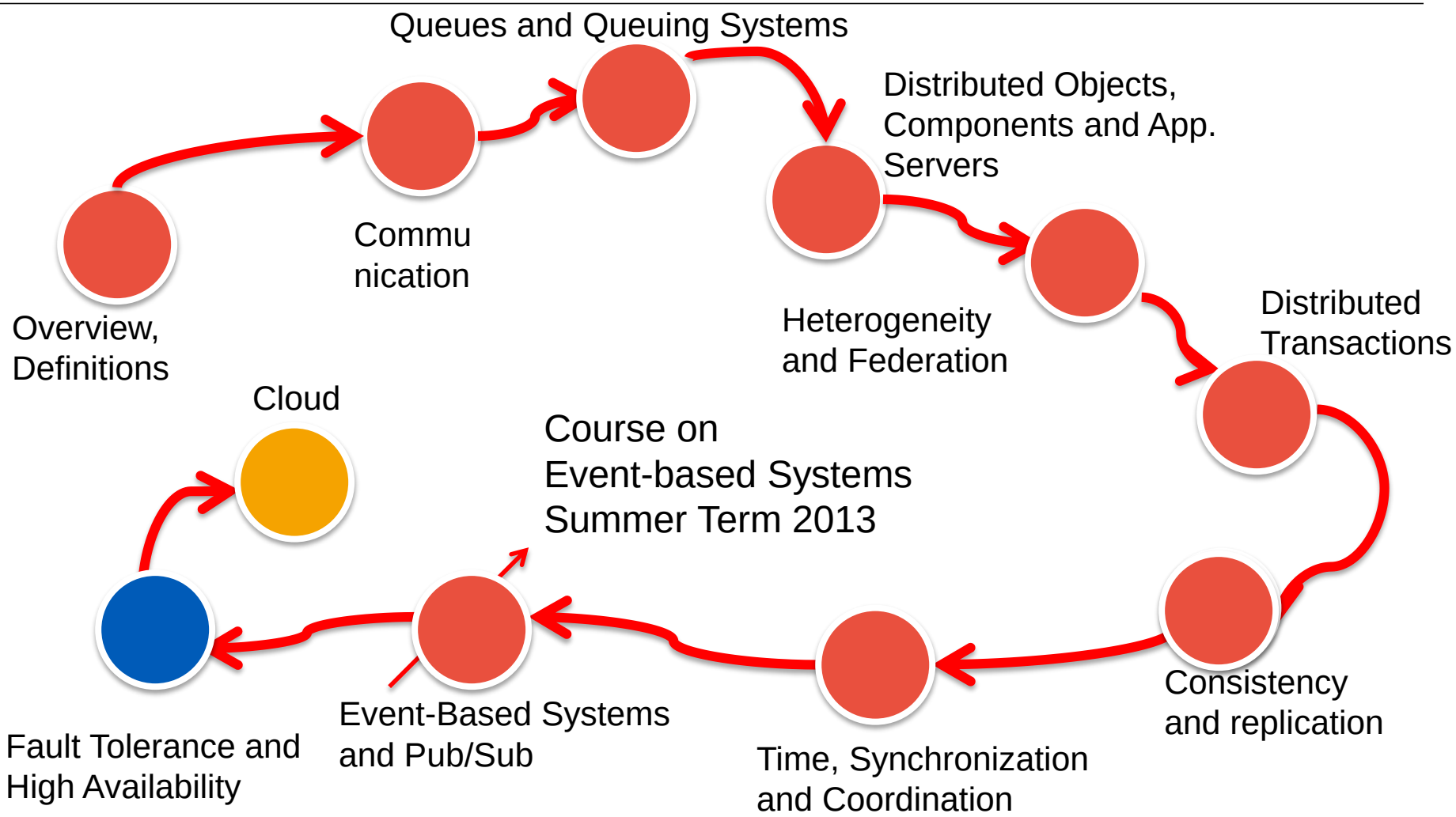
# Middleware:
## 11. Fault Tolerance

TECHNISCHE UNIVERSITÄT DARMSTADT

**A. Buchmann**
**Wintersemester 2011/2012**

# Topics


Queues and Queuing Systems
Distributed Objects, Components and App. Servers
Commu nication
Overview, Definitions
Heterogeneity and Federation
Distributed Transactions
Cloud
Course on Event-based Systems Summer Term 2013
Consistency and replication
Event-Based Systems and Pub/Sub
Time, Synchronization and Coordination
Fault Tolerance and High Availability

# Reading for THIS Lecture

TECHNISCHE UNIVERSITÄT DARMSTADT

- The slides for the lecture are based on material from:
  - Andrew S. Tanenbaum and Maarten Van Steen. 2001. **Distributed Systems: Principles and Paradigms.** Prentice Hall.
    - Chapter 7
  - George Coulouris, Jean Dollimore, and Tim Kindberg. 2005. **Distributed Systems: Concepts and Design.** Addison-Wesley Longman.
    - Chapter 16
  - Jim Gray, Andreas Reuter
    - **Dependable Computing Systems**.
  - M. Tamer Özsu, Patrick Valduriez
    - **Principles of Distributed Database Systems**, 3rd Ed., Springer
    - Chapter 12

# Fault Tolerance Basic Concepts

TECHNISCHE UNIVERSITÄT DARMSTADT

- Fault tolerance is strongly related to dependable systems

- Dependability:
  - Availability – system is available to serve user requests at any given time
  - Reliability – system runs continuously, without failure
  - Safety
  - Maintainability

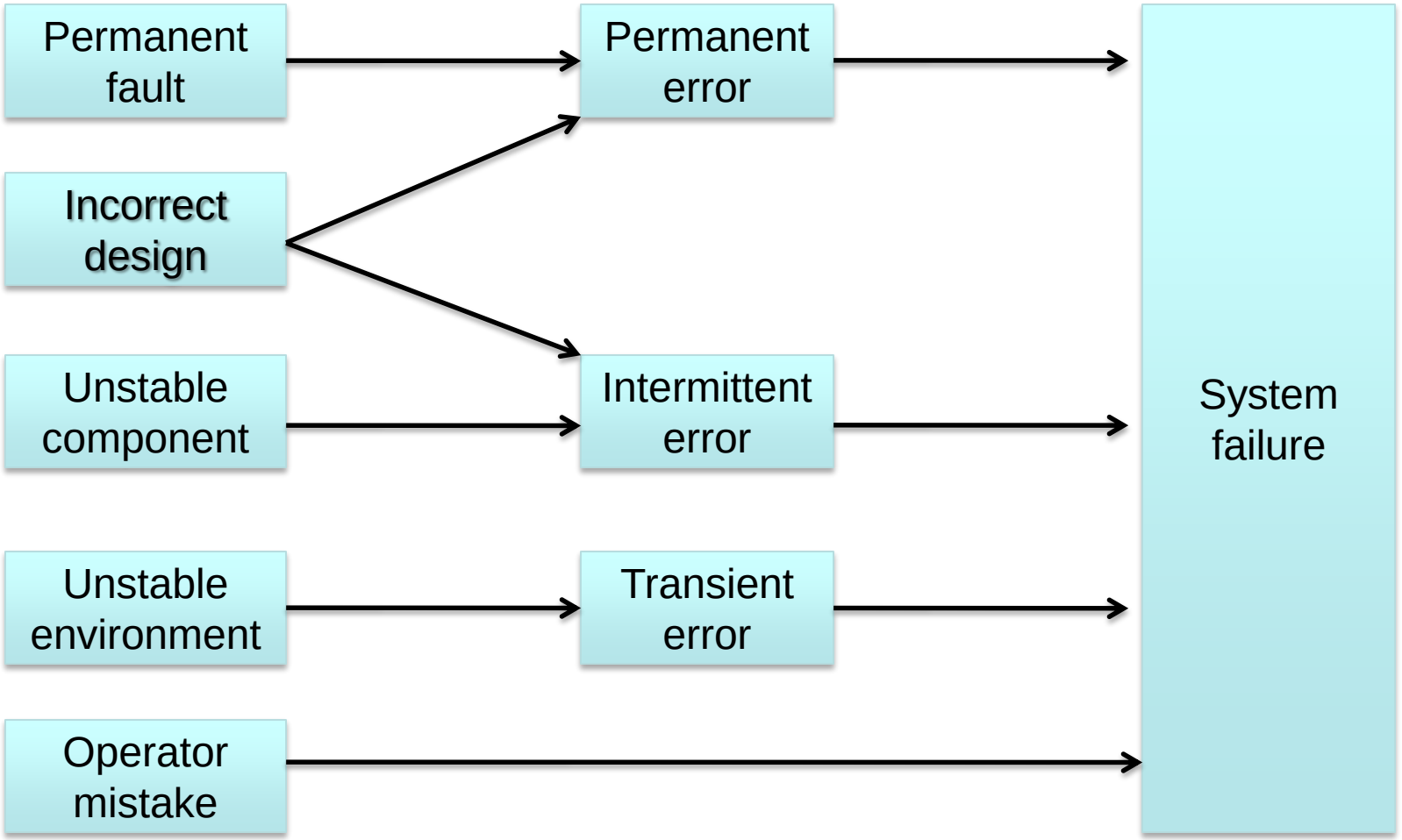
- Question: What is the difference between Reliability and Availability
  - System down for 1 sec every hour → 99.97% availability but unreliable
  - System down for 2 weeks but otherwise runs continuously → 96% available → low availability but highly reliable

# Faults, Errors, Failures

TECHNISCHE UNIVERSITÄT DARMSTADT


- Hard or permanent fault → irreversible change in behavior of system
  - Requires intervention to repair
- Soft fault → reversible change in behavior of system
  - Intermittent fault → shows up occasionally due to unstable components
  - Transient fault → shows up due to changes in environment
  - Timing faults → shows up when particular sequences of states occur
- Design fault
- Operator error

# Sources of System Failure



# Reliability

TECHNISCHE UNIVERSITÄT DARMSTADT

- Reliability refers to the probability that a system does not experience failures in a given time interval

  R(t) = P {0 failures in time interval [0,t]}  (starting from a correct state at t=0)

  P {k failures in [0,t]} = $e^{-m(t)}$ $[m(t)]^k$ / k!  where m(t) = $\int_0^t z(x)dx$

  z(x) is the hazard function that gives the time-dependent failure rate of a component

- The expected (mean) number of failures in time [0,t] can be computed as

  $E[k] = \sum_{k=0}^{\infty} k\, e^{-m(t)} [m(t)]^k / k! = m(t)$

  $Var[k] = E[k^2] - (E[k])^2 = m(t)$  and $R(t) = e^{-m(t)}$

# Reliability for multicomponents Availability

TECHNISCHE UNIVERSITÄT DARMSTADT

- For e system with n independent components (all must function properly for the system to work)

$$R_{sys} = \prod_{i=1}^{n} Ri(t)$$

- Availability refers to the probability that a system is operational according to its specification at a given point in time (prior failures have been repaired)
- In the limit (as time goes to infinity) availability refers to the percentage of time the system is able to perform the desired task
- If failures follow a Poisson distribution with failure rate $\lambda$ and that repair time is distributed exponentially with mean repair time $1/\mu$ the steady state availability of the system is

$$A = \mu / (\lambda + \mu)$$

# Failure Models

TECHNISCHE UNIVERSITÄT DARMSTADT

| Type of failure | Description |
|---|---|
| Crash failure | A server halts, but is working correctly until it halts |
| Omission failure<br>*Receive omission*<br>*Send omission* | A server fails to respond to incoming requests<br>A server fails to receive incoming messages<br>A server fails to send messages |
| Timing failure | A server's response lies outside the specified time interval |
| Response failure<br>*Value failure*<br>*State transition failure* | A server's response is incorrect<br>The value of the response is wrong<br>The server deviates from the correct flow of control |
| Arbitrary failure | A server may produce arbitrary responses at arbitrary times |

# Mean time between failures
# Mean time to repair

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mean Time Between Failures (MTBF) is the expected time between failures in a system with repair
- Mean Time To Failure (MTTF) is the expected time between time = 0 and the first failure in a system without repair

$$\text{MTBF} = \int_0^\infty R(t)dt$$

- Mean Time To Repair (MTTR) is the expected time to repair a failed system
- The availability of the system is then given by

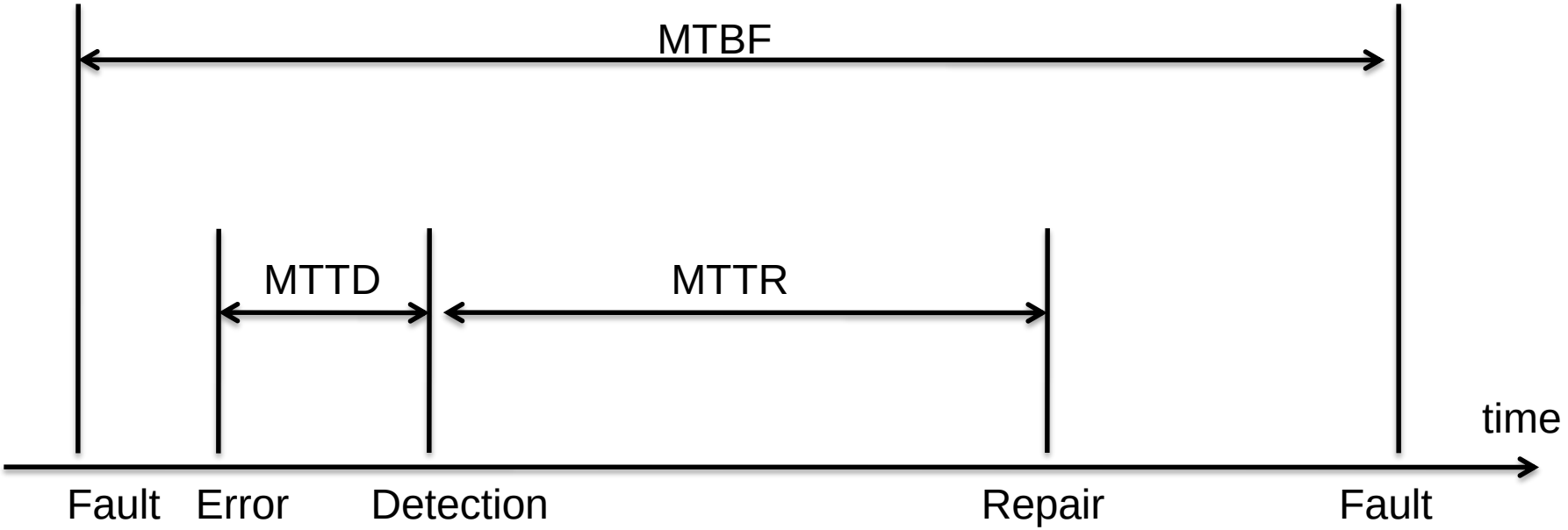
$$A = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

# Latency in fault/failure detection

TECHNISCHE UNIVERSITÄT DARMSTADT

- System failures may be latent (i.e. they are discovered some time after they occur)

- Average error latency (assuming identical systems) is called Mean Time To Detect (MTTD)



# Reliability – MTTF storage systems

TECHNISCHE UNIVERSITÄT DARMSTADT

**MTTF** *(Mean Time To Failure)*

- High quality disk (2)($10^5$) - $10^6$ h or 20-100 yrs
- Standard components ~ 10 yrs

**MTTDL** *(Mean Time To Data Loss)*

- assume independent failures
- without redundancy and *n* disks MTTDL = MTTF/*n*
- *Example:*
  MTTF = 10 yrs., n = 100 disks ➔MTTDL ~ 36 days
- Not acceptable ➔ need redundancy to increase reliability (concept underlying RAID)

# High Availability System Classes
# Goal: Build According to Application Needs

TECHNISCHE UNIVERSITÄT DARMSTADT

| System Type | Unavailable (min/year) | Availability | Availability Class |
|---|---|---|---|
| Unmanaged | 50,000 | 90.% | 1 |
| Managed | 5,000 | 99.% | 2 |
| Well Managed | 500 | 99.9% | 3 |
| Fault Tolerant | 50 | 99.99% | 4 |
| High-Availability | 5 | 99.999% | 5 |
| Very-High-Availability | .5 | 99.9999% | 6 |
| Ultra-Availability | .05 | 99.99999% | 7 |

J.Gray and A.Reuter. Fault Tolerance

# Sources of Failures

TECHNISCHE UNIVERSITÄT DARMSTADT

| Source | MTTF | MTTR |
|---|---|---|
| Power Failure | 2000 hr | 1 hr |
| Phone Lines: Hard | 4000 hr | 10 hr |
| Phone Lines: Soft | >.1 hr | >.1 hr |
| Hardware Modules | 100 000 hr | 10 hr |
| | | |

Software:

1 Bug/1000 Lines Of Code (after vendor-user testing)

➔ Thousands of bugs in System!

Most software failures are transient: dump & restart system.

J.Gray and A.Reuter. Fault Tolerance