Testing security of CPS

with Formal Methods

Application (in progress) to

industrial protocols

IoS & IoT

Roland Groz, Jean-Luc Richier, Maxime Puys, Laurent Mounier

Univ. Grenoble Alpes

LIG/Vasco + Vérimag/PACSS

Kobe – Université Grenoble-Alpes Workshop

Industrial systems & security

- Hot topic since Stuxnet (Iran 2009)
  - Even military-nuclear protected industrial sites can be damaged by cyberattacks
    - ~1000 centrifuges destroyed
  - Significant attacks (before and) after Stuxnet
    - E.g.: Ukraine black out (2015), German steel plant, Finland heating breakdown…
- Protection becoming a priority for government agencies (France: ANSSI, LPM 2013, OIV)

Testing for security (LIG)

- Goal: early detection of vulnerabilities (security flaws) in systems
- Approach: based on models, Model-Driven Engineering (MDE) and Model Based Testing (MBT), with Formal Methods
- Main techniques:
  - Model learning, reverse engineering
  - Model checking and analysis
  - Test generation, fuzzing

Past and current projects

SPaCIoS

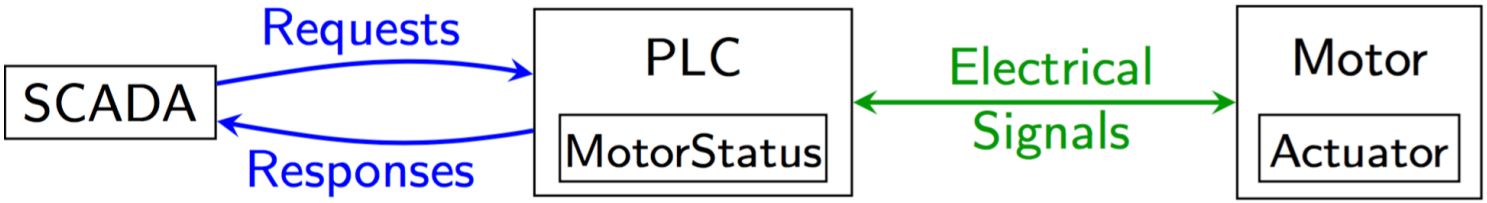
- European: Diamonds, SPaCIoS
  - + many national projects
- Application domains:
  - Internet of Services (web applications)
  - Communication protocols
  - Transport systems (automotive, aerospace, rail)
  - Industrial systems, CPS

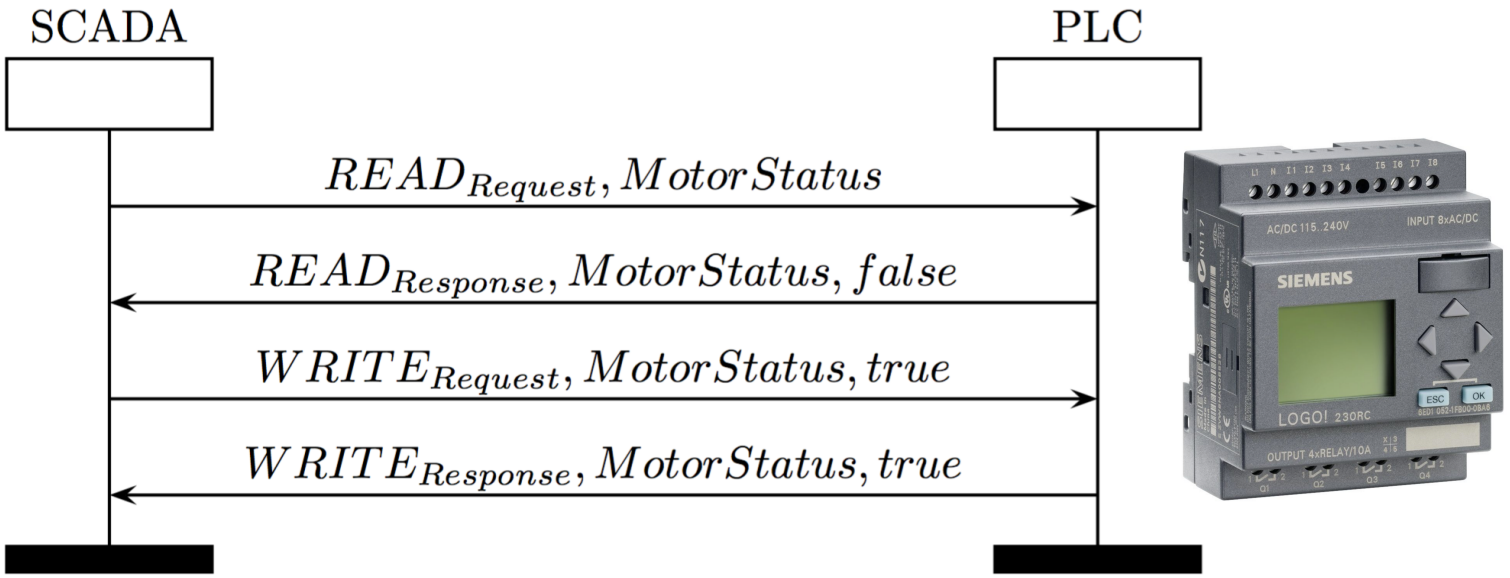
Industrial (&IoT) vs Business IT

- Security priorities differ from IT
  - IT: Confidentiality > Authentication > Integrity > Availability
  - (Indus) Availability > Integrity > Authent. > Confid.
  - IoT: domain dependent, e.g. Integrity > Avail. > Conf.
- Long lived, hard to patch, legacy (+Indus, -IoT)
- Proprietary protocol implementations (~IoT)
- Real-time
- Cyber Physical Systems (physical/vital hazards)

SCADA (Supervisory Control and Data Acquisition)

- SCADA controls variable Motor Status on PLC (Programmable Logic Controller)





Industrial Communication Protocols

- MODBUS (1979)
  - Mostly read/write PLC variables (+config…)
  - No security
- OPC-UA (2006)
  - Open PF Communications, Unified Arch.
  - Complex standard (978 pages)
  - Provisions for security
    - Signed or encrypted messages
    - OPC-UA SecureConversation (similar to TLS with handshake)

Current responses to threats

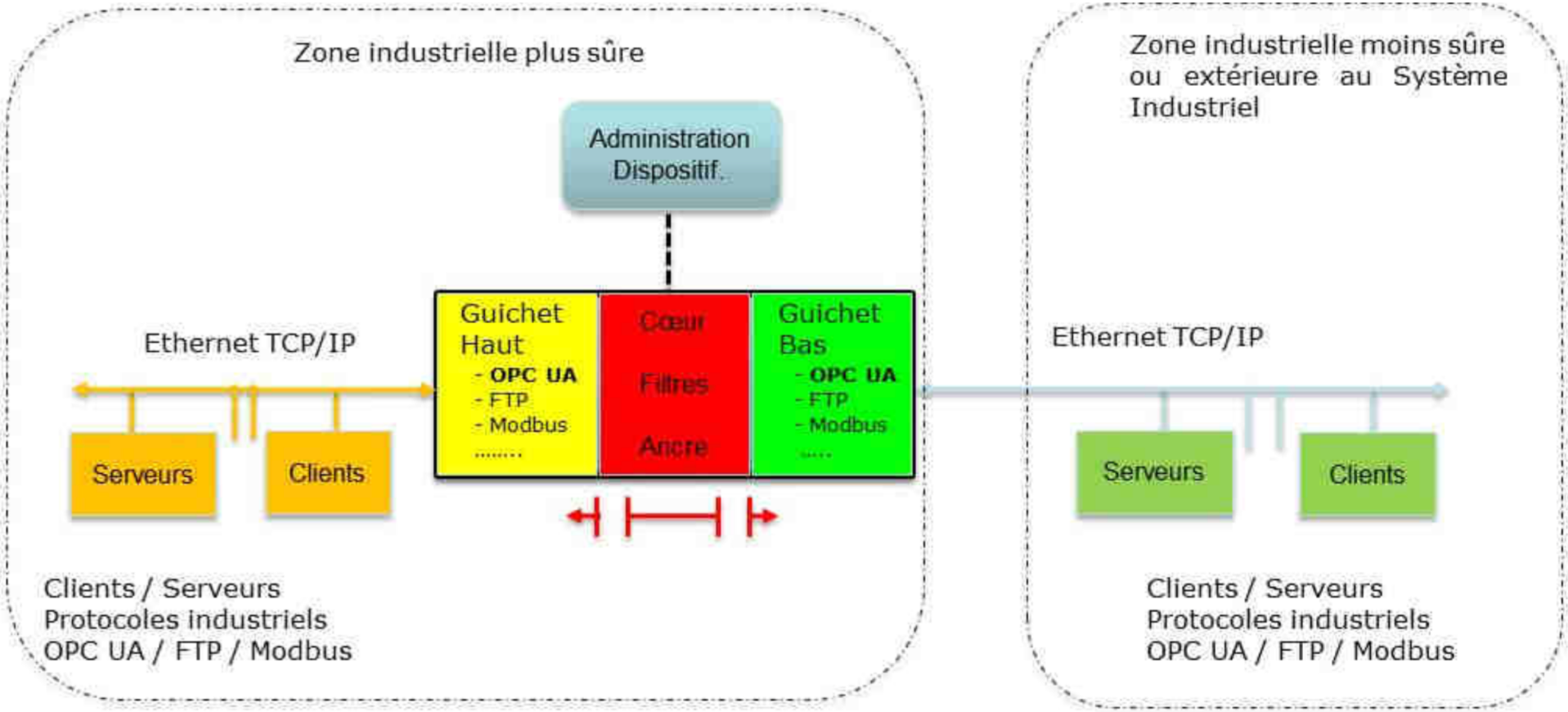
- Legal requirements on companies: risk analysis, human and technical measures
- Zoning: Data diodes, Firewalls for ICS protocols…
- Intrusion Detection Systems & IPS
  - Multiple systems (cf hierarchical distributed structure)
- Research: advanced IDS/IPS, vulnerability detection in protocol specs & implems

LIG security projects for SCADA

- ARAMIS (PIA) — **Zoning**
  - Isolation device (~firewall based on protocol rupture)
- SACADE (ASTRID) — **Attack prevention & detection**
  - SCADA platform for detecting and playing attacks
- SRED (PIAVE) — **On-line protection**
  - Intrusion detection for electric distribution

ARAMIS security gateway

- Deep Packet Inspection
- Rewriting packet contents, with protocol-specific rules
- Physically separated processors



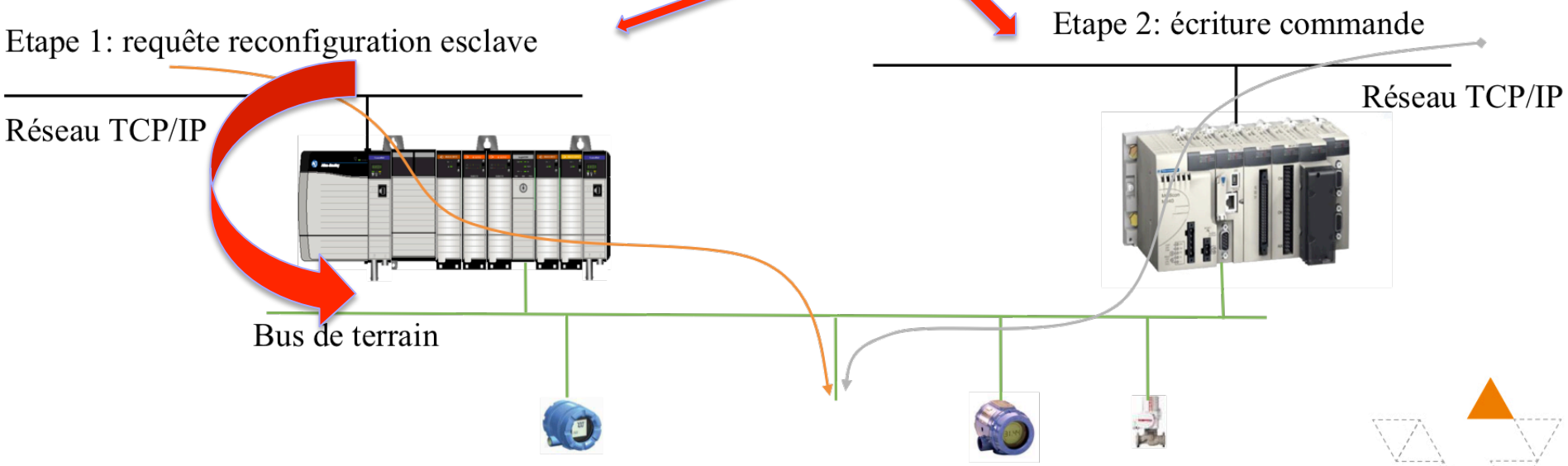
SACADE

- Started 2017
- Investigating Attack scenarios against PLC
- Special focus on scenarios combining Distribution and Encapsulation:
  - Payload recombined from multiple encapsulated sources

Focus: encapsulation+distrib

Encapsulation:
Payload is legal at all levels of protocols, so espaces protocol filtering

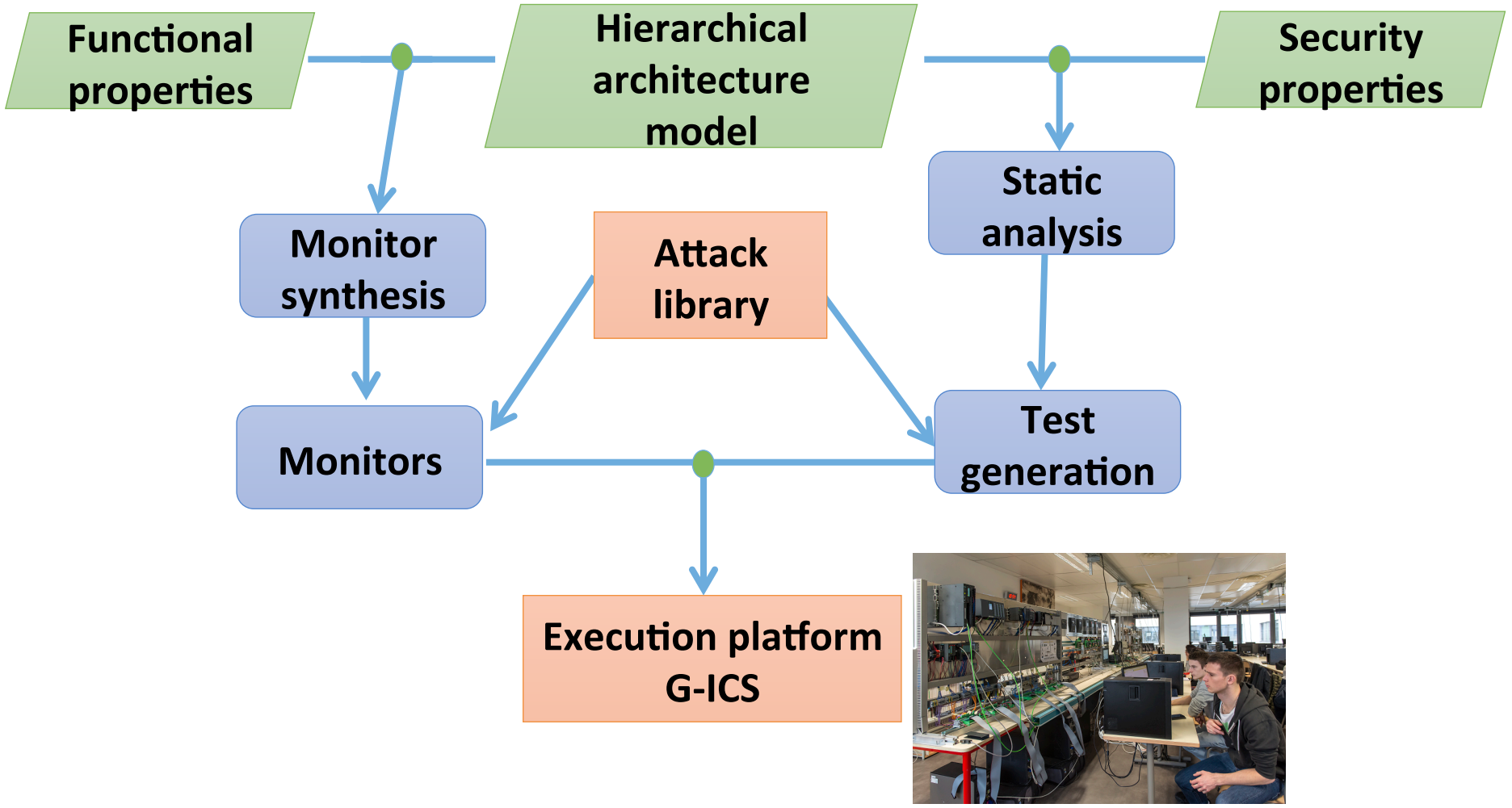
Distribution: Noxious behaviour is obtained by combining commands from several legal commands from distributed sources



Examples of attack elements

- Playing on protocol levels
  - Data injection to move towards dangerous states
  - Device reconfiguration
  - Combining reconfiguration followed by injection
- Timing dependence
  - Commands sent in transient states of CPS

Experimental Platform (Grenoble)



Model Based analysis to detect vulnerabilities in protocol implem.

- Previous work for vulnerability detection

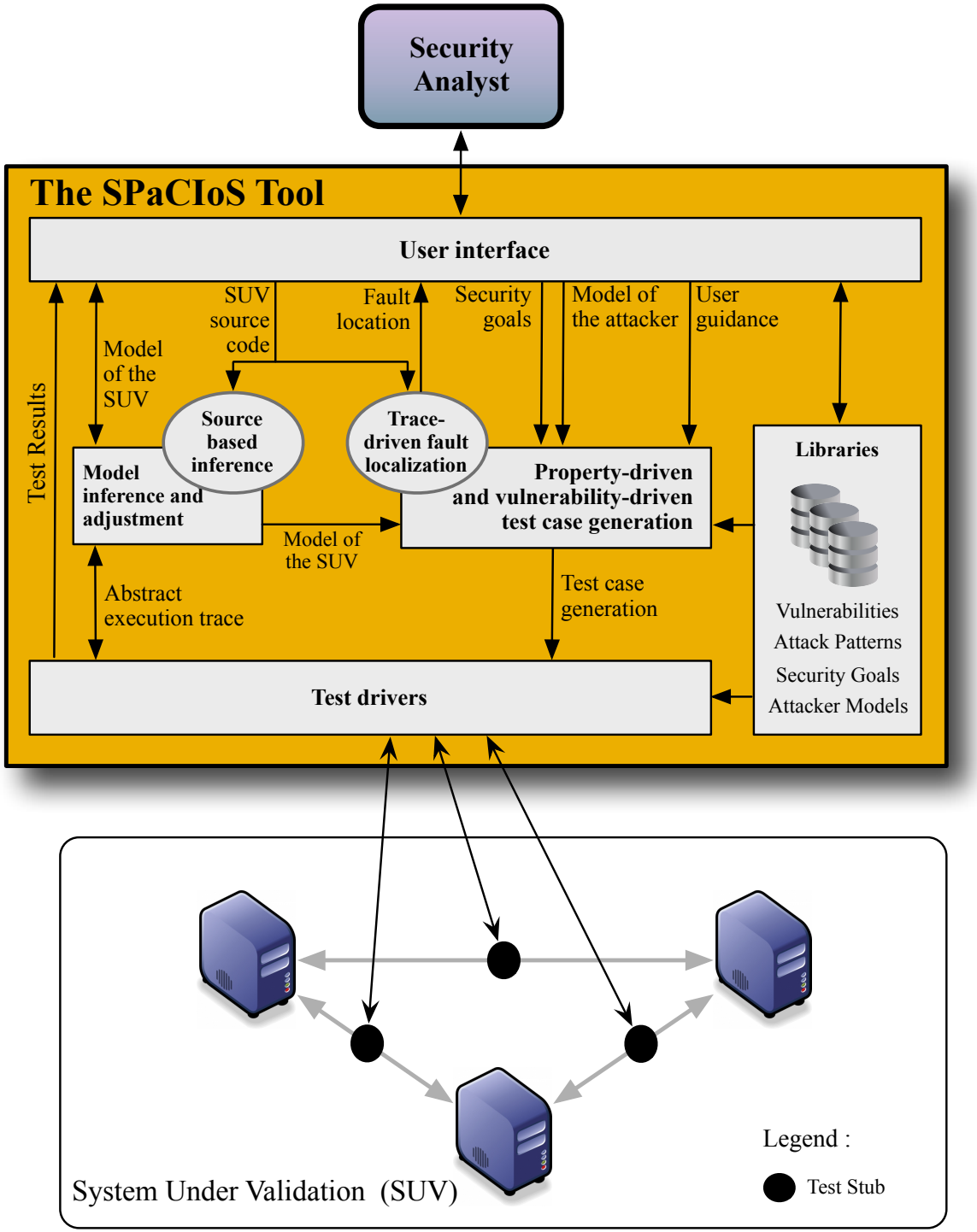
SPaCIoS

  - SPaCIoS: tool box for vulnerability in IoS (Internet of Services: Web applications),
    - Based on model of cryptographic protocols + Model-checking, model based testing, model inference…
  - Kameleon Fuzz: smart fuzzing

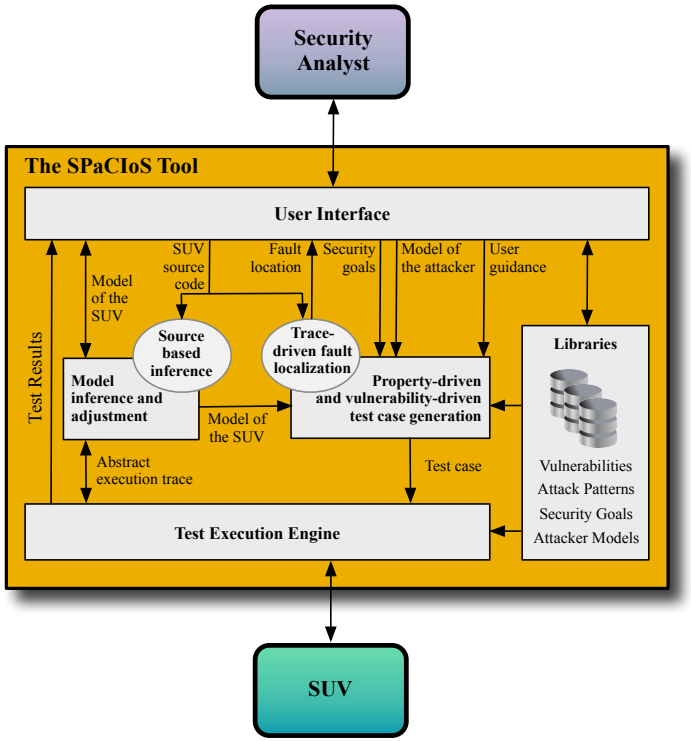
    - Based on protocol model and grammar

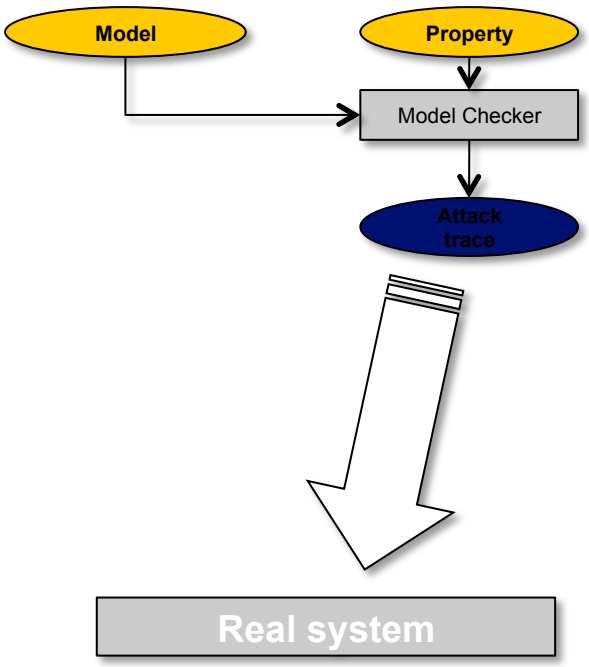
SPaCIoS tool

- Modelling with
  - ASLan++
- Models can be retrieved
  - From source code (jModex)
  - Black box testing (SIMPA)



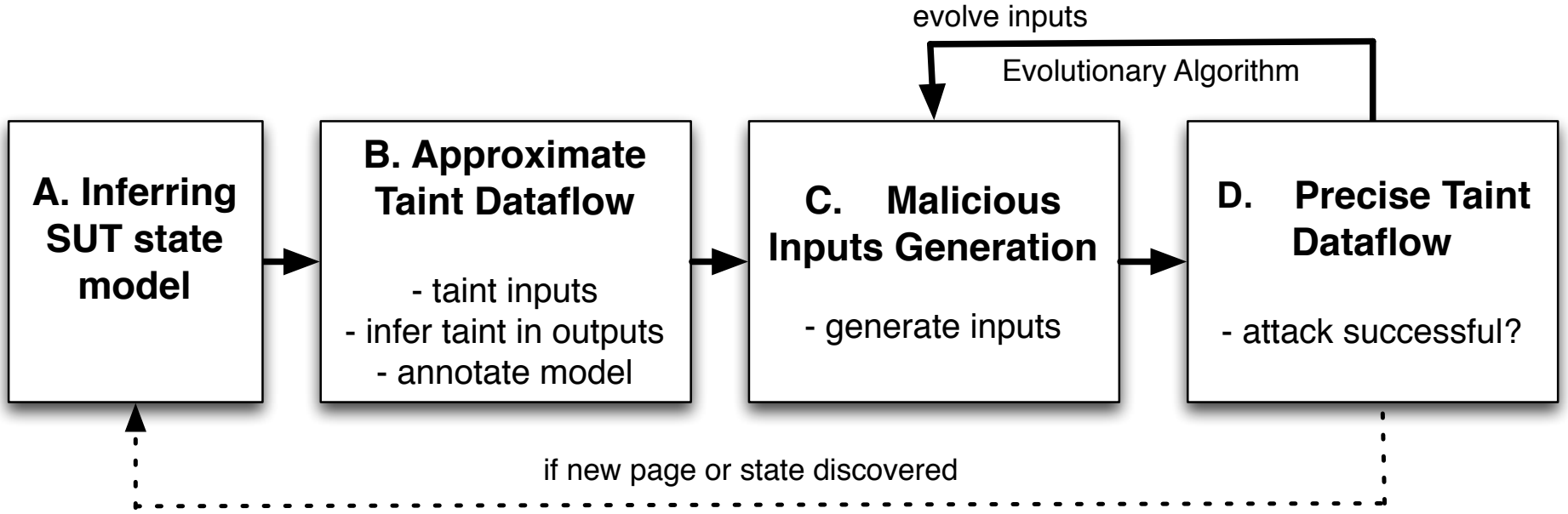
	Objectives	Results
WP 3	▪ Validation techniques • model checking • property-driven testing • vulnerability-driven testing • bridge components	✓ Working prototypes of (and more) • Model inference **SIMPA** • Model extraction **jModex** • Mutation-based testing **SPaCiTE** • Instrumentation-based testing **IBT** • LTL separation for testing **Fred** • Low level attacker models **Vera** • Fuzzing **KameleonFuzz & SVCov**
WP 4	▪ SPaCIoS Tool ▪ Validation methodology patterns	✓ **SPaCIoS Tool** released ✓ Valid. method. patterns & **Tutorials**



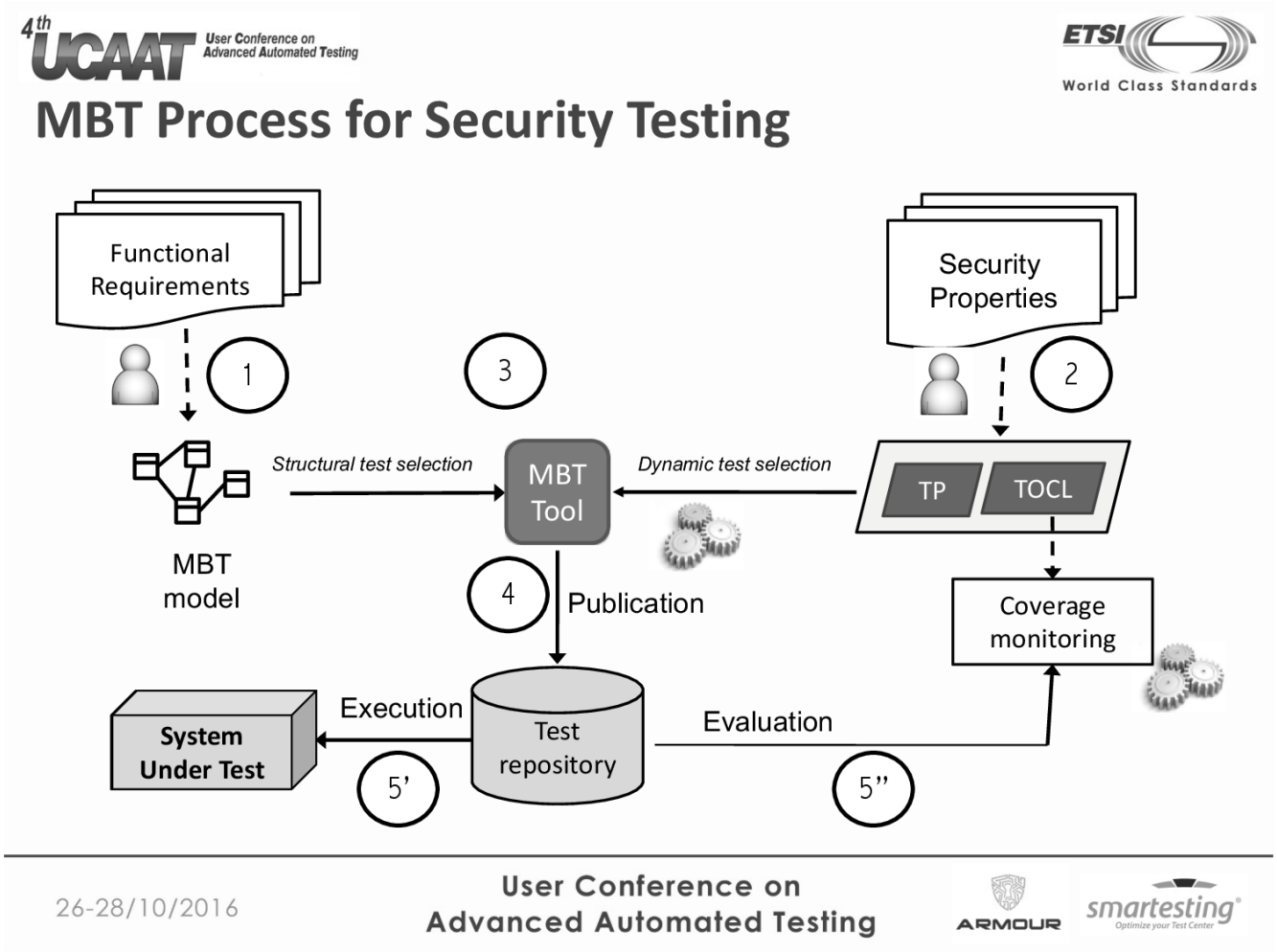


SPaCIoS

KameleonFuzz overview



Other approach: test patterns



Back to SCADA & CPS security

- Weak protocols => easily found attacks
- Difficulties lie in concretization
  - Bypassing security architecture
    - firewall through encapsulation & multiple interface
    - IDS through non monitored channels
  - Dealing with proprietary undocumented protocols and features
- Methods above might be too sophisticated ?

THANK YOU
FOR YOUR ATTENTION.

Contact:
Roland.Groz@univ-grenoble-alpes.fr
Professor at Grenoble INP Ensimag

BACKUP SLIDES

Architecture of Industrial Control Systems

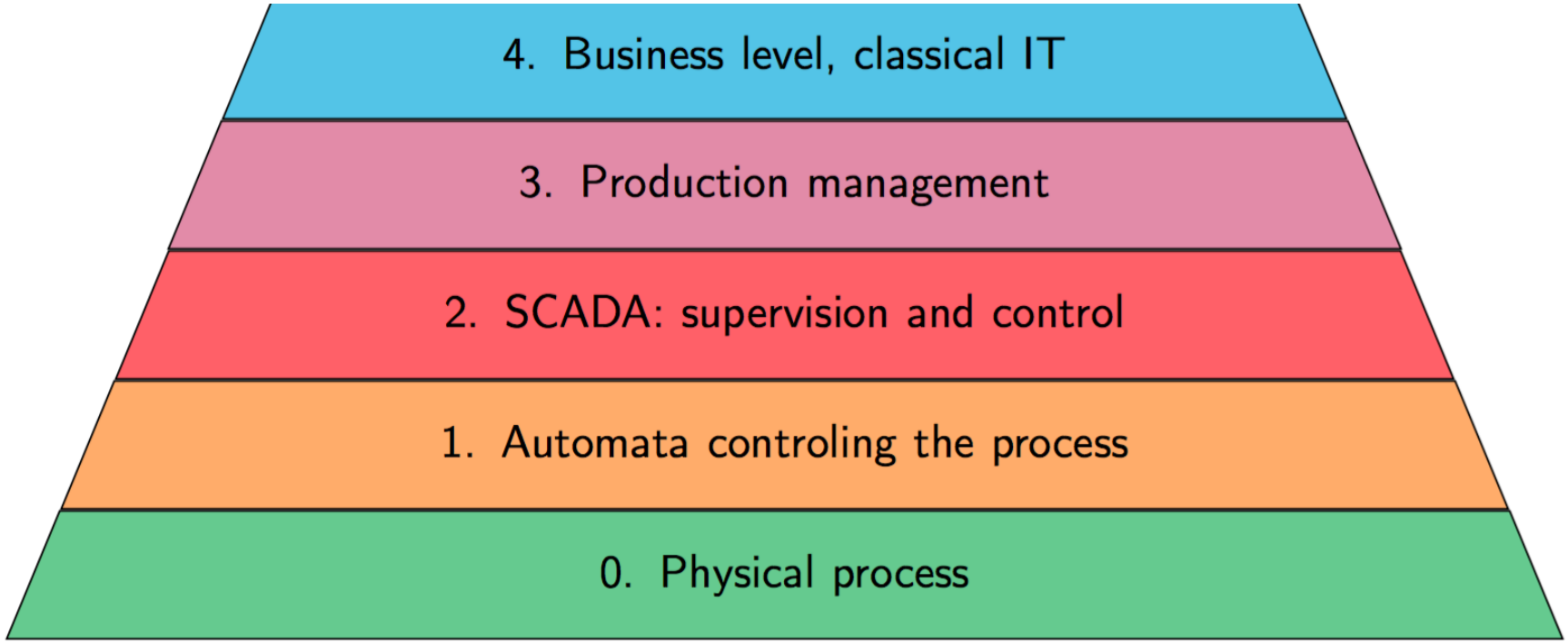
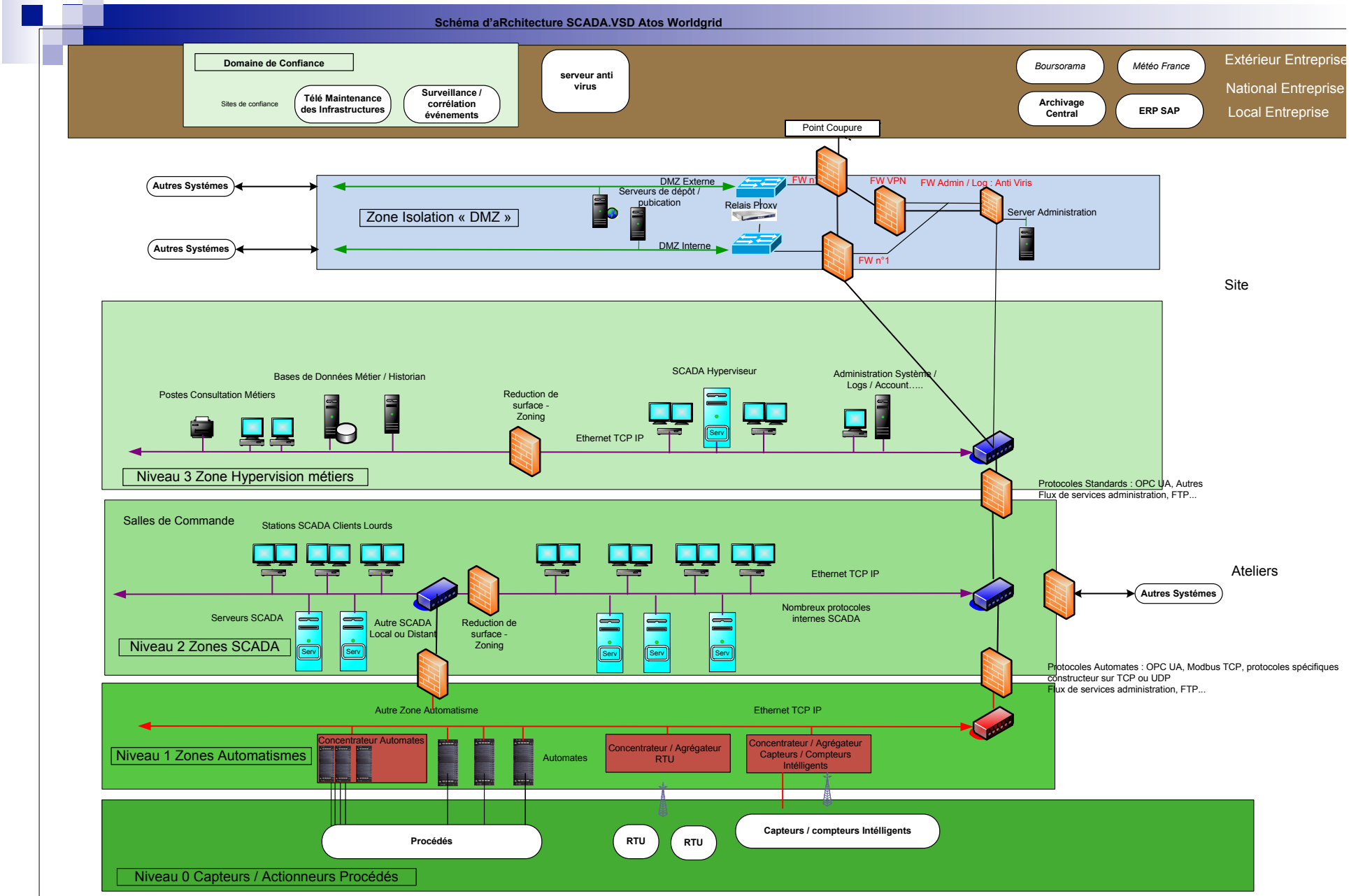


Figure : Purdue model [Wil91]

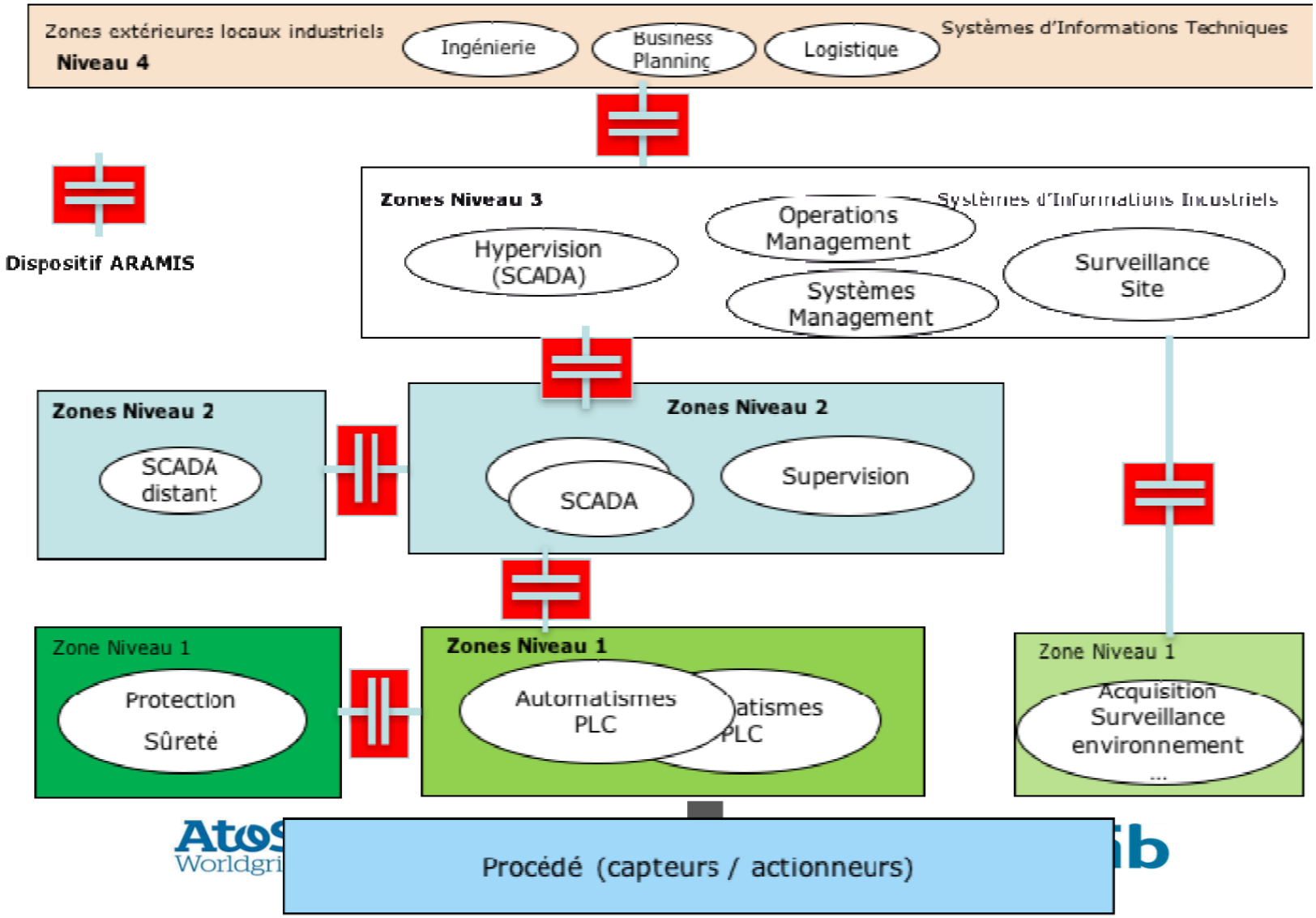


Légende
RTU = Remote Transmission Unit

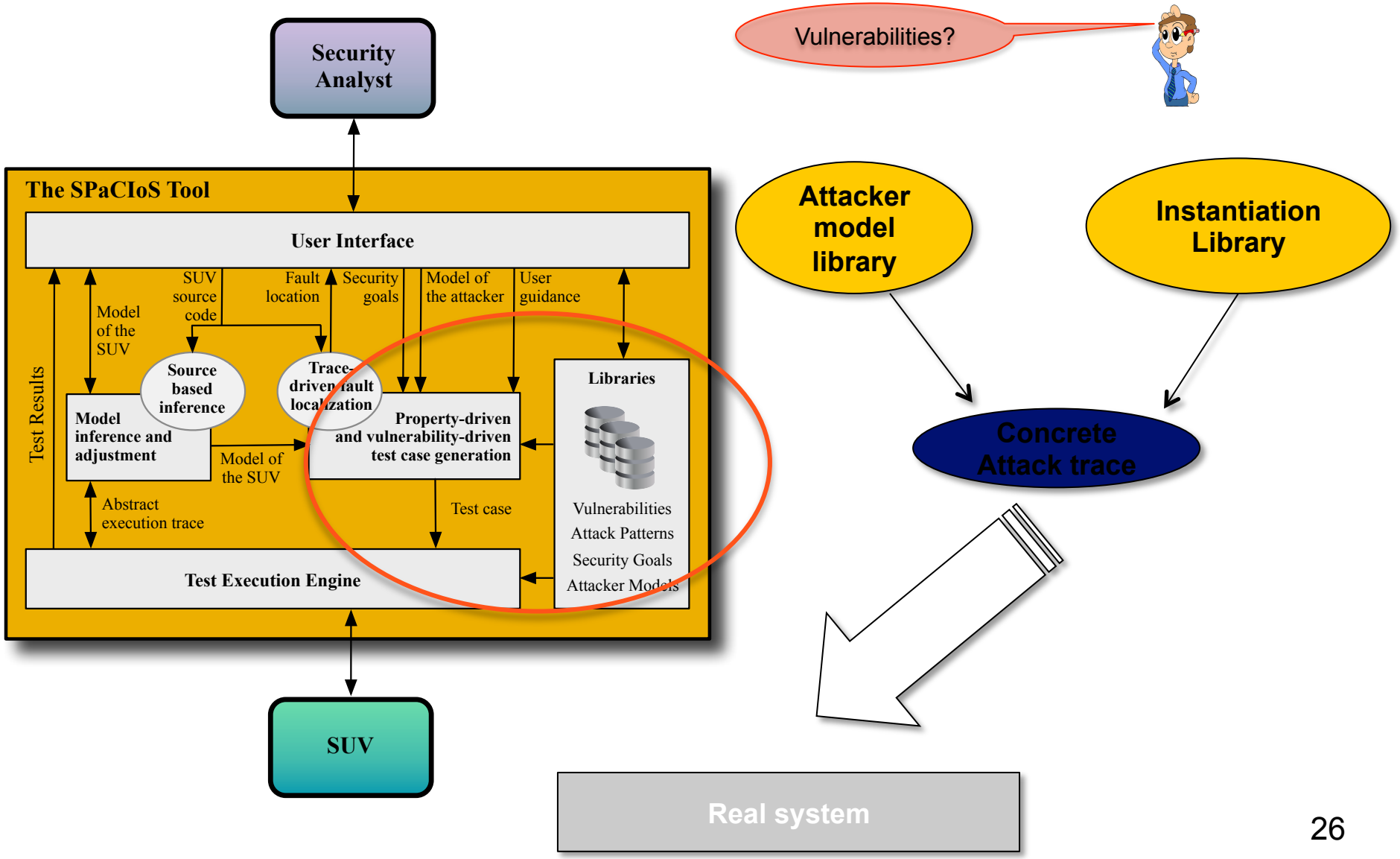
Equipements Réseaux Switchs etc

Firewall IPS/IDS – Data Diode

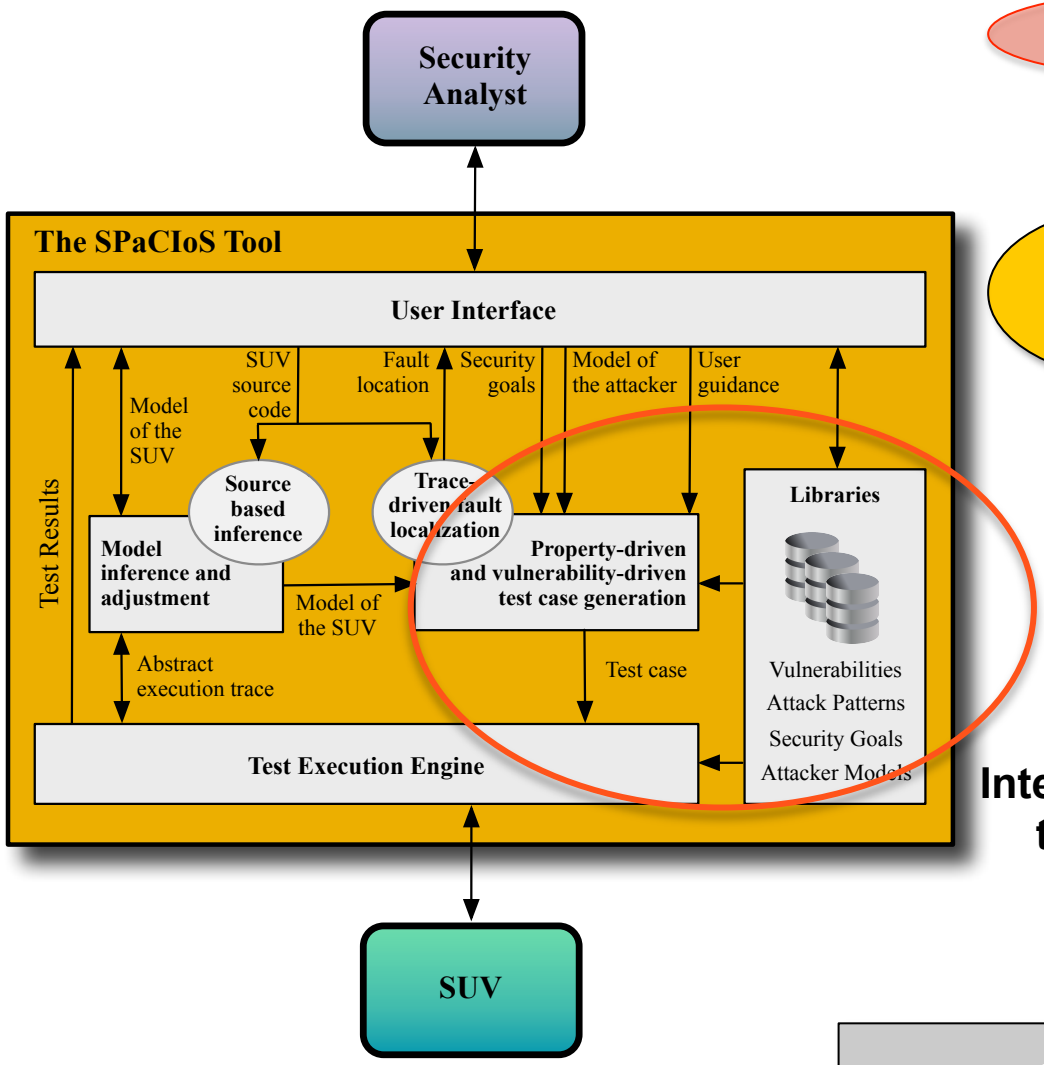
ARAMIS

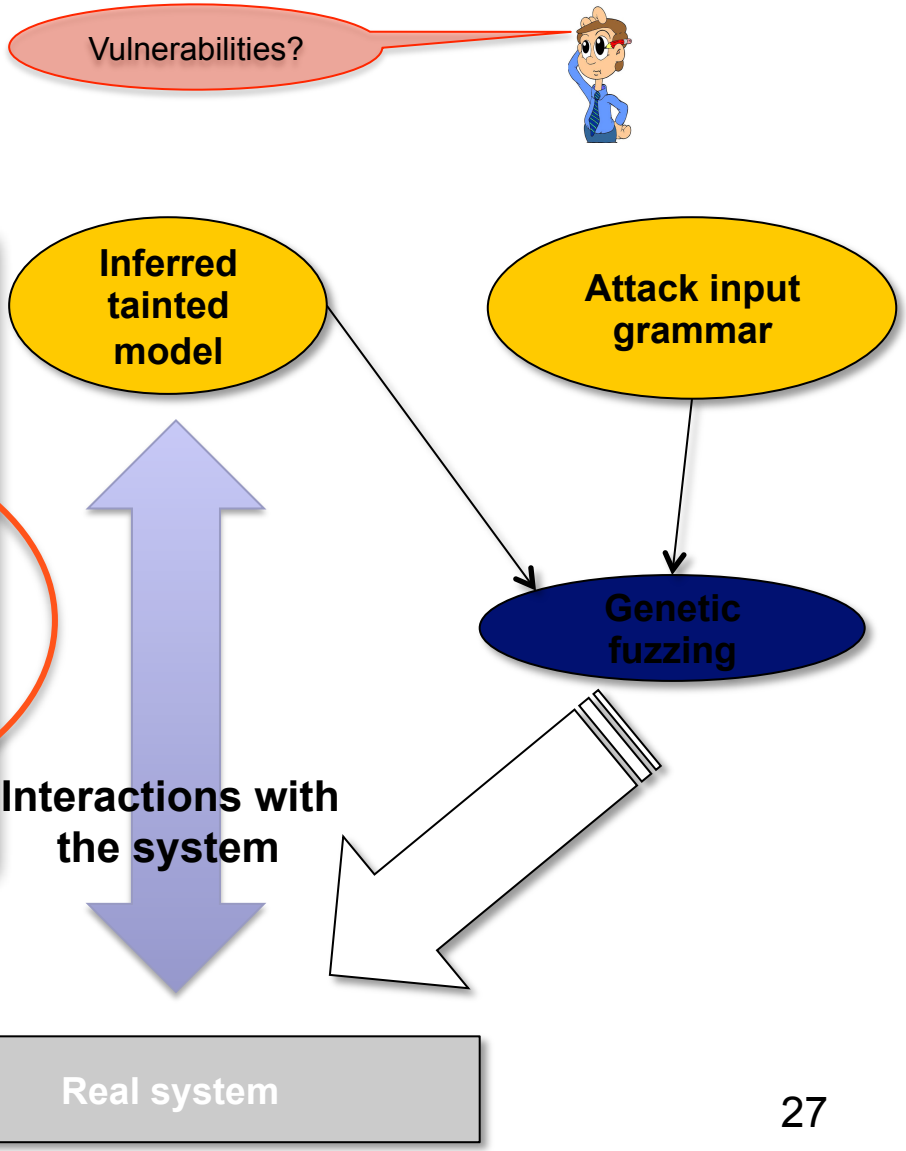

Zones extérieures locaux industriels
Niveau 4
Ingénierie
Business Planning
Logistique
Systèmes d'Informations Techniques
Dispositif ARAMIS
Zones Niveau 3
Hypervision (SCADA)
Operations Management
Systèmes Management
Surveillance Site
Systèmes d'Informations Industriels
Zones Niveau 2
SCADA distant
Zones Niveau 2
SCADA
Supervision
Zone Niveau 1
Protection
Sûreté
Zones Niveau 1
Automatismes PLC
Zone Niveau 1
Acquisition
Surveillance
environnement
...
Procédé (capteurs / actionneurs)

Low-level attacker models: Vera

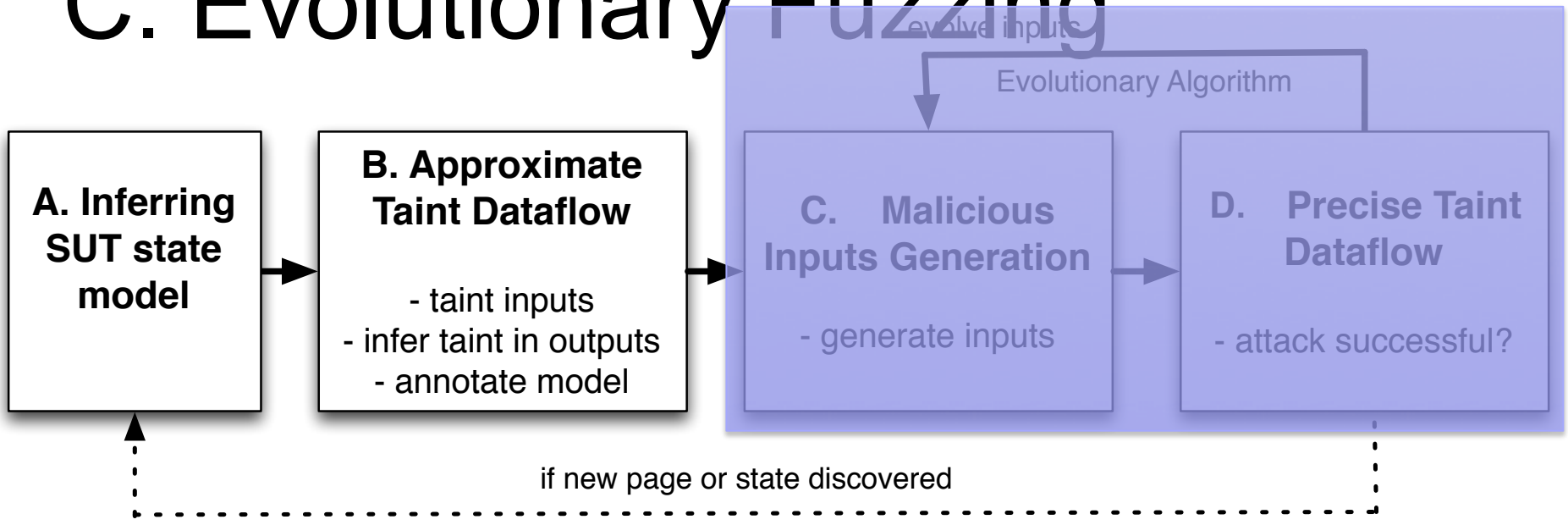


KameleonFuzz



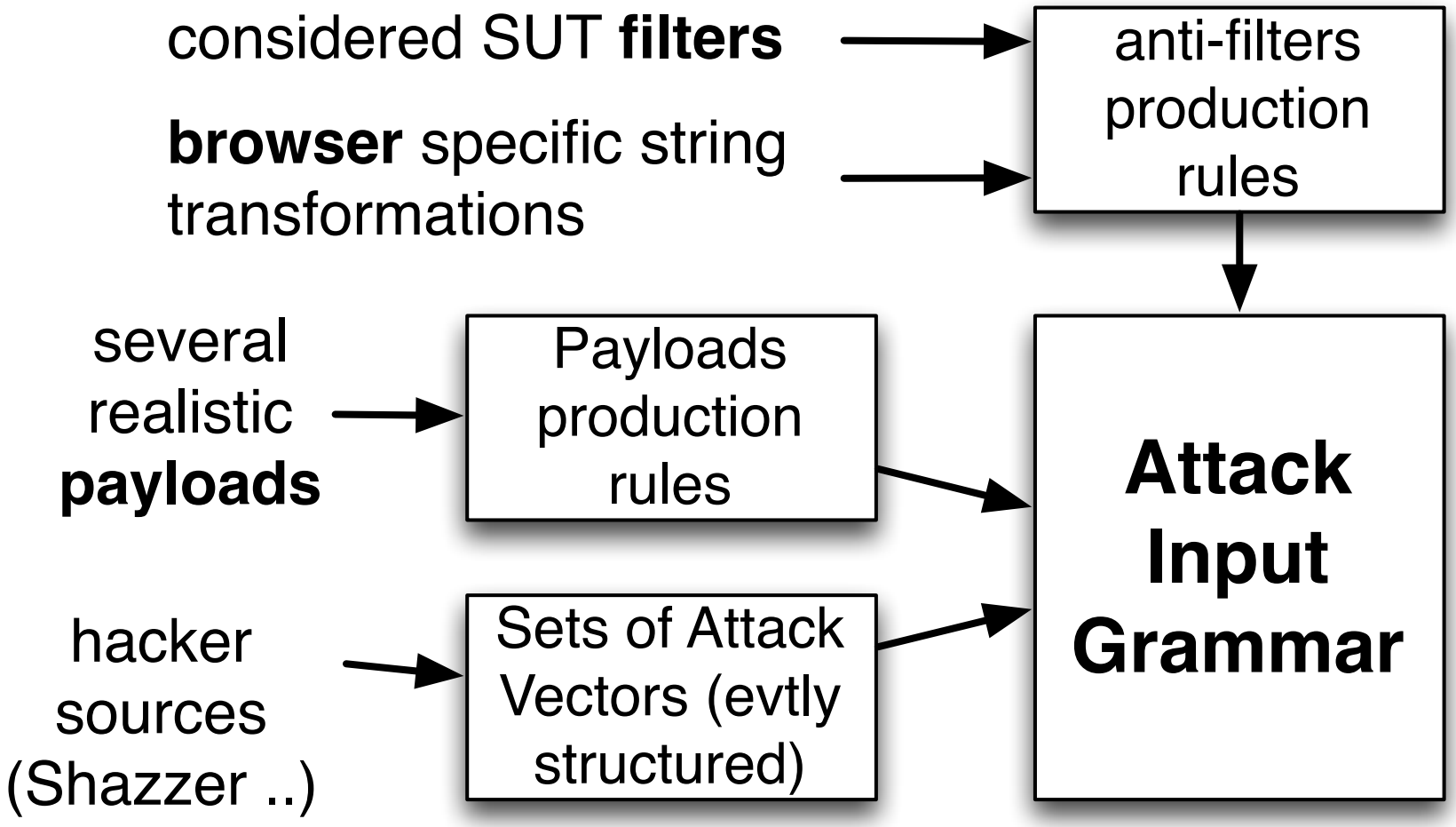


C. Evolutionary Fuzzing



- Attack Input Grammar
- Mutation & Crossover
- Fitness & Test Verdict

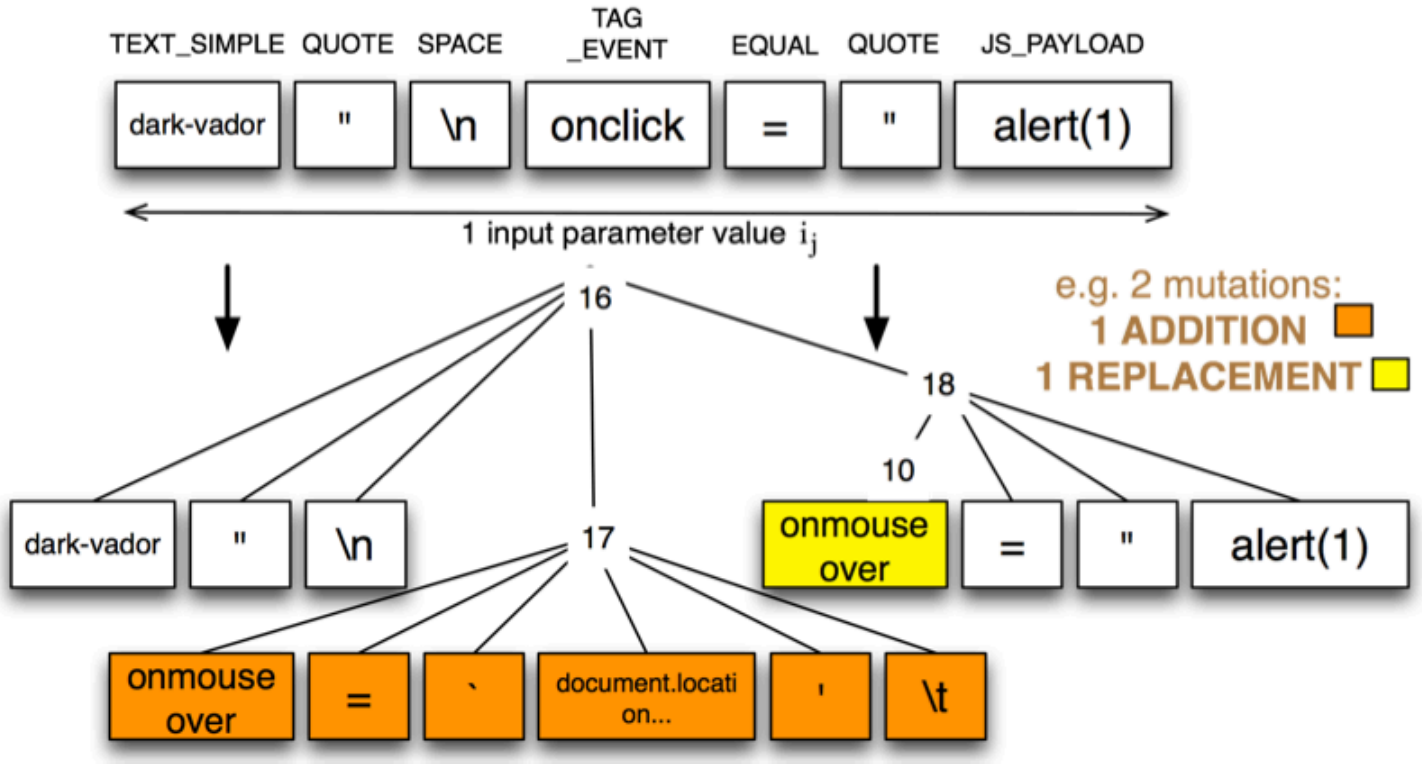
Attack Input Grammar



Mutation (at input param. value)

w.r.t. Attack Input Grammar G_{AI} add, delete, replace elements:

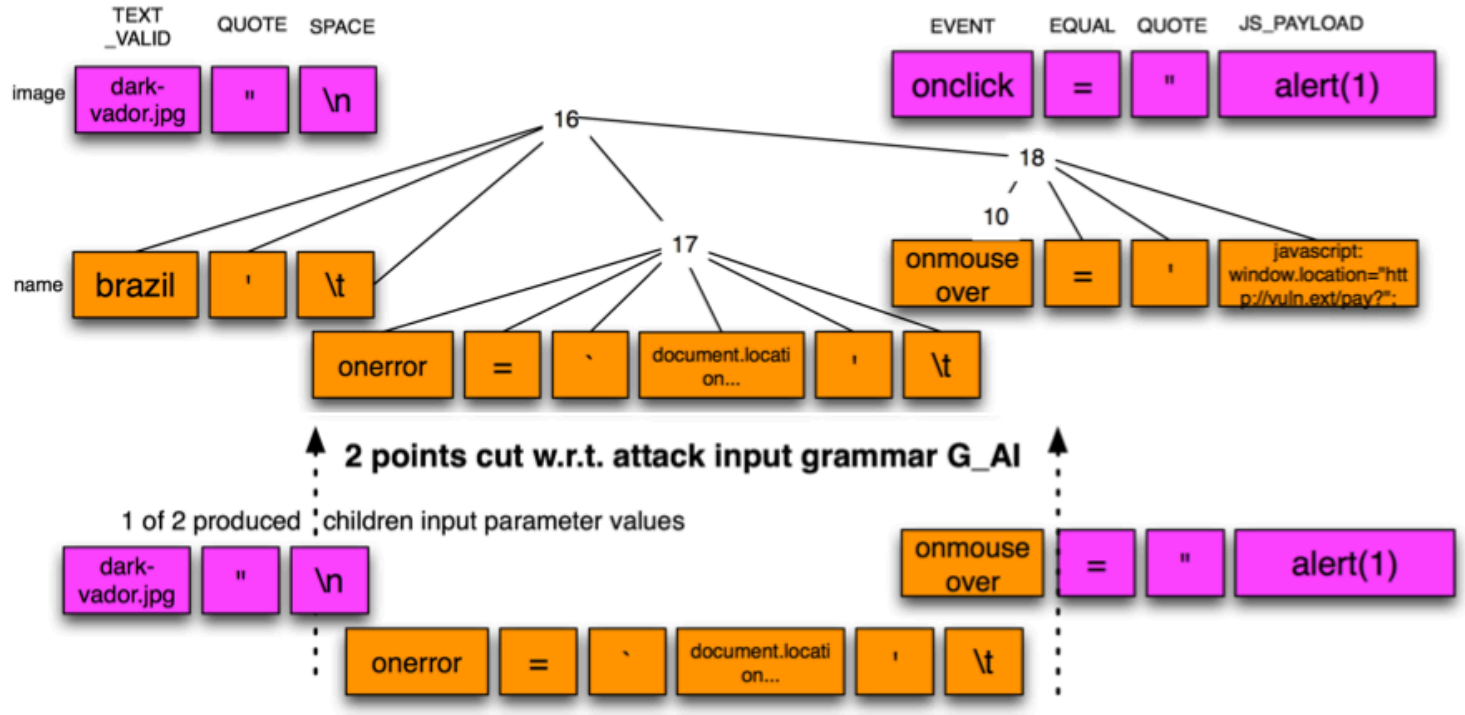
10: html_tag_event = "onabort" | ... | "onclick" | ... | "onwaiting"
16: xss_in_attribute = text_simple quote space xss_at_least_1_attribute html_attribute_before_end_quote
17: xss_at_least_1_attribute = [0:15] (html_attribute_before_end_quote quote text_simple space)
18: html_attribute_before_end_quote = tag_event "=" quote js_payload



Crossover (at input param. value)

Let $I = (prefixSeqI, (image =$ `dark-vador.jpg" \n onclick="alert(1)`$))$
and $J = (prefixSeqJ, (name =$ `brazil'\t ...onmousover='javascript:...`$))$

- **input parameter level**:



Other approach: test patterns

- Developed by FEMTO-ST & Smartesting (Besançon)
- Based on CertifyIt tool
- « High level » test patterns based on
  - Expert knowledge
  - Known high level vulnerability classes
  - Tool uses TP to guide symbolic exploration of model
  - Unfolding of patterns based on constraint solving

Language for security properties

- TOCL: Temporal OCL (Dwyer99, patterns)
- TOCL property = Pattern+Scope
- Pattern: event ordering
  - (always, never, eventually k times, precedes, follows)
- Scope: restricts pattern to interval
  - (globally, between, after, before)