Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Software Architecture
(Software-ontwikkelmethoden en verder...)

Jan Martijn van der Werf

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

A (too) short introduction to Software Architecture

Jan Martijn van der Werf

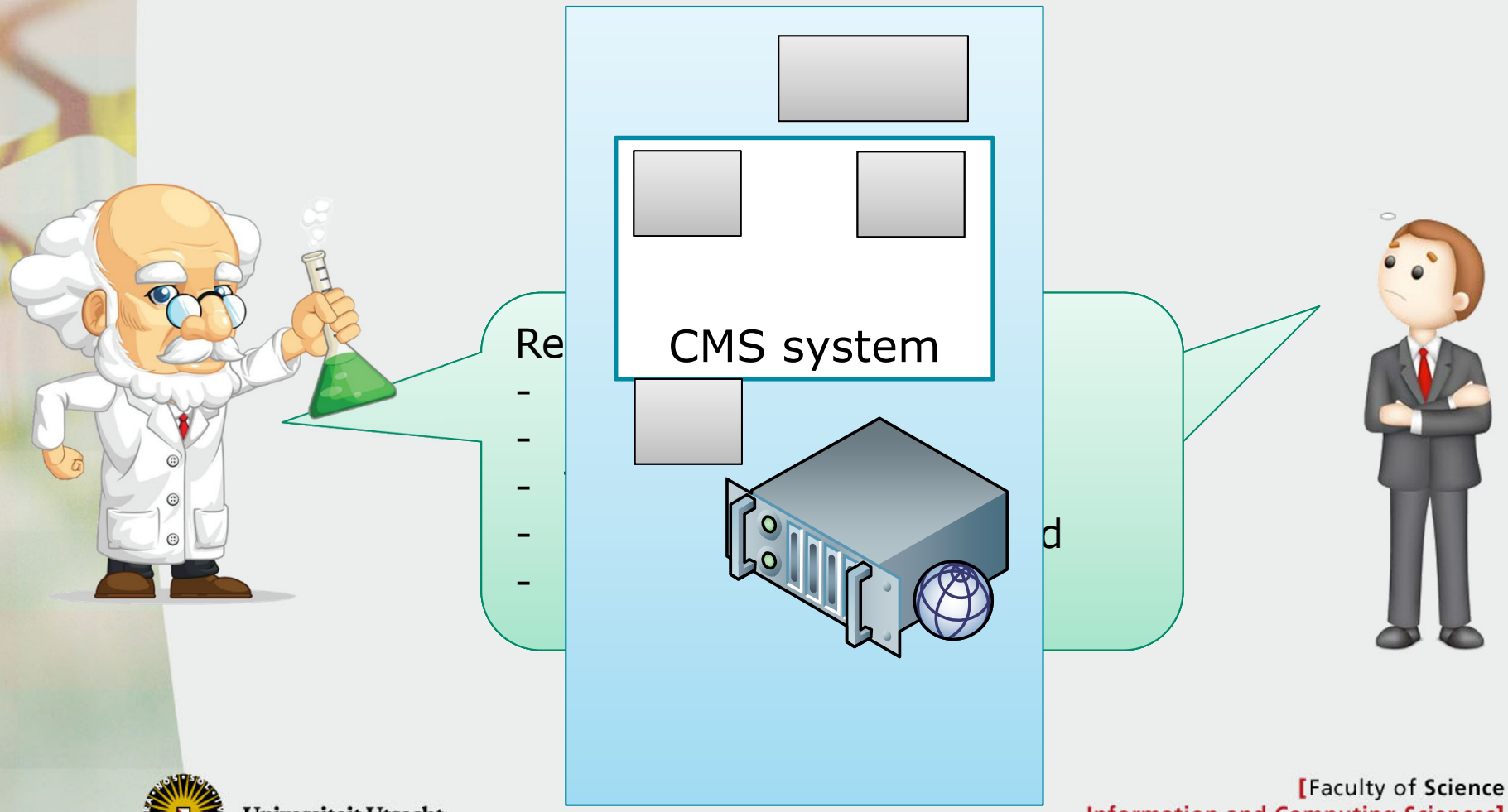
Building a software system

I need an online ticket sale for my conference

That is not too difficult…

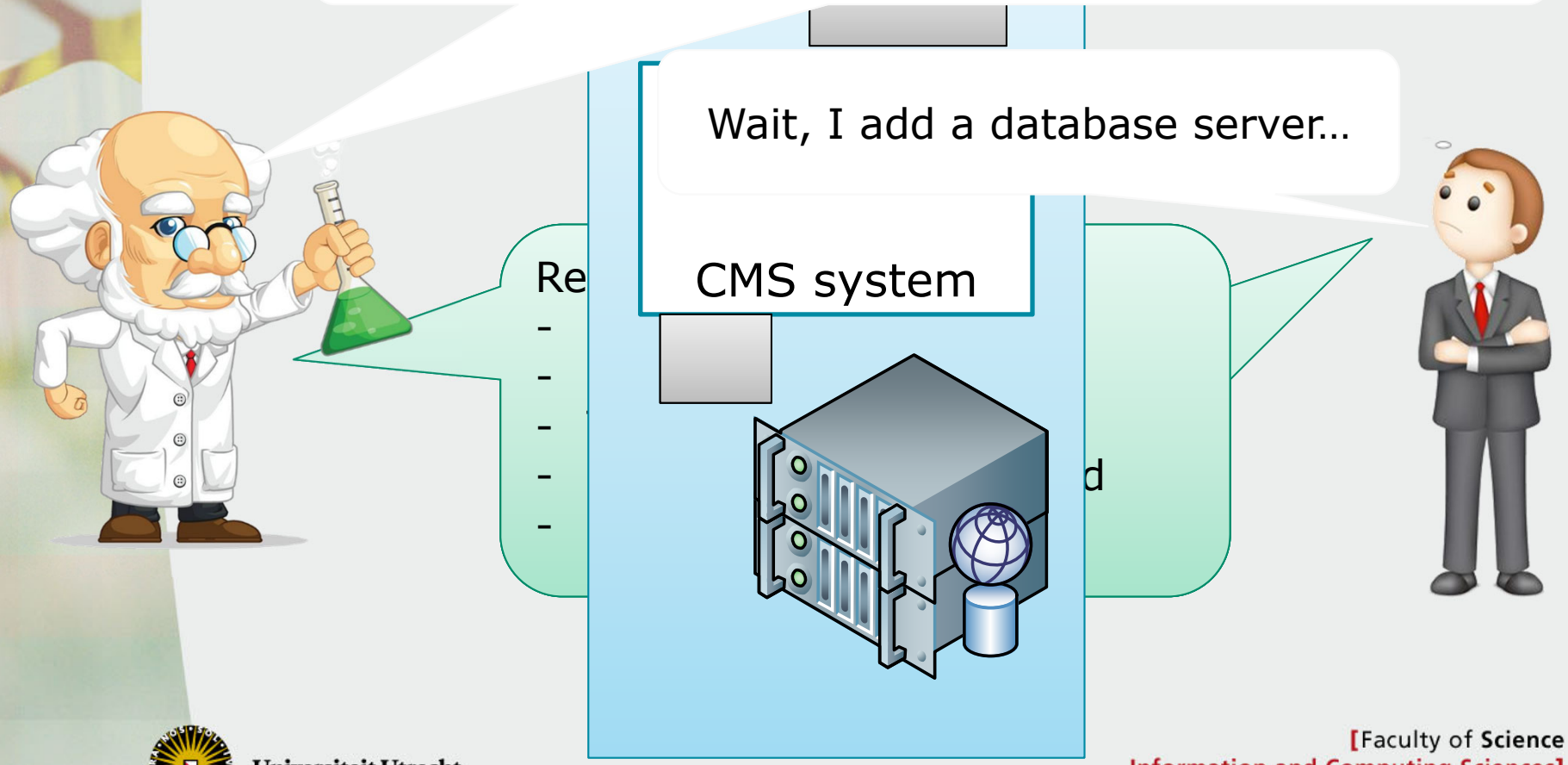
Requirements:
- Event listing
- Event overview
- Ticket selection
- Payment with credit card
- Financial overview

Building a software system

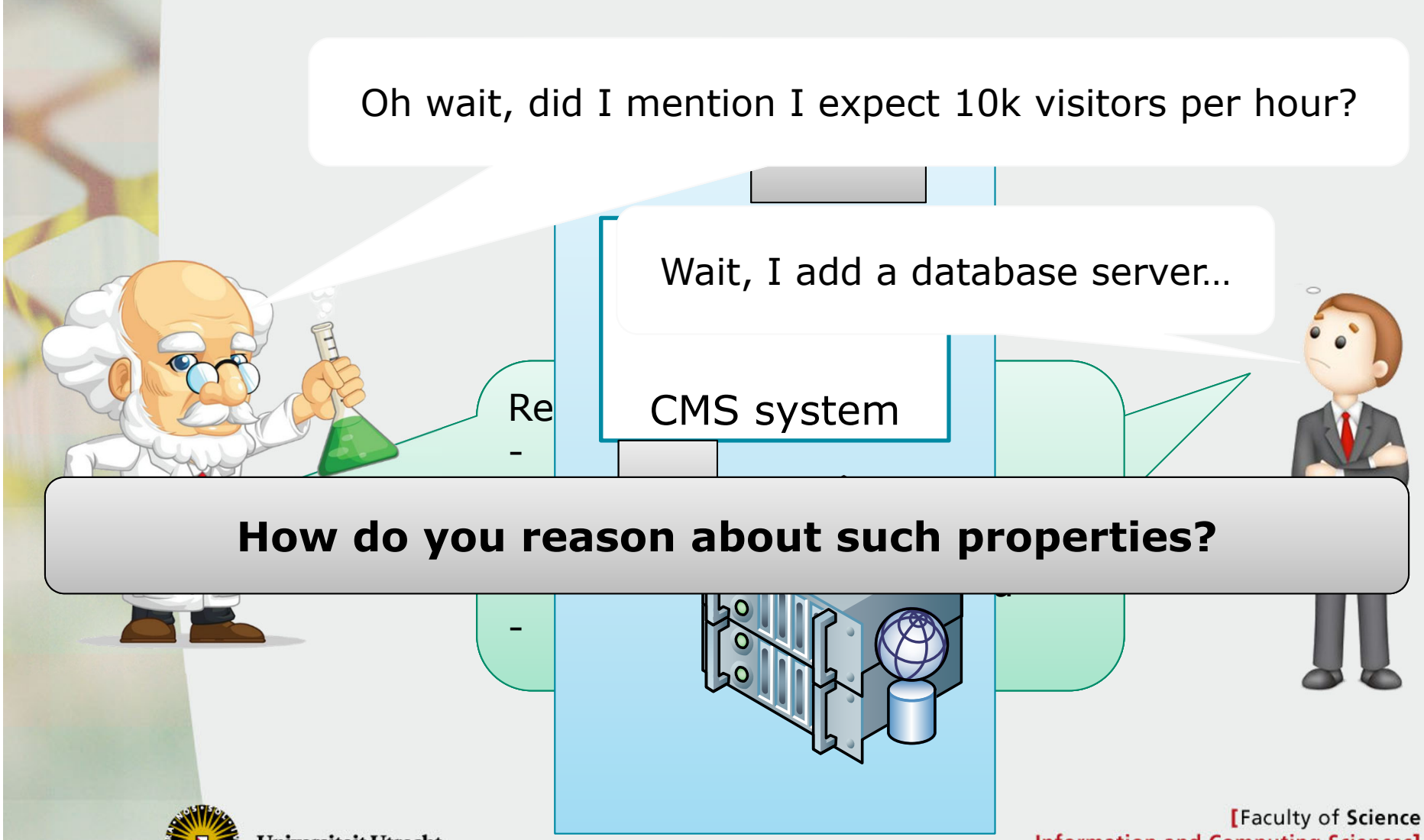


Building a software system

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

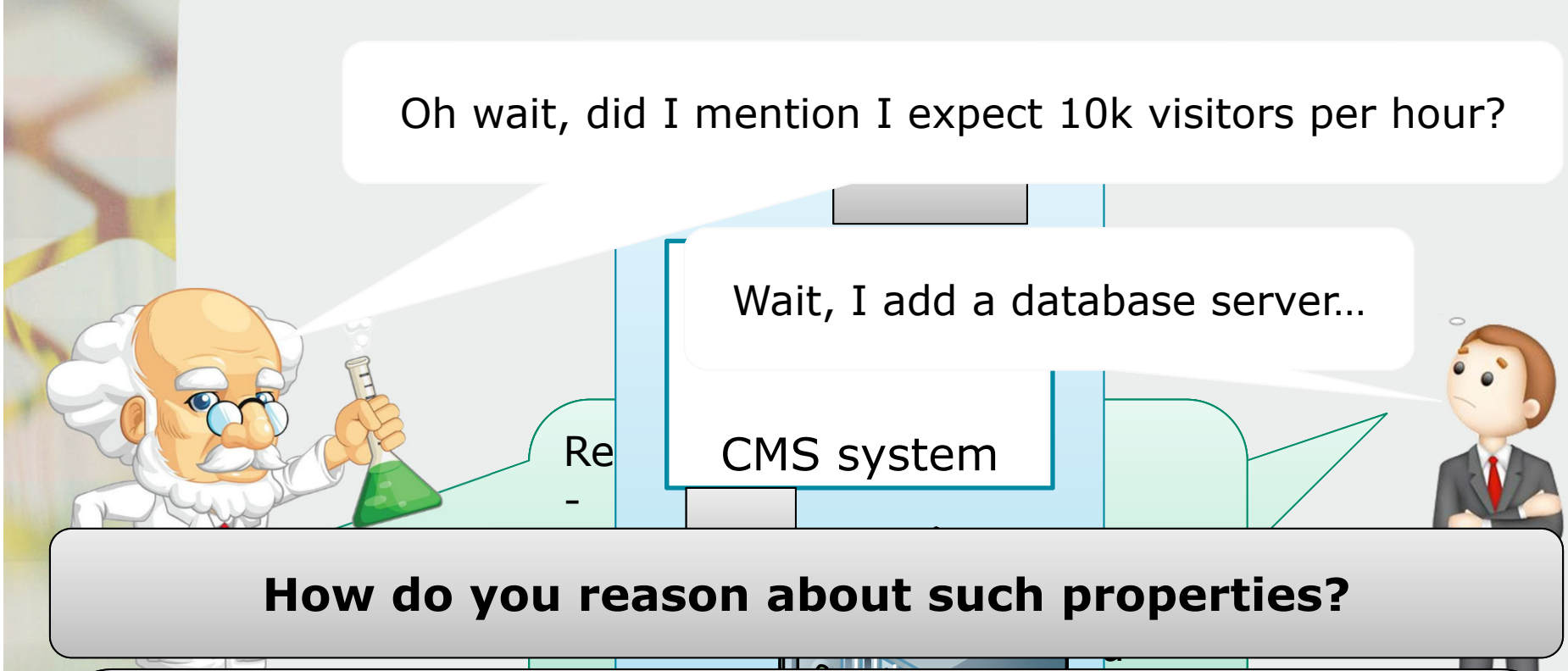


Building a software system



How do you reason about such properties?

Building a software system



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

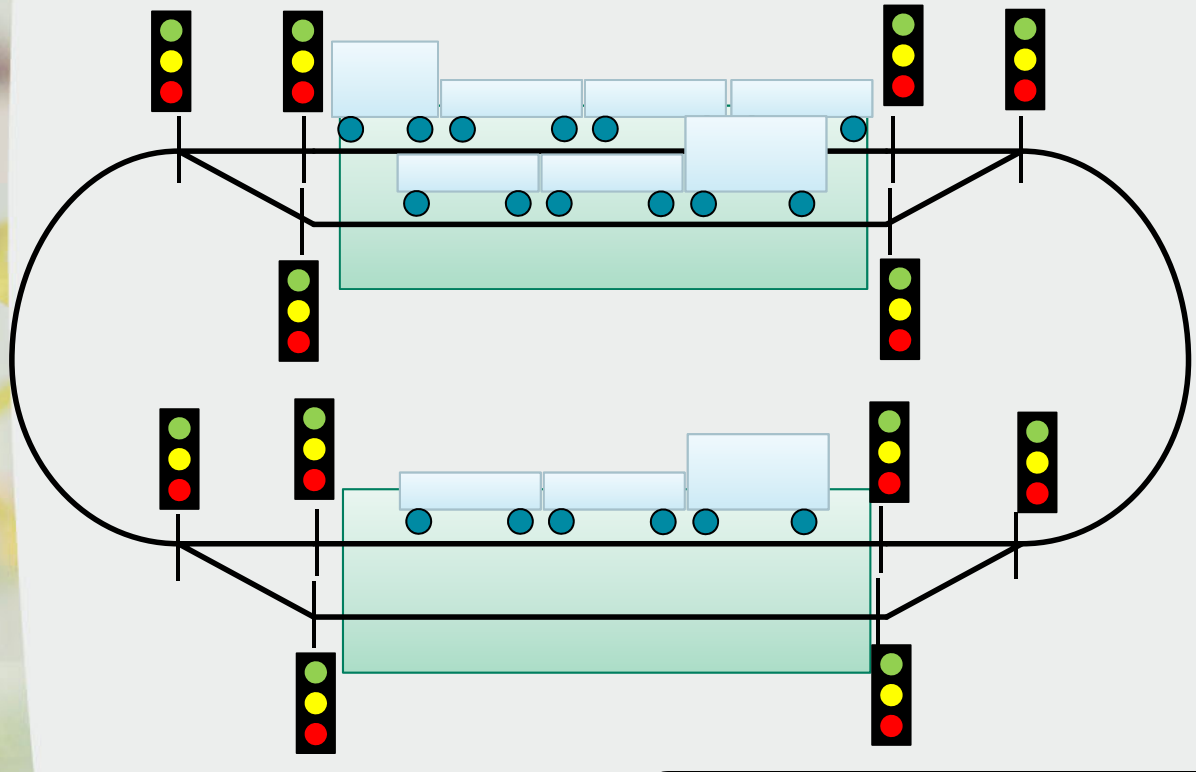
(Clements et al, 2003)

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Let’s move to a simple example

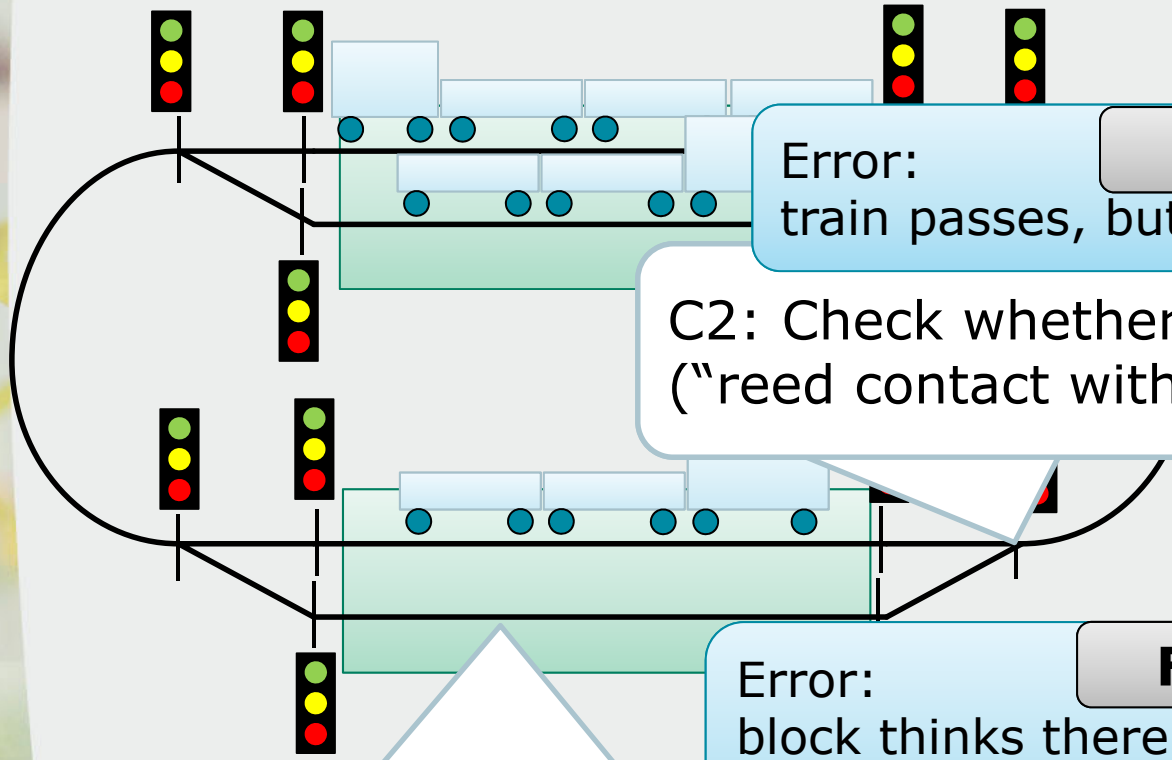
A simple railroad



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

Universiteit Utrecht

Step 1: detect trains



False negative

Error:
train passes, but reed did not notice

C2: Check whether a train passes
("reed contact with magnets")

False positive

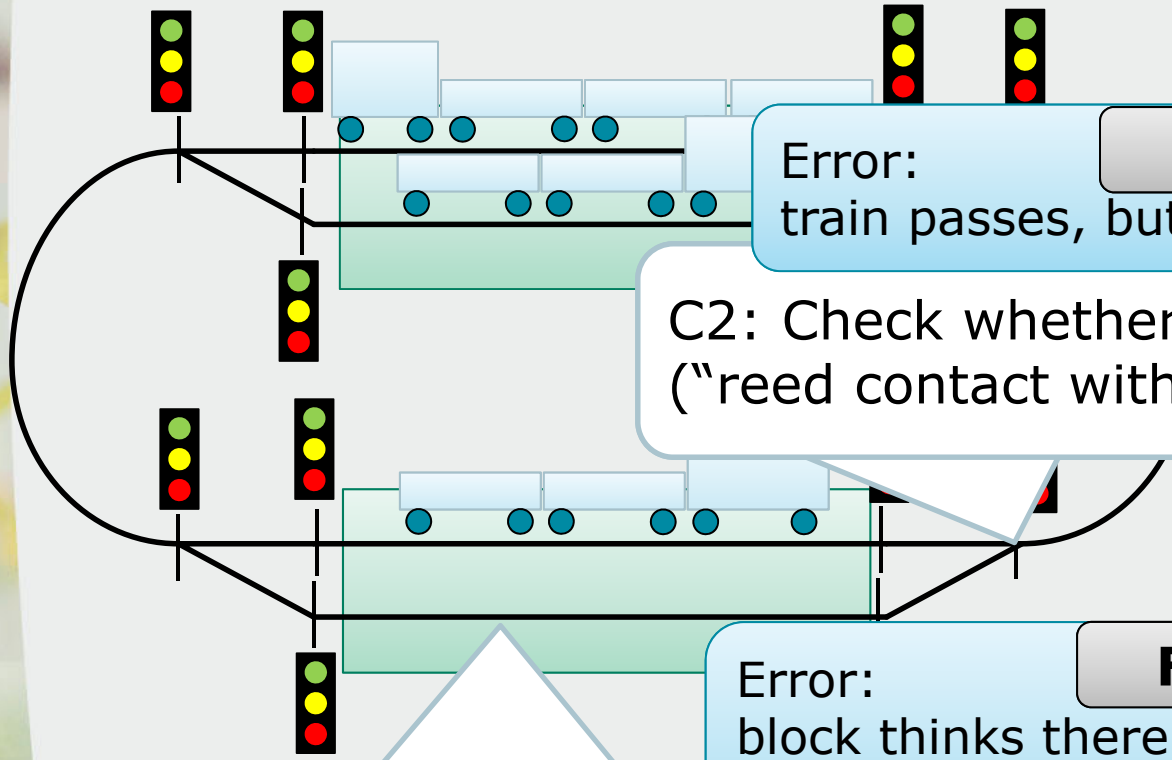
Error:
block thinks there is a train, but
there is none

C1: Check whether there is some current
("Stroomdetectie")

- Switches handled automatically

Universiteit Utrecht

Step 1: detect trains



Error:
train passes, but reed did not notice

False negative

C2: Check whether a train passes
("reed contact with magnets")

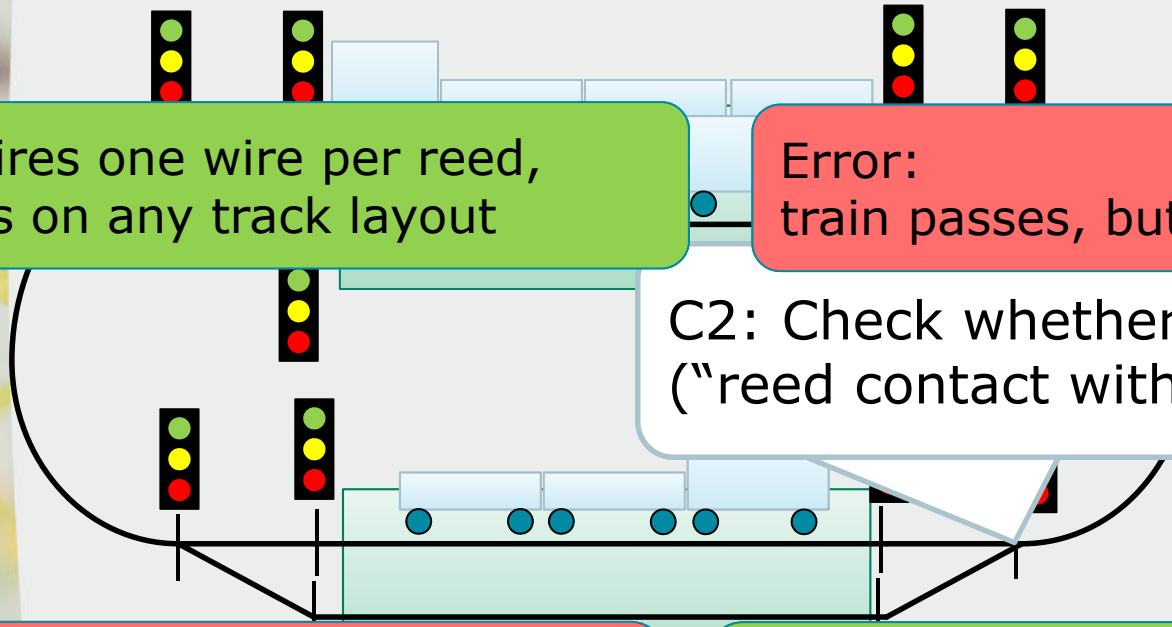
Error:
block thinks there is a train, but
there is none

False positive

C1: Check whether there is some current
("Stroomdetectie")

- Switches handled automatically

Step 1: detect trains



Requires one wire per reed, works on any track layout

Error:
train passes, but reed did not notice

C2: Check whether a train passes ("reed contact with magnets")

Requires a lot of wires and tracks need to be adapted

Error:
block thinks there is a train, but there is none

C1: Check whether there is some current
("Stroomdetectie")

Software Architecture:
the composition of a set of architectural design decisions

(Jansen & Bosch, 2005)

Let's structure and think over the options…

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

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Let's structure and think over the options…

Context:

Train detection

Facing concern:

Safety

Criteria:

- Ease of implementation
- Collisions
- Availability
- Performance

Options

Candidate 1	Candidate 2
Current detection	Reed contacts
Idea: divide track in blocks, measure for each block the current. If ≥ 0, then there is a train	Idea: place magnets under train, and place reeds that detects whether a magnet passes
- Cut tracks to create blocks	+ Add reeds to track
+ Not possible	- Possible if not detected

Decision:

Candidate 1

Because:

There are no collisions possible

Consequences:

Additional wiring

A change to the tracks by changing the connections between blocks

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Let's structure and think over the options…

Context:

Train detec

Facing co

Safety

Criteria:

Ease of impl

Collisions

Availability

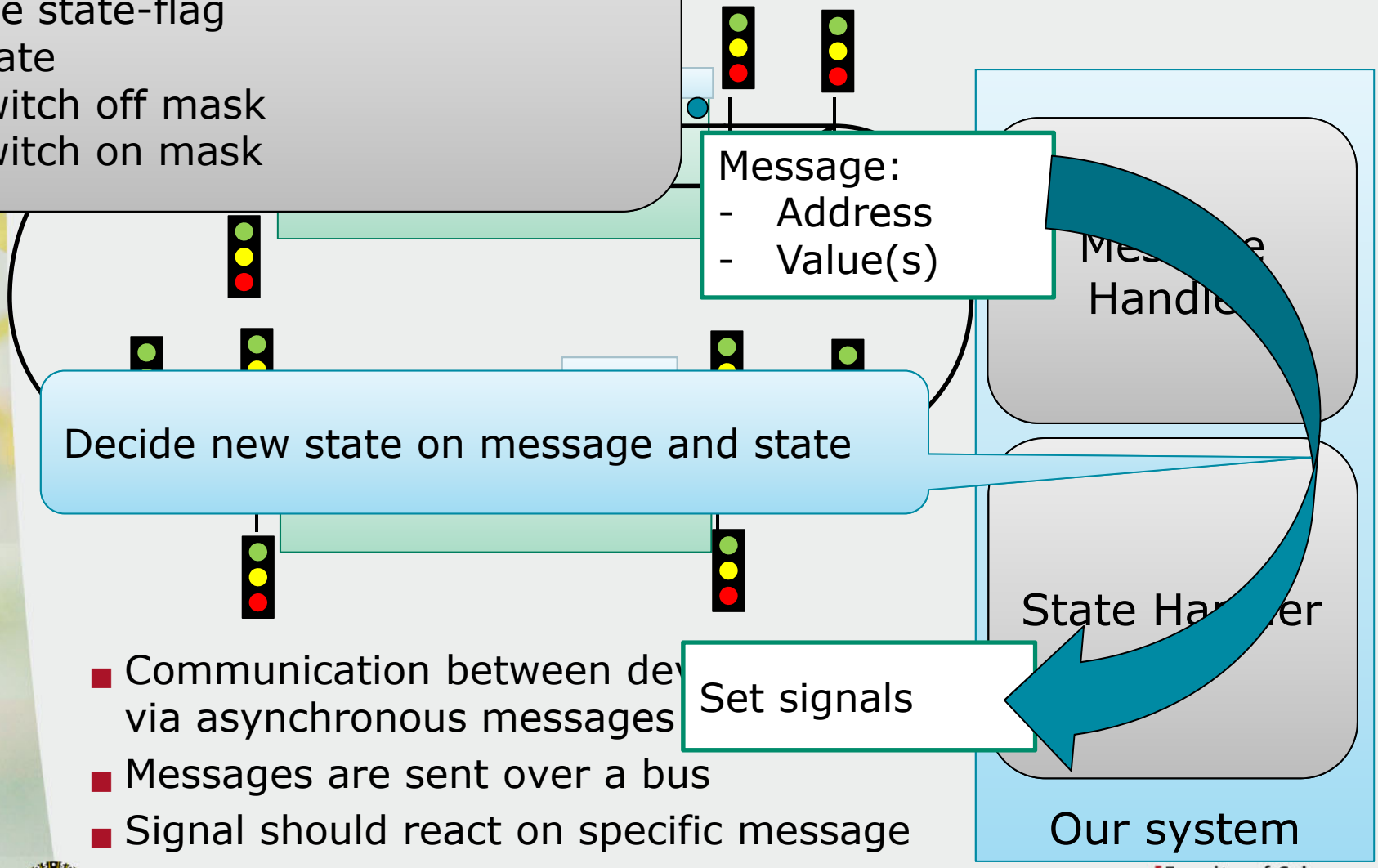
Performanc

Decision story: Train detection

In the context	of our model railroad,
facing	Safety,
We decided to	use block detection
And not to	use reed contacts
Because	in case of failure, trains stop thus not colliding
Accepting and	the additional work in wiring changing the track connections.

Decision table: 128 entries with:
- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

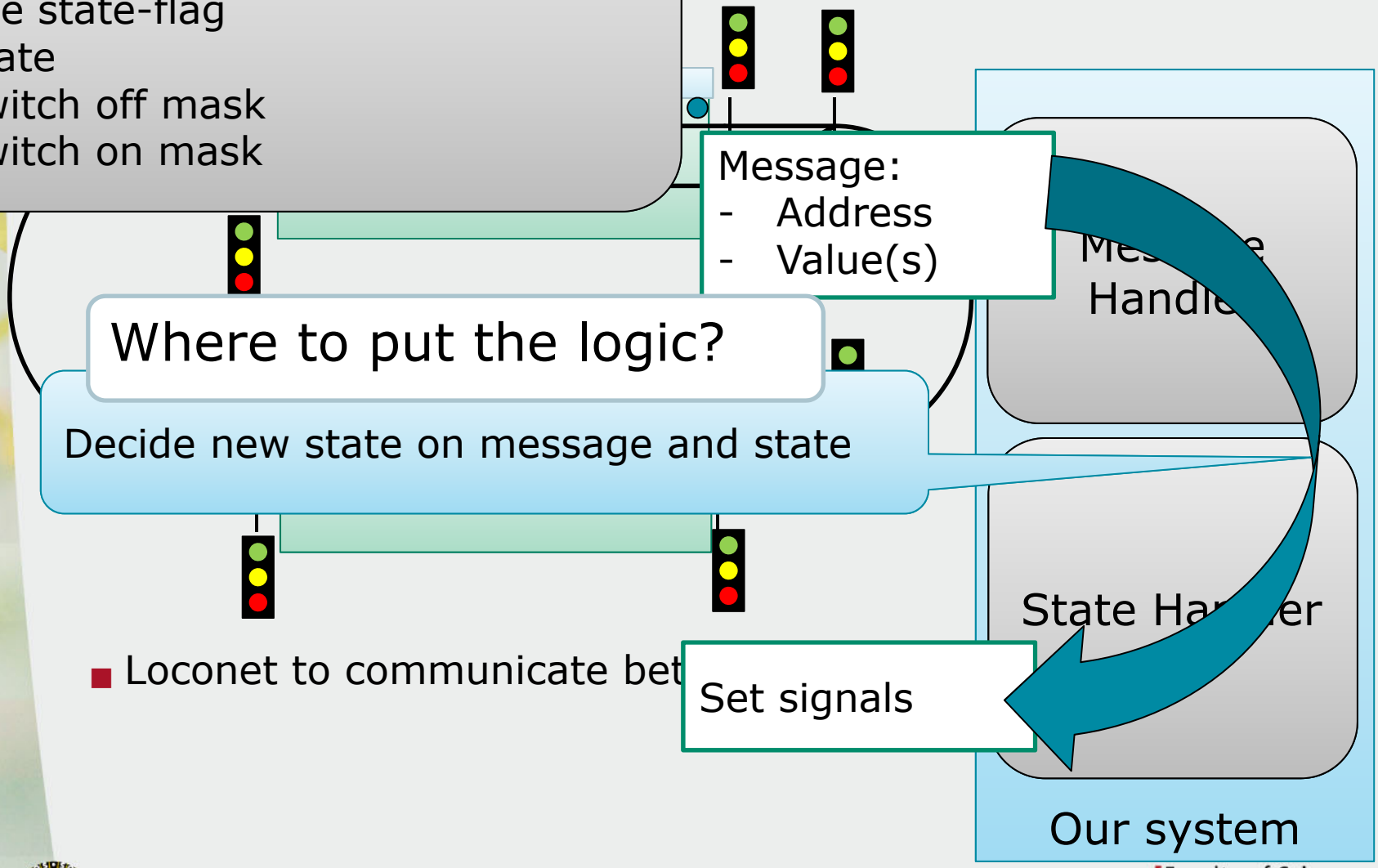
ve to software!



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

Decision table: 128 entries with:
- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

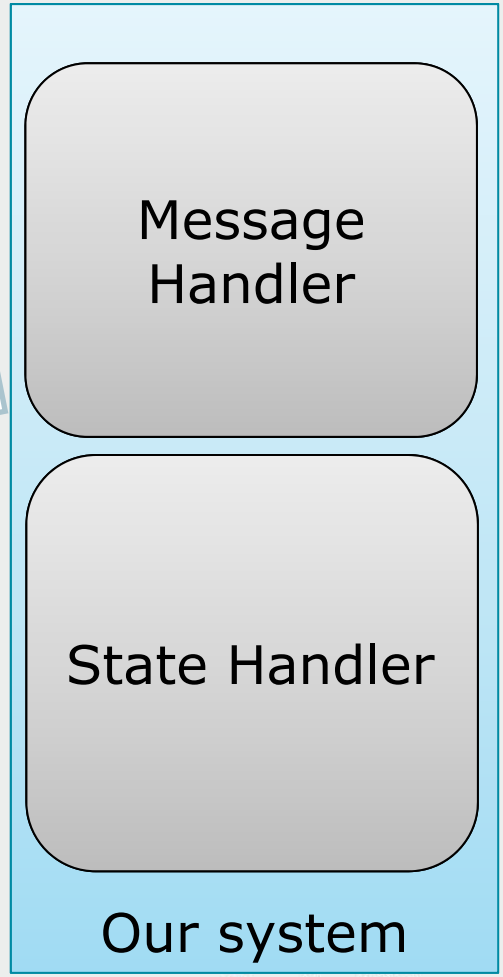
ve to software!



- Loconet to communicate bet

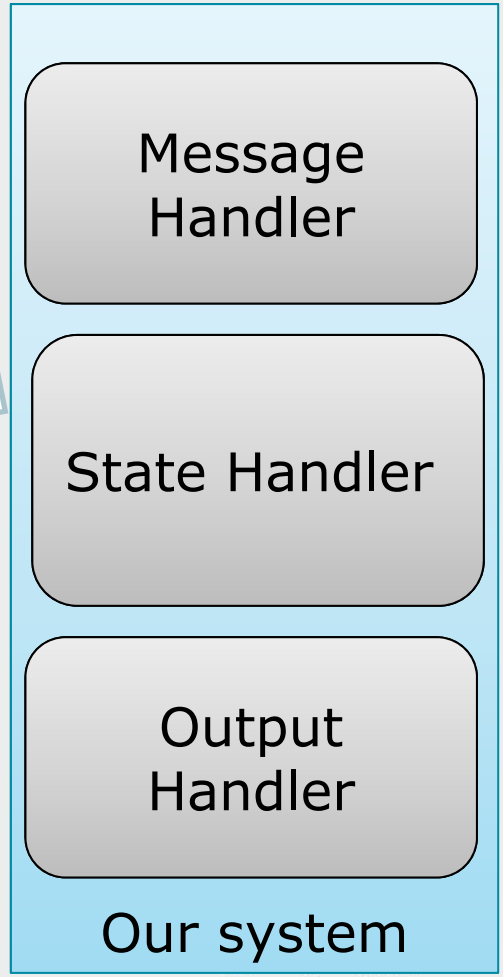
Decision table: 128 entries with:
- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

Where to put the logic?



Decision table: 128 entries with:
- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

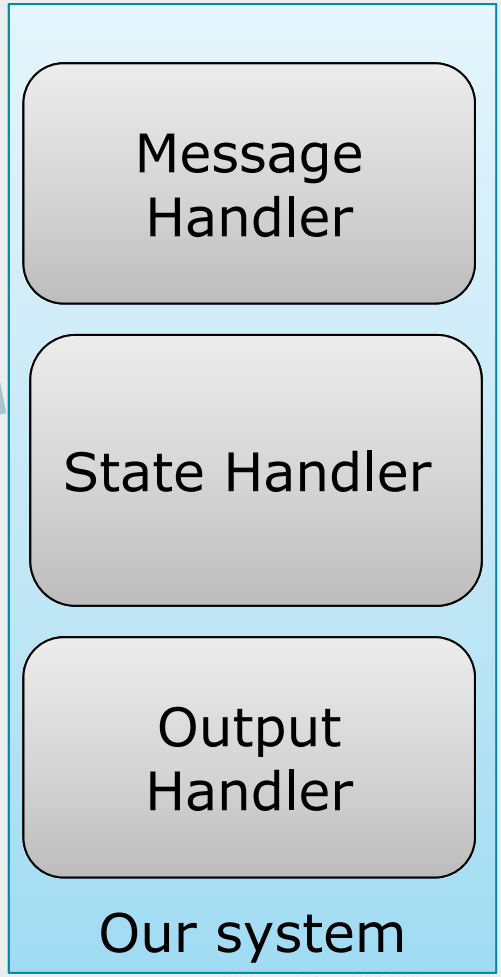
Where to put the logic?



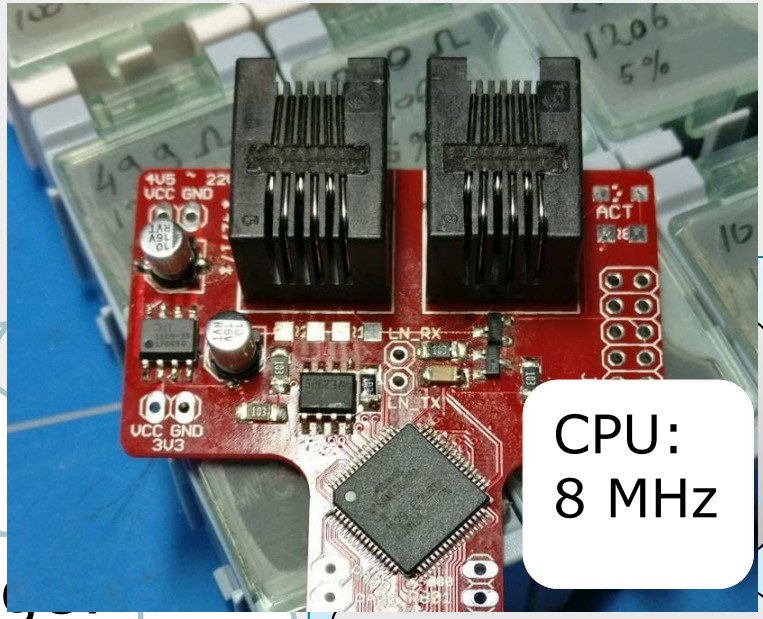
Decision table: 128 entries with:
- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask

For each arrived message:

1. Check if address & state in table
2. If found, update state with entry



Decision table: 128 entries with:
- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask



CPU:
8 MHz

For each arrived messag

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

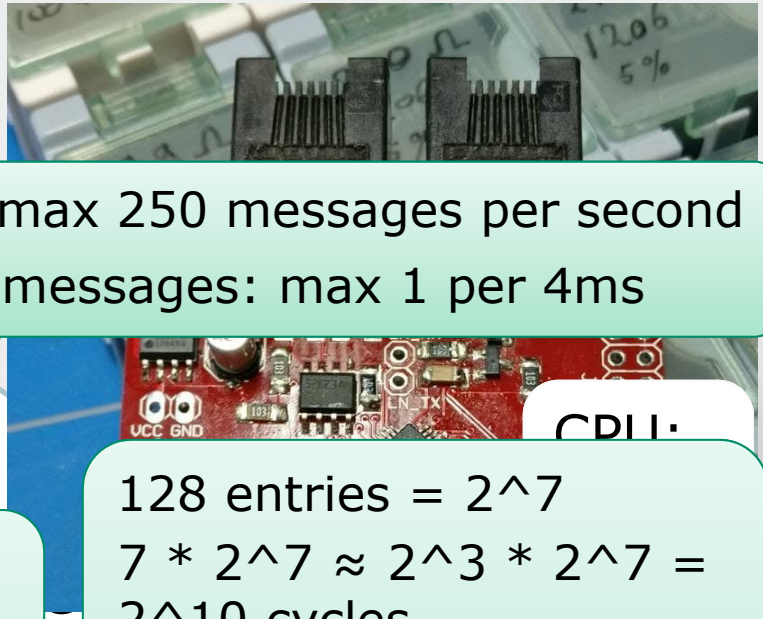
State Handler

Output Handler

Our system

[Faculty of Science
formation and Computing Sciences]

Decision table: 128 entries with:
- Address
- Use state-flag
- State
- Switch off mask
- Switch on mask



Loconet: max 250 messages per second

Max 250 messages: max 1 per 4ms

128 entries = 2^7

7 * 2^7 ≈ 2^3 * 2^7 = 2^10 cycles

@ 8MHz gives 0.128 ms

128 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

Binary Search

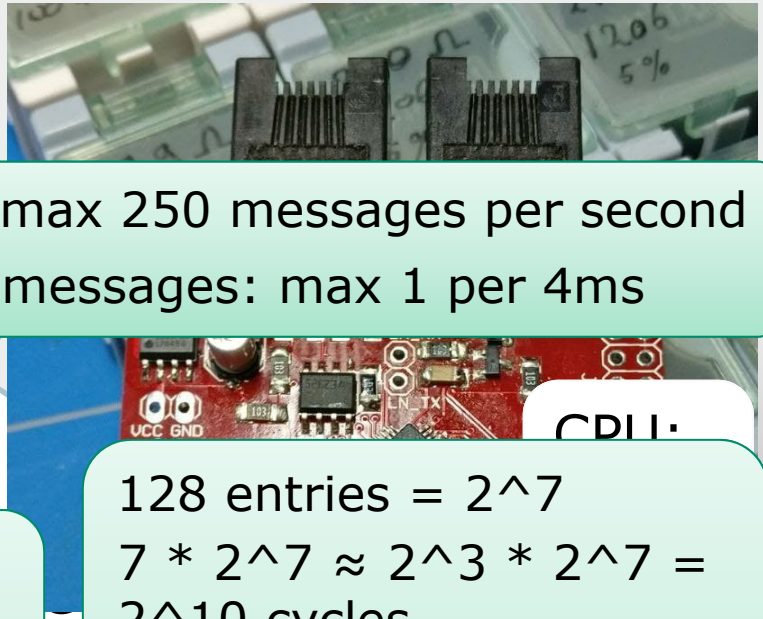
- Logarithmic
- Entries sorted
- Tree-based structure

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



Loconet: max 250 messages per second

Max 250 messages: max 1 per 4ms

128 entries = 2^7

7 * 2^7 ≈ 2^3 * 2^7 = 2^10 cycles

@ 8MHz gives 0.128 ms

128 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

Binary Search

- Logarithmic
- Entries sorted
- Tree-based structure

Output Handler

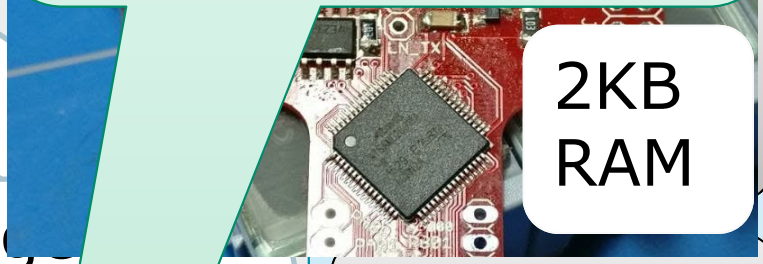
Our system

[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!

State Handler

Output Handler

Our system

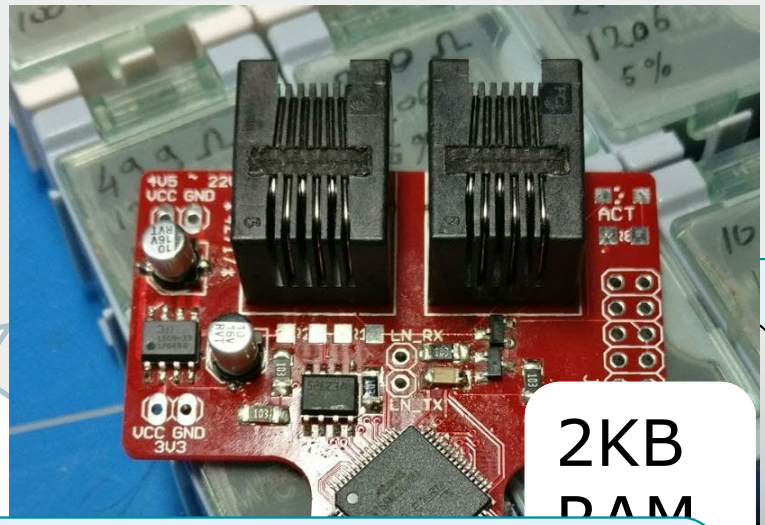
Bounded Linear Search
- Linear
- Entries in any order
- Simple data structure

Binary Search
- Logarithmic
- Entries sorted
- Simple data structure

[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 [illegible]sk — 2 bytes



2KB
RAM

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

Constraint: use simple array!

Bounded Linear Search	Binary Search
Linear	Logarithmic
Entries in any order	Entries sorted
Simple data structure	Simple data structure

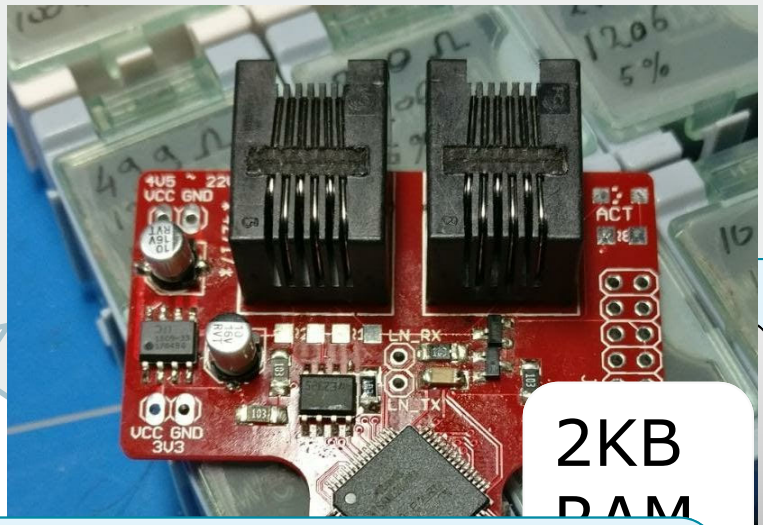
Output Handler

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 [illegible]sk — 2 bytes



2KB
RAM

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

Constraint: use simple array!

Bounded Linear Search	Binary Search
Linear	Logarithmic
Entries in any order	Entries sorted
Simple data structure	Simple data structure

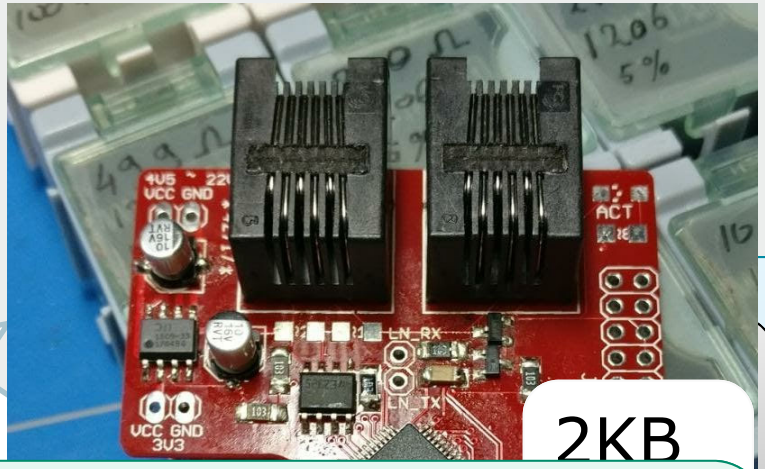
Output Handler

Our system

[Faculty of Science
[illegible]formation 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



2KB

Maintaining the table:

Insert: add to the back O(1)
Delete: Place last over removed entry O(1)

Maintaining the table:

Insert: ad O(n)
Delete: back O(n)

1. Che dress & state in table

Bounded L r Search

- Linear
- Entries in any order
- Simple data structure

Binary Se

- Logarithmic
- Entries sorted
- Simple data structure

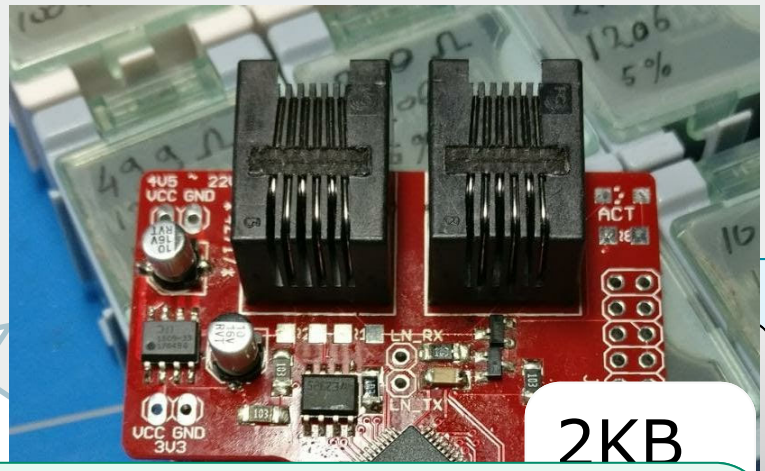
Output Handler

Our system

[Faculty of Science
formation 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



2KB

Maintaining the table:

Insert: add to the back O(1)
Delete: Place last over removed entry O(1)

Maintaining the table:

Insert: ~~ad~~ O(n)
Delete: ~~e al~~ back O(n)

1. Che dress & state in table

Bounded L r Search

- Linear
- Entries in any order
- Simple data structure

Binary Se h

- Logarithmic
- Entries sorted
- Simple data structure

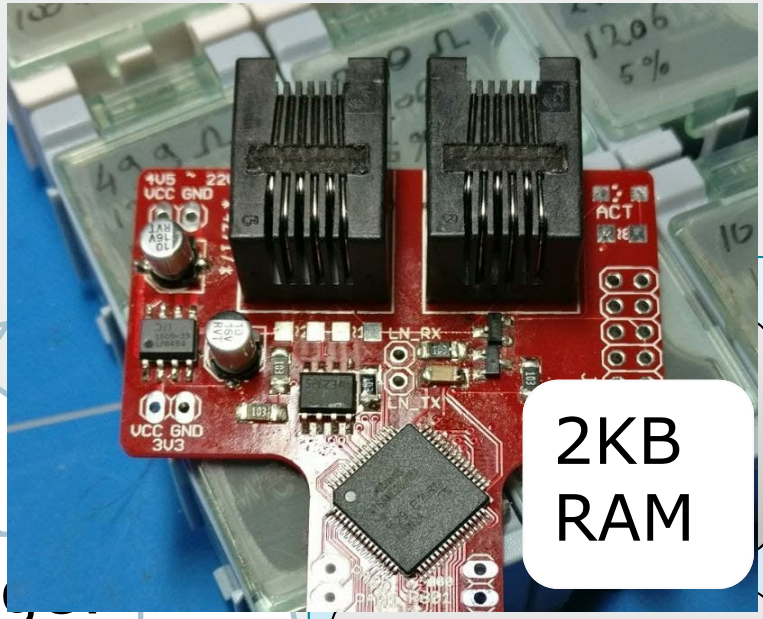
Output Handler

Our system

[Faculty of Science
formation 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



2KB RAM

For each arrived messag

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

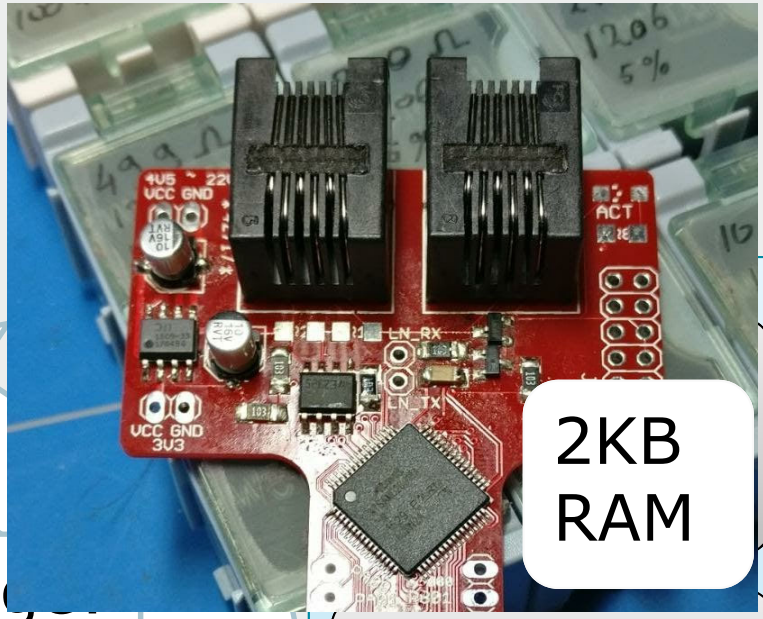
Output Handler

Our system

[Faculty of Science
formation 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



2KB RAM

For each arrived messag

State Handler

We favor maintainability over performance

Bounded Linear Search

- Linear
- Entries in any order
- Simple data structure

~~Log… inary Sear…~~

- ~~Log…~~
- ~~Ent…~~
- ~~Sim…le data stru…ure~~

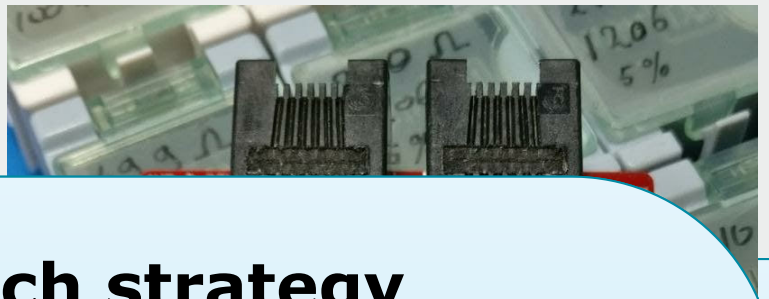
Output Handler

Our system

[Faculty of Science
…formation and Computing Sciences]

Decision table: 128 entries with:
- Address
- Use state-fla
- State
- Switch
- Switch

2 bytes



We f

Bounded l

Linear

Entries i

Simple da

Decision story: Search strategy

In the context	of our RailGuard PCB,
facing	Maintainability,
We decided to	use the Bounded Linear Search
And not	the Binary Search
Because	It is fast enough,
and	It keeps the code base simpler
Accepting	the longer runtime required for a single pass.

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

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"

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

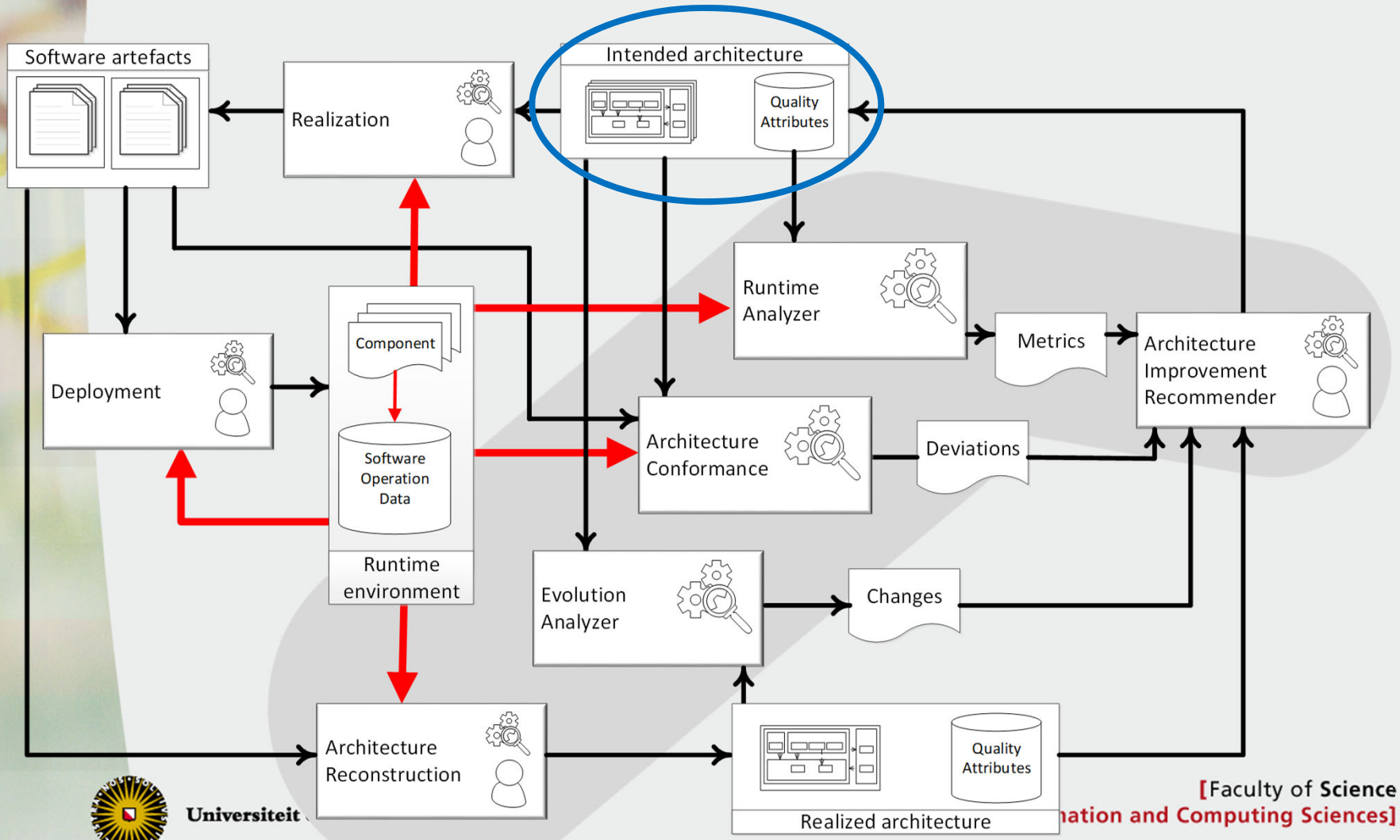
Software Architecture:

1) structures, relations and properties, and

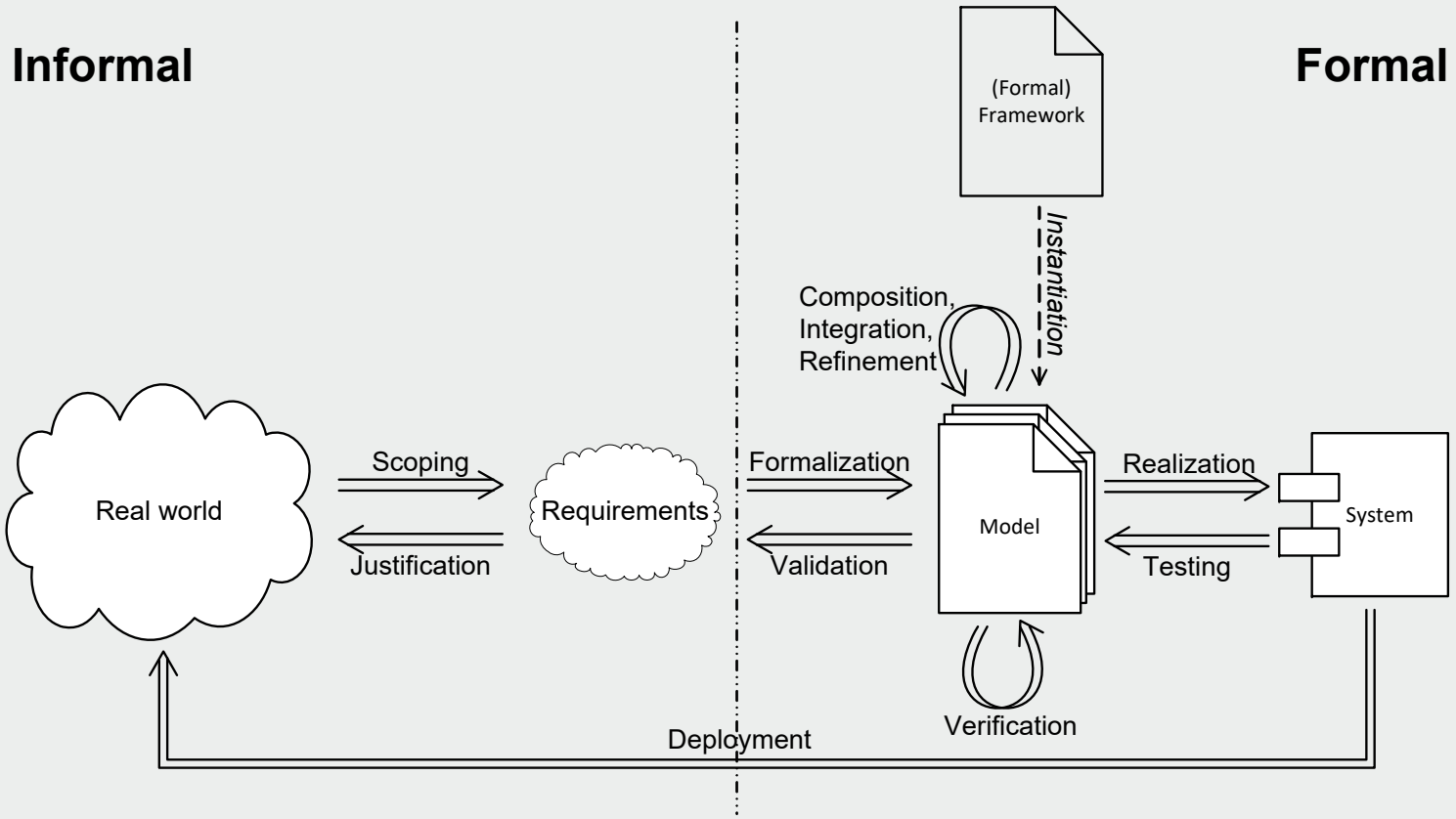
2) a set of architectural design decisions

Jan Martijn van der Werf

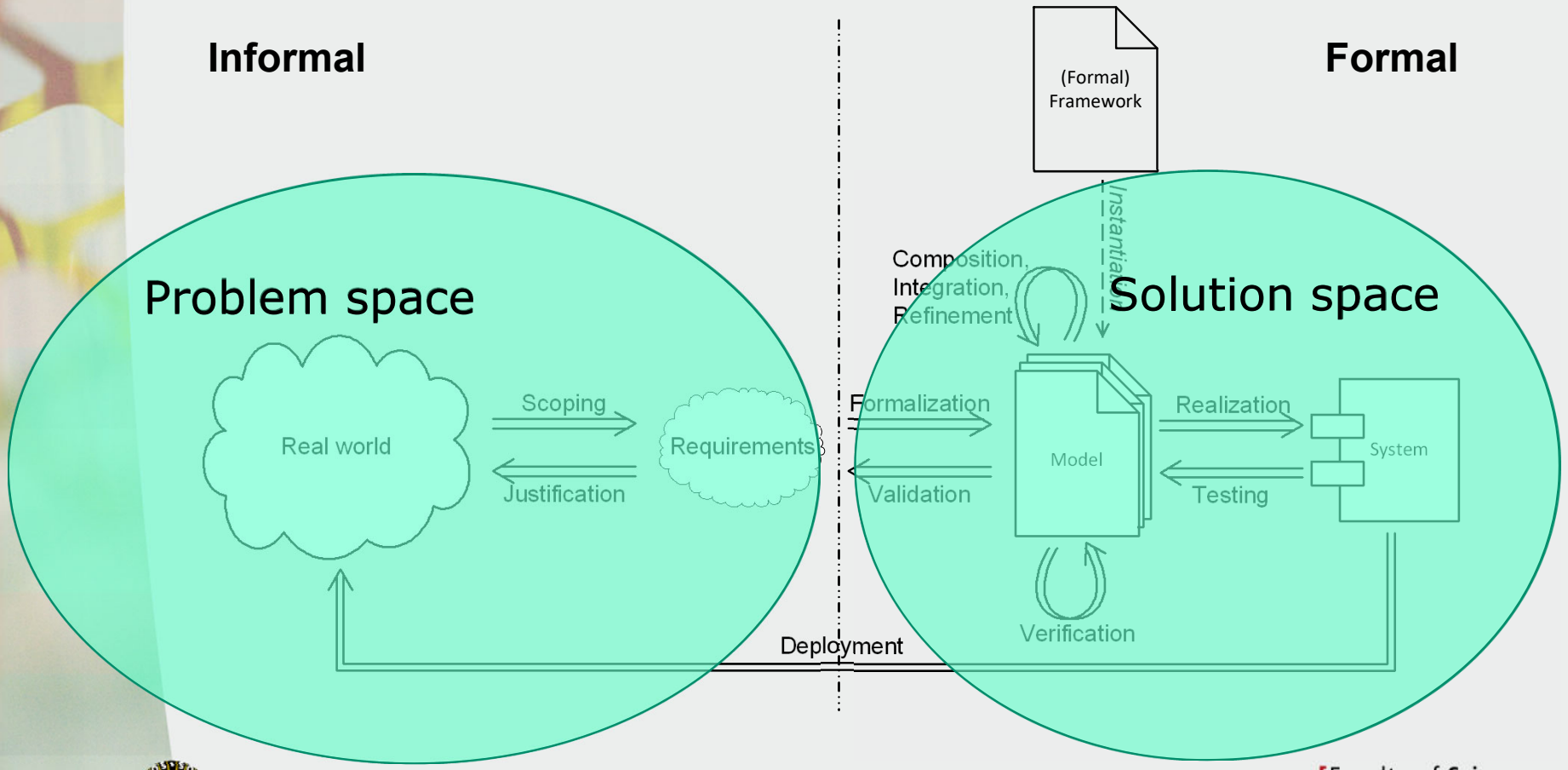
Software Architecture: overview of the field



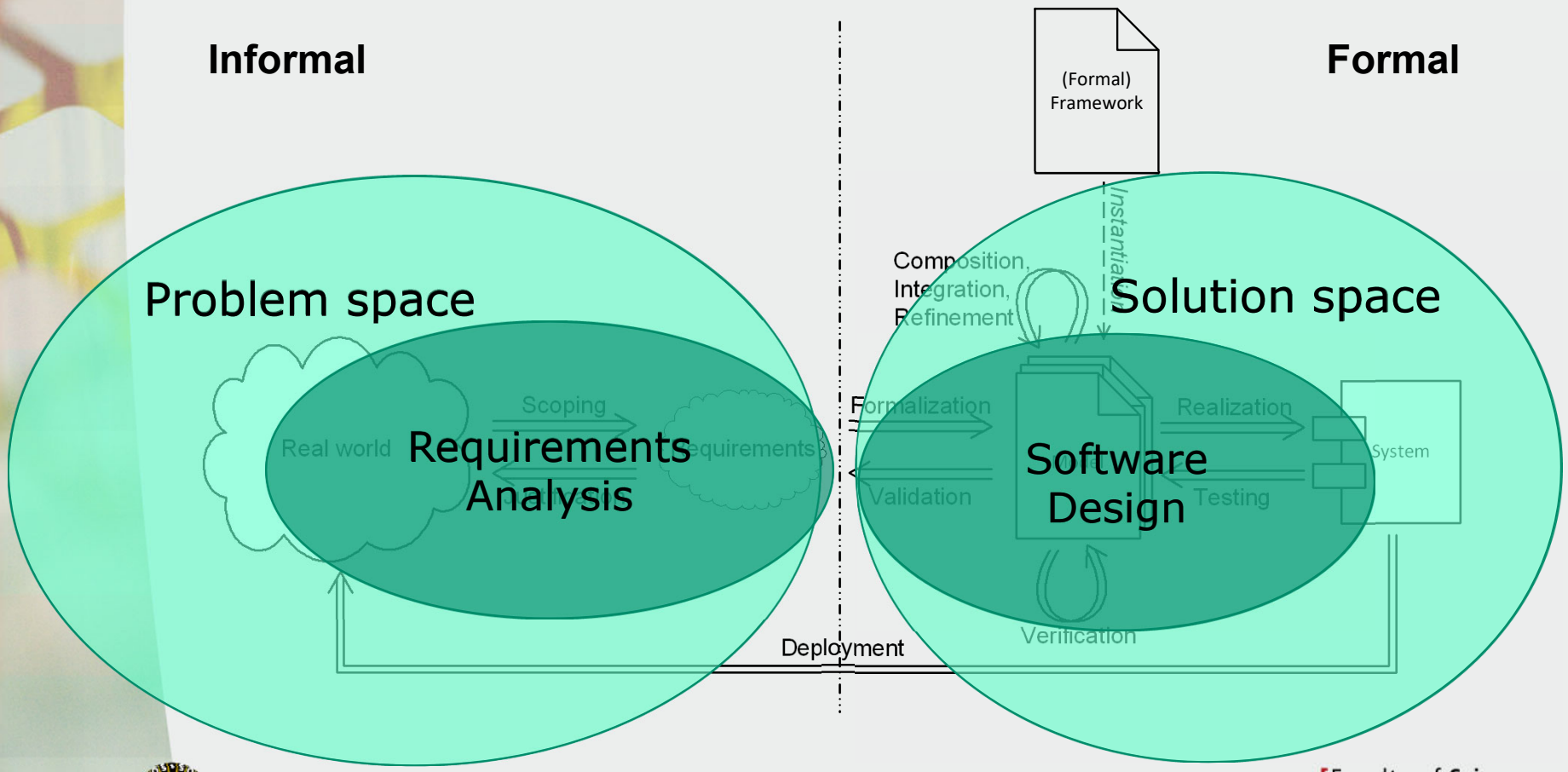
The role of Software Architecture



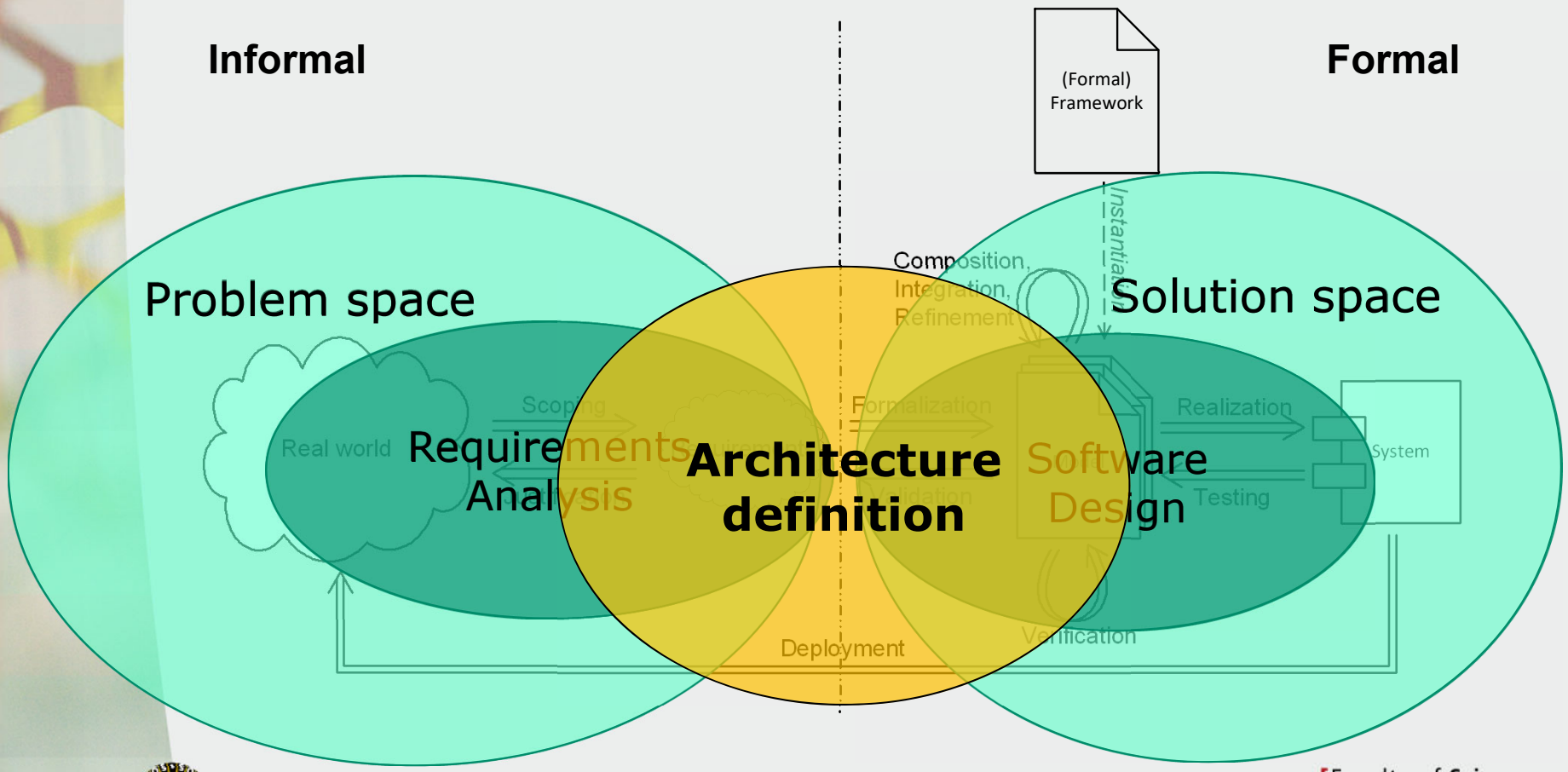
The role of Software Architecture



The role of Software Architecture



The role of Software Architecture



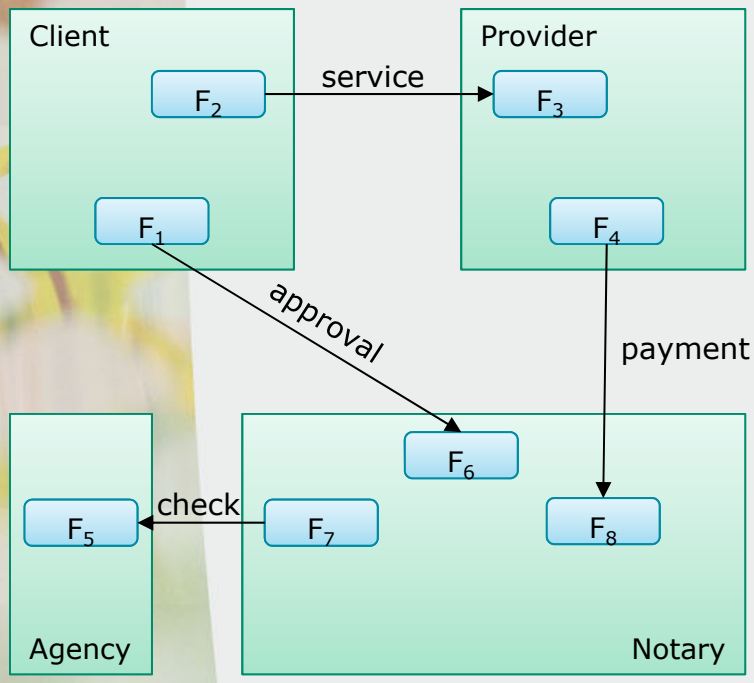
Architecture is similar to building a house

- 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!

Software Architecture considers:



- Static structures
  - Modular decomposition
  - Design patterns
- Dynamic design:
  - Protocols
  - Concurrency & component behavior
  - Infrastructure
- Perspectives (-ilities)
  - Performance
  - Availability
  - Scalability
  - Usability
  - Maintainability
  - Portability
  - ….

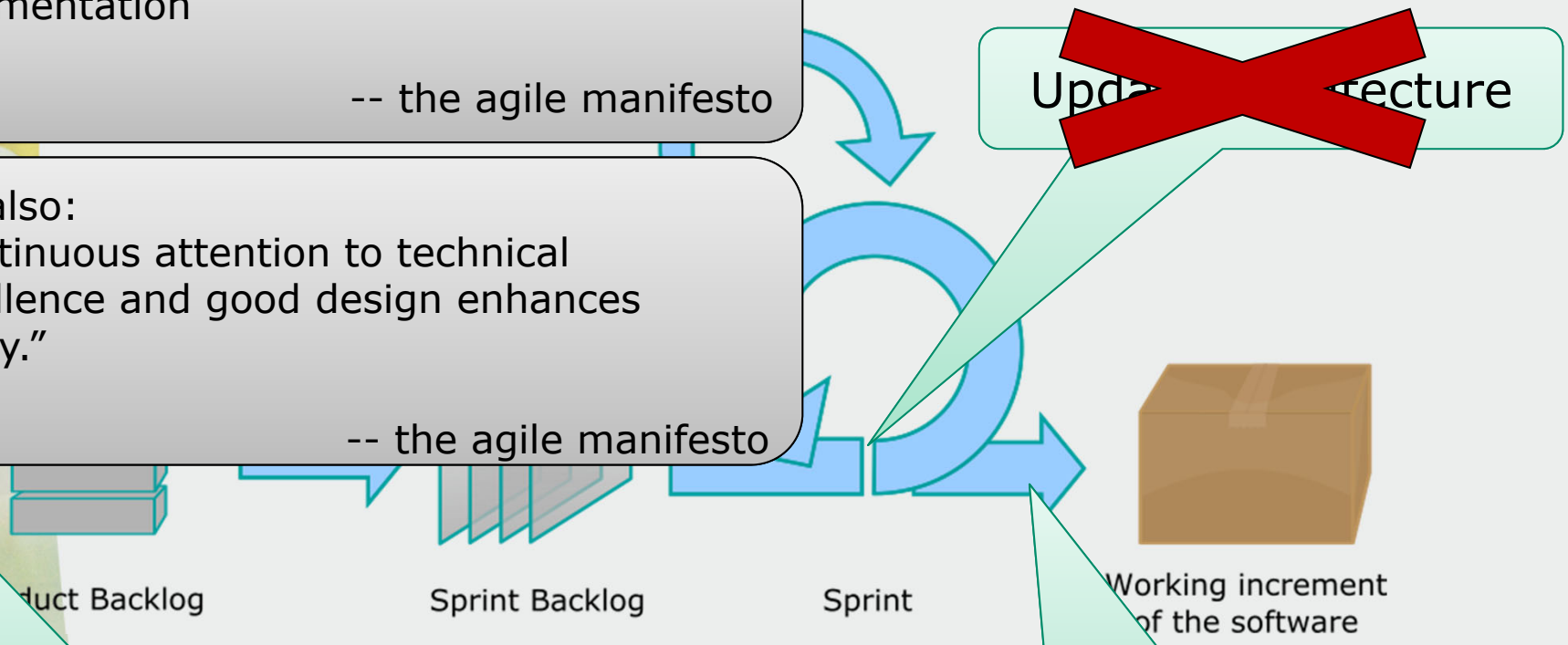
Architecture in practice

"Working software over comprehensive documentation"

-- the agile manifesto

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

-- the agile manifesto



PSA: Project Start Architecture

PSA ≠ PEA

PEA: Project End Architecture

Software architecture

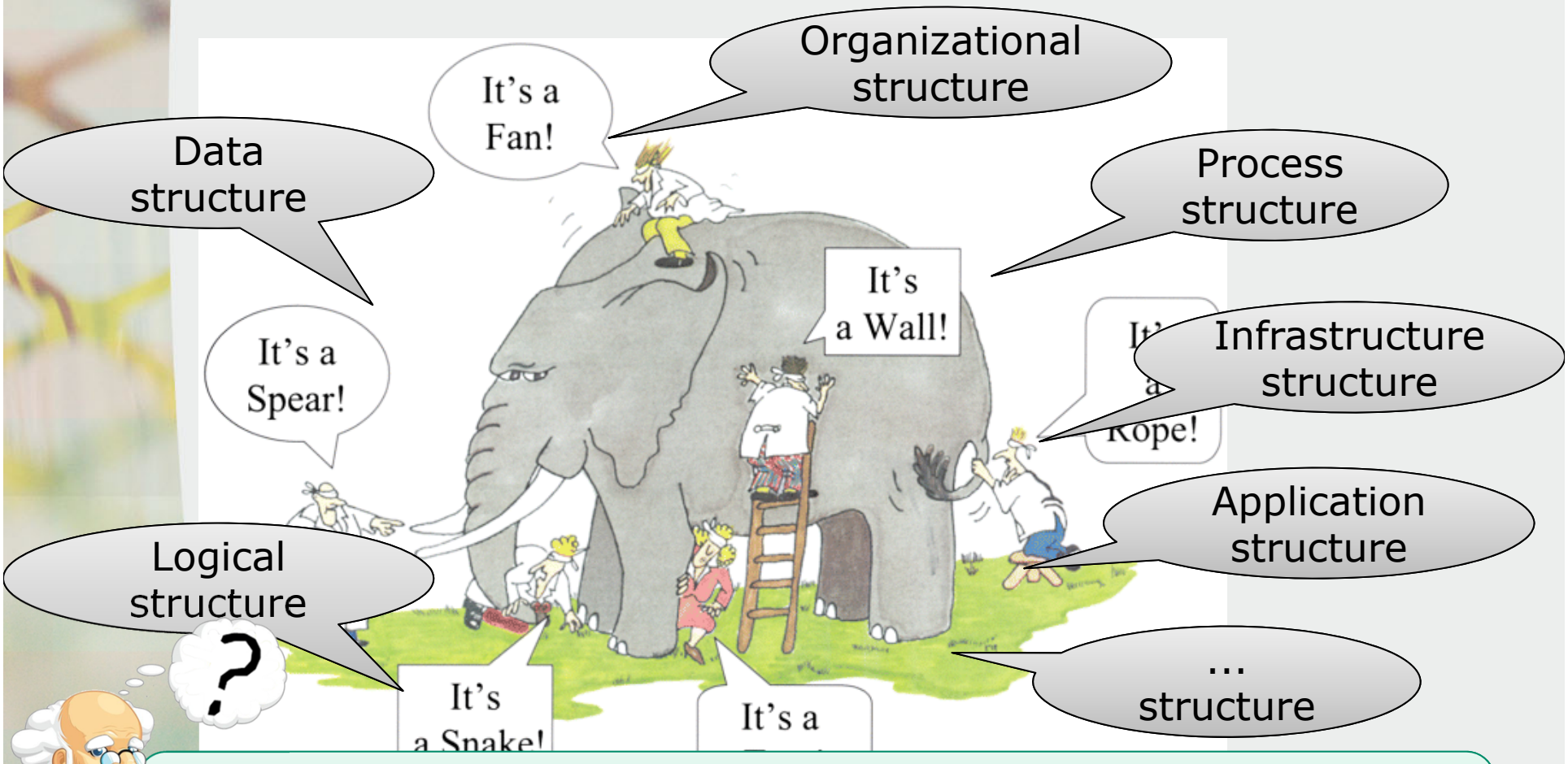
- Combines
  - Requirements
  - Design
  - Trade-off
  - Roadmapping
  - People-management
- To build software in a structured way

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

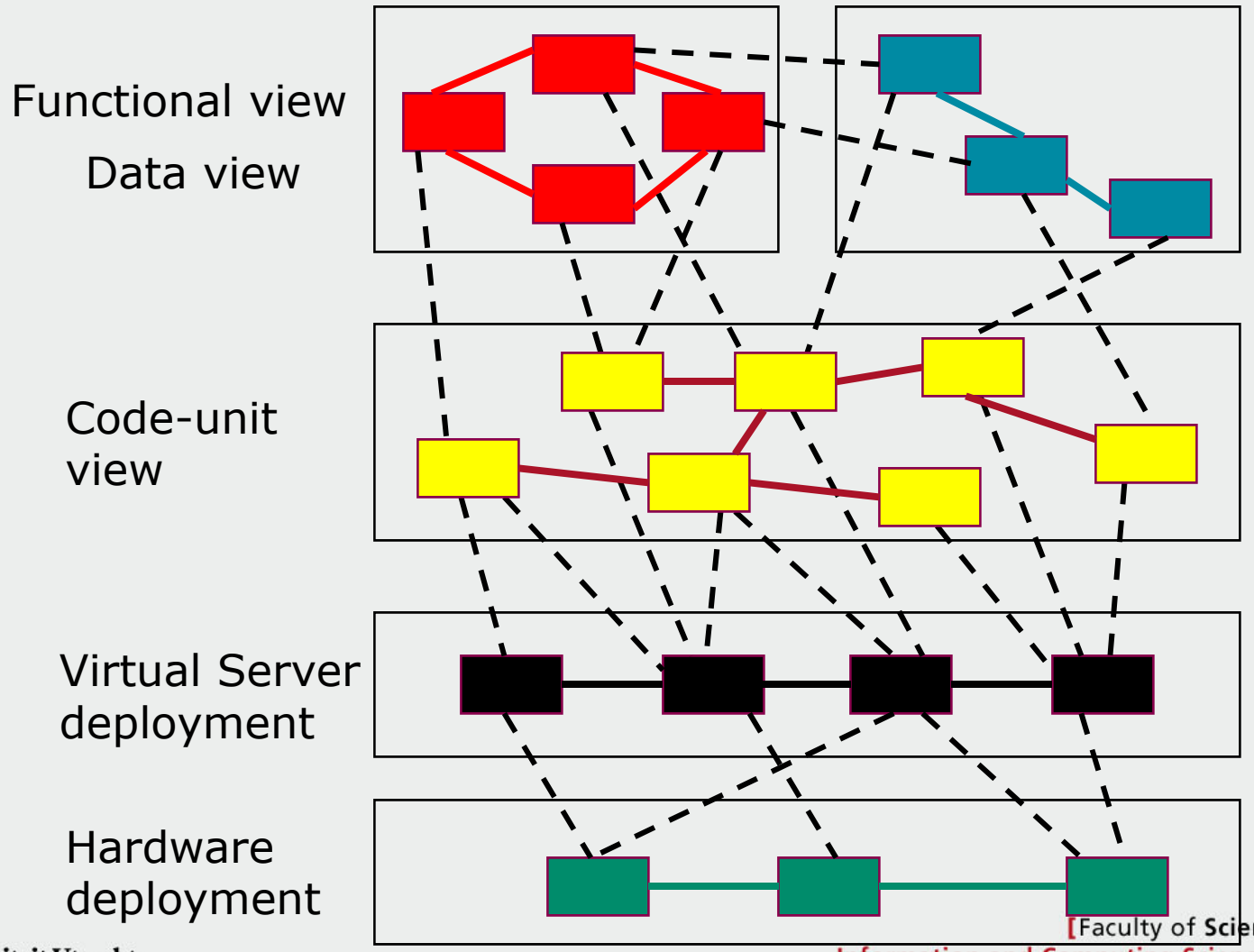
Software as a set of views

A set of structures

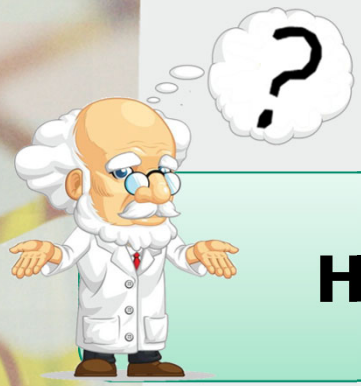


What is the relation between these views?

Structure of structures



Structure of structures



Functional view

Data view

How to keep those consistent?

Code-unit view

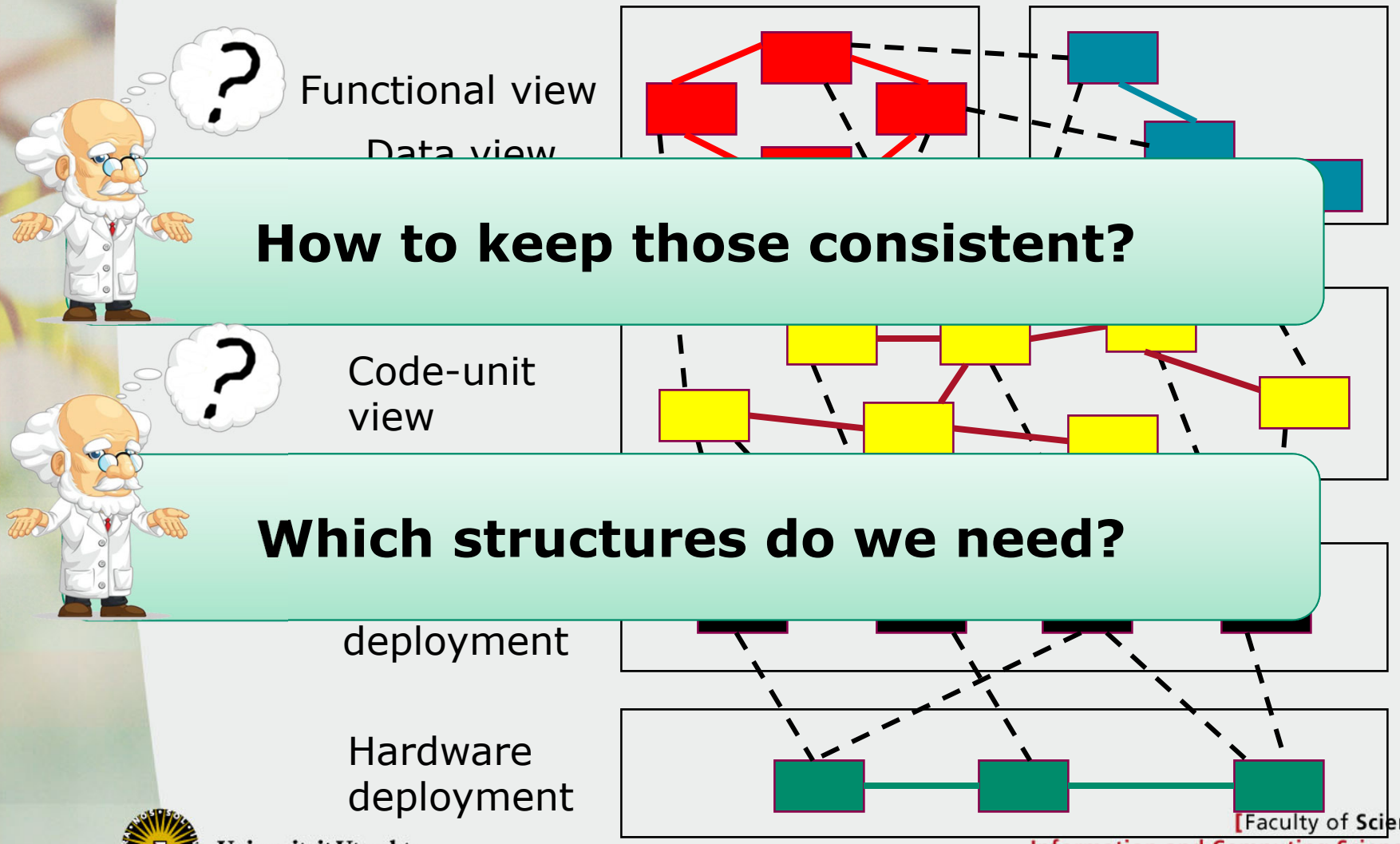
Virtual Server deployment

Hardware deployment

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

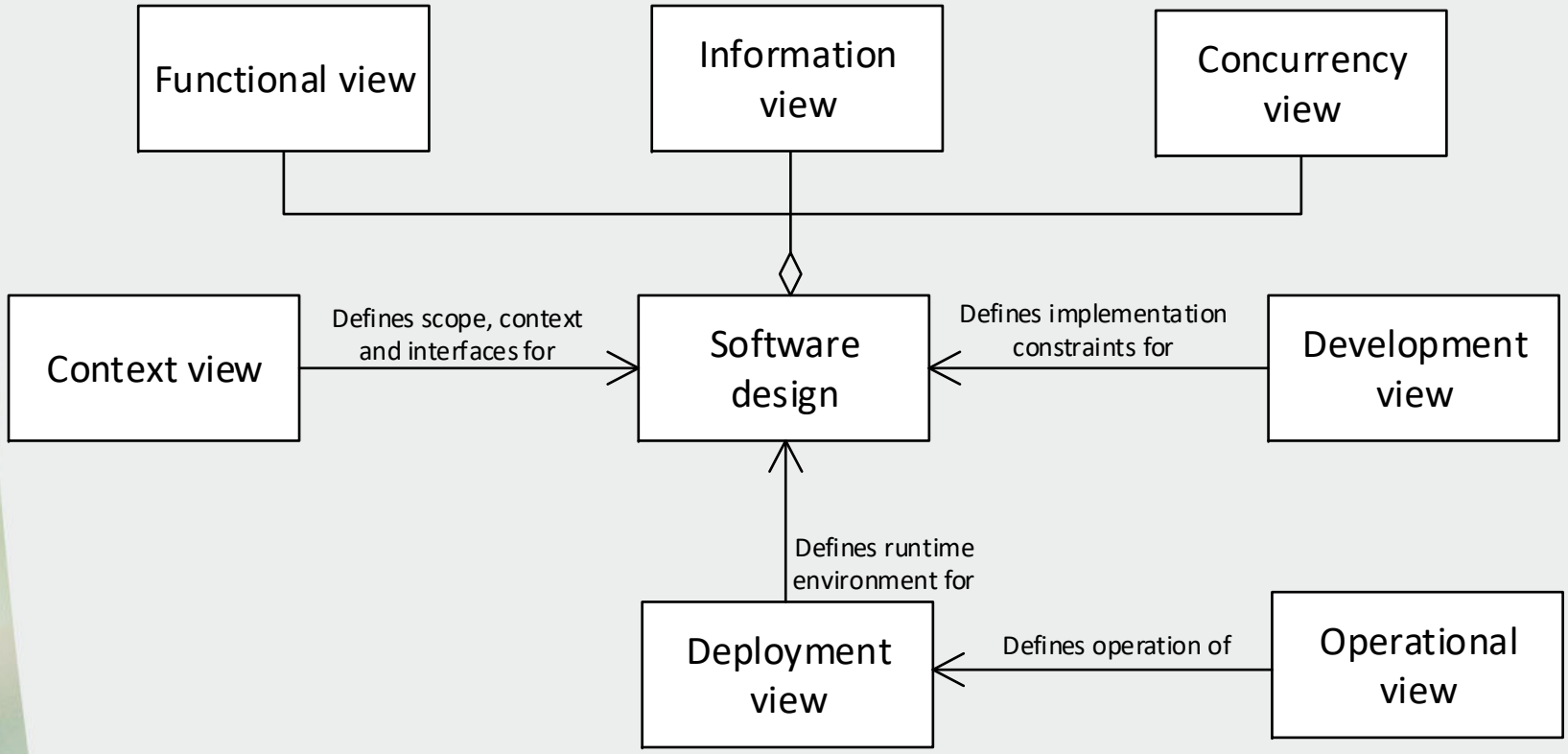
Structure of structures



How to keep those consistent?

Which structures do we need?

Classifying the different views...



Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

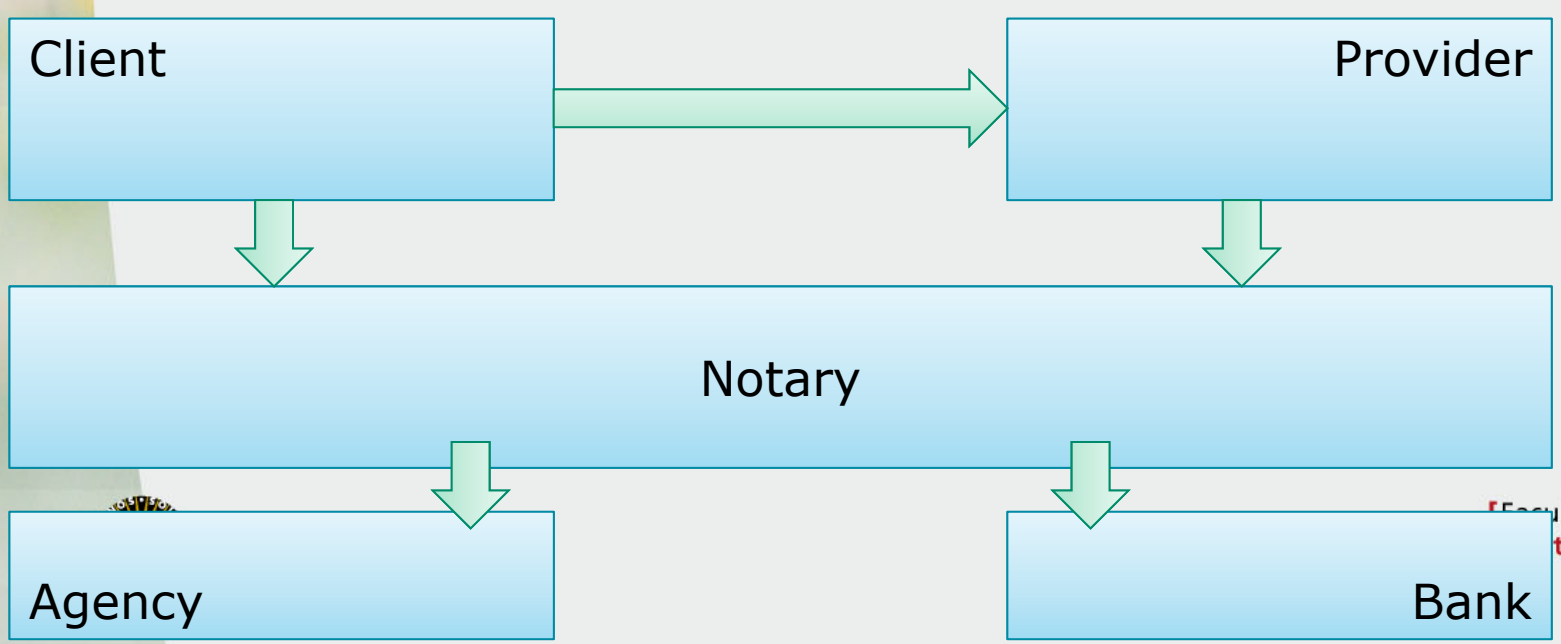
Functional architecture: static and dynamic structures

Suppose the following situation

- Providers offer services to clients
  - Clients first need to be checked at **the** Notary
  - Clients send their approval to the provider
  - Payment is only done once the client signed the bill
  - Multiple payments per service possible
  - Payment is done by the notary
  - The notary can use one or more agencies to get more info

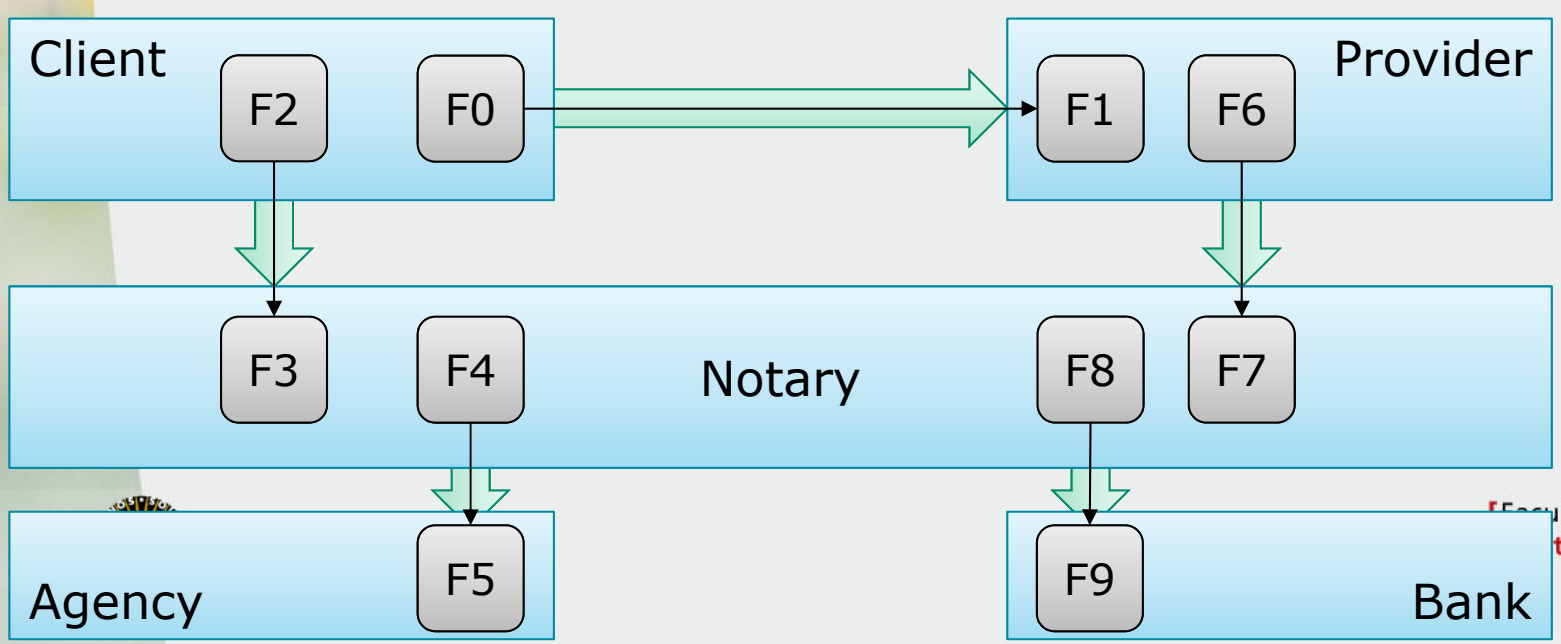
Suppose the following situation

- Providers offer services to clients
  - Clients first need to be checked at **the** Notary
  - Clients send their approval to the provider
  - Payment is only done once the client signed the bill
  - Multiple payments per service possible
  - Payment is done by the notary
  - The notary can use one or more agencies to get more info



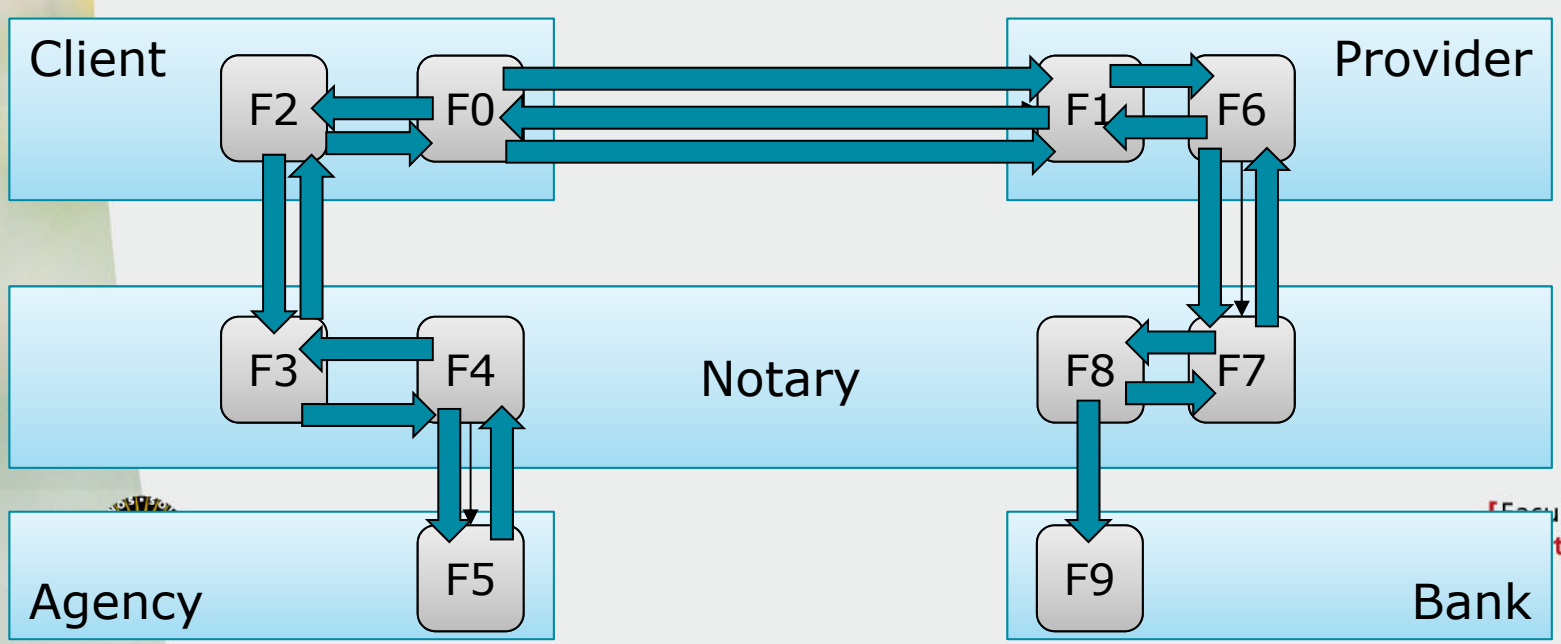
Suppose the following situation

- Providers offer services to clients
  - Clients first need to be checked at **the** Notary
  - Clients send their approval to the provider
  - Payment is only done once the client signed the bill
  - Multiple payments per service possible
  - Payment is done by the notary
  - The notary can use one or more agencies to get more info



Suppose the following situation

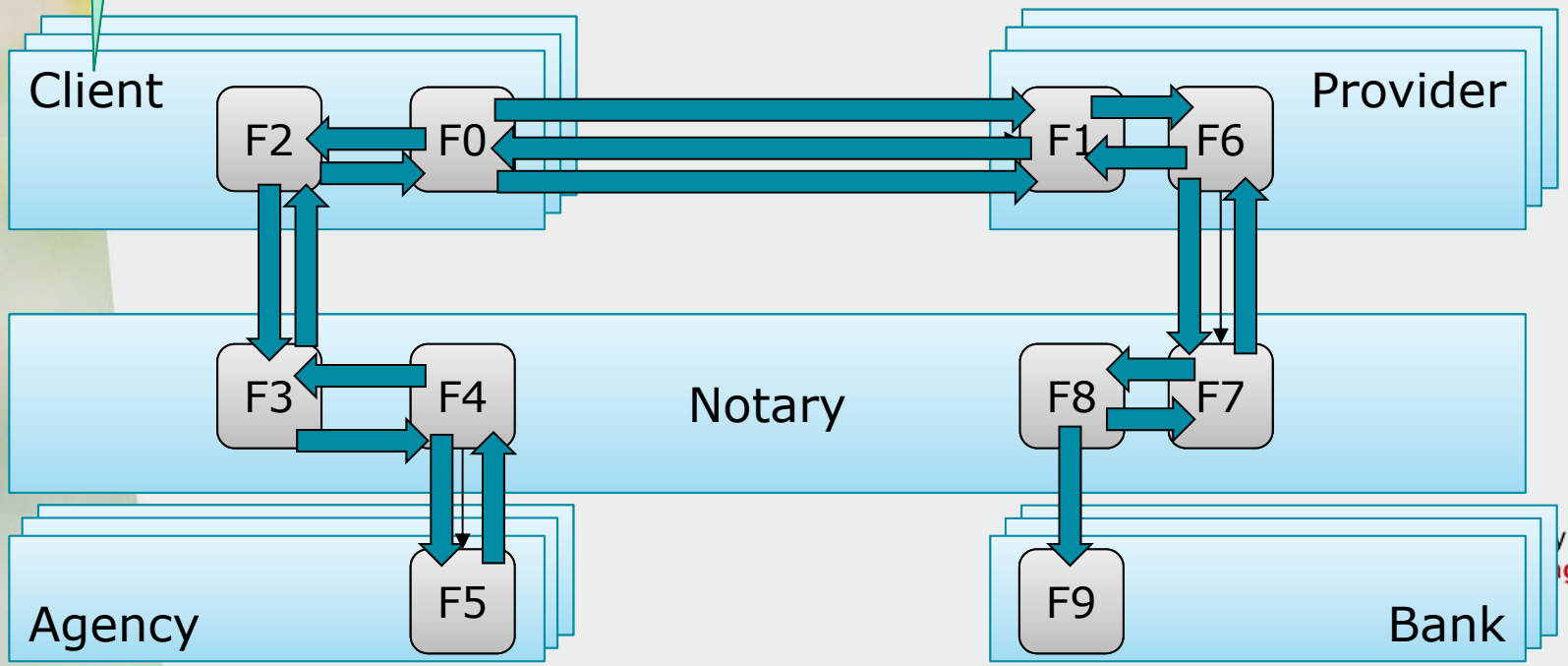
- Providers offer services to clients
  - Clients first need to be checked at **the** Notary
  - Clients send their approval to the provider
  - Payment is only done once the client signed the bill
  - Multiple payments per service possible
  - Payment is done by the notary
  - The notary can use one or more agencies to get more info



Multiple instances?

Suppose the following situation

- Providers offer services to clients
  - Clients first need to be checked at **the** Notary
  - Clients send their approval to the provider
  - Payment is only done once the client signed the bill
  - Multiple payments per service possible
  - Payment is done by the notary
  - The notary can use one or more agencies to get more info



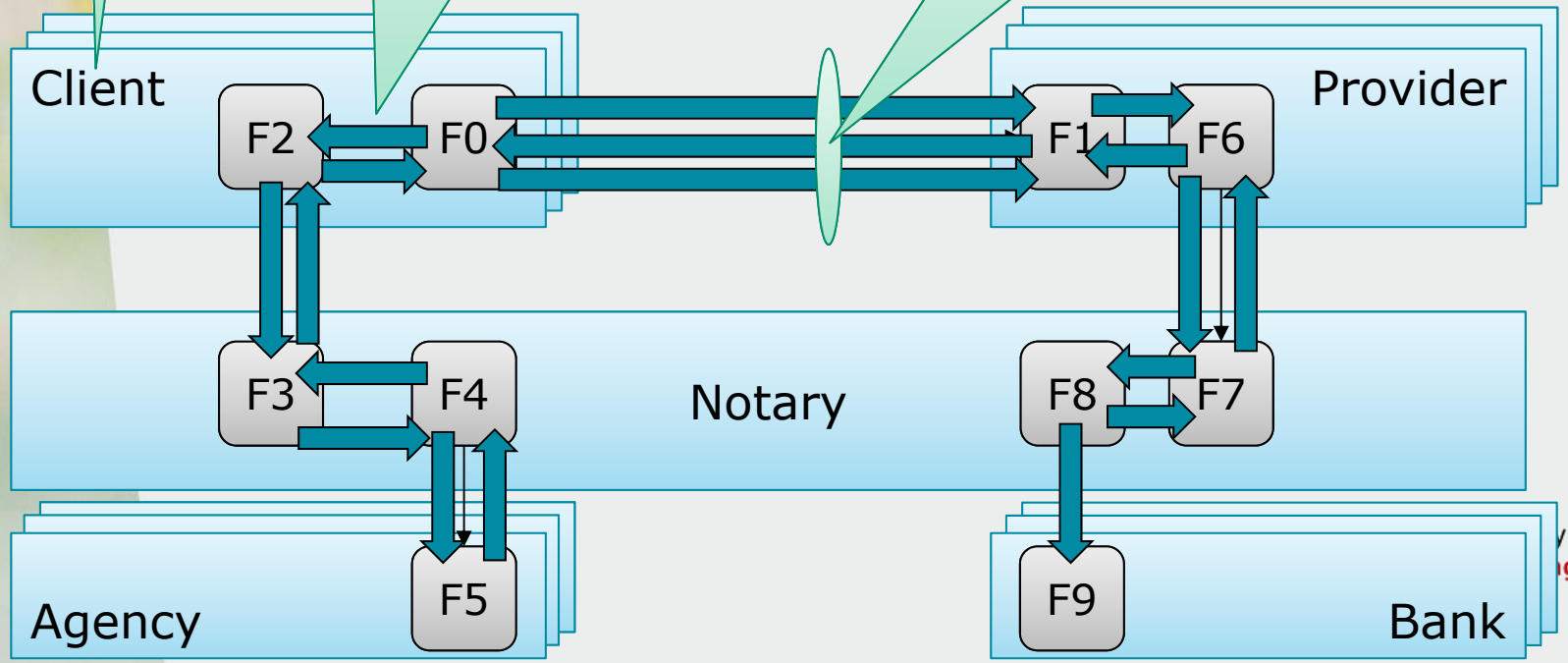
Suppose the following situation

Multiple instances?

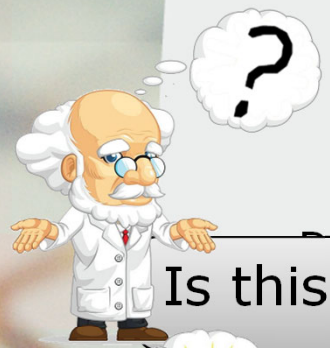
- Providers offer services to clients
  - Clients first need to be checked at **the** Notary
  - Clients send their approval to the provider
  - one once th
  - ments per service pos
  - Pay s done by the notary
  - The ry can use one or mor ncies to get more info

Dependency between features

There is a protocol here!



Suppose the following situation



Is this the only scenario?

o clients

hecked at **the** Notary

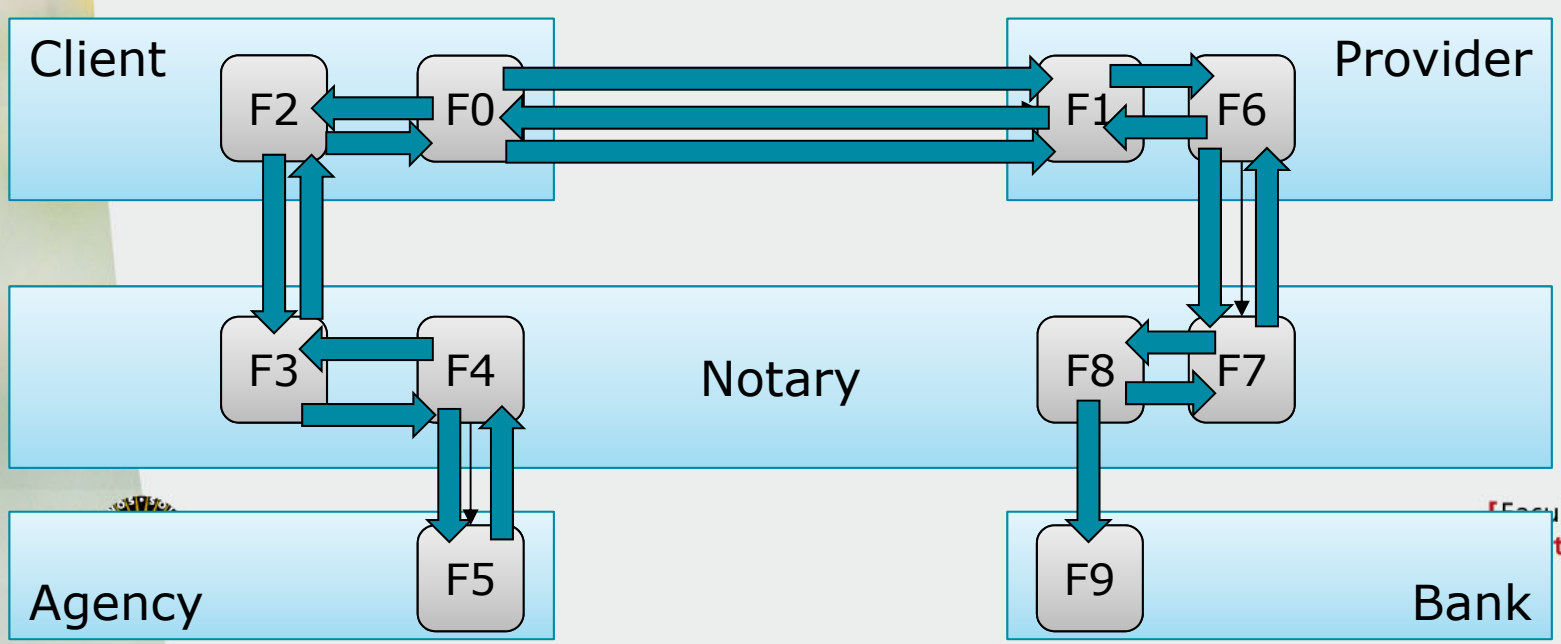
- Clients send their approval to the provider

Are all scenarios consistent with each other?

- Payment is done by the notary

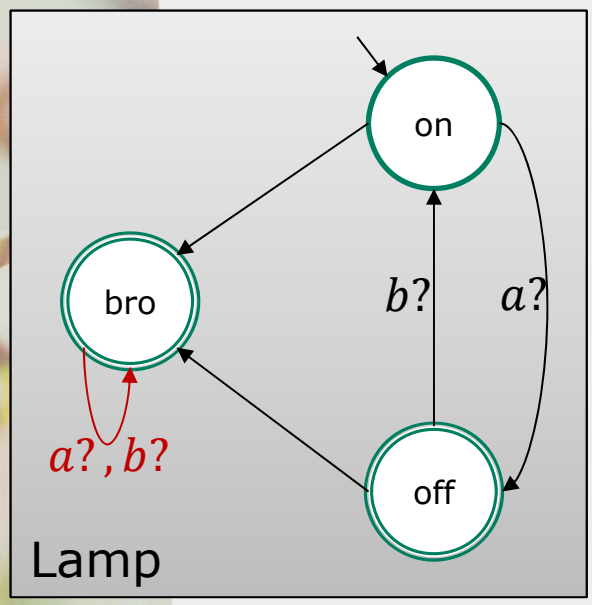
Formal methods to the rescue!

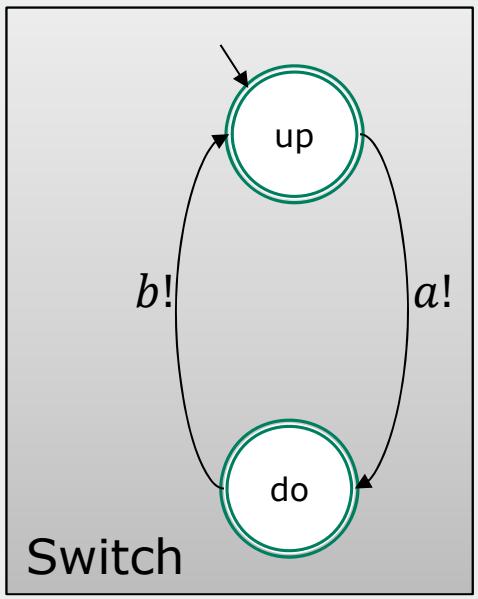
e info

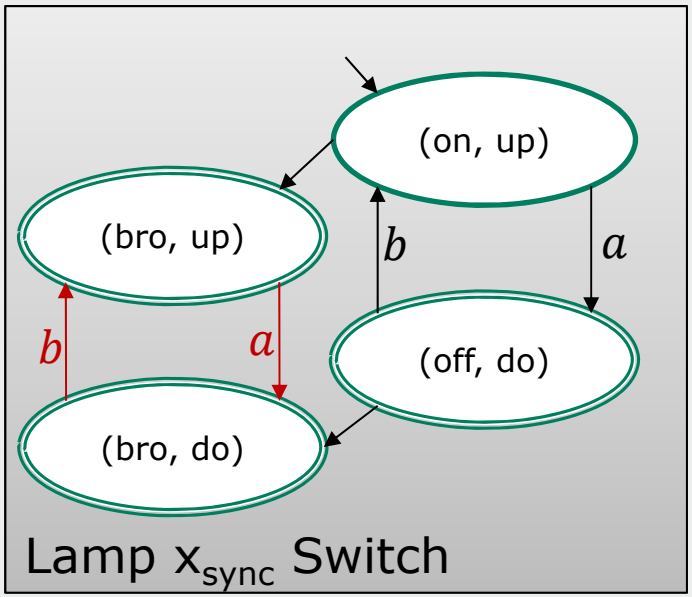


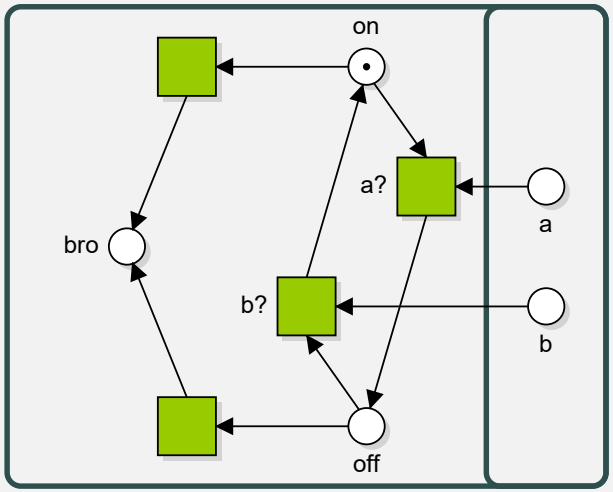
[Faculty of Science
ting Sciences]

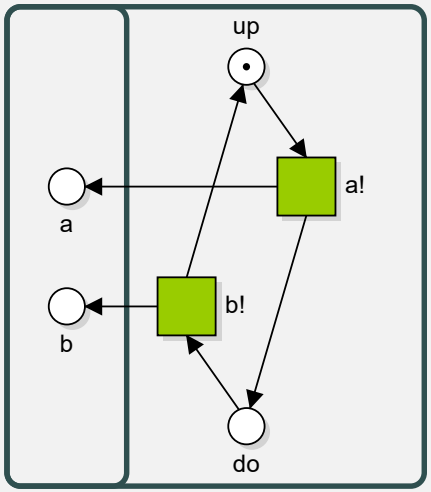
Formal methods to analyze communication



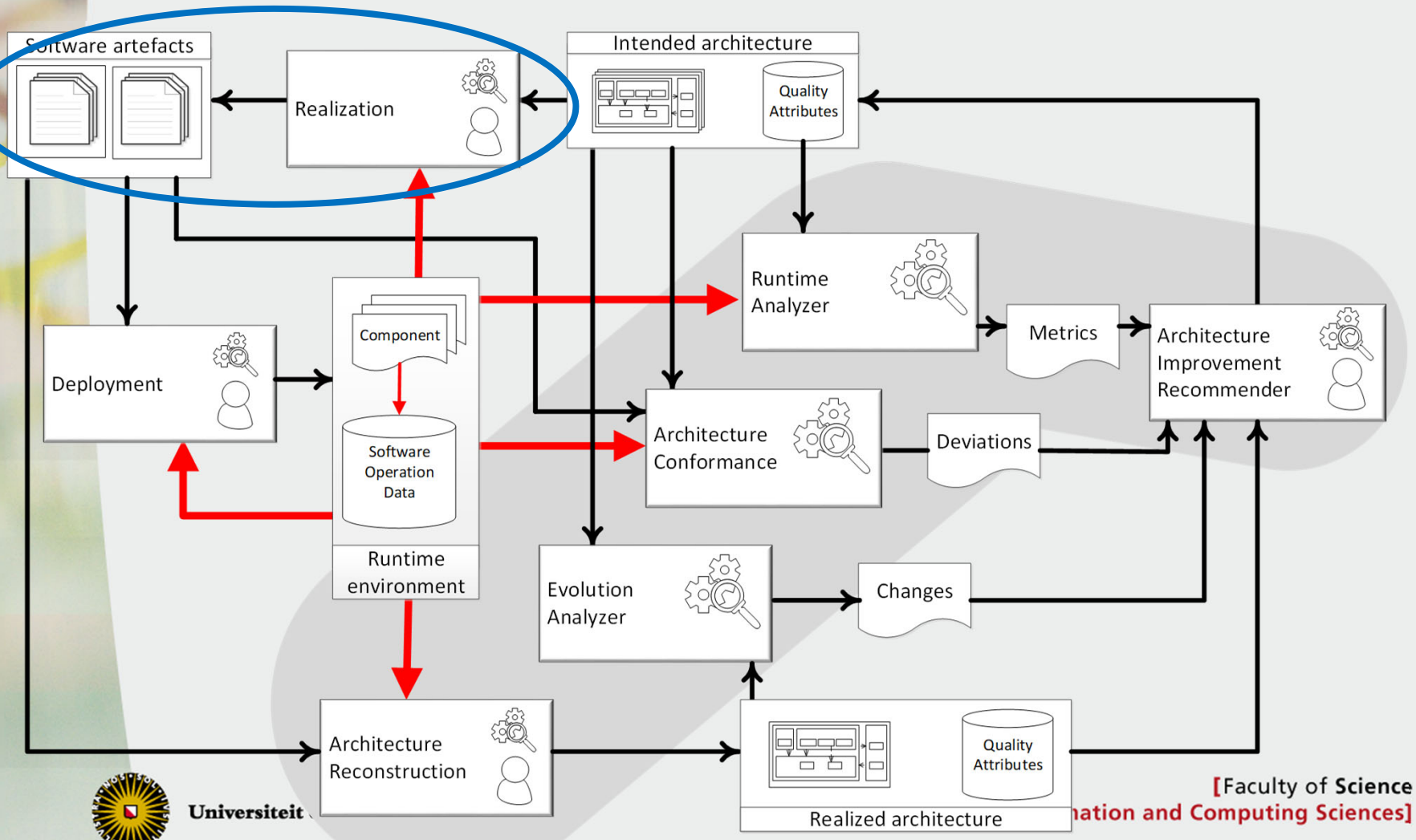






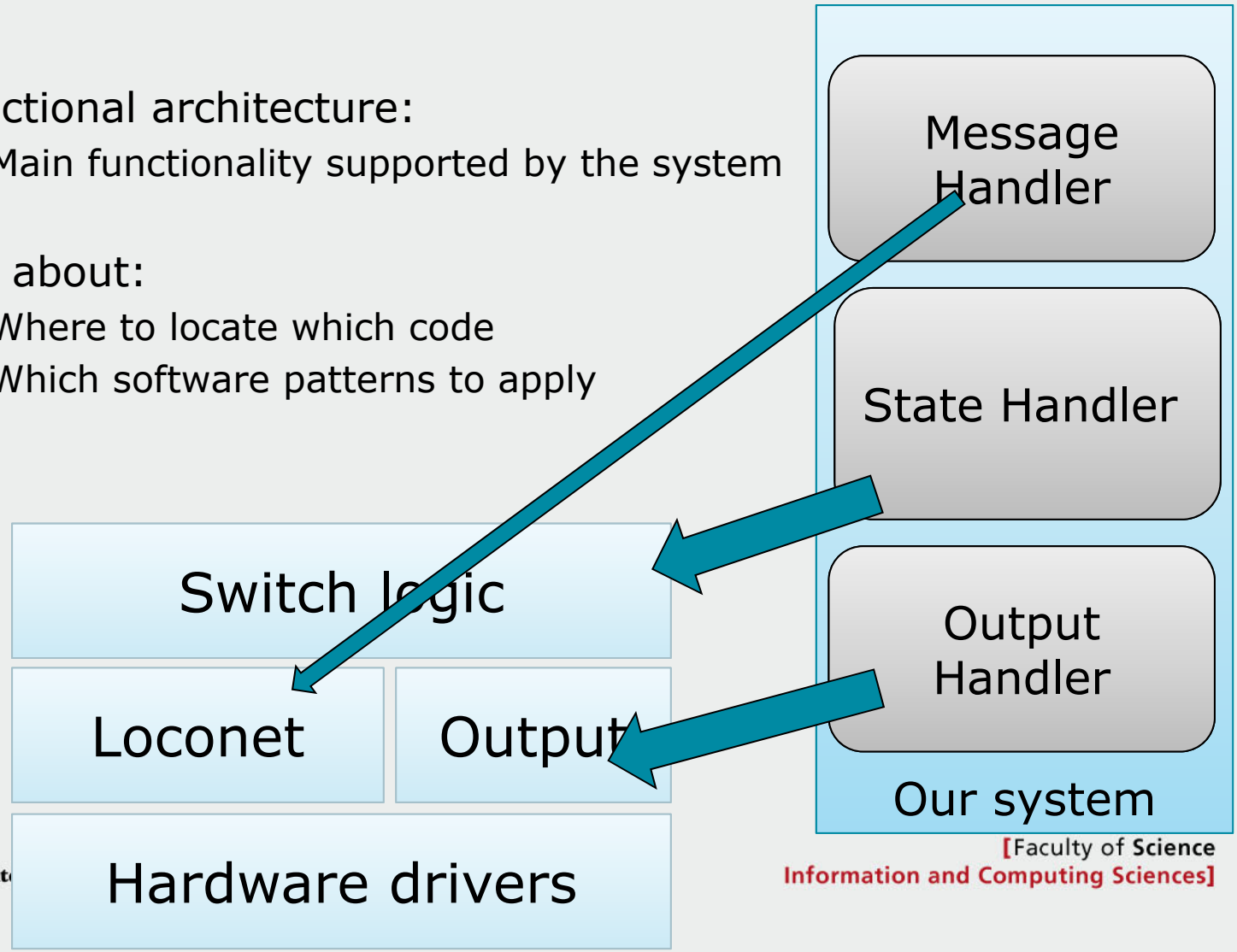


Software Architecture: overview of the field

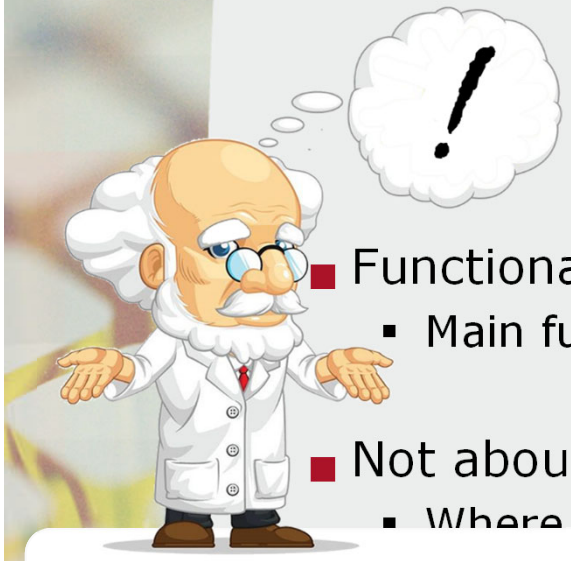


Software architecture & source code

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

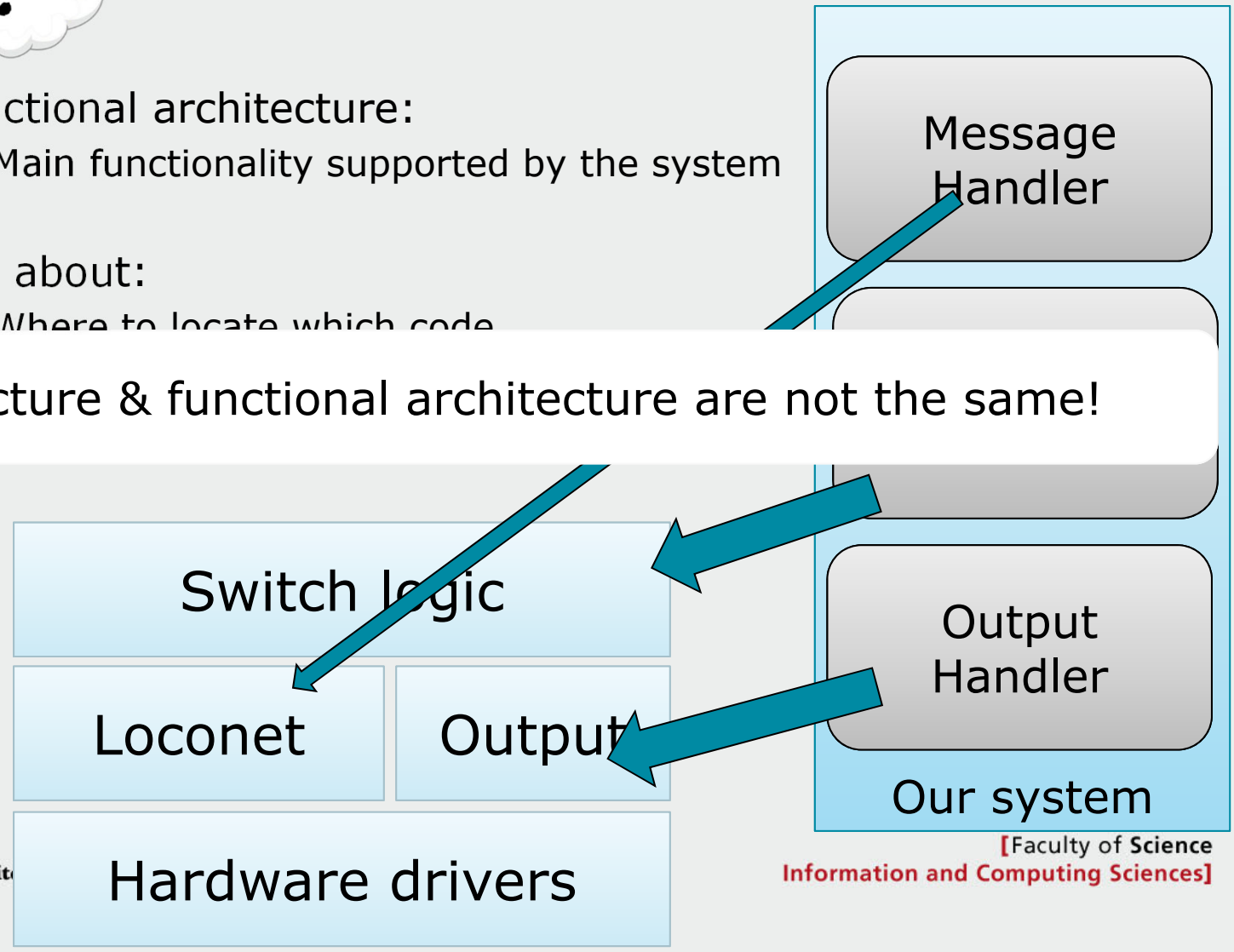


Software architecture & source code



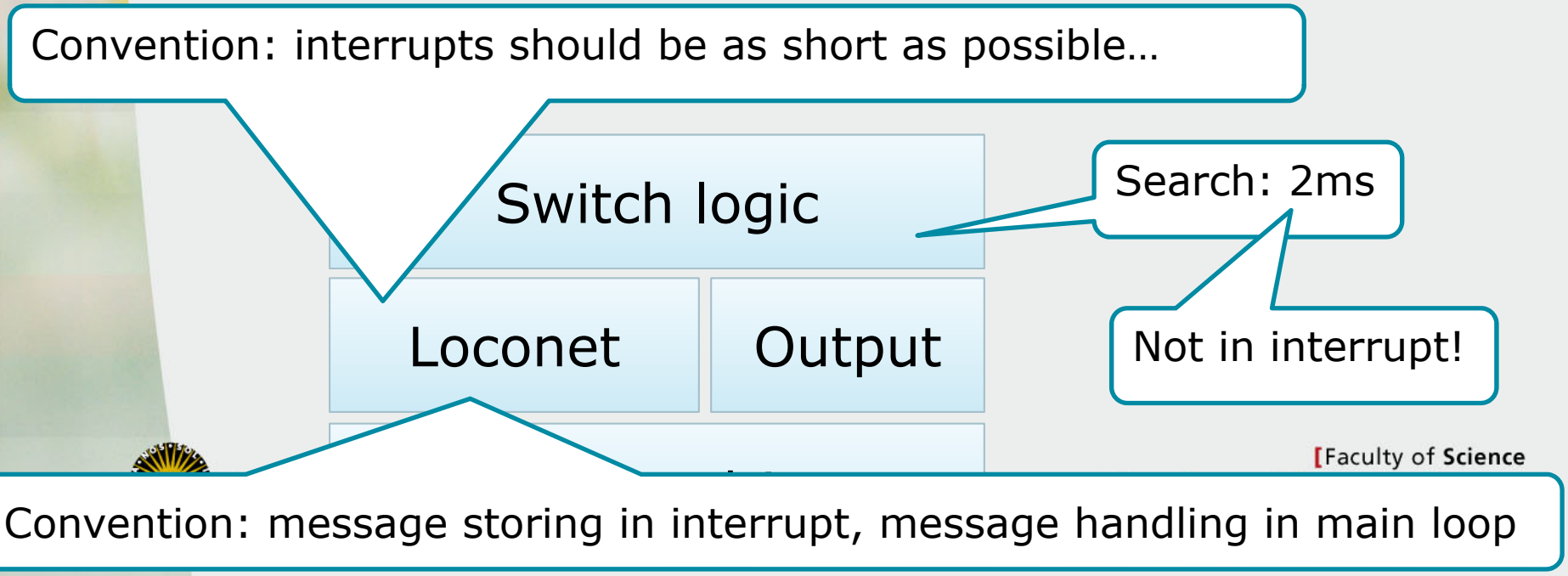
- Functional architecture:
  - Main functionality supported by the system
- Not about:
  - Where to locate which code

Code structure & functional architecture are not the same!



Architecture: guidelines to follow

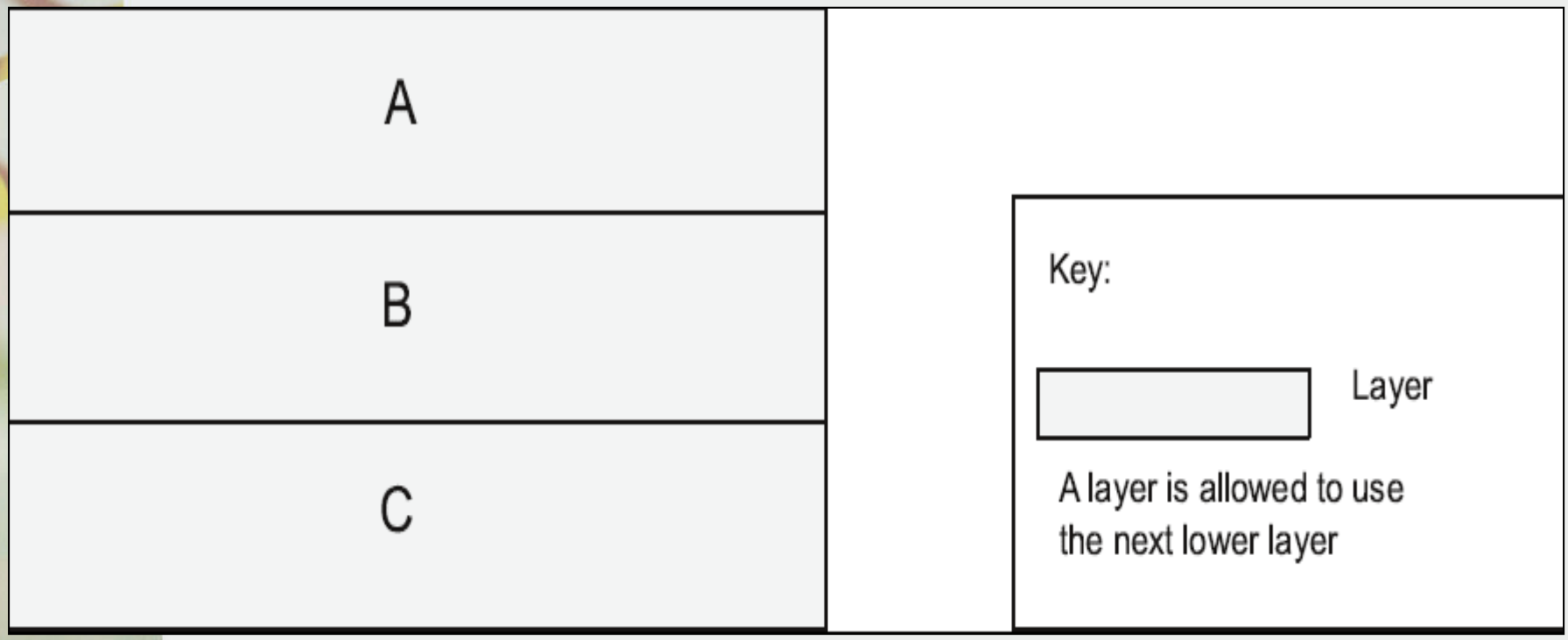
- Part of the Developer viewpoint:
  - Code templates
  - Programming conventions
  - Frameworks…
  - Development process (agile, SCRUM, testing, …)



Architectural styles & patterns

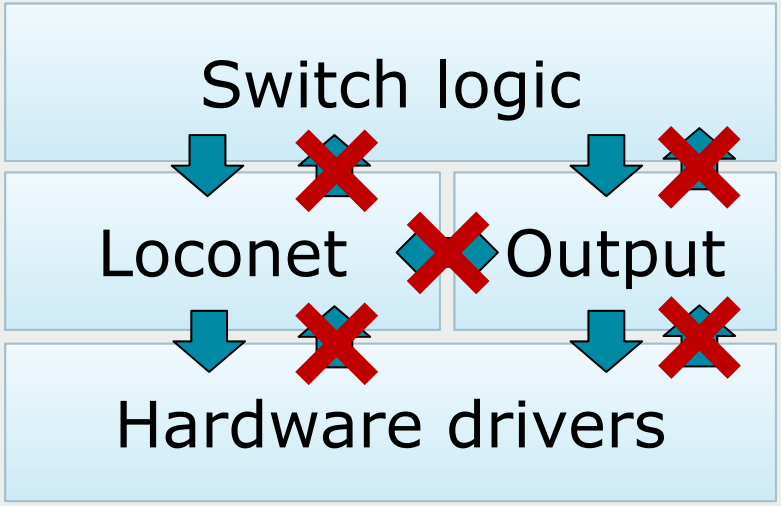
- Architectural style: fundamental structural organization schema for software systems
- Pattern: documents commonly recurring and proven structure
- Think of it as in language idiom: idiom describes the patterns of the language

Layer pattern: topoligical layout



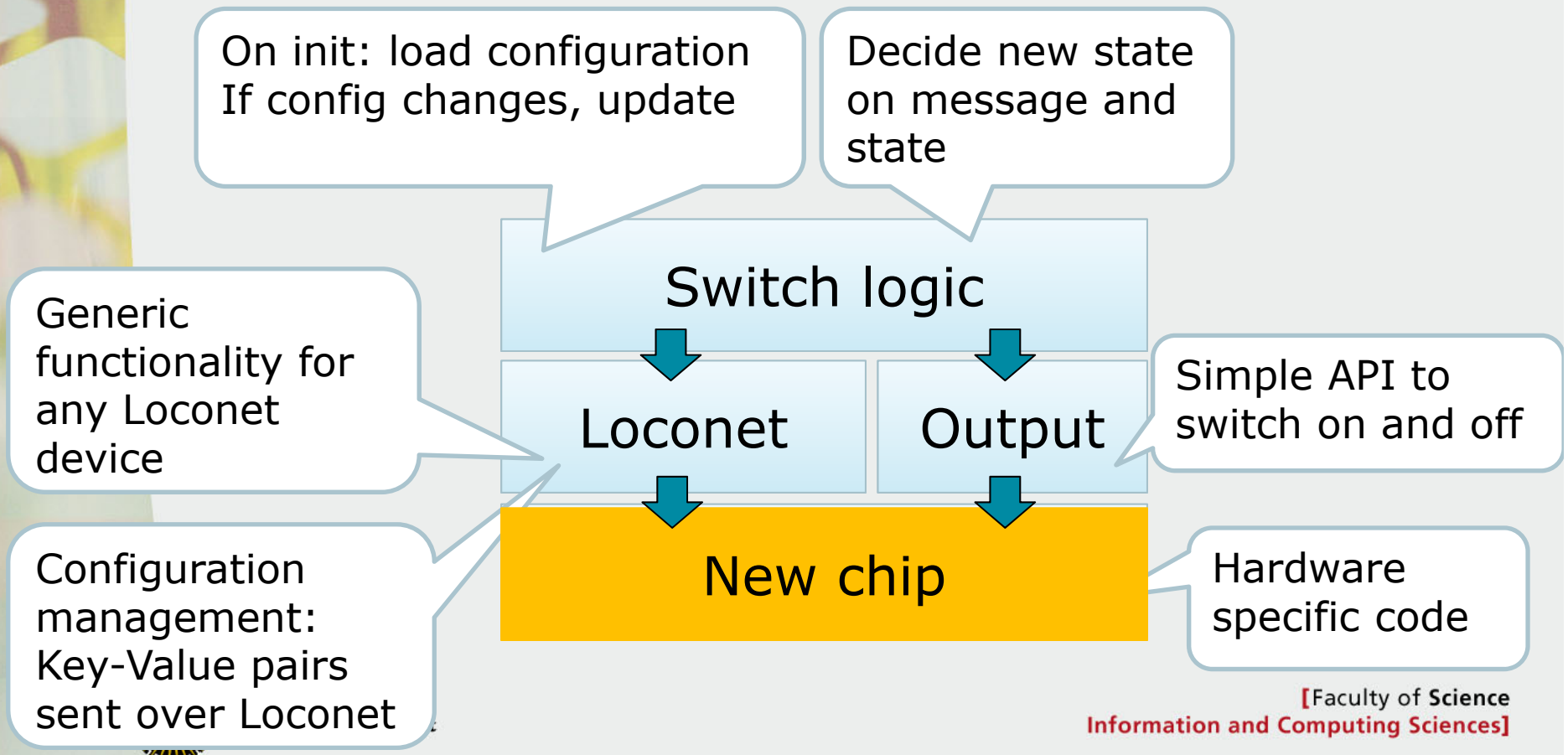
Layers: back to our example…

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



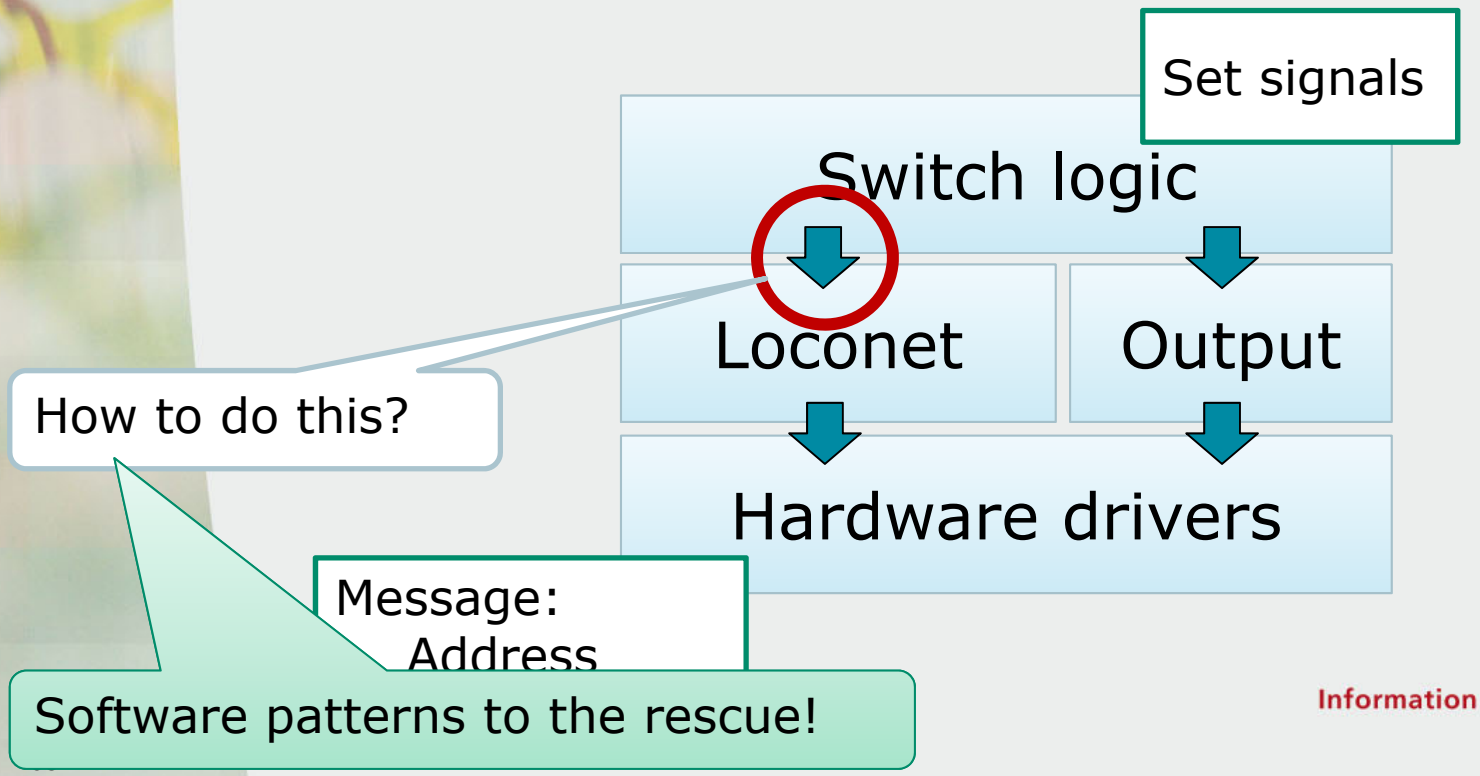
Layers: back to our example…

- 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 d

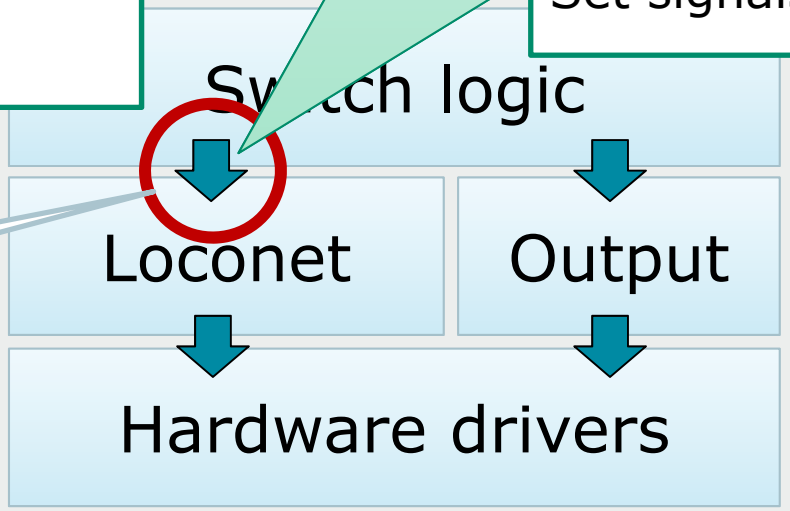
```
extern bool ln_register(uint8_t msgType, ln_fun
fun*)

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)

Set signals

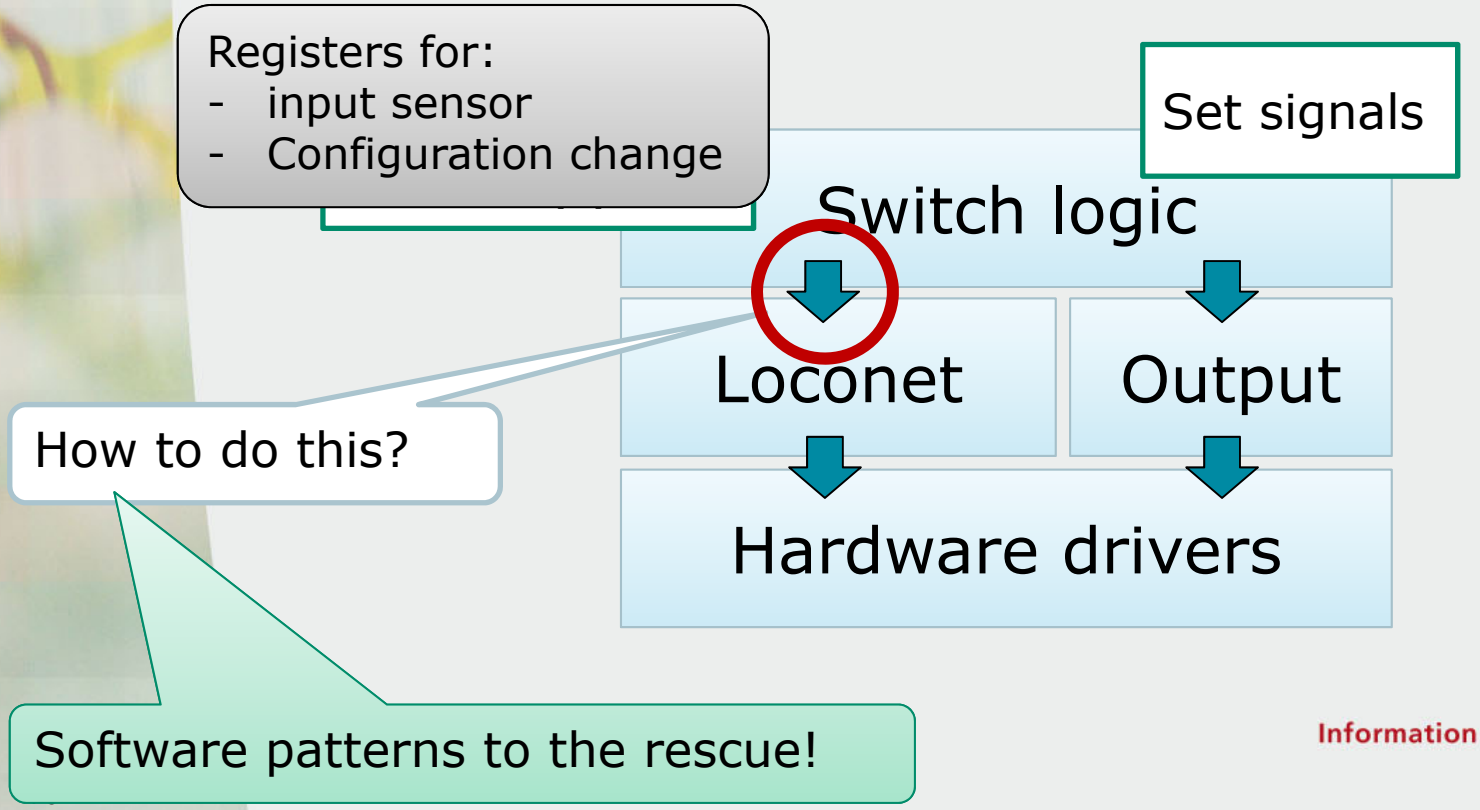


How to do this?

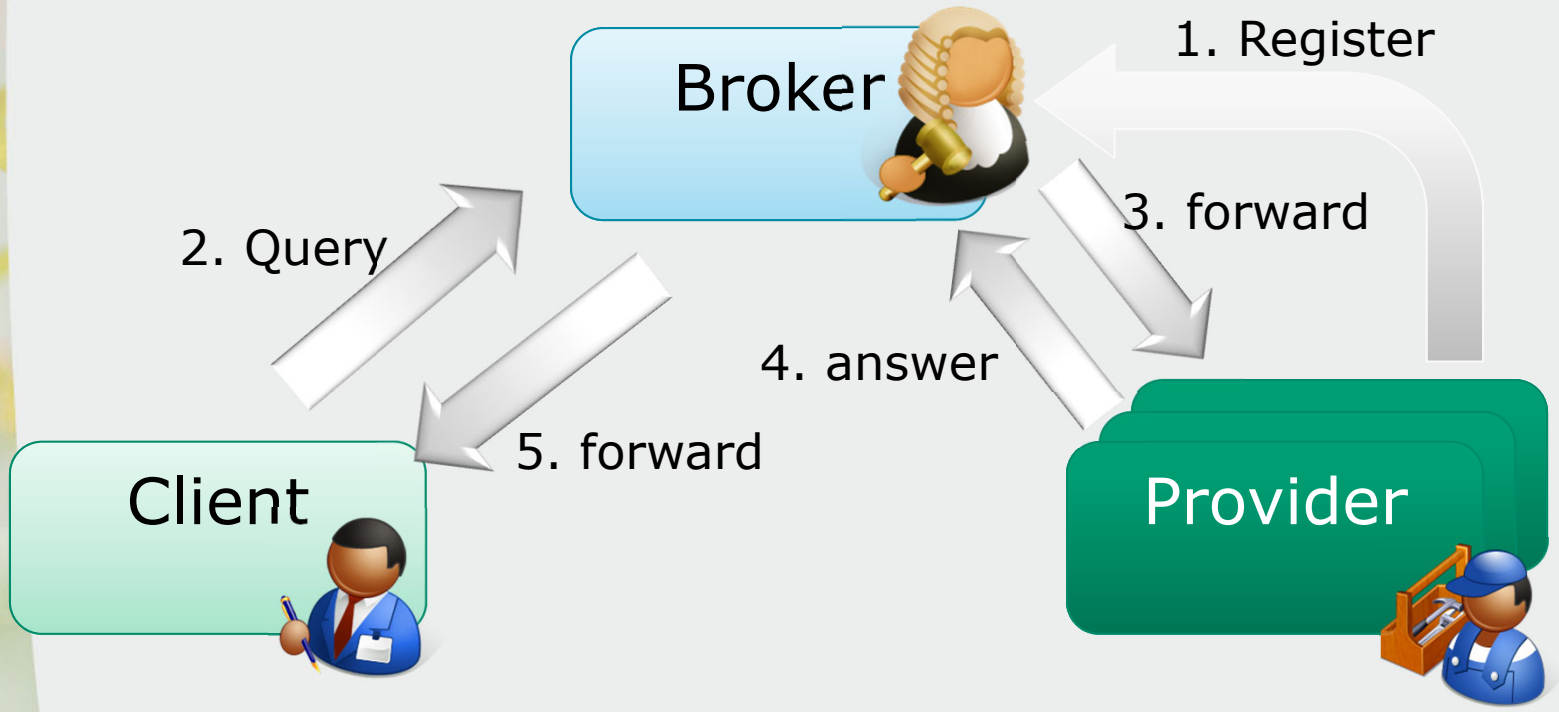
Software patterns to the rescue!

Layers: back to our example...

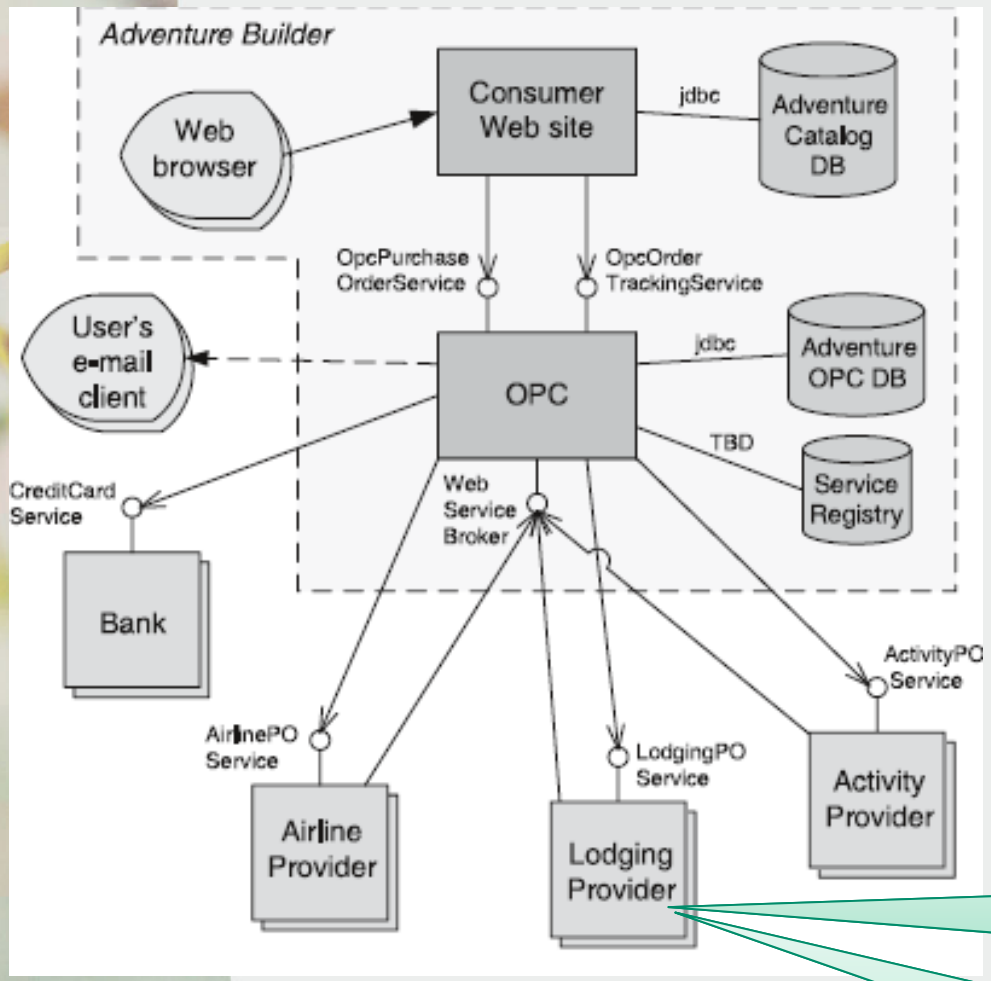
- Idea: lower layers do not know anything of higher layers

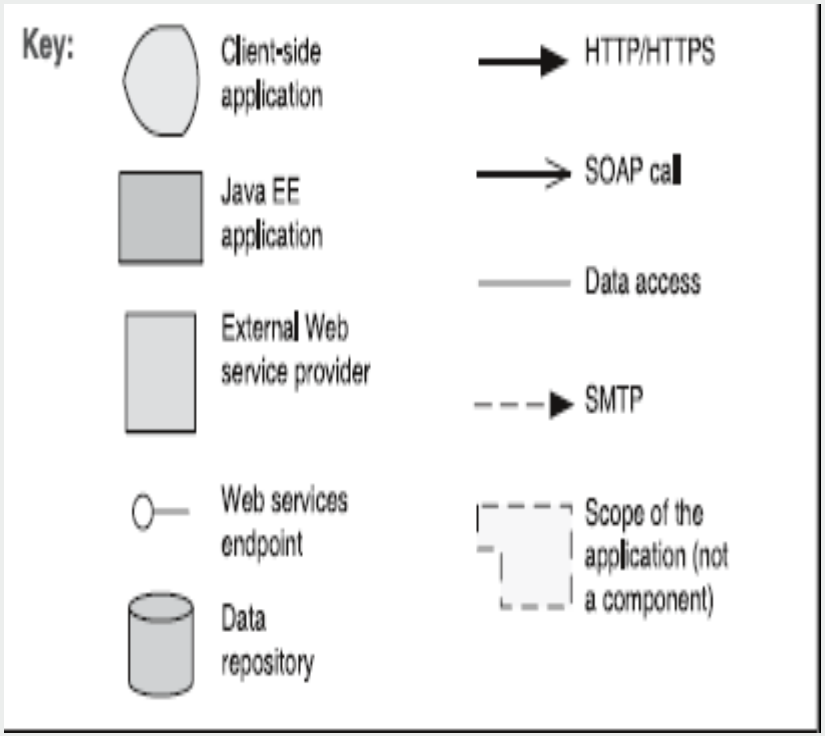


Broker pattern



Service Oriented Architecture pattern



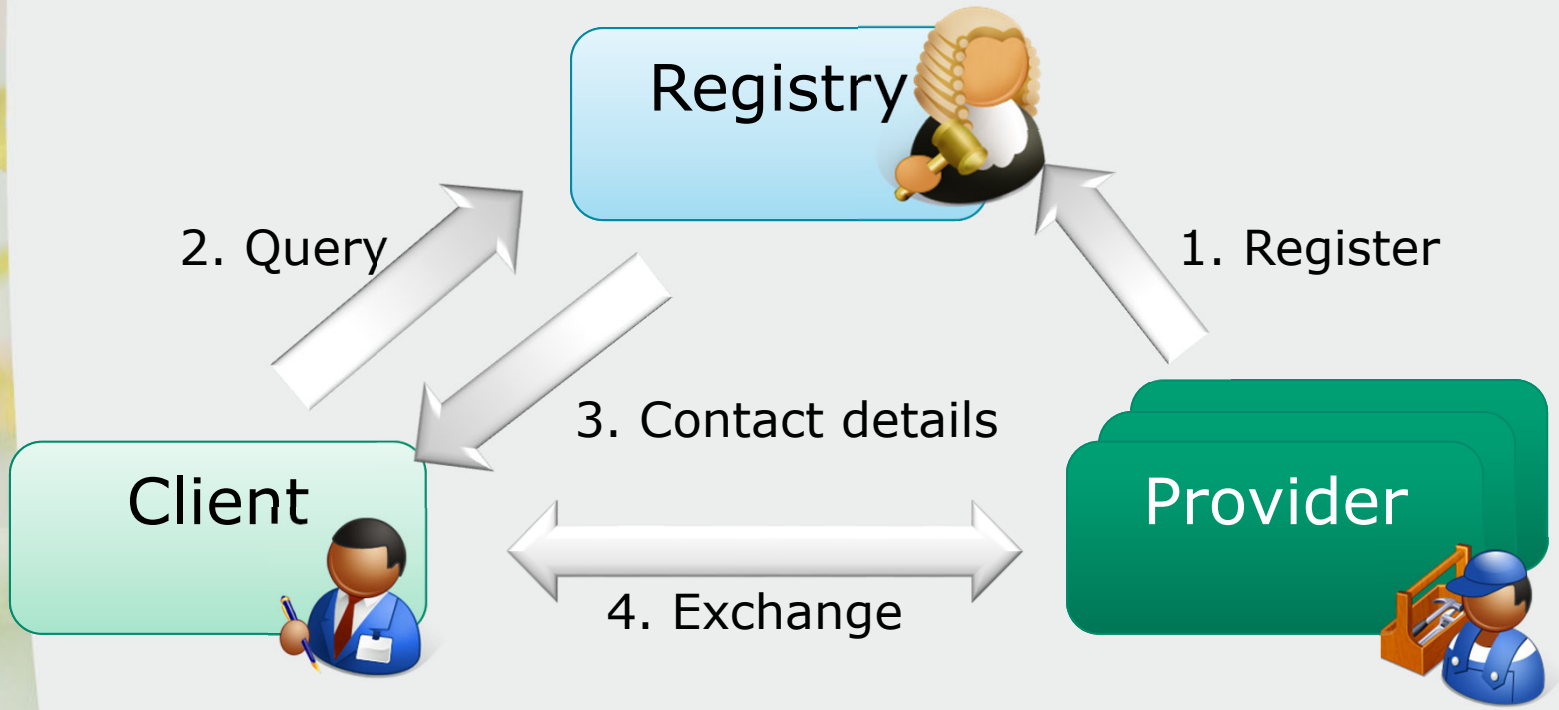


Where are these services located?

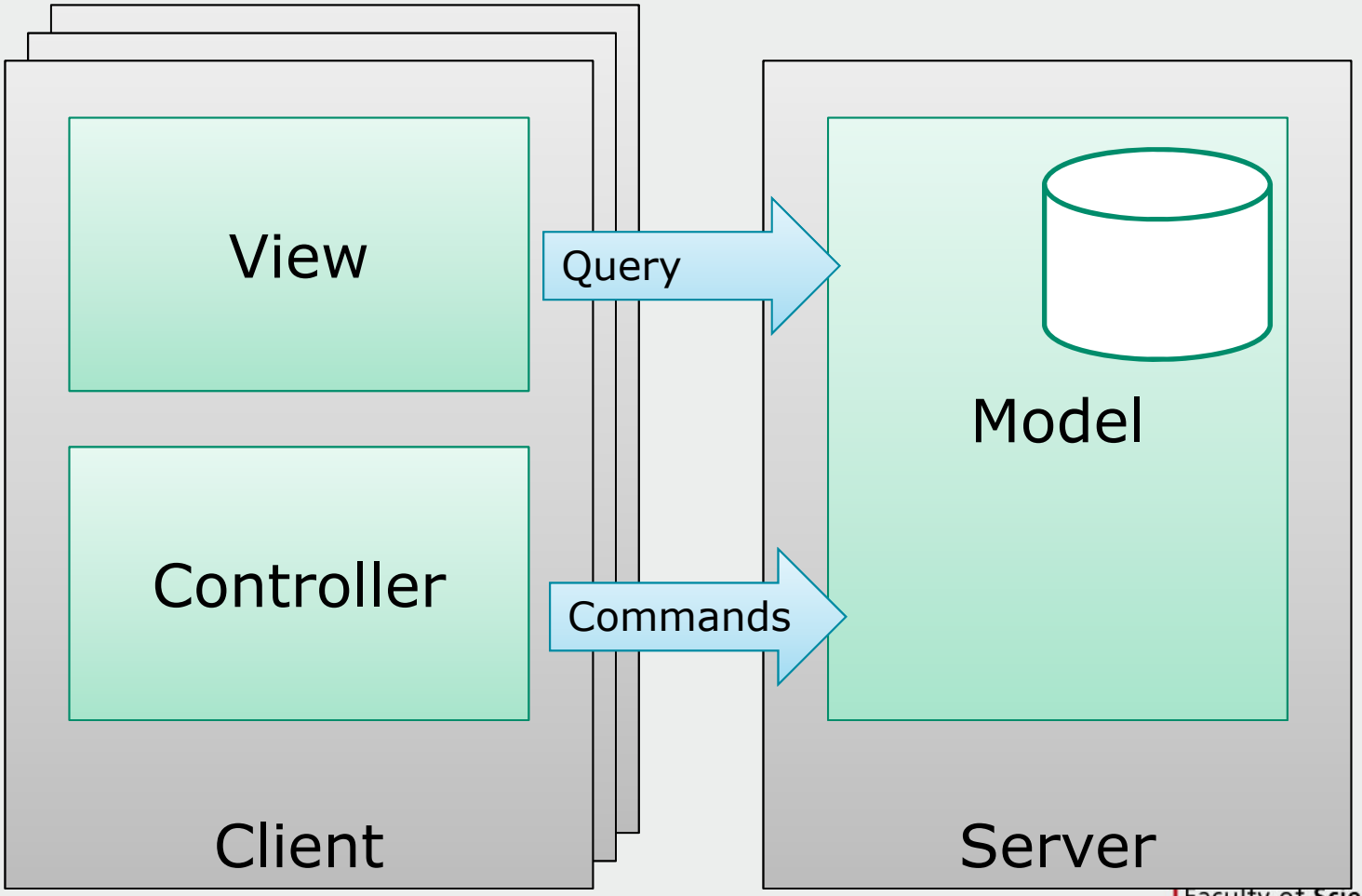
How to find these services?

Universiteit Utrecht

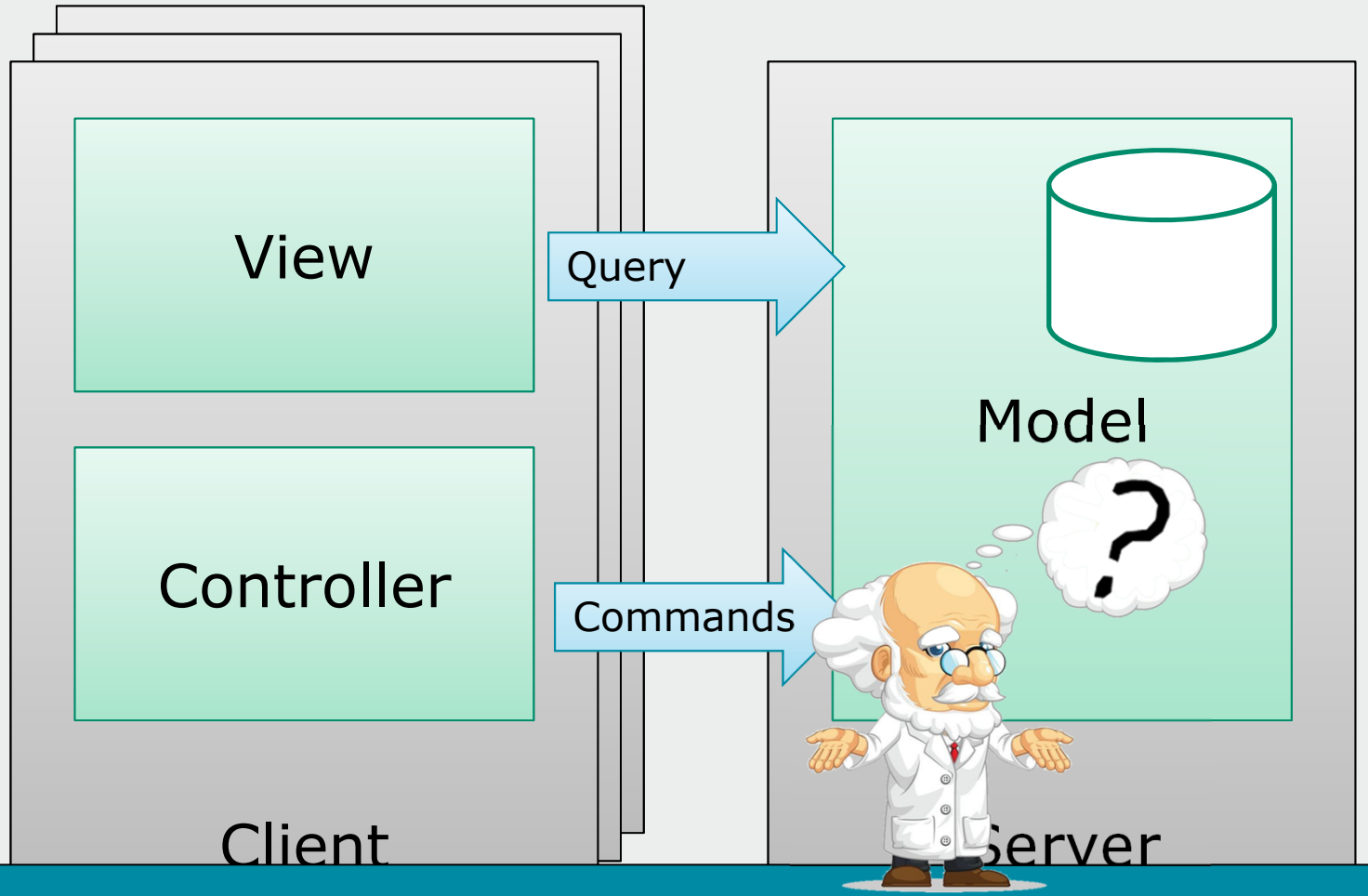
Registry in SOA



Model-View-Controller

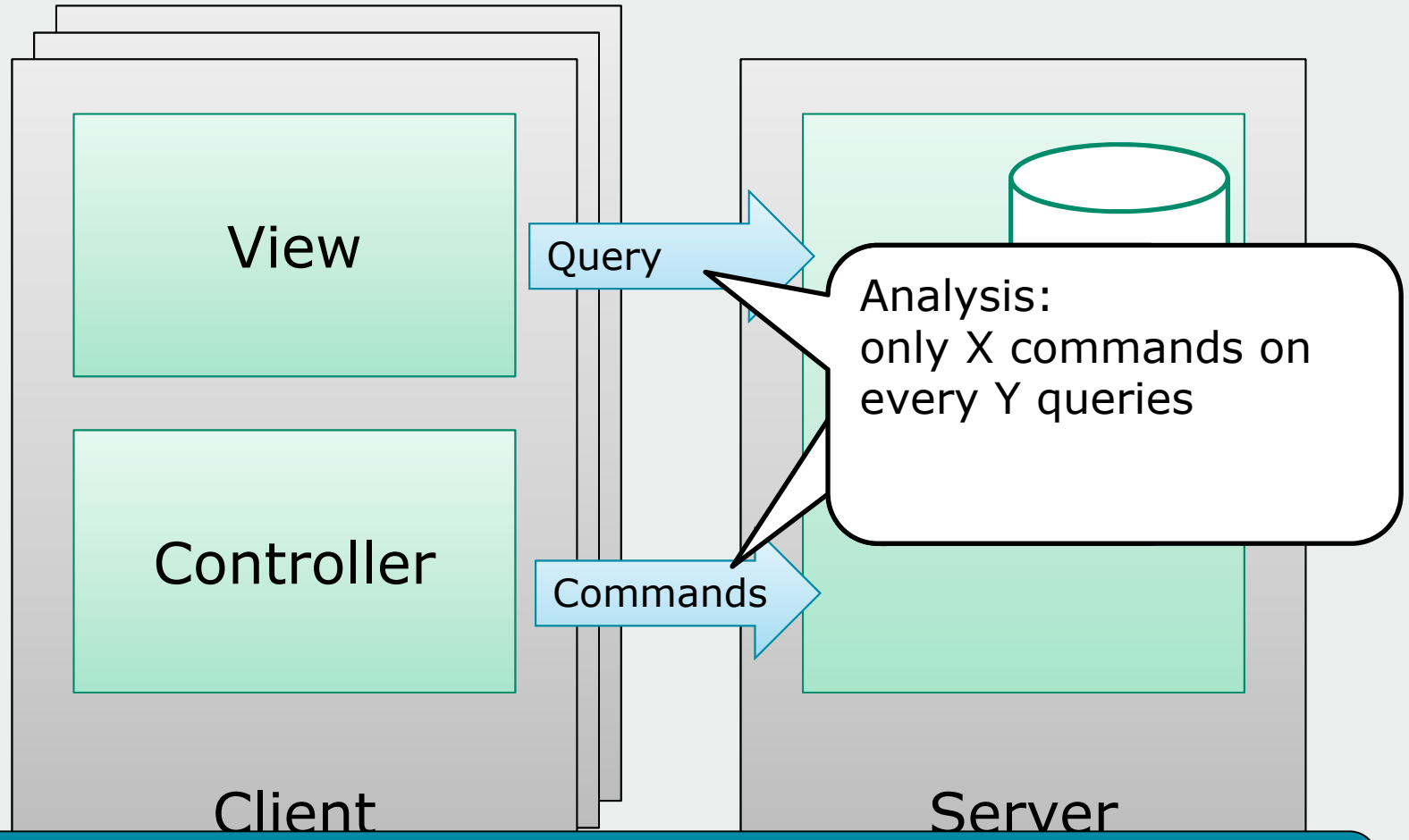


Model-View-Controller



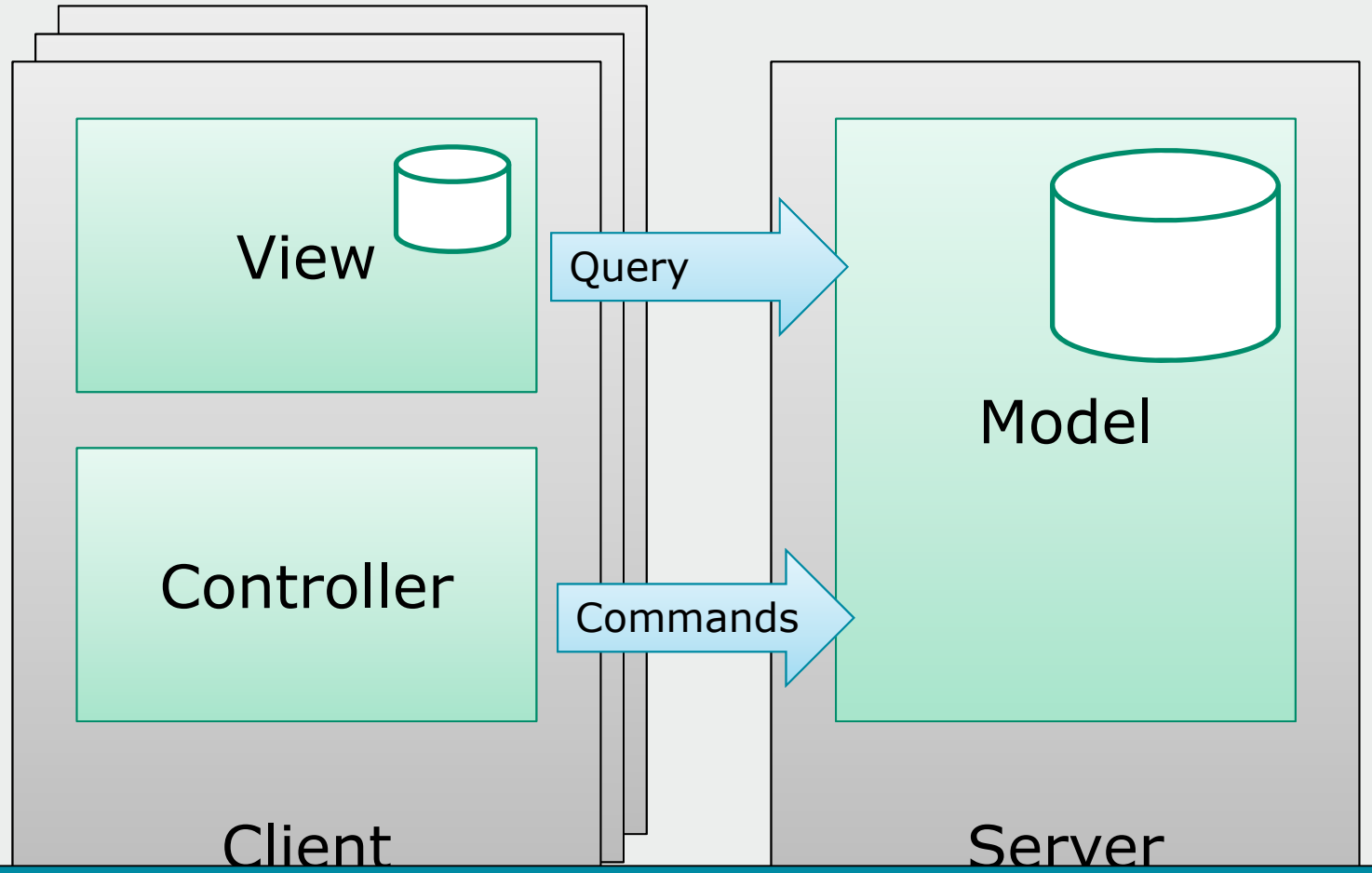
What if there are 10.000 clients running concurrently?

Model-View-Controller



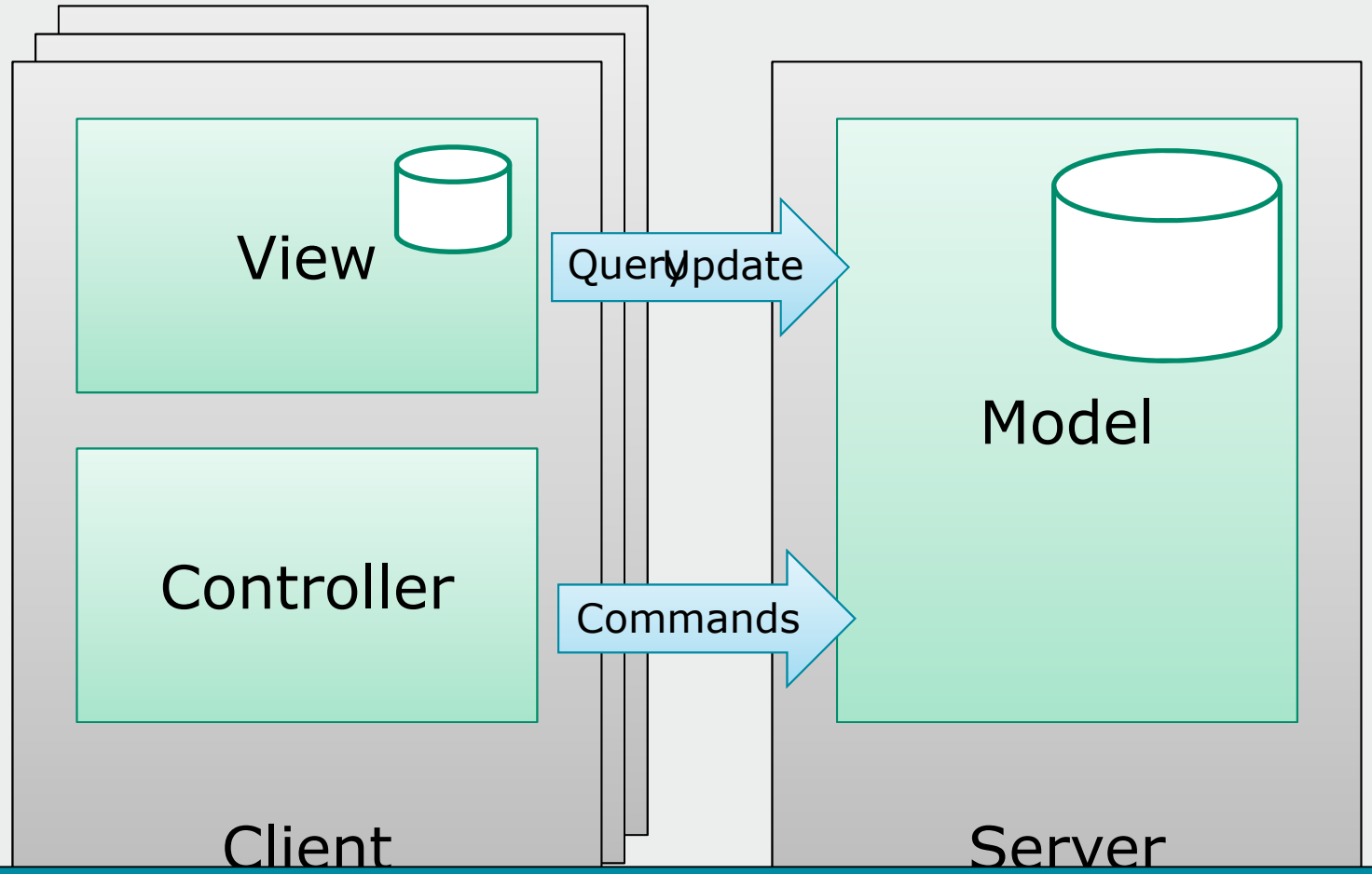
What if there are 10.000 clients running concurrently?

Model-View-Controller



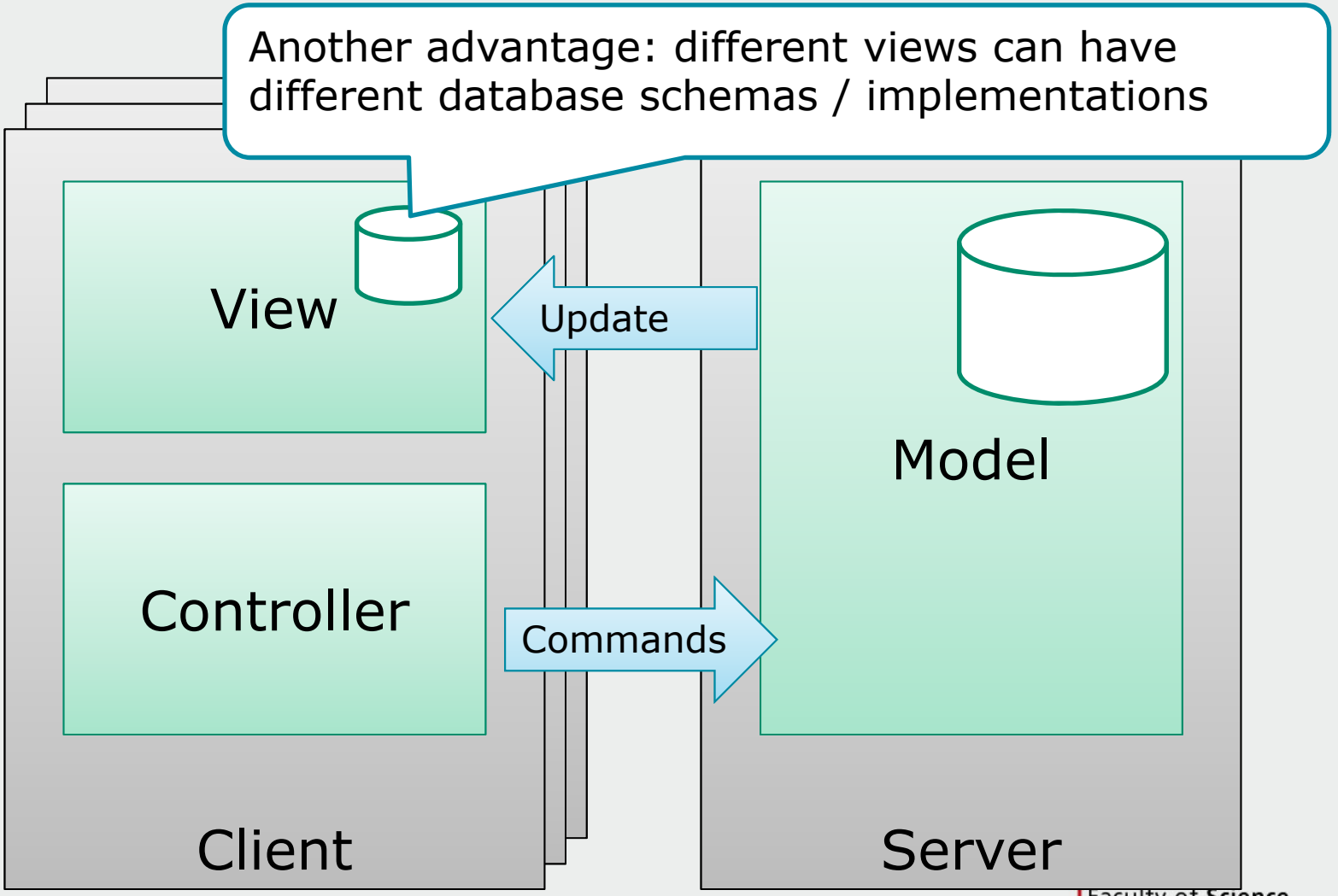
What if there are 10.000 clients running concurrently?

Model-View-Controller

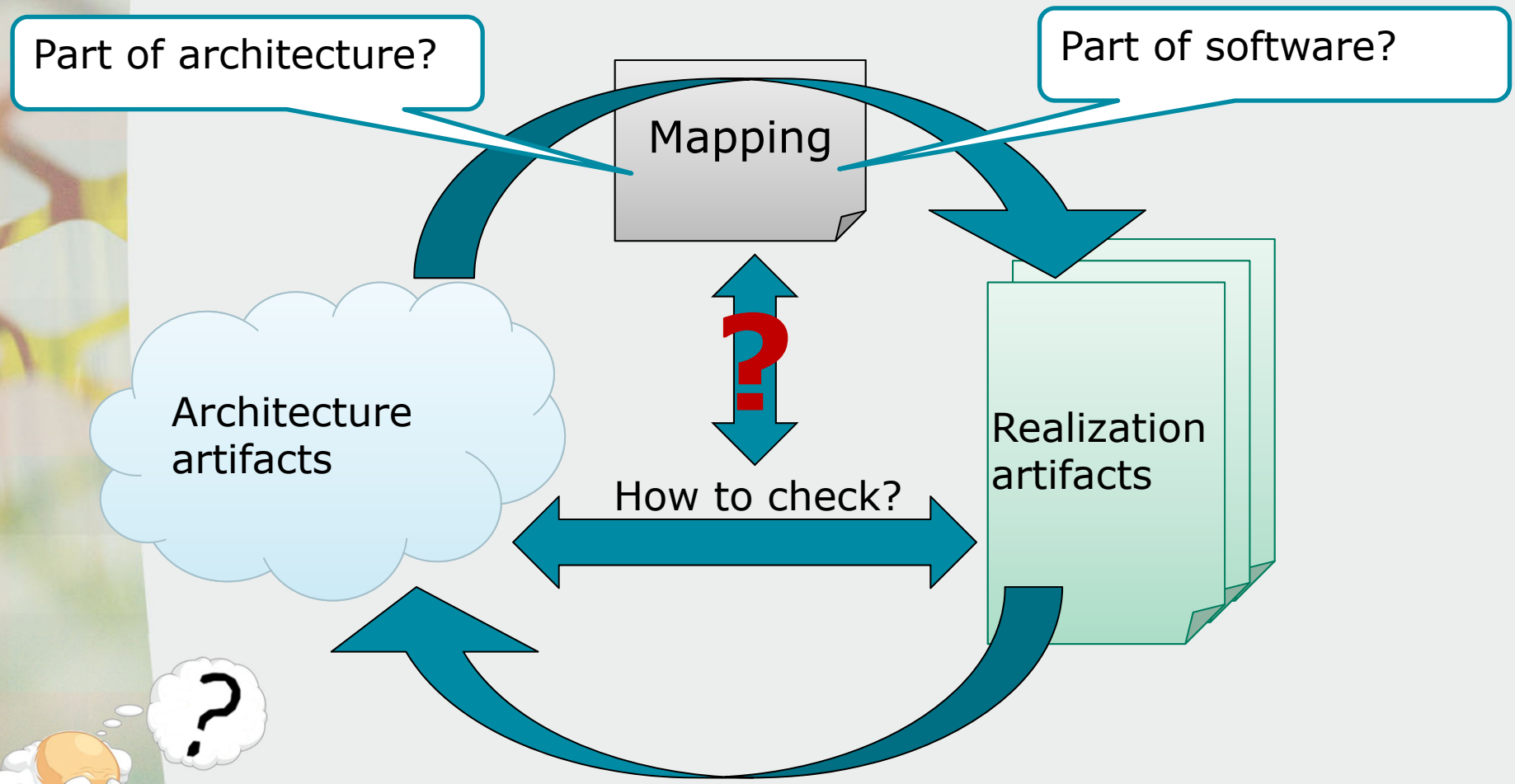


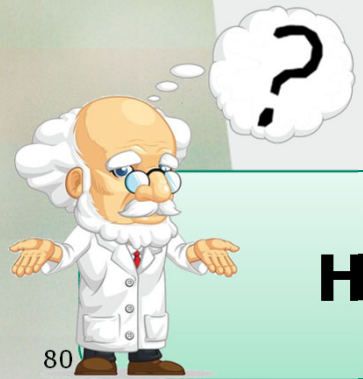
What if there are 10.000 clients running concurrently?

CQRS: Command Query Responsibility Segregation



Software architecture & source code





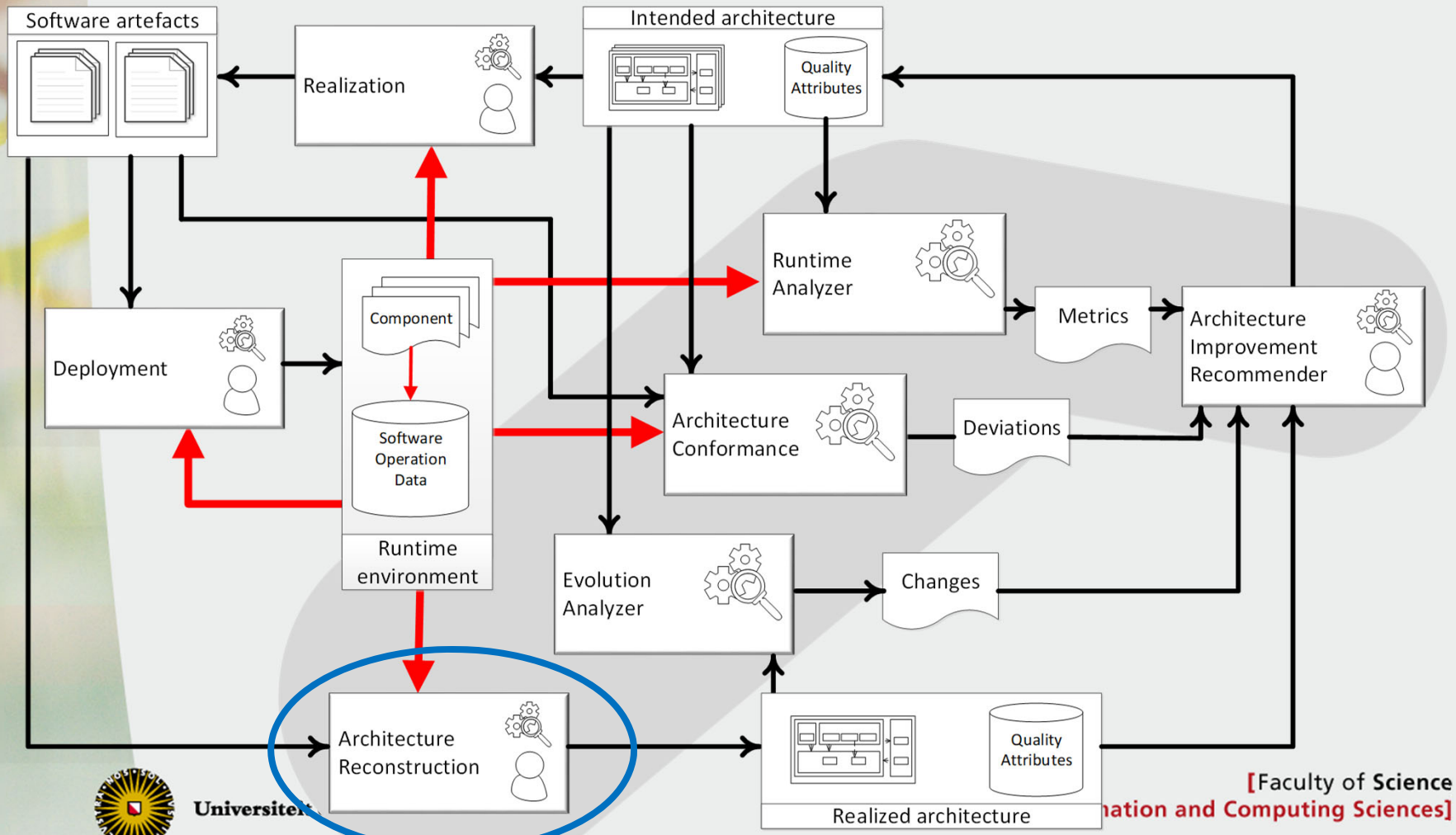
How to keep those consistent?

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

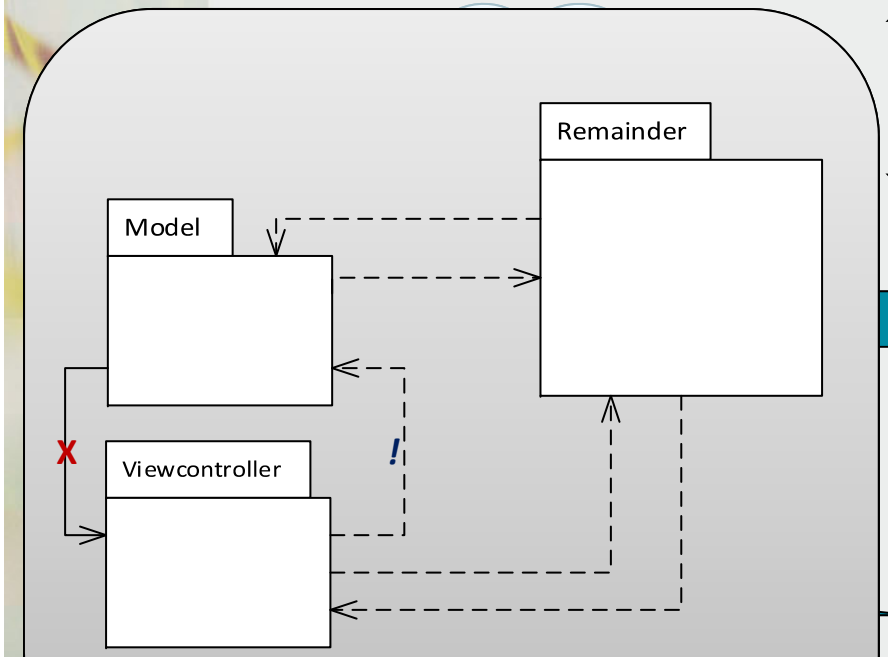
Architecture mining:
providing objective insights to the architect

Software Architecture: overview of the field

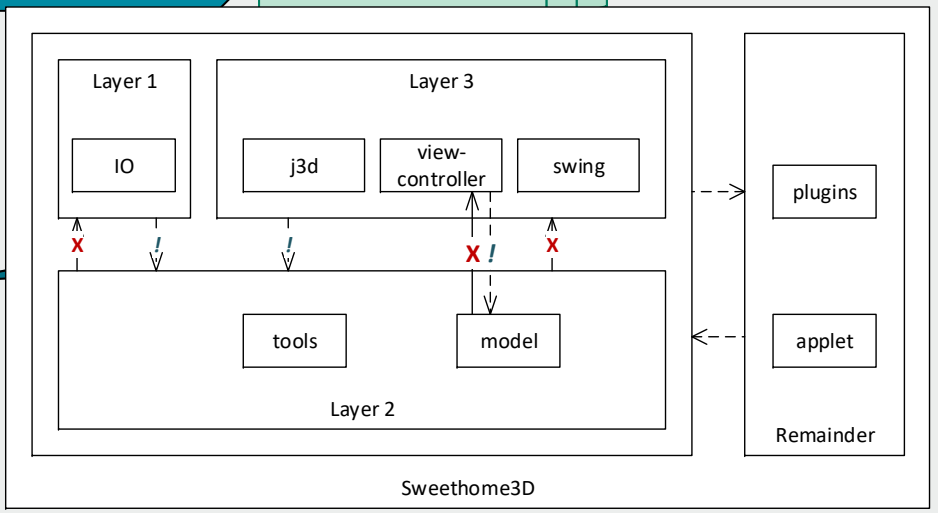


A Genetic Approach to Architectural Pattern Discovery (MSc. project of Joeri Peters, MBI)

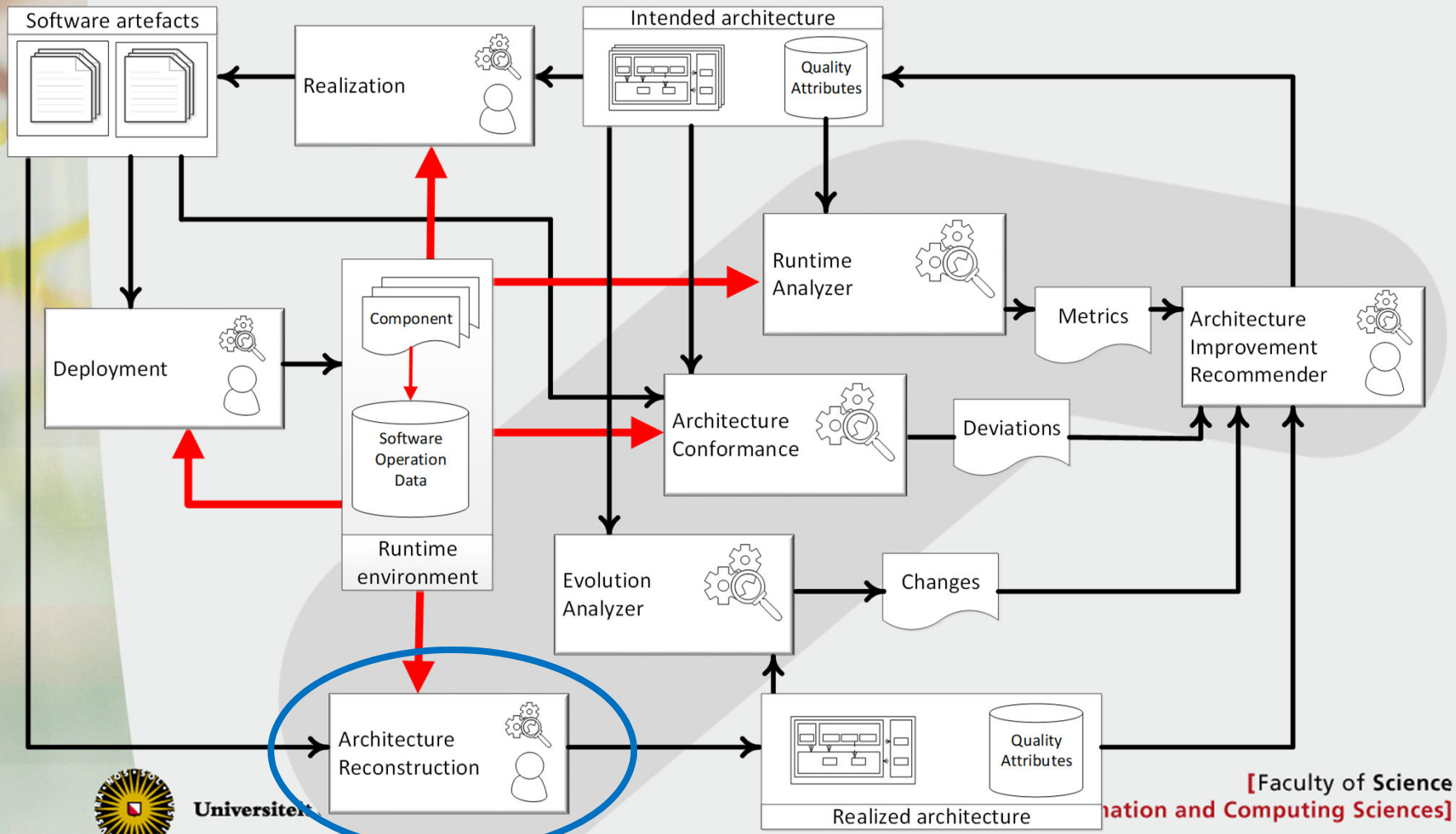
Genetic algorithm to generate an optimal mapping!



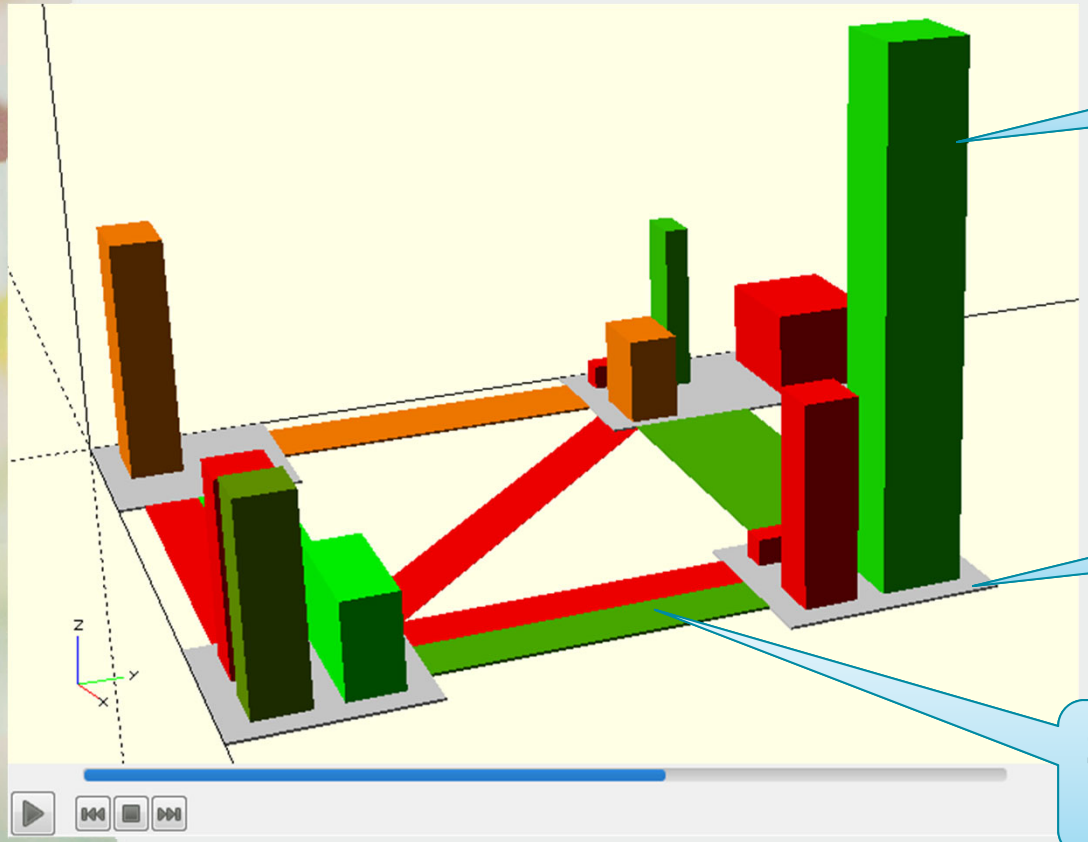
What is an architecture pattern?



Software Architecture: overview of the field



ArchitectureCity
(MSc-project Rens Rooimans , COSC)

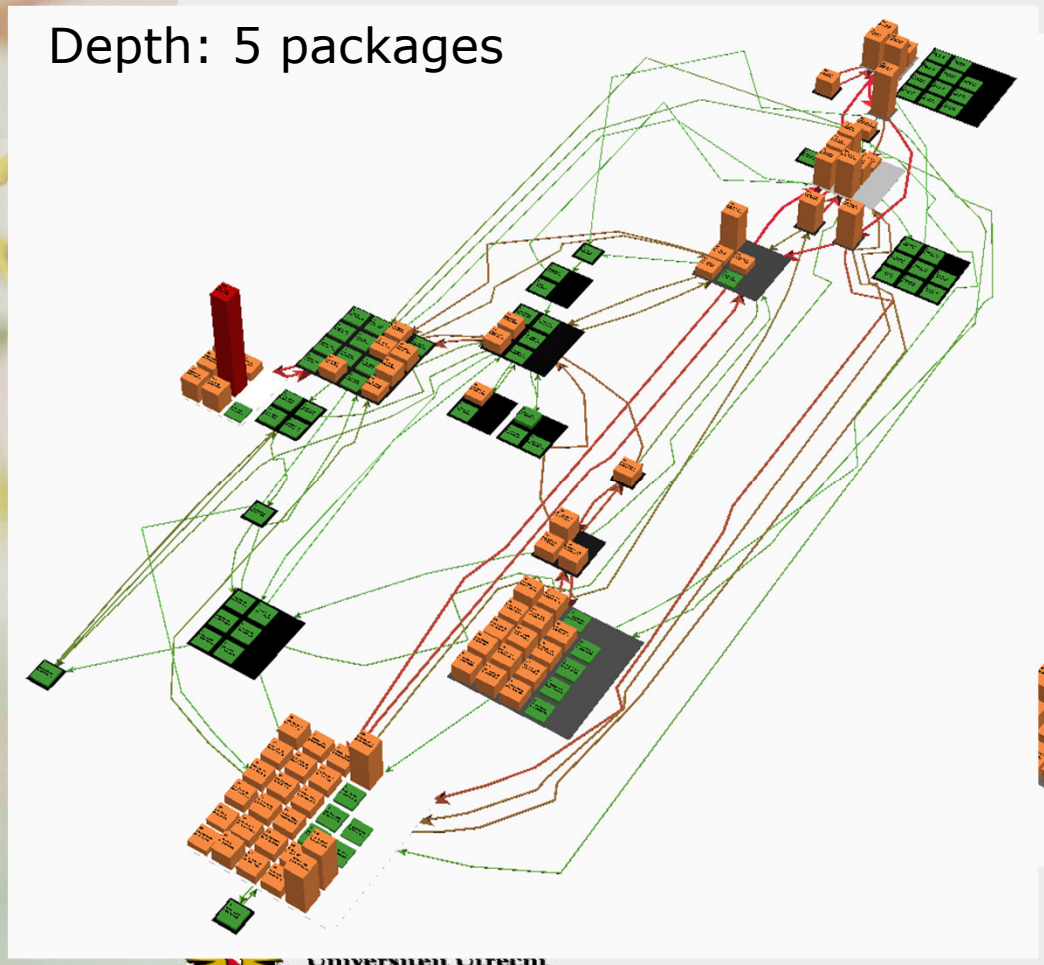


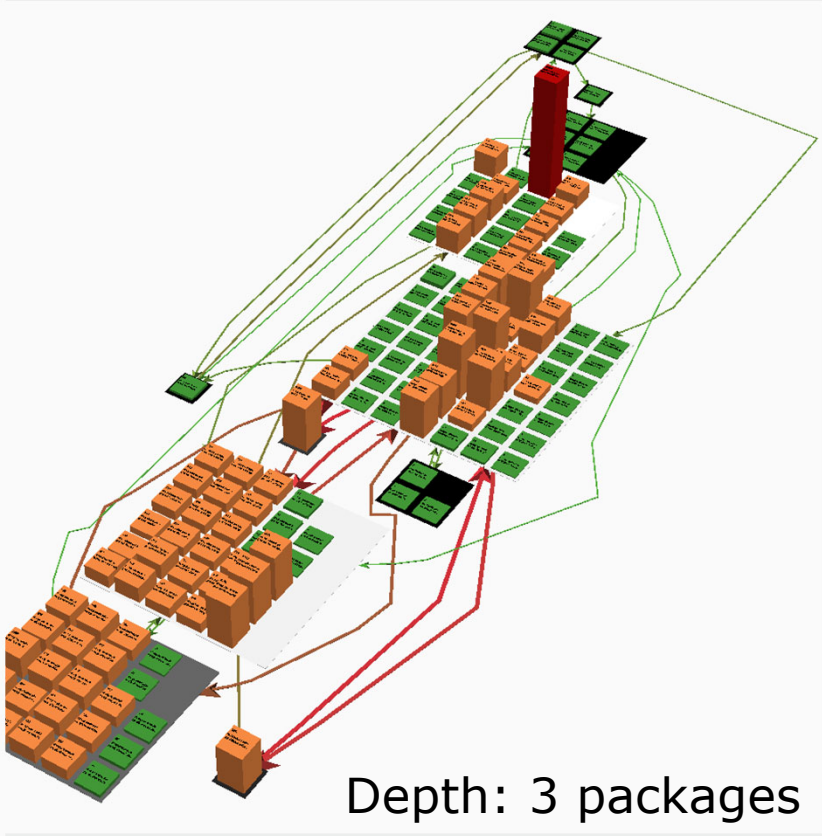
Buildings: functions

Quarter: module

Streets: dependencies between buildings

ArchitectureCity
(MSc-project Rens Rooimans , COSC)

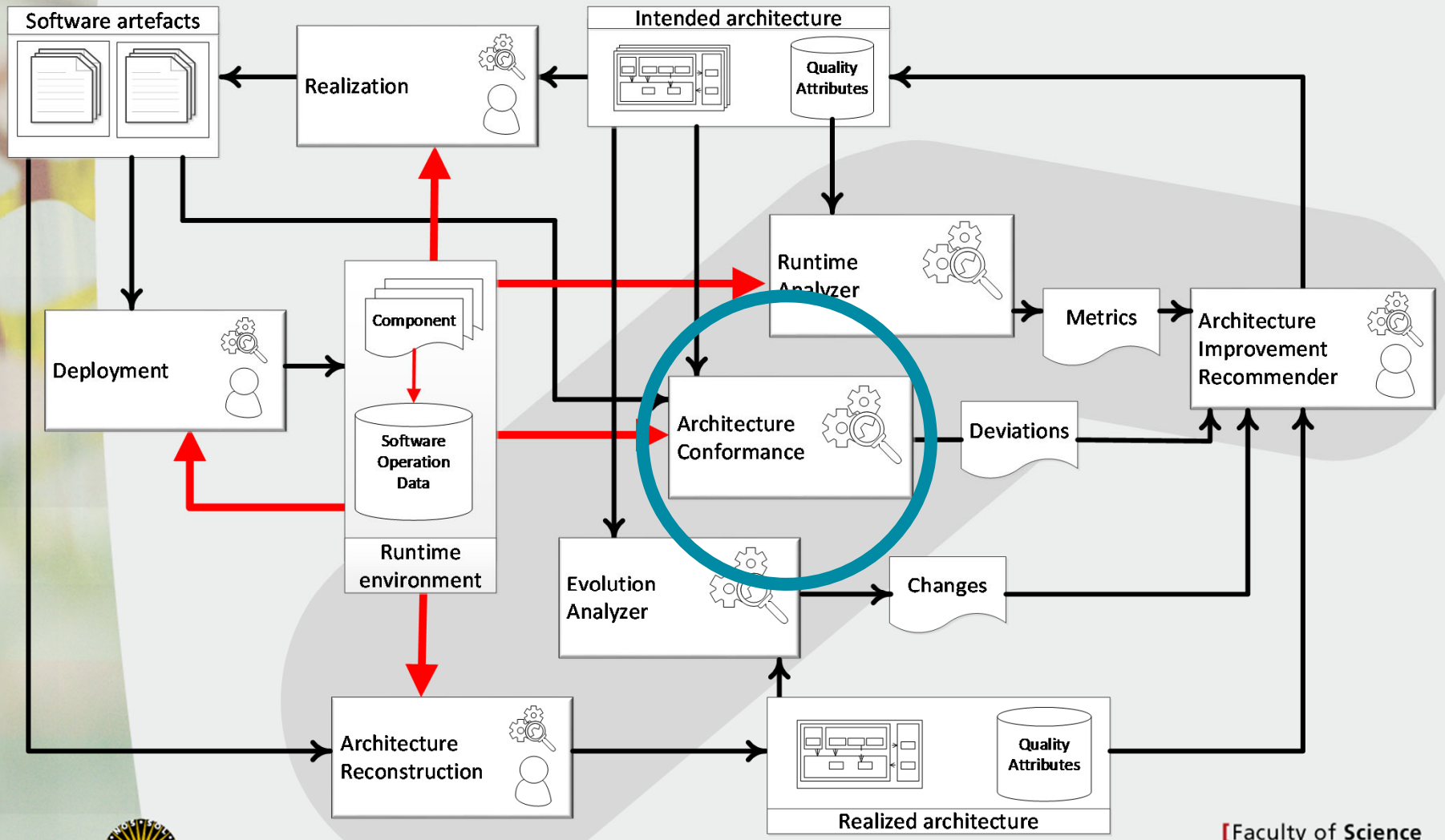




Universiteit Utrecht

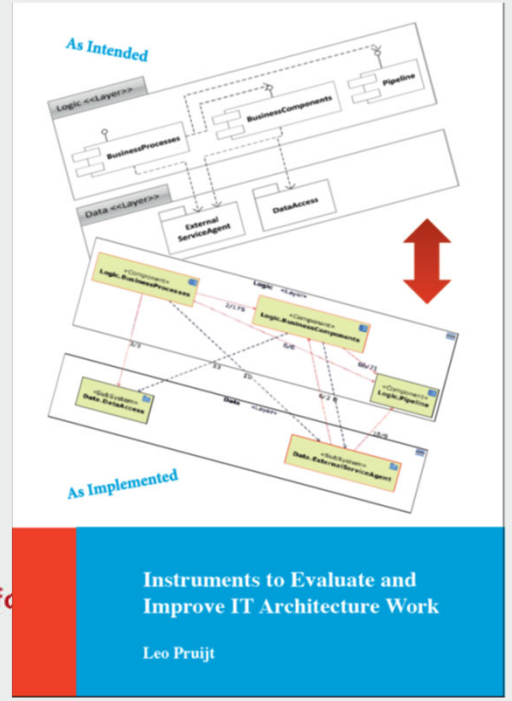
[Faculty of Science
Information and Computing Sciences]

Software Architecture: overview of the field



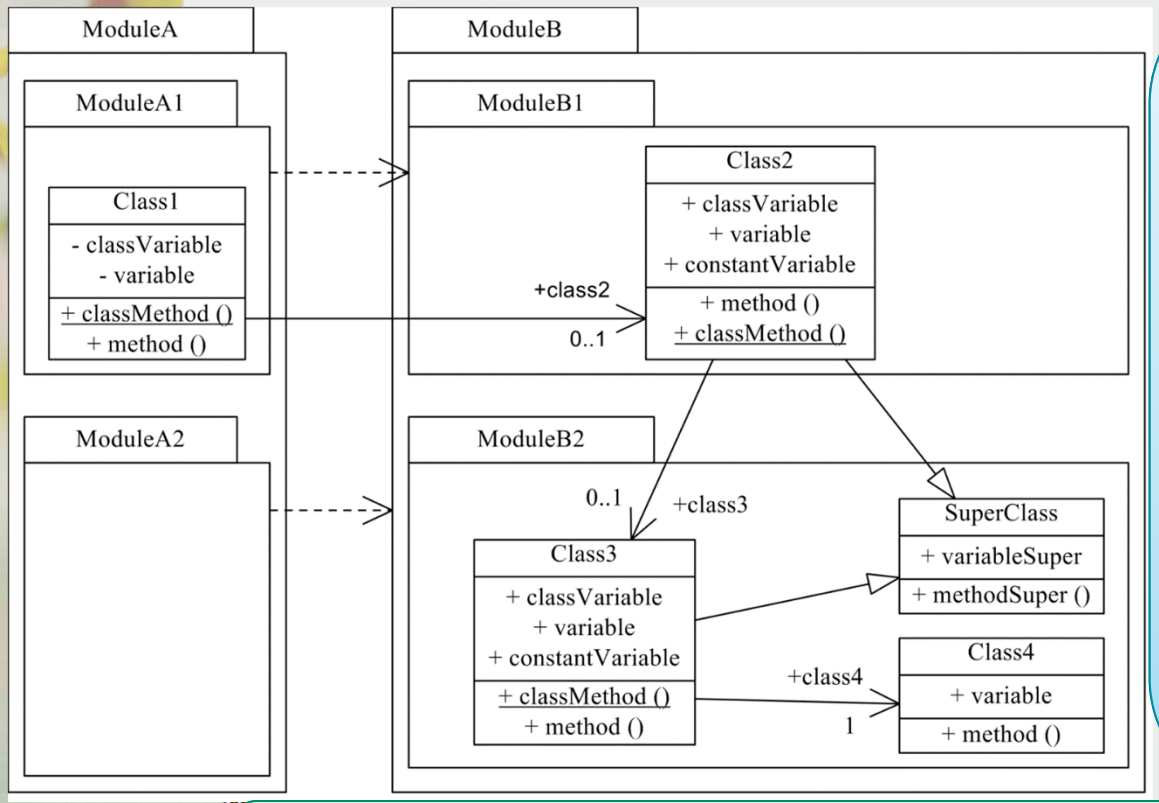
Software Architecture Conformance Checking (SACC) (PhD-topic of Dr. Leo Pruijt)

- 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".



Universiteit Utrecht

Example: Intended Architecture

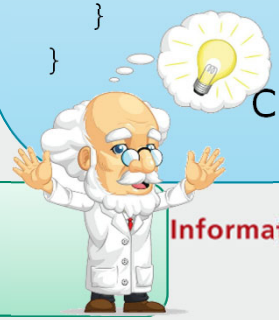



```
import
    ModuleB.ModuleB2.Class3;

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



Who spots the violations?

[Faculty of Science
Information and Computing Sciences]

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

To summarize

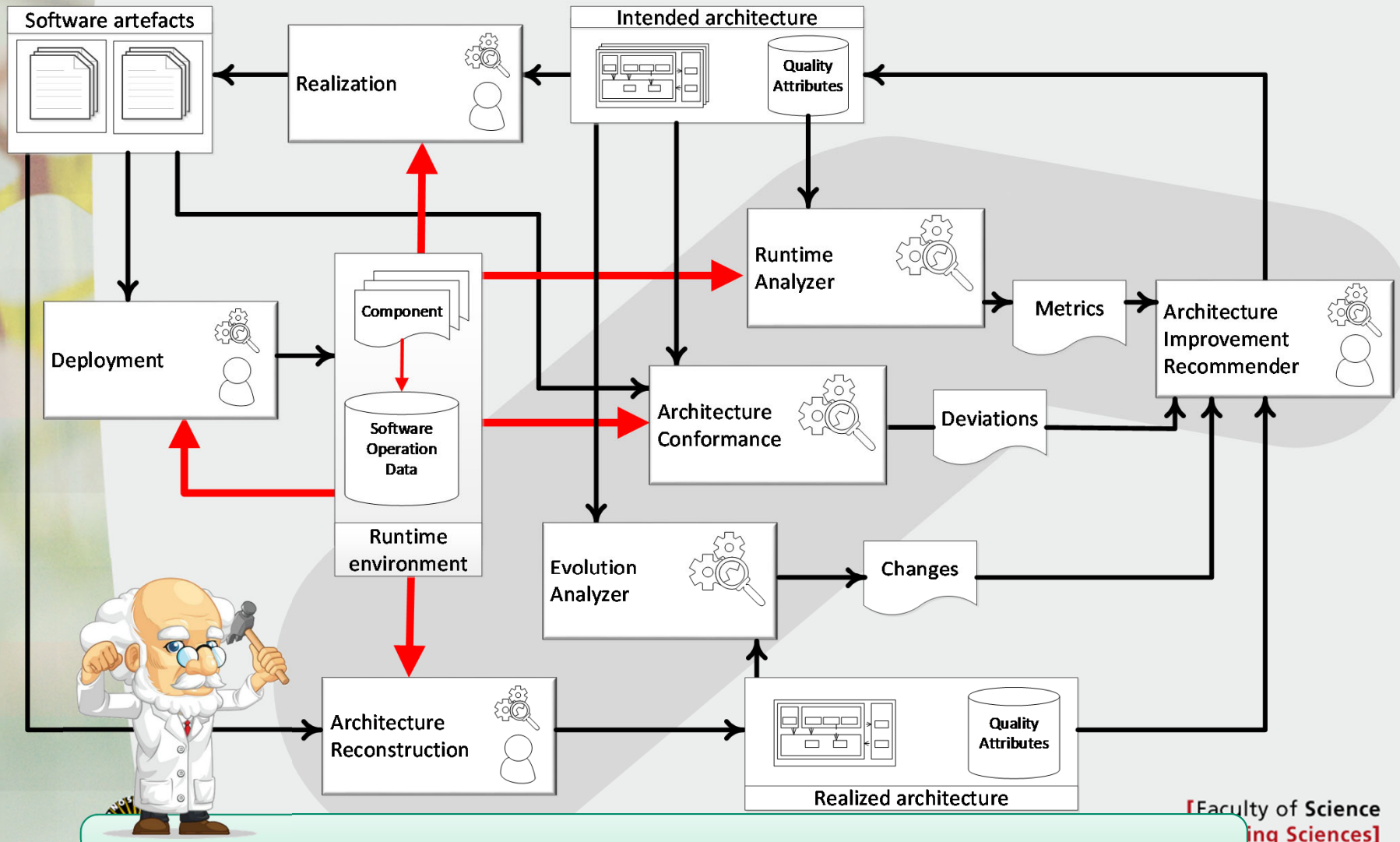
Software architecture

- 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 architectural design decisions
- Trade-off analysis of requirements
  - Functionality
  - Characteristics (“ilities”)

Document your design decisions!

Science
ciences]

Software Architecture: overview of the field



Many OZP-opportunities as well!