

# Dependability and its Threats: A Taxonomy

Al Avizienis    Jean-Claude Laprie    Brian Randell



UNIVERSITY OF  
NEWCASTLE



18th IFIP World Computer Congress

**Dependability**: ability to deliver service that can justifiably be trusted

**Service** delivered by a system: its behavior as it is perceived by its user(s)

**User**: another system that interacts with the former

**Function** of a system: what the system is intended to do

(Functional) **Specification**: description of the system function

**Correct service**: when the delivered service implements the system function

**Service failure**: event that occurs when the delivered service deviates from correct service, either because the system does not comply with the specification, or because the specification did not adequately describe its function

**Failure modes**: the ways in which a system can fail, ranked according to failure severities

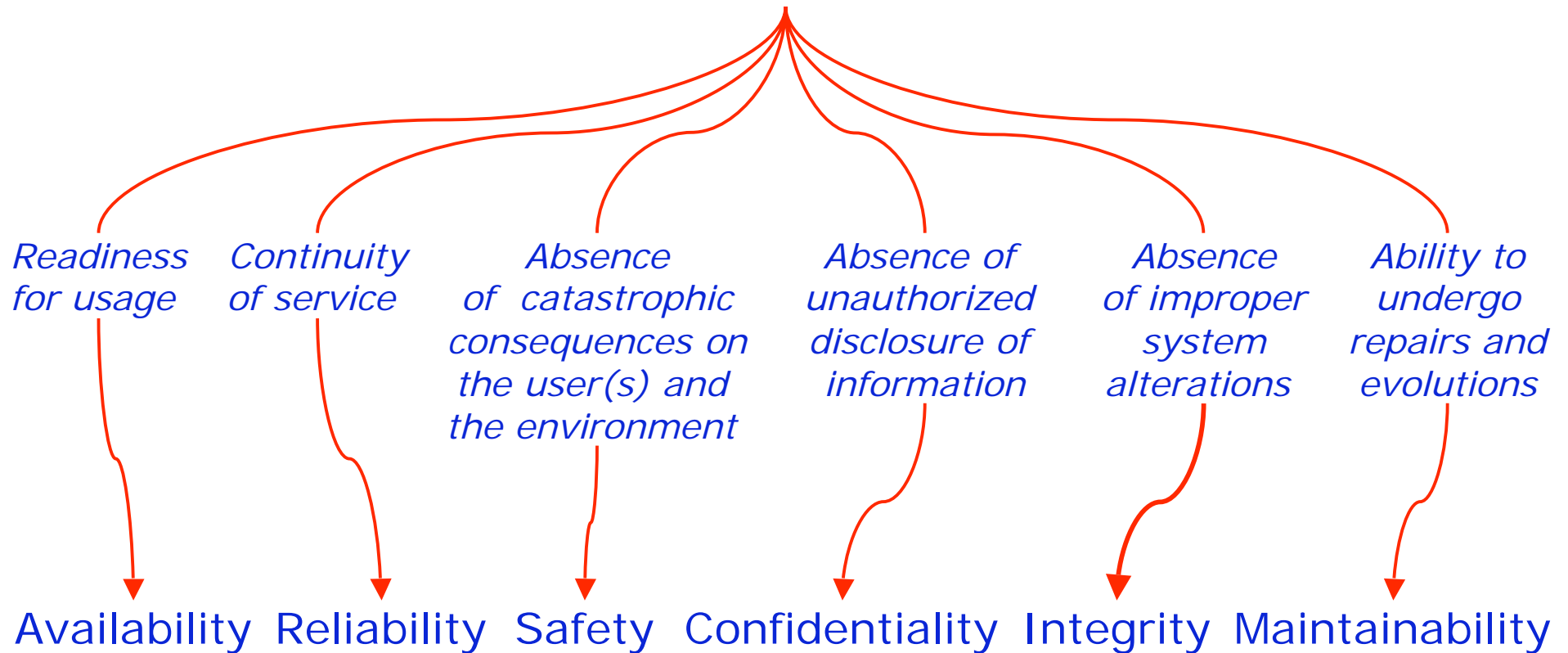
Part of system state that may cause a subsequent service failure: **error**

Adjudged or hypothesized cause of an error: **fault**

**Dependability**: ability to avoid service failures that are more frequent or more severe than is acceptable

When service failures are more frequent or more severe than acceptable: **dependability failure**

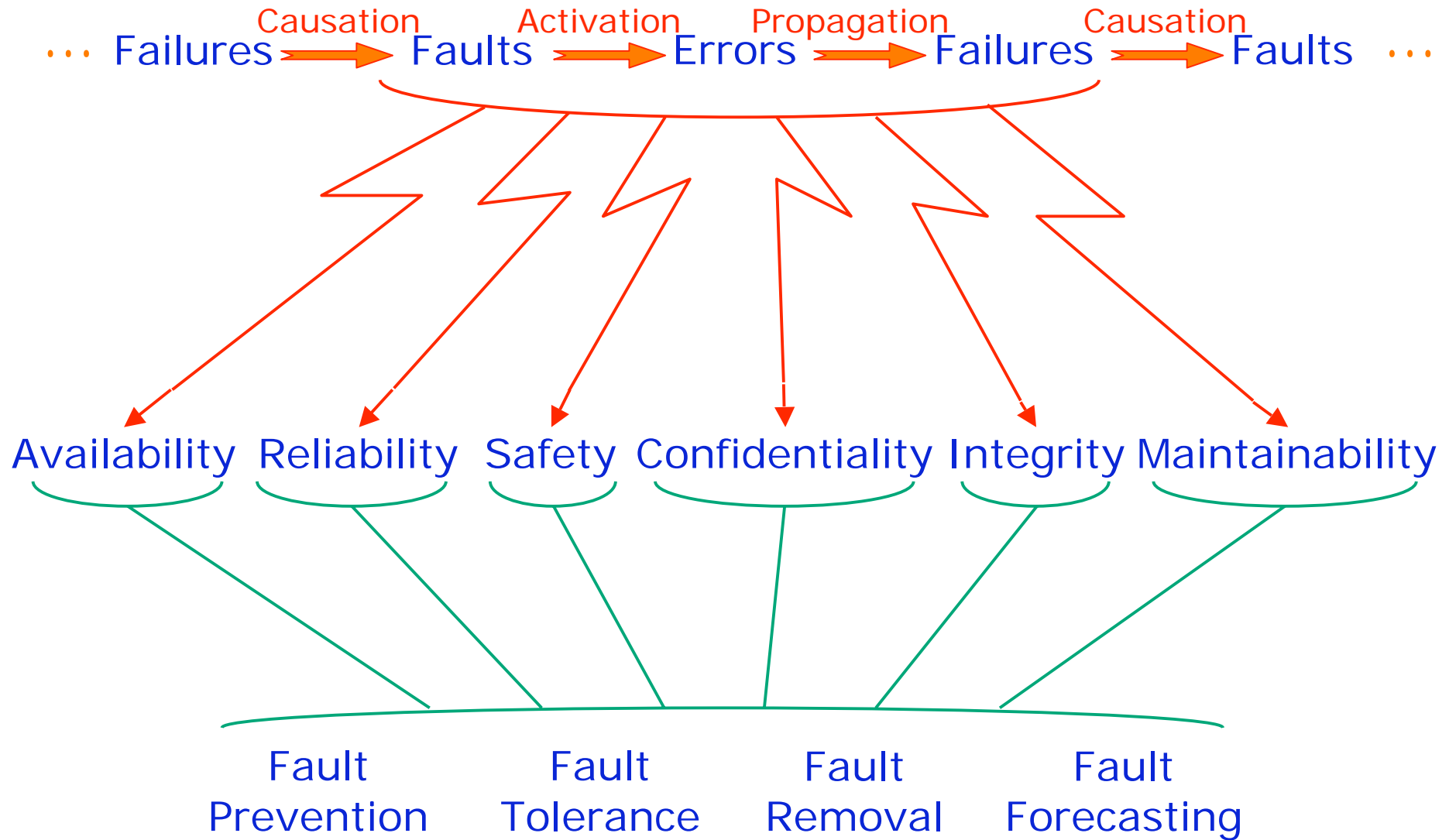
# Dependability

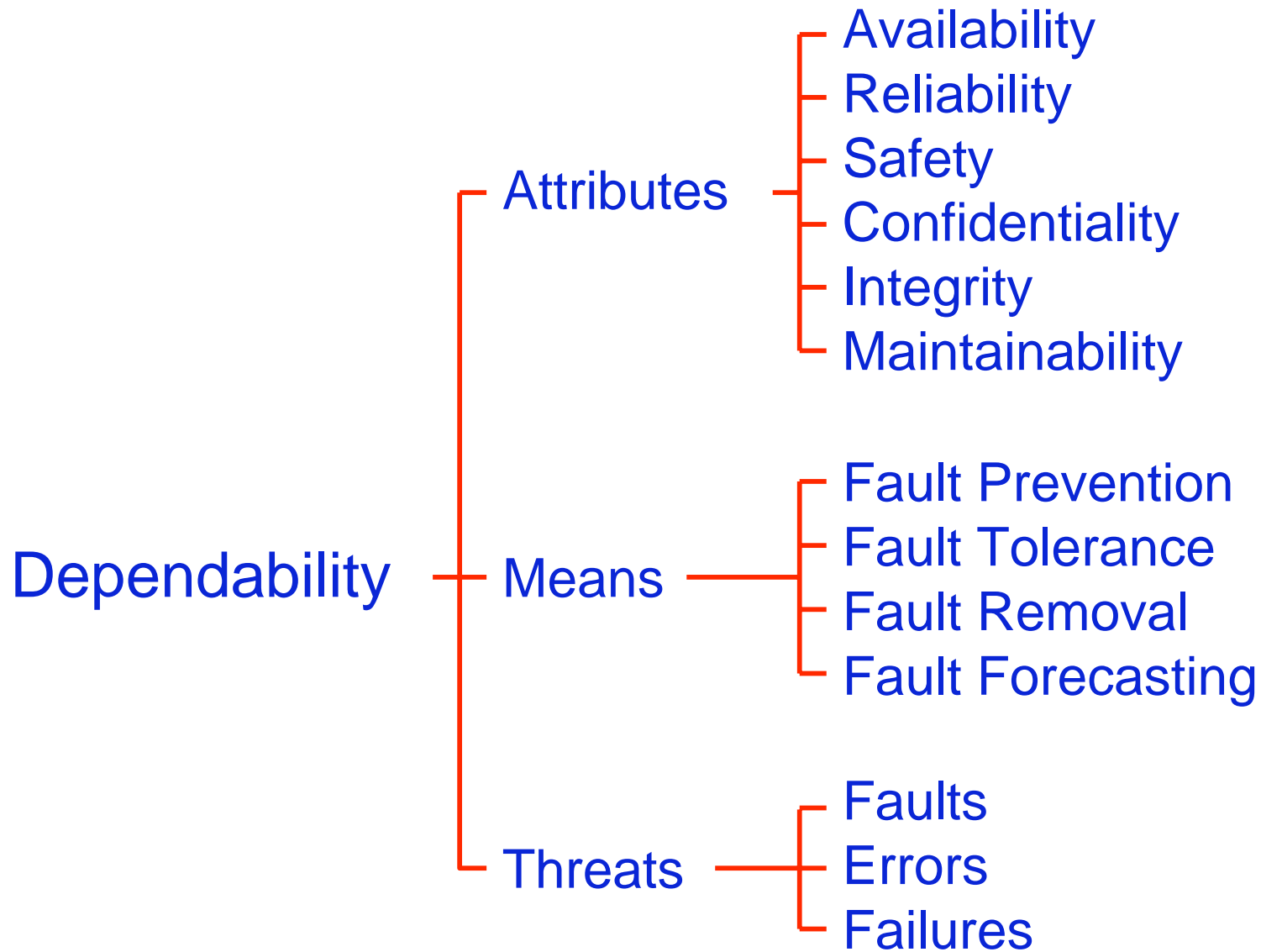


*Authorized actions*

## Security

Absence of unauthorized access to, or handling of, system state



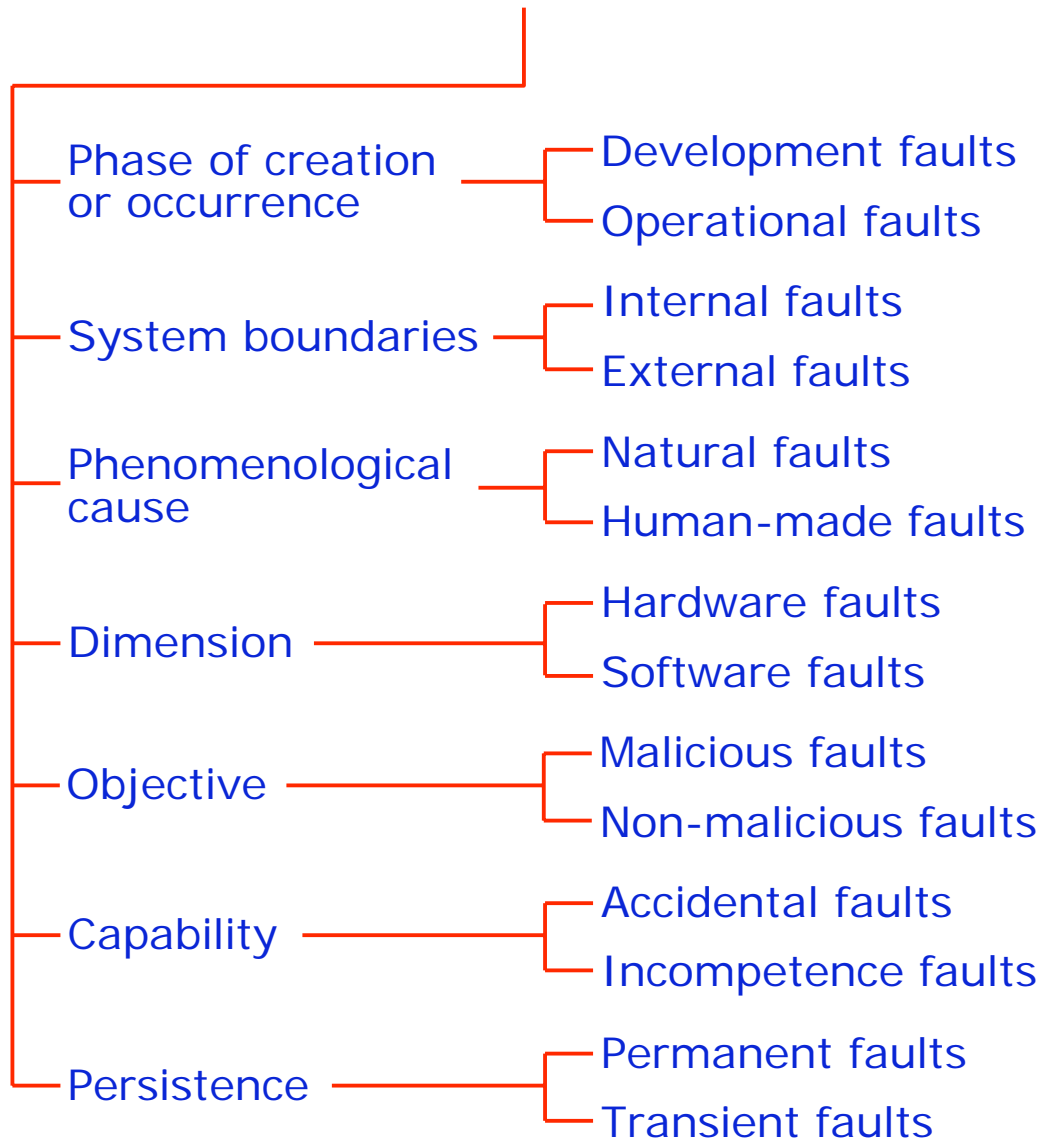


# Situation

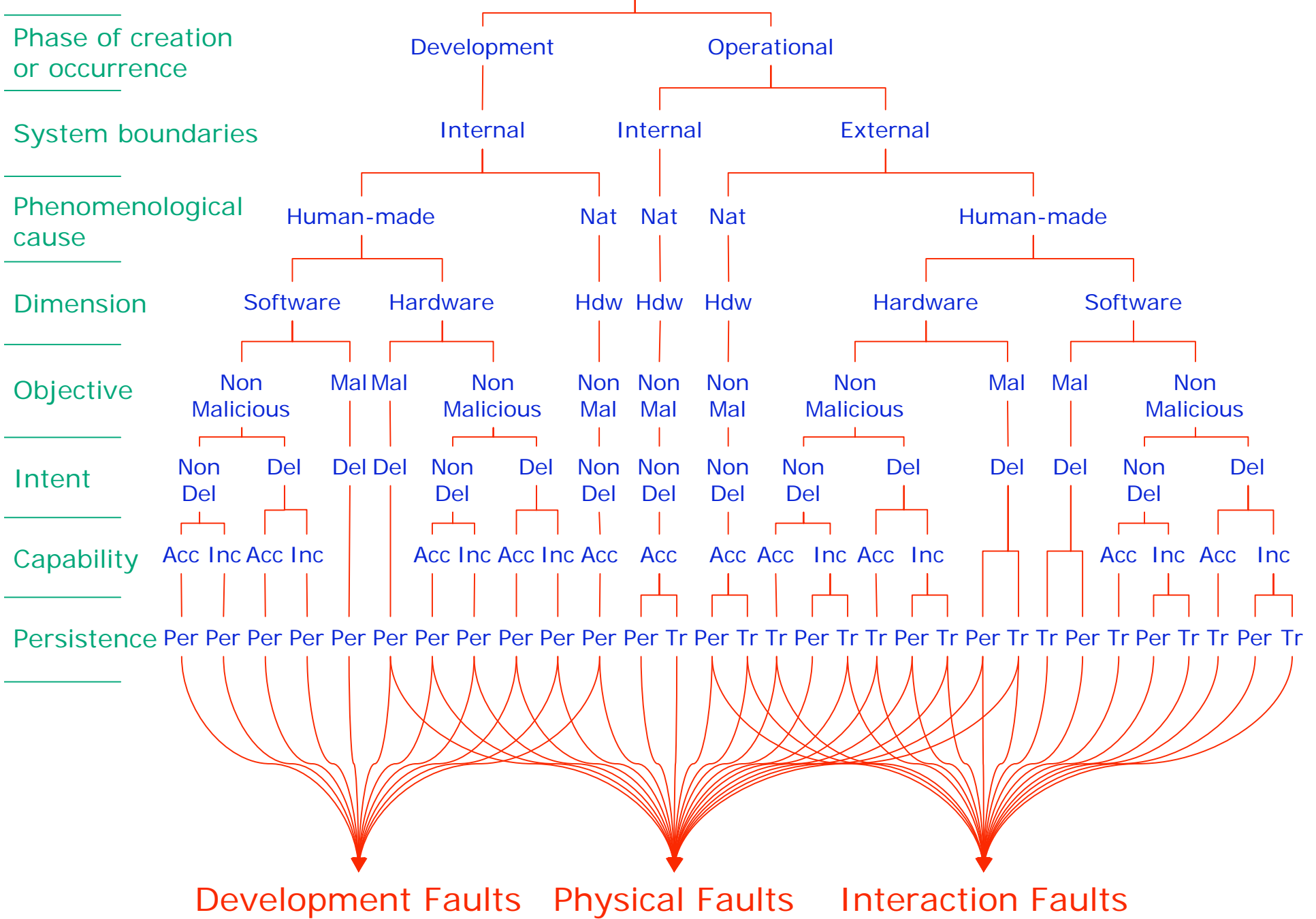
- ✓ Relationship between dependability and security
- ✓ Alternate definition of dependability
- ✓ Service failures distinguished from dependability failures
- Expanded classification of faults, including criterion of capability in the classification of human-made non-malicious faults → competence
- Dependability issues of the development process → development failures
- Dependability related to dependence and trust
- Dependability compared with high confidence, survivability, trustworthiness

# Service Threats

... Failures → Faults → Errors → Failures → Faults ...



# Faults







# Faults

Phase of creation or occurrence

System boundaries

Phenomenological cause

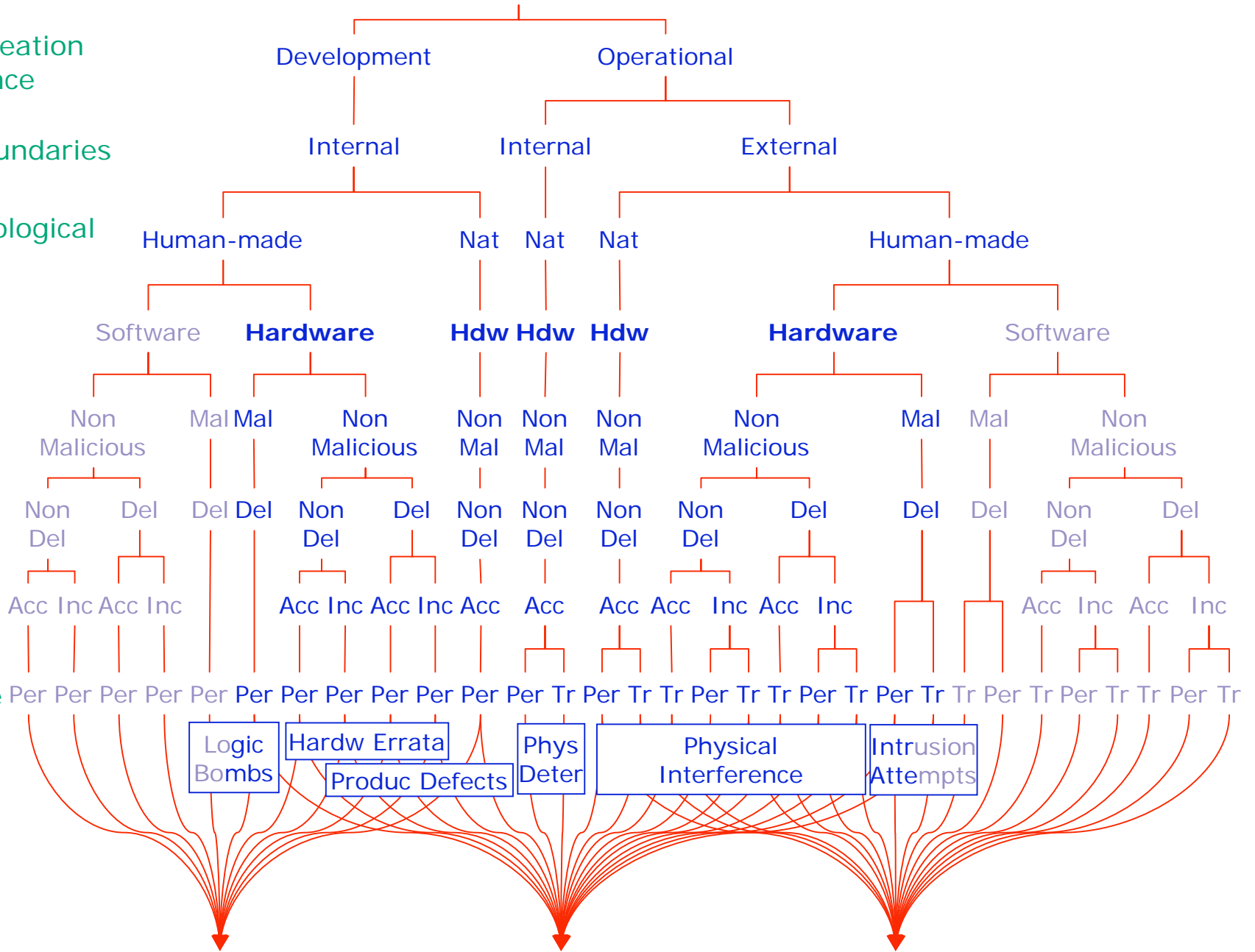
Dimension

Objective

Intent

Capability

Persistence



Development Faults

Physical Faults

Interaction Faults

# Faults

Phase of creation or occurrence

System boundaries

Phenomenological cause

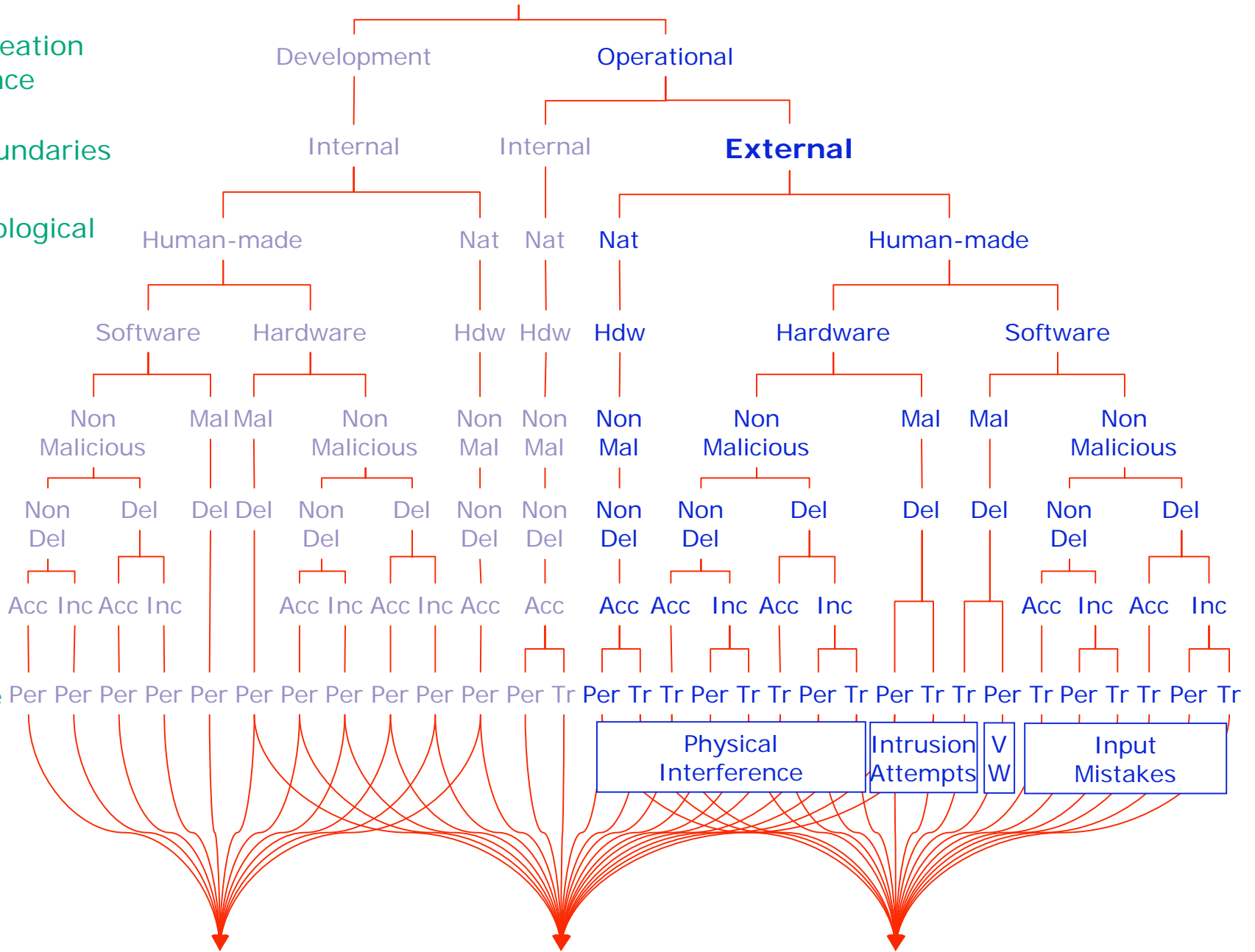
Dimension

Objective

Intent

Capability

Persistence

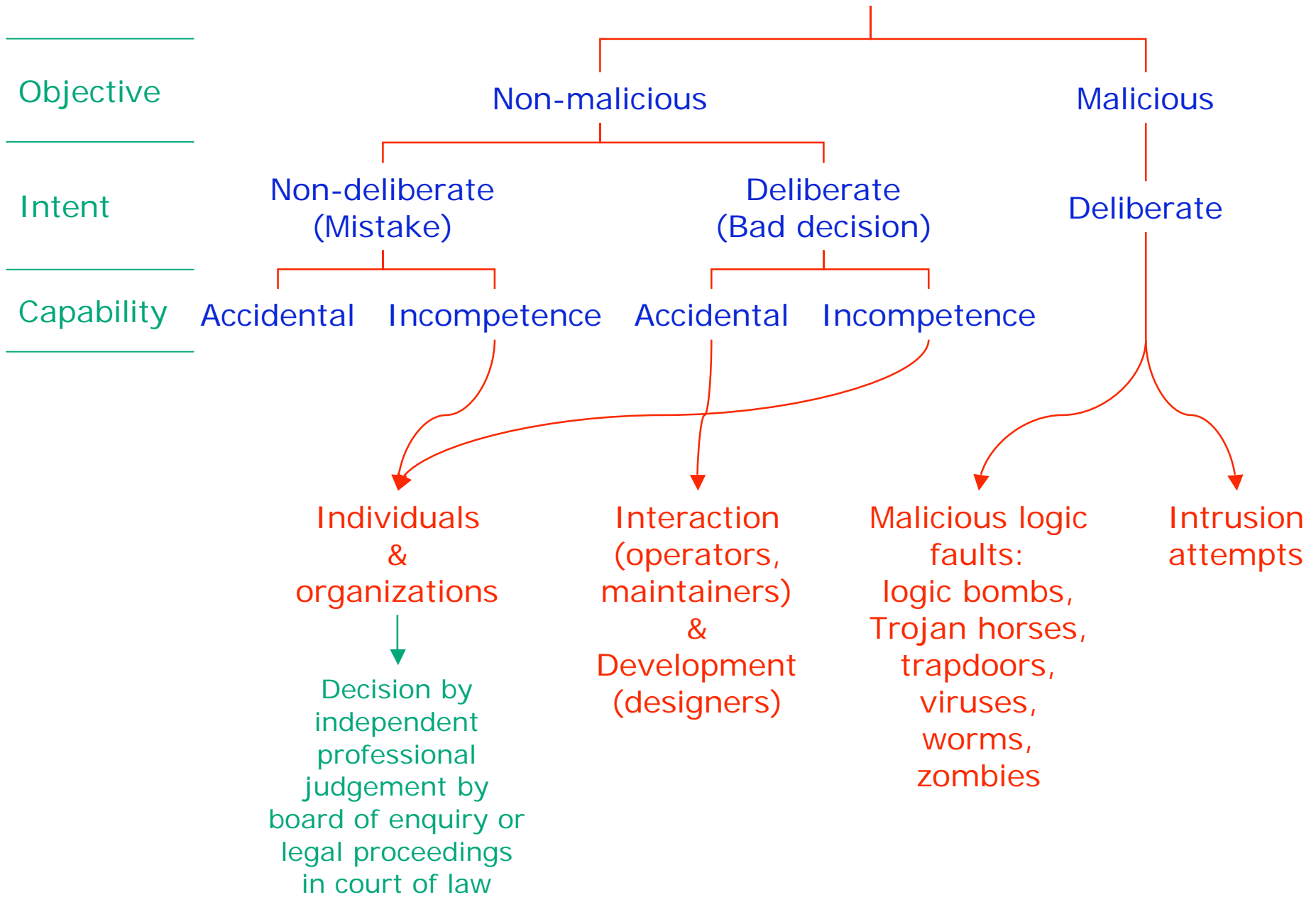


Development Faults

Physical Faults

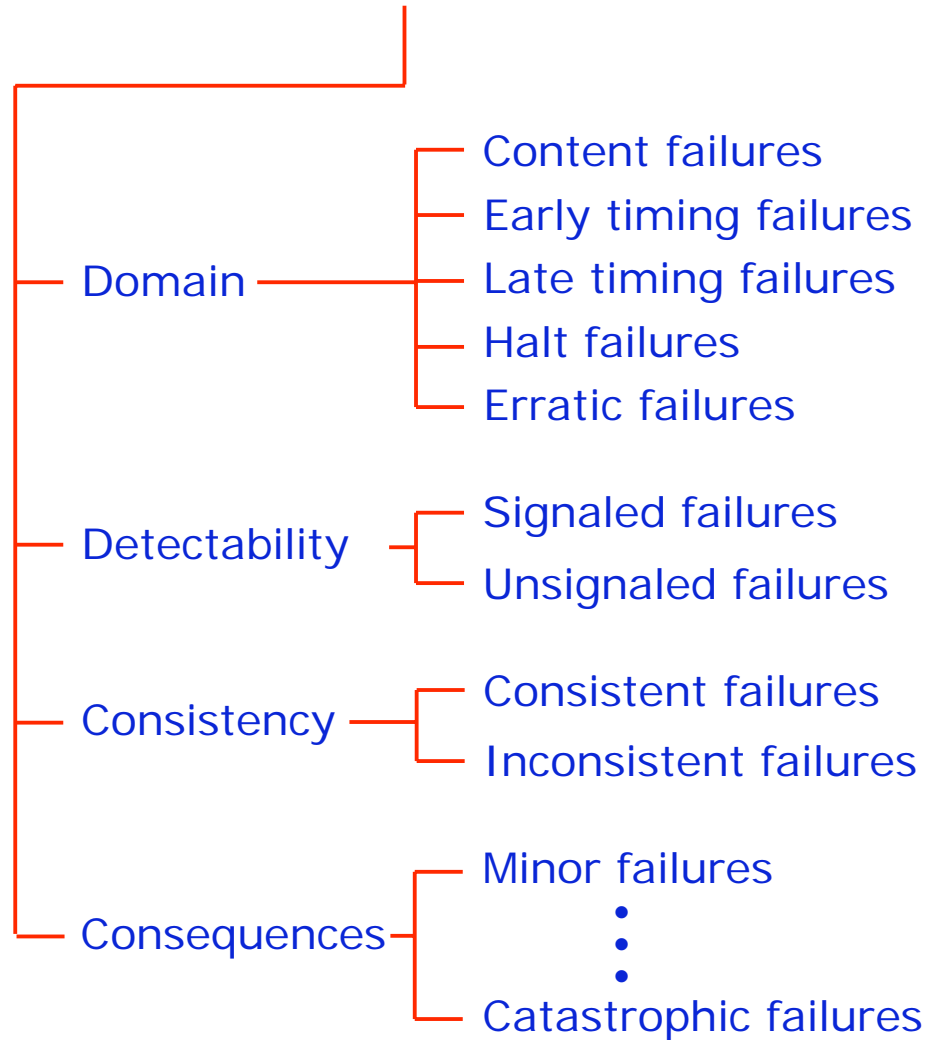
Interaction Faults

# Human-made Faults



# Service Threats

... Failures → Faults → Errors → Failures → Faults ...



... → Failure → Fault → Error → Failure → Fault → ...

Activation Propagation Causation

Error alters service

Facility for stopping recursion  
↓  
Context dependent

Interaction faults  
↓  
Prior presence of a *vulnerability*:  
Internal fault that enables an external fault to harm the system

Activation reproducibility

Solid (hard) faults

Elusive (soft) faults

Interaction or composition

Elusive permanent faults and Transient faults

↓  
*Intermittent faults*

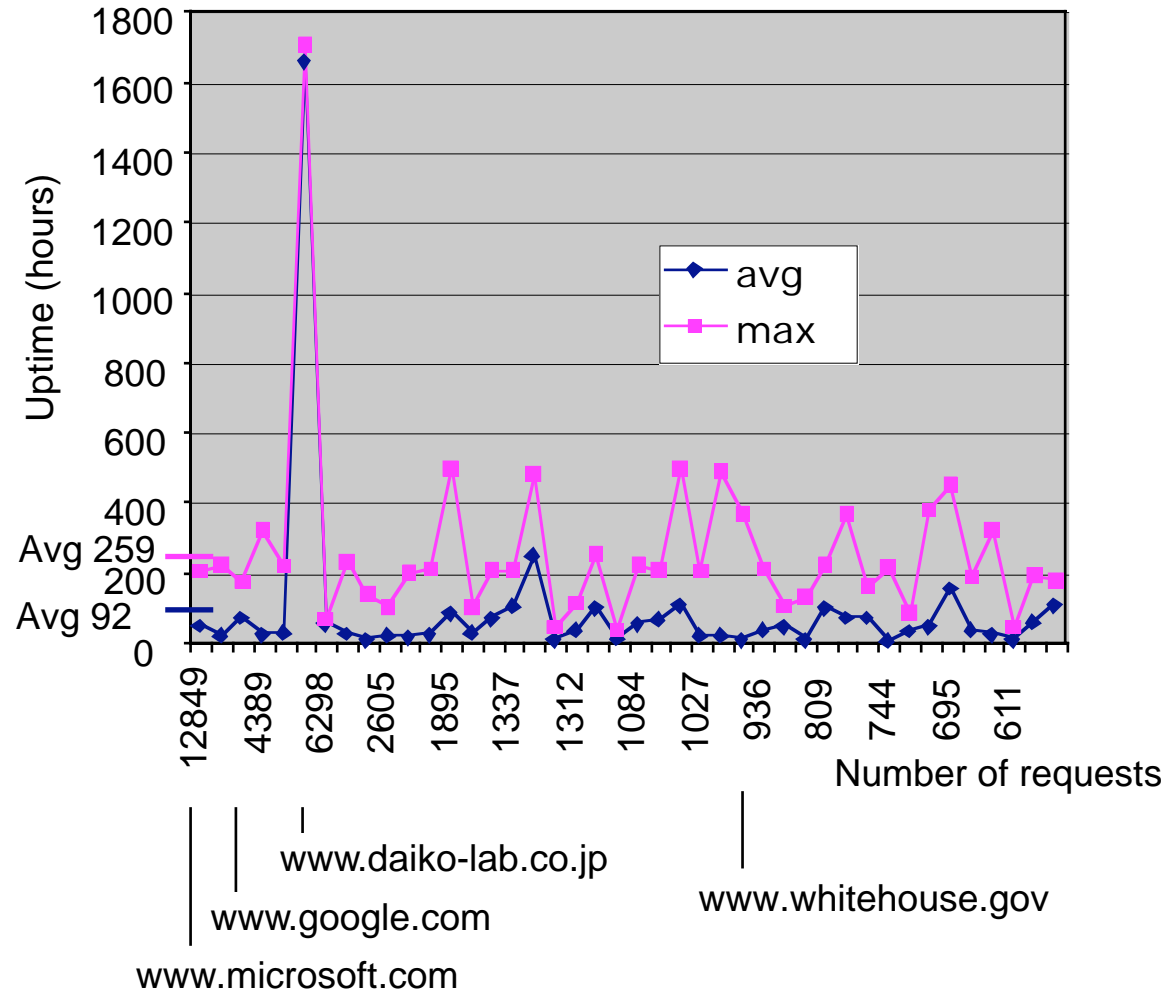
# Non-malicious faults

Number of failures [consequences and outage durations highly application-dependent]	Computer systems (e.g., Transactions, Electronic switching, Back-end servers)		Larger, controlled systems (e.g., Commercial airplanes; telephone network; web applications)	
Faults	Rank	Proportion	Rank	Proportion
Physical internal	3	~ 10%	2	15-20%
Physical interaction	3	~ 10%	2	15-20%
Human-made interaction *	2	~ 20%	1	40-50%
Development	1	~ 60%	2	15-20%

\* Root analysis evidences that they often can be traced to development faults

# NetCraft — Uptime statistics (Dec 1, 2003)

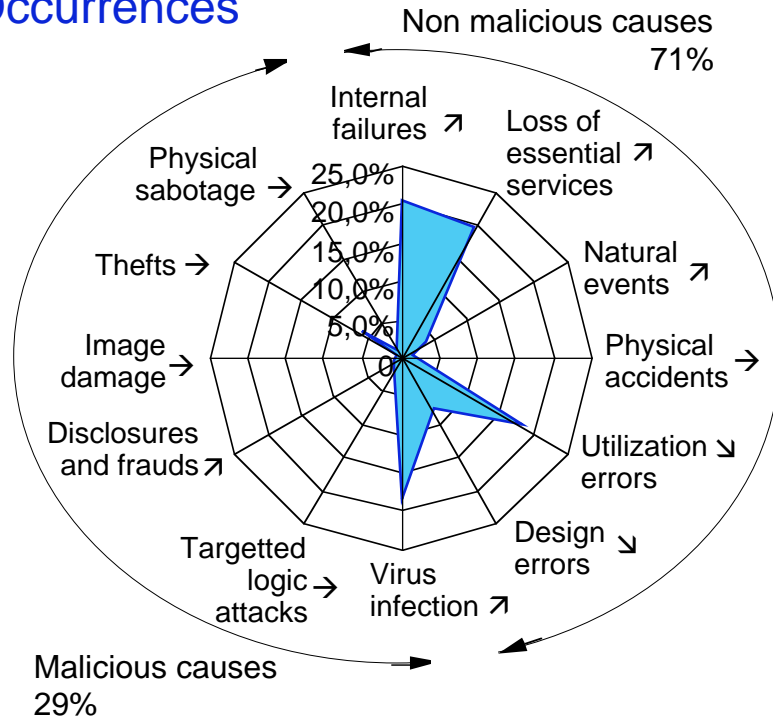
## Top 50 most requested sites



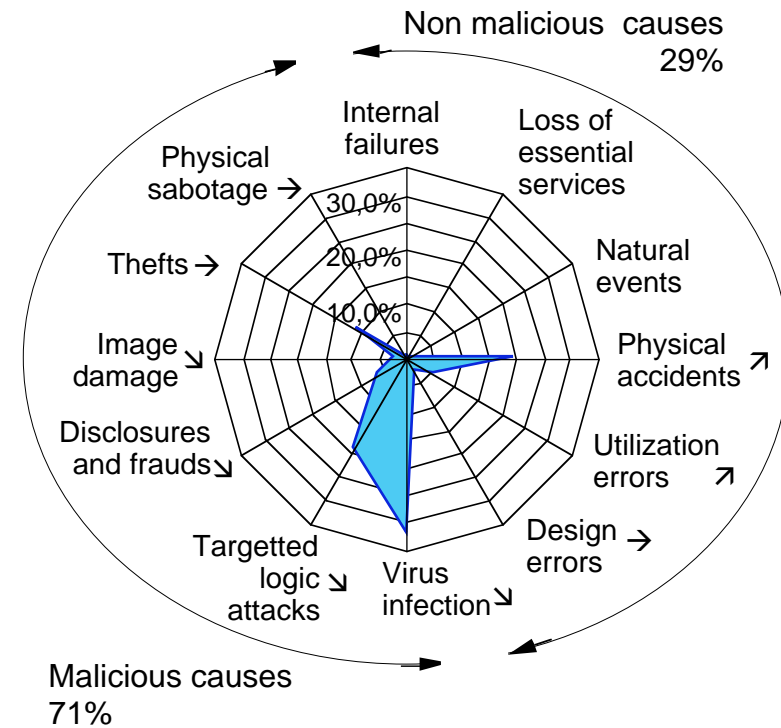


# Yearly survey on computer damages in France — CLUSIF (2000, 2001, 2002)

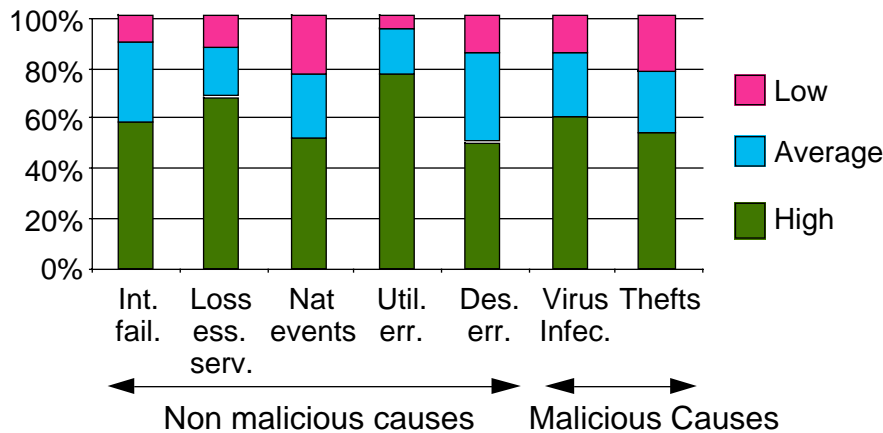
## Occurrences



## Risk perception



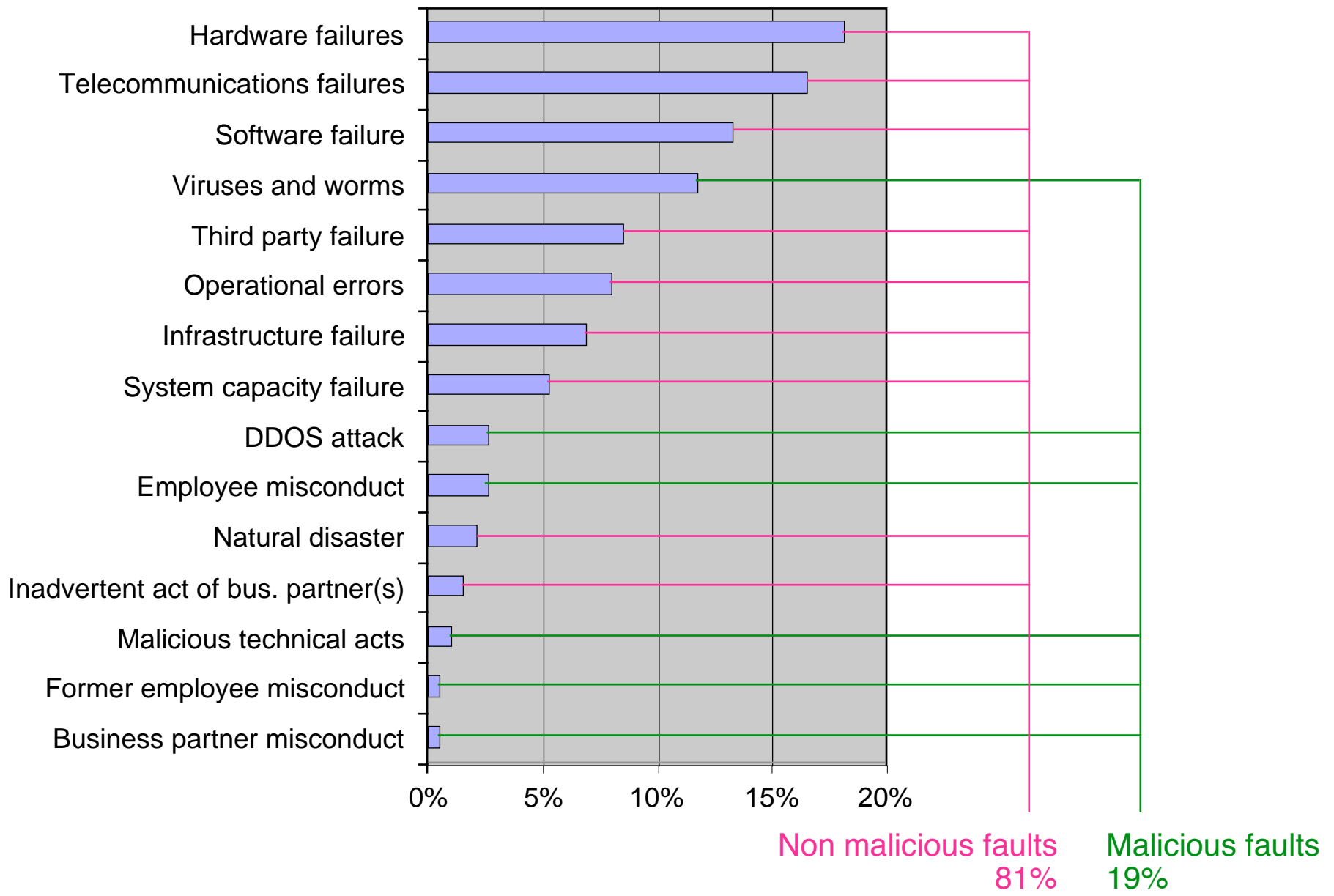
## Occurrence impact



3 year trends

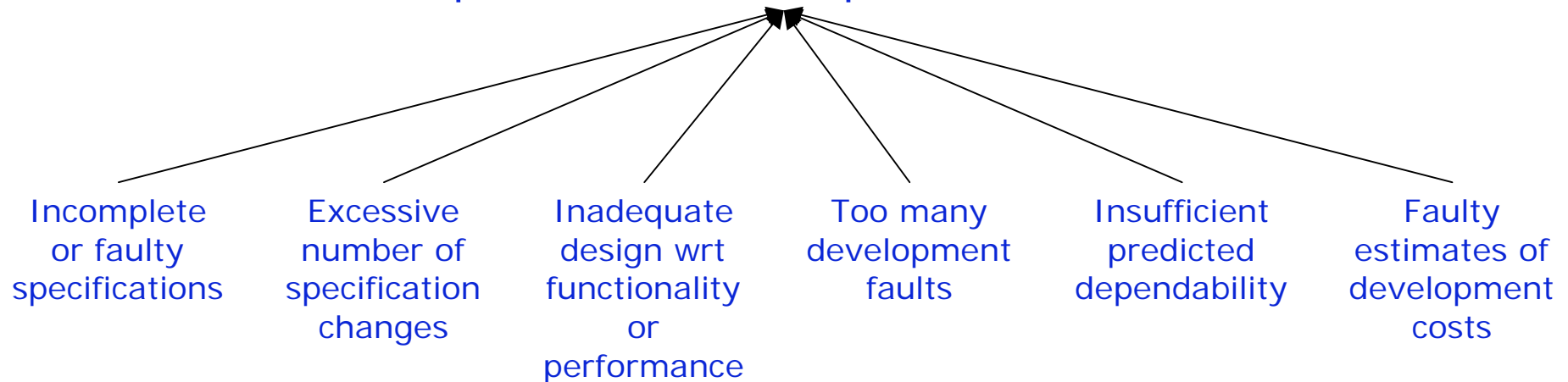
- stable
- ↗ increase
- ↘ decrease

# Global Information Security Survey 2003 — Ernst & Young



# Development failures

Development process terminates before the system is accepted for use and placed into service



## Partial development failures

- Budget or schedule overruns
- Downgrading to less functionality, performance, dependability

## Standish Group (*Chaos reports*)

	1994	2002
Number of surveyed projects	8,380	13,522
Successful projects (completed on-time and on-budget, with all features and functions as initially specified)	16%	34%
Challenged projects (completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified)	53%	51%
Canceled projects	31%	15%
Overruns for challenged projects	89%	82%
Left functions for challenged projects	61%	52%
Total estimated budget for software projects in the USA, in G\$	250	225
Estimated lost value for software projects in the USA, in G\$	81	38

# Dependability and its attributes

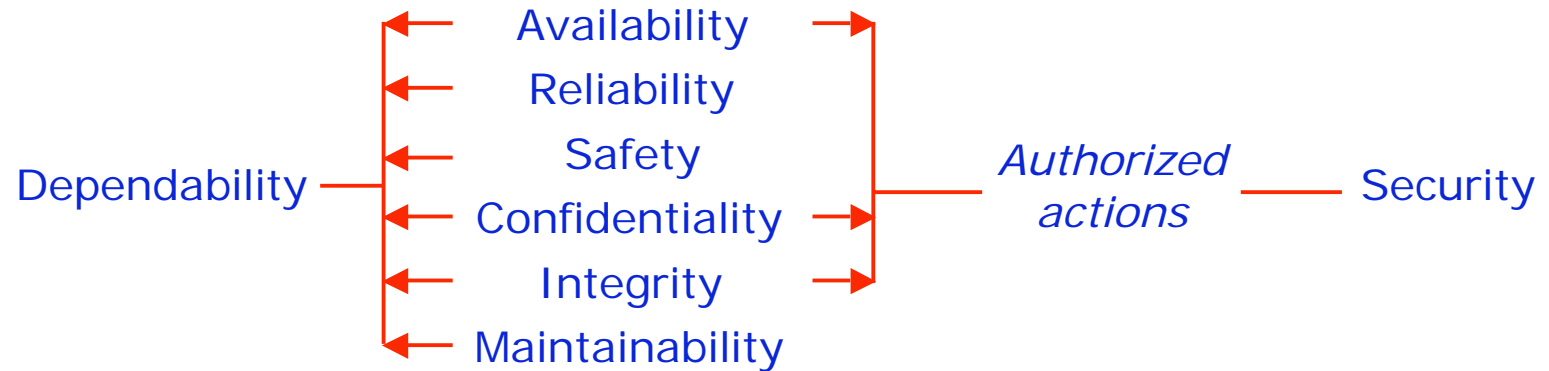
## ❖ Definitions of dependability

- Original definition: ability to deliver service that can justifiably be trusted
  - ☞ Aimed at generalizing availability, reliability, safety, confidentiality, integrity, maintainability, that are then attributes of dependability
- Alternate definition: ability to avoid service failures that are more frequent or more severe than is acceptable
  - ☞ A system can, and usually does, fail. Is it however still dependable ? When does it become undependable ?



criterion for deciding whether or not, in spite of service failures, a system is still to be regarded as dependable.

## ❖ Dependability and security



## ❖ Dependence and trust

- Dependence of system A on system B is the extent to which system A's dependability is (or would be) affected by that of system B
- Trust: accepted dependence

Concept	Dependability	High Confidence	Survivability	Trustworthiness
Goal	<p>1) ability to deliver service that can justifiably be trusted</p> <p>2) ability of a system to avoid service failures that are more frequent or more severe than is acceptable</p>	<p>consequences of the system behavior are well understood and predictable</p>	<p>capability of a system to fulfill its mission in a timely manner</p>	<p>assurance that a system will perform as expected</p>
Threats present	<p>1) development faults (e.g., software flaws, hardware errata, malicious logic)</p> <p>2) physical faults (e.g., production defects, physical deterioration)</p> <p>3) interaction faults (e.g., physical interference, input mistakes, attacks, including viruses, worms, intrusions)</p>	<ul style="list-style-type: none"> <li>• internal and external threats</li> <li>• naturally occurring hazards and malicious attacks from a sophisticated and well-funded adversary</li> </ul>	<p>1) attacks (e.g., intrusions, probes, denials of service)</p> <p>2) failures (internally generated events due to, e.g., software design errors, hardware degradation, human errors, corrupted data)</p> <p>3) accidents (externally generated events such as natural disasters)</p>	<p>1) hostile attacks (from hackers or insiders)</p> <p>2) environmental disruptions (accidental disruptions, either man-made or natural)</p> <p>3) human and operator errors (e.g., software flaws, mistakes by human operators)</p>

# Conclusion

## ❖ Further discussion

- Confidentiality
- Trust and risk management
- Human-machine interactions
- Unified measures of dependability wrt non malicious and malicious faults

+

- New technologies, such as emerging from bio-info-nano convergence