Towards the Model-Driven Engineering of Secure yet Safe Embedded Systems

Ludovic APVRILLE (Télécom ParisTech),
Yves ROUDIER (EURECOM)

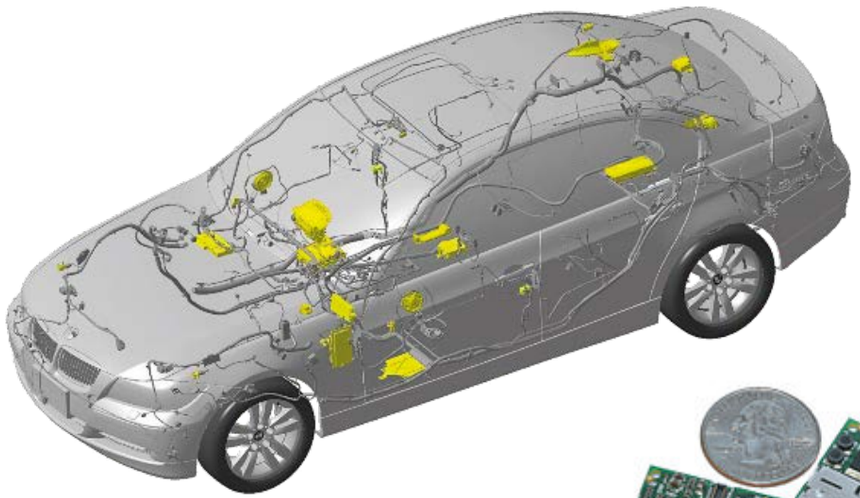
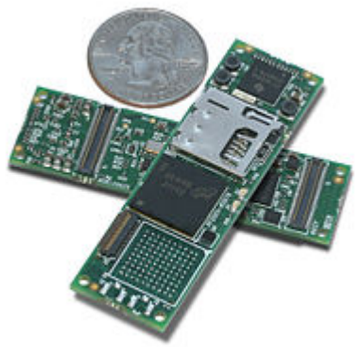
Outline

- **Context: Security for Embedded Systems**
- **Security Requirements and HW/SW Partitioning**
- **Design of Cryptographic Protocols**

Context: Embedded Systems

- **Embedded systems?**
  - “Computer system with a dedicated function within a larger mechanical or electrical system” [Wikipedia]
  - Designed on-purpose for specific control functions
  - Integrated: Software + Hardware
    - ☞Many technologies, increasingly distributed and communicating systems



Embedded Systems: Examples of Threats

- **Automotive Systems**
  - Tire Pressure Monitoring System wireless link [Rouf 2010]
  - Keyfob authentication [Francillon 2011]
  - Vulnerabilities of Onboard Network [Koscher 2010]
  - HU remotely exploitable vulnerabilities [Checkoway 2011]
  - Locksmith tool(CAN/LIN injection) [MultiPick 2012]

© 2012, MultiPick

- **Avionics Systems**
  - Abusing the Automatic Dependent Surveillance Broadcast (ADS-B) protocol [Costin 2012]
  - Use of exploits in Flight Management System (FMS) to control ADS-B/ACARS [Teso 2013]

© 2013, Teso

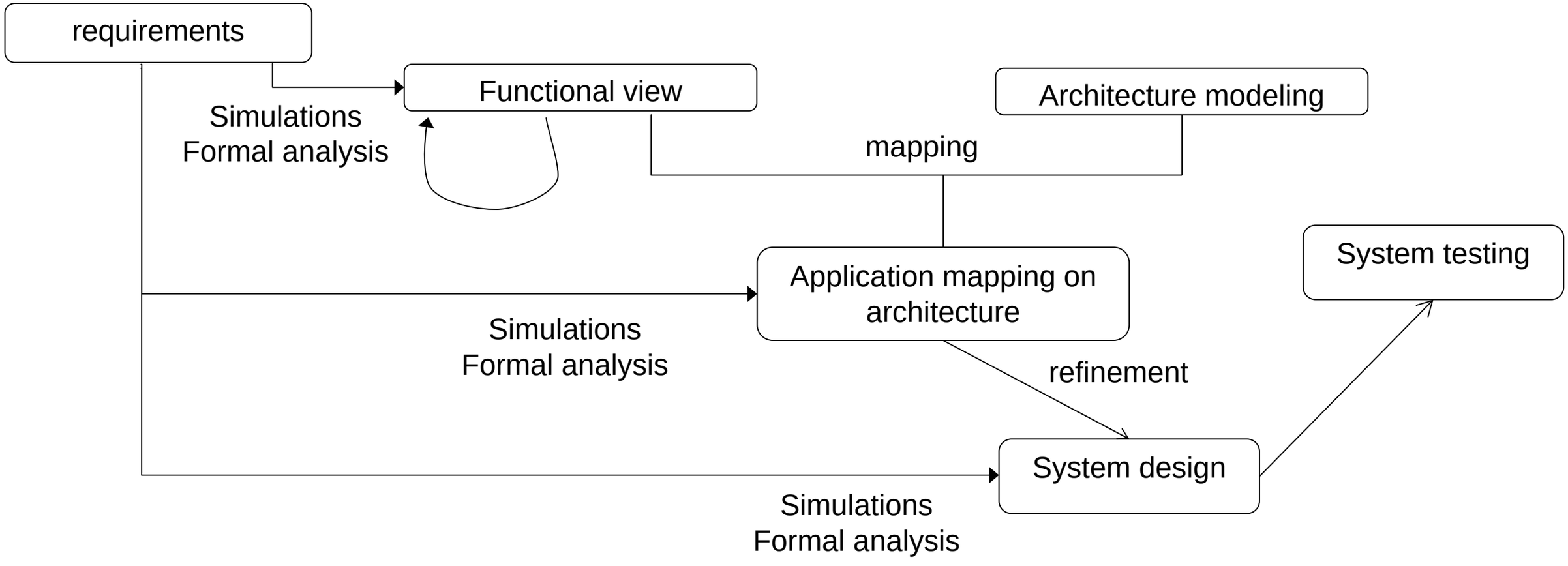
- **Internet of Things**
  - 750000 spams sent in 2 weeks from compromised refrigerators [ProofPoint 2014]
  - Proof of concept of attack on IZON camera [Stanislav 2013]

Our Proposal: SysML-Sec

- **Focus: holistic approach**
  - Bring together system engineers & security experts
- **Security is not supported by SysML**
  - Yet security is not an add-on
  - Can have adverse effects on safety/real-time properties
- **Security requirements: available tools …**
  - … don't address functional and safety requirements
  - Some tools directly address security mechanisms configuration
  - Do not handle hardware or HW/SW mapping (elicitation)
- **Hardware/Software partitioning is central**
  - Support in MDE approaches is often limited, lacks integration with architecture
  - Architecture = CPUs + memories + buses + OS + middleware + software
  - Fails to capture environmental constraints of system (esp. realtime ones)

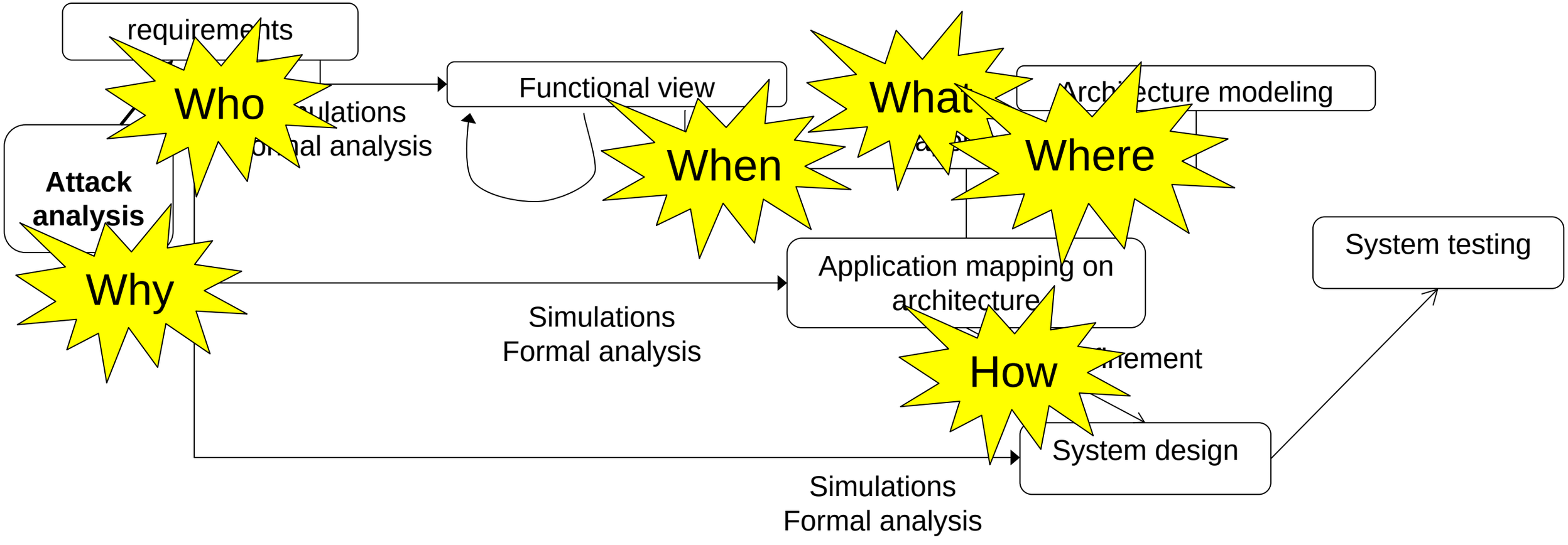
MDE: Y-Chart, V-Cycle

- **System partitioning between HW and SW**
  - Mapping process
  - Objective is to optimize the system wrt. various criteria (cost, area, power, performance, flexibility…)

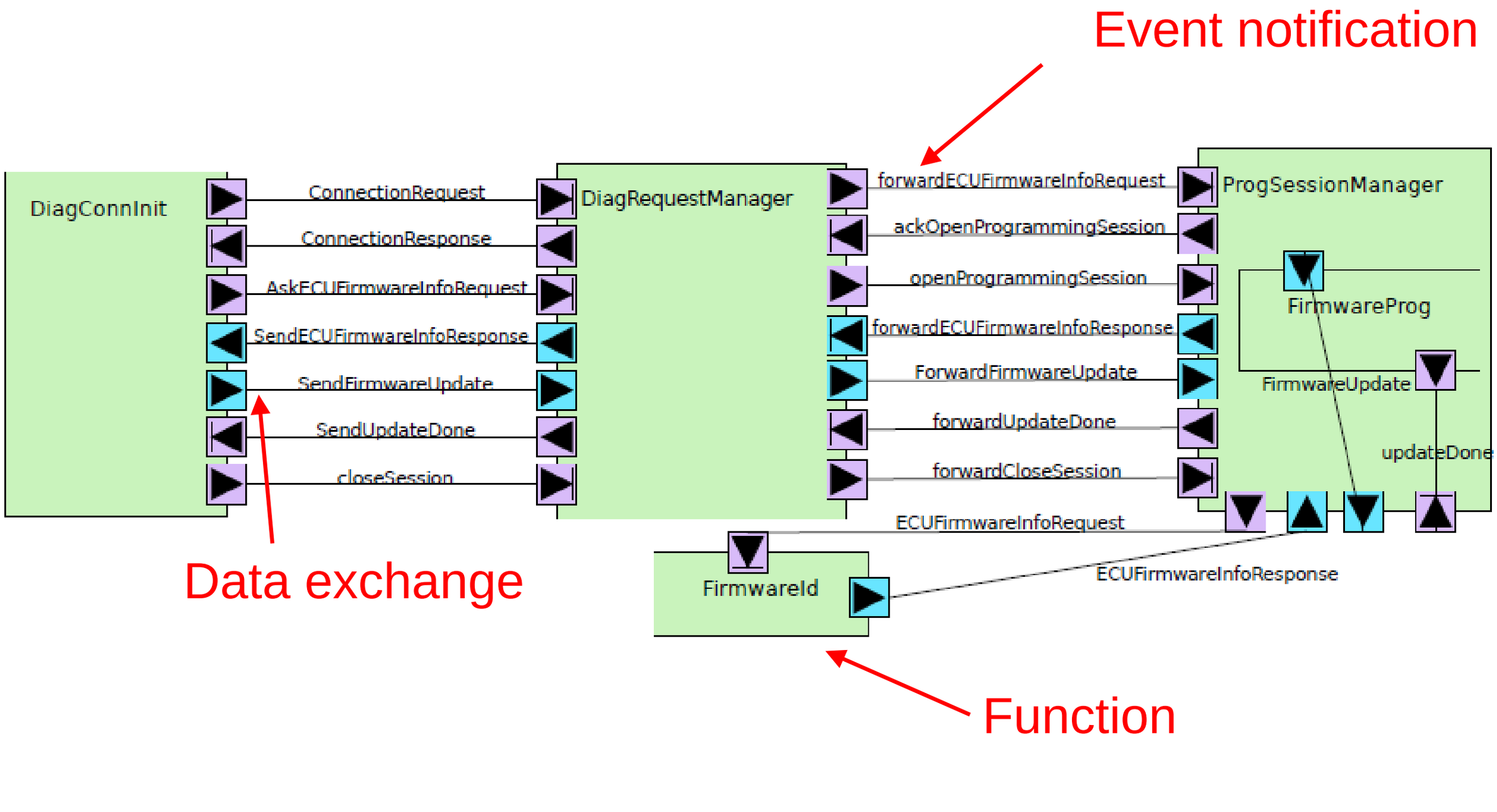


Methodology: The Y-Chart Revisited

- *What*: assets to be protected
- *When*: operation sequences in functions involving those assets
- *Where*: architecture mapping of functions involving those assets
- *Why*: attacks envisioned that motivate security countermeasures
- *Who*: stakeholders + attackers & capabilities (risk analysis)
- *How*: security architecture (e.g., network topology, process isolation, etc.)

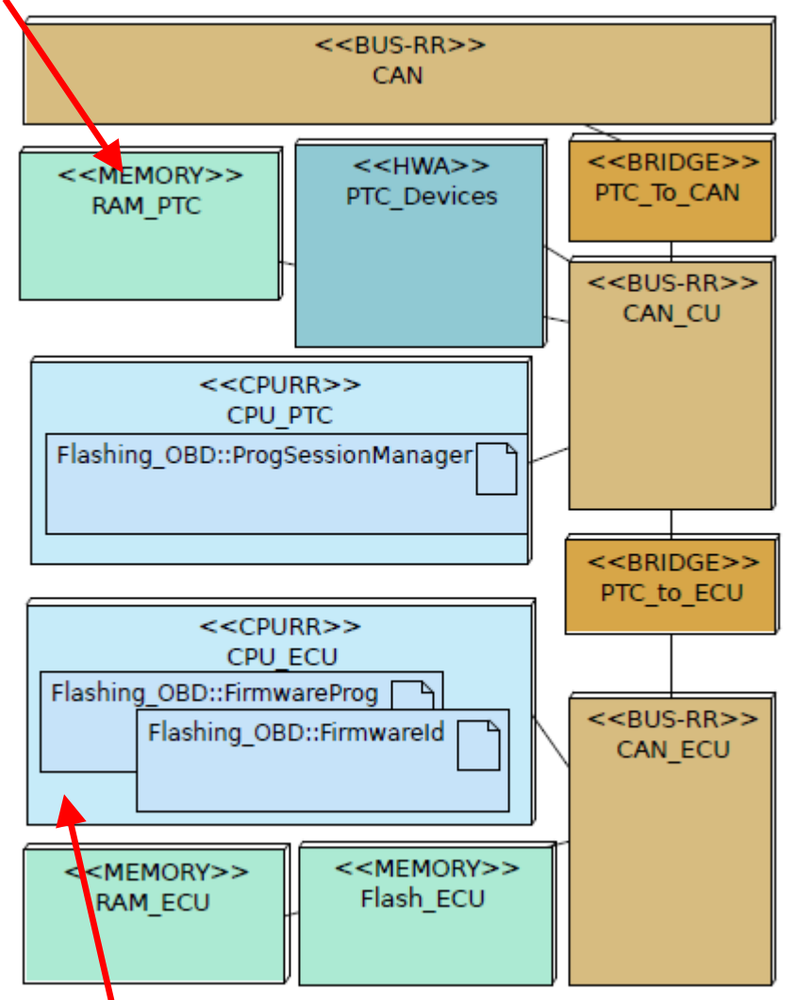


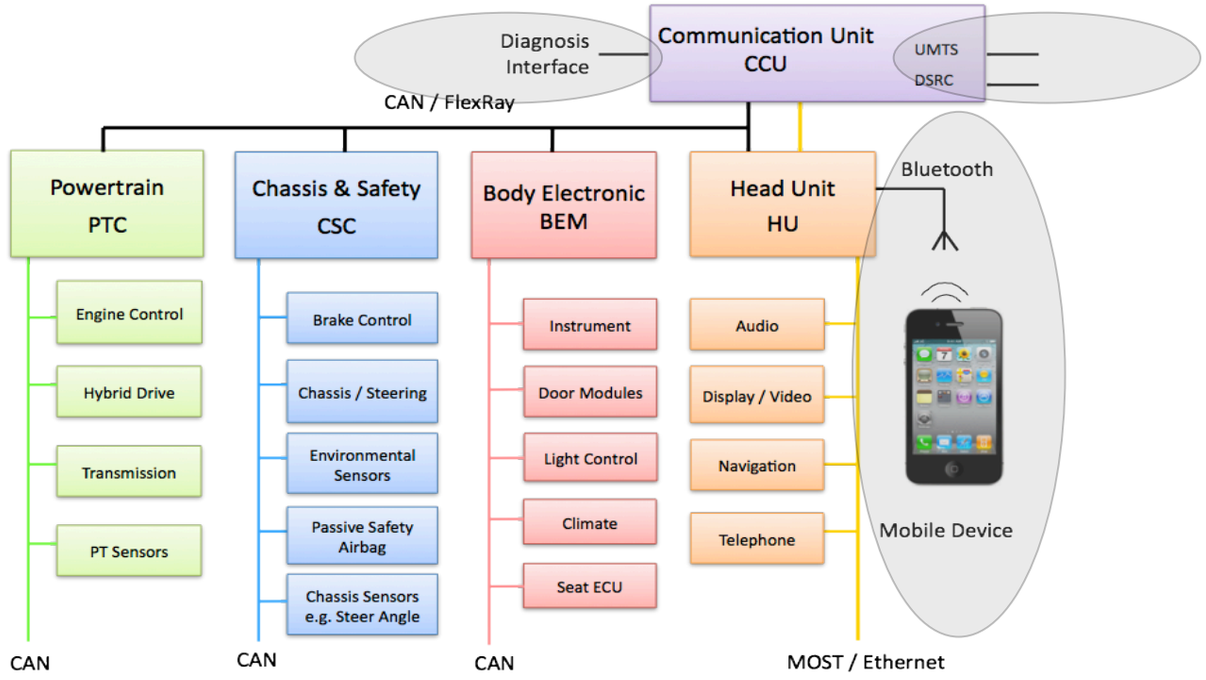
Functional View
Internal Block Diagram



Architectural Mapping Model
Deployment Diagram

Processing Units



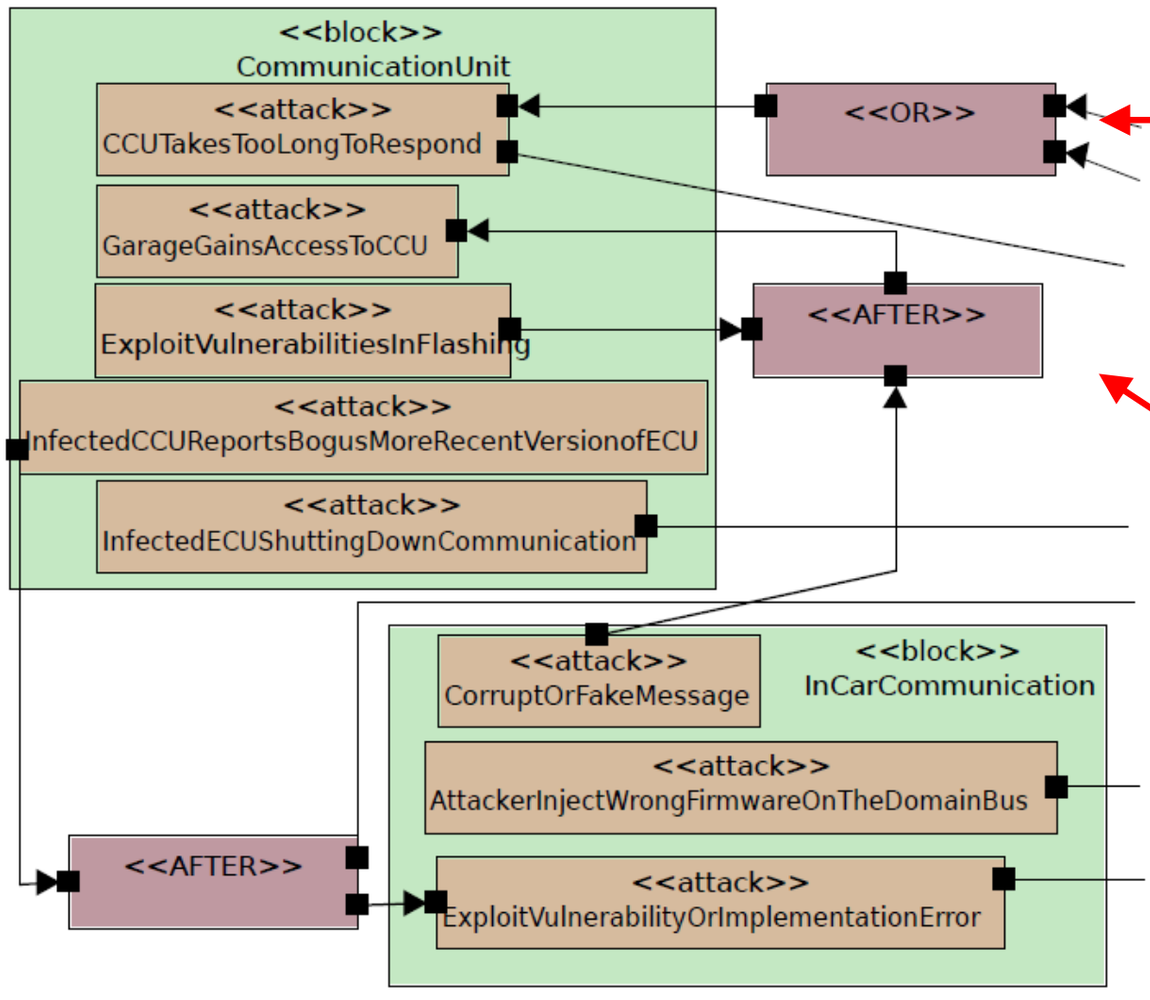


Event/dataflow mapping

Function mapping

Identification of Assets and Vulnerabilities Parametric Diagram

- **HW/SW mapping is very important**
  - To determine perimeter of assets and attacks
  - For traceability of attacks and countermeasures

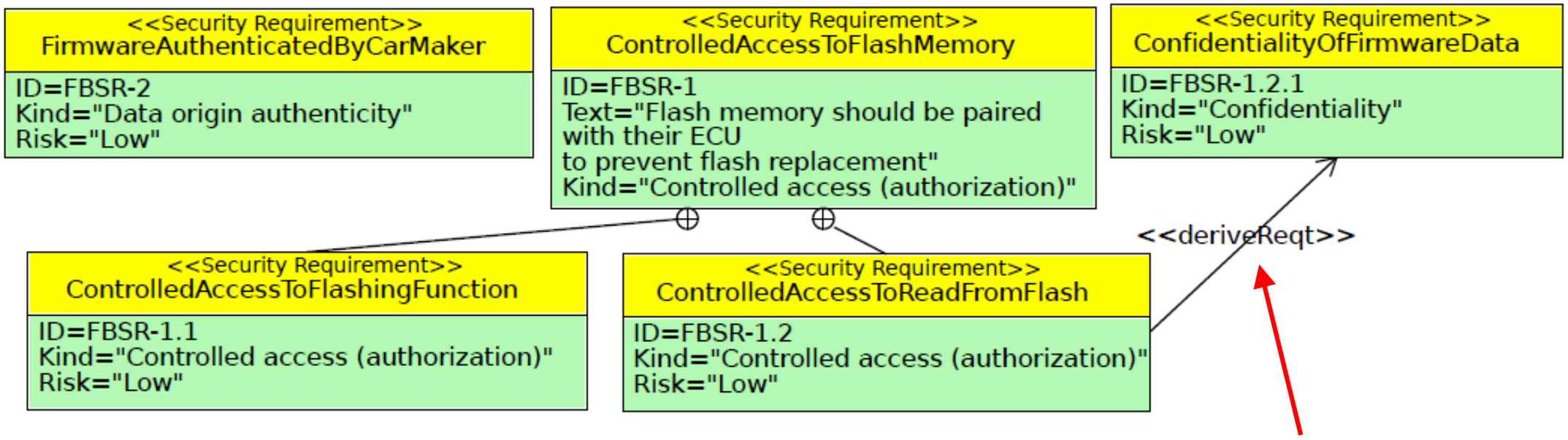


Attack Trees mapped to components

Equations

Security Properties and Types of Countermeasures Requirements Diagram

Security Properties: e.g., Confidentiality, Authenticity, Integrity, Freshness, Availability…



Objective: Trace refinements (more details about requirements) and dependencies (requirements upon which one depends)

SysML Extensions for Security in Software Design

- **Design of a cryptographic protocol also lacks support in SysML:**
  - Trust assumptions: private and public channels
  - Security manufacturing: modelling distribution of cryptographic material (e.g., PSK)
  - Cryptographic components: supporting algorithms
- **Modeling security properties**
  - Formal translation of the semi-formal security requirements
  - Formal semantics (Pi-calculus, as supported by ProVerif)
  - Dolev-Yao attacker model

Conclusions and Future Work

- **Hybrid Approach**
  - Goal-Oriented security requirements engineering integrated in SysML
  - MDE approach: exploits knowledge resulting from HW/SW mapping and model translation
- **Results**
  - Covers the whole methodological development of a system: partitioning, attacks, security requirements, design, validation
  - Software and hardware semantics
  - TTool was used to validate formally and experimentally (simulations and tests) the impact of security mechanisms
    - ☞ Available free of charge (open-source environment)
- **Future directions**
  - Semi-formal checks: requirements consistency / attack coverage
  - Combining security and safety requirements

Thank You!

Questions?