SAFURE

Modeling and analysis of Mixed-Critical systems

Marco Di Natale
SSSA

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644080."

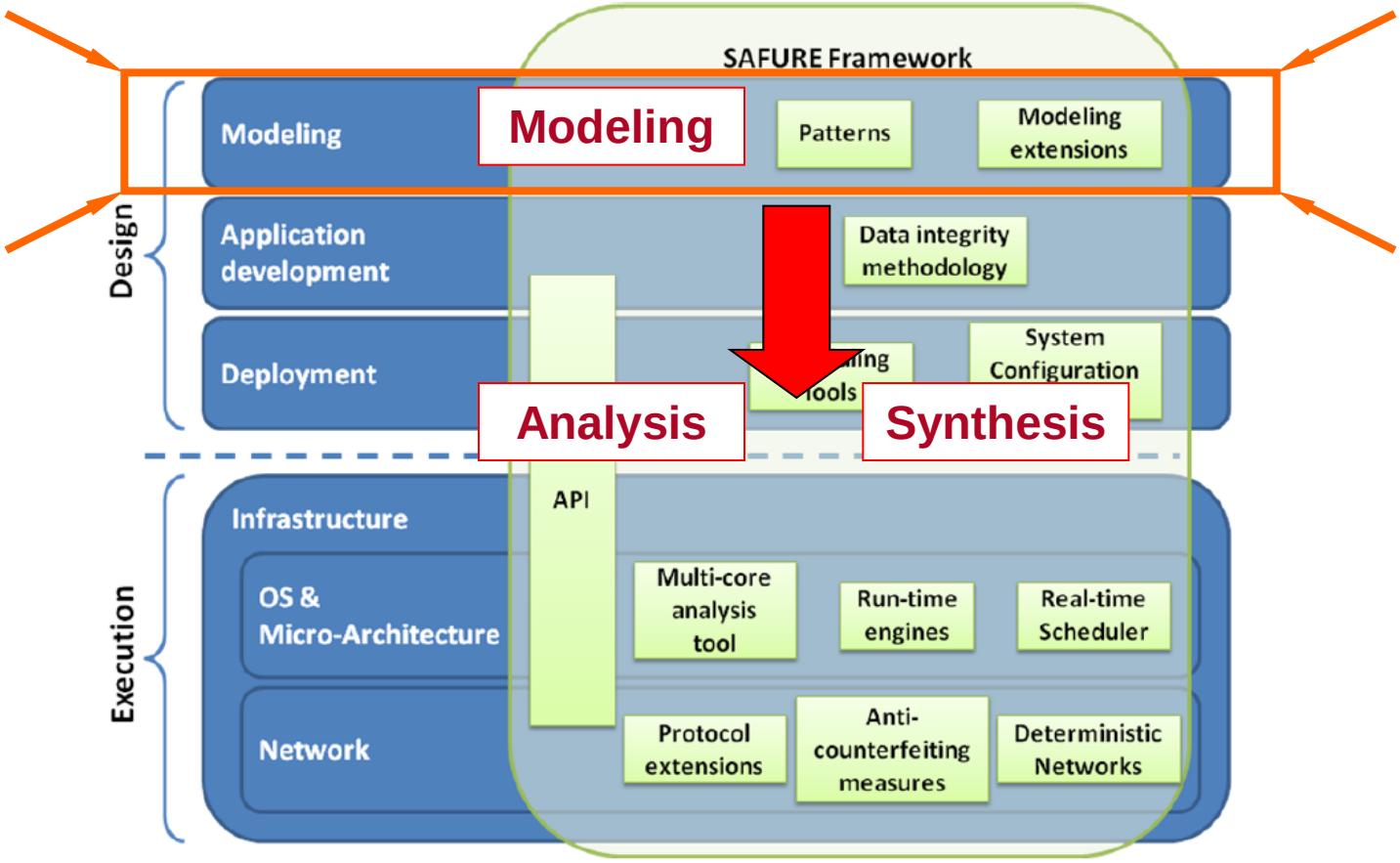
SAFURE

SAFety and secURity by dEsign for interconnected mixed-critical cyber-physical systems

Outline

- **Motivation and outline of methodology**
- **Extensions to modeling languages for safe and secure functionality**
  - From abstract concepts to concrete implementations targeting AUTOSAR and UML/SysML
  - Extensions (profiles) and architecture patterns
- **Using models for synthesis of implementations**
  - Security: synthesis of adapters
  - Time: synthesis of OS mechanisms for isolation in AUTOSAR
    - Connection to automotive use case
- **Using models for (timing) analysis**
  - m-k models

SAFURE architecture



Modeling extensions

Modelling safe and secure functionality

- Need to represent safety and security requirements at the functional level for the system and its components
  - Currently emphasis on mechanisms
  - Examples: taint (dependency) analysis, security needs on connections/ports, timing…
- Need to represent timing requirements for mixed-critical systems
  - Extensions to common modeling paradigms such as in MARTE or AUTOSAR
- Show connection to analysis methods and opportunities for synthesis
- Driven by needs and requirements (standards) of the automotive and telecom industrial cases.
- Support for the identification of data dependencies (taint analysis)
- Support for the definition of mixed-critical real-time systems as in use by the RT research community (and challenged in a recent RTNS paper)
- The same model can be used for the definition of mode-dependent behaviors or AVR tasks
- Support the integration of security faults in “traditional” fault tree analysis
- Definition of patterns for protection kernel and definitions to identify the mapping of functional units to partitions

Modeling extensions: Methodology

- **Step 0**: analysis of state of the art, projects, standards
  - Research papers, Safe and Evita projects (and more), MARTE (UML/SysML), AUTOSAR, Capella
- **Step 1**: Abstract modeling
- *Result*: metamodels expressed using Eclipse/Ecore
- **Step 2**: Selection of concrete target modelling languages (meta-models) and implementation using a target tool;
- *Result*: UML Profiles defined in Rhapsody and applicable to AUTOSAR elements in Rhapsody

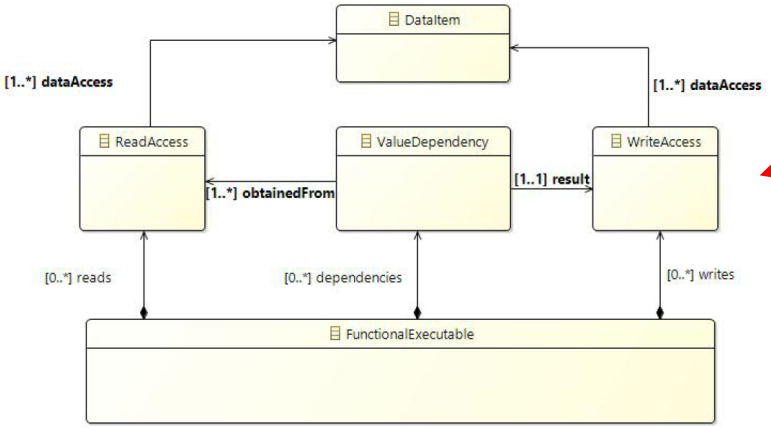
Modeling extensions: Methodology

- **Step 3a**: Definition of analysis methods and tools at the system and components level;
- **Step 3b**: Opportunities for automatic synthesis and architecture optimizations.

Modeling extensions

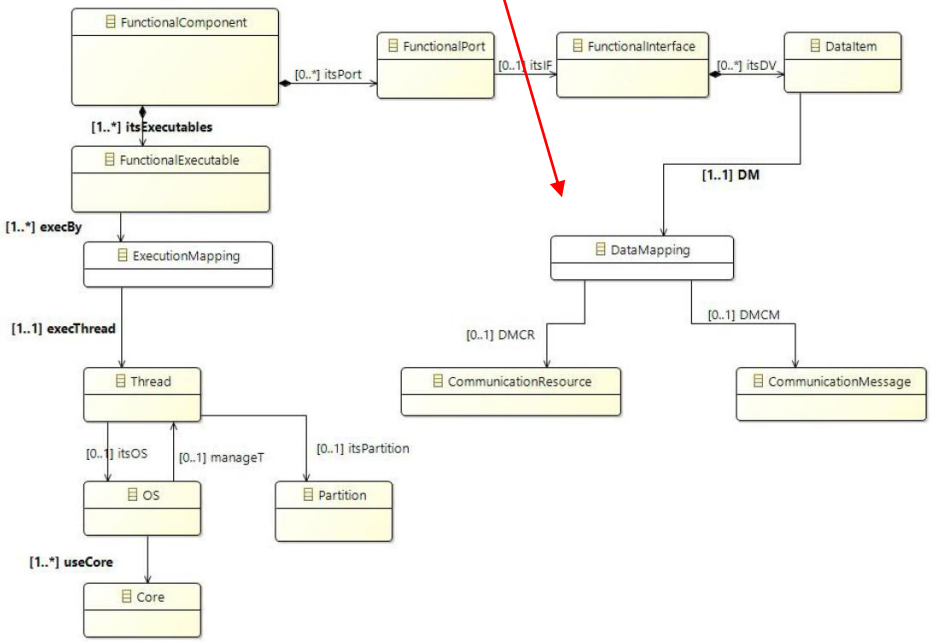
- **Architecture models and patterns for safe and secure systems**
- Define execution architectures (computation and communication HW, operating system and communication layers) using modelling languages that allow the specification and analysis of time, safety and security constraints.
- Define components or architecture patterns that are capable of providing desirable safety and security levels (sharing resources with guaranteed separation of concerns at the functional and timing level).

Abstract Models (examples - General)

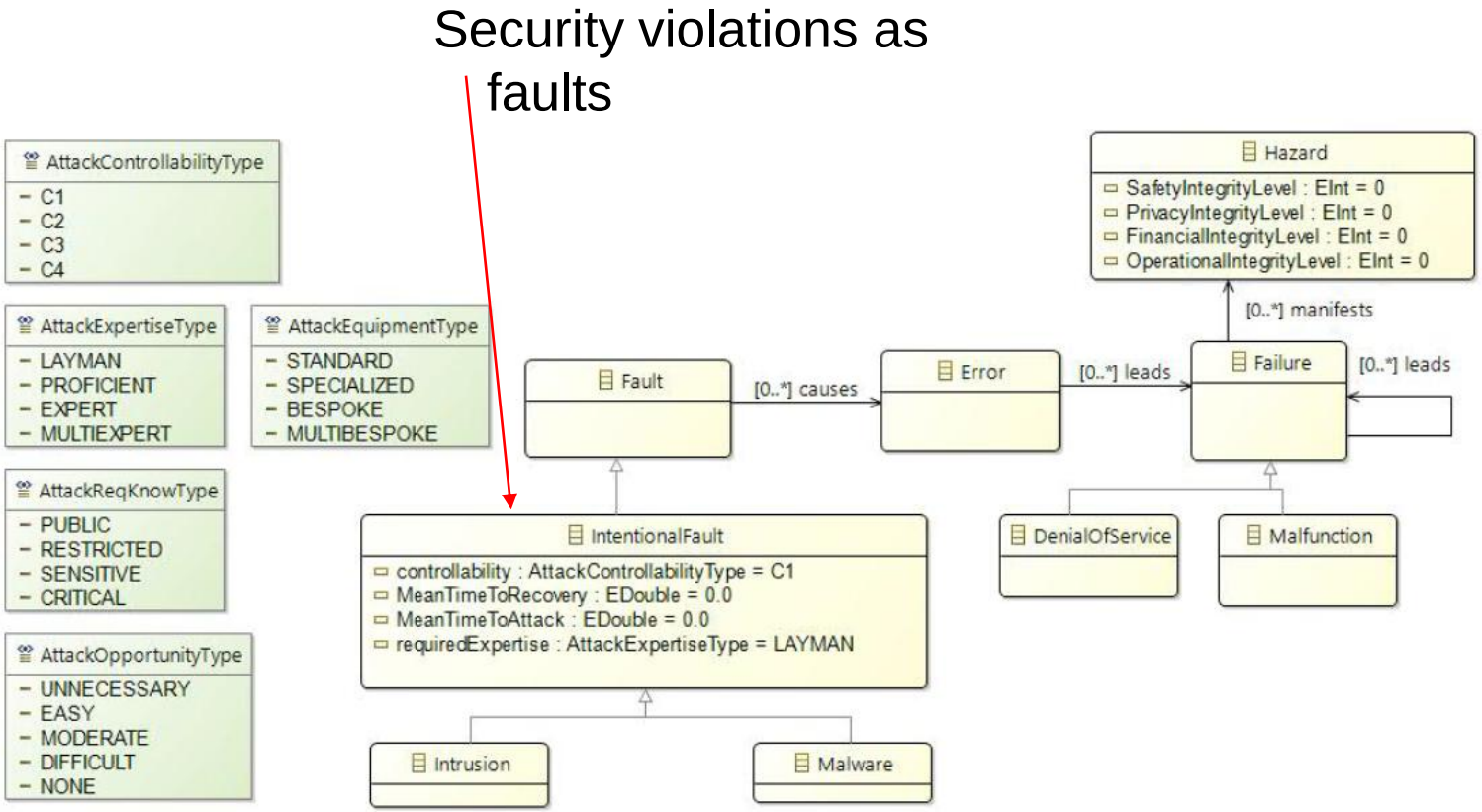


Value dependencies

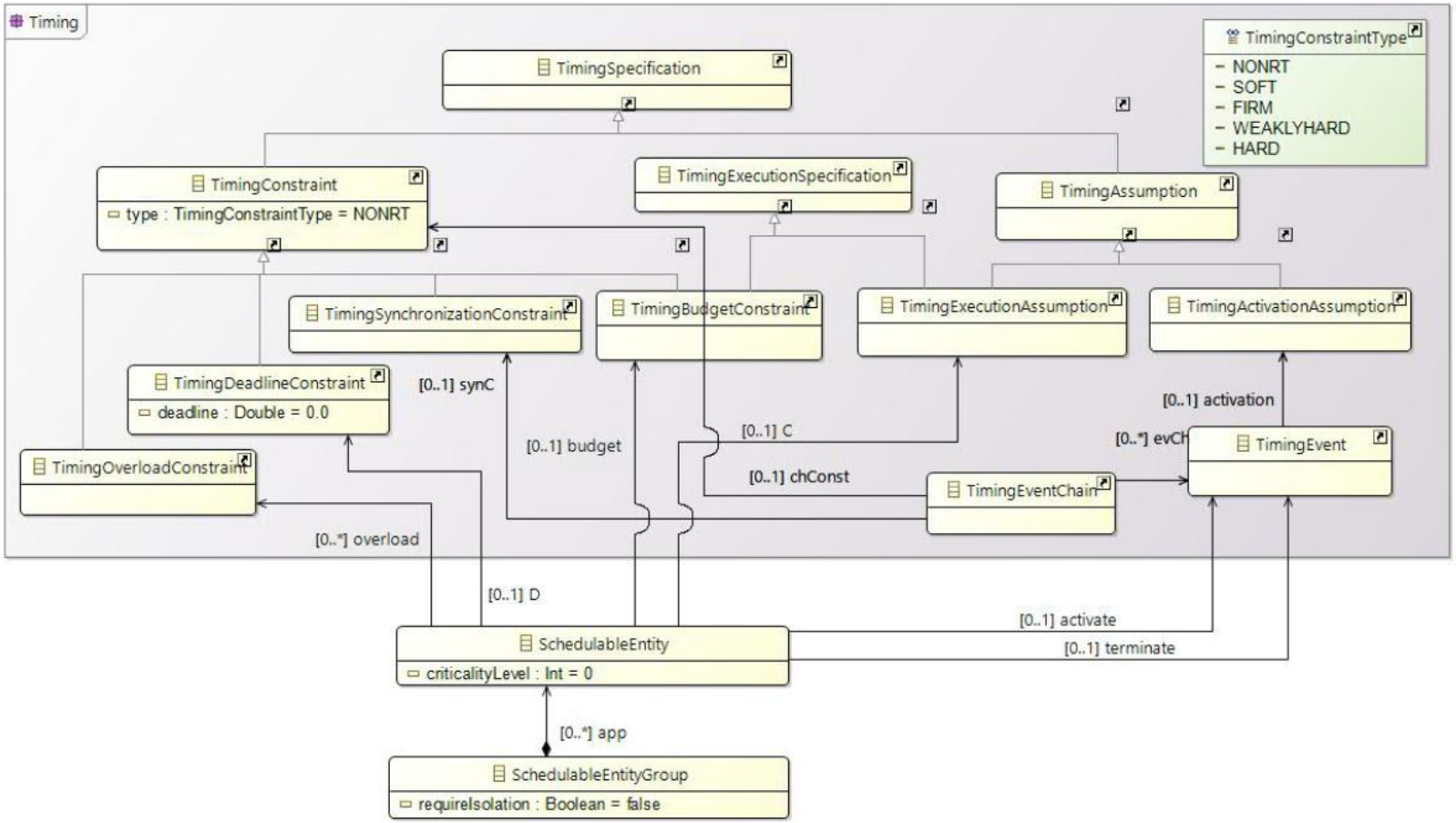
Mapping to partitions



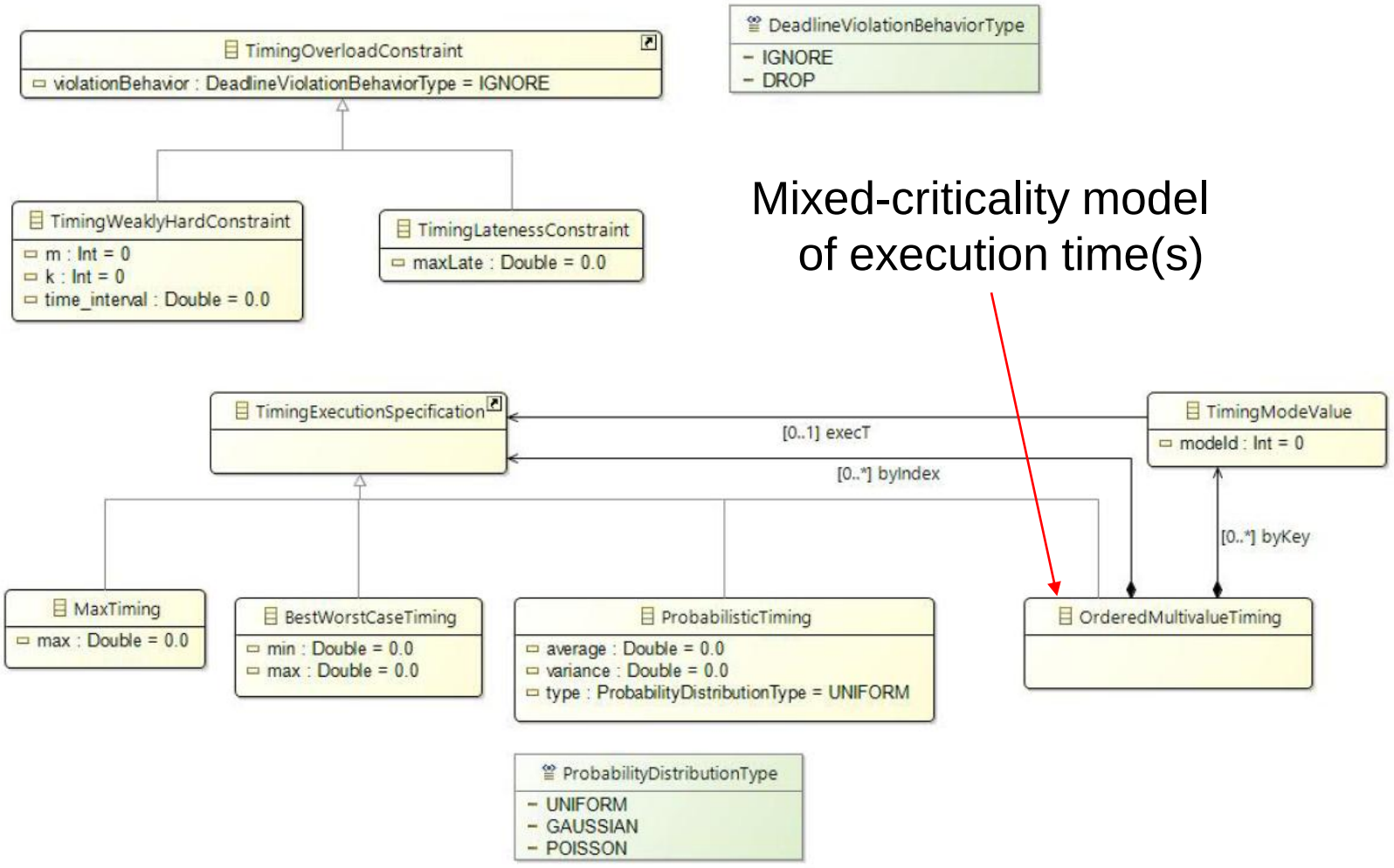
Abstract Models (Security & Safety)



Abstract Models (Time – general)

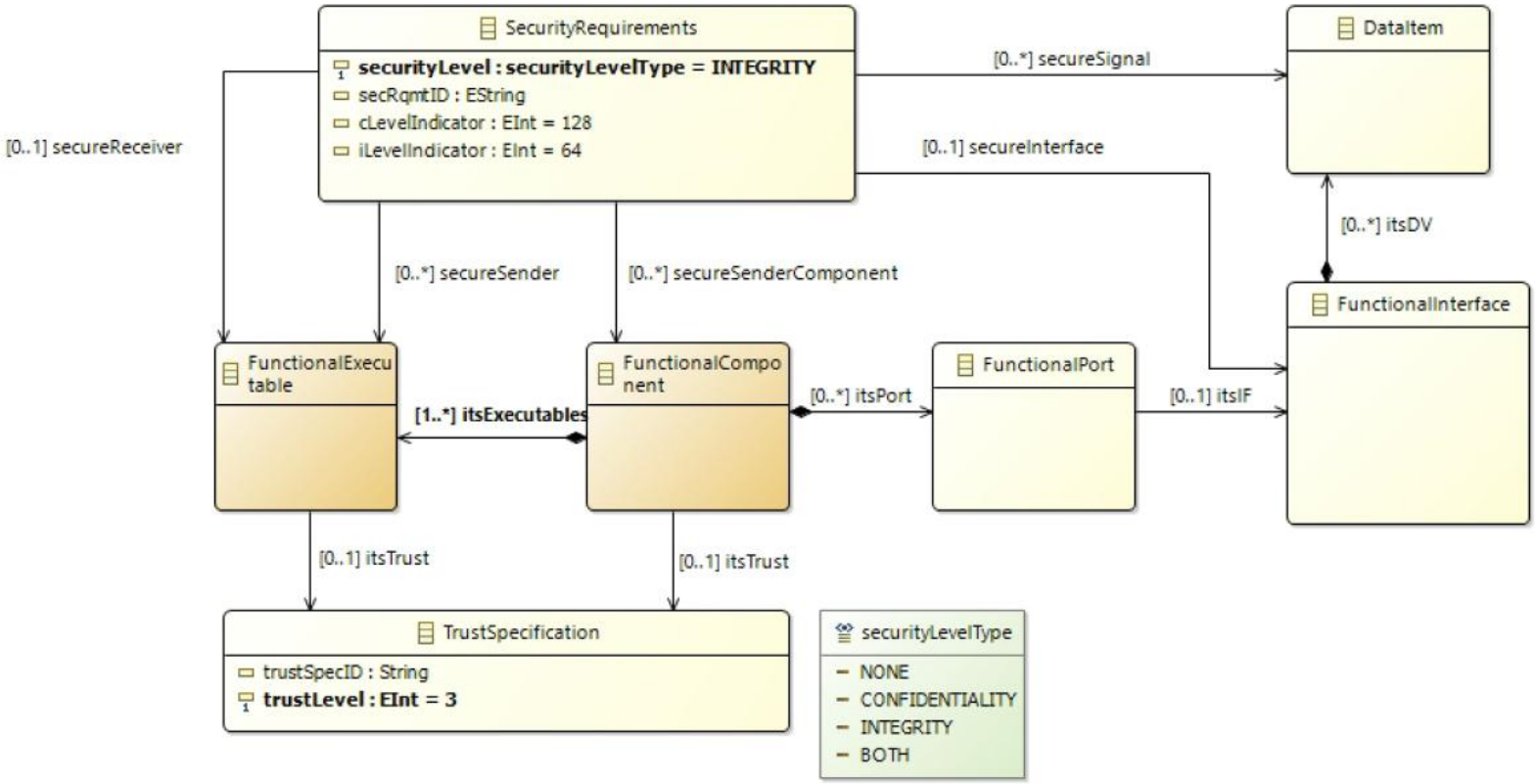


Abstract Models (Time)



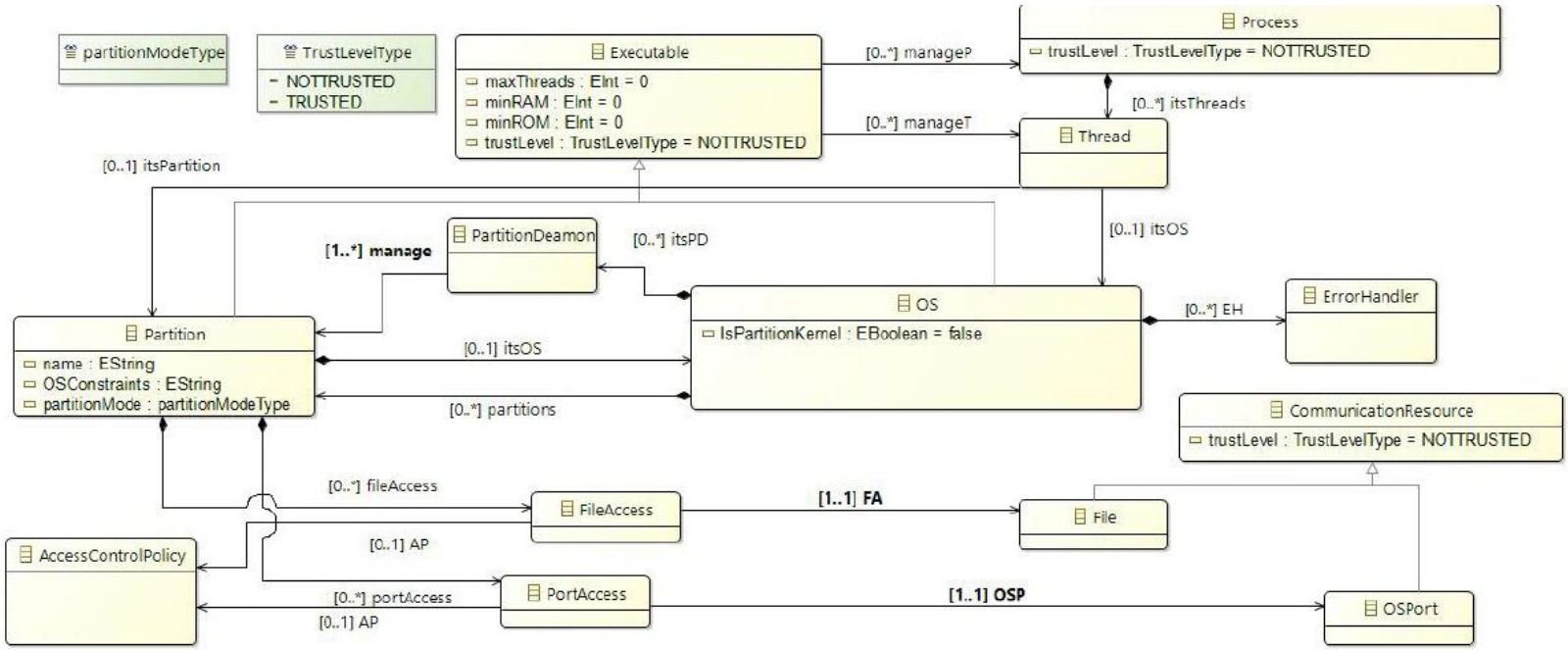
Abstract Models (Security)

A security level specification applies to a component, a port, an interface or a connection



Abstract Models

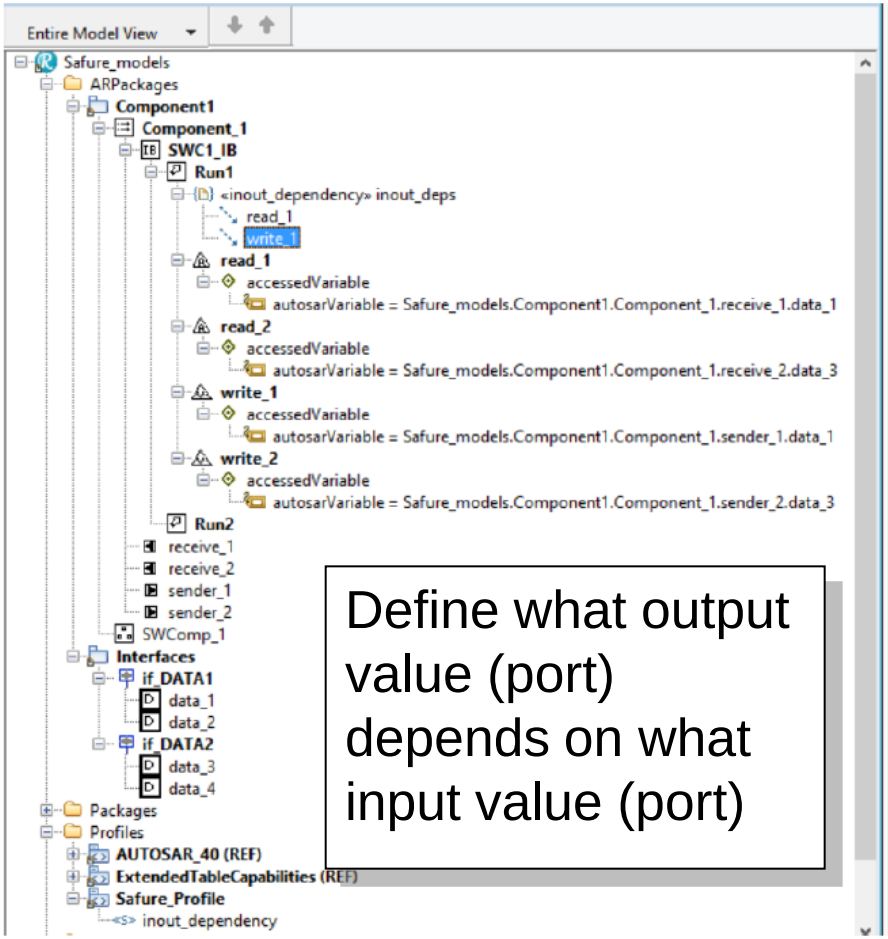
Patterns: Protection kernel



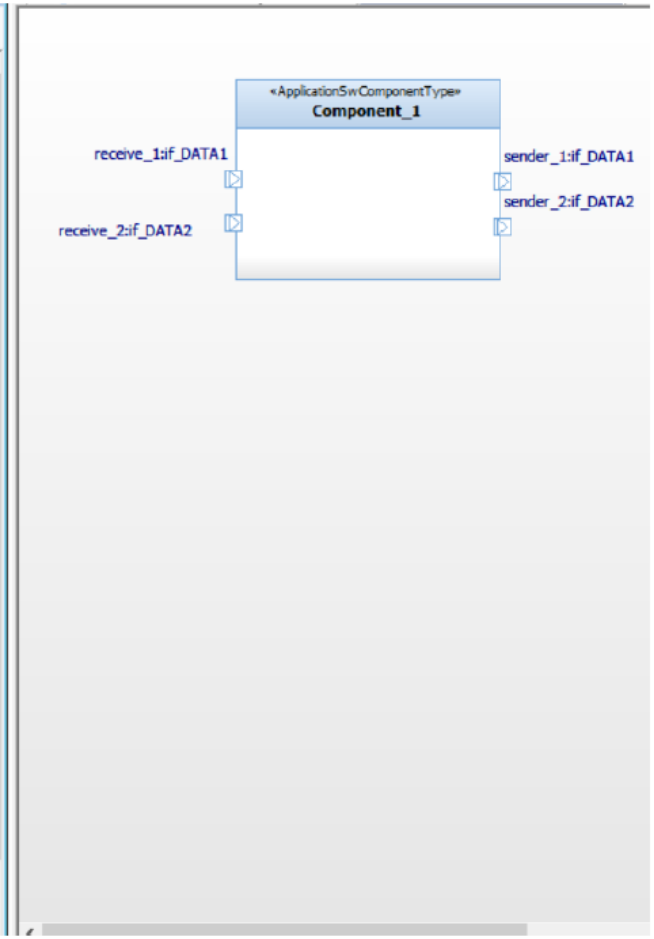
Concrtete Modeling

- **Performed using Rhapsody**
- Rhapsody implements AUTOSAR as an UML profile
  - Getting both worlds at once
- Identify possible ways of representing the concepts
  - In most cases multiple options available
  - Strive for simplicity
- Select which base metaclass needs to be extended
  - Sometimes constrained by Rhapsody implementayion on AUTOSAR stereotypes

Concrete Models – Functional dependencies

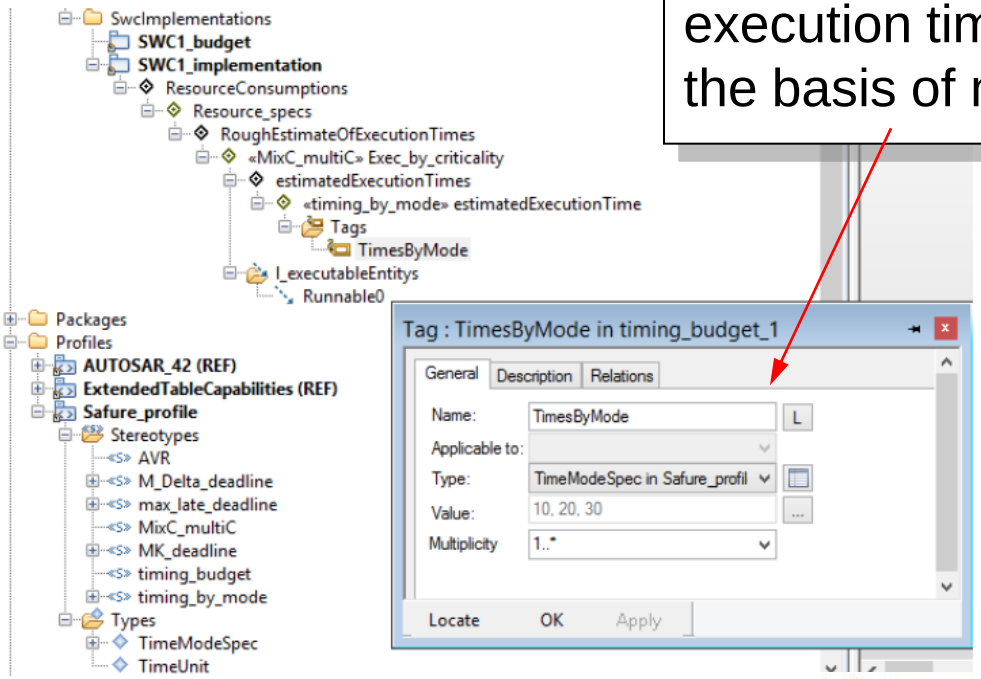


Define what output value (port) depends on what input value (port)

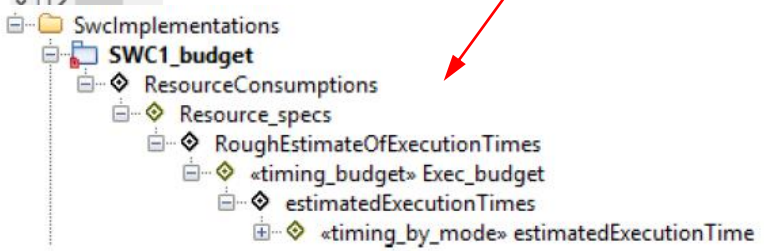


Concrete Models - Timing

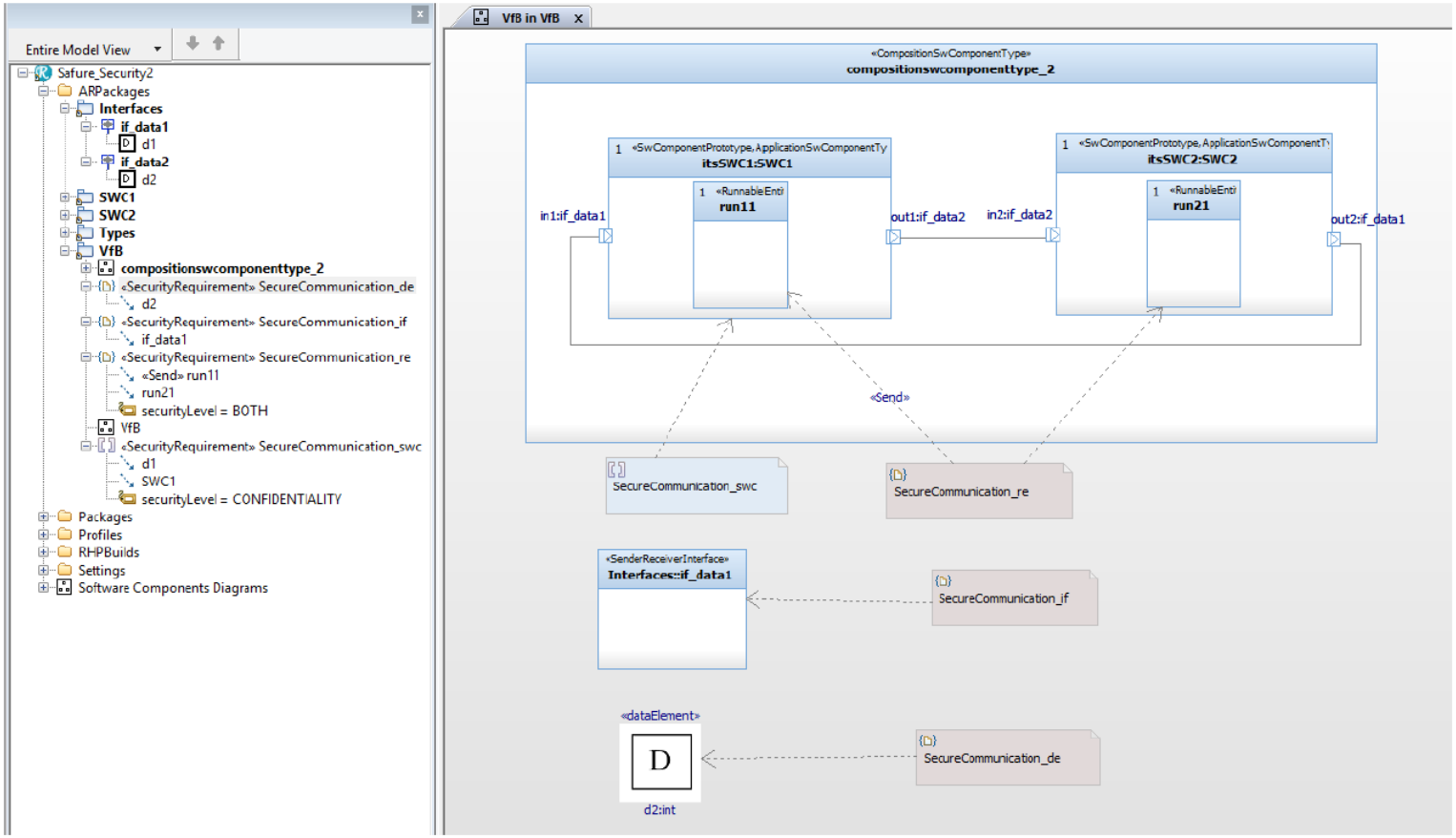
Selecting execution times on the basis of modes



Assigning timing budgets



Concrete Models - Security



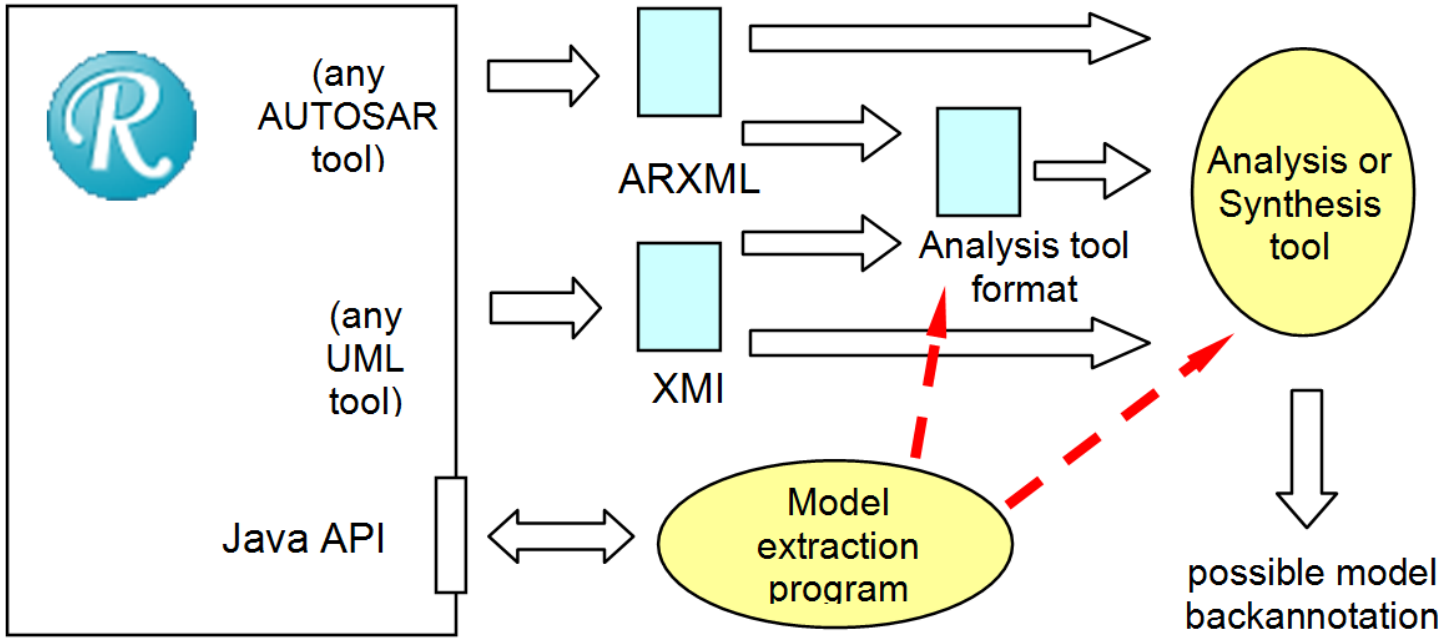
Applications

- Case 1: Synthesis of security implmentations or component proxies
- Case 2: m-k analysis of real-time systems
- Case 3: automatic generation of task code using the AUTOSAR features for timing isolation from specifications (models) of mixed-critical systems.

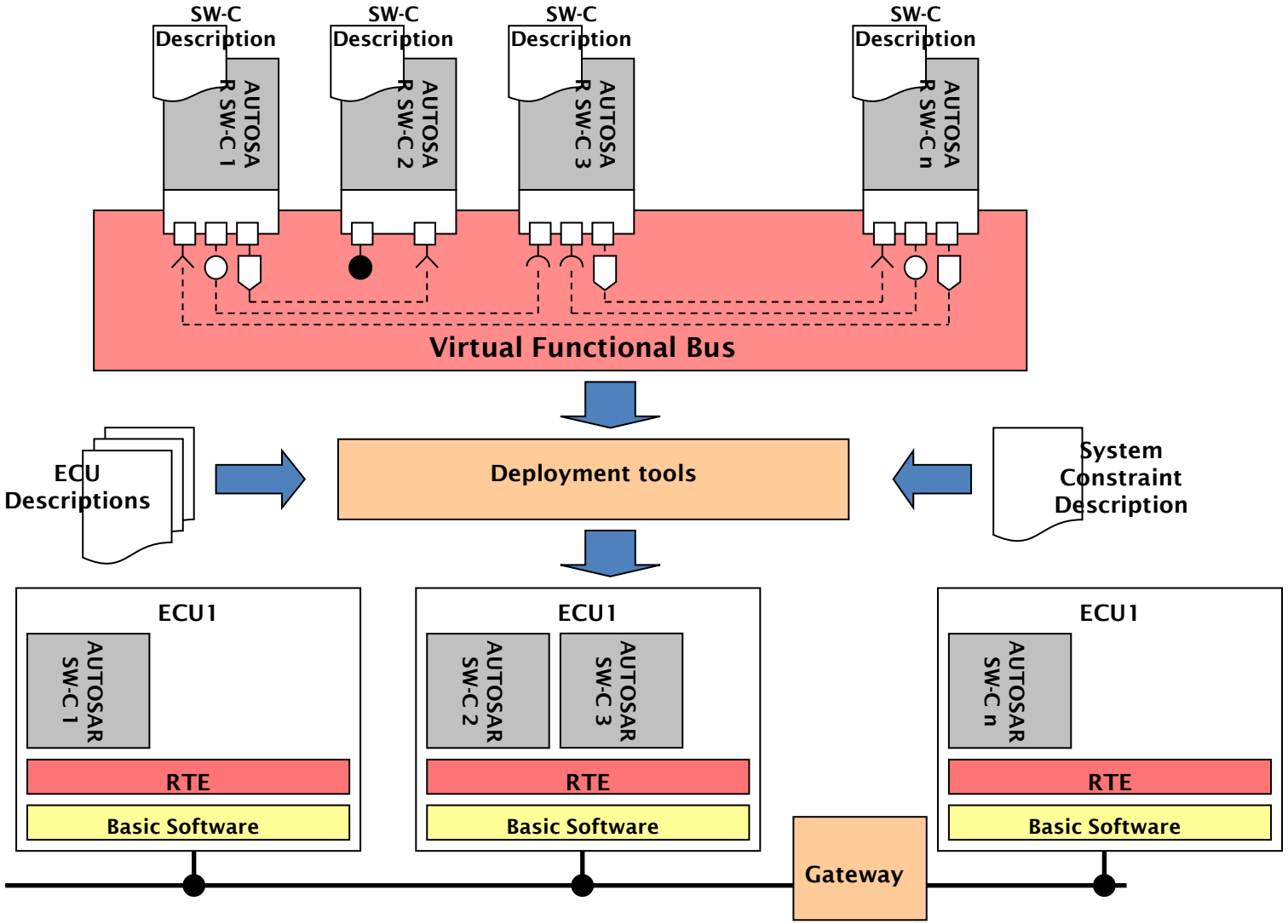
From models to analysis and synthesis

Flow with Rhapsody

- The model with extensions is now available as ARXML, XMI or through the Rhapsody API for analysis and synthesis



Background: The AUTOSAR Process



Background: The AUTOSAR Process - RTE

The task model and the communication implementation are automatically generated by tools (RTE generator) based on the system model specifications

The RTE generator is responsible for generating code, which implements the connectors between the ports.

This generated code is dependant on the mapping of the software components to ECUs.

If the connector connects two components on the same ECU a local communication stub can be generated.

Otherwise, a stub that uses network communication must be generated.

Intra-ECU connector

```
Rte_Write_Client_wiperWasher_start(…) {
        modify variable
}
```

Inter-ECU connector

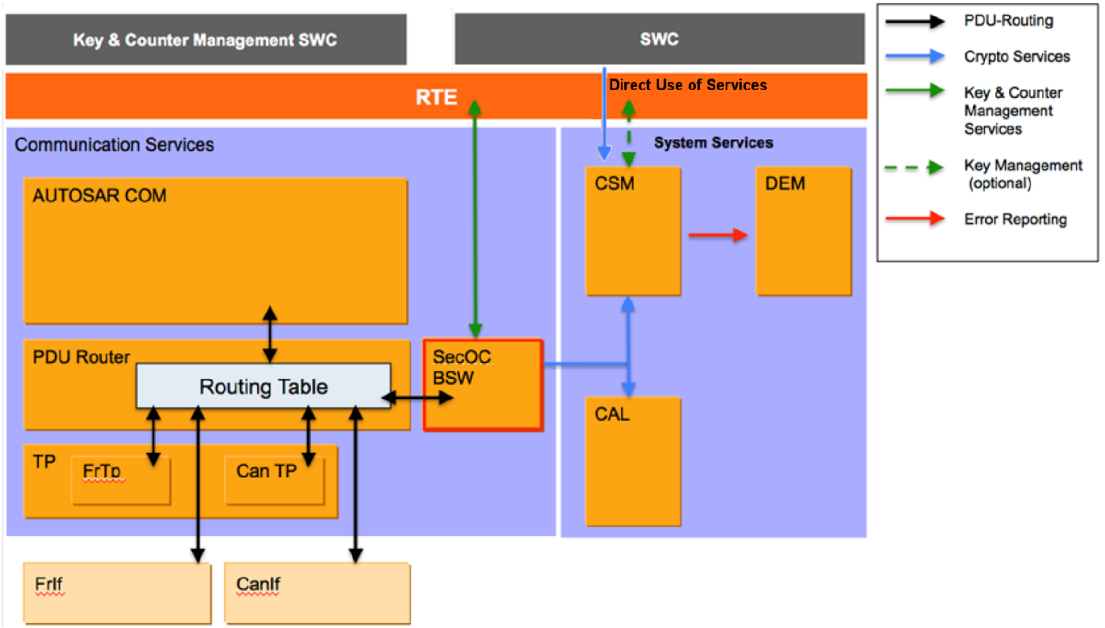
```
Rte_Write_Client_wiperWasher_start(…) {
        access COM
}
```

AUTOSAR Modeling Extensions for Security

Secure On-board Communication (SecOC) allow the transmission of secured data between two or more peers over a network;

Crypto Abstraction Library (CAL) provides other BSW modules and application SW components with cryptographic functionality.

CSM: an AUTOSAR service which provides cryptographic functionalities based on a SW or hardware (HSM) module.

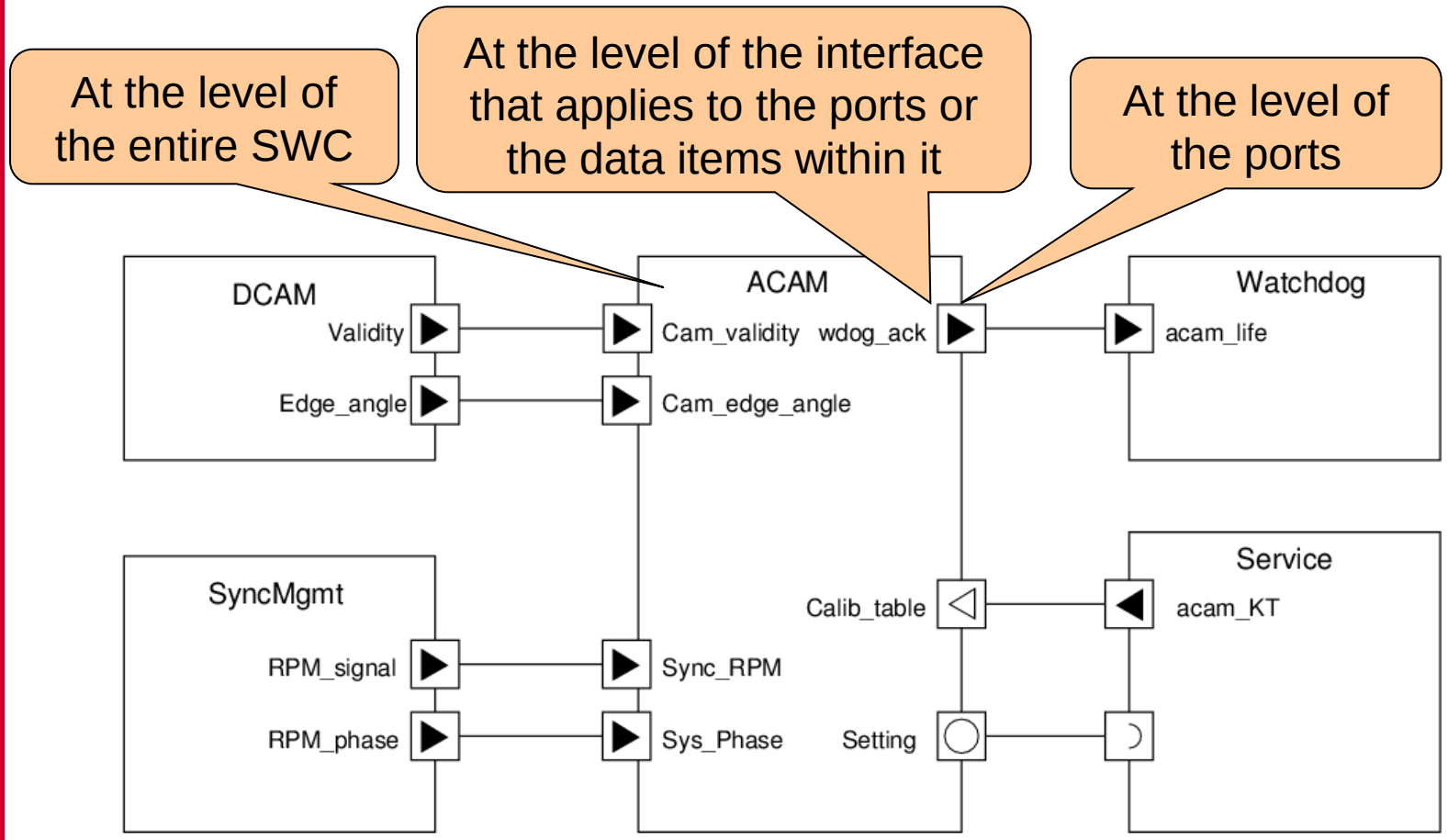


Only services are provided – **no application-level specifications**

SWC access directly CSM: **no hiding of services from SWC**

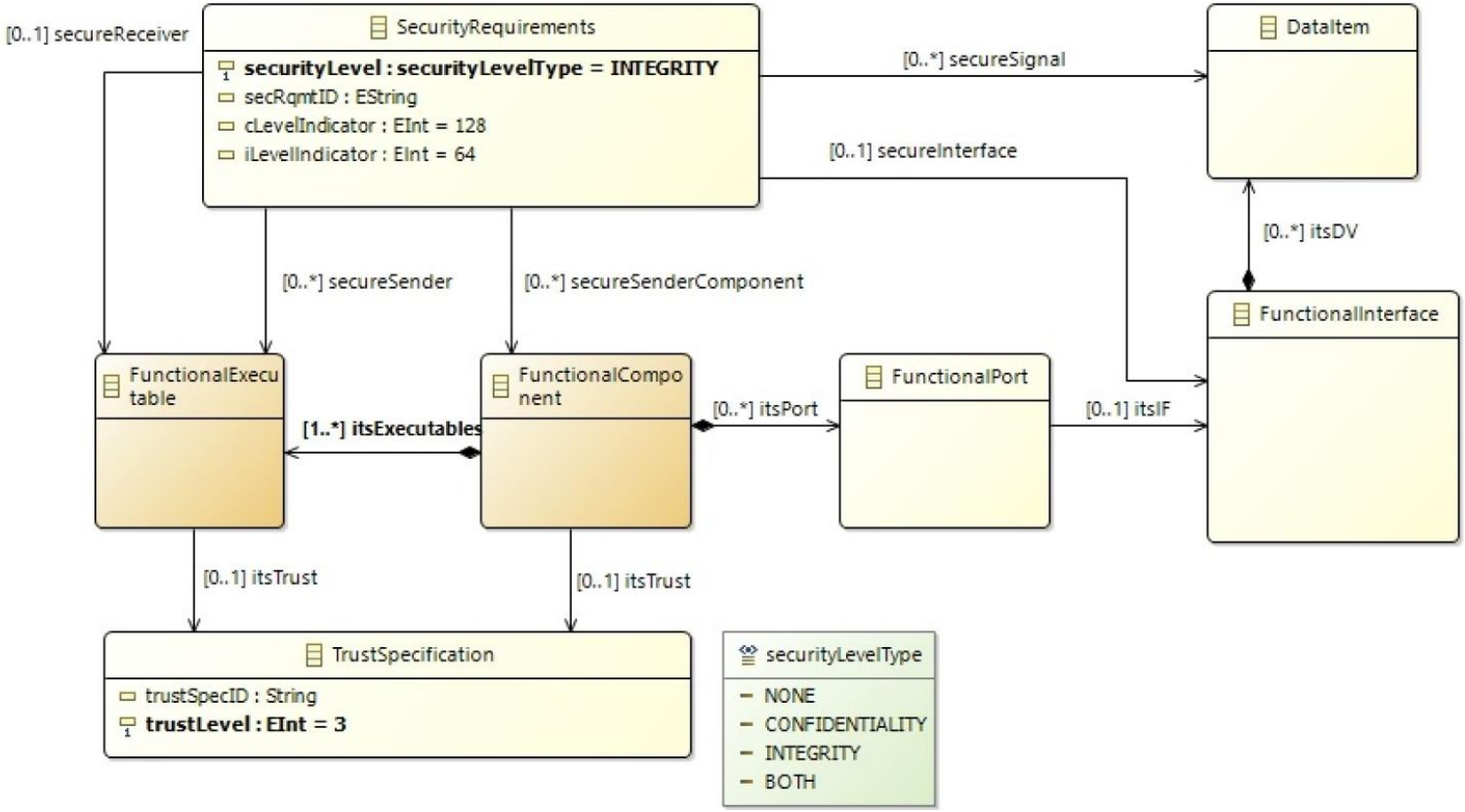
AUTOSAR Modeling Extensions for Security

- Desirable for a user to define the level of confidentiality and integrity of its communications



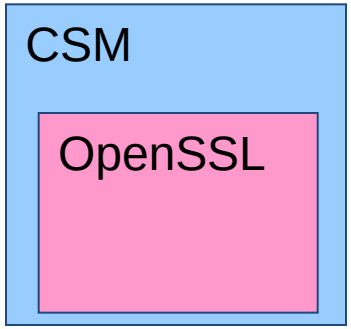
AUTOSAR Specifications for Security: Extensions

- Safure metamodel for Security Reqmts to be applied to the AUTOSAR SWC modeling elements



Building the CSM Module

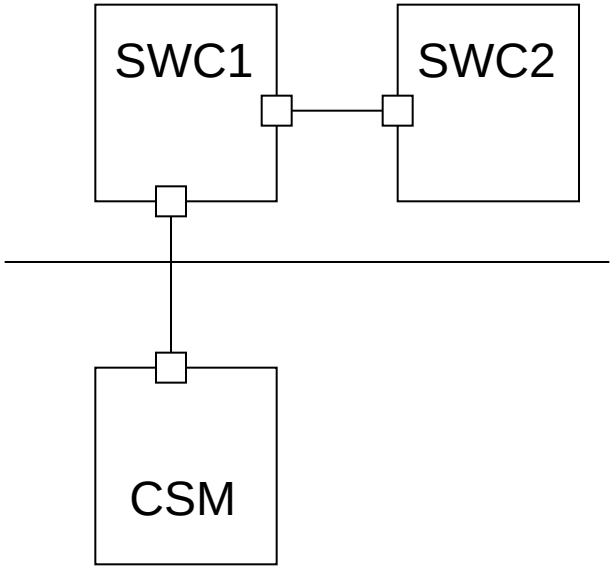
- To support our experiments, we implemented a CSM library compliant with the AUTOSAR standard.
- The library is implemented in C-language and relies on OpenSSL.
- The services implemented are symmetrical encryption/decryption and MAC generation/verication,
  - To test confidentiality and integrity levels.
- The CSM supports the processing of a single instance of each service at a time.
  - Each service makes use of a conguration structure, where all the information regarding the current request are stored.
  - When a new instance of a service is created, a configuration structure is allocated.
- Only synchronous services are implemented

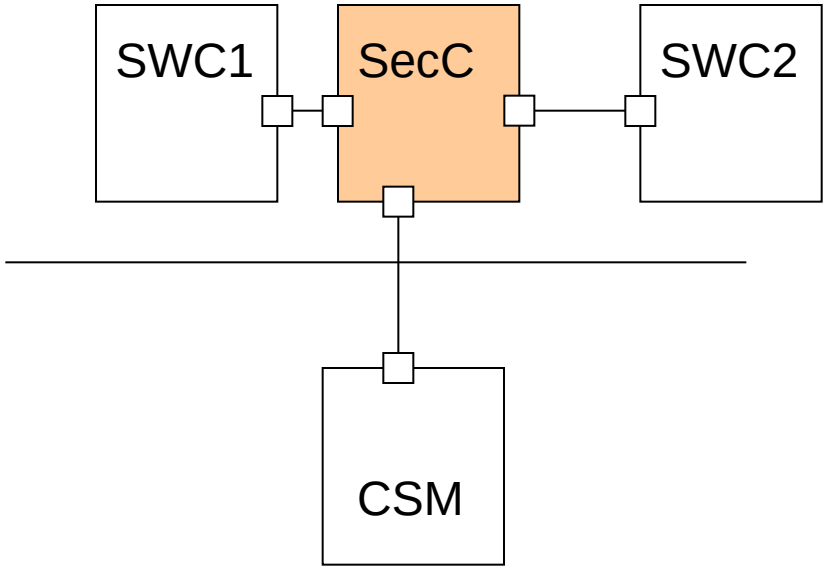


Option 1: Generation of Filter component

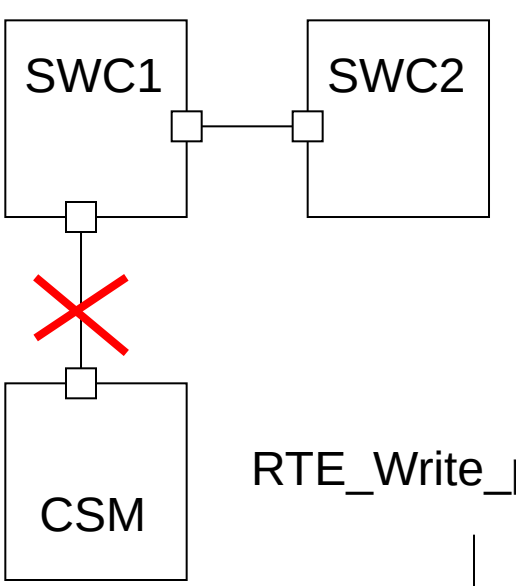
Two approaches to provide an automatic implementation of security requirements and avoid the direct use of crypto services from the user code

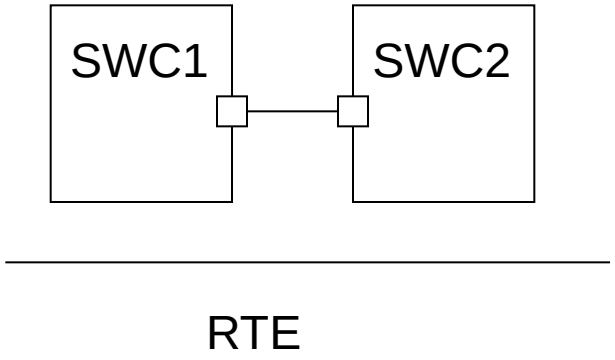
1. Automatic generation of a filter component
2. Customized generation of RTE code performing encryption before transmission





Option 2: Generation of RTE Code

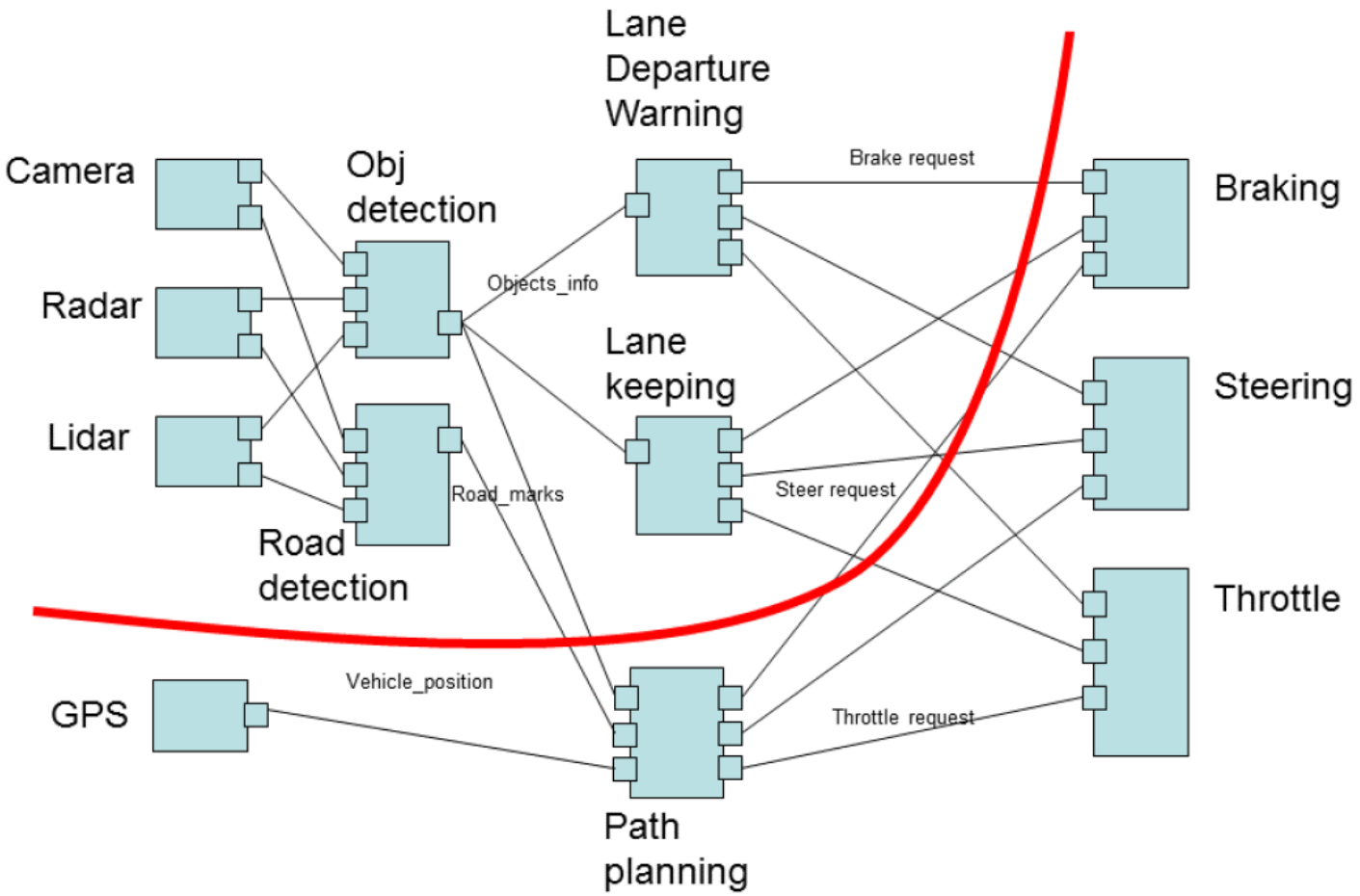




RTE_Write_port_data(value)

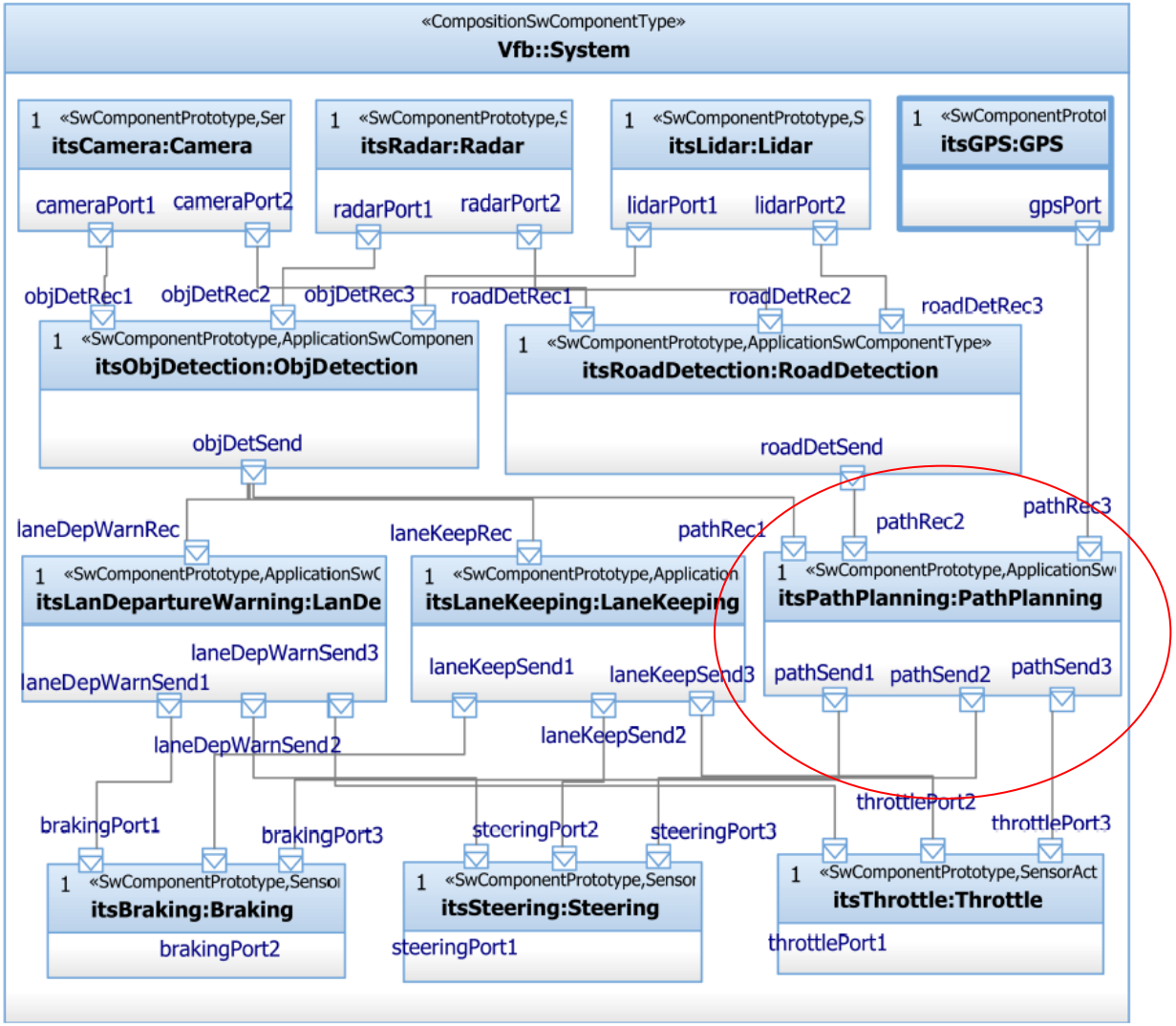
Implementation that uses the CSM API to provide the level of confidentiality specified in the model

Applied to an example derived from active safety functions



An example derived from active safety functions

The AUTOSAR model of the sample application with the component under analysis



Adding RTE-level code

When the encryption is performed in the RTE code, it is completely hidden in the model view.

A customized RTE generator is required

- Implemented with ARXML parsing and python processing

```
Std_ReturnType Rte_Write_gpsPort_GPS(uint32 datum) {
    ...
    Csm_SymEncryptStart(cfgId, ..., sizeof(uint32));
    Csm_SymEncryptUpdate(cfgId, datum_buf, ... );
    Csm_SymEncryptFinish(cfgId, ...);
    res = Com_SendSignal(sigID, datum_buf);
    ... // AUTOSAR COM
    return res;
}
```

Code for filter component

Encryption code added to the runnable code performing forwarding of data from the Filter

```
FUNC(void, RTE_APPL_CODE) run0(void)
{
    ...
    Rte_Read_Port0in_GPSFilter(datum_buf);
    Csm_SymEncryptStart(cfgId, ..., sizeof(uint32));
    Csm_SymEncryptUpdate(cfgId, datum_buf, ... );
    Csm_SymEncryptFinish(cfgId, ...);
    Rte_Write_Port0out_GPSFilter(*datum_buf);
}
```

m-k modeling and analysis

- Very small number of “true” hard real-time systems
- Need for models that account for overload in several control domains (where deadline can be missed)
  - Informally, we can miss deadlines but not too many and not too many in a row
- m-k model: You can miss at most m deadlines out of k instances
- Not the only possible abstraction
- Length and number of deadline misses in longest busy period

(all these specifications are available from out modeling extensions)

The m-k model: Our solution

Work on m-k analysis presented at EMSOFT

Y. Sun, M. Di Natale “Weakly Hard Schedulability Analysis for Fixed Priority Scheduling of Periodic Real-Time Tasks”

Contributions wrt previous work

- Not restricted to offset determined systems
- Can sweep a range of m-k options and even find the minimum m for a given k
  - Easily done since it is based on an optimization formulation

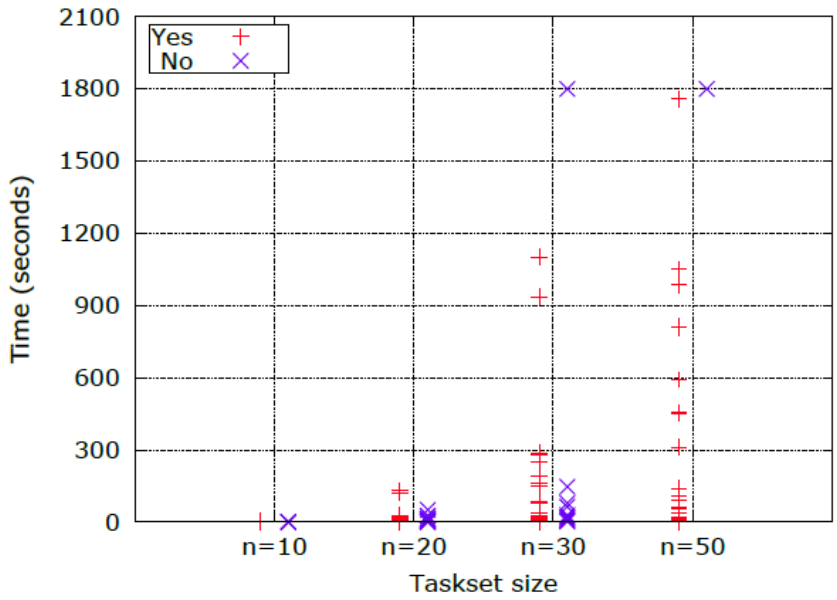
Limitations

- Still limited to periodic load
- Does not scale beyond 20 tasks and k>10
  - Do we really need those?

The m-k model: an MILP solution

- Relaxing some constraints (number of interferences) and then refining to limit pessimism
- Feasibility or optimization formulation
  - ***Maximize # of misses m in any given window of k***
- Results *very* close to true optimum
- Runtimes acceptable for many configurations

m=2, k=5



SAFURE Grant Agreement No. 644080

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644080."

"This work was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 15.0025. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Swiss Government."

If you need further information, please contact the coordinator:

TECHNIKON Forschungs- und Planungsgesellschaft mbH
Burgplatz 3a, 9500 Villach, AUSTRIA
Tel: +43 4242 233 55 Fax: +43 4242 233 55 77
E-Mail: coordination@safure.eu

WP2: Abstract Models

Protection kernel pattern