Future Computing Architectures

Nishant Saurabh

Utrecht University

Agenda for today

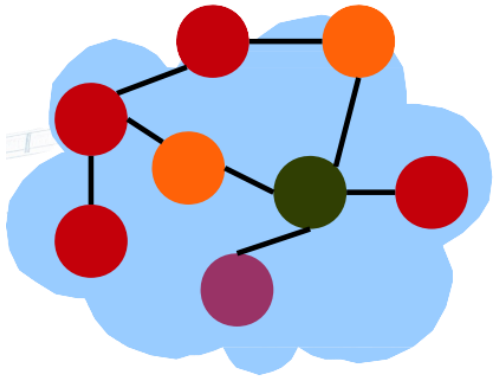
- 09:00 – 09:30: Recap
- 09:30 – 10:30: Lab Assignment-I
- 10:45 – 11:15: Future Computing Architectures
- 11:15 – 12:30: Assignment Time
- 12:30 – 12:45: Wrap up

SOA

Service –oriented architectures (SOAs) are way of developing distributed systems where system components are stand-alone services , executing on geographically distributed servers

Utrecht University

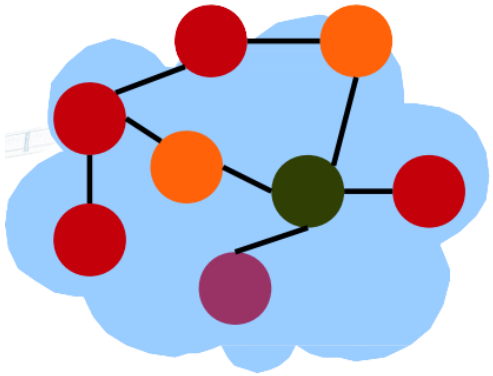
Components of SOA



- Services
  - **Loosely-coupled reusable software component**
  - **Encapsulates discrete functionalities**
  - **Distributed and programmatically accessed**
- Messages
  - **Exchange of information**
- Meta-data
  - **Service description**
  - **Service interface**
  - **Service metadata**

Utrecht University

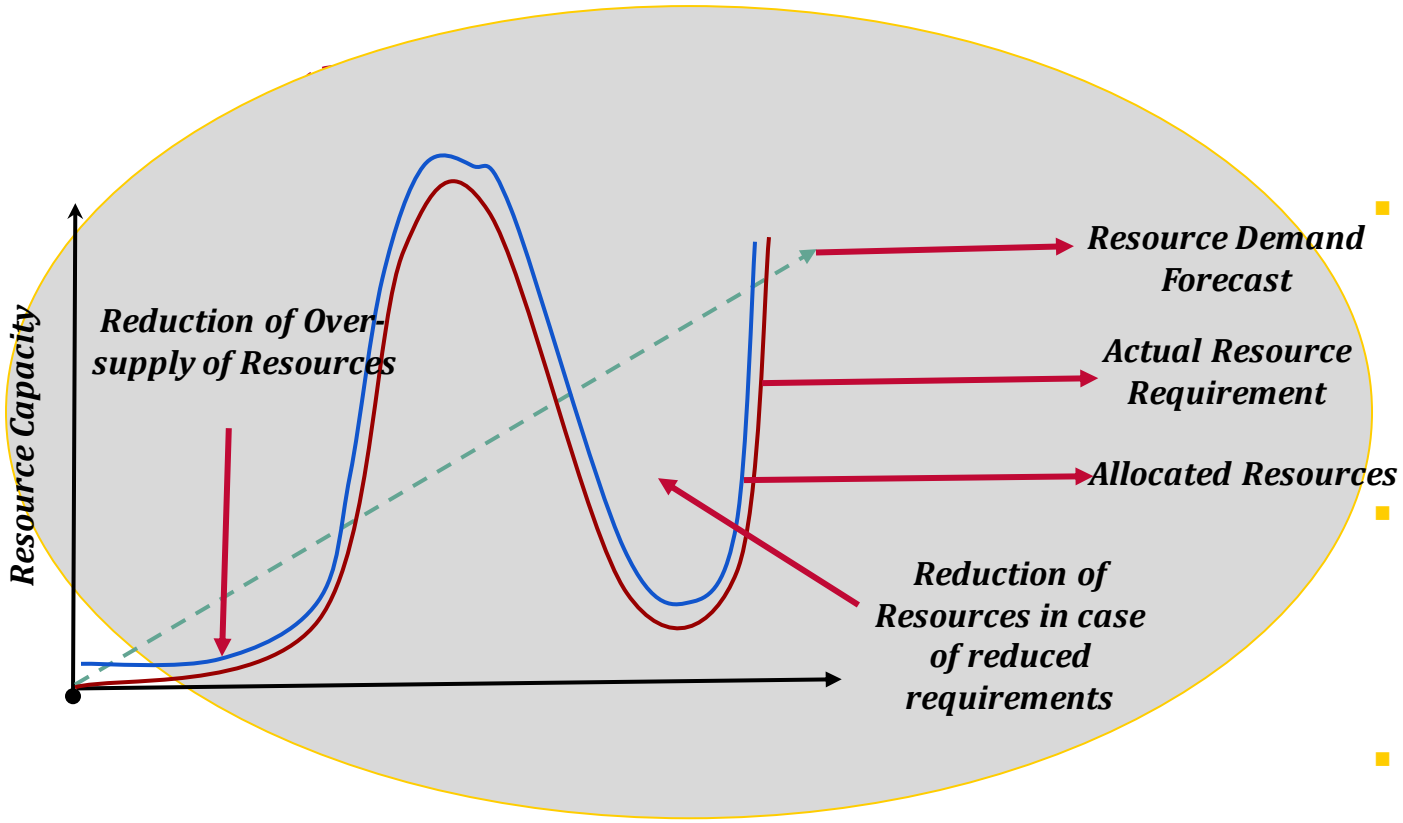
SOA considerations?



- Computational penalties
- Extra layers
- Higher communication latency
- Supporting native applications
- Service granularity (reusability or performance)
- Fault tolerance (partial or full failure)
- Service agreements (performance and cost)
- Governance (how to manage and orchestrate)

Utrecht University

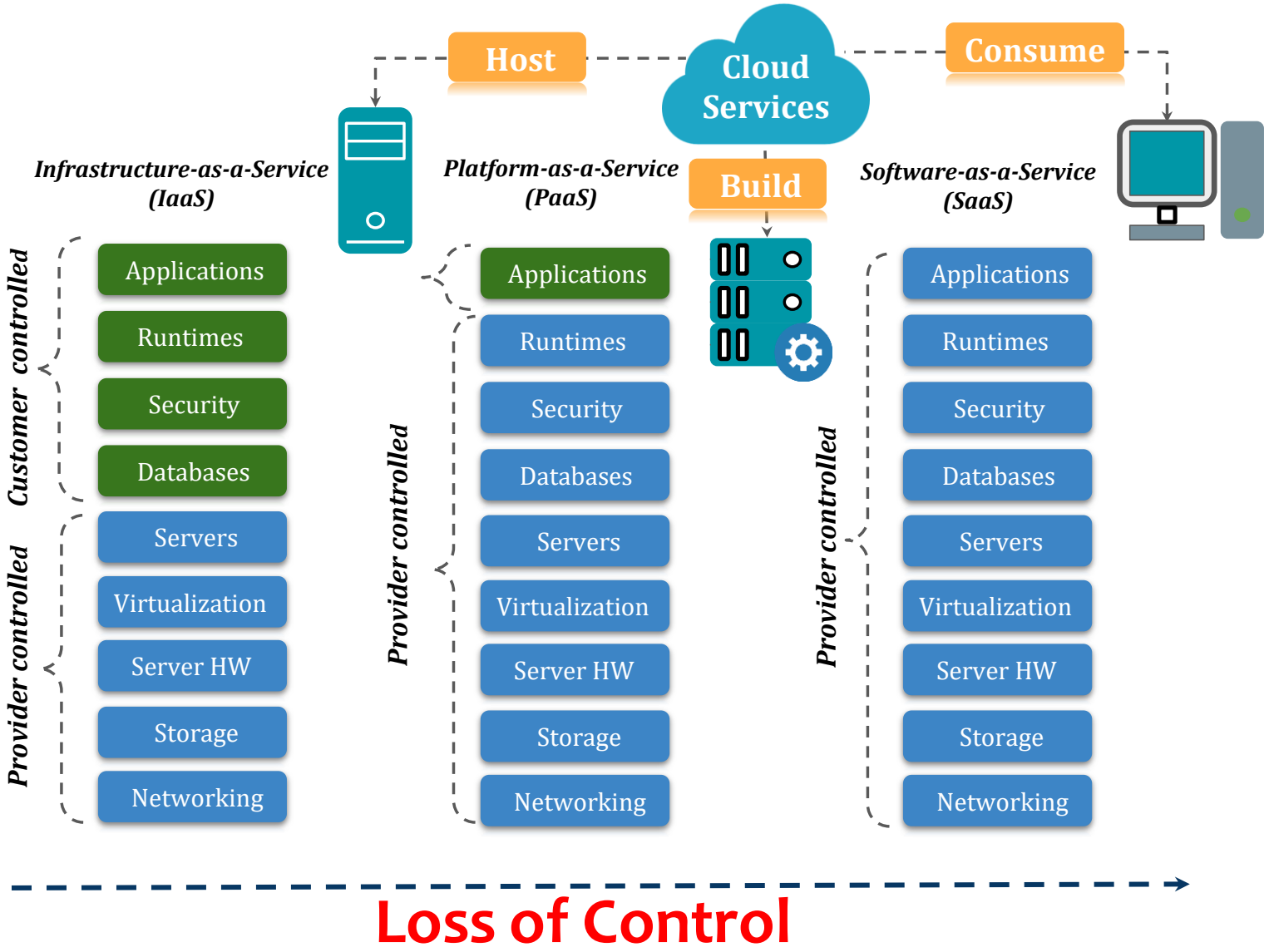
Ubiquitous Cloud



- On-demand self service
  - **Unilaterally provision computing capabilities without requiring human interactions**
- Broad network access
  - **Available over the network**
  - **Heterogenous thin or thick client platforms (mobile phones, laptops)**
- Resource pooling
  - **Compute resources serve multiple users**
  - **Multi-tenant model**
- Metering capability
- Rapid elasticity
  - **Scale out**
  - **Scale in**

Utrecht University

Cloud service models



- Software-as-a-Service
- Platform-as-a-Service
- Infrastructure-as-a-Service

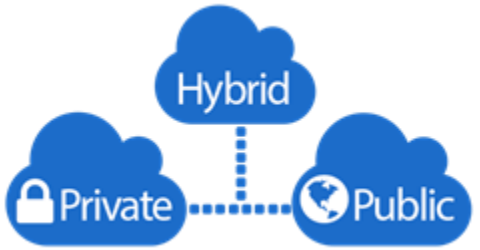
Utrecht University

Cloud deployment models

- Public Cloud
  - **Available to all**
  - **Example: AWS, GCP**

- Private Cloud
  - **On-premise**
  - **Example: OpenStack**

- Hybrid Cloud
  - **Private + Public**
  - **E.g. Rackspace Cloud**



- Community Cloud
  - **Shared by organisations with similar goals**
  - **Example: salesforce**

Utrecht University

Virtualization

- Create a software-based-or virtual-representation of applications, servers, storage and networks to reduce IT expenses while boosting efficiency and agility.

- Hardware-level virtualization
  Virtual Machines (VMs)

- OS-level virtualization
  Containers (Dockers, Lxc containers)

- Serverless
  function based invocations

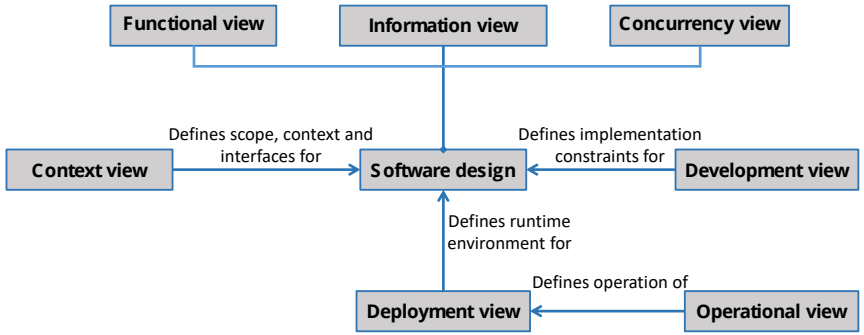
Utrecht University

Architecting in a Cloud Environment

- Quality attributes that are different in a cloud
  - **Security**
  - **Performance**
  - **Availability**

Utrecht University

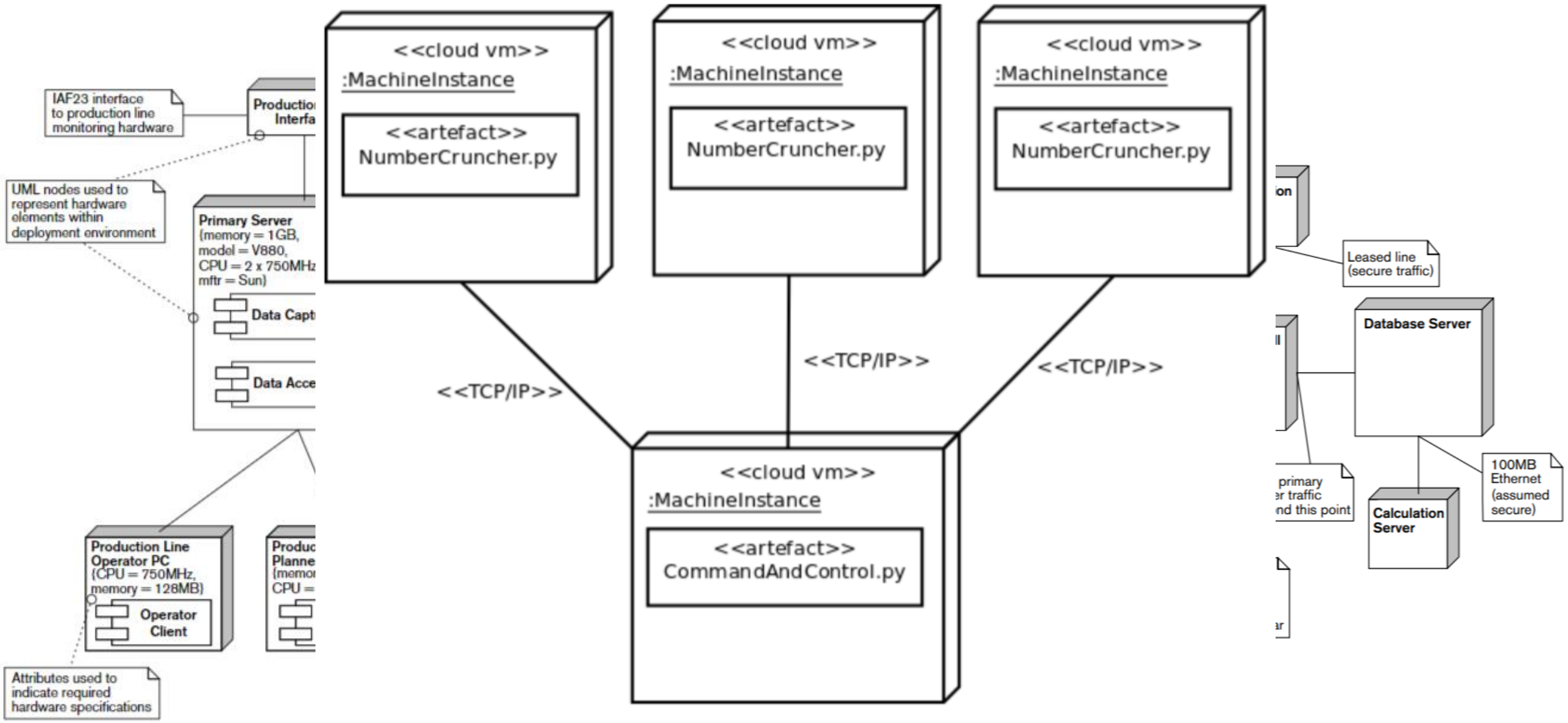
Deployment view



- Deployment view:
  - **Describes the environment into which the system will be deployed, and the dependencies that the system has on elements of it**
- Concerns
  - **Runtime platform,**
  - **Specification of hardware or hosting, network requirements**
  - **Physical constraints**
- Models and views
  - **Runtime platform models,**
  - **Network models**
  - **Technology-dependent models**
- Problems and pitfalls
  - **Unproven technology,**
  - **Unsuitable / missing Service Level agreements**
  - **Ignoring inter-site complexities**
  - **Disaster recovery environment**

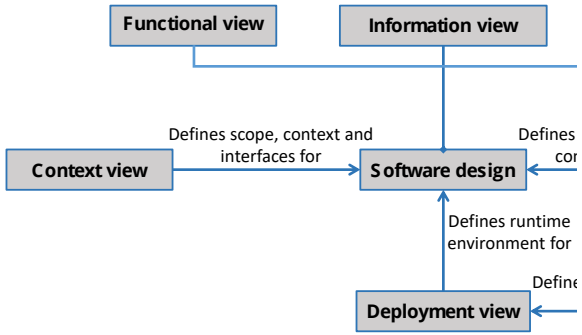
Utrecht University

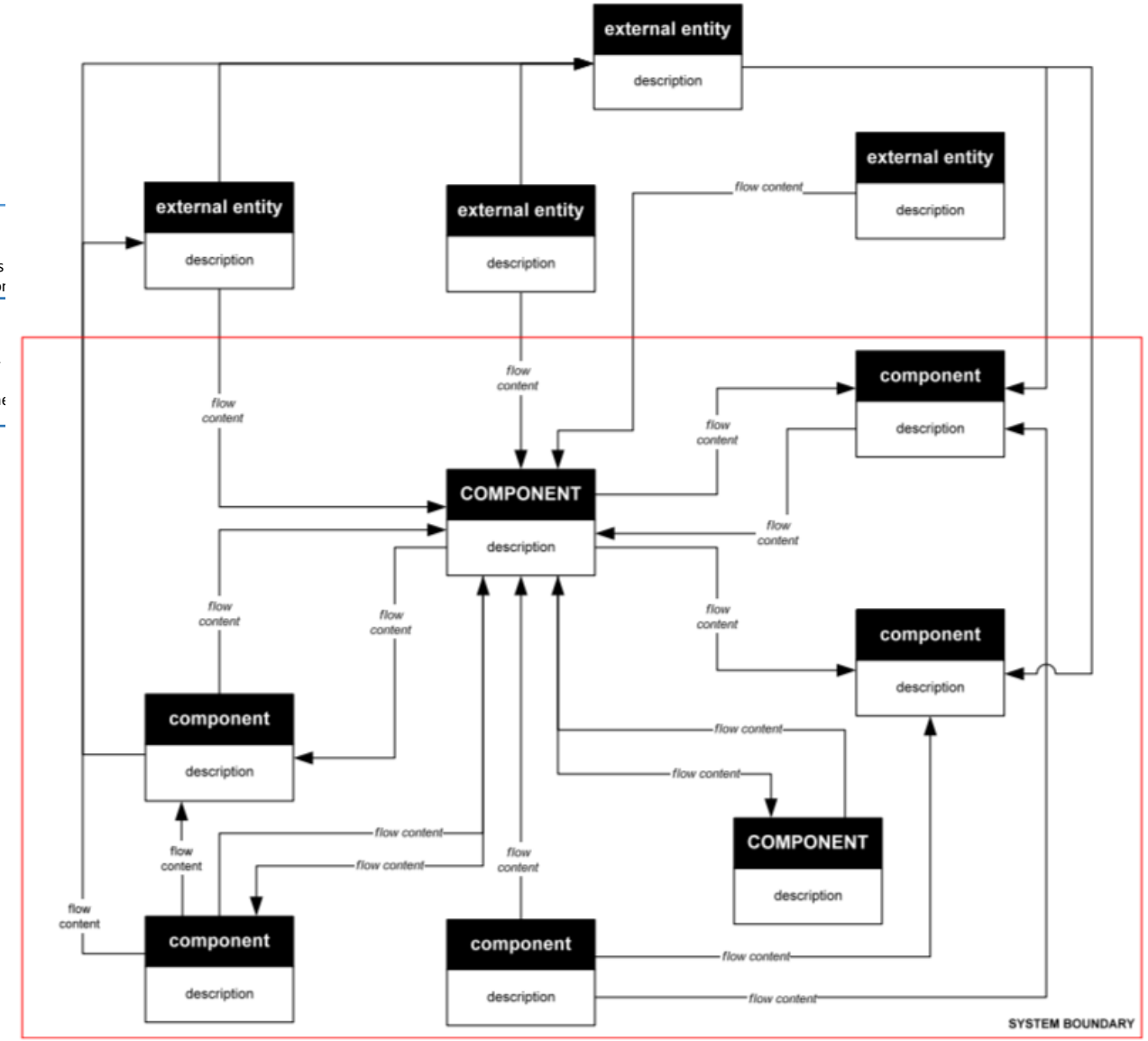
Deployment View: Runtime and Network Models



Utrecht University

Information view





, manipulates, manages

latency
uality

els (BPMN, Petri nets, ...)
ship models, ...

complexity,
distributed DB

Multiple concurrent updaters ...

Utrecht University

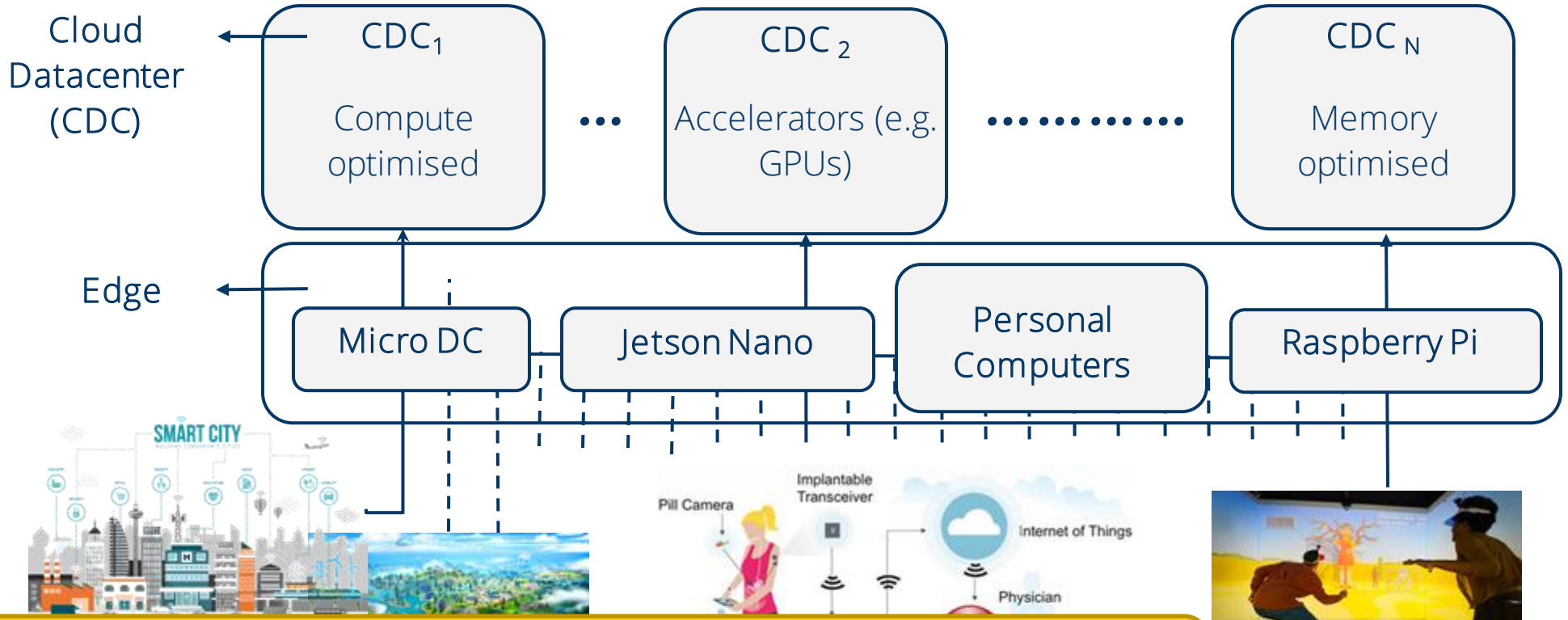
Agenda for today

- 09:00 – 09:30: Recap
- 09:30 – 10:30: You work on Lab Assignment-I
- 10:45 – 11:15: Future Computing Architectures
- 11:15 – 12:30: Assignment Time
- 12:30 – 12:45: Wrap up

Utrecht University

Edge Architecture

Cloud versus Edge



Fast Data Processing: Low Latency and Response time

Typical Concerns: Mobile Edge Systems: Energy Usage, Network Connectivity, Constrained Resources

Utrecht University

Quantum Edge Model

- Classical resources

  Based on von Neumann architectures: typically Cloud, Edge resources with shared memory

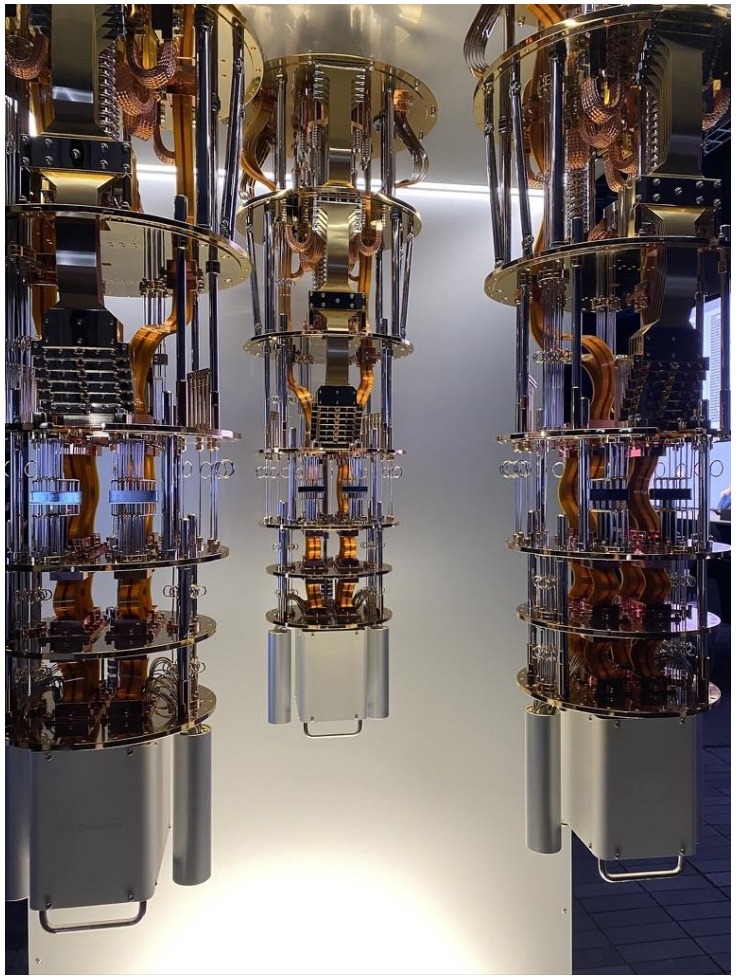
- Quantum resources

  Superposition and entanglement

  NISQ: Noisy Intermediate Scale Quantum Computers

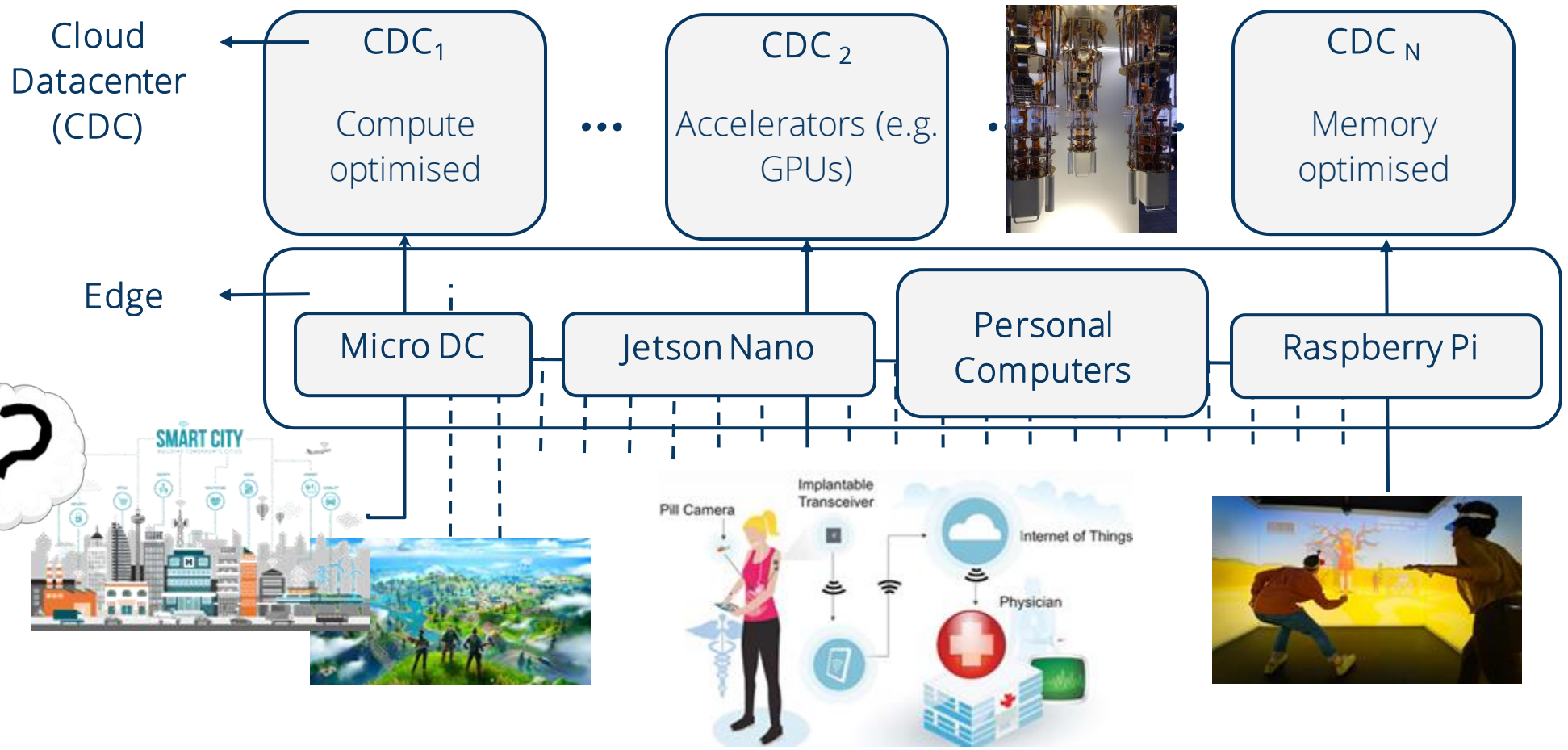
  FTQC: Fault-tolerant Quantum Computers

Characteristics	Classical Computing	Quantum Computing
Computing units	Calculates using bit levels 0 and 1	Calculates with Qubits, represents 0 and 1 simultaneously
Computing Capacity	Capability increased linearly with number of transistors	Capability increases exponentially with Qubits
Error rates	Low error rates, operate at room temperature	High error rate, some quantum systems need to be ultra cold



Edge systems

Utrecht University



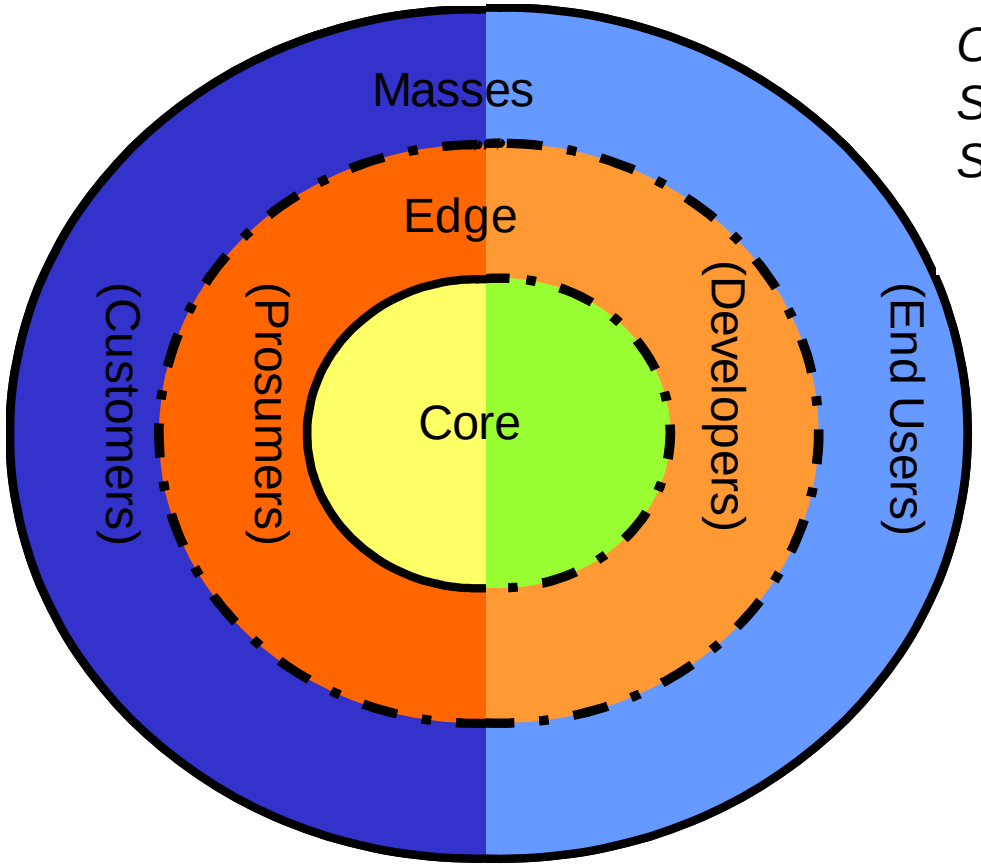
What does an Edge System look like ?

Utrecht University

Structure of Edge systems

Open Content Systems

Open Source Software

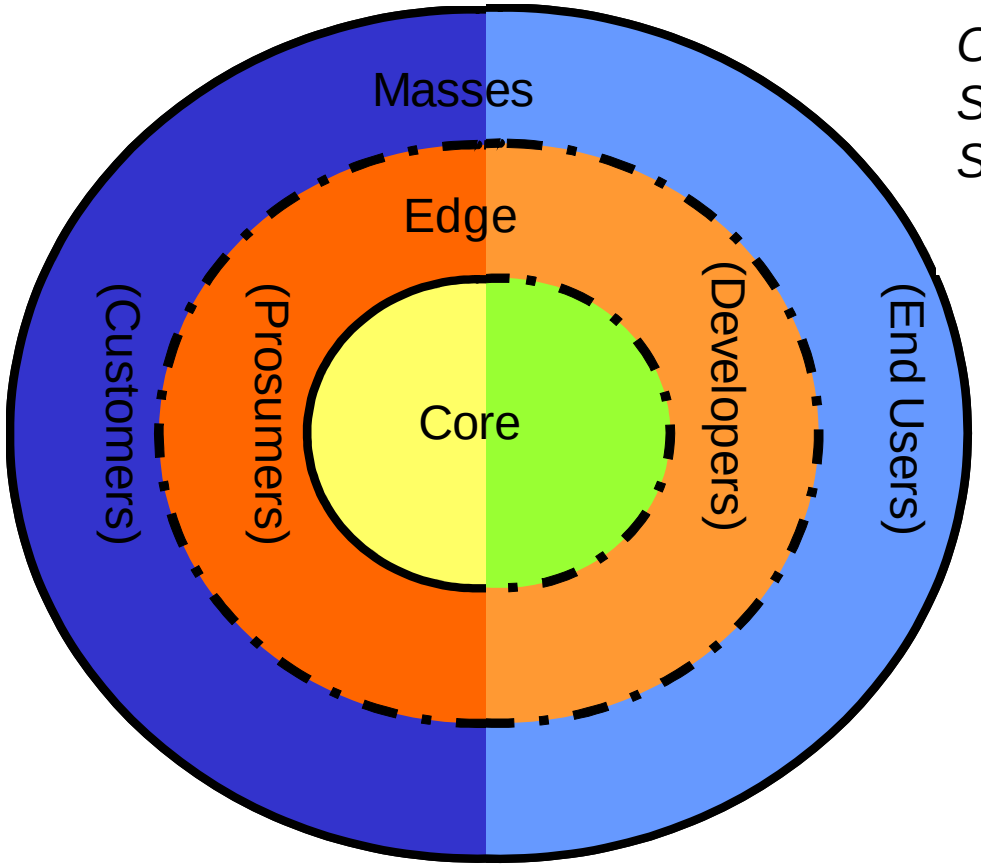


Edge is an open source and decentralised system

Utrecht University

Structure of Edge systems

Open Content Systems



Open Source Software

Edge is an open source and decentralised system

Edge systems need input from its users.

Utrecht University

Structure of Edge systems

- Stakeholders
  - **End users and customers**
  - **Prosumers**
  - **Developers**

Utrecht University

Structure of Edge systems

- Stakeholders
  - **End users and customers**
  - **Prosumers**
  - **Developers**
- Roles
  - **Masses create requirements**
  - **Prosumers and developers create value**
  - **Core provides a set of services**

Utrecht University

Structure of Edge systems

- Stakeholders
  - **End users and customers**
  - **Prosumers**
  - **Developers**
- Roles
  - **Masses create requirements**
  - **Prosumers and developers create value**
  - **Core provides a set of services**

Edge is an open source and decentralised system ?

Roles are interchangeable?

Utrecht University

Is CORE permeable?

Utrecht University

How should we architect the core?

Utrecht University

Edge systems considerations

- Requirements
  - **Unknown ?**
  - **Desires of a community**
- System development
  - **Incremental changes**
  - **Context-aware development**
- Resources
  - **Grow with use (i.e. resources at every level)**
- Manageability
  - **Developers are volunteers**

Edge is an open source and decentralised system ?

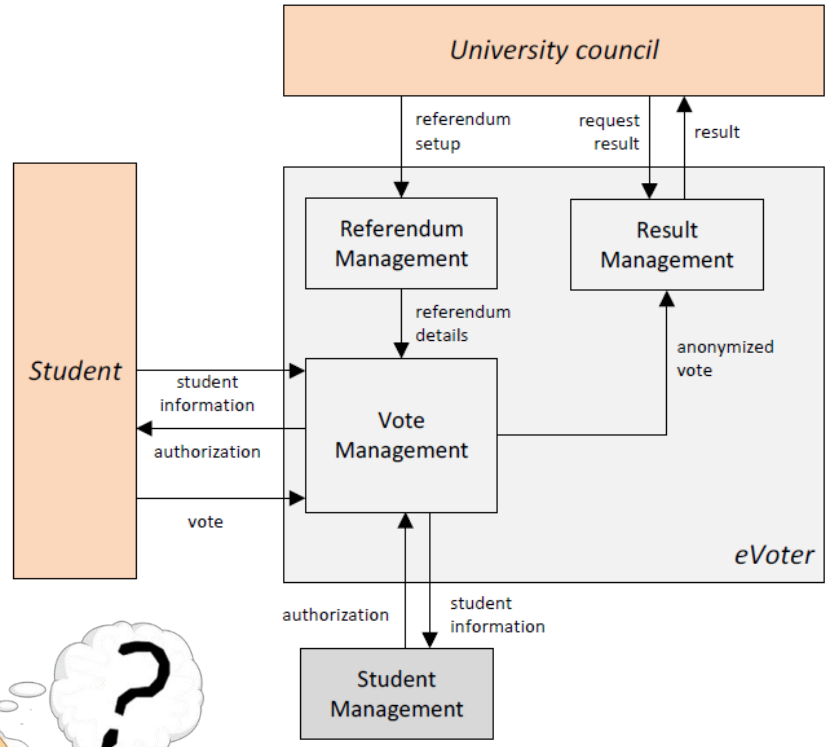
Utrecht University

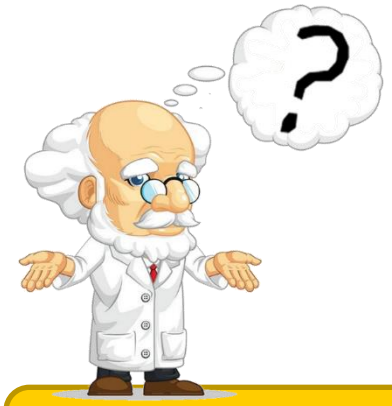
Edge architecture

- Core
  Kernel of the system
- Periphery
  Services built on top of core

Utrecht University

Edge architecture



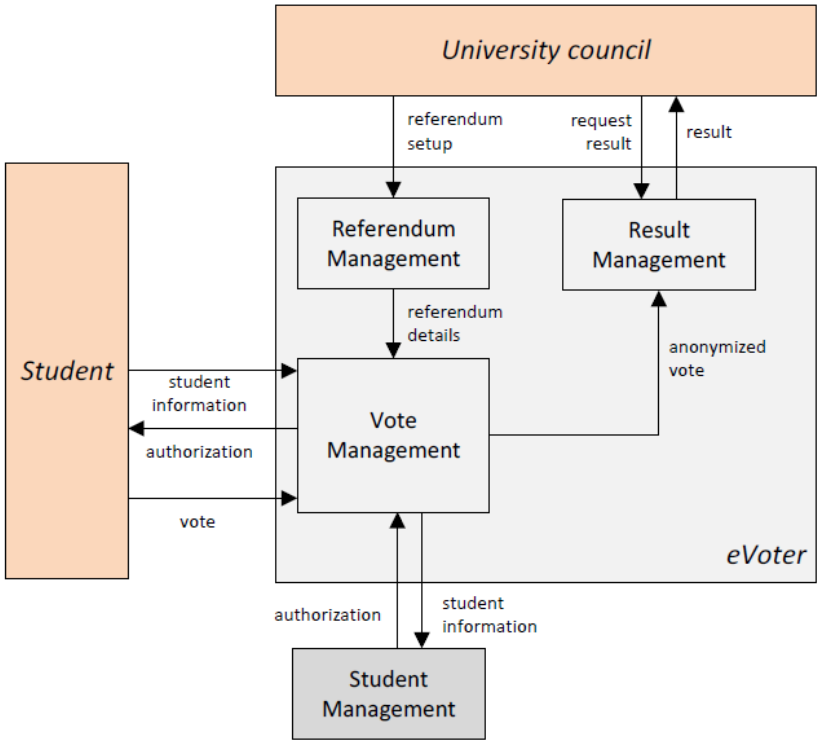


- Core
  Kernel of the system
- Periphery
  Services built on top of core

What is Core in your assignment?

Utrecht University

Edge architecture

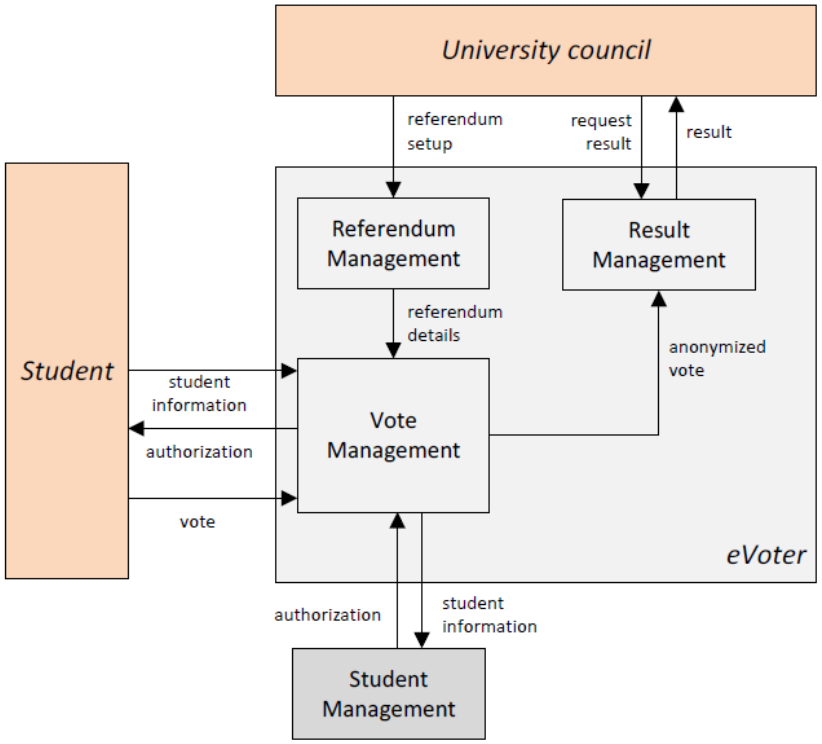


- Core:
  - **Quality attributes for periphery**
  - **Should be modular and multilayered**
  - **Highly reliable**
  - **Robust to errors**

Core is the backbone of Edge system!

Utrecht University

Edge architecture



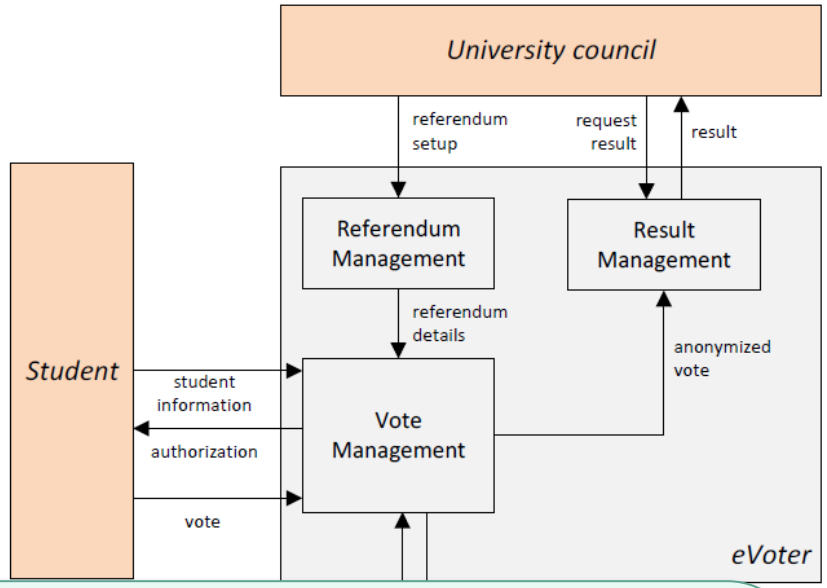
- Core:
  - **Quality attributes for periphery**
  - **Should be modular and multilayered**
  - **Highly reliable**
  - **Robust to errors**

Core is the backbone of Edge system!

A stable Core provides services to peripheral developers

Utrecht University

Edge architecture



- Core:
  - **Quality attributes for periphery**
  - **Should be modular and multilayered**
  - **Highly reliable**
  - **Robust to errors**

Conditions

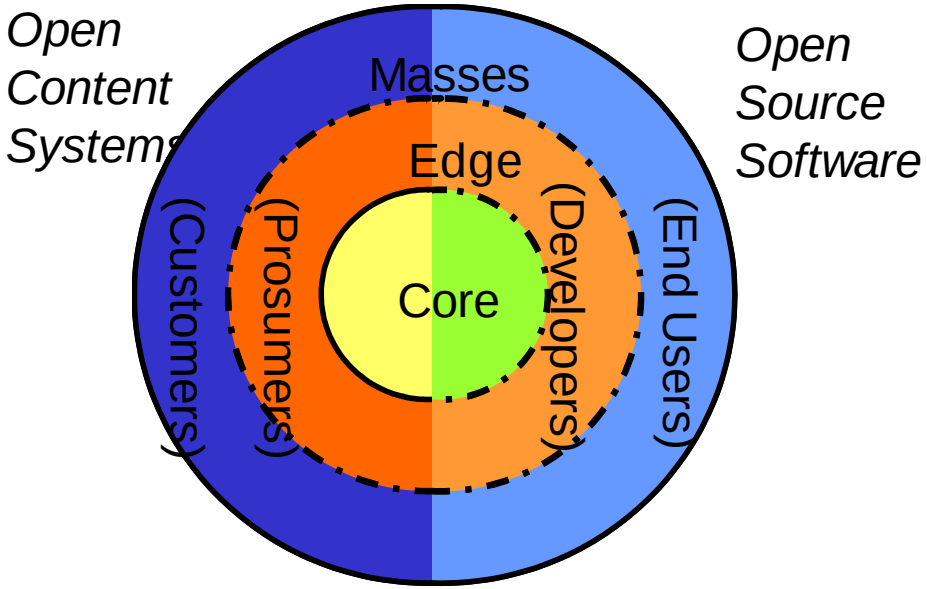
- Organized and up-to-date documentation
- Core services should be easily discoverable
- Appropriate monitoring
- Adequate response to peripheral requests

Core is the backbone of Edge system!

A stable Core provides services to peripheral developers

Utrecht University

Edge perspectives



- Managerial attention on inclusion of periphery for system development
- Crowd governance, Core versus Periphery trade-off analysis
  Proactive and reactive enforcement models
- Requirement forums for periphery
- Architecture consideration for open systems and open content
- Distributed testing, automated and asynchronous delivery

Utrecht University

Agenda for today

- 09:00 – 09:30: Recap
- 09:30 – 10:30: Lab Assignment-I
- 10:45 – 11:15: Future Computing Architectures
- 11:15 – 12:30: Assignment Time
- 12:30 – 12:45: Wrap up

Utrecht University

For Next Time

- Lab-Session Assignment –I Deadline : 10th March 11:59 CET time

Utrecht University

DISCLAIMER

The information in this presentation has been compiled with the utmost care, but no rights can be derived from its contents.