Software Architecture for Critical Infrastructures – Engineering for the Unknown –

Barbora Buhnova, **Lasaris** Summer School**’21**, September 9, 2021

“Bridging communities to foster innovation.”

Barbora Bühnová

Vice-dean, Masaryk University
Chair of ICSA Steering Committee
Co-founding & Gov. Board, Czechitas

czechitas

Masaryk University, Brno, Czech Republic

- Masaryk University (MU)
  - Established in 1919
  - 2nd largest in Czechia
  - Over 30,000 students
- Faculty of Informatics, MU
  - Established in 1994
  - 1st faculty of comp. science
  - Over 2,000 students

MUNI
FI

Czech CyberCrime Centre of Excellence C4e

- A multidisciplinary center that brings together expert academic departments to address complex cyberspace problems

MUNI

MUNI ICS

MUNI FI

MUNI LAW

NÚKIB

CONCORDIA

Cyber Security for Europe

National Cybersecurity R&D Laboratory

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

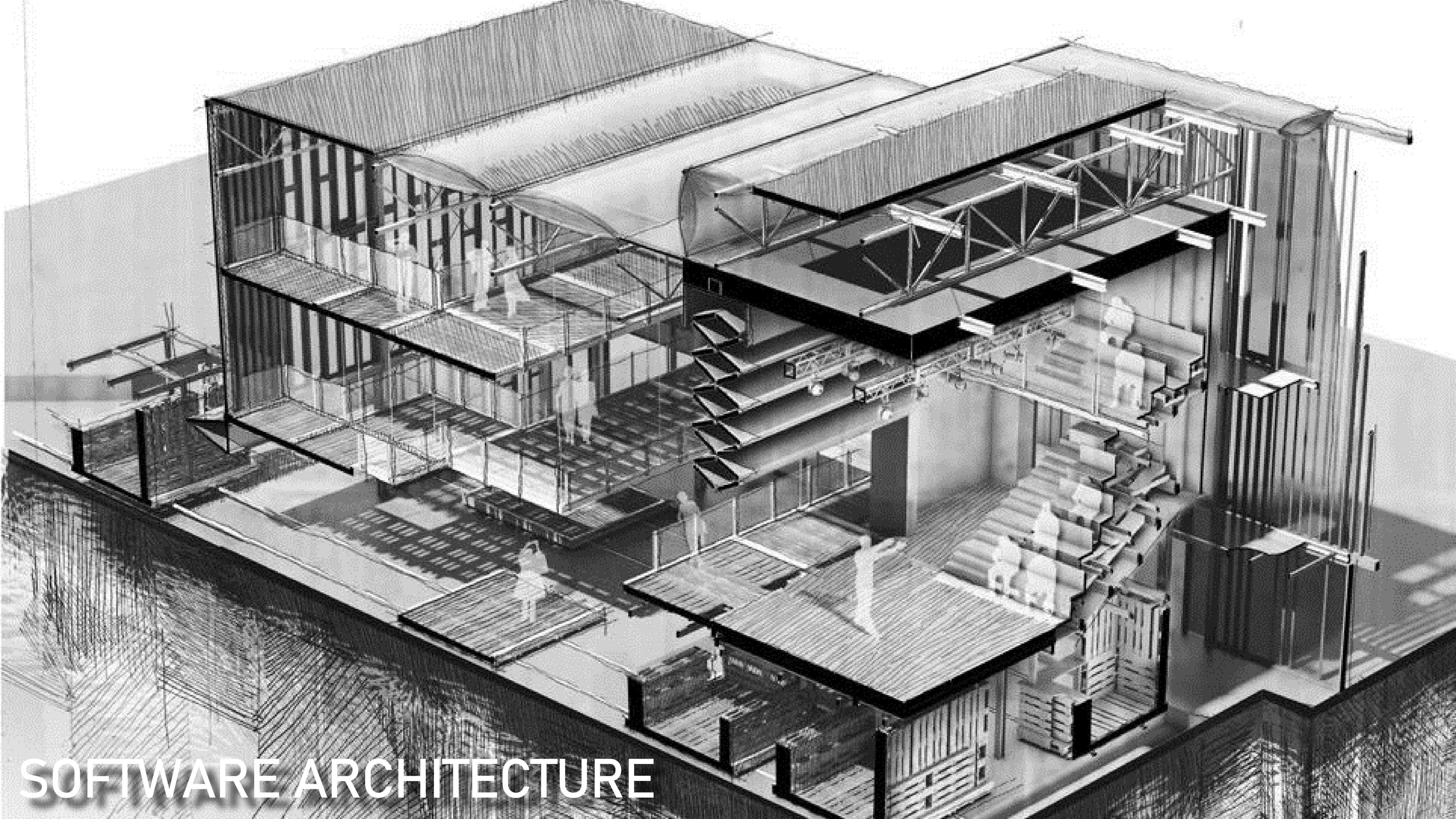
MINISTRY OF EDUCATION, YOUTH AND SPORTS

Cybersecurity Innovation Hub

Coordinated by National Cyber Security Competence Centre (NC3)

- Key initiatives
  - Computer Security Incident Response Team (CSIRT) of MU https://csirt.muni.cz
  - Lab of Software Architectures and Information Systems https://www.lasaris.cz
  - Institute of Law and Technology at MU https://cyber.law.muni.cz
  - CyberRange (Kybernetický polygon, KYPO) https://www.kypo.cz
- Collaboration on
  - Cybersecurity Education (National CyberCzech Technical Exercise, Cybersecurity Qualification Framework)
  - Policy and Legislation in Cybersecurity (Cyber Security Act, Methodology)
- Partners
  - Masaryk University, Brno University of Technology
  - Czech National Cybersecurity Agency, Network Security Monitoring Cluster
  - Regional Chamber of Commerce, Industry Cluster 4.0

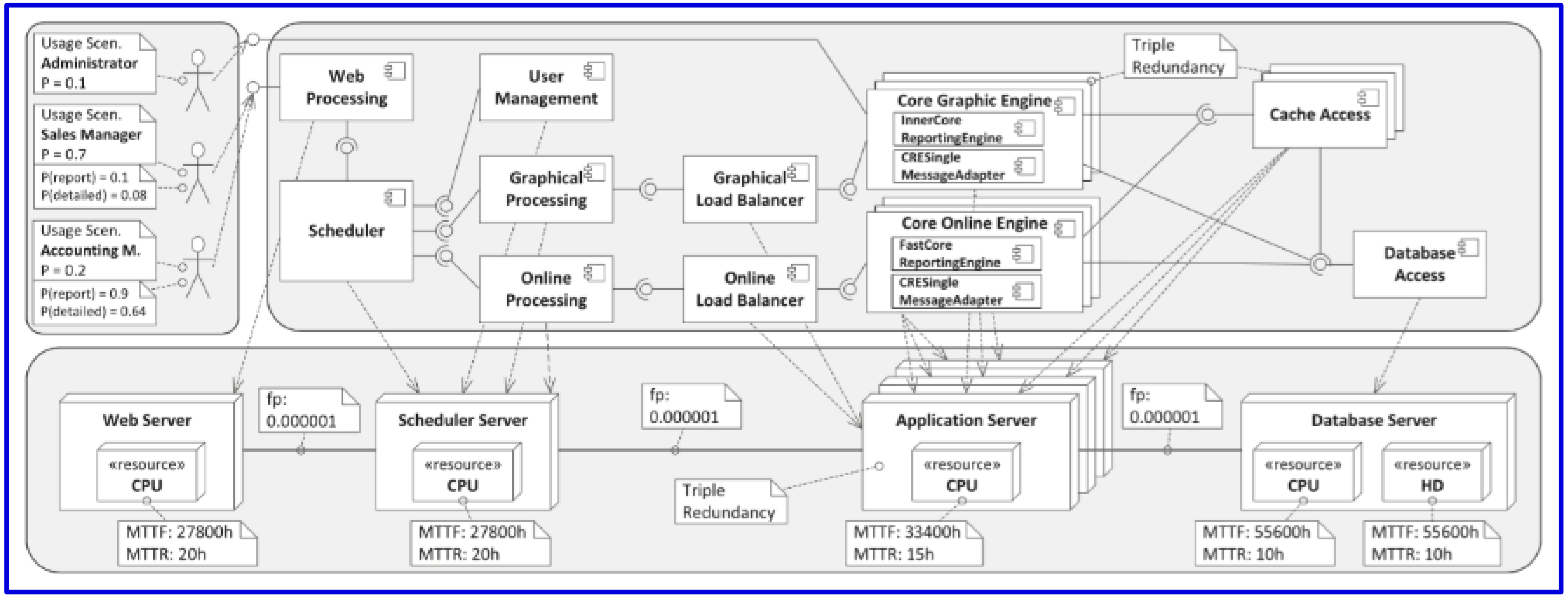
SOFTWARE ARCHITECTURE

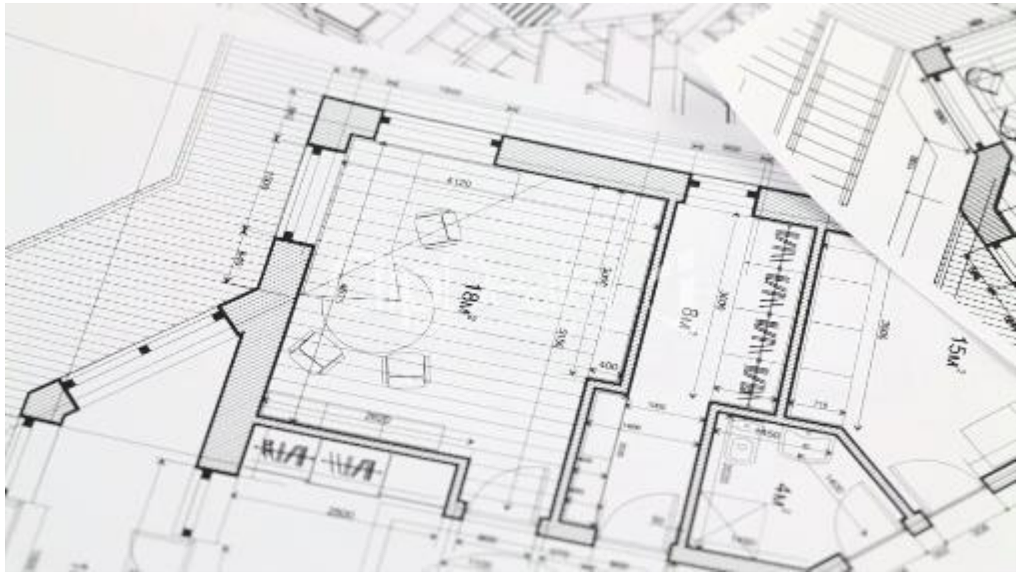


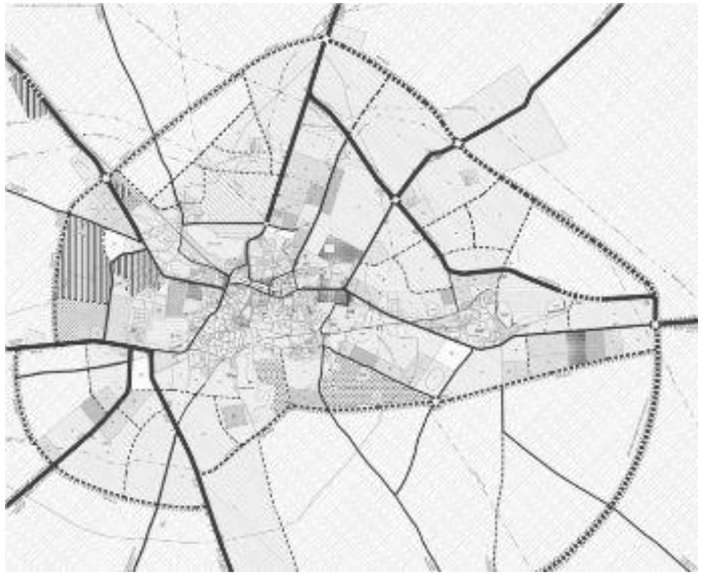
SOFTWARE ARCHITECTURE

SOFTWARE ARCHITECTURE

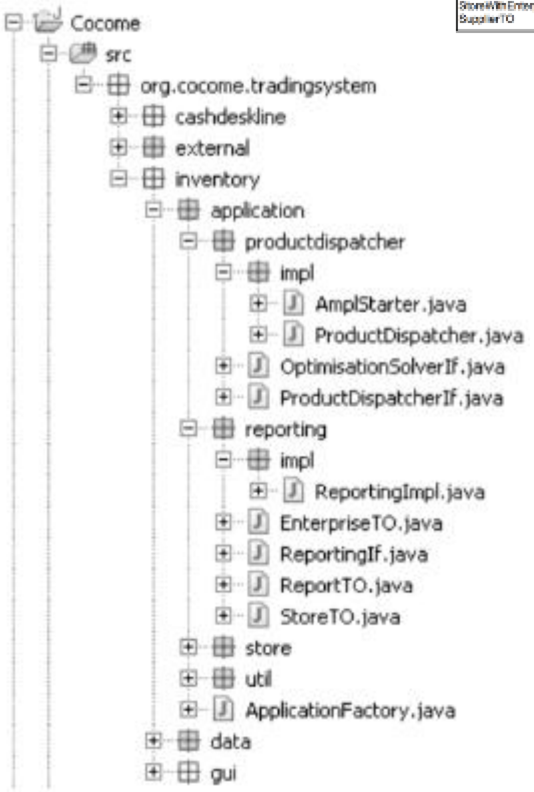
Software Architecture



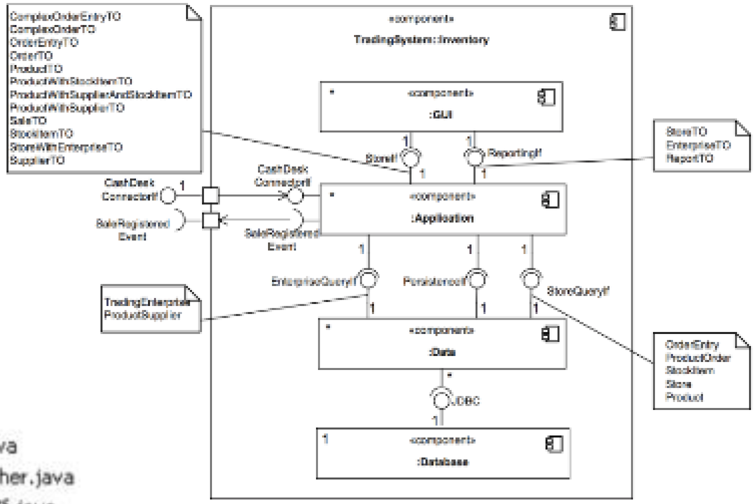


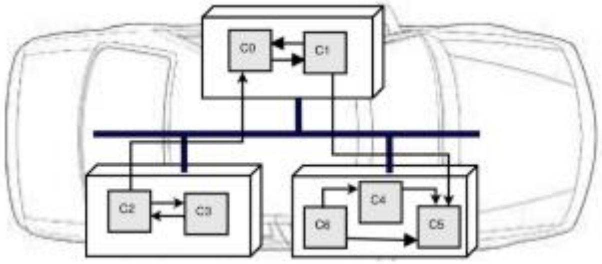


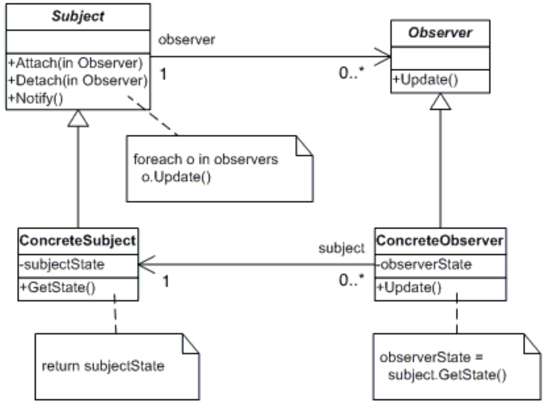
Where do we find SA?

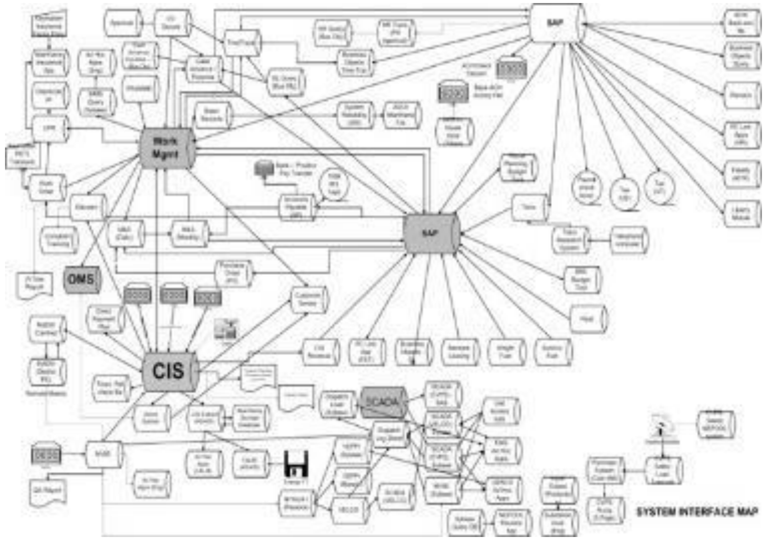


```
Cocome
  src
    org.cocome.tradingsystem
      cashdeskline
      external
      inventory
        application
          productdispatcher
            impl
              AmplStarter.java
              ProductDispatcher.java
            OptimisationSolverIf.java
            ProductDispatcherIf.java
          reporting
            impl
              ReportingImpl.java
            EnterpriseTO.java
            ReportingIf.java
            ReportTO.java
            StoreTO.java
          store
          util
          ApplicationFactory.java
        data
        gui
```









HOWEVER, ARCHITECTURE IS NOT ITS BLUEPRINT

What is then a SW Architecture?

Till 2000

- Software architecture refers to **the fundamental structures** of a software system… [IEEE 1471:2000]

Since 2000

- Software architecture encompasses **the set of significant design decisions** that shapes a software system… [RUP, 1998]

What is then a SW Architecture?

- The architecture is **the set of significant design decisions** that shape a software system, where significant is measured by cost of change. [Grady Booch, 2006]

- Expert developers' **shared understanding of the system design**.
- The decisions that you wish you could get right early. [Martin Fowler, 2015]

Those principles that drive all your future design decisions**.**

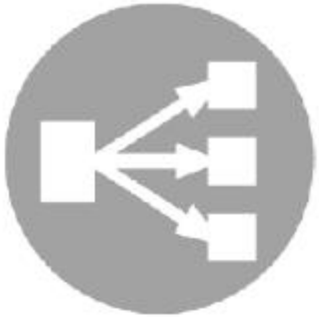
Quality Criteria

- **Reliability** – The probability of correct/failure-free system operation.
- **Performance** – The degree to which a system meets its requirements for timeliness, i.e. response time or throughput.
- **Security** – The ability of a system to prevent unauthorized access and protect the confidentiality, integrity and availability of data.
- **Safety** – The ability of a system to operate without the danger of causing serious harm (e.g. human injury).
- **Robustness** – Degree to which a system is able to withstand an unexpected event without quality degradation.
- **Resilience** – The ability of a system to recover quickly after a disaster.

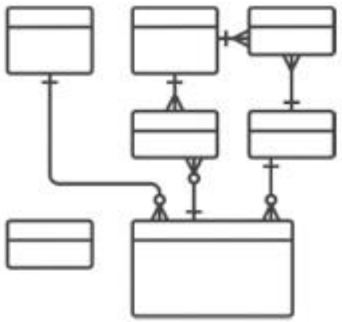
Dimensions and Guidelines

Quality Criteria

Architectural Tactics



Architectural Patterns



Reference Architectures

Technologies

Methods and processes

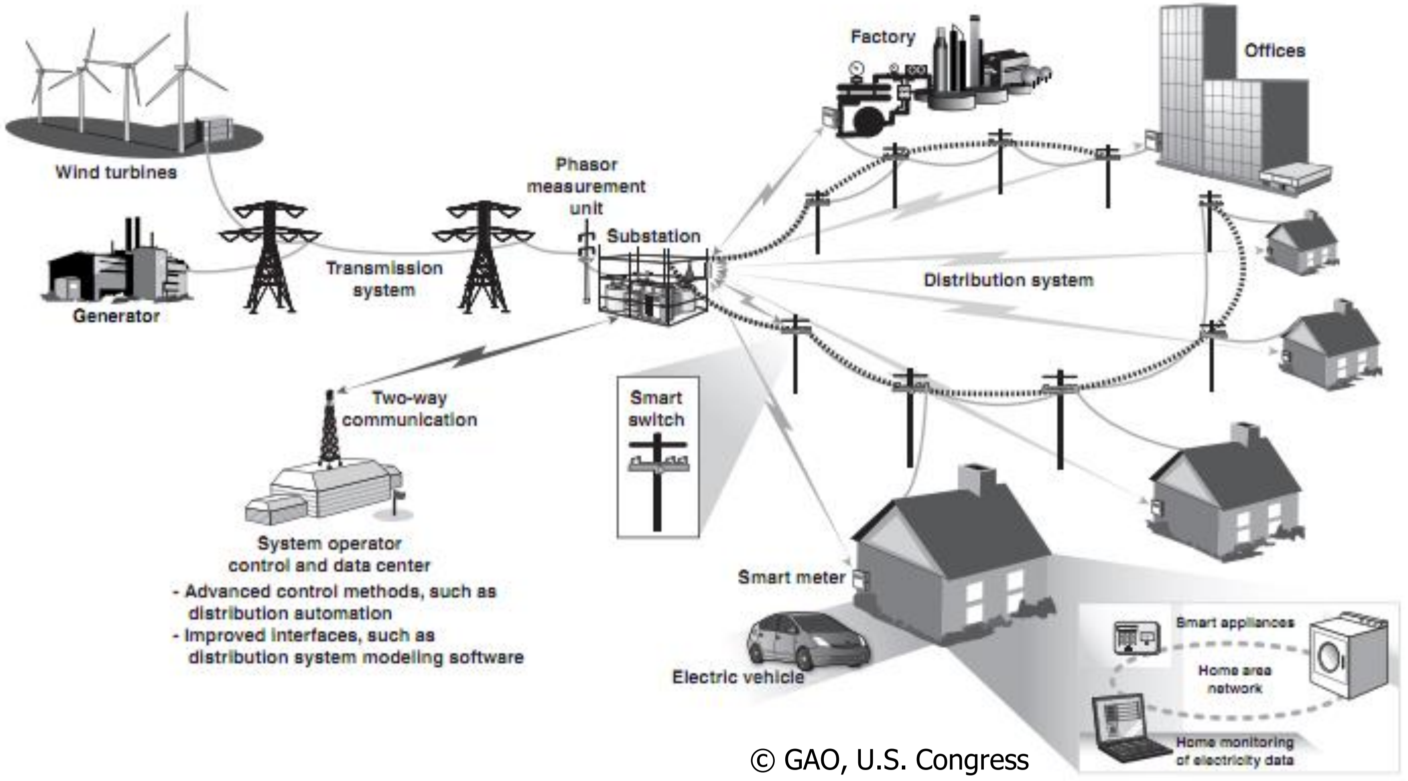
/ Pictures from Awais Rashid, University of Bristol

WHAT MAKES ARCHITECTING DIFFICULT?

Digitalization meets Critical Infrastructures

What makes these infrastructures critical?

- The cyber and physical space merged into one
- If we stayed all digital, not much would be in danger, but we go into remote control of everything


Wind turbines
Generator
Transmission system
Phasor measurement unit
Substation
Factory
Offices
Distribution system
Two-way communication
Smart switch
System operator control and data center
- Advanced control methods, such as distribution automation
- Improved interfaces, such as distribution system modeling software
Smart meter
Electric vehicle
Smart appliances
Home area network
Home monitoring of electricity data

© GAO, U.S. Congress

Context-related Challenges

- Hyperconnected world and business landscape, problem cascading, unpredictable impacts
- Uncertainty about the trustability of connected devices
- Highly distributed environment, entry points to secure, data inconsistency, unreliable sensors, partial failures
- Securing against threats that are not existing yet

Engineering for the Unknown

It is no longer enough to engineer systems for **problem avoidance**

- We need to anticipate **intentional & unintentional** problems on all levels

Prebuilt mechanisms for:

- recognizing an attack/fault,
- stopping it from propagating,
- ensuring safety under attack/fault,
- recovering from an attack/failure,
- forensics after the attack/failure

Engineering for the Unknown

It is no longer enough to engineer systems for **problem avoidance**

- We need to anticipate **intentional & unintentional** problems on all levels

Prebuilt mechanisms for:

Detection of insider attacks in organizations

- recognizing an attack/fault,
- stopping it from propagating,
- ensuring safety under attack/fault,
- recovering from an attack/failure,
- forensics after the attack/failure

Engineering for the Unknown

It is no longer enough to engineer systems for **problem avoidance**

- We need to anticipate **intentional & unintentional** problems on all levels

Prebuilt mechanisms for:

- recognizing an attack/fault,
- stopping it from propagating,
- ensuring safety under attack/fault,
- recovering from an attack/failure,
- forensics after the attack/failure

Forensic-ready software systems

Engineering for the Unknown

It is no longer enough to engineer systems for **problem avoidance**

- We need to anticipate **intentional & unintentional** problems on all levels

Prebuilt mechanisms for:

- recognizing an attack/fault,
- stopping it from propagating,
- ensuring safety under attack/fault,
- recovering from an attack/failure,
- forensics after the attack/failure

Trust in Autonomous Ecosystems

NEED FOR EXTENSIVE MINDSET STRETCH

Bridging Communities & Thinking out of the Box

- Building trust in digital autonomous ecosystems
  - Technical, methodological, legal, psychological, sociological, environmental, economical and other aspects need to meet in one solution

- References
  - Cioroaica, Emilia, Thomas Kuhn, and Barbora Buhnova. **"(Do not) trust in ecosystems."** In Proceedings of ICSE NIER 2019
  - Cioroaica, Emilia, Barbora Buhnova, Thomas Kuhn, and Daniel Schneider. **”Building Trust in the Untrustable”**. In Proceedings of ICSE SEIS 2020

THANK YOU