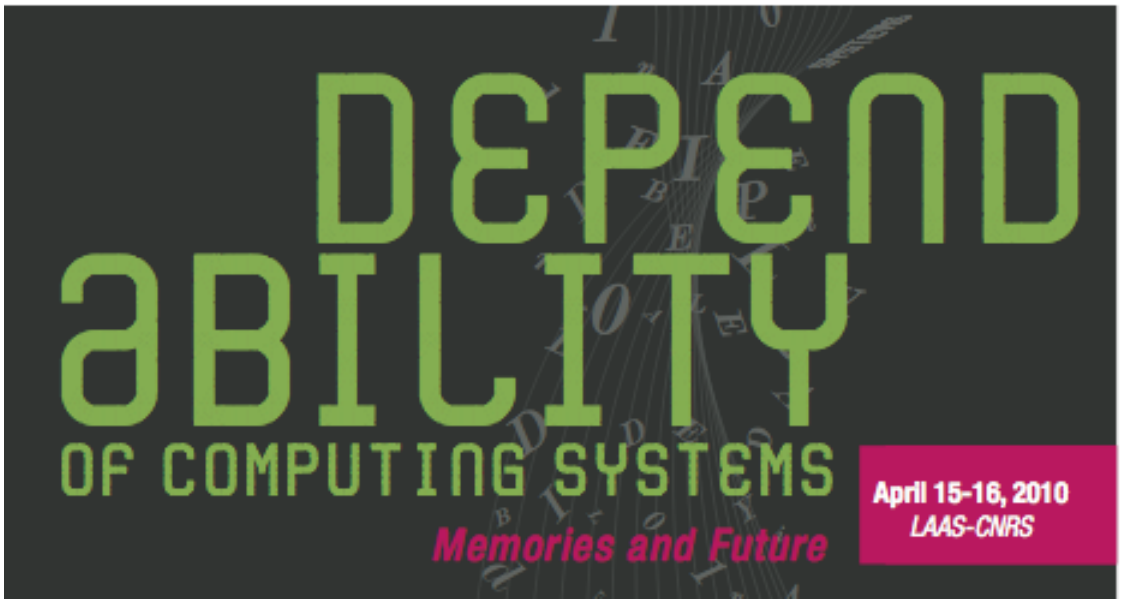
Dependability and Resilience of Computing Systems

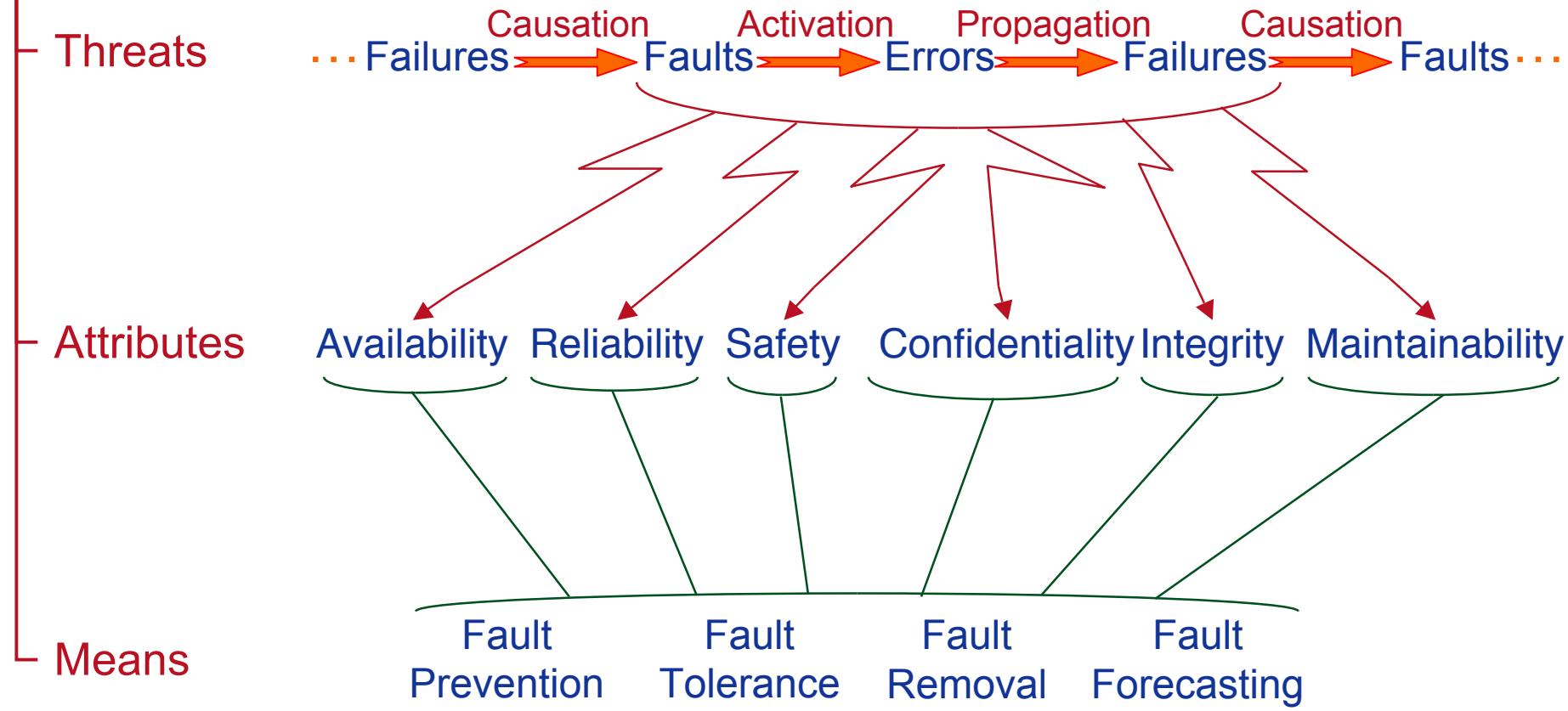
Jean-Claude Laprie

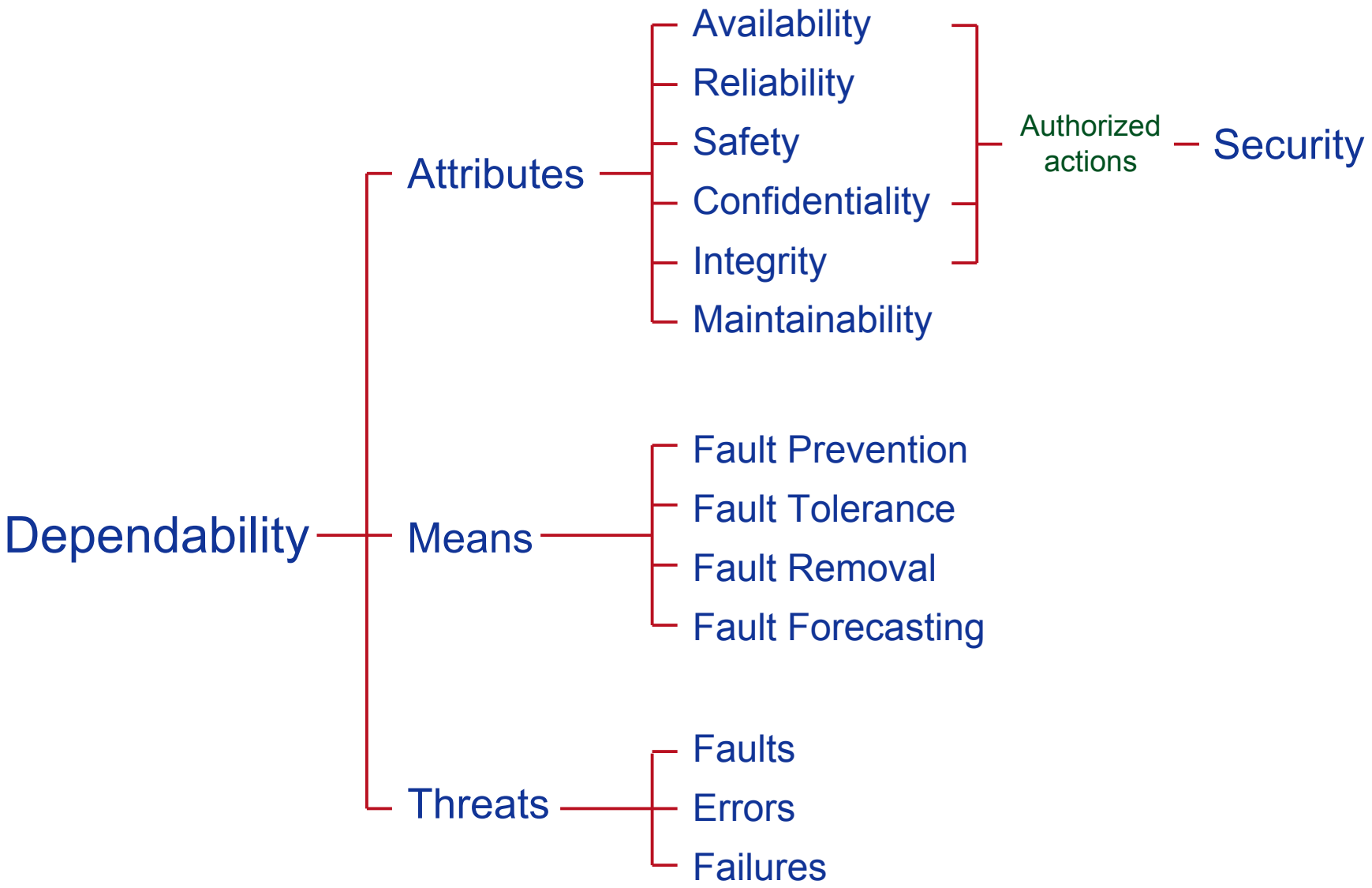


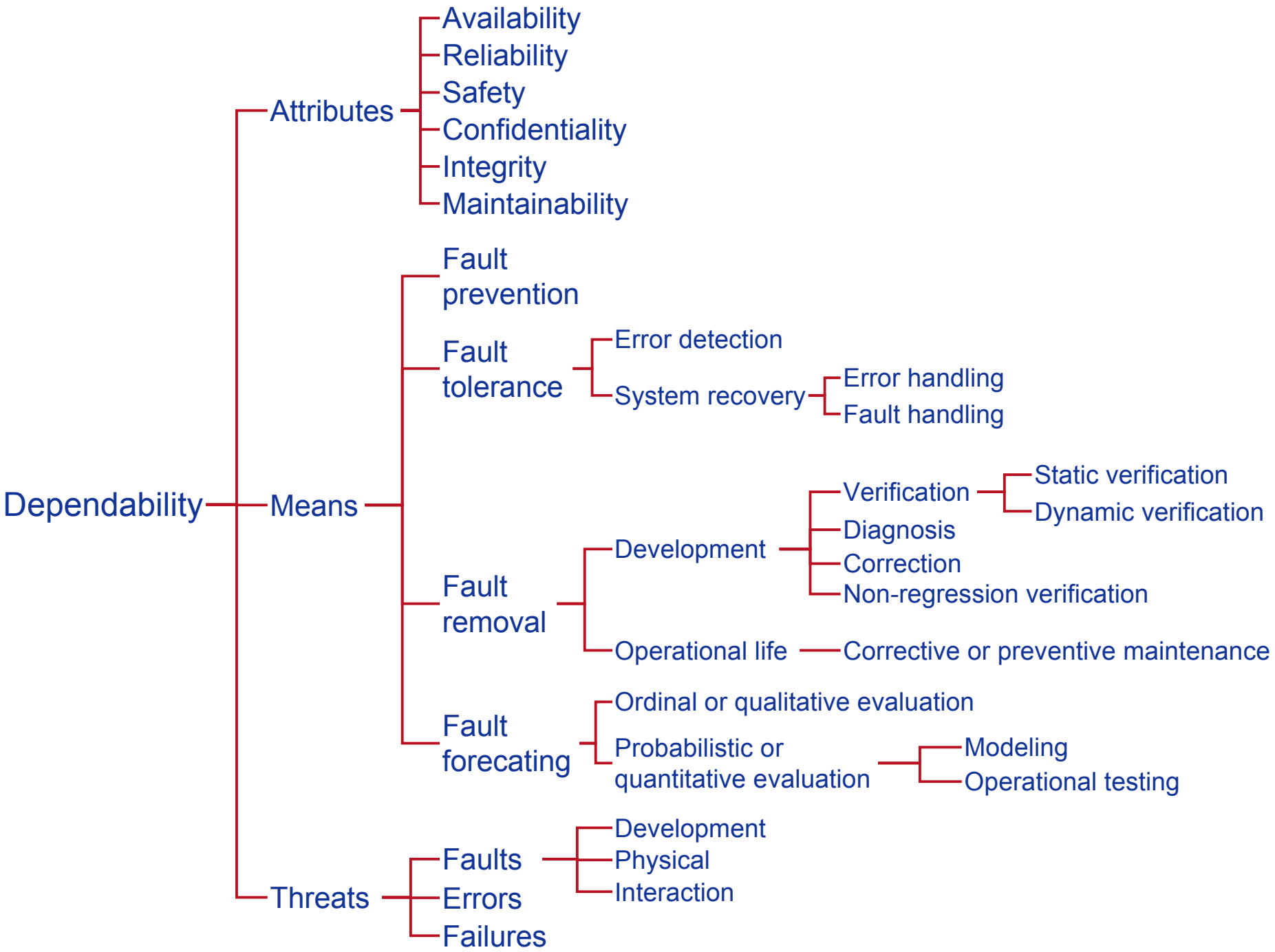
- Dependability: an integrative concept
  - Based on, and elaboration upon 'A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr, *Basic Concepts and Taxonomy of Dependable and Secure Computing*, IEEE Tr. Dependable and Secure Computing, 2004'
- Resilience: a framework for ubiquitous computing challenges
  - Based on, and elaboration upon outcomes of the European Network of Excellence ReSIST (Resilience for Survivability in Information Society Technologies)

- Conclusion

Dependability: delivery of service that can justifiably be trusted, thus avoidance of failures that are unacceptably frequent or severe




Dependability
Attributes
Availability
Reliability
Safety
Confidentiality
Integrity
Maintainability
Authorized actions
Security
Means
Fault Prevention
Fault Tolerance
Fault Removal
Fault Forecasting
Threats
Faults
Errors
Failures


Dependability
Attributes
Availability
Reliability
Safety
Confidentiality
Integrity
Maintainability
Means
Fault prevention
Fault tolerance
Error detection
System recovery
Error handling
Fault handling
Fault removal
Development
Verification
Static verification
Dynamic verification
Diagnosis
Correction
Non-regression verification
Operational life
Corrective or preventive maintenance
Fault forecating
Ordinal or qualitative evaluation
Probabilistic or quantitative evaluation
Modeling
Operational testing
Threats
Faults
Development
Physical
Interaction
Errors
Failures

Dependability definition

- First part: delivery of service that can justifiably be trusted
  - Enables to generalize availability, reliability, safety, confidentiality, integrity, maintainability, that are then attributes of dependability
- Second part: avoidance of service failures that are unacceptably frequent or severe
  - Failure frequency and severity ➜ risk assessment and management
  - A system can, and usually does, fail. Is it however still dependable? When does it become undependable?

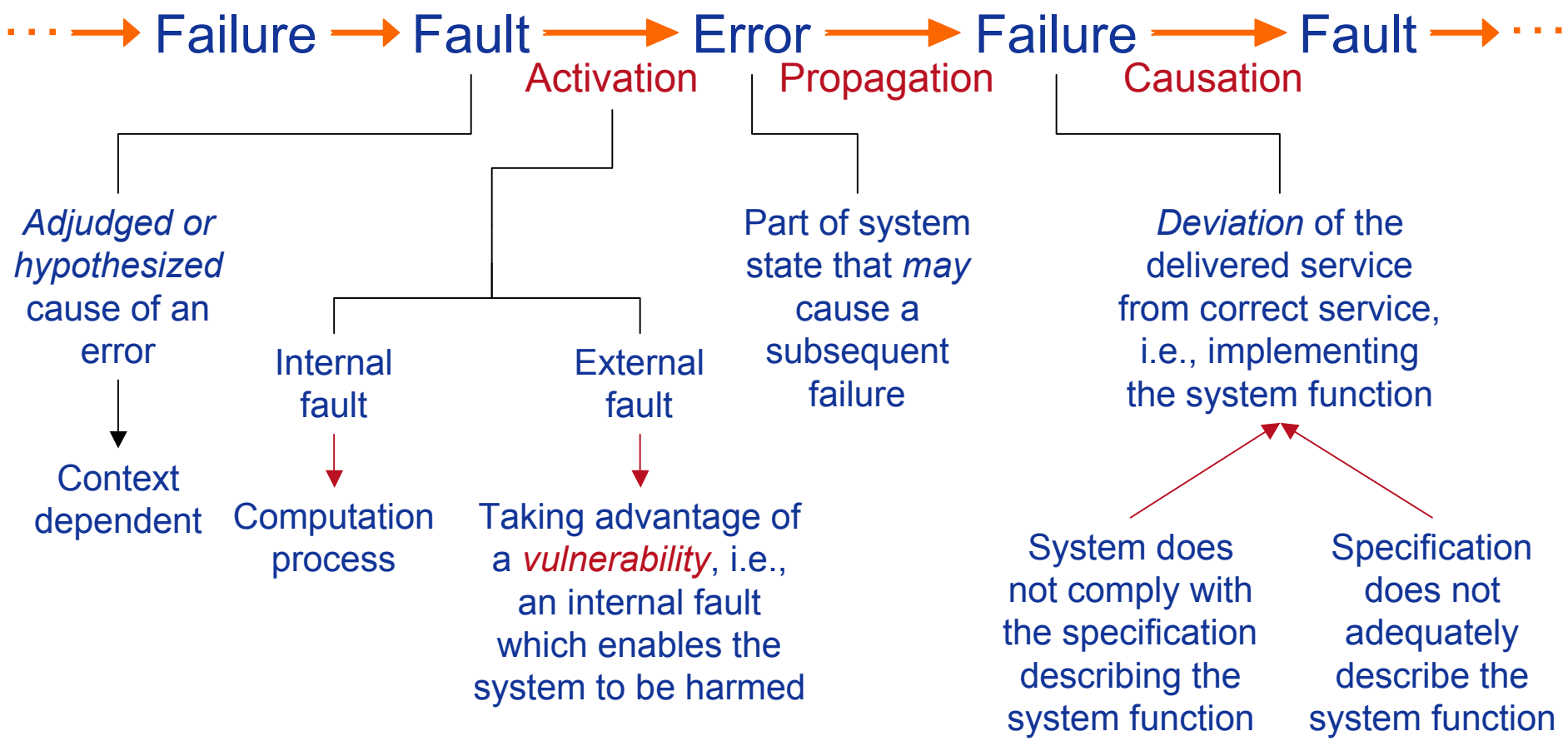
    ⬇

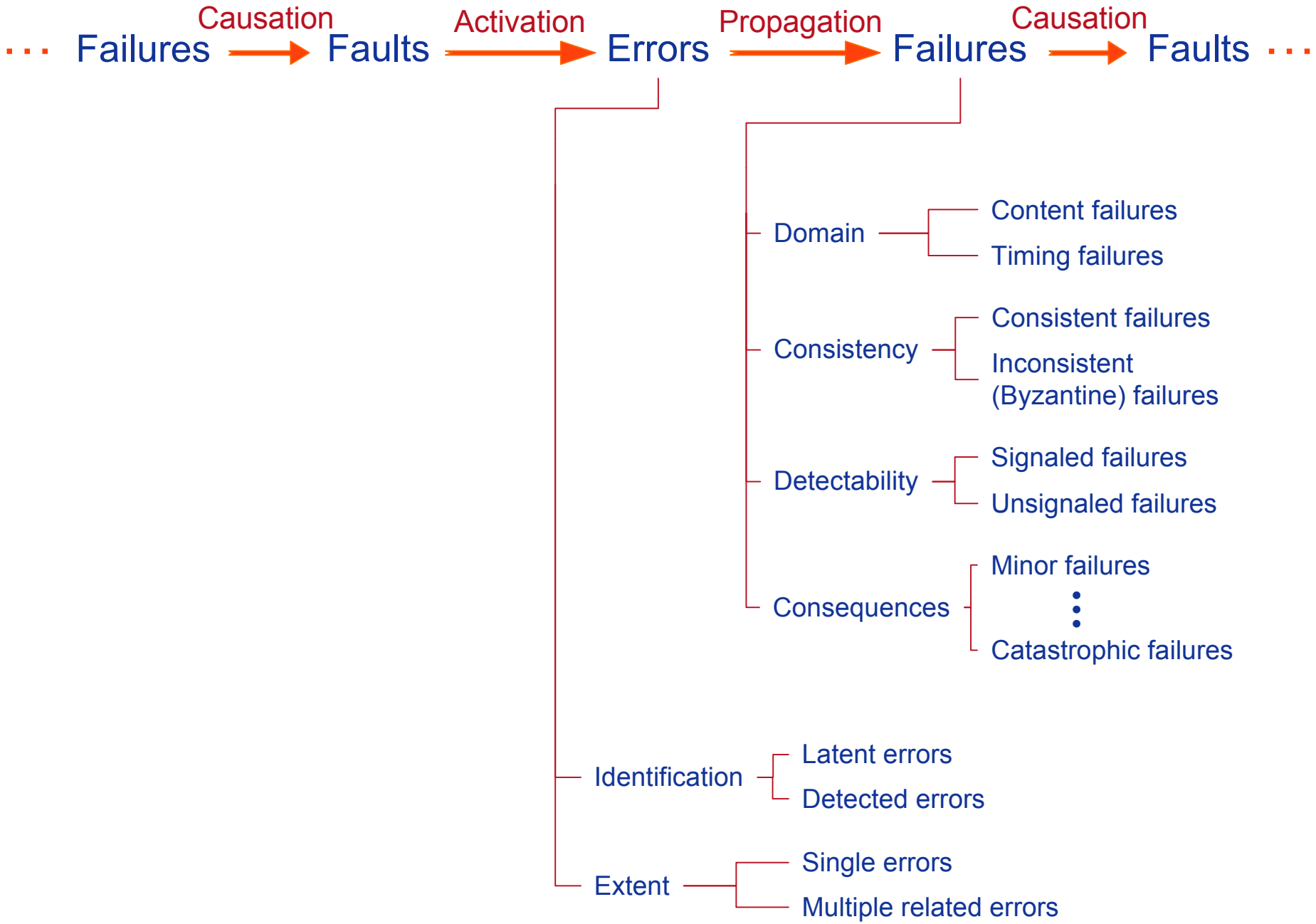
    criterion for deciding whether or not, in spite of service failures, a system is still to be regarded as dependable
  - Unacceptably frequent or severe service failures ➜ dependability failure, connected to development process failure

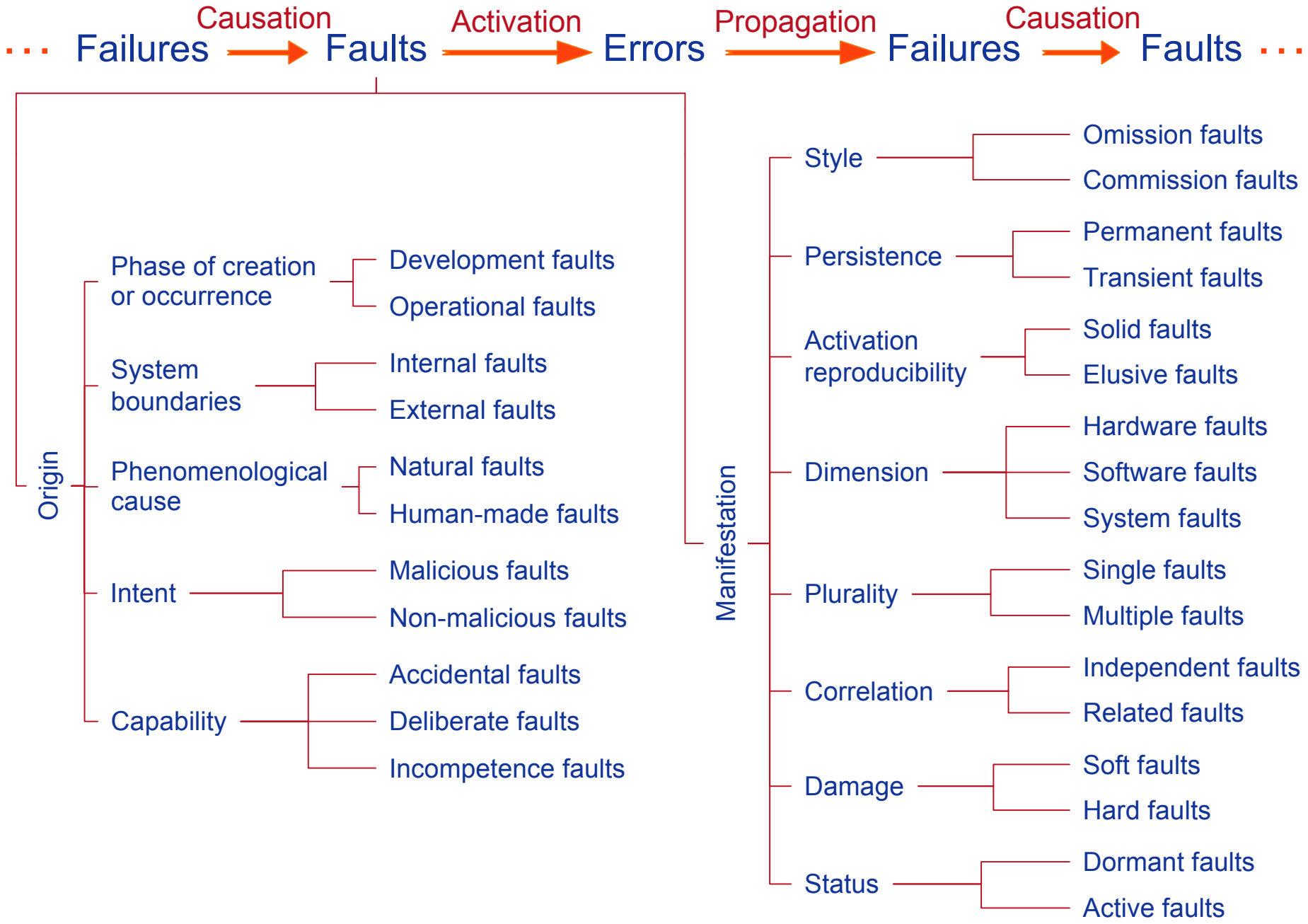
Dependability attributes

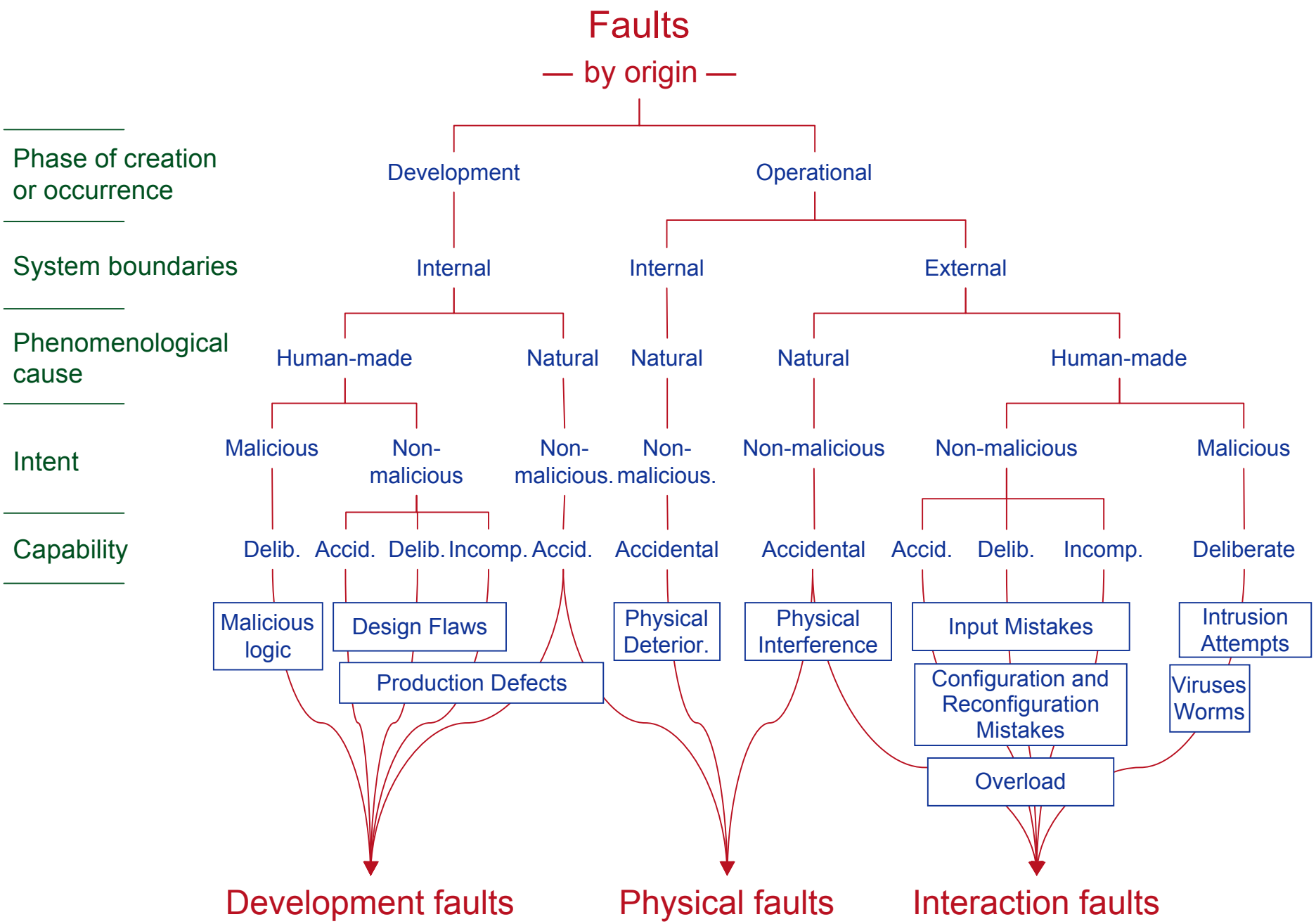
- Availability, Reliability, Safety, Confidentiality, Integrity, Maintainability: Property-based attributes
- Event-based attributes
  - Robustness: persistence of dependability with respect to external faults
  - Survivability: persistence of dependability in the presence of active fault(s)
  - Resilience: persistence of dependability when facing functional, environmental, or technological, changes
- Attributes based on distinguishing among various types of (meta-)information
  - Accountability: availability and integrity of the person who performed an operation
  - Authenticity: integrity of a message content and origin, and possibly some other information, such as the time of emission
  - Non-repudiability: availability and integrity of the identity of the sender of a message (non-repudiation of the origin), or of the receiver (non-repudiation of reception)

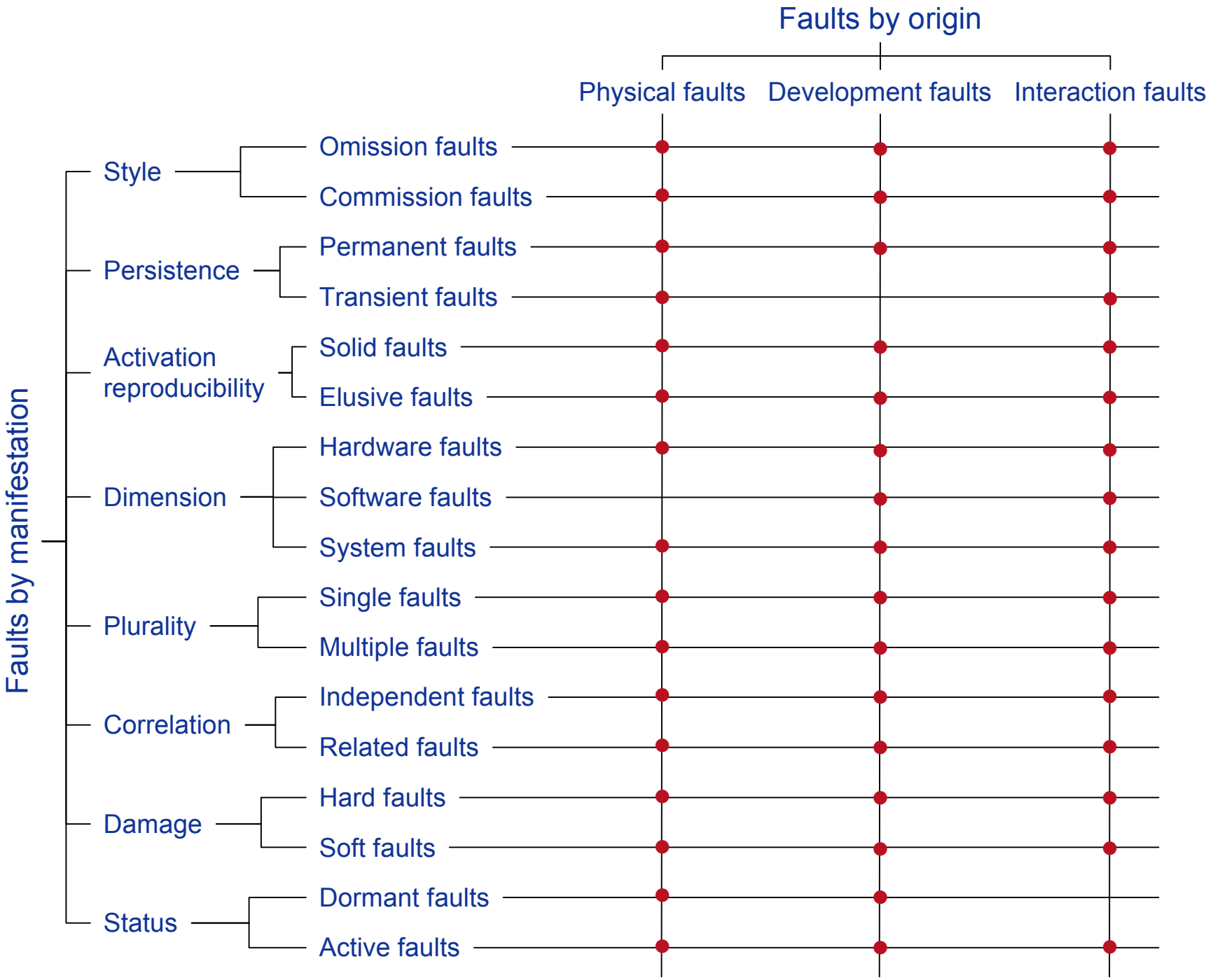
Dependability Threats




Causation
Activation
Propagation
Causation
Failures
Faults
Errors
Failures
Faults
Domain
Content failures
Timing failures
Consistency
Consistent failures
Inconsistent
(Byzantine) failures
Detectability
Signaled failures
Unsignaled failures
Consequences
Minor failures
Catastrophic failures
Identification
Latent errors
Detected errors
Extent
Single errors
Multiple related errors


Causation
Activation
Propagation
Causation
... Failures
Faults
Errors
Failures
Faults ...
Origin
Phase of creation or occurrence
Development faults
Operational faults
System boundaries
Internal faults
External faults
Phenomenological cause
Natural faults
Human-made faults
Intent
Malicious faults
Non-malicious faults
Capability
Accidental faults
Deliberate faults
Incompetence faults
Manifestation
Style
Omission faults
Commission faults
Persistence
Permanent faults
Transient faults
Activation reproducibility
Solid faults
Elusive faults
Dimension
Hardware faults
Software faults
System faults
Plurality
Single faults
Multiple faults
Correlation
Independent faults
Related faults
Damage
Soft faults
Hard faults
Status
Dormant faults
Active faults


Faults
— by origin —
Phase of creation or occurrence
System boundaries
Phenomenological cause
Intent
Capability
Development
Operational
Internal
Internal
External
Human-made
Natural
Natural
Natural
Human-made
Malicious
Non-malicious
Non-malicious.
Non-malicious.
Non-malicious
Non-malicious
Malicious
Delib.
Accid.
Delib.
Incomp.
Accid.
Accidental
Accidental
Accid.
Delib.
Incomp.
Deliberate
Malicious logic
Design Flaws
Production Defects
Physical Deterior.
Physical Interference
Input Mistakes
Configuration and Reconfiguration Mistakes
Overload
Intrusion Attempts
Viruses Worms
Development faults
Physical faults
Interaction faults

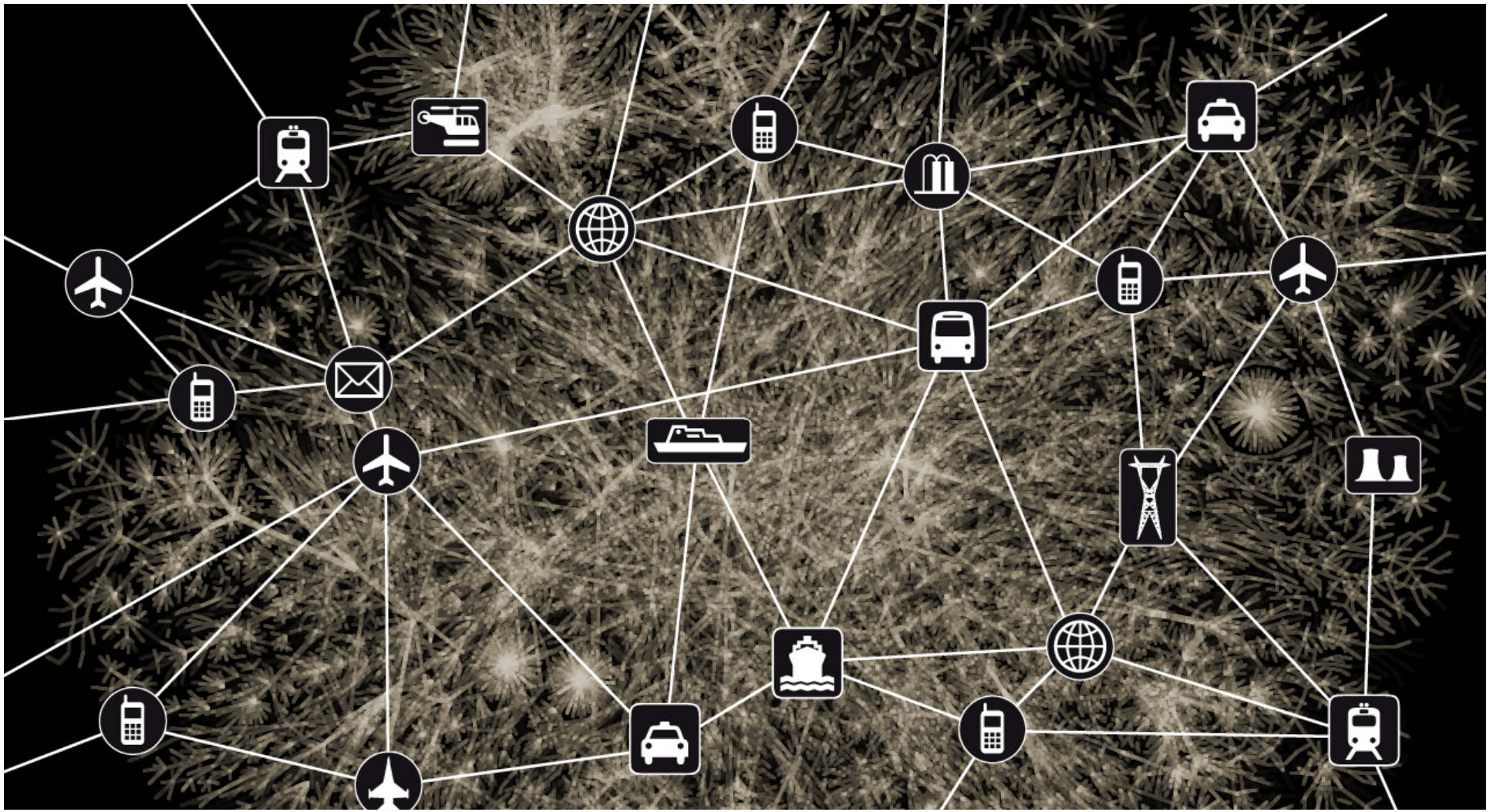

Faults by origin
Physical faults
Development faults
Interaction faults
Faults by manifestation
Style
Omission faults
Commission faults
Persistence
Permanent faults
Transient faults
Activation reproducibility
Solid faults
Elusive faults
Dimension
Hardware faults
Software faults
System faults
Plurality
Single faults
Multiple faults
Correlation
Independent faults
Related faults
Damage
Hard faults
Soft faults
Status
Dormant faults
Active faults

A few milestones

- 12th IEEE Int. Symposium on Fault-Tolerant Computing, Santa Monica, June 1982
  - Session ‘Fundamental Concepts of Fault Tolerance’, papers by A. Avizienis, H. Kopetz, J.C. Laprie and A. Costes, A.S. Robinson, T. Anderson and P.A. Lee, P.A. Lee and D. Morgan
  - Session ‘Prospective on the State-of-the-Art’, position paper by W.C. Carter
- Paper by J.C. Laprie, 15th IEEE Int. Symposium on Fault-Tolerant Computing, Ann Arbor, June 1985, *Dependable Computing and Fault Tolerance: Concepts and Terminology*
- Publication by J.C. Laprie of the book *Dependability: Basic Concepts and Terminology — in English, French, German, Italian, Japanese*, Springer Verlag, 1992, vol. 5 of the series ‘Dependable Computing and Fault-Tolerant Systems’, A. Avizienis, H. Kopetz, J.C. Laprie, eds.
- Paper by A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr, *Basic Concepts and Taxonomy of Dependable and Secure Computing*, IEEE Transactions on Dependable and Secure Computing, 2004

- Strength of the dependability concept: integrative role
  - Enables the more classical notions of reliability, availability, safety, confidentiality, integrity, maintainability, etc. to be put into perspective, as attributes of dependability
  - Model provided for the means for achieving dependability
    - Those means are much more orthogonal to each other than the more classical classification according to the attributes of dependability
    - Facilitates the unavoidable trade-offs, as the attributes tend to conflict with each other
  - Fault - Error - Failure model
    - Understanding and mastering the threats
    - Unified presentation of the threats, while explaining their specificities via the various fault classes
- Concept and terminology widely accepted by international community
  - IEEE Fault-Tolerant Computing Symposium + IFIP Dependable Computing for Critical Applications Conference ➠ IEEE/IFIP Conference on Dependable Systems and Network
  - Regional (Europe, Pacific Rim, South America) Fault Tolerant Computing Conferences ➠ Dependable Computing Conferences

Ubiquitous systems: future large, networked, continuously evolving systems constituting complex information infrastructures —perhaps involving everything from super-computers and huge server "farms" to myriads of small mobile computers and tiny embedded devices



At stake: Maintain dependability in spite of changes

Resilience: persistence of dependability when facing changes

Nature

- Functional
  [incl. Specification faults]
- Environmental
  [incl. interaction faults]
- Technological
  [incl. development and physical faults]

Prospect

- Foreseen
  [e.g., new versioning]
- Foreseeable
  [e.g., advent of new hardware platforms]
- Unforeseen
  [e.g., drastic changes in servivce requests or new types of threats]

Timing

- Short term
  [e.g., second to hours, as in dynamically changing systems]
- Medium term
  [e.g., hours to months, as in reconfigurations or new versioning]
- Long term
  [e.g., months to years, as in reorganisations resulting from mergers, or in military coaltions]

Threat evolution

- Attacks
- Mismatches in evolutionary changes
- Side-effects in emerging behaviors
- Increasing importance of configuration faults
- Increasing proportion of hardware transient faults

Resilience

➠ in dependability and security of computing systems

- Adjective Resilient
  - In use for 30+ years
  - Recently, escalating use ➔ buzzword
  - Used essentially as synonym, or substitute, to fault tolerant
  - Noteworthy exception: preface of *Resilient Computing Systems*, T. Anderson (Ed.), Collins, 1985

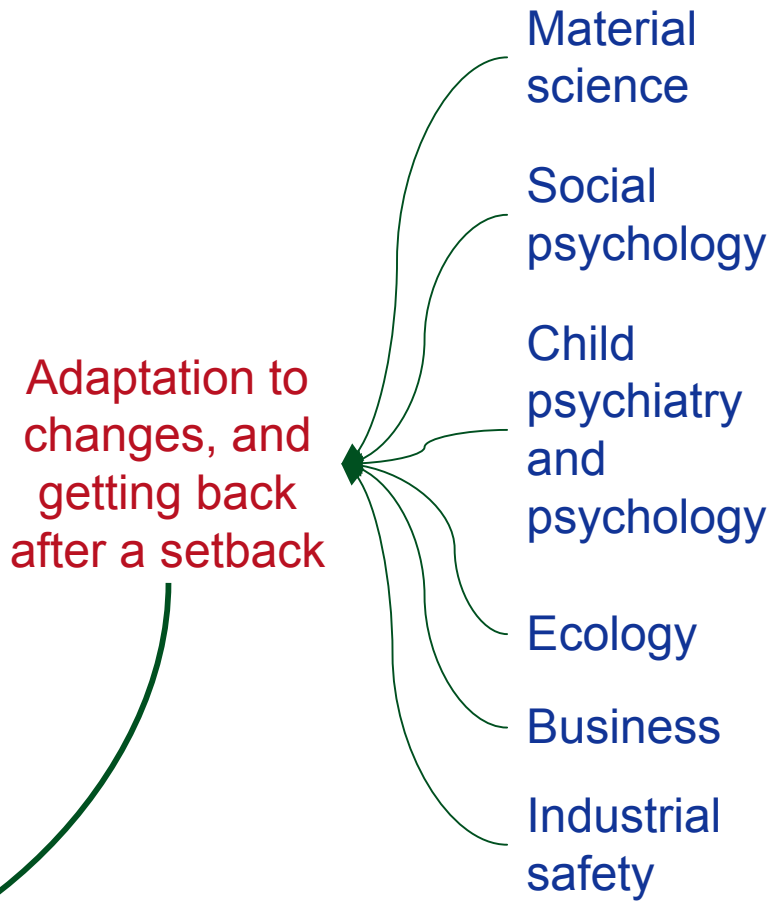
    «The two key attributes here are dependability and robustness. […] A computing system can be said to be *robust* if it retains its ability to deliver service in conditions which are beyond its normal domain of operation»

- Generalisation of fault tolerance

  ⬇

  change tolerance

➠ in other domains



Technologies for resilience

Changes → Evolvability

- ☞ Self evolvability: adaptation

Trusted service → Assessability

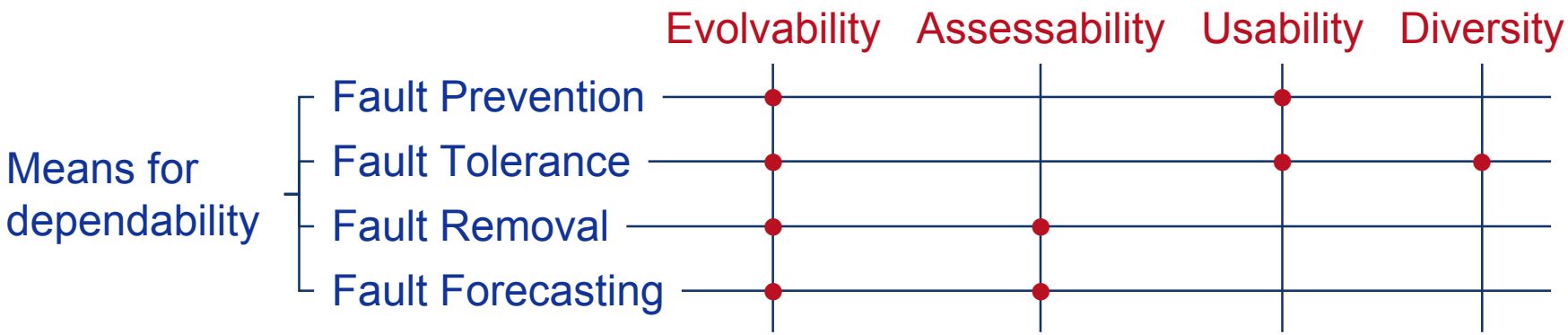
- ☞ Verification and evaluation

Ambient intelligence → Usability

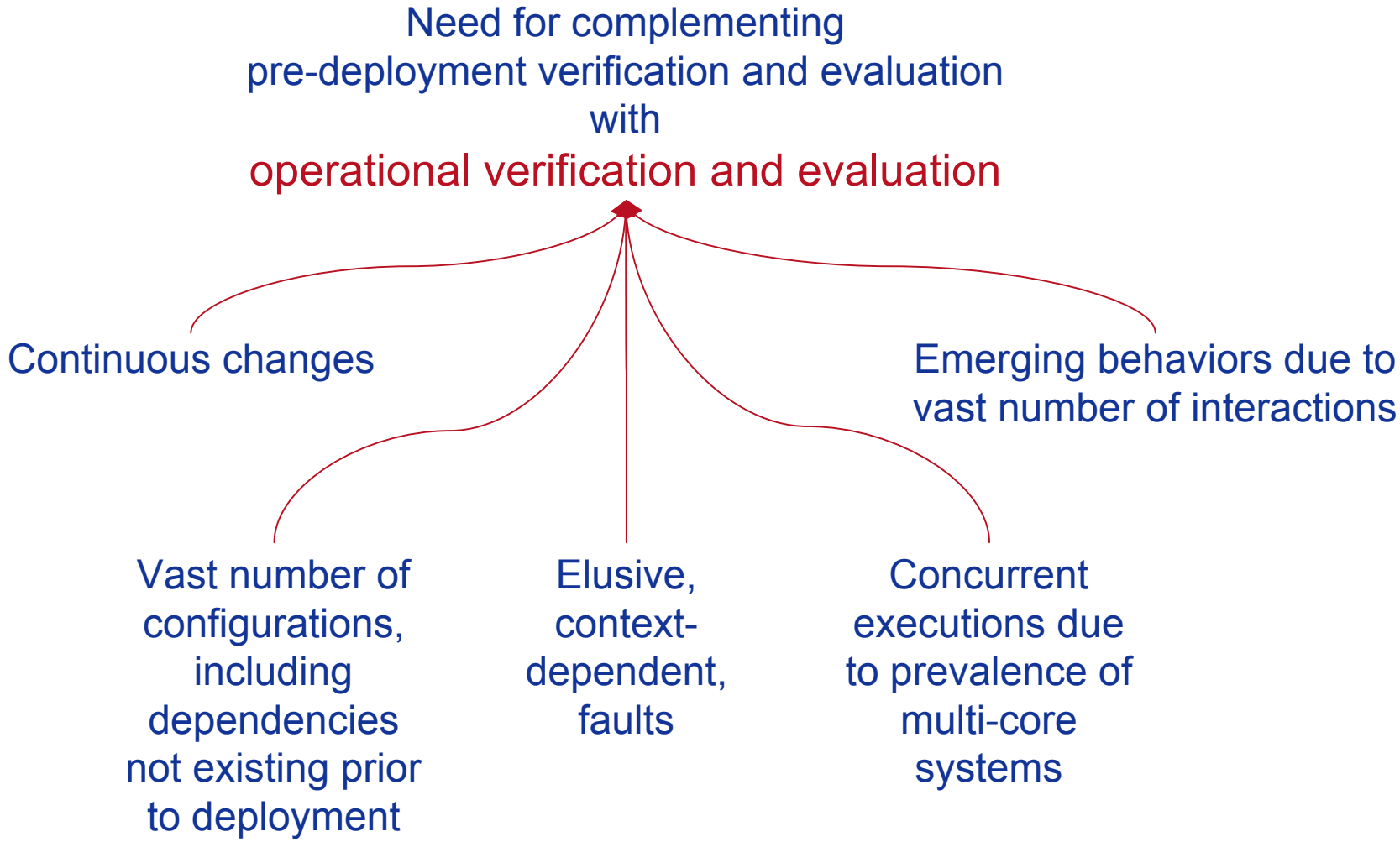
- ☞ Human and system users

Complex systems → Diversity

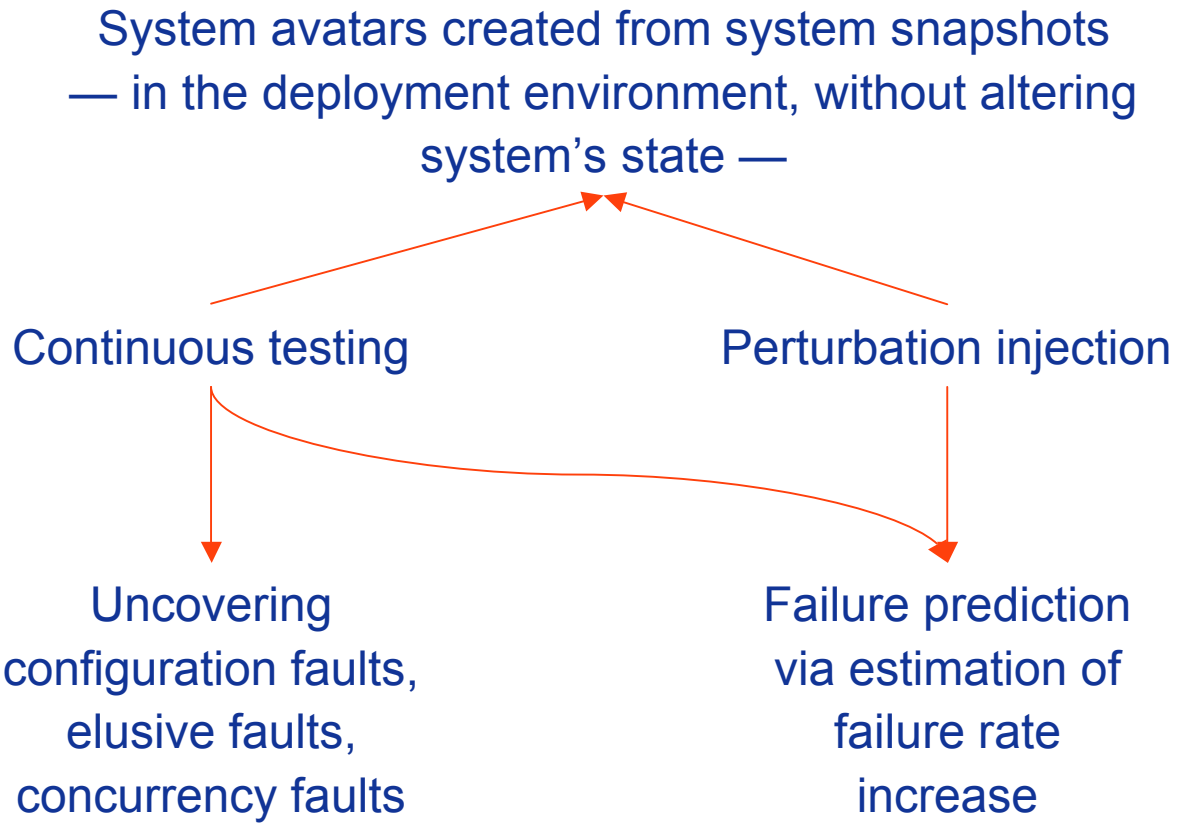
- ☞ Taking advantage of existing diversity, and augmenting it



Assessability



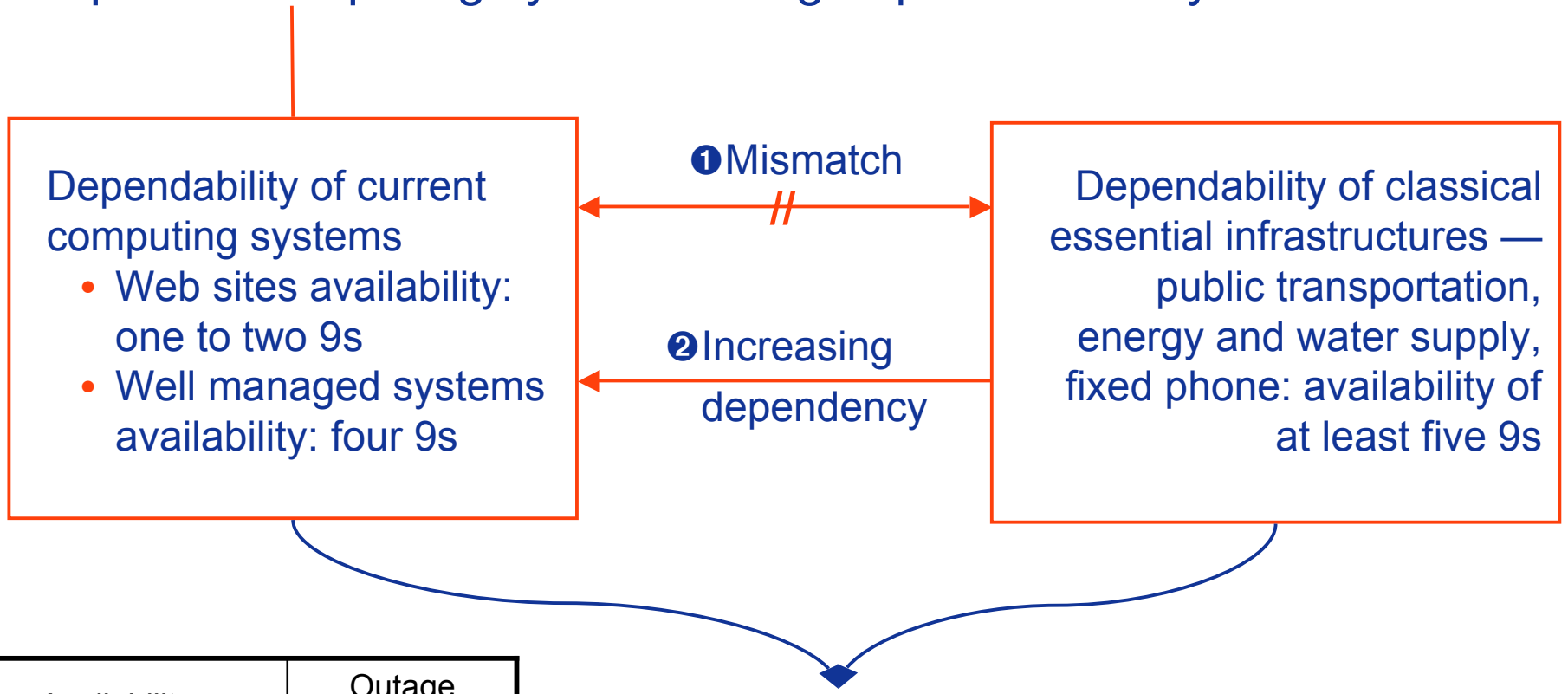
Project on operational assessment under elaboration



Expectations of the computing science and engineering community

- 50th anniversary issue of Communications of the ACM (January 2008) : most articles mention dependability or resilience as a major factor (Jeannette Wing, Rodney Brooks, Gul Agha, John Crowcroft, Gordon Bell)
  - Rodney Brooks : « New formalisms will let us analyze complex distributed systems, producing new theoretical insights that lead to practical real-world payoffs. Exactly what the basis for these formalisms will be is, of course impossible to guess. My own bet is on resilience and adaptability »
- March 2008 issue of IEEE Computer, feature section devoted to software engineering in the 21st century
  - Barry Boehm : « In the 21st century, software engineers face the often formidable challenges of simultaneously dealing with rapid change, uncertainty and emergence, dependability, diversity, and interdependence »

Ubiquitous computing systems: integral part of society



Availability		Outage duration/year
Six 9s	0,999999	32s
Five 9s	0,99999	5mn 15s
Four 9s	0,9999	52mn 34s
Three 9s	0,999	8h 46mn
Two 9s	0,99	3j 16h
One 9	0,9	36j 12h