Summary of Lecture 10.01.2018

http://englishskills.se

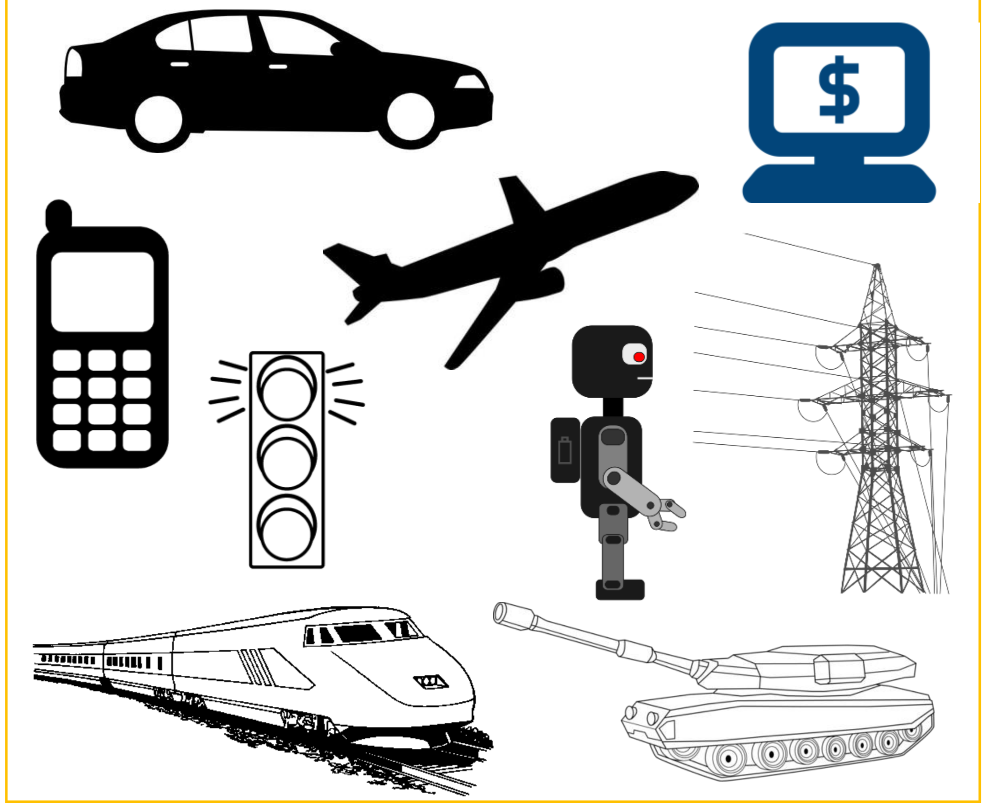
Dependability **expectations** → Dependability **properties**

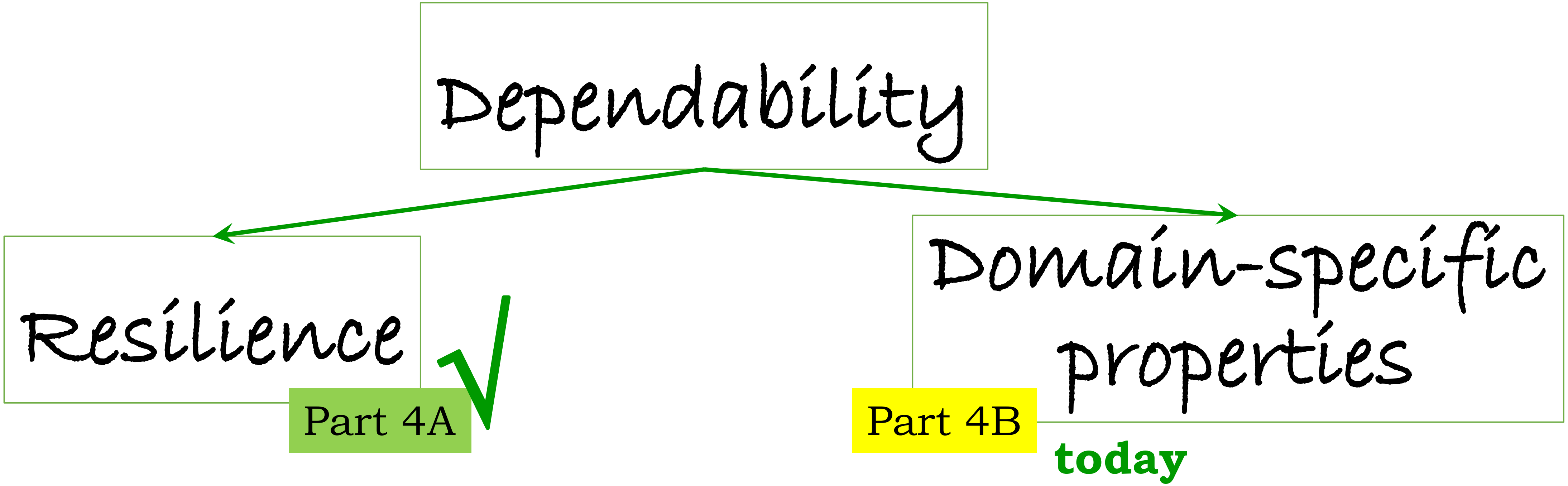
Resilience ← General property

- **Safety**
- **Security**
- **Integrity**
- **Confidentiality**
- **Real-time capability**
- …

← Specific properties

Application Domain




Dependability
Resilience
Part 4A
√
Domain-specific properties
Part 4B
today

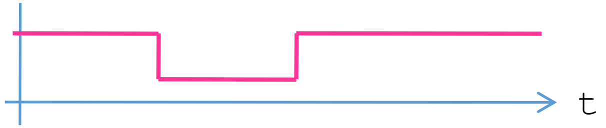
Resilience: Reaction

Incident

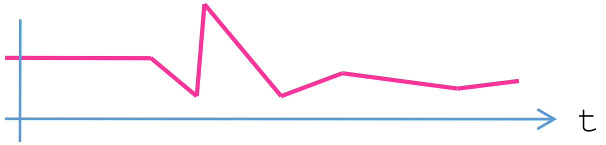
https://soundcloud.com

Crash ① **NOT acceptable**

Degraded operation ② **Acceptable in some cases**



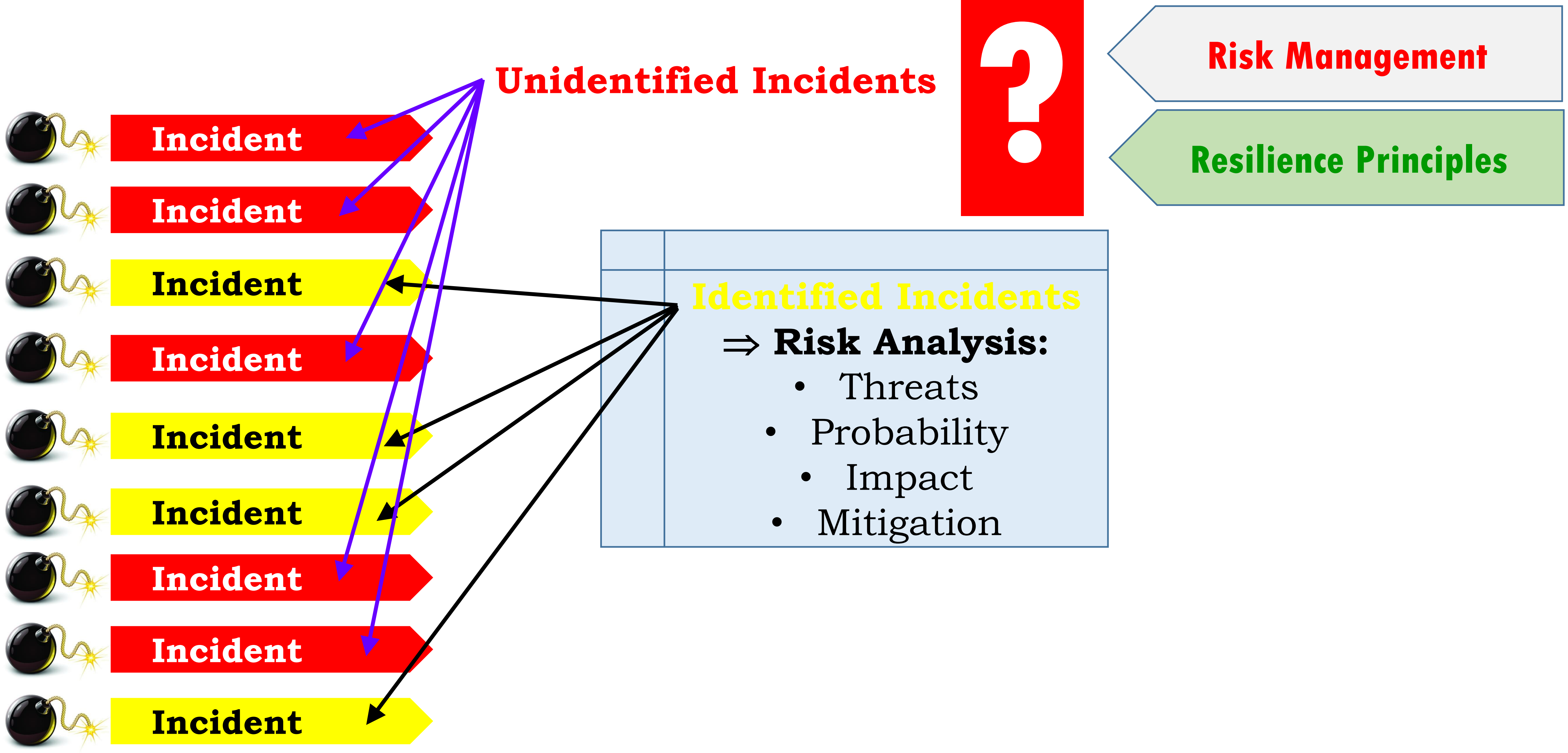
Recovery ③ **Acceptable in some cases**



Malfunction ④ **NOT acceptable**

http://kanto.stripes.com

Immunity ⑤ **Desirable**


Unidentified Incidents
?
Risk Management
Resilience Principles
Incident
Incident
Incident
Incident
Incident
Incident
Incident
Incident
Incident
Identified Incidents
⇒ Risk Analysis:
• Threats
• Probability
• Impact
• Mitigation

General (Overarching) Architecture Principles for Resilience

- R1: Policies
- R2: Vertical Architectures
- R3: Fault Containment Regions
- R4: Single Points of Failure
- R5: Multiple Lines of Defense
- R6: Fail-Safe States
- R7: Graceful Degradation
- R8: Dependable Foundation (Infrastructure)
- R9: Monitoring

R1

Resilience Architecture Principle R1:

Policies

1. Develop and enforce good, comprehensive, consistent policies for the infrastructure of the company or organization
2. Support the policies by carefully selected standards, methodologies and frameworks
3. Keep all policies and supporting material up-to-date and adapted to the changing environment
4. Consequently apply the policies to the evolution of the infrastructure (enforcement)

Justification: Technical infrastructure has become so complex and important that it needs to be governed by a consistent set of policies – otherwise the resulting divergence is a massive risk for the company

Future-Proof Software-Systems: Summary 10.01.2018


http://www.lighthousecounselingomaha.com

Good policies are a great help for the people implementing company objectives, especially ***infrastructure projects***

SECURITY Policy

→ **Security Standards**

→ **Security Methodology**

→ **Security Frameworks**

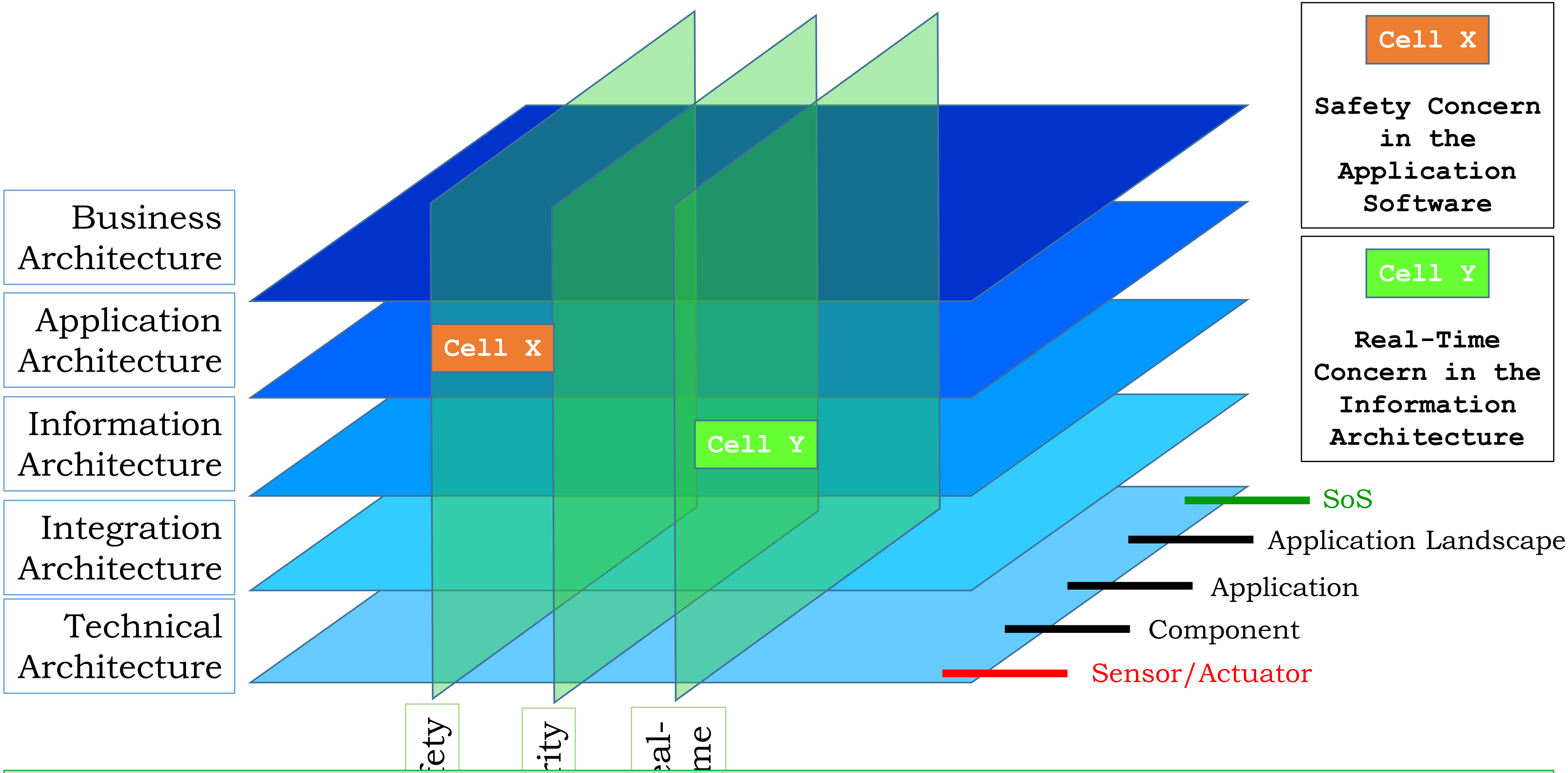
R2

Resilience Architecture Principle R2:

Vertical Architectures

1. The dependability of the system is directly dependent on the quality of the vertical architectures (and the quality of their implementation)
2. Match the quality of the vertical architectures to the risk/damage potential of your application (Caution: be on the safe side)
3. Continuously maintain/evolve your vertical architectures in sync with changing environments and requirements

Justification: Vertical architectures are at the core of dependability. A great bandwith in quality of vertical architectures exists. If the quality is too low, your systems/applications may be at risk.



⇒ **Formulation of Powerful Set of Architecture Principles**, **NEVER** implement security functionality in the applications software

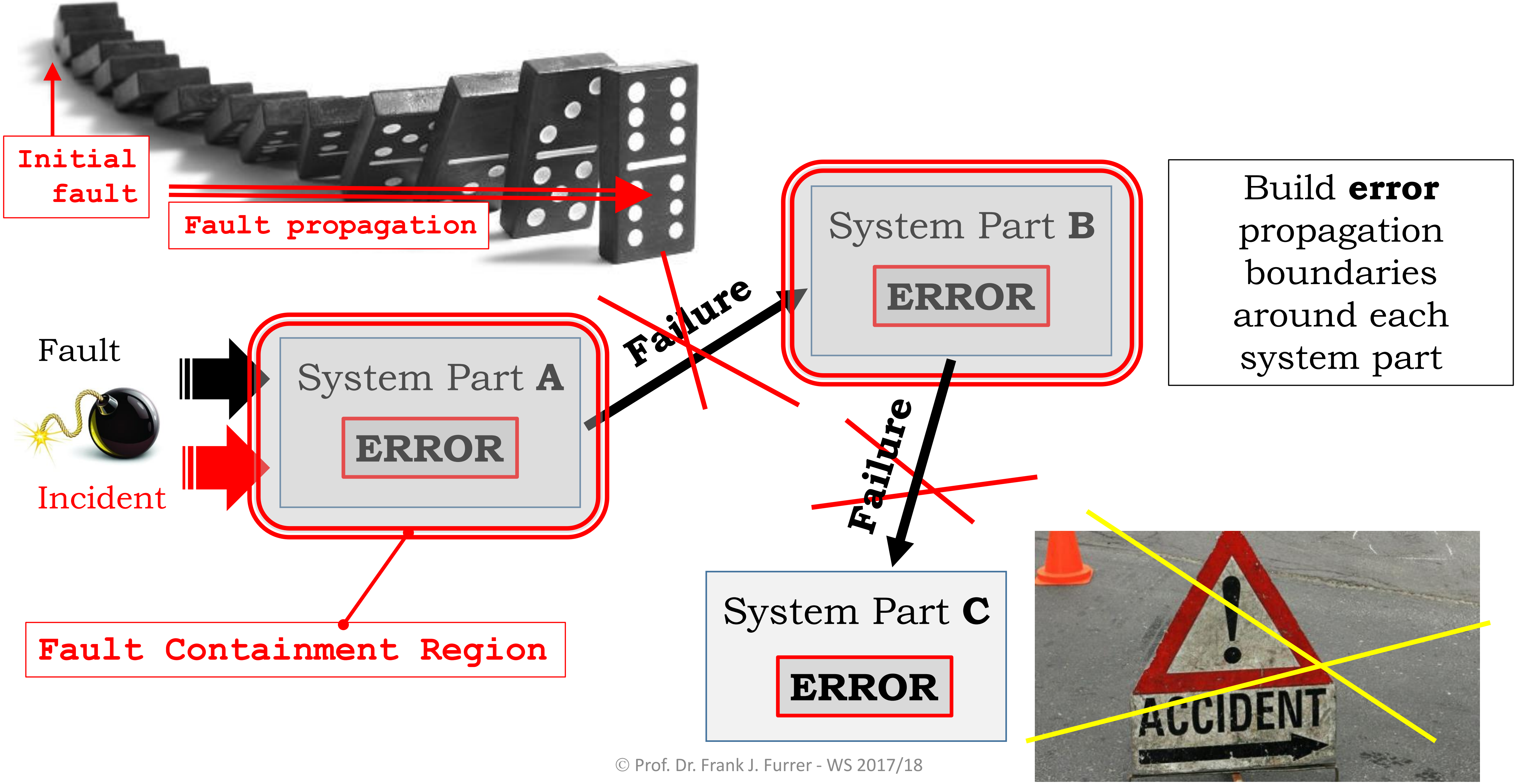
R3

Resilience Architecture Principle R3:

Fault Containment Regions

1. Partition the system into fault containment regions
2. Build error propagation boundaries around each system part ($\Rightarrow$ Interfaces)
3. Provide sufficient redundant information about the intended behavior of the system parts (components)

Justification: A fault or incident causing an error or disruption in one part (component) of the system should not propagate to other parts of the system and thus cause a sequence of errors and failures


Initial fault
Fault propagation
Fault
Incident
System Part A
ERROR
Failure
System Part B
ERROR
Failure
System Part C
ERROR
ACCIDENT
Fault Containment Region
Build **error** propagation boundaries around each system part

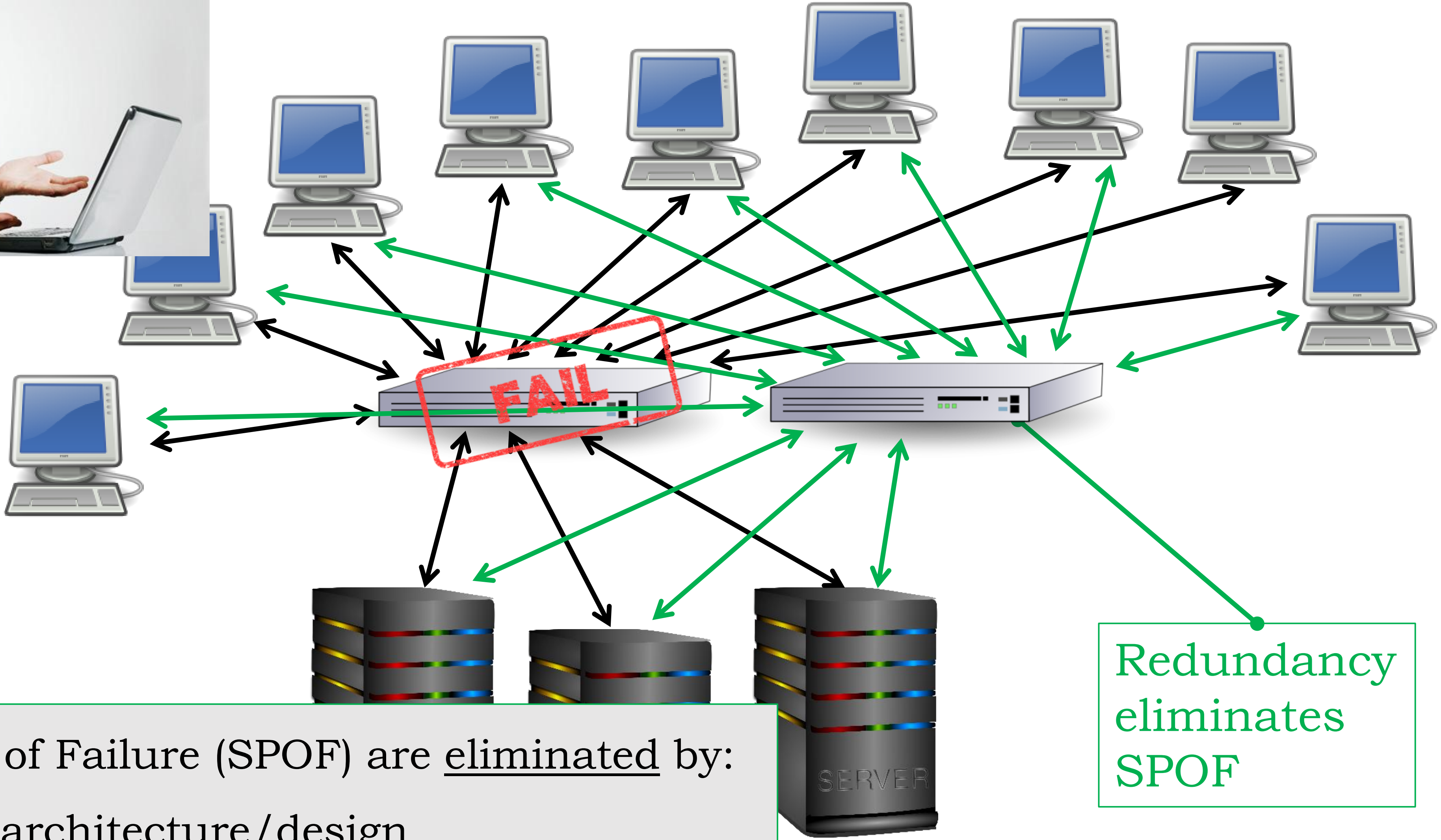
R4

Resilience Architecture Principle R4:

Single Points of Failure

1. Identify possible single points of failure early in the architecture/design process (Note: single points of failure can occur on all levels of the architecture stack)
2. Eliminate single points of failure, e.g. by introducing redundancy

Justification: Any single point of failure is a great risk for a dependable system. They must therefore be avoided



Redundancy eliminates SPOF

Single Points of Failure (SPOF) are eliminated by:

a) Intelligent architecture/design

b) Introduction of *redundancy*

R5

Resilience Architecture Principle R5:

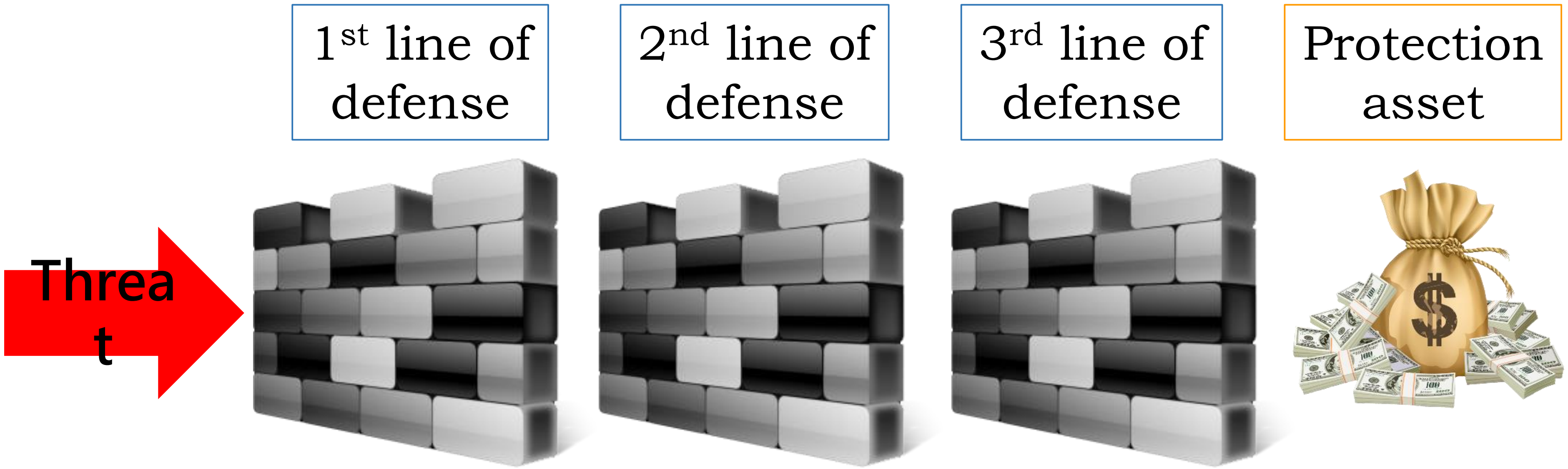
Multiple Lines of Defense

1. For each threat and incident implement multiple, independent lines of defense
2. For each line of defense use different methods, techniques and technologies

Justification: If a line of defense is overcome as a consequence of an incident, the second (third, …) line of defense may mitigate the impact of the incident

DEFINITIONS

Multiple lines of defense represents the use of *multiple* computer techniques to help mitigate the risk of one component of the defense being compromised or circumvented



R6

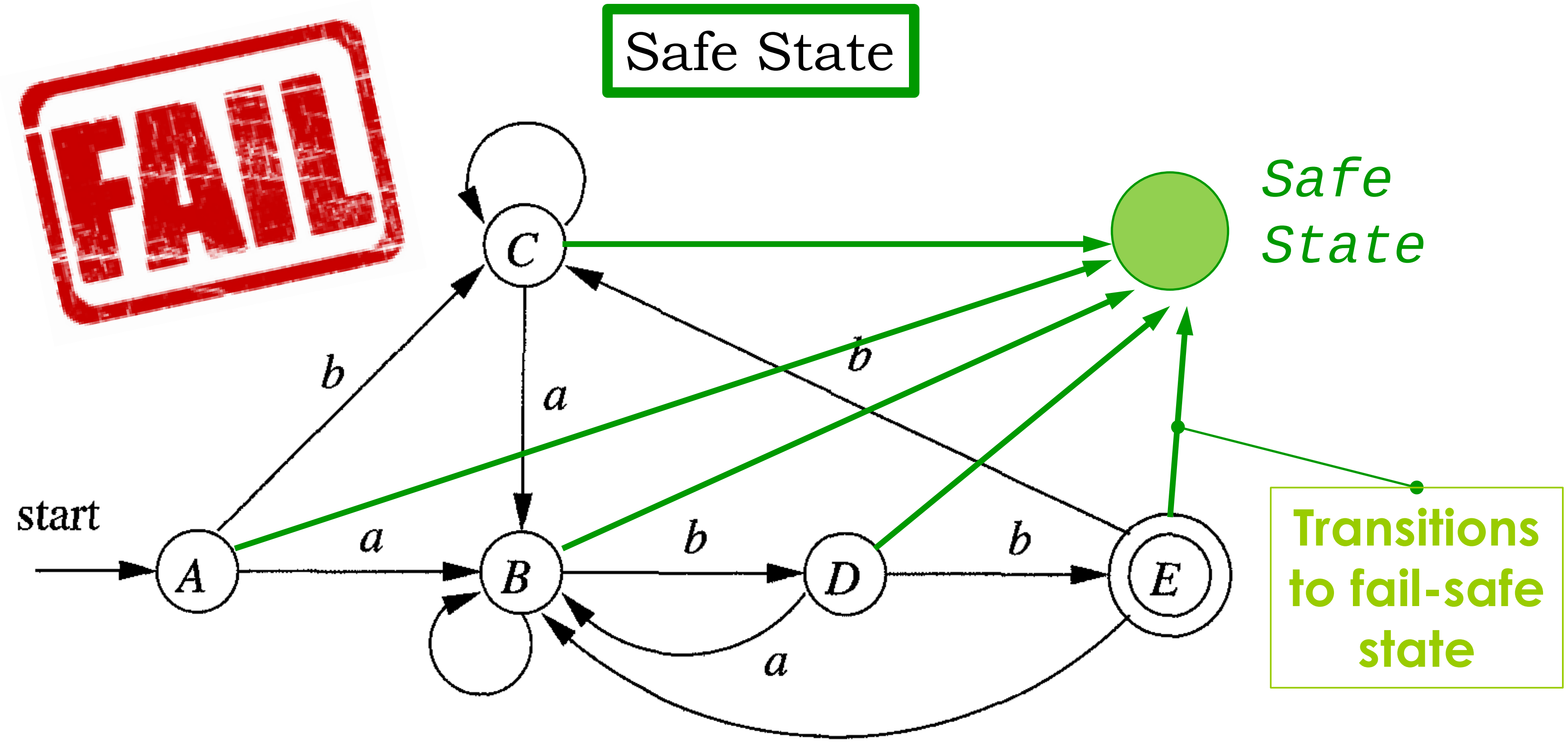
Resilience Architecture Principle R6:

Fail-Safe States

1. Execute a careful hazard analysis of your *full system* to identify all (= goal) critical or harmful states
2. Document all paths to the critical or harmful states in a formal way, such as state chart diagrams
3. Model your application (or the software part of it) as a finite state machine
4. Define fail-safe state(s)
5. Implement reliable paths from all nodes to the fail safe state(s)

Justification: If a failed system can transition into a fail-save state, then damage, loss of life or property or other negative consequences may be avoidable (or minimized)

Safe State



Safe State

Which is a safe state?

How can we find a safe state?

Difficult Engineering Task

https://www.npmjs.com

https://img.clipartxtras.com

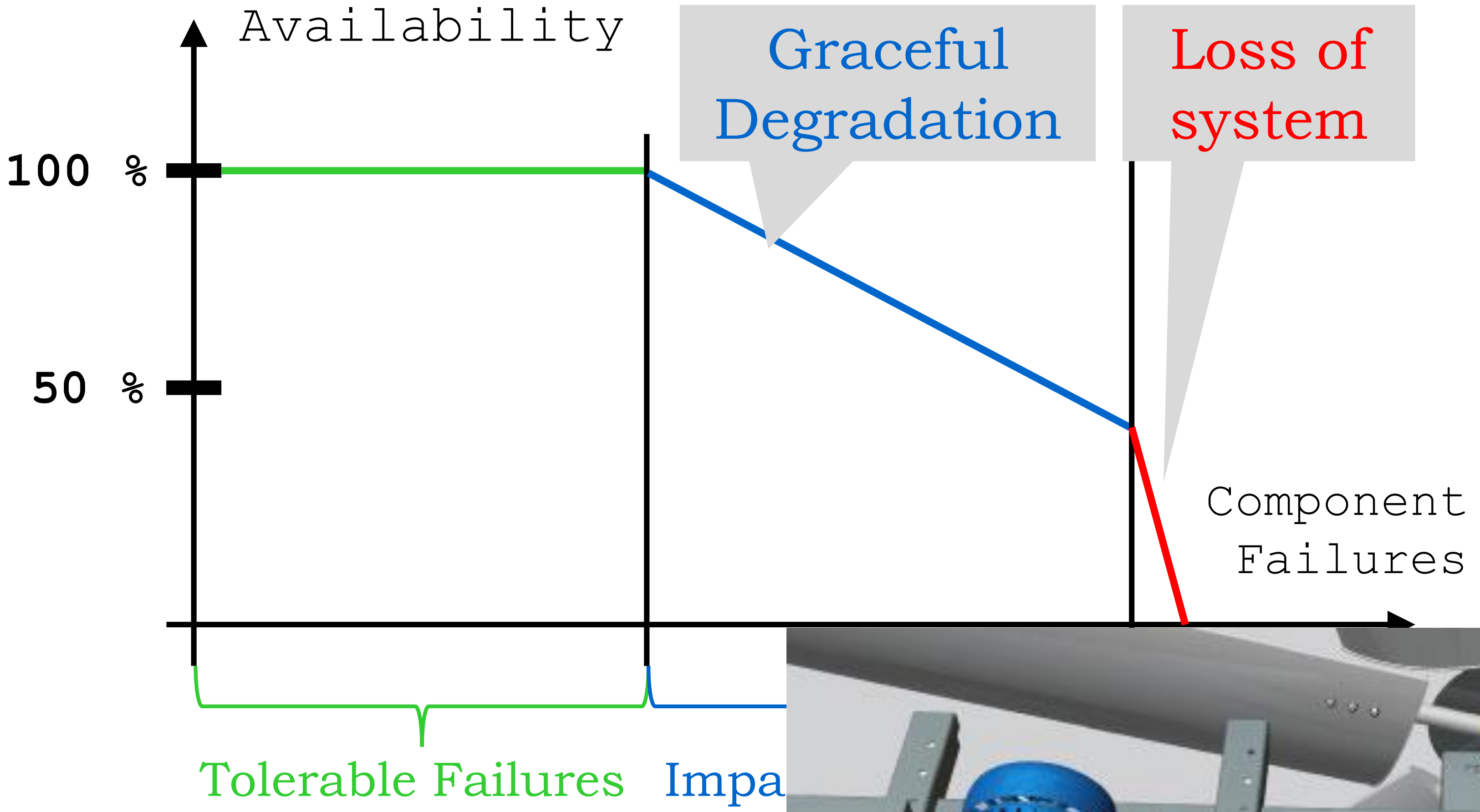
R7

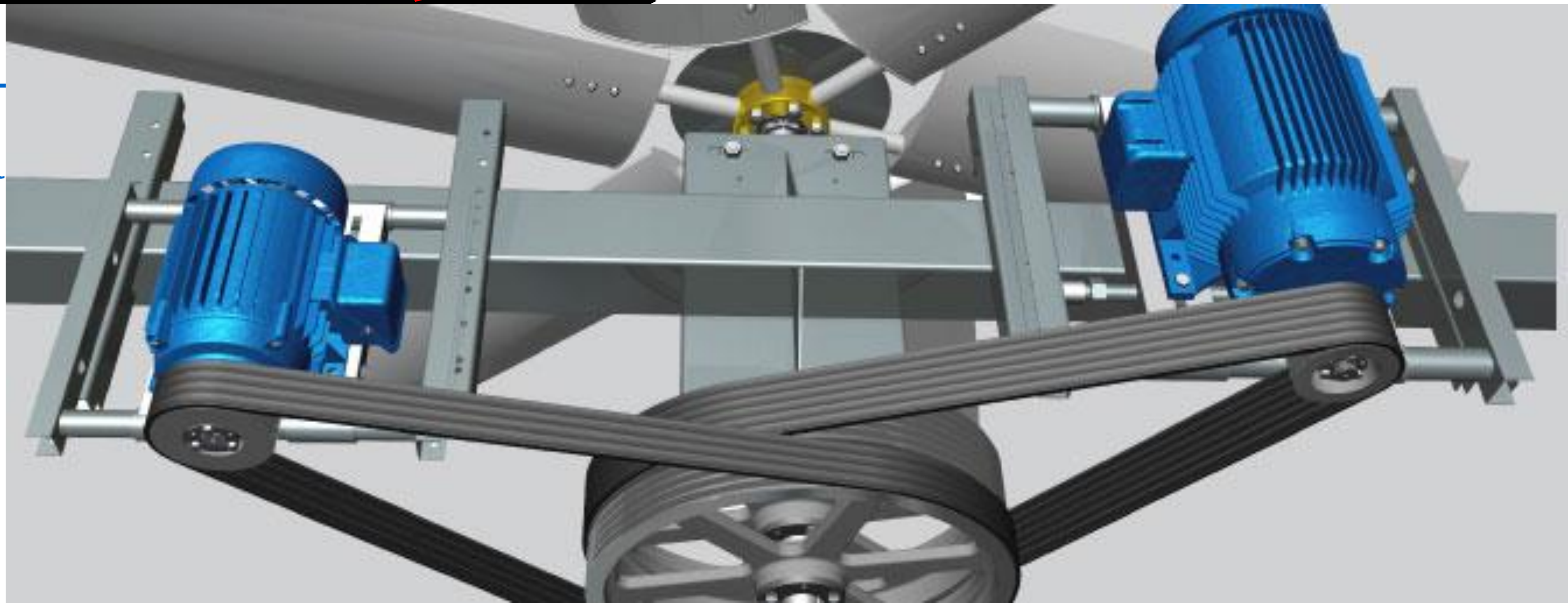
Resilience Architecture Principle R7:

Graceful Degradation

1. Investigate the possibility for graceful degradation in your planned system (= Business task)
2. Architect and implement proven graceful degradation technologies (for specific resilience properties, such as availability, performance, safety, security, …)
3. Compensate component failures by carefully planned *redundancy*

Justification: The value of many systems is significantly improved if after a failure of a component the system operates in a (planned) degraded mode instead of stopping service


Availability
Graceful Degradation
Loss of system
100 %
50 %
Component Failures
Tolerable Failures
Impa



R8

Resilience Architecture Principle R8:

Dependable Foundation (Infrastructure)

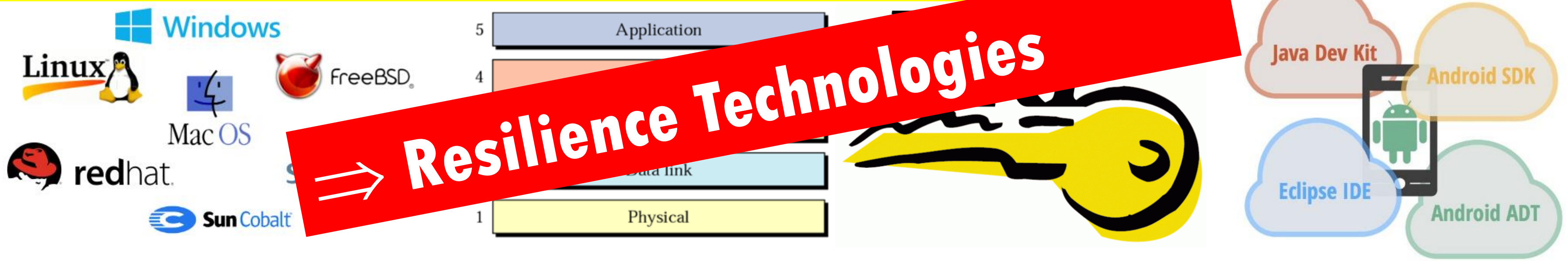
1. Use a resilience infrastructure as part of a dependable foundation for resilient software-systems
2. Only use *proven* resilience technologies and services supporting the resilience properties (availability, security, performance, …)
3. Whenever possible use industry-standard based resilience techniques (Avoid vendor lock-in)

Justification: An implementation of proven resilience techniques in the form of industry-standard products forms a valuable, trustable resilience foundation

APPLICATIONS

⇒ Resilience Principles

Execution Infrastructure: *System Software*



⇒ Resilience Technologies

Execution Infrastructure: *System Hardware*

⇒ Resilience Technologies

R9

Resilience Architecture Principle R9:

Monitoring

1. Define the objectives of monitoring, both for technical monitoring and the business monitoring
2. Carefully specify the metrics, analytics, results, and alerts to be extracted from monitoring
3. Define the processes for data analysis, including incident/emergency response
4. Specify the actions following alerts – whenever fully automated responses
5. Recommendation: Use commercial monitoring tools whenever possible

Justification:

- Technical monitoring is a strong weapon for assuring the non-functional properties of the system and for defending the system against incidents
- Business monitoring strongly contributes to customer satisfaction

An **IT system monitor** is a hardware and software component used to measure resource consumption and performance in a computer system.

Any ***anomaly*** in operating parameters (load, response time, …) is automatically detected and an alarm is triggered

⇒ **Automatic** or **human** intervention

Technical Information

Objectives:

- Early problem warning
- System defense
- System optimization
- System intelligence information
- Failure tracing

⇒ *Assure the **non-functional** requirements*

https://www.istockphoto.com

Business Information

Objectives:

- Customer satisfaction
- Financial optimization
- Contract (SLA) supervision
- Audit/Compliance
- Business intelligence information
- ⇒ Assure the ***functional*** requirements

https://www.jobdiagnosis.com

Specific
Dependability
Properties

Part 4B

https://i2.wp.com

1. Security
2. Confidentiality
3. Integrity
4. Availability
5. Safety
6. Real-Time Capability