U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Dependable Systems

Basic Concepts and Terminology

Basic Concepts and Terminology

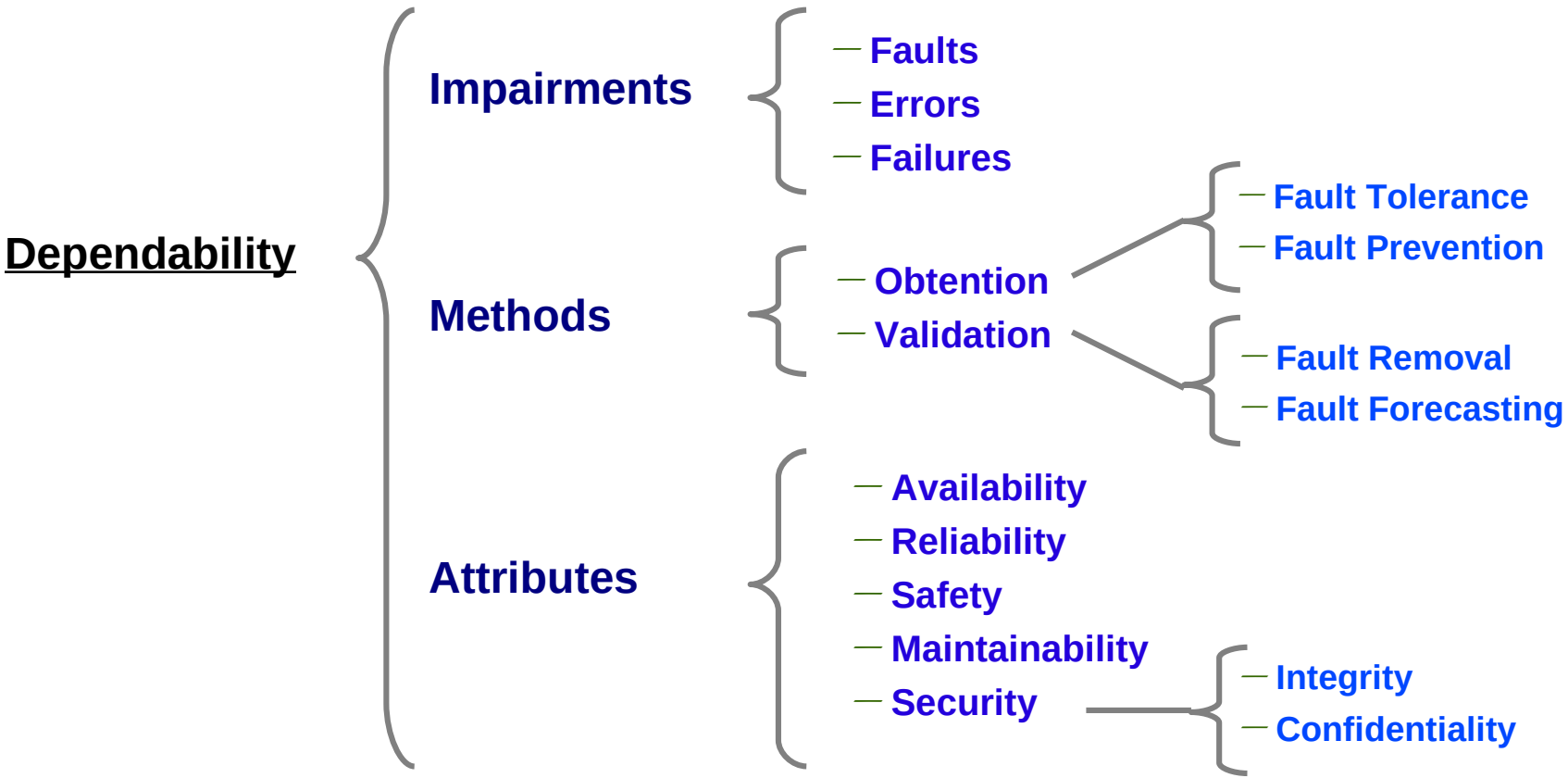
- Basic Definitions
- Faults, Errors and Failures
- Dependability Attributes
- Means of Attaining Dependability
- Computational Systems and Safety

Basic Definitions

- Dependability:
  - "[..] the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers [..]" (IFIP WG10.4 definition)
  - Ability to avoid service failures that are more frequent or more severe than is acceptable.

A system can, and usually, does fail. When does it become undependable?
For each system/user, we need criteria for deciding
when the system becomes undependable.

Basic Definitions



Basic Definitions

- Dependability Impairments
  - The Failure of a system occurs when the service provided is no longer in accordance with the specification;
    - This definition was later changed in order to include the behaviours which, although satisfying the specification, are unacceptble to the system's users

      (due to a fault in the specification).
  - Error is a state of the system that may lead to a failure.
  - The hypothetical cause of an error is a Fault.

Basic Definitions

- Methods of obtaining Dependability:
  - fault prevention → don’t insert faults
  - fault tolerance → tolerate existing faults
  - fault removal → find and remove any faults
  - fault forecasting → forecast remaining unknown faults

- The Dependability attributes
  - allow us to express the properties that are desired from the system;
  - allow us to quantitatively evaluate the system's qualities

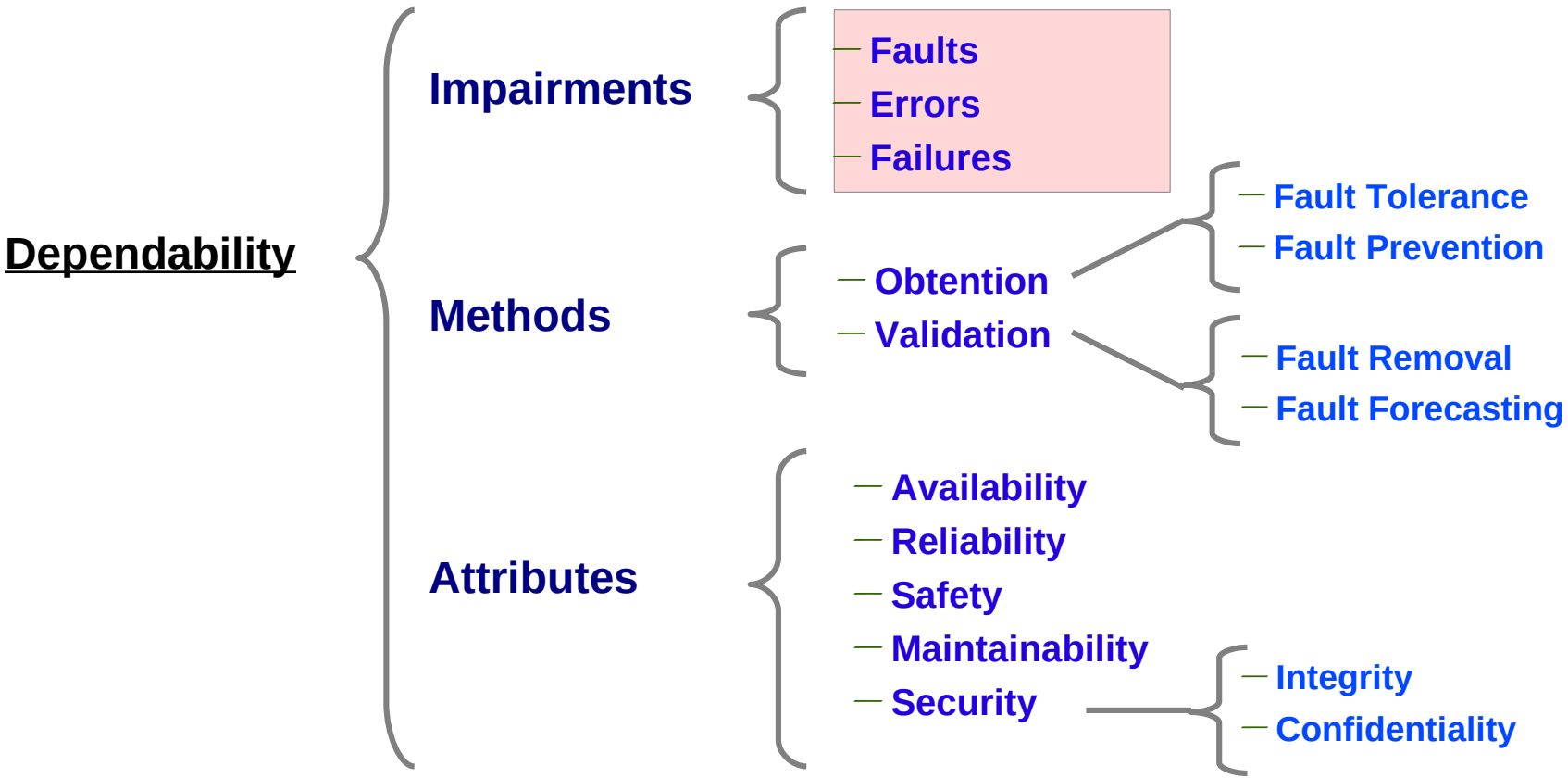
Basic Definitions

- Dependability attributes:
  - *Availability* → readiness to provide the desired service
  - *Reliability* → continuity of providing the service
  - *Safety* → capability of avoiding catastrophic failures
  - *Maintanibility* → ease of being corrected
  - *Confidentiality* → avoid unauthorized information disclosure (control reading)
  - *Integrity* → avoid unauthorized information alteration (control writing)

Basic Concepts and Terminology

- Basic Definitions
- Faults, Errors and Failures
- Dependability Attributes
- Means of Attaining Dependability
- Computational Systems and Safety

Basic Definitions



Faults, Errors and Failures

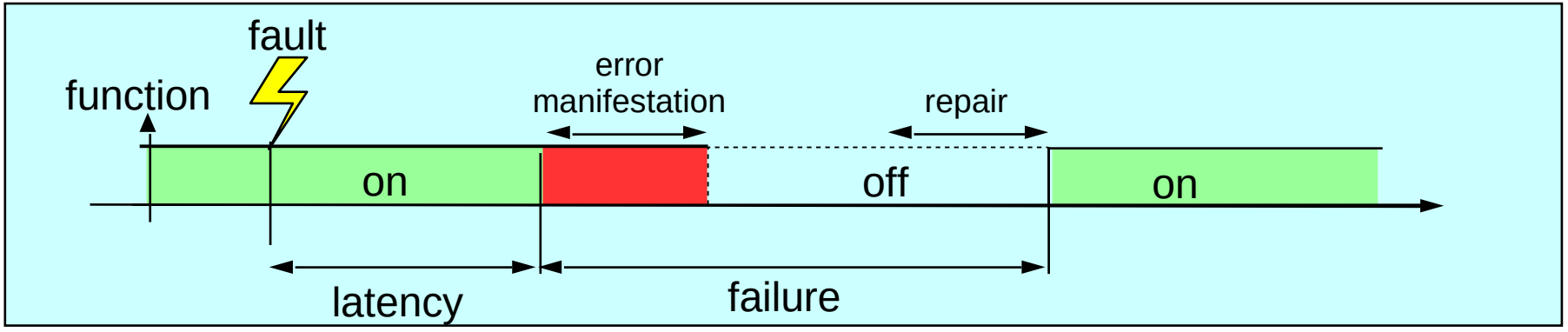
- *Mission*:
  - desired device function (including correct specification)

- *Failure*:
  - the non-fulfillment of the mission
  - this may be:
    - Momentary, Tempory (e.g. due to repairs), or Permanent
    - there are many other ways to classify these...

Faults, Errors and Failures

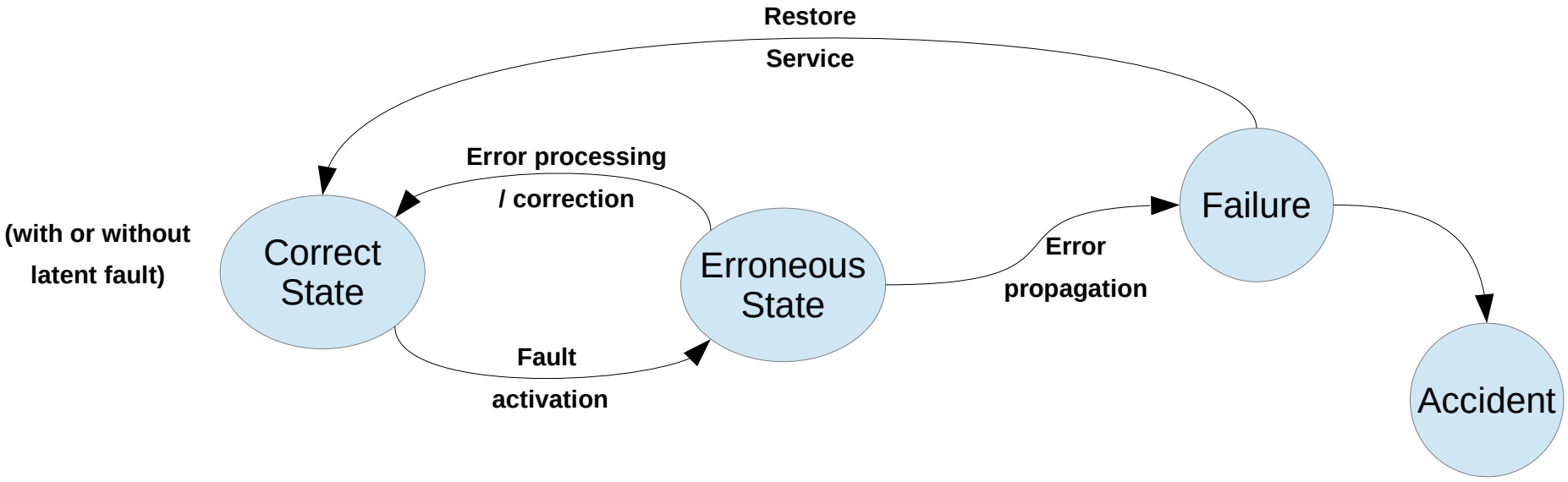
- *Fault*:
  - the cause of an error
    (may occur long before the failure itself)
- *Error*:
  - the manifestation of the fault
    - An error may (and usually does) propagate, causing other errors.
- Failure
  - When an error transposes the user/system interface, ...

Faults, Errors and Failures



FAULT → may cause → ERROR → may cause → FAILURE

Faults, Errors and Failures



FAULT may cause ERROR may cause FAILURE

Faults, Errors and Failures

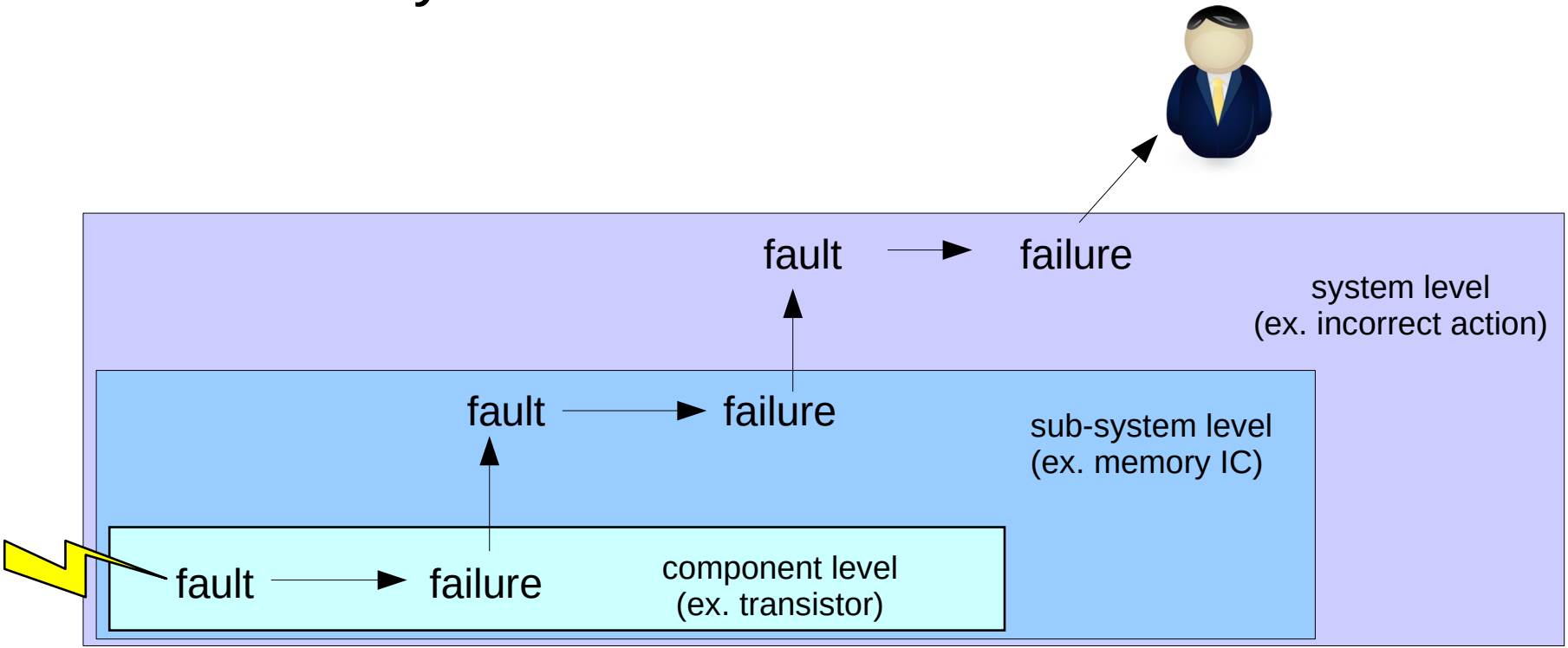
- Examples of a **Fault → Error → Failure** chain...
  - A software 'bug' introduced by a programmer is a fault
    - Once activated the fault may produce an error
      (ex.: activation → invocation of function containing the 'bug'),
    - The error is an incorrect internal state
      (ex.: error → variable with incorrect value);
    - A failure occurs if/when the incorrect internal state affects the service being provided (either in value or time domain).
  - An electromagnetic interference / cosmic ray is a fault;
    - this fault may result in an internal error
      (error → flipping of bits in communication wire / memory);
    - these errors may result in failures
      (failure → when reading the data of the wire / memory);

Faults, Errors and Failures

- Examples of a **Fault → Error → Failure** chain...
  - An incorrect operation made by a user is a fault
    (from the point of view of the overall system)
    - this fault may result in an incorrect internal state (error)
      (ex.: error → the vehicule accelerates instead of breaking)
    - The error results in a failure
      (ex. failure → vehicule crashes into garage door)
  - A failure by the editor of a maintenance or user manual
    - may result in a fault becoming present in the manual
      (ex. Apollo 11 moon landing procedures);
    - This fault will remain inactive until someone follows the incorrect procedures detailed in the manual (ex. rendezvous radar switched on instead of off);
    - The incorrect procedures leads to a system failure (ex. software crashes);

Faults, Errors and Failures

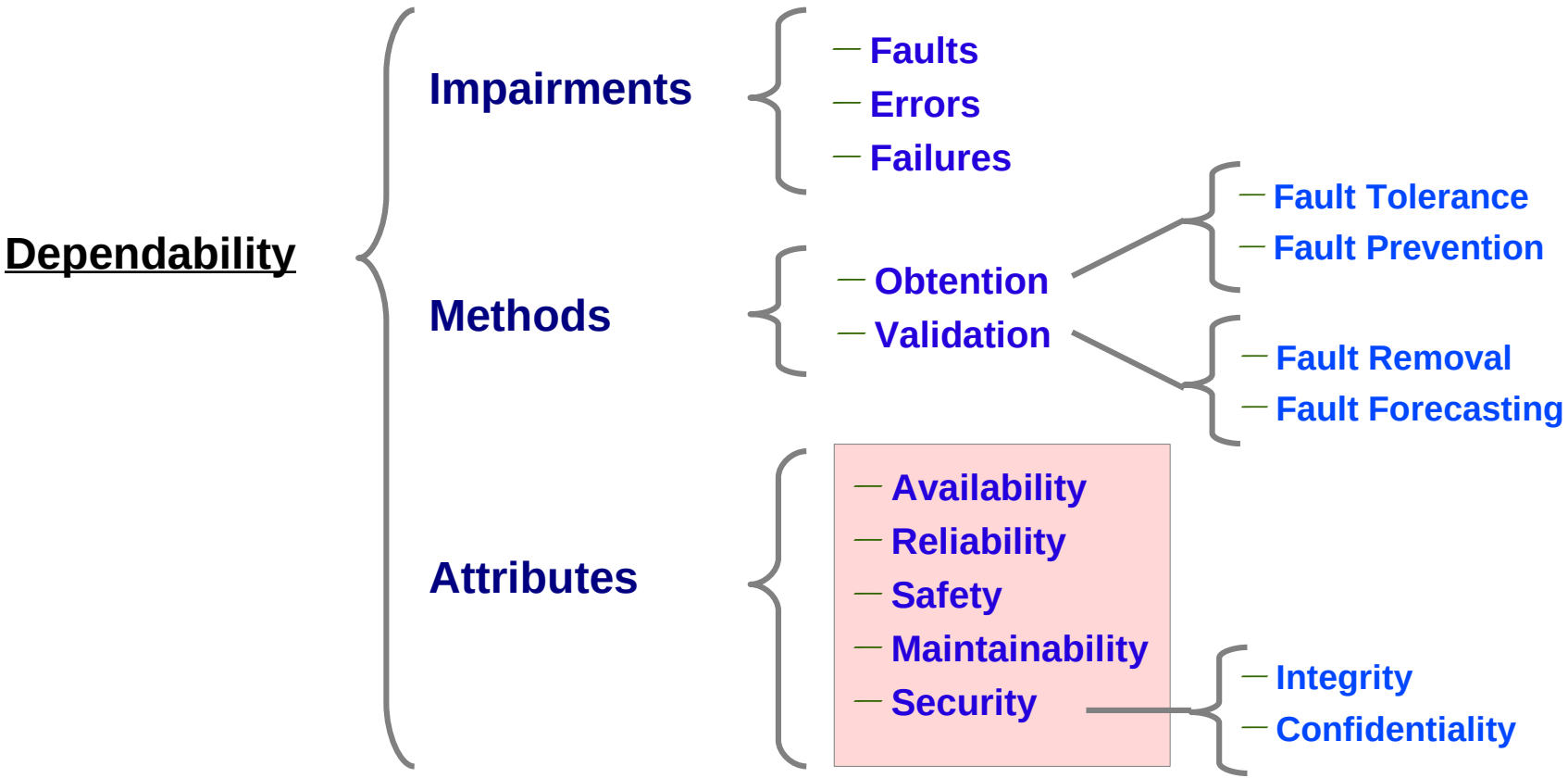
- Hierarchy of faults and failures



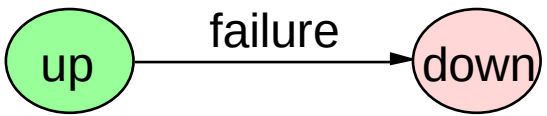
Basic Concepts and Terminology

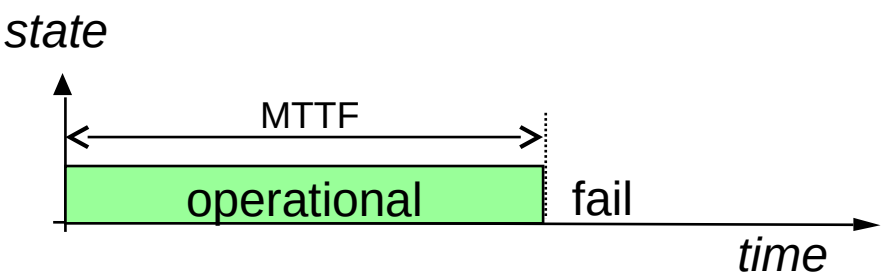
- Basic Definitions
- Faults, Errors and Failures
- Dependability Attributes
- Means of Attaining Dependability
- Computational Systems and Safety

Basic Definitions



Dependability Attributes: Reliability

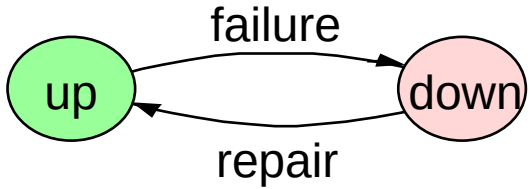


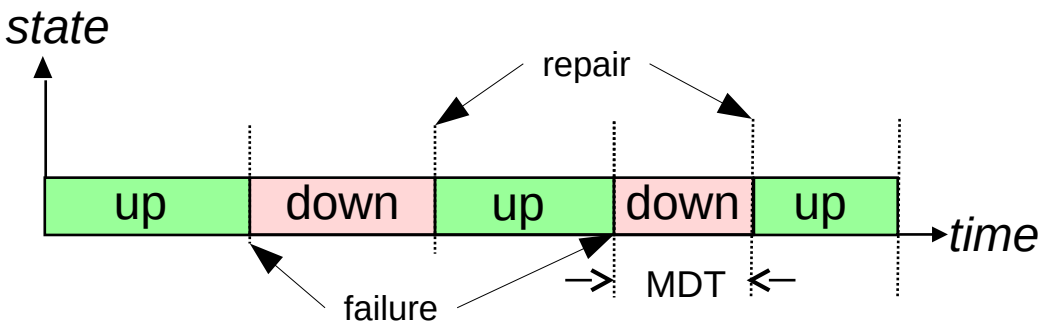


- Definition: R(t)
  ”probability that an item will perform its required function in the specified manner and under specified or assumed conditions *over a given time period*”

- Sometimes expressed as MTTF:
  Mean Time To Failure
  (years, days, hours, seconds, ...)

Dependability Attributes: Availability





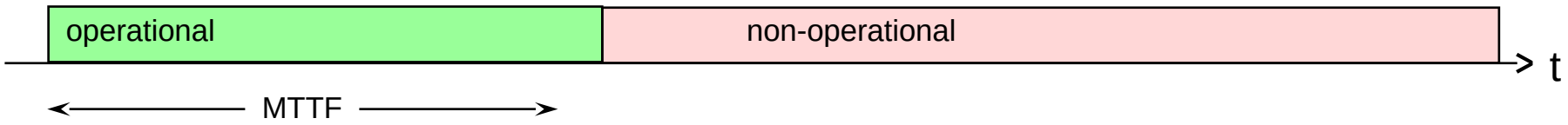
- Definition:
  ”probability that an item will perform its required function in the specified manner and under specified or assumed conditions *at a given time*”

- Expressed as: A
  proportion (eg. in %)
  of time that an item is 'up'.

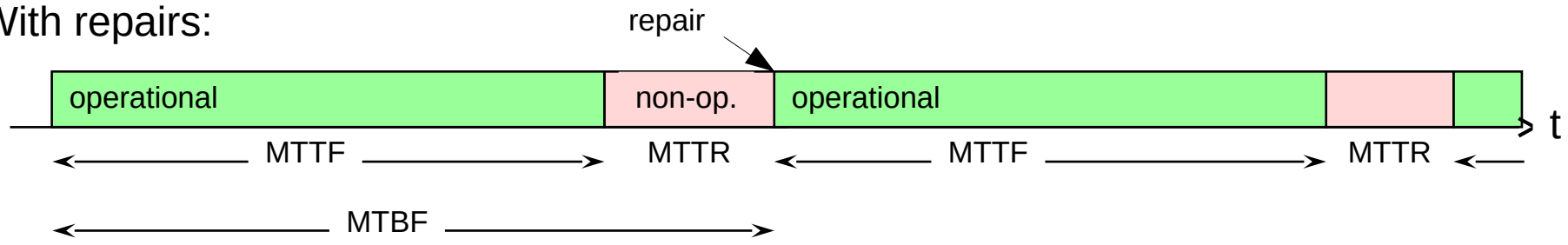
MDT - Mean Down Time

Dependability Attributes: Reliability vs Availability

Without repairs:



With repairs:



MTBF – Mean Time Between Failures
MTTF – Mean Time To Fail
MTTR – Mean Time To Repair

NOTE: sometimes MTBF – Mean Time Before Failures == MTTF

MTBF = MTTF + MTTR

if MTTF >> MTTR
then MTBF ≈ MTTF

Dependability Attributes: Reliability vs Availability

- Example:
  - A system that fails, on average, once an hour, but that reboots automatically in 10 ms, has low reliability, but high availability.
- Example of systems that require high availability, but low reliability:
  - stateless web servers, ...

Dependability Attributes: Safety

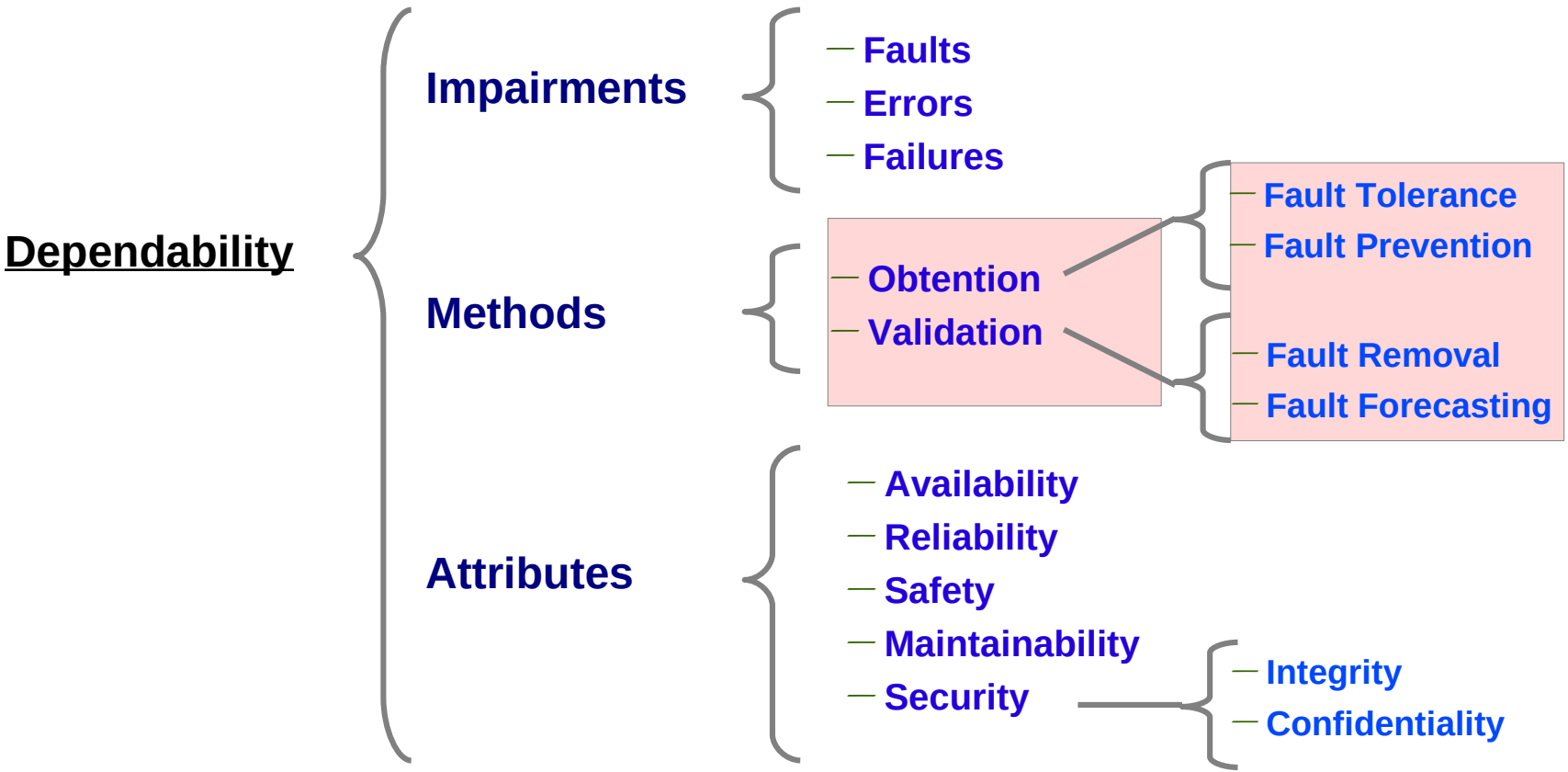
- Definition:
  - “safety is defined to be the absence of catastrophic consequences on the environment”.
  - “freedom from accidents and loss” [Leveson 95]

- What is a safety critical system?
  - this is a system for which safety (absence of catastrophic failures) is guaranteed to a specific value.

Basic Concepts and Terminology

- Basic Definitions
- Faults, Errors and Failures
- Dependability Attributes
- Means of Attaining Dependability
- Computational Systems and Safety

Basic Definitions



Means of Attaining Dependability

- Fault Prevention:
  Do whatever it takes to prevent the faults from occurring, or from introducing them during design and development.
  In software Development, this usually means:

  - Formal, Semi-formal, and engineering methods to be used and applied during software development;
  - These methodologies are to be used throughout the software's life-cycle (requirements analysis, specification, design, implementation, verification, validation, ...)

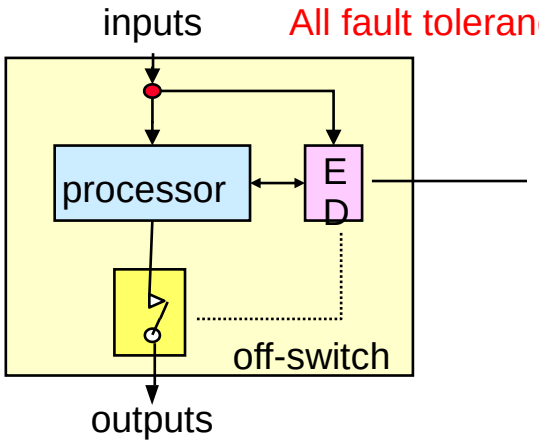
Means of Attaining Dependability

- Fault Tolerance:
  No matter how hard we try, we will never be able to prevent all faults from occurring. Therefore, the next step is to tolerate the occurrence of faults, i.e. allow the system to continue operating correctly even in the presence of faults that have caused an error.
  This is achieved through redundancy.
  - Hardware redundancy;
  - Software redundancy / diversity;
    (time diversity, data diversity, design/implementation diversity)

Means of Attaining Dependability

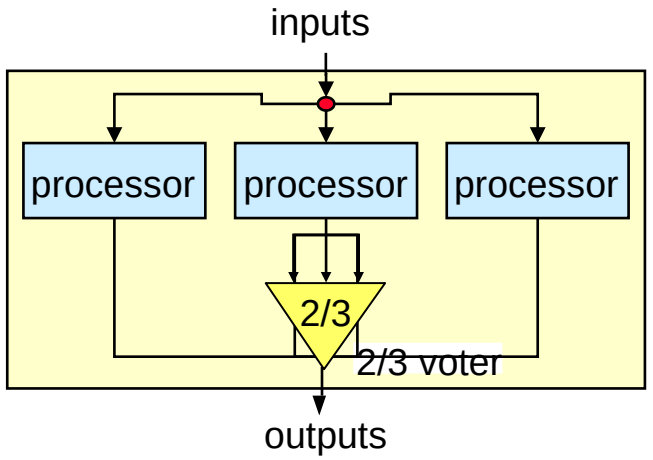
- Fault Tolerance: the three main architectures

All fault tolerance mechanisms go through several stages...



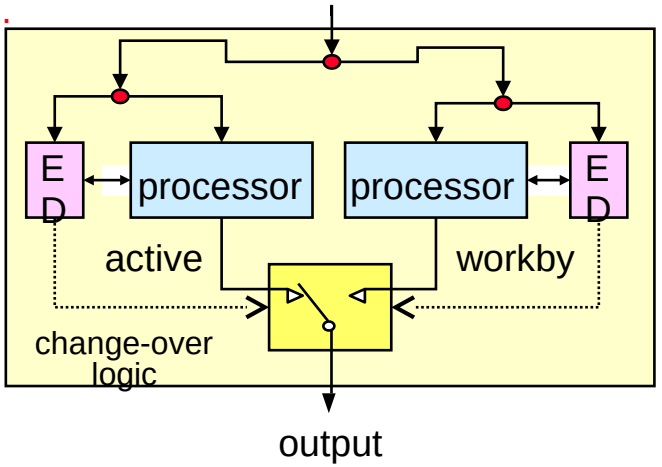
a) Integer

" rather nothing than wrong "
(fail-silent, fail-stop, fail-safe)



c) Integer & persistent

error masking, massive redundancy



b) Persistent

" rather wrong than nothing "
fail-operate

Means of Attaining Dependability

- Fault Removal:
  Reduce the number and severity of faults

  - During System Use
    This is essentially Preventative or Corrective Maintenance;
  - During System Development
    consists of three steps: verification, diagnosis, and correction.
    verification : the process of checking whether the system adheres to given properties (the term validation is often used when checking the specification).

Means of Attaining Dependability

- Fault Forecasting:
  Estimate the number and consequence of any remaining faults that we were unable to prevent and remove.
  - How do you do this for software?

Basic Concepts and Terminology

- Bibliography

- "Safety-Critical Computer Systems",
  Neil Storey, Chapters 1 and 2
- "Fundamental Concepts of Dependability",
  A. Avizienis, J-C. Laprie, B. Randell
- "Dependability and its Threats: A Taxonomy",
  A. Avizienis, J-C. Laprie, B. Randell
- "Quality Attributes", M. Barbacci, T. H. Longstaff, M. H. Klein, C. B. Weinstock, , CMU/SEI-95-TR-021, December 1995.
- "Quality Attributes and Service-Oriented Architectures",
  L. O'Brien, L. Bass, P. Merson, CMU/SEI-2005-TN-014, September 2005.