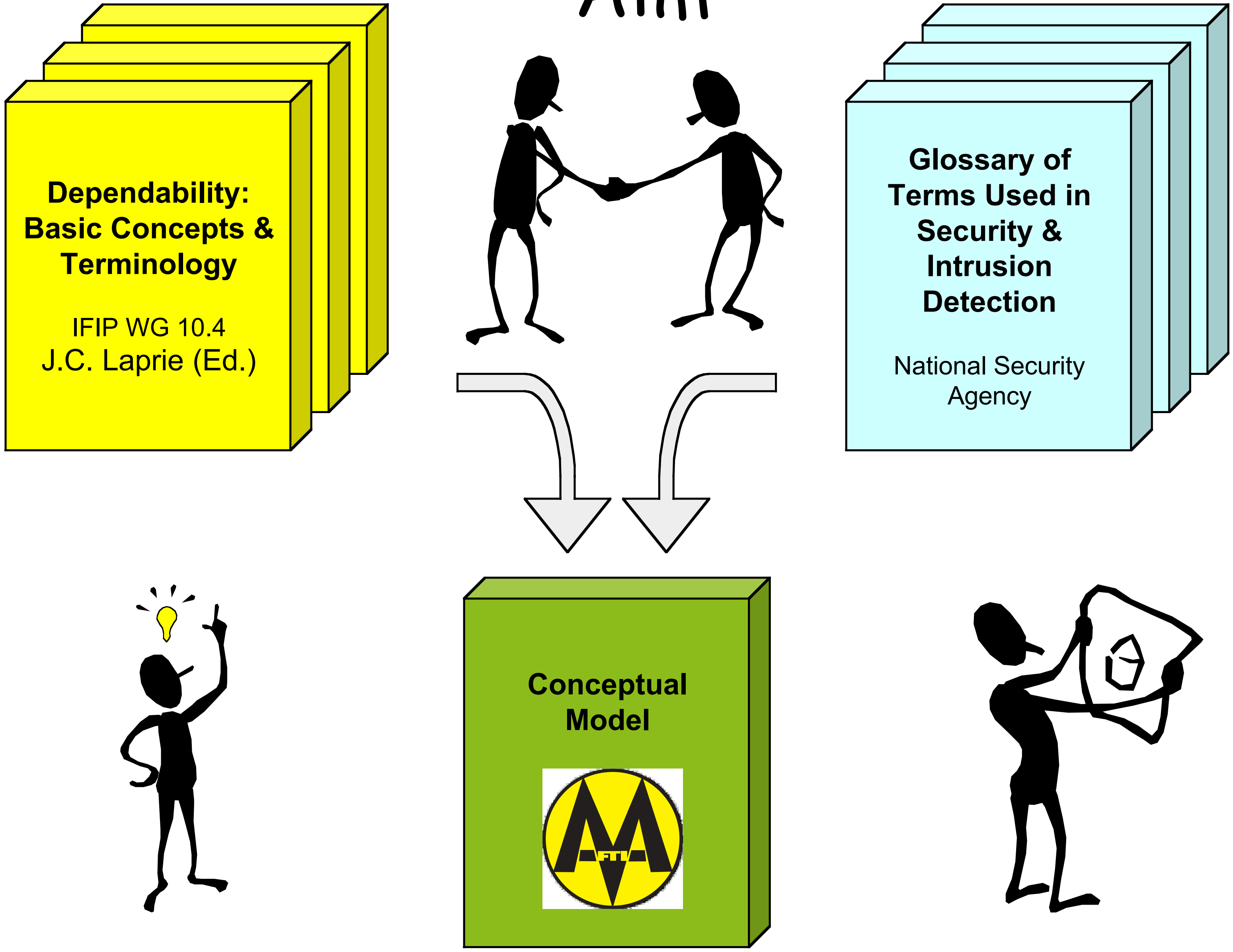
# Dependability Concepts for Malicious Faults

*David Powell & Yves Deswarte*

Dependability & Survivability Workshop
IFIP WG 10.4 meeting, Hilton Head Island, SC, USA
27 June - 1 July 2002

# Aim



Dependability: Basic Concepts & Terminology

IFIP WG 10.4
J.C. Laprie (Ed.)

Glossary of Terms Used in Security & Intrusion Detection

National Security Agency

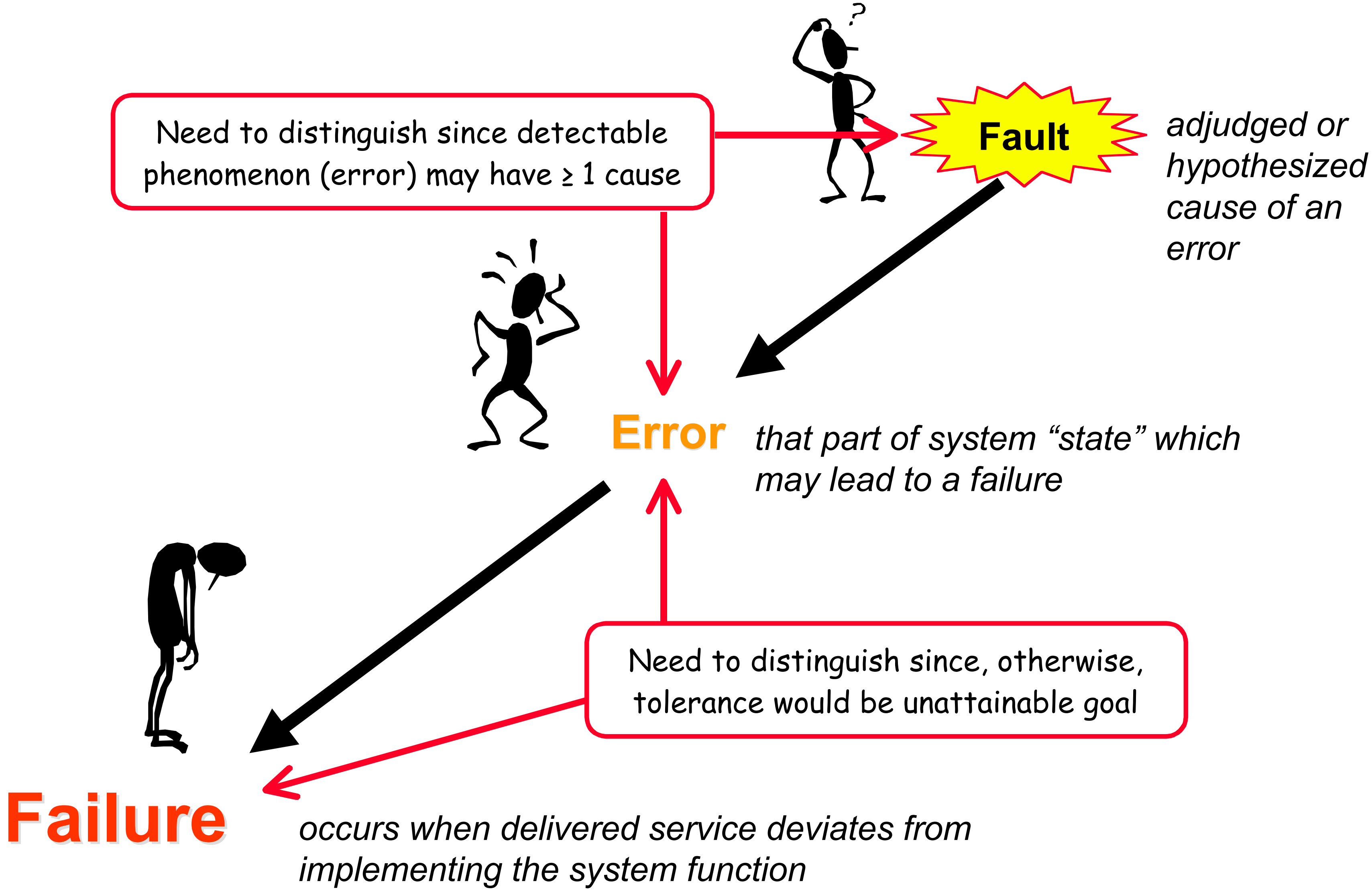
Conceptual Model

http://www.research.ec.org/maftia/

# Summary

- Causal chain of threats
- Security policy and security failure
- Intrusion, attack and vulnerability
- Security methods
- Intrusion detection
- Fault tolerance
- Integrated intrusion detection/tolerance framework

# Causal Chain of Threats



Need to distinguish since detectable phenomenon (error) may have ≥ 1 cause

**Fault**

*adjudged or hypothesized cause of an error*

**Error**

*that part of system "state" which may lead to a failure*

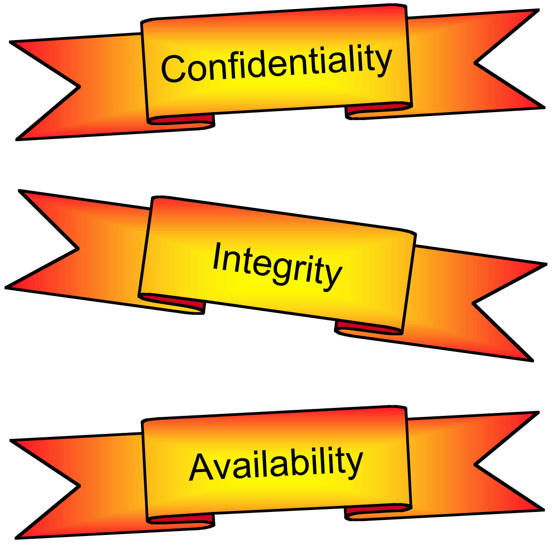
Need to distinguish since, otherwise, tolerance would be unattainable goal

**Failure**

*occurs when delivered service deviates from implementing the system function*

# Security Policy

- Security properties which are to be fulfilled by the system

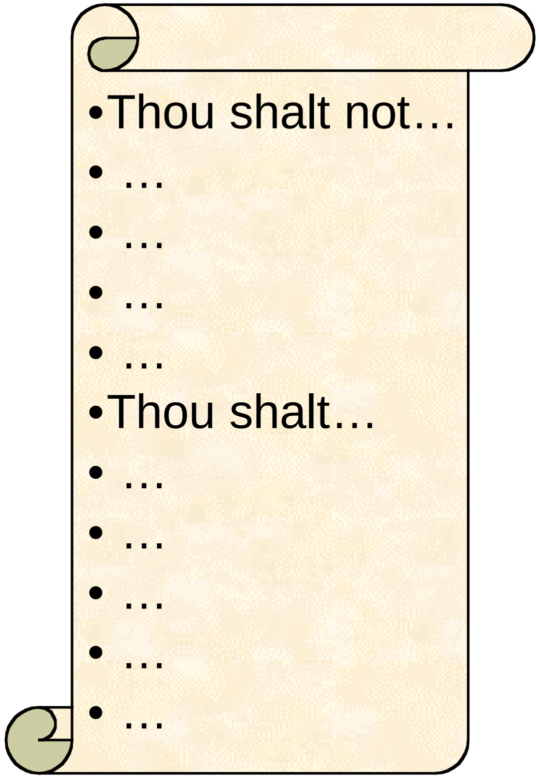


Confidentiality

Integrity

Availability

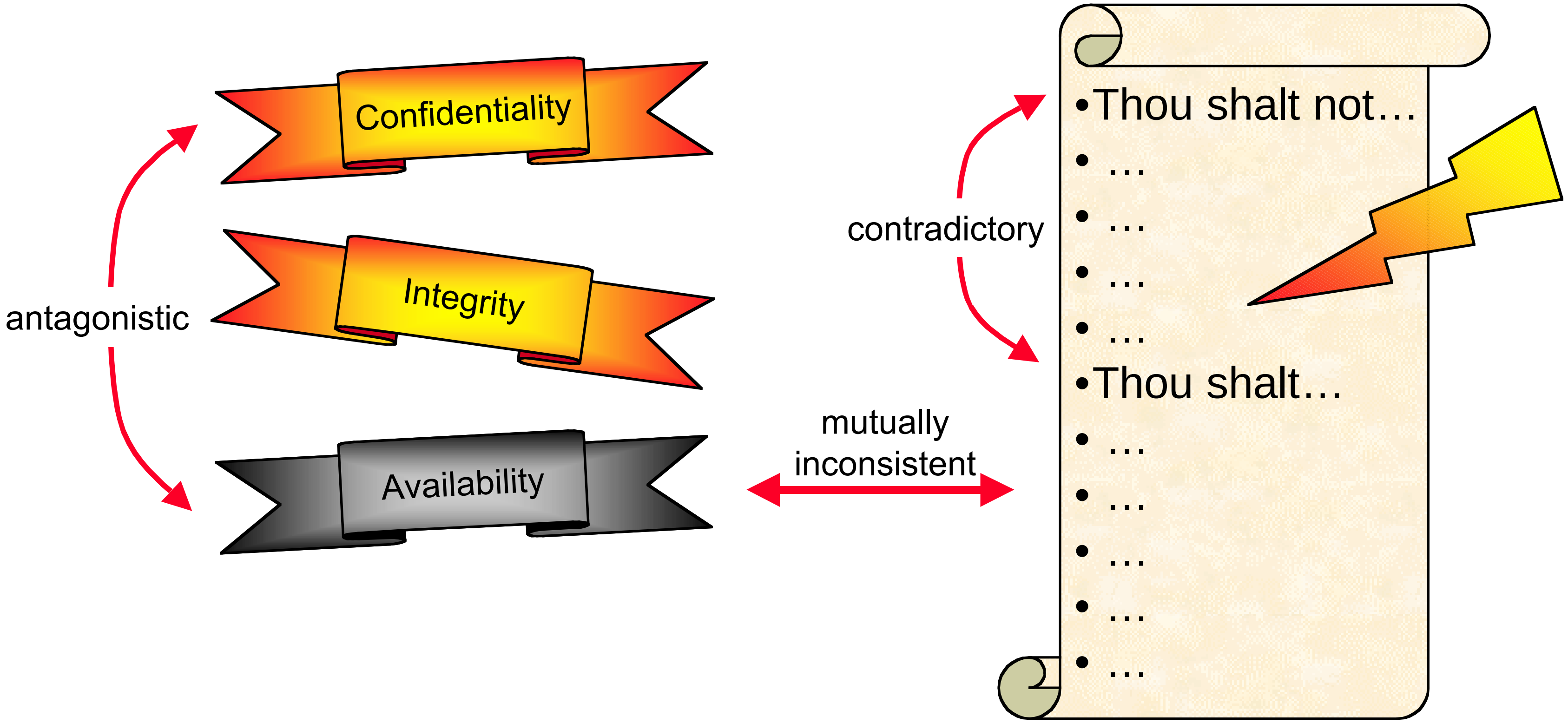
- Rules according to which the system security state may evolve



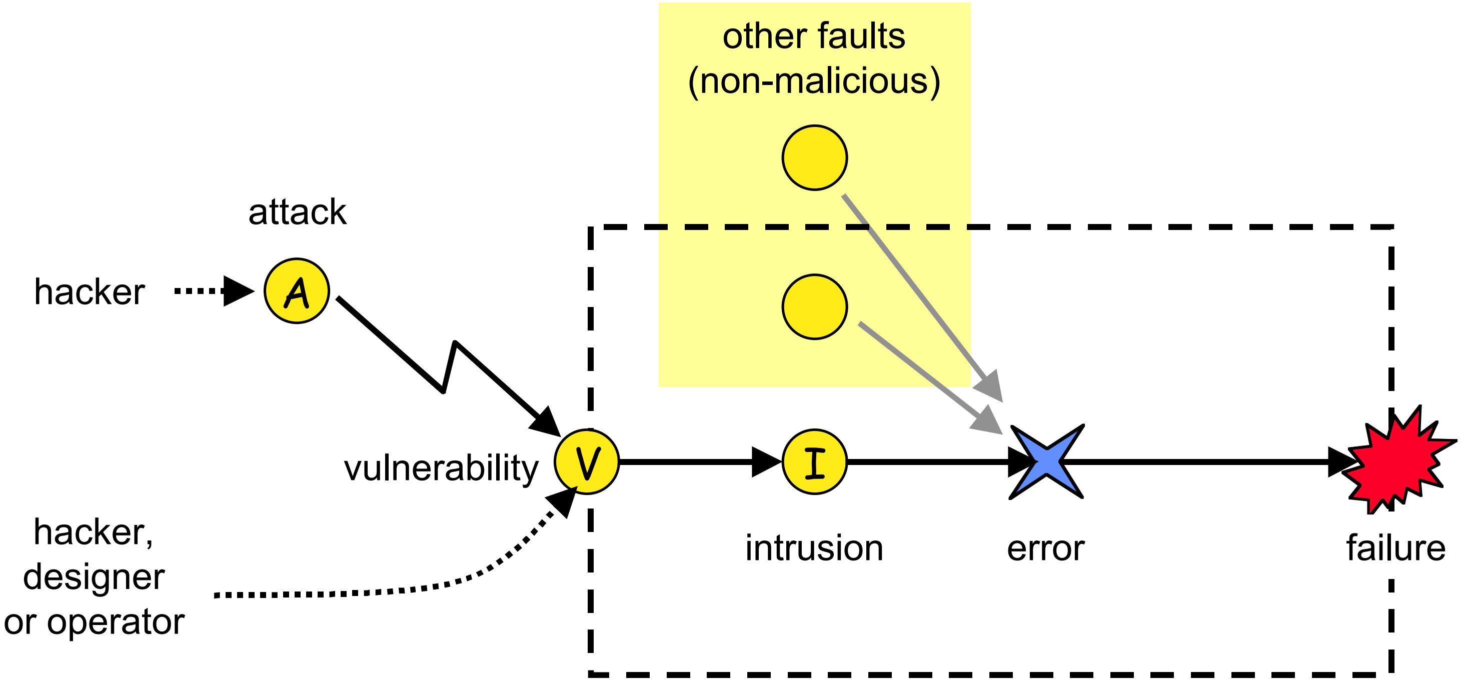
- Thou shalt not…
  - …
  - …
  - …
  - …
- Thou shalt…
  - …
  - …
  - …
  - …
  - …

# Security Failure

- Violation of a security property of intended security policy

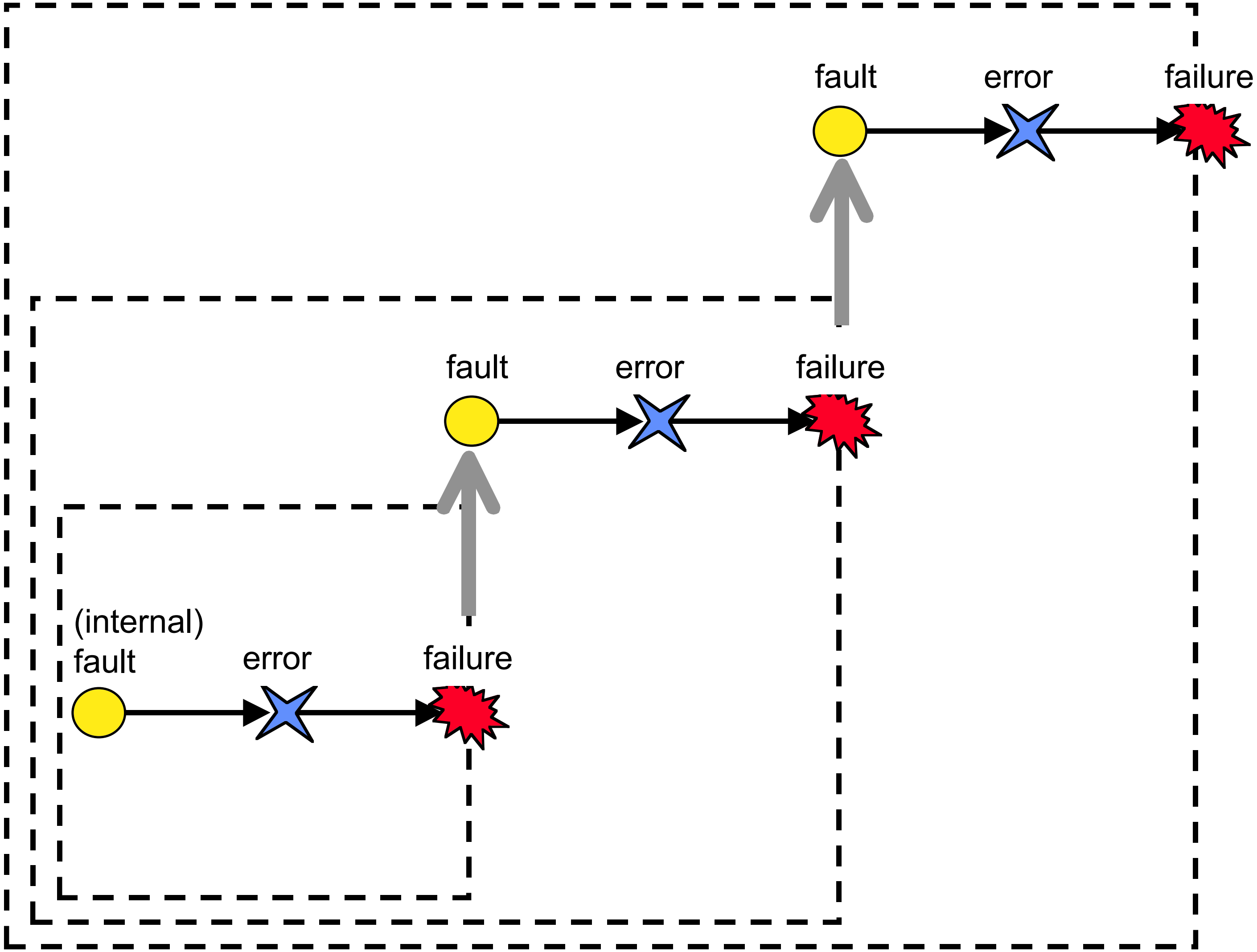


# Fault Model

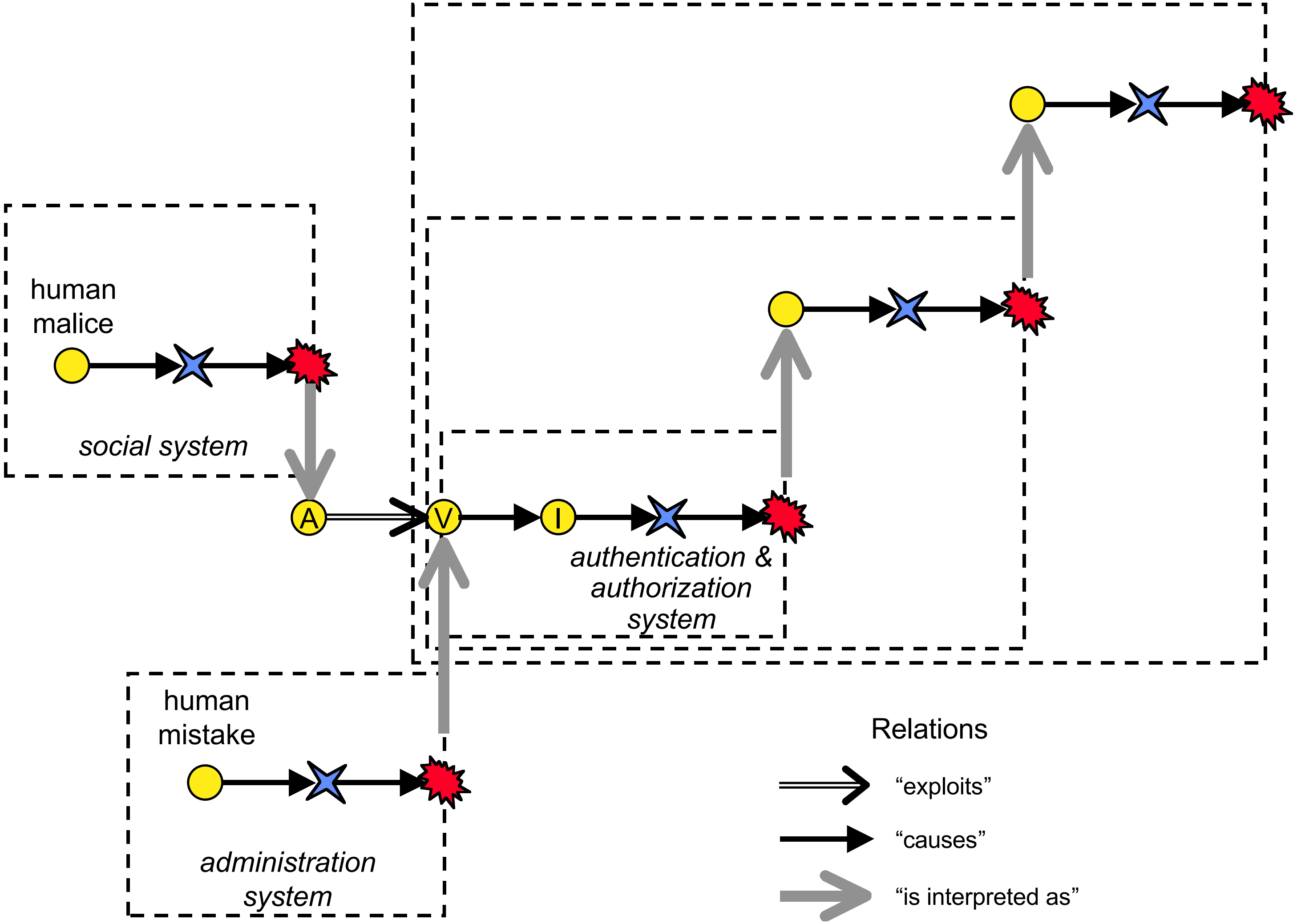


- **attack** - malicious external activity aiming to intentionally violate one or more security properties; an *intrusion* attempt
- **vulnerability** - a malicious or non-malicious fault, in the requirements, the specification, the design or the configuration of the system, or in the way it is used, that could be exploited to create an *intrusion*
- **intrusion** - a malicious interaction fault resulting from an *attack* that has been successful in exploiting a *vulnerability*

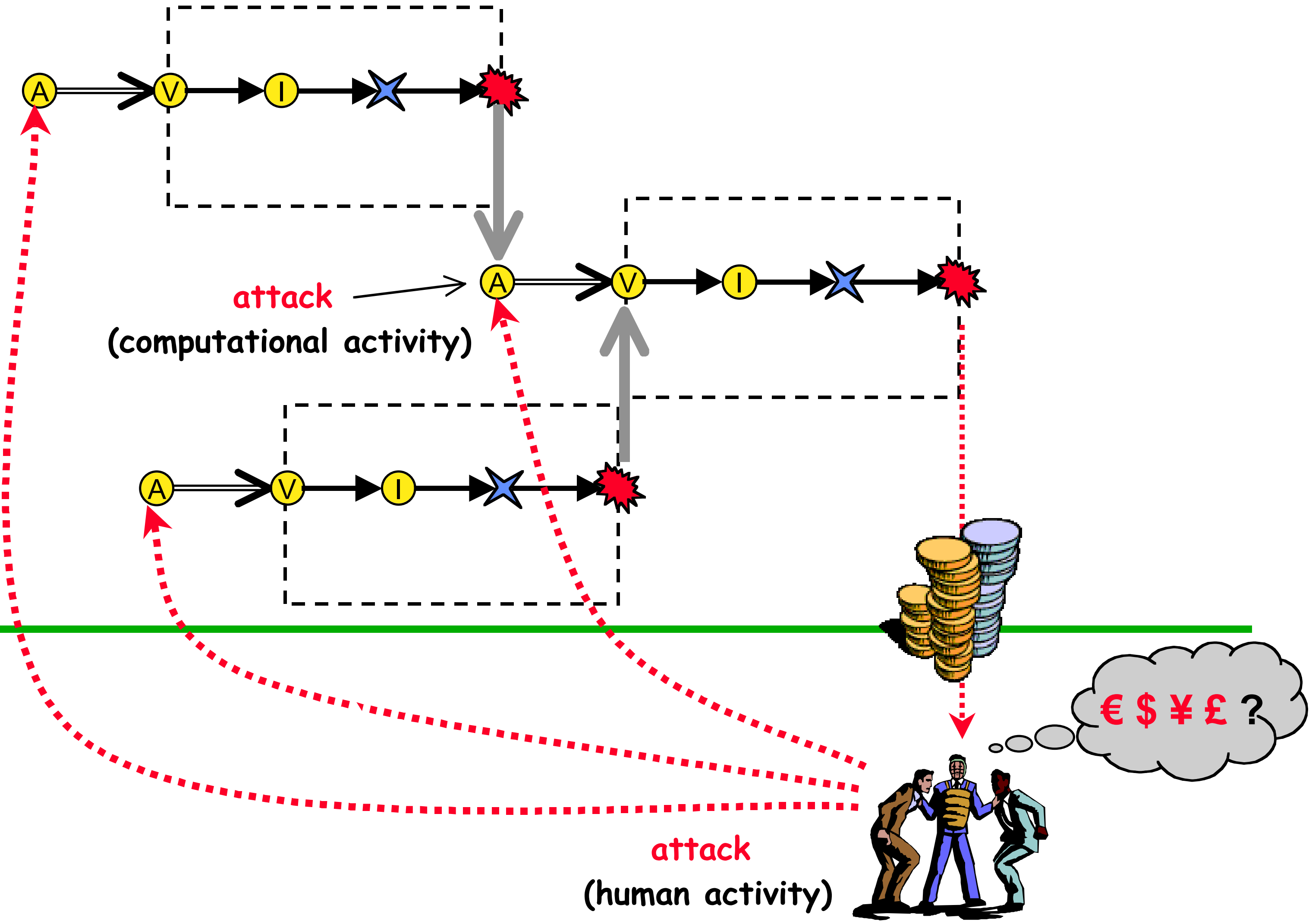
# Fault Model: Recursion



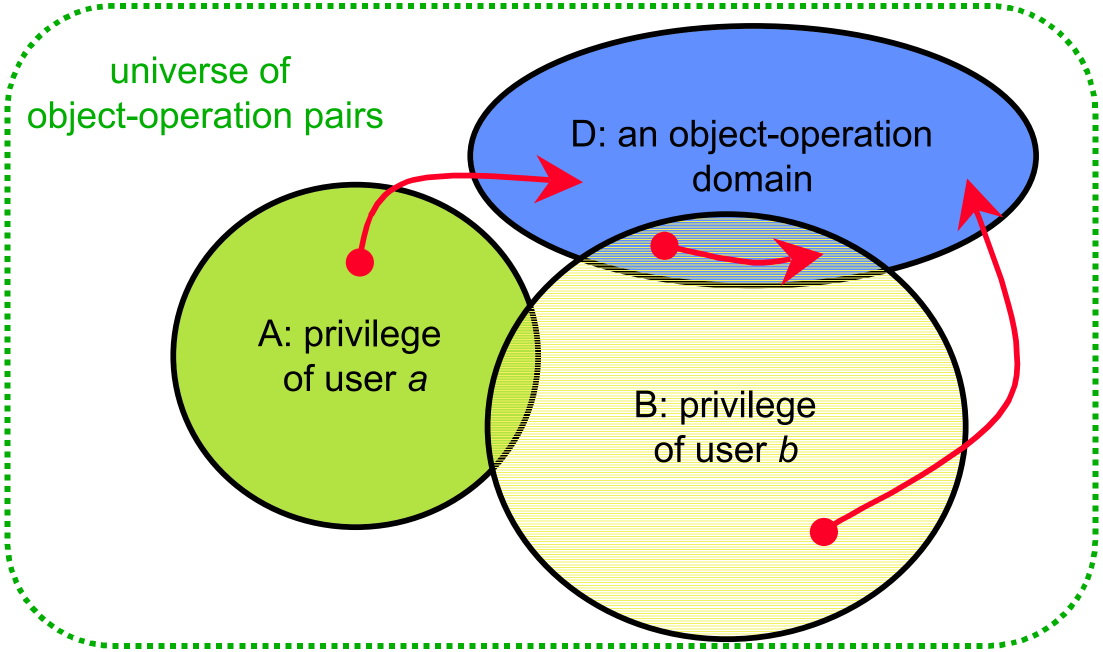
# Fault Model: Recursion?



# Fault Model: Propagation?

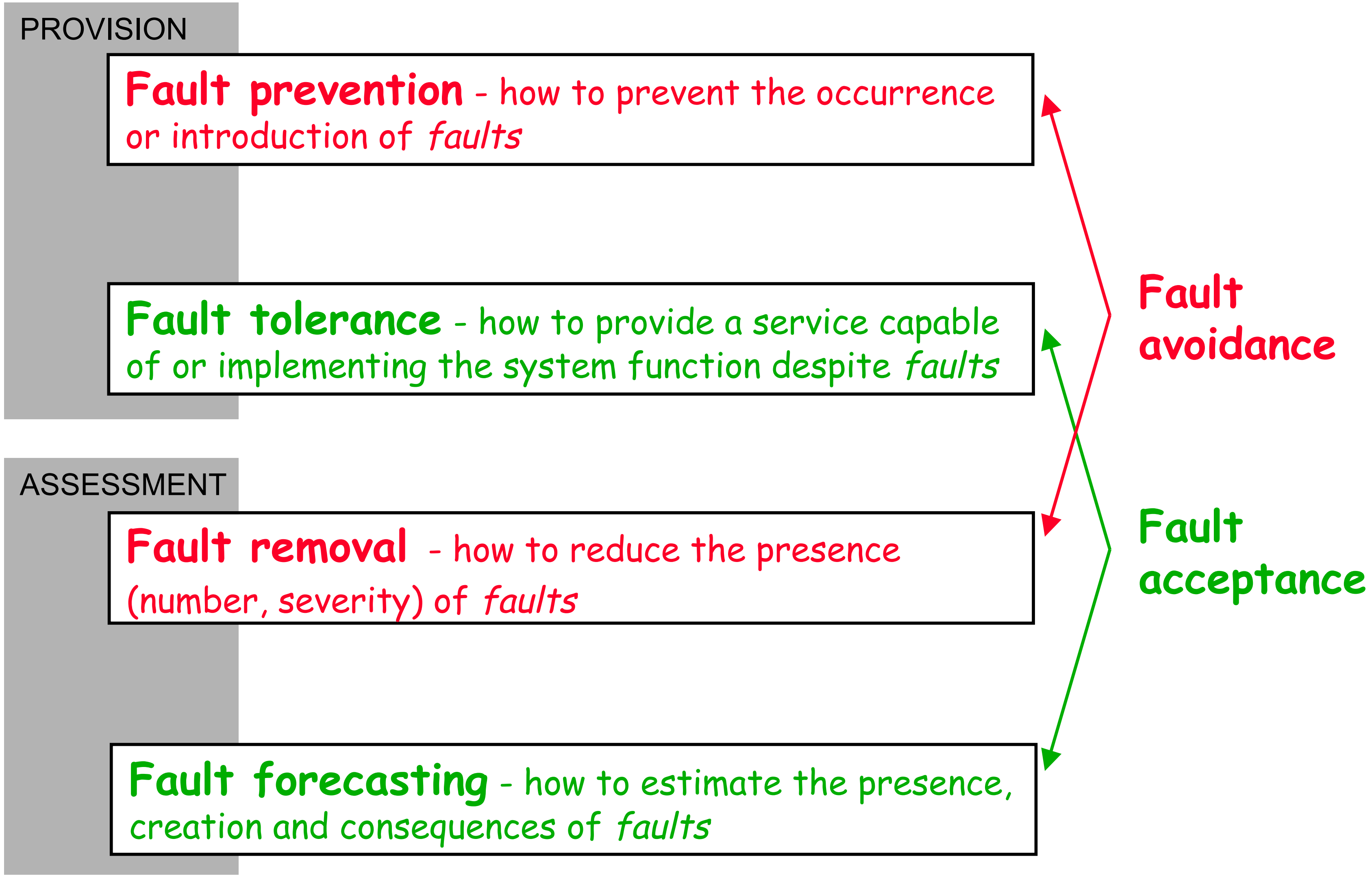


# Outsiders or Insiders: Privilege



- **Theft of privilege**: unauthorized increase in privilege
- **Abuse of privilege**: improper use of authorized operations
- **Outsider**: current privilege does not intersect considered domain
- **Insider**: current privilege intersects considered domain

# Dependability Methods



PROVISION

**Fault prevention** - how to prevent the occurrence or introduction of *faults*

**Fault tolerance** - how to provide a service capable of or implementing the system function despite *faults*

ASSESSMENT

**Fault removal** - how to reduce the presence (number, severity) of *faults*

**Fault forecasting** - how to estimate the presence, creation and consequences of *faults*

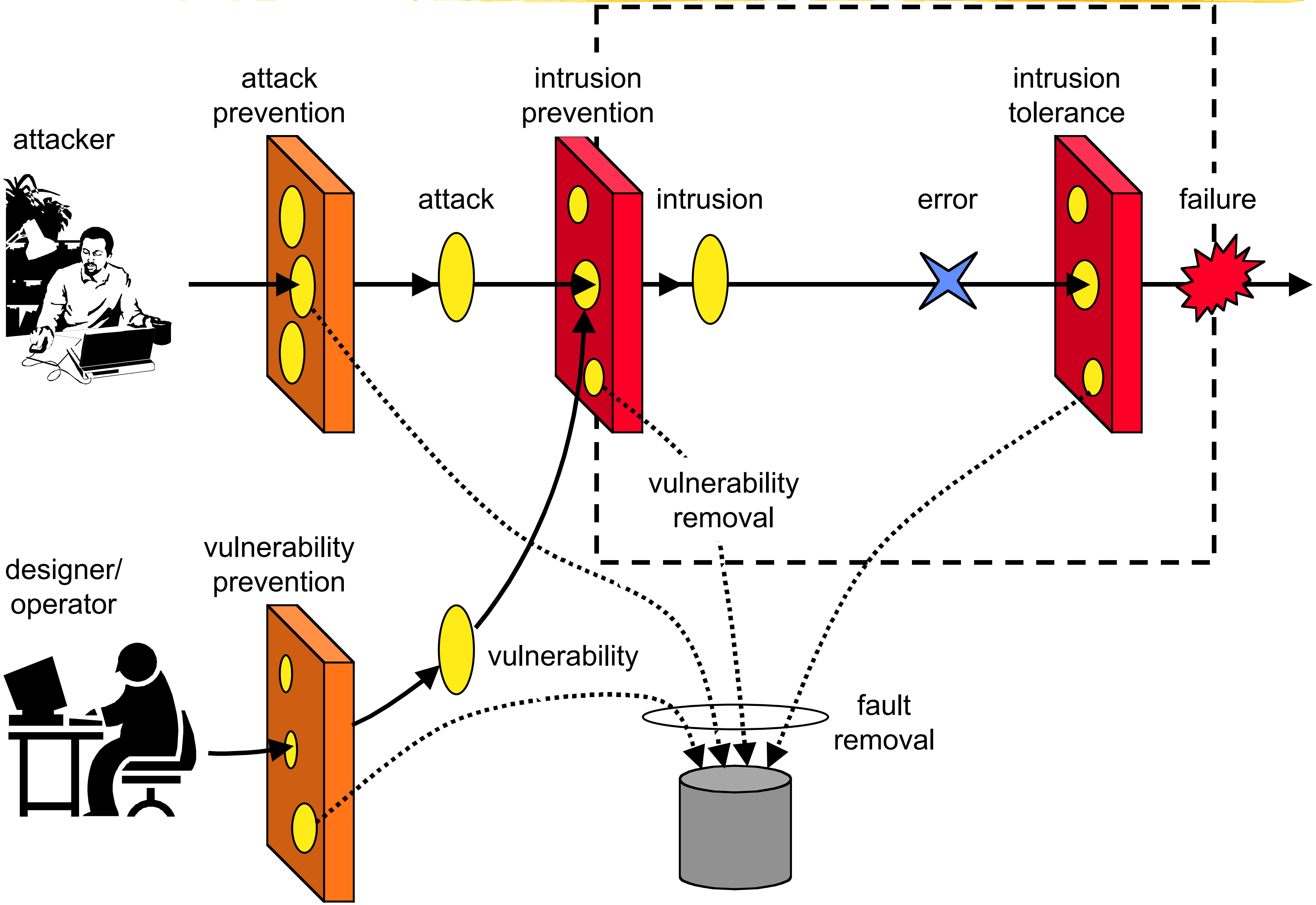
**Fault avoidance**

**Fault acceptance**

# Security Methods

| | | Attack | Vulnerability | Intrusion |
|---|---|---|---|---|
| **Prevention** | *how to prevent the occurrence or introduction of…* | deterrence, laws, social pressure, secret service… | security policy, semi-formal and formal specification, rigorous design and management… | firewalls, authentication, authorization… (+ attack prevention vulnerability prevention) |
| **Tolerance** | *how to provide a service capable of or implementing the system function despite…* | vulnerability prevention<br>vulnerability removal<br>intrusion tolerance | = intrusion tolerance | confinement, detection/recovery, masking (eg FRS), + intrusion detection for fault treatment |
| **Removal** | *how to reduce the presence (number, severity) of…* | not applicable | formal proof, model-checking, inspection, test… | not applicable |
| **Forecasting** | *how to estimate the creation and consequences of…* | intelligence gathering, threat assessment, attack warning… | assess presence of vulnerabilities, exploitation difficulty, potential consequences | vulnerability forecasting, attack forecasting |

# Prevention, Tolerance and Removal



# Intrusion Detection: Purpose

- Intrusive behavior => alarms
- Alarms
  - to a system security officer (SSO), to gather information about attacks, vulnerabilities and intrusions, and possibly to initiate manual countermeasures and/or litigation, retaliation
  - to an automatic countermeasure mechanism in order to avert security failures, i.e., to *tolerate* intrusions

- *Response* to intrusion is not part of intrusion *detection!*

# Intrusion Detection: Definition

[NSA 1998]

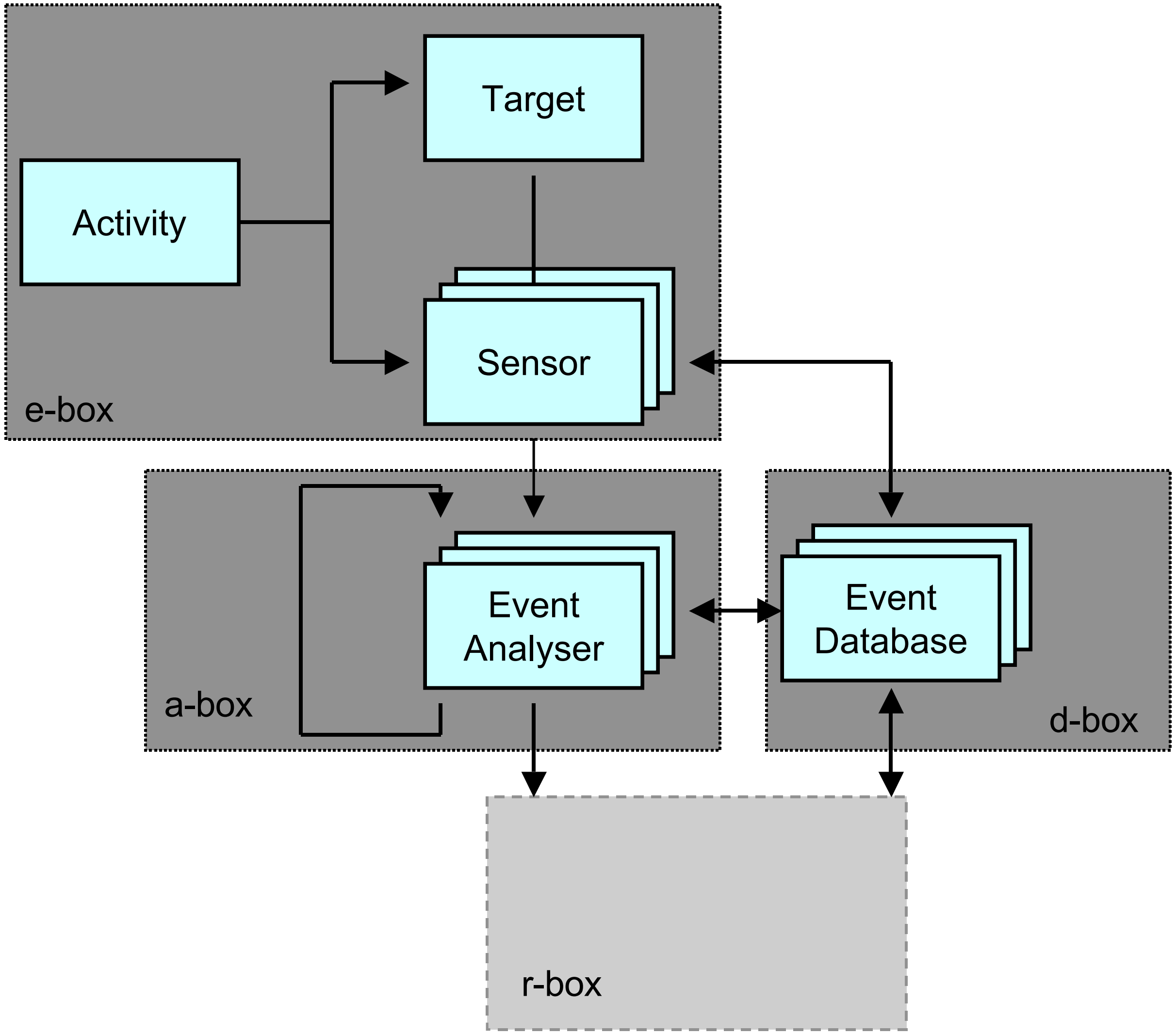
> " *Pertaining to techniques which attempt to detect intrusion into a computer or network by observation of actions, security logs, or audit data. Detection of break-ins or attempts either manually or via software expert systems that operate on logs or other information available on the network* "

**intrusion detection**: concerns the set of practices and mechanisms used towards:

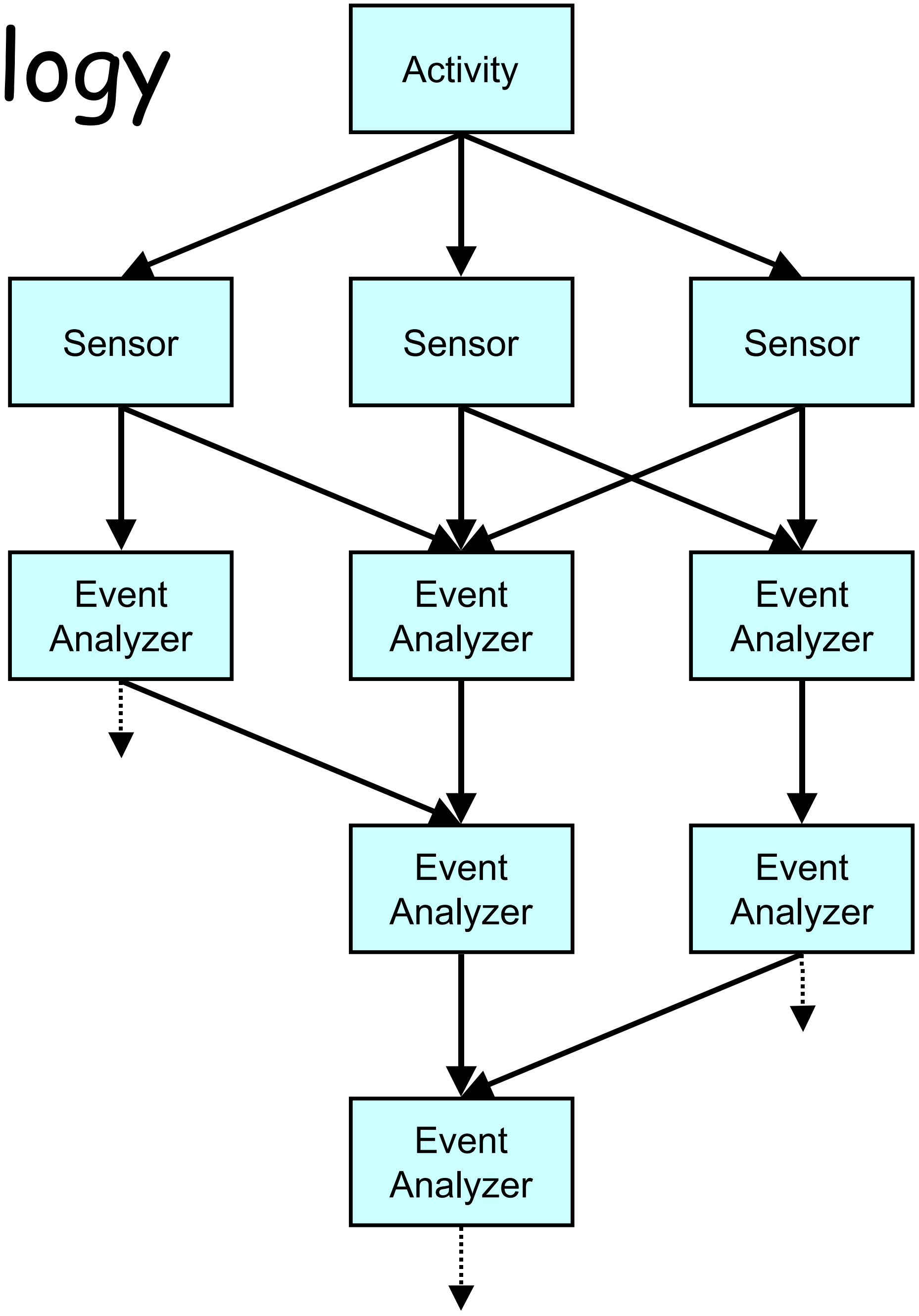
- detection of errors that may lead to security failure
- diagnosing intrusions, vulnerabilities and attacks

**intrusion detection system**: is an implementation of the practices and mechanisms of intrusion detection

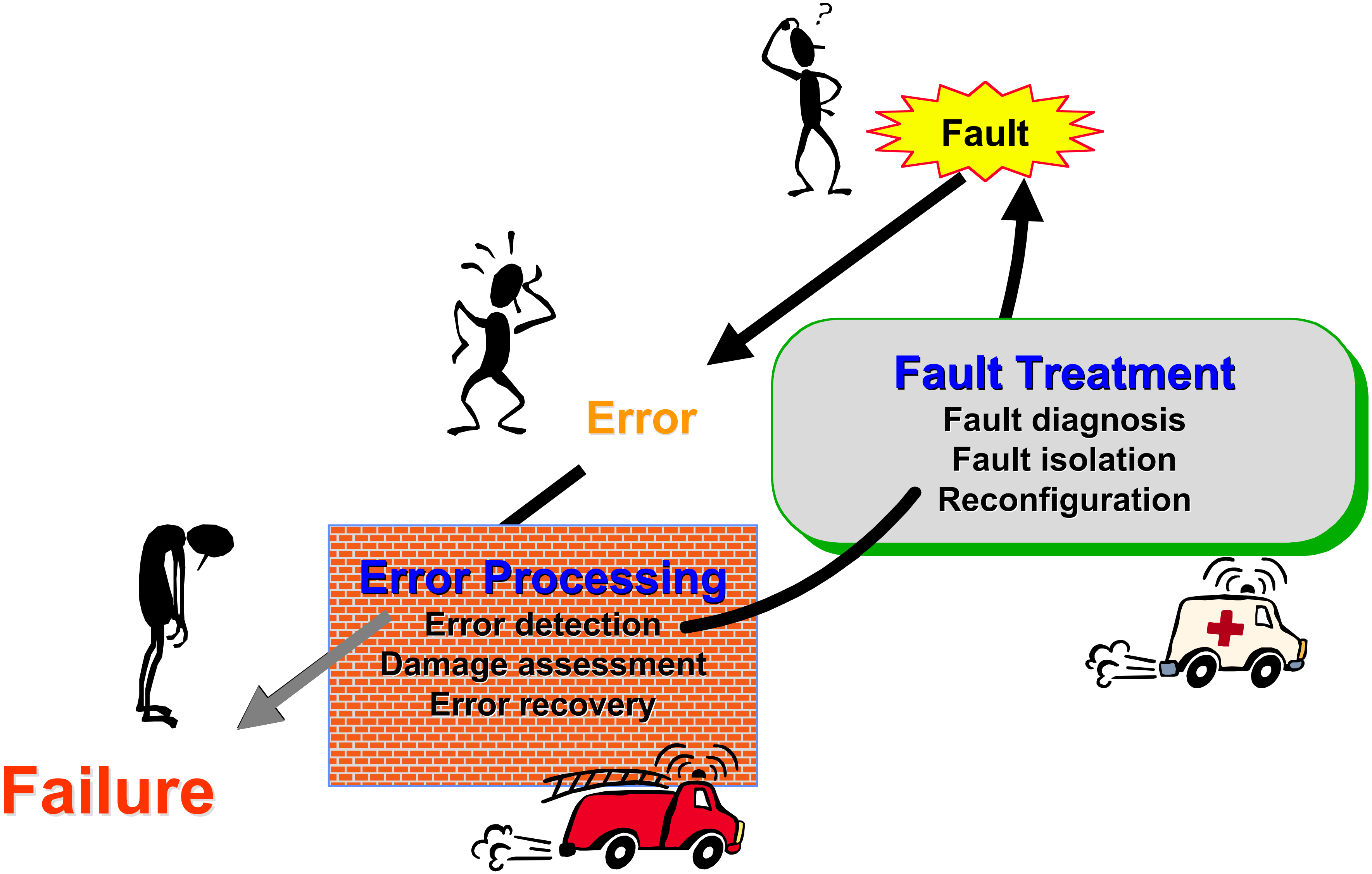
# ID Components



# ID Topology


Activity
Sensor
Sensor
Sensor
Event Analyzer
Event Analyzer
Event Analyzer
Event Analyzer
Event Analyzer
Event Analyzer

# Fault Tolerance



Fault

Error

Fault Treatment

Fault diagnosis

Fault isolation

Reconfiguration

Error Processing

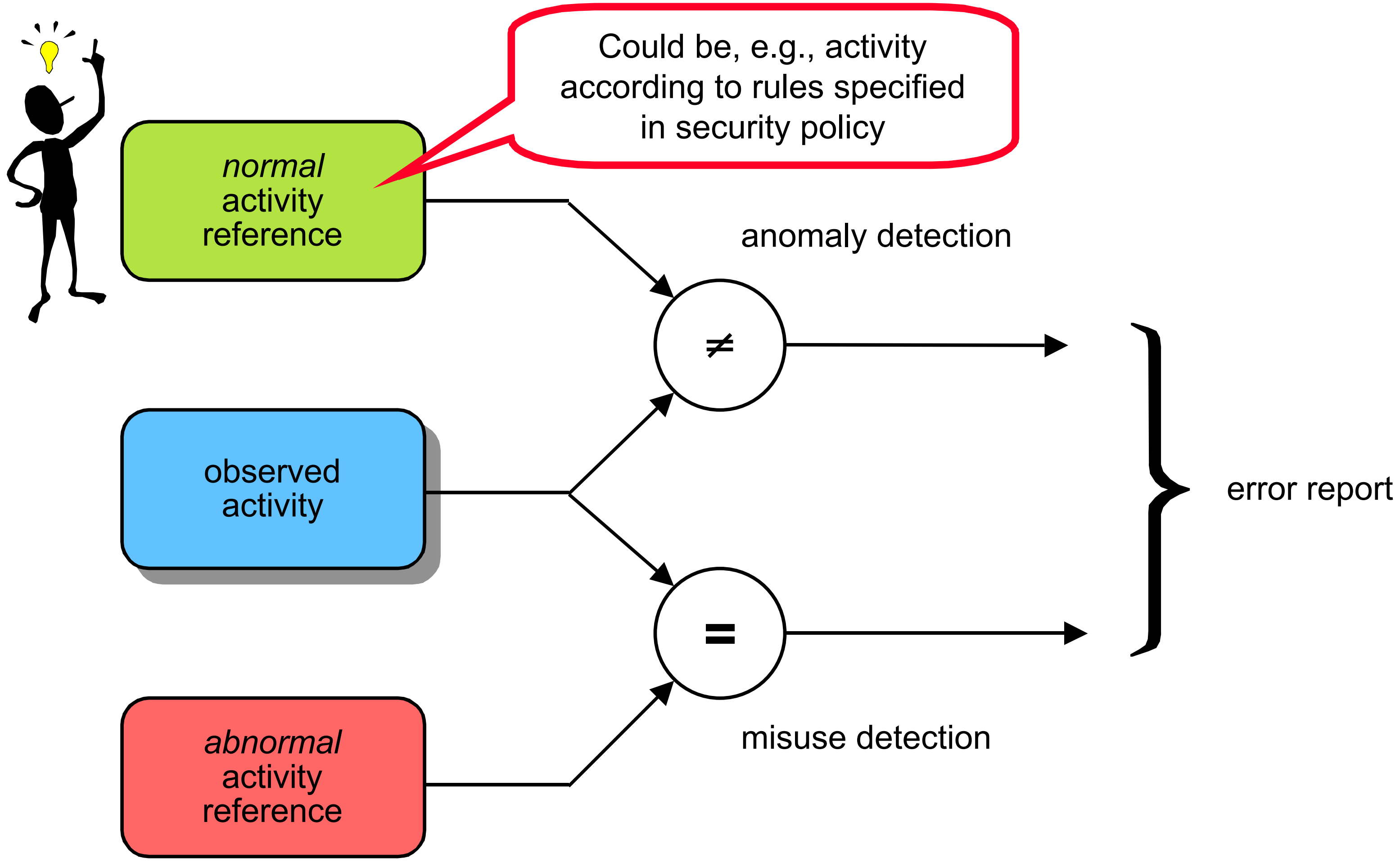
Error detection

Damage assessment

Error recovery

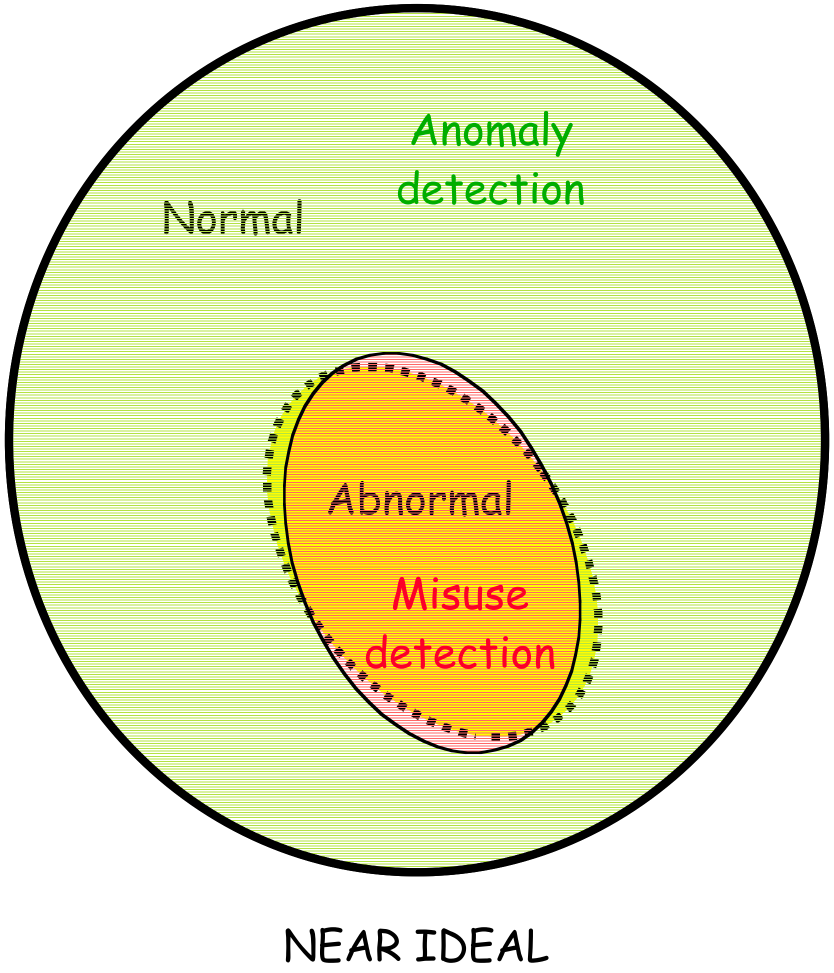
Failure

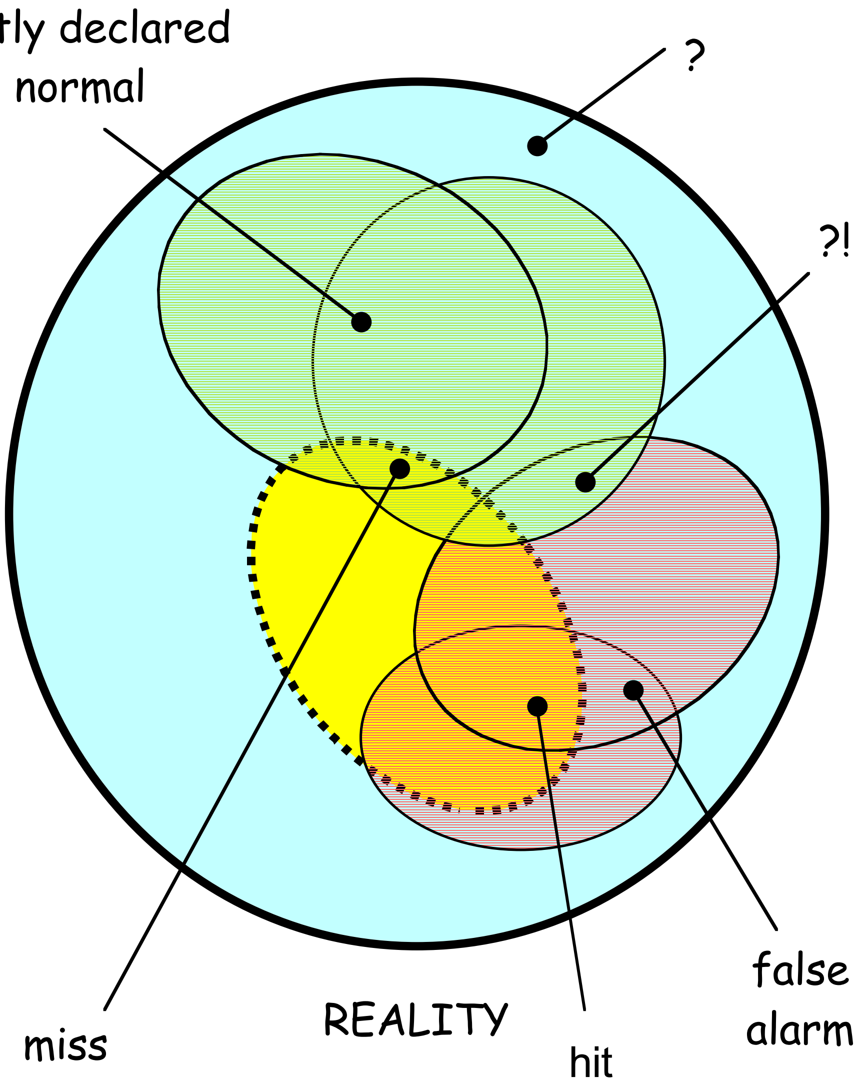
# Error Detection



Could be, e.g., activity according to rules specified in security policy

# Anomaly vs Misuse Detection





# Preemptive Error Detection

[Avizienis, Laprie & Randell 2001]

(as opposed to concurrent error detection)

- Core concepts: AKA "built-in test"
  - -> Memory scrubbing
  - -> Software rejuvenation

- Interpretation wrt malicious faults
  - Vulnerability scanning
  - Configuration checking

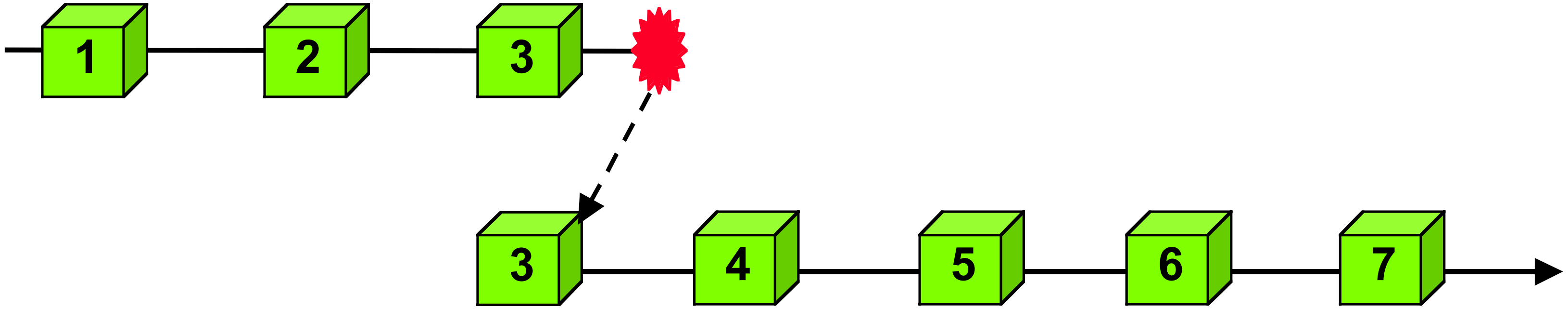
# (Damage assessment)

- Core concepts: aims to evaluate extent of error propagation before initiating recovery
  - How many checkpoints to rollback?
  - How many processes affected before detection?

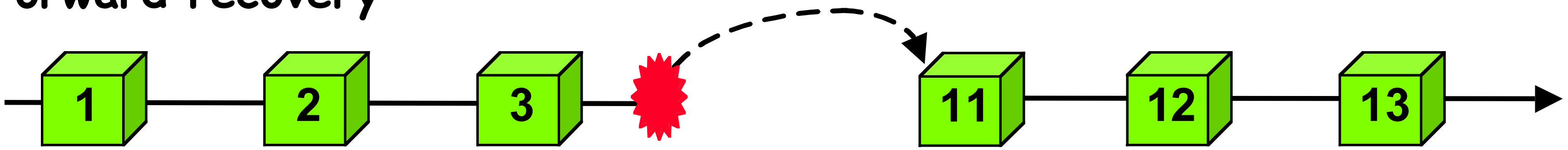
- Interpretation?
  - How many files have been corrupted by an intruder, and thus need to be restored *before use*?

# Error Recovery

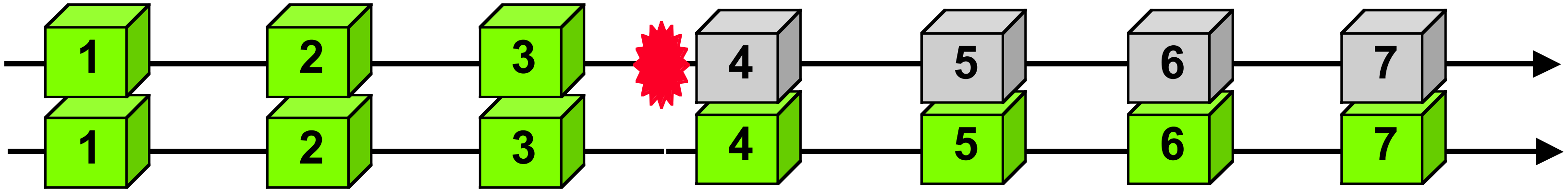
Backward recovery



Forward recovery



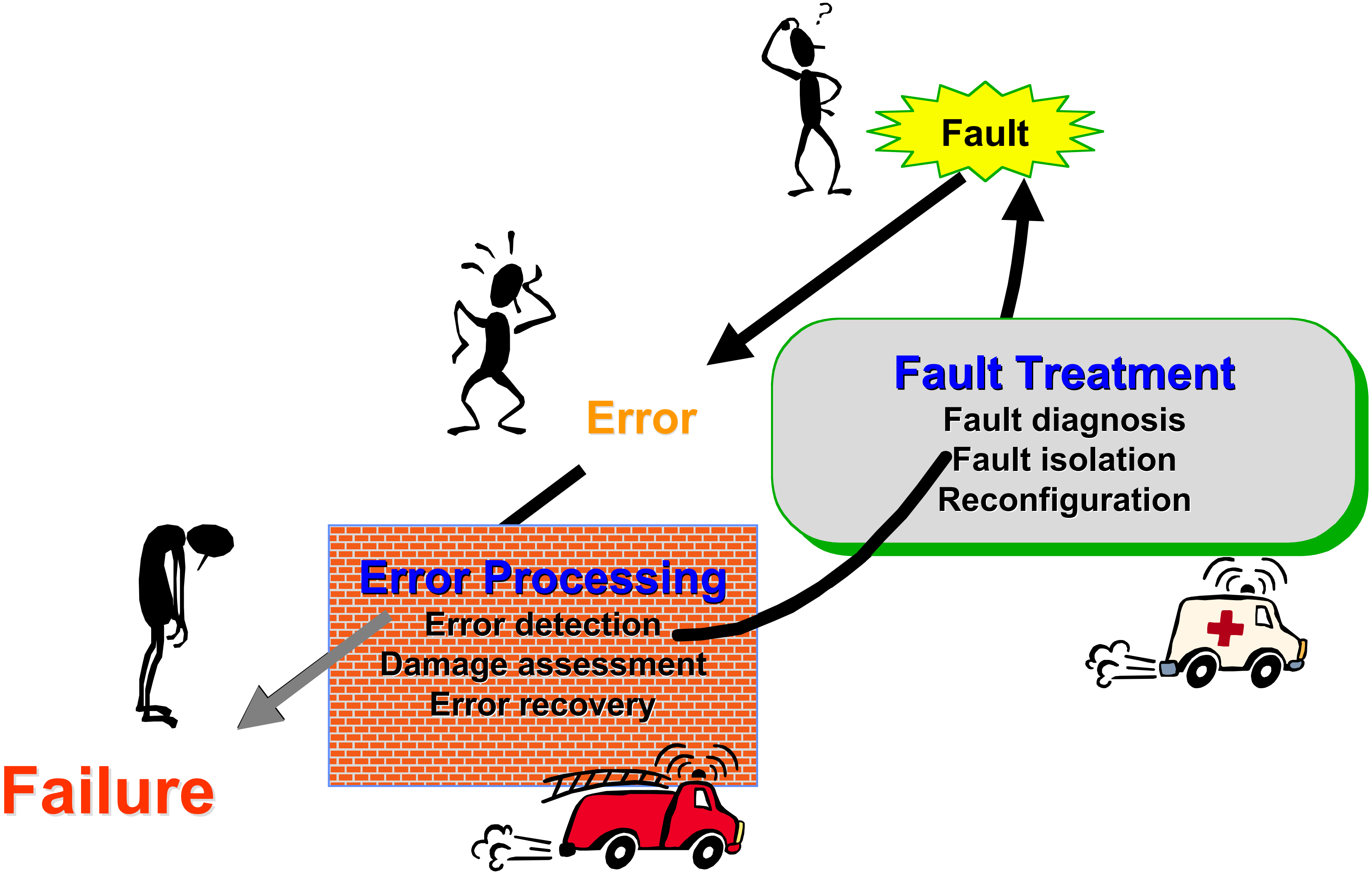
Compensation-based recovery (fault masking)



# Error Recovery

- Backward recovery
  - Operating system re-installation
  - TCP/IP connection resets
  - System reboots and process re-initialisation
  - Software downgrades
- Forward recovery
  - Automated re-keying procedures
  - Switching to diminished “safe” mode.
  - Software upgrades
- Masking
  - Voting mechanisms
  - Fragmentation-Redundancy-Scattering
  - Sensor correlation

# Fault Tolerance



Fault

Error

Fault Treatment

Fault diagnosis

Fault isolation

Reconfiguration

Error Processing

Error detection

Damage assessment

Error recovery

Failure

# Fault Diagnosis

- Core concepts: identification and locations of faults; prerequisite to isolation & reconfiguration
- **Intrusion diagnosis**, i.e., trying to assess the degree of success of the intruder in terms of system penetration
- **Vulnerability diagnosis**, i.e., trying to understand the channels through which the intrusion took place so that corrective maintenance can be carried out

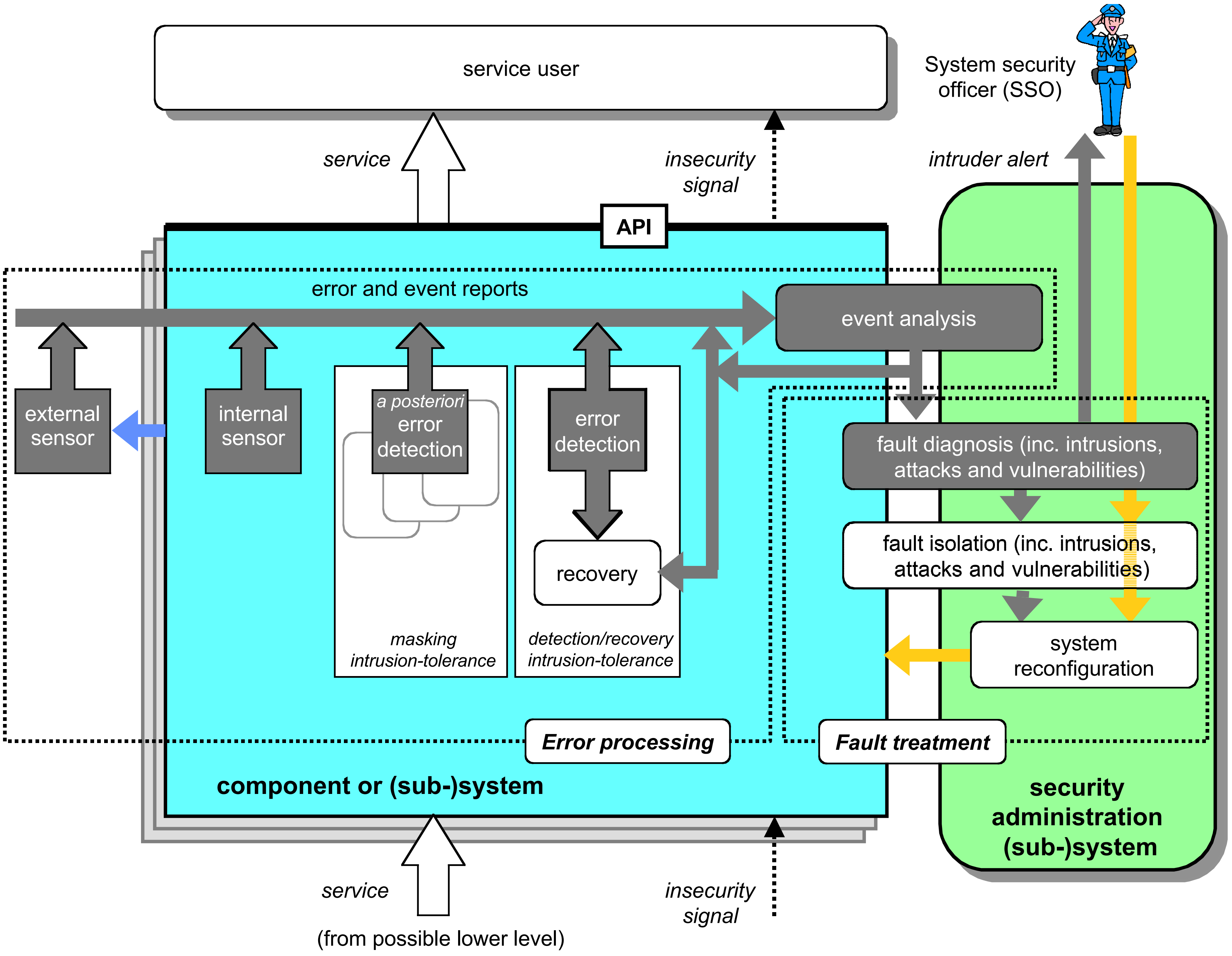
  (diagnosis immediate if errors signaled by vulnerability scanner or configuration checker)
- **Attack diagnosis**, i.e., finding out who or what organisation is responsible for the attack in order that appropriate litigation or retaliation may be initiated

# Fault Isolation

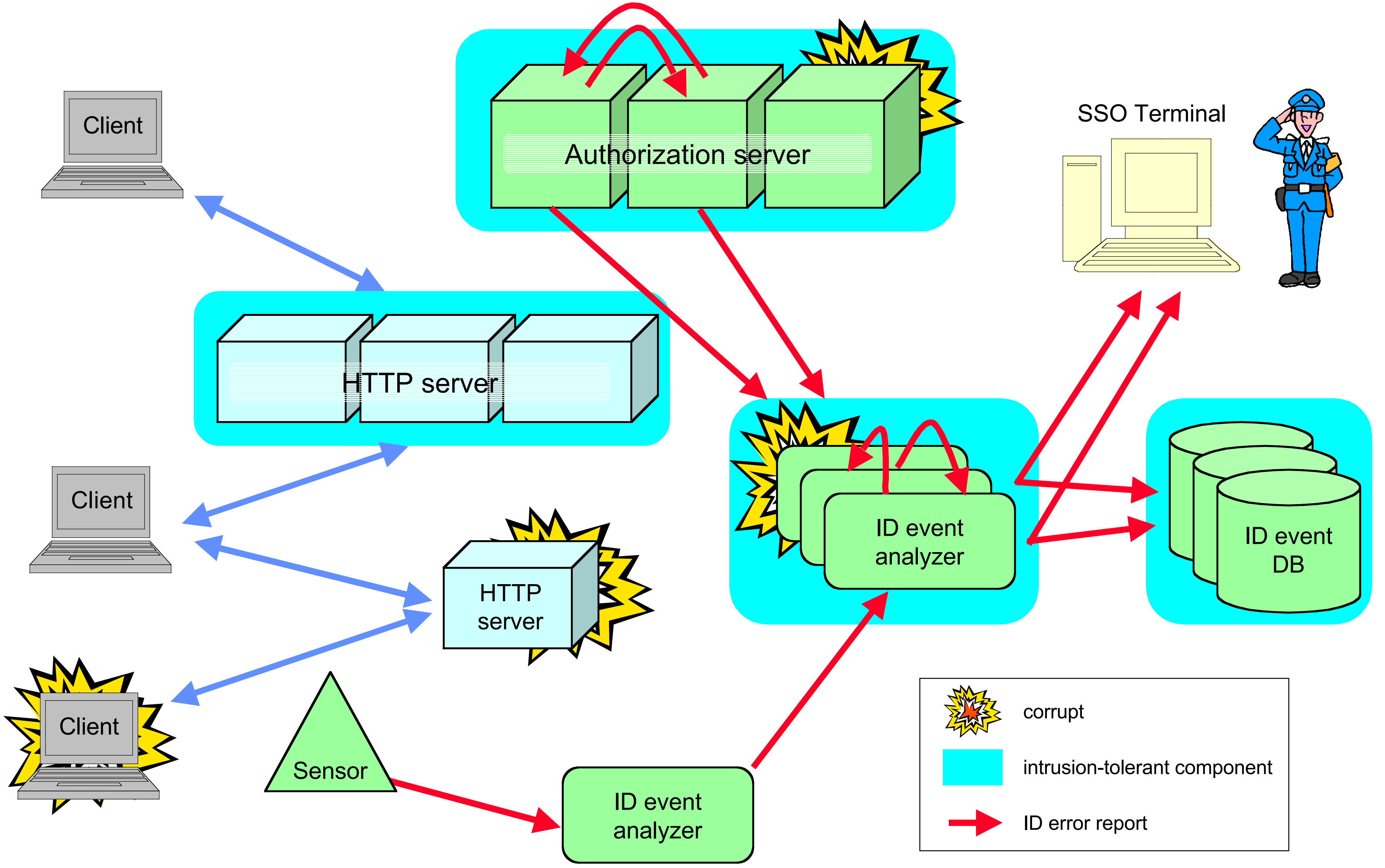
- Core concepts: needed to prevent further errors
- Interpretation wrt intrusions
  - Blocking traffic from an intrusion containment domain that is diagnosed as corrupt, by, for example, changing the settings of firewalls or routers
  - Removing a corrupted file from the system
- Interpretation wrt root causes (vulnerability/attack)
  - Taking off line software versions with newly-found vulnerabilities
  - Arresting the attacker

# System Reconfiguration

- Core concepts: redeployment of fault-free resources + corrective maintenance
- Interpretation wrt intrusions
  - Change a voting threshold, e.g., 3/5 => 2/3 after 2 corruptions
  - Deployment of countermeasures, inc. probes and traps
- Corrective maintenance actions
  - Vulnerability removal
    - software revision and upgrade
    - security patches
  - Attacker rehabilitation


service user
System security officer (SSO)
service
insecurity signal
intruder alert
API
error and event reports
event analysis
external sensor
internal sensor
a posteriori error detection
error detection
recovery
masking intrusion-tolerance
detection/recovery intrusion-tolerance
fault diagnosis (inc. intrusions, attacks and vulnerabilities)
fault isolation (inc. intrusions, attacks and vulnerabilities)
system reconfiguration
Error processing
Fault treatment
component or (sub-)system
security administration (sub-)system
service
insecurity signal
(from possible lower level)

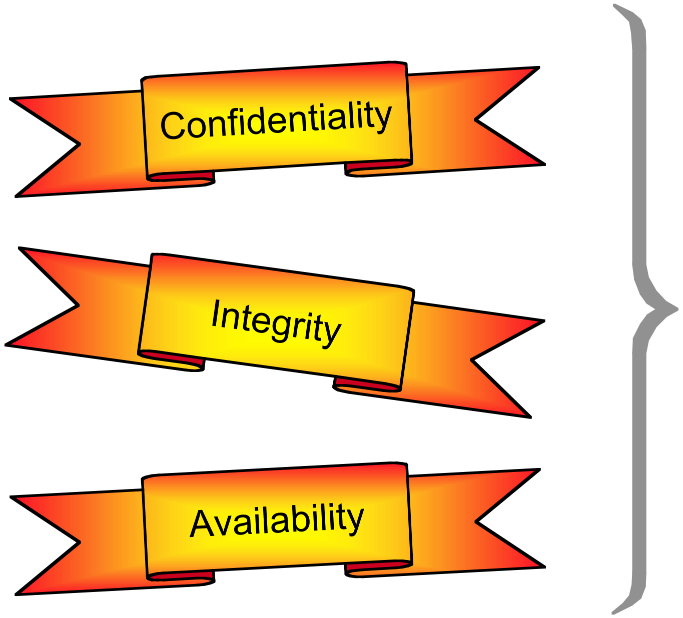
# A (very) Simple Example



# Security Properties

- Auditability
- Confidentiality
- Integrity
- Availability
- Tracability
- Anonymity
- Authenticity
- Accountability
- Imputability
- Irrefutability
- Privacy
- Secrecy
- Non-repudiability
- Opposability

# Security Properties



Confidentiality

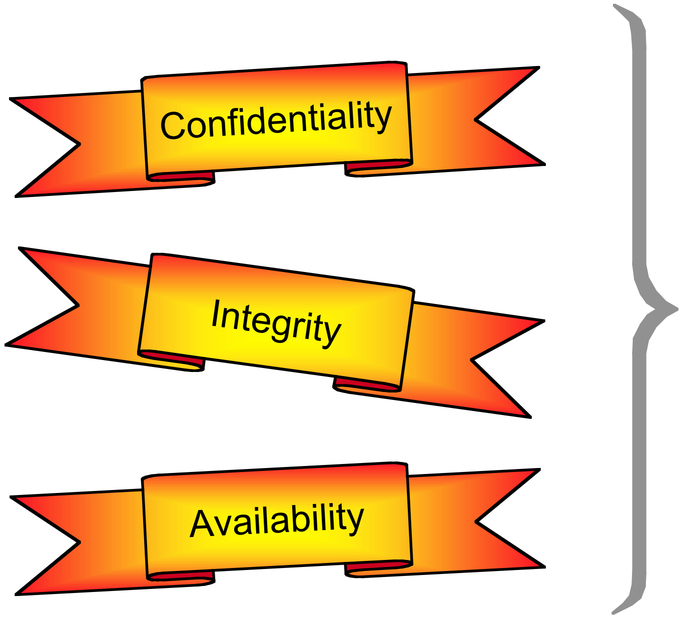
Integrity

Availability

Data

Meta-data

# Security Properties



personal data

message content

etc.

identity of person

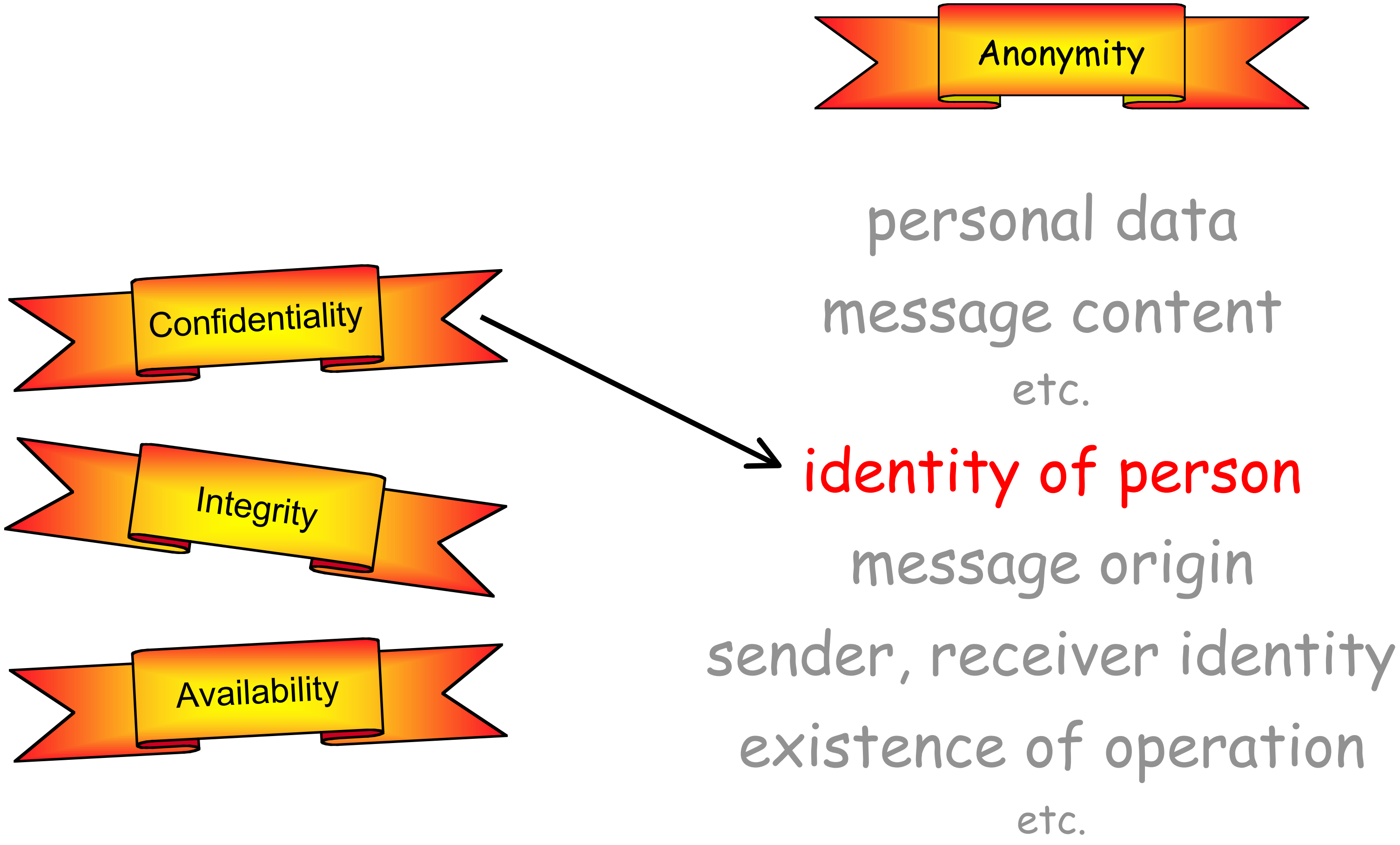
message origin

sender, receiver identity

existence of operation

etc.

# Security Properties



Anonymity

Confidentiality

Integrity

Availability

personal data

message content

etc.

identity of person

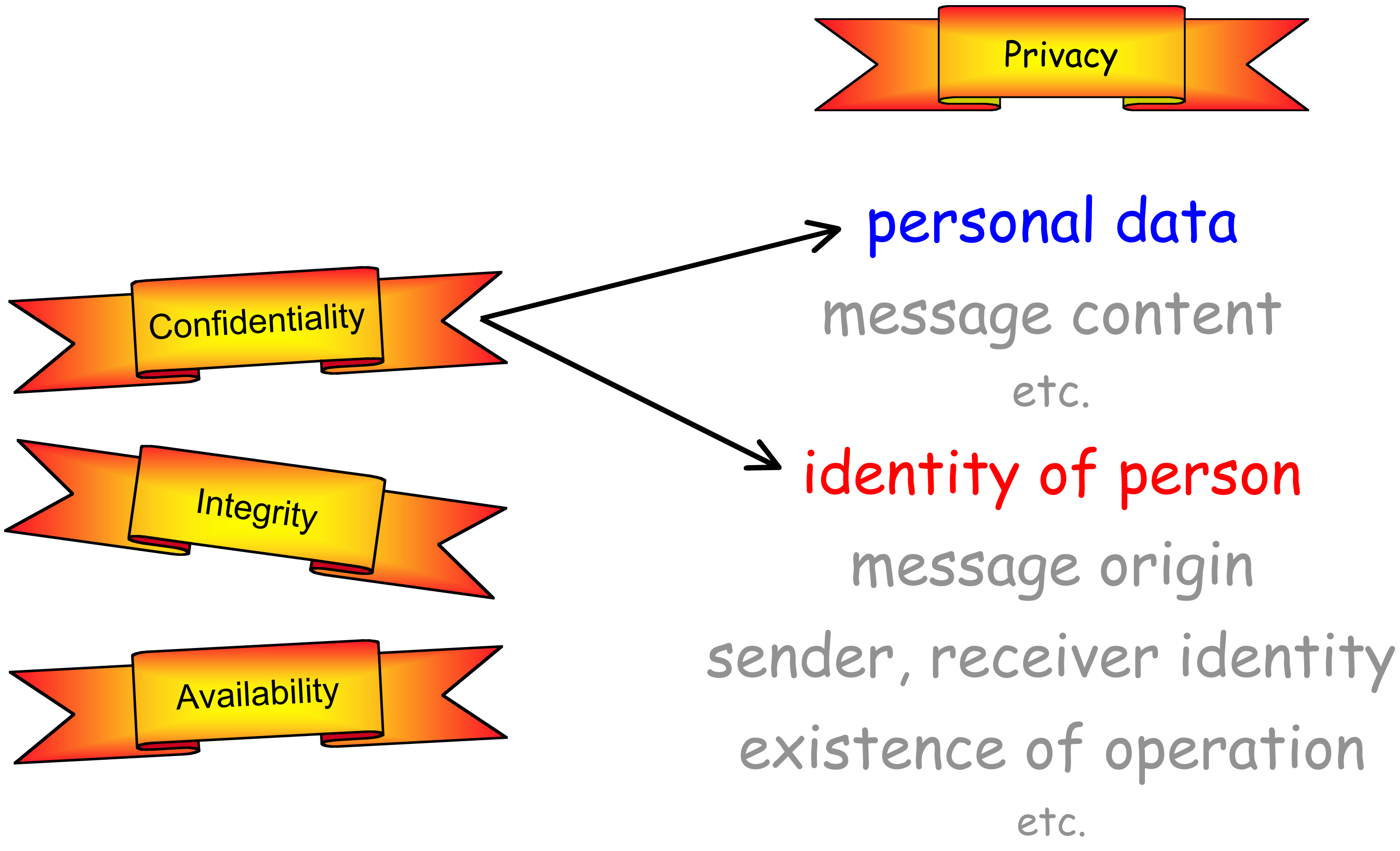
message origin

sender, receiver identity

existence of operation

etc.

# Security Properties



Privacy

Confidentiality

Integrity

Availability

personal data

message content

etc.

identity of person

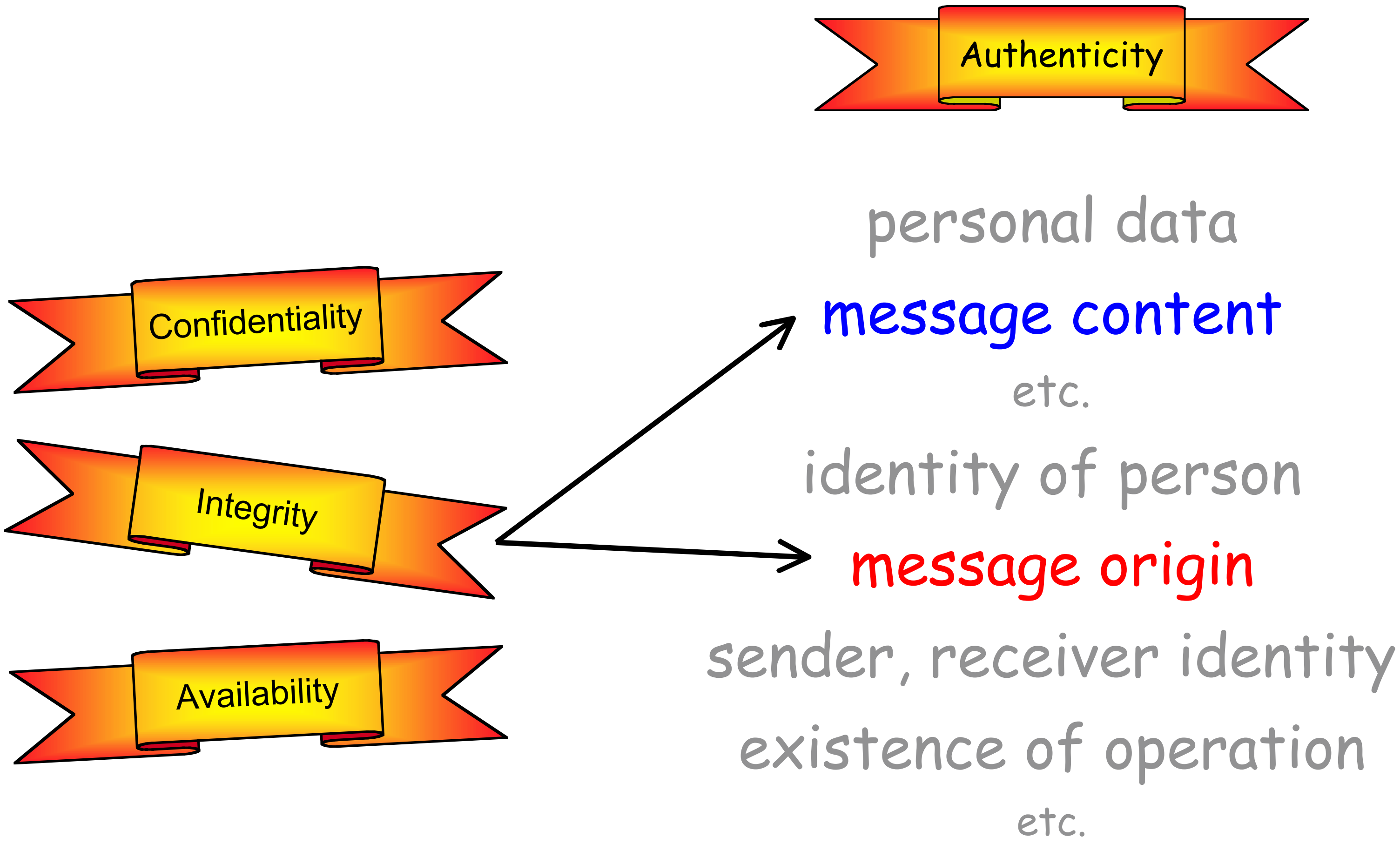
message origin

sender, receiver identity

existence of operation

etc.

# Security Properties



Authenticity

Confidentiality

Integrity

Availability

personal data

message content

etc.

identity of person

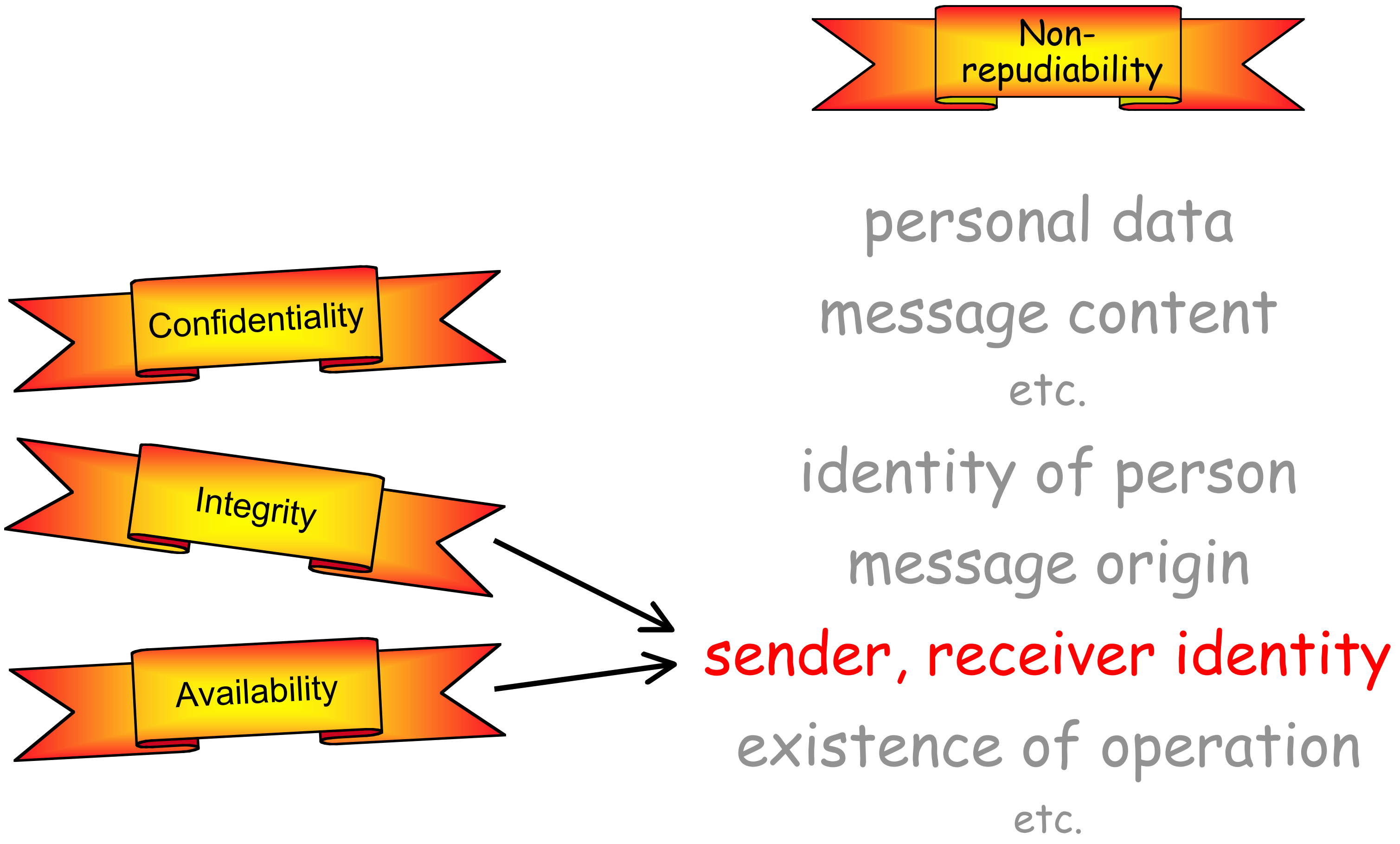
message origin

sender, receiver identity

existence of operation

etc.

# Security Properties



Non-repudiability

Confidentiality

Integrity

Availability

personal data

message content

etc.

identity of person

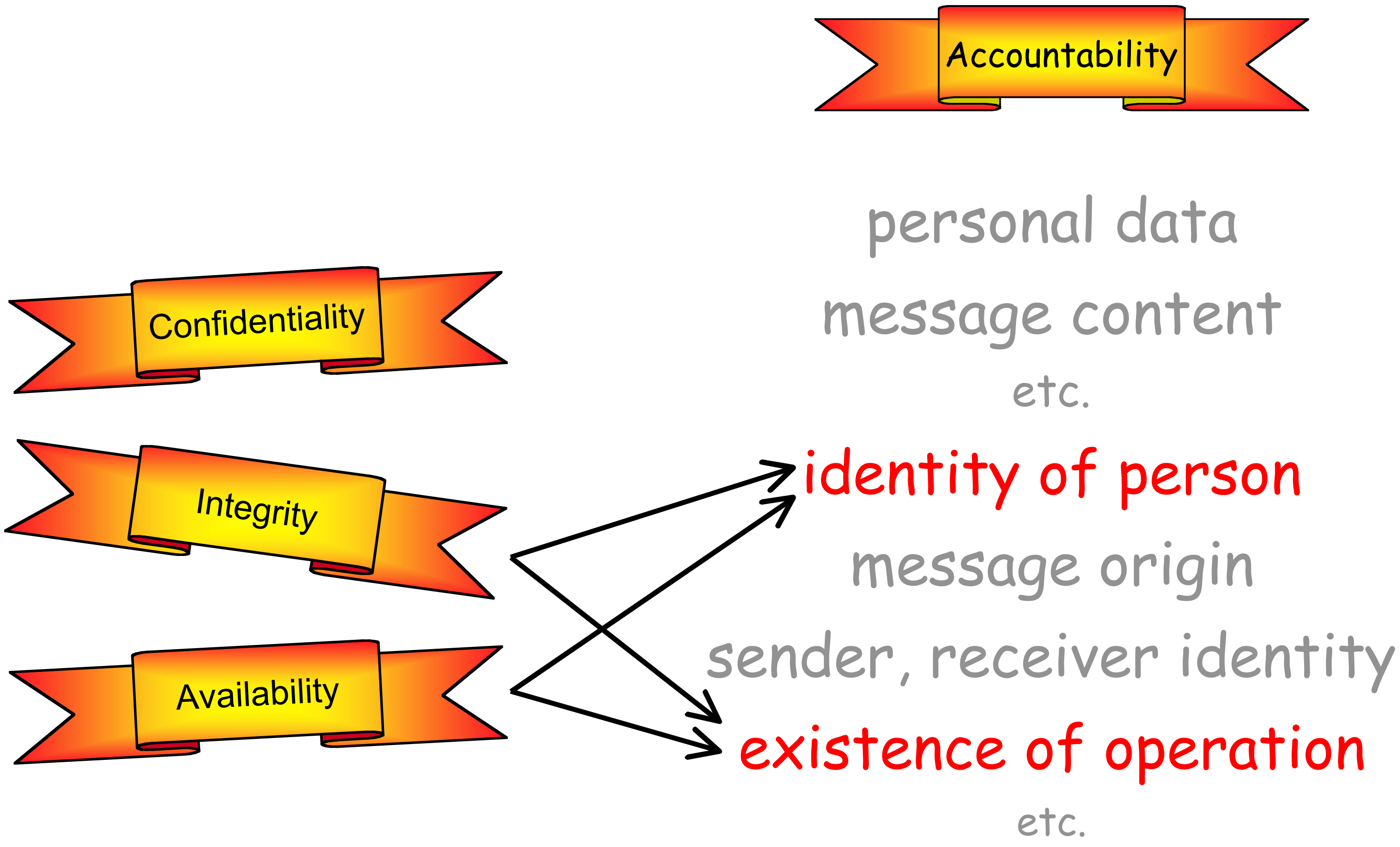
message origin

sender, receiver identity

existence of operation

etc.

# Security Properties



Accountability

personal data

message content

etc.

identity of person

message origin

sender, receiver identity

existence of operation

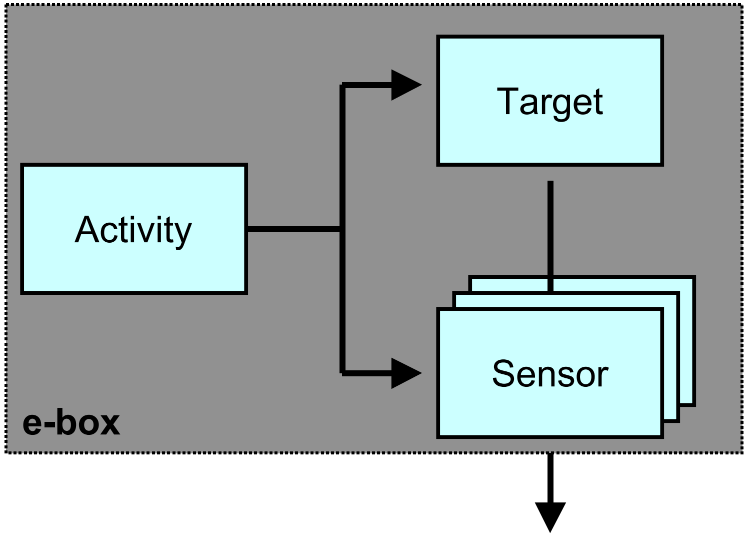
etc.

Confidentiality

Integrity

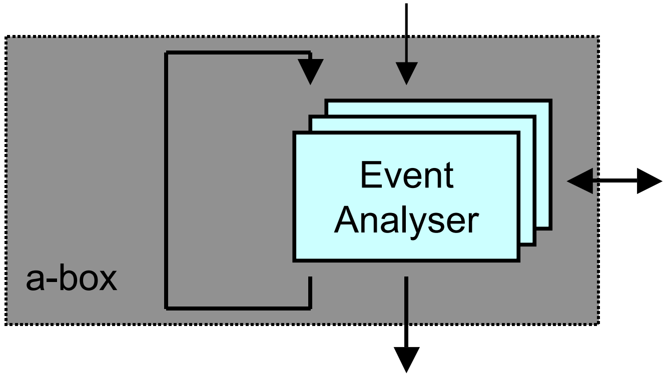
Availability

# ID Event Generator



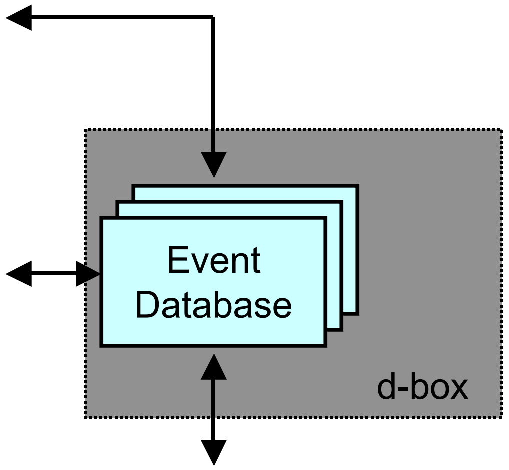
- Target: monitored component
- Sensor: raw data collector (e.g., sniffer, audit log)
- Deployment trade-offs
  - Sensitivity: false alarms vs. misses
  - Deployment: ease vs. completeness
  - User rights: privacy vs. visibility
  - Encryption: attacker view vs. system-administration view

# ID Event Analysis



- Successively transform, filter, normalize, and correlate data, adding semantic relevance and reducing volume at each stage
- Single event analysis box
  - May take its input from several different producers (both from sensor boxes and other event analysis boxes)
  - May feed its output to several different consumers in a topologically arbitrary manner

# ID Event Database



- Provides persistence to IDS
  - Off-line error detection
  - Intrusion analysis
  - Evidence justifying response
- Text file or relational database
- Need to be able to view data with varying degrees of resolution