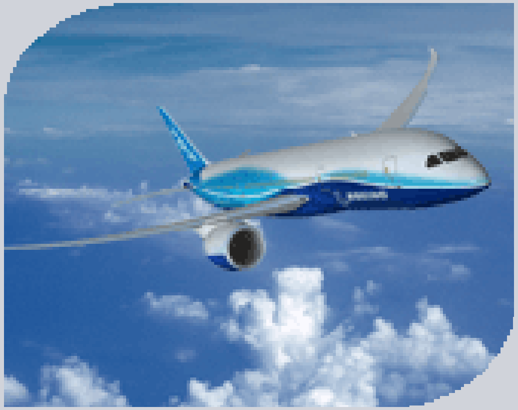
Corporate Technology

Formal security analysis and certification in industry, at the example of an AADS[1]



Dr. David von Oheimb
Siemens Corporate Technology, Security

Guest lecture on invitation by Dr. Ricarda Weber at
CS department of TU Munich, Germany, 04 June 2009

http://www11.in.tum.de/Veranstaltungen/SecurityEngineering2009/

[1]**A**irplane **A**ssets **D**istribution **S**ystem

Overview

- IT Security at Siemens CT
- Software Distribution Systems
- Common Criteria certification
- Formal Security Analysis
- Alice-Bob protocol model
- Validation with AVISPA Tool
- Conclusion

Siemens Corporate Technology: About 1,800 Researchers and Developers Worldwide …

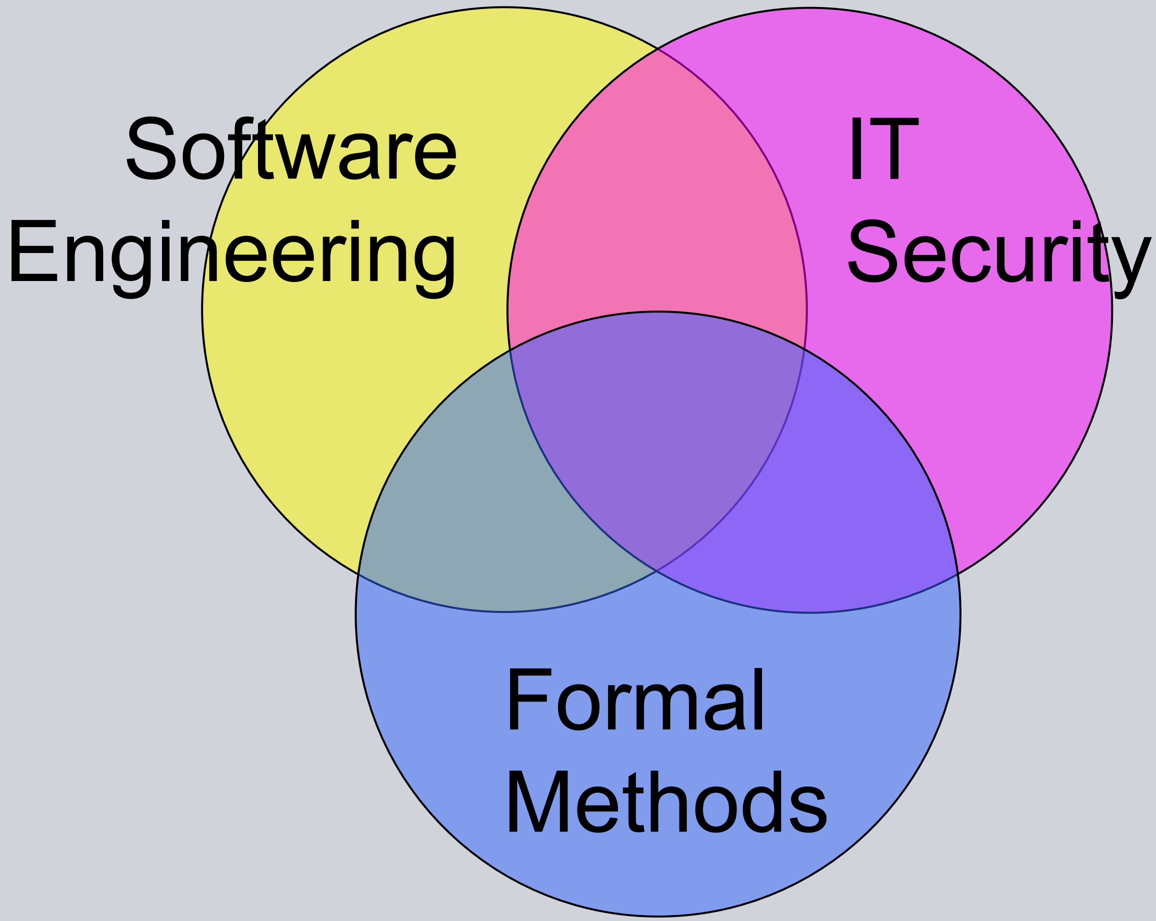
SIEMENS

SIEMENS

Security Applications & Methods

- Secure Operating Systems, Trusted Platform Modules (TPM)
- General Purpose Security Mechanisms, like:
  - Role / Policy Based Access Control (RBAC)
  - Public Key Infrastructure (PKI),
  - Single Sign-On (SSO)
- Security of Service Oriented Architecture (SOA): Web Services etc.
- Application-level security: e-health, e-government, e-Commerce
- Enterprise Rights Management (ERM)
- Formal Methods and Certification

Fields

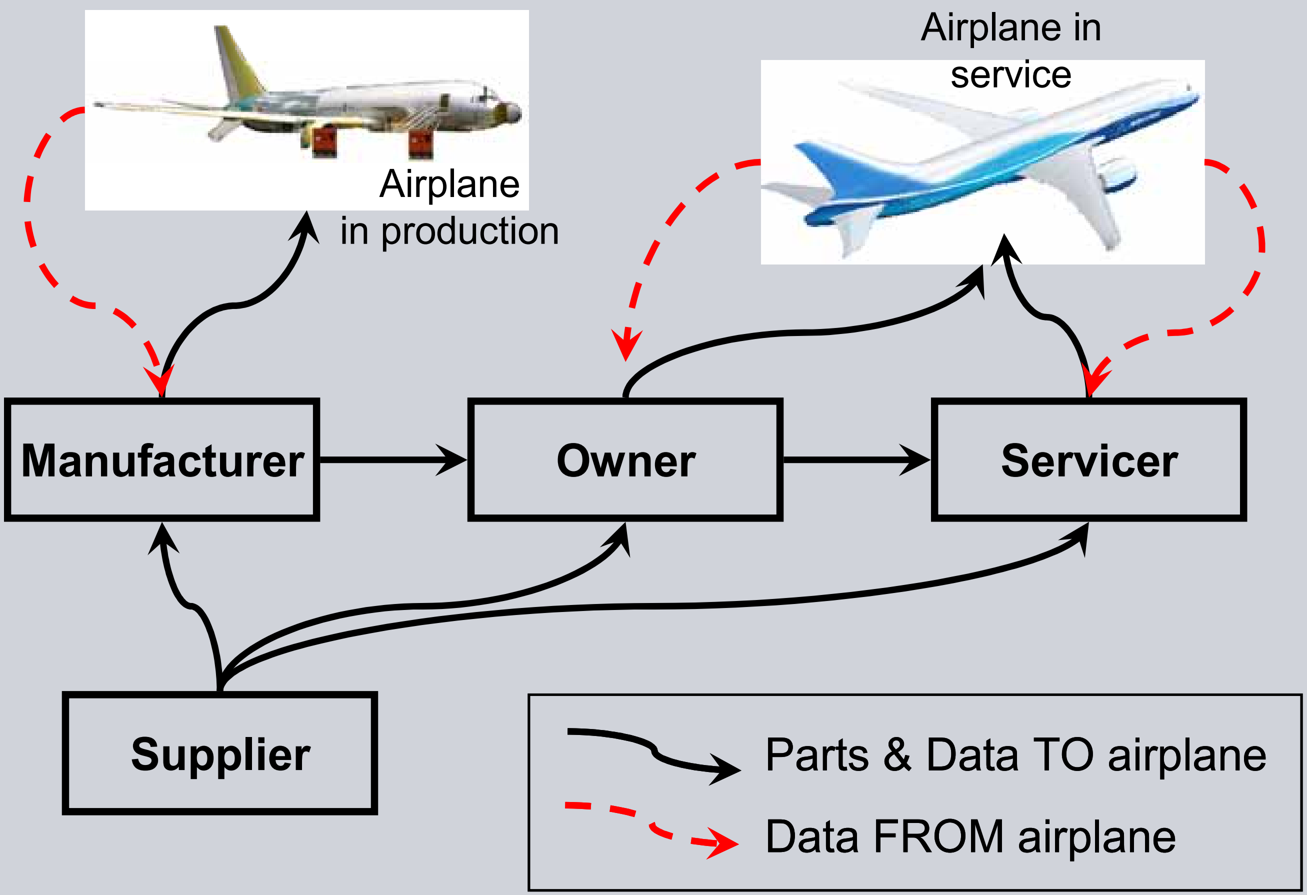


Overview

- IT Security at Siemens CT
- **S**oftware **D**istribution **S**ystems
- Common Criteria certification
- Formal Security Analysis
- Alice-Bob protocol model
- Validation with AVISPA Tool
- Conclusion

Airplane Assets Distribution System

AADS is a system for storage and distribution of airplane assets, including *Loadable Software Airplane Parts* (LSAP) and airplane health data



AADS architecture

A complex distributed store-and-forward middleware with OSS components

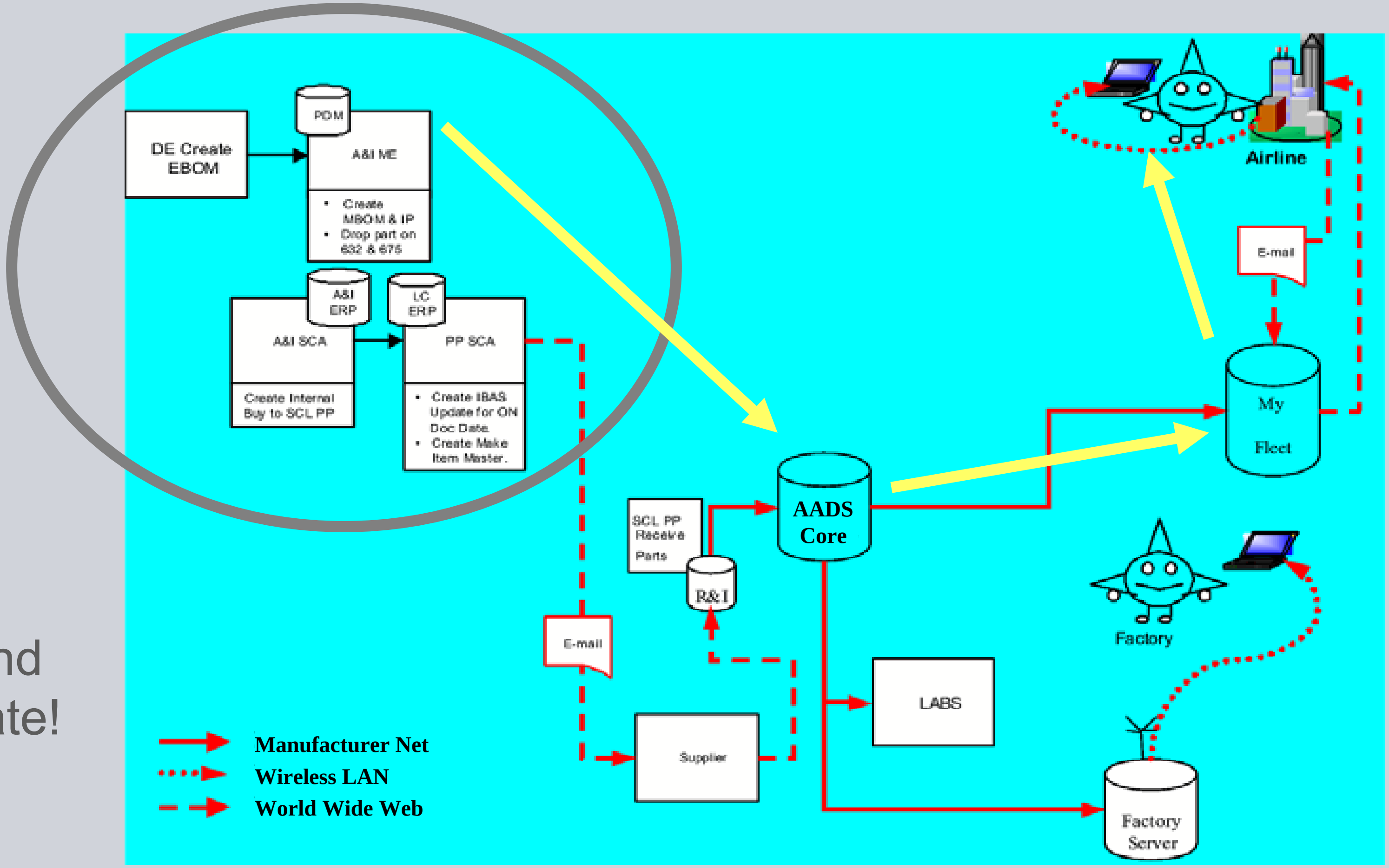
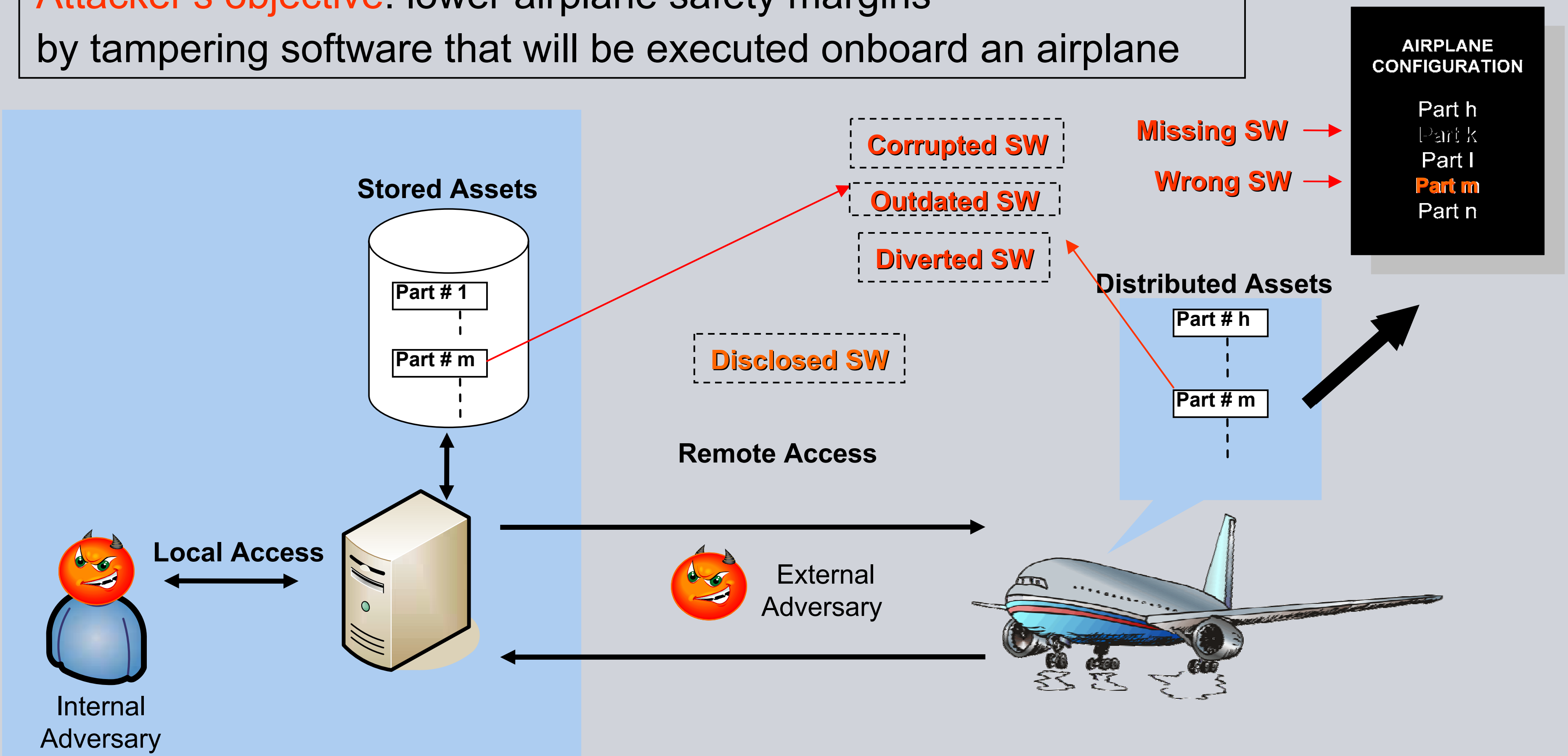


Figure is simplified and not up-to-date!

Security threats at the airplane example

SIEMENS

Attacker's objective: lower airplane safety margins
by tampering software that will be executed onboard an airplane

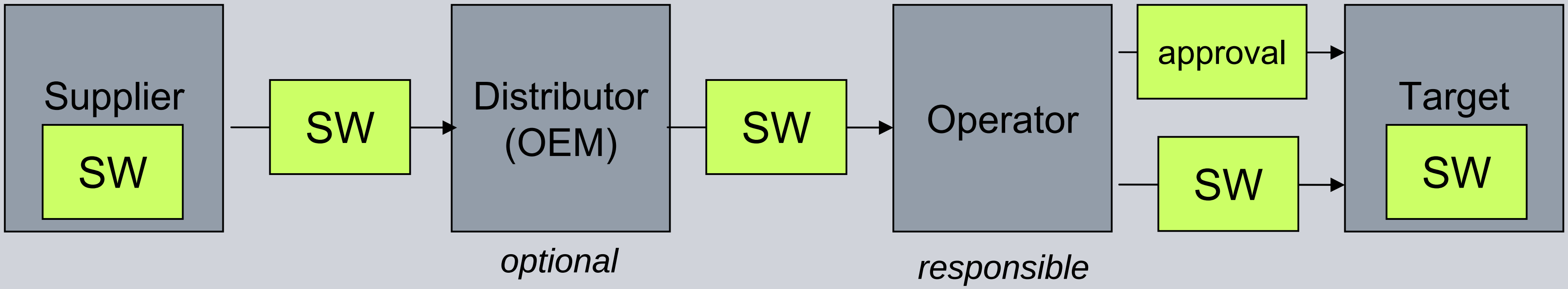


Corruption/Injection **Wrong Version** **Diversion** **Disclosure**

Software Distribution System (SDS)

ICT systems with networked devices in the field
performing safety-critical and/or security-critical tasks.
Field devices require secure software update.

→ Software Distribution System (SDS):
System providing secure distribution of software (SW)
from software supplier to target devices in the field

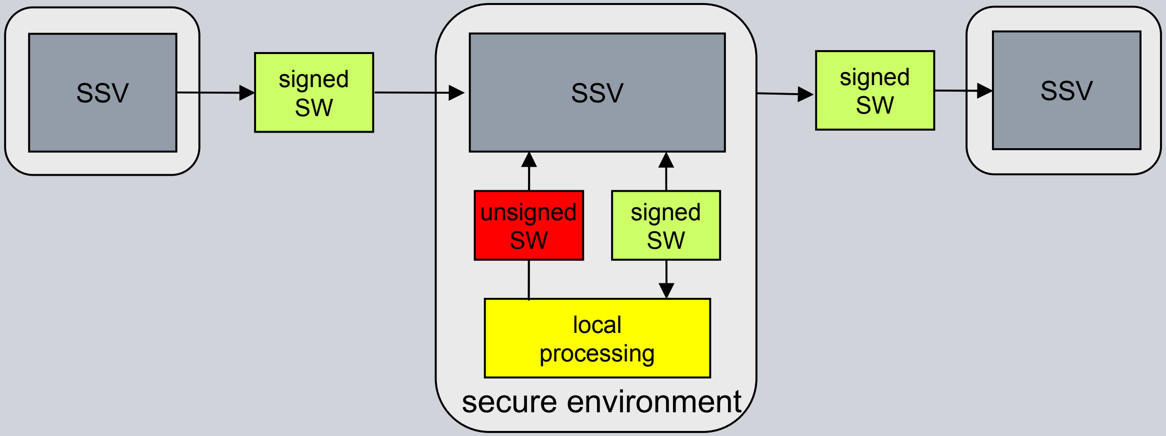


Transition from media-based (CD-ROMs etc.) to networked SW transport increases security risks due to transport over open, untrusted networks

Software Signer Verifier (SSV)

Each node in SDS runs an SSV instance, used for:

- Introducing unsigned software into the SDS, by digitally signing and optionally encrypting it
- Verifying the signature on software received from other SSVs, checking integrity, authenticity and authorization of the sender
- Approving software by adding an authorized signature
- Delivering software out of the SDS after successfully verifying it



Overview

- IT Security at Siemens CT
- **S**oftware **D**istribution **S**ystems
- Common Criteria certification
- Formal Security Analysis
- Alice-Bob protocol model
- Validation with AVISPA Tool
- Conclusion

SIEMENS

IT Security as a System Engineering Problem

- IT security aims at preventing, or at least detecting, unauthorized actions by agents in an IT system.

In the AADS context, security is a prerequisite of safety.

- Safety aims at the absence of accidents (→ airworthiness)

Situation: security loopholes in IT systems actively exploited

Objective: thwart attacks by eliminating vulnerabilities

Difficulty: IT systems are very complex. Security is interwoven with the whole system, so very hard to assess.

Remedy: evaluate system following the Common Criteria approach

- address security systematically in all development phases
- perform document & code reviews and tests
- for maximal assurance, use formal modeling and analysis

Common Criteria (CC) for IT security evaluation

Common Criteria

product-oriented methodology

for IT security assessment

ISO/IEC **standard** 15408

Current version: 3.1 of end-2006

Aim: gain confidence in the security of a system

- What are the objectives the system should achieve?
- Are the measures employed appropriate to achieve them?
- Are the measures implemented and deployed correctly?

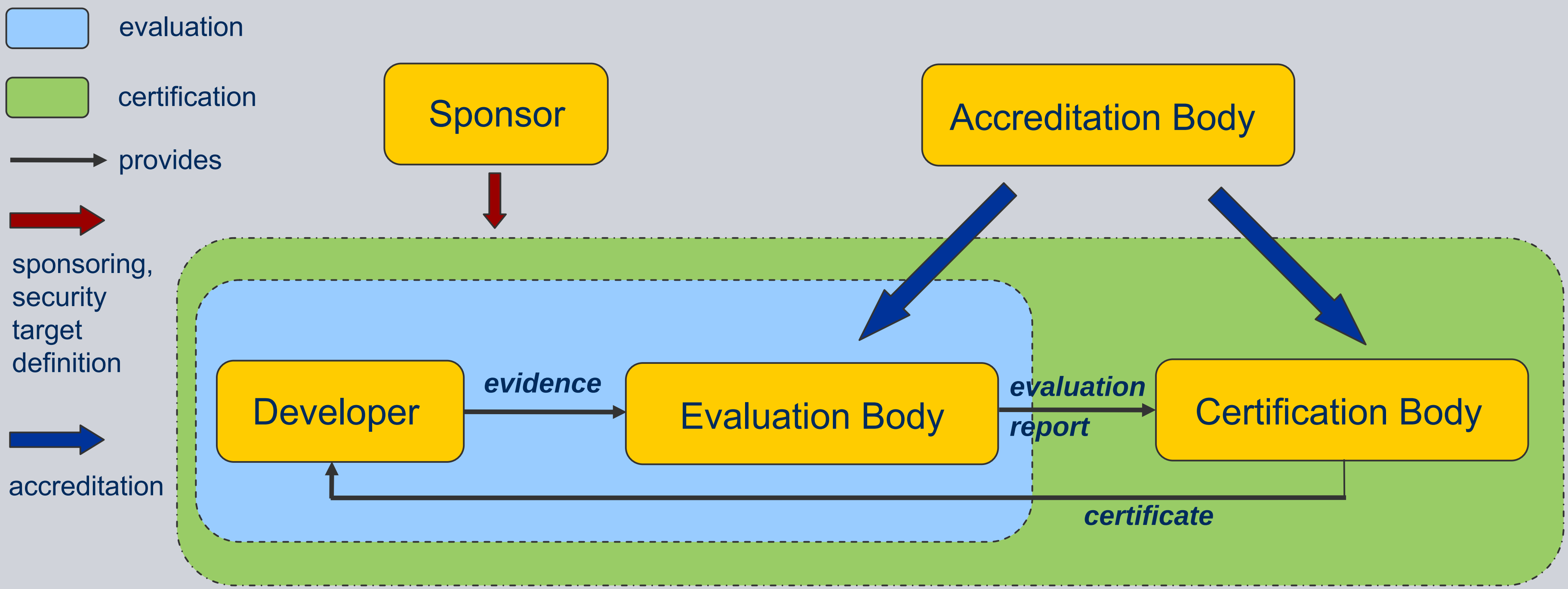
CC General Approach

Approach: assessment of system + documents by neutral experts

- Gaining understanding of the system's security functionality
- Checking evidence that the functionality is correctly implemented
- Checking evidence that the system integrity is maintained

SIEMENS

CC Process Scheme



Certification according to the Common Criteria is a rather complex, time consuming and expensive process.

A successful, approved evaluation is awarded a certificate.

SIEMENS

CC: Security Targets

Security Target (ST): defines extent and depth of the evaluation

for a specific product called *Target of Evaluation (TOE)*

Protection Profile (PP): defines extent and depth of the evaluation

for a whole class of products, i.e. firewalls

STs and PPs may inherit (‘*claim*’) other PPs.

ST and PP specifications use **generic** “construction kit”:

- Building blocks for defining *Security Functional Requirements (SFRs)*
- Scalable in depth and rigor: *Security Assurance Requirements (SARs)*

  layered as *Evaluation Assurance Levels (EALs)*

AADS Security Specification: CC Protection Profile (1)

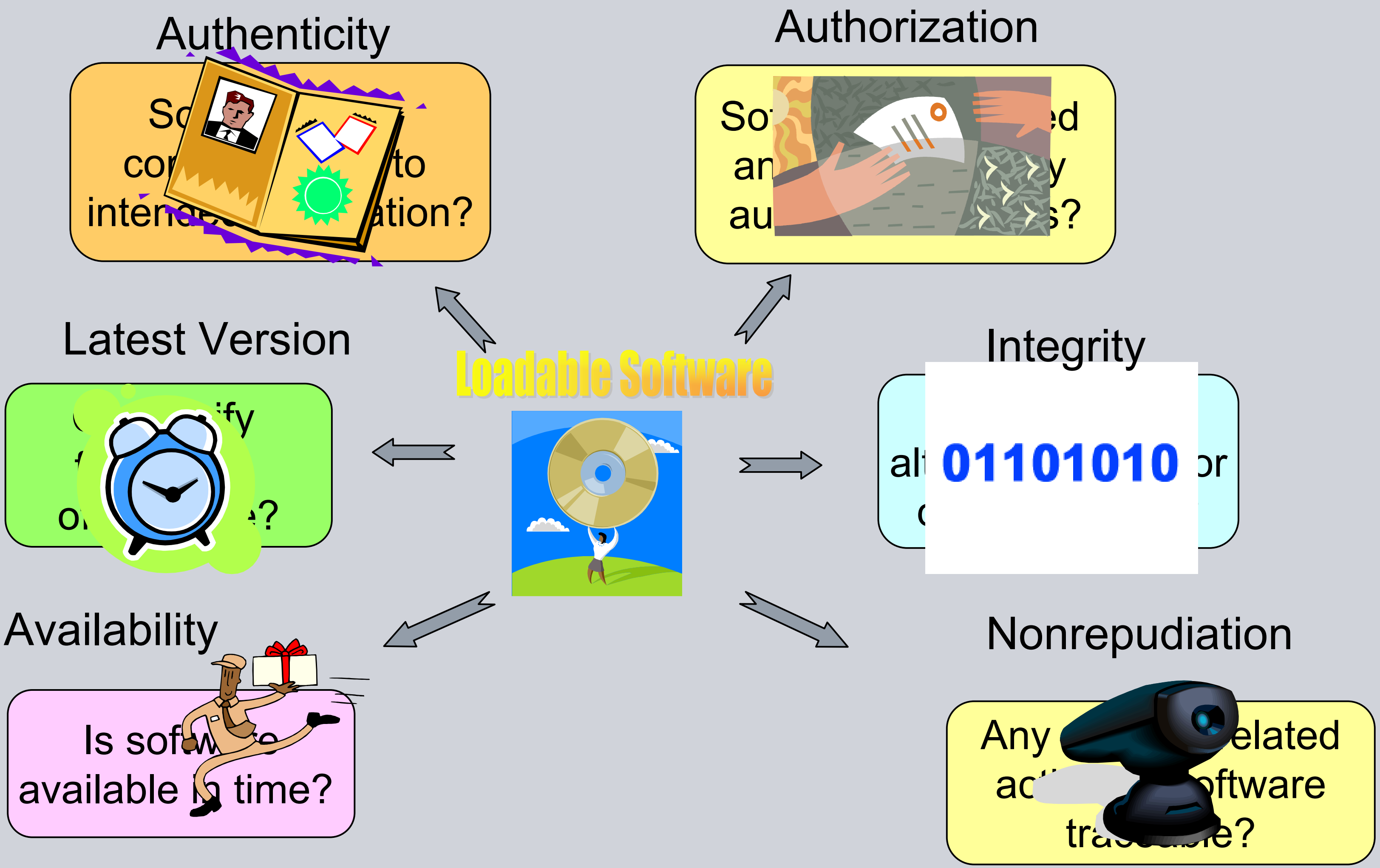
1. Introduction
2. System Description - Target of Evaluation (TOE)
3. Security Environment

   - Assets and Related Actions
   - Threats
   - Required Assurance Level
   - Assumptions

4. Security Objectives

   - …
   - Rationale

Security Objectives for AADS



SIEMENS

Threats Addressed by the AADS Security Objectives

Objectives	Threats	Safety-relevant				Business-relevant			
		Corruption	Misconfiguration	Diversion	Staleness	Unavailability	Late Detection	False Alarm	Repudiation
Safety-relevant	Integrity	√							
	Correct Destination			√					
	Latest Version				√				
	Authentication	√	√						√
	Authorization	√	√						
	Timeliness				√				
Business-Relevant	Availability					√			
	Early Detection						√		
	Correct Status							√	
	Traceability	√	√						√
	Nonrepudiation								√
Environment	Part_Coherence	√	√	√					
	Loading_Interlocks	√	√	√					
	Protective_Channels	√							
	Network_Protection				√	√			
	Host_Protection	√							√
Assumptions	Adequate_Signing	√							
	Configuration		√						
	Development	√	√	√	√	√	√	√	√
	Management	√	√						√

SIEMENS

AADS Security Specification: CC Protection Profile (2)

1. Introduction
2. System Description
3. Security Environment

   - Assets and Related Actions
   - Threats
   - Required Assurance Level
   - Assumptions

4. Security Objectives

   - …
   - Rationale

5. Security Functional Requirements

   - …
   - Rationale

SIEMENS

CC: Security Functional Requirements (SFRs) overview

FAU: Security audit

- Security audit automatic response (FAU_ARP)
- Security audit data generation (FAU_GEN)
- Security audit analysis (FAU_SAA)
- Security audit review (FAU_SAR)
- Security audit event selection (FAU_SEL)
- Security audit event storage (FAU_STG)

FCO: Communication

FCS: Cryptographic support

FDP: User data protection

FIA : Identification and authentication

FMT: Security management

FPR: Privacy

FPT: Protection of the TSF

FRU: Resource utilization

FTA: TOE access

FTP: Trusted path/channels

CC: EALs

Security Assurance Requirements (SARs)

grouped as

Evaluation Assurance Levels (EALs)

Assurance class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Development	ADV_ARC		**1**	1	1	1	1	1
	ADV_FSP	**1**	**2**	**3**	**4**	**5**	5	**6**
	ADV_IMP				**1**	1	**2**	2
	ADV_INT					**2**	**3**	3
	ADV_SPM						**1**	1
	ADV_TDS		**1**	**2**	**3**	**4**	**5**	**6**
Guidance documents	AGD_OPE	**1**	1	1	1	1	1	1
	AGD_PRE	**1**	1	1	1	1	1	1
Life-cycle support	ALC_CMC	**1**	**2**	**3**	**4**	4	**5**	5
	ALC_CMS	**1**	**2**	**3**	**4**	**5**	5	5
	ALC_DEL		**1**	1	1	1	1	1
	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD			**1**	1	1	1	**2**
	ALC_TAT				**1**	**2**	**3**	3
Security Target evaluation	ASE_CCL	**1**	1	1	1	1	1	1
	ASE_ECD	**1**	1	1	1	1	1	1
	ASE_INT	**1**	1	1	1	1	1	1
	ASE_OBJ	**1**	**2**	2	2	2	2	2
	ASE_REQ	**1**	**2**	2	2	2	2	2
	ASE_SPD		**1**	1	1	1	1	1
	ASE_TSS	**1**	1	1	1	1	1	1
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	**2**	**3**	3	**4**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Vulnerability assessment	AVA_VAN	**1**	**2**	2	**3**	**4**	**5**	5

SIEMENS

CC: Evaluation Assurance Level 2

Development	ADV_ARC.1 Security architecture description ADV_FSP.2 Security-enforcing functional specification ADV_TDS.1 Basic design
Guidance documents	AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures
Life-cycle support	ALC_CMC.2 Use of a CM system ALC_CMS.2 Parts of the TOE CM coverage ALC_DEL.1 Delivery procedures
Security Target Evaluation	ASE_XXX *(6 families of components)*
Tests	ATE_COV.1 Evidence of coverage ATE_FUN.1 Functional testing ATE_IND.2 Independent testing - sample
Vulnerability analysis	AVA_VAN.2 Vulnerability analysis

CC: Evaluation Assurance Level 4

Development	ADV_FSP.4 **Complete** functional specification **ADV_IMP.1 Implementation representation of the TSF** ADV_TDS.**3** Basic **modular** design
Guidance documents	
Life-cycle support	ALC_CMC.**4 Production support, acceptance procedures and automation** ALC_CMS.**4 Problem tracking** CM coverage **ALC_DVS.1 Identification of security measures** **ALC_LCD.1 Developer defined life-cycle model** **ALC_TAT.1 Well-defined development tools**
Security Target Eval.	
Tests	ATE_COV.**2 Analysis** of coverage **ATE_DPT.2 Testing: security enforcing modules**
Vulnerability analysis	AVA_VAN.**3 Focused** vulnerability analysis

SIEMENS

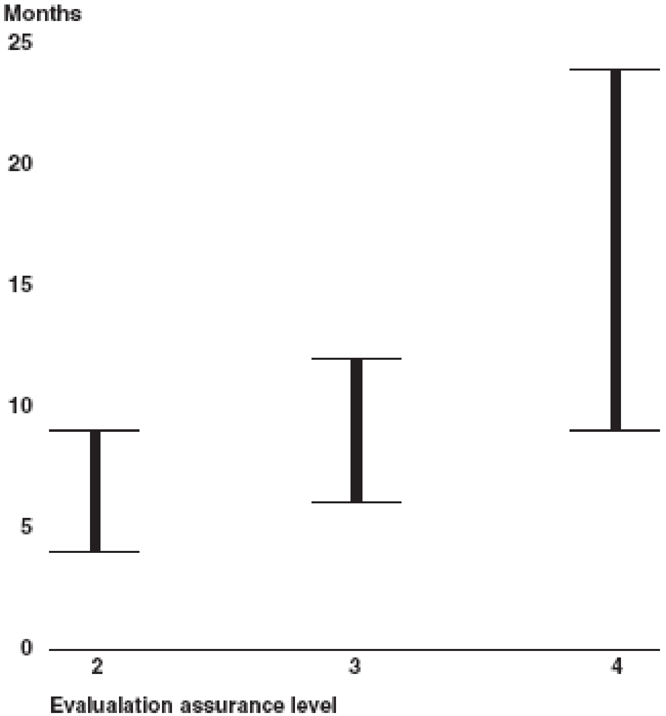
CC: Evaluation Assurance Level 6

Development	ADV_FSP.**5** Complete semi-formal functional spec. **with additional error information**
	ADV_IMP.**2 Implementation** of the TSF
	ADV_INT.3 Minimally complex internals
	ADV_SPM.1 Formal **TOE security policy model**
	ADV_TDS.**5 Complete semiformal** modular design
Guidance documents	
Life-cycle support	ALC_CMC.**5 Advanced** support
	ALC_CMS.**5 Development tools** CM coverage
	ALC_DVS.**2 Sufficiency** of security measures
	ALC_TAT.**3 Compliance with implementation standards – all parts**
Security Target Eval.	
Tests	ATE_COV.**3 Rigorous** analysis of coverage
	ATE_DPT.**3** Testing: **modular design**
	ATE_FUN.2 Ordered functional testing
Vulnerability analysis	AVA_VAN.**5 Advanced methodical** vulnerability analysis

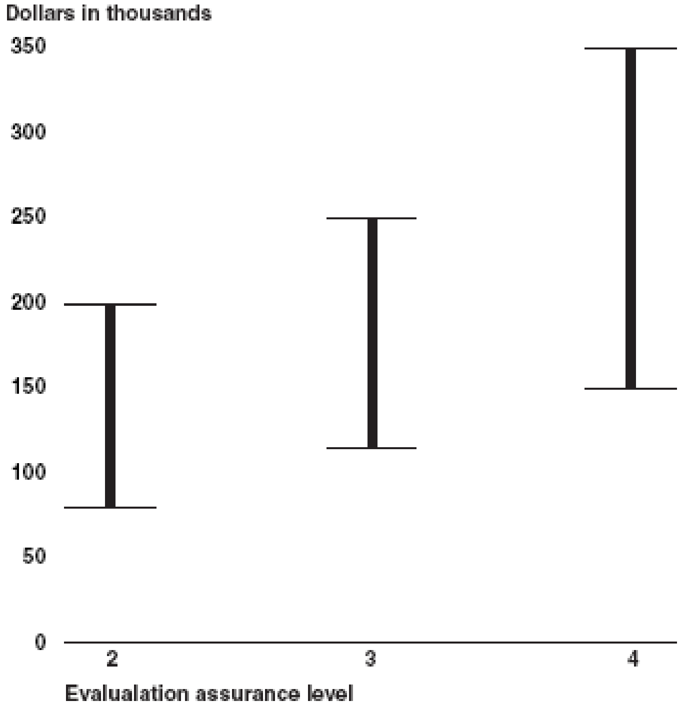
SIEMENS

CC: Factors determining the evaluation effort

- Definition of TOE vs. TOE environment
- Definition of Treats and Security Objectives for the TOE
- Definition of Security Functional Requirements (SFRs)
- Selection of Evaluation Assurance Level (EAL)



Source: GAO analysis of data provided by laboratories.



Source: GAO analysis of data provided by laboratories.

SIEMENS

Selection of Evaluation Assurance Level (EAL) for AADS

	Flight safety	Airline business
Threat Level assume sophisticated adversary with moderate resources who is willing to take XXX risk	**T5**: XXX = significant e.g. intl. terrorists	**T4**: XXX = little e.g. organized crime, sophisticated hackers, intl. corporations
Information Value violation of the protection policy would cause YYY damage to the security, safety, financial posture, or infrastructure of the organization	**V5**: YYY= exceptionally grave Risk: loss of lives	**V4**: YYY = serious Risk: airplanes out of service, or damage airline reputation
Evaluation Assurance Level for the given Treat Level and Information Value	**EAL 6**: semiformally verified design and tested	**EAL 4**: methodically designed, tested, and reviewed

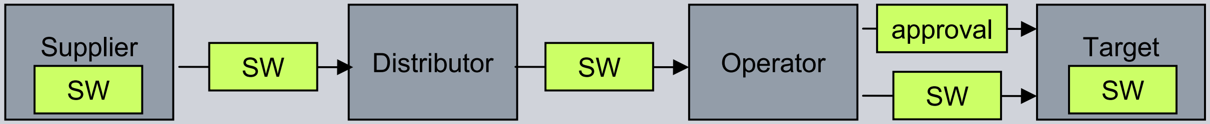
Evaluating the whole AADS at EAL 6 would be extremely costly.

Currently available Public Key Infrastructure (PKI) certified only at EAL 4.

Two-level approach: evaluate only LSAP integrity & authenticity at EAL6.

Hybrid security assessment

- Highest CC evaluation assurance levels (EAL 6-7) require formal analysis
- SDS usually are complex distributed systems with many components



General problems:

- Highly critical system, but (complete) formal analysis too costly
- CC offer only limited support ("CAP") for modular system evaluation

Pragmantic approach:

- Define confined security kernel with generic component: SSV
- Software Signer Verifier (SSV) handles digital signatures at each node
- Evaluate SSV according to Common Criteria EAL4 (non-formal)
- Analyze the interaction of SSVs in a formal way (→ crypto protocol)

SIEMENS

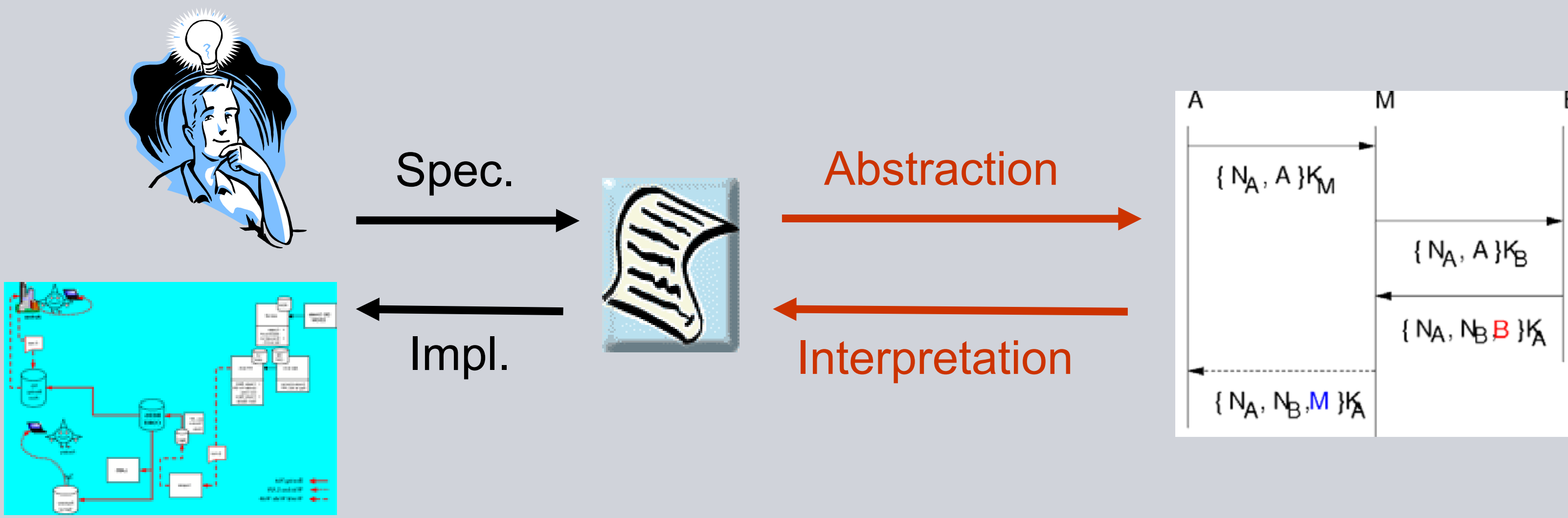
Overview

- IT Security at Siemens CT
- **S**oftware **D**istribution **S**ystems
- Common Criteria certification
- Formal Security Analysis
- Alice-Bob protocol model
- Validation with AVISPA Tool
- Conclusion

SIEMENS

Formal Security Analysis: Approach and Benefits

Mission: security analysis with maximal precision
Approach: formal modeling and verification



Improving the quality of the system specification

Checking for the existence of security loopholes

High-Level Protocol Spec. Language
- Model checkers (AVISPA tools)

Interacting State Machines
- Interactive theorem prover (Isabelle)

Security Models

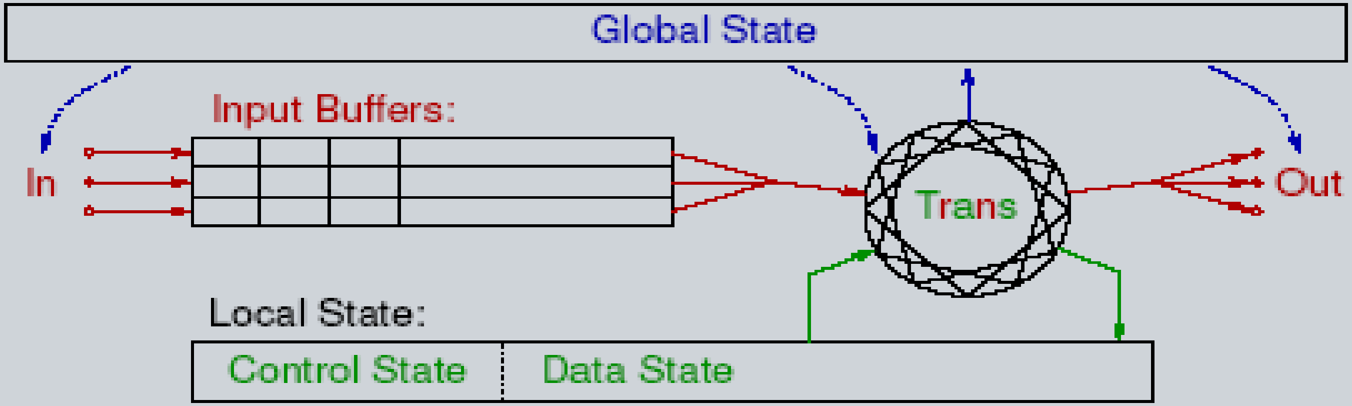
SIEMENS

- A security policy defines what is allowed (actions, data flow, ...) typically by a relationship between subjects and objects.
- A security model is a (+/- formal) description of a policy and enforcing mechanisms, usually in terms of system states or state sequences (traces).
- Security verification proves that mechanisms enforce policy.
- Models focus on specific characteristics of the reality (policies).
- Types of formal security models
  - Automata models
  - Access Control models
  - Information Flow models
  - Cryptoprotocol models

SIEMENS

Interacting State Machines (ISMs)

Automata with (nondeterministic) state transitions + buffered I/O, simultaneously on multiple connections.



Transitions definable in executable and/or axiomatic style.

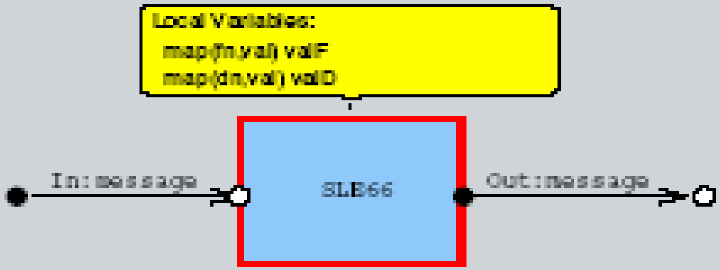
An ISM system may have changing global state.

Applicable to a large variety of reactive systems.

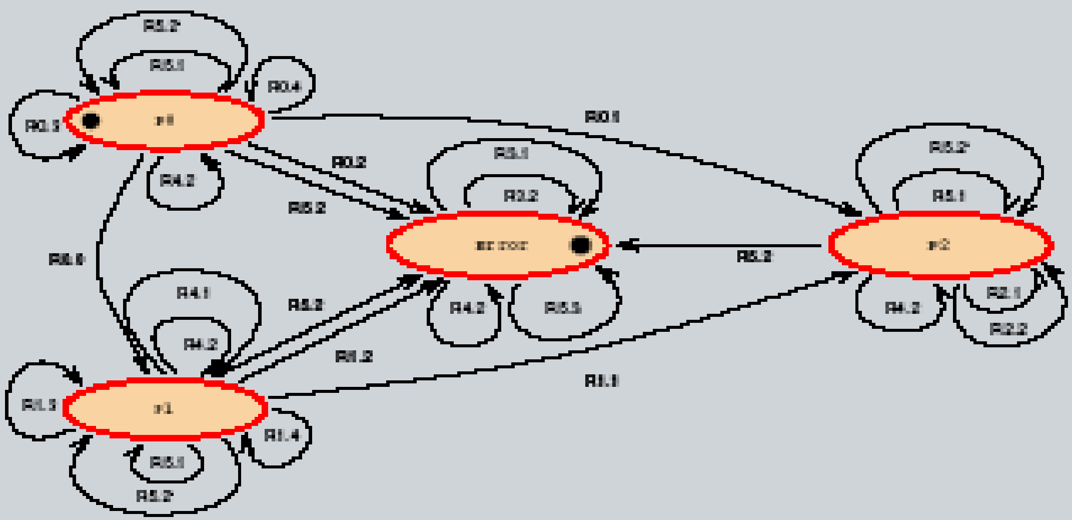
By now, not much verification support (theory, tools).

Model of Infineon SLE 66 Smart Card Processor

System Structure Diagram:



State Transition Diagram (abstracted):



First higher-level (EAL5) certification for a smart card processor!

RBAC of Complex Information System

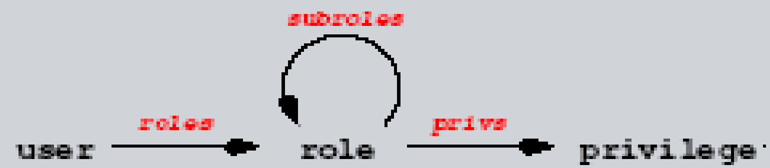
Is the security design (with emergency access etc.) sound?

Privileges:

$\mathit{roles} \subseteq \mathtt{user} \times \mathtt{role}$

$\mathit{subroles} \subseteq \mathtt{role} \times \mathtt{role}$

$\mathit{privs} \subseteq \mathtt{role} \times \mathtt{privilege}$



$$(u, p) \in \mathit{roles} \circ \mathit{subroles}^* \circ \mathit{privs}$$

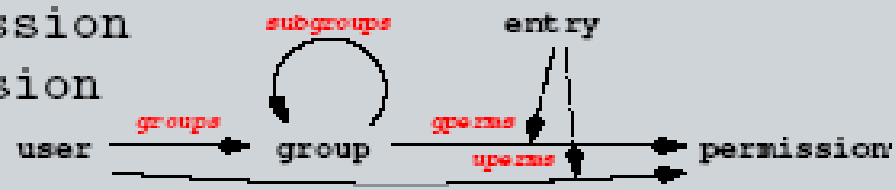
Permissions:

$\mathit{groups} \subseteq \mathtt{user} \times \mathtt{group}$

$\mathit{subgroups} \subseteq \mathtt{group} \times \mathtt{group}$

$\mathit{gperms} \subseteq \mathtt{group} \times \mathtt{permission}$

$\mathit{uperms} \subseteq \mathtt{user} \times \mathtt{permission}$



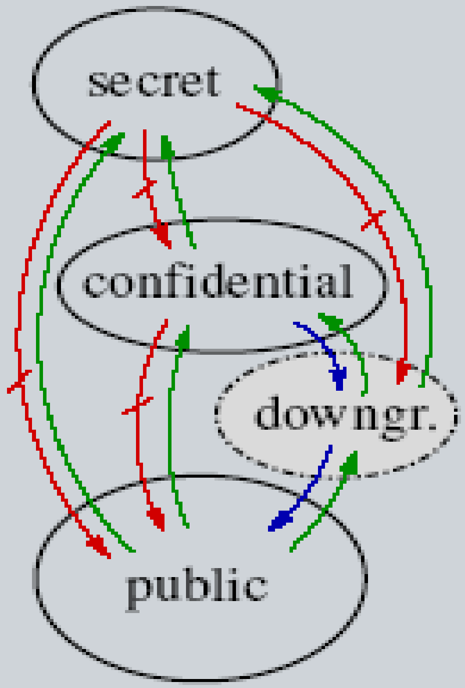
$$(u, p) \in (\mathit{groups} \circ \mathit{subgroups}^* \circ \mathit{gperms}(e)) \ \cup \ \mathit{uperms}(e)$$

"nagging questions" $\rightsquigarrow$ clarifications improving specification quality.

Open issue: relation between model and implementation ($\rightsquigarrow$ testing).

Information Flow Models

- Identify knowledge/information domains
- Specify allowed flow between domains
- Check the observations that can be made about state and/or actions
- Consider also indirect and partial flow



- Classical model: Noninterference (Goguen & Meseguer)
- Many variants: Non-deducability, Restrictiveness, Non-leakage, ...

Very strong, but rarely used in practice

Available: connection with ISMs

Language-based Information Flow Security

SIEMENS

Policy: no assignments of high-values to low-variables, enforced by type system

Semantically: take (x, y) as elements of the state space with high-level data (on left) and low-level data (on right).

Step function $S(x,y) = (S_H(x,y), S_L(x,y))$ does not leak information from high to low if $S_L(x_1,y) = S_L(x_2,y)$ (functional independence).

Observational equivalence $(x,y) \stackrel{L}{\sim} (x',y') \;:\longleftrightarrow\; y = y'$ allows re-formulation:

$$\boxed{s \stackrel{L}{\sim} t \;\longrightarrow\; S(s) \stackrel{L}{\sim} S(t) \quad (\text{preservation of } \stackrel{L}{\sim})}$$

Generalization to action sequences α and arbitrary policies $\leadsto$

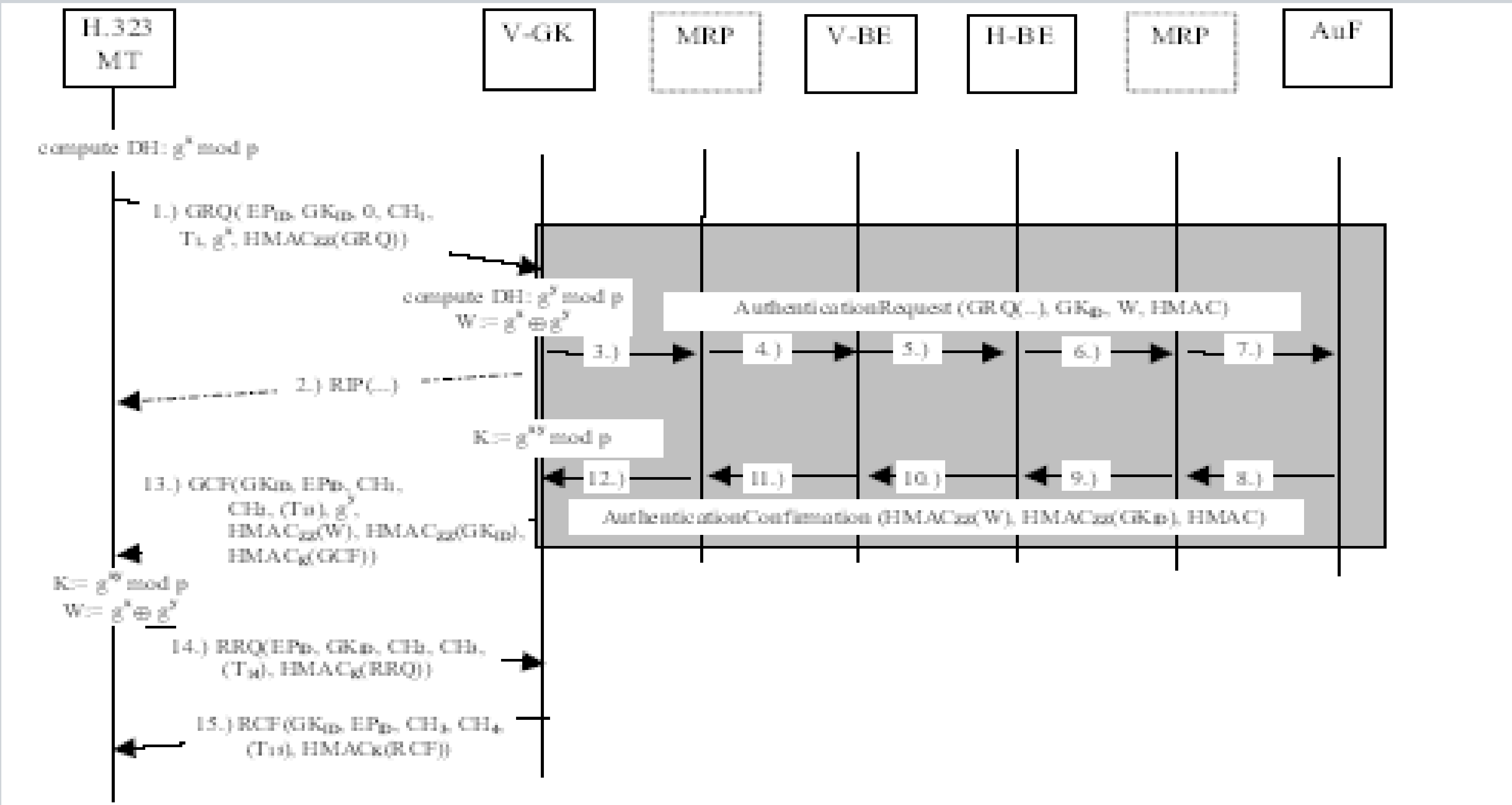
Cryptoprotocol models

- Describe message exchange between processes or principals

- Take cryptographic operations as perfect primitives
- Describe system with specialized modeling languages
- State secrecy, authentication, . . . goals
- Verify (mostly) automatically using model-checkers

EU project AVISPA , ...

H.530 Mobile Roaming Authentication



Two vulnerabilities found and corrected. Solution standardized.

Shaping a Formal Model

SIEMENS

Formality Level: should be adequate:

- the more formal, the more precise,
- but requires deeper mastering of formal methods

Choice of Formalism: dependent on ...

- application domain, modeler's experience, tool availability, ...
- formalism should be simple, expressive, flexible, mature

Abstraction Level: should be ...

- high enough to achieve clarity and limit the effort
- low enough not to loose important detail

refinement allows for both high-level and detailed description

SIEMENS

Formal Security Analysis: Information Required

- Overview: system architecture (components and interfaces),
  e.g. databases, authentication services, connections,…
- Security-related concepts: actors, assets, states, messages, …
- Threats: which attacks have to be expected.
- Assumptions: what does the environment fulfill.
- Security objectives: what the system should achieve.
  Described in detail such that concrete verification goals can be set up
  e.g. integrity: which contents shall be modifiable by whom, at which times,
  by which operations (and no changes otherwise!)
- Security mechanisms: relation to objectives and how they are achieved.
  e.g. who signs where which contents, and where is the signature checked
  Described precisely but at high level (no implementation details required),
  e.g. abstract message contents/format but not concrete syntax

SIEMENS

Development Phases and the Benefits of Formal Analysis

Requirements analysis:

understanding the security issues

- abstraction: concentration on essentials, to keep overview
- genericity: standardized patterns simplify the analysis

Design, documentation:

quality of specifications

- enforces preciseness and completeness

Implementation:

effectiveness of security functionality

- formal model as precise reference for testing and verification

Overview

- IT Security at Siemens CT
- **S**oftware **D**istribution **S**ystems
- Common Criteria certification
- Formal Security Analysis
- Alice-Bob protocol model
- Validation with AVISPA Tool
- Conclusion

Formal modeling: Alice-Bob notation

```
SUP - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP}_KDIS -> DIS
DIS - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP
            .{h(Asset).OP }_inv(KDIS).CertDIS}_KOP  -> OP
OP  - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP
            .{h(Asset).OP }_inv(KDIS).CertDIS
            .{h(Asset).TD }_inv(KOP ).CertOP }_KTD  -> TD
```

`A - M -> B`	message `M` sent from A to B
`Asset`	a software item including its identity
`h(M)`	the hash value (i.e. crypto checksum) of content `M`
`M.N`	the concatenated contents of `M` and `N`
`{M}_inv(K)`	content `M` digitally signed with private key K
`{M}_K`	content `M` encrypted with public key K

Formal modeling: SDS protocol structure

```
SUP - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP}_KDIS -> DIS
DIS - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP
            .{h(Asset).OP }_inv(KDIS).CertDIS}_KOP  -> OP
OP  - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP
            .{h(Asset).OP }_inv(KDIS).CertDIS
            .{h(Asset).TD }_inv(KOP ).CertOP }_KTD  -> TD
```

SUP: software supplier	with private key `inv(KSUP)`
DIS: software distributor	with private key `inv(KDIS)`
OP : target operator	with private key `inv(KOP)`
TD : target device	with private key `inv(KTD)`

Signatures comprise hash value of asset and identity of intended receiver

Signatures are applied in parallel (rather than nested or discarded)

SIEMENS

Formal modeling: SDS approvals and certificates

```
SUP - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP}_KDIS -> DIS
DIS - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP
            .{h(Asset).OP }_inv(KDIS).CertDIS}_KOP  -> OP
OP  - {Asset.{h(Asset).DIS}_inv(KSUP).CertSUP
            .{h(Asset).OP }_inv(KDIS).CertDIS
            .{h(Asset).TD }_inv(KOP ).CertOP }_KTD  -> TD
```

- Approval information partially modelled: **operator** determines **target**
- Certificate of a node relates its identity with its public key, e.g. certificate of supplier SUP: `CertSUP = {SUP.KSUP}_inv(KCA)`
- Certificate authority (CA) with private key `inv(KCA)`
- Certificates are self-signed or signed by CA
- Locally stored sets of public keys of trusted SSVs and CAs

SIEMENS

Overview

- IT Security at Siemens CT
- **S**oftware **D**istribution **S**ystems
- Common Criteria certification
- Formal Security Analysis
- Alice-Bob protocol model
- Validation with AVISPA Tool
- Conclusion

Verification goals

Show asset authenticity & integrity (end-to-end) and confidentiality:

- assets accepted by target have indeed been sent by the supplier
- assets accepted by target have not been modified during transport
- assets remain secret among the SSV instances

Proved asset authenticity & integrity also hop-by-hop

Correct destination covered:

- Name of the intended receiver in signed part, checked by target.
  Signature of the operator acts as installation approval statement

Correct version not modelled:

- Version info is integrity protected, but
  checks delegated to SSV local environment

Results of formal verification

- Alice-Bob notation not detailed and precise enough
- Use the specification language of the AVISPA Tool: HLPSL
- Software Signer Verifier (SSV) as parameterized role (node class)
- SDS as communication protocol linking different SSV instances
- Multiple protocol sessions describing individual SW transports

- Modelcheckers at their complexity limits, due to
  - parallel signatures, only the latest one being checked
  - multiple instances of central nodes (e.g. manufacturer)
  - …?

Overview

- IT Security at Siemens CT
- **S**oftware **D**istribution **S**ystems
- Hybrid security assessment
- Alice-Bob protocol model
- Formal Security Analysis
- Validation with AVISPA Tool
- Conclusion

SIEMENS

Conclusion (1) on AADS

- Challenges for AADS development
  - pioneering system design and architecture
  - complex, heterogeneous, distributed system
  - security is critical for both safety and business

- Common Criteria offer adequate methodology for assessment

- Systematic approach, in particular formal analysis, enhances
  - understanding of the security issues
  - quality of specifications and documentation
  - confidence (of Boeing, customers, FAA, etc.) in the security solutions

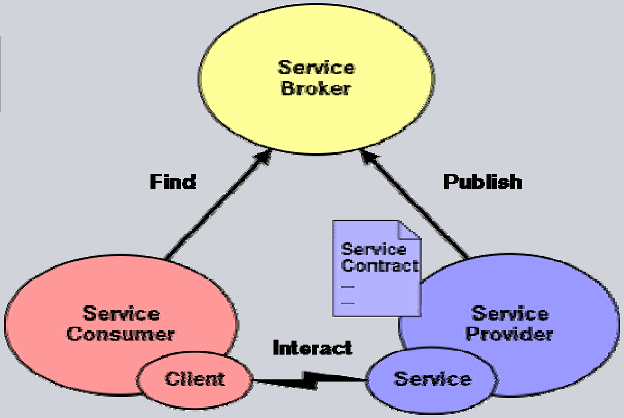
Conclusion (2) on AADS

- Experience with SDS evaluation
  - Common Criteria most widely accepted methodology
  - Problem of compositional security evaluation not solved
  - Use formal analysis where cost/benefit ratio is best
  - Highly precise design and documentation:
    assumptions, requirements
  - Shape system architecture to support security evaluation

- Future steps
  - Key management aspects:
    Public Key Infrastructure (PKI) components
  - Configuration management
    with installation instructions and reports

AVISPA successor: AVANTSSAR

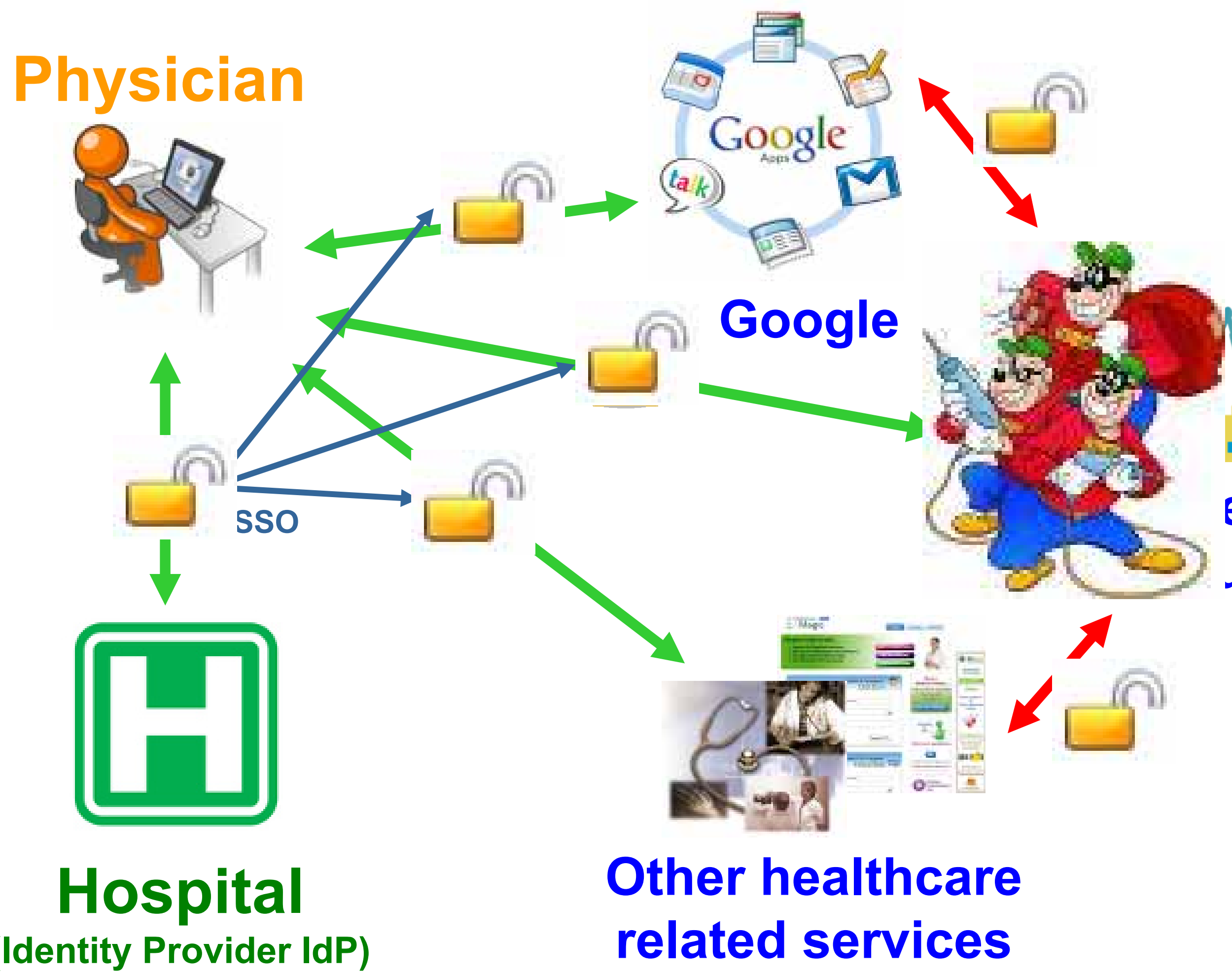
Automated VAlidatioN
of
Trust and Security
of
Service-oriented ARchitectures



FP7-2007-ICT-1, ICT-1.1.4, Strep project no. 216471
(36 months duration, 590 PMs, 6M€ budget, 3.8M€ EC contribution)

Single Sign-On (SSO): One access point for everything

AVANTSSAR analysis of Google SAML SSO: also for attackers!



A malicious service provider can access the data of the physician located at all other services connected via Google SSO

Further info on AVANTSSAR

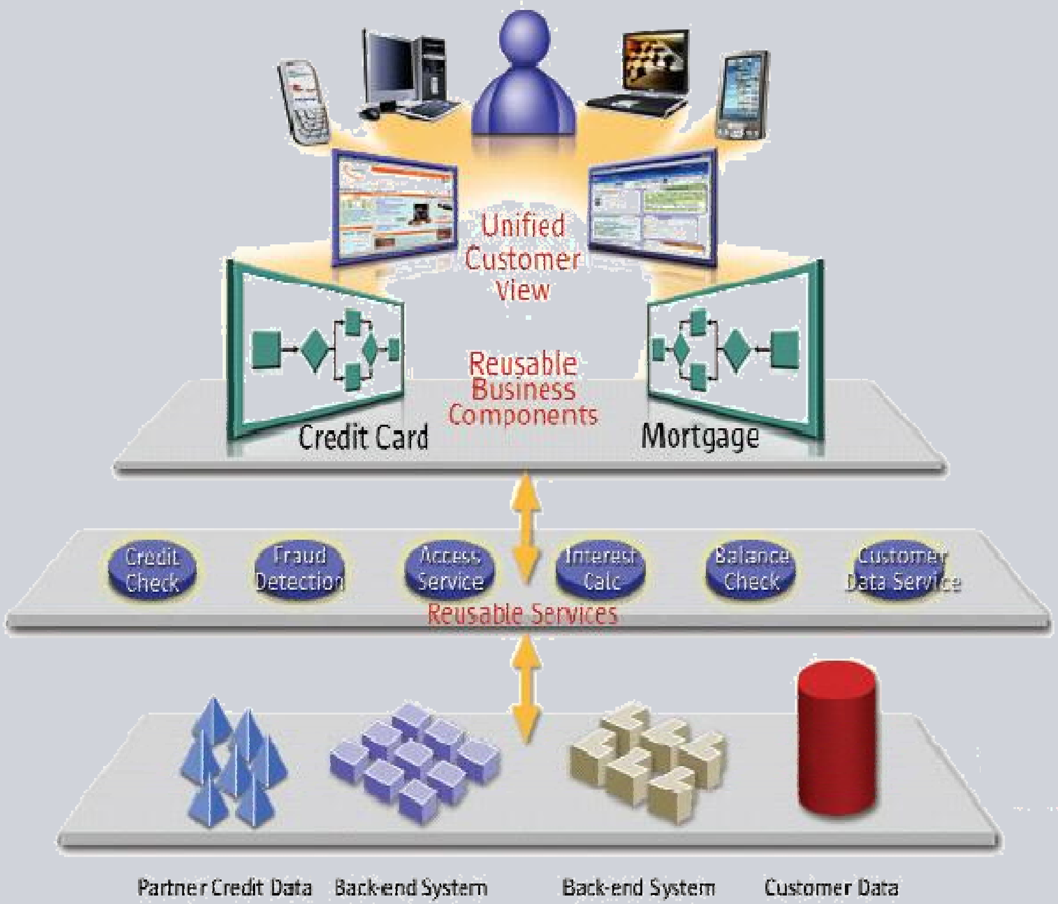
AVANTSSAR project motivation

SIEMENS

ICT paradigm shift: from components to services, composed and reconfigured dynamically in a demand-driven way

Trustworthy service may interact with others causing novel trust and security problems

Validation of composition of individual services into service-oriented architectures dramatically needed



AVANTSSAR consortium

Industry

IBM Zurich Research Labs

OpenTrust Paris

SAP Research France

Siemens AG Munich

Academia

Università di Verona

ETH Zurich

INRIA Lorraine

UPS-IRIT Toulouse

Università di Genova

IEAT Timisoara

Expertise

Service-oriented enterprise architectures

Security solutions

Standardization and industry migration

Automated security validation

Formal methods

Security engineering

SIEMENS

AVANTSSAR main objectives and principles

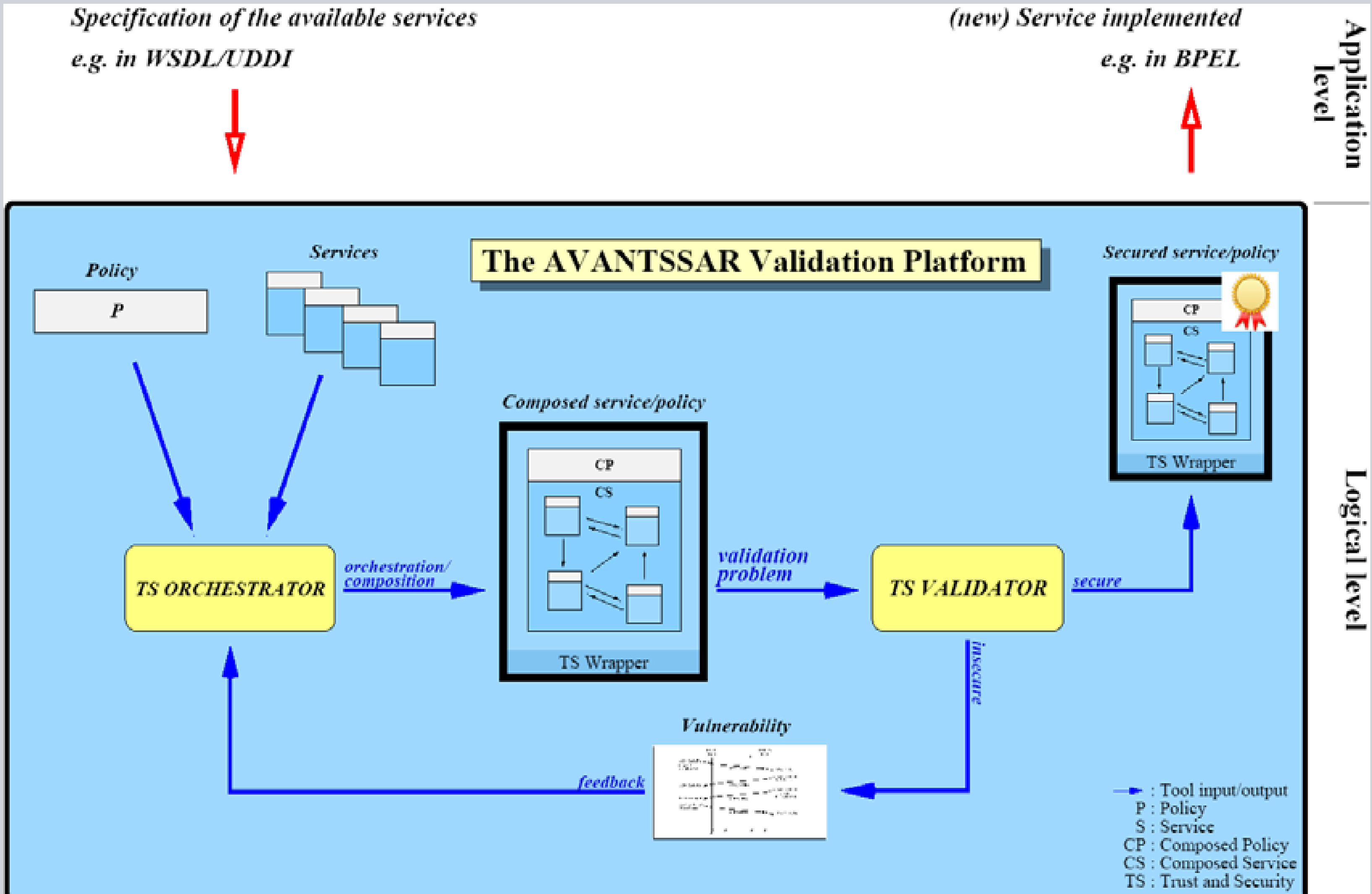
Platform for formal specification and automated validation of trust and security of SOAs

- First formal language for specifying trust and security properties of services, their policies, and their composition into service-oriented architectures
- Automated toolset supporting the above
- Library of validated industrially-relevant case studies

Migration of platform to industry and standardization organizations

- Speed-up development of new service infrastructures
- Enhance their security and robustness
- Increase public acceptance of SOAs

AVANTSSAR project results and innovation



AVANTSSAR impact: industry migration

Services need to be securely combined according to evolving trust and security requirements and policies

A rigorous demonstration that a composed SOA meets the security requirements and enforces the application policy will

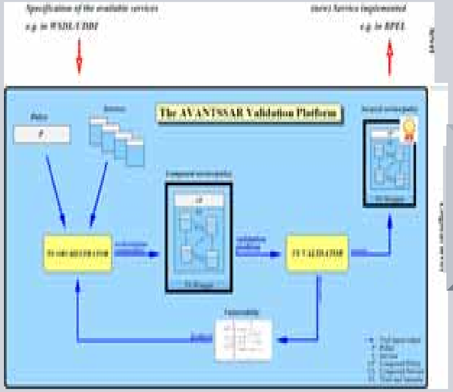
- significantly increase customers' confidence
- enable them to fully exploit the benefits of service orientation

Integration of AVANTSSAR Platform in industrial development environment

The AVANTSSAR Platform will advance the security of industrial vendors' service offerings: validated, provable, traceable

AVANTSSAR will thus significantly strengthen the competitive advantage of the products of the industrial partners

LOP



Health care