



Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Software Architecture: The consequences of your choices...

Jan Martijn van der Werf



Jan Martijn van der Werf

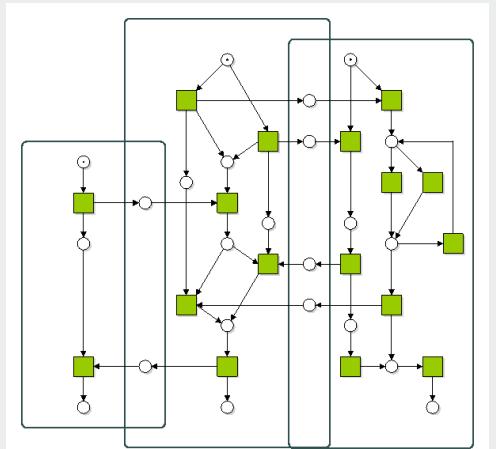
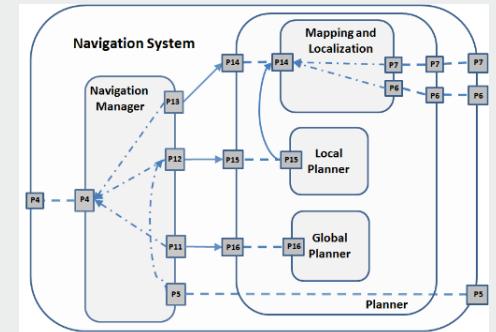
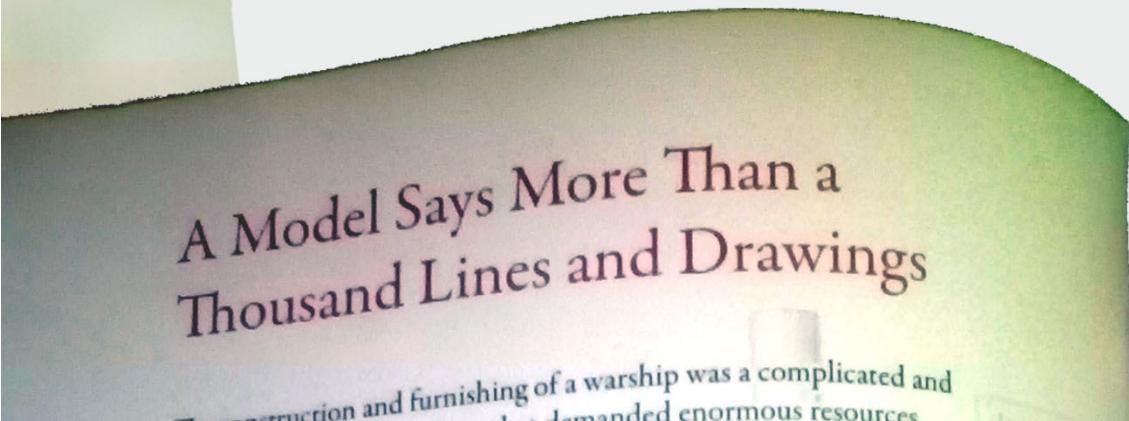


■ Education

- Information systems (BSc)
- Software Architecture (MSc)

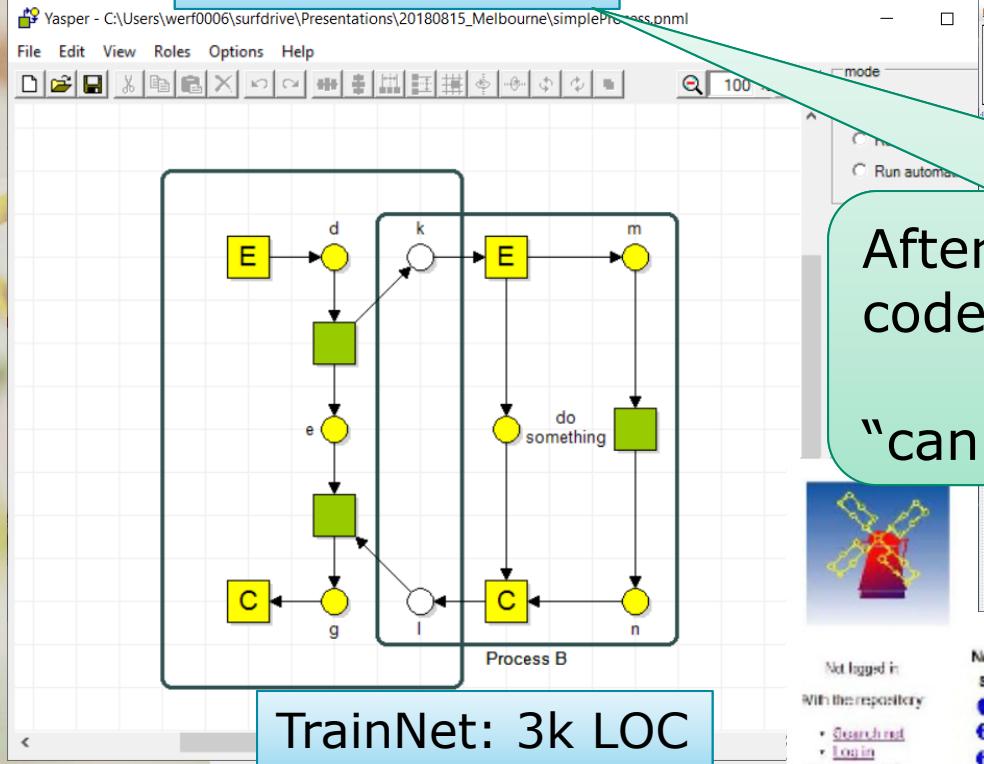
■ Research areas

- Software Architecture
- Process modeling & mining
- Modelling (behavior of/between/with) systems
- Architecture Rationale

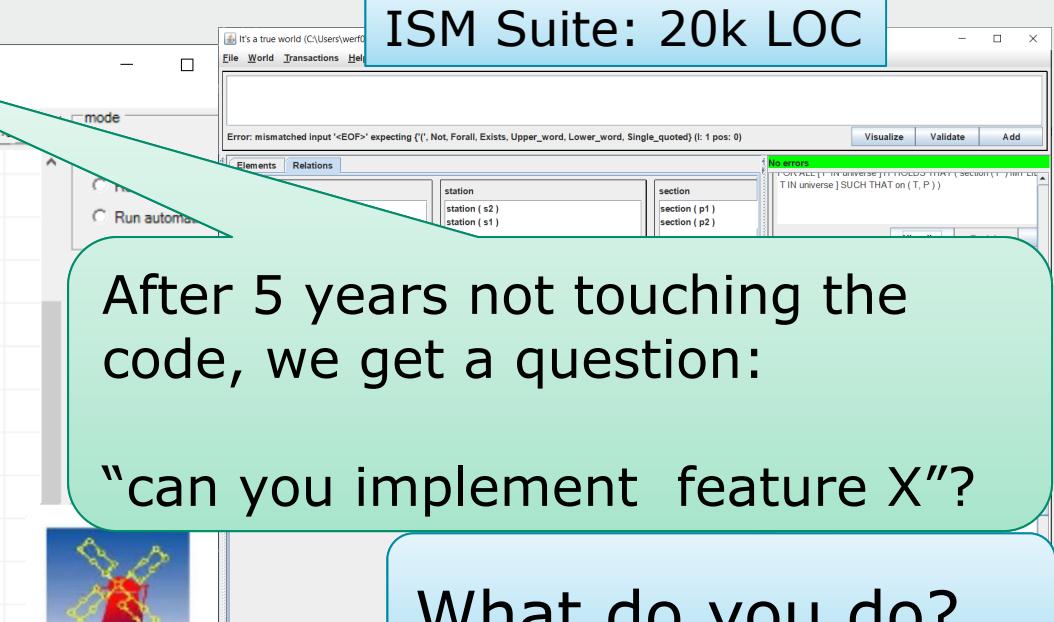


Sometimes I implement some stuff myself...

Yasper: 30k LOC



ISM Suite: 20k LOC

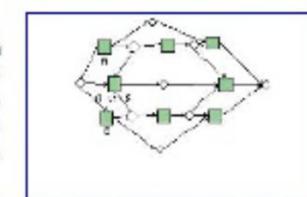


TrainNet: 3k LOC



3

Name: A or B
Structural properties of the graph
• Strongly_connected:
• Connected:
• Places:
• Free_choice:
• Communication_protocol
Behavioural properties dependent on initial marking
• Live:
• Bounded:
Exported File formats
• Pnml (regenerated):
Pnml (original):
I want to change the properties of this net | I want to delete this net



Name: basictransformations
Structural properties of the graph

Petriweb: 20k LOC



Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Software Architecture: The consequences of your choices...

Jan Martijn van der Werf

In one of the first lectures:

SOFTWARE ARCHITECTURE

- Software Architecture is a very popular term in the context of software development
- From Larman (p. 448):

[...] the architecture includes the organization and structure of the major elements of the system. Beyond this static definition, it includes the system behavior, especially in terms of large scale responsibilities of systems and subsystems, and their collaborations. In terms of a description, the architecture includes the motivations or rationale for why the system is designed the way it is.





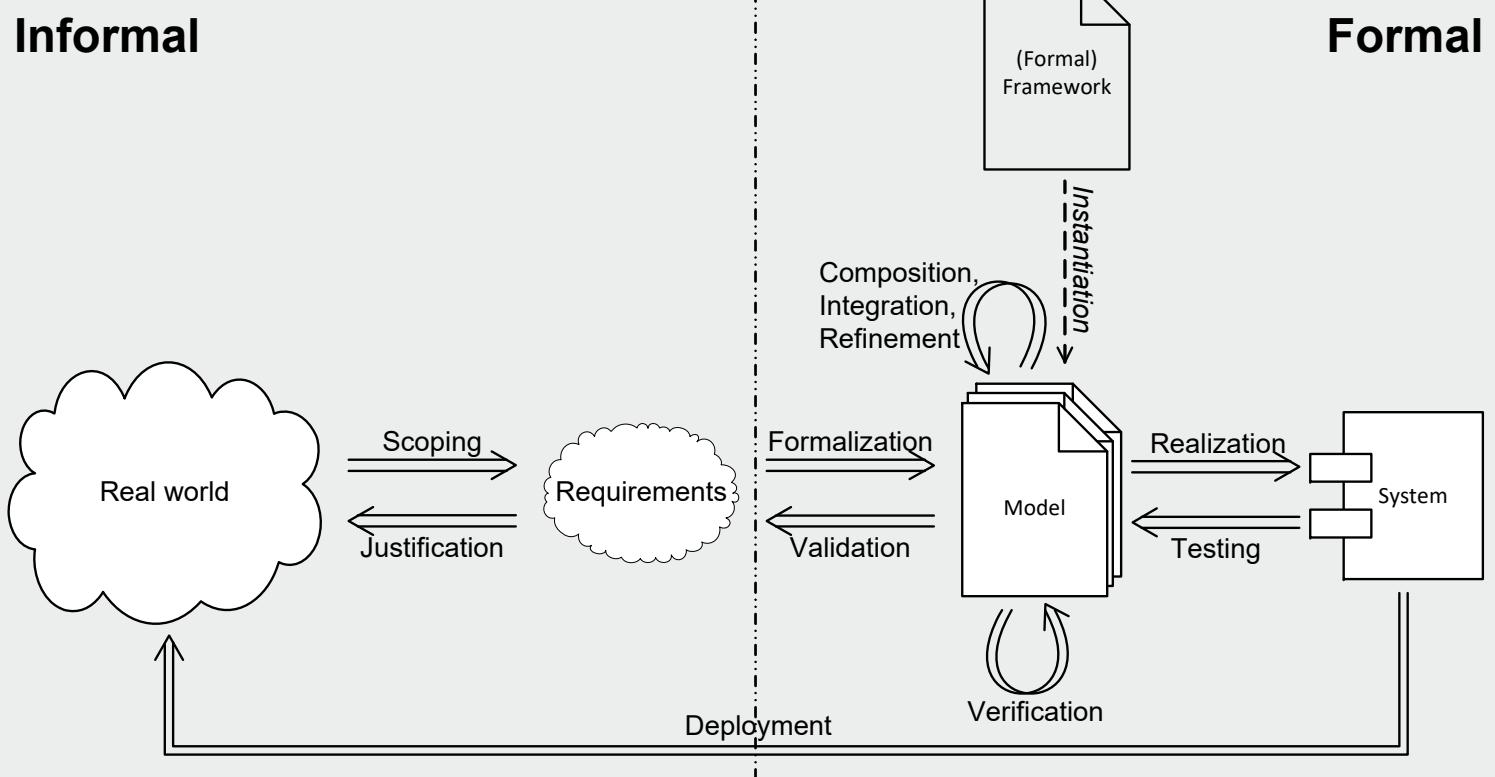
Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

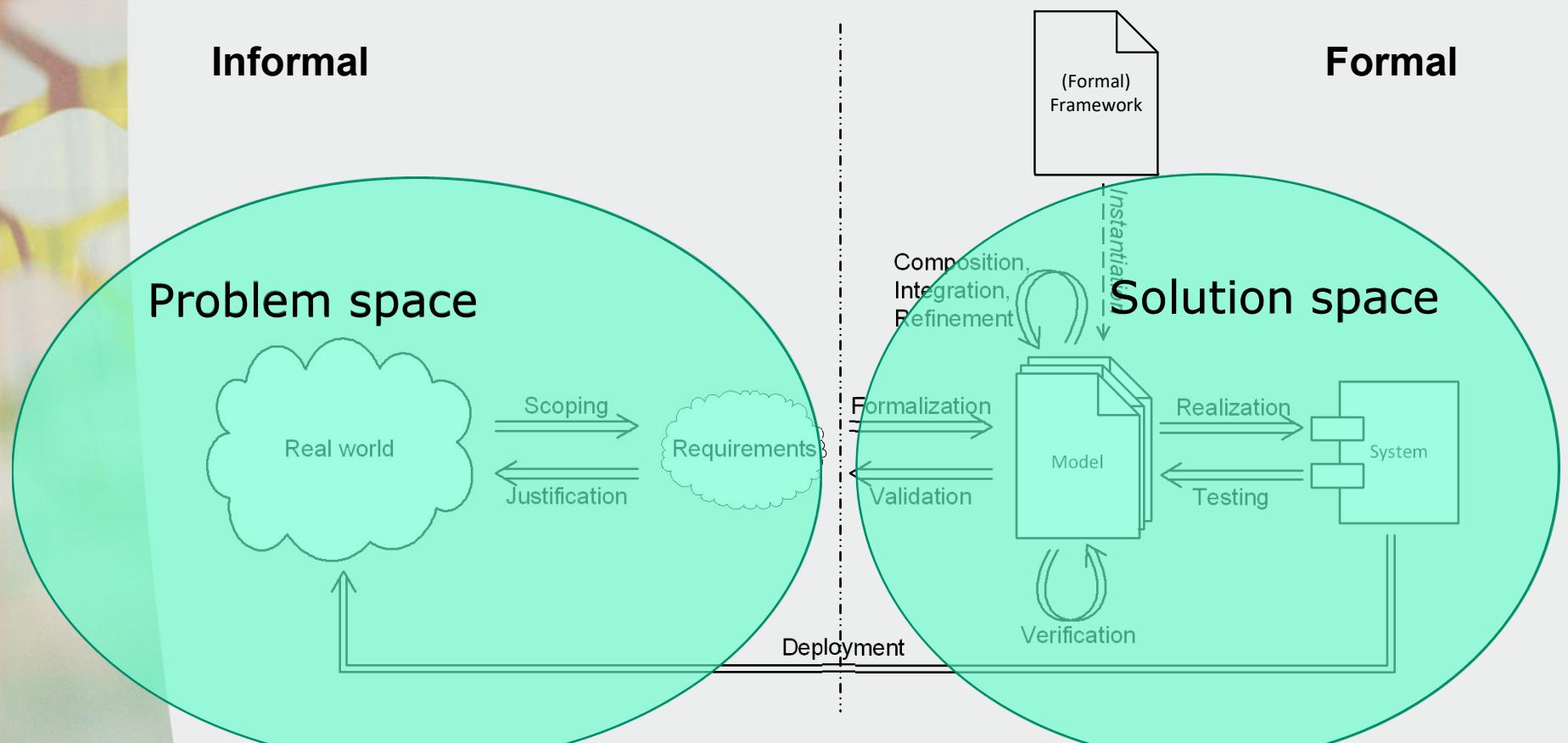
A (too) short introduction to Software Architecture

Jan Martijn van der Werf

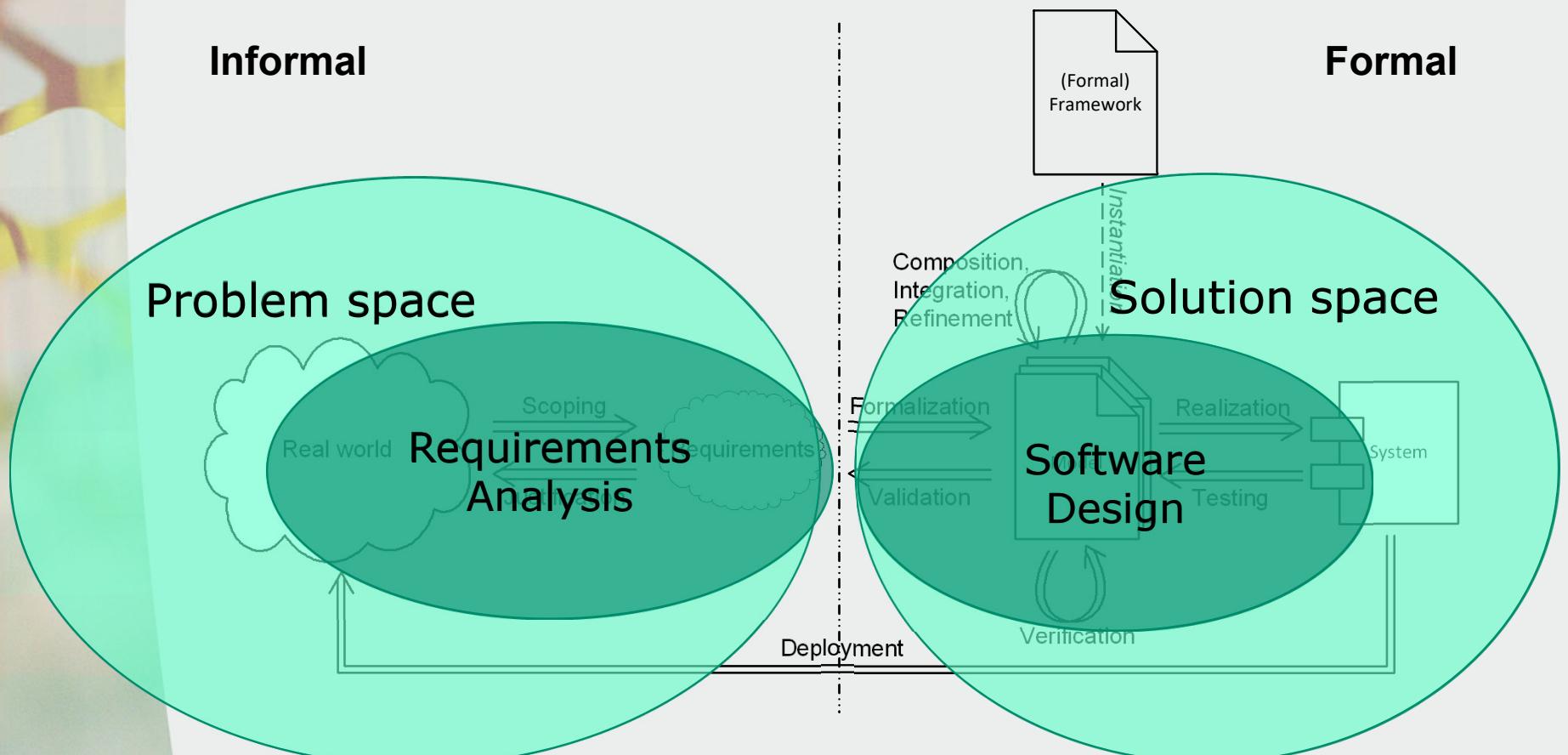
The role of Software Architecture



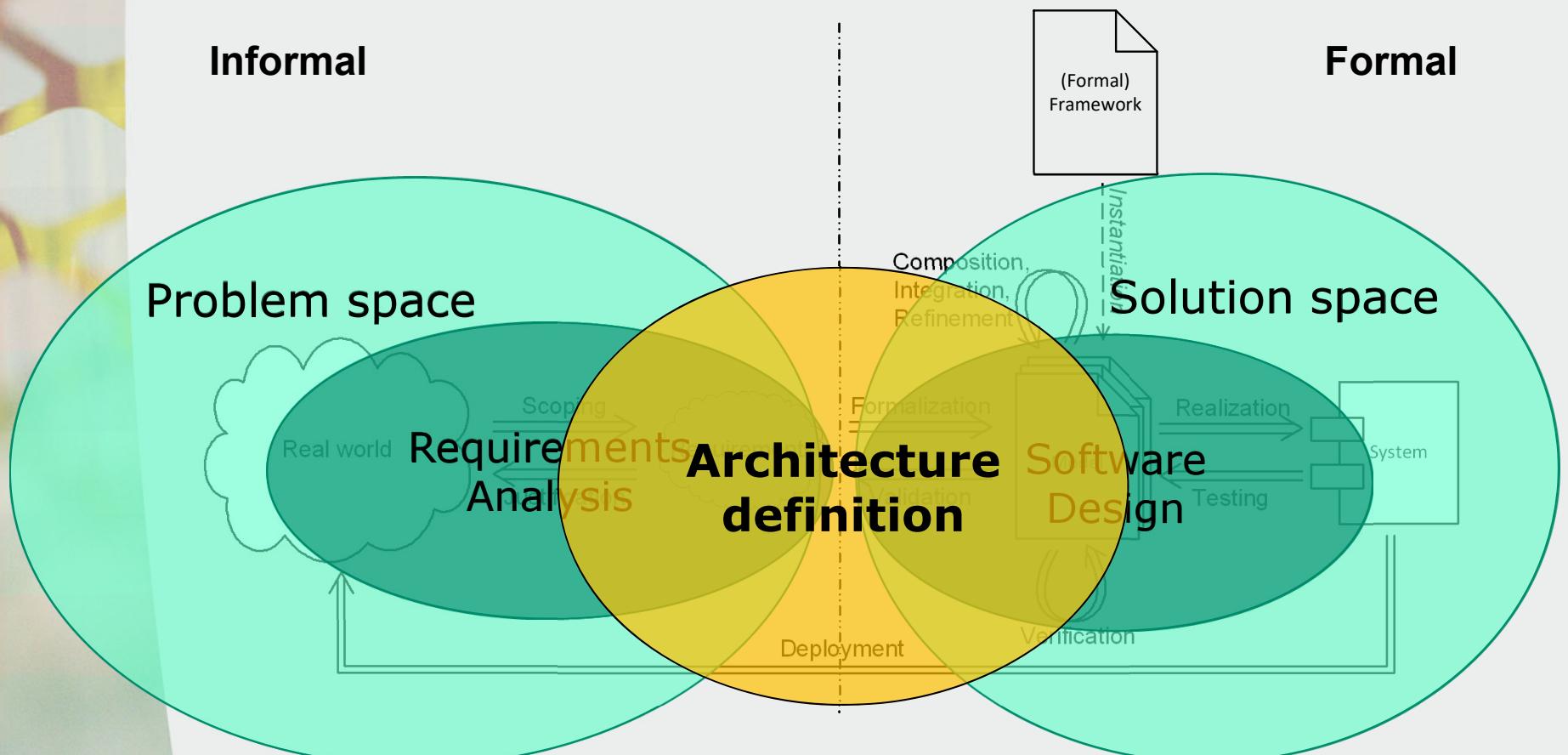
The role of Software Architecture



The role of Software Architecture



The role of Software Architecture



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:

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



Universiteit Utrecht

Why architecture?

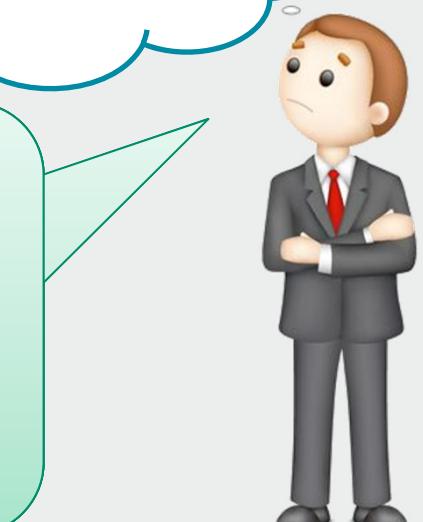
- “can’t we generate the design from the code?”
- “We communicate most decisions via code”
- “SA tends to embrace engineering concerns too strongly and too early”
- “Code is the best way to capture the end-user mental models in a form suitable to the shaping and problem solving that occur during design”



Building a software system



I need an online ticket sale for my conference



That is not too difficult...

Requirements:

- Event listing
- Event overview
- Ticket selection
- Payment with credit card
- Financial overview

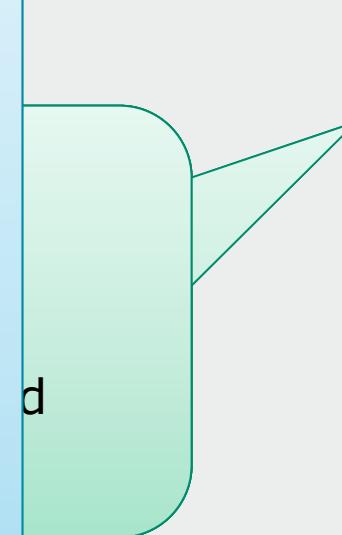
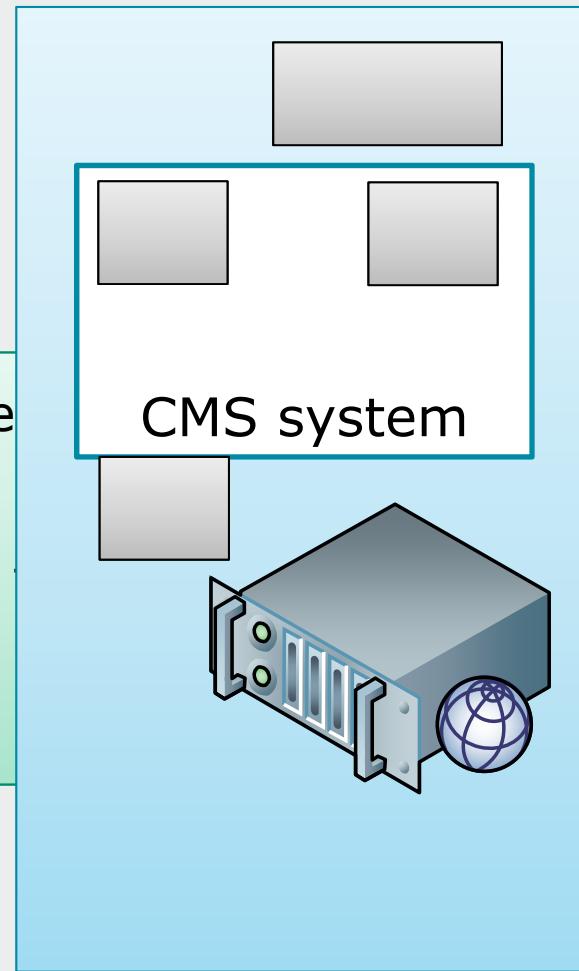


Building a software system



Re

-
-
-
-
-



d

[Faculty of Science
Information and Computing Sciences]



Universiteit Utrecht

Building a software system

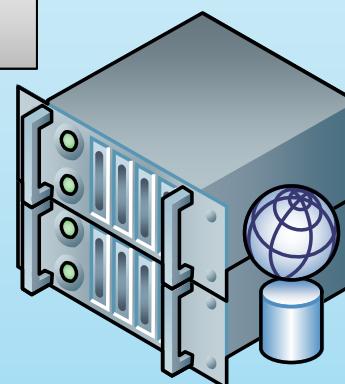
Oh wait, did I mention I expect 10k visitors per hour?



Re
-
-
-
-
-

Wait, I add a database server...

CMS system



d



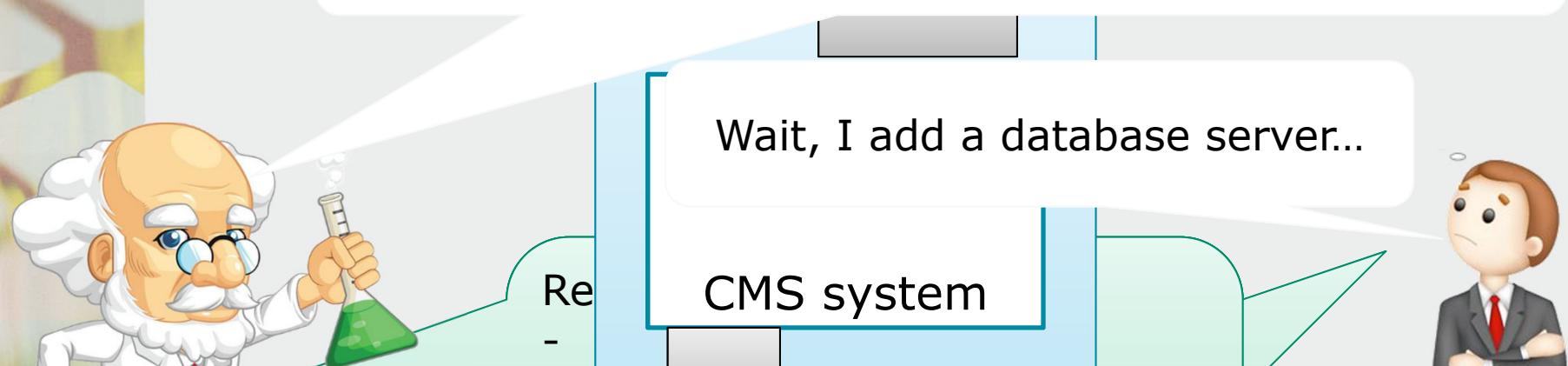
[Faculty of Science
Information and Computing Sciences]



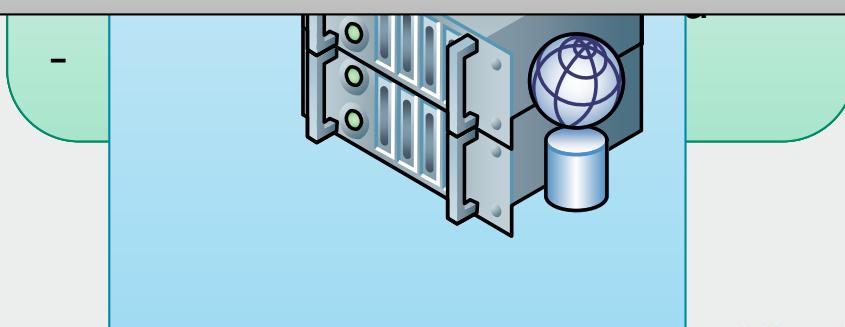
Universiteit Utrecht

Building a software system

Oh wait, did I mention I expect 10k visitors per hour?



How do you reason about such properties?



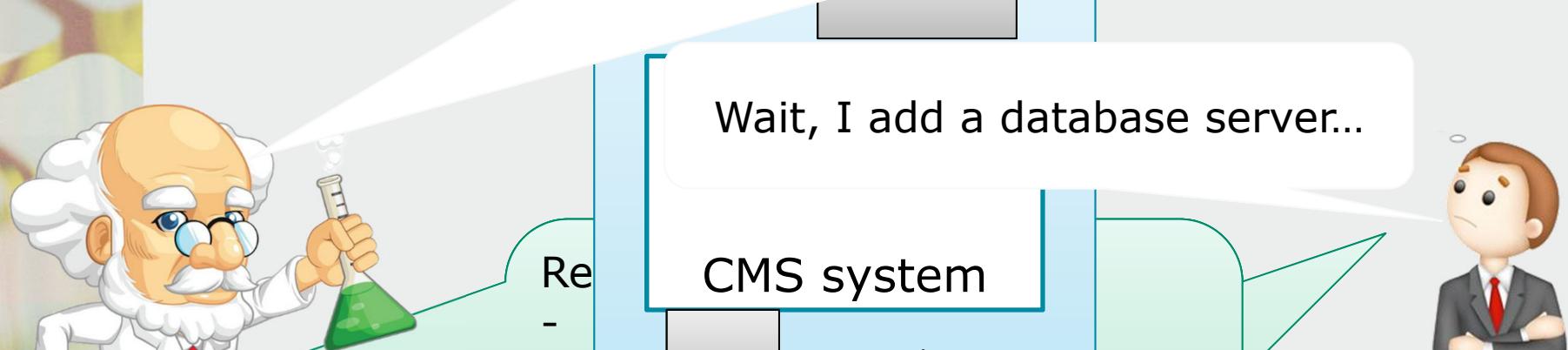
[Faculty of Science
Information and Computing Sciences]



Universiteit Utrecht

Building a software system

Oh wait, did I mention I expect 10k visitors per hour?



How do you reason about such properties?

Software Architecture:

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)



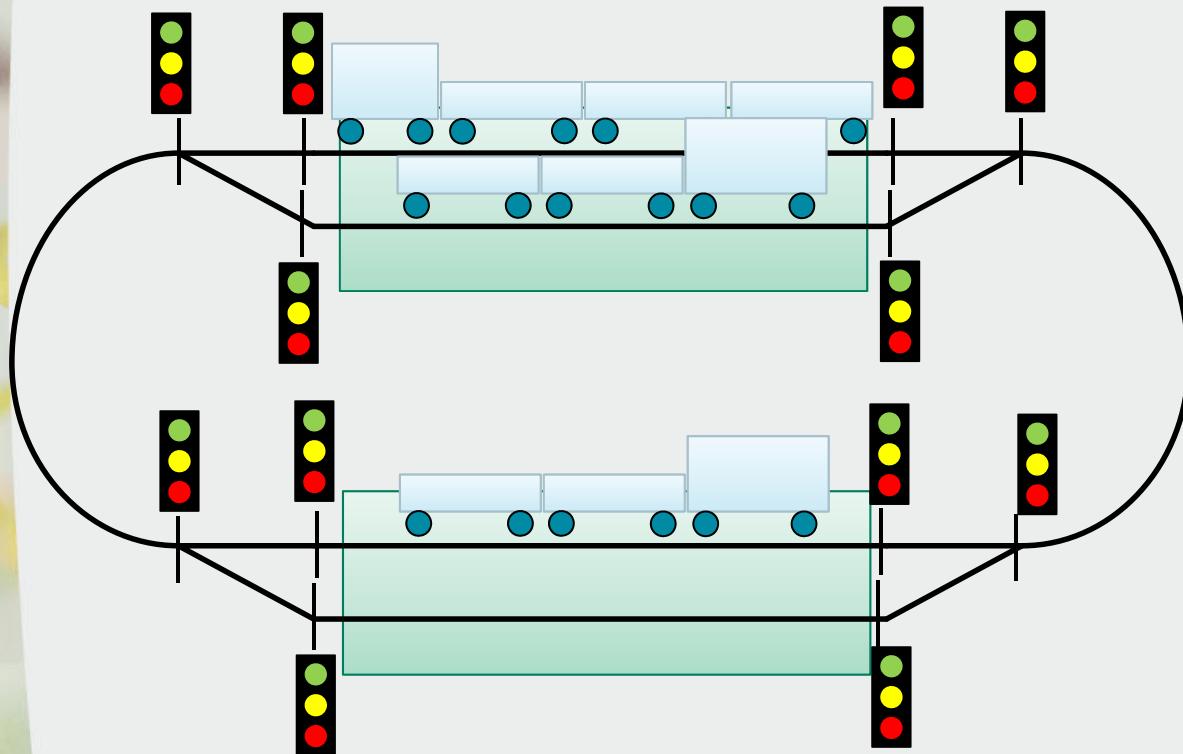
Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Let's move to a simple example



A simple railroad



Automatic driving:

- Trains should never collide
- Train drives between stations
- Signals handled automatically
- Switches handled automatically



Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

that ensures safe driving...

Decide new state on message and state

- Communication between dev via asynchronous messages
- Messages are sent over a bus
- Signal should react on specific message

Set signals

Our system

[Faculty of Science
Information and Computing Sciences]



Universiteit Utrecht

Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

that ensures safe driving...

Where to put the logic?

Decide new state on message and state

■ Loconet to communicate bet

Message:
- Address
- Value(s)

Message Handler

Set signals

State Handler

Our system

[Faculty of Science
Information and Computing Sciences]



Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

Where to put the logic?

Message Handler

State Handler

Our system

[Faculty of Science
Information and Computing Sciences]



Universiteit Utrecht

Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

Where to put the logic?

Message Handler

State Handler

Output Handler

Our system

[Faculty of Science
Information and Computing Sciences]



Universiteit Utrecht

Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

For each arrived message:

1. Check if address & state in table

2. If found, update state with entry

Message Handler

State Handler

Output Handler

Our system

[Faculty of Science
Information and Computing Sciences]



Universiteit Utrecht

Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

For each arrived message

1. Check if address & state in table

Bounded Linear Search

Linear

Entries in any order

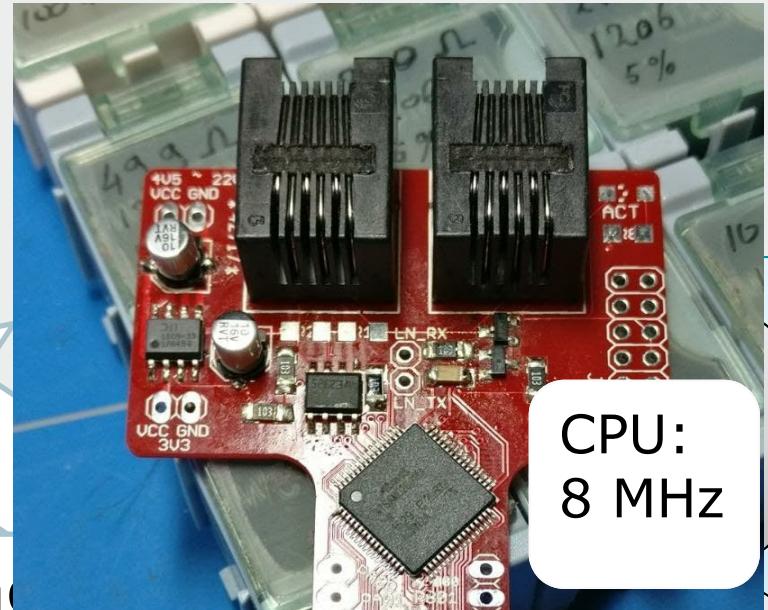
Simple data structure

Binary Search

Logarithmic

Entries sorted

Tree-based structure



State Handler

Output
Handler

Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

$$128 \text{ entries} = 2^7$$

Educated guess: max 128 cycles per entry

$$2^7 * 2^7 = 2^{14} \text{ cycles} @ 8\text{MHz} = 2\text{ms}$$

No decision: both fit within the 4ms!

Bounded Linear Search

Linear

Entries in any order

Simple data structure

Loconet: max 250 messages per second
Max 250 messages: max 1 per 4ms



CDLI

128 entries = 2^7
 $7 * 2^7 \approx 2^3 * 2^7 =$
 2^{10} cycles
@ 8MHz gives 0.128 ms

State Handler

Binary Search

Logarithmic

Entries sorted

Tree-based structure

Output Handler

Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

$$128 \text{ entries} = 2^7$$

Educated guess: max 128 cycles per entry
 $2^7 * 2^7 = 2^{14}$ cycles @ 8Mhz = 2ms

No decision: both fit within the 4ms!

Bounded Linear Search

Linear

Entries in any order

Simple data structure

Loconet: max 250 messages per second
Max 250 messages: max 1 per 4ms



CDLI

128 entries = 2^7
 $7 * 2^7 \approx 2^3 * 2^7 = 2^{10}$ cycles
@ 8MHz gives 0.128 ms

State Handler

Binary Search

Logarithmic

Entries sorted

Tree-based structure

Output Handler

Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address 2 bytes
- Use state-flag 2 bytes
- State 2 bytes
- Switch off mask 2 bytes
- Switch on mask 2 bytes

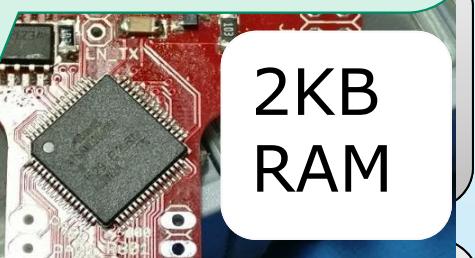
128 entries = 2^7

8 bytes for message

4 bytes for left subtree

4 bytes for right subtree

Structure: 2048 B = 2KB



128 entries = 2^7

8 bytes for message

Simple array: 1024 B = 1KB

Decision: use simple array!

Bounded List

Search

Linear

Entries in any order

Simple data structure

Binary Search

Logarithmic

Entries sorted

Simple data structure

State Handler

Output Handler

Our system

Decision table: 128 entries with:

- Address 2 bytes
- Use state-flag 2 bytes
- State 2 bytes
- Switch off mask 2 bytes
- Switch on mask 2 bytes



**Does not say anything about the algorithm...
Both can be implemented on a simple array...**

Constraint: use simple array!

Bounded Linear Search

Linear

Entries in any order

Simple data structure

Binary Search

Logarithmic

Entries sorted

Simple data structure

Output
Handler

Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address 2 bytes
- Use state-flag 2 bytes
- State 2 bytes
- Switch off mask 2 bytes
- Switch on mask 2 bytes



**Does not say anything about the algorithm...
Both can be implemented on a simple array...**

Constraint: use simple array!

Bounded Linear Search

Linear

Entries in any order

Simple data structure

Binary Search

Logarithmic

Entries sorted

Simple data structure

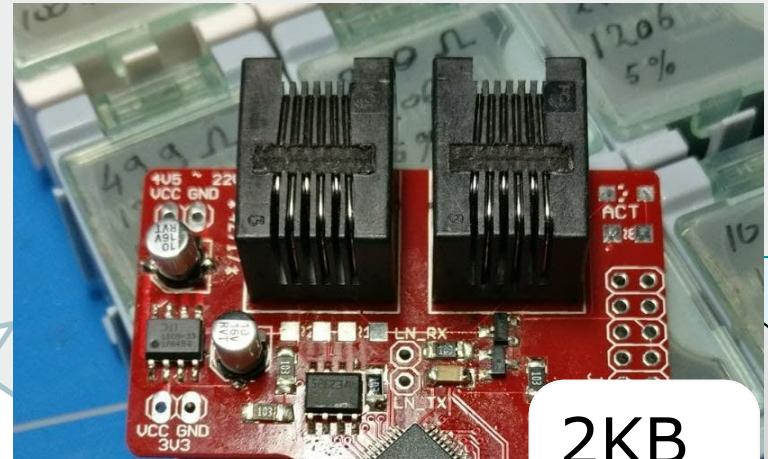
Output
Handler

Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address 2 bytes
- Use state-flag 2 bytes
- State 2 bytes
- Switch off mask 2 bytes
- Switch on mask 2 bytes



Maintaining the table:

Insert: add to the back O(1)

Delete: Place last over removed entry O(1)

1. Check

address & state in table

Maintaining the table:

Insert: ~~add to the back~~ O(n)

Delete: ~~Place last over removed entry~~ back O(n)

Bounded L

Linear

Entries in any order

Simple data structure

Linear Search

Binary Se

Logarithmic

Entries sorted

Simple data structure

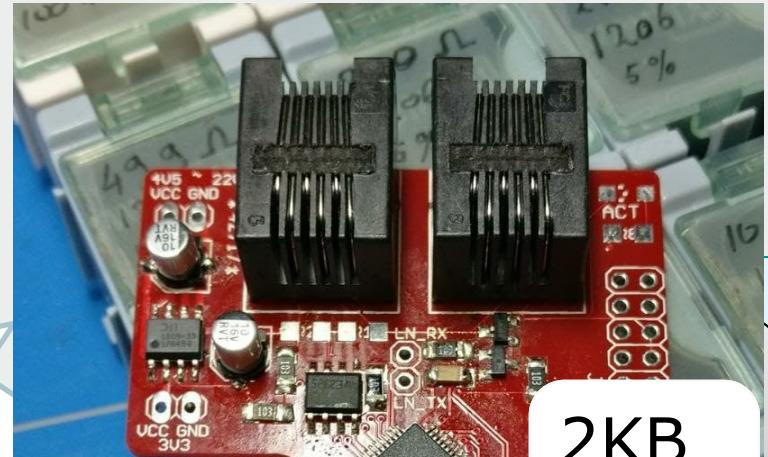
Output Handler

Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address 2 bytes
- Use state-flag 2 bytes
- State 2 bytes
- Switch off mask 2 bytes
- Switch on mask 2 bytes



Maintaining the table:

Insert: add to the back O(1)

Delete: Place last over removed entry O(1)

1. Check

address & state in table

Maintaining the table:

Insert: ~~add to the back~~ O(n)

Delete: ~~Place last over removed entry~~ back O(n)

Bounded L

Linear

Entries in any order

Simple data structure

Linear Search

Binary Se

Logarithmic

Entries sorted

Simple data structure

Output Handler

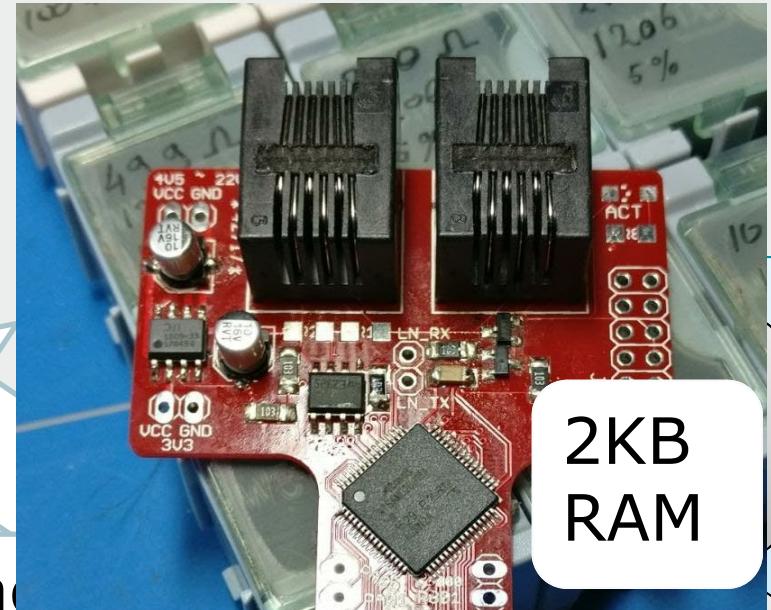
Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address 2 bytes
- Use state-flag 2 bytes
- State 2 bytes
- Switch off mask 2 bytes
- Switch on mask 2 bytes

For each arrived message



State Handler

We favor maintainability over performance

Bounded Linear Search

Linear

Entries in any order

Simple data structure

Binary Search

Logarithmic

Entries sorted

Simple data structure

Output Handler

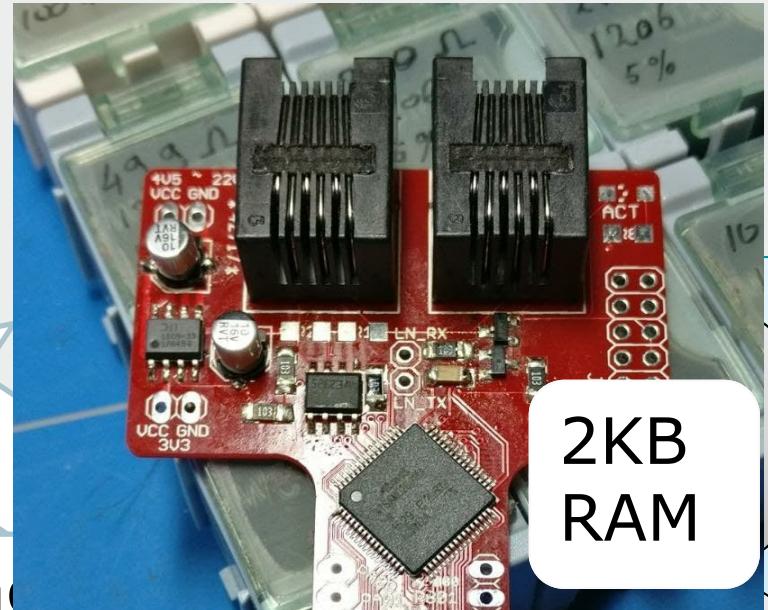
Our system

[Faculty of Science
Information and Computing Sciences]

Decision table: 128 entries with:

- Address 2 bytes
- Use state-flag 2 bytes
- State 2 bytes
- Switch off mask 2 bytes
- Switch on mask 2 bytes

For each arrived message



State Handler

We favor maintainability over performance

Bounded Linear Search

Linear

Entries in any order

Simple data structure

Binary Search

Log

Entries

Simple data structure

Output Handler

Our system

[Faculty of Science
Information and Computing Sciences]

Let's structure and think over the options...

Context:	Options	Decision:
Facing concern:		Because:
Criteria:		Consequences:



Let's look at the decision we just took...

Context:	Options		Decision:
Table search	Candidate 1	Candidate 2	Candidate 1
Facing concern:	Bounded linear search in a table of max 128 entries	Binary search in a table of max 128 entries	Because:
Just in time message handling	0 Within 2ms < 4ms, thus ok	++ Within 0.125ms < 4ms, thus ok	It requires simpler code, and is fast enough
Criteria:	0	++	Consequences:
Performance	Simple array, max 1 KB	Possible with simple array	Slower performance per search
Memory	0	0	
Maintenance	++ All table operations O(1)	-- All table operations O(n)	



Decision table: 128 entries with:

- Address
- Use state-fla
- State
- Switch
- Switch

2 bytes

We f

Bounded

Linear

Entries i

Simple da

Decision story: Search strategy

**In the context
facing
We decided to
And not
Because
and
Accepting**

of our RailGuard PCB,
just in time message handling,
use the Bounded Linear Search
the Binary Search
It is fast enough,
It keeps the code base simpler
the longer runtime required for
a single pass.

Decision table: 128 entries with:

- Address
- Use state-fla
- State
- Switch
- Switch

2 bytes



Decision story: Search strategy

**In the context
facing
We decided to
And not
Because
and
Accepting**

We f

Bounded I

of our RailGuard PCB,
just in time message handling,
use the Bounded Linear Search
the Binary Search
It is fast enough,
It keeps the code base simpler
the longer runtime required for
a single pass.

Software Architecture:

The composition of all principle design decisions

(Jansen & Bosch, 2005)

Decision table: 128 entries with:

- Address
- Use state-fla
- State
- Switch
- Switch

2 bytes

We f

Bounded

Linear

Entries i

Simple da

Decision story: Search strategy

**In the context
facing
We decided to
And not
Because
and
Accepting**

Current
research!

of our RailGuard PCB,
just in time message handling,
use the Bounded Linear Search
the Binary Search
It is fast enough,
It keeps the code base simpler
the longer runtime required for
a single pass.





Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Software Architecture:

- 1) structures, relations and properties,
and**
- 2) The set of principle design decisions**

Jan Martijn van der Werf



Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Software architecture ≠ Software design

Software architecture vs Software design

- Software design:
 - functional requirements
 - Patterns to be used
- Software architecture adds to that:
 - Software characteristics ("ilities")
 - Tradeoff analysis between requirements
 - Stakeholders and their concerns
 - People (not only developers!) management



Architecture in practice

"Working software over comprehensive documentation"

-- the agile manifesto

Upd~~ate~~ecture

But also:

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

-- the agile manifesto

Product Backlog

Sprint Backlog

Sprint



Working increment
of the software

PSA: Project Start
Architecture

PSA ≠ PEA

PEA: Project End
Architecture



Universiteit Utrecht

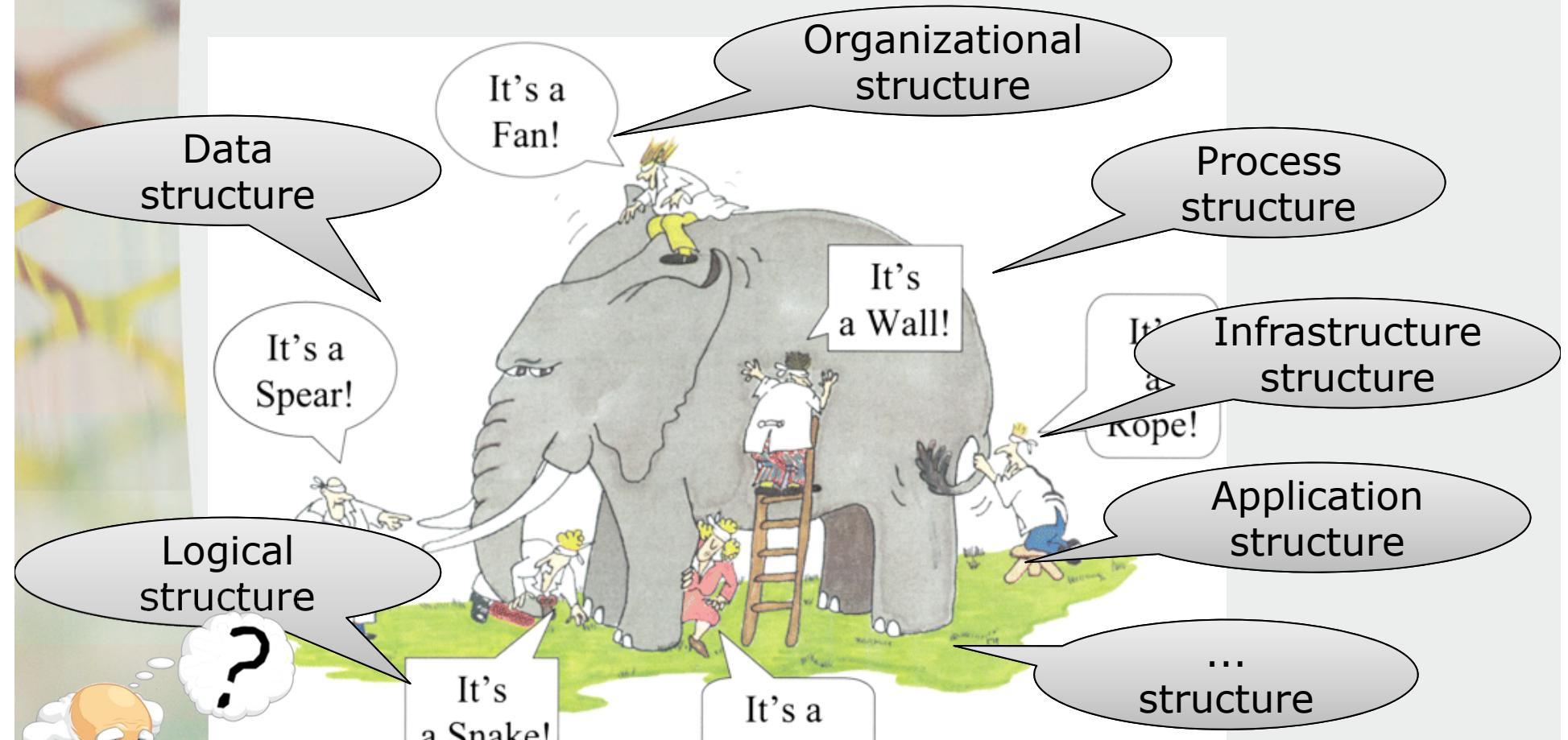


Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Software as a set of views

A set of structures



What is the relation between these views?

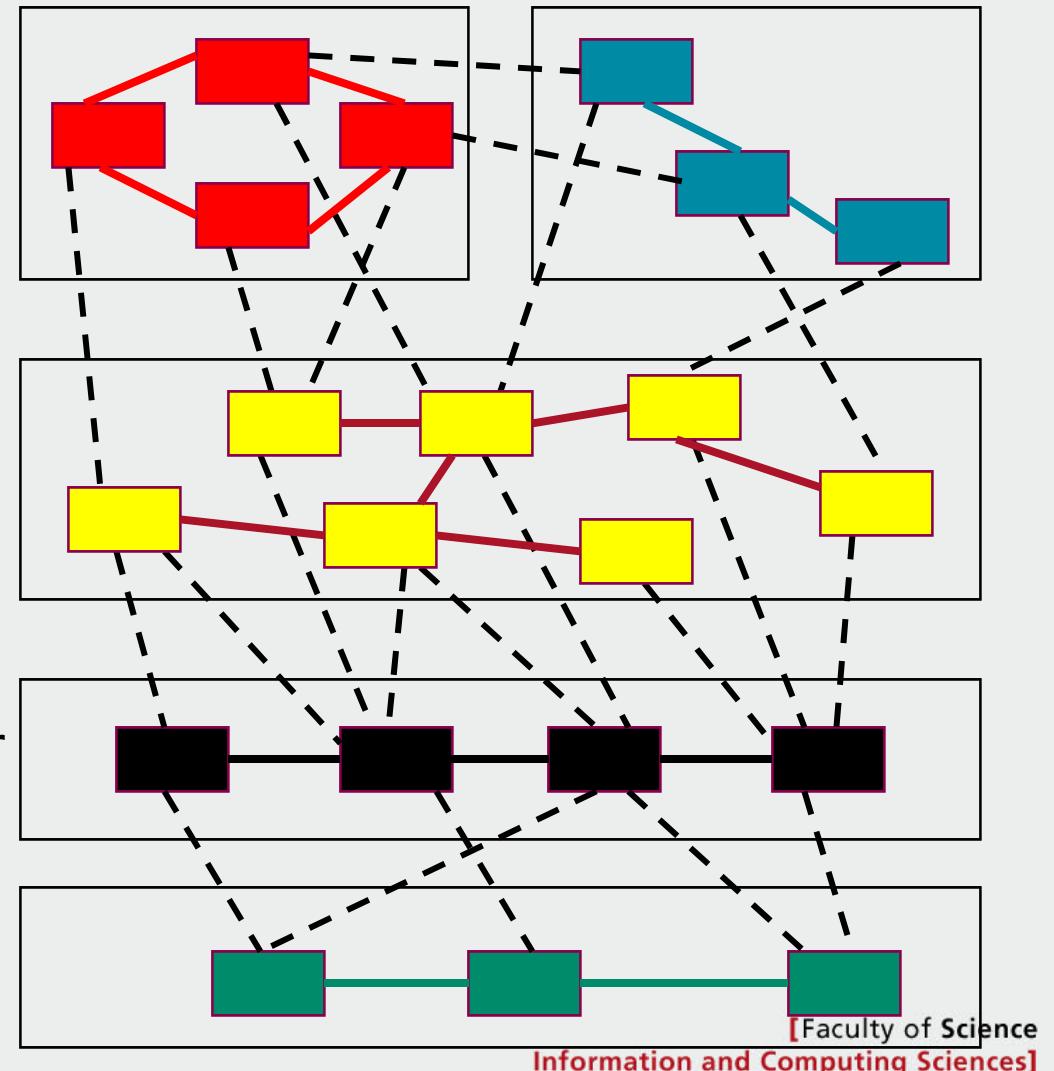
Structure of structures

Functional view
Data view

Code-unit view

Virtual Server deployment

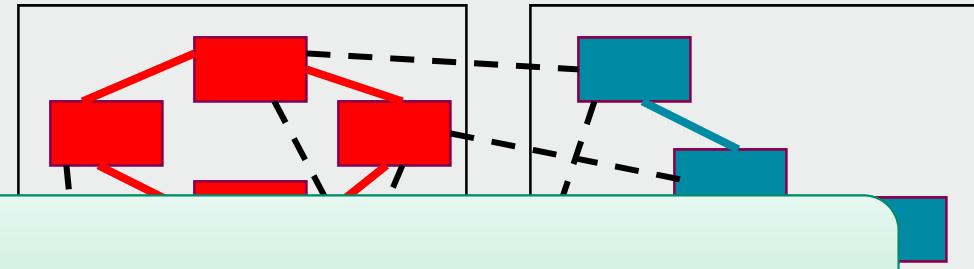
Hardware deployment





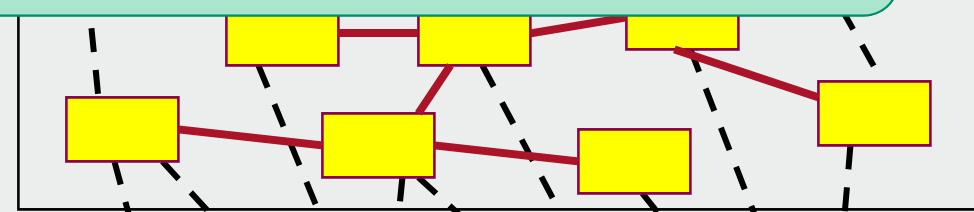
Structure of structures

Functional view
Data view

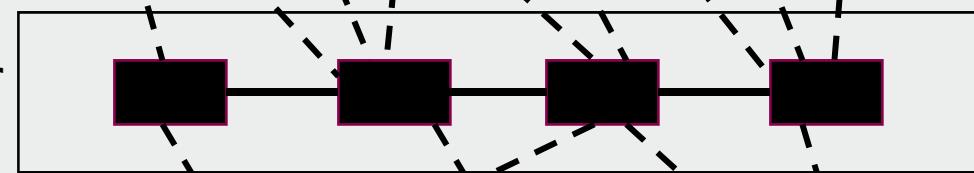


How to keep those consistent?

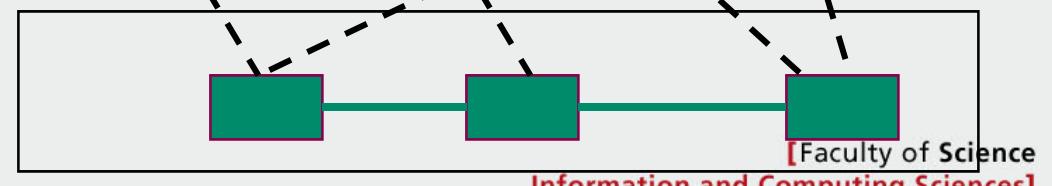
Code-unit view



Virtual Server deployment



Hardware deployment



Faculty of Science
Information and Computing Sciences

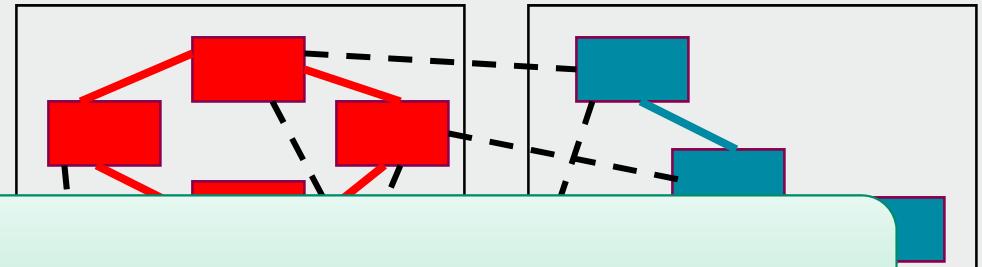


Universiteit Utrecht

Structure of structures



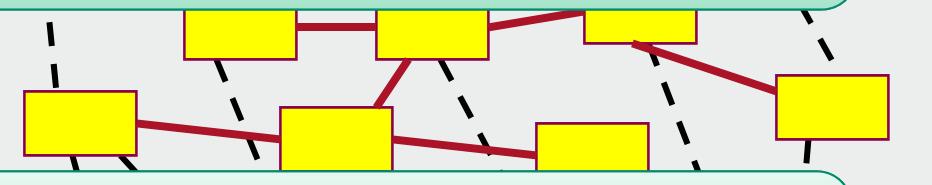
Functional view
Data view



How to keep those consistent?



Code-unit
view

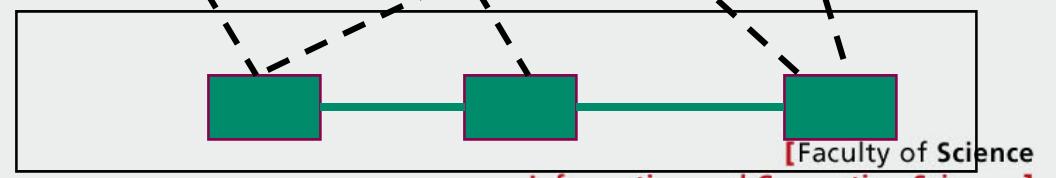


Which structures do we need?

deployment



Hardware
deployment

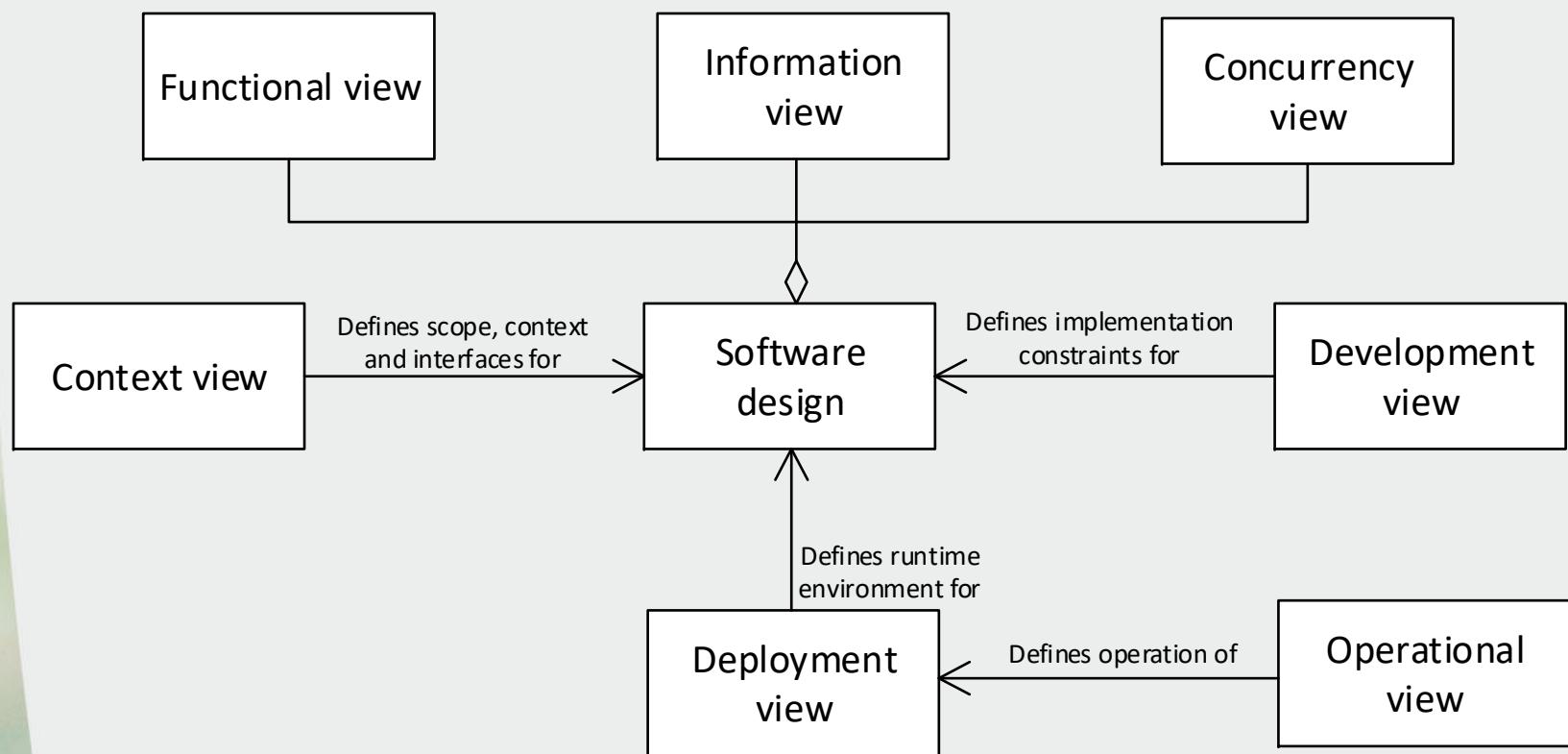


Faculty of Science
Information and Computing Sciences

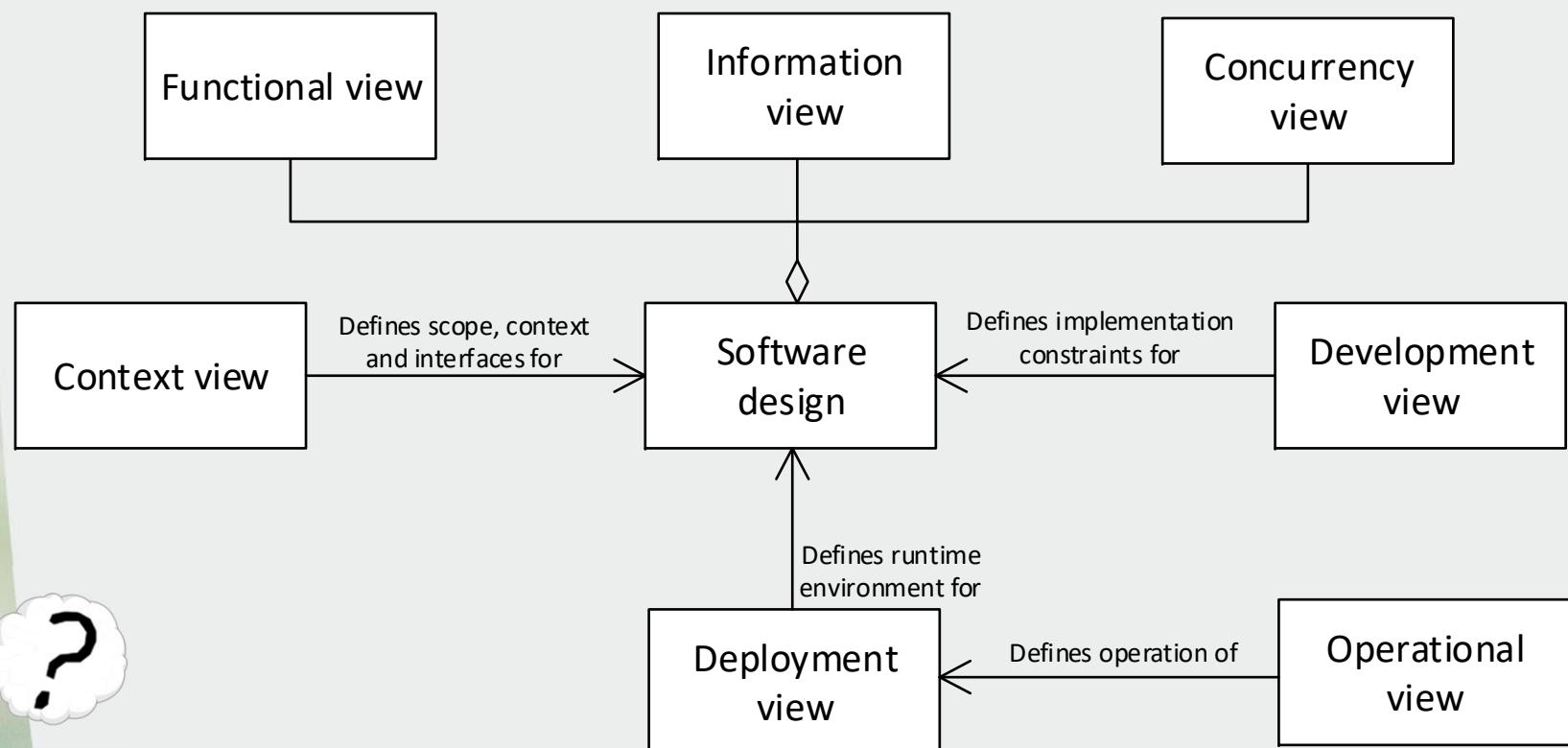


Universiteit Utrecht

Classifying the different views...

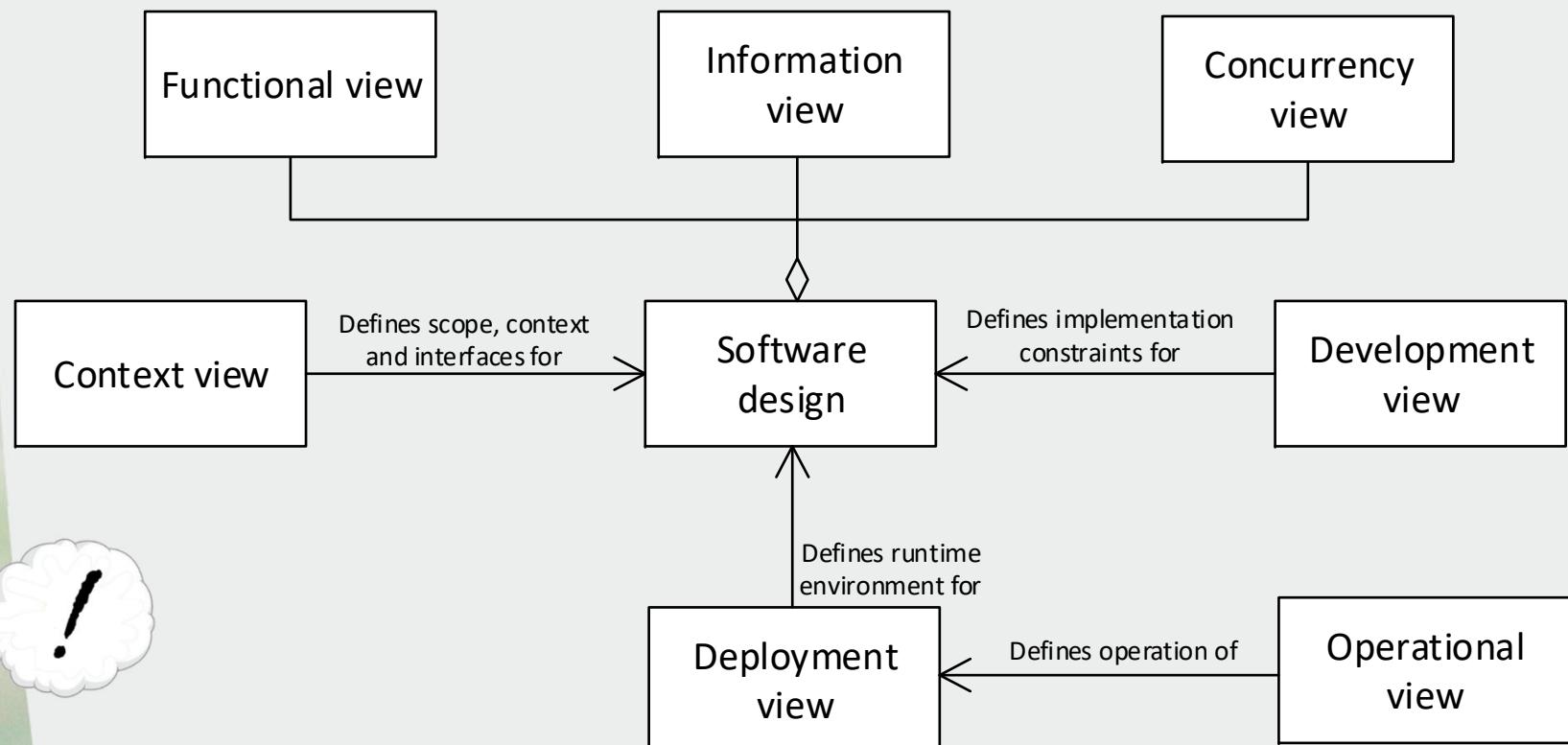


Classifying the different views...



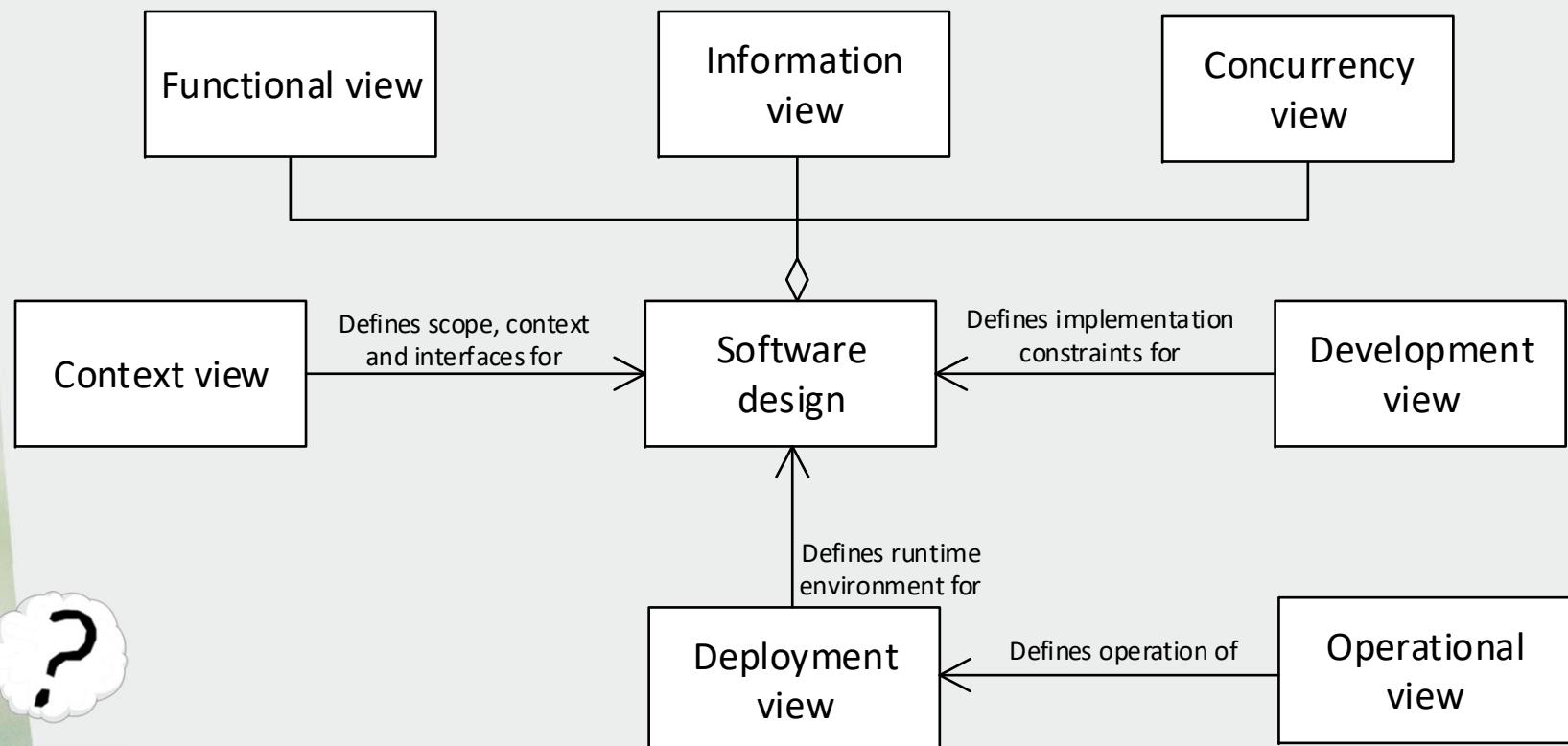
Where are properties of the system?

Classifying the different views...



Perspective: how a certain property is addressed in each view...

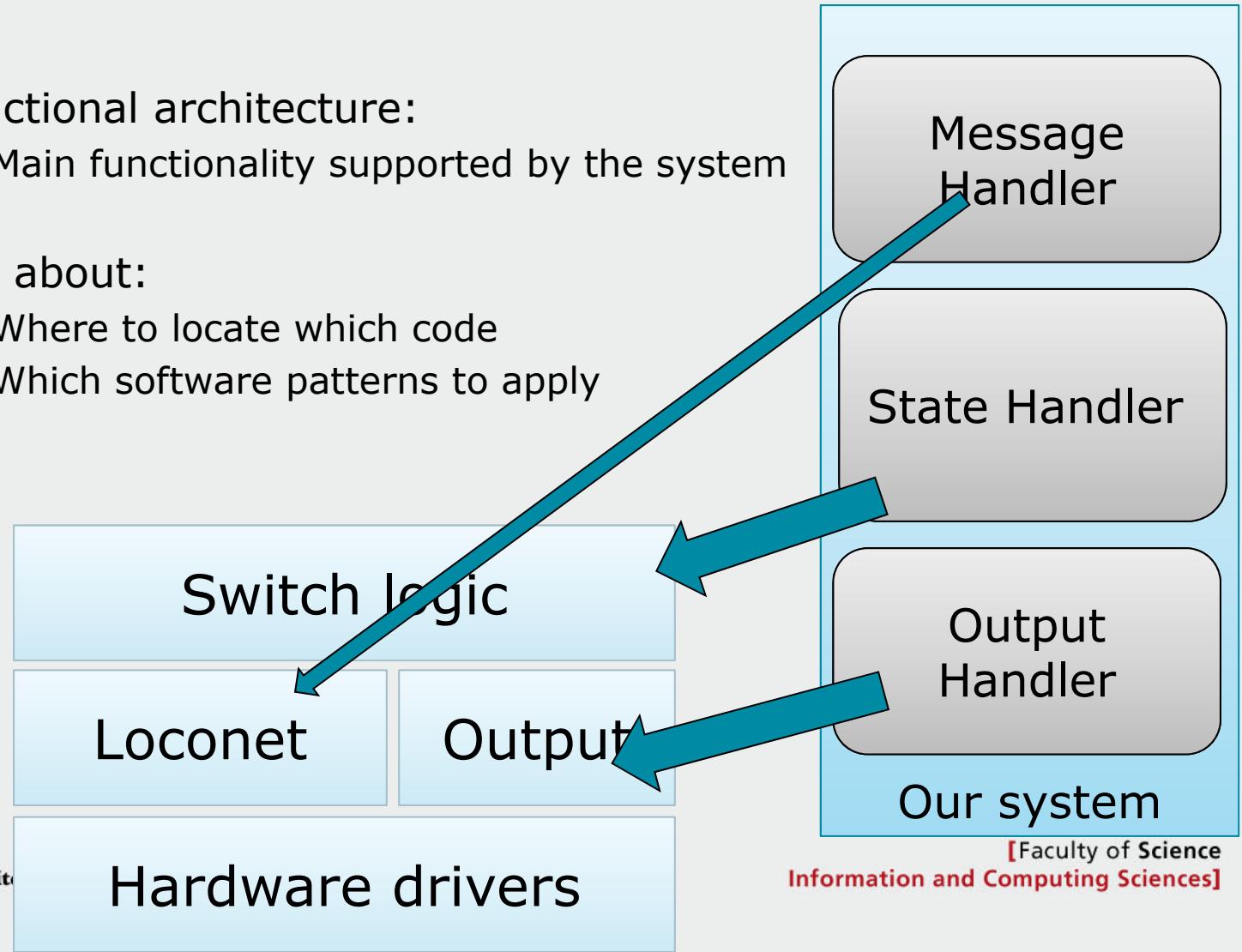
Classifying the different views...

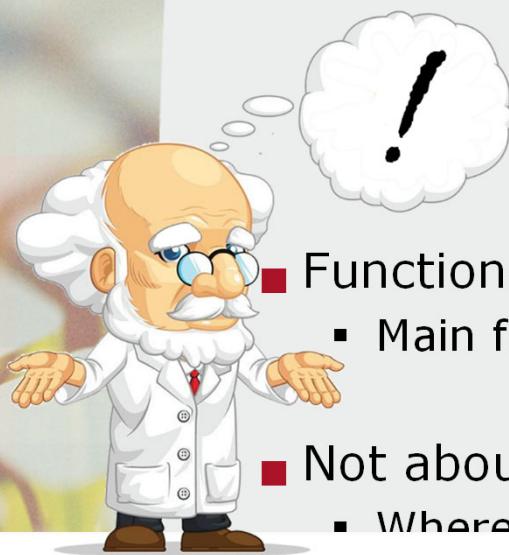


Where is the code of the system?

Software architecture & source code

- Functional architecture:
 - Main functionality supported by the system
- Not about:
 - Where to locate which code
 - Which software patterns to apply

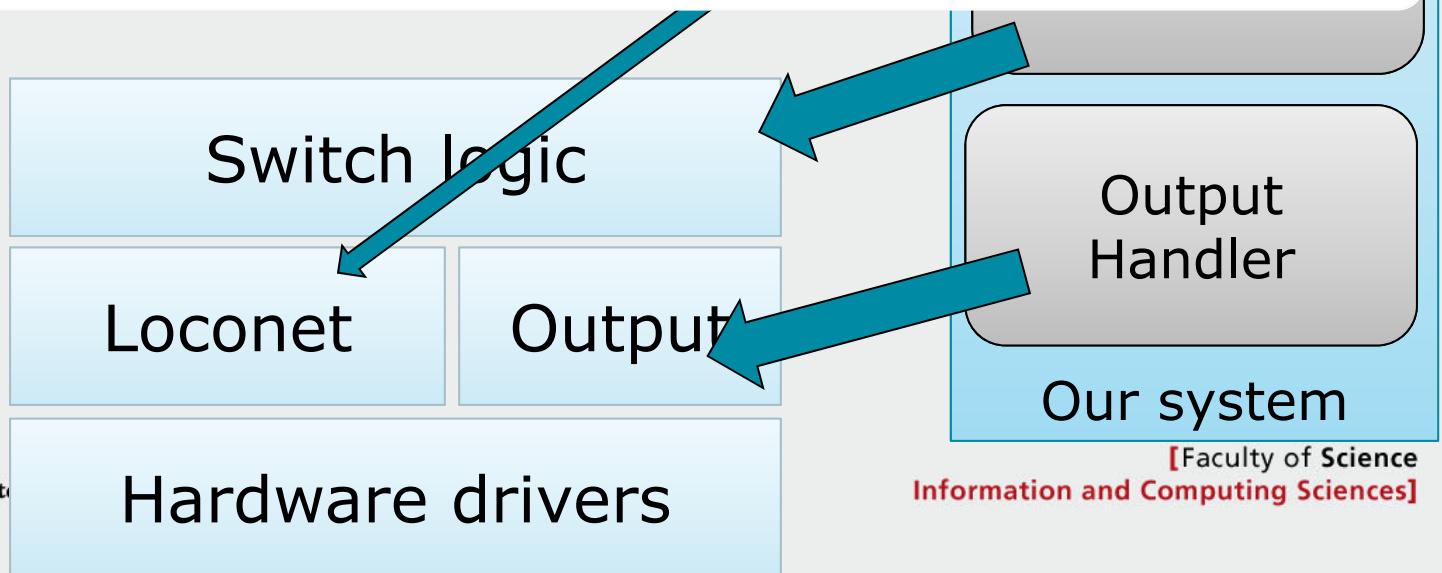
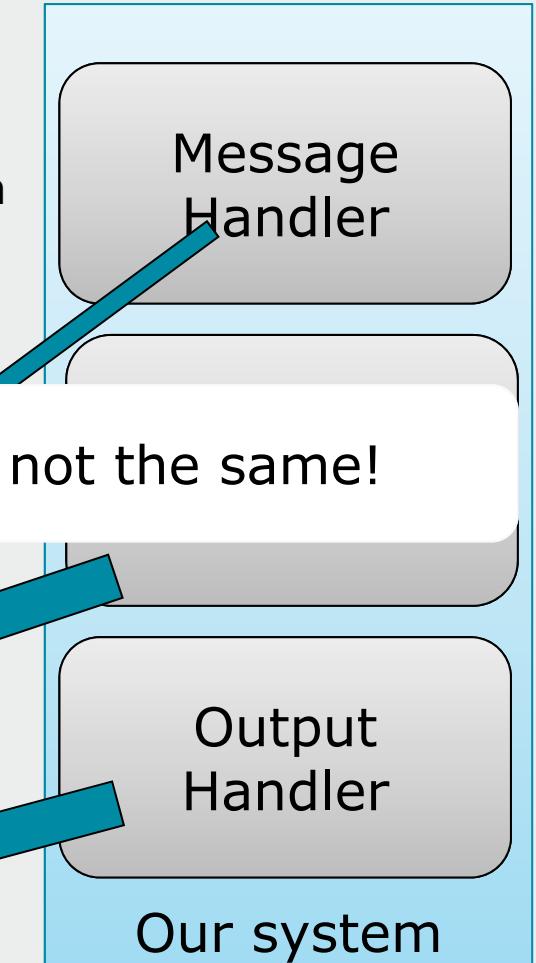




Software architecture & source code

- Functional architecture:
 - Main functionality supported by the system
- Not about:
 - Where to locate which code

Code structure & functional architecture are not the same!



Architecture: guidelines to follow

- Part of the Developer viewpoint:

- Code templates
- Programming conventions
- Frameworks...
- Development process (agile, SCRUM, testing, ...)

Convention: interrupts should be as short as possible...

Switch logic

Loconet

Output

Search: 2ms

Not in interrupt!

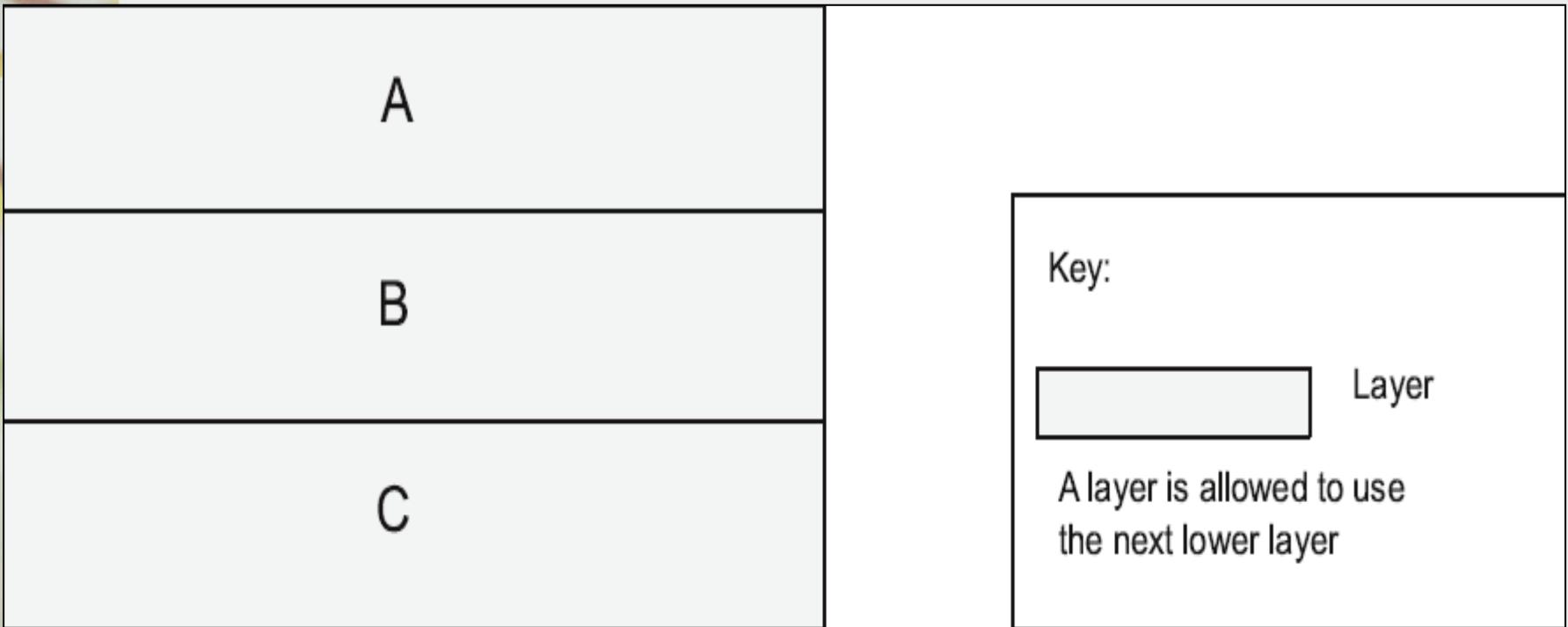
Convention: message storing in interrupt, message handling in main loop

Architectural styles & patterns

- Architectural style: fundamental structural organization schema for software systems
- Pattern: documents commonly recurring and proven structure
- Think of it as in language idiom: idiom describes the patterns of the language

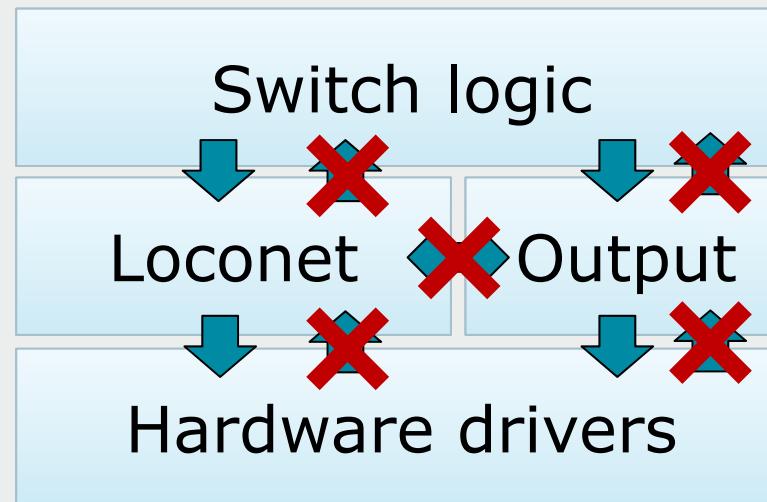


Layer pattern: topological layout



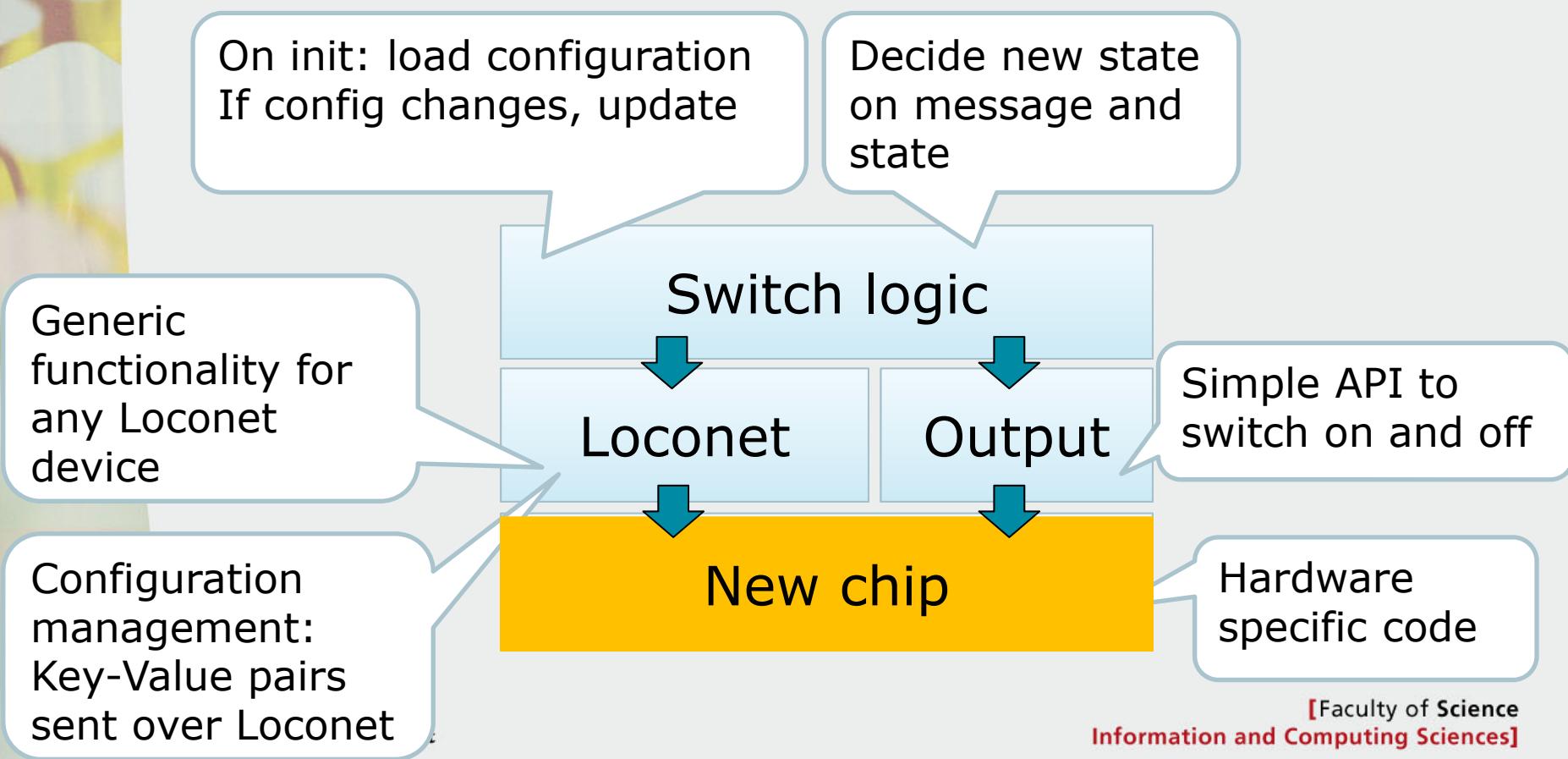
Layers: back to our example...

- Idea: lower layers do not know anything of higher layers



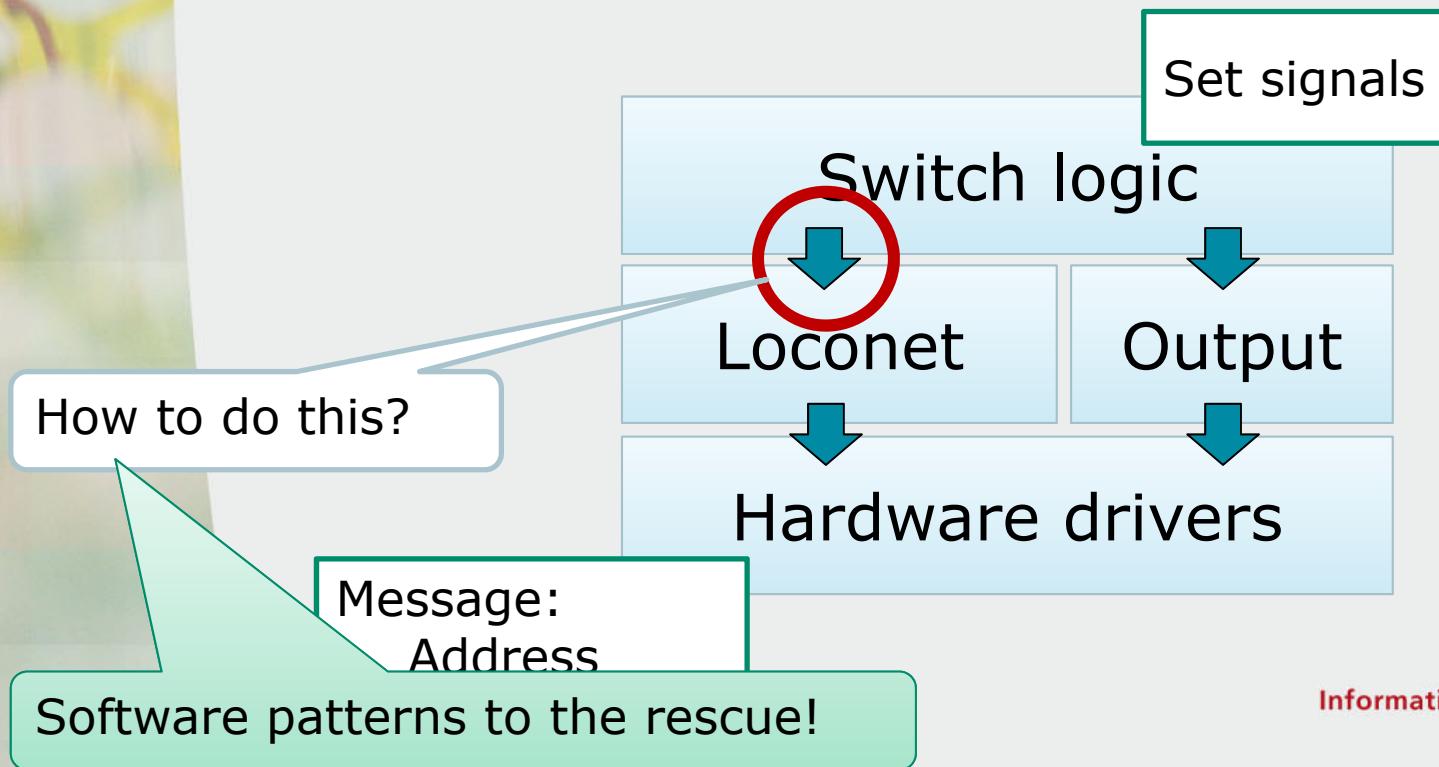
Layers: back to our example...

- Idea: lower layers do not know anything of higher layers



Layers: back to our example...

- Idea: lower layers do not know anything of higher layers



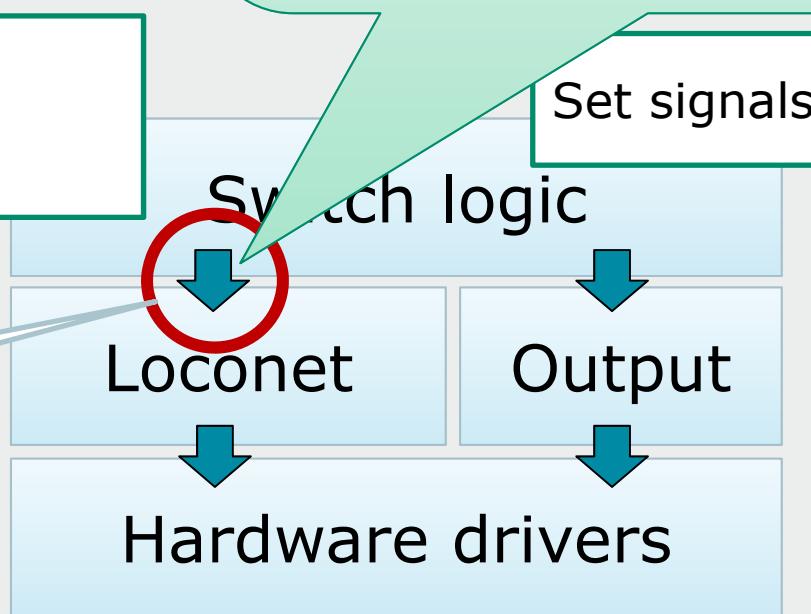
Layers:

- Idea: lower layers do

```
extern bool ln_register(uint8_t msgType, ln_fun  
fun*)  
  
extern bool ln_unregister(uint8_t ms, ln_fun  
fun*)  
  
bool ln_notify(uint8_t msgType, uint8_t msg*,  
uint8_t ln) {  
    // notify all registered functions...  
}
```

Message:
- Address
- Value(s)

How to do this?



Software patterns to the rescue!

Software architecture & source code

Part of architecture?

Part of software?

Mapping

Architecture artifacts

Realization artifacts

How to check?



How to keep those consistent?

Faculty of Science
[Computer Sciences]

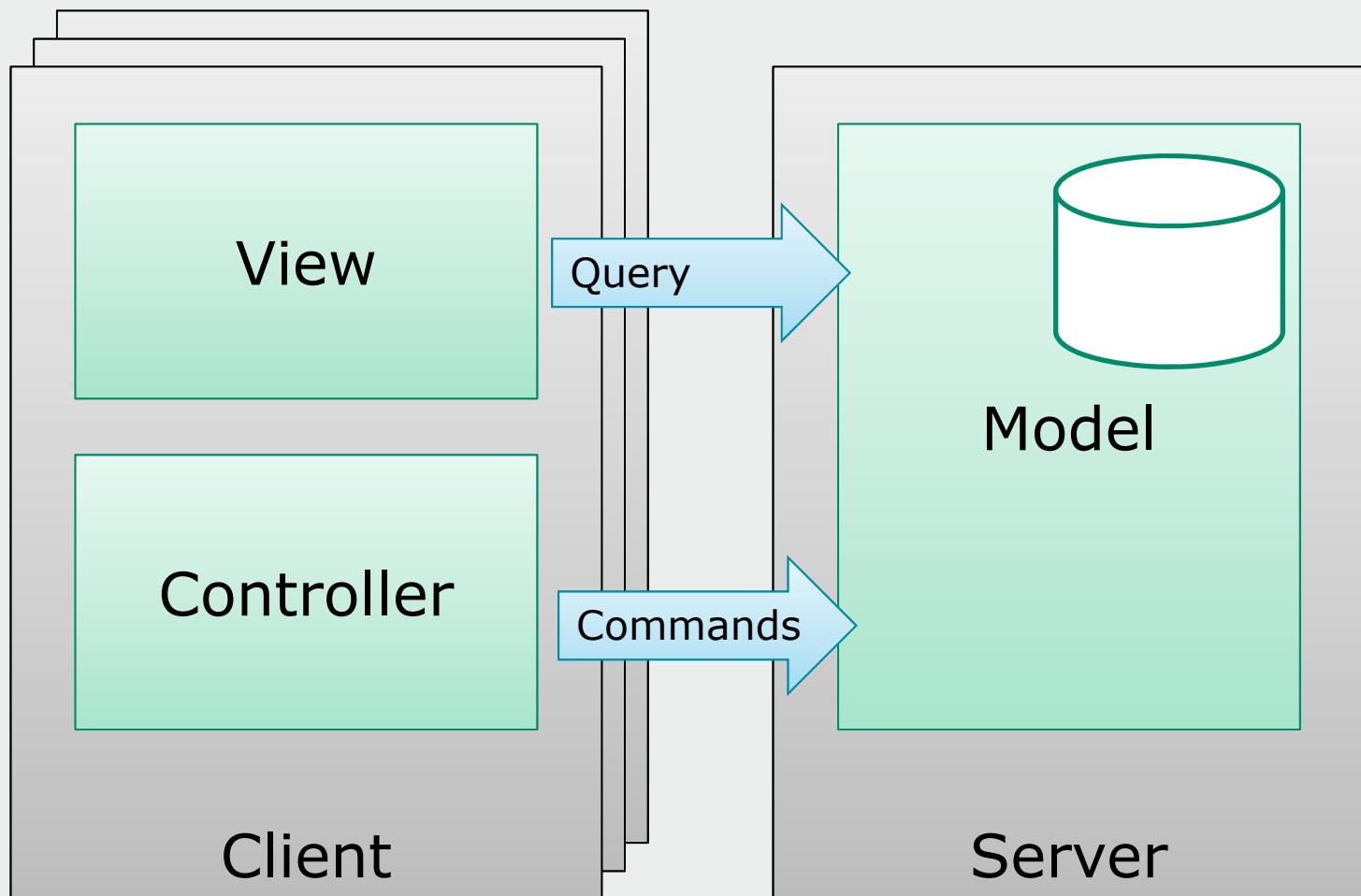


Universiteit Utrecht

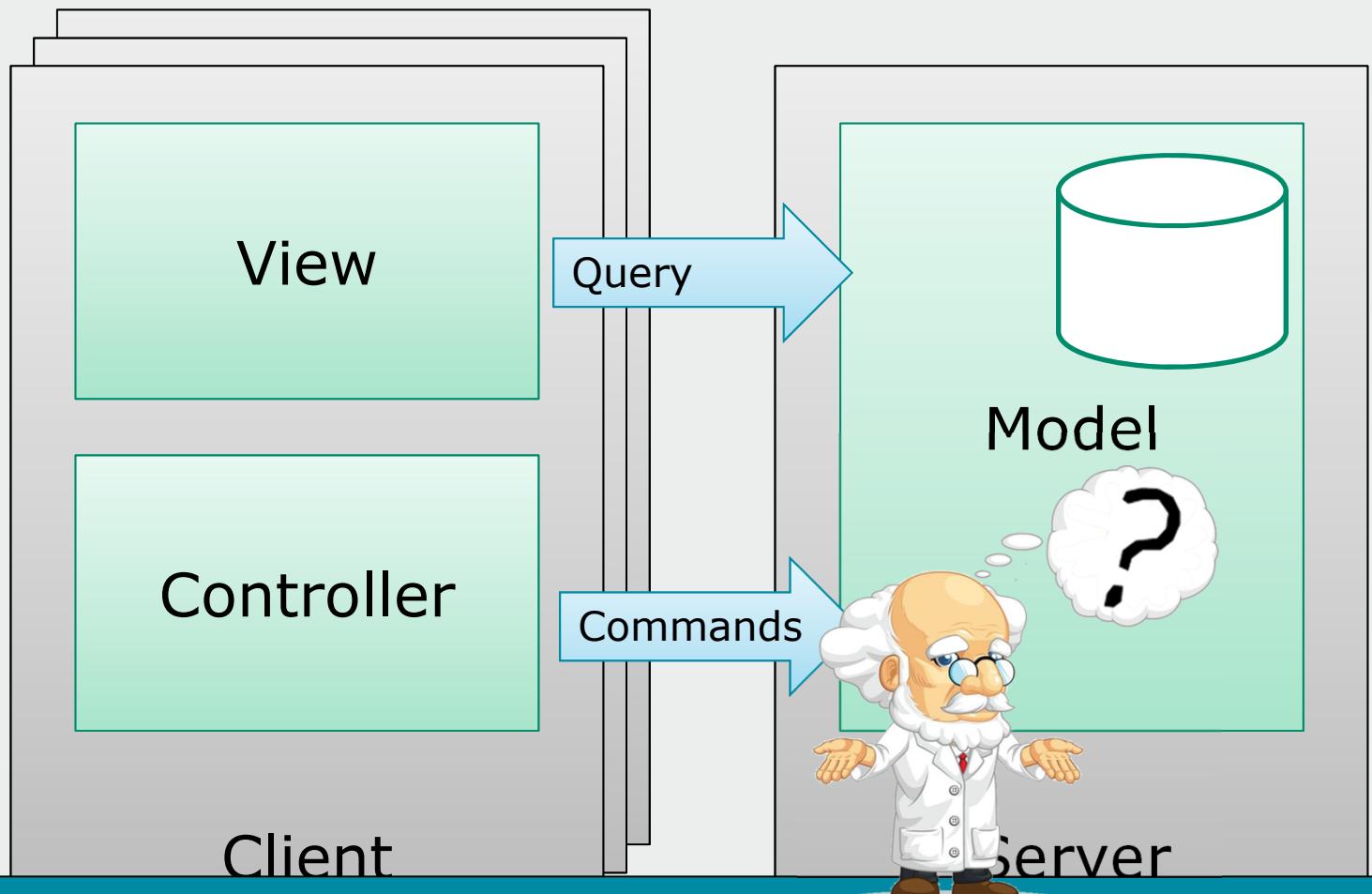
[Faculty of Science
Information and Computing Sciences]

Software architecture & patterns

Model-View-Controller

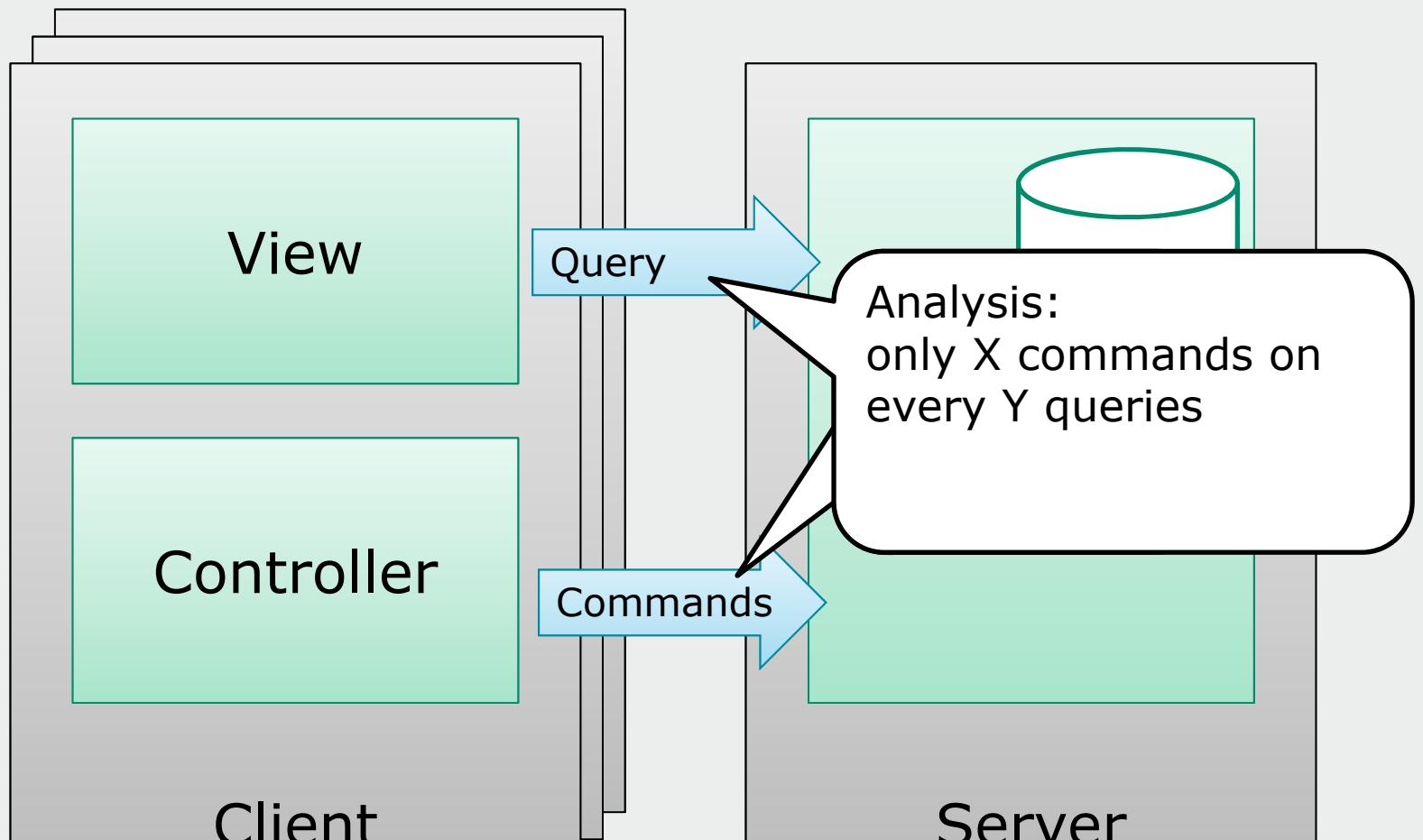


Model-View-Controller



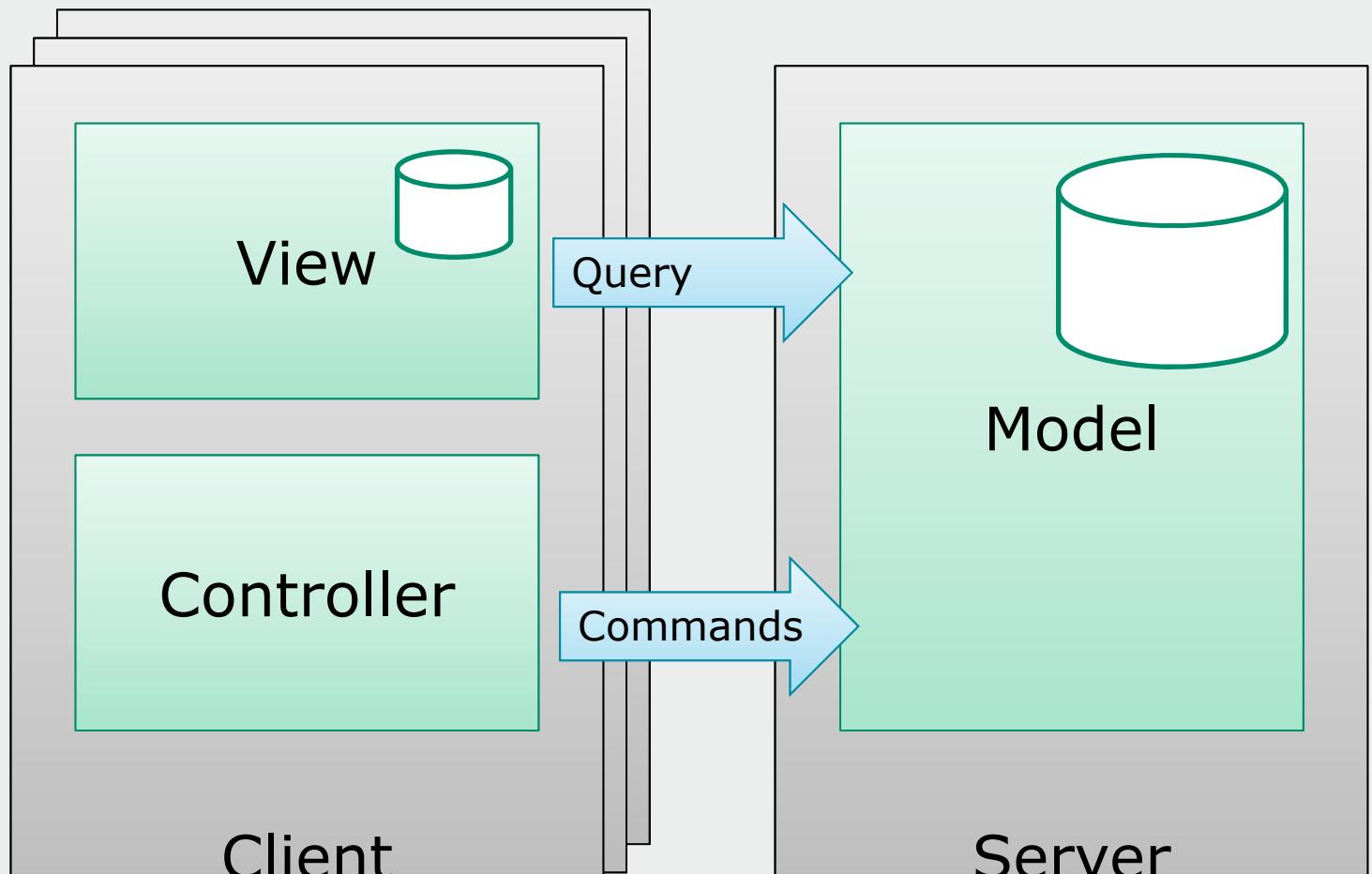
What if there are 10.000 clients running concurrently?

Model-View-Controller



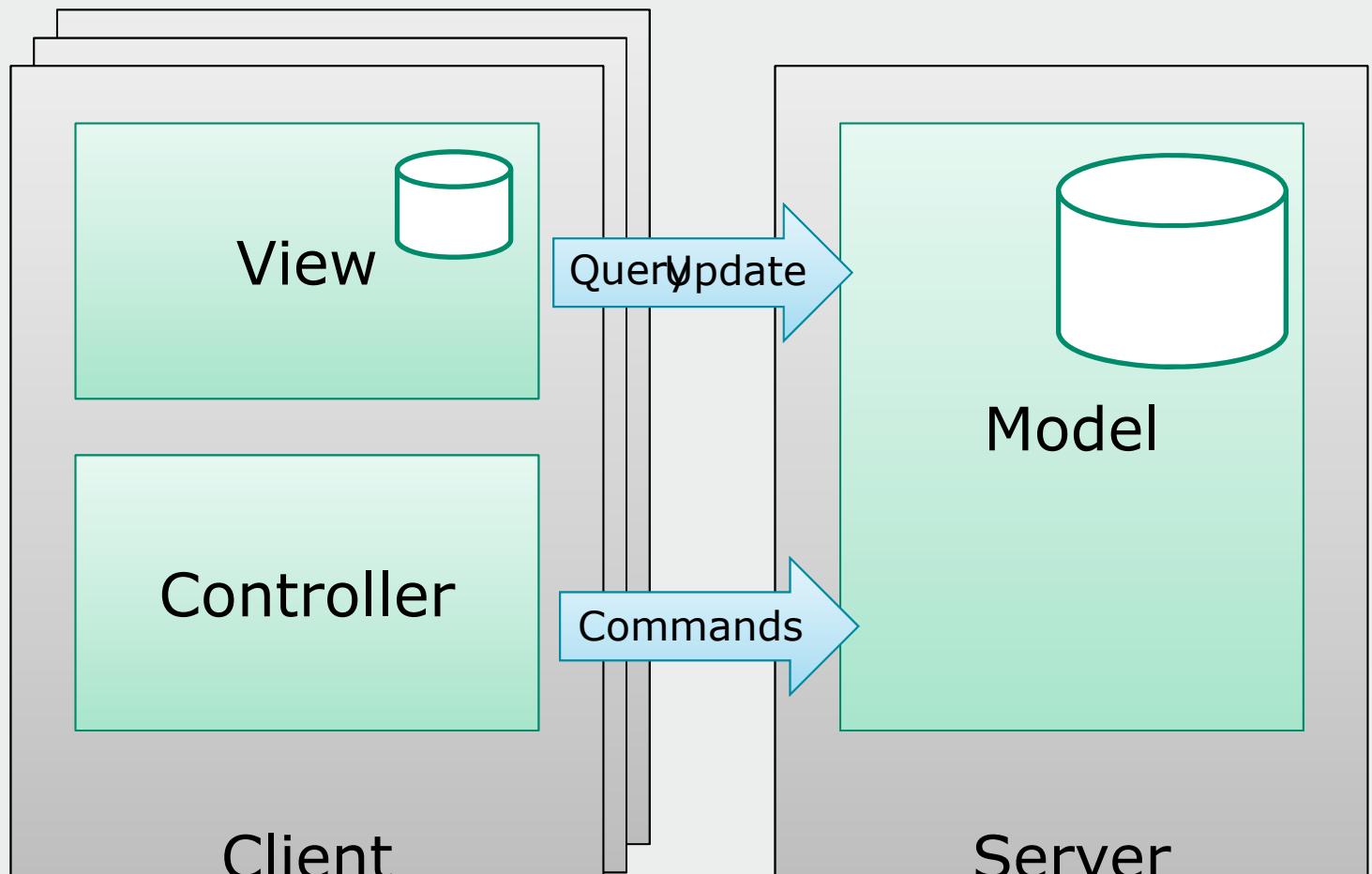
What if there are 10.000 clients running concurrently?

Model-View-Controller



What if there are 10.000 clients running concurrently?

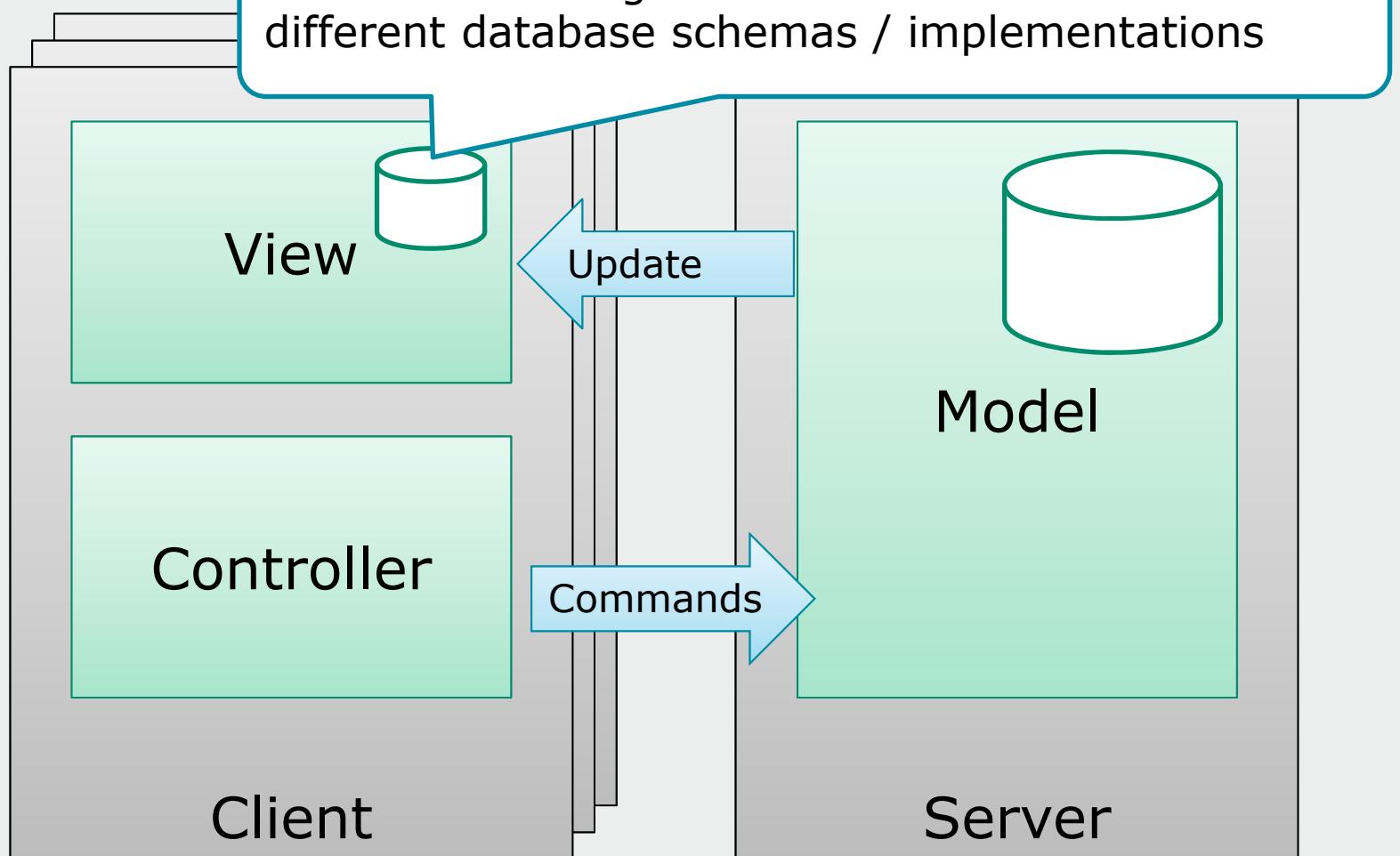
Model-View-Controller



What if there are 10.000 clients running concurrently?

CQRS: Command Query Responsibility Segregation

Another advantage: different views can have different database schemas / implementations

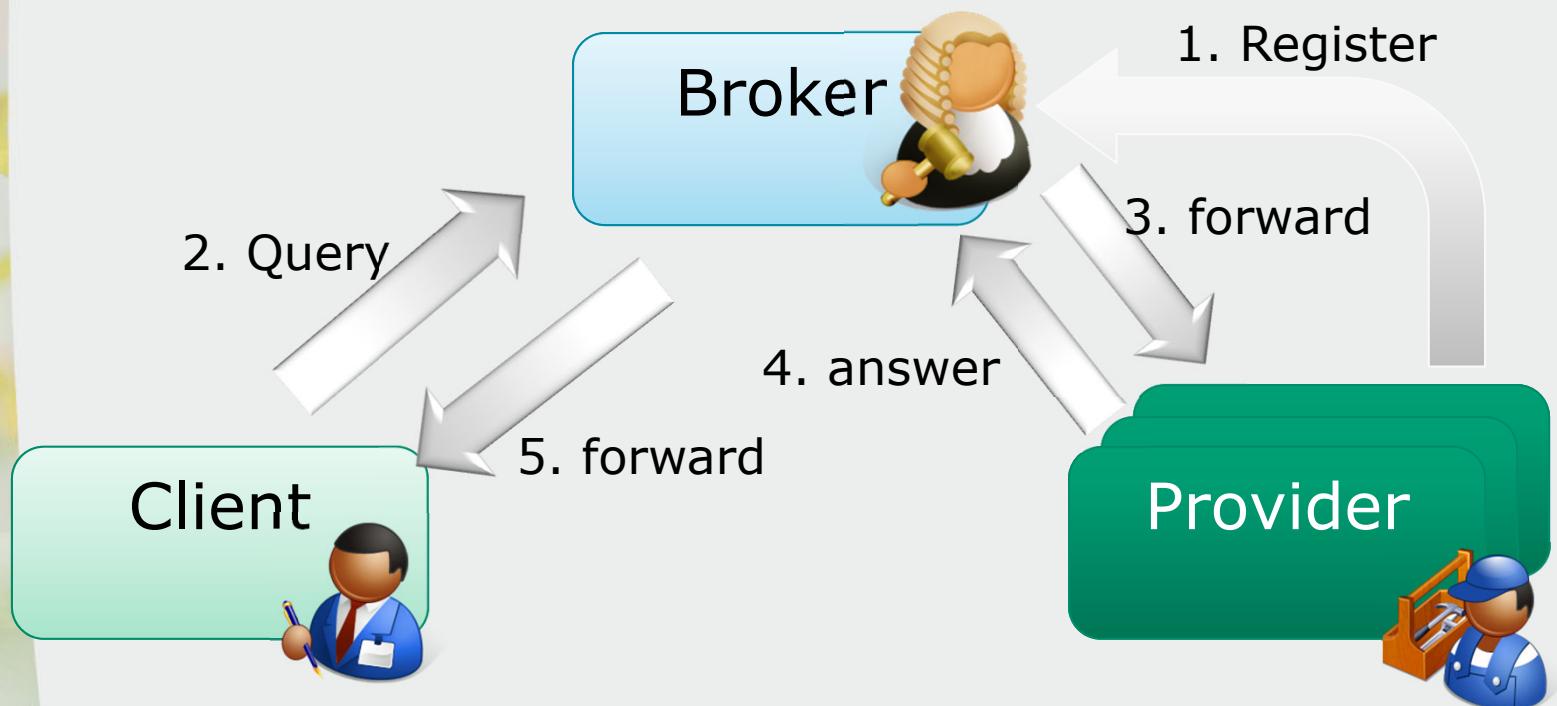


How would you solve the following problem?

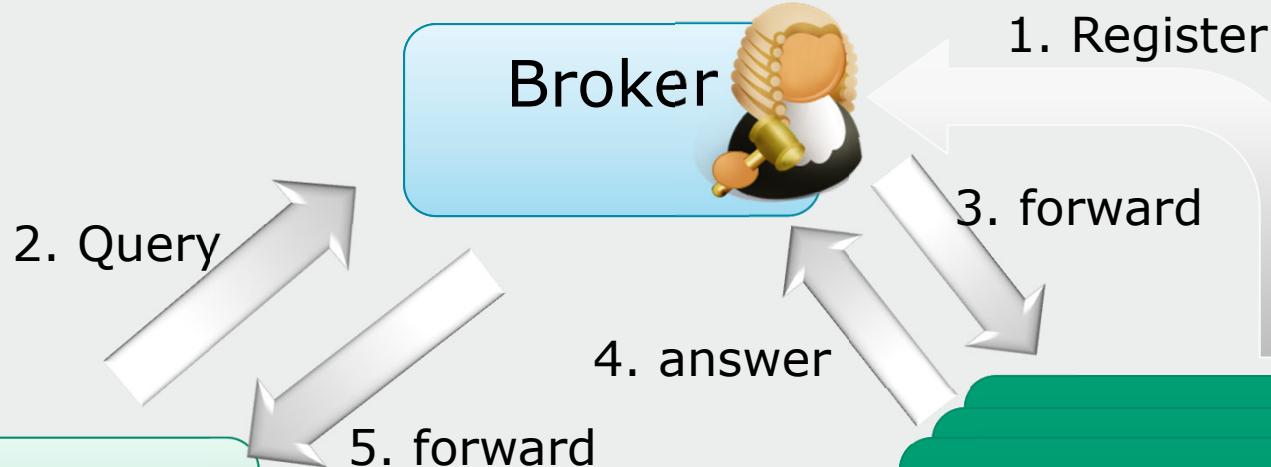
- Many clients
- Many providers delivering a service
 - Some providers may deliver the same service
 - Runtime providers can come and go
- Clients can use the providers' services



Broker pattern



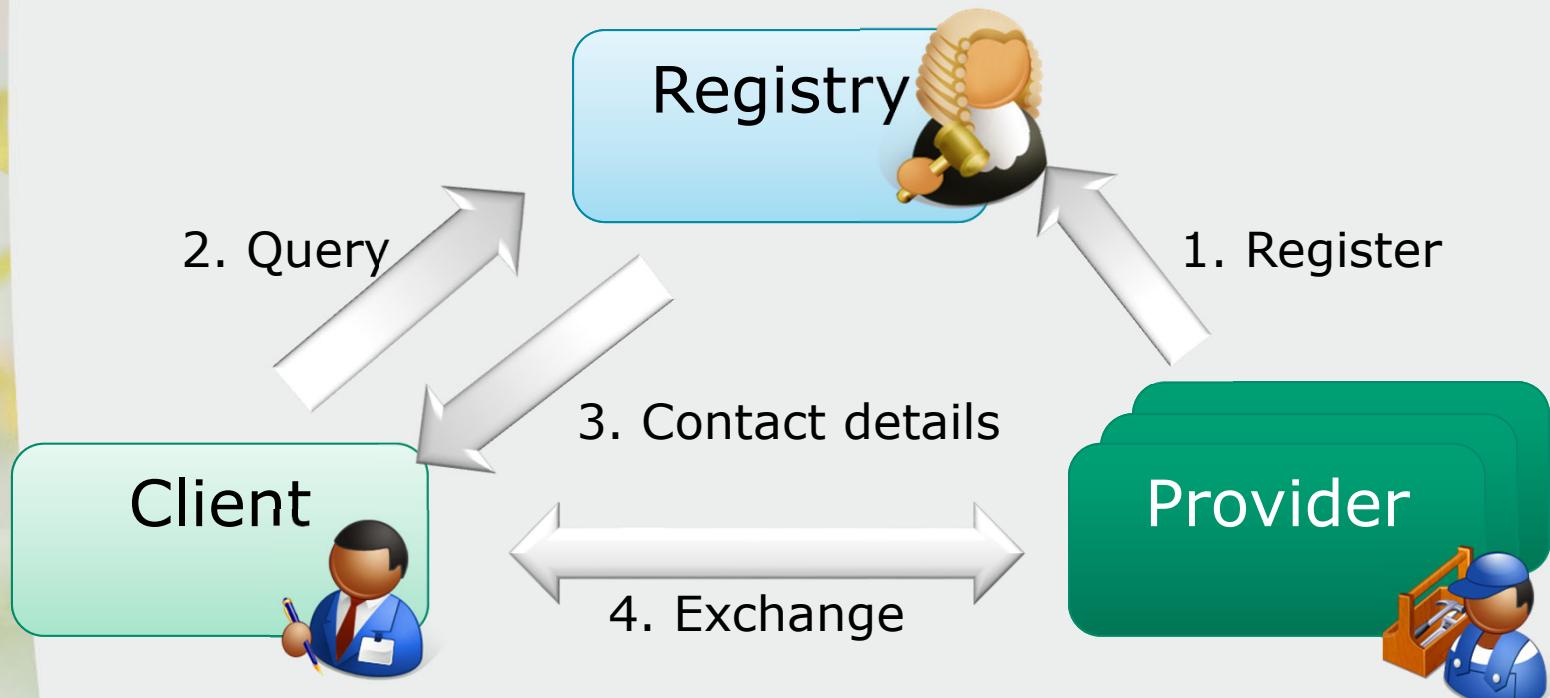
Broker pattern



What is the disadvantage of this pattern?

re
s]

Service oriented architecture





Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

How do you benefit from patterns?

Consider the following situation

- A crowd simulation tool with editing functionality
- Edit operations
 - Design time: to construct a world with obstacles
 - Run time: add obstacles & “agents”
- Simulation of agents
 - Notebook: max 3k agents
 - “Large server”: max 10k agents
- Architect a system that can handle 3M agents **real time**
 - Cloud-based
 - Multi-tenant
 - Different viewers (Unity, web, ...)





Questions for the architect

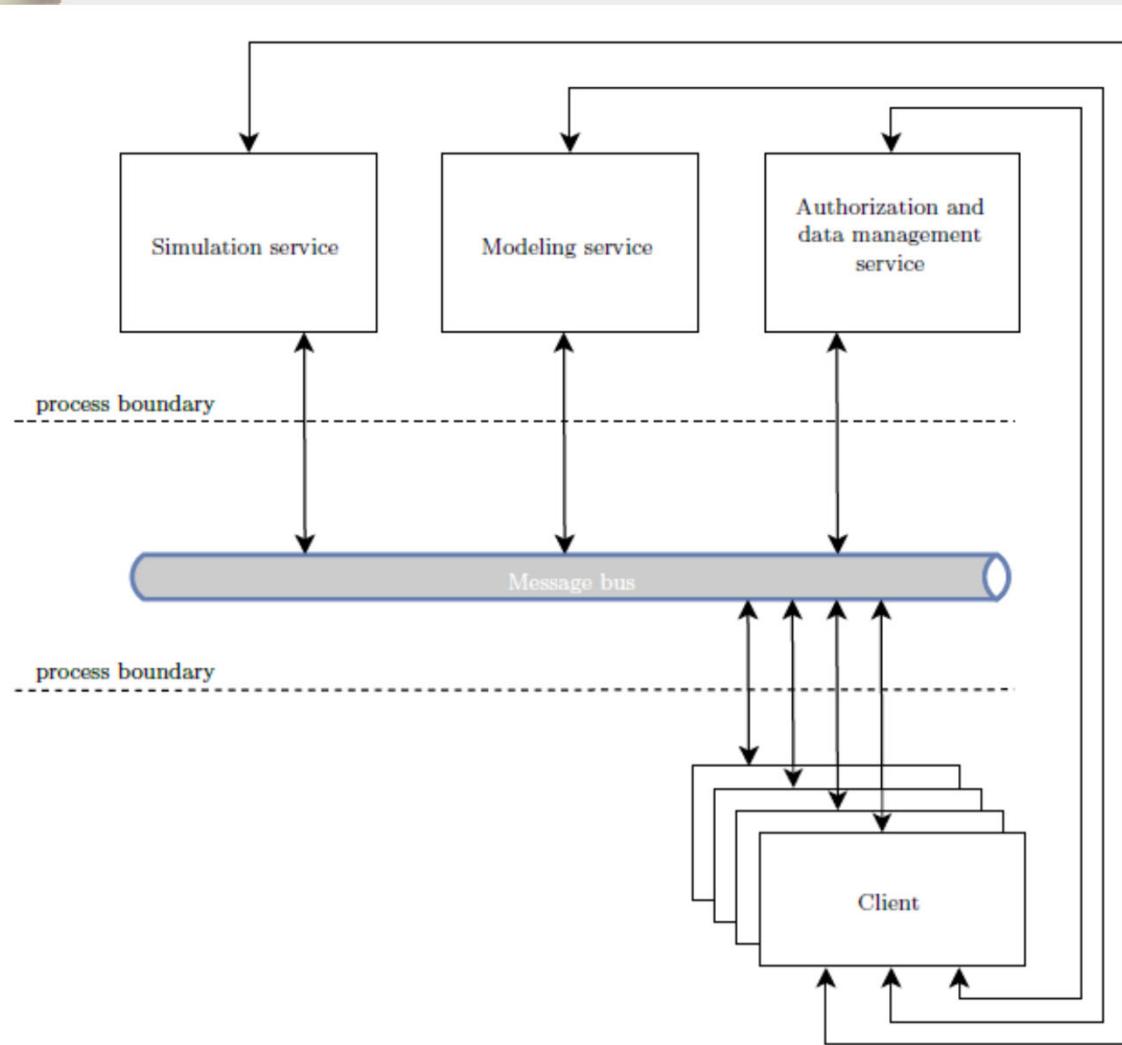
- How many agents can the system handle real time?
 - What is the performance?
 - How scalable is the solution?
 - How large should a deployment be?
- What will be the business model?
- Intended uses?
 - Real time prediction & forecasting?
 - Upfront simulation analysis?
 - ...



Let's architect a system ☺

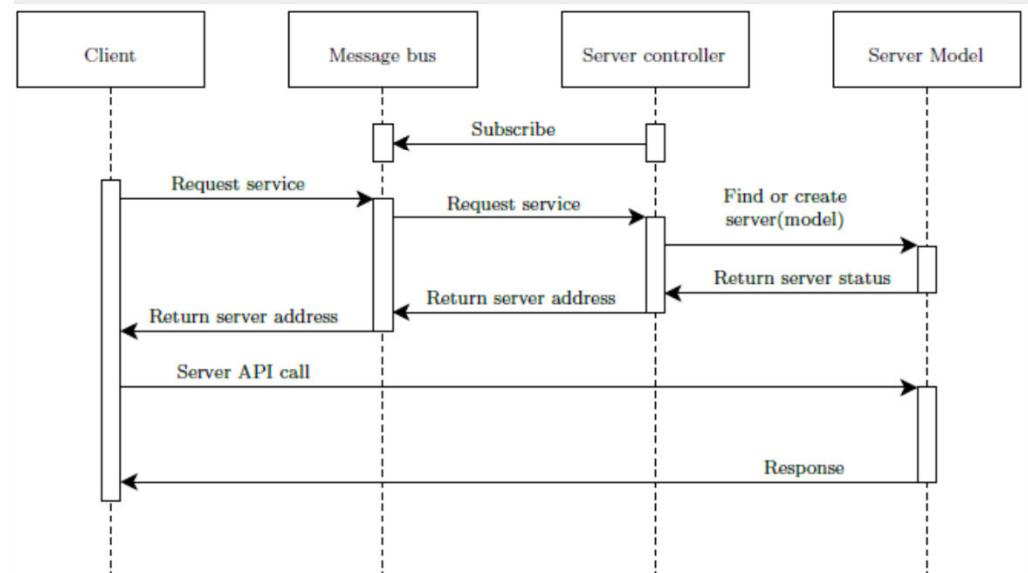
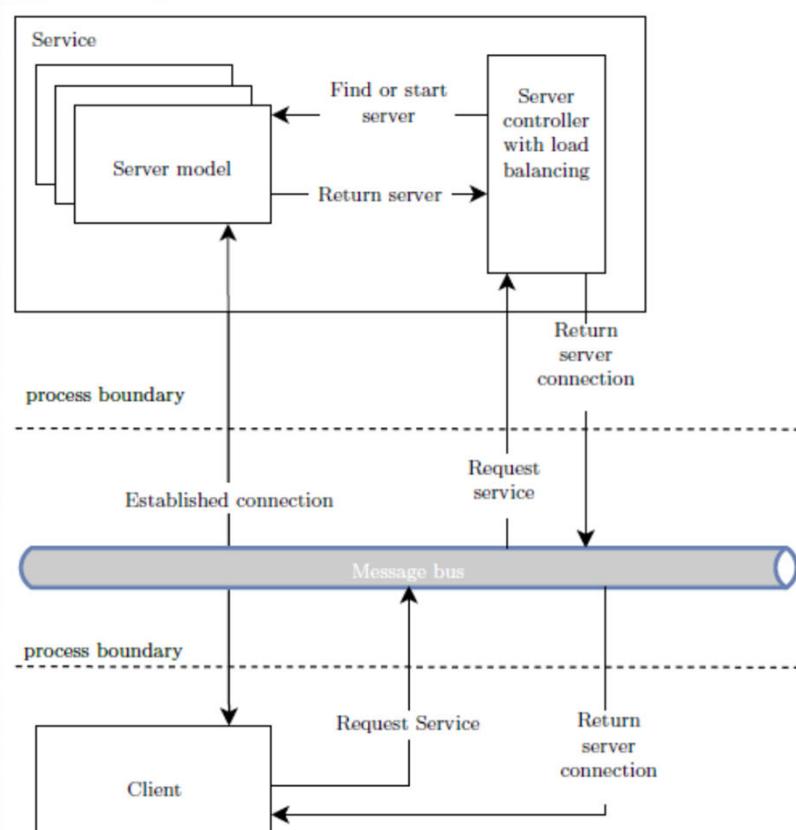
ce
s]

Our solution...



[Faculty of Science
Information and Computing Sciences]

With our own defined pattern



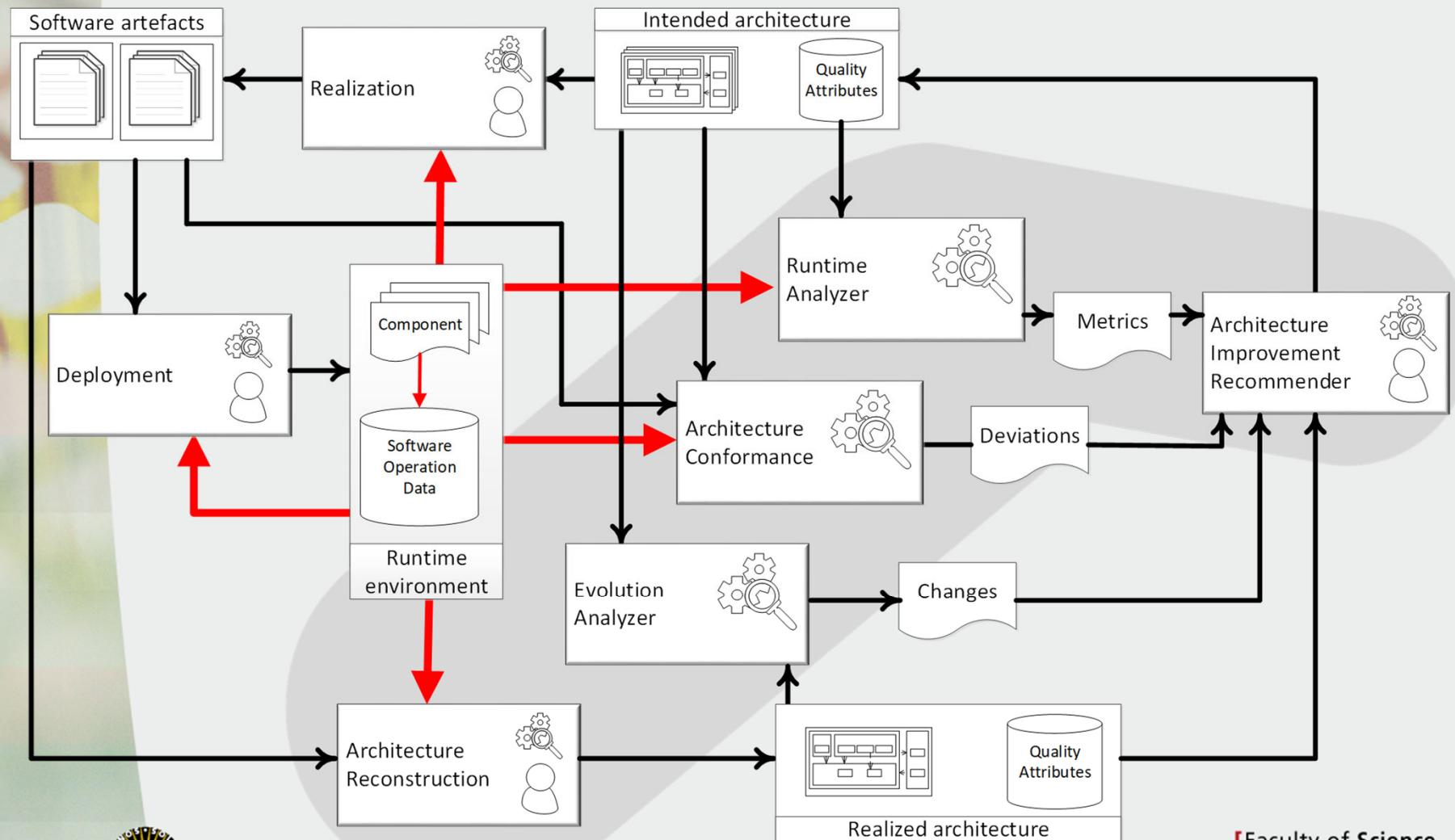


Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

To summarize

Software Architecture: overview of the field



Software architecture

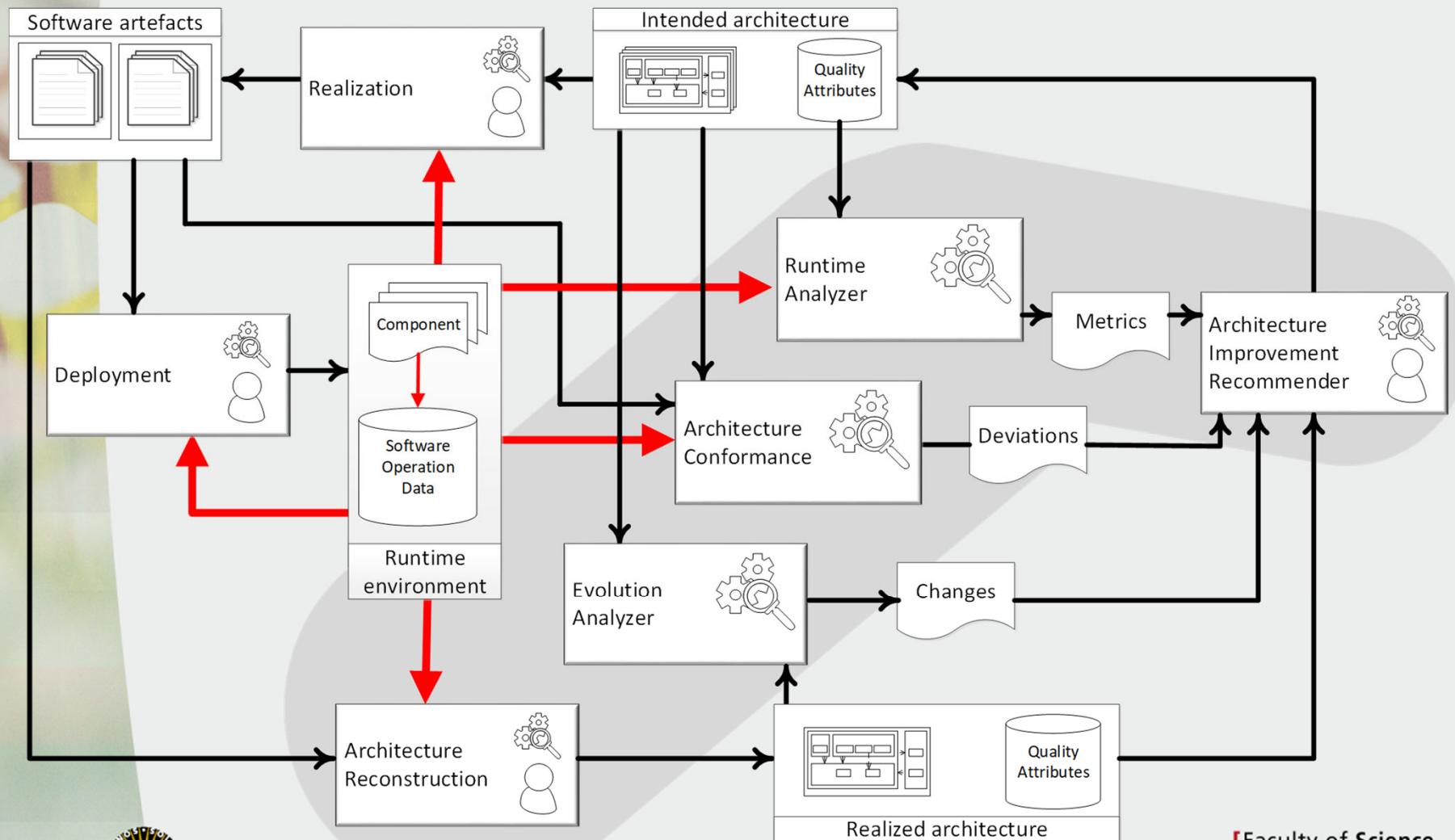
- The bigger picture of organizing your software
 - Software design ≠ software architecture
- Two important definitions:
 - Set of structures comprising software elements, their relations and properties to reason over a system
 - Composition of the set of principle design decisions
- Trade-off analysis of requirements
 - Functionality
 - Characteristics ("ilities")



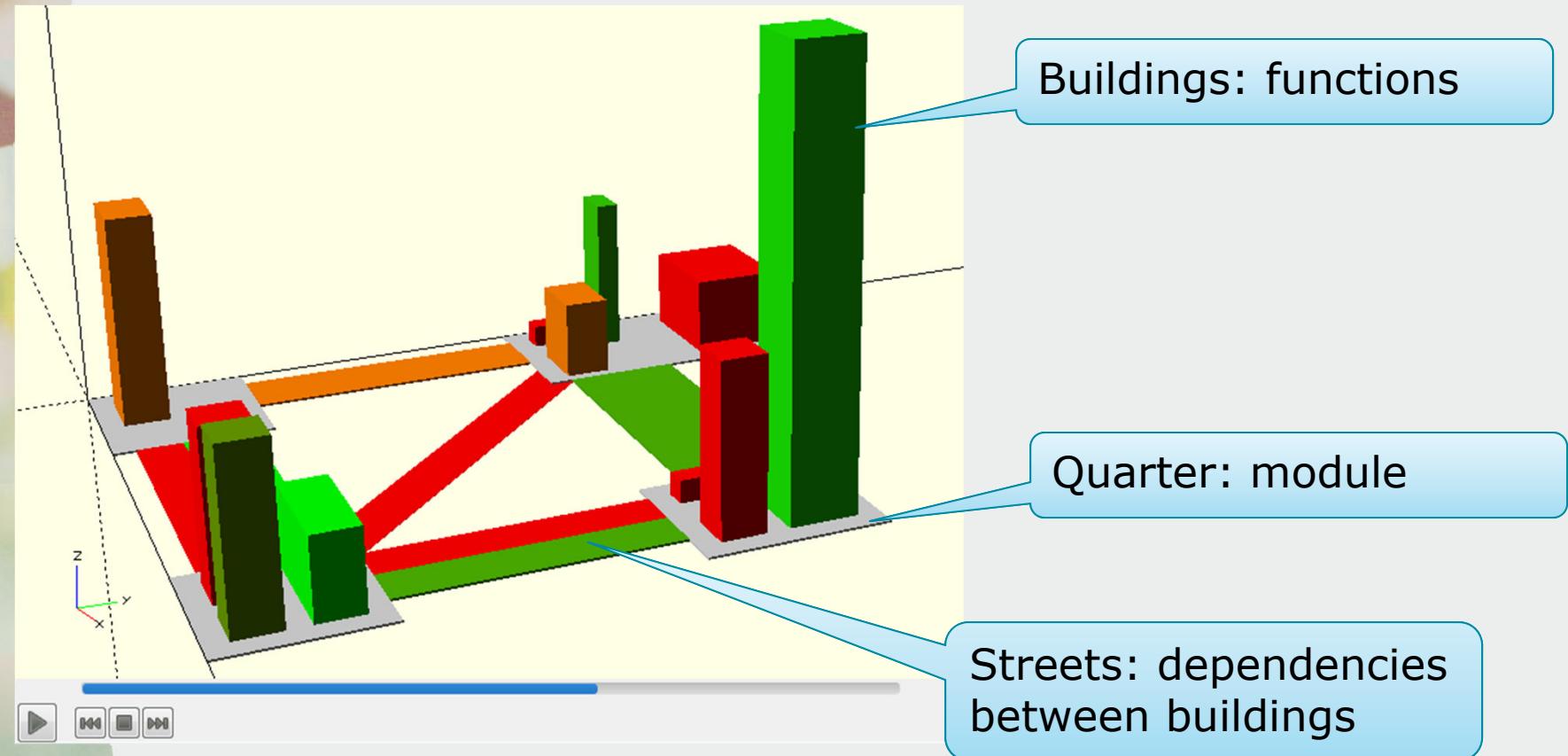
Document your design decisions!

Science
sciences]

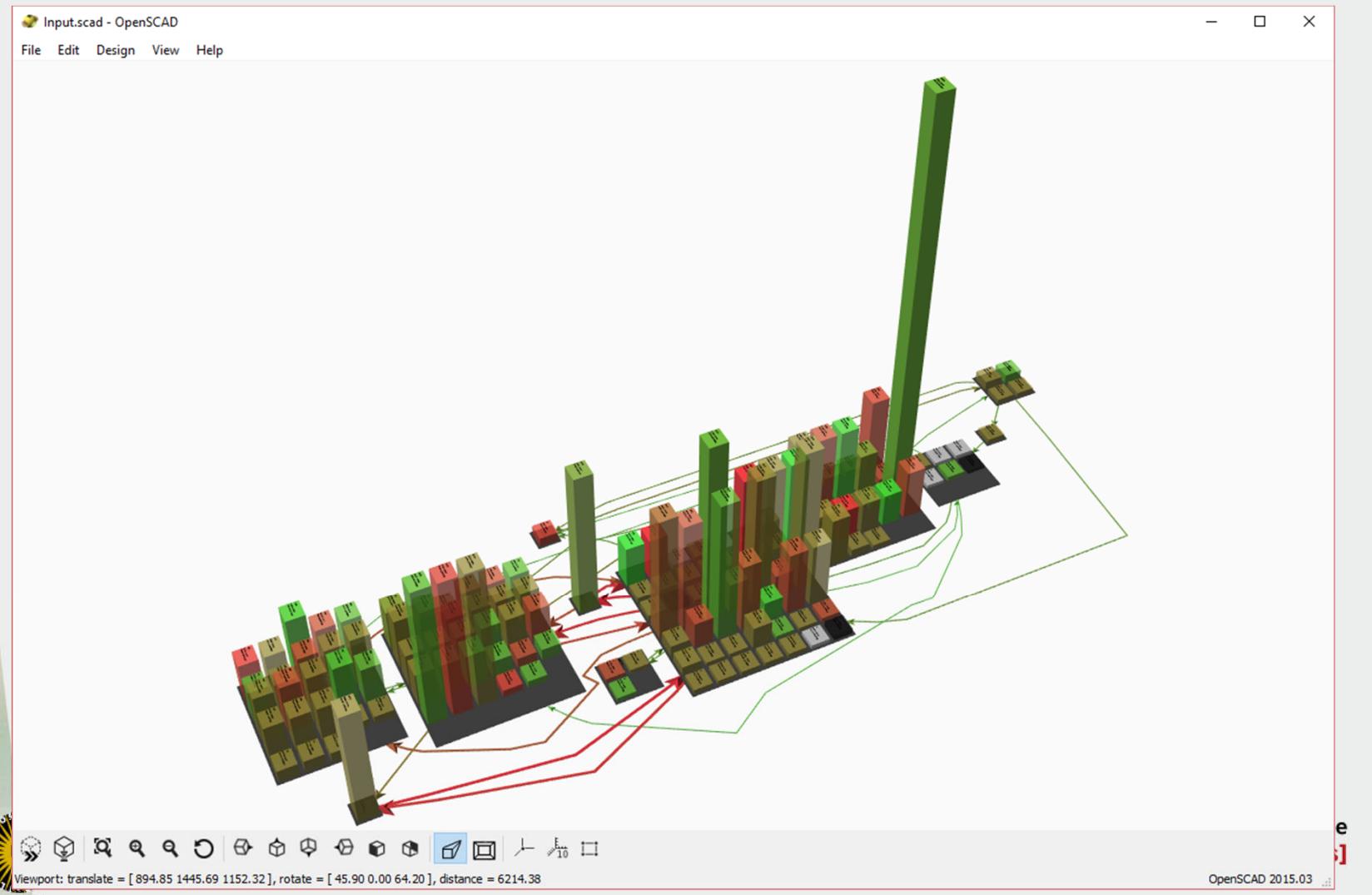
Software Architecture: overview of the field



ArchitectureCity (MSc-project Rens Rooimans)

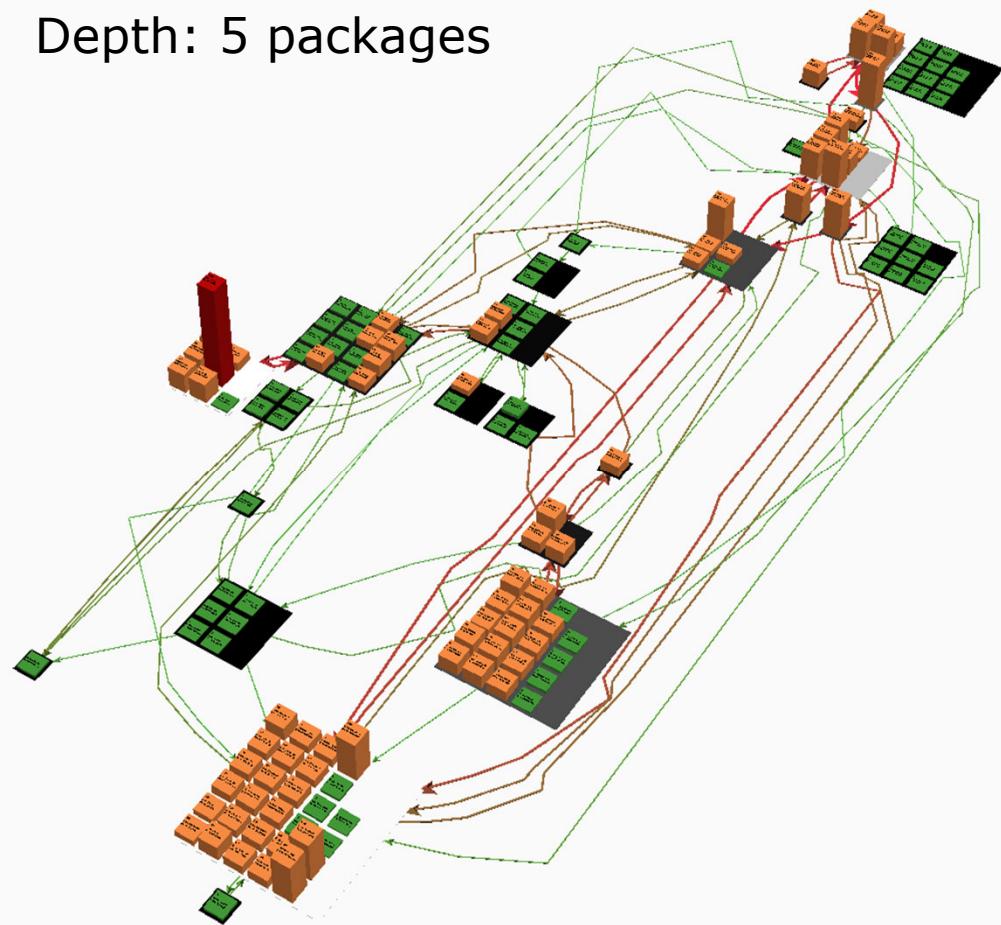


ArchitectureCity (MSc-project Rens Rooimans)

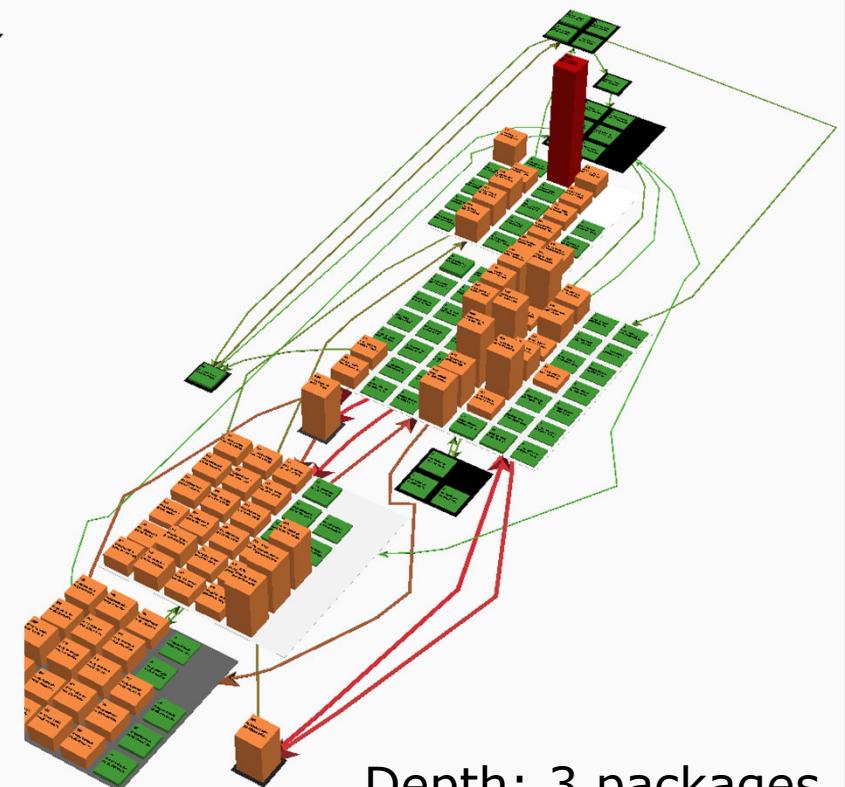


ArchitectureCity (MSc-project Rens Rooimans)

Depth: 5 packages

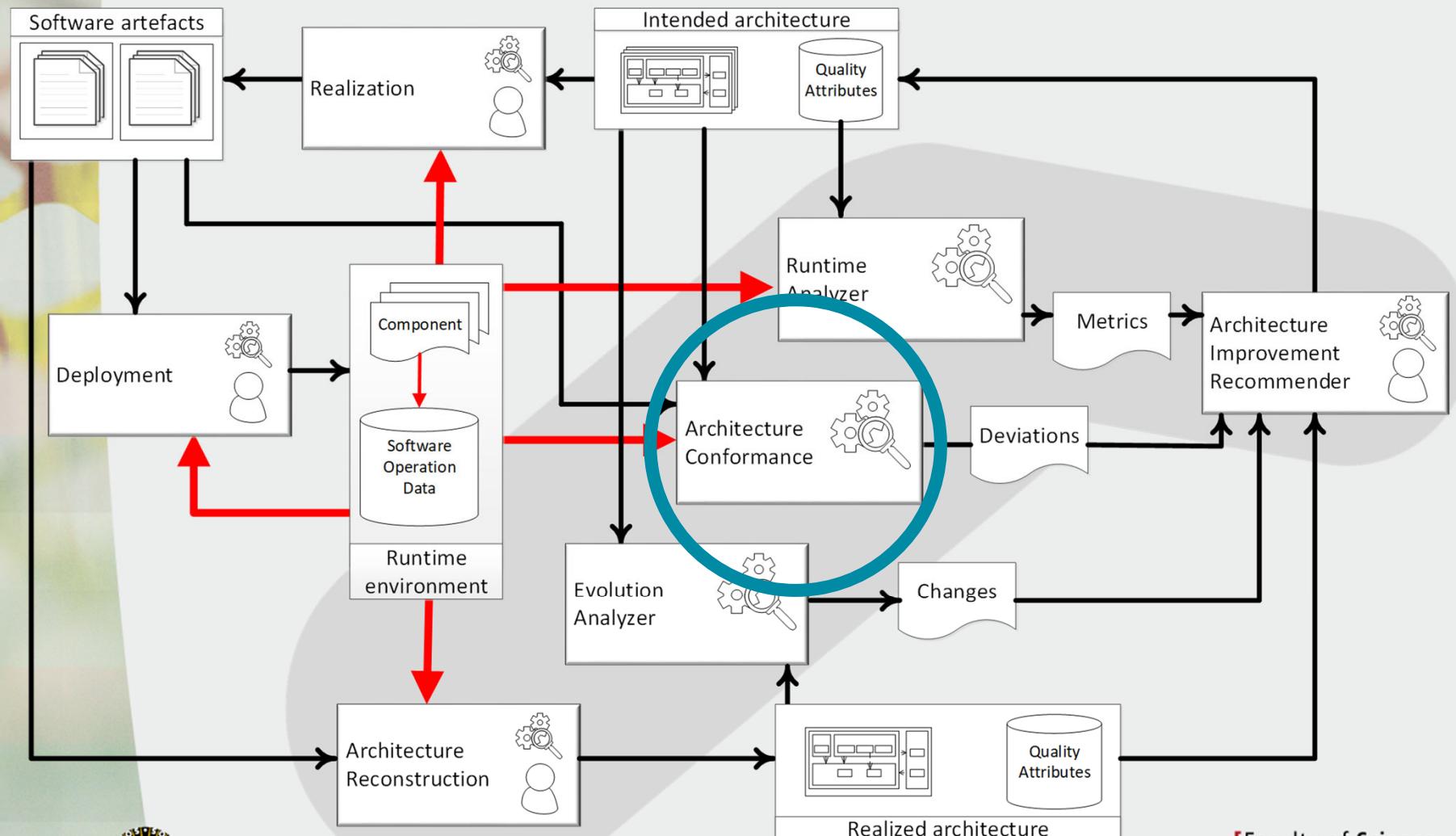


Depth: 3 packages



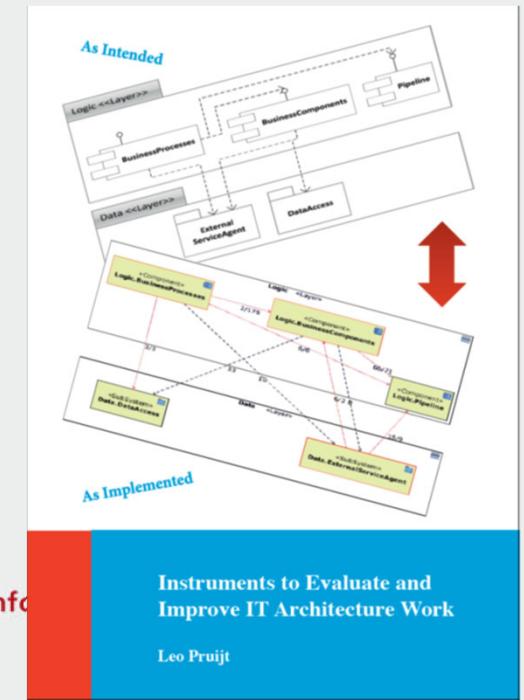
[Faculty of Science
Information and Computing Sciences]

Software Architecture: overview of the field

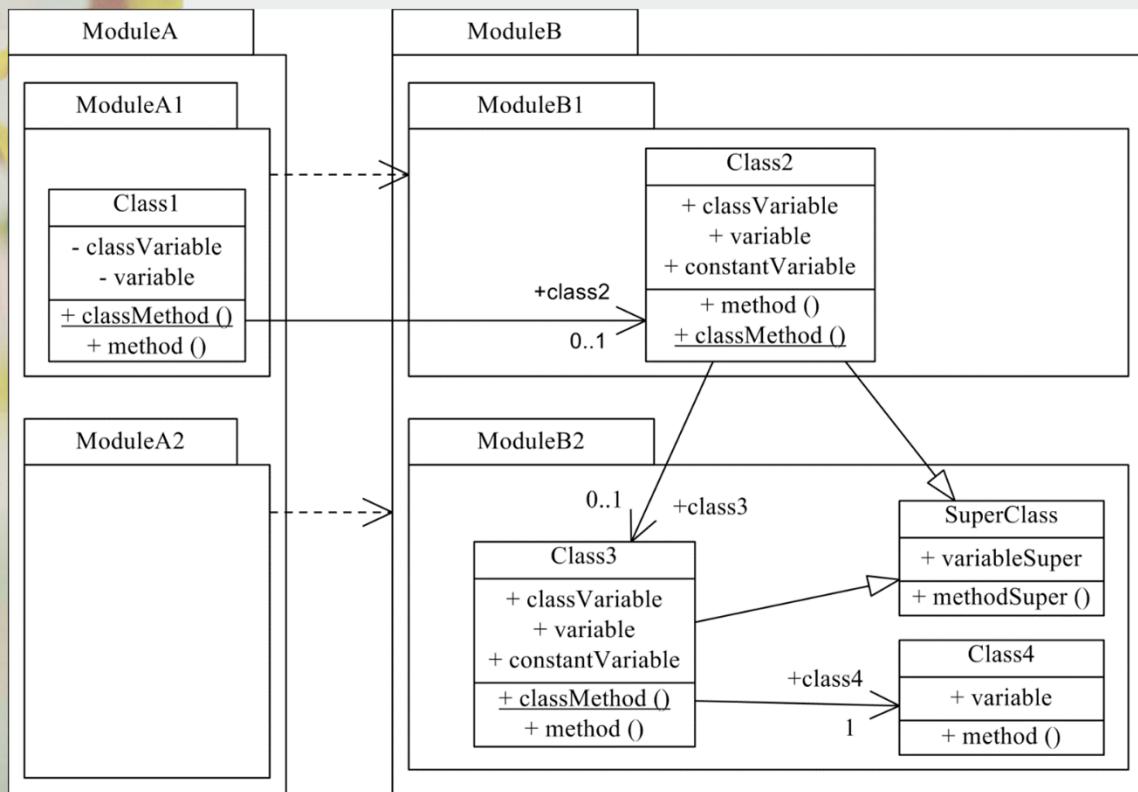


Software Architecture Conformance Checking (SACC) (PhD-topic of Dr. Leo Pruijt)

- SACC is an approach to bridge the gap between the high-level models of architectural design and the implemented program code, and to prevent architectural erosion.
- **Architecture conformance** is “a measure to which degree the implemented architecture in the source code conforms to the planned software architecture”.



Example: Intended Architecture



```
import ModuleB.ModuleB2.Class3;

public class Class1
    extends SuperClass {

    /** @var Class3 */
    private Class3 class3;
    /** constructor */
    public Class1 {
        class3 = new Class3();
        int a = class3.method();
        int b = class3.variable();
    }
}
```

Class1.java



Software Architecture: overview of the field

