Future-Proof Software-Systems

(Zukunftsfähige Software-Systeme)

Frank J. Furrer

Prof. Dr. sc. techn. ETH-Zürich

TU Dresden WS 2015/2016

Part 4B: Architecting for Resilience

V0.5 / 17.01.2016

Content

Part 4A:

1. Managed Evolution (Repetition)
2. Resilience
3. Architecting for Resilience
4. Resilience Architecture Principles
   - *Universal* Resilience Architecture Principles

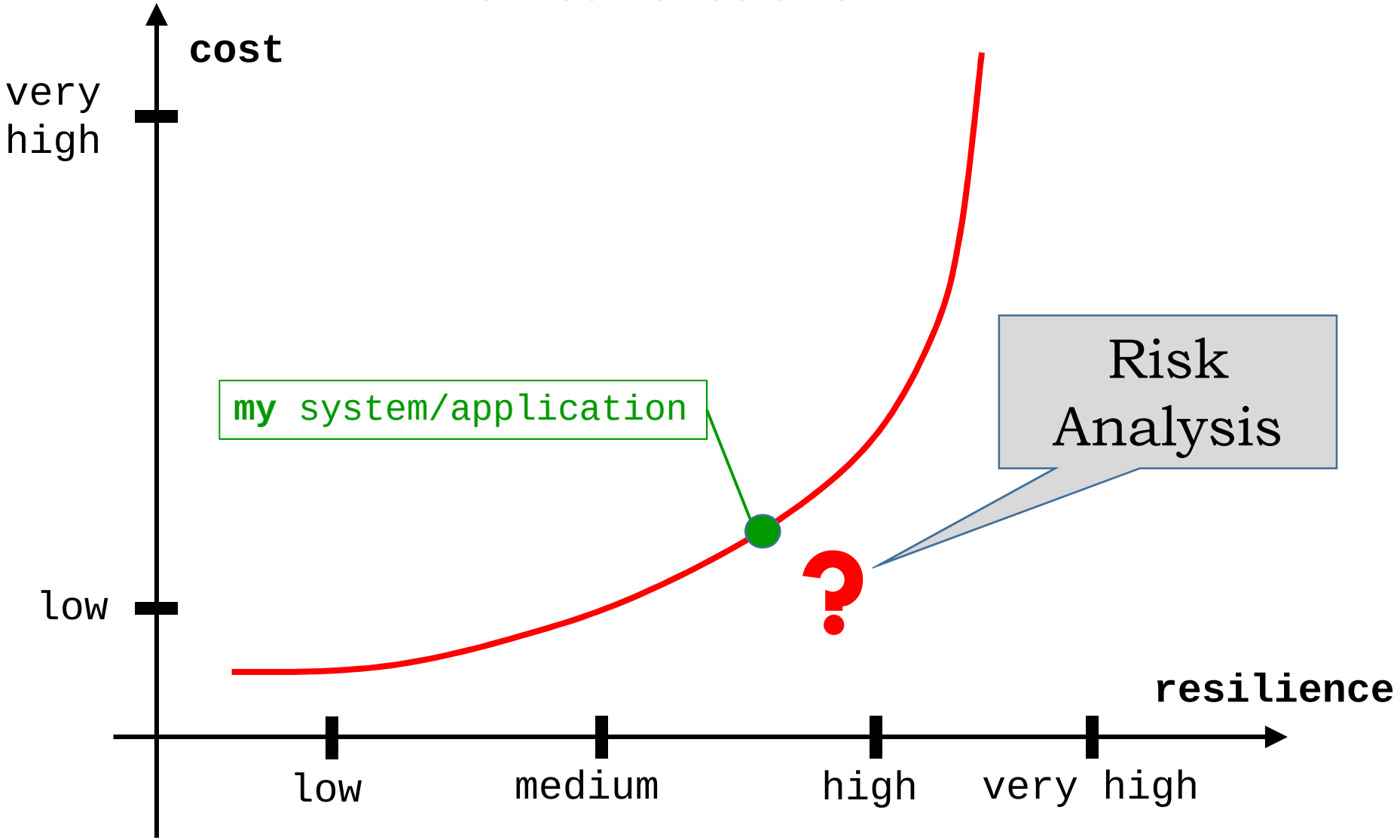
Part 4B:

- *Specific* Resilience Architecture Principles
- Autonomic Computing

Future-Proof Software-Systems:

Risk Analysis

The «resilience dilemma»



http://www.aeroartinc.com

Resilience properties of the SW-system

- **Safety**
- **Security**
- **Integrity**
- **Availability**
- **Confidentiality**
- **Certifiability**
- **Reliability**
- **Diagnosability**
- **Standards adherence**
- **Accountability**
- **Traceability**
- **Non-Repudiation**
- **Fault-Tolerance**
- **Business Continuity**
- **Recoverability**
- **Graceful Degradation**
- **Auditability**
- **Real-TimeCapability**
- **Fail-Save Behaviour**
- **Survivability**
- **Performance**

etc.

Resilience Profile

Quality Property	*Type*
Business Value	Primary Characteristic
Agility	Primary Characteristic
Availability	√
Security	√
Safety	√
Privacy/Confidentiality	√
Performance	
Usability	
Robustness	√
Operating Cost	
Reusability	
Compliance to laws and regulations	√
Adherence to industry-standards	
Memory Size	
Power consumption	
Testability	
etc.	
	√ = Resilience

identical for all systems

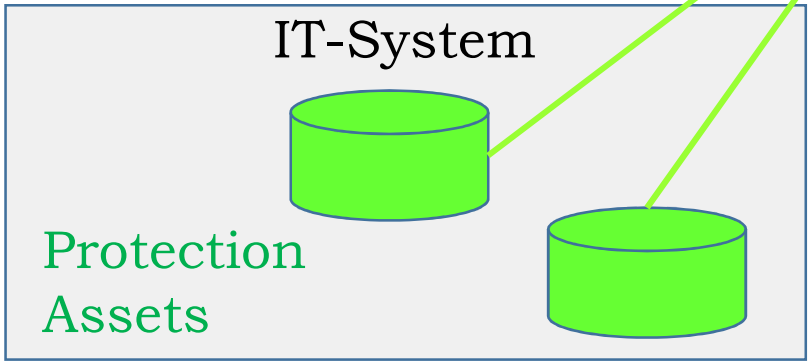
Resilience set: dependent on application area

Additional quality properties: dependent on application area

How do I find out which specific resilience *principles* and *patterns* are important for my application?

http://managementresearchdevelopment.com

⇒ by executing a *risk analysis*

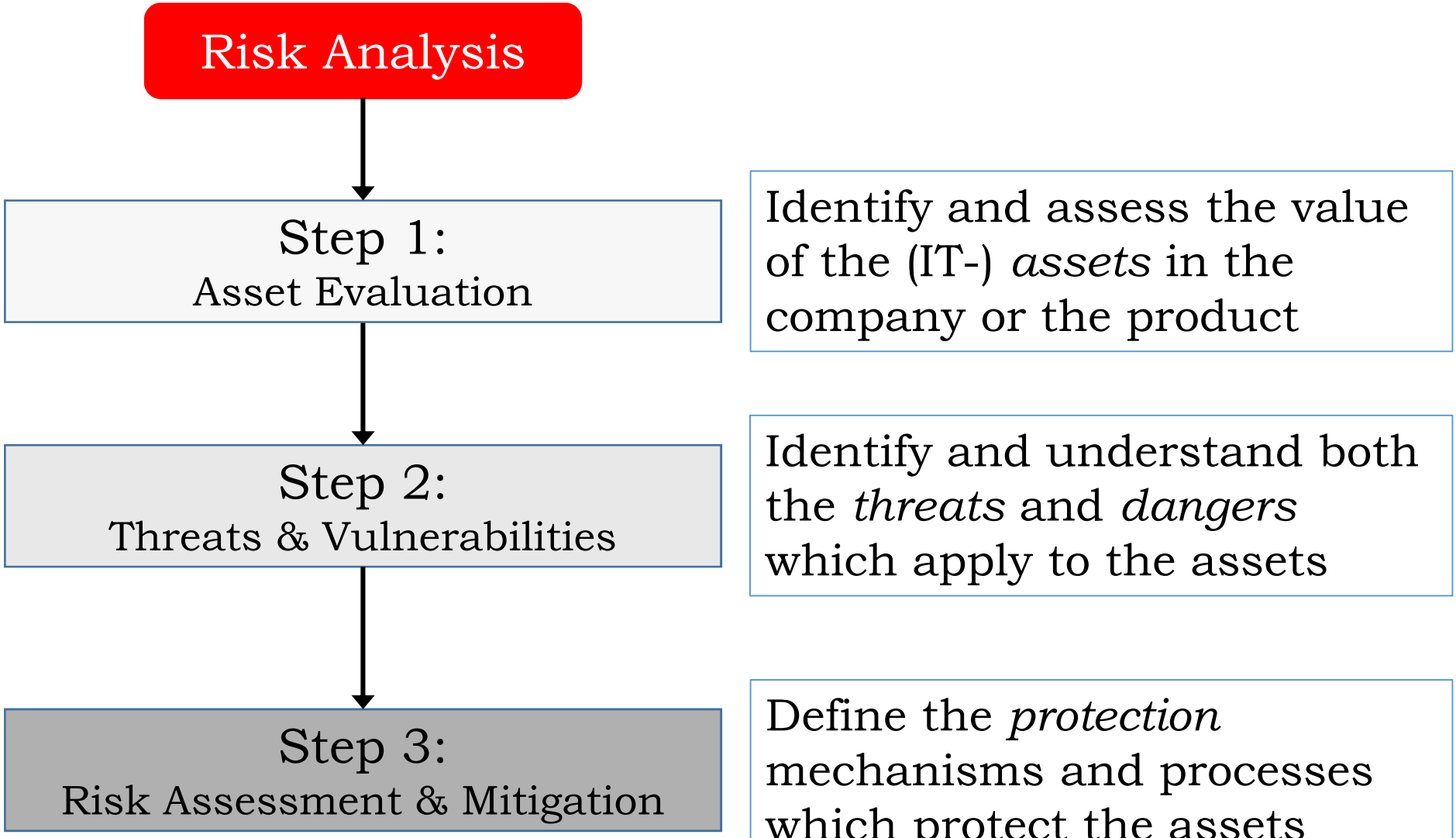


Risk Analysis

Step 1:
Asset Evaluation

Step 2:
Threats & Vulnerabilities

Step 3:
Risk Assessment & Mitigation


Risk Analysis
Step 1:
Asset Evaluation
Identify and assess the value of the (IT-) *assets* in the company or the product
Step 2:
Threats & Vulnerabilities
Identify and understand both the *threats* and *dangers* which apply to the assets
Step 3:
Risk Assessment & Mitigation
Define the *protection* mechanisms and processes which protect the assets

http://seattlewebdesign

Example: System = Financial Institution

Protection Assets

- …
- Customer Data
- …

Threats & Dangers

- …
- Unauthorized Access
- …

Protection Means

- Authentication
- Authorization
- Access Control
- …

https://roadloans.com/

Example: System = Passenger Car

Protection Assets

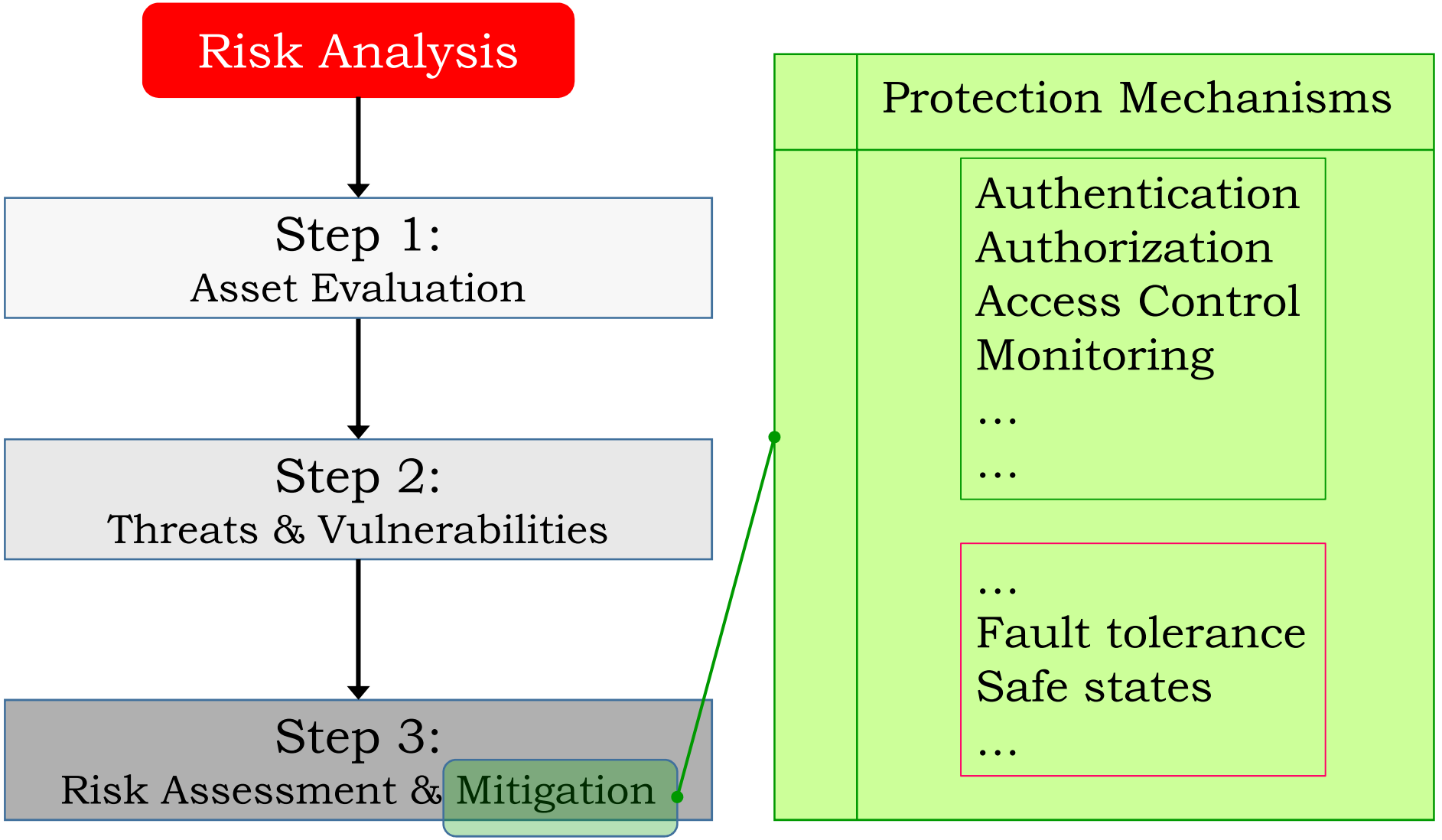
- …
- Life & Health of Passengers
- …

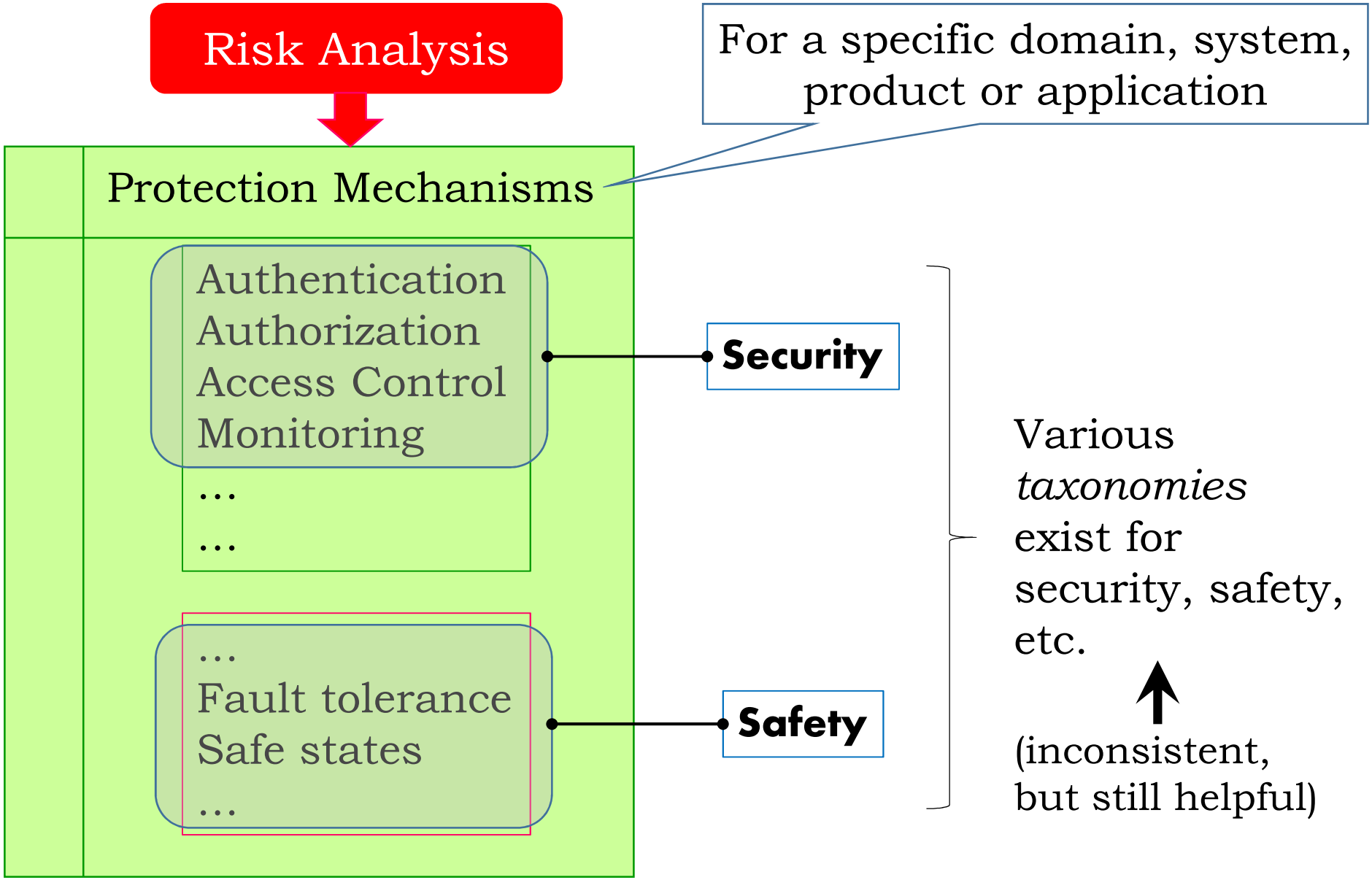
Threats & Dangers

- …
- Electronic System Failure
- …

Protection Means

- …
- Fault tolerance
- Safe states
- …


Risk Analysis
Step 1:
Asset Evaluation
Step 2:
Threats & Vulnerabilities
Step 3:
Risk Assessment & Mitigation
Protection Mechanisms
Authentication
Authorization
Access Control
Monitoring
…
…
…
Fault tolerance
Safe states
…


Risk Analysis
For a specific domain, system, product or application
Protection Mechanisms
Authentication
Authorization
Access Control
Monitoring
...
...
Security
...
Fault tolerance
Safe states
...
Safety
Various *taxonomies* exist for security, safety, etc.
(inconsistent, but still helpful)

Risk Analysis

Protection Mechanisms

Authentication
Authorization
Access Control
Monitoring
…
…

Security

…
Fault tolerance
Safe states
…

Safety

Applicable Resilience *Principles* and *Patterns*

Summary

1. Specify your system
2. Resilience Analysis
3. Applicable Principles and Patterns

http://seattlewebdesign

https://roadloans.com/

Decide on general (overarching) principles

Risk Analysis

Protection Mechanisms

Resilience Principles and Patterns

… others

https://www.npmjs.com

Summary

Applicable Resilience Principles and Patterns

Resilience Engineer:
- Security Engineer
- Safety Engineer
- Certification Engineer
- …

Important and interesting role

Summary

3. Applicable Principles and Patterns

Resilience Principles and Patterns

→ **Specific** (Property-dependent) Resilience Architecture Principles & Patterns







Future-Proof Software-Systems:

Specific (Property-dependent) Resilience Architecture Principles & Patterns

General Resilience Architecture Principles

Resilience

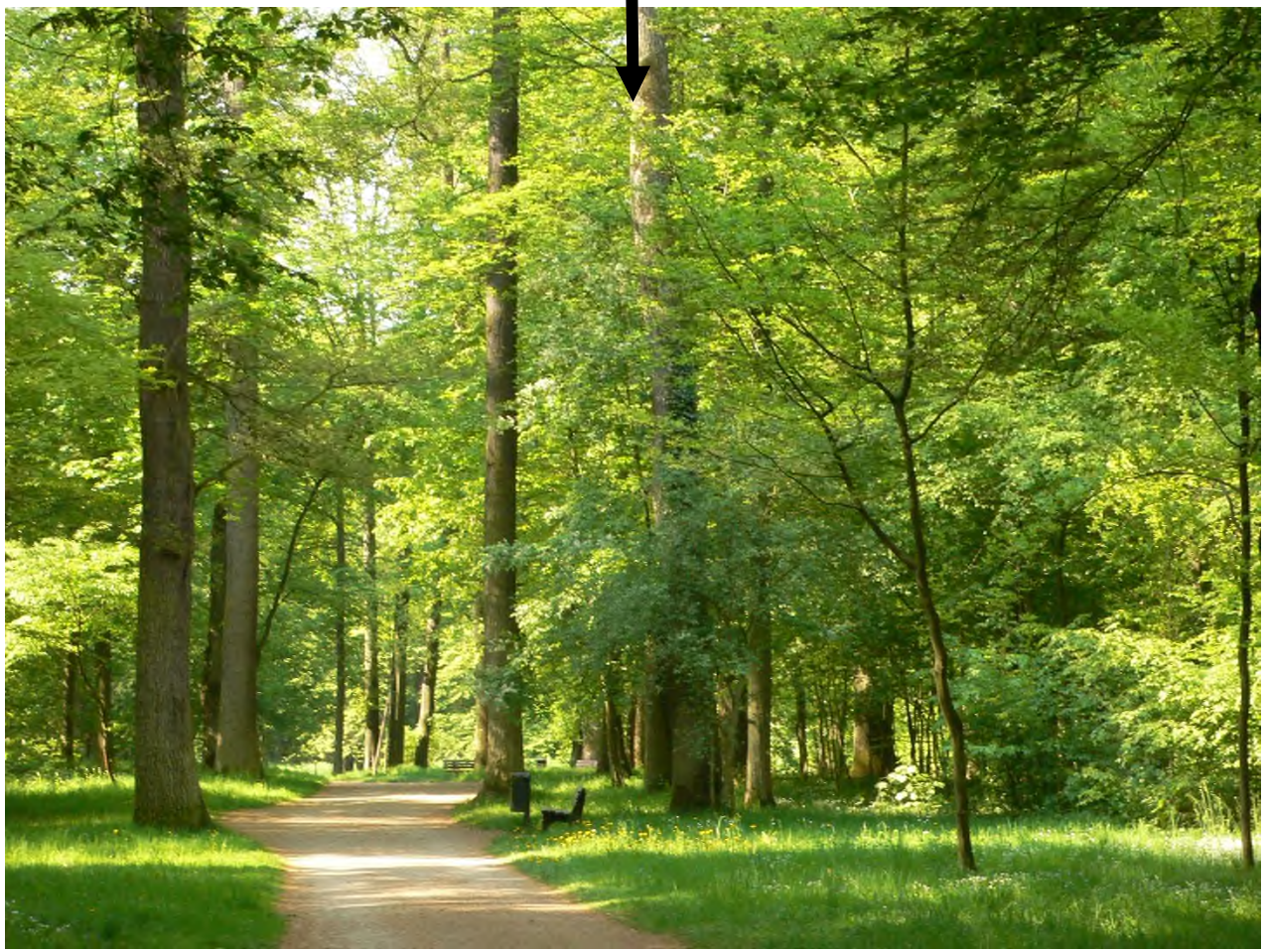
http://en.wikipedia.org/wiki/Czechoslovak_border_fortifications

- R1: Fault Containment Regions
- R2: Single Points of Failure
- R3: Multiple Lines of Defense
- R4: Fail-Safe States
- R5: Graceful Degradation
- R6: Dependable Foundation

√

Specific (Property-dependent) Resilience Architecture Principles

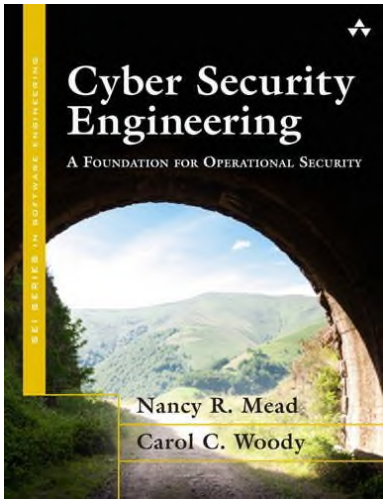
Specific IT Resilience Architecture Principles & Patterns

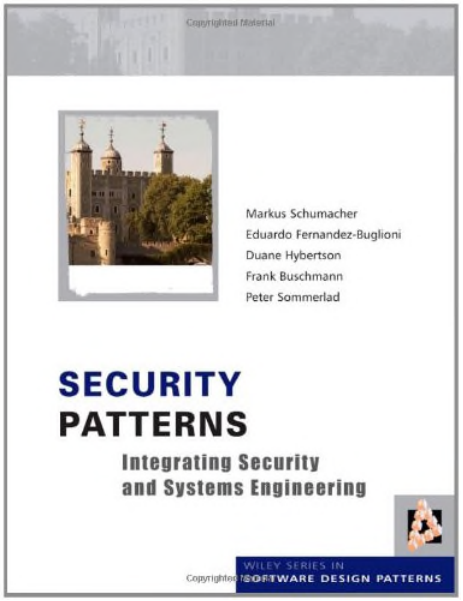


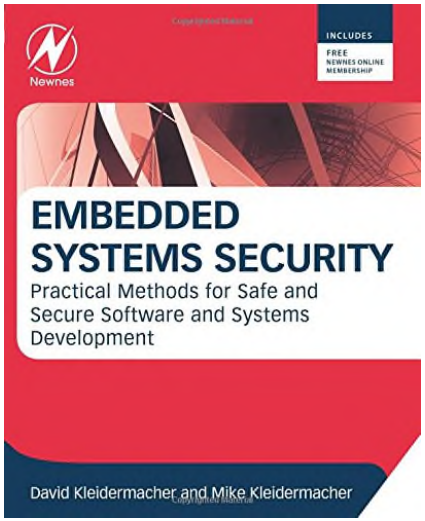
http://www.outdoor-magazin.com

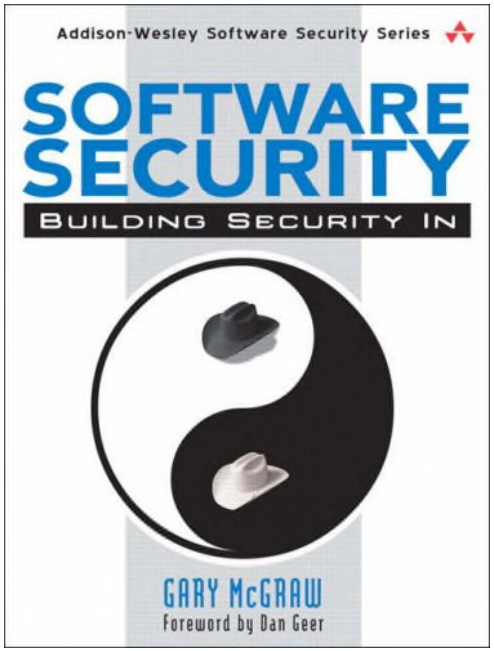
… a very large and rich field of engineering

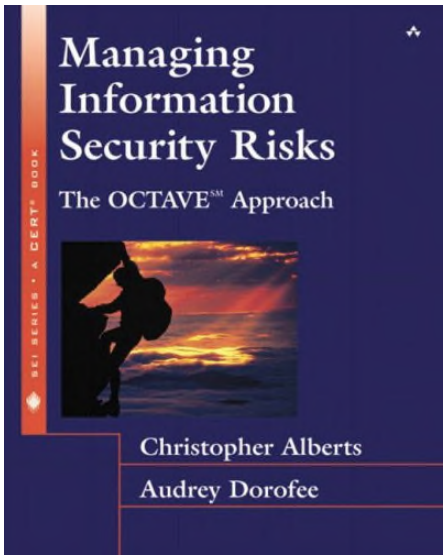
Security

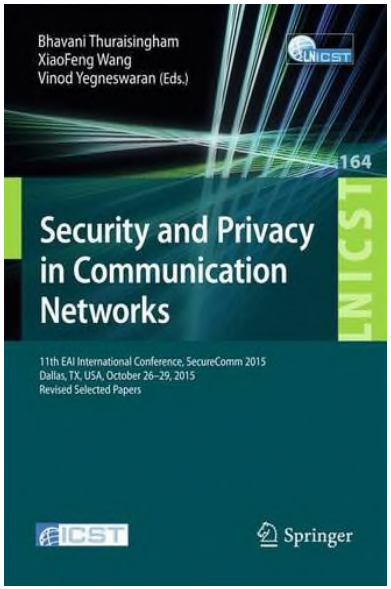

Cyber Security Engineering
A Foundation for Operational Security
Nancy R. Mead
Carol C. Woody

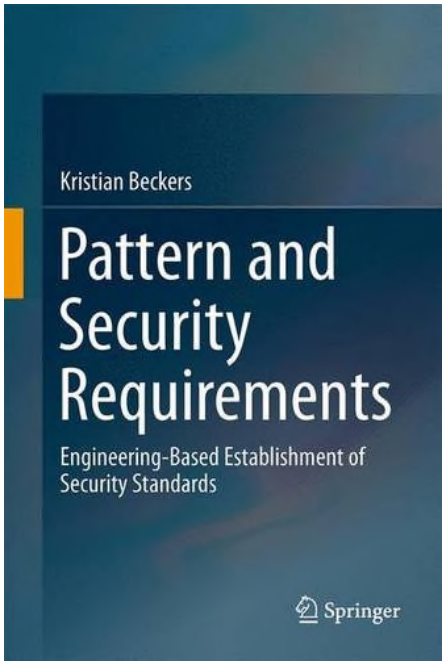

Markus Schumacher
Eduardo Fernandez-Buglioni
Duane Hybertson
Frank Buschmann
Peter Sommerlad
SECURITY PATTERNS
Integrating Security and Systems Engineering
WILEY SERIES IN SOFTWARE DESIGN PATTERNS

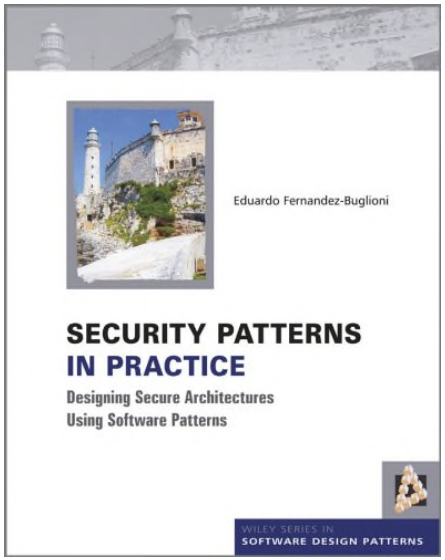

Newnes
INCLUDES
FREE NEWNES ONLINE MEMBERSHIP
EMBEDDED SYSTEMS SECURITY
Practical Methods for Safe and Secure Software and Systems Development
David Kleidermacher and Mike Kleidermacher


Addison-Wesley Software Security Series
SOFTWARE SECURITY
BUILDING SECURITY IN
GARY McGRAW
Foreword by Dan Geer


SEI SERIES • A CERT® BOOK
Managing Information Security Risks
The OCTAVE℠ Approach
Christopher Alberts
Audrey Dorofee

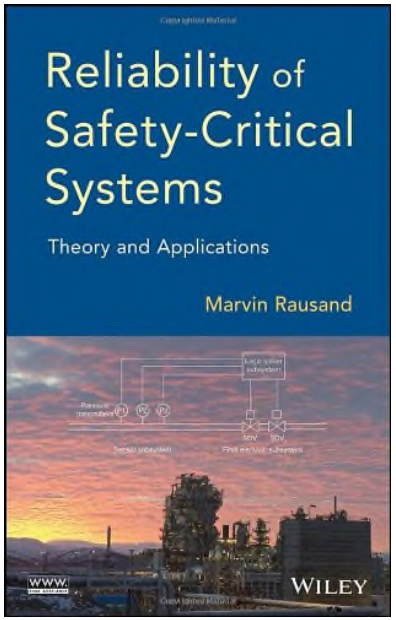

Bhavani Thuraisingham
XiaoFeng Wang
Vinod Yegneswaran (Eds.)
LNICST
164
Security and Privacy in Communication Networks
11th EAI International Conference, SecureComm 2015
Dallas, TX, USA, October 26–29, 2015
Revised Selected Papers
ICST
Springer

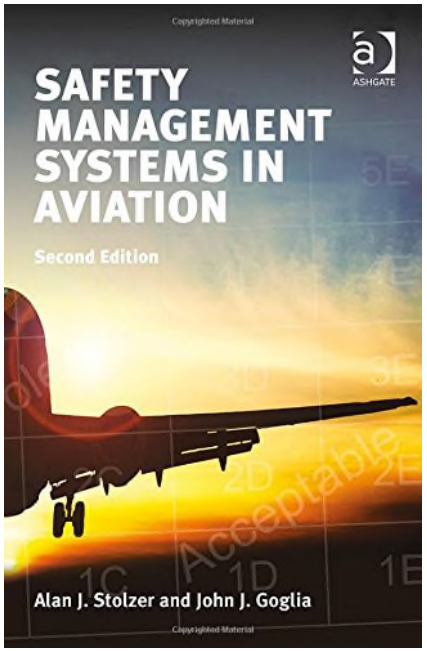

Kristian Beckers
Pattern and Security Requirements
Engineering-Based Establishment of Security Standards
Springer

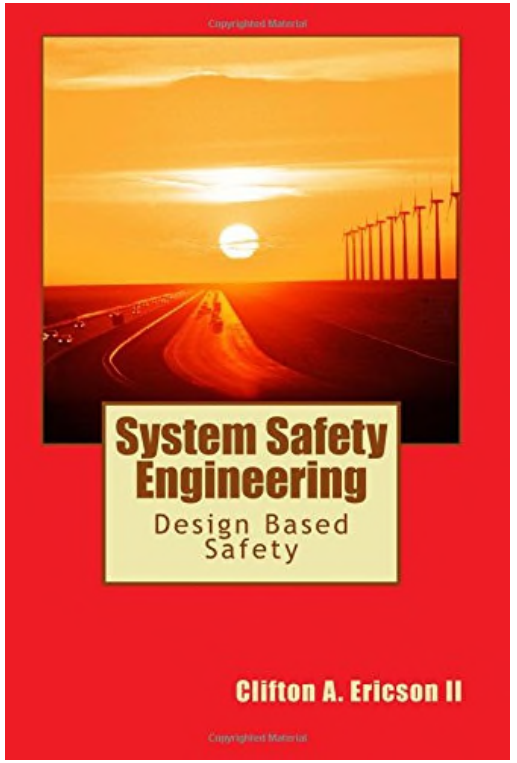

Eduardo Fernandez-Buglioni
SECURITY PATTERNS IN PRACTICE
Designing Secure Architectures Using Software Patterns
WILEY SERIES IN SOFTWARE DESIGN PATTERNS

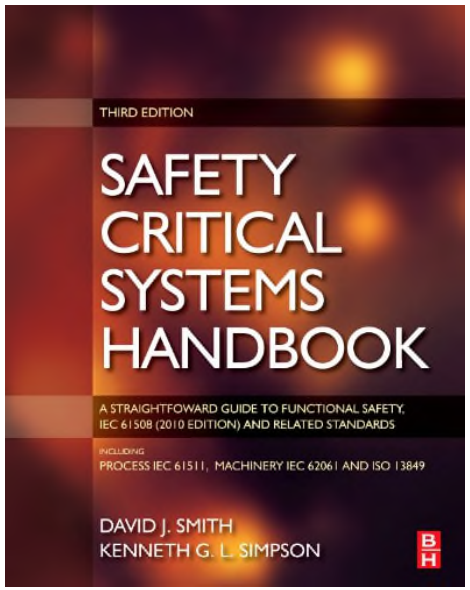
Safety

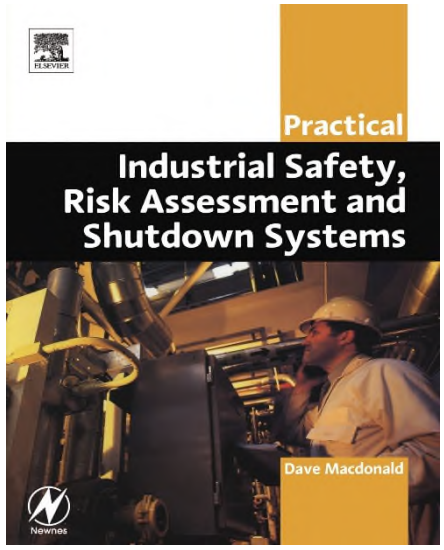

Engineering a Safer World
Systems Thinking Applied to Safety
Nancy G. Leveson

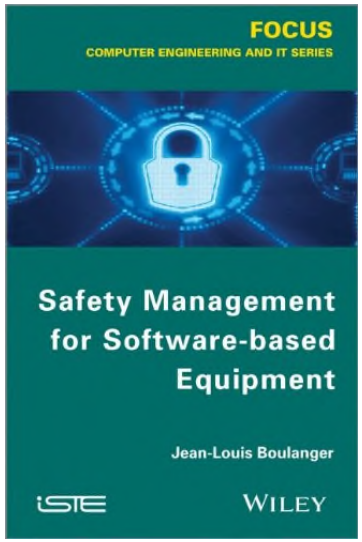

Reliability of Safety-Critical Systems
Theory and Applications
Marvin Rausand
WILEY

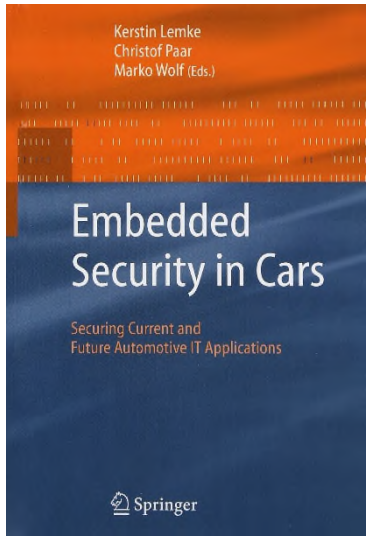

SAFETY MANAGEMENT SYSTEMS IN AVIATION
Second Edition
ASHGATE
Alan J. Stolzer and John J. Goglia

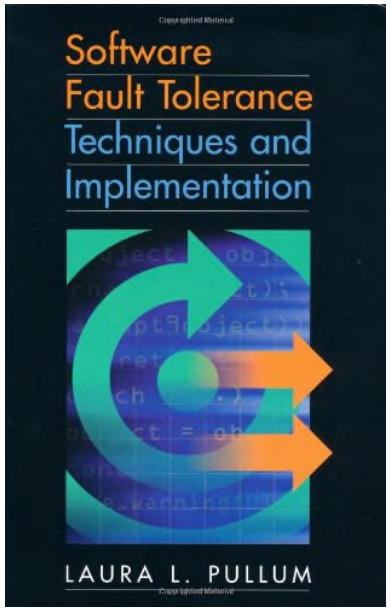

System Safety Engineering
Design Based Safety
Clifton A. Ericson II

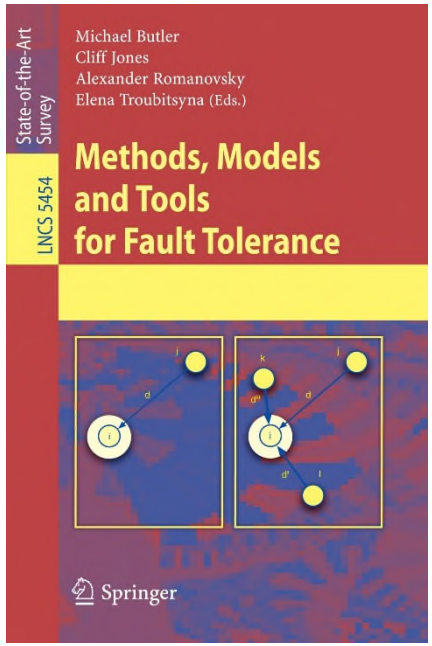

THIRD EDITION
SAFETY CRITICAL SYSTEMS HANDBOOK
A STRAIGHTFOWARD GUIDE TO FUNCTIONAL SAFETY, IEC 61508 (2010 EDITION) AND RELATED STANDARDS
INCLUDING PROCESS IEC 61511, MACHINERY IEC 62061 AND ISO 13849
DAVID J. SMITH
KENNETH G. L. SIMPSON
BH

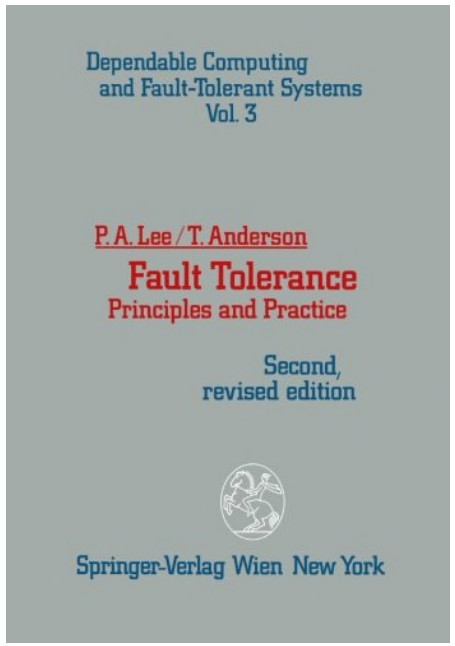

Practical
Industrial Safety, Risk Assessment and Shutdown Systems
Dave Macdonald
Newnes

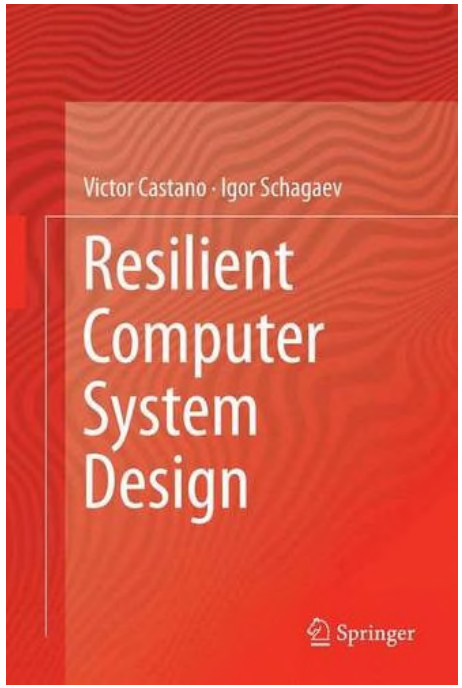

FOCUS
COMPUTER ENGINEERING AND IT SERIES
Safety Management for Software-based Equipment
Jean-Louis Boulanger
ISTE
WILEY


Kerstin Lemke
Christof Paar
Marko Wolf (Eds.)
Embedded Security in Cars
Securing Current and Future Automotive IT Applications
Springer

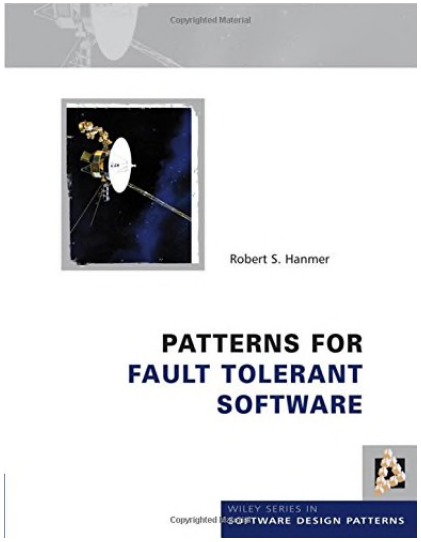


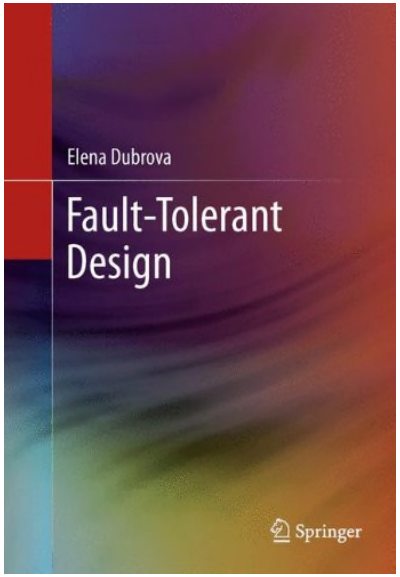


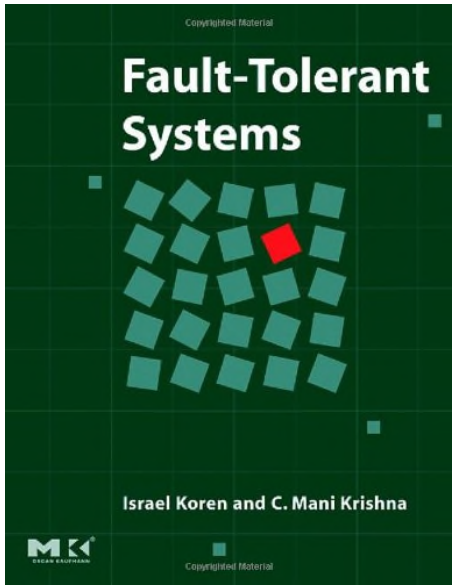




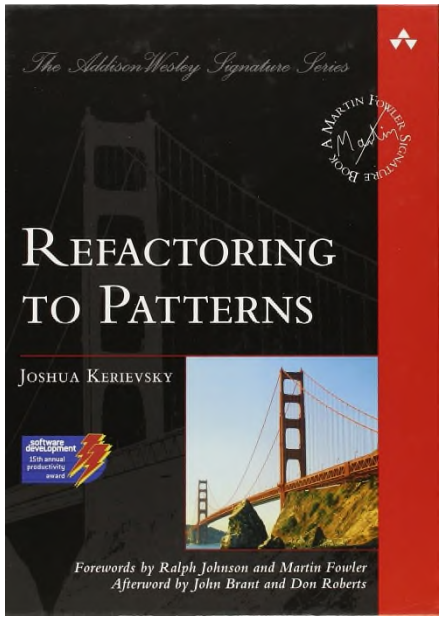
Fault-Tolerance

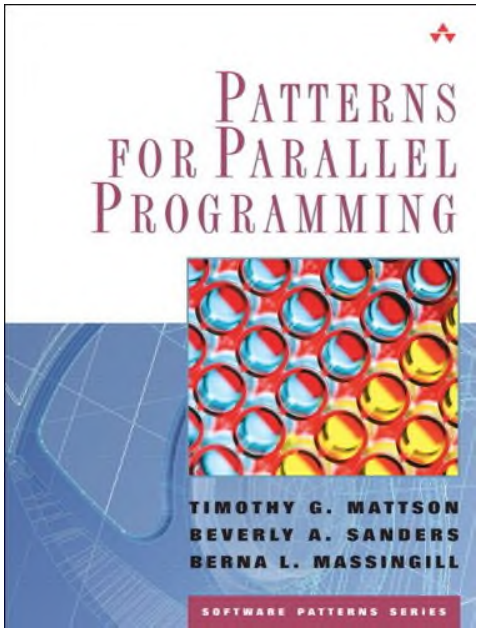




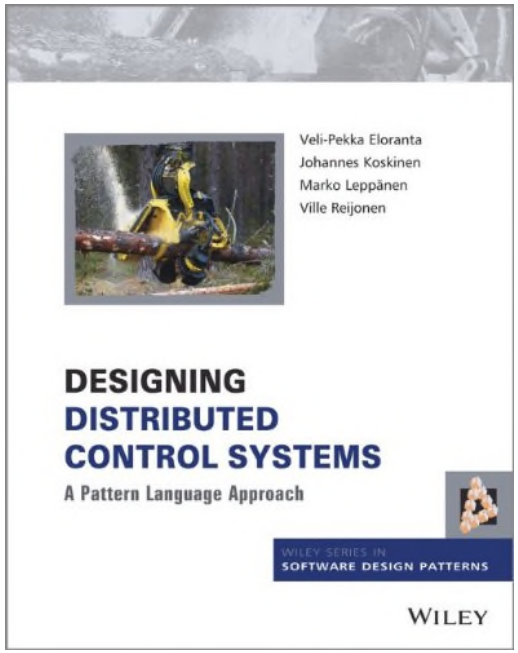


Patterns

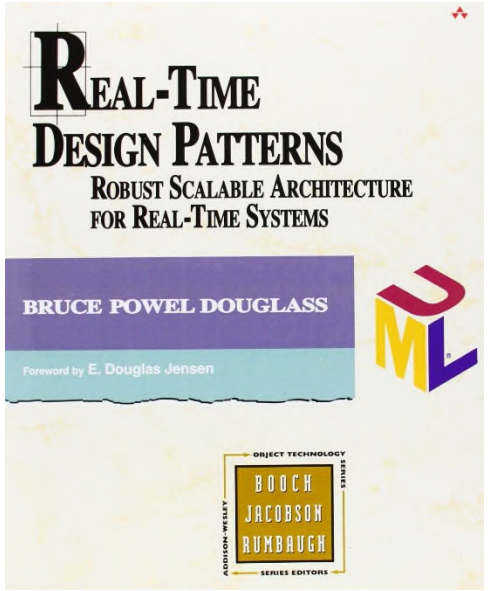

The Addison Wesley Signature Series
A Martin Fowler Signature Book
Refactoring to Patterns
Joshua Kerievsky
software development 15th annual productivity award
Forewords by Ralph Johnson and Martin Fowler
Afterword by John Brant and Don Roberts


Patterns for Parallel Programming
Timothy G. Mattson
Beverly A. Sanders
Berna L. Massingill
Software Patterns Series


Jorge Luis Ortega-Arjona
Patterns for Parallel Software Design
Wiley Series in Software Design Patterns


Veli-Pekka Eloranta
Johannes Koskinen
Marko Leppänen
Ville Reijonen
Designing Distributed Control Systems
A Pattern Language Approach
Wiley Series in Software Design Patterns
Wiley


Microservices: Patterns and Applications
Designing fine-grained services by applying patterns
Lucas Krause


Real-Time Design Patterns
Robust Scalable Architecture for Real-Time Systems
Bruce Powel Douglass
Foreword by E. Douglas Jensen
UML
Object Technology Series
Booch
Jacobson
Rumbaugh
Addison-Wesley
Series Editors

What can we learn in this lecture?

https://www.npmjs.com

We cannot become full **Resilience Engineers** in this lecture

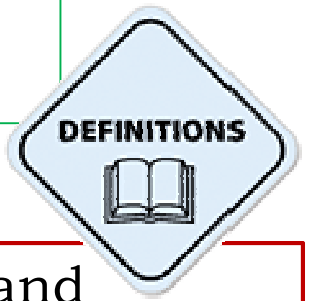
… but we can learn:

- The *methodology*
- Some important *principles & patterns*

Choice for this lecture

We will focus on Principles & Patterns for:

- Security
- Safety
- Integrity



Information **security** protects the confidentiality, integrity and availability of computer system data and functionality from unauthorized and malicious accesses

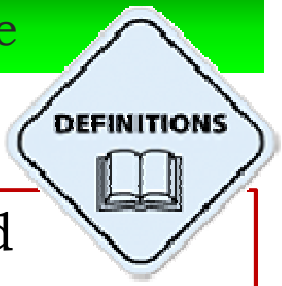
Safety is the state of being protected against failure, damage, error, accidents, harm, or any other event that could be considered non-desirable in order to achieve an acceptable level of risk

Integrity assures that functions or data are executed as implemented and have not been altered, modified, deleted or replaced

Future-Proof Software-Systems:

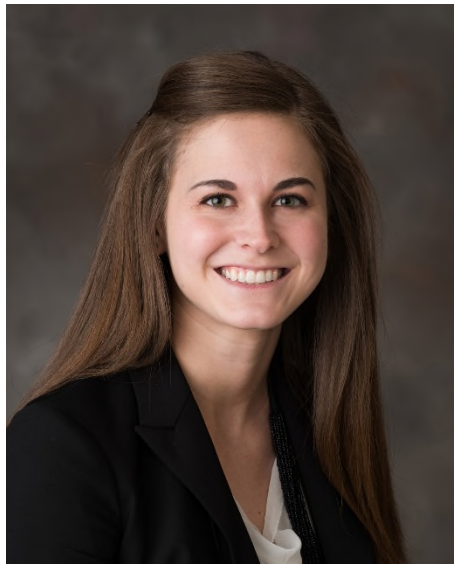
Specific (Property-dependent) Resilience Architecture Principles for:

Security



Information *security* protects the confidentiality, integrity and availability of computer system data and functionality from unauthorized and malicious accesses

Authentication



http://engineering.unl.edu

?

Electronic channel

http://www.rand.org

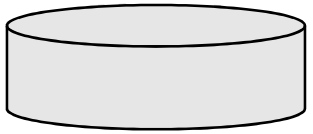
Real world

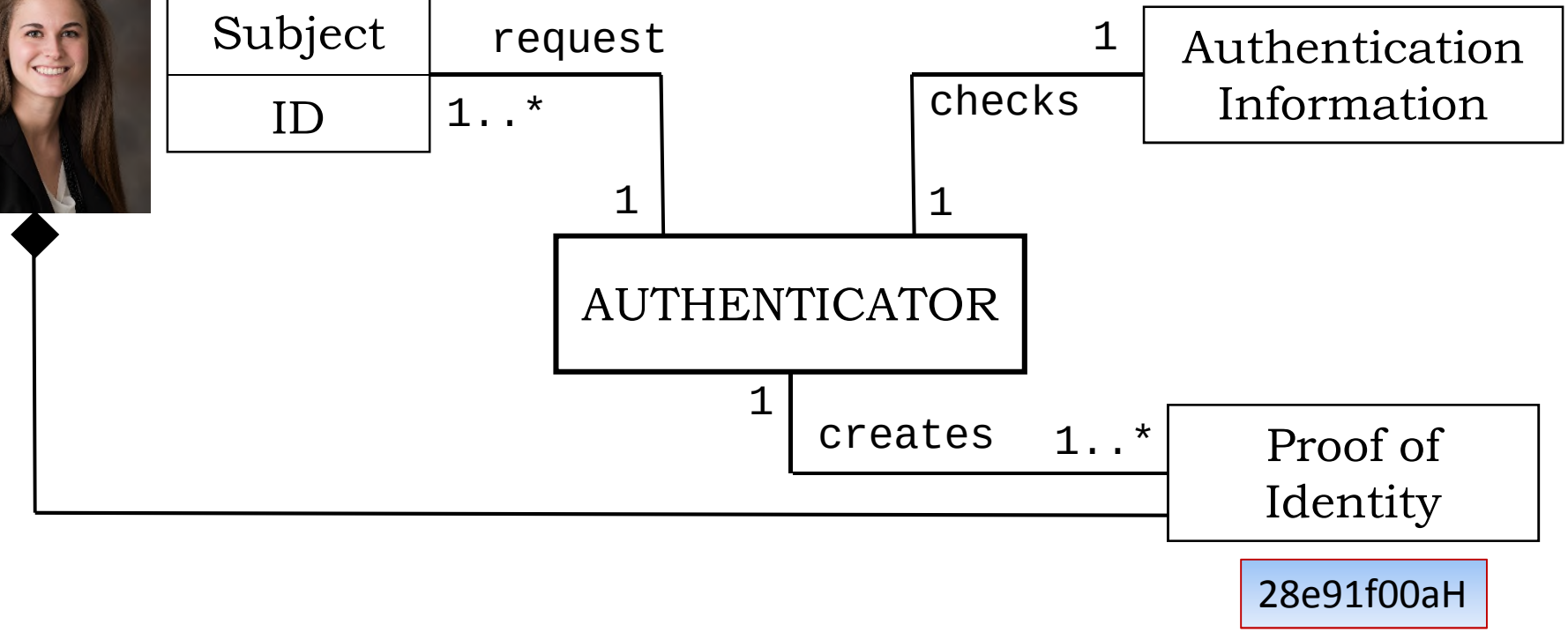
Computer representation

Authentication

Authenticator Pattern

[Schuhmacher, ISBN 978-0-470-85884-4]





Authentication

Authenticator Pattern

[Schuhmacher, ISBN 978-0-470-85884-4]

User Name
Password

Subject

ID

request

1..*

1

checks

1

Authentication Information

AUTHENTICATOR

1

creates

1..*

Proof of Identity

28e91f00aH

Authorization

Authorization:
Allow or deny access
to a computer resource

http://stonehousesigns.com

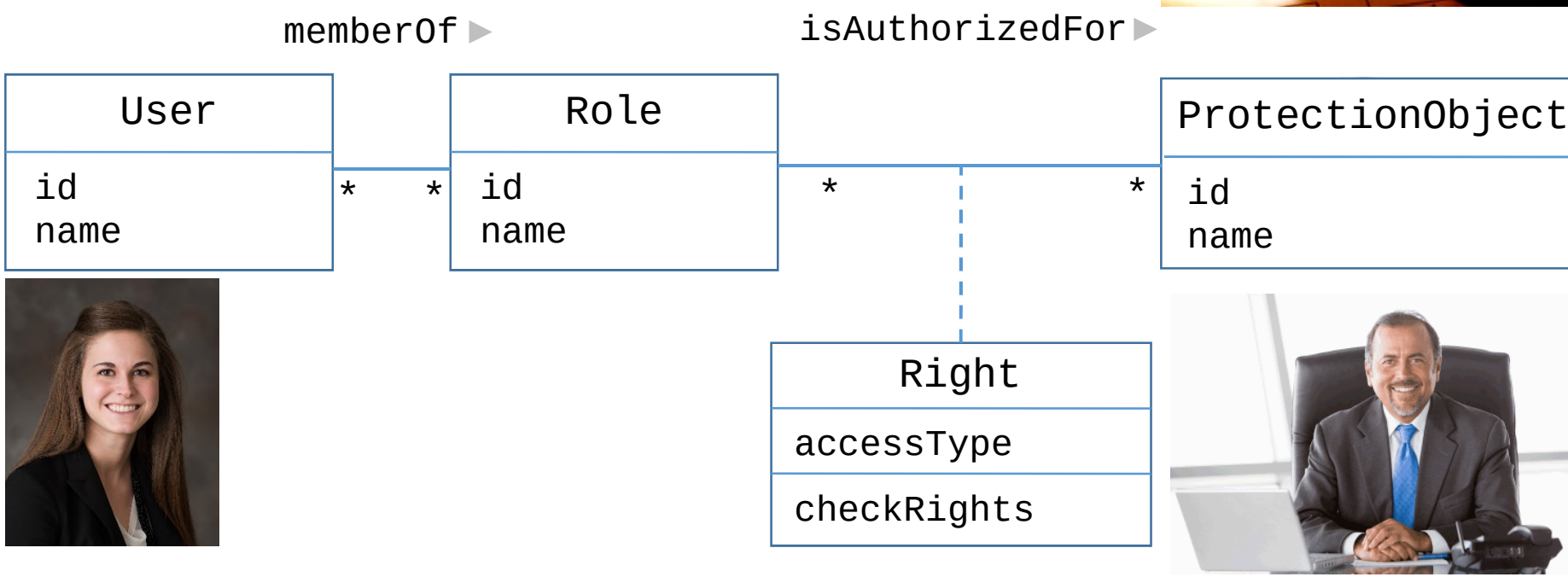
RBAC Pattern
[Fernandez, ISBN 978-1-119-99894-5]

Full Access with Errors Pattern
[Schuhmacher, ISBN 978-0-470-85884-4]

Authorization

RBAC Pattern

[Fernandez, ISBN 978-1-119-99894-5]



The `User` and `Role` classes describe registered users and their predefined roles. Users are assigned to roles, roles are given rights according to their functions

Authorization

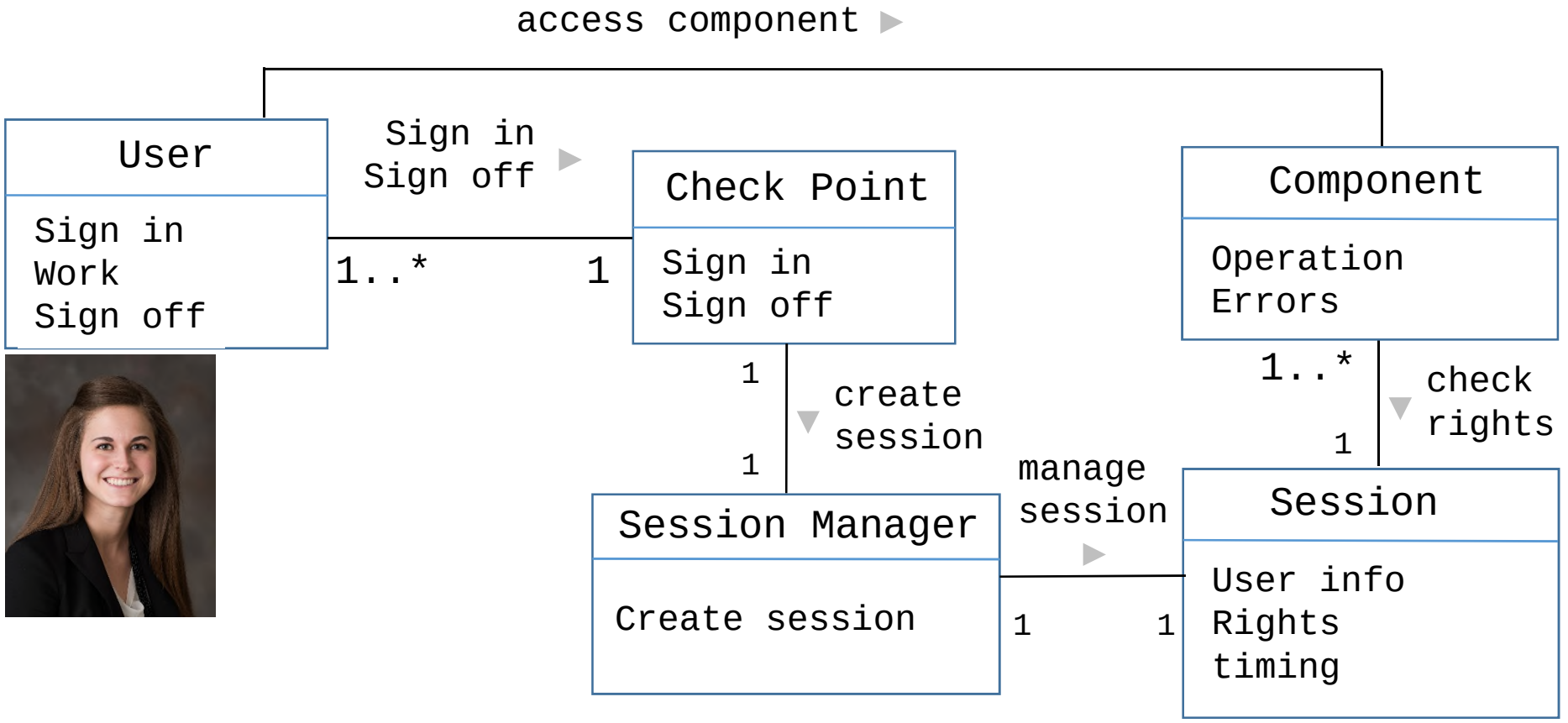
Full Access with Errors Pattern

[Schuhmacher, ISBN 978-0-470-85884-4]

Access Control

Security Session Pattern

[Schuhmacher, ISBN 978-0-470-85884-4]

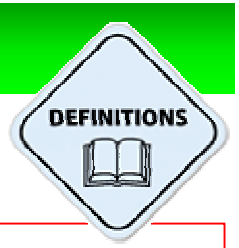


Future-Proof Software-Systems:

Specific (Property-dependent) Resilience Architecture Principles for:

Safety

Safety



Safety is the state of being protected against *failure*, damage, error, accidents, harm, or any other event that could be considered non-desirable in order to achieve an acceptable level of risk

Protection Assets

- …
- Life & Health of Passengers
- …

Threats & Dangers

- …
- Electronic System Failure
- …

Protection Means

- …
- Fault tolerance ❶
- Safe states ❷
- …

Safety

Architectural Patterns
[Hanmer, ISBN 978-0-470-31979-6]

Detection Patterns
[Hanmer, ISBN 978-0-470-31979-6]

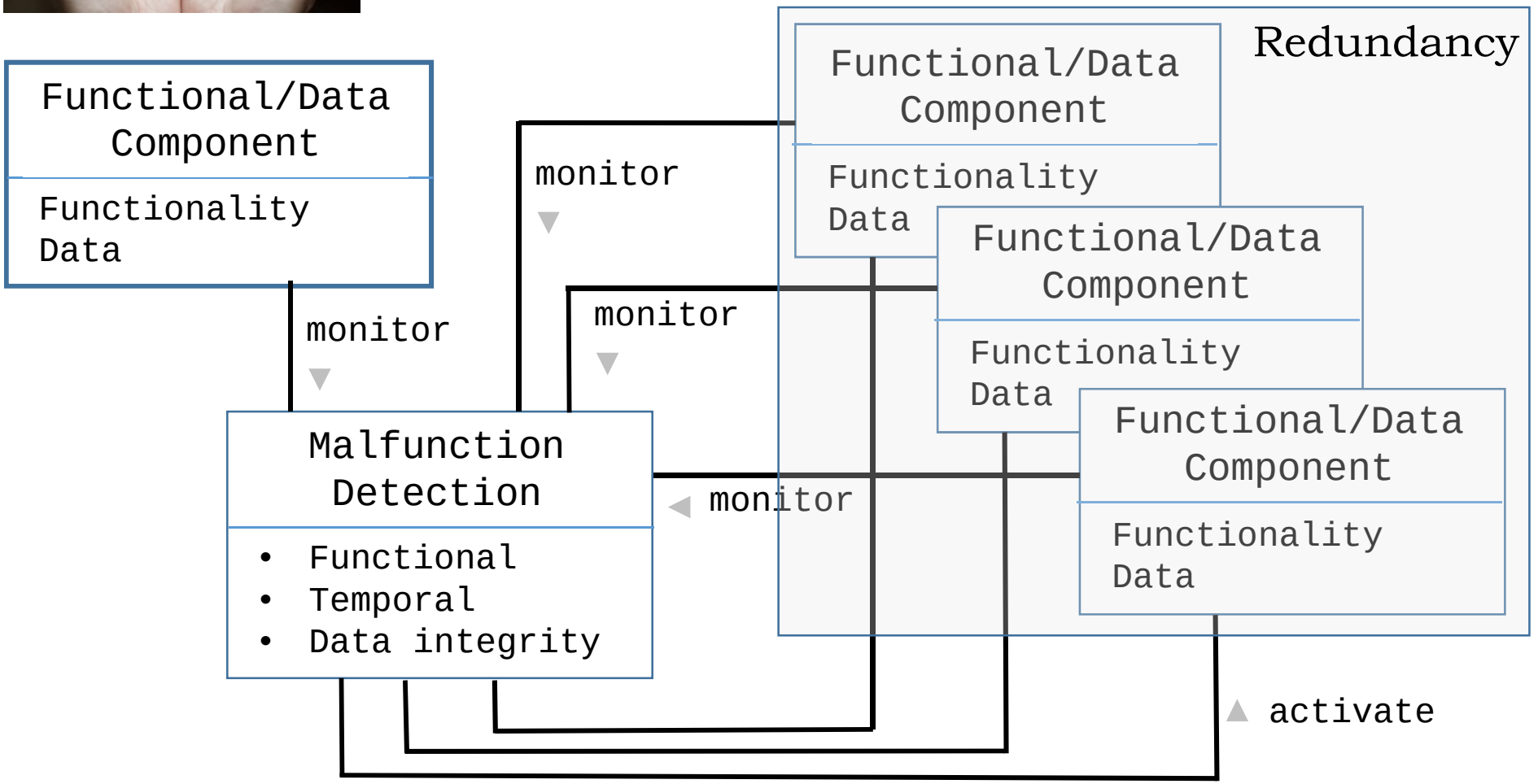
Error Recovery Patterns
[Hanmer, ISBN 978-0-470-31979-6]

Error Mitigation Patterns
[Hanmer, ISBN 978-0-470-31979-6]

Fault Treatment Patterns
[Hanmer, ISBN 978-0-470-31979-6]

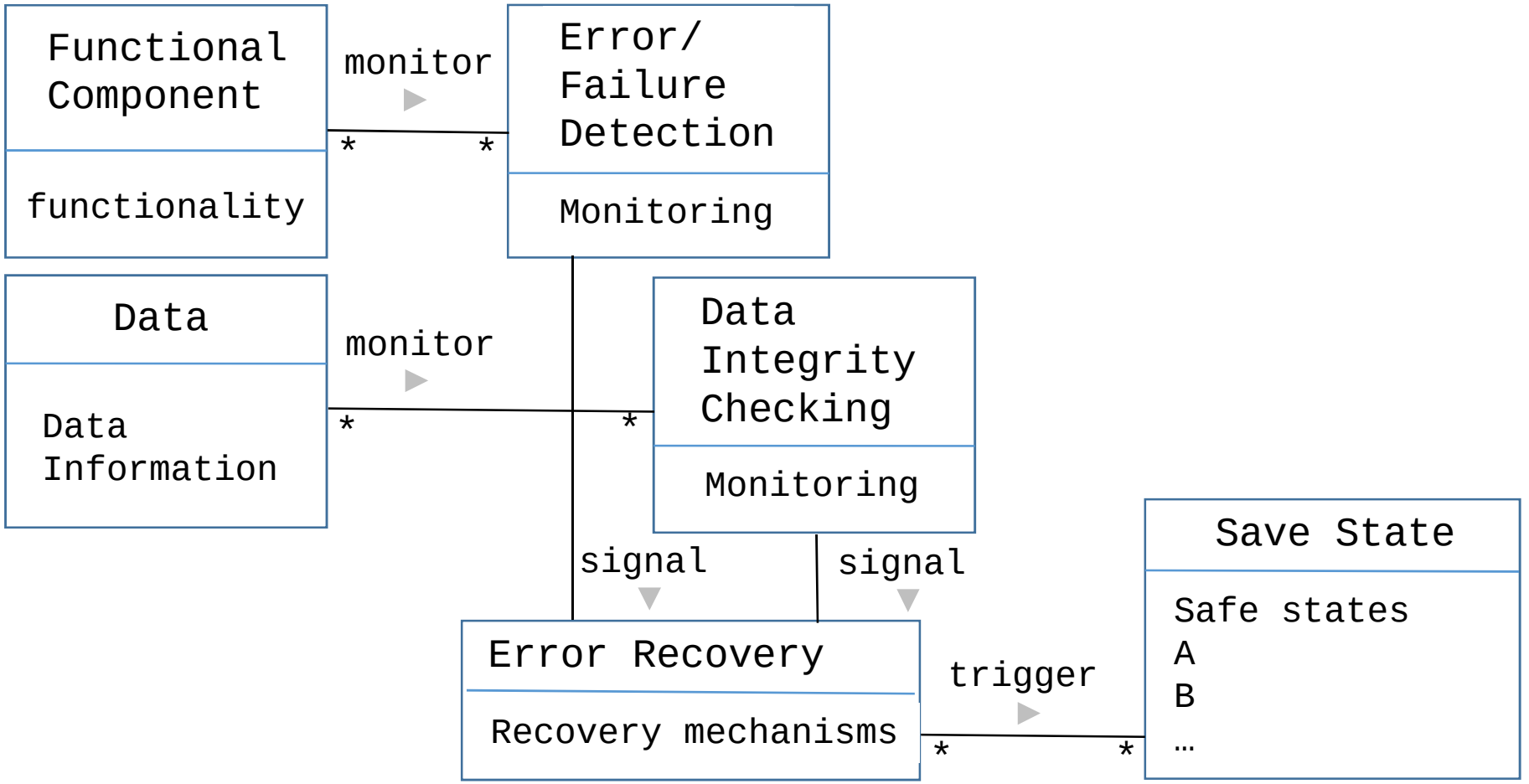
Safety

Fault Tolerance



Safety

Fail Safe State Pattern



Future-Proof Software-Systems:

Specific (Property-dependent) Resilience Architecture Principles for:

Integrity

http://www.phdconstruction.com/

Enterprise Record Archive

Each enterprise is obliged by law to generate records and to keep them available *unaltered* for 10 years (CH)

Protection Assets

- Electronic Records (Archive)

Threats & Dangers

- Record modification

Protection Means

- Integrity protection

http://www.nelbaliproperty.com

Data Integrity

How can you assure and prove that a *digital document* has not been altered?

Digital Data Integrity Protection Technolog
Hashing & Digital Signatures

Hashing & *Digital Signatures* are a secure technique to preserve the integrity of a digital document over long periods of time

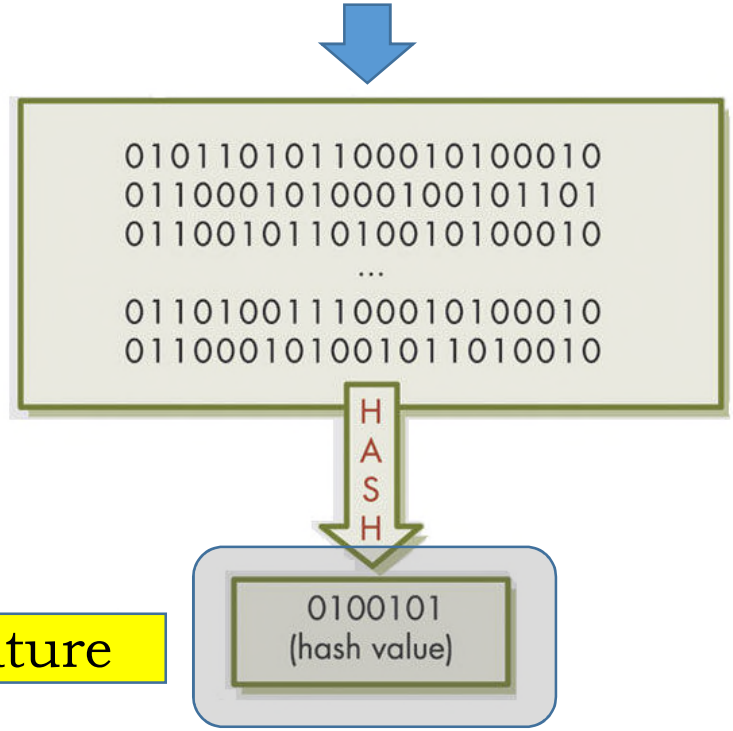
https://www.tsl.texas.gov/

Data Integrity

http://www.nelbaliproperty.com

Hashing & Digital Signatures

http://de.dreamstime.com

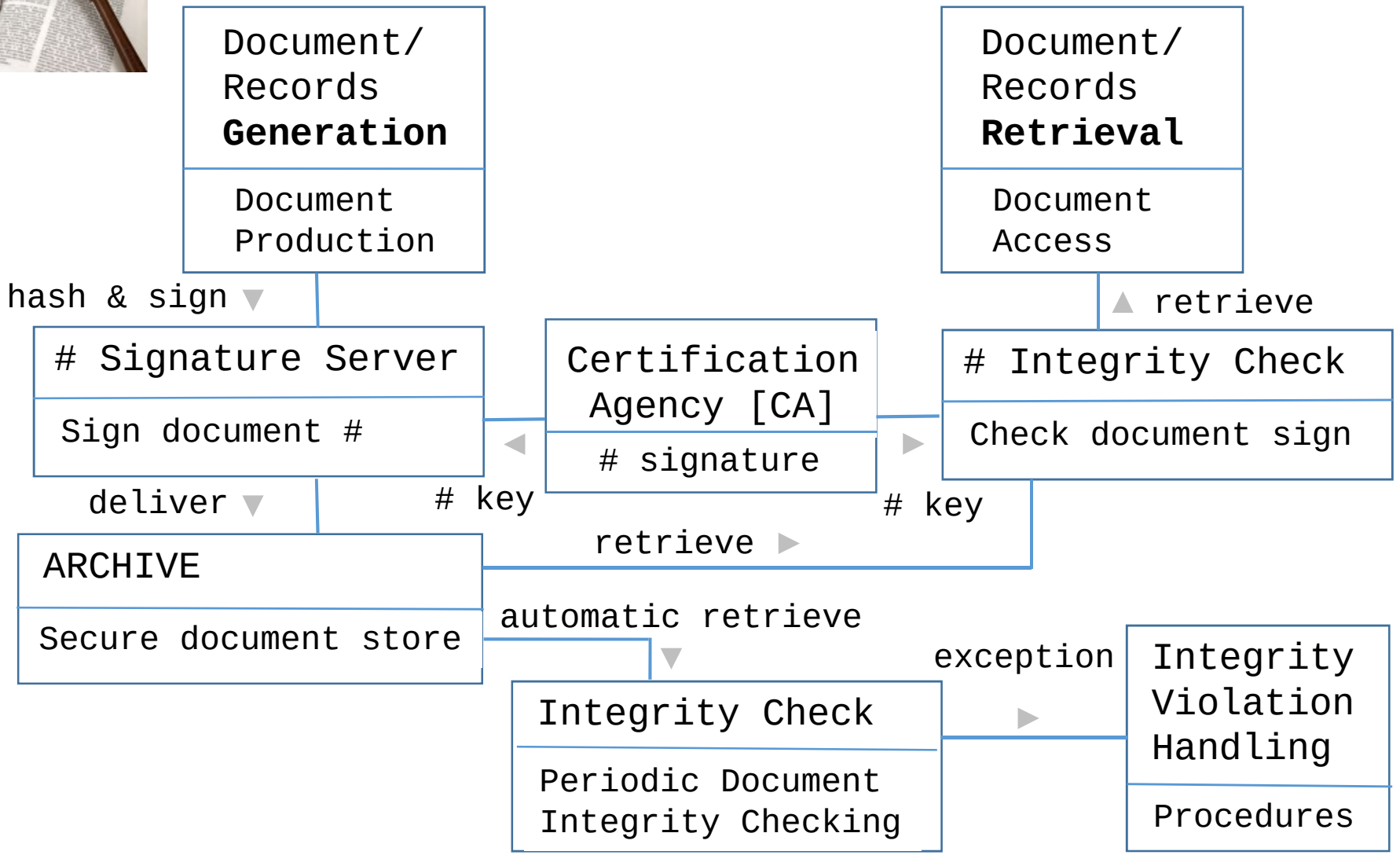


http://www.3gforensics.co.uk

Document ID
Time Stamp
Hash Value

Digital Signature

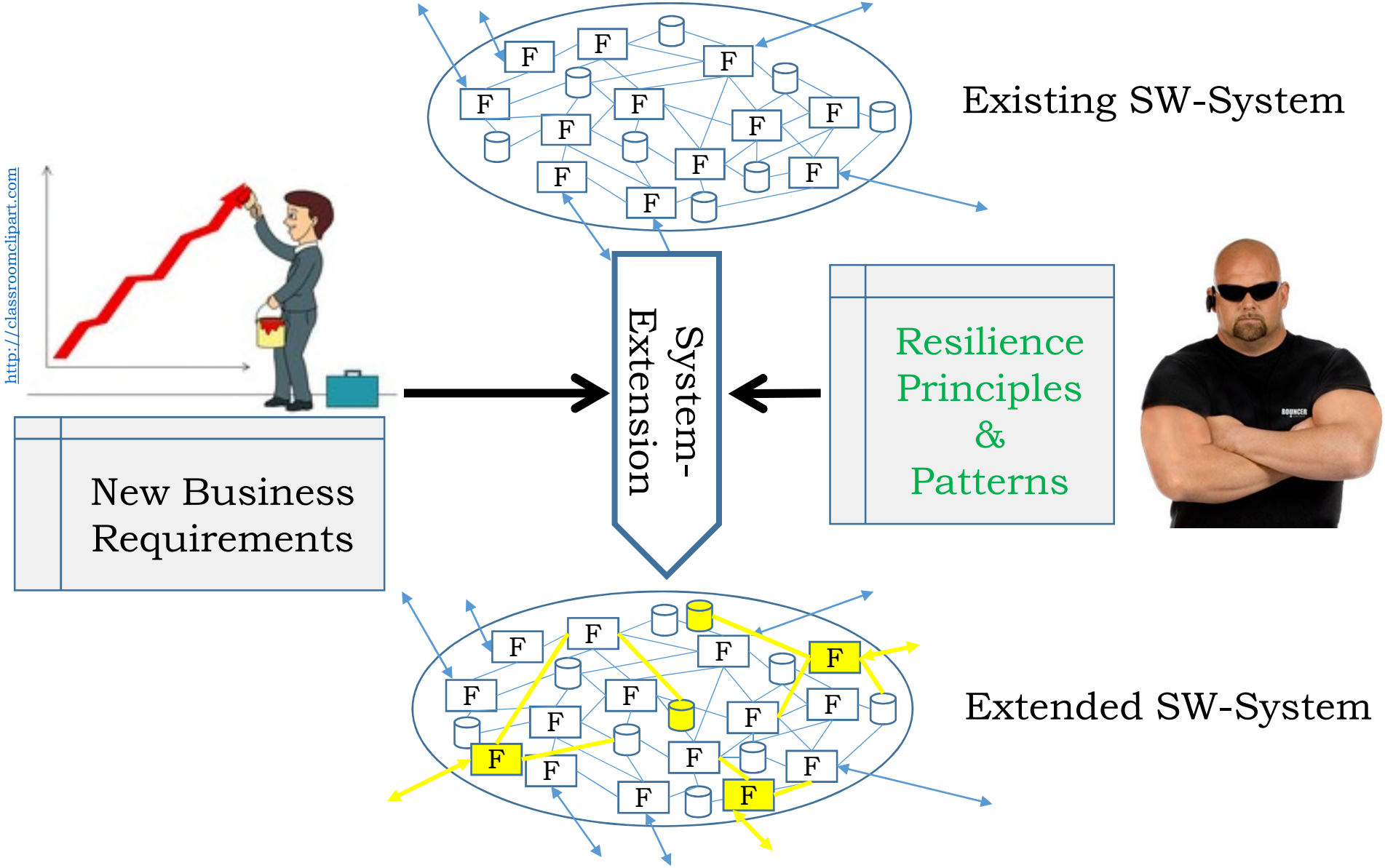
Data Integrity Assurance Pattern



Future-Proof Software-Systems:

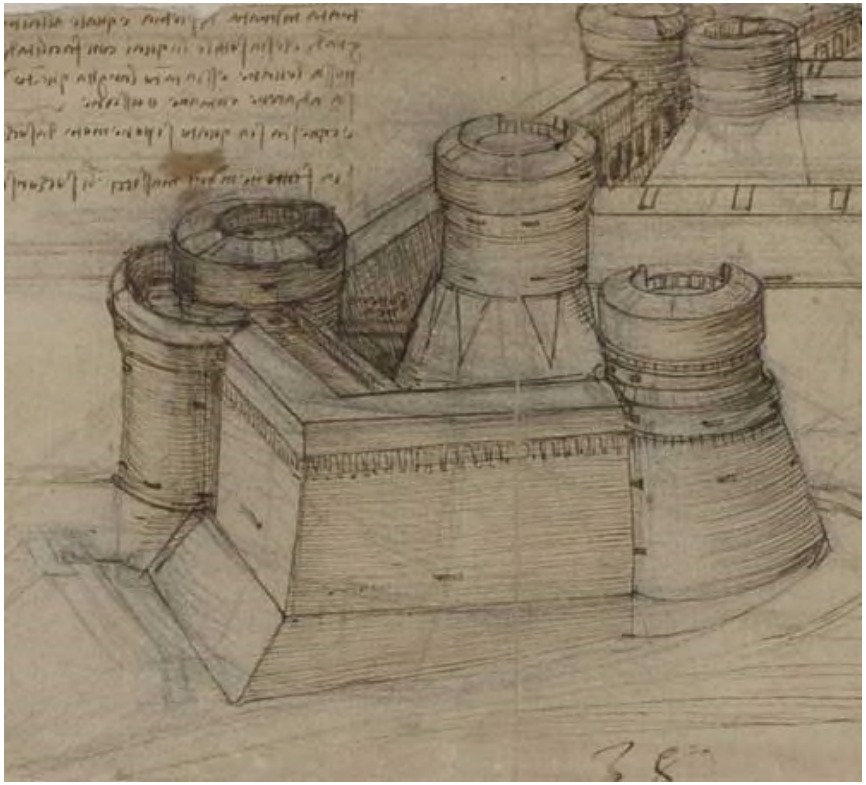
Principle-Based Resilience Engineering [Key Points]

Future-Proof Software-Systems: Resilience Engineering



Resilient Architecture

Resilience Architecture *Principles & Patterns*



http://www.marinabaysands.com

Strong, resilient architecture

The Principles & Patterns Challenge

Principles & Patterns =

Very valuable software/system *architecture knowledge* in proven & easily accessible form

https://pmtgurumantra.wordpress.com

… However,

Principles & Patterns are *not* directly applicable
(⇒ they represent generic solutions)

Principles & Patterns in the literature are often *inconsistent*:

- Different terminology
- Different representations
- Duplicated functionality
- Functional overlaps
- … even some contradictions

https://pmtgurumantra.wordpress.com

Example

Subject
ID

User
id name

Authenticator Pattern
[Schuhmacher, ISBN 978-0-470-85884-4]

RBAC Pattern
[Fernandez, ISBN 978-1-119-99894-5]

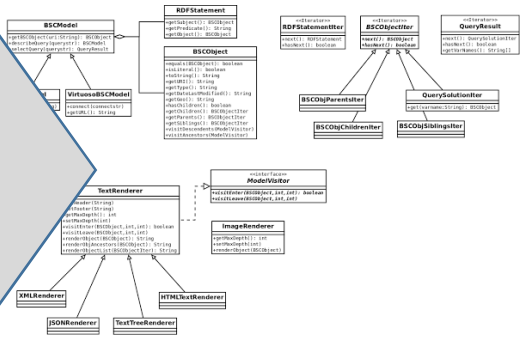
https://cms.sachsen.schule

Principles & Patterns are *not* directly applicable:

⇒ Need for adaptation & integration into *your* system!

Learn & understand

Apply & enforce

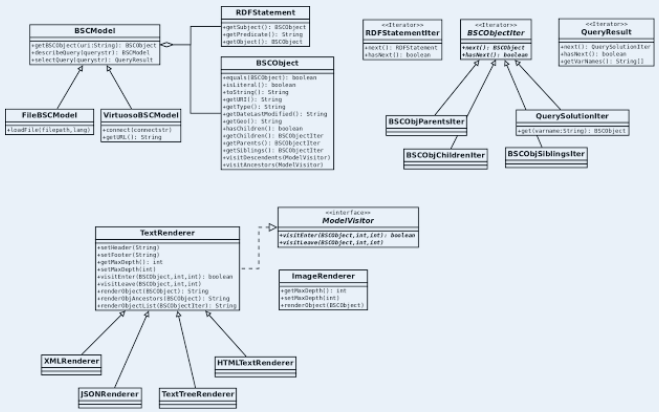


Future-Proof Software-Systems Engineer

Architecture & Resilience *Principles & Patterns*

Future-Proof Software-Systems Engineer

Apply & enforce



Pattern Integration

Future-Proof Software-Systems:

Autonomic Computing

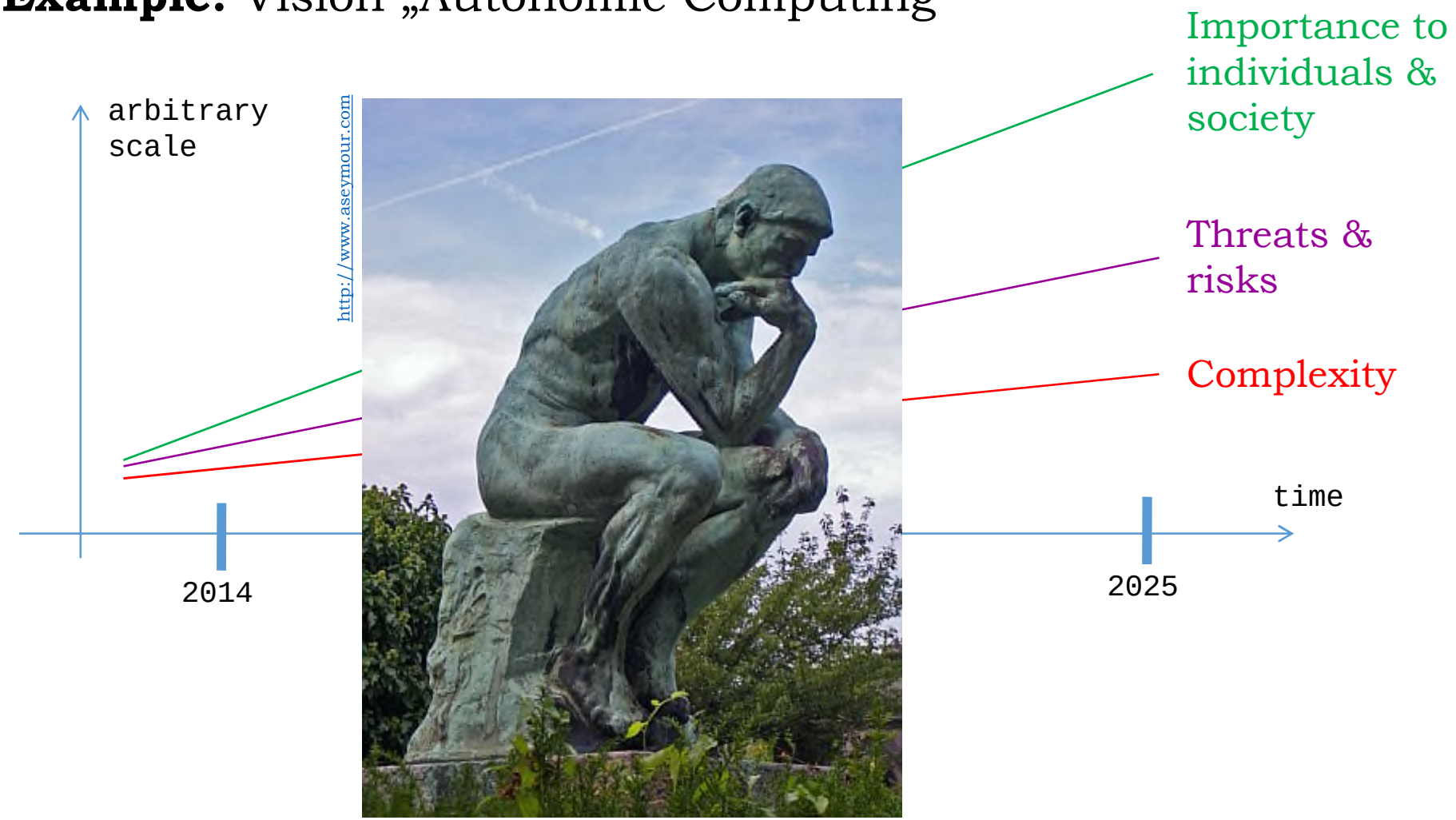
❶ **Specific** Countermeasures (⇔ Risk Analysis) → Resilient Code

❷ **Architectural** Countermeasures (Principles) → Resilient Architecture

❸ **Adaptive Behaviour** («Autonomic Computing») → Resilient System

«Autonomic Computing»

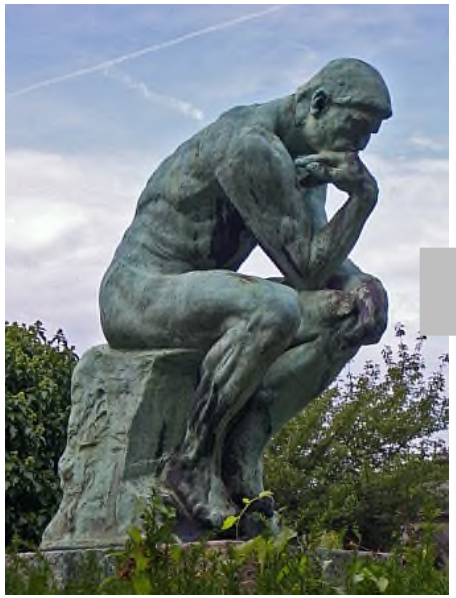
Example: Vision „Autonomic Computing“



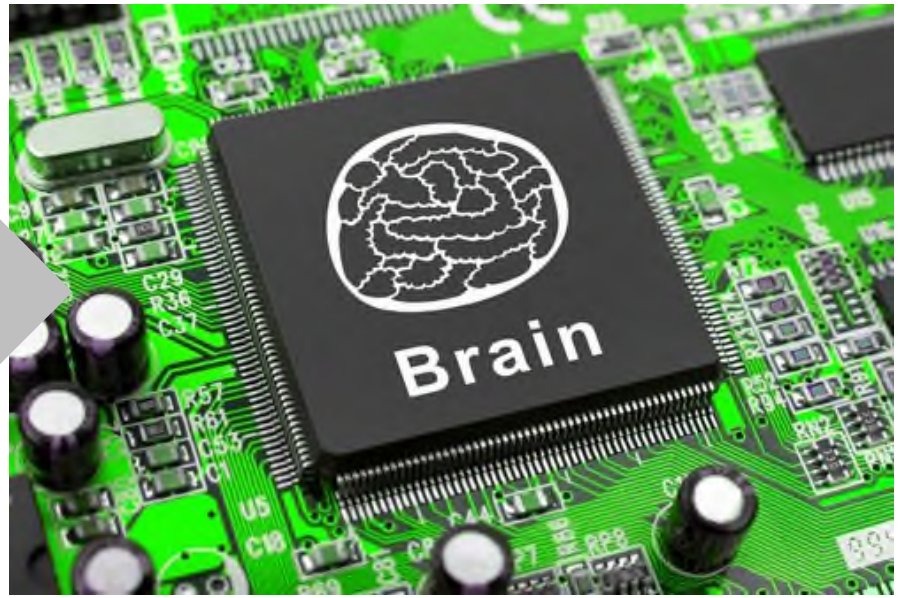
Can we humans successfully cope with these trends ?

Example: Vision „Autonomic Computing“

http://www.aseymour.com



Probably NOT



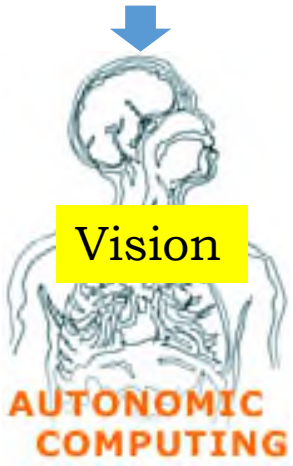
http://mcvcbca.blogspot.ch/2012

Can we humans successfully cope with the trends of:

- increasing complexity
  - raising importance
- accelerating threats & risks

?

… we will need the support of intelligent machines



Paul Horn, **IBM**, 2001

Autonomic Computing (IBM Concept 2001)

https://www.illustrationsource.com

Basic idea: Enable the computer for self-defense

< 2010 ⟶ > 2010

Disruptive Incidents
1. Xxx
2. Yyy
3. Zzz

System

Safety Case

predict

Complexity

t

Tractability

t

Incident

Resilient System

adjust

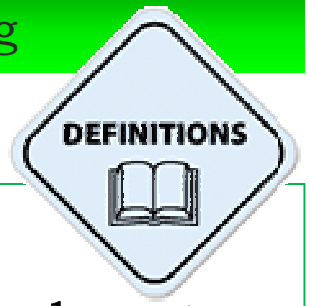
https://www.illustrationsource.com

Autonomic Computing Vision

Autonomic Computing has four areas of concern:

1. Self-Configuration
2. Self-Healing (error correction, fault recovery)
3. Self-Optimization (automatic resource control for optimal functioning)
4. **Self-Protection** (identification and protection from attacks in a proactive manner)

… often called the self-X properties



Autonomic =
Computer systems and networks that configure themselves to changing conditions and are self-healing in the event of failure and self-defending in the event of attacks. "Autonomic" means "automatic responses" to unpredictable events.

Autonomic computing is an approach to address the complexity and evolution problems in software systems.

A software system that operates *on its own* or with a minimum of human interference according to a set of rules is called autonomic.

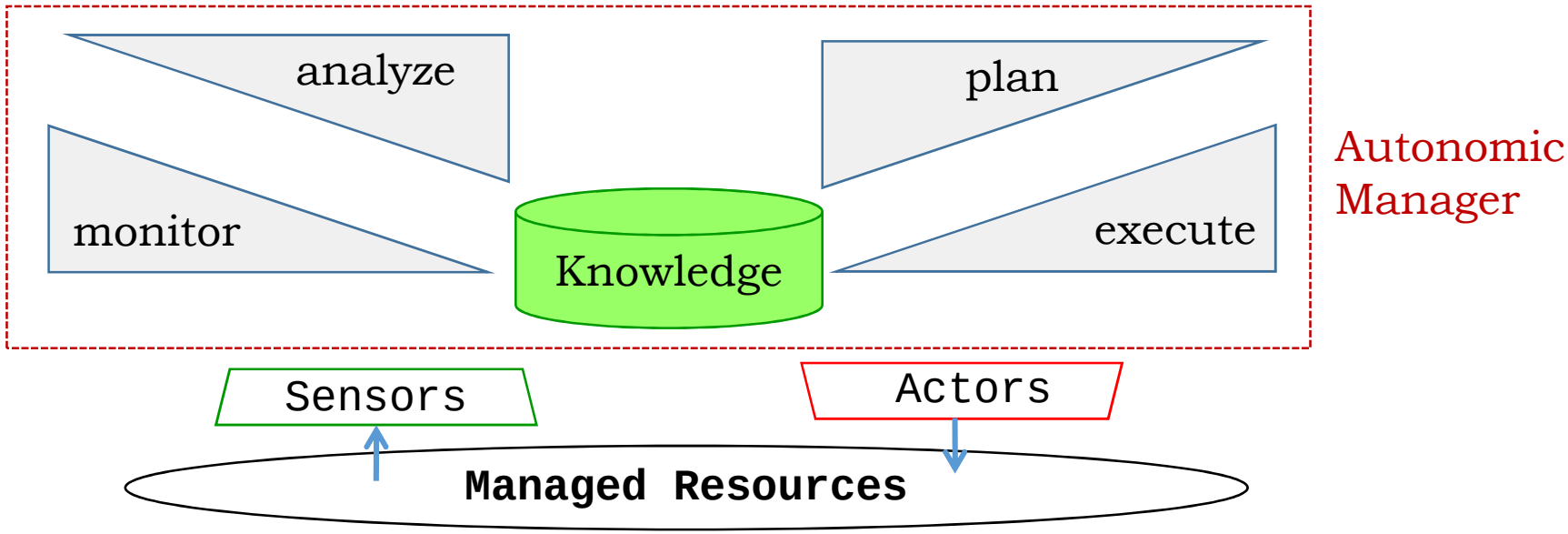
Modern, active research area

Example: Vision „Autonomic Computing“

„Horn envisioned autonomic computers that could sense, analyze and respond to situations automatically“

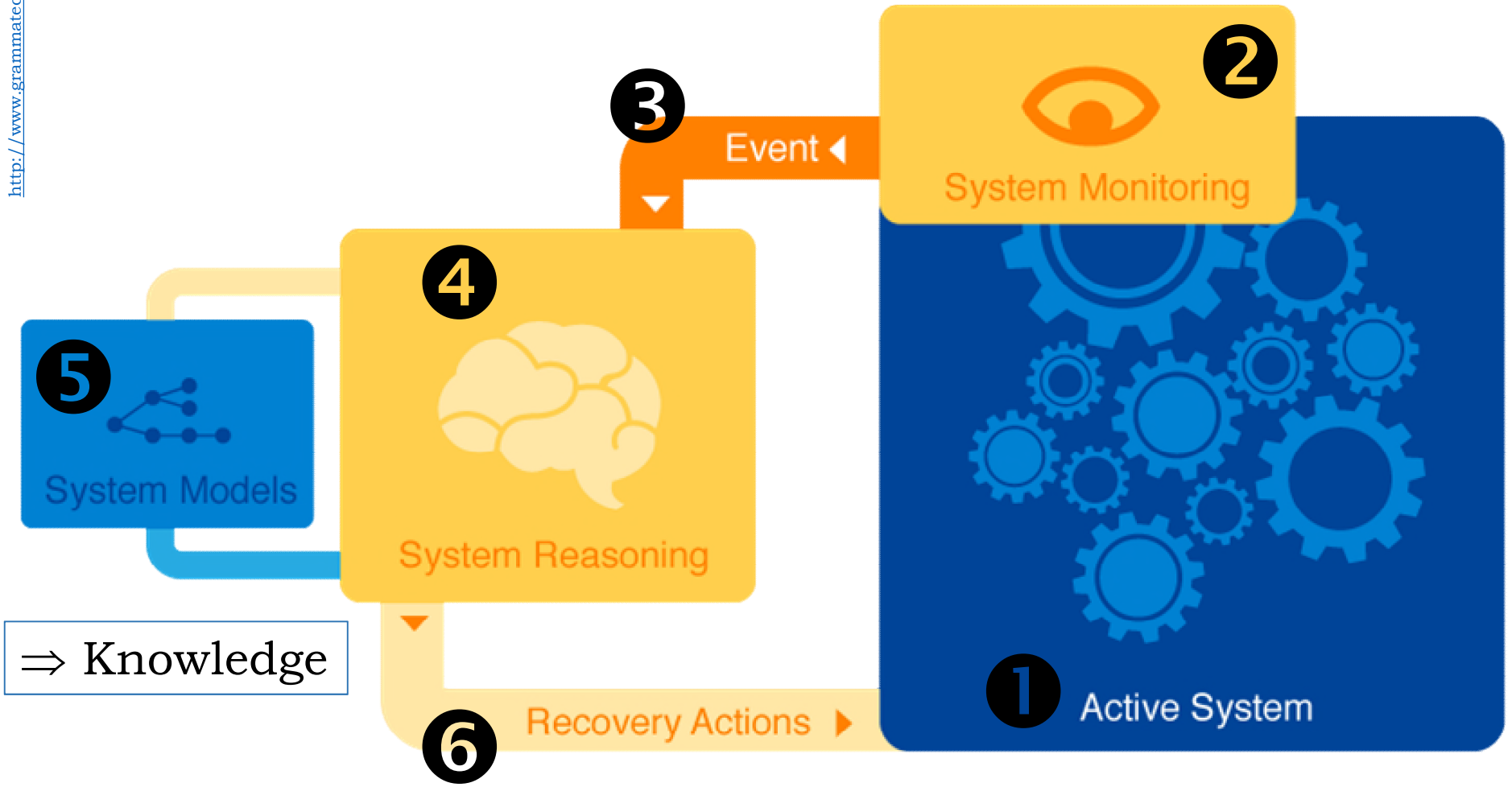
Self-protecting **Self-optimizing**

Self-healing **Self-configuring**



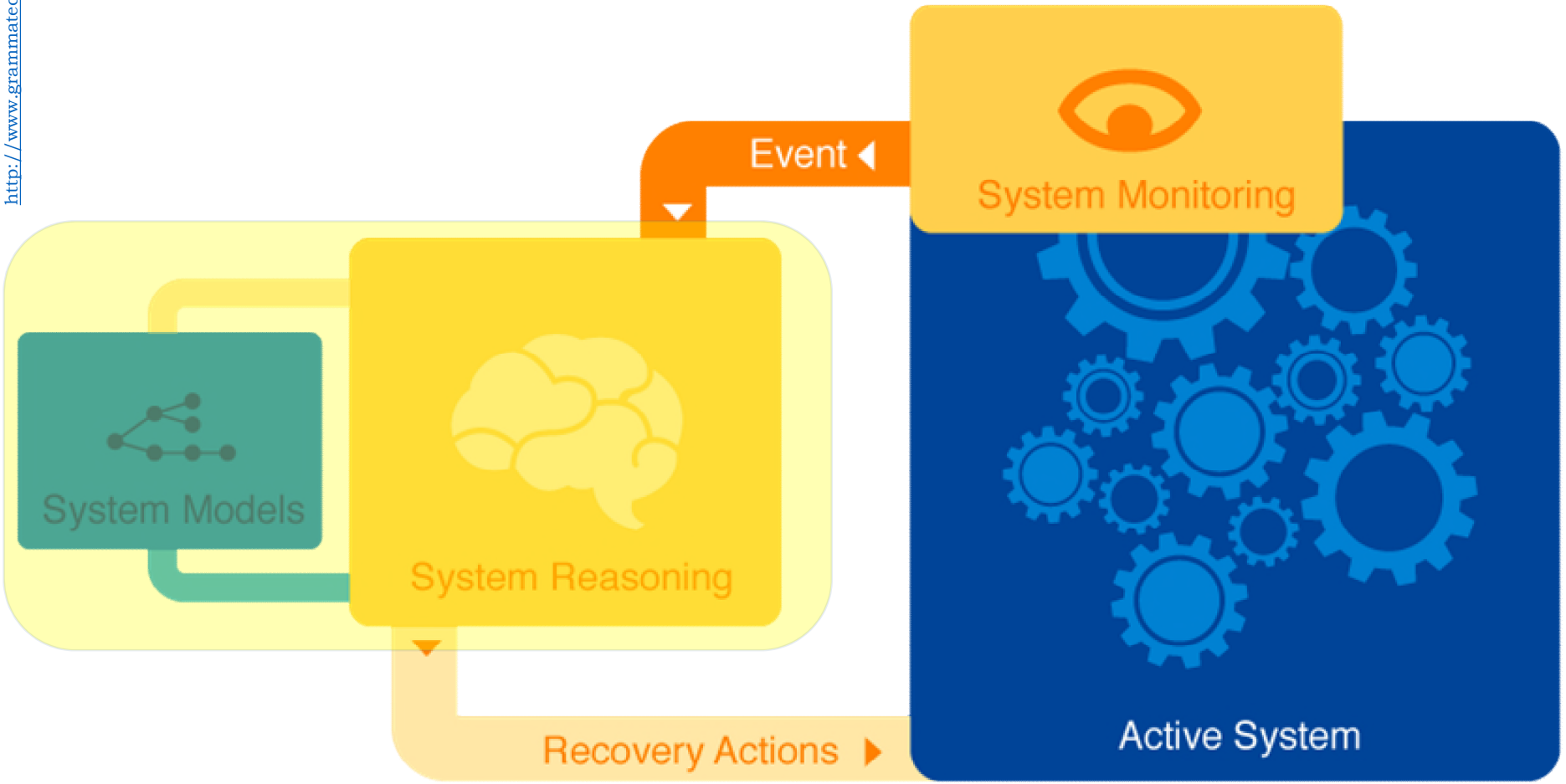
Autonomic Computing Schema:

http://www.grammatech.com



http://www.grammatech.com

Autonomic Computing Schema:



Artificial Intelligence

Example: Autonomic Computing for Traffic Control

http://iict.bas.bg/acomin/events/3-4-October-2013/KStoilova_ppt.pdf

Large area traffic control

http://www.autoevolution.com

- The autonomy is achieved by applying *multilevel optimization* for the control process
- Thus, an increase of the transport parameters, defined as solutions of the optimization problem, is achieved.

Applications:

Rules, optimization and artificial intelligence (cognition) instead of fixed algorithms

References:

References	
Hariri06	Salim Hariri, Manish Parashar (Editors): **Autonomic Computing - *Concepts, Infrastructure, and Applications*** CRC Press Inc., Boca Raton, USA, 2006. ISBN 978-0849393679
IBM06	IBM Business Consulting Services: **An Architectural Blueprint for Autonomic Computing** IBM Autonomic Computing, 4th edition, June 2006. Downloadable from: http://www-01.ibm.com/software/tivoli/autonomic/
Kephart03	Jeffrey O. Kephart, David M. Chess: **The Vision of Autonomic Computing** IEEE Computer Society, New York, January 2003, pp. 41-50. Downloadable from: http://www-01.ibm.com/software/tivoli/autonomic/
Murch04	Richard Murch: **Autonomic Computing** IBM Press, Prentice Hall PTR, NJ, USA, 2004. ISBN 978-0-13-315319-3
Cong-Vinh11	Phan Cong-Vinh (Editor): **Formal and Practical Aspects of Autonomic Computing and Networking – *Specification, Development, and Verification*** Premier Reference Source, Information Science Reference Publishing, 2011. ISBN 978-1-60960-845-3
Müller06	Hausi A. Müller, Liam O'Brien, Mark Klein, Bill Wood: ***Autonomic Computing*** Carnegie Mellon University, Technical Note CMU/SEI-2006-TN-006, 2006. Downloadable from: http://www.sei.cmu.edu/reports/06tn006.pdf [last accessed 14.1.2016]
Lalanda13	Philippe Lalanda: **Autonomic Computing – *Principles, Design and Implementation*** Springer-Verlag, London, 2013. ISBN 978-1-4471-5006-0

Future-Proof Software-Systems: Content

This is the end of Parts 1-4:

You know now the *foundations* and *principles* of future-proof software-systems

Part 5:

Describes the „future-proof software-systems engineer“ and his working context