System-Security Co-design

Janos Sztipanovits
David Lindecker
ISIS-Vanderbilt

Berkeley
UNIVERSITY OF CALIFORNIA

Massachusetts
Institute of
Technology

MICHIGAN

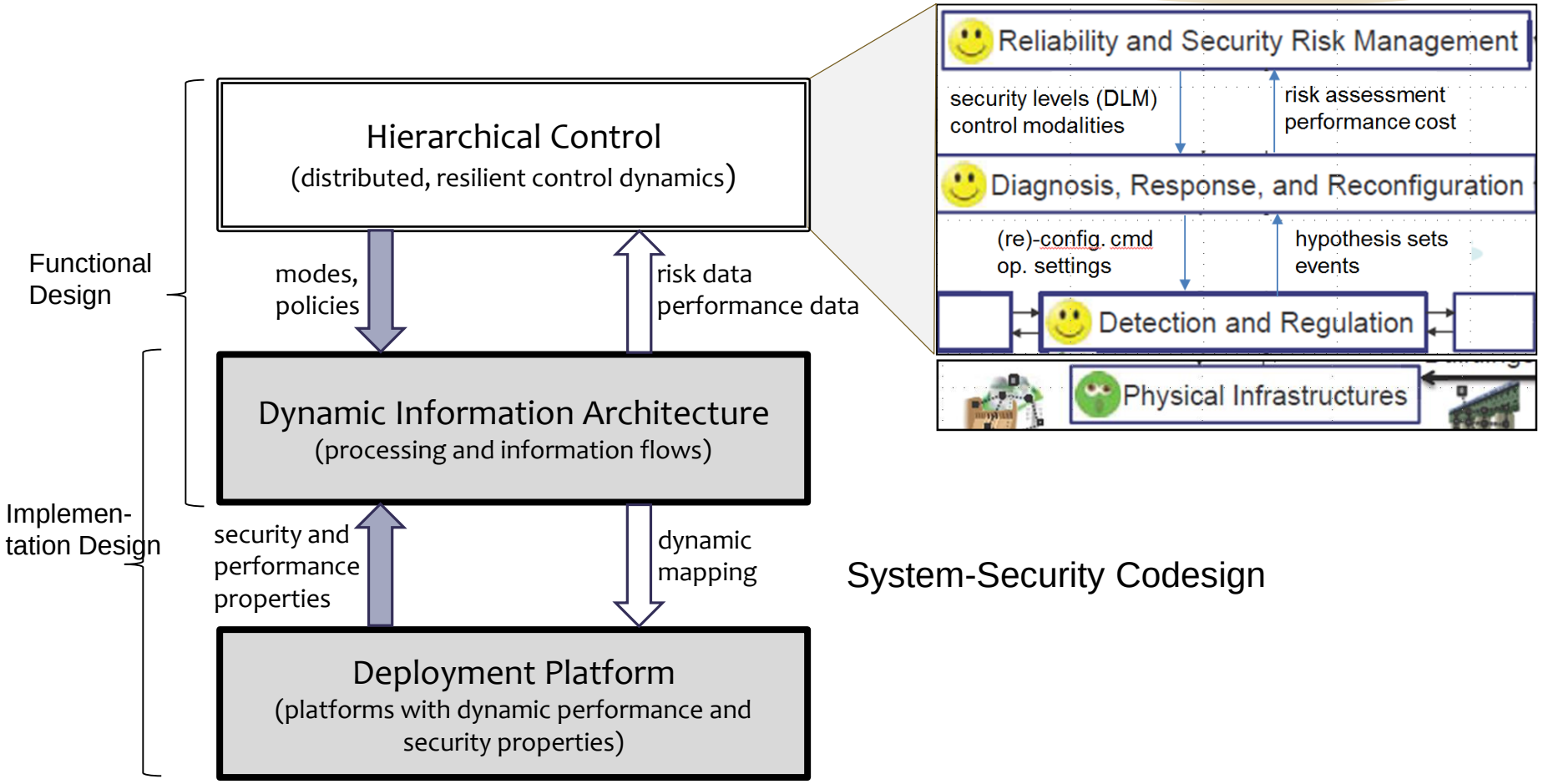
VANDERBILT
UNIVERSITY

Objectives of Reconfiguration

* **Change modes of operation of Detection and Regulation**
  - Diagnosis, Response and Reconfiguration forms a supervisory control mechanism – used in hierarchical control approaches (e.g. Pappas, Tabuada)
* **Re-synthesize implementation architecture**
  - Provide interface for changing required security policies
  - Provide models of information flows required to be implemented
  - Provide models for security and performance characteristics of communication links and computing devices
  - Provide precise specification for the reconfiguration space
  - Develop methods for remapping the information architecture to the implementation architecture subject to functional, performance, timing and security constraints

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

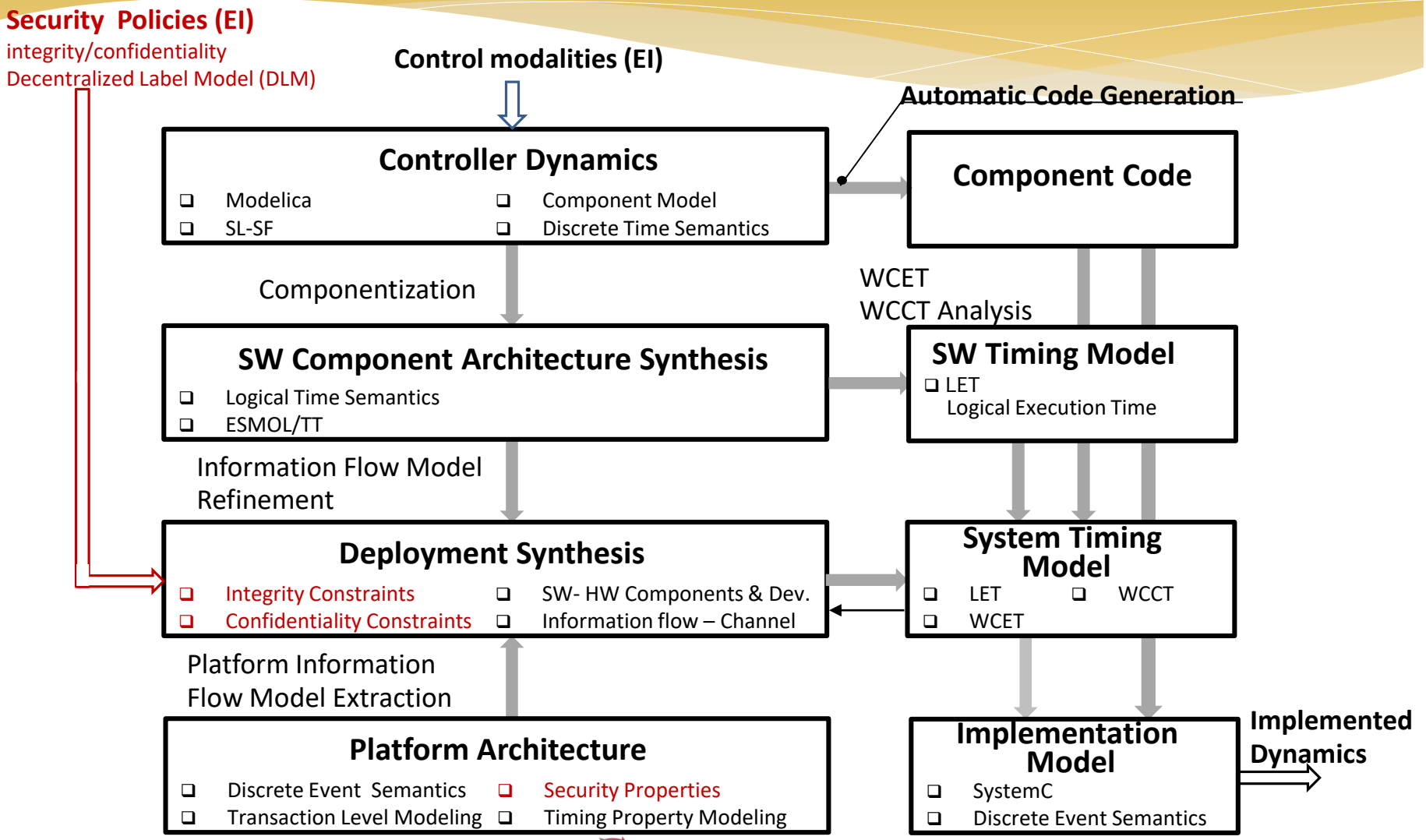
Co-design Problem



System-Security Codesign

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

System – Security Co-design



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Security Concerns Addressed

* **Integrity attacks**
  - Manipulate data (value, timestamp, source identity,..)

* **Confidentiality attack**
  - Leak critical data to unauthorized persons/systems

Integrity and confidentiality restrictions impose constraints on information flows.

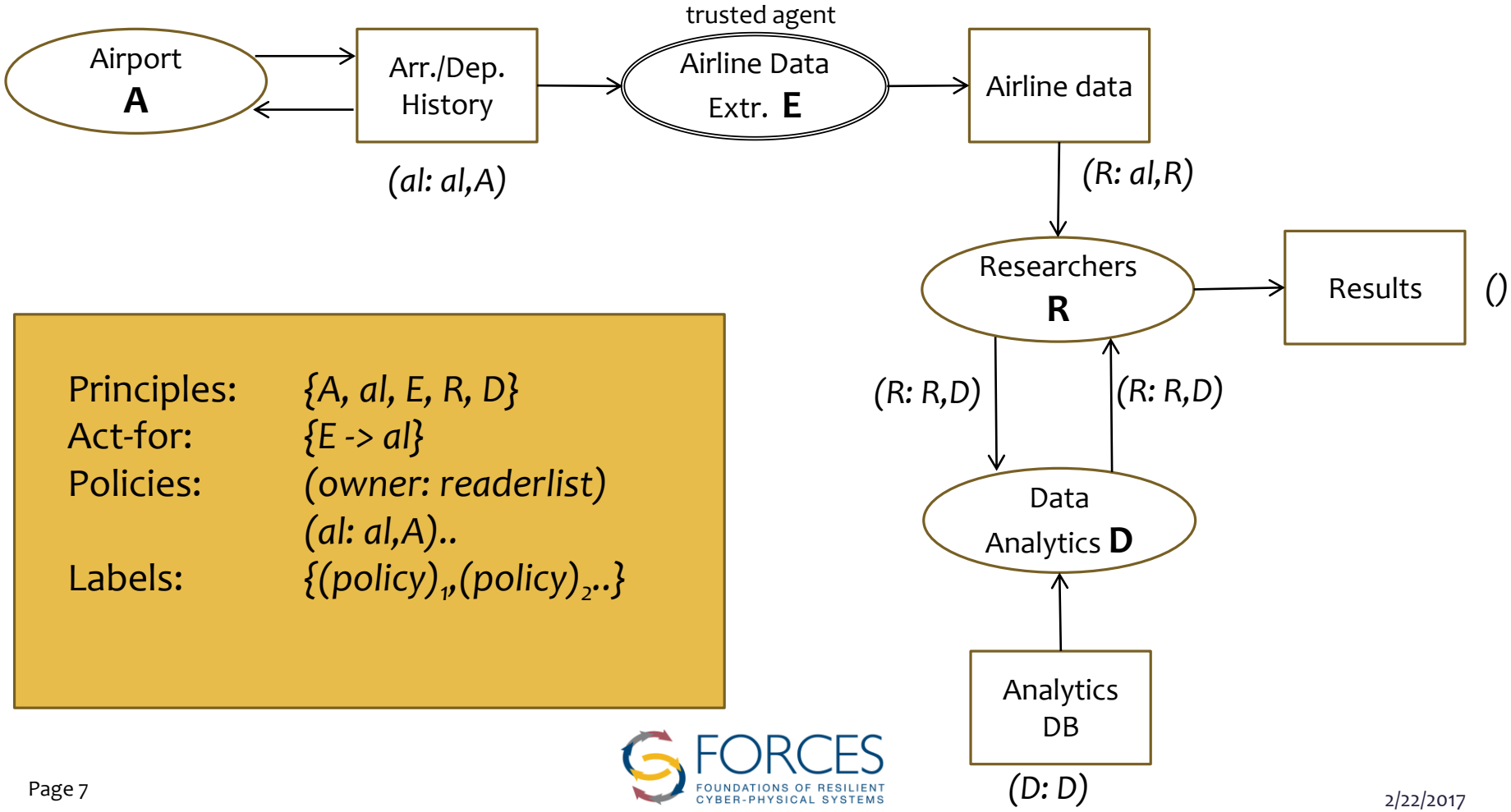
- How to model these restrictions?
- How to integrate these restrictions with others (functional and timing) and formulate a co-design problem?

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Decentralized Label Model for Informaiton Flow Control

* Myers, Liskov (1997): Introduced security-typed languages by labeling variables with information flow security policies
* New semantic concepts:
  - *Principles* that represent authority entities.
  - *Labels expressing* security classes encountered in most information flow models.
  - *Policies* that are elementary security primitives used in *labels.*
  - *Labeled entities* that have attached labels, such as *values, slots (variables, objects, i/o channels).* Copies of *values* can be relabeled, *slots* cannot.
  - *Operators* that can *relabel* or *declassify* values in information flows.
* DLM provides mechanism for static/dynamic type checking of security labels in information flows to detect policy violations.
* Example: *Jif,* a security-typed version of Java

Simple Example

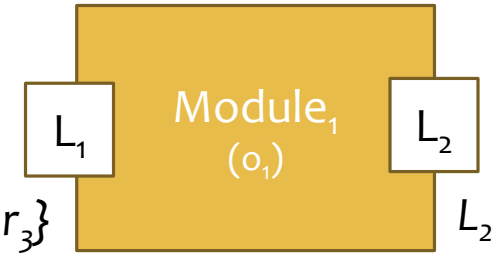


Principles: {A, al, E, R, D}
Act-for: {E -> al}
Policies: (owner: readerlist)
(al: al,A)..
Labels: $\{(policy)_1,(policy)_2..\}$

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Working With Security Labels

* *Labels* contain a set of *policies*. Each *policy* includes an *owner* and a set of *readers* allowed by the *owner*. The *effective reader set* for a *label* is the intersection of every *reader set* in it.
  $L = \{o_1: r_1, r_2; o_2: r_2, r_3\}$
* *Processing blocks* running under the authority of an *owner* can *declassify* the *owner's policy* by adding *readers*.

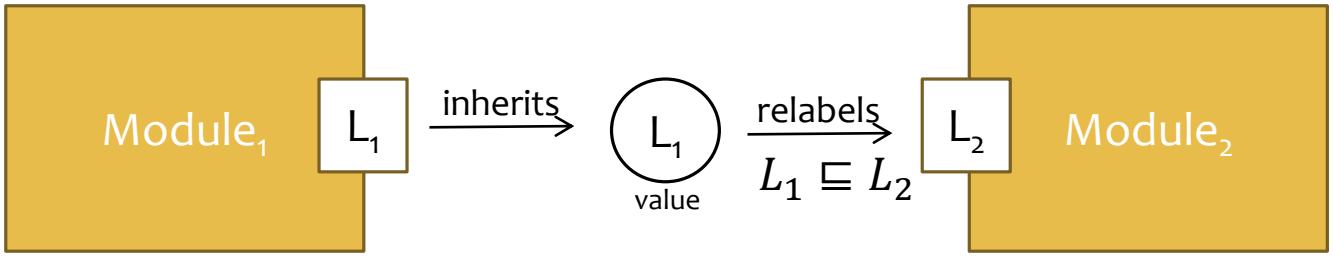


$L_1 = \{o_1: r_1, r_2; o_2: r_2, r_3\}$ $L_2 = \{o_1: r_1, r_2, r_3, ; o_2: r_2, r_3\}$

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Propagation Rules

* *Propagation rule-1:*

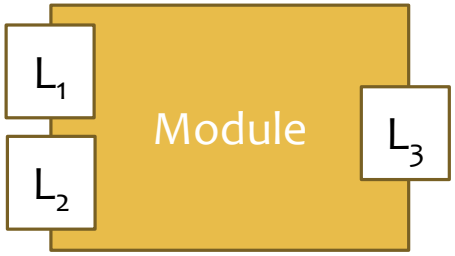


$owners(L_1) \subseteq owners(L_2)$

$\forall o \in owners(L_1), readers(L_1, o) \supseteq readers(L_2, o)$

(Labels form a security lattice.)

* *Propagation rule-2:*



L_3 is the *join* of L_1 and L_2

$L_3 = L_1 \sqcup L_2$

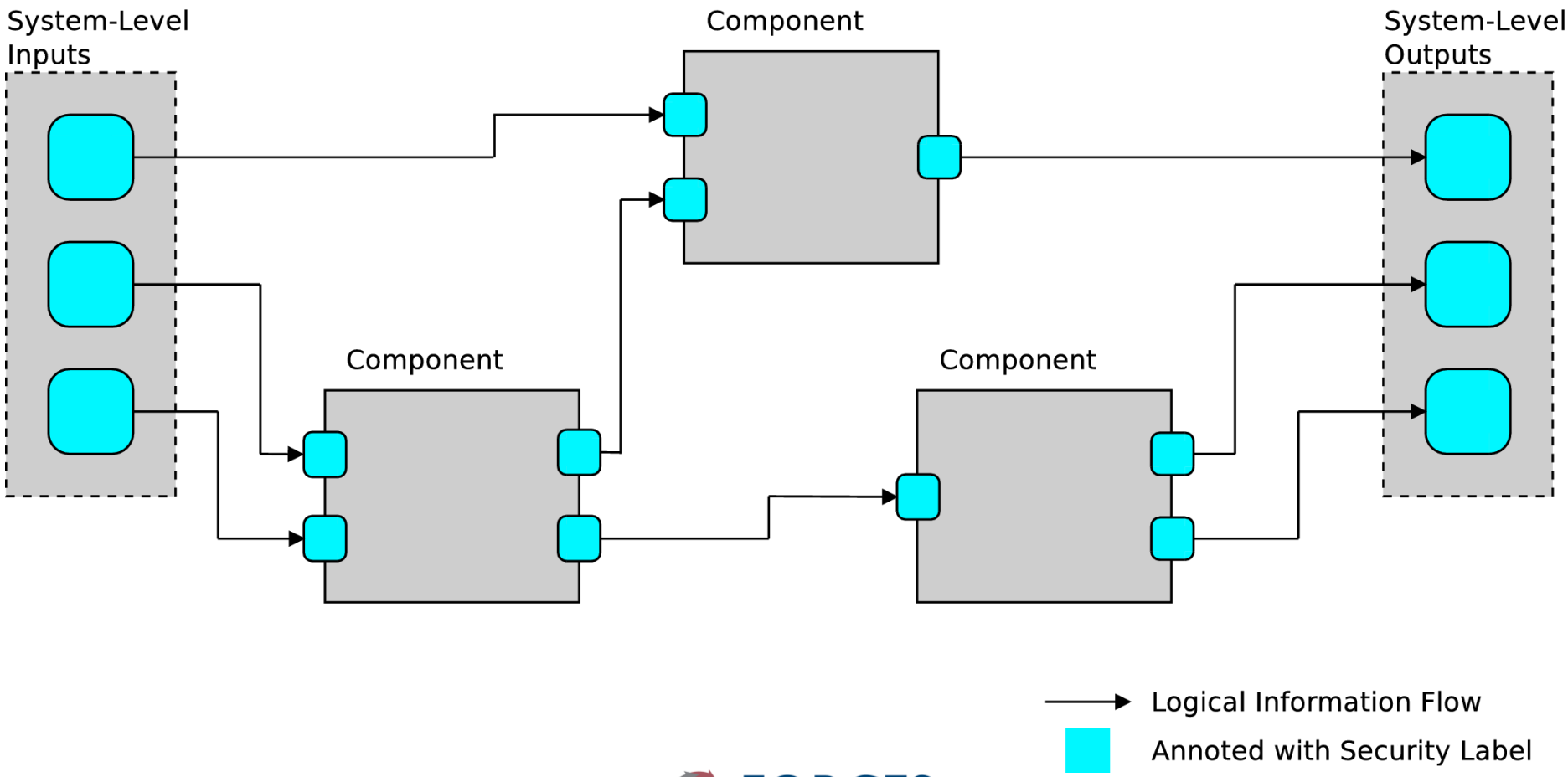
$owners(L_1 \sqcup L_2) = owners(L_1) \cup owners(L_2)$

$readers(L_1 \sqcup L_2, o) = readers(L_1, o) \cap readers(L_2, o)$

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

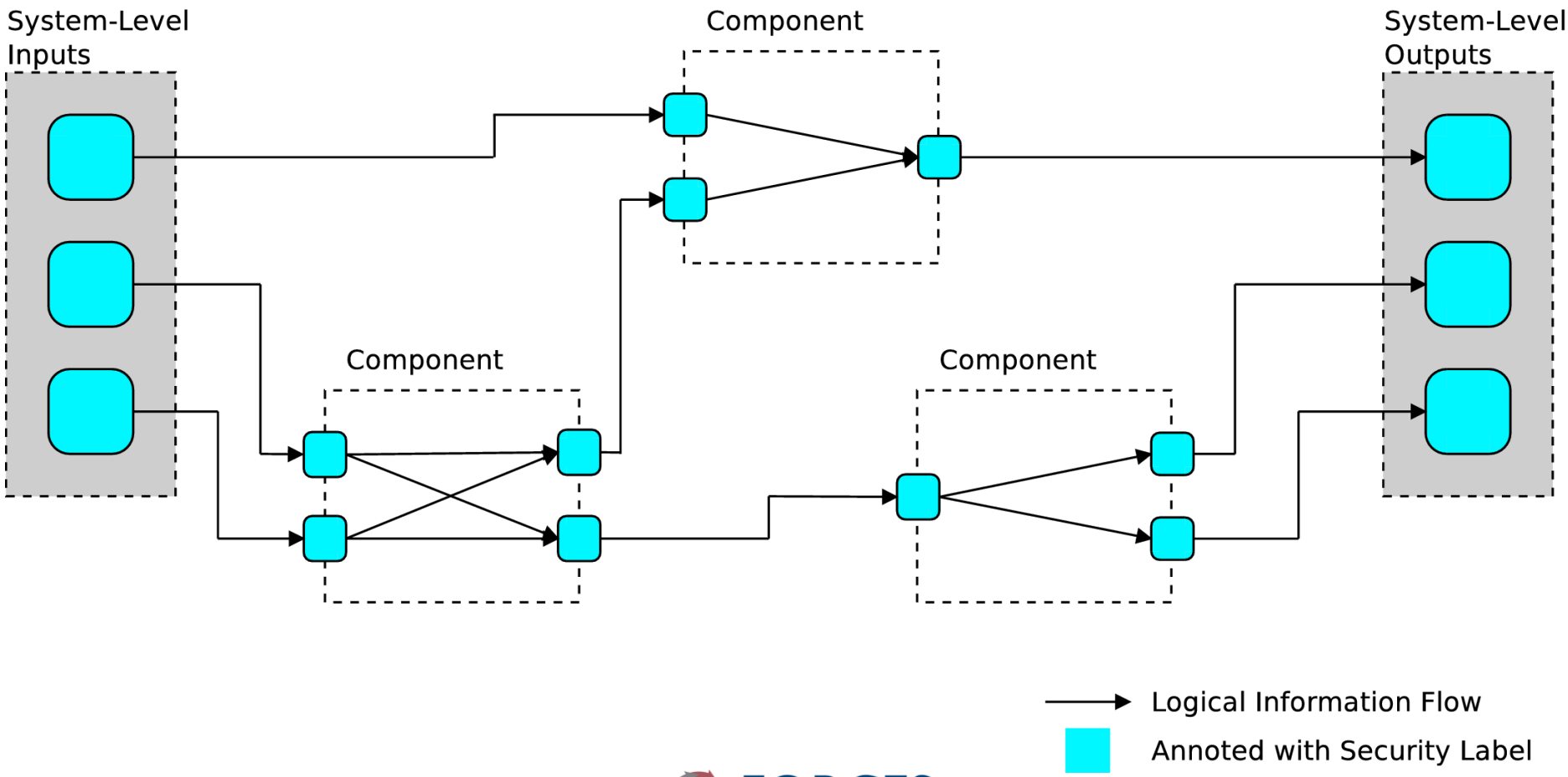
DLM in Model-Based Design

Information Flow Over SW Component Model



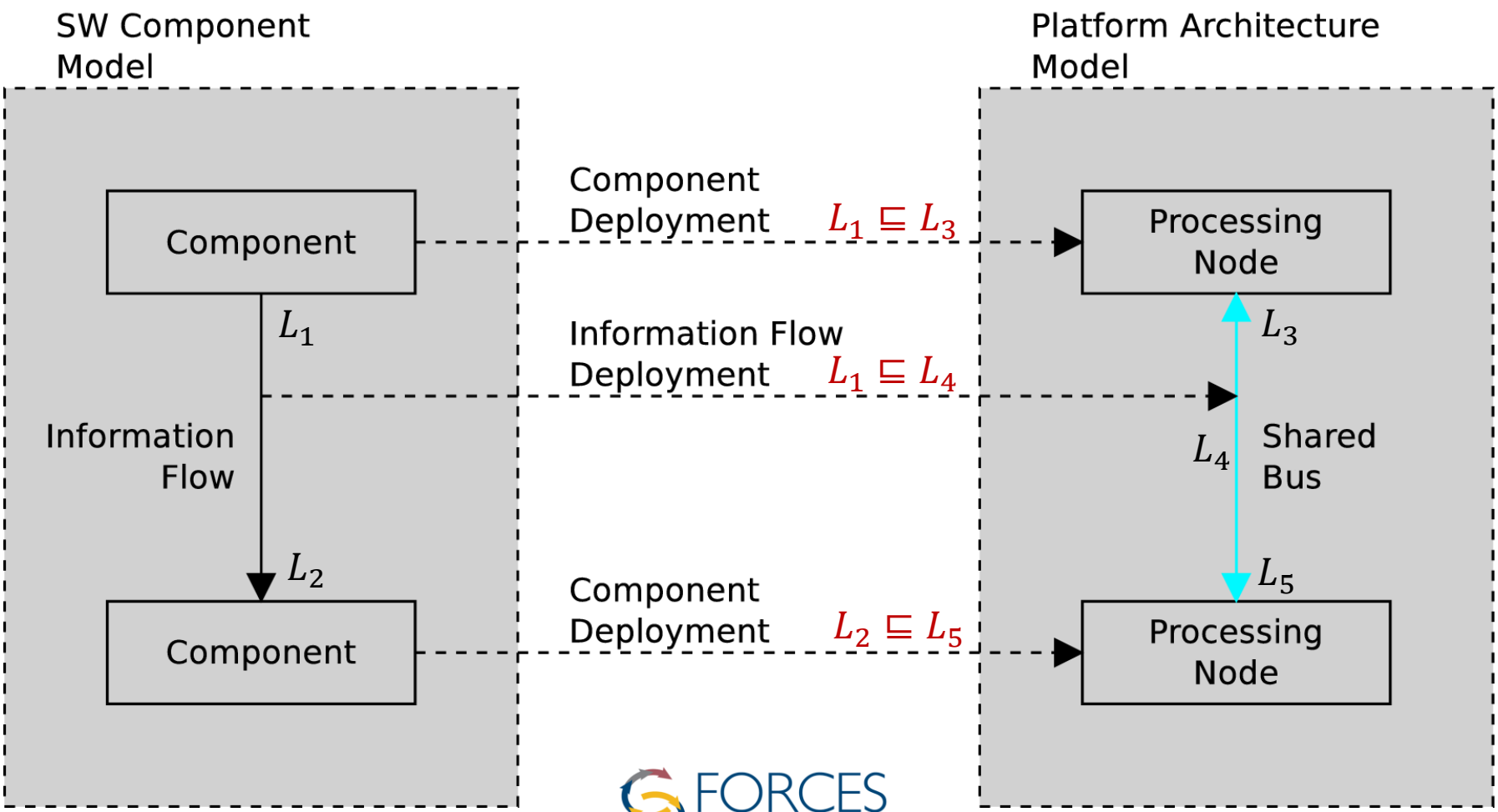
FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Information Flow Over SW Component Model



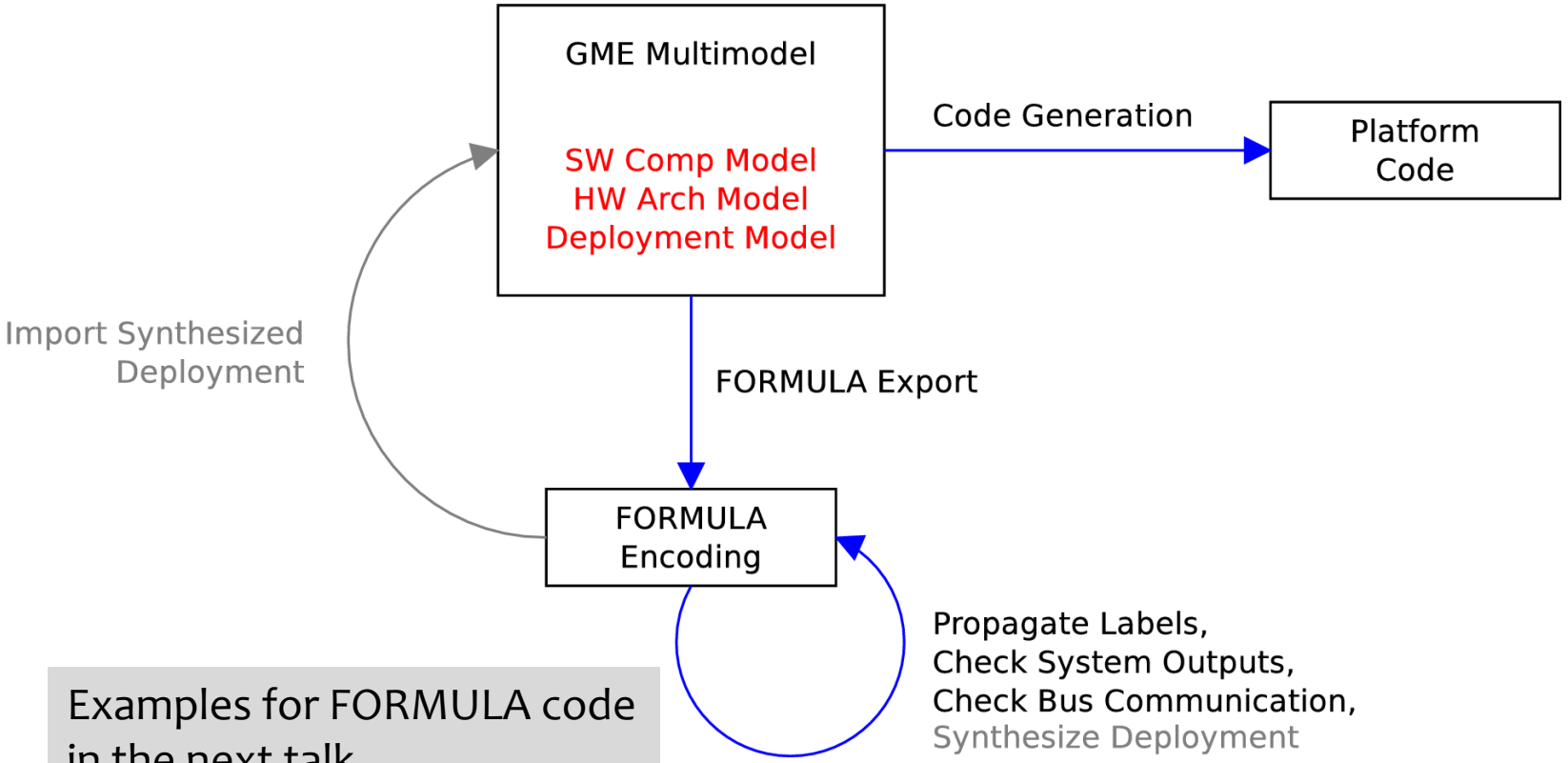
FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Information Flow Over Hardware Buses



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Workflow for Designing Secure Distributed Embedded Systems



Examples for FORMULA code in the next talk.

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS