AMADEOS Time-Aware System-of-Sytems Architecture

In many applications the duration between events that occur in the environment of the different Constituent Systems (CSs) of a System of Systems (SoS) must be determined. If a global timestamp is assigned to every significant event, then the duration between any two significant events occurring at any place within the whole SoS can be calculated easily. For example, we can consider the temporal validity of real-time data. An observation of a dynamic entity, e.g., the state of traffic light, can only be used for control purposes within a validity interval that depends on the dynamics of the entity. If the observation of the environment is performed by a CS that is different from the CS that uses the observation, then, based on the timestamp of the observation, the user can determine if it safe to use a given observation at a particular moment.

A **dependable global** (**physical**) **time** provides the backbone of the temporal infrastructure of an SOS. Some of the benefits that can be achieved are summarized below:

- Interpret timestamps across the diverse CSs.
- Control and synchronize the *durations of the physical time frames.*
- Specify the *temporal properties* of interfaces.
- Synchronize inputs and output actions.
- Synchronize *stigmergic* and *message-based* information.
- Allocate resources *conflict-free* (e.g, in time-triggered communication, scheduling).
- *Detect errors* promptly.
- Strengthen *security protocols.*

Following that ascertainment, AMADEOS brought attention on the relevance of building a reliable time base to ease the development and execution of distributed, time-critical Systems of Systems. Specifically, we propose the Resilient Master Clock (RMC), a low-cost hardware-software solution. The RMC acts as a dependable, accurate global time base that includes local clock correction techniques, self-estimation of time awareness and fault tolerant synchronization solutions. The ultimate goal is to guarantee a consistent global time view across the SoS infrastructure to avoid synchronization problems.

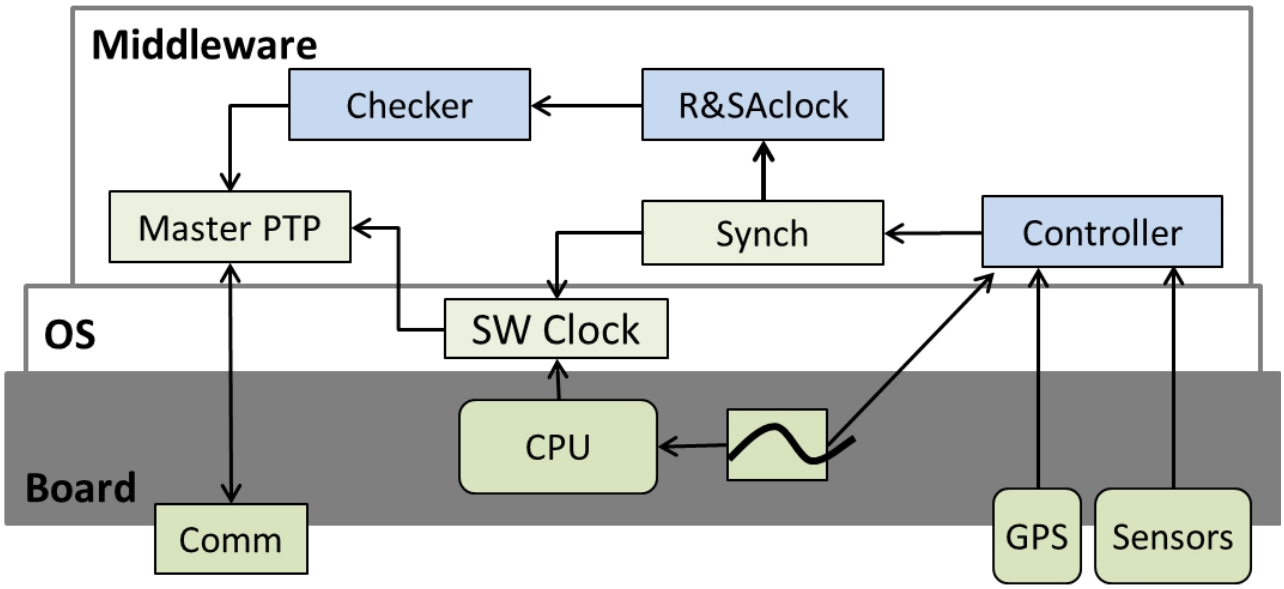

Figure 1 : The Resilient Master Clock architecture

The architecture of the RMC is shown in Figure 1 and it is divided in three layers: *board* (or Hardware layer), *Operating System* (or OS layer) and *Middleware*. Each layer consists of the different constituent blocks which are herewith described. In the picture, the software we developed (Checker, R&SAClock, Controller) is in light blue, while off-the-shelf software and hardware is in green.

Board layer. The components of this layer are only hardware ones :

- A *GPS module* for receiving time messages by the GPS satellite constellation. The messages are then provided to any enquiring hardware or software along with a one-pulse-per-second (1PPS) signal.
- *Sensors* for acquiring information from the environment. Examples are temperature and pressure sensors.
- *Comm*. This block refers to the communication interface. For example, an Ethernet Network Interface Card (NIC) can connect the RMC to a network in which PTP slaves wait for time synchronization packets from a master clock.
- *CPU and the physical oscillator*. These are standard components of any hardware board, necessary to execute and read the physical clock.

OS Layer. This layer includes only one component, which is a common component in modern operating system :

- A local software clock (*SW Clock*) which provides timestamps to the board.

Middleware layer. This layer includes off-the-shelf software components (*Synch*, *master PTP*), and software components we developed on our own :

- A *synchronization module* (*Synch*) based on NTP which uses GPS time signals to discipline the local clock.
- A *master PTP* module for broadcasting a time synchronization packet according to the protocol IEEE 1588 PTP to the nodes of the subnetwork to which the board is connected through *Comm*.
- The *R&SAclock* uses the offset and drift obtained from the synchronization module to estimate the uncertainty of the time provided by the local clock over time.
- The *controller* module disciplines the clock when the GPS signal is unavailable. The disciplining mechanism provided by the controller is based on
    - the values measured by the dedicated sensors (e.g., temperature) and
    - a-priori knowledge of the frequency deviation caused by environmental changes on the onboard crystal oscillator (e.g., temperature variations).
- A *checker* module checks the uncertainty associated to the time of the local clock provided by the R&SAclock; consequently, it decides if the RMC can be considered a reliable time source and allows or blocks the PTP synchronization.