



Software Architecture Introduction

Jan Martijn van der Werf, Claudio Di Ciccio, Nishant Saurabh



Utrecht University

About us



Dr. Ir. Jan Martijn van der Werf

Education

Process modelling (BSc)

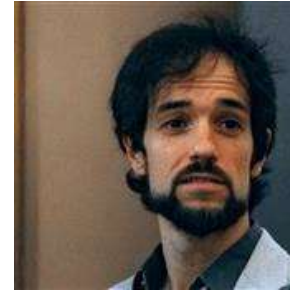
Software architecture (MSc)

Research areas

Modelling & analysis of complex (software) systems

Architecture & process mining

Architecture rationale



Dr. Claudio di Ciccio

Education

Data modelling (BSc)

Process modelling (BSc)

Software architecture (MSc)

Research areas

Formal methods & process mining

Automated reasoning in AI

Blockchain technologies



Dr. Nishant Saurabh

Education

Information security (BSc)

Software architecture (MSc)

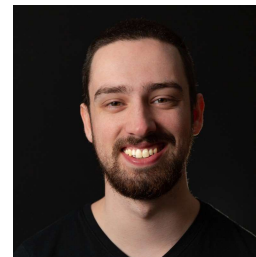
Cloud and edge computing (MSc)

Research areas

Distributed and Hybrid Computing Systems

Workload and Resource Management

Serverless and Runtime Management



Douwe Geurtjens, BSc.

Teaching Assistant for the Course



Utrecht University

About you?



- 1 Go to wooclap.com
- 2 Enter the event code in the top banner

Event code
SAYFBS

 [Copy participation link](#)



Utrecht University

Structure of the course



Utrecht University

Structure of the course



- Lectures on Monday & Thursday
4 hours, blended approach (help us sticking to it! ;-))
- Grading:
 - 30% Oral exam to test your knowledge and skills**
 - 60% Assignment**
 - 10% Lab session assignments (there are 2 of them !)**
 - Mandatory Participation in architecture debates**
 - Mandatory Participation in assignment Peer Review**
- More about the assignment later today!



Utrecht University

Agenda for today

- 09:00 – 09:30: Introduction
- 09:30 – 10:30: DecidArch in groups of 3
- 10:45 – 11:00: recap of DecidArch
- 11:00 – 11:30: Viewpoints
- 11:30 – 12:30: Assignment time
- 12:30 – 12:45: wrap-up





Utrecht University

Let's start!

Imagine the following situation

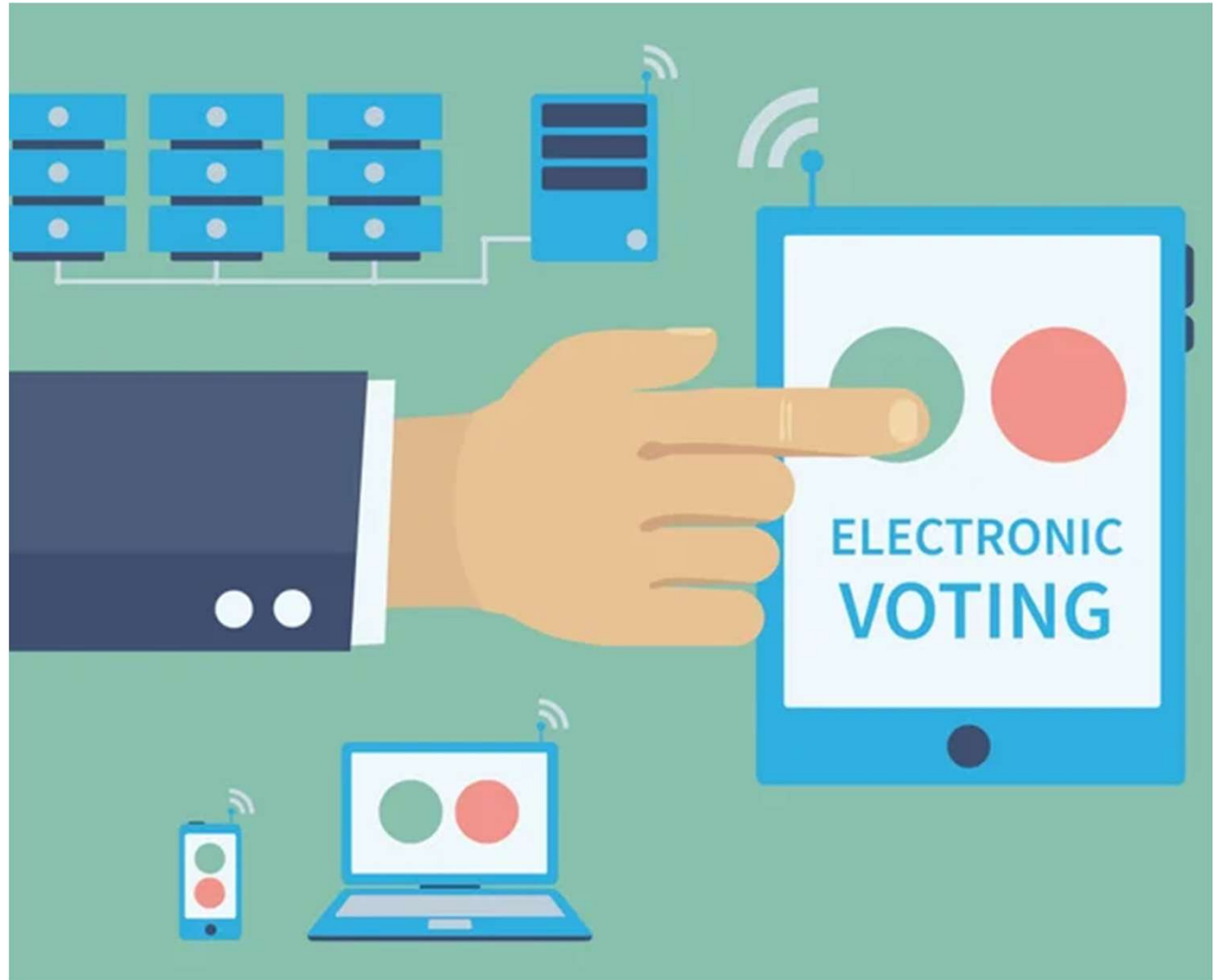
Build me an 'eVoting system'!

What will you build?

How will you start?

Gather requirements?

System design?



Credit: [MIKKO LEMOLA Getty Images](#)

Imagine the following situation

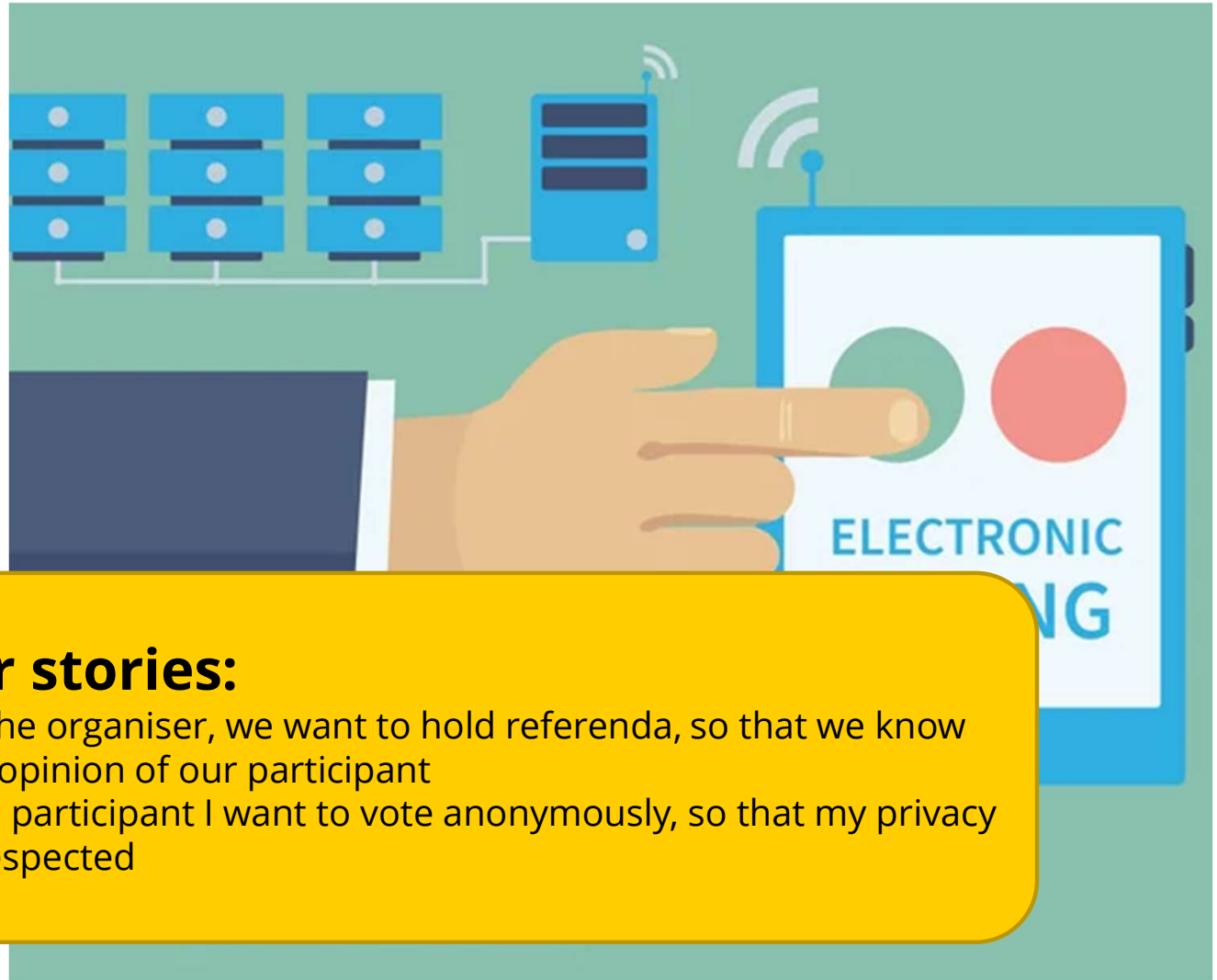
Build me an 'eVoting system'!

What will you build?

How will you start?

Gather requirements?

System design?



User stories:

- As the organiser, we want to hold referenda, so that we know the opinion of our participant
- As a participant I want to vote anonymously, so that my privacy is respected

Credit: [MIKKO LEMOLA Getty Images](#)



Who builds the software system?

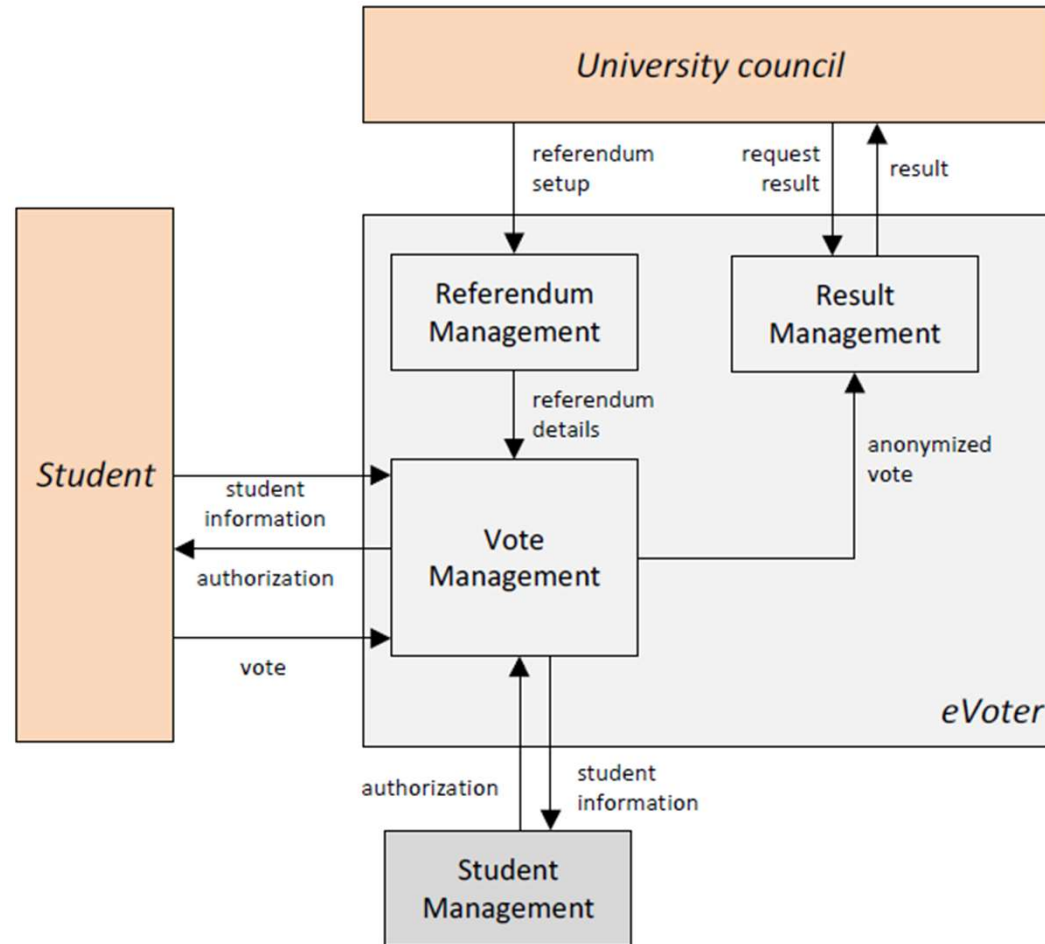


- Is it just you?
- Rough guess: it takes about 30.000 LoC
Developer writing about 100 LoC per day → 300 days!
- So work in a team!
How to communicate?
How to transfer our plans and ideas?
How to divide the work?
Ensure it all fits together?

Make a design!

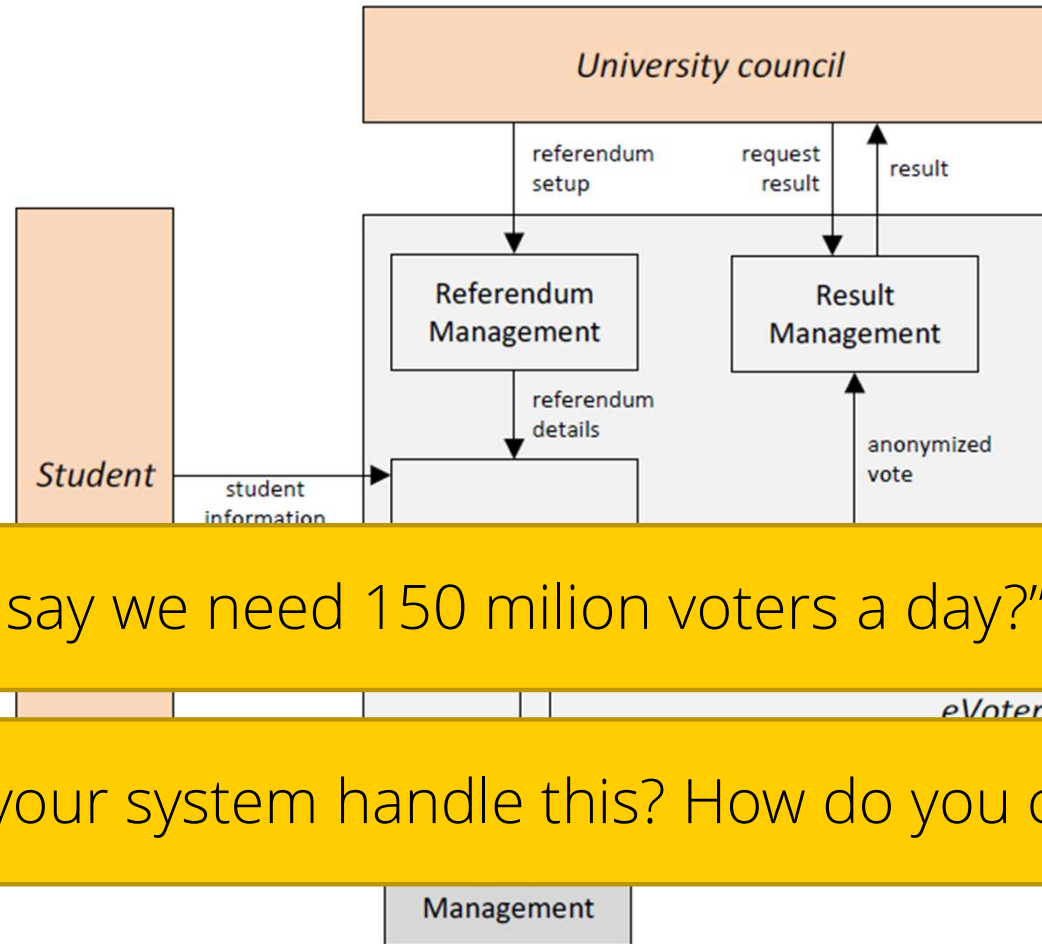


System design





System design



"Oh, did I say we need 150 milion voters a day?"

Can your system handle this? How do you check this?



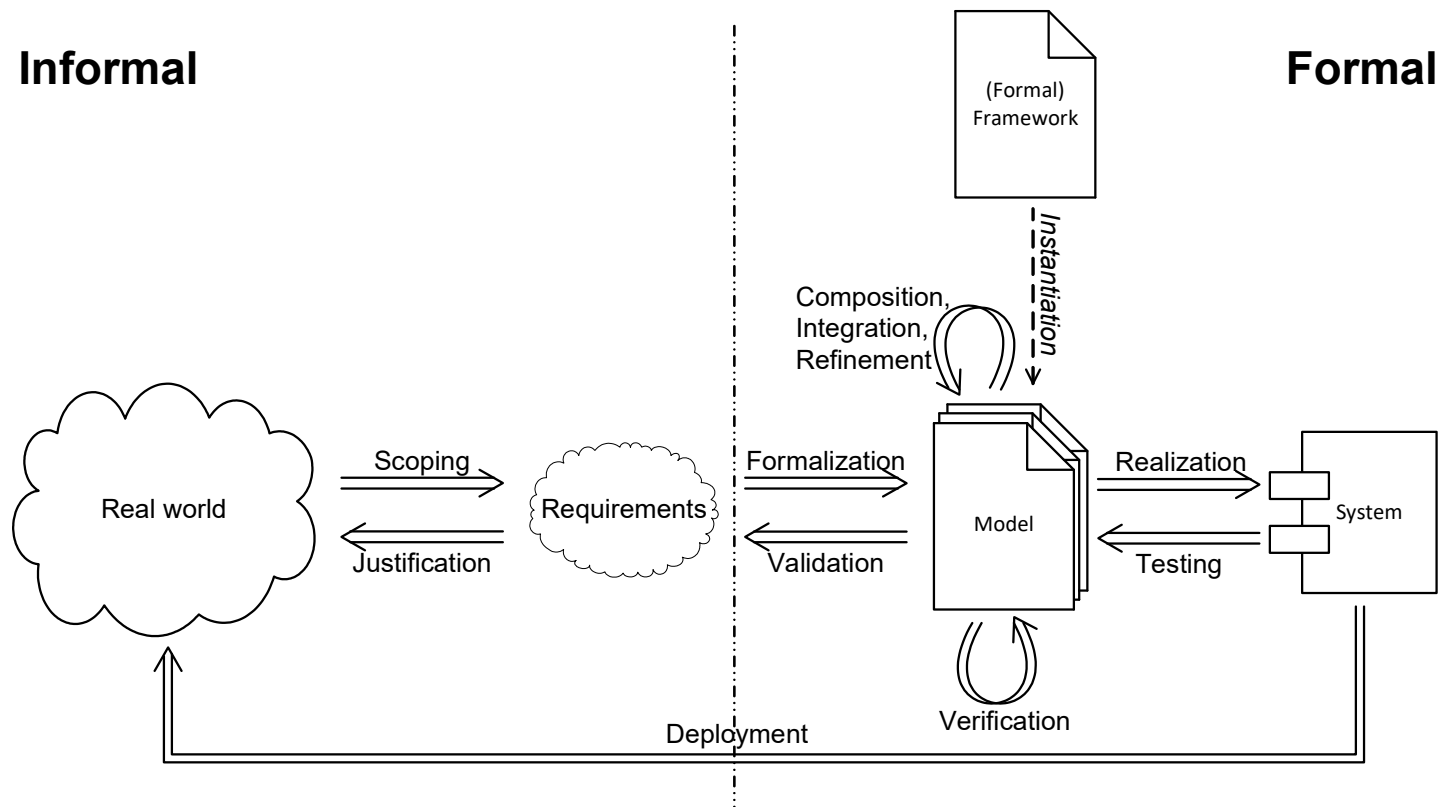
Utrecht University

Software architecture \neq Software design

*In this course, you will meet many (new) modeling notations,
but it is not a course on system design!*

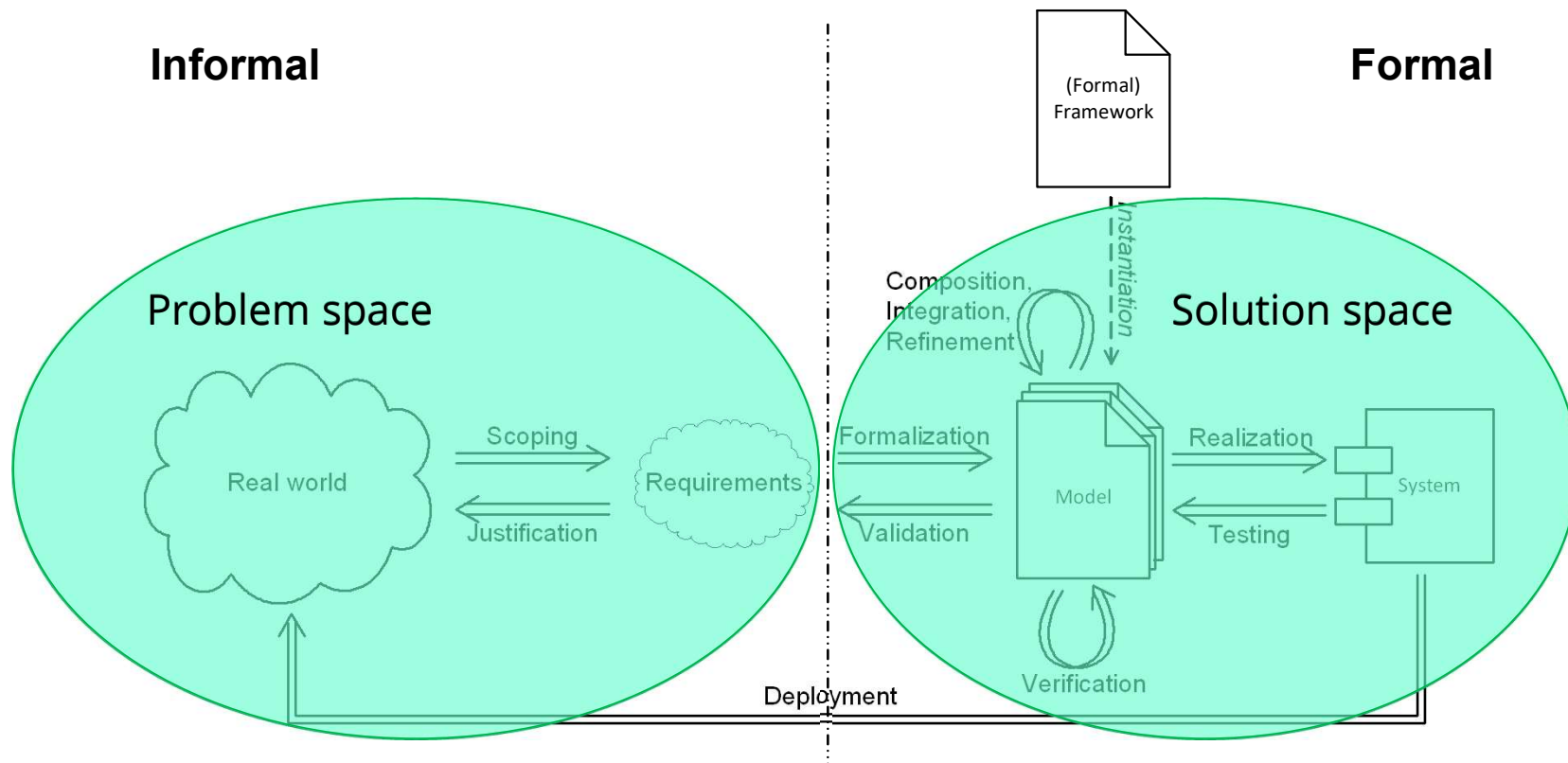


The role of software architecture



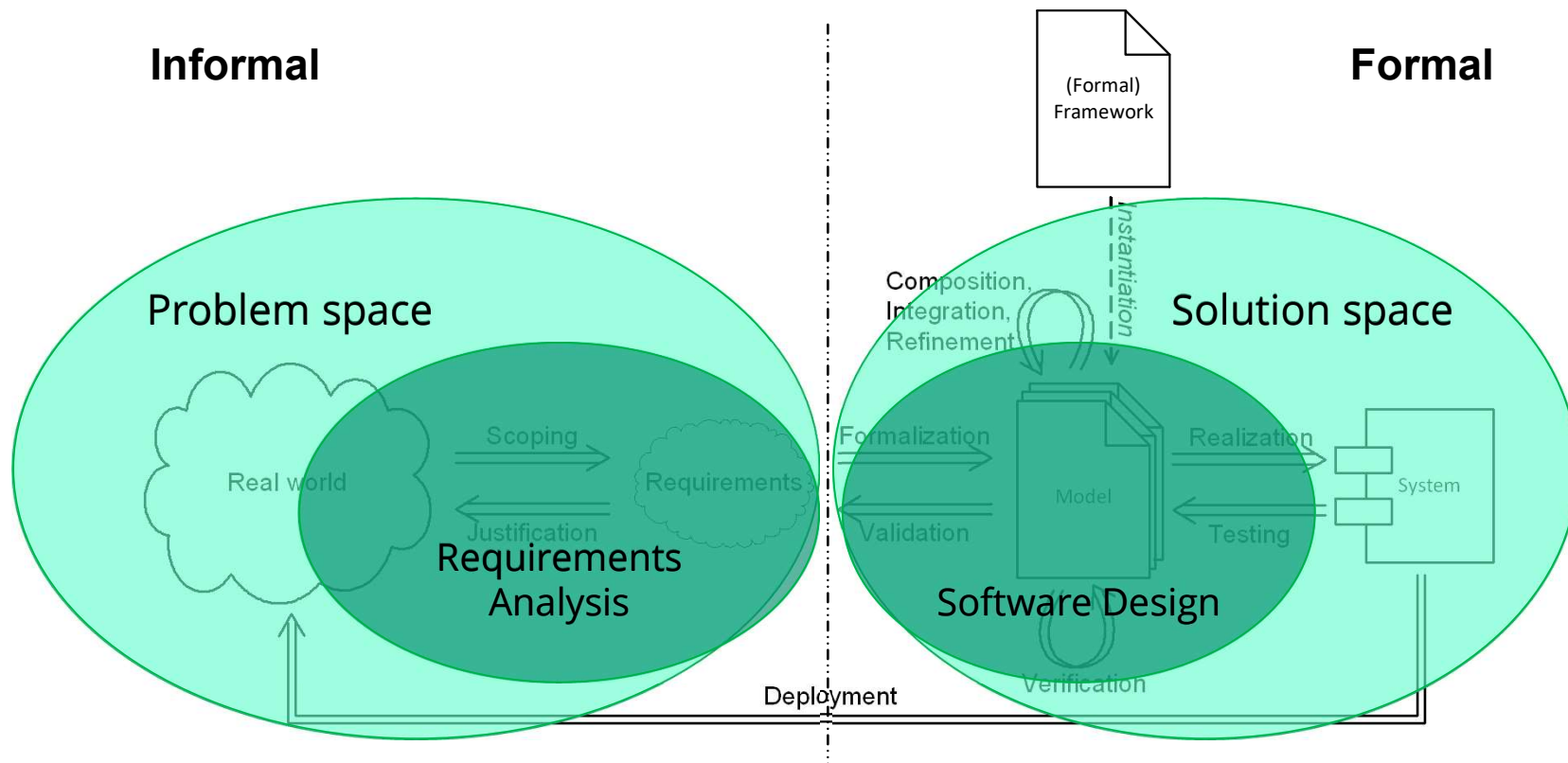


The role of software architecture



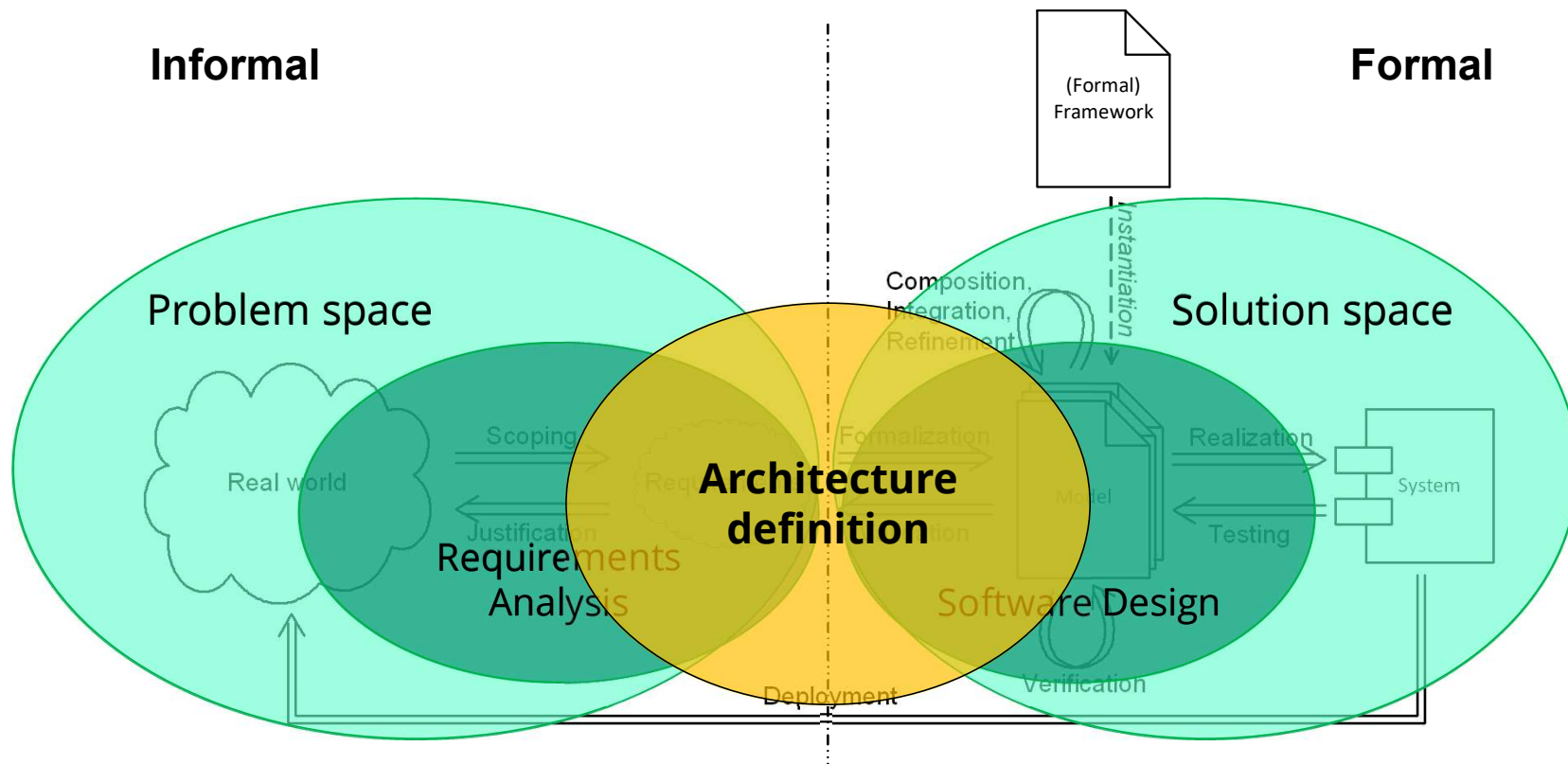


The role of software architecture





The role of software architecture





Utrecht University

Architecture is similar to building a house

- Start at the foundations
- Are the walls strong enough to carry the attics?
- Location of wiring & piping?

But also:

- Architect vs. interior designer
- Architecture is not up to the level of the bricks!





Architecture in practice

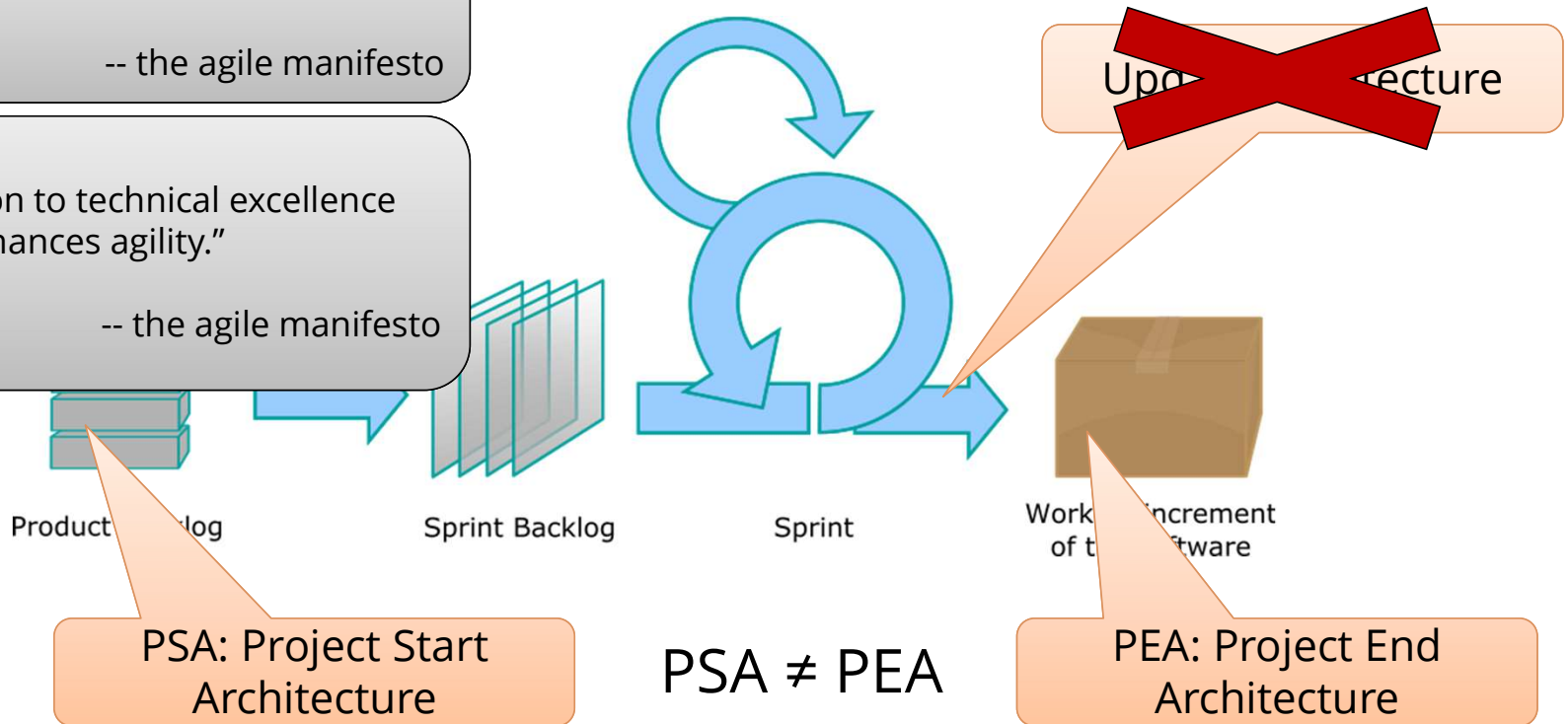
"Working software over comprehensive documentation"

-- the agile manifesto

But also:

"Continuous attention to technical excellence and good design enhances agility."

-- the agile manifesto





Software architecture vs Software design



- *The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both (Clements et al, 2003)*
- Software design:
functional requirements
- Software architecture
Software design
Stakeholders and their concerns
People (not only developers!) management
Trade-off between functional requirements & characteristics



Software architecture vs Software design



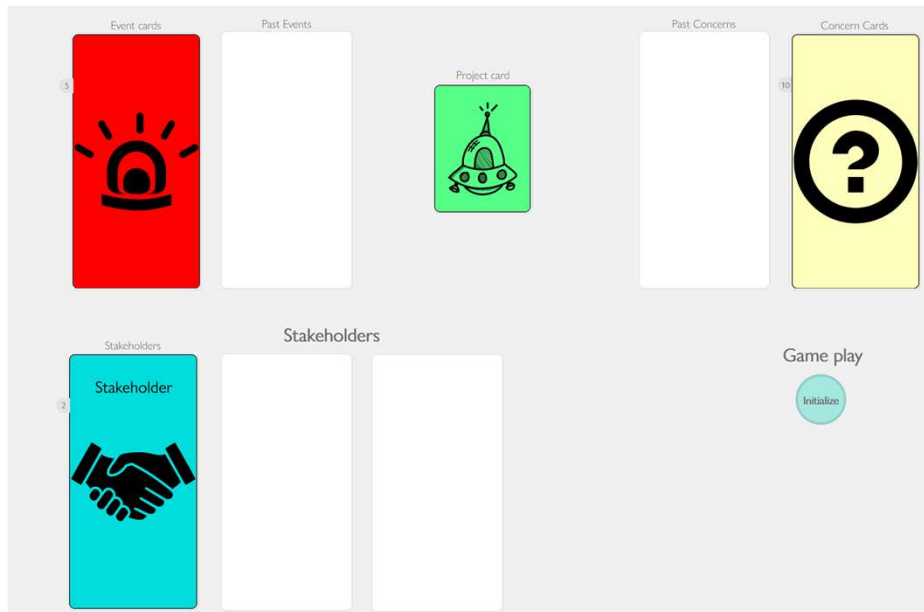
- *Software architecture is the composition of a set of architectural design decisions (Jansen & Bosch, 2005)*
- Software architecture
 - Trade-off between functional requirements & characteristics**
 - Each trade-off analysis results in a decision**
 - Communicate not only the what, but also the why!**



Utrecht University

DecidArch – The architecture game

DecidArch – the architecture game



- Play in rounds
 - Draw a concern card**
 - Individually: choose an option, provide rationale**
 - Discuss in the group which option you choose**
 - Document your decision, update the sketch**
 - Play next concern card**
 - After each round (of 3 concern cards)**
 - Play an event card**
- In Blackboard:
 - Instructions**
 - Excel sheet to fill in (1 per group)**
 - Link to playingcards.io environment**



Utrecht University

DecidArch – The architecture game

we start again at 10:45

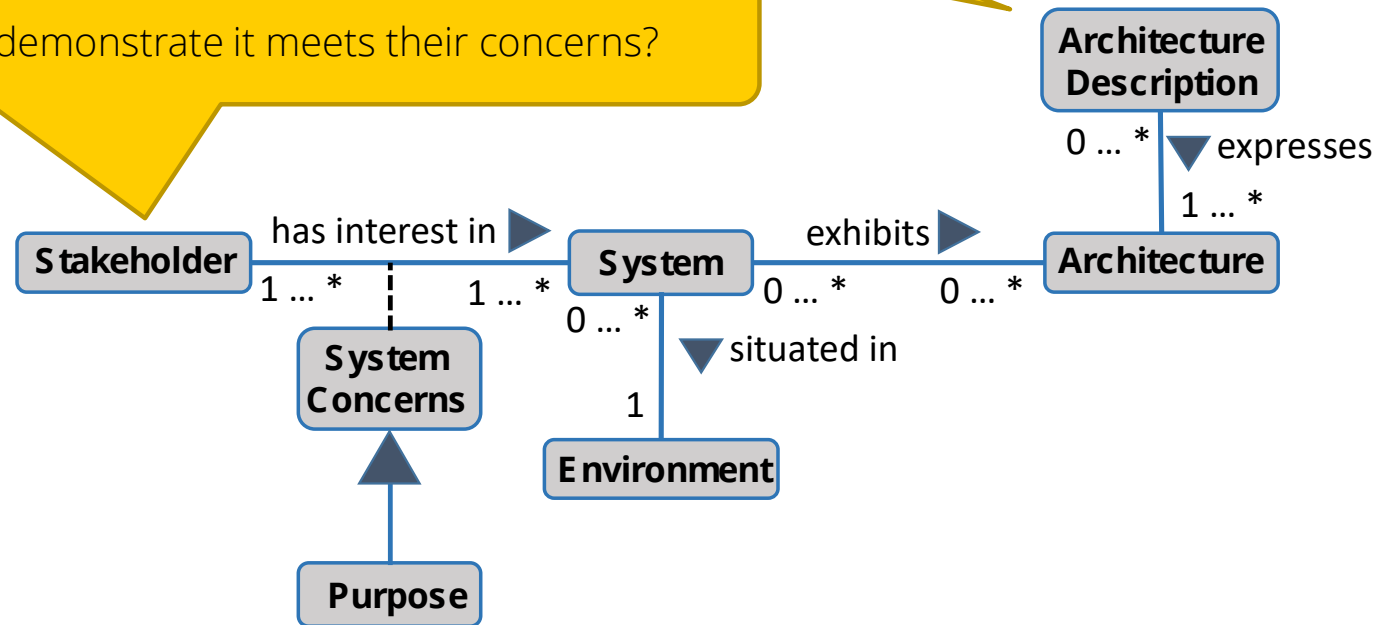


Architecture put in context



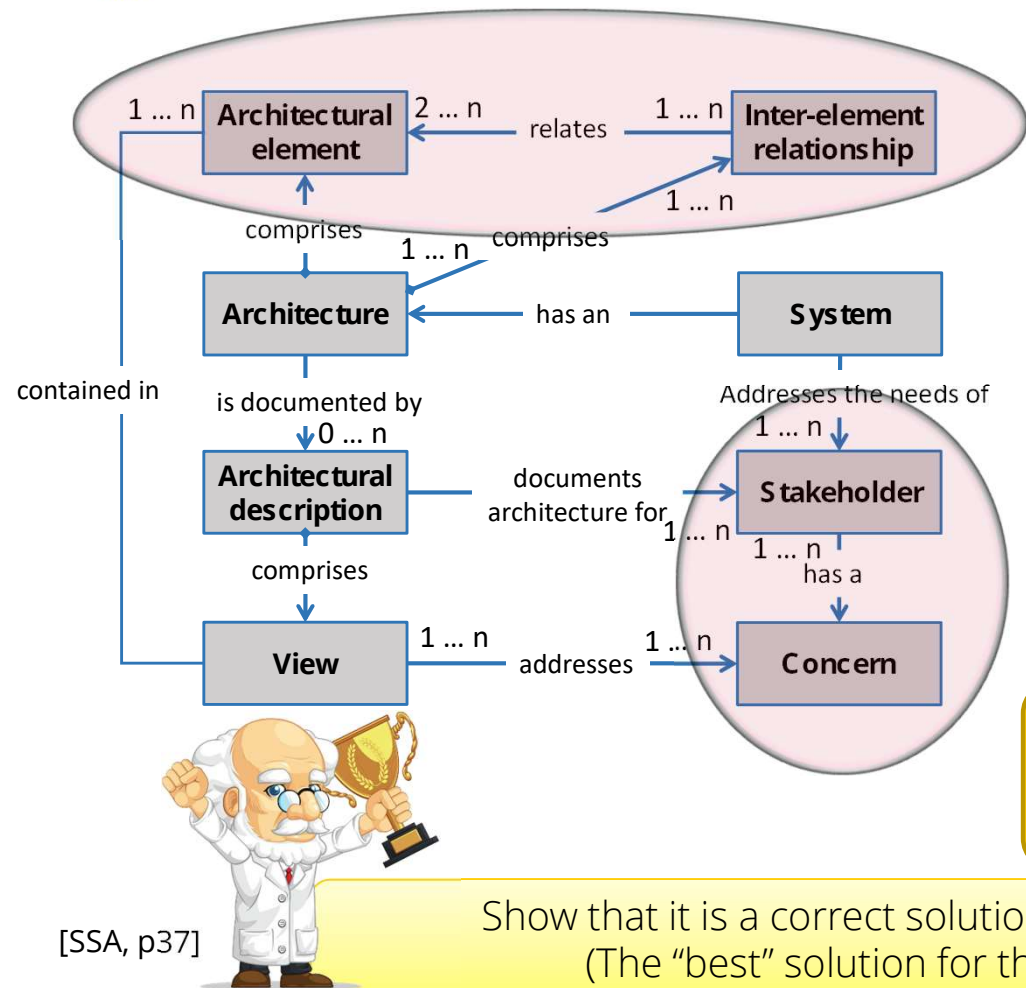
Should all stakeholders understand everything?

How to demonstrate it meets their concerns?





Structures and views



- A structure:
A set of elements itself, and a meaningful relation between them
- A view:
A representation of a coherent set of architectural elements, as written by and read by system stakeholders.

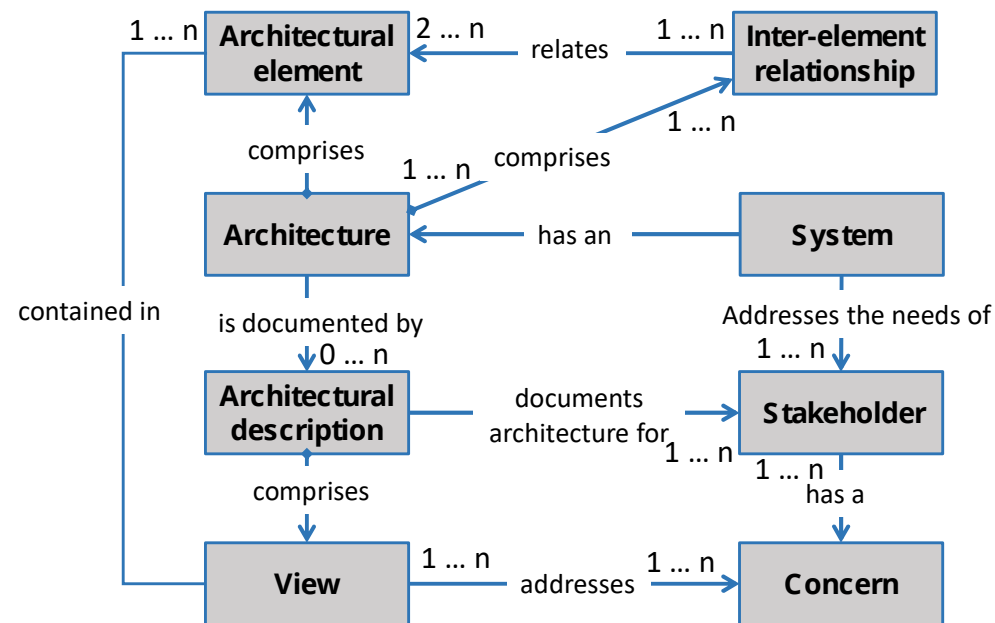
A view is a representation of a set of elements and the relations among them

**Architects design structures,
But document views**

Show that it is a correct solution for the problem!
(The "best" solution for the problem?)



Structures and views

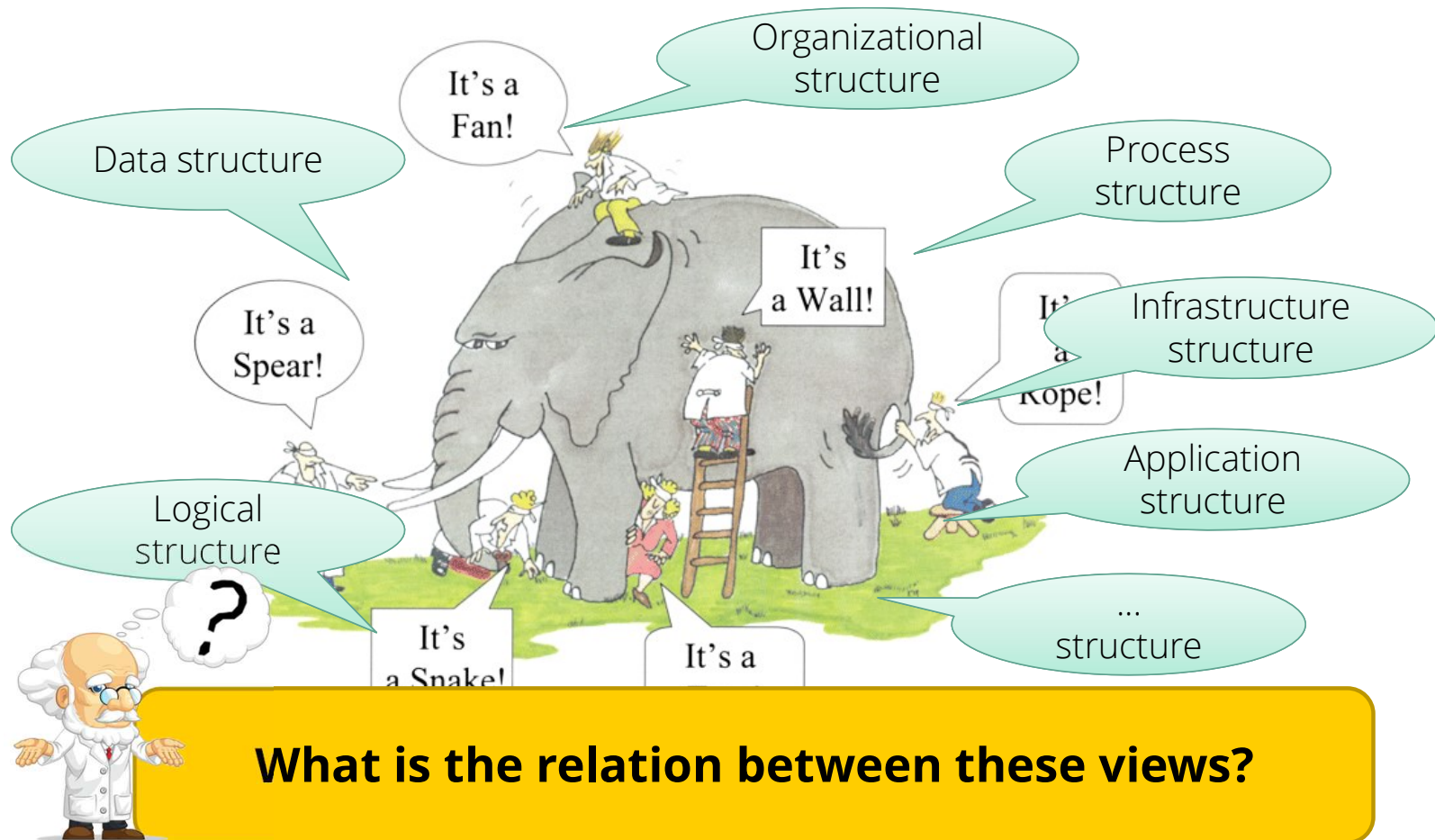


- Static structure
Defines the system's internal design-time elements and their arrangement
- Dynamic structure
Defines the runtime elements and their interactions
- Candidate architecture
A particular arrangement of static and dynamic structures that has the potential to exhibit the system's required externally visible behaviors and quality attributes

Architecture: select the "best" candidate



A set of structures





Structure of structures



How to keep those consistent?



Which structures do we need?

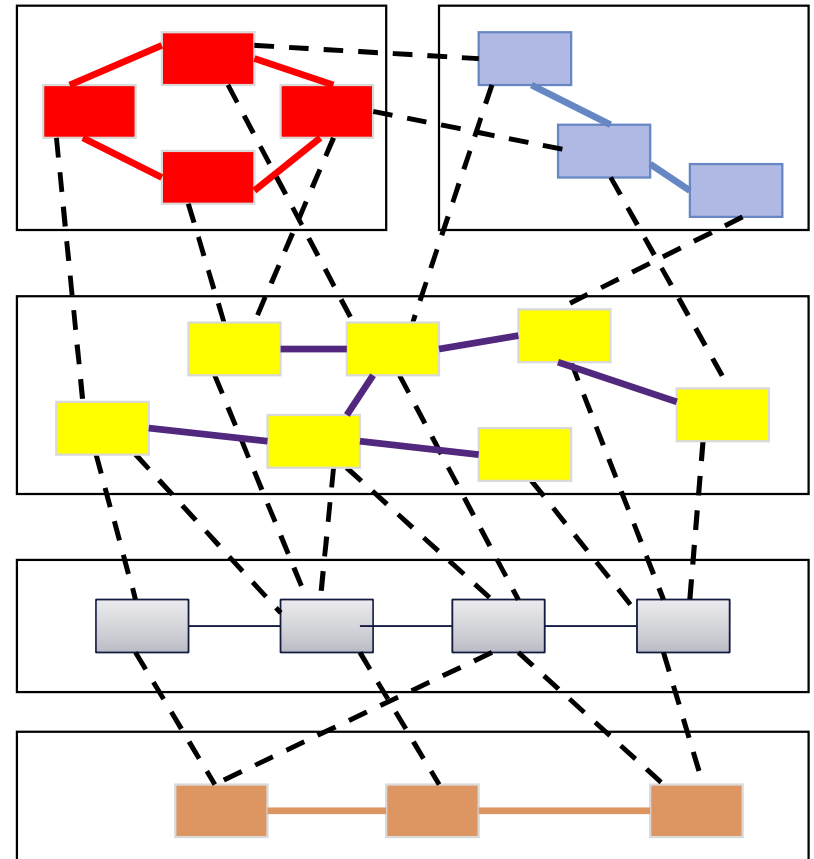
Functional view

Data view

Logical view

Virtual Server
deployment

Hardware
deployment

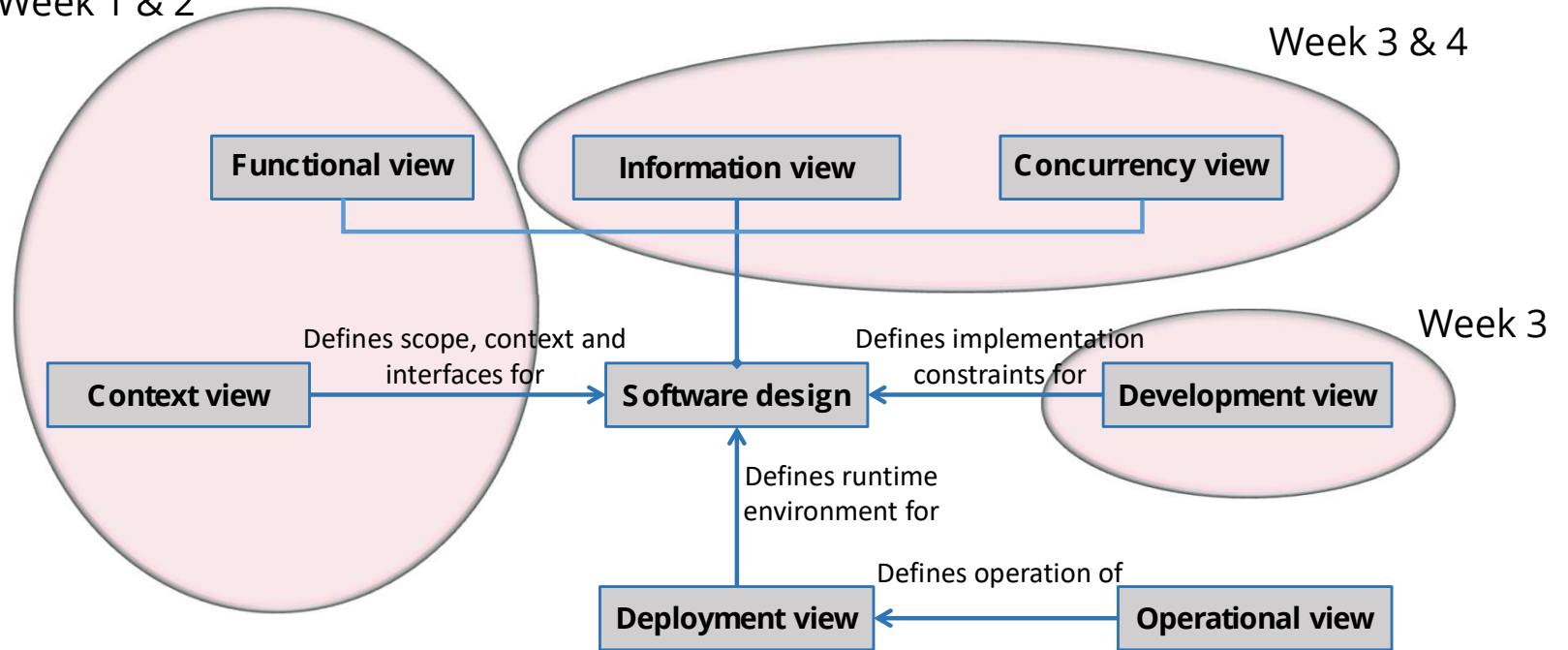




Viewpoints: an aid to structure views

Week 1 & 2

Week 3 & 4



Viewpoint:

Collection of patterns, templates and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint and the guidelines, principles, and template models for constructing its views

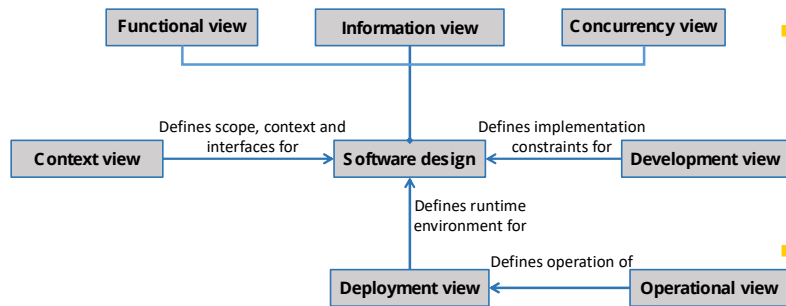


Utrecht University

Context viewpoint



Context view



Context view:

Describes the relationships, dependencies and interactions between the system and its environment

Concerns

**System scope and responsibilities
identity of external entities,
data and services used,
responsibilities of external interfaces,
impact of system on its environment**

Models and views

Stakeholder model

Context model

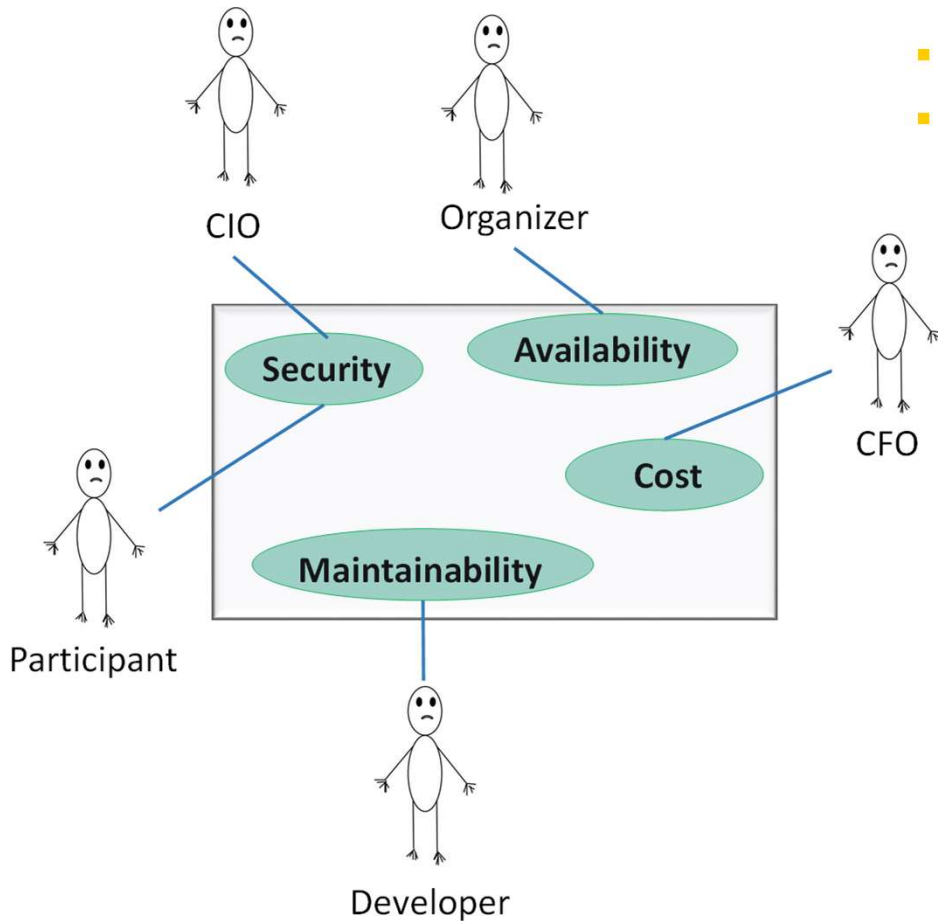
User stories

Use cases and interaction scenarios

...



Stakeholder model



- Who are the stakeholders?
- What are their main concerns?

Informal model

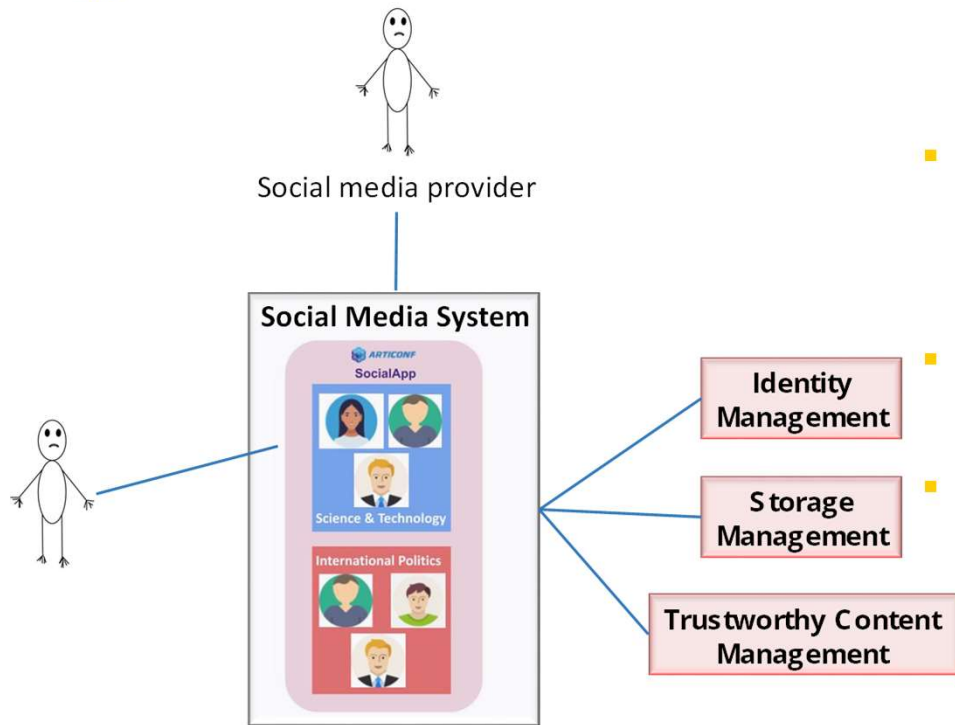
Useful for identifying stakeholders

Useful in communication

Recognize stakeholders explicitly



Context Model



- What are the main external dependencies?

Actors
Systems

- No (formal) diagram techniques available
- System as a black box



User Stories

- Strict format:
As a <role>
I want to <action>
(so that <benefit>)

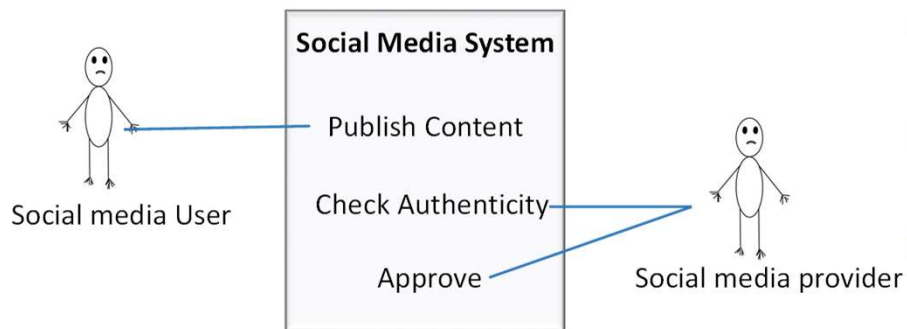
- Examples:
As a social media user, I want to **publish my content without any restriction**, so that **I can voice my opinion**

As a social media provider, I want to **audit the content before publication**, so that **I can prevent misinformation and fake news**





Use case models



- Goal **A future or desired result**
In US: <action>
- Actors **Something / someone interacting with the system**
In US: <role>
- Primary actor
Initiates the interaction to achieve the goal
- Trigger
Event that causes the use case to be initiated
- Precondition
Must be true or happen before the use case to be initiated
- Main success scenario
Sequence of activities required to achieve the goal
- Alternative paths
Variations on the main sequence where things go wrong



Utrecht University

About the assignment

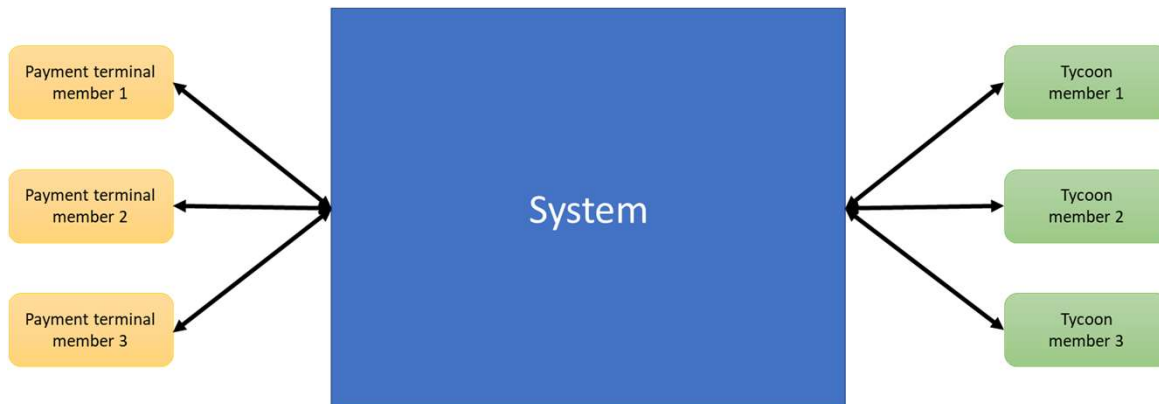
The Trip Inter Payment System



- Context:
 - Three railroad tycoons operate a railway**
 - Each has its own payment system**
 - Many complaints**
- Your task in teams of 3 students:
 - Architect a payment system for the three tycoons.**
- Stakeholders:
 - TrIP owner**
 - The three Railroad Tycoons**
 - Passengers**

The Trip Inter Payment System

- Constraints:
Microservice architecture
- We follow the ideas of the DecidArch game!
Events will occur...





Utrecht University

For today

- Develop the context viewpoint
 - Stakeholder model**
 - Context diagram**
 - User stories**
 - Prioritize the user stories**





Utrecht University

For Monday



Utrecht University

For Next Time

- Study the following paper:
M.D. McIlroy (1968). Mass Produced Software Components
- Questions:
What problems do the authors observe?
What solutions do they propose?





Utrecht University

DISCLAIMER

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

© Utrecht University