Project acronym: AMADEOS
Project full title: Architecture for Multi-criticality Agile Dependable Evolutionary Open System-of-Systems
Grant Agreement no.: 610535

Partners:
[1] Università degli Studi di Firenze
[3] Université Joseph Fourier Grenoble 1
[6] European Network For Cyber Security Cooperatief Ua

D1.1 – SoSS, COMMONALITIES AND REQUIREMENTS
(01.10.2013-30.06.2014)

Revision of the document: 3.11
Responsible partner: UNIFI
Contributing partner(s): all
Reviewing partner(s): TUW, TNL
Document date: 23/06/2014
Dissemination level: RE
## Change Records

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Changes</th>
<th>Authors</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>22/01/2014</td>
<td>Merged all contributions.</td>
<td>TUW</td>
<td>NV</td>
</tr>
<tr>
<td>1.1</td>
<td>23/01/2014</td>
<td>Merged updates from UJF, RES, THALES.</td>
<td>TUW</td>
<td>NV</td>
</tr>
<tr>
<td>1.2</td>
<td>21/02/2014</td>
<td>Added Energy example scenarios MVC and EVC and updated energy dependability Section</td>
<td>RES</td>
<td>NV</td>
</tr>
<tr>
<td>2.0</td>
<td>24/01/2014</td>
<td>Inserted an Executive Summary and a Section describing the Roadmap to the end of deliverable. Version distributed to AB members for comments/ suggestions/ improvements.</td>
<td>UNIFI</td>
<td>NV</td>
</tr>
<tr>
<td>2.1</td>
<td>14/03/2014</td>
<td>Added Section on common findings</td>
<td>RES</td>
<td>NV</td>
</tr>
<tr>
<td>2.2</td>
<td>27/03/2014</td>
<td>Included first draft of Section 1</td>
<td>UJF</td>
<td>NV</td>
</tr>
<tr>
<td>2.3</td>
<td>31/03/2014</td>
<td>Included part on security</td>
<td>ENCS</td>
<td>NV</td>
</tr>
<tr>
<td>3.0</td>
<td>04/04/2014</td>
<td>Creation of unique document, including draft of SoS requirements Section</td>
<td>UNIFI</td>
<td>NV</td>
</tr>
<tr>
<td>3.1</td>
<td>22/04/2014</td>
<td>Collection of comments and contributions</td>
<td>UNIFI, TUW, RES, TNL, ENCS</td>
<td>NV</td>
</tr>
<tr>
<td>3.2</td>
<td>23/04/2014</td>
<td>New version of Section 1, Section 5, Section 7</td>
<td>UJF, TNL, RES</td>
<td>NV</td>
</tr>
<tr>
<td>3.5</td>
<td>07/05/2014</td>
<td>Updated Section 8 after round of comments</td>
<td>UNIFI</td>
<td>NV</td>
</tr>
<tr>
<td>3.6</td>
<td>07/05/2014</td>
<td>Updated Section 1, executive summary, conclusions, layout of table in Sect. 8.12</td>
<td>UNIFI, UJF</td>
<td>NV</td>
</tr>
<tr>
<td>3.7</td>
<td>30/05/2014</td>
<td>Updated version for first round of review</td>
<td>UNIFI, UJF, RES</td>
<td>NV</td>
</tr>
<tr>
<td>3.8</td>
<td>17/06/2014</td>
<td>Review round 1: integrations</td>
<td>UNIFI, TUW, RES, TNL</td>
<td>NV</td>
</tr>
<tr>
<td>3.9</td>
<td>17/06/2014</td>
<td>Updated version for second round of review</td>
<td>UNIFI</td>
<td>NV</td>
</tr>
<tr>
<td>3.10</td>
<td>23/06/2014</td>
<td>Updates after reviewers comments</td>
<td>UNIFI, TNL, TUW</td>
<td>NV</td>
</tr>
<tr>
<td>3.11</td>
<td>23/06/2014</td>
<td>Final version</td>
<td>UNIFI</td>
<td>V</td>
</tr>
</tbody>
</table>

\(^1\) V - Valid, NV - Not Valid, R - Revision
# TABLE OF CONTENTS

1. STATE OF THE ART ANALYSIS .......................................................... 19
   1.1 Introduction .................................................................................... 19
   1.1.1 Definition .................................................................................. 19
   1.1.2 Classification of Systems-of-Systems ......................................... 20
   1.2 Open Literature Survey ................................................................ 20
      1.2.1 Recently Published Analysis of the State of the Art in SoS .......... 20
      1.2.2 Emergence ............................................................................. 20
      1.2.3 Interoperation ........................................................................ 21
      1.2.4 Time ....................................................................................... 21
      1.2.5 Evolution ................................................................................ 22
      1.2.6 Dependability and Security ..................................................... 23
      1.2.7 Governance .......................................................................... 23
      1.2.8 Quality Metrics ..................................................................... 23
   1.3 EU SoS Research Projects .............................................................. 24
      1.3.1 Current projects ...................................................................... 24
      1.3.2 Finished projects .................................................................... 27
   1.4 Concluding Remarks ...................................................................... 31

2 AUTOMOTIVE DOMAIN ................................................................. 32
   2.1 Introduction .................................................................................... 32
   2.2 Problem Area & Example Scenarios ............................................ 33
   2.3 Constraints .................................................................................... 35
   2.4 Architecture & Relied Upon Message Interface (RUMI).................. 36
   2.5 Semantic of Communication ......................................................... 37
   2.6 Dynamicity ................................................................................... 39
   2.7 Evolution ....................................................................................... 40
   2.8 Emergence .................................................................................... 40
   2.9 Governance .................................................................................. 41
   2.10 Handling of Time ......................................................................... 41
   2.11 Dependability .............................................................................. 41
   2.12 Security ....................................................................................... 42
   2.13 Quality Metrics .......................................................................... 44
   2.14 Standards ................................................................................... 45

3 RAILWAY DOMAIN .......................................................................... 47
   3.1 Introduction .................................................................................... 47
   3.2 Constraints .................................................................................... 48
   3.3 Architecture & Relied Upon Message Interface (RUMI) .................. 48
3.3.1 Railway ERTMS ................................................................. 49
3.3.2 Railway SCMT ................................................................. 51
3.3.3 Assets ........................................................................... 51
3.3.4 SoS Design and Boundaries .............................................. 52
3.4 Semantic of Communication ............................................... 55
3.5 Dynamicity ....................................................................... 55
3.6 Evolution .......................................................................... 55
3.7 Emergence ........................................................................ 56
3.8 Governance ....................................................................... 56
3.9 Handling of Time ................................................................. 57
3.10 Dependability and Security ................................................... 57
  3.10.1 Safety and Dependability .................................................. 57
  3.10.2 Security ...................................................................... 58
  3.10.3 Conclusive notes ............................................................. 59
3.11 Quality Metrics .................................................................. 59
3.12 Standards ........................................................................ 60
3.13 Railway Safety and Fatalities ................................................ 61
  3.13.1 Major Incidents ............................................................... 61
3.14 Ontology .......................................................................... 64

4 GLOBAL AUTOMATED TELLER MACHINE NETWORK ..................... 66
4.1 Introduction .................................................................... 66
4.2 Problem Area & Example Scenarios ...................................... 67
  4.2.1 Example Scenario: Cash Withdrawal with ATM/Debit Card .... 67
4.3 Architecture & Relied Upon Message Interface (RUMI) ............ 68
  4.3.1 Relied Upon Message Interfaces (RUMIs) ......................... 69
  4.3.2 Specifications ................................................................. 70
  4.3.3 Classification of Transactions ............................................ 70
4.4 Constraints ....................................................................... 71
4.5 Dynamicity ....................................................................... 71
4.6 Evolution .......................................................................... 71
4.7 Emergence ........................................................................ 73
4.8 Governance ...................................................................... 73
4.9 Handling of Time ............................................................... 73
4.10 Security ........................................................................... 74
4.11 Standards ...................................................................... 75
4.12 Conclusion ...................................................................... 76
4.13 History of the ATM Network .............................................. 76
4.14 Ontology ...................................................................... 77
## 5 CRISIS MANAGEMENT

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>80</td>
</tr>
<tr>
<td>Problem Area &amp; Example Scenarios</td>
<td>81</td>
</tr>
<tr>
<td>Constraints / Quality Metrics</td>
<td>81</td>
</tr>
<tr>
<td>Architecture</td>
<td>82</td>
</tr>
<tr>
<td>Safety Chain</td>
<td>83</td>
</tr>
<tr>
<td>Responsibilities</td>
<td>83</td>
</tr>
<tr>
<td>Governance</td>
<td>85</td>
</tr>
<tr>
<td>During Pre-phase</td>
<td>85</td>
</tr>
<tr>
<td>During repression phase</td>
<td>86</td>
</tr>
<tr>
<td>Governance in Military Action Planning</td>
<td>87</td>
</tr>
<tr>
<td>Governance: Scaling Up and Down</td>
<td>88</td>
</tr>
<tr>
<td>Semantic of Communication</td>
<td>89</td>
</tr>
<tr>
<td>Dynamicity</td>
<td>89</td>
</tr>
<tr>
<td>Problems with Decision Making under Dynamic Circumstances</td>
<td>89</td>
</tr>
<tr>
<td>Recognitional Planning Model (RPM) in Dynamic Decision Making</td>
<td>90</td>
</tr>
<tr>
<td>Evolution</td>
<td>90</td>
</tr>
<tr>
<td>Drivers for Evolution in Crisis Management</td>
<td>91</td>
</tr>
<tr>
<td>Nationwide Initiatives for Evolution in Crisis Management Organisations</td>
<td>92</td>
</tr>
<tr>
<td>Field study: Veiligheidsregio Haaglanden</td>
<td>92</td>
</tr>
<tr>
<td>Emergence</td>
<td>93</td>
</tr>
<tr>
<td>Handling of Time</td>
<td>94</td>
</tr>
<tr>
<td>Dependability</td>
<td>94</td>
</tr>
<tr>
<td>Security</td>
<td>95</td>
</tr>
<tr>
<td>Standards</td>
<td>95</td>
</tr>
<tr>
<td>Recent History of Crisis Management Systems</td>
<td>95</td>
</tr>
<tr>
<td><strong>ENERGY DOMAIN</strong></td>
<td>96</td>
</tr>
<tr>
<td>Introduction</td>
<td>96</td>
</tr>
<tr>
<td>Problem Area &amp; Example Scenarios</td>
<td>96</td>
</tr>
<tr>
<td>Constraints</td>
<td>98</td>
</tr>
<tr>
<td>Architecture &amp; Relied Upon Message Interface (RUMI)</td>
<td>98</td>
</tr>
<tr>
<td>Semantic of Communication</td>
<td>102</td>
</tr>
<tr>
<td>Dynamicity</td>
<td>103</td>
</tr>
<tr>
<td>Evolution</td>
<td>105</td>
</tr>
<tr>
<td>Emergence</td>
<td>105</td>
</tr>
<tr>
<td>Governance</td>
<td>106</td>
</tr>
<tr>
<td>Handling of Time</td>
<td>106</td>
</tr>
<tr>
<td>Dependability</td>
<td>107</td>
</tr>
</tbody>
</table>
### 7 CROSS DOMAIN COMMON FINDINGS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Constraints</td>
</tr>
<tr>
<td>7.2</td>
<td>Architecture &amp; RUMI</td>
</tr>
<tr>
<td>7.3</td>
<td>Semantic of Communication</td>
</tr>
<tr>
<td>7.4</td>
<td>Dynamicity</td>
</tr>
<tr>
<td>7.5</td>
<td>Evolution</td>
</tr>
<tr>
<td>7.6</td>
<td>Emergence</td>
</tr>
<tr>
<td>7.7</td>
<td>Governance</td>
</tr>
<tr>
<td>7.8</td>
<td>Handling of Time</td>
</tr>
<tr>
<td>7.9</td>
<td>Dependability</td>
</tr>
<tr>
<td>7.10</td>
<td>Security</td>
</tr>
<tr>
<td>7.11</td>
<td>Quality Metrics</td>
</tr>
</tbody>
</table>

### 8 REQUIREMENTS DEFINITION

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>An Approach to SoS Requirements</td>
</tr>
<tr>
<td>8.1.1</td>
<td>Structure of Requirements</td>
</tr>
<tr>
<td>8.1.2</td>
<td>Glossary</td>
</tr>
<tr>
<td>8.2</td>
<td>Constraints</td>
</tr>
<tr>
<td>8.2.1</td>
<td>General requirements</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Relations with other viewpoints</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Assets</td>
</tr>
<tr>
<td>8.2.4</td>
<td>Standards</td>
</tr>
<tr>
<td>8.2.5</td>
<td>System life</td>
</tr>
<tr>
<td>8.2.6</td>
<td>Costs and financial aspects</td>
</tr>
<tr>
<td>8.2.7</td>
<td>Governance</td>
</tr>
<tr>
<td>8.3</td>
<td>Architecture &amp; RUMI</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Enterprise view</td>
</tr>
<tr>
<td>8.3.2</td>
<td>Information view</td>
</tr>
<tr>
<td>8.3.3</td>
<td>Engineering view</td>
</tr>
<tr>
<td>8.3.4</td>
<td>Technology view</td>
</tr>
<tr>
<td>8.3.5</td>
<td>Computational view</td>
</tr>
<tr>
<td>8.3.6</td>
<td>Relations with other viewpoints</td>
</tr>
<tr>
<td>8.4</td>
<td>Semantic of Communication</td>
</tr>
<tr>
<td>8.4.1</td>
<td>General requirements</td>
</tr>
<tr>
<td>8.4.2</td>
<td>Viewpoints influencing Semantic of Communication</td>
</tr>
<tr>
<td>8.4.3</td>
<td>Viewpoints influenced by Semantic of Communication</td>
</tr>
<tr>
<td>8.4.4</td>
<td>Relations with other viewpoints</td>
</tr>
</tbody>
</table>
8.5 Dynamicity ................................................................................................................... 133
8.5.1 General requirements .............................................................................................. 133
8.5.2 Viewpoints influencing Dynamicity ....................................................................... 134
8.5.3 Viewpoints influenced by Dynamicity ..................................................................... 134
8.5.4 Relations with other viewpoints .............................................................................. 135
8.6 Evolution........................................................................................................................ 136
8.6.1 General requirements .............................................................................................. 136
8.6.2 Viewpoints influencing Evolution ......................................................................... 136
8.6.3 Viewpoints influenced by Evolution ....................................................................... 136
8.6.4 Relations with other viewpoints .............................................................................. 136
8.7 Emergence ..................................................................................................................... 137
8.7.1 General requirements .............................................................................................. 137
8.7.2 Viewpoints influencing Emergence ........................................................................ 138
8.7.3 Viewpoints influenced by Emergence ..................................................................... 138
8.7.4 Relations with other viewpoints .............................................................................. 138
8.8 Governance..................................................................................................................... 139
8.8.1 General requirements .............................................................................................. 139
8.8.2 Viewpoints influencing Governance ...................................................................... 139
8.8.3 Viewpoints influenced by Governance ................................................................... 140
8.8.4 Relations with other viewpoints .............................................................................. 140
8.9 Handling of Time .......................................................................................................... 140
8.9.1 General requirements .............................................................................................. 140
8.9.2 Viewpoints influencing Handling of Time ............................................................... 141
8.9.3 Viewpoints influenced by Handling of Time ............................................................. 141
8.9.4 Relations with other viewpoints .............................................................................. 142
8.10 Dependability................................................................................................................. 143
8.10.1 General requirements .............................................................................................. 143
8.10.2 Viewpoints influencing Dependability ................................................................... 143
8.10.3 Viewpoints influenced by Dependability ............................................................... 144
8.10.4 Relations with other viewpoints .............................................................................. 144
8.11 Security......................................................................................................................... 145
8.11.1 General requirements .............................................................................................. 145
8.11.2 Viewpoints influencing Security ......................................................................... 146
8.11.3 Viewpoints influenced by Security ....................................................................... 146
8.11.4 Relations with other viewpoints .............................................................................. 146
8.12 Quality Metrics ............................................................................................................. 147
8.12.1 General requirements .............................................................................................. 147
8.12.2 Viewpoints influencing Quality Metrics ..............................................................148
8.12.3 Viewpoints influenced by Quality Metrics ............................................................148
8.12.4 Relations with other viewpoints .........................................................................148
8.13 Viewpoints relations ..............................................................................................149

9 CONCLUSIONS, CHALLENGES AND OPEN ISSUES .............................................185
9.1 SoS Constraints ........................................................................................................185
  9.1.1 Architecture and RUMI ......................................................................................185
  9.1.2 Semantic of Communication ............................................................................185
  9.1.3 Dynamicity .......................................................................................................185
  9.1.4 Evolution ...........................................................................................................186
  9.1.5 Emergence .........................................................................................................186
  9.1.6 Governance .......................................................................................................186
  9.1.7 Handling of Time ..............................................................................................186
  9.1.8 Dependability and Security ............................................................................186
  9.1.9 Quality Metrics ...............................................................................................187
9.2 Viewpoints Relations ..............................................................................................187
9.3 Next Steps and Challenges ......................................................................................187

10 REFERENCES .............................................................................................................189
LIST OF FIGURES

Figure 1 – Automotive context [1] .................................................................................................32
Figure 2 – Example of intra-vehicle systems and its strong control connections [2] .................33
Figure 3 – V2X communication scenario ....................................................................................34
Figure 4 – Road vehicle architecture [4] .....................................................................................36
Figure 5 – ERTMS architecture, high level view [64]. .................................................................50
Figure 6 – SCMT Driver Machine Interface [70]. .......................................................................51
Figure 7 – SoS in the Railway domain .......................................................................................53
Figure 8 – Interactions between signalling CSs. .......................................................................53
Figure 9 – Interactions between ground CSs .............................................................................54
Figure 10 – Fatalities on EU railways per year and victim category (2006-2011) [56]. ..............62
Figure 11 – Automated Teller Machine. .....................................................................................66
Figure 12 – Interconnected CSs and their RUMIs ......................................................................69
Figure 13 – Global EMV Adaption Rate, 2011 ..........................................................................72
Figure 14 – Safety Chain of different phases in crisis management ...........................................83
Figure 15 – The Crisis Management Structure in the Netherlands ..............................................85
Figure 16 – Military action planning process. ............................................................................88
Figure 17 – Conceptual model of Smart Grid [102]. .................................................................99
Figure 18 – Detailed Smart Grid Conceptual Model [102] ...........................................................100
Figure 19 – General Smart Grid Architecture ...........................................................................101
Figure 20 – (a) Formal Hierarchy, (b) Non-formal Hierarchy or Holarchy ................................113
Figure 21 - Constraints viewpoint: relations with other viewpoints. ........................................124
Figure 22 - Architecture & RUMI viewpoint: relations with other viewpoints. .......................131
Figure 23 - Semantic of Communication viewpoint: relations with other viewpoints. ..............133
Figure 24 - Dynamicity viewpoint: relations with other viewpoints .........................................135
Figure 25 - Evolution viewpoint: relations with other viewpoints ............................................137
Figure 26 - Emergence viewpoint: relations with other viewpoints ..........................................139
Figure 27 - Governance viewpoint: relations with other viewpoints ......................................140
Figure 28 - Handling of Time viewpoint: relations with other viewpoints ................................142
Figure 29 - Dependability viewpoint: relations with other viewpoints ....................................145
Figure 30 - Security viewpoint: relations with other viewpoints ..............................................147
Figure 31 - Quality Metrics viewpoint: relations with other viewpoints ....................................149
LIST OF TABLES

Table 1 – Generic security threats and security objectives [6]. .......................................................... 44
Table 2 – Recommended techniques for fault detection and handling. .............................................. 58
Table 3 – SIL-Table [39]. .................................................................................................................. 60
Table 4 – Main railway standards and guidelines. ............................................................................. 61
Table 5 – Railway accidents in the EU in 2010 and relations to SoS. .................................................. 64
Table 6 – Timeline............................................................................................................................ 77
Table 7 – Concepts in the ATM use case. .......................................................................................... 78
Table 8 – Similar transitions from traditional to new SoSs showcase the relevance of findings in the Crisis Management area towards the Smart Grid area. ......................................................... 80
Table 9 – Police responsibilities. ..................................................................................................... 84
Table 10 – Fire department responsibilities. ...................................................................................... 84
Table 11 – Municipality responsibilities............................................................................................ 84
Table 12 – Health services responsibilities. ....................................................................................... 84
Table 13 – GRIP (level of severity of incident) related responsibilities. ........................................... 87
Table 14 – Domains and actors in the Smart Grid Conceptual Model. ............................................... 99
Table 15 – Summary of communication system development activities for electric utilities [109]. . 103
Table 16 - Glossary from D2.1.......................................................................................................... 121
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DES</td>
<td>Triple DES</td>
</tr>
<tr>
<td>ABS</td>
<td>Antilock Braking System</td>
</tr>
<tr>
<td>ACH</td>
<td>Automated Clearing House</td>
</tr>
<tr>
<td>AEB</td>
<td>Advanced Emergency Braking</td>
</tr>
<tr>
<td>ASAI</td>
<td>Average Service Availability Index</td>
</tr>
<tr>
<td>ATM</td>
<td>Automated Teller Machine</td>
</tr>
<tr>
<td>BSI</td>
<td>Bundesamt für Sicherheit in der Informationstechnik - Federal Office for Information Security</td>
</tr>
<tr>
<td>CAIDI</td>
<td>Customer Average Interruption Duration Index</td>
</tr>
<tr>
<td>CAIFI</td>
<td>Customer Average Interruption Frequency Index</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>CENELEC</td>
<td>Comité européen de normalisation en électronique et en électrotechnique</td>
</tr>
<tr>
<td>CIII</td>
<td>Customers Interrupted per Interruption Index</td>
</tr>
<tr>
<td>Cloud4SOA</td>
<td>A Cloud Interoperability Framework and Platform for user-centric, semantically-enhanced service-oriented applications design, deployment and distributed execution</td>
</tr>
<tr>
<td>CM</td>
<td>Crisis Management</td>
</tr>
<tr>
<td>CML</td>
<td>COMPASS Modelling Language</td>
</tr>
<tr>
<td>CM-SoS</td>
<td>Crisis Management System-of-Systems</td>
</tr>
<tr>
<td>COA</td>
<td>Courses of action</td>
</tr>
<tr>
<td>COMPASS</td>
<td>Comprehensive Modelling for Advanced Systems-of-Systems, models for SoS development and maintenance</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Emergent Connectors for Eternal Software Intensive Networked Systems</td>
</tr>
<tr>
<td>CONTRAIL</td>
<td>Open Computing Infrastructures for Elastic Services</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off-the-Shelf</td>
</tr>
<tr>
<td>CPS</td>
<td>Cyber-Physical System</td>
</tr>
<tr>
<td>CPSOS</td>
<td>Cyber-Physical Systems-of-Systems</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CS</td>
<td>Constituent System</td>
</tr>
<tr>
<td>CSO</td>
<td>Charging Station Operator</td>
</tr>
<tr>
<td>CU</td>
<td>Communication Unit</td>
</tr>
<tr>
<td>CYPHERS</td>
<td>Cyber-Physical European Roadmap and Strategy</td>
</tr>
<tr>
<td>DA</td>
<td>Distribution feeder Automation</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DANSE</td>
<td>Designing for Adaptability and Evolution in System-of-Systems Engineering</td>
</tr>
<tr>
<td>DDA</td>
<td>Dependability Deviation Analysis</td>
</tr>
<tr>
<td>DEMANES</td>
<td>Design, Monitoring and Operation of Adaptive Networked Embedded Systems</td>
</tr>
<tr>
<td>DERs</td>
<td>Distributed Energy Resources</td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
</tr>
<tr>
<td>DG</td>
<td>Distributed Generation</td>
</tr>
<tr>
<td>DGA</td>
<td>Distribution Grid Area</td>
</tr>
<tr>
<td>DME</td>
<td>Disturbance Measuring Equipment</td>
</tr>
<tr>
<td>DMI</td>
<td>Driver Machine Interface</td>
</tr>
<tr>
<td>DMS</td>
<td>Distribution Management System</td>
</tr>
<tr>
<td>DMZ</td>
<td>Demilitarized zone</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial-of-Service</td>
</tr>
<tr>
<td>DSO</td>
<td>Distribution System Operator</td>
</tr>
<tr>
<td>DSoS</td>
<td>Dependable Systems-of-Systems</td>
</tr>
<tr>
<td>DTC</td>
<td>Diagnostic Trouble Code</td>
</tr>
<tr>
<td>DYMASOS</td>
<td>Dynamic Management of Physically Coupled Systems-of-Systems</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic Control Unit</td>
</tr>
<tr>
<td>EFT</td>
<td>Electronic Funds Transfer</td>
</tr>
<tr>
<td>EFTPOS</td>
<td>Electronic Funds Transfer Point of Sale</td>
</tr>
<tr>
<td>EMS</td>
<td>Energy Management Systems</td>
</tr>
<tr>
<td>EMV</td>
<td>Europay, Mastercard and Visa</td>
</tr>
<tr>
<td>ENISA</td>
<td>European Union Agency for Network and Information Security</td>
</tr>
<tr>
<td>EPG</td>
<td>Electric Power Grid</td>
</tr>
<tr>
<td>EPP</td>
<td>Encrypted Pin Pad</td>
</tr>
<tr>
<td>ERTMS</td>
<td>European Rail Traffic Management System</td>
</tr>
<tr>
<td>ETCS</td>
<td>European Train Control System</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>EV</td>
<td>Electric Vehicle</td>
</tr>
<tr>
<td>FDIR</td>
<td>Fault Detection, Isolation and service Restoration</td>
</tr>
<tr>
<td>FEDERATES</td>
<td>A Foundation for Engineering Decentralized Self-Adaptive Software Systems</td>
</tr>
<tr>
<td>GAO</td>
<td>Government Accountability Office</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographical Information systems</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>GRIP</td>
<td>Gecoördineerde Regionale Incidentbestrijdings Procedure - Coordinated Regional Incident-management Procedure</td>
</tr>
<tr>
<td>GSM-R</td>
<td>GSM for Railways</td>
</tr>
<tr>
<td>HMI</td>
<td>Human Machine Interface</td>
</tr>
<tr>
<td>HV</td>
<td>High Voltage</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>IaaS</td>
<td>Infrastructure as a Service</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technology</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated development environment</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commission</td>
</tr>
<tr>
<td>IEDs</td>
<td>Intelligent Electronic Devices</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronic Engineers</td>
</tr>
<tr>
<td>IOT6</td>
<td>Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability</td>
</tr>
<tr>
<td>IRIG</td>
<td>Inter-Range Instrumentation Group</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>JCB</td>
<td>Japan Credit Bureau</td>
</tr>
<tr>
<td>KDM</td>
<td>Knowledge discovery metamodel</td>
</tr>
<tr>
<td>LCMS</td>
<td>Landelijk Crisis Management Systeem - Countrywide Crisis Management System</td>
</tr>
<tr>
<td>LEU</td>
<td>Lineside Equipment Unit</td>
</tr>
<tr>
<td>LICB</td>
<td>Lasting Infrastructure Cost Benchmarking</td>
</tr>
<tr>
<td>LIN</td>
<td>Local Interconnect Network</td>
</tr>
<tr>
<td>LOCC</td>
<td>Landelijk Operationeel Coördinatiecentrum - crisis management centre</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>LV</td>
<td>Low Voltage</td>
</tr>
<tr>
<td>LVGC</td>
<td>Low Voltage Grid Controller</td>
</tr>
<tr>
<td>MAIFI</td>
<td>Momentary Average Interruption Frequency Index</td>
</tr>
<tr>
<td>MAP</td>
<td>Manifold Absolute Pressure</td>
</tr>
<tr>
<td>MMI</td>
<td>Man Machine Interface</td>
</tr>
<tr>
<td>MODAClouds</td>
<td>MOdel-Driven Approach for design and execution of applications on multiple Clouds</td>
</tr>
<tr>
<td>MV</td>
<td>Medium Voltage</td>
</tr>
<tr>
<td>MVGC</td>
<td>Medium Voltage Grid Controller</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>NATO</td>
<td>North Atlantic Treaty Organization</td>
</tr>
<tr>
<td>NCAP</td>
<td>New Car Assessment Programme</td>
</tr>
<tr>
<td>NCC</td>
<td>National Crisis Centre</td>
</tr>
<tr>
<td>NDM</td>
<td>Naturalistic decision-making</td>
</tr>
<tr>
<td>NER-CIP</td>
<td>North American Electric Reliability Corporation - Critical Infrastructure Protection</td>
</tr>
<tr>
<td>NFC</td>
<td>Near-Field Communication</td>
</tr>
<tr>
<td>NGO</td>
<td>Non-governmental organisation</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NRZ</td>
<td>Non-return-to-zero</td>
</tr>
<tr>
<td>OBU</td>
<td>On-board unit</td>
</tr>
<tr>
<td>ODMP</td>
<td>Operational Decision-Making Process</td>
</tr>
<tr>
<td>OLTC</td>
<td>On Load Tap Changer</td>
</tr>
<tr>
<td>OT</td>
<td>Operation Technology</td>
</tr>
<tr>
<td>PaaS</td>
<td>Platform as a Service</td>
</tr>
<tr>
<td>PCI</td>
<td>Security Standards Council</td>
</tr>
<tr>
<td>PCI DSS</td>
<td>Payment Card Industry Data Security Standard</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>PMI4Cloud</td>
<td>Platform Independent Model for Cloud Computing</td>
</tr>
<tr>
<td>PMU</td>
<td>Phasor Measurement Units</td>
</tr>
<tr>
<td>POS</td>
<td>Point Of Sale</td>
</tr>
<tr>
<td>PPS</td>
<td>Pulse Per Second</td>
</tr>
<tr>
<td>PSOBU</td>
<td>Public safety OBU</td>
</tr>
<tr>
<td>PTP</td>
<td>Precision Time Protocol</td>
</tr>
<tr>
<td>PV</td>
<td>Photo Voltaic</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RBC</td>
<td>Radio Block Center</td>
</tr>
<tr>
<td>REMICS</td>
<td>Reuse and Migration of legacy applications to Interoperable Cloud Services</td>
</tr>
<tr>
<td>RES</td>
<td>Renewable Energy Sources</td>
</tr>
<tr>
<td>RESERVOIR</td>
<td>Resources and Services Virtualization without Barriers [179]</td>
</tr>
<tr>
<td>Road2SoS</td>
<td>Roadmaps for System-of-Systems engineering</td>
</tr>
<tr>
<td>RPDM</td>
<td>Recognition-primed decision-making</td>
</tr>
<tr>
<td>RSU</td>
<td>Roadside unit</td>
</tr>
<tr>
<td>RTU</td>
<td>Remote Terminal Units</td>
</tr>
<tr>
<td>RUMI</td>
<td>Relied Upon Message Interface</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>Term</td>
<td>Definition</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>SAIDI</td>
<td>System Average Interruption Duration Index</td>
</tr>
<tr>
<td>SAIFI</td>
<td>System Any inter Average Interruption Frequency Index</td>
</tr>
<tr>
<td>SAS</td>
<td>Substation Automation Systems</td>
</tr>
<tr>
<td>SCADA</td>
<td>Supervisory Control and Data Acquisition</td>
</tr>
<tr>
<td>SCMT</td>
<td>Sistema di Controllo della Marcia del Treno</td>
</tr>
<tr>
<td>SECURECHANGE</td>
<td>Security engineering for lifelong evolvable systems</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SLA@SOI</td>
<td>Empowering the service industry with SLA-aware infrastructures</td>
</tr>
<tr>
<td>SOA</td>
<td>Service Oriented Architecture</td>
</tr>
<tr>
<td>SoS</td>
<td>System-of-Systems</td>
</tr>
<tr>
<td>SOSI</td>
<td>System-of-Systems Interoperability</td>
</tr>
<tr>
<td>SoSs</td>
<td>Systems-of-Systems</td>
</tr>
<tr>
<td>SoSE</td>
<td>System-of-Systems Engineering</td>
</tr>
<tr>
<td>SSB</td>
<td>Sottosistema di bordo - Onboard Subsystem</td>
</tr>
<tr>
<td>SST</td>
<td>Sottosistema di terra - Ground Subsystem</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>SysML</td>
<td>Systems Modelling Language</td>
</tr>
<tr>
<td>T-AREA-SOS</td>
<td>Trans-Atlantic Research and Education Agenda on Systems-of-Systems</td>
</tr>
<tr>
<td>TC</td>
<td>Transaction Certificate</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TIU</td>
<td>Train Interface Unit</td>
</tr>
<tr>
<td>TPS</td>
<td>Transactions per Second</td>
</tr>
<tr>
<td>TSO</td>
<td>Transmission System Operator</td>
</tr>
<tr>
<td>TVE</td>
<td>Total Vector Error</td>
</tr>
<tr>
<td>UIC</td>
<td>International Union of Railways</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>UoD</td>
<td>Universe of Discourse</td>
</tr>
<tr>
<td>US</td>
<td>United States</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>UTP</td>
<td>UML Testing Profile</td>
</tr>
<tr>
<td>V2I</td>
<td>Vehicle-to-infrastructure</td>
</tr>
<tr>
<td>V2V</td>
<td>Vehicle-to-vehicle</td>
</tr>
<tr>
<td>V2X</td>
<td>Vehicle to X, X being either infrastructure or vehicle</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular ad hoc network</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>WAMS</td>
<td>Wide-Area Measurement System</td>
</tr>
<tr>
<td>WAVE</td>
<td>Wireless Access in Vehicular Environments</td>
</tr>
</tbody>
</table>
Executive Summary

This document contains the conclusive results of Work Package 1 (Analysis of Existing SoSs) of the AMADEOS project.

The analysis carried out through the whole document is organized in viewpoints, especially starting from Section 2, which are considered the core aspects that will be addressed during the AMADEOS project. These viewpoints are: i) System-of-Systems (SoS) constraints, ii) semantic of communication, iii) architecture and Relied Upon Message Interface (RUMI), iv) dynamicity, v) evolution, vi) emergence, vii) governance, viii) handling of time, ix) dependability, x) security, and although with a minor attention through the document, xi) quality metrics.

The document starts (Section 1) with an analysis of current state of the art on the different SoS properties presenting different points of view, needs and lacks, that will lead the AMADEOS goals. The analysis is also accompanied by a short description of current SoS EU FP projects and also past projects related to AMADEOS. The investigation shows that a vast literature is available, but still leaves unanswered a number of questions on the modelling, development and evolution of time-sensitive and emergent SoSs, which are key AMADEOS challenges.

Subsequently (Sections 2-6), the document explores existing System-of-Systems in different domains. Reference sectors that have been explored are transport (specifically the automotive and the railway domains), the global automated teller machine network, the crisis management domain and finally the energy domain, with a specific focus on Smart Grids. Each domain is analyzed considering the specific viewpoints previously reported and including, whenever considered relevant, historical notions on the domain, a list of major incidents at SoS-level, and the available ontology. The exploration of the domains, strongly supported by the industrial partners of the AMADEOS consortium, allowed understanding the application and the diffusion of SoS concepts in current practice and allowed defining the scale and complexity of an SoS-oriented approach to system design.

To perform the synthesis of the findings in terms of common cross-domain approaches and criticalities, the document identifies commonalities between the different domains (Section 7). Common core aspects/characteristics and accepted cross-domains approaches that can be exploited in the AMADEOS framework are identified.

In Section 8 the synthesis of requirements for SoS in general is presented. They will be used as a reference in all the remaining AMADEOS activities. The requirements framework is generic and can be applied to describe any SoS as well as to guide the SoS design. In other words, our SoS requirements can be seen as SoS elements, peculiarities, or characteristics that should be identified when describing an SoS. The definition of the requirements is based on different perspectives, guided by the viewpoints. Requirements regarding the different viewpoints often present overlaps, thus requiring a cross-viewpoint traceability matrix which relates the requirements to one another. The objective of such matrix is to make explicit the connections and dependencies between viewpoints.

Finally, conclusions and open issues are presented in Section 9.

This document will be a fundamental input for the successive activities of the AMADEOS project, in particular for the definition of the conceptual model in WP2 (Conceptual model of an SoS) and the definition of the AMADEOS architecture in WP3 (Design methodology and design guidelines).
READING GUIDELINES

Given the length of the document, we present the following note to guide an interested reader to select the appropriate portions of the document.

A reader interested in a very quick overview to get a grasp of the main concepts expressed in the document should concentrate on reading Section 1.1 (Introduction) and Section 9 (Conclusions, challenges and open issues).

A reader interested in the main findings instead, is required to go into deeper details, focusing on reading the entire Section 1 (Purpose of SoS analysis), Section 7 (Cross domain common findings), Section 8 (Requirements definition) and Section 9 (Conclusions, challenges and open issues).

A reader interested in detailed explorations of the SoS domains should also consider reading Sections 2 to 6, in addition to the above-mentioned Sections.
1. STATE OF THE ART ANALYSIS

This Section surveys the open literature with respect to publications that deal with those properties of Systems-of-Systems (SoS) that are considered in the AMADEOS project. We include also a review of ongoing and completed European research projects in the SoS domain.

1.1 INTRODUCTION

System-of-Systems (SoS) forms an area of computer science whose relevance has recently increased tremendously, since many existing systems are getting interconnected (e.g., by the Internet) in order to realize new services and improve economic processes. An SoS comes about by the integration of constituent systems (CS) e.g., existing legacy systems that might belong to different organizations and newly developed CSs. An SoS is composed of hardware/software systems, communication systems, physical machines and humans. This first Section discusses the different definitions and classifications of SoSs.

1.1.1 Definition

In the literature ([103], page 2) we find several different definitions of a System-of-Systems (SoS). However, the following definition seems to be accepted by a majority of the SoS community [103]:

“An SoS is an integration of a finite number of constituent systems which are independent and operable, and which are networked together for a period of time to achieve a certain higher goal.”

This definition is in concordance with five key characteristics of an SoS postulated by Maier [189], a majority of which most SoSs exhibit. These characteristics are:

- **Operational Independence**: If the SoS is disassembled into its component systems, the component systems must be able to usefully operate independently. That is, the components fulfil customer-operator purposes on their own.
- **Managerial Independence**: The component systems not only can operate independently, they do operate independently. The component systems are separately acquired and integrated but maintain a continuing operational existence independent of the SoS.
- **Geographical Distribution**: The component systems are distributed spatially (and possibly temporally); therefore focus is on transport of information between the component systems.
- **Emergent Behavior**: The purposes expressed by the SoS emerge only through the collective actions of the systems participants.
- **Evolutionary/Adaptive Development**: The SoS is not designed and built as a single entity, but evolves over time. Component systems may join and leave the SoS over time.”

These characteristics are not the only ones that are typical of an SoS. SoSs usually exhibit other features like interdisciplinary, heterogeneity of CS and networks of systems [129], in addition to the mentioned characteristics of Maier - although not all of these characteristics have to be present in any single SoS. Firesmith [101] states that SoSs are the systems with the highest size, complexity, evolution, emergence, and variability.

System-of-Systems Engineering (SoSE), as defined by Keating et al. [130], is the discipline that emerged with the goal of integrating existing legacy systems, or as Jamshidi points out: “the opportunity for the SE community to define the complex systems of the 21st century” [103].

In the interesting article Systems-of-Systems - the Meaning of, Boardman and Sauser [219] analyze the following differentiating characteristics that distinguish an SoS from a conventional system:

- Autonomy of the CSs,
- Belonging of the CS,
1.1.2 **Classification of Systems-of-Systems**

Dahmann and Baldwin [193] introduce the following four categories of System-of-Systems that are based on Maier’s classification and widely cited in the SoS literature:

I. **Directed**: The SoS has clear objectives and is built and managed by a central authority to attain the stated goal. The constituent systems (CS) that form the SoS may operate independently, but they are subordinated to the central purpose.

II. **Acknowledged**: SoS with a recognized objective but the CSs may retain their own management, funding and authority in parallel with the SoS.

III. **Collaborative**: There is no clear objective, management, authority or funding at the SoS level. Systems voluntarily act together to address shared or common interests.

IV. **Virtual**: There is no clear objective, management, authority or funding at the SoS level. Systems don’t even know about each other.

The degree of control and coordinated management of the CSs that form the SoS is relatively tight in a directed SoS, but it gets looser as we move to the acknowledged, collaborative and finally virtual category.

1.2 **Open Literature Survey**

In the following literature survey we focus on existing publications on topics of particular interest to the AMADEOS project. For each topic, we introduce a general view on how that property is understood and what are the needs and lacks that are identified in the literature.

1.2.1 **Recently Published Analysis of the State of the Art in SoS**

In the recent past, a number of excellent surveys on the general State of the Art on SoSs have been published. We do not intend to repeat this work here and point the interested reader to the T-AREA State of the Art Analysis [203] that forms an important input to the SoS Strategic Research Agenda [198]. A reader who is interested in the State of Art in SoS Software Architectures is referred to [167]. There is a focused research effort on Systems-of-Systems at the Software Engineering Institute at Carnegie Mellon University (www.sei.cmu.edu/sos) with pointers to many relevant publications.

We could not find a State-of-the Art Survey that focuses on the topics that are at the core of the AMADEOS project, such as time in an SoS or short term and long term evolution of an SoS.

1.2.2 **Emergence**

Emergence is a term that denotes the appearance of novel phenomena at the system level that are not present at the level of subsystems. The Strategic Engineering Research Agenda of the EU for System-of-Systems [198] states on page 14: **Networking individual systems together to realize a higher goal that none of the individual systems can achieve in isolation defines emergence as a fundamental property of SoSs.**

Emergence can be regarded as an intriguing part-whole relation that considers how the properties, the organization and the interaction of the parts lead to novel phenomena at the level of a whole. Emergence can be seen as positive, negative or neutral and can be part of the global objectives. The interactions of the autonomous entities can result in cascade effects, epidemics or cascading cycles [194]. Several authors have pointed out the need of emergence for the design of adaptable and evolvable systems [201], [141], [142]. Hsu [139] discusses how emergence can contribute to the behavior of the SoS.
In philosophy, the questions of how the diversity of the world emerges out of simple physical building blocks has been a topic of inquiry since the time of the ancient Greeks (we refer to the abundant literature about emergence in philosophy such as [220] or the books by [221], [222], [134]). Computer scientists got interested in the topic of emergence when it was realized that some striking phenomena that are observed at the system level of complex systems cannot be explained by looking at the system’s components in isolation.

A well-publicized example of emergent misbehavior in System-of-Systems is the flash crash of the stock market on May 6, 2010 [133].

John Holland, a computer scientist working in the domain of complex systems, remarks: Despite its ubiquity and importance, emergence is an enigmatic and recondite topic, more wondered at than analyzed [...] It is unlikely that a topic as complicated as emergence will submit meekly to a concise definition and I have no such definition to offer ([137] p.3). In his paper Emergent (Mis)behavior vs. Complex Software Systems J. Mogul [138] describes emergent misbehavior in a number of computer systems, discusses how emergence can manifest itself, and proposes a research agenda for studying the phenomena of emergence in complex computer systems.


Parunak and VanderBrok [132] and Huberman and Hogg [148] observed that variable temporal delays play a key role in the generation of emergent misbehavior in an SoS. In the interesting paper A Turing Test for Emergence, Boschetti and Gray [147] elaborate on the limits of insights gained from computer simulations when modeling emergent phenomena in natural systems.

The study of emergence in system-of-systems is at the core of the AMADEOS project. We plan to make substantial contributions to the understanding of emergence in system-of-systems.

1.2.3 Interoperaion

In the context of an SoS, interoperability is defined as the ability of a set constituent systems (CS) to communicate in order to share information and operate on that information according to previously agreed operational semantics [192].

Over the past decades, the focus of the discussion about interoperability has shifted from the technological level (how to transport a bit-string from one system to another system) via the syntactic level (how to structure a bit string into syntactic chunks) to the semantic level (what is the meaning of a syntactic chunk). While the progress in the solution of the problems at the technological and syntactic level has been excellent (see, e.g., the information access via the Internet), the problem of interoperability at the semantic level remains a serious research challenge (see the survey by Noy [202] and the survey by Wache [209] on information interoperability). The topic of System-of-Systems interoperability (SOSI) in the context of military systems is discussed in [210]. A conceptual model for the information transfer is systems-of-systems has been published in [211].

In AMADEOS we plan to contribute to the investigation of the semantic interoperability problem in System-of-Systems.

1.2.4 Time

As mentioned before, an SoS is composed of hardware/software systems, communication systems, physical machines and humans, i.e., CSs are often cyber-physical systems (CPSs). In many CPSs a computational result must be produced before a physical deadline, otherwise the system is deemed to have failed. Whenever a physical machine is controlled by a computer system, two different notions of time must be carefully distinguished: (i) the physical time that is an
independent variable in the laws of physics that describe the behavior of the physical machine as a function of time, and (ii) the *logical time* that characterizes some aspects of the behavior of the computer system that unfolds as a consequence of the execution of the stored programs. Lee [212] argues that the prevailing abstractions in computer science focus on *logical time* and have ignored the progression of *physical time*. Neglecting the temporal dimension of computations causes many temporal coordination problems in cyber-physical systems.

Whereas in a monolithic computer a *common clock for time-stamping events* can be derived directly from the signals of the central physical oscillator, no such *common clock* exists in an SoS. Each autonomous CS has its own oscillator that swings freely and is uncoordinated with respect to the oscillation of the oscillator in any other autonomous CS. It is thus not possible to measure the duration between events that occur in the physical environment of different CSs if no global notion of time of adequate precision is shared by all CSs of the SoS. The establishment of a global time based in a distributed system is extensively discussed in [146].

To solve the temporal coordination problems in large SoSs involved in the control of the infrastructure, the global *GPS time* has been introduced at the level of the application. For example, in the Smart Grid PMUs (phase measurement units), [213] uses the GPS time to timestamp the collected data; in telecommunication the GPS time is used to synchronize the transmission signals; in air traffic control the GPS system is used to replace the ILS (Instrument Landing System). The pervasive use of the GPS time in the control of the infrastructure leads to a dependence of the infrastructure service on a single point of failure, the failure of the GPS system. In a recent US GAO report [214], these concerns have been articulated.

Given that a global SoS time is available, this global physical time can be used to simplify the solution of many other temporal coordination problems in an SoS, such as [211]:

- Enable the interpretation of timestamps in the different nodes of the system.
- Limit the validity of real-time control data.
- Synchronize Input Actions across CSs.
- Synchronize Output Actions across CSs.
- Provide conflict-free Resource Allocation.
- Perform prompt Error Detection.
- Strengthen Security Protocols.

It is one of the main objectives of the AMADEOS project to bring time-awareness into the design of Systems-of-Systems and to investigate how the availability of such a global time can be used in the SoS design process.

### 1.2.5 Evolution

In the computing literature the term *evolution* refers to the necessary modifications that are required over the lifetime of a computer system, (specifically software system) in order to maintain or increase the utility of the system to its users. These modifications are needed because the external world is dynamic, requirements change, new hardware devices appear on the market, new functionality is desired and latent design errors must be corrected.

Already in 1976 Belady and Lehman [215] observed that the structure of a large centrally controlled software system, such as the IBM OS 360 Operating System, degrades when programming errors are fixed and new functionality is added. They formulated a set of laws describing the process of evolution [208]. In an SoS context, where existing legacy systems must be frequently modified to keep them relevant to their organization, the evolution problem is magnified, since it is hardly possible to coordinate the individual changes on an SoS wide basis.

In 2010, Murer et al. [153] published a seminal book on *Managed Evolution* (with extensive references) that outlines how the evolution problem has been solved in a large banking system. According to Murer et al. [153], precise internal interface specifications (which in AMADEOS are called Relied Upon Message Interfaces (RUMI)) play a key role in managed evolution. Scot et al. [136] postulates that an evolutionary SoS architecture will need standard interfaces, interface
layers and continual system verification and validation process, in order to minimize the effort when modifying constituent systems.

The topic of evolution of an SoS and the related topics of interface placement, interface definition and interface changes will play a dominant role in AMADEOS.

### 1.2.6 Dependability and Security

Whereas the occurrence of a fault is considered an *exceptional event* in classical system designs, in an SoS design the occurrence of a fault must be regarded as a *normal event*, since any CS may decide to leave the SoS for whatever reason (e.g., local maintenance). Borkar [216] states that the next generation of Integrated Circuits (ICs) is likely to exhibit more transient hardware faults and that the handling of these faults will have to be done at the system level. It is also highly improbable that a large system, such as an SoS, if free from latent design faults. This insight—*that faults are normal*—has stern consequences for the system design.

The methods and techniques that have been developed in the domain of fault-tolerant computing are thus highly relevant for the SoS designer. These methods are discussed in the yearly IEEE flagship conference DSN (Dependable Systems and Networks) and published, in addition to the proceedings of the DSN conference in the IEEE Transactions on *Secure and Dependable Computing*. The standard terminology used by this community is contained in [57]. A conceptual model for a dependable SoS has been published in [196].

System-of-Systems pose a formidable security challenge. As pointed out in [217], the primary security architectures of today rely on perimeter control and centralized security models. Since in an SoS the perimeters are eroding and no central security control is feasible, new security mechanisms have to be developed for the SoS domain [218].

The development of design principles for dependable and secure SoSs will be a major effort in the AMADEOS project.

### 1.2.7 Governance

The underlying purpose of *System-of-Systems governance* is to ensure that the interoperation between the component systems will achieve the goals of the SoS, i.e, the enterprises that are part of the SoS [190].

Several governance strategies have been already proposed for IT Systems [143], [144], [145] and for SoSs [200]. Authors in that work propose several characteristics that need to be monitored and also state the need for additional characteristics for system-of-systems governance. Among the proposed characteristics we can find: collaboration and authority, motivation and accountability, multiple models, expectation of evolution, highly fluid processes and minimal centrality.

Based on the AMADEOS design methodology, we will elaborate on an appropriate governance strategy such that the stated goals of an SoS can be achieved.

### 1.2.8 Quality Metrics

Quality metrics are set of qualitative and quantitative metrics to decide on the efficiency and effectiveness of the SoS. They were first proposed as a measure for software quality [20] and, together with Quality of Service, has been a major topic of interest in past Grid Computing research [161], [162] and is nowadays a major topic of interest in Cloud Computing [163], [205].

AMADEOS will define quality attributes and metrics for SoSs. Note that for SoSs as a service oriented architecture (SOA), not much attention has been paid in the recent past to the impact of quality attributes, therefore a more thorough examination of the relationship between SOA and quality attributes is needed [206].
1.3 **EU SoS RESEARCH PROJECTS**

1.3.1 **Current projects**

- **CPSOS: Cyber-Physical Systems-of-Systems** [152]. CPSOS is a Support Action started in October 2013 and lasting for 30 months. The objective is to develop a roadmap on research and innovation in engineering and management of cyber-physical Systems-of-Systems. This project will provide an exchange platform for the communities and projects related to Systems-of-Systems. That platform focuses on the challenges that engineering and the operation of technical systems pose and in which computing and communication systems interact with large complex physical systems. The different approaches pursued by different communities in theory and applications to the design, analysis and control of SoSs will be spanned in CPSoS, together with a deep exam of the application-specific issues that capture cross-industry and cross-application findings, enabling cross-domain developments and opening new paths for the analysis, design and control of Systems-of-Systems. Concretely, as stated in the project objectives, the project proposes the following outcomes:
  
  - "Identification of the industrial & societal needs and of the state of the art of tools and theories for cyber-physical SoS.
  - Identification of synergies, open issues and promising trans-disciplinary research directions, which will go into work programs of the EU, national funding programs, etc.
  - Building up a network of key researchers and application domain experts in the area.
  - Stimulate the take-up of research by industry (vendors and end-users).
  - Raise public awareness of the impact of research on Systems-of-Systems engineering, analysis and control (Public events will be organised in conjunction with some of the Working Group meetings)."

- **CYPHERS: Cyber-Physical European Roadmap and Strategy** [151]. CYPHERS is a 18-month support action started in July 2013. CYPHERS aims at developing an integrated cyber-physical roadmap and strategy research and innovation agenda for Europe to ensure its competitiveness. Among the different objectives of the project, we can find the following set of actions:
  
  - Elaborating a systematic classification of the domain
  - Market modeling and identification of relevant players.
  - Developing an analysis and assessment of core technologies and current state in science and technology (in the CPS field).
  - Analyzing future implications of CPS regarding technology, economy and society.
  - Identifying challenges, bottlenecks and risks for the research and development in CPS.

  Comprehensive recommendations will also address: identification and prioritization of research areas, measures for both horizontal and vertical cooperation, funding policies, and finally, training and elaboration of standards.

- **DYMASOS: Dynamic Management of Physically Coupled Systems-of-Systems** [150]. DYMASOS is a three-year project started in October 2013. Its main objective is to develop new methods for the distributed management of large physically connected systems having a distributed autonomous management and global coordination. The use cases considered in the project belong to electrical grid management and control, in which we can find charging electric vehicles or industrial production management. Large-scale simulations will
validate the properties of the distributed management and control techniques. As described in the project overview, the project will exhibit the following outcomes:

- “Innovation in distributed management methods for complex interconnected Systems-of-Systems;
- Progress in methods for the rigorous analysis and validation of Systems-of-Systems;
- Demonstration of advanced methods in realistic large-scale simulations of real use cases of Systems-of-Systems;
- Reduction of the carbon footprint and of the resource consumption of industry and of electric power generation and distribution, including charging of electric vehicles;
- Identification of technology gaps in advanced management and coordination methods for SoS and their implementation.”

- Local4Global: Systems-of-Systems that act locally for optimizing globally [149]. This project started in October 2013 and will last for three years. The main objectives of Local4Global are to develop, test and evaluate in a real-life Traffic System-of-Systems, a generic, integrated and fully functional methodology exposing the following characteristics:
  - Full autonomy of constituent systems that optimize their local environment to emerge the performance at the global level.
  - The methodology will provide the ability to exploit the abilities of each constituent system enriching the latter with learning, evolving and self-organizing capabilities.
  - The infrastructure to be elaborated will provide all constituent systems information collected from all the traffic System-of-Systems in a non “expensive” way.

The approach this project will follow is to provide situation awareness mechanisms to constituent systems in conjunction with a distributed optimizer responsible for deciding the optimal actions of the constituent system. That is, the traffic System-of-Systems will be embedded with the ability of identifying and predicting emerging and/or evolutionary characteristics globally and control the behavior of the traffic System-of-Systems inside the specification. As a main result, the project will elaborate a final product which will be fully functional and ready to use. That product will be tested in two different real-life traffic Systems-of-Systems.

- DANSE: Designing for Adaptability and evolution in System-of-Systems Engineering [172]. DANSE is a running three-year project expected to finish at the end of 2014. This project aims to bring evolution and adaptation to an iterative SoS life. This project raises the difficult technical, managerial and political challenges that organizations face when trying to unify large infrastructures. Among the different goals of the project, we can find:
  - Develop a technical approach for SoS engineering.
  - Define a methodology to support an evolutionary, adaptive and iterative SoS life-cycle.
  - Semantic contracts for SoS interoperations.
  - Continuous and non-disruptive integration of system components.
  - Develop tools to support the analysis, simulation and optimization of Systems-of-Systems.

Several real-life use cases have been studied, including emergency services or traffic management in collaboration with industrial representatives ranging from aerospace, land, to automotive systems. Finally, the expected impacts of the project are:

- Providing methods and tools tailored to SoS enabling the improvement of European industrial competitiveness.
Building new markets for the SoS development and management tools.
Taking European industry beyond the current state-of-the-art in the development and management of Systems-of-Systems.
Support the cooperation with international experts and working groups.

DEMANCES: Design, Monitoring and Operation of Adaptive Networked Embedded Systems [178]. The DEMANCES project started in May 2012 and is expected to end in April 2015. The main project goals are to provide component-based methods, framework and tools that can be used to develop runtime adaptive systems. That will make adaptive systems more flexible and capable of evolving and reacting to changes inside them or in their environment or even changes in the requirements. Recent advances from systems and control engineering are combined inside this project. Concretely, the objectives of this project are listed as follows:

Providing a model of the architecture and operation of adaptive systems.
Development of environments for simulation and evaluation, in order to support the design process of adaptive systems.
Integration of services in the execution environment offering self-organization, self-reconfiguration and self-optimization. Those services will support the implementation of adaptive systems.
Means to validate and test adaptive systems.
Propose runtime monitoring for internal and external operational conditions of such systems.

Three use cases of smart adaptive systems (urban transport, airport and home) will be used to validate and demonstrate the concept, methodology and tools proposed by DEMANCES.

COMPASS: Comprehensive Modeling for Advanced Systems-of-Systems [174]. This three-year project will end in September 2014. The objective of COMPASS is to integrate engineering notations, methods and tools in the modeling and analysis process of SoS. The core is a new modeling language called CML (COMPASS Modeling Language) [154]. With CML, developers can choose among several levels of description, like a graphical architectural view using SysML, which is easy to understand for stakeholders. SysML has been linked to SoS specific properties allowing to describe assumptions and guarantees of Constituent Systems. Moreover, it can be processed by static analysis tools like theorem provers and model checkers. In this way, inconsistencies and potential deviations from the contract can be detected automatically. The semantics in CML are given in UTP (Unifying Theories of Programming) which enables consistency between models and analyses. COMPASS has applied CML to several use cases like Accident Response (coordination of diverse healthcare services), or an Audio/Video/Home Automation Ecosystem (aggregating AV and home automation in multiples locations like house, car or the office).

FEDERATES: A Foundation for Engineering Decentralized Self-Adaptive Software Systems [182]. This project started in April 2012 and will end in March 2016. This project aims at studying and developing scientific foundations for engineering self-adaptive systems moving from centralized or hierarchical control of adaptation to a decentralized environment. As stated in the project proposal, the project will study and develop:

“A set of formally founded high-level design models for decentralized self-adaptation.
A set of supporting coordination mechanisms.
An approach for formal analysis of decentralized self-adaptation that combines static with runtime verification.”
The goals will be based on robustness and openness of modern software systems and applying them to the domain of robotics.

- **IOT6**: Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability [183]. This three-year project started in October 2011 and will run until September 2014. The main objective of the project is to study the potential of IPv6 to support the future Internet of Things. It will develop a IPv6-based Service Oriented Architecture. Is architecture will allow to achieve interoperability, mobility, cloud integration and intelligence distribution for its heterogeneous components, applications and services. Finally the project explores novel forms of interactions including multi-protocol integration and interoperability with heterogeneous devices, mobile and cellular networks, cloud services at the SaaS level, among others. Several outcomes in this project are expected: recommendations of IPv6 features for exploitation of the Internet of Things, the well defined and highly scalable IPv6 service oriented architecture (previously mentioned), and validation and testing the architecture in a real environments with a smart and green building perspective.

- **MODAClouds**: MOdel-Driven Approach for design and execution of applications on multiple Clouds [184]. MODAClouds is a three-year project ending in 2015. The main objective of this project is to provide methods, a system for decision support (DSS) and an open source IDE and run-time environment for the design, prototyping, automatic code generation, and automatic deployment of applications on multi-Clouds, guaranteeing a certain degree of quality of service. The research will be carried to attain the following sub-objectives:
  - **Deliver an advanced software engineering model-based approach and an IDE to support systems developers in building and deploying applications, with related data, to multi-Clouds spanning across the full Cloud stack (IaaS/PaaS/SaaS).**
  - **Define quality measures, monitoring mechanisms, prediction models, and adaptive policies to provide quality assurance in Clouds and multi-Clouds.**
  - **Provide support to costs and risks assessment to increase trust in Clouds.**
  - **Develop an integration framework between design tools and run-time.**
  - **Create relevant and complex case studies for the entire risks assessment and software engineering methodologies based on practical industrial scenarios.**
  - **Analyze and validate project outcomes through case studies.**
  - **Ensure distribution of project results via (i) dissemination activities on relevant publication channels, (ii) training and (iii) standardization of results.**
  - **Provide the MODACLOUDS results as an open source solution supporting the full lifecycle of applications on multiple Clouds.**

### 1.3.2 Finished projects

- **CONTRAIL**: Open Computing Infrastructures for Elastic Services [177]. The CONTRAIL project ran from October 2010 till January 2014. The focus of this project is the development of a cloud IaaS and PaaS infrastructure based on federated clouds. The federated infrastructure is able to provide service guarantees based on SLAs. The goal of CONTRAIL was the integration of an open-source operating system for the management of autonomous resources in the context of infrastructure cloud services (IaaS) with high level platform services (PaaS). Among the achievements in the project, the main one is an open source software stack that includes an exhaustive set of systems, runtime services and high level services. All the elements expose a standardized interface allowing federated clouds to cooperate and share their different resources. This has been translated into an integrated approach to virtualization, offering infrastructure and platform services for federated clouds. CONTRAIL had to address key technological challenges in both commercial and academic existing Clouds. It had to face the absence of standardized
interfaces, combined with the oversized limitations imposed to customers by the cloud providers and the lack of Quality of Service guarantees in terms of the performance and dependability of the resources offered by Cloud providers.

- **Road2SoS: Roadmaps for System-of-Systems engineering** [173]. Road2SoS was a three-year project that ended in 2013. The objectives of the project were to define action plans for the research in System-of-Systems Engineering in different fields. Four roadmaps were proposed inside the following application domains:
  1. Integrated multi-site industrial production.
  2. Multi-modal traffic control.
  3. Emergency and crisis management.
  4. Distributed energy generation and Smart Grids.

- **T-AREA-SOS: Trans-Atlantic Research and Education Agenda on Systems-of-Systems** [175]. This project was a support action that ran from September 2011 to August 2013. It aimed at creating a strategic research agenda [198] in SoS engineering enabling the development of concrete research initiatives through which both the EU and the US will collaborate. This collaboration could be transformed into the enhancement of research programs and to prepare future ones. The proposed agenda has been formulated comprising twelve activities ranging from Theoretical foundations, Emergence, Human Aspects, Energy Efficient, Prototyping and Evaluation of SoS. Specific research questions were identified for each theme. Finally, the agenda concluded with seven recommendations:
  1. Use the identified themes for call for proposals in Horizon 2020.
  2. Use the identified themes as the basis for collaborations between EU and US.
  3. Use the identified themes as the basis EU and national founded projects.
  4. Research on SoSE should be carried out by teams involving people from academia and industry.
  5. Research on SoS and SoSE should be enhanced by case studies.
  6. Maintain a roadmap with respect to EU research [173].
  7. Appropriate funding for training should be focus on skills development in SoS and SoSE.

- **CONNECT: Emergent Connectors for Eternal Software Intensive Networked Systems** [176]. CONNECT was a 42-month project started in February 2009. This project had as goal to enable continuous composition of networked systems to respond to the evolution of functionalities provided to and required from the networked environment. The solution has been made by means of synthesizing on the fly the connectors (emergent middleware) via which networked systems communicate. The project experimented with scenarios form the area of Global Monitoring for Environment and Security (GMES). This area is composed of a mixed set of legacy and new systems, focuses on facilitating the collaboration of international communities and most of their components have to be composed in runtime.

- **RESERVOIR: Resources and Services Virtualization without Barriers** [179]. This three-year project ended in March 2011 and aimed at defining an infrastructure for effective delivery of services as utilities. This service-oriented infrastructure builds on open standards and new technologies, and provides a framework for delivering elastic and energy efficient services as utilities by means of controlling resources on demand. Inside this project several spin-out technologies where created and/or enhanced:
1. Claudia from Telefonica R&D [155]: The Claudia platform is a toolkit for controlling elasticity and scalability of an IaaS, and provisioning of PaaS and SaaS, regarding elasticity rules, business rules, and SLAs.

2. OpenNebula from Complutense University of Madrid [156]: OpenNebula is a framework to build private, public and hybrid cloud infrastructures. During the project, OpenNebula was enhanced with advanced scheduling policies, support for elastic multi-tier services, support for Cloud federation, among others features.

3. Kernel based Virtual Machine (KVM) from IBM and SAP [157]: KVM is a complete virtualization solution for Linux that can be installed as a kernel module. It was developed to support several virtual machines at a time, both Linux and Windows images, with private virtualized hardware. The KVM module forms part of the official Linux kernel since its version 2.6.20.

• REMICS: Reuse and Migration of legacy applications to Interoperable Cloud Services [180]. REMICS was a three-year project that ended in August 2013. The main goal of this project was to specify, develop and evaluate a tool-supported model-driven methodology to modernize legacy applications into the form of interoperable cloud service or platforms. The project has driven the standardization works in the Object Management Group (OMG) [158]:

1. PMI4Cloud: Platform Independent Model for Cloud Computing. It is an extension to the OMG SoaML (Service-oriented architecture Modeling Language) [159]. PIM4CLOUD details viewpoint models adapted to the paradigm of cloud computing from the different perspectives of the stakeholder.

2. KDM: Knowledge discovery metamodel. This standard includes how to represent in a general manner how to store the information obtained from legacy artifacts. The project proposed extensions to KDM to integrate the former knowledge with business models.

3. PIM4ServiceInteroperability: This standard focuses on solving interoperability issues in service-oriented architectures like discrepancy in the format of XML files. PIM4ServiceInteroperability provides the support for representing data model transformations or mediators

4. PIM4Models@Runtime: concentrates on the adaptation at runtime and the management of the resulting system.

5. UTP (UML Testing Profile). This standard from the OMG is a specification defining test-related artefacts. The REMICS project contributed to its revision to make it more usable and also to cover more testing concepts.

6. Software Metrics Metamodel: involves the definition of metrics and metric computation results used during the migration process.

7. ESSENCE: it is the standard for a domain-specific language and a kernel of essentials for software engineering. REMICS supported this standardisation.

• Cloud4SOA: A Cloud Interoperability Framework and Platform for user-centric, semantically-enhanced service-oriented applications design, deployment and distributed execution [181]. Cloud4SOA was a three-year project running from September 2010 until August 2013. The objective of this project was to provide an open framework for PaaS developers and providers, with emphasis in semantics and user-centric design and development principles. Among the project results we can find:

1. The Cloud4SOA Cloud Semantic Interoperability Framework: This framework provides, among other features, a typology identifying semantic interoperability conflicts in the Cloud and a set of good practices and recommendations for
transparent data and application portability and semantic interoperability on different cloud platforms.

2. The Cloud4SOA Reference Architecture: An open and generic architecture following a scalable, reusable and transferable approach. This approach facilitates the migration, management and monitoring of applications by interconnecting Clouds in a semantic manner.

3. The Cloud4SOA Models for Resources and Services: This model provides necessary semantics for annotating resources and services in Cloud Computing.

4. The Cloud4SOA System: This system is the implementation of the Cloud4SOA Reference Architecture.

- **SECURECHANGE.** Security engineering for lifelong evolvable systems [185]. This project ran from 2009 until 2012. The main objective of this project was to develop a set of techniques and tools that ensure compliance to evolving security, privacy and dependability requirements for long-running software systems with evolving properties. The project has been validated on the basis of real industrial use cases from several domains ranging from Air Traffic Management, Smart Cards Software Evolution, to Home Appliances. Inside the project a wide range of new tools have been developed and others have been enhanced, like the Move Tool, the SecMer tool (and the underlying engines EMF-IncQuery and OpenArgue), the Rinforzando tool, the CARISMA tool, among others. All the tools are available through the project Web page. Some of the tools have been put into a production environment. A spin-off company was founded named QE LaB Business Services [160]. This company provides services for collaborative IT-Systems.

- **SLA@SOI: Empowering the service industry with SLA-aware infrastructures [186].** This three-year project ended in 2011. The objective of SLA@SOI was to deliver and showcase a new open Service Level Agreement (SLA) Management Framework offering holistic support for service level objectives. The framework enables an open, dynamic, and SLA-aware market for service providers. Three key features were taken into consideration:
  1. Prediction and dependability: providing means to predict and enforce the quality of the service at runtime.
  2. Transparent SLA Management: defining the conditions under which provided/consumed service can be transparently managed.
  3. Automation of SLA negotiation and provisioning, delivery and monitoring of services.

Regarding the achievements of the project we can find:
- A public reference SLA management architecture: that integrates SLA in the design, offering, negotiation, providing, monitoring and adjustment of a service.
- A flexible and extensible model for SLA: this model provides a language to describe SLAs independently from any technology.
- A reference SLA management framework: the implementation of the proposed reference architecture.
- A reference demonstrator in a reference retail scenario: This demonstrator includes main aspects of service/SLA design, negotiation, provisioning, monitoring, and adjustment.
- Industrial use cases proving the business benefits of applying the SLA@SOI framework: an ERP Hosting service for SMEs, an SLA aware provisioning of computer platforms, a multi-domain SLA for aggregated services and an eGovernement domain integrating human-based services.
• Contributions to open standards: the project has contributed to some open standard initiatives among which we can find the Unified Service Description Language USDL from W3C, and the Open Cloud Computing Interface (OCCI) and WS-Agreement from OGF.

• DSoS: Dependable Systems-of-Systems [195], [196]. This project ran from April 2000 until March 2003. Its main goal was to develop significantly improved means for the composition of a dependable System-of-Systems. Dependability in the context of this project encompasses reliability, security and maintainability, taking into account also timeliness and functionality issues. The project proposed a dependability modeling framework and architectures and dependable mechanisms for dependable SoSs. Several use cases were used to validate the contributions like a Travel Agency Service, or a middleware base system.

1.4 CONCLUDING REMARKS

This Section has presented an analysis of current state of the art on the different SoS properties presenting different points of view, needs and lacks, that will lead AMADEOS goals. The analysis is also accompanied by a short description on current SoS EU FP projects and also past projects related to AMADEOS.

In the following Section 2 to Section 6, we describe five reference domains from a SoS-like perspective. These are the automotive and the railway domains, the global automated teller machine network, the crisis management domain and finally the energy domain, with a specific focus on Smart Grids. Each domain is analyzed considering ten viewpoints, derived from and integrating the key elements identified in this Section. The considered viewpoints are: SoS constraints, semantic of communication, architecture and RUMI, dynamicity, evolution, emergence, governance, handling of time, dependability, security, and quality metrics. Whenever considered relevant, historical notions on the domain, a list of major incidents at SoS-level, and the available ontology are included. The analysis of such domains, and the relations between them defined in Section 7, shall be at the basis for the definition of SoS requirements in Section 8.
2 AUTOMOTIVE DOMAIN

2.1 INTRODUCTION

Examples of Systems-of-Systems, either existing or proposed, can be found in all societal sectors including road transportation.

Today’s cars are collections of embedded systems on wheels. Much of the innovation in the automotive industry in the last decade or two has been as a result of onboard computing, control, and communication, and this innovation has dramatically improved safety, fuel economy, emissions, and reliability. The solution of using the concept of SoSs when designing automotive systems may be taken into consideration in order to overcome a number of problems, as will be illustrated in the following Sections.

Moreover it is foreseen that in the near future autonomous functionality in vehicles will allow the dynamic formation of collective groups for collaboration and the realization of services beyond the ability of isolated vehicles.

In the automotive context the passage from advanced control to System-of-Systems (SoS) can be summarized in Figure 1.

![Figure 1 – Automotive context [1].](image)

A number of separate embedded systems exist in a modern road vehicle (those related to safety include collision impact warning, airbag deployment and seatbelt pre-tensioners, antilock and differential braking, intelligent cruise control, and traction and stability control). Often designed independently, these systems are nevertheless interdependent through the physics of the vehicle and the environment and the actions of the driver. Thus failure modes arose such as cars that locked themselves if the driver got out with the engine running and shut the door, or cars whose antitheft systems disengaged and doors unlocked if the cars were rocked side-to-side, triggering rollover detection.

The solution, evidently, is to adopt an SoS viewpoint when designing automotive systems. An example of “embedded” SoS within the automotive domain is extracted by [1] and it is shown in Figure 2. Standard network protocols and buses have already been adopted in vehicles. Some level of algorithmic integration has also occurred. Some systems coordinate traction control and...
antilock braking, for example. But much remains to be done, and with the continuing rollout of X-by-wire systems\(^2\) (e.g., active steering), more opportunities will arise.

![Graph showing intra-vehicle systems and control connections](image)

**Figure 2 – Example of intra-vehicle systems and its strong control connections** [2].

But the SoS approach is not only applicable at the level of individual vehicles. Future technology will enable vehicles to communicate and autonomously interact with their environment. Cars will exchange information in order to prevent collisions, optimize the traffic flow, or facilitate autonomous driving. The data exchange between vehicles or with roadside infrastructure is denoted as vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communications, respectively, which are often subsumed by V2X [8]. At this level the vehicle itself is seen as a subsystem of a huge traffic management system that continuously evolves and whose borders cannot be consistently determined due to the dynamic nature of that SoS. Such an SoS is not controlled by a single entity, but by the totality of its Constituent Systems (CSs), their interactions and the resulting emergent properties. According to the definitions of Dahmann [15], this type of SoS can be classified as **Collaborative SoS**.

In order to provide individual descriptions for the In-Vehicle SoS and the SoS enabled by V2X Communication, the following subSections are split into two parts.

### 2.2 Problem Area & Example Scenarios

**In-Vehicle SoS**

The automotive domain is mainly concerned – beside smaller niches, e.g., entertainment sector – with the transportation of people and goods on short and medium distances. Therefore, safety for

\(^2\) Electro-mechanical systems for performing vehicle functions traditionally achieved by mechanical linkages/actuators.
humans, goods and the environment is of utmost importance. Besides that, transportation in minimum time and with maximum comfort is desired. This requires optimal control by the CSs forming the vehicle, but it also involves the management of the traffic flow. In order to satisfy these challenges, the automotive domain puts a lot of effort in the design and implementation of modern automotive systems and infrastructure.

The SoS approach is introduced in the automotive domain at different levels, whereas this analysis focuses on the following two substantial views:

The first consists in considering the vehicle itself as the SoS, and as CSs the elements that make up the architecture of a vehicle (e.g., the engine control, safety subsystem, multimedia system, etc.). Some of these CSs can be regarded as an SoS on their own. At this level, the SoS controls the vehicle's engine, brakes, passenger room, etc. based on the driver input, current vehicle conditions and measurements from its immediate environment. The safety system is designed to reduce injuries and damages in case of hazardous situations, but not to prevent these situations before they even occur. Each vehicle acts as an individual system.

**V2X Communication SoS**

The second point of view is the system that arises when many different vehicles are interconnected, exchanging data on the current and future behavior of the vehicle in order to avoid dangerous situations and optimize the traffic flow. In this system, the constituent systems are the individual vehicles and infrastructure elements (i.e., roadside units), while the actual System-of-Systems is abstract and has a continuously changing dimension. At this level, intercommunication between vehicles and infrastructure allows to coordinate the traffic that emerges from the behavior of individual vehicles. Traffic jams can be prevented or reduced by balancing the traffic load among different roads. As position and speed of vehicles, road conditions and obstacles are communicated to other vehicles, the safety system is able to avoid hazardous situations even before they might appear. Figure 3 provides an example scenario of V2X communication, where the SoS implicitly emerges from the conjunction of individual vehicles, roadside units (RSUs) and cloud services.

![Figure 3 – V2X communication scenario](http://www.kapsch.net/ktc/its-solutions/V2X-Cooperative-Systems)
2.3 CONSTRAINTS

In-Vehicle SoS
A typical modern vehicle contains between a dozen and nearly 100 electronic control units (ECUs). Current electronic systems are typically partitioned by application domains.

There are two main classes of electronic systems: hard-real-time control of mechanical parts and information-entertainment. The first category includes

- chassis control;
- automotive body, including components such as interior air conditioning, dashboard, power windows, and control subsystems;
- powertrain, including the engine, transmission, and emission and control systems; and
- active safety control.

The second category includes information management, navigation, computing, external communication, and entertainment.

Each domain has its own requirements for computation speeds, time scales, reliability, flexibility, and extensibility.

For communications, a typical vehicle today contains two or three controller area network buses, with rates from 25 to 500 Kbytes, two or three lower-speed local interconnect network buses, and, optionally, some dedicated high-speed links for infotainment.

Experimental vehicles now being developed have up to 10 Controller Area Network (CAN) buses, with additional buses almost invariably providing 500-Kbps links. A further increase in the number of buses is unlikely because of the additional gateways and consequent increased latencies and jitter.

The critical architecture-evaluation and selection design-process phase affects profoundly a product line’s cost, performance, and quality [3].

V2X Communication SoS
V2X communication uses wireless communication technology, e.g., the IEEE 802.11p/Wireless Access in Vehicular Environments (WAVE) standard. Vehicular ad hoc networks (VANETs) [9] are established between vehicles and RSUs that are within the range of the wireless signals. This allows the exchange of information between vehicles and infrastructure in the vicinity of a vehicle. But with VANETs, applications are still restricted to a short range of interaction, which precludes wide-range control applications, like traffic flow control.

Therefore, additional wireless transmitters with a different communication technology (e.g., UMTS, LTE) are contemplated that enable the communication over longer distances. The single usage of long range technology is not reasonable, since also information which is only relevant for other vehicles in the near proximity would be transmitted on longer distances and to vehicles that do not gain a benefit. This possibly leads to overload of the communication medium.

However, also in the case of VANETs capacity overload has to be prevented. Especially in the case where many vehicles are in the operating distance of the wireless network (e.g., traffic jams, city traffic), data relay functionality and indiscriminate data exchange can lead to the breakdown of the wireless communication. The maximum number of wireless nodes in an area is restricted by the number of available communication channels and the media access technology.

Furthermore, the V2X technology might suffer a causality dilemma, since the beneficial application of information exchange between vehicles requires that a certain percentage of other vehicles also
dispose of wireless communication interfaces. But as this is currently not the fact, consumers are not encouraged to invest in this technology, unless they are forced by governmental authorities.

2.4 ARCHITECTURE & RELIABLE UPON MESSAGE INTERFACE (RUMI)

In-Vehicle SoS

An example of a vehicle architecture overview is shown in Figure 4.

![Road vehicle architecture](image)

**Figure 4 – Road vehicle architecture [4]**

In particular, in Figure 4 the CSs and their connections are shown.

A vehicle bus is a specialized internal communication network that interconnects components inside a vehicle (e.g., road vehicle, bus, train, industrial or agricultural vehicle, ship, or aircraft). Special requirements for vehicle control such as assurance of message delivery, assured non-conflicting messages, assured time of delivery as well as low cost, redundant routing and other characteristics mandate the use of less common networking protocols. Protocols include Controller Area Network (CAN), Local Interconnect Network (LIN), that are present in the considered architecture in Figure 4.

An electronic control module typically gets its input from sensors that it uses in its computation. Various actuators are used to enforce the actions determined by the module. The modules need to exchange data among themselves during the normal operation of the vehicle. For example, the engine needs to tell the transmission what the engine speed is, and the transmission needs to tell other modules when a gear shift occurs. This need to exchange data quickly and reliably led to the development of the vehicle network. The vehicle network is the medium of data exchange.

The automotive industry quickly realized the complexity of wiring each module to every other module. Such a wiring design would not only be complex, it would have to be altered depending on which modules were included in the specific vehicle. For example, a car without the anti-lock brake module would have to be wired differently than one that included anti-lock brakes.

For network spanning communication, automotive bus systems require appropriate bridges or gateways processors to transfer messages among each other despite their different physical and logical operating properties. Gateways processors read and write all the different physical
interfaces and have to manage the protocol conversion, error protection and message verification. Depending on their application area, gateways include sending, receiving and/or translation capabilities as well as some appropriate filter mechanisms.

While so-called super gateways interconnect in a centralized manner all existing bus systems, local gateways are linking only two different bus systems together. Therefore, super gateways require some kind of sophisticated software and plenty of computing power in order to accomplish all necessary protocol conversions, whereas local gateways realize only the hard- and software conversion between two different bus backbones.

**V2X Communication SoS**

Compared to the in vehicle communication architecture, where a fixed number of CSs are interconnected, in the SoS emerging from V2X communication the number of interacting CSs continuously changes. On the one hand, RSUs can be constantly interconnected with cloud service providers (e.g., centralized traffic management centers, toll collect services). And on the other hand, as vehicles move into or out of the operating range of wireless transmitters of other vehicles or roadside infrastructure, ad hoc network connections are dynamically established and shut down, respectively. Thus, the actual extent of the SoS cannot be consistently determined.

In general, the communication infrastructure consists of the following three network types:

- Wired networks between RSUs and service providers, which are based on standard network technology (e.g., Ethernet).
- Wireless ad hoc networks using short range WIFI technology (e.g., IEEE 802.11p, Bluetooth, ZigBee), that are used to exchange data relevant for the traffic in the vicinity of a vehicle. This is also described as *Dedicated Short Range Communication (DSRC)* [9].
- Wireless communication based on mobile telecommunication infrastructure (e.g., UMTS, LTE), which can be used for authentication of vehicles (e.g., by using certification services provided by certification authorities), exchange of traffic information with global impact (e.g., data on planned route to balance traffic flow) or to retrieve data from the internet.

### 2.5 Semantic of Communication

**In-Vehicle SoS**

The information that are exchanged among the CSs in a vehicle SoS can be split in two groups which in some cases can overlap: core services information and Infotainment information.

Core services regards all the information that are needed for the transport services, whereas infotainment are information-based media content or programming that also includes entertainment content. In the former, several subgroups can be identified, e.g. sensor signals, commands/driver actions and actuation. Sensor signals can be oil temperature, vehicle speed and engine speed, which can be used for example to activate or deactivate some vehicle component (e.g., the engine). Light sensor signal can exist as well and can be used, for example, to activate or deactivate the lighting system. Other signals, instead, can be aimed to status reporting of vehicle components (e.g., DTC, Diagnostic Trouble Code). As for commands and driver actions there could be: key signal which is used to power on/off the vehicle; blinker commands used to activate/deactivate the blinkers; lighting commands used to activate/deactivate some particular lights (for example high beams); steer commands, accelerator commands and braking commands used by the driver to drive the vehicle. As for actuation, we can identify engine driving information, used to activate/deactivate the engine, suspension control information used to tune the dampers and many others. On the other hand, infotainment information could be the digital radio, the audio signals to the speakers, GPS signal used by the navigation system. Sometimes infotainment information could be also used for core service purposes. Indeed, European Commission with the regulation N° 661/2009 [16] states that Advanced Emergency Braking (AEB) systems will be mandatory on heavy-duty vehicles from 1 Nov 2013 for new types of vehicle and from 1 Nov 2015 for all new vehicles. Starting from 2014, Euro NCAP (New Car Assessment Programme) [17]
foresees tests for verifying AEB system [18]. These AEB systems improve safety in two ways: firstly, they help to avoid accidents by identifying critical situations early and warning the driver; and secondly they reduce the severity of crashes which cannot be avoided by lowering the speed of collision and, in some cases, by preparing the vehicle and restraint systems for impact. In particular AEB systems employ sensors and/or cameras to monitor the proximity of vehicles in front and detects situations where the relative speed and distance between the host and target vehicles suggest that a collision is imminent. In such a situation, emergency braking can be automatically applied to avoid the collision or at least to mitigate its effects.

It is worth noting that also a lot of effort was made in order to facilitate the integration of legacy systems (i.e., CSs), in particular SAE International [19] published several standards about message formats.

As for the technology utilized for in-vehicle SoS, different communication protocols are used (see Section 2.4), including for example the CAN, which is a message-based protocol designed specifically for automotive applications.

CAN is a multi-master broadcast serial bus standard for connecting ECUs. Each node is able to send and receive messages, but not simultaneously. A message consists primarily of an ID (identifier), which represents the priority of the message, and up to eight data bytes. It is transmitted serially onto the bus. This signal pattern is encoded in non-return-to-zero (NRZ) and is sensed by all nodes.

The devices that are connected by a CAN network are typically sensors, actuators, and other control devices. These devices are not connected directly to the bus, but through a host processor and a CAN transceiver.

Another protocol named in Section 2.4 is the LIN that is a serial network protocol used for communication between components in vehicles.

LIN is a broadcast serial network comprising one master and typically up to 16 slaves. All messages are initiated by the master with at most one slave replying to a given message identifier. The master node can also act as a slave by replying to its own messages. Because all communications are initiated by the master, it is not necessary to implement a collision detection.

Current uses combine the low cost efficiency of LIN and simple sensors to create small networks. These sub-systems can be connected by a back-bone-network. (i.e., CAN in cars).

Coupled with growing bandwidth requirements with today's advanced vehicles utilizing over five separate CAN busses, the FlexRay standard [22] has emerged as the in-vehicle communications bus to meet these new challenges in the next generation of vehicles. The FlexRay protocol is a unique time-triggered protocol that provides options for deterministic data that arrives in a predictable time frame (down to the microsecond) as well as CAN-like dynamic event-driven data to handle a large variety of frames. FlexRay accomplishes this hybrid of core static frames and dynamic frames with a pre-set communication cycle that provides a pre-defined space for static and dynamic data. This space is configured with the network by the network designer. While CAN nodes only needed to know the correct baud rate to communicate, nodes on a FlexRay network must know how all the pieces of the network are configured in order to communicate. As with any multi-drop bus, only one node can electrically write data to the bus at a time. If two nodes were to write at the same time, the communication would end up with contention on the bus and data would become corrupted. FlexRay manages multiple nodes with a Time Division Multiple Access or TDMA scheme. Every FlexRay node is synchronized to the same clock, and each node waits for its turn to write on the bus. Because the timing is consistent in a TDMA scheme, FlexRay is able to guarantee determinism or the consistency of data deliver to nodes on the network. This provides many advantages for systems that depend on up-to-date data between nodes [23].

In addition to the signals sent on the communication bus, also analog signals are exchanged among the CSs. In some cases the control unit needs precise signals that are sent from the sensors to work properly. These sensors are mounted at precise locations of the engine. One example is the signal sent by the temperature sensor. Other examples of messages exchanged
are related to light sensors which allow switching on the lights, the oxygen sensor used to monitor the amount of oxygen in the exhaust, the vehicle speed sensor that is used to measure the speed of the vehicle, the turbine speed sensor, used to measure the rotational speed of the input shaft or torque converter, the MAP sensor, Manifold Absolute Pressure, used in regulating fuel metering and still others. Another example of message exchanged is related to commands sent from the dashboard to the calibration of the suspension.

**V2X Communication SoS**

While communication within the vehicle is mostly used to control the actual behavior of the vehicle or to improve the safety after an accident and the driving experience for the passenger, V2X communication targets an increased safety by preventing hazardous situations to occur and the optimization of the traffic flow.

Essentially, two different types of communication can be observed:

- The periodic transmission of movement or state information (e.g., position, heading direction, speed and acceleration values, emergency information of ambulance, etc.).
- The sporadic communication of events (e.g., vehicle wanting to turn, breaking) and infrequent states (e.g., accidents, slippery road sections).

The semantic of messages exchanged between subsystems has to be defined exactly, which should also be enforced by standardization. This enables the usage of information that originates from other subsystems, even if both systems never exchanged information before. At the end, the receiving subsystem is responsible for the actual usage of the exchanged data.

However, the availability of V2X communication cannot be assumed for all vehicles even one decade after the introduction to the mass market – unless authorities enforce the existence of V2X communication modules in vehicles. Furthermore, wireless communication can hardly be protected against intentional manipulation by overloading the communication channels. Thus, safety relevant applications within vehicles, and also traffic flow control systems, cannot simply rely on the existence of V2X information. This means that V2X communication has to be regarded as an added value for those systems that are enabled to use it, but not as an ensured source of information concerning the vehicle’s environment.

2.6 **DYNAMICITY**

**In-Vehicle SoS**

Inside a vehicle, it is possible to identify some components that play a role in recovery. These CSs allow performing a reconfiguration of the system in response to an unforeseen event.

For example, it is possible to think of the damper valve. The function of the damper valve is to reduce the unpleasant effect that occurs during travel on uneven ground. The function is also aimed to avoid adherence loss and, consequently, vehicle control loss due to vehicle wheels pulling away from the ground because of road surface deformation.

The valve allows compression or extension as a consequence of the variation of the forces in play and it thus allows the reconfiguration of the system.

**V2X Communication SoS**

The system emerging from V2X communication is able to react to lost wireless connections between its subsystems as the same information is typically available at different constituent systems. A connection to another subsystem is established where the information is provided. For instance, if an oncoming car on a highway missed to communicate an accident it realized some kilometers ahead, the next oncoming car will provide the same information. This principle works best if many subsystems appear in the operating rage of the wireless transmitters.
2.7 Evolution

In-Vehicle SoS

Today, drivers and vehicle occupants enjoy improved performance, comfort, and safety thanks to functions like the anti-lock braking system, Electronic Stability Program, active parking aid, emergency brake assistant, lane departure warning system, and proximity-controlled cruise control. However, the associated ICT that has grown up in vehicles over many years is becoming increasingly complex.

The future architectures should enable the vehicle to communicate with a future intelligent power grid and transport system and allow the development of completely new functions, such as an "autopilot" that could steer the vehicle autonomously in the distant future. Fully autonomous vehicles, also known as driverless cars, already exist in prototype, and are expected to be commercially available around 2020.

Another possible evolution concern, for example, is the possibility that a traffic information message is generated by in-vehicle sensors, which collect data and send them to the traffic information application that processes and aggregates the information. Then, this information is sent out via the Communication Unit (CU).

A further possible improvement could consist in enabling a remote control of car functions from both outside and inside the vehicle via mobile devices, e.g., open and close of windows with a smart phone.

Finally, we expect that multimedia buses and wireless communication interfaces will be soon available in the most modern cars.

New materials which may replace steel car bodies include duraluminum, fiberglass, carbon fiber, and carbon nanotubes.

V2X Communication SoS

Currently, V2X communication is only available in test environments and prototypical setups. Therefore, it is expected that in future this technology will evolve as a whole and will be a widely deployed feature in modern vehicles. Especially with the introduction of the envisioned autonomous driving functionality in vehicles, V2X communication can be used as an additional source of information which helps to improve safety and comfort. With the availability of V2X communication in vehicles, different cloud services will arise, like traffic control, participant authentication or toll collect services.

Due to the dynamic and decentralized nature of the SoS emerging from V2X communication, it will hardly be possible to introduce major changes at once. Hence, the system has to be capable of operating with different software and hardware versions, which provide different features.

2.8 Emergence

In-Vehicle SoS

A meaningful analysis of the emergent properties of the system requires an analysis of the interactions between components in the system as well as an analysis of the system's interaction with its environment. The vast majority of failures in complex systems are the result of systemic flaws based on the unexpected or hidden interactions between components.

V2X Communication SoS

Due to the loose coupling of individual CSs (i.e., vehicles and RSUs), V2X communication has a high potential for emergent properties (e.g., optimized traffic flow). It is the responsibility of each individual CS to decide how received information from different communication partners is evaluated and used for safety and planning mechanisms. Only if the majority of CSs acts as an intelligent swarm, positive emergent behavior can emerge. For instance, in case all vehicles decide
to take the same highway to reach their destination, the result will be a traffic jam on that road. If the information about the planned routes is exchanged among the vehicles, the traffic load can be divided on different roads such that almost all vehicles reach their destination faster. But still each vehicle decides individually whether to choose the best road for itself or for the group of all vehicles. The individuality and loose coupling makes it difficult to design intentional positive emergent behavior.

2.9 Governance

In-Vehicle SoS
Since the SoS environment is restricted to a vehicle, no relevant governance features are applicable.

V2X Communication SoS
Currently V2X communication is in the focus of research activities by multiple car manufacturers and technology providers in different regions of the world (i.e., Europe, America, Japan). Since the usefulness of V2X communication strongly depends on a large distribution and a high acceptance rate, and therefore on the interoperability between systems from different technology providers and car manufacturers, an appropriate governance model has to be supported by the largest community possible. This should also include national authorities as safety, security and data protection constitute important issues. A prominent consortium – with focus on Europe – that is working on the definition of standards and common V2X methodologies, which comprises global car manufacturers and technology providers, is the Car 2 Car Communication Consortium⁴.

2.10 Handling of Time

In-Vehicle SoS
Since this is a closed SoS, in the sense that there is no interaction with other vehicles, only a local clock reference is used, as typically done in these kinds of SoSs.

V2X Communication SoS
The optimization of the traffic flow as well as preventive safety measures require the communication of accurate position and speed information of vehicles in the neighbourhood. Due to the driving dynamics of modern vehicles, these values might change within milliseconds. Therefore it has to be known at which instant in time these values were valid and how much time passed since they have been measured. Otherwise, if outdated values are used, safety could be compromised or the positive effect of traffic flow control could be reduced or even reversed. In order to determine whether the data is still valid or to estimate new values based on the passed time, the clocks of sender (i.e., vehicle producing information) and receiver (i.e., vehicle using the information) have to be synchronized with an appropriate precision.

While distributed clock synchronization in rapidly changing networks, where the number of participants cannot be consistently determined, is cumbersome, GPS based clock synchronization provides the necessary precision with relatively low effort. This synchronization technique is also supported by the fact that GPS receivers will already be available in most vehicles which use V2X communication.

2.11 Dependability

In-Vehicle SoS
The dependability is a measure of a system’s safety, availability, reliability, and its maintainability.

⁴ www.car-to-car.org

An example of a safety mechanism in our System-of-Systems can be considered the fact that if the CAN is broken, the control unit that manages the engine supply disables the engine itself.

**V2X Communication SoS**

Since the subsystems using V2X communication cannot rely on the availability of other subsystems as sources of information which are connected via some V2X network, these systems have to be designed to tolerate the failure of other subsystems. This implies that each constituent system has other means to obtain information about its environment (e.g., cameras, laser scanners, etc.), or a fail safe mode is implemented. For instance, if one vehicle driving in front of another vehicle is breaking and it does not communicate this information via V2X, the other vehicle must still be able to stop before it crashes. The benefits resulting from V2X communication have to be regarded as added value which helps to increase safety or to optimize traffic flow and passenger experience.

Furthermore, a piece of information is usually not only available at one single system, but is given at different constituent subsystems (i.e., vehicles and in the cloud). In case some information cannot be directly obtained from the generating subsystem, it might be forwarded or also generated by another subsystem. For example, if the first car in a platoon on a highway is breaking, this information is relayed by other car to the last car in the platoon, even if the first and the last car are not within the operating range of their wireless transmitters.

### 2.12 Security

The increasing complexity of car functionality also requires more communication, which then introduces new risks. For example, some modern car radios automatically increase the volume at higher speeds to compensate for the engine noise. To be able to do that, the radio needs to be able to communicate with the speed measurement – i.e., a part of the car entertainment system is on the same communication bus as a critical control system in the car. This fact allowed for an attack where an update of the car radio (delivered via a CD) allowed taking over the brakes of the car. Other examples of such connections are door locks that open in case of an accident (meaning the lock, which has a wireless interface, and many of which have been broken in the past, is connected to the critical bus), or, for an electric car, the charger (which communicates with the outside charging station) needs to know the battery levels (which also need to be known by critical components).

In addition to massive connectivity, a car also has stringent real time requirements. Most message authentication in a car is – if not all – done by means of a shared symmetric key, as any asymmetric cryptography would be too slow. Experiments in the past have shown that the components in the car are not very good at protecting their memory, and many components have debug commands that essentially allow anyone on their communication bus to obtain a memory dump (which then includes the authentication key).

In addition to attacks on the car-IT itself, it has also recently been demonstrated that the algorithms can be attacked directly by a misbehaving environment. Concretely, it has been shown that some algorithms used for “caravan-mode” – i.e., a system that allows cars to automatically drive at a constant distance to each other – can be confused by erratic behavior of a single car, sending shockwaves through the whole caravan and causing undesirable behavior. With more and more automation, there is an increased danger of confusing algorithms with specially designed environment behavior (e.g., painting a wrong sideline on the road to force a car to drive in the wrong direction.)
While there is a large body of work on automotive security, and entire conferences are dedicated to that topic, the advance of use cases – smart charging, car-to-car communication etc. – is advancing rapidly, adding new threat scenarios.

**In-Vehicle SoS**

Automotive security is different to PC security for different reasons:

1. Combination of safety and infotainment
   a. Buyers demand modern connected infotainment systems;
   b. If a PC is hacked, data is lost. If a car is hacked, life is at stake.

2. Automotive software cannot easily be updated
   a. No monthly security update.

3. Attacker might have physical access to vehicle

4. Automobile IT systems have limited resources, real time constraints, and a long lifetime (15+ years)

It is also important to note that nowadays issues of security are more and more related: the violation of a safety requirement can be seen as a security threat and vice versa.

A very interesting research has been presented in [24], where authors demonstrate that an attacker who is able to infiltrate virtually an Electronic Control Unit (ECU) can leverage this ability to completely circumvent a broad array of safety-critical systems.

The security objectives are been summarized in the following categories:

- operational;
- safety;
- privacy;
- financial.

These security objectives counter generic security threats, as outlined in the EVITA project [6] and as summarized in Table 1.
The reference standard that deals with issues related to security aspects in the IT field is the ISO/IEC 15408 [7]. Since in the automotive field there is no legislation governing the issues of security, there is a need to adapt the approaches defined in the IT security evaluation standard ISO/IEC 15408 in order to address particular issues of automotive applications, such as the possibility that a security threat may also have safety implications. However in the automotive domain the functional safety standard ISO26262 [5] can be used as reference to describe the process flow to design and verify implementation of security requirements.

### V2X Communication SoS

Security threats in systems with V2X communication are very crucial since safety can be compromised, personal data can leak or malicious individual try to maximise their benefit by manipulating the communication. Due to the random, rapidly changing connectivity of subsystems, secure and authentic communication is harder to solve than for common internet connections. For instance, how can it be ensured that a communication partner that appeared for the very first time is really the person it pretends to be, and is the information he provides really correct? How can revocation be handled if a vehicle is stolen or disposed of? On the other hand, having central authorities for the exchange of crucial data could solve the problem, but it also enables them to monitor individuals and create movement profiles.

[9] lists many different security threats to availability, authenticity and confidentiality of VANETs. IEEE standard 1609.2 [11] is dedicated to standardize solutions for security issues with respect to management and data messages in VANETS. The standard proposes the Elliptic Curve Digital Signature Algorithm (ECDSA) [14] and certificates for authentication of messages.

### 2.13 Quality Metrics

**In-Vehicle SoS**

---

**Table 1 – Generic security threats and security objectives [6].**

<table>
<thead>
<tr>
<th>Aims</th>
<th>Target</th>
<th>Approach</th>
<th>Motivation</th>
<th>Security Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harming individuals</td>
<td>Driver or passenger</td>
<td>Interference with safety functions of a specific vehicle</td>
<td>Criminal or terrorist activity</td>
<td>Safety</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Privacy</td>
</tr>
<tr>
<td>Harming groups</td>
<td>City or state economy, through vehicles and/or transport system</td>
<td>Interference with safety functions of many vehicles or traffic management functions</td>
<td>Criminal or terrorist activity</td>
<td>Safety</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Operational</td>
</tr>
<tr>
<td>Gaining personal advantage</td>
<td>Driver or passenger</td>
<td>Theft of vehicle information or driver identity, vehicle theft, fraudulent commercial transactions</td>
<td>Criminal or terrorist activity</td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Financial</td>
</tr>
<tr>
<td>Vehicle</td>
<td></td>
<td>Interference with operation of vehicle functions</td>
<td>Build hacker reputation</td>
<td>Operational</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td>Transport system, vehicle networks, tolling systems</td>
<td>Interference with operation of traffic management functions or tolling systems</td>
<td>Enhanced traffic privileges, toll avoidance,</td>
<td>Operational</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Financial</td>
</tr>
<tr>
<td>Gaining organizational advantage</td>
<td>Driver or passenger</td>
<td>Avoiding liability for accidents, vehicle or driver tracking</td>
<td>Fraud, criminal or terrorist activity, state surveillance</td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Financial</td>
</tr>
<tr>
<td>Vehicle</td>
<td></td>
<td>Interference with operation of vehicle functions, acquiring vehicle design information</td>
<td>Industrial espionage or sabotage</td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Operational</td>
</tr>
</tbody>
</table>

---

AMADEOS - Architecture for Multi-criticality Agile Dependable Evolutionary Open System-of-Systems
Quality metrics were originally applied to software [20]. Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. In general, software quality metrics are more closely associated with process and product metrics than with project metrics.

Relevant quality criteria are: analyzability; changeability; stability; testability. These criteria (all together contributing to maintainability) are associated to the corresponding metrics. The user can associate threshold values with each of the quality model metrics, indicating minimum and maximum reference values accepted for the metric.

For example, it is possible to consider for the criterion of analyzability, the metric $lc\_stat$ [21] that represents the number of executable statements in the function; for the changeability it is possible to consider the $ic\_param$ metric that indicates the number of formal parameters of the function; for the stability one of the metrics is $dc\_calling$ that concerns the number of functions calling the designated function; for the testability it is possible analyze the $ct\_vg$ metric that indicates the cyclomatic number of the control graph of the function.

As for the hardware, some quality characteristics can also be found in terms of robustness and reliability. In particular, stress tests to verify both electrical and mechanical robustness are foreseen in the development process. In addition, ISO 26262 foresees the evaluation of HW metrics to evaluate the robustness of automotive-related systems against single point and multiple point failures.

### V2X Communication SoS

Beside generic and well accepted quality metrics for software, additional quality metrics can be defined for V2X communication. On the one hand, these metrics describe the quality for the pure communication between vehicles and RSUs, and on the other hand quality attributes for the actual application can be defined.

For instance, metrics related to communication can express the quality based on the time to establish a new ad-hoc connection with another vehicle or RSU, the robustness of connections, or the ability to rebuild broken connections. Examples for application specific metrics are the reduction of traffic load measured by the average time or carbon emission on a specific route, and increase of safety given by the number of avoided hazardous situations.

A special quality attribute might be the degree of dissemination and usage of V2X communication, since the quality of most applications improves as the number of participating vehicles – and thereby the amount of provided information – increases. For instance, traffic flow optimization performs best if all vehicles provide their routing information.

### 2.14 Standards

#### In-Vehicle SoS

It is worth mentioning the ISO/TS 16949:2009 [223] which defines the quality management system requirements for the design and development, production and, when relevant, installation and service of automotive-related products.

Then SAE standards [19] aim to regulate the integration of legacy systems (i.e., CSs), standardizing message formats.

As for the communication means, the most popular standard is the ISO 11898 [224] which specifies the CAN bus. CAN is the most widely used for data transfer in the automotive domain.

Finally it is worth to mention the ISO 26262 [225] whose aim is to standardize the activities to be performed during the vehicle lifecycle, in order to guarantee a certain safety degree.

#### V2X Communication SoS

The prevailing standard in V2X communication is IEEE 802.11p, which is also known as *Wireless Access in Vehicular Environments* (WAVE). This standard is an extension to the well-known
802.11 standards for wireless LAN communication that specify the physical layer (PHY) and the media access layer (MAC). Additionally, the IEEE 1609 standard family specifies complex DSRC related functionality on top of IEEE 802.11p. The most relevant standards are:

- IEEE 1609.1 – Resource Manager [10]
- IEEE 1609.3 – Networking Services [12]
- IEEE 1609.4 – Multi-Channel Operation [13]

These standards define the operation and intercommunication of vehicular on-board units (OBUs) and roadside units (RSUs), as well as public safety OBUs (PSOBUs) which are mounted in vehicles like police cars, ambulances or fire trucks. But while the existing standards address the basic functionality and operation of V2X communication at the network level, the actual exchange of data and its semantics is still not covered by standardization.
3 RAILWAY DOMAIN

3.1 INTRODUCTION

The railway system moves people and goods within a country and between countries; this system is critical to the economic and social wellbeing of several, if not all, EU nations. In Europe, based on quarterly figures, rail passenger transport performance at EU-27 level continued to increase by around 3 billion passenger-kilometres between 2010 and 2011 (+0.7 %). Note that many Member States have recorded a decrease due to the economical crisis, the highest decreases can be observed in Italy (-10.4 %) followed by Greece (-3.7 %, after a decline of 11.0 % in 2010) [33].

Statistics on rail freight transport in the European Union (EU) reports that the total performance in the EU-27 (European Union of 27 Member States) was estimated at 389 billion tonne-kilometres in 2010, a rise of 7.9 % compared with 2009 [33].

The LICB (Lasting Infrastructure Cost Benchmarking [25], [27], [26]) project from the International Union of Railways defines the railway infrastructure and identifies assets that compose it. We consider this classification as a starting point towards the exploration of the railway domain, as this is in our view able to define the complexity of this domain. We will then go deeper defining a composition schema of the infrastructure and its assets and we will put all components in place for an SoS-oriented classification.

All railways have the same basic targets. Beyond a safe railway, they are all working to maximise the capacity at which they can operate their networks, minimise passenger and freight delays, maximise the reliability of the infrastructure and rolling stock, and do all of these at minimum cost [34]. Since disruptions of the railway infrastructure can have a significant negative impact on the economy and security of an individual country [29] and the current railway system depends on ICT (typically to increase performance), the security and safety aspects (especially for wireless communications) become critical [28]. Especially for passengers transportation, additional undeniable requirements of the railway infrastructure are availability and timeliness of the train transport service, as well as accessibility and functionality of train stations, together with a sufficient degree of comfort.

The railway infrastructure is able to sustain such enormous quantity of people and goods moved daily. This makes it a complex infrastructure, composed of several items, players, and dependencies with other connected infrastructures. To identify the main items, we refer to the Lasting Infrastructure Cost Benchmark (LCiB) project. LCiB defines the railway infrastructure as consisting of the following items, assuming they form part of the permanent way, including sidings, but excluding lines situated within railway repair workshops, depots or locomotive sheds, and private branch lines or sidings:

- ground area;
- track and track bed, etc.;
- engineering structures; bridges, culverts and other overpasses, tunnels, etc.;
- level crossings, including appliances to ensure the safety of road traffic;
- superstructure, in particular: rails;
- sleepers, small fittings for the permanent way, ballast;
- points, crossings;
- access way for passengers and goods, including access by road;
- safety, signalling and telecommunications installations on the open track, in stations and in marshalling yards, etc.;
- lighting installations for traffic and safety purposes;
- plant for transforming and carrying electric power for train haulage: sub-stations, supply cables between substations and contact wires, catenaries.
The complexity of the railway SoS, together with its classification as a major critical infrastructure, calls for the investigation of the SoS to design solutions for better, safer and securer interoperability of the different CS and with respect to the adjacent SoSs.

### 3.2 CONSTRAINTS

The following are the main constraints that can be easily identified in the railway domain.

**Constraints of railway assets (general).** With respect to the existing assets identified above, it appears obvious that several of them are supposed to last through time and not subject to frequent changes. This is due to the intrinsic characteristic of the asset: we are referring for example to tracks, bridges, tunnels, etc. which are built with reasonable effort and supposed to last decades.

Most of the assets identified have safety-critical requirements, thus their development, approval from the certification authority, deployment and maintenance is regulated by standards (electrical, mechanical, or even civil engineering standards or regulation). Again, these are supposed to change very slowly through years: the cost in certification and the requirements from standards constitute a severe limitation and discourage frequent changes of an item.

**Hardware and Software.** Hardware and software of safety-critical components must undergo severe assessment and certification activities. This imposes limitations on the variety and characteristic of the hardware and software that can be used, especially for the usage of Off-The-Shelf (OTS) components. While hardware OTS are nowadays widely accepted and used, the usage of software OTS components is severely limited due to the risk that they may introduce new hazards in the system (thus the OTS software shall undergo a full certification process or the isolation of faults generated within the component shall be proved). This is somehow conflicting with the general trends in the industry, which rapidly turns to system integration based on the reuse of hardware and software components [61].

Similarly for the assets, the need of rather expensive and long certification procedure makes changes in the hardware and software architecture very slow, or avoided at all if possible. This is one of the reasons why many train components and railway equipment are produced and used for decades with almost no change at all.

For non-critical items, any kind of hardware or software is applicable. However non-critical items are in general ancillary to the railway SoS, as from the asset list reported in the previous Section and the SoS description reported in the Section 3.3.

See also Section 3.10, Section 3.11, and Section 3.12 for more details on software and hardware techniques and certification.

**Economic constraints.** The railway industry has historically been thought as a natural monopoly requiring unitary ownership of both infrastructure and operations. And this has been the way in which the industry has evolved in most countries around the world. Being a sort of ‘protected industry’, the concerns of safety and proper operations have not been coupled with equivalent concerns on costs and efficiency. However, there has been recently an increasing interest in privatisation and deregulation, freeing railways from government control. At the same time, governments have been keen on reducing public expenditure and seemed willing to transfer new rail investments to the private sector [30] with an associated push for diminishing costs and increasing market competition. Further details are reported in Section 3.8.

### 3.3 ARCHITECTURE & RELIED UPON MESSAGE INTERFACE (RUMI)

Railway systems are composed of several subsystems and several parts. In general, the current scenario sees the coexistence of different systems. Given the generality of the discussion and the impossibility to provide distinct information for each country in the world, we restrict our focus to Europe. Before 1989, in the countries of the European Union different control mechanisms often incompatible among each other were adopted. The European Rail Traffic Management System (ERTMS) has been the initiative aimed at overcoming this situation, by defining a shared standard...
to enhance the interoperability among the railway systems of different countries. As per today, ERTMS is the major initiative to uniform the railway domain between countries although local signalling systems are still used (and their full dismantle is still far to be reached). Actually the emerging model is to have the ERTMS compliant systems (e.g., for high speed and capacity lines and trains) coupled with ‘local’ individual and independent systems for the local lines (due to costs and limited need to be compliant with ERTMS). Consequently, we describe the railway SoS in the European Union starting from a description of the ERTMS, then we describe an alternative signalling subsystem that operates in parallel and independently to ERTMS (specifically, we will refer to the Italian case [62], although the same applies for most EU countries). Then, we report on assets and items that are part of the railway domain. Finally, we describe the railway SoS interfacing these two parallel signalling systems to the various assets that are part of the railway domain, and to boundary SoSs.

3.3.1 Railway ERTMS

The ERTMS system aims at remedying the fragmentation of the European rail network, identified as a major obstacle to the development of international rail traffic [63]. In fact, the principle of the system is to standardize several signalling systems currently coexisting in Europe and to produce an economic and technical solution to railway interoperability. From a functional point of view, interoperability is characterized by four major points: at the borders, the train should not change locomotives, the train should not stop, it should not have a change of control agent and the driver should not carry out control actions other than ERTMS standardized ones [63].

The ERTMS system has two basic components [65], [66]:

- **ETCS** (European Train Control System) allows transmitting the permitted speed and movement information to the train driver and constant monitoring of the driver’s compliance with these instructions. An on-board computer compares the train speed with the permitted speed and brakes automatically when it is necessary.

- **GSM-R** (GSM for Railways) is the radio system used for information transmission between the track and the train. It is based on the standard GSM but using different frequencies specific to rail. Together, these components form the new signalling and management system for Europe enable interoperability throughout the European Rail Network.

To deal with the very different configurations in the signalling equipment in the member states, ETCS has been conceived with three so-called application levels, which are a way to express the possible relationships between track and train. In application level 1, the track transmits to the train information allowing it to calculate constantly its maximum authorized speed. This information is transmitted to the trains by means of Eurobalises placed along the track and connected to the existing signalling system. The train position is detected by traditional trackside occupancy controlling devices, which are linked to the interlocking.

The track is considered as a set of structured fixed block sections. In application level 2, the ETCS uses a continuous radio communication mean, the GSM-R, to exchange information between the track and the train. The interlocking transfers the status of trains routes to the RBC (Radio Block Centre) which, in turn, calculates the correct movement authorities and the safe speed instructions it transmits to trains. In this level, balises are mainly used for odometry purposes and the train position is detected by the track. Compared to Level 1, the track-train communications are done by GSM-R.

Finally, in application level 3, the RBC uses the GSM-R transmission between the track and the trains like for Level 2. But at level 3, the track receives the train location and the train integrity from trains. Compared to level 2, this configuration offers a great simplification with cost reduction of the equipment on the track and an independence from structured fixed block sections by using moving blocks. Required for any new line in Europe, we focus only on application level 2 of ERTMS. For this application level, as any other level, the ETCS is distributed partly trackside and partly on board the trains (Figure 5).
The main equipment proposed in ERTMS can be separated in two groups [64]: trackside and onboard. The trackside equipment comprises (Figure 5):

- Eurobalises: transmission of fixed or variable data at a specific point of the line;
- Lineside Equipment Unit (LEU): calculates the variable data transmitted to the train by Eurobalises.
- GSM-R: Radio Transmission System for Lineside/Onboard bidirectional communications.
- Radio Block Center (RBC): calculates the variable data (movement authorities...) transmitted to the train by radio.
- Euroloop or infill loop: A loop allowing a transmission of additional data not essential for safety but avoiding unnecessary delays (e.g., transmission at distance of a signal clearance).
- Radio infill: transmission of additional data by radio.

The onboard equipment comprises (Figure 5):

- European Vital Computer (EVC, called Kernel in Figure 5): implements the onboard signalling functions.
- Transmission Modules (BTM, LTM, EURORADIO, STM) for the transmission of respectively Balises, Loops, Radio and Specific (national system's) data to EVC.
- Train Interface Unit (TIU): Interface between the EVC and the train.
• Man Machine Interface (MMI, also often called Driver Machine Interface DMI): the interface between onboard equipments and the driver.
• Odometry: speed measurements and distance / position calculation.
• Juridical Recorder Unit: Records the mission data (“Black Box”).

3.3.2 **Railway SCMT**

The Sistema di Controllo della Marcia del Treno (SCMT, [67], [68], [69]) is a discontinuous train Cab signalling system used in Italy. The SCMT is divided in two parts:

- **SSB**: Sottosistema di bordo (“Onboard Subsystem”);
- **SST**: Sottosistema di terra (“Ground Subsystem”).

The SST is composed, as a minimum, of a couple of fixed eurobalises. These communicate to the SSB a unique, pre-configured message, which contains indications on speed variations, slope, characteristics of the line, and temporary slowdown.

The set of two or more balises that are installed on the line and that transmit information are called Punto Informativo (Information Point, IP), and their redundancy is used to augment the availability of the SCMT system in case of failures of the balises [70].

The SSB is composed mainly of a computer which analyses data that are provided at the beginning of the train mission, and compares them during the train mission with data acquired from the balises. This allows generating a braking curve which the driver must respect, otherwise an emergency braking is forced. One of the fundamental components of the SSB is the Odometry system, which uses information from sensors to compute the speed and the distance between two IPs. Figure 6 shows the Driver Machine Interface for the SCMT.

![Figure 6 – SCMT Driver Machine Interface [70].](image)

3.3.3 **Assets**

Writing an exhaustive list of items that are part of the railway domain is a complex task. We report the asset list previously identified:

- ground area;
- track and track bed, etc.;
- engineering structures; bridges, culverts and other overpasses, tunnels, etc.;
- level crossings, including appliances to ensure the safety of road traffic;
- superstructure, in particular: rails, grooved rails;
- sleepers, small fittings for the permanent way, ballast;
- points, crossings;
- access way for passengers and goods, including access by road;
- safety, signalling and telecommunication installations on the open track, in stations and in marshalling yards, etc.;
• lighting installations for traffic and safety purposes;
• plant for transforming and carrying electric power for train haulage: sub-stations, supply cables between substations and contact wires, catenaries.

And obviously, there are:
• Trains and rolling stocks.

It should be also noted that there are infrastructures that may not be specifically part of the railway domain, but that are fundamental or ancillary to it. In particular, those that we identify are:

• **Power distribution infrastructure.** Power supply is fundamental for the operability of the tracks, as well as of the train stations.
• **Paved Road infrastructure.** Not fundamental for the operability of the track, but obviously it is relevant for the connection of people and goods to the railway infrastructure. Also fundamental to bring maintenance workers in specific points to operate maintenance procedures,
• **Water infrastructure.** Obviously this is not fundamental for the operability of the track, but it may be useful for the assets that are ancillary to the track, as for example railway stations, in order to guarantee a reasonable level of comfort.

We should also note that people have a key role and interact with all the assets, items and infrastructure mentioned. Amongst people, we include train driver and train personnel, passengers, maintenance workers, etc.

### 3.3.4 SoS Design and Boundaries

We present a first organization of the System-of-Systems in the railway domain, with the help of Figure 7. As while exploring the railway domain we did not identify any reference to SoSs, we try to draft the railway SoS using the knowledge on components, items or assets acquired in the previous part of this Section.

At the top of Figure 7, we present the ERTMS, divided in GSM-R and ETCS, and the SCMT (or analogous local signalling system) that defines interactions between the signalling items and constitute our first two Constituent Systems (CS; we remind from D2.1 [81] that a CS consists of computer systems and possibly of controlled objects and/or human role players that interact to provide a given service).

The following CS specifically for signalling can be identified:

• the GSM-R antennas;
• the railway on-board subsystems. Two CSs are identified here, which are the SCMT on-board system and the ERTMS on-board systems; they are composed of the items discussed in the previous part of this chapter.
• the trackside (ground) subsystems, which include the items discussed in the previous part of this chapter. Again, these are a CS for ETCS ground subsystem and one for SCMT ground subsystem. Note that the CS for ETCS is reasonably wide, including the RBC. From Figure 5, we observe that control rooms are not part of this CS, thus we do not include humans at this stage.
These CSs interface following the related standards ERTMS and SCMT; the connections are schematized in Figure 8 below.

**Figure 7 – SoS in the Railway domain.**

**Figure 8 – Interactions between signalling CSs.**

We now move forward from the railway signalling system, to identify connected or dependent CSs. The ETCS onboard CS and the SCMT onboard CS are part of the train, which itself can be considered a CS. It can be a passenger train or a freight train. Humans are an active part of this CS; in the case of passenger trains, we have train operators, train drivers, and passengers, while in the case of freight trains only train drivers are present.

Similarly as for the onboard signalling subsystems, the ETCS trackside CS and the SCMT trackside CS are part of a bigger CS, that is the tracks CS (Figure 9). Connected to the tracks CS, there are other two CSs: control rooms (control rooms and interlocking adjacent to the RBC, as from Figure 5), and train stations (Figure 9). These two CSs are in fact very essential structures for the existence of the tracks: without control rooms or train stations, it would be impossible to run trains. Humans are also part of these CS; we have specialized operators in control rooms, while in
the train station a large variety of roles coexist, ranging from passengers to security personnel, cleaning or maintenance workers, etc.

Figure 9 – Interactions between ground CSs.

As it can be seen from Figure 7, the CSs identified up to now constitute the core of the railway domain, that is, these CSs are considered essentials to the existence of the railway SoS.

Additional CSs or components part of the identified CSs can be identified; we organize them in supporting infrastructures, supporting components and assets, and adjacent infrastructures not fundamental for operation.

Supporting infrastructures are broad CSs that are fundamental for the core CSs. We say "broad" as each of these CSs is an infrastructure, which could be decomposed in several smaller CSs. At the present status of this work, the supporting infrastructure we identify is the electric infrastructure. In fact, power supply is fundamental for the train operation: most of the trains rely on power, and it is also necessary for operating GSM-R antennas, ground subsystems, and control rooms. Electrification of the tracks requires an entire new infrastructure to be built around the existing tracks; this binds significantly the SoSs in the electric and the railway domains. As this Section is centred on the railway domain, we consider the electric SoS as a "leaf" of the railway SoS (see Figure 9).

Supporting components and assets are very bounded to the identified CSs and very relevant to their operations. Thus we can enrich the tracks CS including bridge and tunnels, as well as maintenance workers and related equipment (e.g., Automatic Track Warning Systems temporarily deployed for maintenance operations on a line). Similarly we add maintenance equipment and personnel to control rooms and train stations CSs.

Adjacent infrastructures not fundamental for operation are CSs that have some degree of interconnection with the railway SoS, but are historically considered not relevant for the proper operation of the railway SoS. In the cases reported in Figure 7, we refer to:

- the road and building infrastructure and related SoS. Roads are necessary to bring goods and people at the train station, and it is a necessary support for maintenance track workers. Also, road may constitute a source of hazards, due to the proximity of roads and tracks. For example, consider the Helsinki accident in Annex A where a train derailed on a hotel.
- the water infrastructure and related SoS. Although any railway system can work well without water, it is a relevant element for humans i.e., train stations, control rooms, and the train itself (e.g., for toilets). That is, water supplies are somehow bound to the railway domain. A noteworthy case is that the interactions and proximity of water and railway infrastructures may be a source of hazards, for example see in Annex A the Merano accident where the explosion of an irrigation pipe was at the origin of a train derailment.

From the above mapping of CSs in the railway SoS, it emerges that boundaries of the railway SoS are very difficult to define, and that new hazards or emergent phenomena may rise due to the interactions of boundary systems and also the surrounding environment. At present, sensors are deployed and used e.g., in tunnels or bridges, or to detect landslides, but still the monitoring
sensors are not integrated with the signalling system and thus a unique and efficient monitoring and surveillance system is missing. A question which we are posing is whether such integrated monitoring system could result in quicker dispatching time of alerts to the trains, thus providing additional time to completely stop the train (we remember that a train due to its physical properties requires a considerable time to stop even when brakes are fully activated).

3.4 *Semantic of Communication*

The signalling systems ERTMS and SCMT use independent interfaces and communication means, described in the respective standards. We are not aware of standards that regulate message exchange between the electronic devices of the different CSs and SoSs identified in the previous Section.

3.5 *Dynamicity*

Whenever an unforeseen event is detected or suspected which potentially could lead to an accident (this is called in general *hazard* in the railway terminology), railway equipment as the trainborne equipment move to a safe state. Such safe state prevents further operations of the equipment, and usually requires handover procedures to restart or substitute the equipment. For example, a safe state commanded by the EVC constitutes in braking the train and shutting down the whole trainborne ERTMS signalling system; the manual intervention of the driver is then required to restart the train, causing a few minutes of train unavailability.

While this could be defined as a very conservative approach with respect to adaptiveness, it is widely considered reasonable for CSs, as for example the trains, where a potential unavailability affects the CS itself and at worst a few others connected CSs (e.g., nearby trains on the same track). Other railway CSs as the RBC are instead in charge of managing several CSs (e.g., the RBC guides several trains) and thus their unavailability would have a high impact on the railway network. In such cases, approaches for highly available systems as Triple Modular Redundancy (TMR) and majority voting are applied. With reference to adaptiveness, we note that such CSs are operating in a controlled environment with fixed physical (secured buildings) and logical (limited and known number of connections endpoint) boundaries.

3.6 *Evolution*

The past 50 years have seen a nadir in rail's fortunes, in the face of competition from road and air transport, but the mode has proved to be remarkably resilient, and its fortunes and its prospects have improved considerably in the latter part of this period [59]. Especially high speed rail developments in Europe have occurred for many reasons, although without any clear overall plan [60]. The European railway scene is now changing very rapidly, partly under EU pressure. The two major changes are [53]:

I. Fragmentation of the industry. Until a decade or so ago, all EU Member States had a single nationalised main line railway system. However, all States are moving in the direction of separating the provision of infrastructure from train operation, and some (Sweden, UK) have taken this a long way. There are many new potential new train operators, especially freight. This fragmentation requires a response to ensure the maintenance of safety. It also means that the UIC’s statistical task becomes much more difficult; reports by a limited number of public companies are no longer sufficient to cover the whole industry.

II. There is pressure towards increased "interoperability", that is, the movement of trains and locomotives across international boundaries. This requires steps towards the harmonisation of technical standards. At present, there is a Directive on the harmonisation of technical standards for high-speed lines, and a draft Directive on conventional lines. These Directives are in the process of being backed up by technical standards. All these developments have
implications for the management and regulation of safety; the most important implication is that safety decisions will inevitably tend to move from national to European bodies. 

Increasing concerns are raising about the environmental consequences of human activities, particularly with respect to the long-term climatic effects of the use of fossil fuels. The course of the next 50 years of evolution of the railway network is obviously unclear, but rail’s prospects seem favourable, especially in the light of its relatively low environmental impact. Assuming the necessary investment and the ongoing availability of suitable energy sources, the continued expansion of high-speed rail services is likely, releasing capacity for freight and other services on conventional routes. In the event of energy scarcity and economic stagnation or contraction, rail still has a valuable role to play, making efficient use of available energy sources to provide strategic passenger and freight transport services [59].

3.7 Emergence

Surveying the state of the art, we were not able to match the topic of emergence to the railway network and components. Anyway, the two following elements should be considered as potential sources of detrimental emergent phenomena.

The first element is the interactions between different domains or different SoSs. An example is the Merano accident where the explosion of an irrigation pipe nearby the track was at the origin of a deadly train derailment.

The second element is the growing complexity of the railway SoS and the increasing number of CSs. The absence of clear interfaces or a clear classification of the involved CSs makes very difficult to model or monitor the whole SoS, including assessing or mitigating hazards of the SoS.

We refer to the Annex A on railway accidents for potential examples of negative emergence in the railway domain.

While detrimental emergence examples can be identified analyzing safety reports, it is not trivial to identify positive emergence examples, as we are aware of no design-for-emergence approaches, or SoS-oriented design approaches, in the railway domain.

3.8 Governance

The railway system is a supranational critical infrastructure and it has a wide range of stakeholders and players. Considering the European railway system, the major stakeholders are: European Commission, that defines guidelines for railway system integration; Member States, that supervise the system; private/public companies, that implement and manage the infrastructure (e.g., the Italian RFI - Rete Ferroviaria Italiana - which manages the Italian railway infrastructure, by i) guaranteeing the safe circulation of trains, ii) assuring maintenance and efficiency of the network, iii) giving access to the railway infrastructure to companies, and finally iv) cashing in on fares for the usage of the railway network [31]) and local communities, that benefit from the service to transport goods and people [28].

Also the list of players involves several actors, from International or National companies to passengers. A simple classification - while not limited to - includes [28]:

- rail transportation companies (public and private) for passengers and goods (e.g., the Italian Trenitalia and Nuovo Trasporto Viaggiatori);
- supply companies: e.g., railway signalling system, trains, IT services, etc.; shipping companies; passengers;
- local transportation companies (e.g. GTT - Gruppo Torinese Trasporti);
- employees of different companies (e.g., Trenitalia, GTT, etc.).

Historically, the rail industry has been one of the most heavily regulated sectors of the economy [30]. Government controlled through history entry, exit, prices, technology, operating practices,
inter-company relationships and ownership. As a result of these considerations, the railway industry has historically been thought as a natural monopoly requiring unitary ownership of both infrastructure and operations. And this has been the way in which the industry has evolved in most countries around the world. However, there has been an increasing discontent with the traditional public utility regulation of railway, mainly due to its declining market share and the worsening financial performance, which has resulted in the increasing interest in privatisation and deregulation, freeing railways from government control. At the same time, governments have been keen on reducing public expenditure and seemed willing to transfer new rail investments to the private sector [30], [72].

Such reasoning has therefore triggered a process of dramatic changes. Starting in the early 90s, the business has suffered an important structural reform with the separation of roles and organisations within the railway system. Continued efforts have been applied by the European commission towards the implementation of Railway reforms in Europe, which helped paving the way for drastic changes in the manner in which railways are managed. The ultimate aim of these changes is the overall modernisation of the railway business in order to render this industry less dependent on subsidies for its financing, along with improved flexibility and capacity to face complex environments, helping its integration into the global transport system [30].

3.9 Handling of Time

Several components and equipment in the railway domain have hard real-time requirements. In some cases, these requirements are tough to meet, for example in the news it has been reported that the D-Bahn rejected 10 fast trains from Siemens because the braking system did not respect the activation deadline for brakes set to 0.1 second. Other components may have soft real-time requirements, or longer deadlines easier to meet. In general, the number of existing components implies a relevant variety in terms of real-time requirements and deadlines to meet.

A concept of global time is not widely adopted. The ERTMS and SCMT signalling systems do not rely on time and clock for the execution safety-critical functions; while clock is acquired by GPS for non-safety critical items as the SCMT Diagnostic system.

3.10 Dependability and Security

From standards [37], [38], [39], [40] the dependability of railway systems is based on reliability, availability, maintainability, safety and security (RAMSS). These attributes drive the definition of requirements that must be fulfilled to avoid, or to limit, accidents and attacks.

3.10.1 Safety and Dependability

The railway system is composed of different components and a holistic approach is required to satisfy its safety and dependability (hereafter simply dependability, from [57]) requirements.

Railway standards define recommended design and evaluation techniques to apply to hardware [39] and software [38] components to guarantee safety of electronic equipment. The recommendations for the different techniques are dependent on the required/desired Safety Integrity Level (SIL, see Section 3.11). Note that mechanical equipment has its own set of standards and test as well as maintenance procedures but it is not considered here.

EN 50129 [39] requires assurance that no single random hardware component failure mode is hazardous. This fail safety property can be achieved by (i) composite, (ii) reactive or (iii) inherent fail-safety techniques [58].

Composite fail-safety means that each safety-related function is performed by at least two independent items and non-restrictive activities are allowed only if the necessary number of items agree.

Reactive fail-safety assures safe operation by proper detection and negation of hazardous faults that occur in a single item. However, the fault detection function is regarded as a second item that
shall be independent in order to avoid common-cause failures. Independence could be lost by physical internal/external influences, or functional internal/external influences.

Inherent fail-safety is considered if all non-negligible failure modes of a single item are nonhazardous. Inherent fail-safety can be used for certain functions (e.g., to enforce shutdown). Justification shall be provided for any failure mode which is considered to be incredible.

EN 50129 contains recommended architecture-related techniques, as for example "dual electronic structure", "diverse electronic structure with fail-safe comparison", "single electronic structure with self tests and supervision", “single electronic structure based on reactive fail-safety”.

Moreover, the following techniques are highly recommended: (i) dynamic fault detection: on-line dynamic testing should be performed to check the proper operation of the safety-related system and provide an indication to the operator; (ii) program sequence monitoring: temporal or logical monitoring of the program sequence and indication of faults to the operator; (iii) measures against voltage breakdown, voltage variations, overvoltage, low voltage and temperature increase (to detect overtemperature).

EN 50128 [38] recommends a set of software techniques and measures to be implemented in the design phase. The main proposed techniques related to fault detection and handling are summarized in Table 2.

<table>
<thead>
<tr>
<th>Recommended Techniques</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defensive programming</td>
<td>Detect anomalous control flow, data flow or data values</td>
</tr>
<tr>
<td>Fault detection and diagnosis</td>
<td>Check for erroneous states based on a principle of redundancy and diversity on the physical, logical, functional or external levels</td>
</tr>
<tr>
<td>Error detecting codes</td>
<td>Detect errors in sensitive information by e.g., Hamming, cyclic or polynomial codes.</td>
</tr>
<tr>
<td>Failure assertion programming</td>
<td>Detect residual software design faults during the execution of a program by checking pre-conditions and post-conditions of operations</td>
</tr>
<tr>
<td>Safety bag</td>
<td>Protect against residual specification and implementation faults by an independent external monitor responsible for checking that the main system performs safe (not necessarily correct) actions and preventing that an unsafe state is entered</td>
</tr>
<tr>
<td>Diverse programming</td>
<td>Detect residual software design faults by using variants that were designed and implemented independently</td>
</tr>
<tr>
<td>Memorising executed cases</td>
<td>Force the system to a safe state if an unlicensed path (which is not registered as fault-free) is detected; the licensed paths are stored as a reference</td>
</tr>
</tbody>
</table>

Table 2 – Recommended techniques for fault detection and handling.

3.10.2 Security

While to the best of our knowledge specific security standards for railway applications are not published at the moment, safety of communication among electronic systems is addressed by the standard CEI EN 50159 [40], organized in two parts. These parts are the CEI EN 50159-1 [40] for closed transmission systems (from [40], a closed transmission system is constituted by a fixed number or fixed maximum number of participants linked by a transmission system with well-known and fixed properties, and where the risk of unauthorized access is considered negligible) and the CEI EN 50159-2 [40] for open transmission systems (from [40], it is a transmission system with an unknown number of participants, having unknown, variable and non-trusted properties, used for
unknown telecommunication services, and for which the risk of unauthorized access shall be assessed).

This standard discusses in general terms to what extent security is considered an issue for the safety of the railway infrastructure and what is the current level of protection. In closed transmission systems the risk of tampering is usually considered negligible, and the potential actions of malicious authorized users are mitigated by the internal logic of the system: the components which are in charge of taking decisions, e.g., the European Vital Computer (EVC), does not permit determined actions if these are not confirmed by sensors and other (human-independent) indicators (e.g., balises, or the Radio Broadcast Center).

The standard EN 50159-1 provides a set of requirements for closed transmission systems. Briefly, there are six main requirements that shall be provided:

- safety protection shall be applied to the generation of the data to be transmitted;
- safety reaction shall be applied in case of misoperation. This shall be consistent with the safety requirements of the receiver;
- error detection mechanism shall be applied at the receiver and shall be consistent with the safety requirements of the receiver;
- the implementation of the safety reaction shall be functionally independent of the non-trusted transmission system;
- the residual error rate of the safety-related transmission system for each information interchange between transmitter and receiver shall be less than a pre-defined value. This rate shall be compatible with the safety integrity level of each receiver;
- the safety integrity level of the safety-related transmission system shall be consistent with the highest safety integrity level of the safety processes.

Note that these requirements are safety-related requirements and do not explicitly mention security threats that may arise due to tampering, external attackers or malicious authorized users. This follows the definition of closed transmission system reported above.

The standard EN50159-2 provides a set of instructions and requirements for open transmission systems. In particular, seven possible security threats are identified: repetition, deletion, insertion, resequence, corruption, delay and masquerade. The standard presents guidelines to protect the transmission system; these are sequence number, timestamp, time-out, source and destination identifiers, feedback message, identification procedure, safety code, cryptographic techniques. This standard only considers unauthorized users while it does not address the possibility of malicious actions performed by authorized ones.

### 3.10.3 Conclusive notes

Many efforts have been provided to increase safety and interoperability of European railway system, and the available standard defines the interfaces between components as well as safety requirements for their implementation, deployment and maintenance.

However the identification and deployment of shared strategies (e.g., among the European nations) to protect the railway system is still considered challenging and sparsely, if not at all, prosecuted [28]. Therefore, considering the European railway system, a common and holistic approach for safety and security appears a rising necessity, starting from the adoption of shared risk analysis approaches to identify and manage critical aspects and threats of a modern and interoperable railway system, to the shared definition of constituent systems and their interplay, and finally to the definition and management of interactions with adjacent SoSs.

### 3.11 Quality Metrics

Railway standards [39] propose qualitative and quantitative levels for the safety of equipment, called Safety Integrity Levels or SILs. In fact, safety integrity is specified as one of four discrete levels. Level 4 has the highest level of safety integrity; level 1 has the lowest. Level 0 is used to
indicate that there are no safety requirements. A SIL should address qualitative appreciation of factors such as quality and safety management and technical safety conditions. A SIL can be related to the whole system or to a component or subsystem. The association of the SIL to the different subsystems/functionalities are (partially) detailed in EN 50126 [37] and EN 50129 [39].

Quantitative indications matched to each SIL are expressed in terms of Tolerable Hazard Rate (THR) and that are reported in Table 3.

<table>
<thead>
<tr>
<th>Tassò di Pericolo Tollerabile (THR) per ora e funzione</th>
<th>Livello di Integrità della Sicurezza</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-9} \leq \text{THR} &lt; 10^{-8}$</td>
<td>4</td>
</tr>
<tr>
<td>$10^{-8} \leq \text{THR} &lt; 10^{-7}$</td>
<td>3</td>
</tr>
<tr>
<td>$10^{-7} \leq \text{THR} &lt; 10^{-6}$</td>
<td>2</td>
</tr>
<tr>
<td>$10^{-6} \leq \text{THR} &lt; 10^{-5}$</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3 – SIL-Table [39].

The selection of an adequate set of methods, processes and techniques is the basis for the identification of qualitative measures assigned to the different SILs.

While the standards strictly focus on safety-related quality metrics as mentioned above, customers or operators of the railway line may demand other quality metrics. For example, availability and reliability of a train mission are typically measured using the well-known MTBF Mean Time Between Failures and MTTF Mean Time To Failure.

### 3.12 Standards

We analyzed standards and guidelines used in the railway industry. All the following standards provide guidelines to guarantee the safety of a system that has to be applied in the railway domain. IEC 61508 [41] (Functional safety of electrical/ electronic/programmable electronic safety-related systems) is the most important normative from which all the CENELC standards used in the railway industry are derived.

Table 4 presents the main standards and guidelines that are used in the European railway industry in order to demonstrate the safety (and security) of a railway system. Note that availability or reliability of a train mission are usually demanded to the specific requirements of the company in charge of managing the railway infrastructure, while the standards just focus on safety.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Name</th>
<th>Published by</th>
<th>Date</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN 50119</td>
<td>EN 50119:2009 - Railway applications - Fixed installations - Electric traction overhead contact lines</td>
<td>CENELEC</td>
<td>2009</td>
<td>Standard</td>
</tr>
<tr>
<td>EN 50121</td>
<td>EN 50121-1, 2, 3-1,3-2:2006 - Railway applications - Electromagnetic compatibility (Part 1, Part 2, Part 3-1, Part 3-2)</td>
<td>CENELEC</td>
<td>2006</td>
<td>Standard</td>
</tr>
<tr>
<td>EN 50122</td>
<td>EN 50122-1, 2, 3: 2010 - Railway applications - Fixed installations - Electrical safety, earthing and the return circuit (Part 1, Part 2, Part 3)</td>
<td>CENELEC</td>
<td>2010</td>
<td>Standard</td>
</tr>
<tr>
<td>EN 50123</td>
<td>EN 50123-7-1:2003 - Railway applications - Fixed installations - D.C. switchgear - Part 7-1: Measurement, control and protection devices for specific use in d.c. traction systems - Application guide</td>
<td>CENELEC</td>
<td>2003</td>
<td>Guideline</td>
</tr>
<tr>
<td>EN 50124</td>
<td>EN 50124-1:2001 - Railway applications - Insulation</td>
<td>CENELEC</td>
<td>2001</td>
<td>Standard</td>
</tr>
</tbody>
</table>
### Table 4 – Main railway standards and guidelines.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Name</th>
<th>Published by</th>
<th>Date</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN 50126</td>
<td>EN 50126-1:2012 - Railway applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)</td>
<td>CENELEC</td>
<td>2012</td>
<td>Standard</td>
</tr>
<tr>
<td>EN 50128</td>
<td>EN 50128:2011 - Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems</td>
<td>CENELEC</td>
<td>2011</td>
<td>Standard</td>
</tr>
<tr>
<td>EN 50155</td>
<td>EN 50155:2007 - Railway applications - Electronic equipment used on rolling stock</td>
<td>CENELEC</td>
<td>2007</td>
<td>Standard</td>
</tr>
<tr>
<td>EN 50159</td>
<td>EN 50159:2010 - Railway applications - Communication, signalling and processing systems - Safety-related communication in transmission systems</td>
<td>CENELEC</td>
<td>2010</td>
<td>Standard</td>
</tr>
<tr>
<td>EN 50163</td>
<td>EN 50163:2004 - Railway applications - Supply voltages of traction systems</td>
<td>CENELEC</td>
<td>2004</td>
<td>Standard</td>
</tr>
</tbody>
</table>

### 3.13 Railway Safety and Fatalities

The overall level of railway safety in Europe, as measured by fatal train collisions and derailments per billion train-kilometres, has gradually improved since 1990, although there is considerable scatter from year to year. The estimated overall trend is a reduction in the accident rate of 6% per year (4). This gives a fall of 70% from 1990 to 2012 [56].

Available historical data on fatal railway accidents shows a gradual improvement in safety over the past three decades, however restricting the analysis to the past eight years creates uncertainty about the trend in railway safety in Europe in recent years. This stems from the low number of fatal accidents and from their random nature.

#### 3.13.1 Major Incidents

Around 2400 significant accidents occur each year on the railways of the EU Member States. For all participating countries, the most common types of accident with victims are accidents caused by rolling stock in motion and those happening at level-crossings [55]. Accidents to persons caused by rolling stock in motion and level-crossing accidents constitute more than three quarters of the total number of accidents, excluding suicides. The Member States reported 1,480 accidents to persons caused by rolling stock in motion in 2011 [56].

Rail deaths are defined in terms of any person who is killed immediately or dying within 30 days as a result of an accident, excluding suicides. Accidents in workshops, warehouses and depots are excluded [55]. Above 4000 deaths including suicides were registered in 2012; these data is approximately in line with the previous years (see Figure 10 [56]).
Figure 10 – Fatalities on EU railways per year and victim category (2006-2011) [56].

Main EU railway train accidents and relations to the railway SoS are reported in the table below, for the year 2010, from the databases at [54], [56], [73].

<table>
<thead>
<tr>
<th>Accident name</th>
<th>Place, Date</th>
<th>Description</th>
<th>Relation to SoS view</th>
</tr>
</thead>
<tbody>
<tr>
<td>(no specific name)</td>
<td>Helsinki, Finland, 4 January 2010</td>
<td>An intercity train broke in two, and four carriages crashed into the main railway station's concrete buffer stop. Timely actions of operators and a guard diverted the train to a secondary track, and the platform was evacuated. One of the carriages plunged into a Holiday Inn hotel's offices causing severe structural damage. There were no casualties. Only the conductor of the train, which carried no passengers, was slightly hurt.</td>
<td>Manual usage of some of the complex items of the SoS (secondary track). Rely on personnel skills. Proximity of railway infrastructure to other structure (boundaries), which may lead to emergent phenomena.</td>
</tr>
<tr>
<td>(no specific name)</td>
<td>United Kingdom, 4 January 2010</td>
<td>A freight train hauled by a Class 66 locomotive derailed at Carrbridge in snowy weather, blocking the Highland Main Line. The freight train passed a signal at danger (occurs when a train passes a stop signal without authority to do so) and was deliberately derailed on trap points. The line is reopened on 12 January 2010.</td>
<td>Interplay of CSs (signalling, tracks, etc.)</td>
</tr>
<tr>
<td>Halle train collision</td>
<td>15 February 2010, Belgium</td>
<td>Two passenger trains collide head-on in Buizingen near Brussels. 18 people were killed and 162 were injured. The investigation revealed electrical problems with the red signal, which may have made it less visible. The train was not fitted with additional safety systems which could have reduced the risk.</td>
<td>Two CSs of the same kind (trains), and problems with the signalling CSs. Note that this accident would not happen using ERTMS.</td>
</tr>
<tr>
<td>Sjurøya train accident</td>
<td>24 March 2010, Norway</td>
<td>Sixteen goods wagons runaway for 8 km (5 mi) from a goods yard at Alnabu in Oslo hit and destroyed a quayside warehouse. Three people died and four were seriously injured. The line leading to the container and petroleum port at Sjurøya is a branch of Østfold Line, and is only used at low speeds. However, the empty carriages crashed into the harbour terminal at an estimated speed of more than 100 km/h (62 mph)</td>
<td>CS signalling not properly configured; bad interplay CS-CS (SoS) to monitor train speed. Proximity of railway SoS to other critical infrastructure (gas and oil infrastructure)</td>
</tr>
<tr>
<td>Event</td>
<td>Date</td>
<td>Location</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>A locomotive ran into a stationary passenger train after its brakes failed during a test ride in Spišská Nová Ves, Slovakia, killing three people and seriously injuring eight.</td>
<td>1 April 2010</td>
<td>Slovakia</td>
<td>A niche part to the railway SoS, which is the test ride and the interaction with the signalling system. Note that test ride are usually done in test circuits.</td>
</tr>
<tr>
<td>A passenger train is hit by a landslide and partially derails near Merano, South Tyrol, killing 9 and injuring 28. The landslide which reportedly derailed the train is believed to have been caused by a burst irrigation pipe, or by a valve inadvertently left open. Advanced sensors to stop trains in case of landslides were present and activated, but the train was too close to stop timely.</td>
<td>12 April 2010</td>
<td>Italy</td>
<td>Interaction of the railway domain with a boundary SoS (water distribution), mixed with characteristics of the environment may lead to new emergent phenomena.</td>
</tr>
<tr>
<td>A passenger train collided with boulders that had fallen on the line near Falls of Cruachan, derails and catches fire. Several people were hospitalised and the line was blocked for a week</td>
<td>6 June 2010</td>
<td>United Kingdom</td>
<td>--</td>
</tr>
<tr>
<td>12 people died and 14 were injured when they were struck by an Alaris while crossing the railway tracks at Castelldefels Playa station. A pedestrian underpass under the line was closed, although a footbridge over the line was available for use. A member of the Ecuadoran consulate in Barcelona, who had managed to cross the tracks with his family just before the accident, said that the signage of the exits from the platform was &quot;inadequate&quot;.</td>
<td>23 June 2010</td>
<td>Spain</td>
<td>Relations of the alerting systems with items that are placed at the very boundary to the railway SoS, if not excludable from the railway SoS: underpass, footbridge, signage.</td>
</tr>
<tr>
<td>A CityElefant train derailed. It was reported that at the time of the accident, the train was travelling at 108 kilometres per hour (67 mph) while the speed limit at that location was 50 kilometres per hour (31 mph). The most probable cause of the disaster is thus either a fault of the brakes or the driver's inattention</td>
<td>28 June 2010</td>
<td>Czech Republic</td>
<td>Lack of control of all CSs involved in the SoS and missing movement coordination for the different CSs.</td>
</tr>
<tr>
<td>A head-on collision between a passenger train and a freight train occurred on a single-track non-electrified line between Kępice and Korzybie. According to the initial results of the investigation, the passenger train left Korzybie station without permission, which led to a head-on collision with the freight train approaching the station.</td>
<td>13 July 2010</td>
<td>Poland</td>
<td>Lack of control of all CSs involved in the SoS and missing movement coordination for the different CSs.</td>
</tr>
<tr>
<td>A grinding train ran through the buffer stop at Stavoren station and caused substantial damage to a store. The train belonged to a Swiss operator; it was being driven by an Italian driver (not familiar with the route), who was piloted by a local Dutch train driver, who was distracted as he was talking with a third person in the cab. The ATP system on the track did not intervene as the on-board system was not compatible with the track-side system. Investigation was looking into both organisational aspects and human factors.</td>
<td>25 July 2010</td>
<td>the Netherlands</td>
<td>Different aspects concerning human contribution to the CSs and the SoS. In this case, the accident was due to: humans not doing their job properly, plus lack of automated monitoring system of the train.</td>
</tr>
<tr>
<td>A train derailed leading to a single fatality and thirteen other people with severe or critical wounds. The main cause of the accident was excessive speed. One witness said the driver was talking on his mobile phone. Maintenance work to provide structural reinforcement to the side walls had been completed three days before the accident, but allegations that the line had structural problems and reports of anomalies were dismissed.</td>
<td>6 August 2010</td>
<td>Italy</td>
<td>Human factor has their impact in the SoS; distraction caused by the phone possibly together with some safety holes on the lines were at the origin of the disaster.</td>
</tr>
</tbody>
</table>
Although the train was equipped with an automatic speed control system, the required rail components were not installed.

A diesel multiple unit was involved in a collision with a lorry on a level crossing. The primary cause of the accident was that the lorry driver drove onto the crossing when it was unsafe to do so. A lack of track circuits on the line meant that sometimes users of the level crossing faced a wait of up to 19 minutes before being given permission to cross.

A train crashed into a garbage truck that had slid down an embankment and wound up stranded on the track in the western state of Rhineland-Palatinate.

An X 2000 high speed train collides with a backhoe loader, which was there for maintenance work.

40 people were injured when a train derails at Skotterud due to a cracked wheel.

A lorry fell off an overbridge onto a passing train.

A passenger train ran into a large amount of ice that had fallen onto the tracks underneath one of the tunnel's ventilation shafts and derailed.

**Table 5 – Railway accidents in the EU in 2010 and relations to SoS.**

### 3.14 Ontology

The most typical terminology adopted in the railway domains comes from National and International standards. Additionally, there are instead several train dictionaries, for example (for a more complete list, refer to [43]):

- The Railway Age's Comprehensive Railroad Dictionary [45].
- Lexique Anglais-Français - La revue Le Rail Miniature [46].
- the Learner's Dictionary of Railway Rolling Stock [47]
- the Wordsworth Railway Dictionary [48].

An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary [51]. As per today, an ontology which is widely affirmed and commonly used amongst researcher centres and industries does not exist.

The most known attempt to define an ontology for the railway domain is constituted as part of the outcome of the FP6 INTEGRAIL project [34], [35]. INTEGRAIL developed a Railway Domain Ontology (RDO), extracted on the basis of Use Cases, aiming at building the corresponding ontology models. The RDO provides a means of creating a machine interpretable conceptual
model of physical components and domain data concepts. The model was created using the Web Ontology Language (OWL [44]), which is a W3C standard for encoding knowledge. The proposed solution implements a semantically enabled network of reasoning nodes, where information is integrated and shared using the RDO and distributed reasoning over a service oriented architecture (SOA). The application of the RDO aims at solving integration challenges within the railway domain [42].

The ISO 15926 standard [49] was initially developed for the integration and exchange of information relating to process plants including oil and gas production facilities. ISO 15926 takes a very “ground-up” approach to the modelling of processes, allowing each piece of equipment to be described in terms of its component parts as well as its temporal existence. Recently, an Ontology Web Language (OWL) implementation of the ISO 15926 standard has been produced, which could serve as an upper-level ontology for the project.

In [50] the SRO-ontology, constructed using the Protegé tool, is presented for the Southern Railway (India) using the Web Ontology Language and the Semantic Web Rule Language. Consistencies checking of ontology and classification of classes have been done with the support of Pellet reasoner. The ultimate objective of the ontology is supporting a person which wants to travel from one place to another place by train in India, where he has to plan for the shortest route, minimum traveling time or cost effective route and the details of the trains operating in the identified route for the comfort of his journey.

In [52] the authors build and exploit a railway ontology with the objective to develop a possibility for a semi-automated verification process of the static railway infrastructure with focus on the safety components. In the ontological framework of [52], implicit expert knowledge of the railway domain as well as the legal German guideline for railway infrastructures and their correlations are made explicit and stored in a conceptual model. The key concept used as starting point is the track, where all other physical elements are aligned with. Tracks as well as switches have a spatial extent and can be interconnected among each other. All other physical elements like signals, balises, train detection elements and so forth are point-shaped and related to tracks or switches. There do exist virtual elements like routes and railway control centers, but these elements are an overlay of physical elements and subsume their spatial extent.
4 GLOBAL AUTOMATED TELLER MACHINE NETWORK

4.1 INTRODUCTION

Automated Teller Machine (ATM) terminals are networked embedded systems that provide cardholders access to financial transactions in public places worldwide.

Currently, there exist approximately 2.2 million ATM terminals\(^5\) connected to financial (inter)bank networks. VISA is one of the largest companies operating in the international financial interbank network ‘Plus’ which is also used to process transactions originating from ATM terminals. VISA reports in 2010 that this network supports more than 24,000 Transactions per Second (TPS)\(^6\). VISA’s main rival MasterCard claims for its international interbank network ‘Cirrus’ an even higher number of around 44,000 TPS with an average response time of 130 milliseconds [75]. Both interbank networks have an availability of at least 99.9%.

We consider the global ATM network as one of the largest man-made System-of-Systems (SoS) that emerged from the corporation of many financial institutions under the governance of global standardization efforts.

Worldwide, there are more than 80 partially interconnected (national) interbank networks\(^7\). Further, the global ATM network has to handle lots of ATM terminal types from many manufacturers. The single Constituent Systems (CSs) have to deal with diverse dependability constraints, and there is no central control. According to the SoS classification [82] suggested by Dahmann and Baldwin, the global ATM network represents a Collaborative SoS where CSs “interact more or less voluntarily to fulfil agreed-upon central purposes”.

![Figure 11 – Automated Teller Machine\(^8\).](http://en.wikipedia.org/wiki/File:Atm_blockdiagram.png)

A typical ATM consists of a card reader for interacting with a payment card, buttons to select services, a display to present the user with information, an Encrypted Pin Pad (EPP) to securely handle Personal Identification Number (PIN) and cash amount inputs, a CPU system to control the

\(^5\) http://www.atmia.com/

\(^6\) http://investor.visa.com/phoenix.zhtml?c=215693&p=irol-IRHome

\(^7\) http://en.wikipedia.org/w/index.php?title=Interbank_network&oldid=569900439

\(^8\) http://en.wikipedia.org/wiki/File:Atm_blockdiagram.png
ATM terminal’s periphery and communication with the (inter)bank network and a cash dispensing mechanism.

The ATM terminal is only one of many payment card accepting devices that use the same payment processing infrastructure. Other popular payment card accepting devices are Electronic Funds Transfer Point of Sale (EFTPOS) terminals and various vending/ticketing machines.

4.2 **Problem Area & Example Scenarios**

The purpose of the worldwide ATM network is to provide bank/credit account self-service to payment cardholders or human users at an ATM terminal. This account self-service allows the user to perform cash withdrawals at most ATM terminals in the world. In case the financial account of the user is in a different currency than the local currency of the cash contained in the ATM, an automated currency exchange to the local currency occurs. This exchange usually takes place at a very good exchange rate. Hence, ATM terminals are the preferred means of travellers to transfer funds from the home currency to the local currency.

Beside these basic account self-services, there are many other additional features that financial institutions offer only to customers using the financial institute’s own ATM network: e.g., cash deposits, transfer of funds between account and an electronic purse, bill payment and many more financial related services. These additional services are not internationally standardized and require customizations of the ATM terminal, the network(s) in between, the payment card and the host equipment of the financial institutions that issued the bank card.

In the global ATM network the solving process of the account self-service problem involves most importantly:

- **Cardholder verification:** To prevent fraud the global ATM network needs to verify that the person using a payment card is the actual cardholder. The most established cardholder verification method is PIN-based and requires the user to enter a four digit number before any cash payout can take place.

- **Secure cash dispensing:** The ATM terminal needs to count and to hand out money according to the financial transaction.

- **Secure financial transactions among unrelated financial institutions:** Financial institutions are interconnected by a network of networks. An **acquiring** financial institution (e.g., a bank that operates the network an ATM is connected to) needs to interact with the **issuing** bank to authorize withdrawals/advances, handle fees, and issue clearing/settlement.

4.2.1 **Example Scenario: Cash Withdrawal with ATM/Debit Card**

We briefly describe the steps of an EMV (smartcard based payment card) cash withdrawal process. This example scenario introduces important terms used in the payment card industry.

1. **Cardholder initiates transaction by giving ATM/debit payment card:** The ATM terminal contains a card reader that interacts with the payment card. In older systems only the magnetic stripe is read. Newer systems communicate with the embedded system contained in the payment card.

2. **ATM terminal authenticates payment card and it verifies the cardholder:** Depending on the payment card technology, several payment card authentication mechanisms exist. For magnetic stripe these mechanisms are simple and can be easily forged. Payment cards based on smartcards offer more sophisticated authentication mechanisms based on cryptographic signatures. The most often used cardholder verification method is PIN checking. More advanced verification methods that are based on biometric scanners exist, but they are not globally deployed.

3. **Cardholder specifies amount above floor limit to withdraw:** The ATM terminal offers a user interface that allows the cardholder to specify an amount to withdraw. The ATM terminal...
also checks if the amount entered is above a floor limit. In case the amount is below the floor limit, the financial transaction is carried out offline, i.e., the issuing bank is not contacted for authorization during the cardholder’s ATM terminal session, but at a later time during batch processing. This significantly speeds up the withdrawal process for the cardholder, but bears the increased risk of fraud. For ATM terminals the floor limit is usually set to zero which enforces online processing in all cases.

4. In case the ATM terminal and payment card agree on online processing:
   
   1) Acquiring bank enriches transaction with data concerning settlement: The ATM terminal is directly connected to the network of an acquiring bank that operates and maintains the ATM terminal. Hence, the acquiring bank is the previous owner of the dispensed cash and needs to be compensated by the issuing bank which deducts the withdrawal amount (plus fees) from the cardholder’s account. This compensation process is called clearing and settlement.

   2) Bank that issued payment card contacted to authorize transaction: Payment cards are tied to accounts of customers of the issuing bank. Only the issuing bank has records about the cardholder’s accounts (e.g., balances, withdraw limits). The process of checking cardholder account data against the ongoing financial transaction is called authorization. After an authorization is given, the available amount of the cardholder’s account for further authorization is deduced accordingly.

   3) Immediate clearing/charging of funds for cardholder: After the payment card acknowledges the successful authorization of the issuing bank, it creates a Transaction Certificate (TC). Upon reception of a TC the issuing bank charges the account of the cardholder.

   4) Settlement for acquiring bank handled via payment processor and/or ACH (Automated Clearing House) credits/debits: Settlement is carried out by a payment processor or ACH financial transfers (e.g., wire transfers) between banks. Only the net difference is transferred.

5. ATM terminal dispenses cash: The ATM terminal contains a vault with cash and dispenses it according to the financial transaction.

In the background of any financial transaction counter-fraud measures and a complex fee distribution to entities that participate in financial transaction takes place.

4.3 ARCHITECTURE & RELIED UPON MESSAGE INTERFACE (RUMI)

The Constituent Systems (CSs) of the global ATM network are:

- Issuing Bank: The issuing bank authenticates the payment card, authorizes the transaction of the ATM/debit cardholder, and handles accounting of the cardholder.

- Acquiring Bank: The acquiring bank operates an access point where ATM terminals or POS (Point of Sale) terminals belonging to merchants can connect to. The acquiring bank manages the ATM or merchant accounts (i.e., those accounts that acquire funds). Especially for POS terminals, part of the acquiring bank’s services is often outsourced to payment processors. For our case study, we do not treat the payment processor as its own CS, because we focus on the global ATM network.

- ATM/debit Payment Card: This is a sealed computing device for establishing authentication/integrity between card, acquiring, and issuing bank. Also, the payment card incorporates mechanisms to support the verification of the cardholder (e.g., PIN, signature, offline PIN…). The card enforces transaction rules (e.g., limited number of consecutive offline payments …). Further, the payment card is able to execute ‘script commands’ from the issuing bank that are related to, for example: card blocking and changing limits. Finally, the payment card may house add-on features, e.g., e-purse.
• ATM/POS Terminal: The ATM/POS terminal is connected to the acquiring bank or a payment processor that aggregates single terminal setups before connecting to an acquiring bank. The ATM/POS terminal connects the payment card to the other CSs and makes the services of the payment card accessible to the cardholder. If a financial transaction is successful, the ATM terminal also dispenses money or in case of a POS terminal, the merchant hands over products.

• Networks: Several networks interconnect all CSs, i.e., the issuing bank, the acquiring bank, and the ATM/POS terminal. We can distinguish between direct point to point connections (e.g., payment card to ATM/POS terminal) and networks of networks, for example an interbank network that connects multiple bank networks. Payment processors are located between acquiring bank and merchant. They aggregate single terminal setups and offer special services intended for smaller merchants (e.g., few/single POS terminal(s), e-commerce shop, …). Interbank networks are realized as Electronic Funds Transfer EFT networks consisting of EFT switches. An EFT switch must adhere to all implementation details of the interbank networks it supports. The interbank network knows about all participating financial institutions and provides payment information validation: e.g., any financial transaction must be cleared by the interbank network operator.

4.3.1 Relied Upon Message Interfaces (RUMIs)

Figure 12 gives an overview on the interconnected CSs and their RUMIs during the time of a financial transaction: Human, (a), (b), (c), and (d). A RUMI is located at the boundaries of a CS. The RUMI Human consists of the Encrypted Pin Pad (EPP), buttons, display and cash dispenser. One might find it appropriate to consider the human interaction with the terminal as part of the CS ATM/POS terminal, because the human interaction determines the terminal's behavior and is the source of its autonomous interactions with the remaining CSs.

The neighbor relationships of any two CSs represented in Figure 12 express direct communication dependencies. For example, the ATM/debit card CS can interact with the issuing bank CS as long as it is inside (or in case of contactless cards: in close vicinity of) the ATM terminal.

In Figure 12 we represent several transactions among the CSs:

• Offline transaction (a1): An offline transaction only involves the payment card, cardholder and the ATM/POS terminal. For example, if the payment card authenticity cannot be established (e.g., the payment card is unreadable or has expired), no other CSs of the SoS need to be contacted. Other examples of offline transactions are: offline financial transactions (e.g., payments at a POS terminal below the floor limit), viewing the amount of money in the e-purse, …

• Online transaction (a2): In an online transaction all types of the CSs are involved. An example for an online transaction is a cash withdrawal financial transaction that requires online authorization given in Section 4.2.1.
• Batch processing (b1): In case of offline financial transactions\(^9\) (usually at POS terminals) that do not require online authorization and/or where clearing/settlement has not been carried out yet, the set of financial transactions are submitted to the acquiring bank by the merchant once per day.

• Clearance/Settlement of funds (cd1): Clearing and settlement concludes any financial transaction. In the clearing process also the interbank network is involved which validates the payment information contained in the financial transaction.

Some transactions are not represented in Figure 12: e.g., key exchange, parameter updates, ... Also there are fees related to the usage of interbank networks that we do not consider in our architectural model of the global ATM network SoS.

Finally, we regard the sum of all possible transactions where a specific CS can participate as its RUMI.

### 4.3.2 Specifications

The human interface is defined by the terminal manufacturer and is constrained by public expectations as well as the ultimate purpose of the underlying ATM terminal. Even though there seem to be no standards, this apparently works: most ATM terminals are usable by cardholders.

The other RUMIs are only constrained by standards (for details about the involved standards, see Section 4.11). For example the financial transactions and messages are only partially defined (e.g., EMV, ISO 8583, …). These partial definitions do not contain any routing and they are either interpreted or extended by the respective CS operator or manufacturer which often leads to property mismatch and it increases the effort to manage interoperability among CSs.

Concerning timing, the accessible standards specify timeouts only for the parts of the SoS which is most often replicated: e.g., the physical point-to-point connection of (a) is defined in the EMV standard. For financial transactions, the EMV standard only mentions the existence of timeouts that in the end are defined by the respective CS operator.

All RUMIs certainly adhere to rigorous specifications such that the overall SoS is able to reach its goal. However, for the global ATM network, these specifications are only partially public and/or standardized.

### 4.3.3 Classification of Transactions

We refer to the definition of a transaction in the context of SoS as follows: “A transaction is a sequence of computational and communication actions between connected interfaces conforming to rules of a protocol” \(^{[81]}\).

In the global ATM network SoS we identify two types of transactions:

• Local Transaction, e.g., see Figure 12, (a1):
  
  There are only a few underlying protocols (e.g., contact, contactless, …), and the overall transaction activity interval is influenced by a few factors.

• SoS Transaction, e.g., see Figure 12, (a2), (cd1):
  
  There is a vast amount of underlying protocols (interconnected Interbank-networks) and multiple CSs involved. An SoS transaction takes place under a complex mix of message recoding/enrichment. Hence, the transaction activity interval is influenced by many factors.

All those transactions activity intervals are bounded by timeouts.

\(^9\) Also called ‘dual message transactions’ \(^{[76]}\).
4.4 **CONSTRAINTS**

Payment cards as well as terminals to process them represent a considerable investment for the companies involved. Hence they are designed to operate for many years. While there are mechanisms to ‘nudge’ participants of the payment card industry into updating their systems (e.g., liability shift, penalties, exclusion from networks, …) interoperability with especially older technology is the generally accepted practice.

Consequently most constraints in the global ATM network at the macro-level appear to stem from economics. However, some of the CSs like payment cards also need to adhere to physical constraints (e.g., size, weight, durability, …), hardware constraints (e.g., processing power, memory size, available network connectivity, …), or software constraints (e.g., terminal operating systems). These constraints partly influence each other.

4.5 **DYNAMICITY**

Interbank network operators, as well as acquiring and issuing banks constantly monitor financial transactions to detect and prevent fraud caused by stolen payment cards. In case fraud is detected, there exist mechanisms (lock card, reverse transactions, insurance, …) to limit the effects of ongoing fraud and to stop fraudulent use of a payment card.

Besides fraud, interbank network operators as well as participating financial institutes are interested in high availability of their network and systems. Any outages mean a considerable loss of revenue. Hence this possible loss of revenue caused by the outage of a CS determines the scale of the used fault tolerance and load balancing mechanisms. For example, payment cards or ATM terminals usually do not tolerate faults (e.g., loose contacts, loss of power, …). A customer is easily able to get a replacement card or use another ATM terminal in his/her vicinity. For these cases, dynamicity is constrained by economics (e.g., the cost of a second chip on every card, or a backup ATM terminal compared to the occurring faults wouldn’t be justifiable). In contrast, banks and interbank network operators employ sophisticated fault tolerant and load-balancing systems and networks that aim to maximize service availability and, in case it is unavoidable, minimize service outage. These systems depending on the occurring fault scenarios can exhibit high dynamicity, i.e., react very fast to changes (faults) in the environment.

4.6 **EVOLUTION**

The global ATM network evolved from many small separated bank and interbank networks to a global scale network of bank networks. In the US, the number of interbank networks peaked around 1985 and declined to 25 interbank networks in 2006 because of consolidation and a decreased number of new entrants into the industry [79], [80].

Another angle of evolution is the adoption of new technology. Approximately 20 years ago, an upgrade from magnetic stripe based payment cards to chip based payment cards started. In 1994, French bank cards included a chip and through issuing payment cards with PINs fraud could be considerably reduced [78]. Based on this success, the technology was standardized (EMV) and today an end of insecure magnetic stripe cards is conceivable. In Figure 13, the EMV adoption rate is illustrated from the year 2011. Although for the US there are no official data available, the adoption rate is very low because the upgrade process started only very recently in 2010.
For the global ATM network we conclude that any advancement is designed to be coexisting with old and preexisting processes, services, and infrastructure. There are no sudden changes and fallbacks are implanted. For example, EMV cards also contain a magnetic stripe which is currently used in case the EMV infrastructure is unavailable or the chip in the payment card is not readable.

Evolution is driven by aspects of the goals of the global ATM network itself. For example, there is the need to protect stakeholders against frauds. In case of frauds, there is financial damage that one or multiple involved parties need to compensate, i.e., who is held liable for the damage. If a merchant supports old/legacy technology (e.g., operates a magnetic stripe based terminal in the EU) that is considered insecure, then the merchant is liable for all damages caused by fraud and his/her terminal. In the US, this liability is partly still at the interbank network operators. However, a liability shift away from the card schemes is already scheduled. Thus, all involved parties (especially terminal operators) are motivated to update at some point. This point is reached when the costs of fraud are higher than the costs to upgrade. Another aspect of the goals of the global ATM network is to increase accessibility and convenience regarding payment services itself. For example, faster transactions benefit all stakeholders: shorter waiting times for the cardholder, more customers/users can be served at the ATM/POS terminal and more financial transactions are carried out. Finally, another driver of evolution for network operators, acquiring and issuing banks is the reduction of costs concerning network usage fees, equipment development and maintenance.

Most recent advancements concerning evolution of the global ATM network have started from 1985 with the consolidation of interbank networks and payment processes (credit and debit card). In the past 10 years, the main advancements are the move to chip based payment cards, and very recently the move to contactless payment cards and mobile phone payment via Near-Field Communication (NFC).

In the future we expect that standards and specifications are further refined to fix known problems (e.g., the ISO8583 leaves many details concerning message padding open which require handling of many special cases). Currently, standards cover only the most extreme points (payment cards) of the global ATM network well. We expect that standards are going to cover more and more core

---

aspects of the interbank networks themselves to decrease interoperability efforts. Finally, the impact of decentralized virtual currencies (e.g., Bitcoin [83]) and the omnipresence of mobile computing devices (for instance, mobile phone, Google glass, …) might heavily influence the current global ATM network realization.

4.7 Emergence

In our opinion, the worldwide financial account self-service that is established by the global ATM network emerged because of the transition from many interbank networks to relatively few interconnected shared networks and because of the standardization efforts concerning accessibility (ATM terminal and payment card).

With global accessibility there comes also a detrimental emergence effect: global scale fraud. For example “In Hours, Thieves Took $45 Million in A.T.M. Scheme”, 201311: The criminals obtained 12 card numbers, hacked into the issuing financial institution and increased the floor limit. Multiple ‘cash crews’ in several countries conducted approximately 36 000 transactions within 10 hours. In this scheme over $40 Million have been stolen. Two months earlier, the same crew stole $5 million by applying the same scheme.

4.8 Governance

Governance of the global ATM network is structured hierarchically starting at governmental regulations which define only a most general set of rules and end at the interbank operators’ own network specification that refines all superordinate hierarchical level such that the interplay of the involved CSs works towards the goal of the overall SoS.

Governmental regulations represent the first hierarchical level of governance. They concern, e.g., information obligations, liabilities, and (de-)authorization rules (i.e., how long an issuing bank needs to respect hold requests on the withdrawal amount of a financial transaction before it either times out or it is transferred to the acquirer). Governmental regulations can be further decomposed to international, nation compound (e.g., U.S., European Union) and regulations established by governments in single countries.

The second hierarchical level of governance is represented by standards like EMV and PCI DSS (see Section 4.11). These worldwide standards are freely accessible and the whole payment card industry adheres to them. The major participants of the payment card industry (e.g., Visa, MasterCard, American Express, JCB, Discover, …) practically enforce the use of the standards by:

- holding violating banks or merchants liable for damages caused by fraud, and
- excluding (repeating) violators from their interbank network.

Changes of these standards are worked out and carried out by working groups or a standards council. These entities consist of members and associates in the respective standardization organizations (i.e., EMVCo and the PCI Security Standards Council).

Finally, at the lowest hierarchical level of governance there are the interbank network specifications of each operator. They are in the responsibility of the respective operator, and equipment manufacturer or software suppliers need to work with them to build compatible products.

4.9 Handling of Time

In the global ATM network there is no explicit global time requirement. All effects related to timing violations are handled. In case of a timeout during a financial transaction, the transaction is rolled back by the mechanism of reversal. In the event of continuous timeouts, there simply is (at the inconvenience of the payment cardholder) no service.

However, the local time (i.e., the time at the place of the installation) is used to check certificates (extensively used for EMV transactions) and process expiration/effective dates. Also the local time is used to initialize a pseudo random number generator for the cryptographic operations required in the ATM terminal such that transaction numbers remain unique.

Although the EMV standard sometimes mentions ‘real-time authorization’, they apparently mean that the issuing bank is contacted for online authorization and that this process has a timeout limit. However, it is in the responsibilities of the interbank network operators to specify these timeouts concerning transactions, e.g., VISA defines acquirer transaction timeout of 45 seconds.

Finally, the EMV specifications also define requirements concerning the interaction of an ATM/POS terminal and the chip based payment card. These timeouts are specified in elementary time units (etus) which depend on the frequency of the clock signal of the serial interface used for this interaction.

4.10 SECURITY

The case of ATMs is special in two ways. Firstly, it is the only case where an attacker can get direct payoff from a successful attack. Secondly, an ATM is somewhat between a control system and an IT system – most ATMs have actual PCs as their underlying hardware, though usually with quite outdated software. In addition, ATM security is the most mature of the fields we are looking into.

Most past attacks on ATMs where relatively simple and low-level. Skimming attacks add a special hardware to the ATM that copies the information of the user’s card, and records the PIN code to reuse the card elsewhere. Another popular attack based on the fact that many ATMs have a mastercode that allows entering the maintenance menu (which then allows to empty the money container). In many ATMs, the mastercode was never changed, and can be found in the user’s manual which is freely available on the internet.

A countermeasure against skimming attacks was the introduction of chip & PIN, i.e., use the chip on the card rather than the (easy to copy) magnetic stripe. While this is more complicated to break, it has been shown in 2010 that this approach too can be hacked with a slightly more complex attack.

Another recent attack has been directly on the ATM. By drilling a small hole at the point where the internal computers’ USB connection is, the attacker can enter a USB stick which forces a software update and a reboot of the machine (the operating system in question here was Windows XP, which is phased out in the IT world and no longer receives security updates). The new software included a special menu that can be called with a special 16 digit pin, which allows the attacker to empty the machine (after buying an extra code online), block the ATMs network connection (to prevent it from raising an alarm) and even clean out the malware.

ATMs and evolvability: ATMs have been around for many years now, and in this demonstrated a number of evolvability issues. The first one is the aforementioned old version of the operating systems. Another issue in banking systems is that the cryptographic primitives, e.g., the DES cipher, are deeply embedded into the design, and cannot be replaced without shutting down and updating the whole system and even replacing some hardware – thus, many banking systems still require the use of DES (or 3DES), which otherwise has been phased out years ago. Finally, ATMs have shown the issues with interoperability; many still run Windows XP. With the attempt to support a multitude of protocols in the same system, and translate messages from protocols with different security designs, researchers have shown how to strip messages of all security features [86], or to extract keys from an cryptographic coprocessor [87].

All aspects of security [57], i.e., confidentiality, integrity and availability are of concern in the global ATM network SoS. Availability is increasingly important towards the interbank network

infrastructure where availability is higher than 99% [76]. In general the availability requirements appear to get lower towards endpoint CSs of the SoS: ATM/POS terminals have the least availability whereas the infrastructure of issuing and acquiring banks as well as the interbank networks seem to have highest availability requirements (unfortunately, there is no public data available). Integrity and confidentiality is established/controlled partly by standards and partly by the interbank network specifications. For example, parts of the interbank networks are realized over public networks (Internet) and the communication is protected by Virtual Private Networks (VPNs)\(^\text{13}\). Hence, there exists end-to-end data protection, because of this involvement of public networks. Data integrity is ensured by forgery-proof signatures where no trust is required. Data confidentiality (e.g., PINs) is protected by cryptographic procedures which are periodically reevaluated and updated (e.g., with respect to key sizes) by EMVCo and the interbank operators. Additionally, there are protocols in place to update keys and certificates at CSs.

Extensive monitoring and logging is realized at most CSs to handle disputes, detect fraud, and fulfill governmental regulations.

When looking at various attack scenarios, the CSs located at the endpoints of the SoS are preferred targets:

Attackers often modify the ATM terminal to add a payment card skimmer and camera for obtaining payment card data and its associated PIN. Countermeasures are defined in the standard PCI DSS (see Section 4.11). The PCI DSS contains guidelines to prevent acquiring illegal card data and the PIN.

Another attack uses counterfeit ATM/debit card for criminal activities. To counteract this attack, the EMV standard introduces smartcards to prevent (easy) card copies.

Finally, there are physical attacks that involve the robbery directly at the ATM terminal or of the whole ATM terminal. Countermeasures for these class of attacks are the usage of a strong vault, ink-bombs, video surveillance and guards.

### 4.11 Standards

We identify the usage of the following standards (chronologically sorted):

- ISO/IEC 7811, ISO/IEC 7813: These standards concern financial transaction cards and their recording techniques and characteristics of magnetic stripe cards.
- ISO 8583: Financial transaction card originated messages -- Interchange message specifications: Only the message layout (syntax and semantics) is defined, not the routing. Routing is realized e.g., by TCP/IP networks.
- Other standards bodies: ISO/IEC 7810 (physical card characteristics), ISO/IEC 7812 (issuer identification), ISO/IEC 7816 (smartcards), ISO/IEC 14443 (contactless cards), GlobalPlatform (shared secure chip platforms), ...
- EMV (Europay, Mastercard, Visa), builds on 28 ISO standards, e.g., ISO/IEC 7816, ISO/IEC 14443, ...
  - "global interoperability and compatibility of chip-based payment cards and acceptance devices"
  - Protocol and interface standards and requirements
  - Can use (legacy) ISO 8583 compliant networks for message transport
- PCI DSS (Payment Card Industry Data Security Standard)

4.12 CONCLUSION

The global ATM network is one of oldest and largest SoSs built by mankind. It is owned and operated by billions of humans, thousands of financial institutions, and a few global interbank networks.

Throughout the whole development of the global ATM network, there is “Backward compatible” evolution. Not all CSs are at same level of evolution, but they support older technology and can still interact with each other.

Virtual decentralized currencies, and mobile computing devices might have (r)evolutionary impact on cash and ATM terminals. In case either of these get adopted by the public, there might not be a need for paper cash or ATM terminals anymore: money would be transferred solely over networks that are accessed directly by the account holders.

Current standardization efforts focus on CSs that are at the endpoints of the SoS, i.e., payment cards, and ATM/POS terminals. These SoS endpoints need to exist in great quantity while being relatively cheap and geographically densely distributed.

RUMIs are defined by standards, their interpretation, and the interbank network operators themselves. From the discussion on RUMIs, we have seen that they can be associated with a set of transactions. SoS Transactions that involve multiple CSs over complex networks appear most complex and require further investigation.

The timing requirements in this use case concern security and quality of service, but are ‘indulgent’, i.e., no hard deadlines. In case a timeout occurs, there is simply no service at that specific ATM terminal. Even though time plays a role with respect to the duration of financial transactions and expiration of security certificate, there is no high precision global time in the global ATM network SoS.

Lastly, security is about costs caused by fraud related to the costs of updating to more secure equipment. In the US, the widely spread magnetic stripe-only ATM/POS terminals have not been replaced for a long time, because fraud did not exceed these upgrade costs.

4.13 HISTORY OF THE ATM NETWORK

Table 6 gives a brief overview of the development of the global ATM network.

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1939</td>
<td>Luther George Simjian invents and patents the Bankmatic ATM which was not successful at the time.</td>
</tr>
<tr>
<td>1966</td>
<td>Debit card pilot program, Bank of Delaware.</td>
</tr>
<tr>
<td>1967</td>
<td>Barclays Bank in the UK installs the first cash dispenser.</td>
</tr>
<tr>
<td>1968</td>
<td>Don Wetzel conceives of an ATM while waiting in line at a Dallas bank.</td>
</tr>
<tr>
<td>1969</td>
<td>An ATM (produced by Docutel) is operated by New York’s Chemical Bank.</td>
</tr>
<tr>
<td>70s</td>
<td>• Magnetic stripe cards.</td>
</tr>
<tr>
<td></td>
<td>• ATMs developed that are online, take deposits, transfer money between savings and checking, provide cash advances from credit card and take payments.</td>
</tr>
<tr>
<td></td>
<td>• ATM networks established and growing, fast growing interbank networks appear.</td>
</tr>
<tr>
<td></td>
<td>• First POS terminal installations.</td>
</tr>
<tr>
<td>1977</td>
<td>• Midwest Payments Systems develops Jeanie, the first online shared ATM</td>
</tr>
</tbody>
</table>
network.
  - Microprocessor chip card invented.

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1982</td>
<td>Visa acquires Plus and starts to build national interbank networks.</td>
</tr>
<tr>
<td>1984</td>
<td>French banks begin rollout of chip cards.</td>
</tr>
<tr>
<td>1985</td>
<td>U.S. supreme court holds that ATM terminals do not represent bank branches, encourages interstate networks, consolidation of interbank networks starts.</td>
</tr>
<tr>
<td>1988</td>
<td>MasterCard acquires Cirrus.</td>
</tr>
<tr>
<td>late 80s</td>
<td>ATM systems flourish, POS languishes (merchant/bank conflicts, multiple standards).</td>
</tr>
<tr>
<td>1990</td>
<td>Plus and Cirrus enter ‘duality agreement’: enables ATM terminal owners belonging to one of the networks to serve cardholders of the other network without additional fees.</td>
</tr>
<tr>
<td>1994</td>
<td>Europay, MasterCard and Visa start work on EMV.</td>
</tr>
<tr>
<td>1997</td>
<td>EMV field trials.</td>
</tr>
<tr>
<td>1999</td>
<td>EMVCo founded.</td>
</tr>
<tr>
<td>2000 – 2010</td>
<td>Several EMV versions published, JCB joins EMVCo, American Express joins EMVCo, Contactless protocol published, 1 billion EMV cards today</td>
</tr>
<tr>
<td>today</td>
<td>Move towards contactless payment, multi-functionality cards (not only payment card), mobile phone payment.</td>
</tr>
</tbody>
</table>

Table 6 – Timeline.

4.14 Ontology

The concepts in Table 7 are obtained from [84][85] and partly adjusted for this use case.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank/Financial Institution</td>
<td>A financial institution is an institution that provides financial services for its clients or members(^{14}).</td>
</tr>
<tr>
<td>ATM terminal</td>
<td>Networked embedded systems that provide humans access to financial transactions in public places worldwide.</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number – a numeric code used to validate the individual who initiated the transaction via a payment card. The number is entered into a keypad and is encrypted to travel along with the authorization. [85]</td>
</tr>
<tr>
<td>Issuing Bank/Issuer</td>
<td>Entity that issues payment cards or performs, facilitates, or supports issuing services including but not limited to issuing banks and issuing processors [84].</td>
</tr>
<tr>
<td>EFT Network/Interbank Network/Card Scheme</td>
<td>Are the payment brands including: Visa, MasterCard, Interac Association and American Express. The payment brands establish industry operating regulations for Acquirers and Issuers to facilitate coordination with merchants and cardholders. Payment Brands in other countries also manage their proprietary networks:</td>
</tr>
</tbody>
</table>
  - Visa – VisaNet, Plus |

- THE EXCHANGE ATM network
- MasterCard – BankNet, Maestro and Cirrus
- Interac Association – Interac Network
- American Express -American Express Network

[85]

| **Payment Card** | A card that is accepted at ATM/POS and other payment terminals to conduct financial transactions. A payment card is usually tied to a single cardholder. |
| **Acquiring Bank/Acquirer** | An Acquirer is an organization authorized by the payment schemes to enable merchants to process debit and (or) credit transactions. The acquirer establishes a contractual relationship with the merchant and assigns the relevant fees/discount rates for the merchant and ensures the merchant complies with all regulations stipulated by the card schemes [85]. |
| **Financial Transaction** | The act between a cardholder and a merchant or acquirer that results in the exchange of goods or services against payment [85]. |
| **Smartcard/Chipcard** | A payment card based on a microcontroller and memory, i.e., the payment card has computing and storage capabilities. A card that utilizes a chip technology to process some applications such as, payment, loyalty and/or related transactions [85]. |
| **Cardholder/Human User** | Non-consumer or consumer customer to whom a payment card is issued to or any individual authorized to use the payment card [84]. |
| **POS terminal** | Point of Sale – Refers to a payment terminal or the device used at point of purchases within the merchant retail location [84]. |
| **Payment Processor** | A Payment Processor is contracted by an Acquirer and can provide the technology to enable a merchant to accept credit and debit card transactions and processes these transactions on behalf of the merchant and the acquirer [85]. |

*Table 7 – Concepts in the ATM use case.*
5 CRISIS MANAGEMENT

This Section briefly summarizes the findings for the SoSs in the crisis management domain of application.

The crisis management domain is characterised by multiple professional organisations (i.e., Police, Fire-brigade, Health-services, and municipality) that jointly have to configure an ad-hoc SoS to address an incident at a specific location and time. Thus, the component systems for this SoS are primarily these professional organizations. The context of the incident has large influence on the characteristics of the SoS – e.g., an incident involving a fire on a ship requires different resources and organisation, then flooding of an area by a river.

The SoS for crisis management is best characterised as a virtual SoS that wants to be collaborative [90]. Within the crisis management domain, there is no single party responsible for integration and testing of the resulting SoS. Furthermore, the SoS has characteristics of a socio-technical system: humans are active parts of the SoS, as sensors, actuators, information assessors, decision-makers, etc. (including combinations). The challenge is, at design-time, to put the mechanisms in place to have run-time reconfiguration and adaptability of the SoS. This holds for both civil and military crisis management – whereby the latter, seemingly a hierarchically controlled organisation, is nowadays forced to engage in inter-agency collaboration with various public, private and non-governmental entities.

A brief characterisation of crisis management System-of-Systems (CM-SoS) using information from The Netherlands is given below:

- The constraints for CM-SoS derive from law and regulations, the actual incident that needs mitigation, and the effects of mitigation measures.
- There is no single architecture for CM-SoS, although disaster plans provide an initial governance model, suitable for run-time (i.e., ad-hoc) configuration of the CM-SoS, depending on the characteristics and dynamics of the incident.
- Semantic of communication is a big issue in CM-SoS, where typically people and machines have to collaborate, that originate from different organisations with their own culture, doctrines, procedures, and ontologies. Currently, the pragmatic approach is to train professionals to work collaboratively, and foster understanding and awareness of differences in semantic of communication. For government-centralised systems, specific data models for e.g., GIS and population registration information are used – but not necessarily already throughout the entire country.
- The dynamicity in CM-SoS is three-fold: the configuration of the CM-SoS changes, the world (including the incident) changes, as does the interaction between these two.
- The evolution of the CM-SoS occurs on different levels. On a high level, it is directed by regulations, specific efforts of organisations. On the work floor level, it is done by connecting (sub)systems, by improving existing formats for data sharing, etc.
- One of the open challenges in CM-SoS is to detect emergent behavior at runtime, and be able to use that information for reconfiguration purposes.
- Time-criticality is typically present in CM-SoS, where the consequences of an un-mitigated incident may have large repercussions on quality of life, economy, environment, etc. The actual time criticality depends on the incident characteristics.
- Dependability in the CM-SoS is an important aspect. The most important aspects for crisis management are availability (e.g. is the fire department functional when you need it?), reliability (e.g. are cooperating subparts performing within agreed limits?) and safety (e.g., what is the risk taken by personnel when extinguishing a fire?).
- Security in crisis management scenarios often creates a difficult trade-off: the concatenation of various stakeholders (police, fire brigade, etc.) in an organisation calls for an overall security strategy that suits each partner. However, the unfolding crisis response action may call for quick and pragmatic loosening of security on certain aspects, as it could jeopardize the crisis management actions.
- Standardisation is an on-going effort in CM-SoS, which is currently best characterised by the existence of multiple standards (both at national and international level). In Crisis Management practice, standards appear to be seldom followed. Instead, improvisation based on experience leads action.
- The boundaries of CM-SoS are dynamic and difficult to delineate, as this is a highly ad-hoc and context-dependent SoS, whose configuration is not determined a priori, neither remains unchanged during the incident. This relates to dynamicity in CM-SoS.
- The degree of heterogeneity is very high in CM-SoS, which means that the constituent systems differ very much in a number of aspects. These include their goals and requirements, their functionalities and behaviors as well as the procedures and technologies they use.
- The degree of autonomy of the various constituent systems is also high, as these are typically loosely coupled and act rather independently in the field, focusing each on their goals, realm of responsibilities and way of work. Collaboration does happen, however, e.g., as organizations inform one another on an incident.

The lessons learned in the Crisis Management area are helpful in the AMADEOS SoS context, especially the Smart Grids environment. Crisis Management, when viewed as an SoS, differs in various ways from ‘traditional’ SoSs. A number of these differences are outlined in Table 8. However, the same differences can be observed in Smart Grids as well. Therefore, lessons learned from the Crisis Management area could be applicable to that field, too.

<table>
<thead>
<tr>
<th>Traditional SoS</th>
<th>Crisis Mgt./Smart Grid SoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single operator perspective</td>
<td>Multiple ‘prosumers’ perspective</td>
</tr>
<tr>
<td>Controlled use of standards</td>
<td>Risk of multitude of (proprietary) standards</td>
</tr>
<tr>
<td>Design-time configuration</td>
<td>Run-time configuration</td>
</tr>
<tr>
<td>Design-time testing</td>
<td>Integrated, large-scale crash-testing</td>
</tr>
<tr>
<td>Separate infrastructure</td>
<td>Integrated infrastructure, with emphasis on reliability and privacy, through adequate security, to guarantee sufficient levels of trust</td>
</tr>
<tr>
<td>Splendid isolation of providing organisations</td>
<td>Forced collaboration of providing organisations</td>
</tr>
</tbody>
</table>

Table 8 – Similar transitions from traditional to new SoSs showcase the relevance of findings in the Crisis Management area towards the Smart Grid area.

5.1 **INTRODUCTION**

Crisis management is a somewhat odd ‘duck’ in a pool of Systems-of-Systems discussions (SoS). The main differences between this Section and other Sections is that it does not discuss a large-scale, technical system, that might be prone to unsustainability in the future – and accompanying mitigations. Rather, this Section discusses crisis management organizations, mostly from the Dutch perspective, that are set-up to deal with large-scale disruptions. Nevertheless, there are similarities with other SoSs as well. Crisis management is done by large, ad hoc SoSs configured from existing organizations, resulting in an SoS that consists of constituent systems that are loosely coupled, and of different nature. Managing a crisis takes in the order of hours to days, which is a much shorter timespan than the other SoSs discussed within the AMADEOS context. However, the organizations that participate in the SoS to actually perform the work, are long-standing organizations that are for a large part of the time busy preparing for action during a possible disaster. Typically the preparation time is considerably larger than that of disaster repression.

Typicalities occurring in crisis management provide valuable lessons to the area of Systems-of-Systems. In order to effectively suppress a disaster, services (such as police and fire-brigade)
have to closely collaborate and inform each other. All parts within an SoS have to communicate well, in order to have the other parts functioning optimally. However, due to time pressure and stress during an incident, it is difficult to follow procedures exactly. This originates from the fact that the SoS is built up from people that make their own decisions, and each disaster has its own specific properties. This Section investigates the structure of the crisis management SoS collaboration and coordination, shedding light on the question whether emergency management organizations are well connected, and optimally suppress the disaster.

Closely related to crisis management is the military area. Since the organizations behind those areas are similar (large-scale, strongly hierarchical, performing a complex task in a limited amount of time), some lessons learned from that area will be discussed as well.

Parts of this chapter are based on own previous work and work in (EU and National) subsidised projects, especially [99], [93], [97]. Furthermore, public government sources are used, such as [95].

### 5.2 Problem Area & Example Scenarios

Crisis management situations exist along a large range. A car accident may require only monodisciplinary action (police), could be managed at the location of the incident, and does not call for a hierarchically formed crisis management SoS. It is an emergency, not a crisis. Yet – emergencies may have unpredictable effects, therefore although some incident may start as ‘routine’, it may emerge into a larger-scale incident, requiring more effort from different disciplines. In this case, the subsequent SoS must dynamically adapt and reconfigure itself.

Some incidents start ‘big’, e.g., a large fire in a health-care centre requires much more in terms of coordination, and all disciplines are present: fire brigade, police, health services and the municipality. The largest crises have a major impact, “entailing a threat to the core values of a system or the functioning of life-sustaining systems, which must be urgently dealt with under conditions of deep uncertainty” [96]. Examples of such large-scale breakdowns include floods, (chemical/nuclear) plant disasters, breakdowns of critical infrastructure(s), and natural disasters such as tsunamis and earthquakes. An enormous amount of tasks has to be carried out by crisis management personnel in such circumstances, such as keeping order, network maintenance, rescuing people, extinguishing fires, providing logistics, etc.

### 5.3 Constraints / Quality Metrics

Constraints on crisis management services fall into two different categories, namely hard constraints and soft constraints (e.g., quality metrics). Therefore this section jointly addresses the Constraints and Quality metrics viewpoints.

The constraints that are put on crisis management services may be difficult to provide. In ‘normal’ emergency situations (small fires, accidents, 911/112 calls), response times may be imposed by law upon the services. In the Netherlands, these laws have been set nationwide; it is the time between a call and a service being on-site. For an ambulance, this is 15 minutes; for the fire department, 8 minutes; for police, 15 minutes. In practice, however, services seldom succeed in arriving on time, which costs lives and other damage (environmental, economic, …). Police and ambulance take more than 15 minutes to arrive about 10% of the time. Due to cost cutting, the fire department will increase its response times to 15 minutes as well.

However, in large-scale crisis scenarios, a large ad-hoc SoS-organization has to be built up. The speed of setting up such an SoS greatly depends on the crisis scale and location – to name a few factors. Furthermore, assessment of the seriousness of the situation is important to decide on scaling up; this is usually done only after professionals arrive at the scene.

Apart from time constraints, the availability of personnel and material (such as cars, ambulances, etc.) and refuge locations (e.g., hospitals) are important, and are as difficult to prescribe as time constraints, since these are greatly dependent on scale and location as well.
Furthermore, constraints are imposed by the location of the incident, and the progression of the incident. For example— a fire in a building with a (possibly harmful or toxic) plume blowing towards an inhabited area, quickly acquires additional constraints (e.g., on available time for informing the public on sheltering actions, on evacuation of vulnerable people, etc.). When crisis management takes places in a complex environment, every action may result in additional constraints. Typically, given plumes across shipping-lanes, these lanes are (temporarily) closed, which may impose constraints on e.g., availability of water-transportation of resources to mitigate the incident.

For the SoS itself, constraints are also imposed. It is currently difficult to construct the SoS to effectively include ‘non-standard’ organisations. That is, it is difficult to include, e.g., companies, inhabitants, and the general public, into the SoS. It may be beneficial to do so— e.g., consider having an easier access to possible transportation means by contacting local taxi/bus/freight companies, but also contact general audience who typically bring lightly wounded people to first-aid posts in hospitals. Also, in a large-scale crisis event, citizens can often provide valuable information about a developing crisis. For example, in [100] citizens can be used to detect and localize (harmful) gases that were inadvertently released into the atmosphere. Citizens can report their olfactory observations by calling an environmental protection agency that can then reason about the type of gas and possible source (e.g., factory) of the gas release.

All of this shows that it is difficult to impose (hard) constraints in the area of Crisis Management. The same may therefore be true for ‘soft constraints’, or optimality/quality criteria. These are the metrics that have to be satisfied as much as possible. Examples for the Crisis Management area include response times (as short as possible), time needed to set up a CM organisation (a little as possible), number of properties/lives saved, level of security maintained, etc. Introducing such and more detailed quality metrics will enable better assessment after a crisis, as it helps setting goals for CM personnel, and may in the future enable more explicit hard constraints as well. Note that the quality metrics are closely related to (hard) constraints: constraints usually have a ‘soft’ counterpart as a quality metric. E.g., the hard constraint “be on scene within 15 minutes” has the quality metric counterpart “minimize the time of arrival on the scene”.

5.4 ARCHITECTURE

There is no general SoS architecture for crisis management in The Netherlands. Basically, the ‘disaster plans’, and technical preparations form an ‘activation pattern’, to make resources become available for self-configuration into an SoS for a specific incident. A characterisation of an SoS for crisis management is as a socio-technical system. For the crisis management SoS, which is configured ad-hoc, the ‘architecture’ is closely related to the ‘governance’ (see subSection 5.5).
5.4.1 Safety Chain

In the Netherlands a so-called safety chain is used, see Figure 14. It is a cyclic chain, used to describe different phases of emergency management. The pre-phase consists of pro-action, prevention and preparation. This part of the chain concerns the prevention of incidents, taking away obstacles that could create unsafe situations and optimally preparing for an eventual incident. The latter part is the most interesting, since the preparation phase decides on the actual collaboration during an incident, what the mutual relations are between services, and how they influence operations. Most of this work is done in work groups, team meetings, etc.

The repression is the phase in which the incident is actually ongoing and the actual SoS is configured. Services act as quickly as possible with the least possible damage. Information, collaboration and coordination are important here, since they directly influence the speed and accuracy of all actions.

After repression, a post-phase exists, in which the incident is reduced to the preferred state, and in which evaluations are done about the incident. Lessons learned are used in the pre-phase to better organize emergency management for future incidents, thus allowing the SoS to evolve by e.g., changing its mode or organization into a more effective and efficient one.

5.4.2 Responsibilities

The four emergency management services (Police, Fire department, Municipality, and Health services) that together configure an SoS (therefore being the typical CSs in the CM-SoS) have a number of duties imposed on them by law during a crisis (see also subSection 5.5). These are listed in Table 9, Table 10, Table 11 and Table 12.

<table>
<thead>
<tr>
<th>Police</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Keeping order</td>
<td>Crowd mgmt., crowd control, riot control</td>
</tr>
<tr>
<td>2. Tracking</td>
<td>Tactical tracking, Technical forensic tracking, intelligence and investigation</td>
</tr>
<tr>
<td>3. Tracking expertise</td>
<td>Police negotiating, specialized observation, specialized applications of investigation, specialized forensic tracking, arresting and support.</td>
</tr>
<tr>
<td>4. Intervention</td>
<td>Police negotiating, specialized observation, specialized operations,</td>
</tr>
</tbody>
</table>
explosive assessments, arresting, support.

5. **Network maintenance**
   - Maintaining society networks, maintaining other networks and emerging groups

6. **Security and protection**
   - Security and protection of people, objects and own service

7. **Mobility**
   - Dynamic traffic management, Static traffic management, Traffic control and tracking

8. **Information management**
   - Providing information

9. **Support**
   - General support

### Table 9 – Police responsibilities.

<table>
<thead>
<tr>
<th>Fire department</th>
<th>Fire suppression, incident management hazardous goods and decontamination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Source and emission suppression</td>
<td>Fire suppression, incident management hazardous goods and decontamination</td>
</tr>
<tr>
<td>2. Rescue</td>
<td>Technical assistance, rescuing, urban search and rescue</td>
</tr>
<tr>
<td>3. Decontamination</td>
<td>Decontamination human and animal, vehicles, infrastructure</td>
</tr>
<tr>
<td>4. Information management</td>
<td>Providing information</td>
</tr>
<tr>
<td>5. Logistics and support</td>
<td></td>
</tr>
</tbody>
</table>

### Table 10 – Fire department responsibilities.

<table>
<thead>
<tr>
<th>Municipality</th>
<th>Providing information to press, public, relatives, networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Communication</td>
<td>Providing information to press, public, relatives, networks</td>
</tr>
<tr>
<td>2. Public care</td>
<td>Evacuation of human and animal, primarily life needs, post-mortem care</td>
</tr>
<tr>
<td>3. Environmental care</td>
<td>Environmental hygiene, management of water, nature, landscape, building and space.</td>
</tr>
<tr>
<td>4. Information</td>
<td>CRIB (information on persons), CRAS (information on damage to material), filing information</td>
</tr>
<tr>
<td>5. Support</td>
<td>Supporting management, prepare the post-phase</td>
</tr>
</tbody>
</table>

### Table 11 – Municipality responsibilities.

<table>
<thead>
<tr>
<th>Health services</th>
<th>Triage, caring, transporting, relaying</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Medical aid</td>
<td>Triage, caring, transporting, relaying</td>
</tr>
<tr>
<td>2. Psychosocial aid</td>
<td>Signal affected people, promote self-help, relaying affected people</td>
</tr>
<tr>
<td>3. Prevention</td>
<td></td>
</tr>
<tr>
<td>4. Health investigation</td>
<td>Monitor public health, investigation with groups and individuals</td>
</tr>
<tr>
<td>5. Counter infection</td>
<td>Tracking source and contact, protective measures, hygiene measures, isolation, quarantine</td>
</tr>
<tr>
<td>6. Information</td>
<td>Providing information</td>
</tr>
<tr>
<td>7. Logistics and support</td>
<td></td>
</tr>
</tbody>
</table>

### Table 12 – Health services responsibilities.
The message interfaces between the various hierarchy levels and various different services are related to the governance structure of the organization, which is treated in the next subSection.

5.5 **GOVERNANCE**

In general, the government of The Netherlands, at national, provincial and municipal levels, regulate the professional organisations that have to form the ad-hoc SoS. The SoS is also subject to regulation, although that is (currently) expressed indirectly, through regulations on the individual services/organisations that contribute to the SoS. To enhance formation of an SoS for incident management, the country has been divided into safety regions, 25 of them, corresponding to geography, population, industry and other important assets in The Netherlands.

![Figure 15 – The Crisis Management Structure in the Netherlands](image)

Figure 15 shows the crisis management structure in the Netherlands. At a policy level, the National Crisis Centre (NCC) manages information between the Dutch Government ministries and the crisis management centre (Landelijk Operationeel Coördinatiecentrum - LOCC). The LOCC has operational control in large (GRIP 3 and GRIP 4, see Section 5.5.2) crises. However, decision-making responsibility resides with the Mayor. The internal structure of the LOCC reflects the major agencies in each safety region. It is made up of officers from each of these agencies. These officers coordinate LOCC decisions with the local representatives of their organisation during a crisis.

5.5.1 **During Pre-phase**

About 90% of the time, the services (i.e., professional organisations) are in ‘cold phase’. There is no incident, but each organisation is practicing, making policies, etc. It is difficult to chart the collaboration/coordination into an SoS, as there are no fixed procedures and each safety region works differently. For the safety region Haaglanden (i.e., the region around, and including, the city of The Hague), the pre-phase consists of:

- Meetings of the safety directorate, general management, daily management, meetings of regional management, meetings of domain owners. The meetings decide on the collaborative modes on a strategic level.
- Projects, such as Program ‘Increased Disaster Suppression and Crisis Management’. This project is carried out to make the services better collaborate and reinforce each other’s processes.
- Creating a regional strategic policy. These policies are decided by law, and have to meet certain criteria, like detailing national goals, containing a strategic policy on practicing, and requirements on the information.
- Maintaining contacts.

The idea behind detailing these tasks in laws is that it stimulates collaboration and coordination. These tasks are on a strategic level, leading to agreements on an operational level in the individual organisations. These agreements are mostly related to training and practicing and material use. Examples include:

- Development of a fire department motor bike;
- Operational practice drills such as the national drill ‘Indian Summer’ in 2011\(^{15}\).

5.5.2 **During repression phase**

In the repression phase, there is collaboration and coordination as an SoS during incidents, where the structure is agreed upon in the pre-phase.

- The first thing in the repression phase is the GRIP-procedure. GRIP means *Gecoördineerde Regionale Incidentenbestrijding Procedure* (Coordinated Regional Incident-management Procedure). It consists of five categories. 0 are the simple incidents, such as car accidents or small fires. The worse the incident gets, the more strategic coordination becomes necessary. The SoS is reconfigured as the services are scaled up accordingly. There may be extra services involved when scaling up, such as the National Crisis Centre leader and the National Operational Coordination Centre leader. Table 13 shows coordination and collaboration during repression, as it has been detailed in the law.

<table>
<thead>
<tr>
<th>Monodisciplinary action</th>
<th>Coordinative action on site</th>
<th>Coordinative action general</th>
<th>Formulating municipal policy</th>
<th>Formulating regional policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>‘bonnetmeeting’ (i.e., at place of incident informal meeting)</td>
<td>1 Command on site</td>
<td>2 Regional coordination centre</td>
<td>3 Municipal policy centre</td>
</tr>
<tr>
<td>Fire dept. Officer on duty</td>
<td>Fire dept. Officer on duty</td>
<td>Fire dept. Officer on duty</td>
<td>Fire dept. Regional commander on duty; Municipal commander</td>
<td>Fire dept. Regional commander on duty</td>
</tr>
<tr>
<td>Police Officer on duty</td>
<td>Police Commander on duty, communication advisor</td>
<td>Police General commander</td>
<td>Police District head</td>
<td>Police Head of corps</td>
</tr>
<tr>
<td>Health service</td>
<td>Health service Commander on</td>
<td>Health service</td>
<td>Health service Regional health</td>
<td>Health service Regional health</td>
</tr>
</tbody>
</table>

### Table 13 – GRIP (level of severity of incident) related responsibilities.

- GRIP 1: Suppression at the source. Incident of limited scale. Agreement on various disciplines necessary.
- GRIP 2: Suppression of source and effect. Incident with clear effects on the surroundings.
- GRIP 3: Wellbeing of large groups of people within a municipality at danger.
- GRIP 4: Beyond municipal border, or threat of emergence to larger-scale/broader area, or scarceness of primary needs.

- During an incident, there are processes for which plans have been made, such as clearing a building and evacuating, in which the services have shared interaction processes.
- Countrywide Crisis Management System (LCMS, ‘landelijk crisis management systeem’): a communication system during an incident where strategic professionals can provide each other with information.
- Additionally, the LOCC manages resources when an incident grows beyond the capabilities of a single safety region.
- Response times: in the repression phase, keeping the times by which services are present low, is essential. These response times have been set by law.
- C2000: The communication system by which (operational) services can communicate during the repression phase. The system is maintained by the public safety answering point.

From this outset, the communication should run perfectly smooth. In practice, however, the experience is that the control rooms (‘meldkamers’) point (which answers 112 calls) warns the appropriate services. However, the services that are directed from a control room, do not cooperate in the field, remaining apart as much as possible. This may not have a (large) negative effect on the incident, but cooperation might increase the performance of the incident management. Furthermore, the provision of information could easily be enhanced by better cooperation. Therefore, optimizing the communication between various stakeholders involves more than effectuating the right organizational charts for the SoS.

### 5.5.3 Governance in Military Action Planning

Military decision making can be seen as an area closely related to (civil) crisis management: a large, hierarchically structured, heterogeneous organization has to perform a specific task that is usually unique, under great stress and time pressure. However, the decision making philosophy behind military actions does not always perfectly suit the nature of those actions.

For military planning, the NATO-standard Operational Decision-Making Process (ODMP) is taught to officer cadets of all military services in many nations, including The Netherlands. It is designed for hierarchical organisations, with clear goals, and rational decision makers. Characteristics of the ODMP are shown in Figure 16:

- It emphasises the planning process before an operation begins.
- It assumes a hierarchical organisation.
- It is a successive, top-down decomposition process.
• Planning at any given level can only begin when planning is complete at the next level up. Warning orders allow some overlapping.
• Synchronisation is achieved by resolving conflicts at the next level up, rather than peer-to-peer. Liaison officers allow some cross-hierarchy information flow.
• There is no specific provision for multi-disciplinary collaboration.
• It assumes rational decision makers. While option selection is not exhaustive, decision makers are advised to consider three own courses of action (COAs) and three enemy COAs.
• Step 7 enables limited projection by war-gaming each own COA against each enemy COA. Projection is limited because neither COA is altered by confrontation with the opposing COA.
• Decision makers may partially tailor the process. This makes compliance and interoperability dependent on shared terminology, training, culture, and infrastructure.

![Military action planning process](image)

**Figure 16 – Military action planning process.**

In Section 5.7, we will see that the strictly hierarchical organisation of this planning process falls short of the ideal in actual operations.

5.5.4 **Governance: Scaling Up and Down**

Incidents can grow (or shrink) during the lifetime of the incident. For example, what began as a simple car accident (GRIP 0) can grow if a chemical tanker also becomes involved. As incidents change scale (measured in GRIP levels), the governance of the incident also changes. This can be a minor change in the lower GRIP levels but when an incident grows very large, the number of people and the number of systems involved also grows. For example, in a GRIP 3 situation, the...
Mayor of a municipality is in command. However, if the incident now involves a bigger municipality (GRIP 4 situation), the Mayor of the larger municipality assumes command. The system must support this dynamicity of governance.

5.6 **SEMANTIC OF COMMUNICATION**

Communication between military or crisis management entities preferably takes place using a standardized language, of which all key players have a strong active and passive command. The difficulty with large, heterogeneous organizations is that this ‘common interface’ is not present. In the crisis management area this is clearly visible, since the various parties speak languages and have a strongly different culture (e.g., the fire department talks about *units* when the police say *agents* and the health services refer to *people*). Furthermore, this may also occur within single organisations: due to e.g., local contextual differences, the semantic grounding of ‘large fire’ may be different. Furthermore, standardization on semantic of communication also becomes an international issue – given borders between countries. In the military environment, semantic interoperability plays a role as soon as multiple branches have to cooperate (navy, army, air force, etc.) or when an operation is carried out by an international force, which is more and more the case in worldwide conflicts nowadays. Furthermore, the tasks of the military are extended towards inter-agency collaboration, including collaboration with NGOs (non-governmental organisations), local organisations & companies in an area, etc.

Various efforts have been undertaken to reduce the semantic gap that separates various branches in a crisis management organization, which will come to more close attention in the subSection 5.8.

5.7 **DYNAMICITY**

Both crisis management and military decision-making have to deal with (short term) dynamicity. As the problem changes continuously, the Crisis Management SoS has to change as well. A good high level example of dynamicity has been given in Section 5.5.2: increasing severity of the crisis at hand requires the CM organisation to scale up to higher GRIP-levels.

At the current organisational level, continuous adaptation may call for more elaborate fine-tuning of the CM organisation. Newly encountered problems may call for action that was not foreseen. For example, panic reactions within the population may call for advanced crowd management skills that cannot be performed by police personnel currently on location that were only sent there for generic traffic control, and may not have the number of people or the skills to perform crowd management.

At the lowest levels, that of commanders and personnel at the action scene, dynamicity calls for a type of decision making that is not usually taught or investigated, as the next two subSections show.

5.7.1 **Problems with Decision Making under Dynamic Circumstances**

In military and crisis management scenarios, operational decision-making has many of the characteristics of the rational decision-making model. The ‘rational’ decision-maker gathers an exhaustive set of information, evaluates each option against evaluation criteria, compares the set of options against one another, and selects the best option without regard to previous or future problems. Scientific analysis of rational decision-making centres on option evaluation and selection.

The attraction of the rational decision-making model is that it has been proven to be optimal. Its main disadvantage is that it is a time-consuming process. Moreover, optimality is dependent on the completeness and correctness of the input information. However, this is hardly ever encountered in crisis/combats situations, and as the future cannot be predicted, as of yet, completeness of information can never be gained. Furthermore, the evaluation criteria should yield comparable results, otherwise no sound comparison can be made. In real life however, different criteria are of different types, and consequently cannot be compared without the use of some translation to
comparable amounts. Finally, the process assumes an unchanging world, whereas the real world usually changes during the decision-making process. Enemy movements, for instance, will cause preventive movements on our side, possibly resulting in obsolete plans.

As a consequence, the theoretical guidelines are seldom followed strictly. [92] investigated 101 operational situations in The Netherlands army where difficult decisions had to be made, and found that the model was properly used in only seven cases. In the majority of the cases, a straightforward decision was made based on experience. Hence, theory and practice differ greatly in most crisis response situations.

5.7.2 Recognitional Planning Model (RPM) in Dynamic Decision Making

In real-life, decision makers often have to make up their goals as they go, as their problem is often ill-defined and ill-structured.

[94] and his colleagues studied naturalistic decision-making (NDM) in a wide range of crisis management situations. They showed that expert decision-makers working in their real environments did not follow the rational decision-making model. A major incentive for them to start searching for a new decision-making model was a fire fighter's classic remark: "I don't make decisions. I don't remember when I've ever made a decision." ([94], p. 11).

Klein's recognition-primed decision-making (RPDM) model revolves around a number of main features. One is the power of experience – i.e., the experts' ability to match situations to templates developed from previous experience. A template consists of a set of cues for matching, a course of action to perform if matching succeeds, a set of expectations to check that the course of action is proceeding correctly, and a set of plausible goals. Another feature is the power of mental simulation – i.e., experts' ability to make a mental movie to play a course of action from start to end. In NDM, the scientific focus is on situation assessment, not on option evaluation and selection. The decision maker is primed to act with incomplete information, not to wait for complete analyses.

RPDM was developed as a descriptive approach to crisis response decision-making. Subsequently, [98] derived a prescriptive model of planning, known as the "Recognitional Planning Model" (RPM). It features lessons learned from the previous research. For example, war-gaming is introduced to gain the experience needed to cope with difficult decision-making situations, and to rehearse the planned course of action.

CM-SoSs involve naturalistic decision making processes (as they are prevalent in the domain), thus they are intrinsically dynamic and they need to adapt to changing conditions.

5.8 Evolution

Both in military decision-making and in crisis management the insight has dawned that a new mode of organization and/or training is necessary to increase effectiveness in targeted scenarios. In Section 5.4.1, we have already seen how closing the loop of the safety chain aims at increasing optimality of the organisation over a longer period of time. However, in Section 5.5.2, we have also seen that much of the performance increase comes from optimising the communication between crisis management entities, and that such optimisation cannot take place by simply putting through yet another organisational structure. For such virtual SoSs, prescribing new modes of organization – which basically define the system architecture at the time the SoS must be in place – is at the heart of evolution, which refers to the long-term changes in the SoS.

In the military environment, a shift is occurring that has been called 'power to the edge' [88]. Instead of top-down coordination of actions, both top-down and bottom-up initiatives are welcomed. The notion that the agents in the field have a stronger (local) awareness of the problem, has led to the delegation of responsibilities to that edge, leading to a so-called 'edge-organization'. This development has been fuelled also by changes in the problem to be solved. In the cold-war era, armies were training for a possible tank war along the iron curtain. Strategic weaponry (nuclear armament) was used by both parties to scare the other to not start a war. Current military
conflicts are much more asymmetric in nature. Invasions into much weaker countries (US-Iraq), unequal enemies (high-tech vs. Taliban; high-tech navies vs. Somali pirates). Since armies are trained and outfitted for the very long-term, such a professional organization is difficult to redirect. Evolution and agility towards unknown enemies is therefore not yet something that is standard-issue in defence environments.

In the crisis management domain, a number of external reasons can be seen as drivers for evolution. They are discussed in the next subSection. The two subSections after that discuss several initiatives that are taken to make the organisation evolve, and a field study on the current effects of these initiatives.

5.8.1 Drivers for Evolution in Crisis Management

A particularity of the CM-SoS is that in order to evolve its mode of organization, it is necessary to change the regulations and procedures of the pre-phase, which prescribe the SoS configuration at run time. Changing regulations and procedures alone, however, is not enough as to have a new mode of organisation, the constituent organizations (e.g., Police, Fire Brigade, etc.) need also to change. For instance, they all need to have more willingness to cooperate or to adopt a new communication technology that is to be used in the SoS context.

In general, evolution in CM-SoS is required in light of – or may happen as a consequence of – a number of changes in society and in the environment (i.e., external factors), such as:

- **New safety regulations / New legal requirements**: the CM-SoS should evolve its mode of organization and realm of responsibilities to comply with new safety regulations and / or legal requirements, which reflect changes in society at large.

- **New technologies**: the CM-SoS can evolve as a result of availability of new technologies. For instance, the availability of proven technology that supports communication and cooperation between organizations can lead the CM-SoS to evolve towards a more closely cooperating SoS.

- **Increased citizen participation**: the CM-SoS can and should evolve its mode of communication and organization to be able to handle and incorporate citizen’s information input as well as help in incident handling.

- **New paradigms of communication, e.g., social media**: the CM-SoS can evolve as a result of availability of new paradigms of communication, as it can take into account, and in some cases even adopt, these new modes of communication.

- **New threats**: the CM-SoS should evolve its mode of organization and communication as well as its realm of responsibilities to handle new types of threats, e.g., incidents / crisis caused by terrorist attacks or by social media information propagation.

- **New standards**: the CM-SoS should evolve so as to comply with and take advantage of new standardization initiatives.

Some of the relevant external factors, e.g., new safety regulations and legal requirements as well as standards, may be derived from lessons learned, which in turn are typically derived from previous incidents’ evaluation. Lessons learned might also give rise to other requirements, e.g., performance or efficiency requirements, which are also important drivers for CM-SoS evolution, promoting changes in the prescribed procedures and modes of cooperation. For instance, the emergency services that are directed from a control room to handle an incident typically act separately. i.e., they do not cooperate very strongly in the field. In order to increase the overall performance of the incident management process, however, the CM-SoS may evolve towards more close cooperation between the involved organizations.
5.8.2 Nationwide Initiatives for Evolution in Crisis Management Organisations

Organisations participating in crisis management SoS strive towards a system in which multidisciplinary cooperation is enhanced. This already happens by having people of the different branches regularly meet together in the same room during the pre-phase. During the repression phase, however, people often revert to well-known work modes that are not directed at multidisciplinary cooperation.

Besides the fact that assessing information needs is a complex issue, there are developments in society that change the needs for information. Examples of those are globalisation, relatively smaller distances for human contact, decreasing physical contact between people, extremism, population increase, technology advances and climate change. There are other examples that show that society changes. Change is not bad, but it makes the information needs unstable, and not one solution exactly fits and will stay fitting. New solutions to information needs currently in development are:

- LCMS, Countrywide Crisis Management System, see Section 5.5.2.
- C2000, a communication system, see Section 5.5.2.
- GRIP, Coordinated Regional Incident-management Procedure, see Section 5.5.2.
- GIS, Geographical Information systems. These are, for example, present in fire engines, and give actual maps of the neighbourhood, including information such as water supply.
- Communal public answering points. Those bring together ambulance care, the fire and police departments. They build up a multidisciplinary information image, they start LCMS, and they warn all stakeholders.
- A program for education, training, and exercise that also evaluates information needs.
- Much attention is spent on risk and crisis communications to inform the people as well as possible, before, during and after an incident (of any scope).
- Protocols with designated actors to increase clear agreements on who is to do what during an incident
- Projects to increase multidisciplinary cooperation.

Some of these initiatives bear more fruit than others. In the next subSection, experiences from the field with some of these measures are discussed.

5.8.3 Field study: Veiligheidsregio Haaglanden

In the The Hague Safety Region (VRH, Veiligheids Regio Haaglanden), a project “Programma Versterking Rampenbestrijding en Crisisbeheersing” (Program for the enhancement of disaster repression and crisis management) is running. This project was conceived after the safety regions were put into place. All domains that are handled cooperatively are treated in this project. The main advantage is that people from different organisations get to know each other, increasing communication and trust. In this project, the domains are: warning and scaling up, information management, management and coordination, risk management, policymaking, education & training & exercises. Within these domains, there are subprojects aimed at multidisciplinary crisis management. This project has contributed most to increasing trust and understanding of mutual organisations.

Within the VRH, policy has been written arising from the cooperation agreement. This part has been imposed by law upon the safety regions. Not described (or prescribed) by theory, professionals from various services spend much time on contact maintenance, such as network meetings. This increases trust, and makes it possible for people to reach each other more easily.

Another example of cross-branch cooperation has been the LCMS system. However, this system does not currently live up to its promises. LCMS is akin to an excel sheet, with different tabs for each service. The web application is accessible to anyone. Other safety regions can look on, but not alter information. LCMS is used in GRIP1 or higher GRIP situations, but not regularly. When an incident happens, LCMS is started up within 30 minutes, the same time the CoPi (Commander On
Site) has to be present. Each user account that logs in to the system can see what has not yet been read. Improvements in LCMS are possible, since there is often too much text, which makes the system difficult to oversee. Furthermore, a standardized method of working with this system does not yet exist, leading to multiple usage modes. To counteract this, courses have been set up. Lastly, when the system crashes, the shared information is lost and officers have to revert to using pen and paper.

Various other systems (such as Geographical Information Systems (GIS) and C2000) have seen the light, but are struggling with issues such as limited availability and low trustworthiness. ‘When there is no will, there will be no way’. The services have to want to cooperate each by themselves, not because a law imposes it on them.

Regarding the emergency services, the situation is as follows.

The Fire Department is a complex organisation, mostly because it consists for 80% of volunteers. The fire department has worked alone for years, mostly focusing on the pre-phase of crisis management. Within the safety region, this is where their strong point lies. On a tactical level, they are not inclined to cooperate. Incidents that occur often are reacted to in an automatic fashion. They think they are entitled to work on their own even in less-frequent incidents.

Like the fire department, the Police department is a large organisation, with the speciality that they are quickly on-site and have a mobilising role. Furthermore, the police organisation has a strong sense of hierarchy. The police do not cooperate in policy making for the safety region, but they do cooperate. The police firstly focus on their own organisation, and the safety region comes secondly. This is aggravated by the current change in police regions.

GHOR (Public Health Service) is a rather new and small(er) organisation. They were created when multidisciplinary cooperation became popular. Therefore, they are very inclined to collaborate, maintaining a health network. However, the GHOR lacks people (resources), and is therefore somewhat disabled in setting up collaboration.

Multiple Municipalities make up a region. They are also inclined to focus on their own community. Often, they have a small group working on the theme of the safety region, however, that team may be confronted with conflicting demands from the own municipality and the region. However, as a rule, municipalities are inclined to cooperate, and the mayors actively promote this by taking part in the general direction of the safety region.

All in all, these professional organisations want to cooperate, but due to various reasons of their own, it is (still) difficult to do so. The problem is postponed, leading to an underexposure of the importance of the pre-phase. During repression, the need to cooperate is felt by all involved – but this may be of relatively short duration.

In conclusion, various initiatives have been undertaken to increase to cooperation between various organisations in crisis management. These initiatives are slowed by strong cultures in the various organisations. Also, technical systems to enhance crisis management effectiveness suffer from issues that make crisis management personnel wary of using them.

5.9 Emergence

Emergence concerns phenomena that manifest at the macro-level and are not observable at the micro-level. Whether emergent behaviour is considered to be positive, negative or neutral, is dependent on the observer’s criteria. In general, emergent behaviour is not a-priori positive or negative. Emergent behaviours are likely to increase (in number and severity) as a system becomes more complex. Systems of systems are often considered complex, in the sense that it is impossible to make a precise descriptive or predictive model that relates local interactions to global behaviour. Emergent situations occur continuously in the repression phase of a crisis. This manifests itself in various ways, as the following three examples show:

First, on a strategic level, policy makers are obliged to work together. However, the policy makers tend to overemphasize the importance of their own service, instead of focusing on a common goal.
Furthermore, policy makers are often very busy, leading to the fact that they lack preparation when showing up at an incident, and they lack experience, leading to confusion. In both cases, the organisational structure is considered to be well suited to the problem at hand, but behavioural typicalities lead to the emergence of a different system performance than expected.

Second, at an operational/tactical level: For the various services, not much change seems to have occurred organisationally. However, they are now coordinated from a joint control room and have to practice more often together. When an incident happens, a lot of action boils down to improvising and falling back to known practices, and those practices do not involve cooperation with other services. So again, even though the organisational structure seems fine, the actions being taken do not yet reflect the envisioned change. Initiatives to mitigate this negative emergence have been taken, to some success, as discussed in Section 5.8.

Lastly, evolution of the organisation may lead to emergent behaviour as well. For instance, a single professional organization, e.g., the Police, may adopt a new available communication technology, which is not adopted by the other organizations, leading to problems in communication between these organizations. These are undesirable emergent behaviours in the CM-SoS, caused by the SoS evolution.

Therefore, since it appears very difficult to predict the emergent behaviours of a crisis management organization, the pre-phase focus should be less on organisational design, and more on preparing for runtime configuration.

5.10 HANDLING OF TIME

Time is an interesting issue for SoS for crisis management. In general, time-criticality is dependent on the incident. For example, a possible flooding by prospected higher water-levels in a river, dictates actions on a longer time-scale than an ongoing fire at a factory. Time criticality and global time are of importance in crisis management, also in e.g., the aftermath to establish guilt, liability, etc. As we have seen in Section 5.3, timeliness is an important aspect in defining both hard constraints and quality metrics.

As yet – we are not aware of specific protocols to ensure global time within the entire SoS; although local systems will use their own clocks for audit logs and time stamps.

5.11 DEPENDABILITY

A dependable organisation is an organisation that is available, reliable, safe, secure and maintainable. The dependability of the organisation as a whole depends greatly on the subparts. Within crisis management, an interesting conflict occurs when it comes to dependability. Locally, the professionals in the field should strive for as much independence as possible, to be able to continue their work without hindrance of other parties involved. What they therefore do is optimising their own behaviour by assuming low dependability of their organisational counterparts.

However, for crisis management on a larger scale, cooperation has to take place. Therefore, joint control rooms and command posts where commanders are obliged to sit together are introduced, for which they are trained. Seen from that angle, the assumed level of dependability in crisis management is great, the main care is that various branches do not exploit the use of other branches as best as possible. Furthermore, in larger incidents (e.g., GRIP 3), these branches are permanently collocated in the national coordination centre. This improves knowledge of each other’s capabilities and therefore increases trust in the dependability of other branches.

For crisis management sub-parts and for the crisis management organisation as a whole, two of the dependability factors are most important:

- Availability: the proportion of time that the subpart or organisation can function as it is meant to do,
- Reliability: The performance of the subpart or organisation to the constraints of time and output that were agreed upon.

- Safety: are the consequences of actions to environment and personnel within limits? By nature, this aspect is of paramount importance in Crisis Management (e.g., the safety of firefighters when extinguishing a fire).

Security within crisis management is treated in the next subsection.

### 5.12 Security

Security in crisis management is a complex issue – all of the aspects of security, authorisation, availability, authentication, confidentiality, integrity etc. are required, to a greater or lesser amount, throughout a crisis management incident handling. As always, security is a trade-off – in order to maintain complete confidentiality, information may be prevented from being exchanged. This can lead to problems when managing an incident. For example, when dealing with a fire in a company's factory, knowledge of the contents of the building could be vital to the fire service. However, on an ordinary day, this information is confidential and would not be shared.

Crisis management tends to consist of an evolving set of agencies and actors who form based on the situation. Each actor/agency has their own set of security requirements. Therefore, the SoS that represents this collaboration must reflect the requirements of each component system. The trade-off between the absolute requirements of each agency's security policy and the requirement to solve an incident can lead to conflicts. This trade-off must be managed and, more importantly, must be explainable, after the conclusion of the incident. Allowing the incident responders to make these trade-off decisions is a vital characteristic of a good crisis management SoS.

### 5.13 Standards

Standardisation is a difficult issue in crisis management – see also the Section on governance. Basically, many (local and/or international) standards are in use, warranting the need for 'wrappers' or 'interoperability engines'. Given the federated nature of crisis management organisations (i.e., not acknowledging one party to be the authority to govern them all) it is expected that multiple standards will co-exist – also with respect to cross-border cooperation for crisis management.

### 5.14 Recent History of Crisis Management Systems

In the Netherlands in the 1990s, the professional crisis management organisations scaled up from local to regional level, leading to the creation of police regions and safety regions (‘veiligheidsregio’). In the 21st century, emergency management organizations are increasingly aware of the importance of collaboration and coordination. This is in part due to two large disasters, and lessons learned there, namely the Volendam New Year’s Fire (2000/2001: [89]) and the Enschede Fireworks Disaster (2000: [91]). Based on a large amount of critique in the handling of those disasters, the law on “quality increase in disaster management” was effectuated.

As of 2009, emergency management organisations have been improving their collaboration. 25 safety regions have been created. Each is a cooperative organization in which multiple stakeholders collaborate in emergency management. The head of such a safety region is the mayor of the largest city, and the other mayors take place in the council. Furthermore, police regions and control rooms are reduced from 25 to 10. Remarkably, the 10 police regions overlap the 25 safety regions.

The future will see emergency management organisations aiming at still more collaboration, primarily in making it happen, since much of the awareness has been generated in the previous period. Currently, the ideal scale for this is investigated. By increasing the size of regions, emergency organizations try to become more multi-functionally operational.
6  ENERGY DOMAIN

6.1  INTRODUCTION

System-of-Systems is a concept that has recently been applied to the electrical grid. In particular, its appliance in the energy sector has grown with the shift from the electrical grid to the Smart Grid concept.

An electrical grid is an interconnected network for delivering electricity from suppliers to consumers. It consists of power plants that produce electrical power, high-voltage transmission lines that carry power from distant sources to demand centers, and distribution lines that connect individual customers.

Power plants may be located near a fuel source or in a way to take advantage of renewable energy sources, and are often located away from heavily populated areas. They are usually quite large to take advantage of the economies of scale. The electric power which is generated is stepped up to a higher voltage and then delivered to the transmission network.

The transmission network will move the power over long distances, sometimes across international boundaries, until it reaches its wholesale customer (usually the company that owns the local distribution network).

On arrival at a substation, the power will be stepped down from a transmission level voltage (high voltage) to a distribution level voltage (medium voltage). As it exits the substation, it enters the distribution wiring. Finally, upon arrival at the service location, the power is stepped down again from the distribution voltage to the required service voltage(s) (low voltage).

The electrical grid is expected to evolve to a new grid paradigm, the Smart Grid (i.e., smart electric grid, smart power grid, and intelligent grid). The traditional electrical grids are generally used to carry power from a few central generators to a large number of users or customers. In contrast, the new emerging Smart Grid uses two-way flows of electricity and information to create an automated and distributed advanced energy delivery network.

Thus, a Smart Grid is a modernized international, national, regional, municipal, or residential electrical grid that uses information and communications technology to gather and act on information, such as information about the behaviors of suppliers and consumers, in an automated fashion to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity [101].

As mentioned before, there are several sizes for a Smart Grid, starting from a residential to a national one. A national Smart Grid includes transmission networks for transferring electricity on large distances, and distribution networks for distributing electricity from transmission networks to consumers. It also includes integrated communication networks to support real-time control and data interchange. Advanced sensors (e.g., phasor measurement units) are used for monitoring usage (e.g., power quality, equipment health, transmission line temperature, etc.). Smart meters are used to measure real-time energy consumption. Smart energy panels are used to intelligently distribute electrical power, software intensive control and information systems for operator and manager control and network monitoring, demand management, real-time sensor fusion, decision support, storage of usage data, recording of anomalies, etc.

From this point of view, a Smart Grid is an SoS, where entities such as the controller, sensors, and smart meters represent the constituent systems.

6.2  PROBLEM AREA & EXAMPLE SCENARIOS

The aim of a Smart Grid is to improve the electricity quality (e.g., free of sags, spikes, disturbances and outages) increasing, therefore, its availability, reliability, safety, and security.
Smart Grids are also foreseen to perform self-healing actions in order to tackle emergency situations such as extreme weather, solar storms, electro-magnetic pulse, and terrorist attacks. One peculiar function of the Smart Grids is to support decentralized power generation, so that homes and businesses can be both electricity consumers and suppliers (i.e., “prosumers”). The Smart Grids are flexible enough so that consumers can participate in grid operations by:

- selecting electricity providers based on cost or type (e.g., green suppliers such as solar, wind, and biomass);
- scheduling electricity usage (e.g., during off peak hours), offering increased energy efficiency to support energy independence and reduce global warming.

Two interesting problem scenarios which exemplify the flexibility management of Smart Grids are the Medium Voltage Control and Electric Vehicle Charging [105].

The introduction of Distributed Energy Resources (DERs) can influence the status of the power grid. The behavior of DERs can affect the capacity of the DSO (Distribution System Operator) to comply with the contracted terms established with the TSO (Transmission System Operator) and can directly affect the quality of service of their neighbour grids. The DSO has to deal with units whose behavior is both unknown and uncontrollable, and investments on conventional reactive power control devices in substations may become ineffective. Automatic voltage regulations limited to the OLTC (On Load Tap Changer) of the substation transformers, as usually operated in passive grids, may be not sufficient to meet the supply requirements established by the norm EN 50160 [131].

In order to maintain stable voltages in the distribution grid, the Voltage Control function can be exploited. The main functionality of the medium voltage control function is to monitor the active distribution grid status from field measurements and to compute optimized set points for MV DERs, flexible loads and power equipment deployed in HV/MV substations. The voltage profile optimization is reached by controlling reactive and active power injection by distributed generators, flexible loads and energy storages, and setting On Load Tap Changers (OLTC), voltage regulators and switched capacitor banks. Costs of control actions and load/generation forecasts in the area have to be taken into account to select the appropriate control strategy.

The objectives of Electric Vehicle (EV) charging are, instead, of:

- Satisfying the charging demands of arriving EVs in such a way that the charging load is distributed according to the resource capacities in time and space (geographical routing for public charging).
- Enabling EVs to charge flexibly, a feature that can be used by the local DSO to manage power quality control in the LV grid along with decentralized Photo Voltaic (PV) production as well as other loads (e.g., households), and by the EV aggregator to handle on the energy market.
- Enabling the DSOs to monitor the state of low voltage grid under EV load conditions.

Pre-charging interactions occur before arrival at the charging spot. The interaction of the EV with the Charging Station Operator (CSO) leads to a reservation and the allocation of a charging spot (CS), as well as the communication of desired charging demand, arrival time, leave time, etc. from the EV to the CSO. Also without the pre-charge phase, the smart charging scenario is possible: the EV arrives at a free CS and requests the CSO to charge, while providing following data: arrival time (now), estimated departure time, minimum required amount of energy, maximum required amount of energy (to fill the battery), preferred charging speed. The CSO creates a schedule, based on up-to-date information: a) from the DSO about the charging capacity at that certain grid bus (available power), b) energy bought optimally on the market, following the offered (flexible) demand.
6.3 CONSTRAINTS

The main constraint that a power grid has regards the power generation capacity provided by the power plants. Indeed, the loads that are present on the grid and that can be supplied, cannot demand an amount of energy greater than what is available. There is a need to continuously balance the generation on load in the power grid, to adjust generation output to match load or reduce load to match available generation.

The rise of renewable energy sources causes extensive changes in power systems. Typically, a conventional power system consists of controllable generation units and non-controllable loads. By increasing the intermittent generation, the controllability of generation is lost to a certain extent, such that at times renewable generation potentials may no longer be sufficient to supply the demand. Indeed, renewable energy sources rely on wind turbines or photovoltaic cells. Thus, their energy generation depends on the weather and, consequently, is not constant over time.

Voltage stability should be assured. Indeed due to different load characteristics, the rated power of the loads can only be evaluated at a constant voltage. In order to avoid an undesired power mismatch and a subsequently-supposed damage of the power unit, it is very important to regulate the voltage within a compliant range.

Also reactive power plays an important role in power grid. The generation units in a given network have a limited reactive power capability. Since reactive power and the voltage magnitudes have a tight coupling in a power system, the feasibility of the nodal voltage magnitudes depends on the provided reactive power range.

Finally, thermal rating should be also taken into account. In order to prevent damage to the power system components, it has to be ensured that their specified power capacity is not exceeded during power system operation.

6.4 ARCHITECTURE & RELIED UPON MESSAGE INTERFACE (RUMI)

Figure 17 shows the conceptual model of a Smart Grid provided by the NIST [102]. The Smart Grid is divided in seven domains. Each domain, and its sub-domains, encompasses Smart Grid actors and applications. Actors include devices, systems, or programs (i.e., the constituent systems) that make decisions and exchange information necessary for performing applications. Smart meters, solar generators, and control systems represent examples of devices and constituent systems. Applications, on the other hand, are tasks performed by one or more actors within a domain. For example, corresponding applications may be home automation, solar energy generation and energy storage, and energy management.
Table 14 summarizes the actors and the domains in the conceptual model.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Actors in the Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>The end users of electricity. May also generate, store, and manage the use of energy. Traditionally, three customer types are discussed, each with its own domain: residential, commercial, and industrial.</td>
</tr>
<tr>
<td>Markets</td>
<td>The operators and participants in electricity markets.</td>
</tr>
<tr>
<td>Service Providers</td>
<td>The organizations providing services to electrical customers and utilities</td>
</tr>
<tr>
<td>Operations</td>
<td>The managers of the movement of electricity.</td>
</tr>
<tr>
<td>Bulk Generator</td>
<td>The generators of electricity in bulk quantities. May also store energy for later distribution.</td>
</tr>
<tr>
<td>Transmission</td>
<td>The carriers of bulk electricity over long distances. May also store and generate electricity.</td>
</tr>
<tr>
<td>Distribution</td>
<td>The distributors of electricity to and from customers. May also store and generate electricity.</td>
</tr>
</tbody>
</table>

*Table 14 – Domain and actors in the Smart Grid Conceptual Model.*

Figure 18 shows a composite ‘box” diagram that combines attributes of the seven domain-specific diagrams.

In the figure, Domain stands for each of the seven Smart Grid domains, which is a high-level grouping of organizations, buildings, individuals, systems, devices or other actors that have similar objectives and that rely on, or participate in, similar types of applications. Communications among actors in the same domain may have similar characteristics and requirements. Domains may contain sub-domains. Moreover, domains have much overlapping functionalities, as in the case of the transmission and distribution domains. Transmission and distribution often share networks and, therefore, are represented as overlapping domains.
An **Actor** can be a device, computer system, software program, or the individual or organization that participates in the Smart Grid. Actors have the capability to make decisions and to exchange information with other actors. Organizations may have actors in more than one domain.

A **Gateway Actor** is an actor in one domain that interfaces with actors in other domains or in other networks. Gateway actors may use a variety of communication protocols; therefore, it is possible that one gateway actor may use a different communication protocol than another actor in the same domain, or use multiple protocols simultaneously.

An **Information Network** is a collection, or aggregation, of interconnected computers, communication devices, and other information and communication technologies. Technologies in a network exchange information and share resources. The Smart Grid consists of many different types of networks, not all of which are shown in the diagram. The networks include: the Enterprise Bus that connects control center applications to markets, generators, and with each other; Wide Area Networks that connect geographically distant sites; Field Area Networks that connect devices, such as Intelligent Electronic Devices (IEDs) that control circuit breakers and transformers; and Premises Networks that include customer networks as well as utility networks within the customer domain. These networks may be implemented using public (e.g., the Internet) and non-public networks in combination. Both public and non-public networks will require implementation and maintenance of appropriate security and access control to support the Smart Grid. Examples of where communications may go through the public networks include: customer to third-party providers, bulk generators to grid operators, markets to grid operators, and third-party providers to utilities.

**Comms (Communications) Path** shows the logical exchange of data between actors or between actors and networks. This communications path can be secured against attacks or not.

![Detailed Smart Grid Conceptual Model](image-url)

*Figure 18 – Detailed Smart Grid Conceptual Model [102].*
Figure 19 shows a more detailed Smart Grid architecture in which the domains of the conceptual model are mapped. In particular, the Smart Grid covers the whole distribution system. Service Providers and Bulk Generator are out of the scope of the architecture represented in Figure 19.

The individual control layers correspond to the main voltage levels from the high voltage (HV) grid down to the low voltage (LV) consumers. This provides the potential as well as the need for decentralized fast reacting control solutions, autonomous control domains as well as minimization of communication needs. Each control-layer is able to collect and process data related to the operation of the grid as well as the communication network. At the top layer, central management systems exist to provide overall interactions with Transmission System Operators (TSO), Distribution System Operators (DSO) management systems and operational procedures (energy market, operation policies, supervision, billing systems, business intelligence). The central management systems cover Supervisory Control and Data Acquisition (SCADA) / Distribution Management System (DMS) and flexible demand control.

The Medium Voltage Grid Controller (MVGC) manages the MV grid through a set of monitoring and control functionalities, including several Distributed Energy Resources (DERs) that are directly connected at the MV level. The MVGC further includes functionalities to manage MV level demand flexibility (provided through MV level connected consumers and LV grids) in conjunction with an overall grid demand control strategy, as well as functionality for energy routing and maintenance of grid stability and power quality.

Low Voltage Grid Controller (LVGC) is in charge of managing a set of LV grid consumers and producers via local actuators, sensors and customer energy management systems (e.g., home gateways). Besides supervising and controlling the LV power quality, the LVGC is also responsible for executing local demand flexibility control of e.g., EVs and households in conjunction with a given MV grid level demand control strategy.
6.5 **SEMANTIC OF COMMUNICATION**

Grid communication occurs in two senses: electrical communication and information communication [103]. In the electrical domain, power system events have system-wide effects that are communicated throughout the electrical system, with the system itself as the medium. In the information domain, all information necessary to the operation of the power system with the intent of achieving or maintaining a desirable nonlocal state, and which are not already present in the power system, must be communicated explicitly. For example, the failure of a generator can be remotely detected as a slight reduction in the frequency of the alternating current. The frequency drops because the remaining generators, which in most cases are rotating mass type, have to supply the system's loads. This increases the load at each generator, which is then slowed slightly and consequently reduces the frequency.

The communication system is needed to carry two kinds of information: commands and power system condition information, whether directly from sensors or operational devices (for example, a relay might send a digital signal that it had complied with a command) or from other controllers.

It is consolidated that communication networks should support a two-way flow of information between the various constituent systems in the electric grid [104]. On the contrary, it is still an open issue what specific technologies should be used in each Smart Grid application domain and how they should be implemented. One reason is that the Smart Grid consists of many different types of networks, including for example:

- Enterprise bus that connects control center applications, markets, and generators;
- Wide area networks that connect geographically distant sites;
- Field area networks that connect devices, such as intelligent electronic devices that control circuit breakers and transformers;
- Access network that include customer networks as well as utility networks within the customer domain.

A communication system in an Smart Grid must at least satisfy the following basic requirements:

- The communication system must support the quality of service (QoS) of data. This is because the critical data (e.g., the grid status information) must be delivered promptly.
- The communication system must be highly reliable. Since a large number of devices will be connected and different devices and communication technologies will be used, guaranteeing the reliability of such a large and heterogeneous network is not a trivial task.
- The communication system must be pervasively available and have a high coverage. This is mandated by the principle that the Smart Grid can respond to any event in the grid in time.
- The communication system must guarantee security and privacy.

As mentioned, several technologies can be used, both wireless and wired. Table 15 summarizes the evolution of the communication technologies and protocol standards used in the electric systems.
Table 15 – Summary of communication system development activities for electric utilities [109]

### 6.6 DYNAMICITY

As for other systems, some unforeseen events can happen in a Smart Grid which cause a change from a correct service state to an incorrect one. In the past, grid Fault Detection, Isolation and Restoration mechanisms to bring the system back to the correct service were deployed and evolved over the years.

In the past, legacy SCADA systems used only data-driven and rule-based approaches [110]. Then SCADA/EMS systems (Energy Management Systems targeted on transmission grids) came to effective deployment where, on top of data-driven processing and rule-based techniques, knowledge about transmission grid behavior could be obtained from that data. The communication networks were private, corresponding to independent systems, and were mainly prone to communications errors or infrastructure damage resulting from natural causes, unpredicted events, or, in some cases, due to human activity (accidents, sabotage). Communication redundancy was already implemented, coping, at least at that time, with the main availability requirements.

It was in the 90s that the concept of using other communication networks, namely from public operators, became a reality. Substation Automation Systems (SAS), including protection features, started to be fully digital in the mid-90s, emerging as a standardized method during the 2000s. Standards like IEC 61850 and others brought together the automation and the protection worlds, although residing within a same substation. This was a paradigm shift, only possible due to improvement and evolution of ICT, computerised solutions and embedded systems.

Distribution feeder Automation (DA) became a reality with plenty of deployments during the 2000s. Basically, it corresponds to tele-control and/or automation mechanisms, dealing with MV substation downstream feeders, overhead switches, ring main units, reclosers, etc., normally operated in radial schemes, suitable for reconfiguration and backup from other feeders. The main feature that is covered by DA, besides SCADA monitoring and tele-control, is Fault Detection, Isolation and service Restoration (FDIR). These entire grid self-healing schemes depend on reliable communication networks, with high levels of accuracy and performance.

In conclusion, during the 2000s the Smart Grid concept came up, bridging Electric Power Grid (EPG) and ICT, as a combined challenge, as both kinds of systems rely on each other. The Smart Grid, as we discuss it today, briefly corresponds to the overall EPG, from generation to transmission (HV grid) and distribution (MV grid), down to the LV level.
Many primary distribution systems are designed and constructed as meshed grids, but are operated as radial feeder systems with normally-open tie switches. These tie switches can help transferring unfaulted, but out-of-service, load to neighbouring feeders to reduce the total load that has to be cut off over a prolonged period after a fault.

Whenever there is a fault in an electrical distribution system, in order to ensure minimal reduction of system reliability, the unfaulted de-energized areas should be supplied with power as soon as possible. Although repairing the fault may take a while, it is possible to quickly restore power to areas cut off by the fault or the consequent protections’ tripping, if they can be temporarily connected to neighbouring feeders carrying the extra load.

Automatic feeder sectionalizing and restoration is a core application for DA. In their most basic form, these systems detect a fault, determine its location, and open the nearest available switches or fault interrupters during a tripped state of the fault-clearing recloser or breaker, isolating the faulted segment from the rest of the feeder.

Automatic Fault Detection, Isolation and Restoration (FDIR) algorithms carry out complex grid self-healing. This is especially true if the feeder segments have to deal with Renewable Energy Sources (RES) based generation, such as photovoltaic stations or wind farms, shaping the pattern of the “no longer” top down flow of energy.

Being bidirectional in terms of power flows, the distribution grid becomes more prone to faults, their negative impact being clearly more aggressive and unpredictable.

The main goals of self-healing systems for Medium Voltage (MV) Smart Grids are to supply the maximum load affected by the fault in the shortest period of time, minimizing the number of switching operations and keeping the grid capacity within its operating limits.

There are different ways of implementing self-healing of electric power distribution grids [106]:

- **Centralized solutions**, at the Control Centre (complete model): the SCADA/DMS system concentrates all the modelling, maintenance and intelligence. The solution relies on telemetry and remote control for automatic grid operation (FDIR also provides manual switching when applicable), besides power applications. It runs a complete overall grid model and control options able to analyse multiple faults and solution scenarios across a wide area, suitable for a Smart Grid.

- **Distributed solutions** (static model): consists on a script-based approach using feeder distributed Remote Terminal Units (RTU). It is a cost effective solution when only few switching devices are employed on a restricted area, using a dedicated communication infrastructure allowing fast response actuation with predefined automation schemes. However, it is not ready to operate under non-standard grid topology and is unable to deal with multiple faults. It also lacks flexibility for handling RES distributed generation, storage or electric vehicles penetration, thus, being not completely suitable for a Smart Grid.

Substation-centric DA solutions (dynamic model) are based on a model driven intelligence sited at the primary substation, coordinating feeders and neighbouring substations, with the decision process being made at substation or Control Centre level (pending approval). It grants interoperability supporting different switches, reclosers and RTU vendors, assuring adaptability to any real-time grid configuration, including any protection or automation plans. Though the operation area is restricted to the neighbouring of the substation where the substation-centric solution performs covering a specific Distribution Grid Area (DGA), it is able to dynamically derive complex restoration solutions. These are derived involving multiple feeders and using, where necessary, automatic load transfer schemes to achieve optimal restoration, besides offering flexibility for handling RES distributed generation, storage and electric vehicles penetration.
6.7 Evolution

The Smart Grid is undergoing constant evolution as it evolves from a traditional power grid to a modern intelligently controlled power grid. It is highly likely to change as new requirements are implemented using new and rapidly evolving technologies (e.g., superconductivity, new storage batteries, and advances in green technologies).

The Smart Grid is not just a compilation of smart meters, wind, solar, or plugin electric vehicles. Instead, it is a series of technologies that will allow companies to integrate, interface with and intelligently control all of these innovations. As the technology evolves, the Smart Grid will allow a two-way flow of electricity and information that is capable of monitoring everything, from power plants to customer preferences and individual appliances/equipment. The Smart Grid will provide real-time information and near instantaneous balance of supply and demand at the device level. The Smart Grid will be implemented incrementally over the next two decades as technology, pricing, policy, and regulation change. Smart grid characteristics include more consumer participation in determining electric usage through an interactive electrical network. Consumers will have real-time information on pricing and usage which will allow for informed decisions on equipment/appliance usage. A Smart Grid will increase energy efficiency by allowing consumers to make decisions regarding when to run equipment and when to turn it off, such as during peak time or high cost periods. Two-way digital communications will provide distribution network operators with outage information and equipment in consumers’ premises with demand and pricing information. The Smart Grid will provide the ability to balance reliability and power quality where individually, needed rather than wholesale across the network or a portion of the network.

6.8 Emergence

An electric power transmission system contains many components such as generators, transmission lines, transformers and substations [113]. Each component experiences a certain loading each day and when all the components are considered together they experience some pattern of loadings. The pattern of component loadings is determined by the system operating policy and is driven by the aggregated customer demands at the substations. The power system operating policy includes short term actions such as power dispatch. The operating policy seeks to satisfy customer demands at least cost. Short term customer demand is typically driven by daily and seasonal cycles while long term demand is driven by geographic shifts in population and population changes, development of alternative energy sources and industry growth and changes.

Events can occur on the power grid that either limit the loading of a component to a maximum or zero the component loading if that component trips or fails. Events that can occur include transformer failure, the probability of which generally increases with loading, and operator re-redispatching to limit power flow on a transmission line to its maximum rating. These events can cause a redistribution of power flow in the network and hence an increase in the loading of other system components, and the events can cascade. If a cascade of events includes limiting or zeroing the load at substations, then a blackout results.

The main requirement of the power grid SoS is to reliably generate and supply electricity, wherever and whenever it is needed on the grid. In order to achieve this, there is a need to continuously balance the generation on load in the power grid, to adjust generation output to match load or reduce load to match available generation. Control actions are limited primarily to adjusting generation output and to opening and closing switches to reconfigure the network. Every action affects all other activities on the grid. The activities of all the control actions must be coordinated, often across large areas. If not managed properly, failure of a single element can cause the subsequent rapid failure of many additional elements, disrupting interconnected transmission systems over a broad geographic area.
6.9 **GOVERNANCE**

Due to its physical size, an international or national Smart Grid will be developed by many utilities under partial direction (policy) and partial funding from international (e.g., EU), national (e.g., Italy, Denmark, etc.) governments and regulatory agencies.

Whether Smart Grid deployments are being driven by legislative and regulatory policies, realizing operational efficiencies, or creating customer value, the motivation and pressure to produce has caused the industry to perform Smart Grid implementations in fragmented efforts with limited or no stakeholder coordination or agreed-upon standards [112]. As the technology and interoperability standards mature and gain consensus, some early adopters may be faced with "sunk costs" (retrospective past cost that has already been incurred and cannot be recovered) or, at the very least, some serious integration and interoperability issues going forward.

There are three challenges facing broad Smart Grid standards adoption:

- The large number of stakeholders, different considerations, number and complexity of standards available (and missing) requires a more formal nationally-driven governance structure.
- Since Smart Grid efforts are underway, and in some cases complete, standards adoption must consider work already completed and underway.
- Interoperability discussions and definitions should be expanded to focus on standards across systems (inter-system) rather than just within systems (intra-system).

In the Smart Grid SoS, governance is mostly about standards. A governance model for standards would accelerate the implementation of a secure, intelligent, interoperable, and a fully-connected Smart Grid. Early identification and development of standards for interoperability and for device specification will ensure that pending deployments will offer lasting and extensible value.

6.10 **HANDLING OF TIME**

The North eastern blackout of 2003 is a concrete example of the importance of the time synchronization in Smart Grids [108]. It brought to the attention of the power distribution industry the need for time synchronization requirements for synchronized distributed measurements to prevent cascading failures in the grid. The event also heightened the need for more accurate data recorders on the electrical grid to aid in rapid fault diagnosis. Inaccurate time-stamps caused significant delays in diagnosing the cause of the blackout.

Accuracy, availability, and reliability of time synchronization are required for power systems applications. Recent developments in the Smart Grid have spawned interest in the use of phasor measurement units (PMUs) to help create a reliable power transmission and distribution infrastructure [107][104]. A PMU measures the electrical waves on an electrical grid to determine the health of the system. Technically speaking, a phasor is a complex number that represents both the magnitude and phase angle of the sine waves found in electricity. Phasor measurements that occur at the same time are called synchrophasor, as are the PMU devices that allow their measurement. Typically, PMU readings are obtained from widely dispersed locations in a power system network and synchronized using the global positioning system (GPS) radio clock. With a large number of PMUs and the ability to compare shapes from alternating current (AC) readings everywhere on the grid, system operators can use the sampled data to measure the state of the power system and respond to system conditions in a rapid and dynamic fashion.

Accuracy requirements in industry standards range from 1 ms in Disturbance Measuring Equipment (DME) for event reconstruction to accuracy on the order of microseconds or less for PMUs and fault detection [111]. The IEEE C37.118 Standard for Synchrophasors for Power Systems requires the Total Vector Error (TVE) of PMUs to be 1 percent. For a 60 Hz system, the time synchronization errors can be at a maximum of 26 s. However, the TVE is an aggregation of errors based on instrumentation conversion latencies, phasor measurement processing latencies,
and time synchronization offsets. Substation Intelligent Electronic Devices (IEDs) use a variety of methods to obtain synchronized time including Global Positioning System (GPS), Pulse Per Second (PPS), and IRIG signals. With the increased use of IEDs, there are benefits associated with using distributed time synchronization protocols such as PTP which leverage data networks. One such benefit is to eliminate the need for a separate time synchronization input for each IED.

6.11 Dependability

Reliability is the ability of a component or system to perform required functions under stated conditions for a stated period of time. System reliability is an important topic in power grid research and design. Cascading blackouts could happen. For example, in the infamous 2003 East Coast blackout, 50 million people in the U.S. and Canada lost power for up to several days [108].

An important aspect of the operation of the electric power system is protection [115]. This means ensuring the safety of the system, including generating units and other grid assets, and the people who may come in contact with the system. Protective action must be taken in fractions of a second to avoid equipment damage and human injury. Protection is achieved using sensing equipment as well as circuit breakers and other types of switches that can disconnect and de-energize parts of the system in the case of a fault, such as a damaged transmission line or a short circuit. Once the fault is repaired, that segment of the system can be brought back online. Proactive planning for contingencies also protects the electric power system. Computers are regularly calculating system power flows and voltages under various possible contingencies, for example the failure of a large generator or transmission line, to identify the best corrective action to take in each case.

It is expected that distributed generation (DG) will be widely used in Smart Grid. While using some fluctuant and intermittent renewables may compromise the stability of the grid, innovative architectures and designs can offer great promise to connect DGs into the grid without sacrificing reliability.

As loads are being served locally within a microgrid (i.e., small size grid), less power flows within the entire grid infrastructure. Thus, the reliability and stability of the Smart Grid can be enhanced. Furthermore, the reliability and stability of a Smart Grid also depends on the reliability of the measurement system which is used to monitor the reliability and stability of the Smart Grid. Recently, the wide-area measurement system (WAMS) based on phasor measurement units (PMUs) is becoming an important component for the monitoring, control, and protection functions in Smart Grids.

Failure prediction and prevention play important roles in the smart protection system since they attempt to prevent failures from happening. Once the system does fail, failure identification, diagnosis, and recovery are required to make the system recover from the failure and work normally as soon as possible (see Section 6.6).

Another effective approach of self-healing is to divide the power grid into small, autonomous islands (e.g., microgrids) which can work well during normal operations and also continue working during outage. By appropriately controlling the system reconfiguration, the impact of disturbances or failures can be restricted within the islands or can be isolated. Cascading events and further system failures can hence be avoided. Therefore the overall efficiency of system restoration can be improved. Failures could also occur on smart meters so that load data could contain corrupted or missing data. Processing or even recovering such data is important since it contains vital information for day-to-day operations and system analysis.

Protection of microgrids during normal or island operations is also an important research topic since microgrids will be widely used in Smart Grids. Traditional protection for distribution grids is designed for high fault-current levels in radial networks. However, during an islanding operation of the microgrid, high fault-currents from the utility grid are not present. Moreover, most of the DG units which will be connected to the low voltage microgrid will be converted/interfaced with limited fault-current feeding capabilities. This means that the traditional fuse protection of low voltage
network is no longer applicable for microgrids, and that new protection methods must be developed.

6.12 SECURITY

Security is a never-ending game of wits, pitting attackers versus asset owners. Smart grid security is no exception to this paradigm. Cyber security is regarded as one of the biggest challenges in Smart Grids. Vulnerabilities may allow an attacker to penetrate a system, obtain user private information, gain access to control software, and alter load conditions to destabilize the grid in unpredictable ways.

To define the state of the art in Smart Grid security, it is important to understand that the term Smart Grid itself is a quite undefined term, and refers to a collection of different digital components in the grid that serve vastly different tasks (e.g., billing, substation automation, routing, load balancing, consumer integration, improvement of retail markets, etc.), and that — even though implemented on the same grid — may not even interact with each other. Furthermore, different countries and companies see different use cases for the Smart Grid, and therefore focus on different components and generate different security requirements, which are also still evolving over time. Finally, the same requirements can be met in different ways, sometimes by excluding use cases that are difficult to secure (e.g., a remote off-switch for smart meters), sometimes by stringent security requirements, and sometimes by accepting the risks and implementing alternative mitigation strategies. Thus, the state of the art differs widely for different parts of the grid, although harmonization efforts are underway.

In terms of communication security, many of the legacy protocols used in Smart Grid systems offer no security at all. In some systems, this is being addressed, and security layers are added to some of the most important communication protocols. While the first generation of these security layers were often found to have flaws, second generation protocols are currently under development, though they may not be available within the necessary timeline for all rollouts.

For fault tolerance, most grid systems have been designed in a redundant way, with backups for most critical systems. The redundancy is mostly designed for the crash-failure scenario (i.e., the primary system becomes completely unavailable), and can usually not handle byzantine (e.g., malicious) attacks.

Following standard best practices, most Smart Grid operators implement basic zoning, i.e., separate different systems that have different security requirements with firewalls and DMZs. In some cases, this is done using information diodes, which enforce a strong separation. As most smart grids are currently only in the beginning of being implemented, the current architectures are likely to evolve significantly in the future.

Monitoring is starting to be recognized as an important factor. While intrusion detection on the IT networks is commonplace, security monitoring for the OT (Operation Technology) networks is still in its infancy, though a couple of commercial solutions are available or emerging for specific subsystems.

Some best practices and guidelines are currently under development and partially already enforced, such as the US NERC-CIP guidelines, the ENISA minimum security requirements, the British CPA, and the German BSI Common Criteria Profile for Smart Meter Gateways. While the former two address mostly organizational aspects and policies, the latter defines device security for a specific Smart Grid component. At the present time, however, there are few appropriate testing and certification guidelines for large parts of the Smart Grid, and testing and policies are often left to the individual companies or associations.

One of the security issues comes from the newly deployed smart meters [105], which are also one of the first Smart Grid components deployed on a massive scale. Smart meters are extremely attractive targets for malicious hackers, since vulnerabilities can easily be monetized. Hackers who compromise a smart meter can immediately manipulate their energy costs or fabricate generated energy meter readings to make money. A common consumer fraud in the traditional power grid is
that customers turn a traditional physical meter upside down in the electrical socket to cause the internal usage counters to run backward. Due to the use of smart meters, attacks can even be done with remote computers or instructions downloaded from the Internet. Moreover, widespread use of smart meters may provide a potentially large number of opportunities for adversaries. For example, injecting misinformation could mislead the electric utility into making incorrect decisions about local or regional usage and capacity. Let us consider a simple but probably effective Denial-of-Service (DoS) attack. An adversary forges the demand request of a smart meter, and keeps requesting a large amount of energy. Within the framework of the Smart Grid, it is possible that the electric utility disconnects all the appliances connected to this meter so that all the power services for this user are denied. Widespread deployment of smart meters not only leads to a potentially large number of opportunities for adversaries, but also opens up a door to cyber attacks which could result in broad effects and even large-scale disasters.

Smart meters also have unintended consequences for customer privacy. NIST pointed out that “the major benefit provided by the Smart Grid, i.e., the ability to get richer data to and from customer meters and other electric devices, is also its Achilles' heel from a privacy viewpoint”. The obvious privacy concern is that the energy use information stored at the meter acts as an information-rich side channel, and can be repurposed by interested parties to reveal personal information such as individual’s habits, behaviors, activities, preferences, and even beliefs.

It is well-known that the communication technologies we are using are often not secure enough themselves. It is expected that most of the security and privacy issues existing in the general communication networks (e.g., Internet and wireless networks) could also exist in Smart Grids. Particularly, we need to focus more on wireless communication technologies since wireless networks are expected to be the more prevalent networks in Smart Grids. Malicious attacks on information transmission in Smart Grids can be categorized into the following three major types based on their goals.

- **Network availability**: Malicious attacks targeting network availability can be considered as DoS attacks. They attempt to delay, block, or even corrupt information transmission in order to make network resources unavailable to nodes that need to exchange information in the Smart Grid. The design of information transmission networks that are robust to attacks targeting network availability is the top priority, since network unavailability may result in the loss of real-time monitoring of critical power infrastructures and global power system disasters.

- **Data integrity**: Data integrity attacks attempt to deliberately modify or corrupt information shared within the Smart Grid and may be extremely damaging in the Smart Grid.

- **Information privacy**: Information privacy attacks attempt to eavesdrop on communications in Smart Grid to acquire desired information, such as a customer’s account number and electricity usage.

### 6.13 Quality Metrics


The most common distribution indices include the System Average Interruption Duration Index (SAIDI), Customer Average Interruption Duration Index (CAIDI), System Any Interruption Frequency Index (SAIFI), Momentary Average Interruption Frequency Index (MAIFI), Customer Average Interruption Frequency Index (CAIFI), Customers Interrupted per Interruption Index (CIII), and the Average Service Availability Index (ASAI).

SAIFI is the average number of times that a system customer experiences an outage over a specific time basis (typically a year). SAIFI is evaluated as:
where \( \lambda_i \) is the failure rate and \( N_i \) is the number of customers for location \( i \).

The most often used performance measurement for a sustained interruption is the SAIDI. This index measures the total duration of an interruption for the average customer during a given time period. SAIDI is normally calculated on either monthly or yearly basis. It is calculated as:

\[
SAIDI = \frac{\text{Total Duration of Customer Interruptions}}{\text{Total Number of Customers Served}} = \frac{\sum r_i N_i}{N_T},
\]

where \( r_i \) is the restoration time expressed in the same time basis as SAIDI is evaluated.

Once an outage occurs the average time to restore service is found from the CAIDI. CAIDI is evaluated as:

\[
CAIDI = \frac{\text{Total Duration of Customer Interruptions}}{\text{Total Number of Customer Interruptions}} = \frac{\sum r_i N_i}{\sum \lambda_i N_i} = \frac{SAIDI}{SAIFI}.
\]

Similar to SAIFI is CAIFI, which measures the average number of interruptions per customer interrupted per year. It is simply the number of interruptions that occurred divided by the number of customer affected by the interruption:

\[
CAIFI = \frac{\text{Total Number of Customer Interruptions}}{\text{Total Number of Customers Interrupted}} = \frac{\sum N_i}{CN},
\]

where \( CN \) is the total number of customers who had at least one interruption.

MAIFI is the average number of momentary interruptions that a customer experiences during a given time period (typically a year). Most distribution systems only track momentary interruptions at the substation, which does not account for pole-mounted devices that might momentarily interrupt a customer. MAIFI is rarely used in reporting distribution indices because of the difficulty in knowing when a momentary interruption has occurred. MAIFI is calculated by summing the number of device operations (opening and reclosing is counted as one event), multiplying the operations by the number of customers affected and dividing by the total number of customer served:

\[
MAIFI = \frac{\text{Total Number of Momentary Customer Interruptions}}{\text{Total Number of Customers Served}} = \frac{\sum ID_i N_i}{N_T},
\]

where \( ID_i \) is the number of interrupting device operations.

Finally, ASAI is the ratio of the total number of customer hours that service was available during a given time period to the total hours demanded. This is sometimes called the service reliability index. The ASAI is usually calculated on either a monthly basis (730 hours) or a yearly basis (8760 hours), but can be calculated for any time period. The ASAI is evaluated as:

\[
ASAI = \frac{\text{Customer Hours Service Availability}}{\text{Customer Hours Service Demand}} = \frac{N_T \cdot T - \sum r_i N_i}{N_T \cdot T},
\]

where \( T \) is the time period under study in hours (e.g., 8760 hours for one year).
7 CROSS DOMAIN COMMON FINDINGS

This Section aims to summarize the results of the analysis on the reference domains presented in the previous Sections (i.e., Automotive, Railway, Energy, Banking (ATM), and Crisis Management) in order to identify common core aspects relevant to SoSs.

Results of this analysis are presented following the same structural viewpoints used in the Section above:

1. Constraints;
2. Architecture & RUMI;
3. Semantic of Communication;
4. Dynamicity;
5. Evolution;
6. Emergence;
7. Governance;
8. Handling of time;
9. Dependability;
10. Security;

It is worth noting that Governance appears in this analysis in two ways, namely, as a viewpoint, corresponding to the way in which the SoS is managed, and as a constraint, referring to the limitations imposed by the governance model as described in the Constraints viewpoint Section.

7.1 CONSTRAINTS

A first common aspect identified in all the analyzed domains regards the following constraints, which may refer to different sources of restrictions.

One possible source of constraints is related to standards. Nowadays standards are more and more used to regulate the development and the management of critical systems and infrastructures. The management and maintenance shall be compliant with rigorous procedures. For example, the ISO 26262 standard and the EN 501xx standards regulate the safety lifecycle development process in the Automotive and Railway domains, respectively. These standards are the domain specialization of the IEC 61508 standard, which is the most important legislation in the field of Functional Safety.

Asset constraints are also a source of SoS constraints. Assets are fundamental elements for the SoS (e.g., tracks, bridges, tunnel, GPS, PV panels, PCs, ECU, mobile phones, etc.). They have some constraints derived from the technology with which they are implemented. These constraints are also reflected in the SoS.

The system life and the lifecycle can also be considered as constraints. Note that lifecycle refers primarily to the CSs that constitute an SoS, whereas from the SoS perspective it is more appropriate to talk about system life. Lifecycle is important because development techniques and strategic choices are often related to the lifecycle duration of the CS (car, rolling stocks, ATM, smart meters, etc.). For example in the Railway domain, tracks, bridges, tunnels, etc. are built with reasonable effort and supposed to last decades; in the Global Automated Teller Machine Network, payment cards as well as terminals are designed to operate for 3-10 years. In the Energy domain, MVGCs and Control centers are supposed to last decades. In the Automotive domain, the CSs inside a vehicle are expected to have 10-15 years duration.
Constraints derive also from *economic* aspects. In particular, it is possible to say that the cost-effectiveness is a common driver for the technology progress. The economic aspects certainly influence the strategic choices made as well as the development in all areas under consideration. For example, replacing payment terminals over the ATM network represent a considerable cost for the banks, and for this reason it is strongly required that they are designed to operate for many years, allowing a cost-effective introduction of new technologies. On the other hand, in the Automotive domain, the electronic boards, mounted into the cars, are designed to last few years since the cost of their replacement is low.

The last type of constraints, but with the same importance as the previous ones, concerns the *governance*. Governance describes the way in which the SoS is managed, including defining roles, laws, standards, etc. This is a critical aspect because different owners/regulations related to different CSs may limit the interoperability across them. For example, there are a plenty of players that have a role in the Railway domains. There is the European Commission, which defines guidelines for system integration. Then there are Member States, which supervise the system, and private/public companies, which implement and manage the infrastructure.

### 7.2 Architecture & RUMI

The analysis identified two different architectural perspectives for Automotive and Railway domains. We can represent them as different levels of a hierarchy, that we describe in what follows.

In particular, in the Automotive domain it is possible to consider:

- vehicle as an SoS itself, where the CSs are the components that make it up (intra-vehicle perspective);
- vehicle as a CS, which interacts with other vehicles (CSs) and with the infrastructure, the latter representing the SoS.

In a similar way, it is possible to identify these two perspectives also in the Railway domain:

- train as an SoS and its components as CSs;
- train as a CS, considering at the upper level the railway infrastructure as the SoS.

The two perspectives are also applicable in Crisis Management and in Energy; in the Crisis Management domain, these two perspectives can be analyzed as:

- the Element of a rescue team as CSs acting together as a single SoS (e.g., the police);
- Rescue teams (e.g., from different governance) as CSs, and the whole rescue machine acting in the crisis as the SoS.

In the Energy domain it is possible to consider:

- micro grid as an SoS, whose assets (like Photovoltaic Panels, wind turbines, charging stations, transformers, etc.) which compose it are considered CSs;
- micro grid as a CS and the energy grid, which is composed by several micro grids, is considered an SoS.

Starting from these observations, we can map these two perspectives in two different types of hierarchies for a complex system: a *formal hierarchy* and a *non-formal hierarchy*, also called *holarchy* [117].

In a formal hierarchy each subsystem at level $n$ is linked vertically by a reporting and control relation to its controlling system at Macro-level $n-1$ (i.e., the level above) as depicted in Figure 20-a [118].
In a non-formal hierarchy, or holarchy, there are horizontal interactions among the related sub-systems at Macro-level \( n \) that lead to the formation of a whole with its own characteristic properties from the point of view of the level above, Macro-level \( n-1 \). Starting from the top, it is possible to identify at the Macro-level, i.e., the whole, the purposes of the SoS (Automotive, Railway, Energy, Banking (ATM), and Crisis Management). At the Macro-level-1, the CSs (e.g., banks, ATMs, payment processors, payment cards, cars, trains, rescue teams, DERs, etc.) interact with each other in order to realize the purposes at the Macro-level. At the Macrolevel-2, the CSs are decomposed in subparts such as individual parts within a car, train subsystems, etc. In principle, this iterative decomposition never stops, as it could be always possible to go beyond the atomic level.

We mention two problems that can often be identified in an SoS. The first is the difficulty to clearly and immediately identify what the SoS scope is. For example, in the crisis management domain, there is no specific distinction in what is in the scope of the SoS and what is not, since there is no architecture specified, because it depends on the governance of a specific country. The second is the difficulty related to the definition of the SoS boundary with the external environment, which is not well-defined. For example in Automotive V2X perspective, there is no clear extent of the SoS, since the number of interacting CSs continuously changes.

The lack of accuracy in the definition of both the SoS scope and its boundaries has deep consequences on the definition of a generic design methodology for SoSs. Indeed, in classical system engineering, often the process starts with the definition of the boundary between a system and its environment, which is not always possible with SoSE. In an SoS, CSs need to exchange information and in some cases this should be performed quickly and reliably (e.g., engine speed in In-vehicle perspective in Automotive domain). Sometimes no standards are defined to regulate message exchange between electronic devices of different CSs, as in the Railway domain. In the ATM domain, no standards are specified to regulate the HMI. Furthermore timing requirements must be improved and they have to be reflected also in the interfaces. To address all these problems, well-defined interfaces are required in order to establish the SoS framework. These interfaces are referred to as RUMIs (Relied Upon Message Interfaces).

The CSs of an SoS interact with each other by the exchange of messages. The internal architecture of an SoS is determined by the localization and specification of the Relied Upon Message Interfaces (RUMIs) among the CSs. A RUMI should be a stable interface that establishes the boundaries between two interacting CSs by specifying the data that are exchanged between them and the exact timing of message exchange. RUMIs must be fully specified with respect to their syntax, semantics and temporal behavior [119]. Formal language theory can also be applied...
to the security aspects of protocol design for SoSs. In summary, the definition of the RUMIs and the precise specification of their syntax, semantics and temporal properties are of utmost importance in the design of an SoS.

7.3 **SEMANTIC OF COMMUNICATION**

Semantic of communication refers to the meaning of the information exchanged between different CSs.

In the analyzed domains we pointed out the following categories of information:

- **Actuators commands:** for example Emergency Braking or ABS for the Automotive domain; setpoints in the energy domain; dispatch money related to the Global Automated Teller Machine Network domain.
- **Localization information:** in V2X Automotive domain and in the Railway domain, this type of information is fundamental to know the position of another vehicle (i.e., car or train) or generally the position of an object. For example GPS-based Global Navigation Satellite System (GNSS) receivers are used in many applications.
- **Monitoring information:** information related to the traffic in the case of the Automotive domain, or the status of a place in the case of Crisis Management domain.
- **Infotainment:** information related to the car audio and the trip in the Automotive domain.
- **Personal data:** several examples can be mentioned, such as bank account data for the Global Automated Teller Machine Network domain and information about the car driver in the Automotive domain.

In all domains, the semantic of communication is regulated by standards and regulations that may give more or less strict guidelines. In particular, the ATM, Railway and automotive domains are regulated by strict standards like CENELEC, ISO 26262, ISO8583 and EMV standards. On the contrary, in the energy and crisis management domains there are less strict regulations. For example, in the Crisis Management domain, various initiatives have been undertaken to reduce the semantic gap between the various organizations, however this is an ongoing effort and standards are still under development.

7.4 **DYNAMICITY**

In some domains, the dynamicity aspects are very important because the composition of the SoS may be highly dynamic, not necessarily due to a recovery after a malfunction, but also within the same nominal operation, e.g., in the crisis management domain. In particular, the dynamic groups of entities (CSs) that can be found across the domains are:

- V2X communications, in the Automotive domain;
- Professional emergency organisations, in the Crisis Management domain;
- RBC, in the Railway domain;
- Renewable Energy Prosumers, in the Energy domain;

In V2X communication in the Automotive domain, the number of connected vehicles can quickly change, for example groups of vehicles can be identified, communication with one or more fixed infrastructure devices, and other vehicles may either exit or enter the group. In the Crisis Management domain, new events can occur (e.g., an incident may escalate) and a new organization (e.g., Police, Fire-brigade, etc.) can come into play. In both cases, the role of each actor is not well defined and SoSs may change in a short time. In the Railway domain, trains can enter or exit from the authority region of an RBC. In the Energy domain, renewable energy prosumers can activate or deactivate themselves during the system operation. Their activation or
de-activation, if not controlled, can provoke instability in the grid system which causes unwanted fluctuations of the line voltage.

Aspects of dynamicity can also be related to an online reconfiguration in case of failures and attacks, such as in the following domains:

- ATM
- Energy
  - Island mode in Smart Grids

In the ATM domain, if a fraud is detected, there exist mechanisms (such as card lock, reverse transactions, insurance, etc.) to limit the effects of the ongoing fraud and to stop the fraudulent use of a forged payment card. In the Energy domain, if there is a failure, a reconfiguration is done for example by replacing a CS or shutting down a connection and creating a new one so that the users still continue to be powered.

### 7.5 Evolution

As reported in D2.1, evolution is defined as follows: “a process of gradual and progressive change or development, resulting from changes in the system's environment or the system itself”. This aspect is important because SoSs are prone to evolution in order to support the growing of intra-domain interactions and communication, to ease the adoption of new technologies, to support fragmentation of the operators and to support the adoption of new standards.

In some domains such as Railway, Energy and ATM, the cost of changes could be an issue, while in other domains such as the Automotive this cost is less critical.

For example, in the case of a car, rather than making changes to introduce a new mechanism or an innovation that involves its HW, it is preferred to build a new car given its life time and its cost. Differently, in the case of a train, which is designed to last for decades and for which the replacement costs are elevated, system updates are foreseen.

In order to obtain systems with a high degree of evolvability, different concerns are strongly required, among others, the improvement of communication, a greater cooperation among entities and architectures that are flexible (i.e., easily adaptable to changes) and robust (i.e., performing well in face of changes). For example, in the Automotive domain a possible evolution concerns the possibility that the vehicle can be remotely controlled. To do that, the communication layer among the CSs shall be improved in order to let them interact with each other.

### 7.6 Emergence

Emergence concerns phenomena that manifest at the macro-level and are not observable at the micro-level [118]. The fact that an emergent behavior is considered to be positive, negative or neutral, is dependent on the observer's criteria. In general, emergent behavior is not a-priori positive or negative.

Some examples of bad emergence are:

- Traffic Jams (Automotive)
- Black-outs (Energy)
- Delays on schedule (Railway)
- Miscommunication between organizations (Crisis Management)
- Frauds (ATM)

On the contrary, examples of good emergence are:
• Advanced Emergency Braking (Automotive)
• Voltage Control (Energy)
• Evaluation and actuation of Movement Authorities (Railway)
• The management of an accident by the several actors (e.g., police, firefighters, etc.)
• The possibility to withdraw Euros in China (ATM).

A relation exists between emergence and dynamicity. In particular, it is possible to say that emergent phenomena increase with the increasing of dynamicity. For example in V2X Communication, the traffic flow optimization can be an emergent phenomenon. Indeed if the information about the planned routes is exchanged among the vehicles, the traffic load can be divided on different roads. On the contrary, if some vehicles lose coupling or decide autonomously, then a traffic jam can happen. In crisis management, the configuration of cooperating organizations is very dynamic, typically depending on the incident characteristics. This dynamicity may lead to positive emergent phenomena, such as the successful management of an escalating incident from a number of diverse perspectives (e.g., health, safety, economy, etc.), as well as negative ones, such as a newly added organization's lack of relevant information on the accident during incident escalation.

In all domains it is necessary to predict (unexpected) emergent behaviors, in particular detrimental ones, and this issue requires to enhance exchanging of information between CSs.

7.7 Governance

Governance shows common aspects according to the domains that are under study. Two different kinds of governance schema can be observed. The first involves several actors that have a role in the overall government, but there is no higher authority which dictates guidelines. The second, instead, involves a hierarchical governance schema in which there is an authority which manages the overall regulation.

Typical examples of large amounts of stakeholders are in the Automotive, Railway and Energy domains. Indeed, in the Automotive domain the usefulness of V2X communication strongly depends on a large distribution and a high acceptance rate, and therefore on the interoperability between systems from different technology providers and car manufacturers. Then a governance model supported by a large (as possible) community is required. In the Railway domain, starting in the early 90s, the business has suffered an important structural reform with the separation of roles and organisations. In the same manner, in the Energy domain international or national Smart Grids will be developed by many utilities under partial direction (policy) and partial funding from international, national governments and regulatory agencies. As the technology and interoperability standards mature and gain consensus, some serious integration and interoperability issues are raising.

On the other hand, examples of hierarchical governance are in the ATM and Crisis Management domains. The ATM network is structured starting from Governmental regulations that represent the first hierarchical level of governance. Then at a second hierarchical level there are international standards as EMV and PCI DSS and finally, at the lowest level, there are the interbank network specifications. In Crisis Management the structure of the governance, limiting to a national view, is strictly hierarchical (see Figure 15). In The Netherlands, for example, at a policy level, the National Crisis Centre (NCC) manages information between the Dutch Government and the crisis management centre (LOCC). The LOCC has operational control in a large crisis, however decision-making responsibility resides with the Mayor. The internal structure of the LOCC is made up of officers from each of the major agencies involved, who coordinate LOCC decisions with the local representatives of their organisation during a crisis.
7.8 **HANDLING OF TIME**

A common aspect of all domains is related to the management of time.

The concept of time is a key aspect for all considered domains. The handling of time is a critical aspect in terms of:

- Synchronization requirements,
- Checking real-time requirements.

The concept of a Global Time Base is often required to guarantee a proper functioning of the system. In particular, in the in-vehicle perspective within the Automotive domain and in the Railway domain, the concept of global time is not widely adopted, because the requirements of real-time assume greater importance than the synchronization. Conversely, in the case of V2X Communication in the Automotive domain, in the Energy and in the Crisis Management domains, synchronization requirements and, thus, the global time play a key role. In the Global Automated Teller Machine Network domain both the concept of real-time and the synchronization are important.

In all domains, the definition and the use of methodologies and tools are needed in order to guarantee a tight synchronization to the global time base and estimate resilient distance from the global time base, assuring the concept of time awareness.

It is worth noting that the time source should be reliable as much as possible. Time source redundancies can also be desirable as well. For example, as pointed out in a recent investigation performed by GAO (U.S. Government Accountability Office), GPS provides positioning, navigation, and timing data to users worldwide and is used extensively in many of critical infrastructures sectors, such as communications and transportation [214]. However, sectors’ increasing dependency on GPS leaves them potentially vulnerable to disruptions. A failure in the GPS service which make it unavailable constitutes a common cause failure for all the sectors.

7.9 **DEPENDABILITY**

Another common aspect between the various domains is related to dependability.

SoS dependability may be differently perceived by different classes of users. This is due to the fact that an SoS is comprised of multiple systems capable of providing different services. Different users may invoke different services which rely on different functions of the SoS.

SoS dependability attributes (i.e., availability, reliability, safety, integrity and maintainability) depend on both user operational profiles and the state of the components involved in the accomplishment of the functions invoked by the user. Each attribute of dependability assumes a greater or lesser importance depending on the domain under consideration. For example, the attribute safety is the main aspect in the Automotive (in-vehicle), in the Railway and in the Energy domains. Availability assumes greater importance in ATM, Energy and Crisis Management domains. Reliability is the most significant attribute in the Crisis Management domain. The integrity attribute is particularly important in ATM domain. Finally maintainability is essential in the Railway, ATM and Energy domains.

7.10 **SECURITY**

SoS security is intertwined with many other SoS properties, including dependability, and time. Cross domain security objectives for the ATM, Energy, Automotive, Crisis Management and Railway domains include the following categories:

- Availability/Continuity of operations;
- Safety;
- Data integrity;
- Data privacy and confidentiality.

In the same way as for dependability, each of the above categories/security objectives assumes a greater or lesser importance depending on the domain under consideration. Moreover, these security objectives are inter-related. Compromising data integrity can lead to financial consequences, privacy violations, a lapse in network availability, or an impact on safety. Attackers may be motivated by causing financial damage to their target or by financial gains for themselves through theft of money, energy, or vehicles. Furthermore, a security issue in one CS can quickly spread to other CSs in the SoS; a confidentiality breach in one CS can reveal authorization tokens and endanger integrity for another CS.

Another common aspect in all complex systems is that the subsystems may have undefined assumptions and requirements about the behavior of the systems they interact with. As systems evolve and use cases and components change, these wrong expectations become a major source of security and safety problems. The interaction of the systems is also the reason why an SoS composed of individual, secure CSs is not necessarily secure as a whole. Updates are difficult in an SoS since those updates may affect other CSs. The patches must be tested before being deployed, thus systems are left vulnerable for a period of time. Real-time communication requirements and safety considerations impact how security measures, such as authentication and authorization, are handled within SoSs. For example, some systems do not use cryptography at all or use symmetric cryptography since it is faster than asymmetric cryptography. Additionally, security measures must include protections against both external and internal attackers (malicious authorized users) who may have physical access to a CS.

### 7.11 Quality Metrics

The domains under study in this project concern heterogeneous systems with different characteristics. This heterogeneity is also reflected in their quality metrics. There can be metrics related to SW and HW quality, as in the intra vehicle Automotive, as well as communication and application quality, as in the V2X Communication Automotive domain. In other domains, instead, metrics are more related to objectives or performance, like the number of properties/lives saved in the Crisis Management domain, or the average number of times that a system customer experiences an outage over a specific time basis in the Energy domain.
8 REQUIREMENTS DEFINITION

This Section defines the requirements of an SoS in general, which will constitute the driving references throughout the subsequent WPs of the AMADEOS project. The analysis has been carried out synthesizing and advancing from the output of the previous Sections, feedbacks received from the Advisory Board, and the glossary reported in D2.1 [126]. These requirements will be exploited in WP2 to advance on the SoS conceptual model.

8.1 AN APPROACH TO SOS REQUIREMENTS

The objective of this Section is to define the requirements of SoSs in general, without referring to a specific context. As this can be considered a challenging and non-conventional exercise, and different approaches may be adopted and questioned, we start the discussion by illustrating the rationale we adopted.

We define meta-requirements of an SoS: potentially, our work constitutes a requirements framework, that can be applied to describe an SoS. In other words, our SoS requirements can be seen as SoS elements, peculiarities, or characteristics that should be identified when describing an SoS. This framework complies with the key viewpoints and their relations as they have been discussed for the five domains.

Before presenting the requirements, we remind that we should not confuse requirements with definitions. This Section is not explaining what an SoS, a RUMI or a CS is. These definitions are constructed in the context of WP2 and reported in D2.1, although some of them are also reported here to facilitate reading of the document.

8.1.1 Structure of Requirements

Following the approach adopted in this document, we structure requirements on the basis of the eleven key viewpoints:

1. Constraints;
2. Architecture & RUMI;
3. Semantic of Communication;
4. Dynamicity;
5. Evolution;
6. Emergence;
7. Governance;
8. Handling of time;
9. Dependability;
10. Security;

The viewpoints lead to eleven corresponding blocks of requirements, with partial overlaps and relations. Each viewpoint is further organized in sub-views (sub-blocks of requirements) as follows.

For the Constraints viewpoint, requirements are structured as follows: i) first, general requirements are presented, ii) then, relations with other viewpoints are shown, supported by a graphical representation of the different relations, iii) finally, each constraint class is investigated individually.
For the Architecture & RUMI viewpoint, requirements are organized on the basis of five traditional views for the description of architectures. Additionally, a conclusive Section on relations with other viewpoints is included, supported by a graphical representation of the identified relations.

For the remaining viewpoints, the requirements are structured in:

a. General Requirements: the main part for each viewpoint. It presents general requirements specific to the viewpoint that have no or minimal overlaps with the other viewpoints.

b. Viewpoints influencing the considered viewpoint: requirements which describe how the considered viewpoint is influenced by other viewpoints.

c. Viewpoints influenced by the considered viewpoint: requirements which describe how the considered viewpoint influences other viewpoints.

d. Relations with other viewpoints, supported by a graphical representation: summarizes relations between the viewpoints, also expressed in points b and c above.

8.1.2 Glossary

We report the main terminology from the glossary of basic SoS concepts in D2.1, which will be used in the remainder of this document. To fully understand it, we recommend that the reader be confident with the content of D2.1, and at a minimum with the definitions from D2.1 reported here.

<table>
<thead>
<tr>
<th>Autonomous System</th>
<th>A system that can provide its services without guidance by another system.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constituent System</td>
<td>An autonomous subsystem of an SoS, consisting of computer systems and possibly of controlled objects and/or human role players that interact to provide a given service.</td>
</tr>
<tr>
<td>Cyber-physical System (CPS)</td>
<td>A system consisting of a computer system (the cyber system) a controlled object (a physical system) and possibly interacting humans.</td>
</tr>
<tr>
<td>Emergence</td>
<td>A phenomenon of a whole at the macro-level is emergent if and only if it is new with respect to the non-relational phenomena of any of its proper parts at the micro level.</td>
</tr>
<tr>
<td>Entity</td>
<td>Something that exists as a distinct, and self-contained unit.</td>
</tr>
<tr>
<td>Entourage of a CPS</td>
<td>The entourage is composed of those entities of a CPS (e.g., the role playing humans, controlled objects) that are external to the cyber system of the CPS but are considered an integral part of the CPS.</td>
</tr>
<tr>
<td>Environment of a System</td>
<td>The entities and their actions in the UoD that are not part of a system but have the capability to interact with the system.</td>
</tr>
<tr>
<td>Interval of Discourse (IoD)</td>
<td>The Interval of Discourse specifies the time interval that is of interest when dealing with the selected view of the world.</td>
</tr>
<tr>
<td>Itom</td>
<td>An Itom (Information Atom) is a tuple consisting of data and the associated explanation of the data (i.e., metadata).</td>
</tr>
<tr>
<td>Prime mover</td>
<td>A human that interacts with the system according to their own goal. <em>Prime movers are CSs and have RUMIs.</em></td>
</tr>
<tr>
<td>Role player</td>
<td>A human that acts according to a given script during the execution of a system and could be replaced in principle by a</td>
</tr>
<tr>
<td><strong>RUMI (Relied Upon Message Interface)</strong></td>
<td>A message interface for the exchange of information among two or more CSs that establishes a well-defined boundary between the CSs and forms part of a backbone of an SoS system architecture.</td>
</tr>
<tr>
<td><strong>Strong Emergence</strong></td>
<td>An emergent phenomenon that is observed at the macro level is strongly emergent if, after a careful analysis of the emergent phenomenon, no trans-ordinal law that explains the appearance of the emergent phenomenon at the macro level out of the properties and interactions of the parts at the adjacent micro level is known (at least at present).</td>
</tr>
<tr>
<td><strong>Supervenience</strong></td>
<td>The principle of Supervenience states that (Sup i) a given emerging phenomenon at the macro level can emerge out of many different arrangements or interactions of the parts at the micro-level while (Sup ii) a difference in the emerging phenomena at the macro level requires a difference in the arrangements or the interactions of the parts at the micro level.</td>
</tr>
<tr>
<td><strong>System</strong></td>
<td>An entity that is capable of interacting with its environment and may be sensitive to the progression of time.</td>
</tr>
<tr>
<td><strong>System Boundary</strong></td>
<td>A dividing line between two systems or between a system and its environment.</td>
</tr>
<tr>
<td><strong>System-of-Systems (SoS):</strong></td>
<td>An SoS is an integration of a finite number of constituent systems (CS) which are independent and operable, and which are networked together for a period of time to achieve a certain higher goal.</td>
</tr>
<tr>
<td><strong>Trans-ordinal Law</strong></td>
<td>A Law that explains the emergence of the whole and the new phenomena at the macro-level out of the properties and interactions of the parts at the lower adjacent micro-level.</td>
</tr>
<tr>
<td><strong>Universe of Discourse (UoD)</strong></td>
<td>The Universe of Discourse comprises the set of entities and the relations among the entities that are of interest when modeling the selected view of the world.</td>
</tr>
<tr>
<td><strong>Weak emergence</strong></td>
<td>An emergent phenomenon that is observed at a macro level is weakly emergent if a trans-ordinal law that explains the occurrence of the emergent phenomenon at the macro level out of the properties and interactions of the parts at the adjacent micro level is known (or has been formulated post facto).</td>
</tr>
</tbody>
</table>

**Table 16 - Glossary from D2.1.**

### 8.2 **Constraints**

Amongst the presented viewpoints, the constraints viewpoint is the only one which introduces restrictions in the SoS. Obviously, the other viewpoints also force an SoS designer to introduce requirements, but such requirements are the result of investigations and reasoning and they are not imposed a-priori on the SoS.
Constraints in SoSs may refer to several classes. In the following Section and in the review on SoSs performed in this document, we identify the following classes of constraints (which may overlap with each other):

- **assets constraints**: assets are elements which are fundamental for a SoS. In financial accounting, assets are defined as resource controlled by the entity as a result of past events and from which future economic benefits are expected to flow to the entity. In our context, distinct assets may differ significantly from each other: some of them may last through time, some may need updates or replacements, some are expensive, etc. In general, a list of the assets of an SoS can identify the main elements that are fundamental for the existence of the SoS. An asset may also be a CS of the SoS. We include technology amongst the assets; technology is an element which is often both fundamental and a limitation for the existence of the SoS. Identification of assets is non-univocal. Depending on the SoS and the point of view, assets may change significantly (e.g., a manager of the railway network which describes the railway SoS may identify a train as an asset; an engineer working to design a train should probably not consider the train as an asset, because it is his final objective, the train is the whole SoS).
  
  - **Examples.** Examples of assets may be engineering structures, signalling and telecommunication, plants for transforming or distributing electric power. More specifically, examples of assets in the Railway domain are bridges, tunnels, lighting installation, access way for passengers, tracks, level crossing. In the ATM domain, examples of assets are banks, cash money. The asset Technology in the Crisis Management domain may refer, amongst others, to resilient communications, distributed ad-hoc networks, and privacy-aware collaborative systems. Other examples of assets can be patents, or communication networks.

- **standards constraints**: different standards can be required, new standards can be introduced, others may be declared outdated. The definition of an SoS, or its existence, cannot abstract from the identification of the standards to apply.
  
  - **Examples.** The ISO 26262 standard and EN 501xx family of standards regulate the safety lifecycle development process respectively in the Automotive and Railway domains. These standards are the domain specialization of IEC 61508 standard, which is the most important legislation in the field of Functional Safety.

- **system life constraints**: the system life influences and limits the development techniques, the evolution, the processes, the strategic choices, the time to deployment, the life-span and lifecycle, as well as the costs. An SoS includes several systems; while some system life may be very common in a specific SoS, some others may be unfeasible. Building an SoS may require to consider the characteristics of the SoS components and the SoS in general, to understand if costs, time, and complexity of the system life are acceptable.
  
  - **Examples.** In the Railway domain, tracks, bridges, tunnels, etc. are built with reasonable effort and supposed to last decades; in the Global ATM Network payment cards as well as terminals are designed to operate for many years. In the Automotive domain, the CSs inside a vehicle are expected to have few years of operation.

- **costs and financial constraints**: cost constraints can be very relevant limitations for the definition of an SoS and of its components. Cost effectiveness is often a keyword when thinking of SoS cost constraints.
  
  - **Examples.** Replacing payment terminals over the ATM network represents a considerable cost for the banks. For this reason it is strongly required that they are designed to operate for many years, allowing a cost-effective introduction of new technologies. A train is particularly expensive, but it lasts decades, with the appropriate maintenance. An automotive system is not particularly expensive, and it
is reproduced in thousands of models, thus maintenance due to production defects may be unacceptable.

- **governance constraints**: governance describes the way in which the SoS is managed, including the definition of roles, laws, standards, etc. This is a critical aspect because different owners/regulations related to different Constituent Systems may limit their interoperability. Note that governance is both a constraint and a viewpoint. In fact, while governance establishes constraints on the SoS (see the examples below), governance also includes management and organizational aspects that are not attributable to constraints.
  
  - **Examples**: International agreements for the circulation of money are at the basis of the worldwide diffusion of ATM.

### 8.2.1 General requirements

**[CONSTR 1]** When describing an SoS, constraints shall be identified and defined.

_SoS are characterized by a potentially vast dimension, complexity, unclear boundaries, and by a large numbers of different players involved. Understanding all the relations active within an SoS is often a challenge. Constraints are limitations to an SoS. Thus it is relevant that these are investigated attentively and are used as a guide to set the (physical, financial, interoperability, etc.) boundaries of an SoS._

**[CONSTR 2]** Constraints in an SoS are related as a minimum to the classes: assets, standards, system life, costs and financial, governance.

_Potentially, other constraints classes can be identified and added in the requirements framework, if appropriately justified. This leads to [CONSTR 3]:_

**[CONSTR 3]** Other constraints classes may be introduced if appropriately justified, whenever these cannot be reduced to any of the classes in [CONSTR 2].

**[CONSTR 4]** For each class of constraints, the constraints shall be identified and respected in the system.

_This is true in general for all the constraints we consider. For example, a standard shall be respected, a national or international law shall be applied, services shall be developed in agreement with the cost and technology._

**[CONSTR 5]** Changing the constraints may lead as a consequence to changes in the Architecture & RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics, in the Handling of Time, in the Dependability and in the Security.

_The requirement [CONSTR 5] means that changing the constraints may potentially impact all the other viewpoints. Relations between constraints and the other viewpoints are expressed below and will be further explored for each specific viewpoint._

### 8.2.2 Relations with other viewpoints

_The Constraints viewpoint has relations with all the other viewpoints. The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints will further discuss on their relations with constraints._

**[CONSTR 6]** Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

_The conceptual model that represents the constraints and the above relations is not part of this report and shall be defined in the context of WP2._
8.2.3 Assets

From requirement [CONSTR 4], it follows that assets shall be identified when describing an SoS.

Although assets are the key elements of an SoS, when describing an SoS, some assets may be difficult to identify, or a complete list of individual assets may be unfeasible in several SoS. Thus we introduce the concept of classes of assets. For example, it may be challenging and in some cases may be not useful to identify all bridges that are part of the railway SoS.

[CONSTR 7] Assets that are part of the SoS may be grouped in classes. Each class shall be described by its features, at a minimum their scope, lifespan, economic value, criticality, number of instance.

For example, a class of assets in the railway domain is the class “bridge”. As scope, a bridge allows a train to move above e.g., rivers or valleys. A bridge has a very long lifespan (potentially, centuries), its economic value and criticality are usually very high, although this may vary from bridge to bridge. Regarding number of instances, several bridges exists, although usually different from one another. Computing the exact number of instances may be difficult, thus an estimation is acceptable.

This example leads us to the following requirement [CONSTR 8].

[CONSTR 8] The features lifespan, economic value, criticality, number of instances may be estimated whenever an exact computation is unfeasible or not necessary.

When describing an SoS some assets, or classes of assets, may be later identified, added or removed (also evolution may have an influence in this).

[CONSTR 9] New assets/classes of assets may be added to the SoS, or existing assets/classes of assets may be removed from the SoS.

[CONSTR 10] The features of a class of assets may change through time.

Classes of assets may be added, removed, or changed: the features scope, lifespan, economic value, criticality, number of instance may change through time.

For example, we can consider the automotive systems, where GPS-based navigation is commonly applied. The upcoming GALILEO technology may become a relevant asset substituting GPS to improve the accuracy of localization.
8.2.4 Standards

[CONSTR 11] An SoS and/or its entities may be subject to the application of standards. From the glossary, we remind that an entity is something that exists as a distinct, independent, or self-contained unit. For example, in this case, by entities we mean (SW/HW) components, functionalities, functions, services, etc.

[CONSTR 12] Application of a standard can be mandatory or recommended, depending on the role of the standard in the SoS.

For example, in the automotive domain the standard ISO26262 is well-known and recommended but it is not mandatory. The IEC EN50126 is mandatory for railway electronic equipment.

[CONSTR 13] Standards may be newly introduced, and standards may be removed.

For example, standardization boards may create new standards, while other old standards may be updated.

[CONSTR 14] The definition of an SoS may require the introduction of appropriate standards at SoS level.

To the best of our knowledge, there are no SoS-oriented standards. New standards may be created, or existing standards may be adapted at the SoS level.

From requirement [CONSTR 4], it follows that mandatory standards shall be defined and shall be applied when designing or exercising the SoS.

8.2.5 System life

As the system life is considered an SoS constraint, we report the related requirements below.

[CONSTR 15] An SoS has its own system life which may be described by attributes as lifecycle, process, time of life, role.

In some cases the system life is simple and relatively short from a temporal perspective, in other case it may be complex, with different phases through several years. For example, the system life of a train spans over 20-30 years, with several planned maintenance steps, as prescribed by standards. The system life of the Internet includes complex processes and the difficulty in identifying a unique lifecycle. Thus the following:

Usually the process of a system is organized in steps, each one distinguished by an expected duration, and costs.

System life has relations with the system’s costs, the applied standards, the governance model, the time of life of the system, its usage/activity. Thus the following:

[CONSTR 16] The system life of the different systems may present dependencies, thus leading to connections between system lives.

For example, changes in a system may cause a cascade of updates in other connected systems.

Please note that the word “system” refers to either an SoS or a CS.

From requirement [CONSTR 4], it follows that the system life constraints shall be defined and shall be applied in designing or exercising the SoS. A methodology can be devised to compute or weigh the attributes of the system life (e.g., costs, duration, complexity of lifecycle), but it is not within the scope of this report.

8.2.6 Costs and financial aspects

[CONSTR 17] Costs and financial constraints shall influence the dimension and complexity of an SoS.

A cost sensitive approach impacts the selection of lifecycle, services, functionalities, etc.
[CONSTR 18] The identification of systems and services that are part of an SoS may be driven by a cost-oriented approach.

For example, by cost effectiveness.

From requirement [CONSTR 4], it follows that cost constraints shall be defined and shall not be exceed in designing or exercising the SoS.

8.2.7 Governance

[CONSTR 19] The dimension, complexity, services, usage of an SoS shall be regulated by governance constraints.

[CONSTR 20] Governance constraints include, but are not limited to, constraints to the players (stakeholders, users, satellite activities, funders), the technology, the evolution and progress, the financial aspects, the management of the SoS.

An example on constraints are the management roles of the railway SoS, where the centralized management of public entity was relaxed in favour of privatization and deregulation (although not relevant for this requirement, is it easy to see that this is connected with evolution of the railway SoS).

From requirement [CONSTR 4], it follows that governance constraints shall be defined and respected in designing or exercising the SoS.

Requirement [CONSTR 5] is particularly effective in case of Governance. For example, modification to the business model may imply severe changes to an SoS especially in the long term.

8.3 Architecture & RUMI

To define requirements for the viewpoint “Architecture and RUMI”, we adopt an approach typical of system requirements definition. In this viewpoint we are considering meta-requirements that can be used to derive the architectural requirements of an SoS. Consequently, we believe that a system designer approach can facilitate the structuring of this Section.

Following the ISO 10746 RM-ODP (Open distributed processing) classification [121], [122], [123], [124], [125], we have the following five sub-views:

• **Enterprise viewpoint.** The Enterprise model views the roles of various entities defined in the system, and the environment “around” each entity. It describes the rules (policies) that apply to the various roles, and the activities that are performed. An enterprise specification is a model of the system and the environment with which the system interacts. It covers the role of the system in the business, and the human user roles and business policies related to the system.

• **Information viewpoint.** The Information viewpoint deals with the information objects and their schema. An Information specification represent a system from the perspective of information definition (which part is invariant, which part is exchanged among system components, in which way information are exchanged, which flows exchange information). An information specification is a model of the information that it holds and of the information processing that it carries out. The information model is extracted from the individual entities and provides a consistent common view which can be referenced by the specifications of information sources and sinks, and the information flows between them.

• **Computational viewpoint.** The Computational viewpoint is the view of an application software architect. It represents a system as made of a set of interacting objects, which perform functions by exchanging data at interfaces. Interaction details (the mechanisms, the coding techniques, the system functions that are used to perform interactions) are not considered under this viewpoint, in the same way as, for example, disk access drivers are not seen by an
application programmer. A computational specification is a model of the system in terms of the individual, logical entities, which are sources and sinks of information.

- **Engineering viewpoint.** The Engineering viewpoint is the System Engineer perspective. Here, operating system details and supporting functions and protocols are considered, like for example security, data transfer, physical distribution of applications. An engineering specification defines a networked computing infrastructure that supports the system structure defined in the computational specification and provides the distribution transparencies that it identifies. It describes mechanisms corresponding to the elements of the programming model, by effectively defining an abstract machine that carries out the computational actions and the provision of the various transparencies needed to support distribution.

- **Technology viewpoint.** The Technology viewpoint describes the physical objects in the system, in terms of their characteristics. This includes, for example, the standards that are used for the implementation of the system. A technology specification defines how a system is structured in terms of hardware and software components.

### 8.3.1 Enterprise view

The enterprise view defines the objects, the environment, the roles, and the activities executed in the system. We identify meta-requirements for:

- the SoS, the CS and their composition;
- interactions in an SoS;
- the humans and the entourage;
- boundaries.

First we define the type of SoS.

[ARCH 1] An SoS shall have a type, to be selected amongst Directed, Acknowledged, Collaborative and Virtual SoS.

For the explanation on the types of SoSs, see Section 1.1.2.

Compositionality is a keyword at the basis of our SoS architecture. Thus we need requirements which express the compositionality of an SoS:

[ARCH 2] Two or more CSs may be organized in formal hierarchical or holarchical levels to form another CS or an SoS.

Obviously, the "composition" is driven by the definition of an appropriate RUMI. For the explanation of formal hierarchy and holarchy, see Section 7.

The relied upon message interface (RUMI) is the interface placed at the boundary of two interacting autonomous CSs. Within an SoS, all CSs will interact through a RUMI (the RUMI guides the interaction between CSs). Information is exchanged by *Itoms*. Thus the following set of requirements:

[ARCH 3] CSs shall interact exclusively through RUMIs to exchange Itoms.

The basic elements of the SoS architecture are CS, RUMI and Itoms.

Requirement [ARCH 3] includes the word "exclusively", considering the RUMI as the only means for communication in SoS. Noteworthy, this is valid not only for Machine-2-Machine interactions, but also for Human-Machine interactions (and with living beings in general).

[ARCH 4] The role and actions of the entourage are defined by using RUMIs.

We remember from the glossary that the entourage of a CS includes humans and controlled objects. Thus, a CS will interact with the users or operators through its RUMI.
[ARCH 5] The number and dimension of CSs, RUMIs and their interactions are limited only by the Constraints.

Requirement [ARCH 5] defines a limit on the dimension of the SoS architecture. Requirement [ARCH 5] sets the maximum acceptable architectural limits. For example, in the Railway domain, there is a constraint on the capacity of a line (assets and standards constraints): this limits the number of trains active on the line and interacting with the RBC through a RUMI (in the railway domain, the RUMI and the Itoms in this case are defined by the ERTMS standard).

This means that, in case the constraints do not set precise limits, an SoS can be potentially without boundaries, or with boundaries which are not very clearly defined. In other words, sometimes defining the boundaries of an SoS may be very difficult. Also this implies that boundaries are not born from the identification of the connections between systems, contrary to what usually happens in ICT architectures, but from constraints, e.g., assets (including technology), standards, etc.

[ARCH 6] Changes in the Architecture & RUMI in a CS or an SoS may influence the semantic of communication, dynamicity, emergence, governance, quality metrics, handling of time, dependability, security of the CS or SoS.

Requirement [ARCH 6] builds a relation between the different viewpoints and the Architecture & RUMI viewpoint. For an example on dynamicity, let us consider the Smart Grid. The growth of the Advanced Metering Infrastructure (AMI), which increases the number of Smart Meters in the network, may introduce dynamicity effects as new or different interactions, area-effect intervention for maintenance, new options offered to the operators.

Regarding influence on governance, this is the case for what concerns the management aspects of governance, while it is not true when governance is considered as a constraint.

[ARCH 7] Changes in Architecture & RUMI in a CS or an SoS may promote the evolution of the CS or SoS.

For example, the architecture of a system can be changed from a rigid one to a modular one which is designed for evolution, thus affecting/improving the capability of the system to evolve.

[ARCH 8] When requirements on dependability, security, handling of time, quality metrics are demanded, SoS shall be observed for monitoring purpose.

With this requirement, we say that an SoS includes monitoring solutions. Note that this requirement builds a relation with the dependability, security, handling of time viewpoints.

8.3.2 Information view

In ODP, the information view focuses on the information exchanged. While the Enterprise view mainly focuses on the interacting entities, here the focus is on the structure of the messages, the possible protocols and the semantic of the information.

(see also [ARCH 3])

[ARCH 9] The RUMI shall describe the Semantic of Communication through the definition of Itoms.

Requirement [ARCH 9] says that the semantic of the information exchanged is described in the RUMI, and that this information is contained within Itoms. This requirement establishes a relation between the viewpoints Architecture & RUMI and Semantic of Communication.

[ARCH 10] A precise specification of the RUMI shall be provided in the value domain and in the temporal domain.

A RUMI should be well-specified in the domains of value and time. Changes in one CS should not unintentionally propagate to another CS via an ill-specified RUMI. Thus the following [ARCH 11]:

[ARCH 11] The RUMI shall identify the choreography of the CSs and their entourage.
In the requirement [ARCH 11] we use the term choreography from the Web Service community. A choreography describes collaboration protocols of cooperating Web Service participants, in which services act as peers, and interactions may be long-lived and stateful. We reuse this term with a very similar meaning: in our SoS: the RUMI should explain which Itoms are exchanged, and how they are exchanged between the different CSs. If there is an expected sequence of messages to be exchanged between CSs, it can be specified in the RUMI. Regarding interfaces commonly applied in Web Services (e.g., the standards WSDL and WADL), the RUMI also contains information on the semantic of the interface.

[ARCH 12] Limitations on the possible specifications of RUMIs are due exclusively to the constraints.

This says that potentially a RUMI may describe any interaction between CS, and the Itoms may contain any kind of information. The boundaries on this are given by the constraints of the SoS, which may include formal contracts, SLA (Service Level Agreements), standards, etc. For example, the interfaces for the AMI in the energy domain are defined in standards which limit the syntactic structure and the semantic of the Itoms exchangeable between the CSs. This requirement defines a relation between the Architecture & RUMI viewpoint and the Constraints viewpoint.

8.3.3 Engineering view

In this view we consider:

- time and handling of time;
- dependability (including safety);
- security;
- emergence;
- architectural solutions (algorithms and techniques)

from the viewpoint of a system architect. Note that viewpoints Handling of Time, Dependability, Emergence and Security will be also analyzed in a separate Section. Amongst other things, this view defines relations between the viewpoint Architecture & RUMI and the above mentioned viewpoints.

We first address time and handling of time:

[ARCH 13] The notion of time shall be set for the SoS, defining whether a shared time base is needed as well as solutions for the resilient time synchronization.

Requirement [ARCH 13] follows from the definition of system, which is "sensitive to the progression of time". An SoS as well is sensitive to the progression of time, may have a shared time base (UTC, TAI, or even a self-defined one), and may require synchronization solutions, and a certain level of synchronization accuracy.

[ARCH 14] Solutions for external or internal time synchronization may be identified.

[ARCH 15] The shared time base will be a global time base from standards or a self-defined time base.

Examples of time base from standards are UTC, TAI.

Then we include also architectural requirements on dependability and security:

[ARCH 16] Solutions for handling of time, dependability and security shall be considered at SoS level and may be considered at lower levels than SoS level.

Requirement [ARCH 16] says that it is mandatory to consider architectural solutions for dependability, handling of time and security at SoS level, while it is not mandatory to consider solutions for the lower levels (the systems that compose the SoS).
Then we introduce architectural requirements on emergence. In the literature and in common practice, there are no widely accepted solutions to identify and manage emergence. Thus building architectural requirements for emergence is not easy.

An SoS has emergence phenomena (at least weak emergence). The requirement below introduces this concept. This requirement on emergence should be read together with the Emergence viewpoint.

Emergence phenomena refers to interactions between CSs and entourage that are within the operations allowed by the RUMI, although potentially such interactions were not foreseen.

As all relations are driven by the RUMI, also emergence comes from interactions between RUMIs. We do not explicitly mention SoS in the requirement as it may confuse a reader (the SoS is often the “upper level” mentioned in the requirement).

[ARCH 17] The RUMI interactions of CSs and/or entourages will generate emergent phenomena at the upper level.

Last, we consider the generic architectural solutions (algorithms and techniques):

[ARCH 18] Limitations on the applicable algorithms and techniques are set exclusively by the constraints.

Given requirement [ARCH 18], we consider that assets, standards, costs are the main constraints for the algorithms and techniques. For example:

- **Asset (technology) and cost.** Magnetic Levitation trains have excellent performance but very high costs which limit their usage for long-range trains. This is amongst the limitations to the diffusion of Magnetic Levitation technology.

- **Standards.** Current railway safety-critical systems very rarely, if not at all, adopt Java, PHP or RUBY software due to constraints in the standard IEC/EN 50128.

- **Costs.** Biometric systems as replacement of magnetic cards may represent a possible evolution in the ATM field, for trust and secure authentication (once technological issues are solved). However this would require to replace the whole ATM network worldwide; the cards will most likely continue to exist in our wallets for quite a long time, even after the introduction of biometric ATM.


8.3.4 **Technology view**

[ARCH 19] Limitations on the applicable technologies (standards, instruments, processes and methodologies, hardware) are set exclusively by the constraints.

This requirement defines relations between the Architecture & RUMI and the Constraints viewpoints.

8.3.5 **Computational view**

In ODP, computational view is in particular for the software engineer.

[ARCH 20] Limitations on the applicable software objects are set exclusively by the constraints.

This requirement defines relations between the Architecture & RUMI and the Constraints viewpoints.

8.3.6 **Relations with other viewpoints**

The architecture & RUMI viewpoint has relations with all the other viewpoints. The following requirement explains that these relations should be identified and reported when describing an
SoS. The requirements in the other viewpoints will further discuss on their relations with constraints.


![Diagram of Architecture & RUMI viewpoint: relations with other viewpoints.](image)

**8.4 Semantic of Communication**

Semantic of communication refers to the meaning of the information exchanged between different CSs.

The requirements on the semantic of communication have strong affinity with the requirements on information exchanged presented in the Architecture & RUMI viewpoint.

**8.4.1 General requirements**

[SEM 1] Semantic of communication shall be described by Itoms exchanged through the RUMIs.

*That is, the semantic of communication identifies the information exchanged. Requirement [SEM 1] introduces a relation between viewpoints Semantic of Communication and Architecture & RUMI.*

[SEM 2] Semantic of communication shall be defined for the SoS.

[SEM 3] Semantic of communication may be defined for each CS that is part of the SoS.

*Requirement [SEM 2] says that Itoms exchanged shall mandatorily be identified for the SoS, while requirement [SEM 3] says that it is optional in case of the CSs composing the SoS. The exact definition of each CS and its composition in a SoS is not always mandatory.*
[SEM 4] Semantic of communication shall be defined for the entourage of the SoS.

[SEM 5] Semantic of communication may be defined for the entourage of each CS that is part of the SoS.

Requirements [SEM 4] and [SEM 5] correspond to requirements [SEM 2] and requirement [SEM 3], respectively, considering the entourage of the SoS and the entourage of the CS.

[SEM 6] Sources of the information shall be located in the SoS and its entourage.

Requirement [SEM 6] says that the information may originate from potentially any sources: CSs, autonomous systems, entourages, SoS, etc.

[SEM 7] Semantic of communication shall be defined at the boundaries of any system.

The requirement [SEM 7] introduces the concept of information exchanged at the boundaries.

8.4.2 Viewpoints influencing Semantic of Communication

[SEM 8] The only limitations to the semantic of information exchanged are given by constraints.

Semantic of communication may refer to any kind of information, as far as it can be expressed through a RUMI and it does not violate constraints. Examples of the semantic of information are actuators commands, localization information, monitoring information, infotainment information, personal data. Other kind of information may be defined, basically there are no limitations to the kind of information that can be exchanged in an SoS.

This requirement introduces a relation between the viewpoints semantic of communication and Constraints.

For example, among constraints, we identify that semantic of communication may be regulated by standards. Standards may impose rules on the kind of information exchanged, as well as their format.

[SEM 9] The semantic of communication shall be compliant to the Architecture & RUMI and the constraints and it may change due to changes in the Architecture & RUMI, evolution, and constraints.


We present some examples. Regarding the Constraints viewpoint: a new standard introduced may require new Itoms, for example in case of a new standard for the communication within the AMI. A new technology introduced (a new asset) may allow the exchange of different Itoms.

For example, UMTS was one of the enabling technology for several of the applications running on our mobile phones, and enabled the exchange of complex information. Regarding the Architecture & RUMI viewpoint, changes in the architecture may lead to new or different Itoms and different patterns for the exchange of Itoms, for example in case of dynamic self-organizing systems. Regarding the evolution viewpoint, the Crisis Management systems of the future may include different or new components/information sources, requiring changes in the semantic of communication to exploit the new information offered.

8.4.3 Viewpoints influenced by Semantic of Communication

[SEM 10] Changes in the semantic of communication may influence dynamicity and emergence.

Requirement [SEM 10] introduces a relation between the viewpoints semantic of communication and Dynamicity and between semantic of communication and Emergence.

For example, in the Smart Grid domain, the introduction of new Itoms in a Smart Grid (e.g., collected by some new sensors) may allow to collect different measurements and achieve a more
efficient energy management (dynamicity). Analogously, the introduction of new items in a Smart Grid may lead to semantic incompatibility between CSs; thus it follows that semantic of communication may lead to different interactions among CSs, thus building new relations and leading to new emergence phenomena.

8.4.4 Relations with other viewpoints

The semantic of communication viewpoint has relations with other viewpoints, as reported above and summarized below. The following requirements explicit these relations, which should be identified and reported when describing an SoS.


Figure 23 - Semantic of Communication viewpoint: relations with other viewpoints.

Please note that emergence, dependability, security and quality metrics are not mentioned before in this viewpoint. The relations with semantic of communication are detailed in the following viewpoints emergence, dependability, security and quality metrics.

The conceptual model for semantic of communication as well as for the relations with the other viewpoints is not part of this report.

8.5 Dynamicity

In an SoS frequent changes may happen, that may have many effects on the SoS. For example, changes can lead to new (strong and weak) emergence phenomena. Changes may be well-planned in advance, or may be abrupt, depending on the SoS, its distribution, and its "degrees of freedom".

Dynamicity refers to short-term changes. Note that dynamicity is due to possibly abrupt changes in an SoS which operates within the established constraints.

8.5.1 General requirements

[DYNAM 1] Dynamicity is a potential attribute of systems and/or their environment.
Here note that we do not refer exclusively to CS and SoS and their entourage, but to any system in general. Being "dynamic" is not mandatory; also different systems can be dynamic to different extents.

8.5.2 Viewpoints influencing Dynamicity

[DYNAM 2] The dynamicity of the SoS/CS and its entourage shall be limited in frequency, number, and dimension by its constraints and the Architecture & RUMI.

First of all, note that now we move the perspective from the generic "system" to the CS and SoS. Some examples below.

Constraints. Standards, costs, rules, assets set boundaries to the dynamic behavior, in terms of "dimension" of the change, frequency of the changes, number of changes in the system.

Architecture & RUMI. We have observed that there are different kinds of SoS (directed, acknowledged, collaborative, virtual). Each one of them has different architectural demands, showing a limited dynamicity in directed SoS, and increasing dynamicity as we move towards the virtual SoS. Dynamicity may also refer to system updates, modifications, connections, etc.

Requirement [DYNAM 2] defines a relation of the viewpoint Dynamicity with the viewpoints Constraints, Architecture & RUMI.

Note that requirement [DYNAM 2] also gives information on the origin of dynamicity. Dynamicity is originated by different causes. [DYNAM 2] says that constraints and Architecture & RUMI offer a "space" where dynamicity can originate. For example, a standard which is not restrictive, and a mobiquitous system, can offer the stage for dynamic behaviors.

[DYNAM 3] Dynamicity may be caused by modifications in the Architecture & RUMI, semantic of communication and constraints.

Example on semantic of communication and dynamicity. Several large scale systems nowadays show trends towards self-organization and adaptiveness. The items that are exchanged by these system may change abruptly, due to new services which are required, changes in the environment. Thus semantic of communication is not fixed - and consequently, changing the semantic of communication may cause dynamicity.

8.5.3 Viewpoints influenced by Dynamicity

In general, changes due to dynamicity are possible. As seen in requirement [DYNAM 2], limitation to dynamicity are related to Constraints, and Architecture & RUMI. Still, the dynamic behavior may cause a cascade of modifications in the Architecture & RUMI, and in the semantic of communication. Thus the following requirements:

[DYNAM 4] Changes in the dynamicity of an SoS/CS or its entourage may require to change the Architecture & RUMI and the Semantic of Communication.

[DYNAM 5] In an SoS, the influence of dynamicity on dependability, security, and handling of time shall be analyzed and, if necessary and possible, may be mitigated.

The term influence does not refer to requirements (dependability, security and time requirements are set a-priori to system design and do not change easily), but it refers to the satisfaction of those requirements. Examples:

- **dynamity may cause new security breaches.** For example, a new system administrator of a complex service-oriented system, without appropriate training, may leave open some security breaches because he fails in keeping all the services and systems updated.

- **dynamity may influence system dependability:** for example, erroneous services orchestrations may reduce the Quality of Service offered to the users.
• dynamicity may influence handling of time, for example abruptly increasing the workload of a web market may cause the service to slow down, thus missing deadlines (the service slows down thus users have an unpleasant experience, financial transactions fail, etc.).

Requirement [DYNAM 5] explains that dynamicity influences some viewpoints, and that mitigations to the detrimental effects of dynamicity may be set.

We now consider dynamicity as a possible cause of emergence:

[DYNAM 6] Changes in the dynamicity of a CS/SoS may give rise to emergence phenomena.

We believe that dynamicity does not directly affect the viewpoints Evolution and Constraints. This because:

a. Evolution is for long-term changes, while dynamicity is short-term;
b. Constraints: dynamicity is limited by the constraints. The opposite is not true: dynamicity does not change constraints.

8.5.4 Relations with other viewpoints

The Dynamicity viewpoint has relations with other viewpoints. The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints will further discuss on their relations with constraints.


![Figure 24 - Dynamicity viewpoint: relations with other viewpoints.](image)

Please note that quality metrics is not mentioned before in this viewpoint. The relations with dynamicity are detailed in the following viewpoint quality metrics.

The conceptual model for dynamicity as well as for the relations between dynamicity and the other viewpoint is not part of this report.
8.6 **Evolution**

CSs and SoSs are evolutionary. The SoS and the CSs that compose the SoS may change through time, varying the intra-domain interactions and communication. External boundaries may be subject to modifications.

8.6.1 **General requirements**

[EVOL 1] Evolution may be an attribute of a CS/SoS.

The requirement does not mention explicitly the SoS but obviously the SoS is included. To understand [EVOL 1], it should be compared to [DYNAM 1]. Evolution is not mandatory and it is an attribute exclusively of SoS/CS (not of other kinds of systems).

[EVOL 2] Evolution of a CS may modify the boundaries of the CS.

Evolution may change the boundaries of the CS or SoS. For example, as a result of an evolutionary process, the SoS may have a different entourage, or may be connected to new CSs.

8.6.2 **Viewpoints influencing Evolution**

[EVOL 3] Evolution of SoS/CS shall be governed exclusively by the constraints.

[EVOL 4] Evolution of the SoS/CS may be originated by modifications in the constraints.

For example, we can consider the long-term evolution of the crisis management SoS as a consequence of changes in the governance model “prescribing” the system architecture at incident time (remind that here governance is a constraint).

8.6.3 **Viewpoints influenced by Evolution**

[EVOL 5] Evolution of an SoS/CS may give rise to modifications in the Architecture & RUMI.

Note that as per today, system evolution is essentially maintenance of the RUMI (evolvability of the RUMI).


Requirement [EVOL 6] follows straightforwardly from requirement [EVOL 5].


The above three requirements relate the viewpoint Evolution to the viewpoints Architecture & RUMI, Semantic of Communication, Emergence.

8.6.4 **Relations with other viewpoints**

Summarizing:

- Constraints (including Governance) changes as well as Architecture & RUMI changes may be a stimulus to Evolution;

- Evolution may cause changes to the boundaries, and may influence modifications to the Architecture & RUMI, the semantic of communication, and give rise to emergence.

The Evolution viewpoint has relations with other viewpoints.

The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints further discuss their relations with evolution.

[EVOL 8] Relations in an SoS between the viewpoint Evolution and the viewpoints Architecture & RUMI, Constraints (and especially Governance), Semantic of Communication, Emergence, Quality Metrics shall be identified.
Figure 25 - Evolution viewpoint: relations with other viewpoints.

Please note that quality metrics is not mentioned before in this viewpoint. The relations with evolution are detailed in the following viewpoint quality metrics.

The conceptual model for evolution and the above mentioned relations is not part of this report.

We would like to remark that the viewpoints connected to evolution do not include dependability, security and handling of time. While modifications to a CS or to the SoS may lead to a change in the security, dependability and time handling of the system, we believe that such changes are not the direct result of evolution. Other viewpoints, connected to evolution, are responsible for them. For example, evolution may lead to changes in the Architecture & RUMI and consequently changes in the solutions for security. Or as another example, evolution may lead to (detrimental) emergent phenomena that may impact dependability.

8.7 Emergence

Emergence concerns phenomena that manifest at the macro-level and are not observable at micro-level. The fact that an emergent behavior is considered positive, negative or neutral, is usually dependent on the observer’s criteria. In general, emergent behavior is not a-priori positive or negative.

It is both fundamental and deeply challenging to predict emergent behaviors, especially detrimental ones.

8.7.1 General requirements

Emergence phenomena may happen:

[EMERGE 1] An SoS/CS will be subject to emergent phenomena in a specific Interval of Disclosure.

Emergence is an intrinsic property of SoS/CS, which are at least characterized by weak emergence to reach their higher goal.

We consider first detrimental, then non-detrimental emergence.
[EMERGE 2] Appropriate efforts shall be devoted to observe and predict detrimental emergence phenomena and mitigate their effect on the SoS.

In requirement [EMERGE 2] we concentrate on the SoS and not on the CS. The requirement [EMERGE 2] says that in an SoS, to the best of our effort we should try to predict detrimental emergence phenomena, in order to mitigate their impact on the system.

[EMERGE 3] Efforts may be devoted to observe and predict and measure non-detrimental emergence phenomena in an SoS.

The difference between requirement [EMERGE 2] and requirement [EMERGE 3] is that the first one expresses a mandatory action (shall), while the latter expresses an optional action (may).

8.7.2 **Viewpoints influencing Emergence**

The following viewpoints may originate emergent phenomena:

[EMERGE 4] Modifications to the Architecture & RUMI, the entourage, the constraints, the Semantic of Communication of an SoS/CS may influence emergence phenomena.

Changing the architecture may pave the way to emergence phenomena. For example, adding new components which introduces new functionalities. Or adding new components may change the error model.

Changing the semantic of communication may also pave the way to emergence phenomena. For example, introducing new Itoms which enables new interoperability between CSs.

Similar reasoning for the constraints.

[EMERGE 5] Emergence may be caused both by the normal and the dynamic behavior of an SoS/CS.

Already discussed in the dynamicity viewpoint.


Already discussed in the evolution viewpoint.

8.7.3 **Viewpoints influenced by Emergence**

[EMERGE 7] Emergence phenomena in an SoS/CS may cause violations to the constraints, handling of time, dependability and security of the SoS/CS.

Requirement [EMERGE 7] considers both detrimental and non-detrimental emergence.

8.7.4 **Relations with other viewpoints**

The emergence viewpoint has relations with other viewpoints. The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints further discuss their relations with emergence.

Please note that quality metrics is not mentioned before in this viewpoint. The relations with emergence are detailed in the following viewpoint quality metrics.

The conceptual model for emergence as well as for the above mentioned relations is not part of this report.

8.8 **GOVERNANCE**

Governance describes the way in which the SoS is managed. It includes the definition of roles, laws, standards, but also management and organizational aspects. Consequently governance is considered in this document both as a constraint and a viewpoint.

As governance is also a constraint, relations to the constraints viewpoint are implicit and not reported here.

8.8.1 **General requirements**

[GOV 1] Incentives must be provided and trust must be established among the CSs in order that the selected CSs are cooperative.

[GOV 2] Governance shall be a constraint for an SoS and its entourage, and it may be a constraint for a CS and its entourage.

Governance is an optional constraint for a CS, and a mandatory constraint for an SoS.

[GOV 3] Governance may change through time.

Modification of governance policies may be due to financial analysis, human thinking, etc.

[GOV 4] Governance shall contribute to the definition of SoS boundaries.

8.8.2 **Viewpoints influencing Governance**

Governance is influenced by other constraints, for example for what concerns organizations and management.
[GOV 5] Governance may be influenced by all constraints, in particular costs, system life and assets.

8.8.3 **Viewpoints influenced by Governance**

See constraints viewpoint Section 8.2.7.

8.8.4 **Relations with other viewpoints**

The governance viewpoint has relations with other viewpoints, following the same relations of the constraints viewpoint.

Please refer to requirement [CONSTR 6].

![Figure 27 - Governance viewpoint: relations with other viewpoints.](image)

The conceptual model for governance and for the relations with the viewpoints are not part of this report.

8.9 **Handling of Time**

The handling of time is a critical aspect in terms of synchronization and real-time requirements. It is worth noticing that most of systems and entities are sensitive to the progression of time: living beings, materials, computer systems age. As a consequence, time is a relevant concept for SoSs, CSs, autonomous systems, entourage, environment, etc.

8.9.1 **General requirements**

[TIME 1] A system may be sensitive to the progression of time.

Requirement [TIME 1] is somehow trivial and follows from the definition of system (an entity that is capable of interacting with its environment and may be sensitive to the progression of time). The
most interesting part is that there may be systems which are not sensitive to the progression of
time.

[TIME 2] An SoS shall be sensitive to the progression of time and shall have time
requirements describing its handling of time.

An SoS is always sensitive to the progression of time; from its definition, the SoS exists in a
"period of time to achieve a certain higher goal". When describing an SoS, time requirements shall
be provided that describe how the SoS handles time.

[TIME 3] The entourage of the SoS may be sensitive to the progression of time and it may
have time requirements describing its handling of time.

Requirement [TIME 3] is similar to [TIME 2], but for the entourage it is not explicitly required that it
is sensitive to the progression of time.

[TIME 4] Time requirements shall be organized in: i) timeliness requirements; ii) time
synchronization requirements.

Timeliness requirements are mainly for real-time SoSs, while time synchronization requirements
are mainly for synchronization of the CSs, providing a common view to the CSs in order to allow
them to achieve the higher goals of the SoS.

[TIME 5] Every system in an SoS subject to physical time requirements shall be able to
measure time with an appropriate uncertainty.

If a system wants to satisfy its time requirements, it needs a solution (e.g., a tool) to measure time.
Time may be measured by several different ways e.g., using a meridian, a quartz clock, an egg-
timer, a pendulum, an atomic clock.

[TIME 6] An SoS and its CSs subject to time synchronization requirements may have a global
time reference shared by all CSs.

The above requirement means that for CS (and SoS), there is a global time reference, and the
systems that compose the CS are aware of it.

[TIME 7] A CS shall be able to achieve a quality of time synchronization which is deemed
sufficient to satisfy the time synchronization requirements of the CS.

[TIME 8] A CS shall execute real-time protocols which are deemed sufficient to satisfy the
timeliness requirements of the CS.

Requirements [TIME 7], [TIME 8] set the basis to say that the overall CS or the systems
composing it shall implements solutions such that the CS satisfies its time requirements.

8.9.2 Viewpoints influencing Handling of Time

[TIME 9] (Part of) time requirements may be imposed by the Constraints.

For example, a standardization agency may set requirements on the duration of a financial
transaction.

For simplicity, the opposite of [TIME 11] (i.e., Design solutions to meet dependability and security
requirements may influence time handling design solutions) is not reported here, see instead
[DEPEND 15], [SECURITY 10].

8.9.3 Viewpoints influenced by Handling of Time

[TIME 10] Design solutions to meet time requirements shall be defined in the Architecture &
RUMI.

Solutions may refer to:

- solutions for logical clocks, hardware clocks,
• solutions for resilient time synchronization,
• solutions for real-time distributed algorithms,
• etc.

Note that we do not report on assessment of the Handling of Time requirements. Assessment is done in the Dependability viewpoint. In other words:
• time requirements are set;
• design solutions are in the Architecture & RUMI;
• the Dependability viewpoint will consider the assessment of the Architecture & RUMI.

Note that security, dependability and time requirements often cannot be considered independently, but there are strong relations between them (especially when moving towards the definition of the architectural solutions, each of these viewpoints can influence the other two).

[TIME 11] Design solutions to meet time requirements may influence dependability and security design solutions.

8.9.4 Relations with other viewpoints

The Handling of Time viewpoint has relations with other viewpoints. The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints further discuss their relations with Handling of Time.


This requirement also includes the relations between Handling of Time and Dependability and Security. In fact, these three viewpoints are often tightly connected, and the solutions adopted for one of them may influence the other two.

Figure 28 - Handling of Time viewpoint: relations with other viewpoints.
Please note that emergence and quality metrics are not mentioned before in this viewpoint. The relations with handling of time are detailed in the respective viewpoints emergence and quality metrics.

The conceptual model for Handling of Time and for the above mentioned relations is not part of this report.

8.10 DEPENDABILITY

We remark that from the taxonomy in [57], and contrary to some industrial glossaries as the aerospace one, safety is considered an attribute of dependability.

We present:
a. requirements to specify "dependability requirements". Dependability requirements are usually set prior to system design, a typical example from the railway domain is "The Safety Integrity Level of the component shall be SIL 4".
b. requirements that describe the dependable design and assessment. Note also requirement [ARCH 8] on observation of the system.

8.10.1 General requirements

[DEPEND 1] Dependability requirements may be assigned to a system.

The environment and entourage do not have dependability requirements.

[DEPEND 2] Dependability requirements shall be assigned to an SoS.

An SoS mandatorily has dependability requirements. When describing an SoS, the dependability requirements of the SoS shall be identified, while it is optional to identify the dependability requirements of the systems composing the SoS.

[DEPEND 3] Any system may be assessed to verify that its requirements are respected.

In requirement [DEPEND 3], by system we also mean an SoS. Assessment here includes verification and validation. Assessment may be qualitative or quantitative. Potentially, it may include both offline and runtime approaches; in general, we do not present restrictions on the kind of assessment techniques applicable.

Also requirement [DEPEND 3] should be regarded in relation to requirement [TIME 10] and the related discussion. The assessment of time and security requirements is in [DEPEND 3].

[DEPEND 4] A system may be observed for monitoring purpose.

In general, it is required to instrument the system for monitoring purposes, and collect measurement results.

[DEPEND 5] The entourage of a CS may have non-unique, different perceptions of dependability, and different needs.

[DEPEND 6] The CS and/or entourage of a CS may have trust requirements.

8.10.2 Viewpoints influencing Dependability

[DEPEND 7] (Part of) dependability requirements of an SoS may be imposed by the Constraints. For example, constraints are standards, contracts, service level agreements, etc.

[DEPEND 8] (Part of) assessment and monitoring solutions of an SoS may be limited and/or imposed by the Constraints.
Assessment and monitoring techniques may be defined by the constraints. For example, by the system life constraint. Note that assessment here includes dependability assessment, security assessment, and time analysis.

Examples. Standards may indicate what assessment activities to perform. Note that this includes dependability assessment, security assessment, time assessment. System life, cost, governance, technological constraints may limit the possible monitoring solutions to apply in the SoS. Standards may indicate what should be mandatorily monitored.

[DEPEND 9] Dependability requirements may change due to modifications in the Constraints.

A change in the Constraints is for example a change in a safety standard, and a consequent change in the dependability requirements.

From [EMERGE 7], detrimental emergence phenomena may occur in SoS, and may cause system unreliability, safety violations, etc.:

[DEPEND 10] Detrimental effects of emergence phenomena may influence the dependability of an SoS/CS.

[DEPEND 11] A system shall be mandatorily assessed if it is required by the Constraints.

Example. If a standard requires a mandatory assessment, the system shall be assessed. Note that assessment can be performed even if it is not required by the constraints, but in such case it is on "volunteer" basis and not regulated in the SoS.

8.10.3 Viewpoints influenced by Dependability

[DEPEND 12] Design solutions to meet dependability requirements shall be defined in the Architecture & RUMI.

[DEPEND 13] Dependability requirements may influence the Architecture & RUMI design.

For example, the dependability requirement "The Safety Integrity Level of the component shall be SIL 4" requires to adopt architectural solutions for fail-safe systems.

Changes in the dependability requirements may imply modifications to the Architecture & RUMI and a re-assessment of the system.

The above requirement says that after changing dependability requirements, it may be necessary to modify the Architecture & RUMI with appropriate solutions, and it may be necessary to assess the modified system.

[DEPEND 14] Trust requirements set on CSs and the entourage may influence the Semantic of Communication and the Architecture & RUMI.

From [TIME 11]:

[DEPEND 15] Design solutions to meet dependability requirements may influence handling of time and security design solutions.

8.10.4 Relations with other viewpoints

The Dependability viewpoint has relations with other viewpoints. The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints further discuss their relations with Dependability.

Please note that dynamicity and quality metrics are not mentioned before in this viewpoint. The relations with dependability are detailed in the respective viewpoints dynamicity and quality metrics.

The conceptual model for dependability and for the above mentioned relations is not part of this report.

8.11 Security

Security is the final viewpoint we consider. At a first glance, most of the requirements are similar to the Dependability viewpoint. This is in line with the definition of security that we adopt from [57], where security is a composition of the attributes of confidentiality, integrity, and availability. Note that in this Section we also include privacy requirements.

In the following, we have:

a. requirements that address "security requirements". Security requirements are usually set prior to system design, though they may depend on use cases and change over time.

b. requirements that address "security" in general, also considering privacy.

8.11.1 General requirements

[SECURITY 1] Security requirements may be assigned to a system.

The environment and entourage do not have security requirements. However, the attacker is part of the environment and might influence the requirements.

[SECURITY 2] Security requirements shall be assigned to an SoS.

An SoS mandatorily has security requirements. When describing an SoS, the security requirements of the SoS shall be identified, while it is optional to identify the security requirements of the systems composing the SoS.

[SECURITY 3] A system may have privacy requirements.
This requirement includes SoS, CS, autonomous systems, etc. The system may have or not privacy requirements, depending on the environment, entourage and the information exchanged.

8.11.2 Viewpoints influencing Security

[SECURITY 4] (Part of) security and privacy requirements may be imposed by the constraints.

For example, constraints are standards, contracts, service level agreements, etc. Use cases, often used to derive security requirements of a system, are not explicitly mentioned here: obviously they shall respect the constraints on the SoS and are built within the boundaries set by the constraints.

Detrimental emergence phenomena may occur in a SoS, and may cause security and privacy breaches. From [EMERGE 7], detrimental emergence phenomena may violate the security and privacy requirements of a system. This leads to the following requirement:

[SECURITY 5] The security and privacy of a system may be affected by detrimental emergence phenomena.

It should be noted that in an SoS, if the attacker evolves/new attack paths are available, security needs to evolve too.

[SECURITY 6] The security and privacy of a system may be assessed.

This requirement sets a relation to the Dependability viewpoint.

8.11.3 Viewpoints influenced by Security

[SECURITY 7] Design solutions to meet security requirements shall be defined in the Architecture & RUMI.

[SECURITY 8] Changes in the security and privacy requirements may imply modifications to the Architecture & RUMI and a re-assessment of the system.

The above requirement says that after changing security and privacy requirements, it may be necessary to modify the Architecture & RUMI with appropriate solutions, and assess the modified system.

[SECURITY 9] Privacy requirements set on CSs and the entourage may influence the Semantic of Communication and the Architecture & RUMI.

For example, the Itoms exchanged with the humans, and the RUMI between humans and the CS, may require a trust-sensitive approach (this obviously depends on the services: some are trust-sensitive, others are not).

From [TIME 11]:

[SECURITY 10] Design solutions to meet security requirements may influence dependability and handling of time design solutions.

8.11.4 Relations with other viewpoints

The Security viewpoint has relations with other viewpoints. The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints further discuss their relations with Security.

Please note that dynamicity and quality metrics are not mentioned before in this viewpoint. The relations with security are detailed in the respective viewpoints dynamicity and quality metrics. The conceptual model for security and for the above mentioned relations is not part of this report.

### 8.12 Quality Metrics

The quality metrics viewpoint is a transversal viewpoint that defines metrics to measure the quality of the considered system.

Examples of quality metrics from the domains identified in this document are:

- **Automotive**: analyzability, changeability, stability, and testability as Software Quality criteria (threshold values are associated to each criterion by the user); Hardware quality metrics as robustness and reliability criteria, required by the ISO26262; In VANETS, time to establish and ad-hoc connection with another vehicle.

- **Railway**: Safety Integrity Levels (SIL), each expressing different safety requirements; SILs are related to the whole system or to parts of the system.

- **Energy**: Quality Metrics to evaluate the reliability of the power grid.

- **Crisis Management**: time to establish ad-hoc connections (e.g., communication / cooperation links) between CM organizations.

#### 8.12.1 General requirements

From the above examples, we can derive the following requirement:

[QUALITY 1] Quality metrics may be required by an SoS.

**Quality metrics may be related to any part or perspective of the system.**

[QUALITY 2] Quality metrics may be related to software, hardware, the communication among CSs and to performance of CSs and SoSs, as well as non cyber-physical system-related metrics such as financial, political or economical indicators.
Note that we do not introduce any requirement on how to measure the identified metrics.

8.12.2 **Viewpoints influencing Quality Metrics**

All viewpoints may influence quality metrics. The decision regarding which quality metrics to set, and why, is strictly dependent on the requirements defined in all the other viewpoints.

[QUALITY 3] Quality metrics may be set on the basis of viewpoints constraints, Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

Examples:

- dependability requirements may set Quality of Service metrics.
- constraints may define metrics in the standards.
- dynamicity requirements may lead to the inclusion of metrics that quantity the SoS adaptiveness, flexibility, heterogeneity.
- governance requirements may suggest the inclusion of financial, economical and political indicators.

Many quality metrics requirements overlap with security and dependability requirements i.e., some security and dependability aspects defined in the appropriate viewpoints could be reduced to a subset of the quality metrics.

8.12.3 **Viewpoints influenced by Quality Metrics**

[QUALITY 4] Changing quality metrics may influence the Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

Basically, every viewpoint except constraints is potentially influenced by modifications to the quality metrics identified for the SoS. For example, relaxing the quality metrics of a service (e.g., security-related metrics) may allow it to interact with different components (e.g., new services composition are created). Thus new functionalities may "emerge" from a modifications in the quality metrics.

Moreover, the architecture may have to be modified to satisfy a newly defined quality metrics. In addition, the system security levels may be enhanced given more demanding quality metrics.

8.12.4 **Relations with other viewpoints**

The Quality Metrics viewpoint has relations with other viewpoints. The following requirement explains that these relations should be identified and reported when describing an SoS. The requirements in the other viewpoints further discuss their relations with Quality of Metrics.

[QUALITY 5] Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.
Figure 31 - Quality Metrics viewpoint: relations with other viewpoints.

The conceptual model for security and for the above mentioned relations is not part of this report.

8.13 VIEWPOINTS RELATIONS

We consider only requirements which express relations between viewpoints, and we report them in the following traceability matrix. The objective is to make explicit the connections and dependencies between viewpoints that may be difficult to extract from the list of requirements.
<table>
<thead>
<tr>
<th>VIEWPOINT</th>
<th>Relation (influence)</th>
<th>VIEWPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONSTRANTS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[CONSTR 5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics, in the Handling of Time, in the Dependability and in the Security.</td>
<td>--&gt; [ARCH 5]</td>
<td>The number and dimension of CSs, RUMIs and their interactions are limited only by the Constraints</td>
</tr>
<tr>
<td></td>
<td>--&gt; [ARCH 12]</td>
<td>Limitations on the possible specifications of RUMIs are due exclusively to the constraints</td>
</tr>
<tr>
<td></td>
<td>--&gt; [ARCH 18]</td>
<td>Limitations on the applicable algorithms and techniques are set exclusively by the constraints.</td>
</tr>
<tr>
<td></td>
<td>--&gt; [ARCH 19]</td>
<td>Limitations on the applicable technologies (standards, instruments, processes and methodologies, hardware) are set exclusively by the constraints.</td>
</tr>
<tr>
<td></td>
<td>--&gt; [ARCH 20]</td>
<td>Limitations on the applicable software objects are set exclusively by the constraints.</td>
</tr>
<tr>
<td></td>
<td>--&gt; [SEM 8]</td>
<td>The only limitation to the semantic of information exchanged are given by constraints.</td>
</tr>
<tr>
<td></td>
<td>--&gt; [SEM 9]</td>
<td>The semantic of communication shall be compliant to the Architecture &amp; RUMI and the constraints and it may change due to changes in the Architecture &amp; RUMI, evolution, and constraints</td>
</tr>
<tr>
<td>--&gt; [DYNAM 2]</td>
<td>The dynamicity of the SoS/CS and its entourage shall be limited in frequency, number, and dimension by its constraints and the Architecture &amp; RUMI.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [DYNAM 3]</td>
<td>Dynamicity may be caused by modifications in the Architecture &amp; RUMI, semantic of communication and constraints.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [EVOL 3]</td>
<td>Evolution of SoS/CS shall be governed exclusively by the constraints.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [EMERGE 4]</td>
<td>Modifications to the Architecture &amp; RUMI, the entourage, the constraints, the Semantic of Communication of an SoS/CS may influence emergence phenomena.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [GOV 5]</td>
<td>Governance may be influenced by all constraints, in particular costs, system life and assets.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [TIME 09]</td>
<td>(Part of) time requirements may be imposed by the Constraints.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [DEPEND 7]</td>
<td>(Part of) dependability requirements of an SoS may be imposed by the Constraints.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [DEPEND 8]</td>
<td>(Part of) assessment and monitoring solutions of an SoS may be limited and/or imposed by the Constraints.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [DEPEND 9]</td>
<td>Dependability requirements may change due to modifications in the Constraints.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [DEPEND 11]</td>
<td>A system shall be mandatorily assessed if it is required by the</td>
<td></td>
</tr>
</tbody>
</table>
(Part of) security and privacy requirements may be imposed by the constraints.

Quality metrics may be set on the basis of viewpoints constraints, Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

<table>
<thead>
<tr>
<th>Constraints.</th>
<th>[SECURITY 4]</th>
</tr>
</thead>
</table>

Relations in an SoS between the viewpoint Architecture & RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

<table>
<thead>
<tr>
<th>Relations in an SoS between the viewpoint Architecture &amp; RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</th>
<th>[ARCH 21]</th>
</tr>
</thead>
</table>

Relations in an SoS between the viewpoint Semantic of communication and the viewpoints Constraints, Architecture & RUMI, Dynamicity, Evolution, Dependability, Security, Quality Metrics shall be identified.

<table>
<thead>
<tr>
<th>Relations in an SoS between the viewpoint Semantic of communication and the viewpoints Constraints, Architecture &amp; RUMI, Dynamicity, Evolution, Dependability, Security, Quality Metrics shall be identified.</th>
<th>[SEM 11]</th>
</tr>
</thead>
</table>

Relations in an SoS between the viewpoint Dynamicity and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified.

<table>
<thead>
<tr>
<th>Relations in an SoS between the viewpoint Dynamicity and the viewpoints Constraints, Architecture &amp; RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified.</th>
<th>[DYNAM 7]</th>
</tr>
</thead>
</table>

Relations in an SoS between the viewpoint Evolution and the viewpoints Architecture & RUMI, Constraints (and especially Governance), Semantic of Communication, Emergence, Quality Metrics shall be identified.

<table>
<thead>
<tr>
<th>Relations in an SoS between the viewpoint Evolution and the viewpoints Architecture &amp; RUMI, Constraints (and especially Governance), Semantic of Communication, Emergence, Quality Metrics shall be identified.</th>
<th>[EVOL 8]</th>
</tr>
</thead>
</table>

Emergence phenomena in an SoS/CS may cause violations to the constraints, handling of time, dependability and security of
Relations in an SoS between the viewpoint Emergence and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

The governance viewpoint has relations with other viewpoints, following the same relations of the constraints viewpoint, see Section 8.8

Governance shall be a constraint for an SoS and its entourage, and it may be a constraint for a CS and its entourage

Relations in an SoS between the viewpoint Handling of Time and the viewpoints Constraints, Architecture & RUMI, Dynamicity, Emergence, Dependability, Security, Quality Metrics shall be explored

Relations in an SoS between the viewpoint Dependability and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Handling of Time, Security, Quality Metrics shall be identified.

Relations in an SoS between the viewpoint Security and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Dependability, Handling of Time, Quality Metrics shall be identified.
Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.

<table>
<thead>
<tr>
<th>ARCHITECTURE &amp; RUMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ARCH 6] Changes in the Architecture &amp; RUMI in a CS or an SoS may influence the semantic of communication, dynamicity, emergence, governance, quality metrics, handling of time, dependability, security of the CS or SoS.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>--&gt; [SEM 9]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>--&gt; [DYNAM 2]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>--&gt; [DYNAM 3]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>--&gt; [EMERGE 4]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>--&gt; [QUALITY 3]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>--&gt; Governance</td>
</tr>
<tr>
<td>Requirement</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>Handling of Time</td>
</tr>
<tr>
<td>Dependability</td>
</tr>
<tr>
<td>Security</td>
</tr>
</tbody>
</table>

**D1.1 - SoSs, commonalities and requirements**

**[ARCH 7]** Changes in Architecture & RUMI in a CS or an SoS may promote the evolution of the CS or SoS.

**[ARCH 8]** The RUMI shall describe the Semantic of Communication through the definition of itoms.

**[ARCH 13]** The notion of time shall be set for the SoS, defining whether a shared time base is needed as well as solutions for the resilient time synchronization.

**[ARCH 14]** Solutions for external or internal time synchronization may be identified.

**[EVOL 8]** Relations in an SoS between the viewpoint Evolution and the viewpoints Architecture & RUMI, Constraints (and especially Governance), Semantic of Communication, Emergence, Quality Metrics shall be identified.

**[SEM 9]** The semantic of communication shall be compliant to the Architecture & RUMI and the constraints and it may change due to changes in the Architecture & RUMI, evolution, and constraints.

**[TIME 10]** Design solutions to meet time requirements shall be defined in the Architecture & RUMI.
<table>
<thead>
<tr>
<th>ARCH 15</th>
<th>The shared time base will be a global time base from standards or a self-defined time base.</th>
<th>--&gt;</th>
<th>TIME 10</th>
<th>Design solutions to meet time requirements shall be defined in the Architecture &amp; RUMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCH 16</td>
<td>Solutions for handling of time, dependability and security shall be considered at SoS level and may be considered at lower levels than SoS level.</td>
<td>&lt;--</td>
<td>TIME 10</td>
<td>Design solutions to meet time requirements shall be defined in the Architecture &amp; RUMI</td>
</tr>
<tr>
<td>ARCH 16</td>
<td>Solutions for handling of time, dependability and security shall be considered at SoS level and may be considered at lower levels than SoS level.</td>
<td>&lt;--</td>
<td>DEPEND 12</td>
<td>Design solutions to meet dependability requirements shall be defined in the Architecture &amp; RUMI.</td>
</tr>
<tr>
<td>ARCH 16</td>
<td>Solutions for handling of time, dependability and security shall be considered at SoS level and may be considered at lower levels than SoS level.</td>
<td>&lt;--</td>
<td>DEPEND 13</td>
<td>Dependability requirements may influence the Architecture &amp; RUMI design.</td>
</tr>
<tr>
<td>ARCH 16</td>
<td>Solutions for handling of time, dependability and security shall be considered at SoS level and may be considered at lower levels than SoS level.</td>
<td>&lt;--</td>
<td>SECURITY 7</td>
<td>Design solutions to meet security requirements shall be defined in the Architecture &amp; RUMI.</td>
</tr>
<tr>
<td>ARCH 17</td>
<td>The RUMI interactions of CSs and/or entourages will generate emergent phenomena at the upper level.</td>
<td>--&gt;</td>
<td>EMERGE 4</td>
<td>Modifications to the Architecture &amp; RUMI, the entourage, the constraints, the Semantic of Communication of an SoS/CS may influence emergence phenomena.</td>
</tr>
<tr>
<td>Relationship</td>
<td>Viewpoint</td>
<td>Viewpoints</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>SEC 11</td>
<td>SECURITY</td>
<td>Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Emergence, Dependability, Handling of Time, Quality Metrics shall be identified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME 12</td>
<td>TIME</td>
<td>Constraints, Architecture &amp; RUMI, Dynamicity, Emergence, Dependability, Security, Quality Metrics shall be explored.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVOL 5</td>
<td>EVOLUTION</td>
<td>Evolution of an SoS/CS may give rise to modifications in the Architecture &amp; RUMI.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEM 11</td>
<td>SEMANTIC</td>
<td>Constraints, Architecture &amp; RUMI, Dynamicity, Evolution, Emergence, Dependability, Security, Quality Metrics shall be identified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYNAM 4</td>
<td>DYNAMIC</td>
<td>Changes in the dynamicity of an SoS/CS or its entourage may require to change the Architecture &amp; RUMI and the Semantic of Communication.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYNAM 7</td>
<td>DYNAMIC</td>
<td>Constraints, Architecture &amp; RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Changing quality metrics may influence the Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

### Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.

### Governance

The governance viewpoint has relations with other viewpoints, following the same relations of the constraints viewpoint, see Section 8.8

---

#### SEMANTIC OF COMMUNICATIONS

<table>
<thead>
<tr>
<th>[SEM 8]</th>
<th>The only limitation to the semantic of information exchanged are given by constraints.</th>
<th>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</th>
<th>[CONSTR 5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SEM 9]</td>
<td>The semantic of communication shall be compliant to the Architecture &amp; RUMI and the constraints and it may change due to changes in the Architecture &amp; RUMI, evolution, and constraints</td>
<td>The RUMI shall describe the Semantic of Communication through the definition of Itoms.</td>
<td>[ARCH 9]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics, in the Handling of Time, in the Dependability and in the Security.</td>
<td>[CONSTR 5]</td>
</tr>
<tr>
<td>SEM 10</td>
<td>Changes in the semantic of communication may influence dynamicity and emergence.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVOL 6</td>
<td>Evolution of an SoS/CS may give rise to modifications in the Semantic of Communication.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYNAM 3</td>
<td>Dynamicity may be caused by modifications in the Architecture &amp; RUMI, semantic of communication and constraints.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMERGE 4</td>
<td>Modifications to the Architecture &amp; RUMI, the entourage, the constraints, the Semantic of Communication of an SoS/CS may influence emergence phenomena.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SEM 11</th>
<th>Relations in an SoS between the viewpoint Semantic of communication and the viewpoints Constraints, Architecture &amp; RUMI, Dynamicity, Evolution, Emergence, Dependability, Security, Quality Metrics shall be identified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCH 6</td>
<td>Changes in the Architecture &amp; RUMI in a CS or an SoS may influence the semantic of communication, dynamicity, emergence, governance, quality metrics, handling of time, dependability, security of the CS or SoS.</td>
</tr>
<tr>
<td>ARCH 21</td>
<td>Relations in an SoS between the viewpoint Architecture &amp; RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>DYNAM 7</td>
<td>Relations in an SoS between the viewpoint Dynamicity and the viewpoints Constraints, Architecture &amp; RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------</td>
</tr>
<tr>
<td>EVOL 6</td>
<td>Evolution of an SoS/CS may give rise to modifications in the Semantic of Communication.</td>
</tr>
<tr>
<td>EVOL 8</td>
<td>Relations in an SoS between the viewpoint Evolution and the viewpoints Architecture &amp; RUMI, Constraints (and especially Governance), Semantic of Communication, Emergence, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>EMERGE 8</td>
<td>Relations in an SoS between the viewpoint Emergence and the viewpoints Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>DEPEND 16</td>
<td>Relations in an SoS between the viewpoint Dependability and the viewpoints Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Emergence, Handling of Time, Security, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>SECURITY 11</td>
<td>Relations in an SoS between the viewpoint Security and the viewpoints Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Emergence, Dependability, Handling of Time, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>QUALITY 3</td>
<td>Quality metrics may be set on the basis of viewpoints constraints, Architecture &amp; RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.</td>
</tr>
<tr>
<td>QUALITY 4</td>
<td>Changing quality metrics may influence the Architecture &amp; RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.</td>
</tr>
</tbody>
</table>
Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.

<table>
<thead>
<tr>
<th>DYNAMICITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>The dynamicity of the SoS/CS and its entourage shall be limited in frequency, number, and dimension by its constraints and the Architecture &amp; RUMI.</td>
</tr>
<tr>
<td>Changes in the Architecture &amp; RUMI in a CS or an SoS may influence the semantic of communication, dynamicity, emergence, governance, quality metrics, handling of time, dependability, security of the CS or SoS.</td>
</tr>
<tr>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</td>
</tr>
<tr>
<td>Dynamicity may be caused by modifications in the Architecture &amp; RUMI, semantic of communication and constraints.</td>
</tr>
<tr>
<td>Changes in the semantic of communication may influence dynamicity and emergence.</td>
</tr>
<tr>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>SEM 11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Handling of Time</th>
<th>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYNAM 5</td>
<td>In an SoS, the influence of dynamicity on dependability, security, and handling of time shall be analyzed and, if necessary and possible, may be mitigated.</td>
</tr>
<tr>
<td>Security</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
</tbody>
</table>

| DYNAM 6 | Changes in the dynamicity of a CS/SoS may give rise to emergence phenomena. | [EMERGE 5] | Emergence may be caused both by the normal and the dynamic behavior of an SoS/CS. |

<table>
<thead>
<tr>
<th>Relations</th>
<th>Viewpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relations in an SoS between the viewpoint Architecture &amp; RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
<td></td>
</tr>
<tr>
<td>&lt;--&gt; [ARCH 21]</td>
<td></td>
</tr>
<tr>
<td>Relations in an SoS between the viewpoint Semantic of communication and the viewpoints Constraints, Architecture &amp; RUMI, Dynamicity, Evolution, Emergence, Dependability, Security, Quality Metrics shall be identified.</td>
<td></td>
</tr>
<tr>
<td>&lt;--&gt; [SEM 11]</td>
<td></td>
</tr>
<tr>
<td>Relations in an SoS between the viewpoint Emergence and the viewpoints Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
<td></td>
</tr>
<tr>
<td>&lt;--&gt; [EMERGE 8]</td>
<td></td>
</tr>
<tr>
<td>Quality metrics may be set on the basis of viewpoints constraints, Architecture &amp; RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.</td>
<td></td>
</tr>
<tr>
<td>--&gt; [QUALITY 3]</td>
<td></td>
</tr>
<tr>
<td>Changing quality metrics may influence the Architecture &amp; RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.</td>
<td></td>
</tr>
<tr>
<td>&lt;-- [QUALITY 4]</td>
<td></td>
</tr>
</tbody>
</table>
Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.

Relations in an SoS between the viewpoint Handling of Time and the viewpoints Constraints, Architecture & RUMI, Dynamicity, Emergence, Dependability, Security, Quality Metrics shall be explored.

Relations in an SoS between the viewpoint Dependability and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Handling of Time, Security, Quality Metrics shall be identified.

Relations in an SoS between the viewpoint Security and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Dependability, Handling of Time, Quality Metrics shall be identified.

### EVOLUTION

<table>
<thead>
<tr>
<th>[EVOL 3]</th>
<th>Evolution of SoS/CS shall be governed exclusively by the constraints.</th>
<th>[CONSTR 5]</th>
<th>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[EVOL 4]</td>
<td>Evolution of the SoS/CS may be originated by modifications in the constraints.</td>
<td>[CONSTR 5]</td>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</td>
</tr>
<tr>
<td>[EVOL 5]</td>
<td>Evolution of an SoS/CS may give rise to modifications in the Architecture &amp; RUMI.</td>
<td>--&gt;</td>
<td>Architecture &amp; RUMI</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>[EVOL 6]</td>
<td>Evolution of an SoS/CS may give rise to modifications in the Semantic of Communication.</td>
<td>--&gt;</td>
<td>SEM 9</td>
</tr>
<tr>
<td>[EVOL 8]</td>
<td>Relations in an SoS between the viewpoint Evolution and the viewpoints Architecture &amp; RUMI, Constraints (and especially Governance), Semantic of Communication, Emergence, Quality Metrics shall be identified.</td>
<td>--&gt;</td>
<td>ARCH 21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>--&gt;</td>
<td>ARCH 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>--&gt;</td>
<td>CONSTR 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>--&gt;</td>
<td>CONSTR 9</td>
</tr>
<tr>
<td>Perspectives</td>
<td>Relations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Governance</td>
<td>see constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints</td>
<td>Relations in an SoS between the viewpoint Semantic of communication and the viewpoints Constraints, Architecture &amp; RUMI, Dynamicity, Evolution, Emergence, Dependability, Security, Quality Metrics shall be identified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EMERGE 8</td>
<td>Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture &amp; RUMI, Dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUALITY 3</td>
<td>Quality metrics may be set on the basis of viewpoints constraints, Architecture &amp; RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUALITY 4</td>
<td>Changing quality metrics may influence the Architecture &amp; RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUALITY 5</td>
<td>Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture &amp; RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Governance constraints include, but are not limited to, constraints to the players (stakeholders, users, satellite activities, funders), the technology, the evolution and progress, the financial aspects, the management of the SoS.
EMERGENCE

<table>
<thead>
<tr>
<th>EMERGE 7</th>
<th>Emergence phenomena in an SoS/CS may cause violations to the constraints, handling of time, dependability and security of the SoS/CS.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td></td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td></td>
<td>Detrimental effects of emergence phenomena may influence the dependability of an SoS/CS.</td>
</tr>
<tr>
<td></td>
<td>The security and privacy of a system may be affected by detrimental emergence phenomena.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMERGE 4</th>
<th>Modifications to the Architecture &amp; RUMI, the entourage, the constraints, the Semantic of Communication of an SoS/CS may influence emergence phenomena.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Changes in the Architecture &amp; RUMI in a CS or an SoS may influence the semantic of communication, dynamicity, emergence, governance, quality metrics, handling of time, dependability, security of the CS or SoS.</td>
</tr>
<tr>
<td></td>
<td>The RUMI interactions of CSs and/or entourages will generate emergent phenomena at the upper level.</td>
</tr>
<tr>
<td></td>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</td>
</tr>
<tr>
<td>[EMERGE 5]</td>
<td>Emergence may be caused both by the normal and the dynamic behavior of an SoS/CS.</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

Relations in an SoS between the viewpoint Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Viewpoint</th>
<th>Constraints</th>
<th>Architecture &amp; RUMI</th>
<th>Semantic of Communication</th>
<th>Dynamicity</th>
<th>Evolution</th>
<th>Governance</th>
<th>Emergence</th>
<th>Dependability</th>
<th>Security</th>
<th>Handling of Time</th>
<th>Quality Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVOL 8</td>
<td>Evolution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME 12</td>
<td>Handling of Time</td>
<td>Constraints, Architecture &amp; RUMI, Dynamicity, Emergence, Dependability, Security, Quality Metrics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEPEND 16</td>
<td>Dependability</td>
<td>Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Emergence, Handling of Time, Security, Quality Metrics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECURITY 11</td>
<td>Security</td>
<td>Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Emergence, Dependability, Handling of Time, Quality Metrics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUALITY 3</td>
<td>Quality Metrics</td>
<td>Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Dependability, Handling of Time, Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUALITY 4</td>
<td>Quality Metrics</td>
<td>Constraints, Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Emergence, Dependability, Handling of Time, Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUALITY 5</td>
<td>Quality Metrics</td>
<td>Constraints, Architecture &amp; RUMI, Dynamicity, Evolution, Governance, Emergence, Dependability, Handling of Time, Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### GOVERNANCE

<table>
<thead>
<tr>
<th>GOV 2</th>
<th>Governance shall be a constraint for an SoS and its entourage, and it may be a constraint for a CS and its entourage</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTR 2</td>
<td>Constraints in an SoS are related as a minimum to the classes: assets, standards, system life, costs and financial, governance.</td>
</tr>
<tr>
<td>CONSTR 16</td>
<td>The system life of the different systems may present dependencies, thus leading to connections between system lives.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GOV 5</th>
<th>Governance may be influenced by all constraints, in particular costs, system life and assets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSTR 5</td>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</td>
</tr>
<tr>
<td>CONSTR 19</td>
<td>The dimension, complexity, services, usage of an SoS shall be regulated by governance constraints.</td>
</tr>
<tr>
<td>CONSTR 20</td>
<td>Governance constraints include, but are not limited to, constraints to the players (stakeholders, users, satellite activities, funders), the technology, the evolution and progress, the financial aspects, the management of the SoS.</td>
</tr>
</tbody>
</table>
### TIME

<table>
<thead>
<tr>
<th>TIME 10</th>
<th><strong>Design solutions to meet time requirements shall be defined in the Architecture &amp; RUMI.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--&gt;</td>
<td>[ARCH 8] When requirements on dependability, security, handling of time, quality metrics are demanded, SoS shall be observed for monitoring purpose.</td>
</tr>
<tr>
<td>--&gt;</td>
<td>[ARCH 10] A precise specification of the RUMI shall be provided in the value domain and in the temporal domain.</td>
</tr>
<tr>
<td>--&gt;</td>
<td>[ARCH 13] The notion of time shall be set for the SoS, defining whether a shared time base is needed as well as solutions for the resilient time synchronization.</td>
</tr>
<tr>
<td>--&gt;</td>
<td>[ARCH 14] Solutions for external or internal time synchronization may be identified.</td>
</tr>
<tr>
<td>--&gt;</td>
<td>[ARCH 15] The shared time base will be a global time base from standards or a self-defined time base.</td>
</tr>
<tr>
<td>--&gt;</td>
<td>[ARCH 16] Solutions for handling of time, dependability and security shall be considered at SoS level and may be considered at lower levels than SoS level.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME 11</th>
<th><strong>Design solutions to meet time requirements may influence dependability and security design solutions.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;--&gt;</td>
<td>[SECURITY 10] Design solutions to meet security requirements may influence dependability and handling of time design solutions.</td>
</tr>
<tr>
<td>&lt;--&gt;</td>
<td>[DEPEND 15] Design solutions to meet dependability requirements may influence handling of time and security design solutions.</td>
</tr>
<tr>
<td>TIME 09</td>
<td>(Part of) time requirements may be imposed by the Constraints.</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
</tr>
</tbody>
</table>

| TIME 12 | Relations in an SoS between the viewpoint Handling of Time and the viewpoints Constraints, Architecture & RUMI, Dynamicity, Emergence, Dependability, Security, Quality Metrics shall be explored | <=-> | CONSTR 6 |
|         | Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified. |       |          |

|         | Relations in an SoS between the viewpoint Architecture & RUMI and the viewpoints Constraints, Dependability, Security, Handling of Time, Emergence, Semantic of Communication, Dynamicity shall be identified | <=-> | ARCH 20 |
|         | Relations in an SoS between the viewpoint Architecture & RUMI and the viewpoints Constraints, Dependability, Security, Handling of Time, Emergence, Semantic of Communication, Dynamicity shall be identified. |       |          |

|         | Relations in an SoS between the viewpoint Dynamicity and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified. | <=-> | DYNAM 7 |
|         | Relations in an SoS between the viewpoint Dynamicity and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified. |       |          |

|         | Relations in an SoS between the viewpoint Emergence and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified. | <=-> | EMERGE 8 |
|         | Relations in an SoS between the viewpoint Emergence and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified. |       |          |

|         | Design solutions to meet dependability requirements may influence handling of time and security design solutions. | <=- | DEPEND 15 |
|         | Design solutions to meet dependability requirements may influence handling of time and security design solutions. |       |          |
Relations in an SoS between the viewpoint Dependability and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Handling of Time, Security, Quality Metrics shall be identified.

Relations in an SoS between the viewpoint Security and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Dependability, Handling of Time, Quality Metrics shall be identified.

Quality metrics may be set on the basis of viewpoints constraints, Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

Changing quality metrics may influence the Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified.

Changing the constraints may lead as a consequence to changes in the Architecture & RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.
<p>| [DEPEND 8] | (Part of) assessment and monitoring solutions of an SoS may be limited and/or imposed by the Constraints. | Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified. |
| [DEPEND 9] | Dependability requirements may change due to modifications in the Constraints. | Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security. |
| [DEPEND 10] | Detrimental effects of emergence phenomena may influence the dependability of an SoS/CS. | Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security. |
| [DEPEND 11] | A system shall be mandatorily assessed if it is required by the Constraints. | Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified. |</p>
<table>
<thead>
<tr>
<th>Dependability</th>
<th>Architectural &amp; RUMI</th>
<th>Time</th>
<th>Security</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependability requirements shall be defined in the Architecture &amp; RUMI.</td>
<td>Solutions for handling of time, dependability and security shall be considered at SoS level and may be considered at lower levels than SoS level.</td>
<td>Relations in an SoS between the viewpoint Architecture &amp; RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
<td>The security and privacy of a system may be assessed.</td>
<td>Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>Dependability requirements may influence the Architecture &amp; RUMI design.</td>
<td>Relations in an SoS between the viewpoint Architecture &amp; RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
<td>Design solutions to meet security requirements may influence dependability and handling of time design solutions.</td>
<td>Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
</tr>
</tbody>
</table>
Quality Metrics shall be identified. Relations in an SoS between the viewpoint Architecture & RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified

<--> [ARCH 21]

Relations in an SoS between the viewpoint Emergence and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified

<--> [EMERGE 8]

Relations in an SoS between the viewpoint Handling of Time and the viewpoints Constraints, Architecture & RUMI, Dynamicity, Emergence, Dependability, Security, Quality Metrics shall be explored

<--> [TIME 12]

Design solutions to meet security requirements may influence dependability and handling of time design solutions.

<-- [SECURITY 10]

Quality metrics may be set on the basis of viewpoints constraints, Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

--> [QUALITY 3]

Changing quality metrics may influence the Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

<-- [QUALITY 4]

Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of
<table>
<thead>
<tr>
<th>SECURITY</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[SECURITY 4]</strong></td>
<td>(Part of) security and privacy requirements may be imposed by the constraints.</td>
</tr>
<tr>
<td></td>
<td>Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</td>
</tr>
<tr>
<td></td>
<td>Changing the constraints may lead as a consequence to changes in the Architecture &amp; RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security.</td>
</tr>
<tr>
<td><strong>[SECURITY 5]</strong></td>
<td>The security and privacy of a system may be affected by detrimental emergence phenomena.</td>
</tr>
<tr>
<td></td>
<td>Emergence phenomena in an SoS/CS may cause violations to the constraints, handling of time, dependability and security of the SoS/CS.</td>
</tr>
<tr>
<td></td>
<td>Relations in an SoS between the viewpoint Constraints and the viewpoints Architecture &amp; RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>SECURITY 6</td>
<td>The security and privacy of a system may be assessed.</td>
</tr>
<tr>
<td>SECURITY 7</td>
<td>Design solutions to meet security requirements shall be defined in the Architecture &amp; RUMI.</td>
</tr>
<tr>
<td>SECURITY 8</td>
<td>Changes in the security and privacy requirements may imply modifications to the Architecture &amp; RUMI and a re-assessment of the system.</td>
</tr>
<tr>
<td>SECURITY 8</td>
<td>Changes in the security and privacy requirements may imply modifications to the Architecture &amp; RUMI and a re-assessment of the system.</td>
</tr>
<tr>
<td>SECURITY 9</td>
<td>Privacy requirements set on CSs and the entourage may influence the Semantic of Communication and the Architecture &amp; RUMI.</td>
</tr>
<tr>
<td>SECURITY 10</td>
<td>Design solutions to meet security requirements may influence dependability and handling of time design solutions.</td>
</tr>
<tr>
<td>SECURITY 10</td>
<td>Design solutions to meet security requirements may influence dependability and handling of time design solutions.</td>
</tr>
</tbody>
</table>
Relations in an SoS between the viewpoint Security and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Dependability, Handling of Time, Quality Metrics shall be identified.

<-- [SECURITY 11] [CONSTR 6]

Relations in an SoS between the viewpoints Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

<--> [ARCH 21]

Relations in an SoS between the viewpoint Architecture & RUMI and the viewpoints Constraints, Semantic of Communication, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

<--> [SEM 11]

Relations in an SoS between the viewpoint Semantic of communication and the viewpoints Constraints, Architecture & RUMI, Dynamicity, Evolution, Emergence, Dependability, Security, Quality Metrics shall be identified.

<--> [EMERGE 8]

Relations in an SoS between the viewpoint Emergence and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Evolution, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.

<--> [DEPEND 16]

Relations in an SoS between the viewpoint Dependability and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Dynamicity, Emergence, Handling of Time, Security, Quality Metrics shall be identified.

<--> [DYNAM 7]

Relations in an SoS between the viewpoint Dynamicity and the viewpoints Constraints, Architecture & RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified.
| [QUALITY 3] | Quality metrics may be set on the basis of viewpoints constraints, Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security. |
| [QUALITY 4] | Changing quality metrics may influence the Architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security. |
| [QUALITY 5] | Relations in an SoS between the viewpoint Quality Metrics and the viewpoints constraints, Architecture & RUMI, dynamicity, evolution, governance, emergence, dependability, handling of time, Semantic of Communication, security shall be identified. |

**QUALITY METRICS**

| [CONSTR 5] | Changing the constraints may lead as a consequence to changes in the Architecture & RUMI and in the Semantic of Communication, to Dynamicity, Evolution, Emergence, Governance, Quality Metrics to changes in the Handling of Time, in the Dependability and in the Security. |
| [ARCH 6] | Changes in the Architecture & RUMI in a CS or an SoS may influence the semantic of communication, dynamicity, emergence, governance, quality metrics, handling of time, dependability, security of the CS or SoS. |

Semantic of communication

A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.
<table>
<thead>
<tr>
<th>Column</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamicity</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Evolution</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Governance</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Emergence</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Dependability</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Handling of Time</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Security</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
</tbody>
</table>
Changing quality metrics may influence the architecture & RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality 4</td>
<td>Changing quality metrics may influence the architecture &amp; RUMI, semantic of communication, dynamicity, evolution, governance, emergence, dependability, handling of time, and security.</td>
</tr>
<tr>
<td>ARCH 8</td>
<td>When requirements on dependability, security, handling of time, quality metrics are demanded, SoS shall be observed for monitoring purpose.</td>
</tr>
<tr>
<td>Semantic of</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Communication</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Dynamicity</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Evolution</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Governance</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
<tr>
<td>Emergence</td>
<td>A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.</td>
</tr>
</tbody>
</table>
### Dependability

A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.

### Handling of Time

A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.

### Security

A specific requirement to match the requirement on the left column is not included, as it is considered redundant and not adding significant additional information to the requirements framework.

---

<table>
<thead>
<tr>
<th><strong>Relations in an SoS between the viewpoint</strong></th>
<th><strong>Quality Metrics and the viewpoints</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints, Architecture &amp; RUMI, Dynamicity, Evolution, Emergence, Governance, Handling of Time, Dependability, Security, Quality Metrics shall be identified.</td>
<td><strong>Qualities</strong> 5</td>
</tr>
</tbody>
</table>

| Relations in an SoS between the viewpoint Architecture & RUMI, Semantic of Communication, Dependability, Security, Quality Metrics shall be identified. | **Const raints** 6 |

| Relations in an SoS between the viewpoint Architecture & RUMI, Dynamicity, Semantic of Communication, Emergence, Dependability, Security, Handling of Time, Quality Metrics shall be identified. | **Architecture** 21 |

<p>| Relations in an SoS between the viewpoint Dynamicity and the viewpoints Constraints, Architecture &amp; RUMI, Semantic of Communication, Emergence, Dependability, Security, Handling of Time and Quality Metrics shall be identified. | <strong>Dynam icity</strong> 7 |</p>
<table>
<thead>
<tr>
<th>Relation</th>
<th>Viewpoint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;-&gt;</td>
<td>[Evol 8]</td>
<td>Relations in an SoS between the viewpoint Evolution and the viewpoints Architecture &amp; RUMI, Constraints (and especially Governance), Semantic of Communication, Emergence, Quality Metrics shall be identified.</td>
</tr>
<tr>
<td>&lt;-&gt;</td>
<td>Governance</td>
<td>see constraints</td>
</tr>
</tbody>
</table>
9 CONCLUSIONS, CHALLENGES AND OPEN ISSUES

This report is the final document of Work Package 1 (Analysis of Existing SoSs). The report first presents an analysis of existing and past System-of-Systems (SoS) related projects and main SoS properties from the state of the art. Then, the report investigates SoSs in five representative domains from a number of different perspectives. Subsequently, it illustrates relations between domains and finally the report concludes with the definition of requirements relevant for generic SoSs. Our SoS requirements can be seen as SoS elements, peculiarities, or characteristics that should be identified when describing an SoS. The final outcome of this report aims to be a requirements’ framework that can be applied to describe an SoS and that shall be used as a reference for the definition of the conceptual model in Work Package 2.

The analysis carried out throughout the document is organized in eleven viewpoints, which are considered the core aspects that will be addressed for the whole AMADEOS project. The viewpoints considered and the overall outcome for each viewpoint is briefly discussed below.

9.1 SoS CONSTRAINTS

From D2.1, a constraint is a restriction in the problem space. A common aspect identified in all the analyzed domains is the (expected) existence of constraints, and in particular of some classes of constraints that are common in the analyzed domains. These classes are assets, standards, lifecycles, costs and financial, and governance.

Constraints are key when defining SoS requirements in general. Constraints are related to all the other viewpoints, and typically limit those viewpoints. In particular, the requirements regarding SoS constraints define an approach to identify the main constraints that should be taken into account when building or describing an SoS.

9.1.1 Architecture and RUMI

The architecture and RUMI is a viewpoint that considers how CSs of an SoS are hierarchically organized (see Section 7.2), how they are interconnected and what information they exchange. CSs exclusively interact with other CSs through their Relied Upon Message Interfaces (RUMIs). Hence the precise specification of RUMIs is central to System-of-Systems Engineering (SoSE) and pose probably the most important design artefact that needs to meet SoS requirements.

Requirements on Architecture and RUMI were identified following a system engineer approach, as the objective is to define meta-requirements that can be used to derive the architectural requirements of an SoS. The requirements were organized in five views, to cover the main aspect of software engineering thus guaranteeing completeness of the analysis.

9.1.2 Semantic of Communication

The viewpoint Semantic of Communication describes how and why the information is exchanged among the CSs. This viewpoint refers to the meaning of the exchanged information and it has evident connections to the Architecture and RUMI viewpoint, although the latter one also includes the syntactical aspects of messages and protocols. We identified relevant aspects of semantic of communication in all the considered domains.

Requirements regarding this viewpoint focus on the information contained in the Itom, and on relations with other viewpoints.

9.1.3 Dynamicity

Dynamicity is defined in the concepts glossary of D2.1 as the property of an entity that is constantly changing in terms of offered services, built-in structure, and interactions with other entities. Dynamicity includes the mechanisms implemented to overcome unexpected events and to deal with fast reconfiguration of an SoS.
In an SoS frequent changes may happen and these changes may have many different effects on
the SoS. For example, changes can lead to new emergent phenomena. Thus the viewpoint
dynamicity required a careful analysis, to understand the viewpoints that are related to dynamicity
(i.e., generates dynamicity or are affected by dynamic phenomena). The outcome of our reasoning
is reported in the requirements presented in Section 8.

9.1.4 Evolution
Evolution refers to the long-term and continuous changes in the SoS as new functions are
included, removed or modified according to the evolution of needs.

A key result from the state of the art and the five domains analyses is that evolution is an inherent
property of SoSs. Thus, evolution should be taken into account when building an SoS as well as
when describing it. The limitations to evolution and its effects on the other viewpoints must be well
understood to achieve an attentive SoS design. These are amongst the main considerations
reported in the requirements on evolution in Section 8.

9.1.5 Emergence
Emergence is a distinct feature of SoSs. In D2.1, emergence is defined as follows: “A phenomenon
of a whole at the macro-level is emergent if and only if it is new with respect to the non-relational
phenomena of any of its proper parts at the micro level.”

Although as per today no generally accepted approaches to model or predict emergence are
available, understanding emergence would be a key advantage for SoS engineers. Our
requirements aim to provide the foundations for the definition of an SoS conceptual model which
explicitly includes emergence phenomena and their relations in an SoS.

9.1.6 Governance
From D2.1, governance is the theoretical concept referring to the actions and processes by which
stable practices and organizations arise and persist. These actions and processes may operate in
formal and informal organizations of any size; and they may function for any purpose.

Governance describes the way in which an SoS is managed, including defining roles, laws,
standards, etc. This is a critical aspect because different owners/regulations related to different
CSs may limit the interoperability across them. We concluded from the analysis of our domains,
that governance can be also considered as a part of SoS’s constraints. Thus in our requirements in
Section 8, we maintain governance as a viewpoint by itself, but we maintain governance explicitly
in the constraints viewpoint as well.

9.1.7 Handling of Time
Unlike previous SoS research we also focused in our analysis of the domains on how the
respective CSs handle the progression of physical time. Time plays a key-role in all considered
domains.

Consequently, our SoS requirements focus in particular on timeliness requirements and clock
synchronization requirements, including the problematic of clock synchronization quality.

9.1.8 Dependability and Security
Dependability and security (including trust and privacy) are amongst the main concerns in the
investigated domains. In our domain analysis it became evident that the different attributes of
dependability and security may assume a greater or smaller importance depending on the domain
under consideration.

Our requirements described in Section 8 include:

i) meta-requirements to specify "dependability and security requirements", that are usually set prior
to system design;
ii) requirements that describe the dependable and secure design and assessment of a system.

9.1.9 **Quality Metrics**

Quality Metrics refer to the set of qualitative and quantitative metrics to decide on the efficiency and effectiveness of an SoS. They are related to different aspects of an SoS and they are the basis for imposing constraints. Providing hard constraints upon quality metrics may be not always possible, thus it is relevant to support sub-optimal but still graceful (non-detrimental) degradation of qualities. Quality metrics can be considered as "soft" constraints, in the sense that it is ideal to satisfy these constraints but whether they are achieved or not can be demanded by requirements in the different viewpoints.

**9.2 Viewpoints Relations**

The viewpoints are interconnected and partially overlap. This whole document offers means to identify such relations and ultimately express them in the form of requirements.

Despite all our efforts, we cannot claim that we have identified a mapping between viewpoints which both can be considered complete and does not include unnecessary relations. As a matter of fact, such mapping is extremely difficult to define, and some relations between viewpoints may be (and in our experience, have been) valued differently by different people. Hence, when building an SoS, it is necessary to add or remove relations between viewpoints in case it is appropriately justified by the specifics of the SoS.

**9.3 Next Steps and Challenges**

*Omne ignotum pro magnifico est* - Tacitus: *Agricola*, Book 1, 30

*Everything unknown (is taken) as grand.*

As a consequence of the research carried out in this report, it can be concluded that the several facets of SoSs require the reconsideration of the traditional approaches for system engineering. An SoS-oriented approach to system engineering requires a change in the traditional perspective for system design, assessment, implementation, deployment and maintenance. For example, viewpoints traditionally not considered when building systems here become central (e.g., emergence and governance). Moreover, other viewpoints are intrinsic to the SoSE approach (e.g., emergence, evolution and dynamicity).

This calls for an effort in building a new approach for defining SoS requirements and ultimately for SoS design and assessment. The ultimate outcome of this document, i.e., the definition of SoS meta-requirements to support the successive identification of an SoS conceptual model, is finally tackled in Section 8, which offers a requirements framework for SoS engineers. It will be the objective of WP2 to build a conceptual SoS model.

Still, despite the work done here and in previous projects, the required changes in the perspective of SoSE raise several challenges that we need to take into account in the further course of AMADEOS:

- Being able to completely identify the relations between viewpoints and their interplay for the whole life of an SoS. These are fundamental for an SoS model and for the success of the AMADEOS architectural framework. Note that, as previously said, some relations are SoS-dependent, thus relations may vary from domain to domain and from SoS to SoS.

- Emergence is a critical viewpoint, which was rarely if not at all present in traditional system engineering. While vast literature on emergence in SoSs exists, and despite the fact that emergence is an intrinsic characteristic of SoSs, there are no SoS design approaches which are emergence-oriented or emergence-aware. Similarly, models that forecast emergence are missing nowadays. Techniques for emergence-aware design, design-for-
emergence, or emergence-driven design may become relevant assets in SoS design whenever appropriately identified.

- Time, Dependability and Security. Time, dependability and security requirements may become more complex when the focus is the whole SoS. This is due to the overlaps of other viewpoints that are traditionally ignored in traditional systems. This consideration may affect both the design and the assessment of SoS, thus calling for new solutions.

- Dynamicity and Evolution. An SoS is an evolutionary system that may (and likely will) exhibit dynamic behavior. Also, other viewpoints are involved and may influence the two viewpoints above. This means that adaptiveness, flexibility, and evolvability of an SoS shall be considered from both long-term and short-term perspectives, and consequently engineering approaches may require revisions.

- Constraints and Governance. SoS engineers must move their perspective, from looking at customer requirements only to looking at a plethora of constraints. These requirements should be considered as they may impact the success, the life and the evolution of an SoS. Thus, the collection of requirements requires a deep analysis and understanding of a potentially vast amount of scenarios, where boundaries may be difficult to set and where diverse actions and processes may be defined to govern the system. To be able to build an SoS, engineers need to understand where to set the boundaries that should be considered for their work, and how broad and detailed their view on the SoS should be.
10 REFERENCES


[37] EN 50126-1:2012 - Railway applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)

[38] EN 50128:2011 - Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems.

[40] EN 50159:2010 - Railway applications - Communication, signalling and processing systems - Safety-related communication in transmission systems.


[67] RFI Rete Ferroviaria Italiana, SCMT SRS Sistema (SMT System Requirement Specification - System) - RFI TC.PATC SR IS 13 D21 A01 (14/12/2001)

[68] RFI Rete Ferroviaria Italiana, SCMT SRS Sistema (SCMT System Requirement Specification - System) - RFI TC.PATC SR IS 13 D21 A01 (14/12/2001)

[69] RFI Rete Ferroviaria Italiana, SCMT SSB (SCMT onboard subsystem)- RFI TC.PATC SR CM 03 M 67 F

[70] Il Sistema Controllo Marcia Treno (SCMT) http://www.segnalifs.it/it/bl/N_scmt.htm


[86] Berkman, Ostrovsky: The Unbearable lightness of PIN Cracking

[87] M Bond: Attacks on Cryptoprocessor Transaction Sets


[186] SLA@SOI. Empowering the service economy with SLA-aware infrastructures. [Online]. http://sla-at-soi.eu/


