Guaranteeing that Cyber-Physical Systems (CPS) do not fail is a non-trivial problem, especially with the adoption of complex software solutions; Formal Methods (rigorous mathematical and logical techniques) can aid in the task of obtaining proofs about the correctness of CPS; Runtime Verification (RV) uses monitors and formal specifications to, during runtime, validate a set of properties.

The deployment of RV solutions causes an inevitable overhead in the system; If not correctly deployed, RV solutions can compromise the security and safety requirements of a system; Lack of formalism in the current state-of-the-art when it comes to the integration of monitoring solutions in CPS; Performing formal verifications is a tedious and time-consuming task.

Apply formal methods to guarantee that deployed software monitoring solutions comply with the target system’s safety requirements; Abstract formalism from the system’s developer by creating a Domain-Specific Language (DSL) and a set of tools to support it; Centralize and automate formal verification procedures for the correct deployment of monitors in a given target system.

Our proposed DSL, and its associated toolset, must allow the developer to: Express functional and non-functional properties to be verified during runtime; Verify, statically, the schedulability of a system We plan to leverage the literature on the topic of mode-change schedulability analysis to enforce better use of system resources; We plan on supporting a broad set of scheduling algorithms and hardware architectures; Automate the code generation of the monitors and monitoring architectures following a Correct-by-Construction approach (i.e., complying with a set of formally defined requirements) Leverage state-of-the-art formal tools like model checkers, SMT solvers, and proof-assistants.

Below, we illustrate our initial efforts to design the DSL syntax using an automotive electric/electronic architecture as a use case example:

Enhanced security and safety of monitored systems; Reduced development and testing time; Potential overall project cost reduction; Centralized tool for the deployment of monitoring solutions.