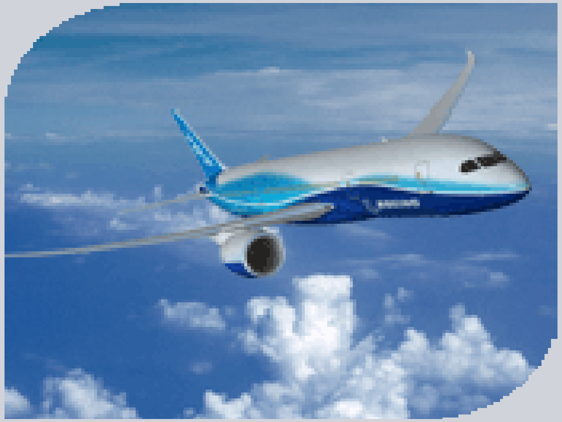
Corporate Technology

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



Dr. David von Oheimb

Siemens Corporate Technology

ISoLA 2006 Security Keynote
Paphos, Cyprus, 16 November 2006

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

Overview

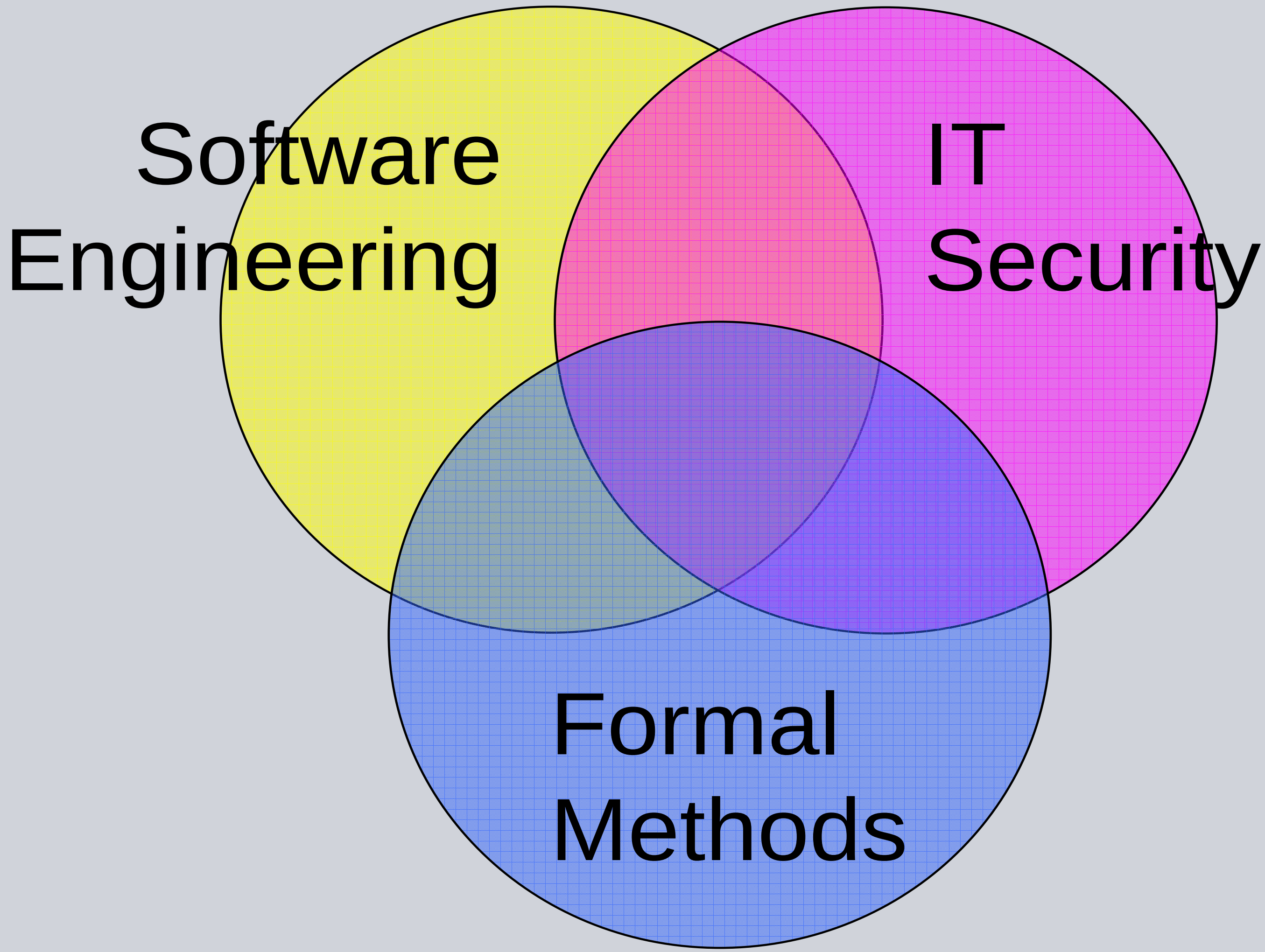
- IT Security at Siemens
- Airplane Assets Distribution System
- Formal Security Analysis
- Modeling and Analysis Techniques
- Security Certification
- Conclusion

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

Security Applications & Methods

- **Secure Operating Systems, Trusted Platform Modules (TPM)**
- **General Purpose Identity Management and Authorization**
  - Role / Policy Based Access Control, Public Key Infrastructure (PKI), SSO
- **Web Services Security**
  - Security of Service Oriented Architecture (SOA)
- **Application level security: e-health, e-gov, e-Commerce**
- **Digital Rights Management (DRM)**
- **Formal Methods and Certification**

Fields

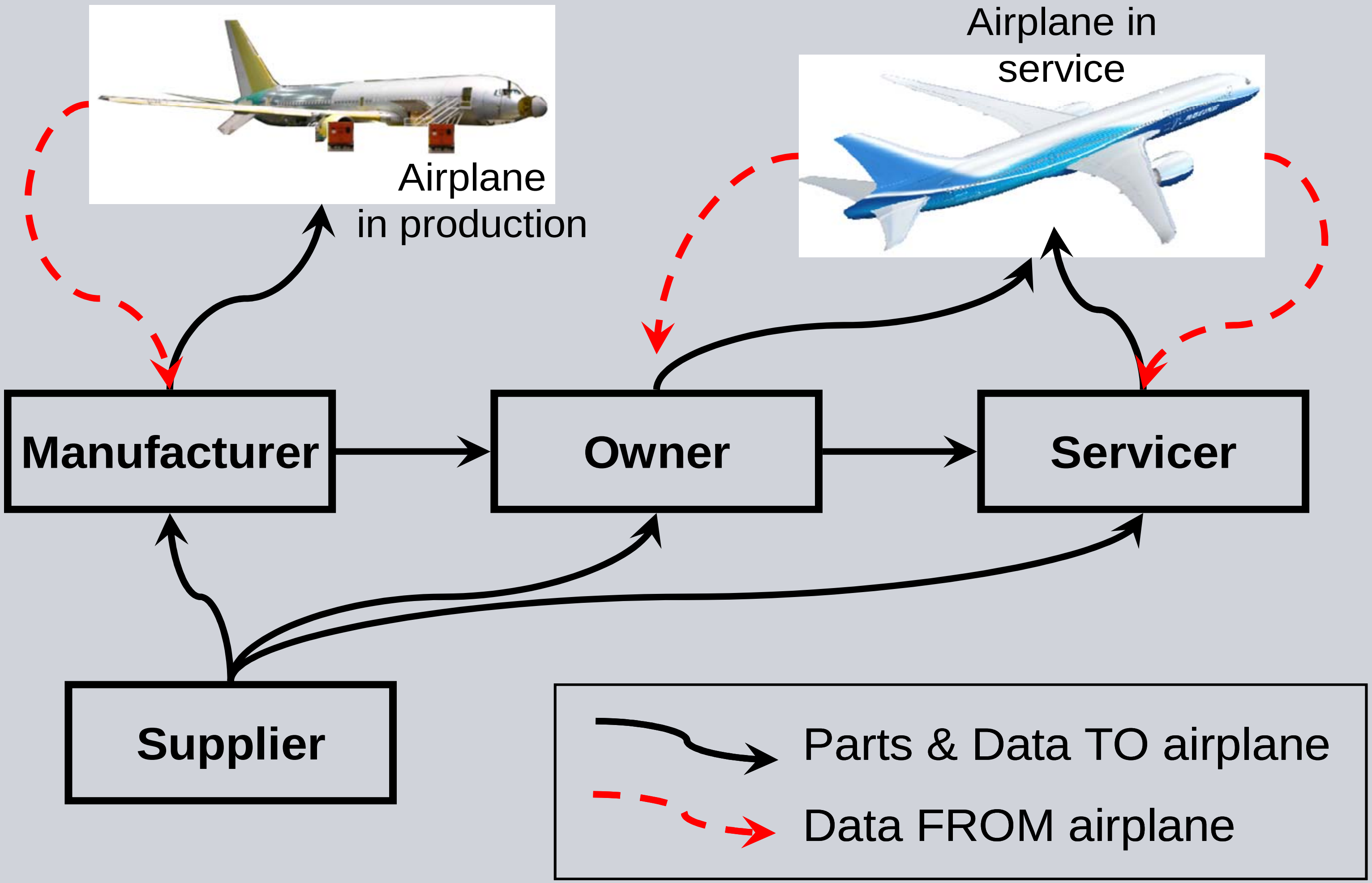


Overview

- IT Security at Siemens
- Airplane Assets Distribution System
- Formal Security Analysis
- Modeling and Analysis Techniques
- Security Certification
- Conclusion

Airplane Assets Distribution System

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



AADS architecture

SIEMENS

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

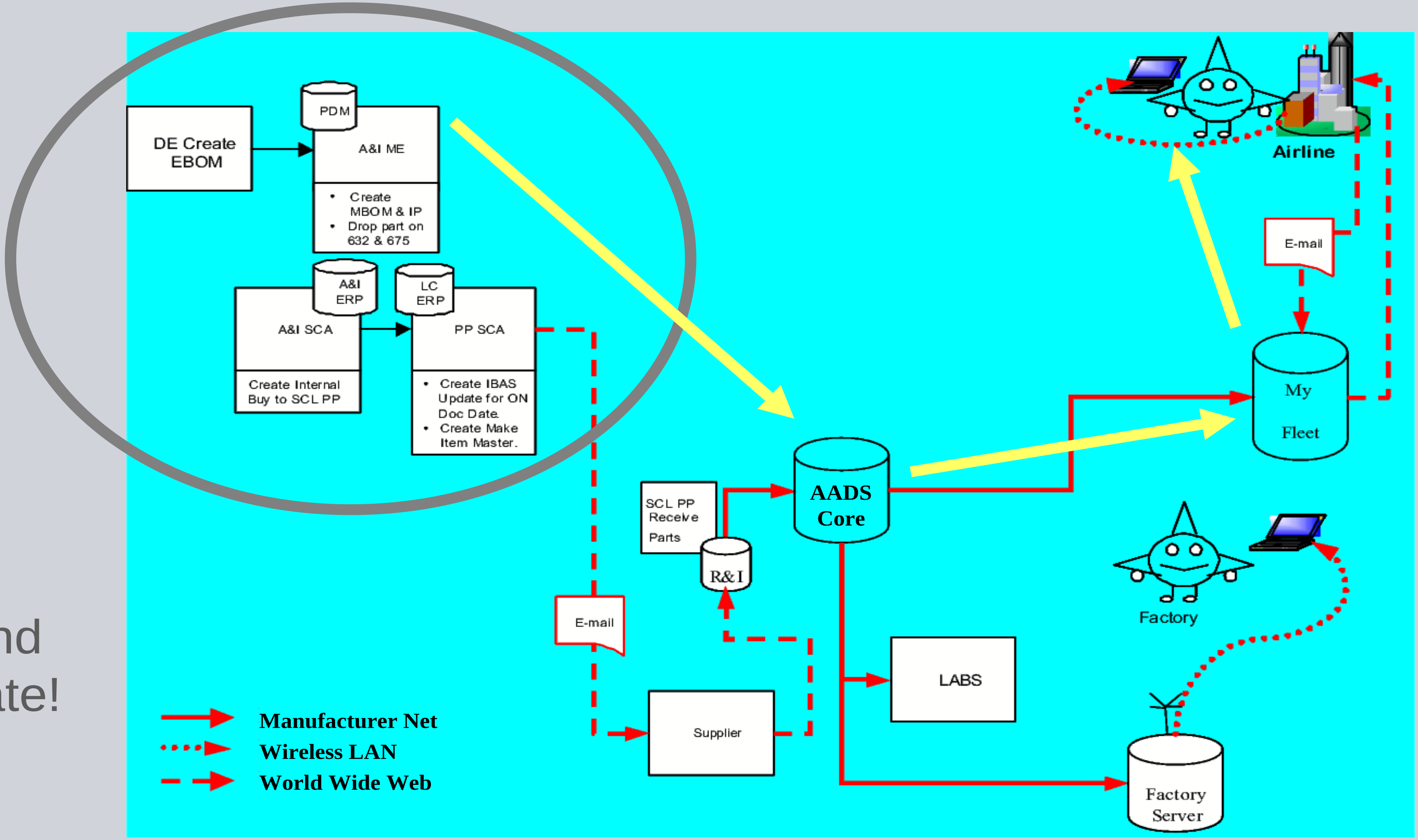
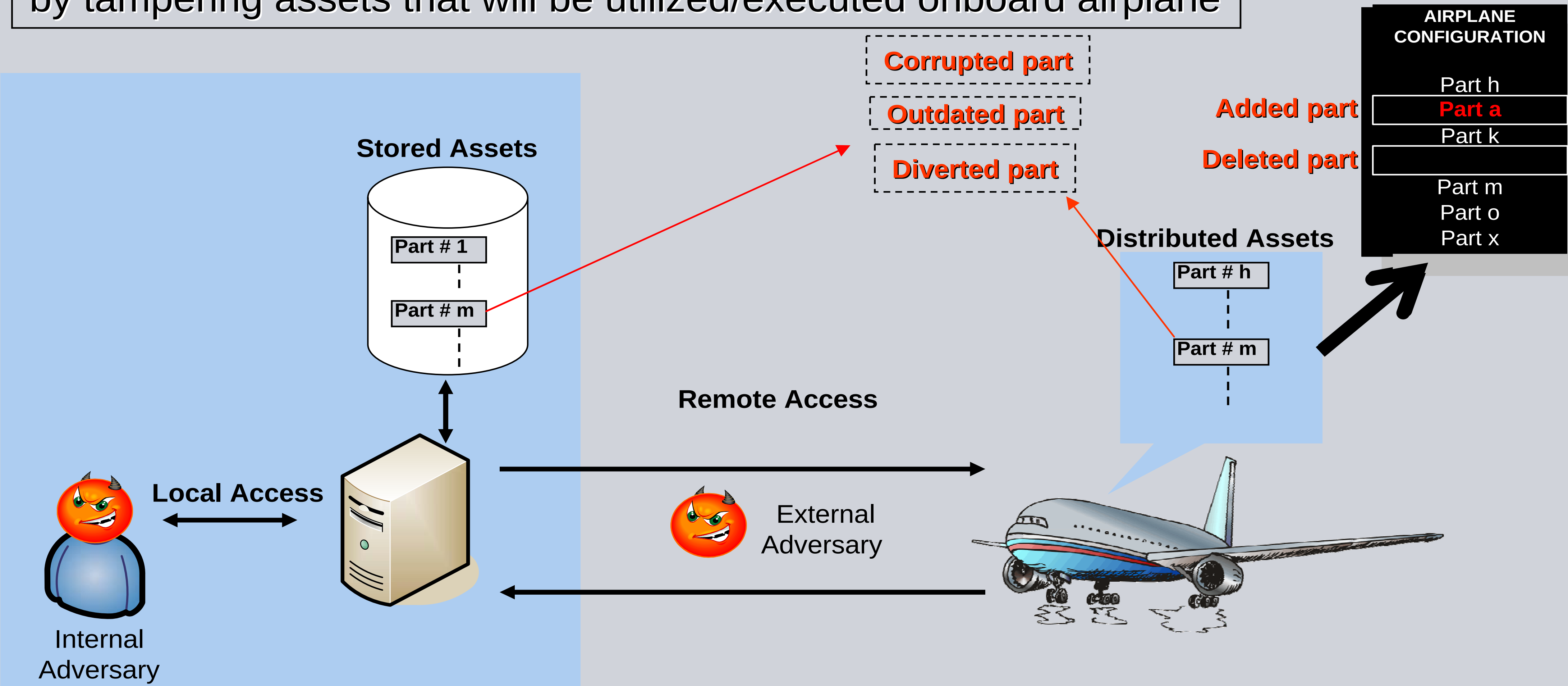


Figure is simplified and not up-to-date!

Safety-relevant Threats

Safety-relevant Threats: lower airplane safety
by tampering assets that will be utilized/executed onboard airplane

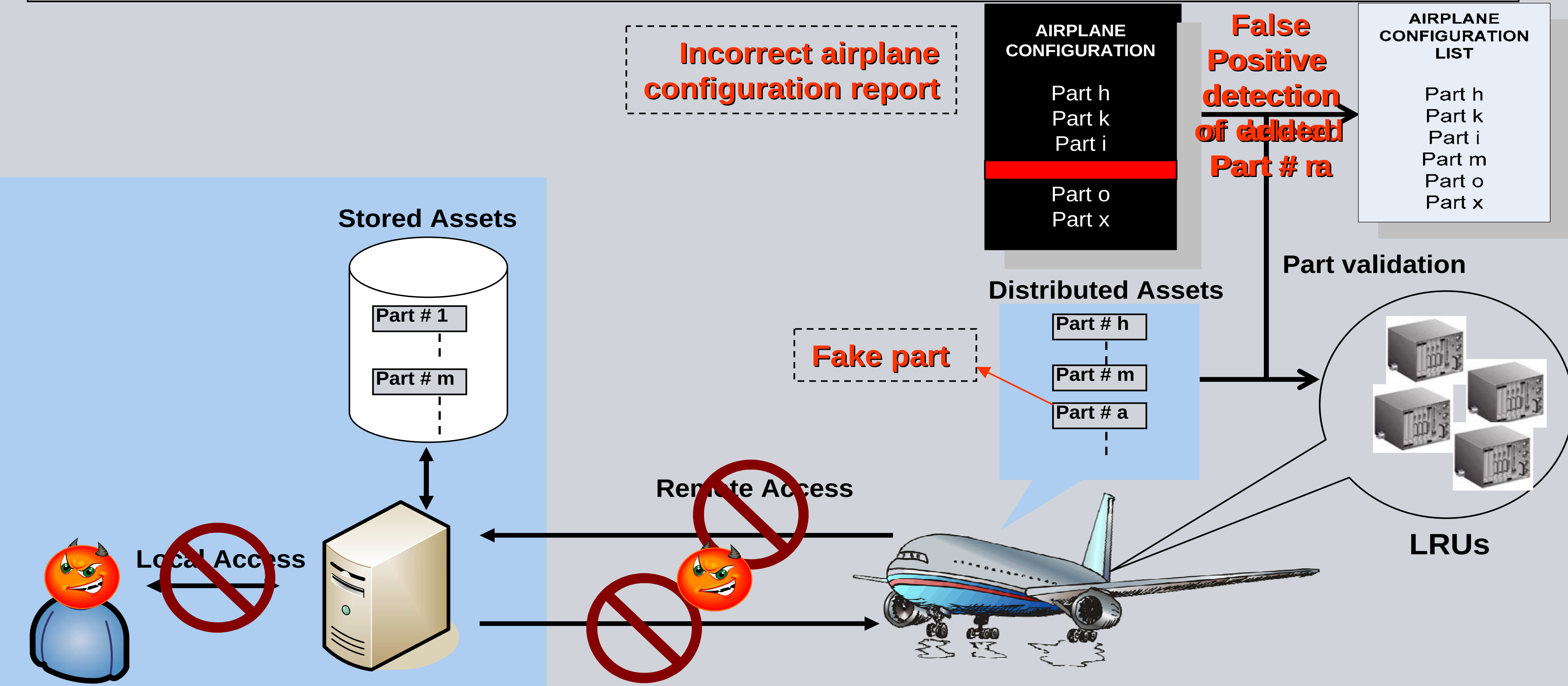


ST.Corruption **ST.Staleness** **ST.Diversion** **ST.Misconfiguration**

SIEMENS

Business-relevant Threats

Business-relevant Threats: impede business of airplane production, operation, and maintenance organizations by disrupting airplane service



BT.Late_Detection **BT.False_Alarms** **BT.Denial_of_Service**

Overview

- IT Security at Siemens
- Airplane Assets Distribution System
- Formal Security Analysis
- Modeling and Analysis Techniques
- Security Certification
- Conclusion

SIEMENS

Security as a SW Engineering Problem

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

This complements

- safety: absence of damage due to mistakes or other unintentional failure

Situation: security loopholes in IT systems actively exploited
Objective: thwart attacks by absence of vulnerabilities
Difficulty: security is interwoven with the whole system.
IT systems are very complex, security flaws hard to find.

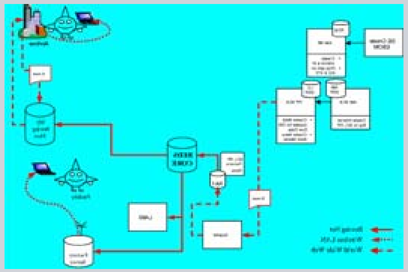
Remedy:

- address security in all development phases
- do reviews and tests
- make use of formal modeling / analysis

SIEMENS

The Promise of Formal Security Analysis

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

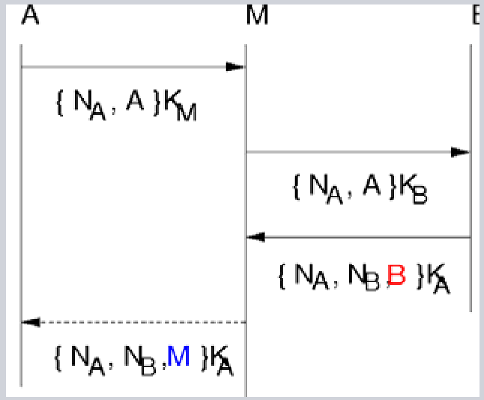


Spec.

Impl.

Abstraction

Interpretation



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)

Information Necessary for Formal Analysis

- Overview: system architecture and components,
  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 (only),
  at which times, by which operations
- Security mechanisms: relation to goals 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

Security Specification (1)

1. Introduction
2. System Description
3. Security Environment

   - Assets and Related Actions
   - Threats
   - Required Assurance Level (for certification)
   - Assumptions

4. Security Objectives

   - …
   - Rationale

SIEMENS

Threats Addressed by the Security Objectives

Objectives	Threats	Safety-relevant				Business-relevant			
		Corruption	Misconfiguration	Diversion	Staleness	Asset Un-availability	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	√	√						√

Security Specification (2)

1. Introduction
2. System Description
3. Security Environment
   - Assets and Related Actions
   - Threats
   - Required Assurance Level (for certification)
   - Assumptions
4. Security Objectives
   - …
   - Rationale
5. Security Functional Requirements
   - …
   - Rationale

Overview

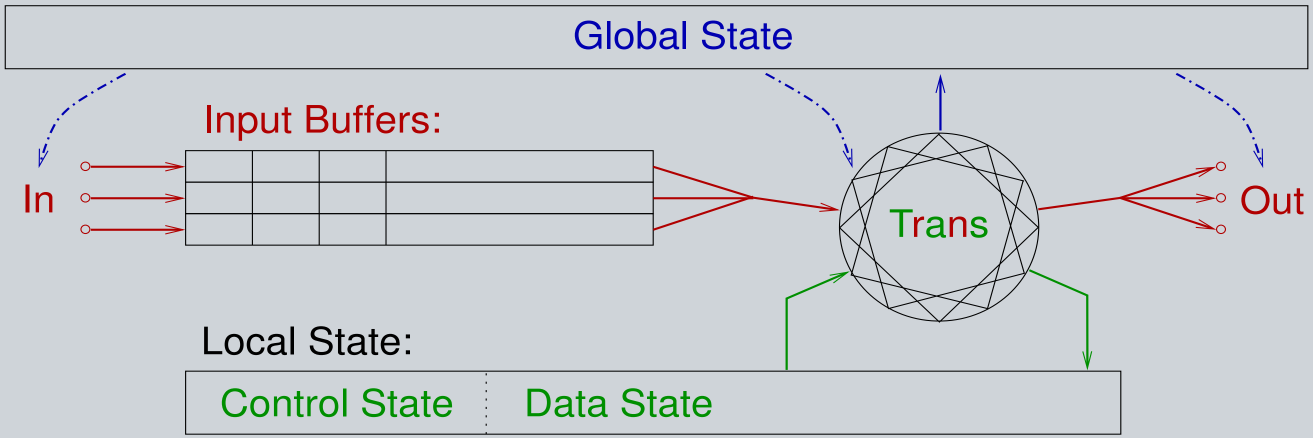
- IT Security at Siemens
- Airplane Assets Distribution System
- Formal Security Analysis
- Modeling and Analysis Techniques
- Security Certification
- Conclusion

Security Models

- 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

Interacting State Machines (ISMs)

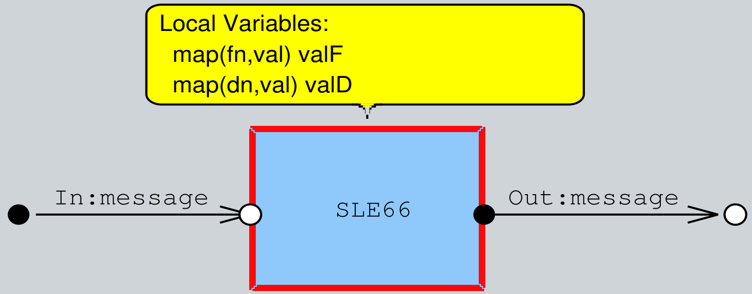
Automata with (nondeterministic) state transitions + buffered I/O, simultaneously on multiple connections.



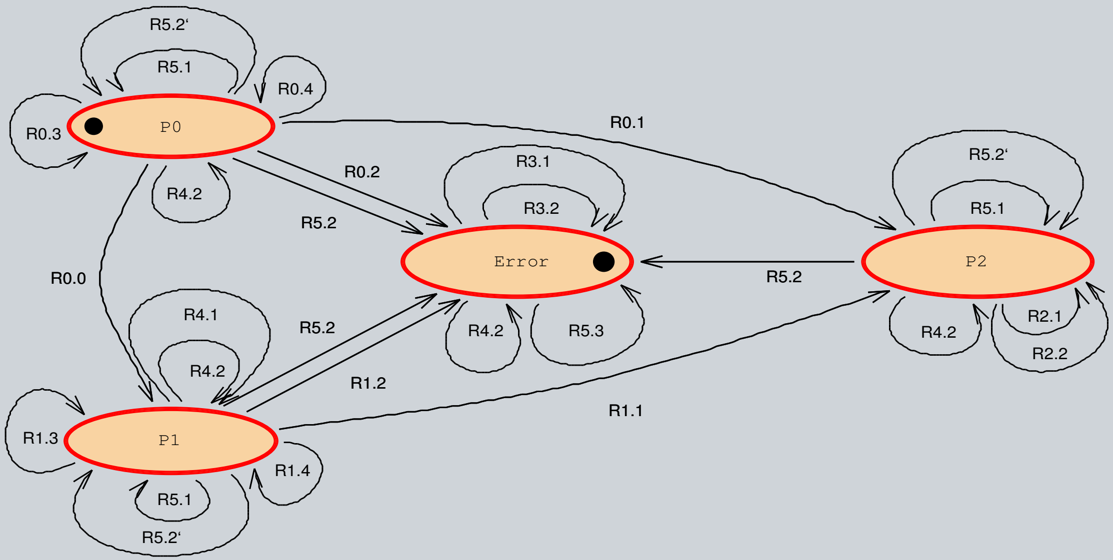
Transitions definable in executable and/or axiomatic style.
An ISM system may have changing global state.
Applicable to a large variety of reactive systems.
By now, not much verification support (theory, tools).

Model of Infineon SLE 66 Smart Card

System Structure Diagram:



State Transition Diagram (abstracted):



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

RBAC of Complex Information System

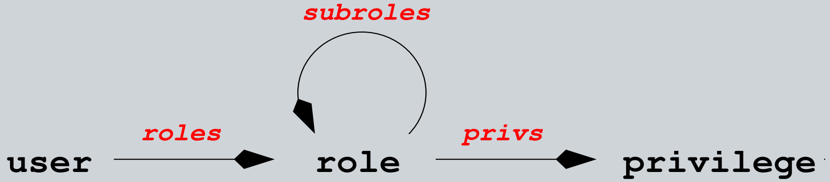
Is the security design (with emergency access etc.) sound?

Privileges:

$roles \subseteq \texttt{user} \times \texttt{role}$

$subroles \subseteq \texttt{role} \times \texttt{role}$

$privs \subseteq \texttt{role} \times \texttt{privilege}$



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

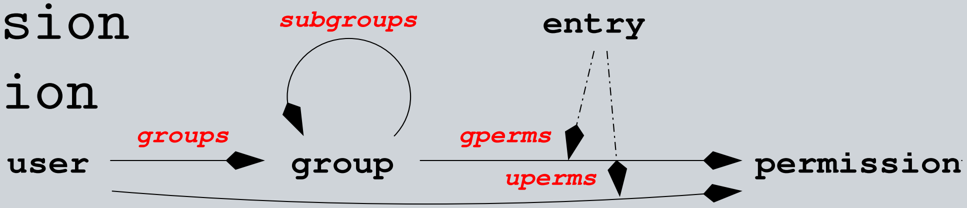
Permissions:

$groups \subseteq \texttt{user} \times \texttt{group}$

$subgroups \subseteq \texttt{group} \times \texttt{group}$

$gperms \subseteq \texttt{group} \times \texttt{permission}$

$uperms \subseteq \texttt{user} \times \texttt{permission}$



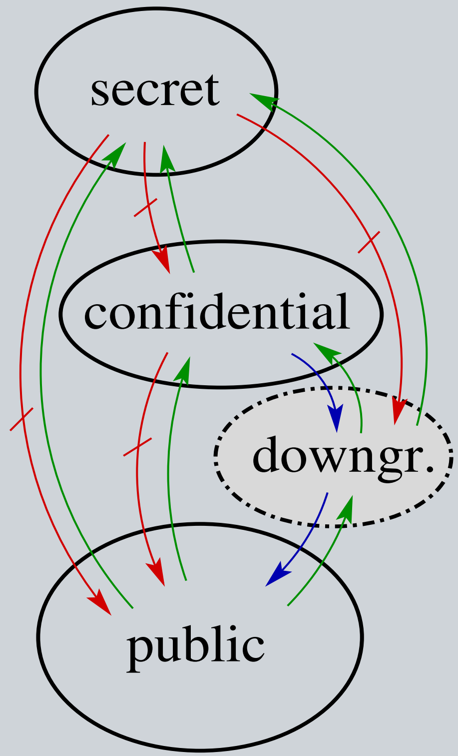
$$(u, p) \in (groups \circ subgroups^* \circ gperms(e)) \cup uperms(e)$$

“nagging questions” $\leadsto$ clarifications improving specification quality.

Open issue: relation between model and implementation ($\leadsto$ 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

In progress: connection with ISMs

Cryptoprotocol models

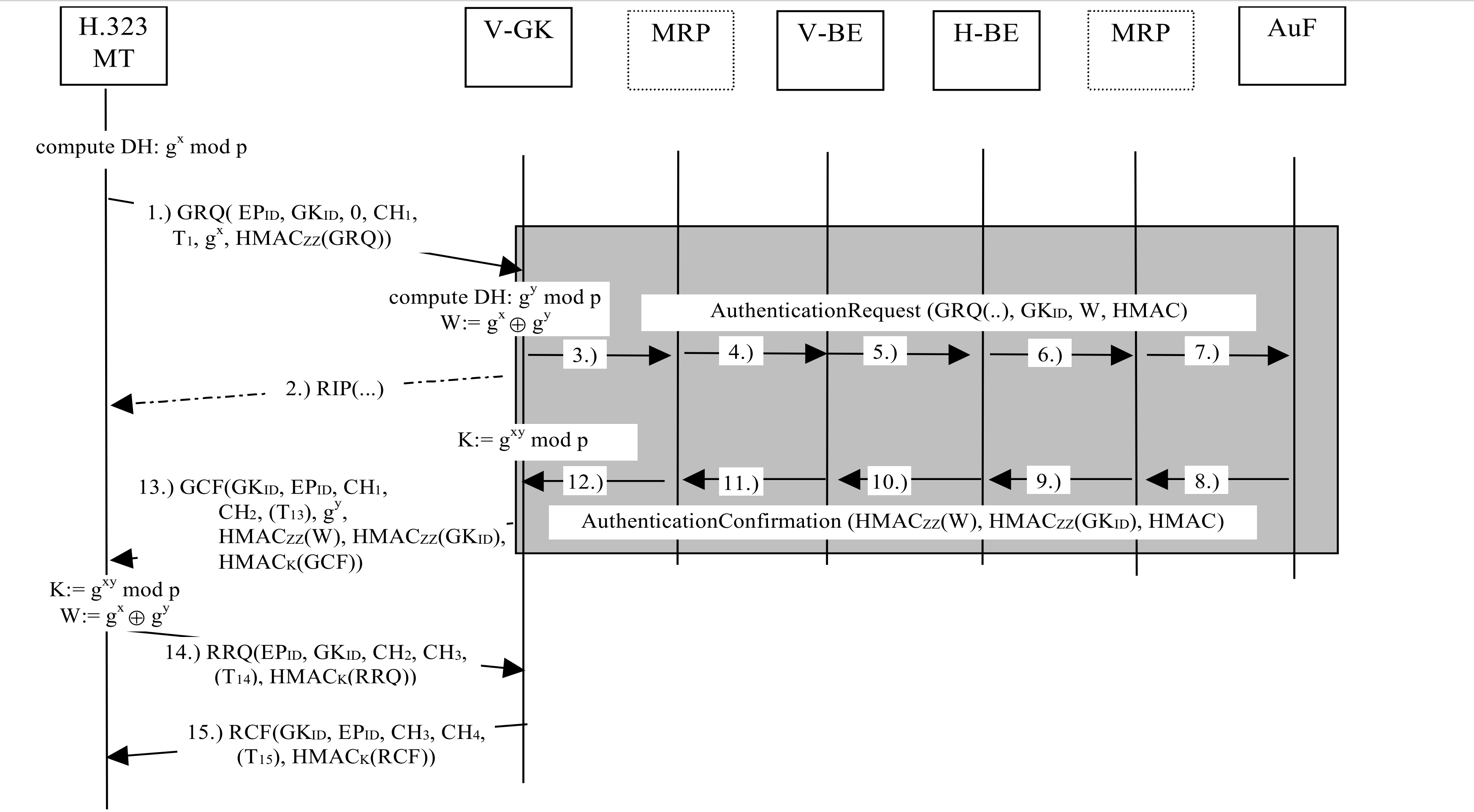
- Describe message exchange between processes or principals

- 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 , ...

SIEMENS

H.530 Mobile Roaming Authentication



Two vulnerabilities found and corrected. Solution standardized.

SIEMENS

Shaping a Formal Model

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

Overview

Security Certification

Conclusion

Certification Goals & General Approach

Goal: gain confidence in the security of a system

- What are the goals to be achieved?
- Are the measures employed appropriate to achieve the goals?
- Are the measures implemented correctly?

Approach: assessment of system security by neutral experts

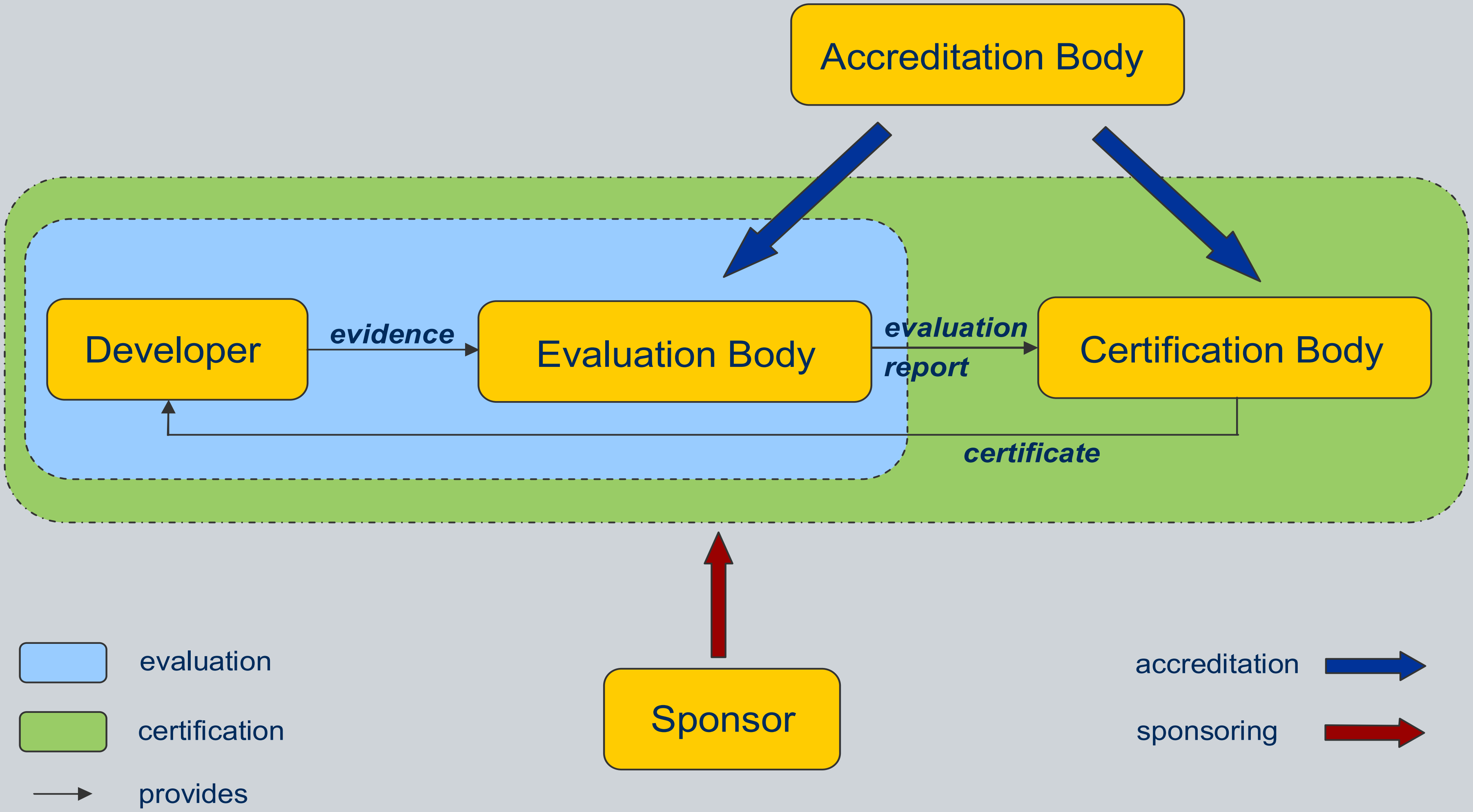
- Understanding the security functionality of the system
- Gaining evidence that functionality is correctly implemented
- Gaining evidence that the integrity of the system is kept

Result: Successful evaluation is awarded a certificate

Common Criteria

- international standard
  - Version 2.1: ISO/IEC 15408:1999
  - Version 3.1: ISO/IEC 15408:2006
- generic approach
  - full range of IT systems
  - scalable level of assurance

Process Scheme



Security Target

- Definition of the Target of Evaluation (TOE) and separation from its environment
- Definition of the security threats and objectives for the TOE
- Introduction of TOE Security Functions (TSF): measures intended to counter the threats
- Determination of Evaluation Assurance Level (EAL)

⇒ The Security Target is the central document to which all subsequent evaluation activities and results refer!

⇒ Interpretation of results is only reasonable wrt. Security Target

Evaluation Assurance Levels

EAL1: functionally tested

EAL2: structurally tested

EAL3: methodically tested and checked

EAL4: methodically designed, tested, and reviewed,

EAL5: semiformally designed and methodically tested
including formal security policy model

EAL6: semiformally verified design and methodically tested

EAL7: formally verified design and methodically tested

Increasing requirements on scope, depth and rigor

EAL example: EAL5

In red: additional requirements compared to EAL4

- Complete source code is subject to analysis
- Formal security policy model
- Semiformal description techniques
- Modular design
- Documentation of developer's tests up to low-level design
- Vulnerability analysis refers to moderate attack potential
- Covert channel analysis
- Comprehensive configuration management

How to scale an Evaluation

- Separation of TOE and TOE environment
- Detail level of TOE summary specification
- Definition of security objectives
- Definition of security functional requirements
- Strength-of-function claims
- EAL selection

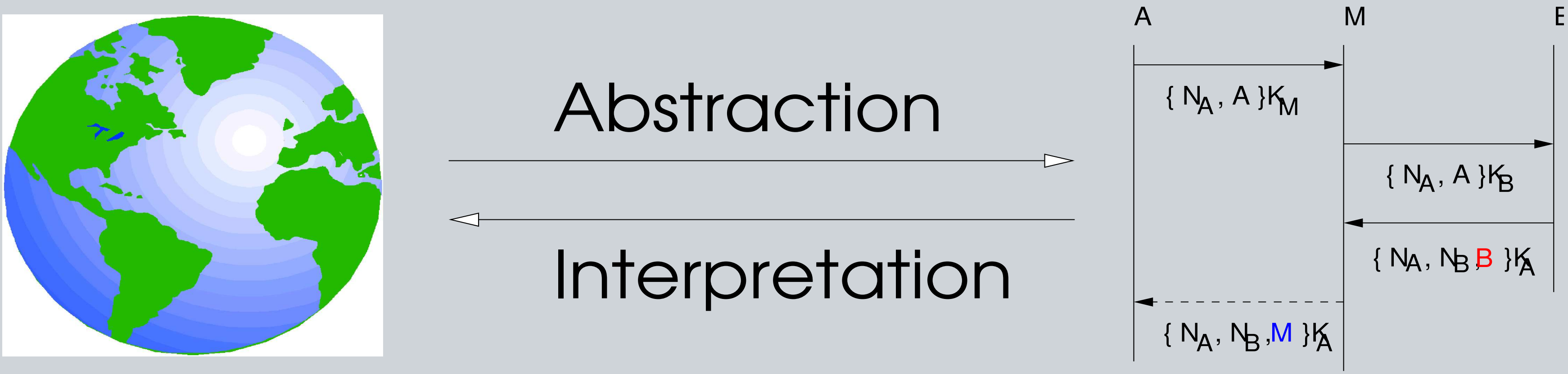
Overview

Security Certification

Conclusion

Benefits of Formal Security Analysis

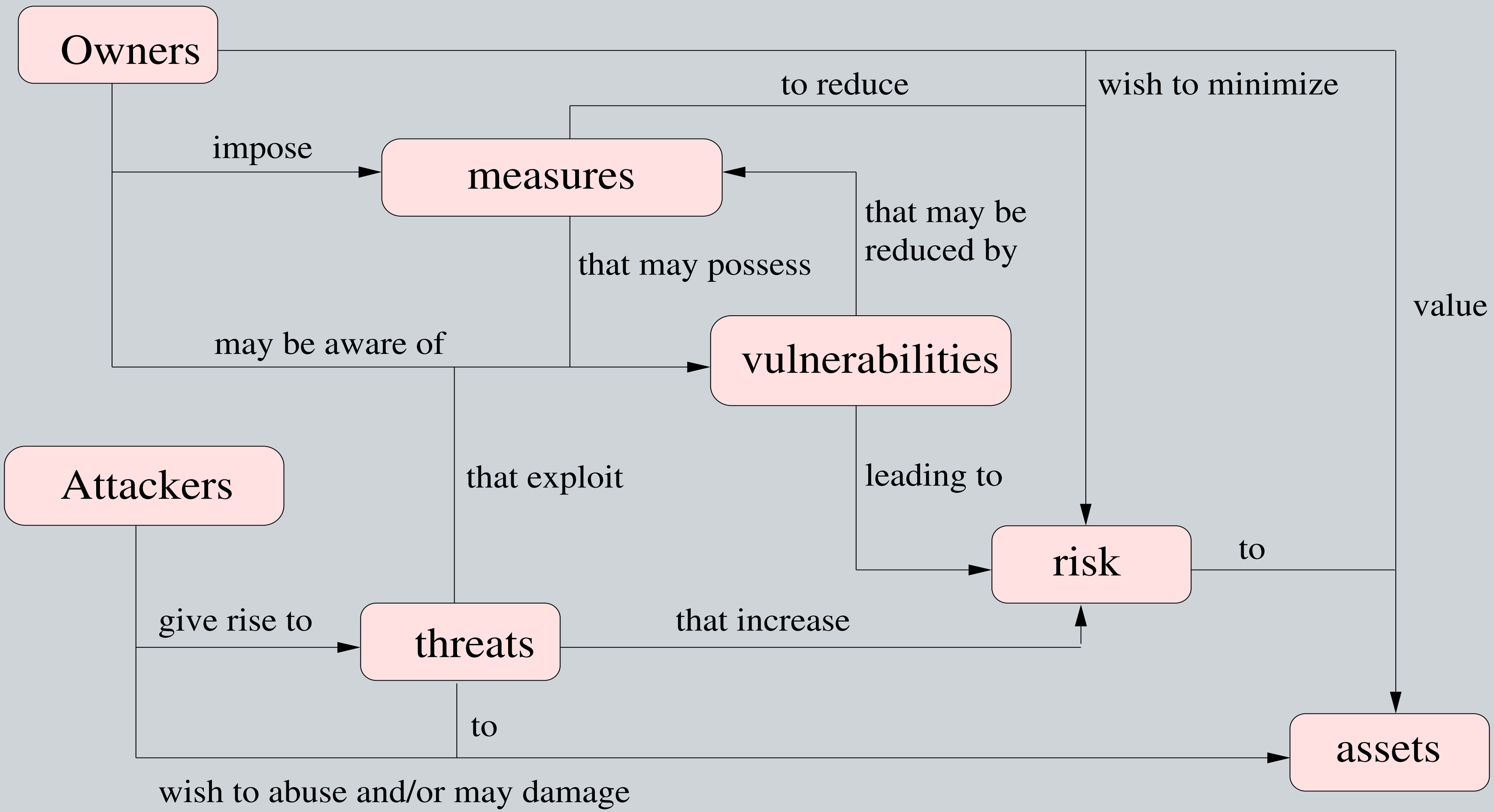
A formal security model of a system is an abstract description with mathematical precision focusing on the relevant security issues.



- provides systematic description with powerful abstractions improving understanding of security issues
- prevents ambiguities, incompleteness, and inconsistencies enhancing the quality of specifications
- provides basis for systematic testing or even formal verification assuring the effectiveness of security measures

Backup slides

Security Concepts and Relationships



Policy (here implicit) defines authorized actions on assets,
i.e., what constitutes legal use (or abuse/damage, respectively).

Goals, Threats, and Mechanisms

Standard breakdown in security engineering:

Goals/Objectives: What to achieve

Threats: Which attacks to counter

Mechanisms: How to achieve goals

Required for certification according to
e.g. ITSEC and Common Criteria

SIEMENS

Security Goals

- Goals: **CIA**

  Confidentiality No unauthorized information disclosure/leakage
  Integrity: No unauthorized modification of information
  Availability: No unauthorized impairment of functionality

  All these require authorization
  = authentication + access control.

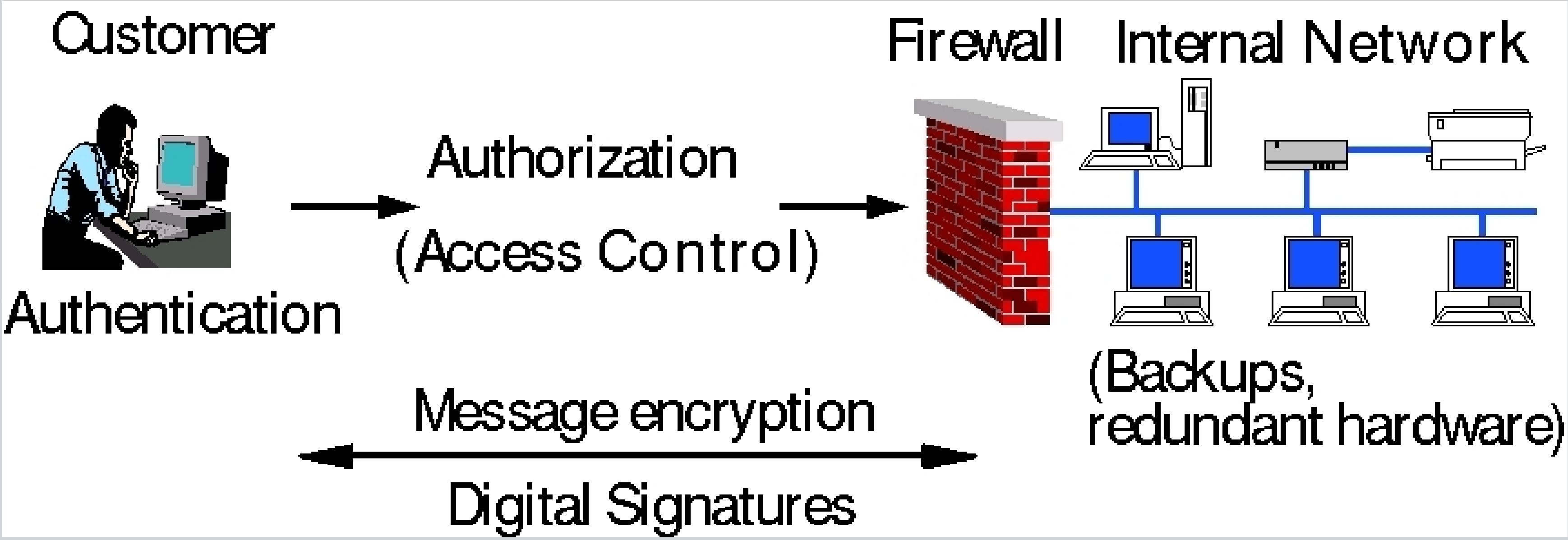
- Other goals

  Privacy: User data is only exposed in permitted ways.
  Nonrepudiation: One cannot deny responsibility for actions.
  Also called accountability
  Application specific requirements and combinations,
  e.g. e-voting

Threats

Confidentiality	Interception	Unauthorized party gains access to information
Integrity	Fabrication	Generation of additional data or activities
Integrity	Modification	Unauthorized tampering of data or services
Integrity, Availability	Interruption	Service or data becomes unavailable or unusable

Security Mechanisms



- Various mechanisms are used to achieve goals.
- Designing adequate mechanisms is challenging.
- One must be cognizant of the tradeoffs and costs involved.

What are Formal Methods?

- A language is formal if it has a well-defined syntax and semantics.
  Examples: Predicate logic, automata, λ-calculus, process algebra, ...
- A model is formal if it is specified with a formal language.
  Example:

$$\forall x.\ bird(x) \longrightarrow flies(x) \qquad bird(tweety)$$

- A proof is formal if it is done using a deductive system
  (i.e., a set of precise rules governing each proof step).
  Examples: Tableau calculus, axiomatic calculus, term rewriting, ...
- A formal proof is machine-assisted if
  it is performed, or at least checked, by an IT system.
  Examples: OFMC (model checker), Isabelle (theorem prover)

Development Phases and the Benefits of Formal Modeling

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

- perfect reference for testing and verification

