



Lecture 3: Architecture decisions and rationale



Agenda for today



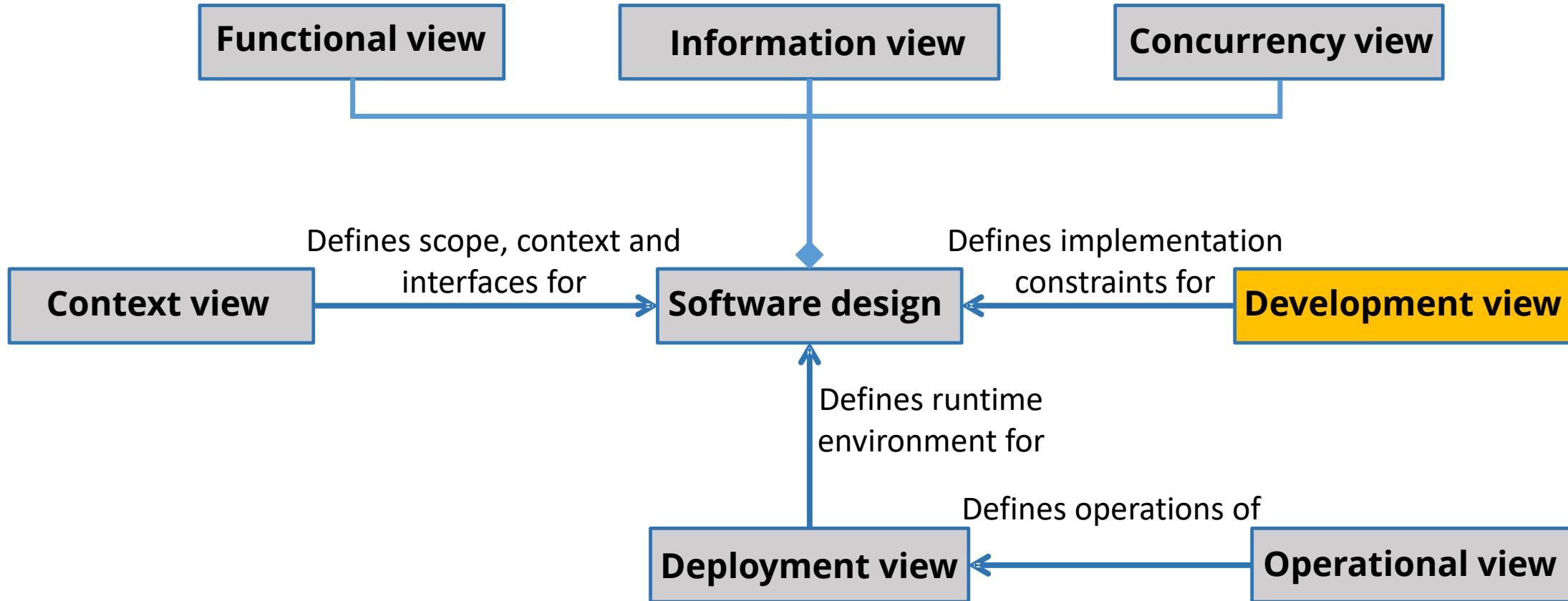
- 09:00 – 09:45: Patterns and styles
- 09:45 – 11:00: You: work on the assignment
- 11:00 – 11:30: Documenting architectures
- 11:30 – 12:40: You: work on the assignment
- 12:40 – 12:45: Wrap-up



Architectural patterns & styles



Viewpoint catalog



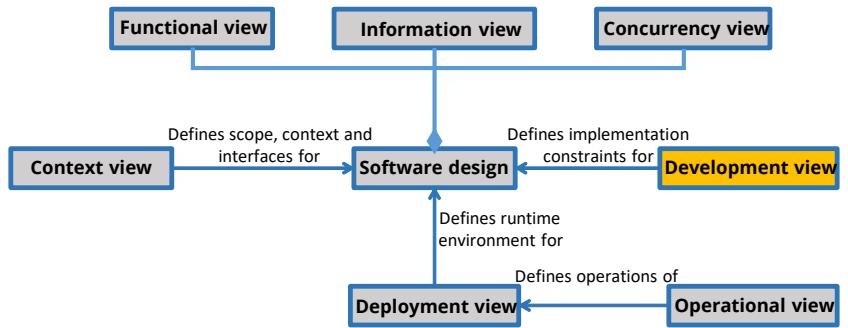
Viewpoint:

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

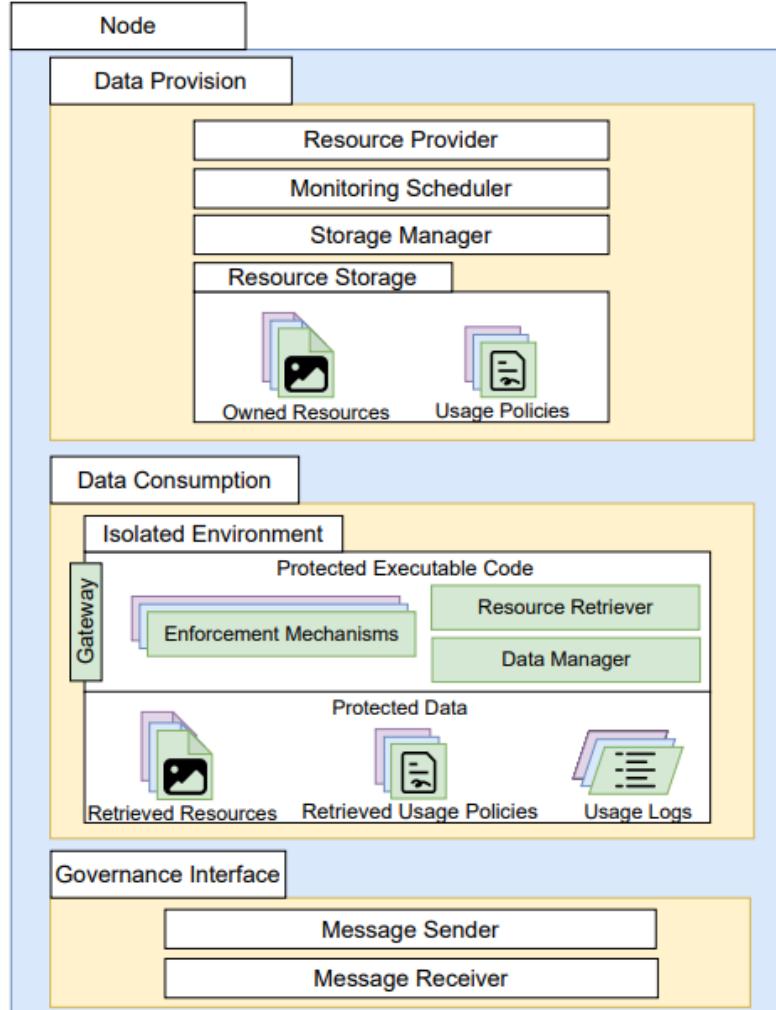


Development view

- Development view:
 - Describes the architecture that supports the **software development process**
(configuring off-the-shelf software or writing from scratch)
- Concerns
 - **Module organization** (manage dependencies!)
 - **Common processing** (e.g., logging)
 - **Standardization of design** (patterns and standards) **and testing**
 - **Instrumentation** (monitoring & debugging; mind performance!)
 - **Codeline organization** (build, test & release)
- Models and views
 - **Module structure models** (logical model)
 - **Common design models** (usually, not class diagrams)
 - **Codeline models**



```
// By using LogMF from the extras companion write, you will not incur the cost of parameter construction if debugging is disabled for logger  
LogMF.trace(logger, "\nStats:\n{0}", statsTable);
```

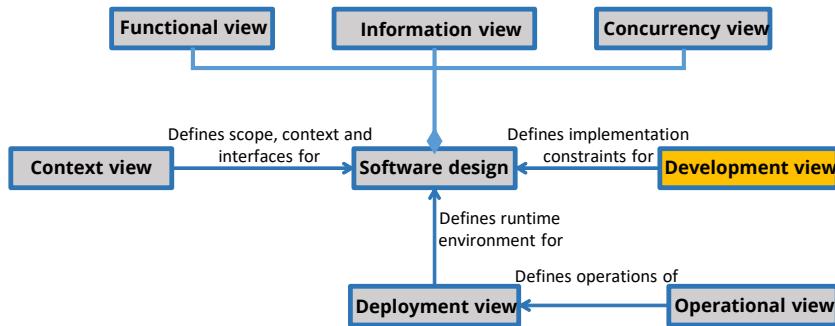


Module structure models

- Organization of the system's logical modules
How will you divide the system into **source code modules?**
- Notations
UML component diagrams
Logical model ("lines-and-boxes notation")
- Risks
 - **Too much detail**
 - **Overburdened architectural description**
In the past, we used to think that architectural models would be 100% formal!
 - **Lack of precision**
 - **Uneven focus** (specific parts are very important)
 - **Lack of developer focus** (how to transition to code?)



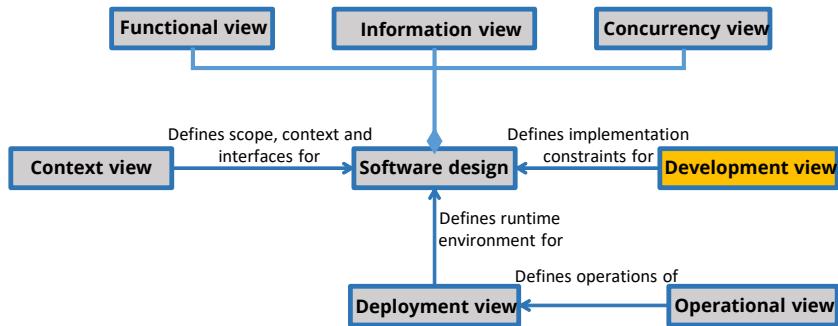
Development view



- Development view:
 - Describes the architecture that supports the **software development process**
(configuring off-the-shelf software or writing from scratch)
- Concerns
 - **Module organization** (manage dependencies!)
 - **Common processing** (e.g., logging)
 - **Standardization of design** (patterns and standards) **and testing**
 - **Instrumentation** (monitoring & debugging; mind performance!)
 - **Codeline organization** (build, test & release)
- Models and views
 - **Module structure models** (logical model)
 - **Common design models** (usually, not class diagrams)
 - **Codeline models**



Development view



- Development view:
 - Describes the architecture that supports the **software development process**
(configuring off-the-shelf software or writing from scratch)
- Concerns
 - **Module organization** (manage dependencies!)
 - **Common processing** (e.g., logging)
 - **Standardization of design** (patterns and standards) and testing
 - **Instrumentation** (monitoring & debugging; mind performance!)
 - **Codeline organization** (build, test & release)
- Models and views
 - **Module structure models** (logical model)
 - **Common design models** (usually, not class diagrams)
 - **Codeline models**



Find the commonalities

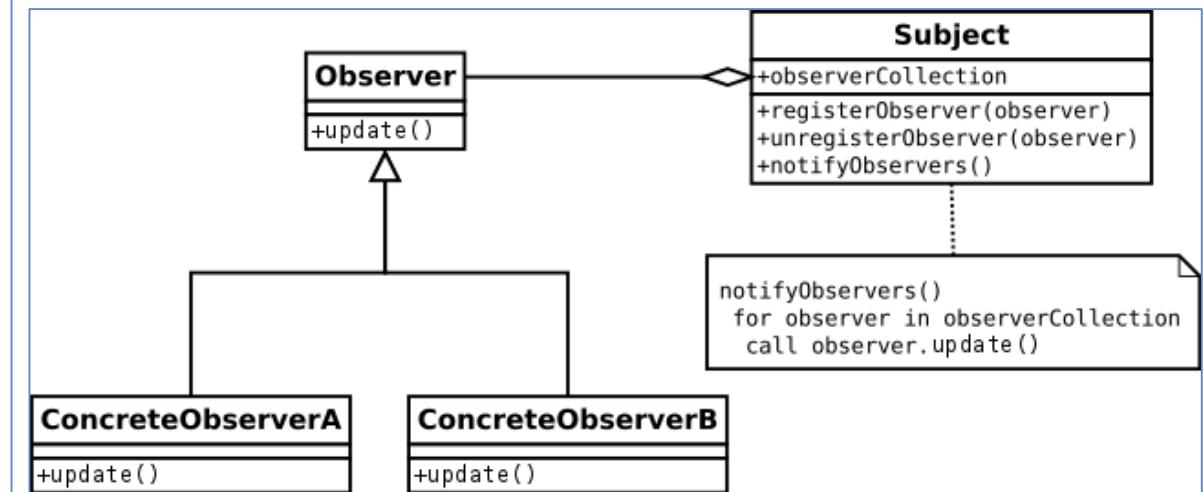
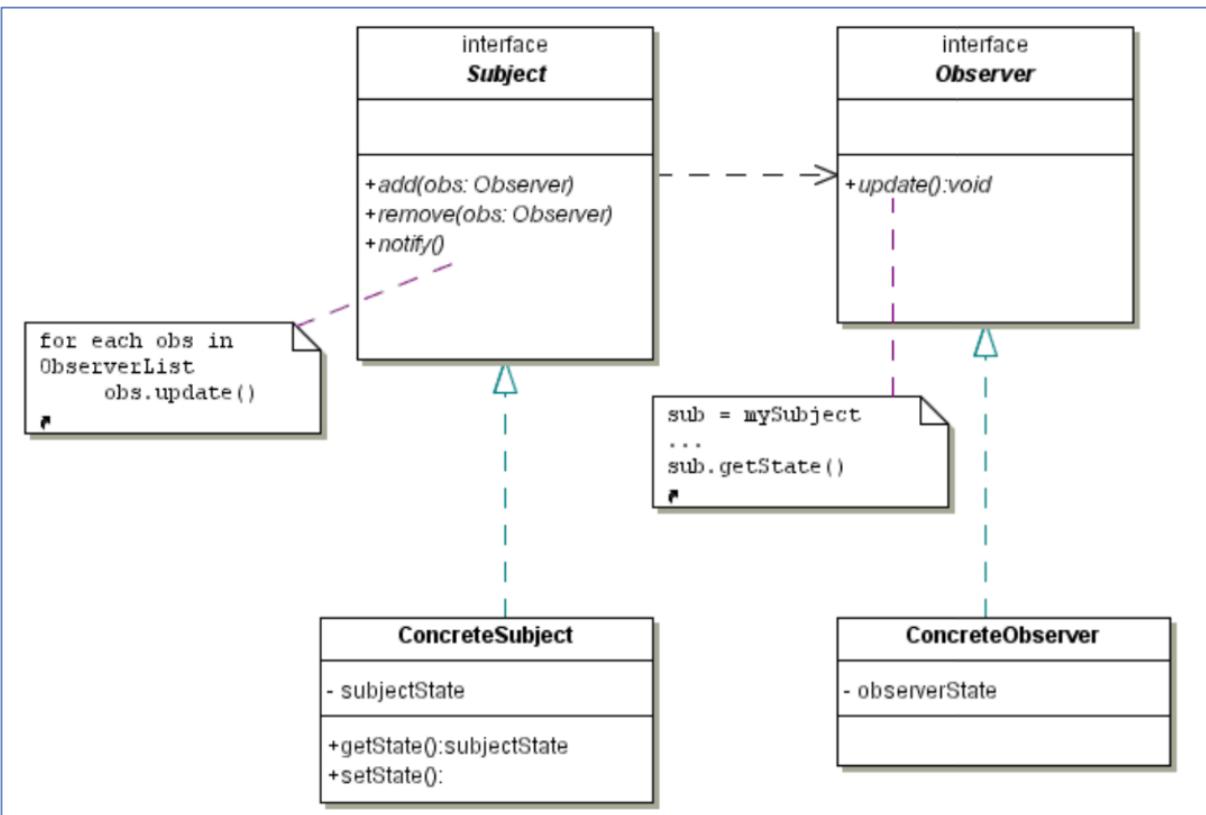
```
try:  
    month = input("Esteemed User, "  
                  "could you gently provide us with a numeric month?\n"  
                  "Month number: ")  
    print_sols_in_a_month(month)  
    correct_input = True  
except ValueError as error_message:  
    print(error_message)  
except KeyboardInterrupt:  
    print("Ouch! That was rude.")  
    correct_input = True # To exit  
except:  
    print("BYE")  
    correct_input = True # To exit
```

```
try:  
    x = int(input("Please enter a number: "))  
    break  
except ValueError:  
    print("Oops! That was no valid number. Try again...")
```

```
try:  
    # Floor Division : Gives only Fractional Part as Answer  
    result = x // y  
    print("Yeah ! Your answer is :", result)  
except ZeroDivisionError:  
    print("Sorry ! You are dividing by zero ")
```

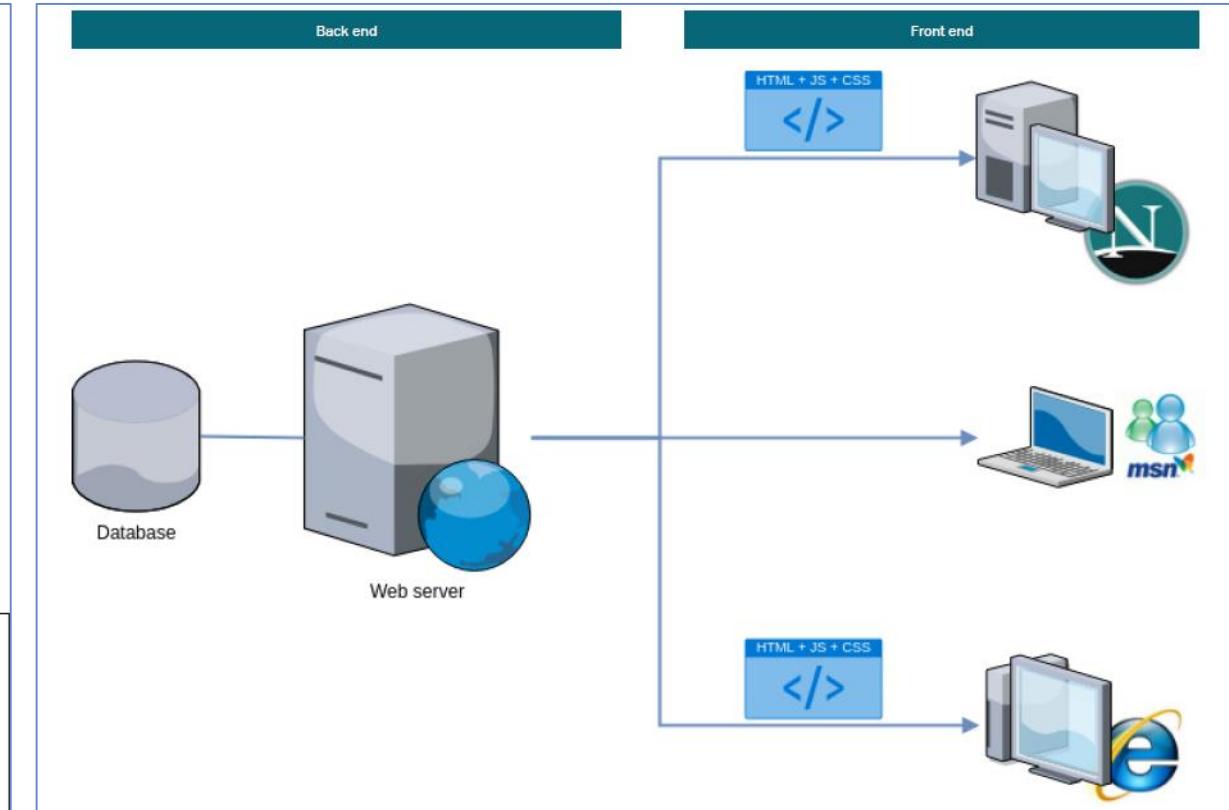
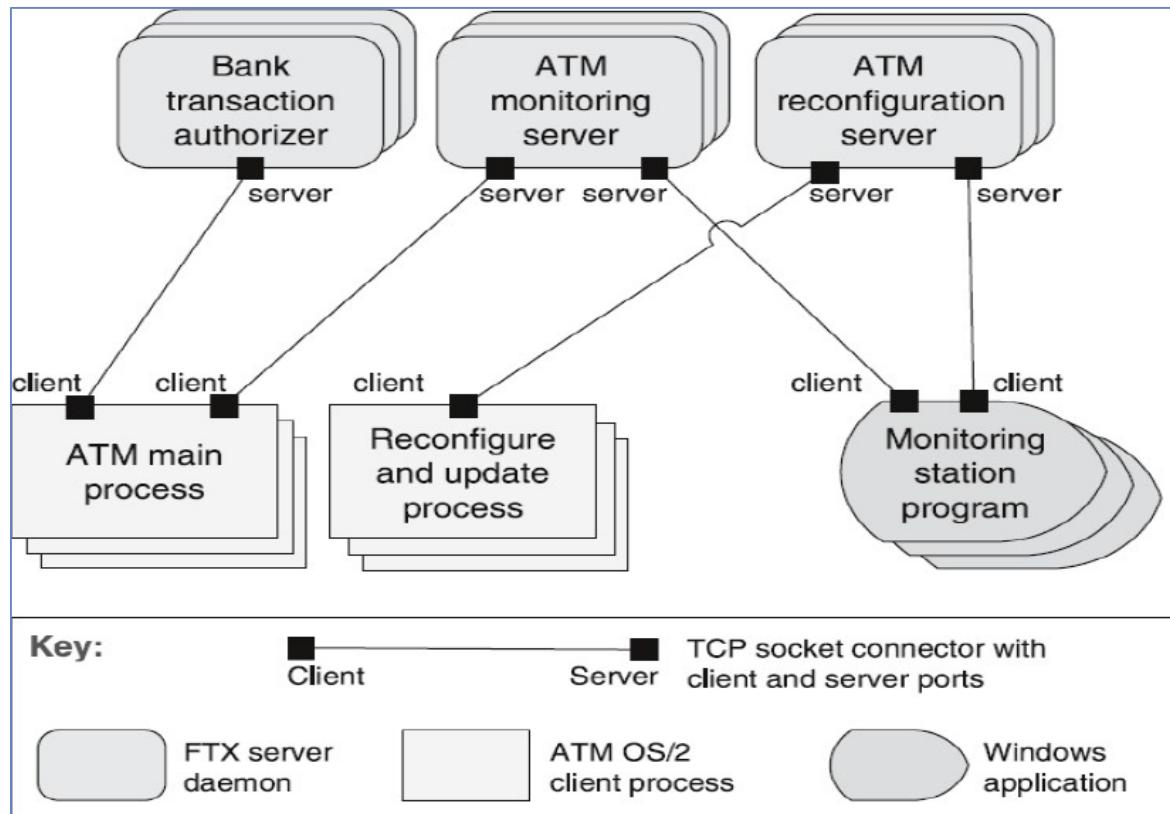


Find the commonalities



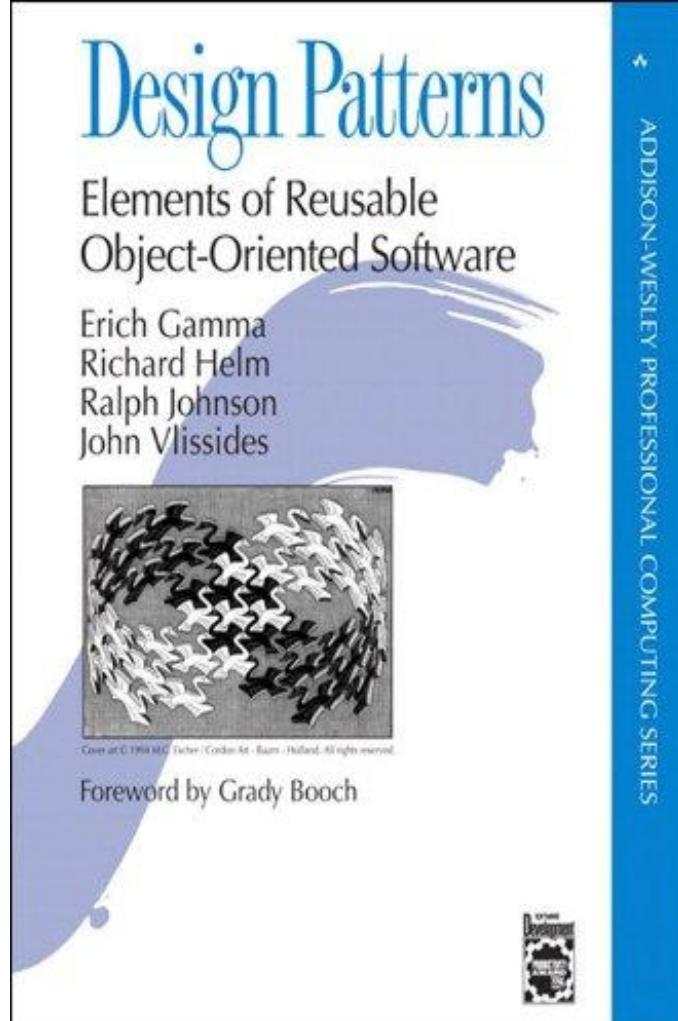


Find the commonalities





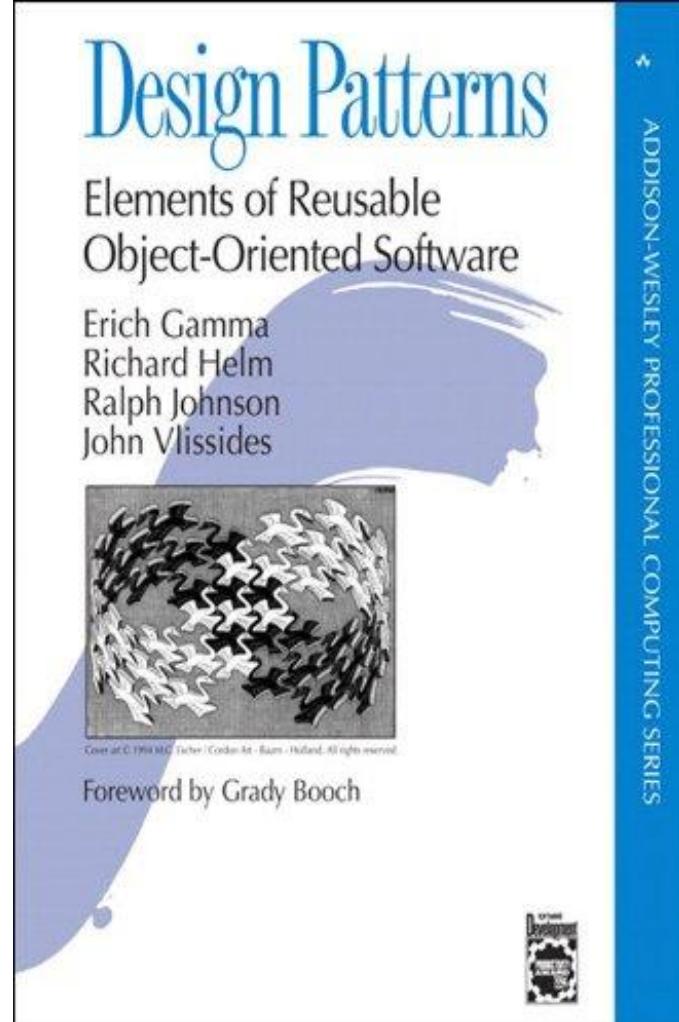
Standardization of design



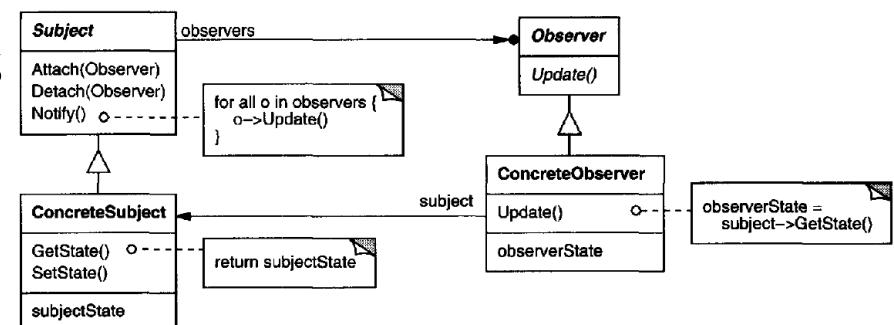
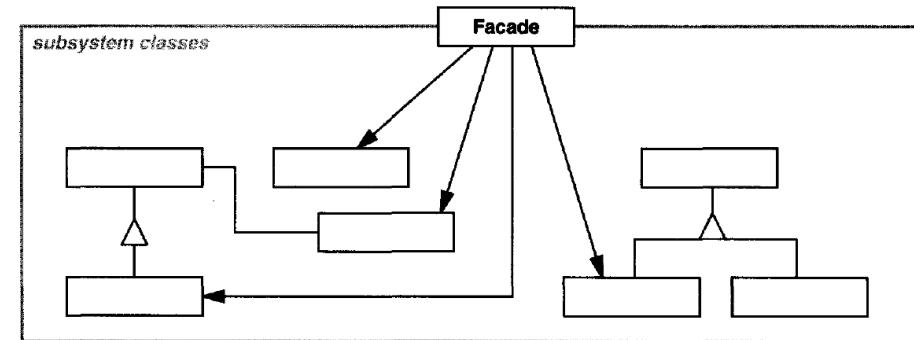
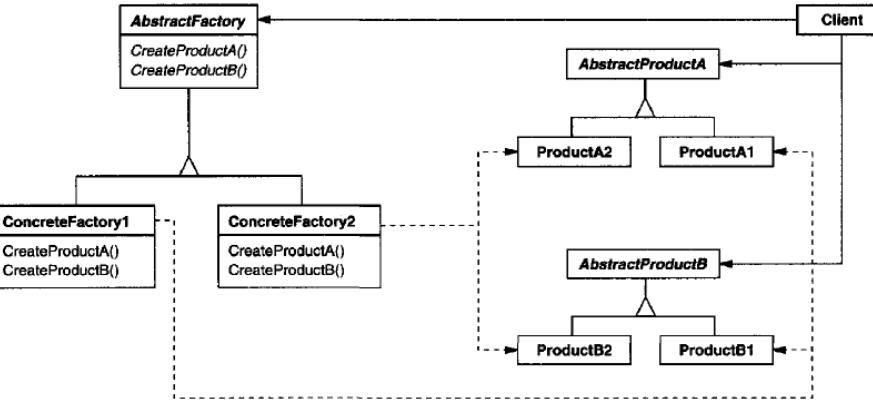
- Architectural style
Expresses a fundamental structural **organization schema** for software systems, providing a set of **predefined element types**, specifying their **responsibilities, rules and guidelines** for organizing them
- Software design pattern
A commonly recurring and proven structure of interconnected design elements that solve a **general design problem** within a particular **context**
- Language idiom
Pattern specific to a **programming language**



A few well-known software design patterns

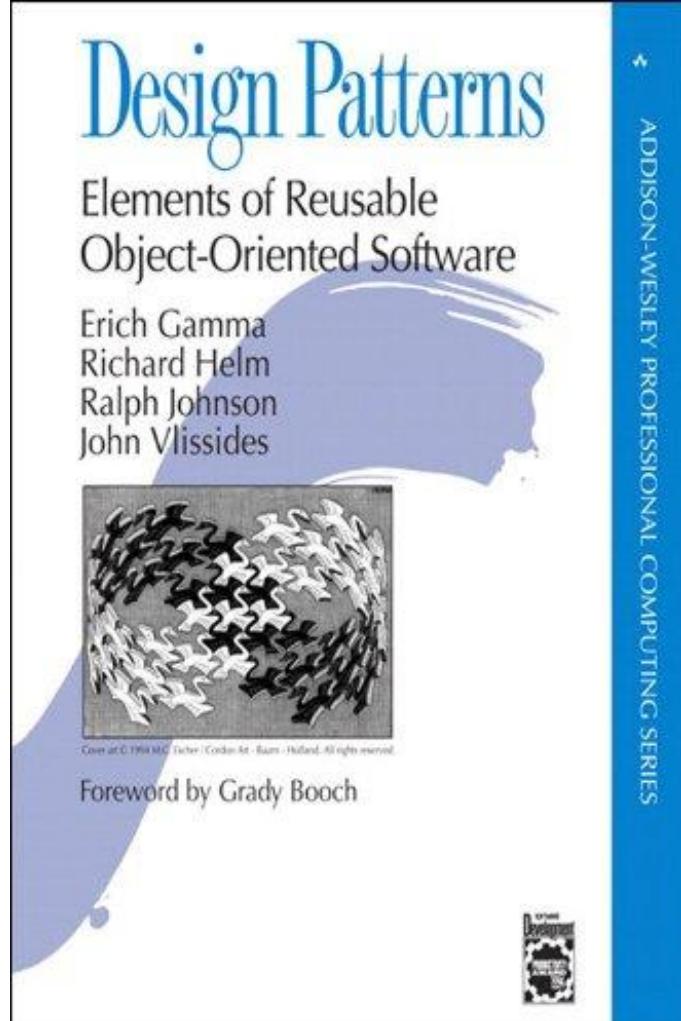


- Creational patterns
Abstract factory
- Structural patterns
Façade
- Behavioural patterns
Observer





Architectural style vs patterns



- Architectural pattern establishes a relationship:
 - **Context:** a recurring, common situation in the world that gives rise to a problem
 - **Problem:** the problem, appropriately generalized, that arises in the given context
 - **Solution:** A successful architectural resolution to the problem, appropriately abstracted
 - **Consequences:** results and trade-offs of the pattern

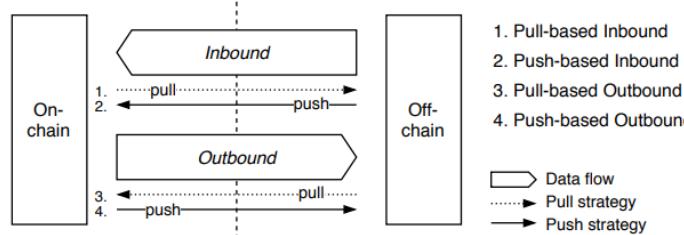


Fig. 1. Conceptual overview of the oracle data flow partitioning.

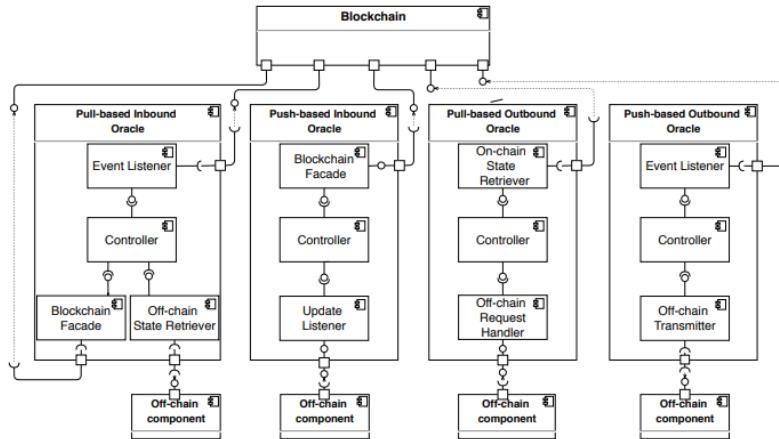


Fig. 2. An overview of the oracle types and conceptual structural components.

When a standard is missing... standardise

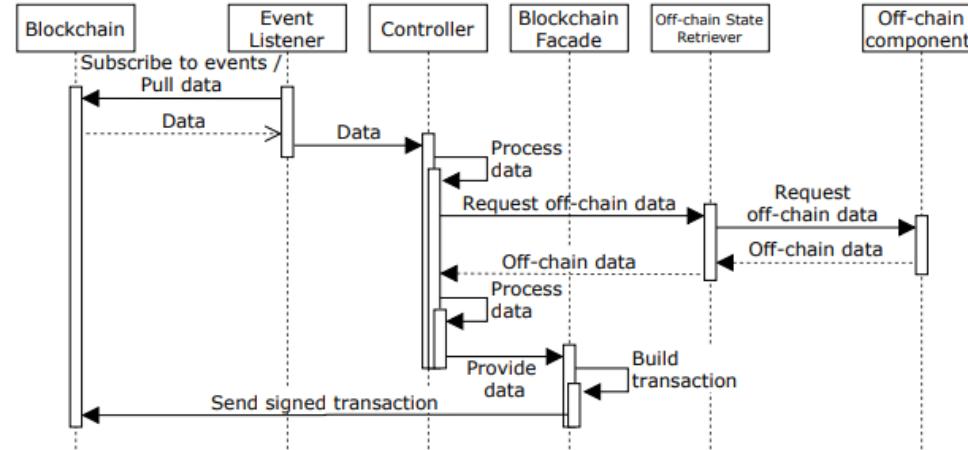


Fig. 3. Sequence diagram showing the component interactions for the *pull-based inbound oracle*.

PATTERN: Pull-based inbound oracle

Problem A blockchain application requires knowledge contained outside of the blockchain, but since blockchains are closed systems, applications cannot directly acquire information from the outside world.

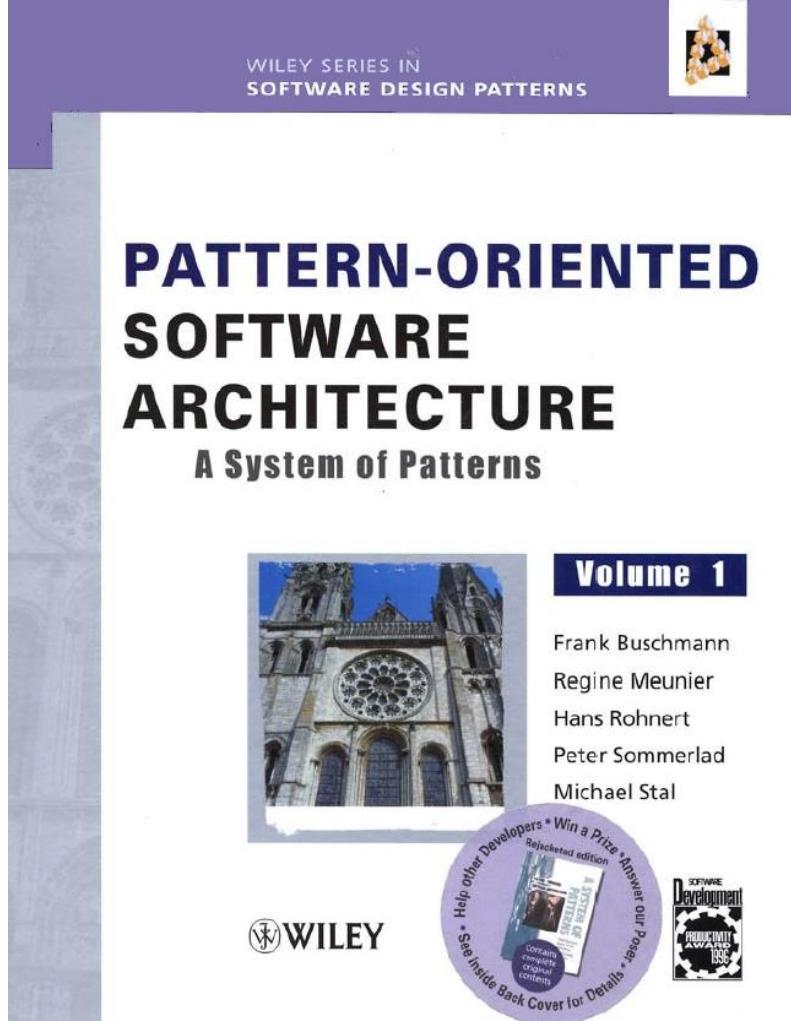
Solution A *pull-based inbound oracle* allows blockchain applications to request states from off-chain components. When a blockchain application requests an off-chain state, the *pull-based inbound oracle* receives this request, gathers the state from off-chain components, and sends the result back to the blockchain (via a transaction).

Benefits State requests are initiated in the blockchain. Thus the whole process is transparent. It can be traced whether off-chain data was successfully provided (in time) or not.

Drawbacks State requests have to be initiated from the blockchain, this induces a passive character. Further, the *pull-based inbound oracle* response time depends on the speed of the blockchain network, which may lead to a bottleneck. Network congestion may result in delayed or missed off-chain state retrieval, as the oracle only starts working after it registers requests from the blockchain.

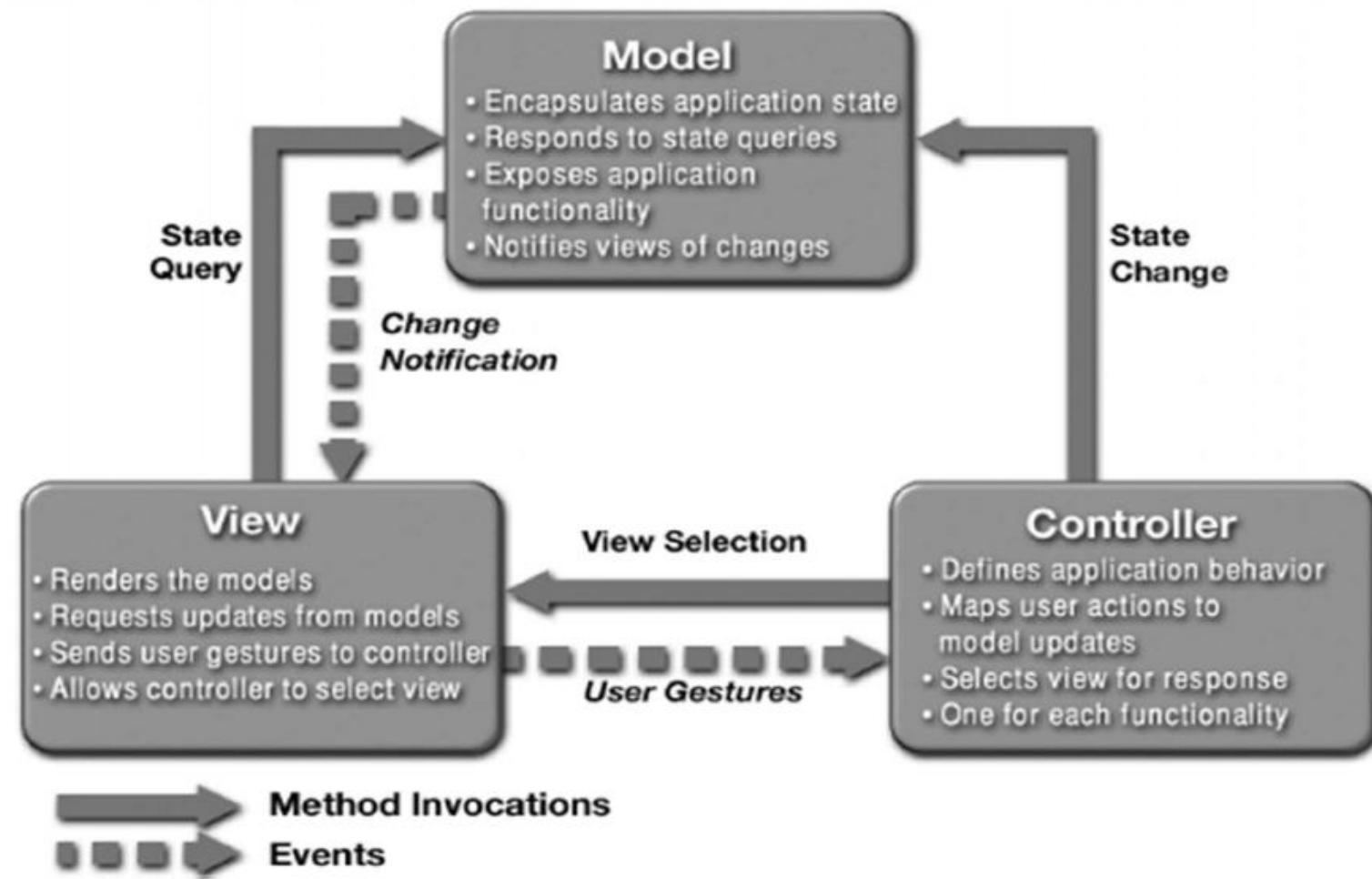


Architectural style vs patterns



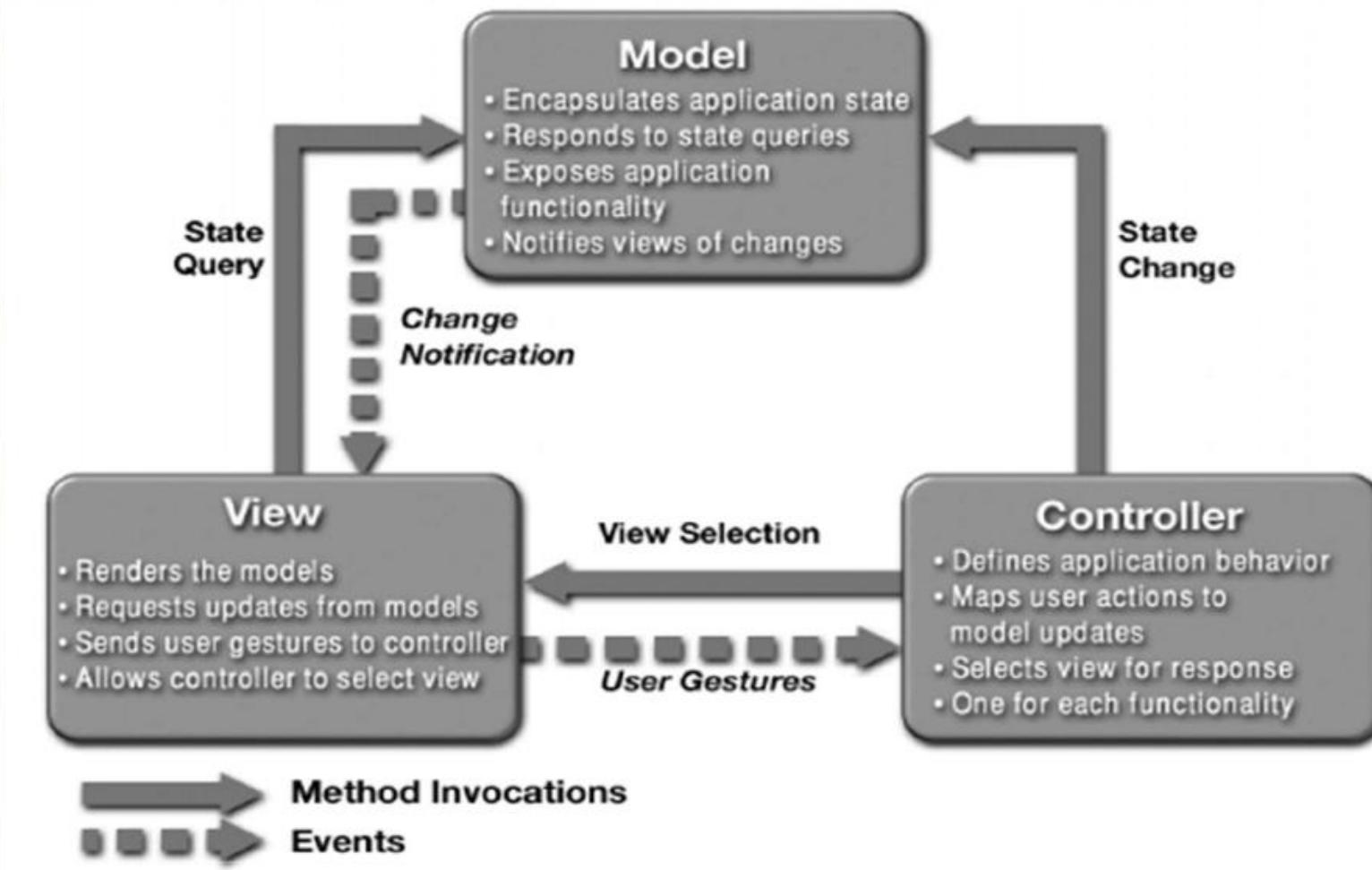
- Architectural pattern establishes a relationship:
 - Context:** a recurring, common situation in the world that gives rise to a problem
 - Problem:** the problem, appropriately generalized, that arises in the given context
 - Solution:** A successful architectural resolution to the problem, appropriately abstracted
 - Consequences:** results and trade-offs of the pattern

Model-view-controller (MVC)



- Context:
- **User interface most frequently updated part of software**
- **Data displayed from different representations**
- **Representations should all reflect current state of the data**
- Problem:
- **Keep UI functionality separate from application functionality**
- **Keep responsive to both data changes and user input**
- **Create, maintain and coordinate views of the UI consistently**

MVC vs Presentation-Abstraction-Control (PAC, J. Coutaz 1987)



Round trip ▾



1

Economy ▾

Amsterdam

Montevideo

Mon, 6 May

Sun, 12 May

All filters

Stops ▾

Airlines ▾

Bags ▾

Price ▾

Times ▾

Emissions ▾

Connecting airports ▾

Duration ▾

Track prices ⓘ 6-12 May

Any dates

Date grid

Price graph

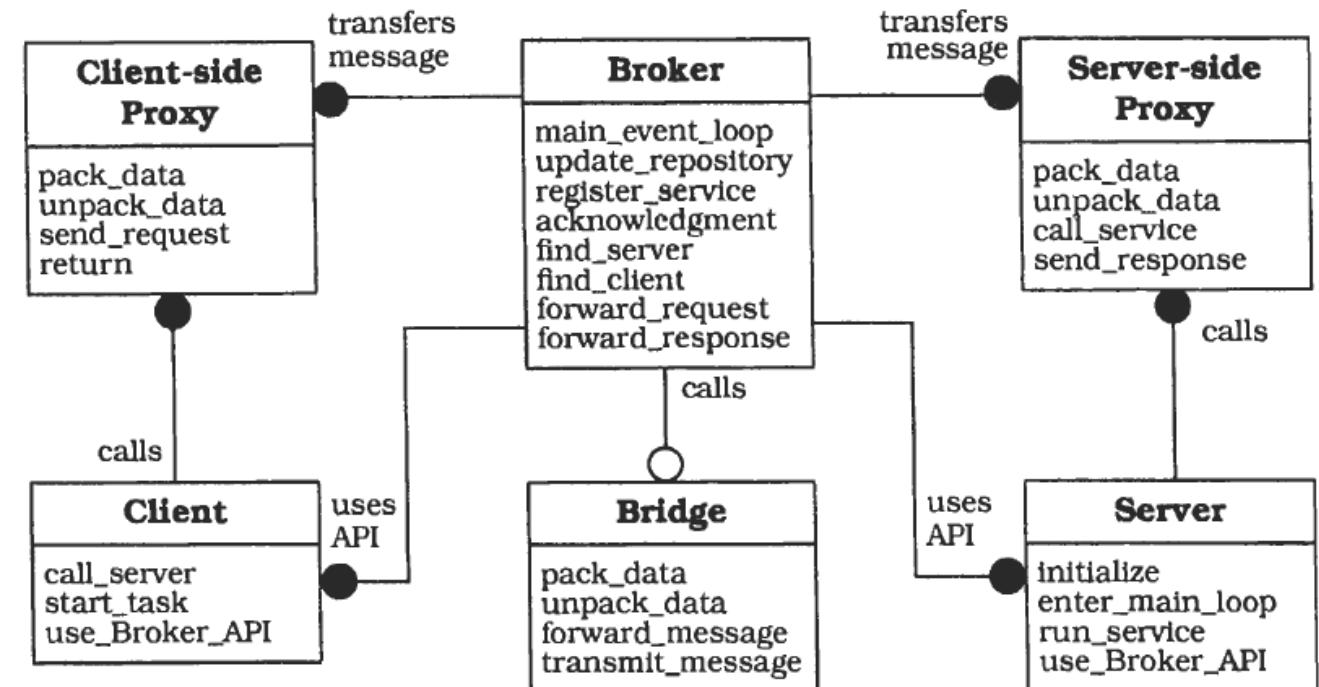
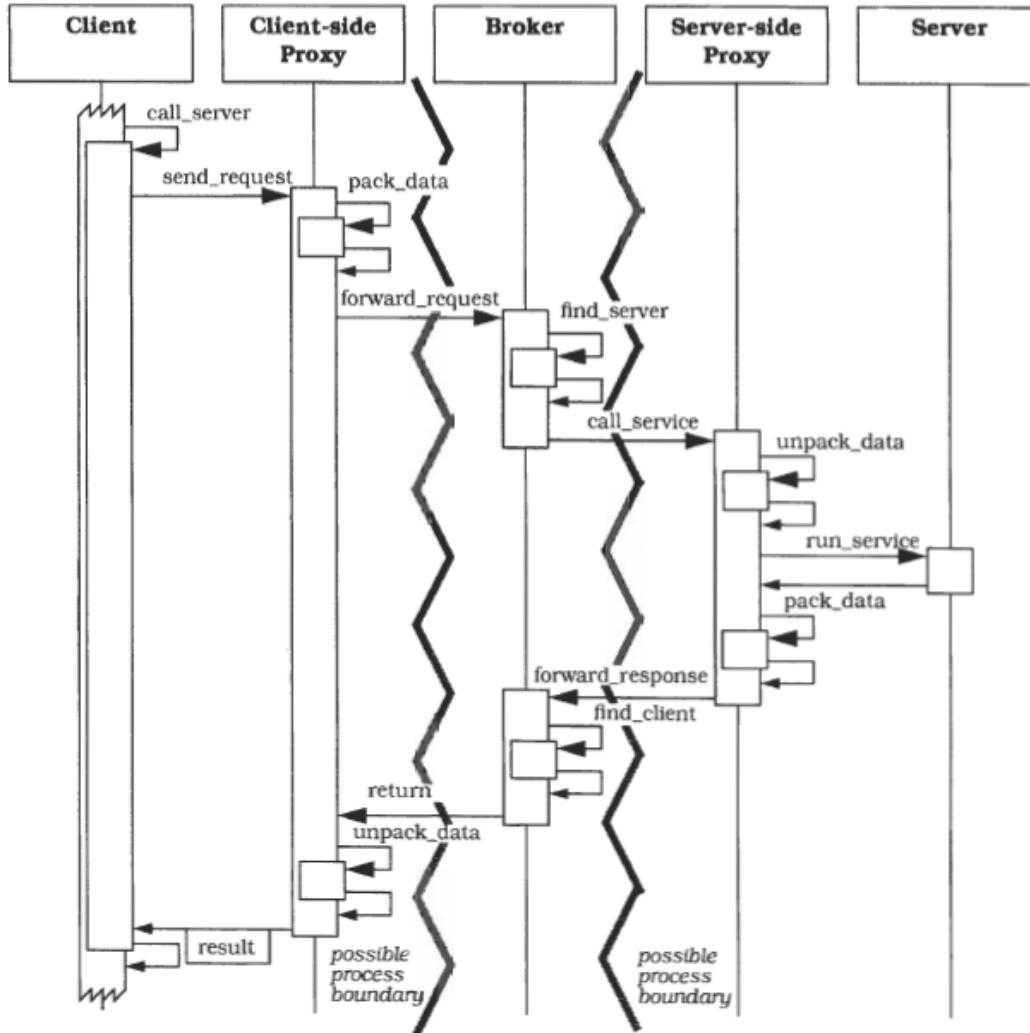
Best departing flights

Ranked based on price and convenience ⓘ Prices include required taxes + fees for 1 adult. Optional charges and [bag fees](#) may apply. [Passenger assistance info.](#) Sort by: ↑

20:35 – 11:30+ ¹ KLM, Aerolineas Argentinas	19 hrs 55 min AMS-MVD	1 stop ⚠ Change of airport	877 kg CO2e -7% emissions ⓘ	€1,091 round trip
12:55 – 00:30+ ¹ KLM, LATAM · Operated by Latam Airlines Brasil	16 hrs 35 min AMS-MVD	1 stop 1 hr 55 min GRU	998 kg CO2e +6% emissions ⓘ	€1,101 round trip
12:55 – 07:24+ ¹ KLM, COPA	23 hrs 29 min AMS-MVD	1 stop 5 hrs 21 min PTY	969 kg CO2e Avg emissions ⓘ	€1,215 round trip
19:30 – 09:50+ ¹ Lufthansa, LATAM · Operated by Latam Airlines Br...	19 hrs 20 min AMS-MVD	2 stops ⚠ FRA, GRU	909 kg CO2e Avg emissions ⓘ	€1,311 round trip
19:10 – 08:05+ ¹ Iberia · Operated by Iberia Express	17 hrs 55 min AMS-MVD	1 stop 2 hrs 20 min MAD	1,011 kg CO2e +7% emissions ⓘ	€1,418 round trip



Broker (ideal for distributed systems)



Bridges allow brokers to relay requests to other brokers

eMule v0.48a

Trennen Kad Server Transfer Suche Dateien Nachrichten IRC Statistik Optionen Tools Hilfe

Downloads (5) Alle

Dateiname	Größe	Fertigge...	Geschwi...	Fortschritt	Quellen	Priorität	Status
eMule0.48a.[content.emule-project.net].zip	2.66 MB	2.66 MB		<div style="width: 100%; background-color: green;"></div>	Auto [Ng]	Vollständig	
Firefox Setup 2.0.0.3.exe	5.73 MB	2.36 MB		<div style="width: 40%; background-color: black;"></div>	64	Auto [No]	Angehalten
KNOPPIX_V5.1.0DVD-2006-12-30-DE.iso	4.02 GB	13.01 MB		<div style="width: 3%; background-color: blue;"></div>	39/40	Auto [No]	Angehalten
ubuntu-6.10-desktop-i386.iso	698.37 MB	150.48 MB	71.65 KB/s	<div style="width: 8%; background-color: blue;"></div>	372 (14)	Auto [Ng]	Ladend
ubuntu-7.04-desktop-i386.iso	697.90 MB	65.58 MB	28.96 KB/s	<div style="width: 2%; background-color: blue;"></div>	100/101...	Auto [Ng]	Ladend

Hochladend (8)

Benutzername	Datei	Gesch...	Übertra...	Wartezeit	Upload ...	Status
SpAcEBoY	ubuntu-7.04-desktop-i386.iso	859 B/s	207.85 KB	2:42 Mins	10:32 ...	Übertrage
m0m	ubuntu-6.10-desktop-i386.iso	1.61 KB/s	165.55 KB	4:18 Mins	4:15 Mins	Übertrage
http://emule-project.net	ubuntu-7.04-desktop-i386.iso	1.31 KB/s	2.38 MB	17:13 ...	30:25 ...	Übertrage
http://emule-project.net	ubuntu-6.10-desktop-i386.iso	7.04 KB/s	6.52 MB	29:15 ...	18:01 ...	Übertrage
http://emule-project.net	ubuntu-7.04-desktop-i386.iso	282 B/s	135.46 KB	9:30 Mins	3:31 Mins	Übertrage
http://emule-project.net	ubuntu-6.10-desktop-i386.iso	8.46 KB/s	1.42 MB	10:04 ...	2:18 Mins	Übertrage
http://emule-project.net	KNOPPIX V5.1.0DVD-2006-12-30-DE.iso	8.25 KB/s	527.11 KB	10:32 ...	1:00 Mins	Übertrage

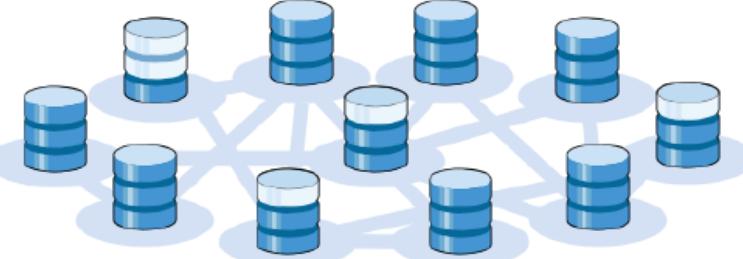
Clients auf der Warteliste: 7 (0 gebannt)

Download vc Benutzer:5.4 M(3.0 M)|Dateien:887.6 M(62| Up: 29.4 | Down: 101.6 eD2K:Verbunden|Kad:Verbunden 29.2 | 91ms | 239%



Peer-to-peer

- Context:
 - Distributed computational, equally important entities**
 - Service to a distributed community of users**
- Problem:
 - How to connect “equal” **distributed computational entities?**
- Solution:
 - Computation is achieved by **cooperating peers** that **request services from** and **provide services to** one another across a **network** (different **topologies** apply)
- Weaknesses:
 - Higher complexity** managing **security**, data **consistency**, data/service **availability, backup, and recovery.**
 - Small peer-to-peer systems may not be able to consistently achieve quality goals such as performance and availability**



[← Back to extra options](#)

How would you like to pay?

Choose a payment method:

 iDEAL Online banking 

no fee

 Credit cardA fee may apply  Card details

Card number *

Security code *

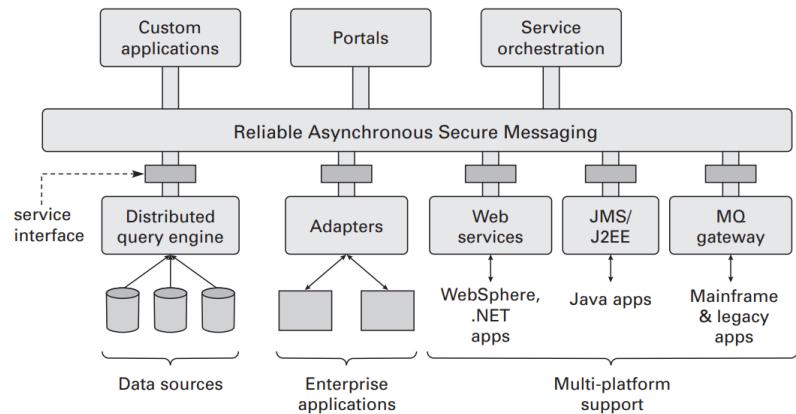
 PayPal 

EUR 1,38 fee

 Bank transfer Pay at KLM office



Service-Oriented Architecture



- Elements:

Component: Service provider (offers a service) / **consumer** (invokes a service)

Enterprise Service Bus: Route and transform messages between service providers and consumers

Registry: records the services the providers offer and their address

Orchestrator: Coordinates the interactions between service consumers and providers

- Weaknesses:

SOA-based systems are typically **complex** to build
Performance overhead with **middleware**

The screenshot shows the Microsoft Teams Chat interface. On the left is a vertical sidebar with icons for Activity, Chat (highlighted with a red notification badge), Teams, Calendar, Calls, Files, and a three-dot menu. The main area has a header with a search bar and user profile. Below the header, the title is "Daniela Mandera (You)" with tabs for Chat, Files, and Organization. A pinned message from "Contact Alex Marshall (Windows Launch Team)" is displayed at the top right. The main chat list shows messages from Daniela Mandera, Ray Tanaka, Beth Davis, Kayo Miwa, Will, Kayo, Eric, +2, August Bergman, Charlotte and Babak, Emiliano Ceballos, Oscar Krogh, Daichi Fukuda, Kian Lambert, Team Design Template, and Reviewers. A message from August Bergman is highlighted with a pink circle and says, "I haven't checked available times yet". The recent messages section follows, with messages from Charlotte and Babak, Emiliano Ceballos, Oscar Krogh, Daichi Fukuda, Kian Lambert, Team Design Template, and Reviewers. At the bottom is a message input field with placeholder "Type a new message" and various emoji and file attachment icons.

Pinned

Daniela Mandera (You)

Ray Tanaka 1:40 PM
Louisa will send the initial list of atte...

Beth Davis 1:43 PM
Thanks, that would be nice.

Kayo Miwa 4/17
I reviewed with the client on Tuesda...

Will, Kayo, Eric, +2 12:00 PM
Kayo: It would be great to sync with...

August Bergman 1:20 PM
I haven't checked available times yet

Recent

Charlotte and Babak 1:58 PM
Babak: I asked the client to send her feed...

Emiliano Ceballos 1:55 PM
😂😂

Oscar Krogh 11:02 AM
You: Thanks! Have a nice weekend

Daichi Fukuda 10:43 AM
No, I think there are other alternatives we c...

Kian Lambert Yesterday
Have you ran this by Beth? Make sure she is...

Team Design Template Yesterday
Reta: Let's set up a brainstorm session for...

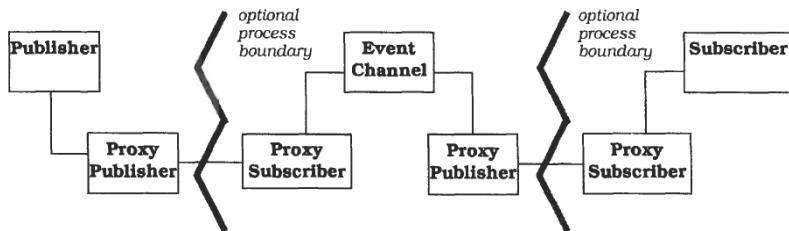
Reviewers Yesterday
Darren: Thats fine with me

Type a new message



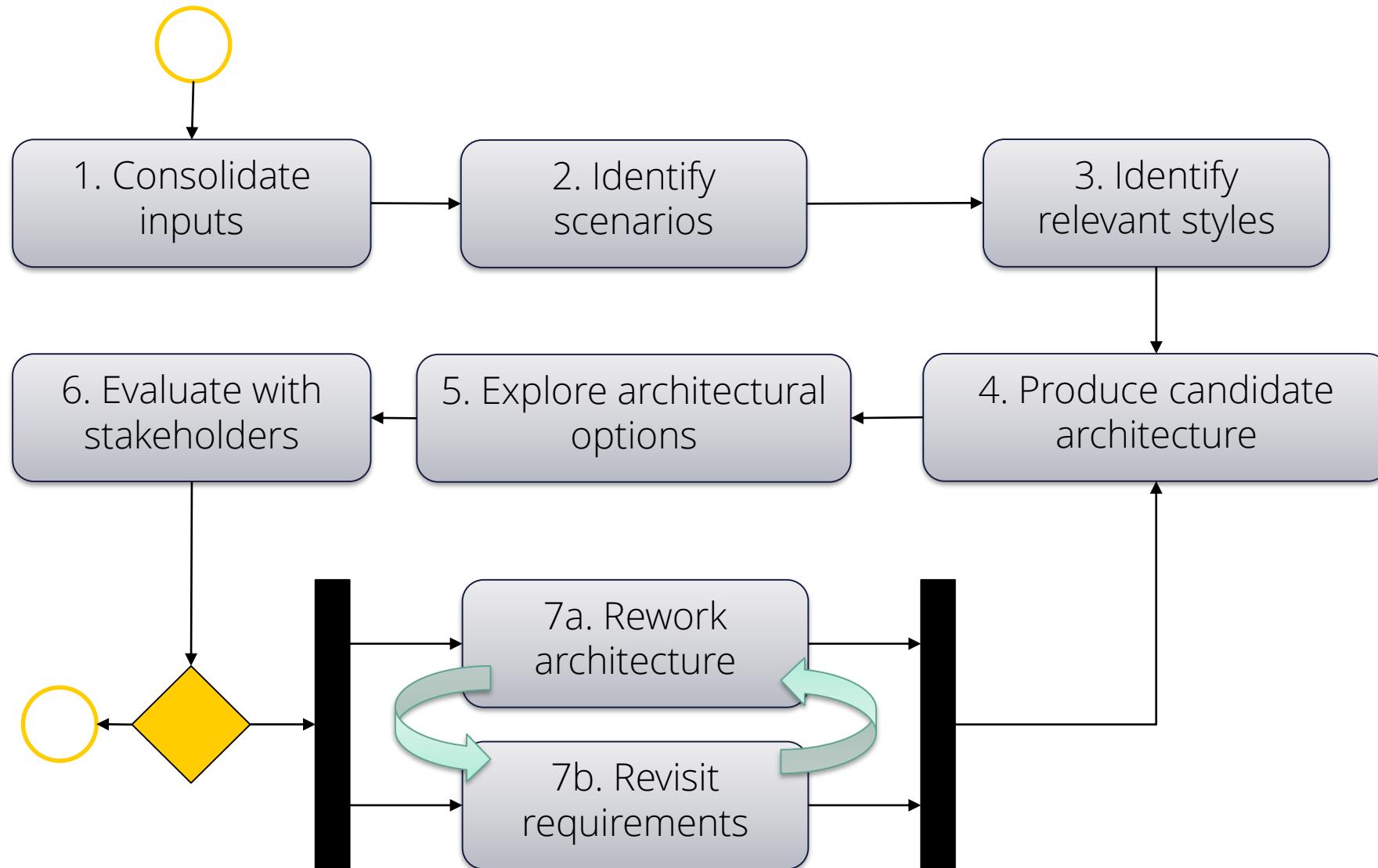
Publisher-subscriber

- Context:
Independent data producers and consumers
Nature of data not predetermined or fixed
- Solution:
Element types:
Publisher, Subscriber, Proxies, Events, Topics, Event Channel
- Weaknesses:
Increases latency
Less control over ordering of messages





Architecture definition





Example

- Shopping cart subsystem for a **very large** on-line retail store
 - Think of Amazon, Alibaba, Jingdong, ...
- Able to cope with extremely high demand
 - **Millions of checkouts in a day**
- Internal use only
 - **No special measures for security, authentication, etc.**
- Focus on writes more than reads
 - **Mostly key-value storage, queries rarely required**
- Always a small but significant number of server and network components fail at any given time.
 - **Network partitions** are a concrete possibility



“...customers should be able to view and add items to their shopping cart even if disks are failing, network routes are flapping, or data centers are being destroyed by tornados.” Anonymous stakeholder*



*Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swami Sivasubramanian, Peter Vosshall and Werner Vogels, “Dynamo: Amazon’s Highly Available Key-Value Store”, in the Proceedings of the 21st ACM Symposium on Operating Systems Principles, Stevenson, WA, October 2007..



Which architectural styles can you use in order to satisfy these requirements?

Let's call this system

abazon

The Amazon logo, which consists of the word "amazon" in lowercase black letters with a red curved arrow underneath forming a smile.



Utrecht University



A possible solution

- Peer to peer architecture (for fault tolerance)
Data is replicated across multiple peers
Distributed Hash Table (DHT)

How are peers managed?



A possible solution

- Peer to peer architecture (for fault tolerance)
Data is replicated across multiple peers
Distributed Hash Table (DHT)
- Service-based architecture (for peer management)
The registry must be made transparently fault-tolerant

Multiple responsibilities for each peer. How to organize them?



A possible solution

- Peer to peer architecture (for fault tolerance)
Data is replicated across multiple peers
Distributed Hash Table (DHT)
- Service-based architecture (for peer management)
The registry must be made transparently fault-tolerant
- For data storage and transfer purposes, peers are layered.
Replication and versioning abstraction layer
Routing of data elements through peers (DHT)

How do **registries** communicate?



A possible solution

- Peer to peer architecture (for fault tolerance)
Data is replicated across multiple peers
Distributed Hash Table (DHT)
- Service-based architecture (for peer management)
The registry must be made transparently fault-tolerant
- For data storage and transfer purposes, peers are layered.
Replication and versioning abstraction layer
Routing of data elements through peers (DHT)
- Registries adhere to a publisher-subscriber style
They register for membership changes and are notified about them
Membership management and failure detection are gossip-based
Each peer is responsible for both data storage and peer management

Styles are just the **first step!**



Agenda for today



- 09:00 – 09:45: Patterns and styles
- 09:45 – 11:00: You: work on the assignment
- 11:00 – 11:30: Documenting architectures
- 11:30 – 12:40: You: work on the assignment
- 12:40 – 12:45: Wrap-up



Architecture documentation



Architecting: find the differences

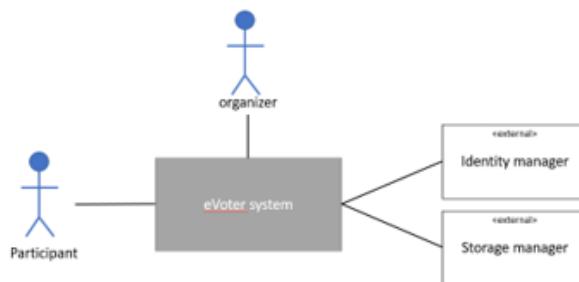




Traditional documentation of views

VIEW: EXTERNAL ENTITIES

MODEL



The system depends on two external systems, one to validate the identity of the voter and one to store the results of the referendum. There are two actors involved in the system: the participant and the organizer. The participant has a management interface to the system, while the organizer can cast its vote.

GLOSSARY OF ELEMENTS

Id	Name	Description
C1	Participant	The actor participant
C2	Organizer	The actor organizer
C3	Identity manager	System that checks for the identity of the voter
C4	Storage manager	System that maintains a secure storage of the referendum results

ANALYSIS ON PERSPECTIVES

Availability.

Requirements on identity manager:

- A model without description is no model!
Description should be concise & precise
Describe the *intent* of the view
- Define the elements in the model
- Analysis on the perspectives, if applicable
- If the view cannot be linked to a stakeholder?
The view is unnecessary!
- If a stakeholder has no views?
Is the stakeholder really involved?



Architecture documentation



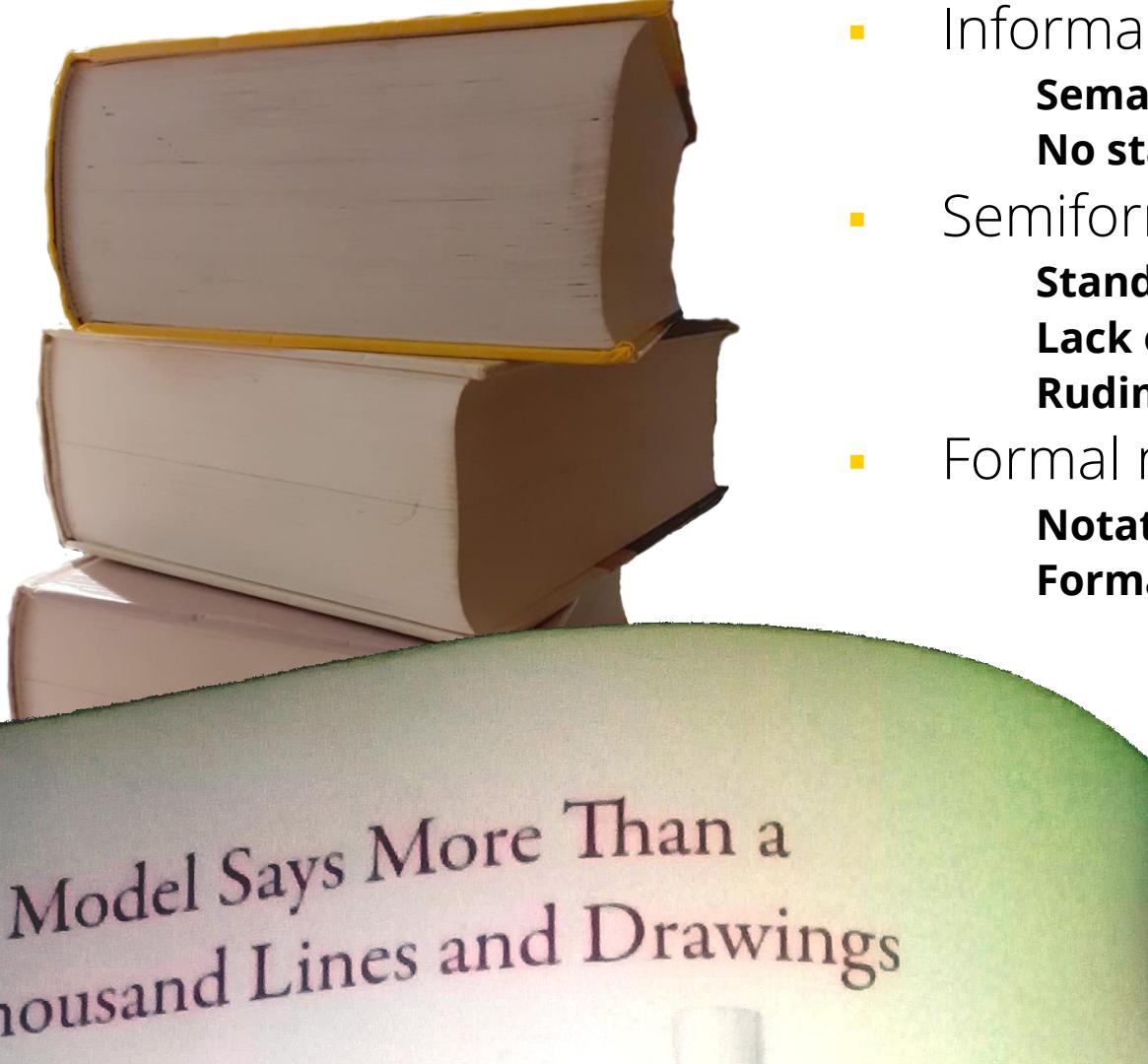
- Even the best architecture is **useless** if the people who need it:
Do not know what it is
Cannot understand it well enough to use, build or modify it
Misunderstand it and apply it incorrectly

- Architecture must be:
sufficiently transparent and accessible
sufficiently concrete to serve as a blueprint
sufficiently complete to serve as a basis for analysis



Notations in architecture documentation

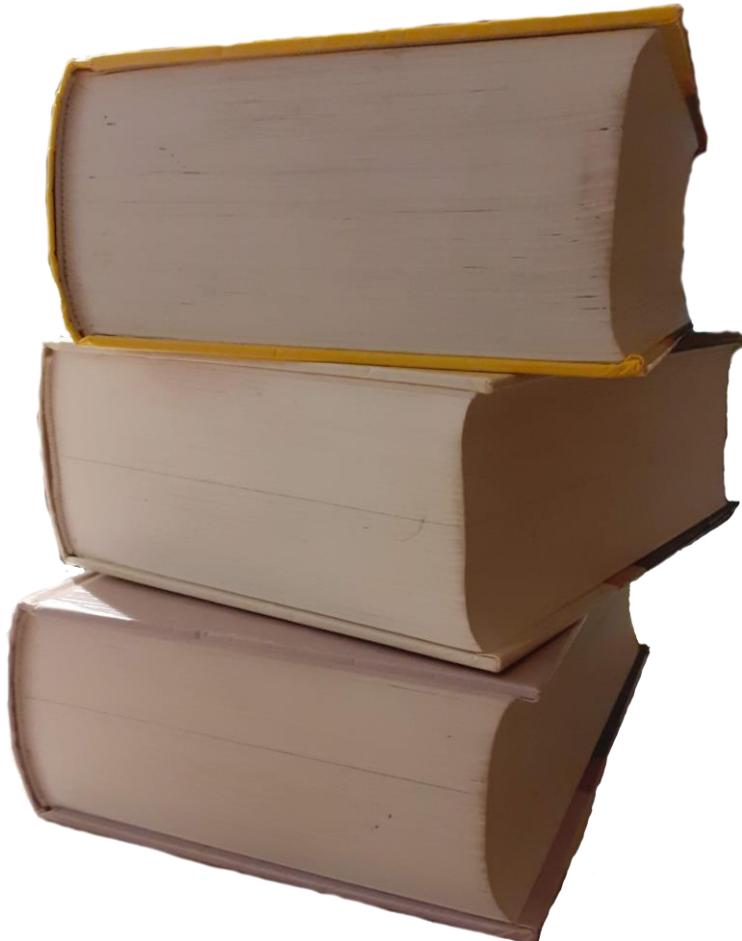
- Informal notation
 - Semantics in natural language**
 - No standardized notation**
- Semiformal notation
 - Standardized notation prescribing graphical elements and rules**
 - Lack of complete formal semantics**
 - Rudimentary analysis based on notation**
- Formal notation
 - Notation has precise semantics (mathematical!)**
 - Formal analysis of both syntax and semantics is possible**



A Model Says More Than a
Thousand Lines and Drawings



Which notation do you choose?



- Trade-off effort vs. effect
- Formal notations:
Very high effort to build the models!
Offer reduced ambiguity
Allows for analysis
- Informal notations:
Easy to create
Fewer guarantees
- Depends on the **intent** of the model!

Elephant in the room:
Architecture is also about the why of the system!!!



Decision making in Software Architecture

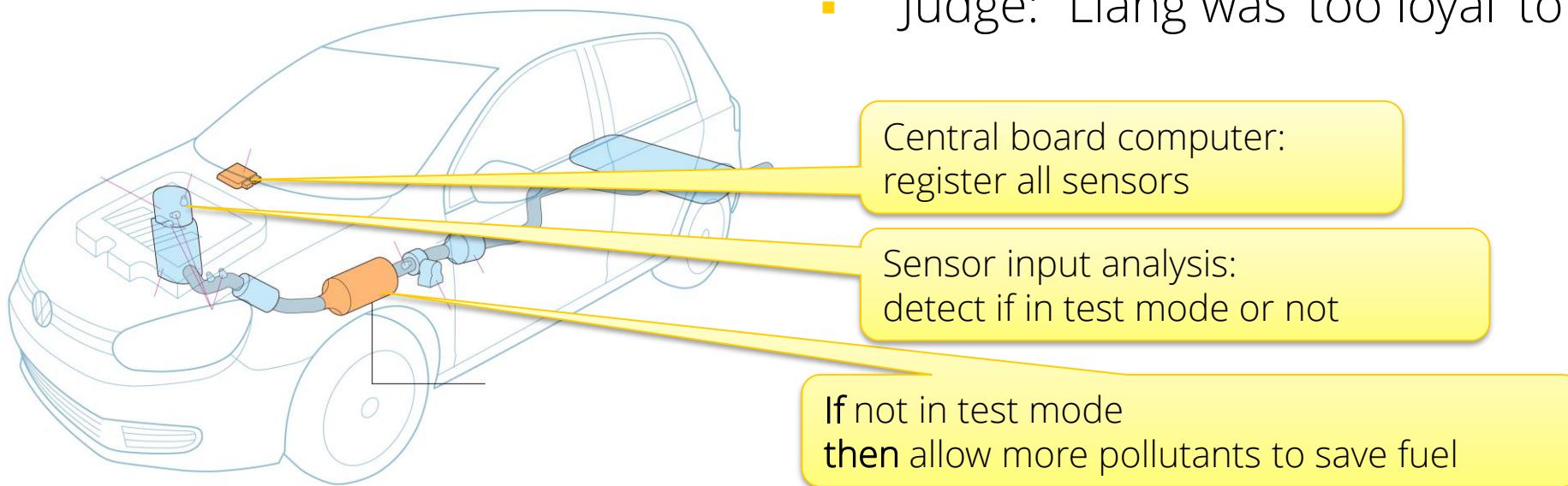


Who knows this person?



First person in prison because of a software architecture decision!

- Sentenced 40 months in prison for Diesel Gate
Engineer
Developed software that concealed high levels of pollutants
Received a harsher sentence than recommended
- Judge: "Liang was 'too loyal' to the German automaker"





(Jansen & Bosch, 2005): What did you notice?

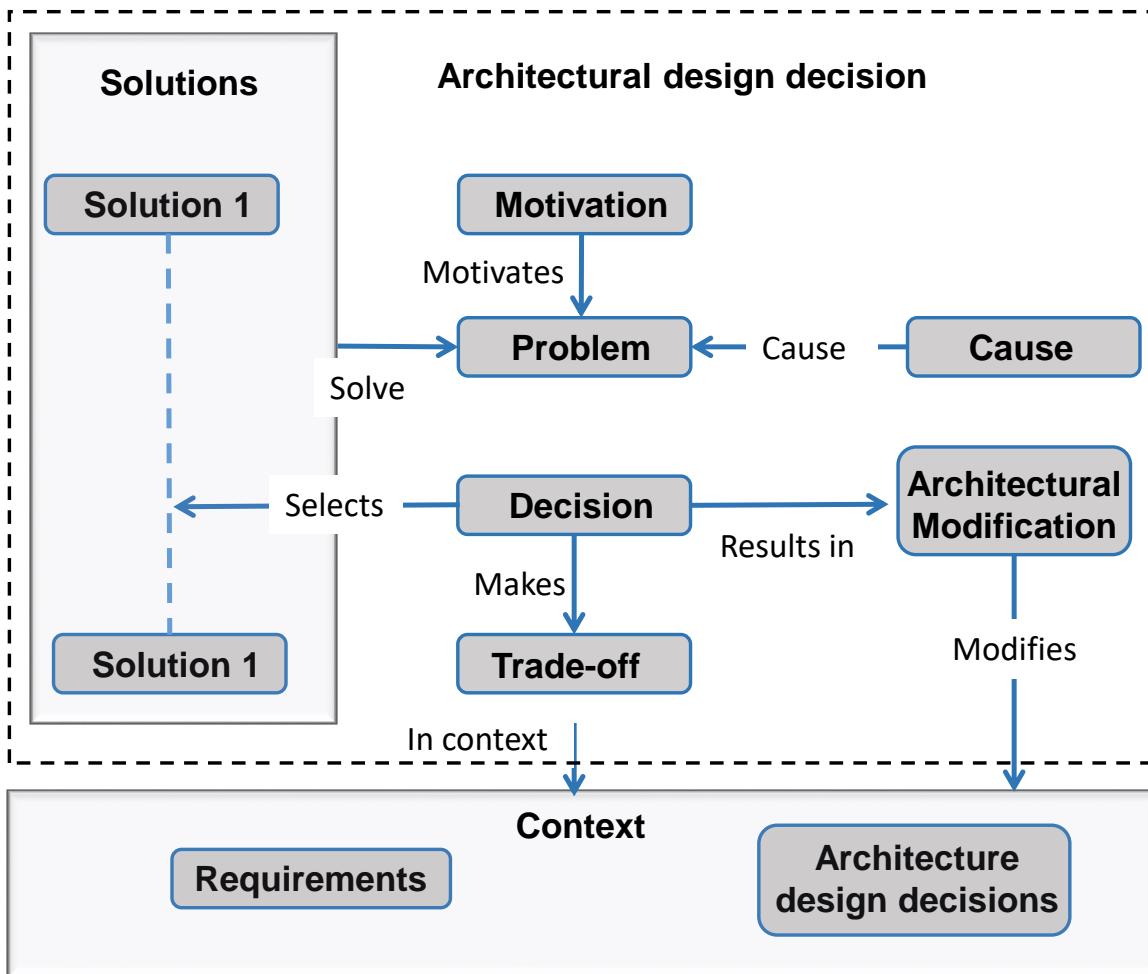


Jansen & Bosch: what's their vision?

- Why is the architecture as it is?
 - Intertwined/fragmented design decisions**
 - Evolution: rules and constraints are violated, architecture drift**
 - Obsolete decisions are never removed**
- Architectural design decisions as first class elements?
 - Guarding conceptual integrity**
 - Explicit design space exploration**
 - Traceability!**
- Requirements
 - Explicit architectural changes**
 - Corrective, perfective and adaptive support**
 - Clear relationship between architecture and system realization**



Jansen & Bosch: what's their vision?



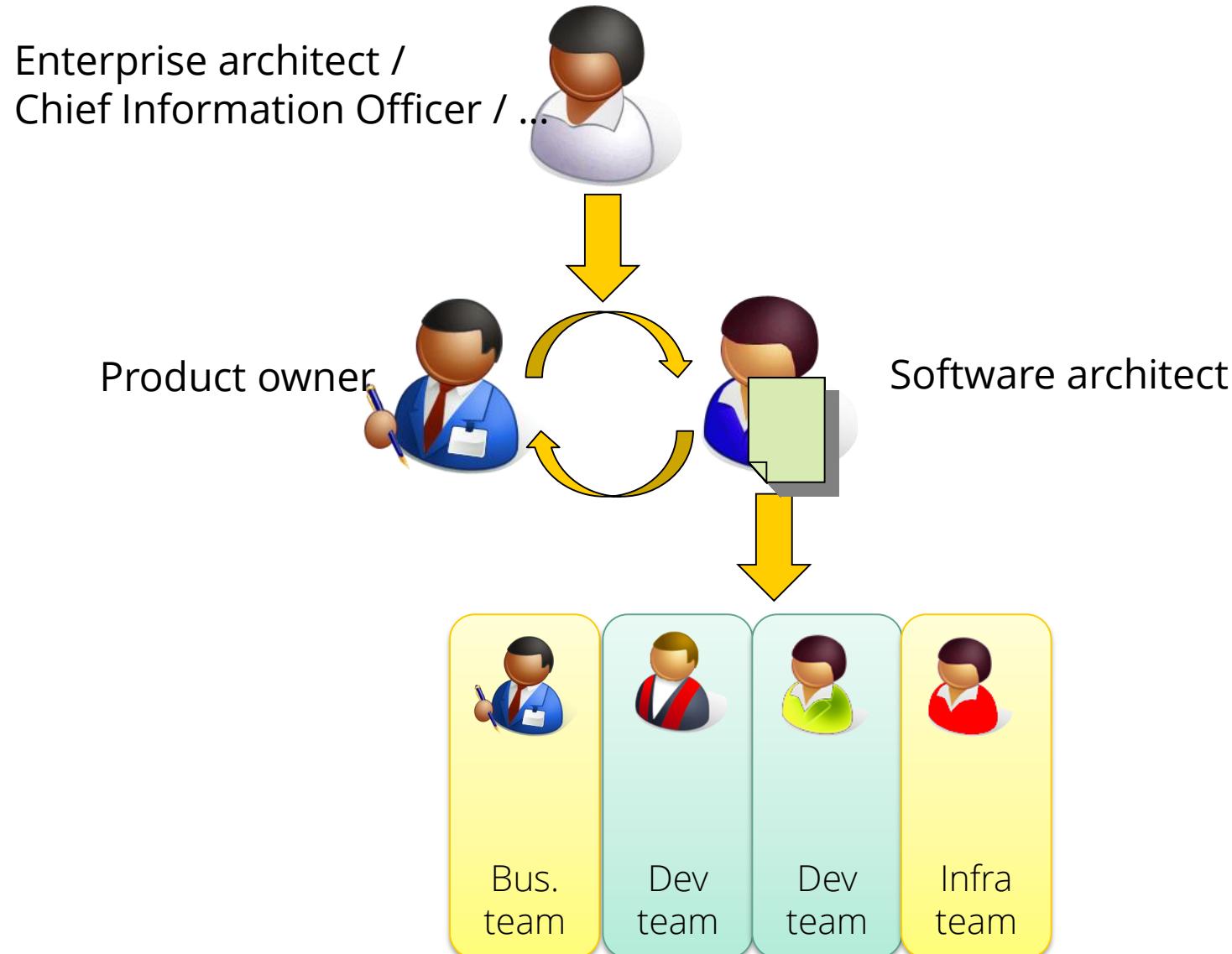
- Why is the architecture as it is?
 - Intertwined/fragmented design decisions**
 - Evolution: rules and constraints are violated, architecture drift**
 - Obsolete decisions are never removed**
- Architectural design decisions as first class elements?
 - Guarding conceptual integrity**
 - Explicit design space exploration**
 - Traceability!**
- Requirements
 - Explicit architectural changes**
 - Corrective, perfective and adaptive support**
 - Clear relationship between architecture and system realization**



The process of making decisions...



Traditional approach: single decision maker





Evolution in Software



Consequence: Architecture evolves much quicker

New term: "Fluid architecture work"

DevOps

Microservices

Cloud computing

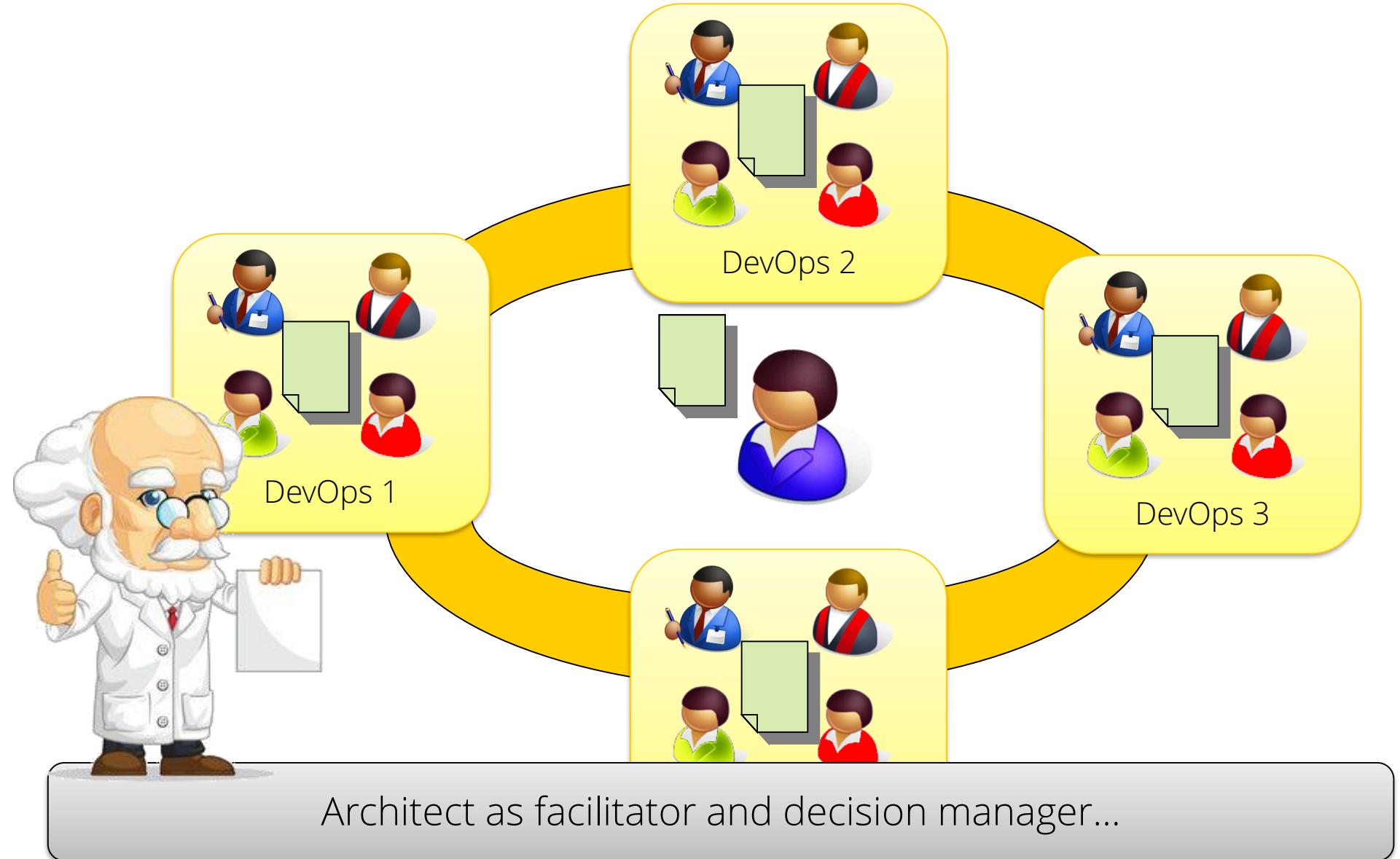
Agile

Common problems:

- Sharing architecture information
- Understanding system-wide view
- Resistance to **any** design work
- Desire for "complete" AD upfront
- Enterprise architect (!)



Shift towards facilitating decision making

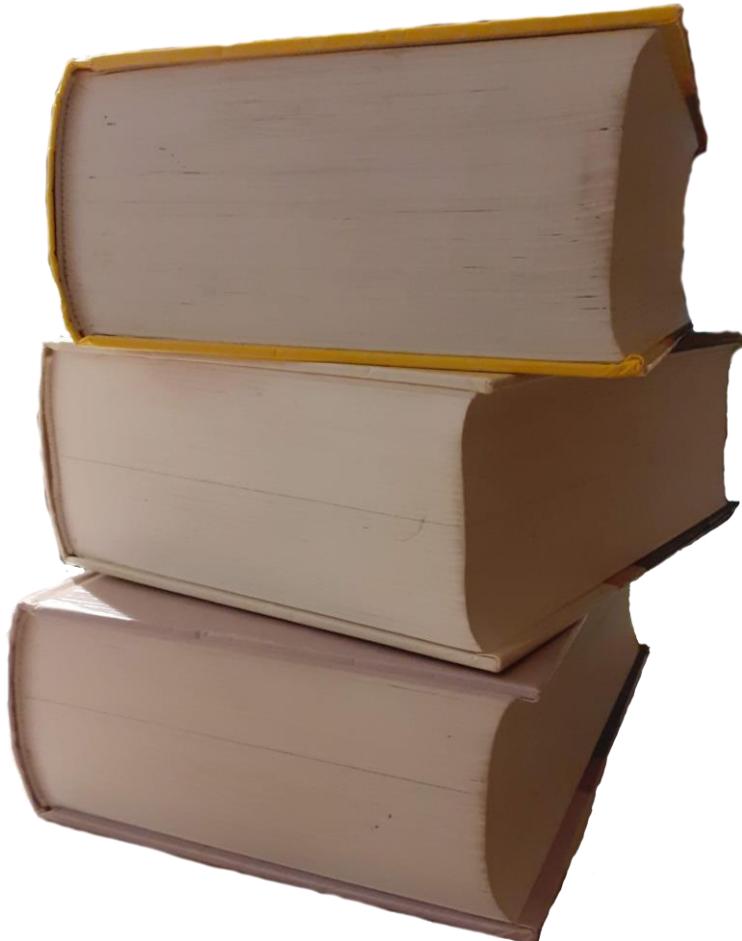




First step: recording decisions...



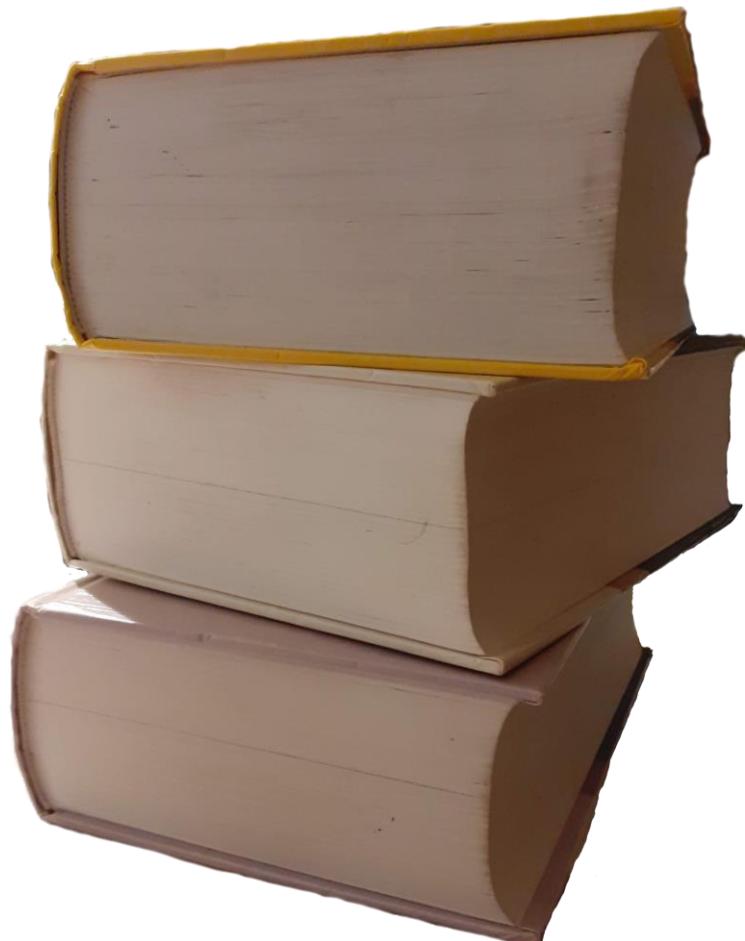
Documenting Architecture Decisions



- Remember the agile manifesto:
**Working software over comprehensive documentation
Continuous attention to technical excellence and
good design enhances agility**
- This means:
**Agile methods are not opposed to documentation
Only to valueless documentation**
- And above all: nobody ever reads large documents!



How do you document design decisions?



- It's tough!
Extensive decision document: Newcomers may be perplexed, delighted, or infuriated by some past decision
- What can this person do?
Blindly accept the decision
Blindly change the decision
- Another problem:
Do you still remember after a year why you decided things?
How about a system that is 50 years old?



Architecture decision records

A screenshot of a Sublime Text editor window titled "git • (CAiSE 2019) - Sublime Text (UNREGISTERED)". The file menu is open, showing "File", "Help". The main pane shows a directory structure under the path "adr":

```
>adr
  - ->0001-decision.md
  - ->0002-decision.md
  - ->0003-decision.md
  - ->0004-decision.md
```

- Old idea, revived by Nygard in 2011
 - Simple text files**
 - Stored with the code**
- Nygard's form:
 - Title**
 - Status**
 - Context**
 - Decision**
 - Consequences**



22 lines (11 sloc) | 466 Bytes

Raw Blame History

s Help

```
>adr
-->0001-dec
-->0002-dec
-->0003-dec
-->0004-dec
```

in 2011

3. Use JUnit 5 for testing

Date: 2018-09-27

>Status

Accepted

Supercedes [2. Use JUnit 4.12 for testing](#)

Context

JUnit 5 was released and has nice additional features

Decision

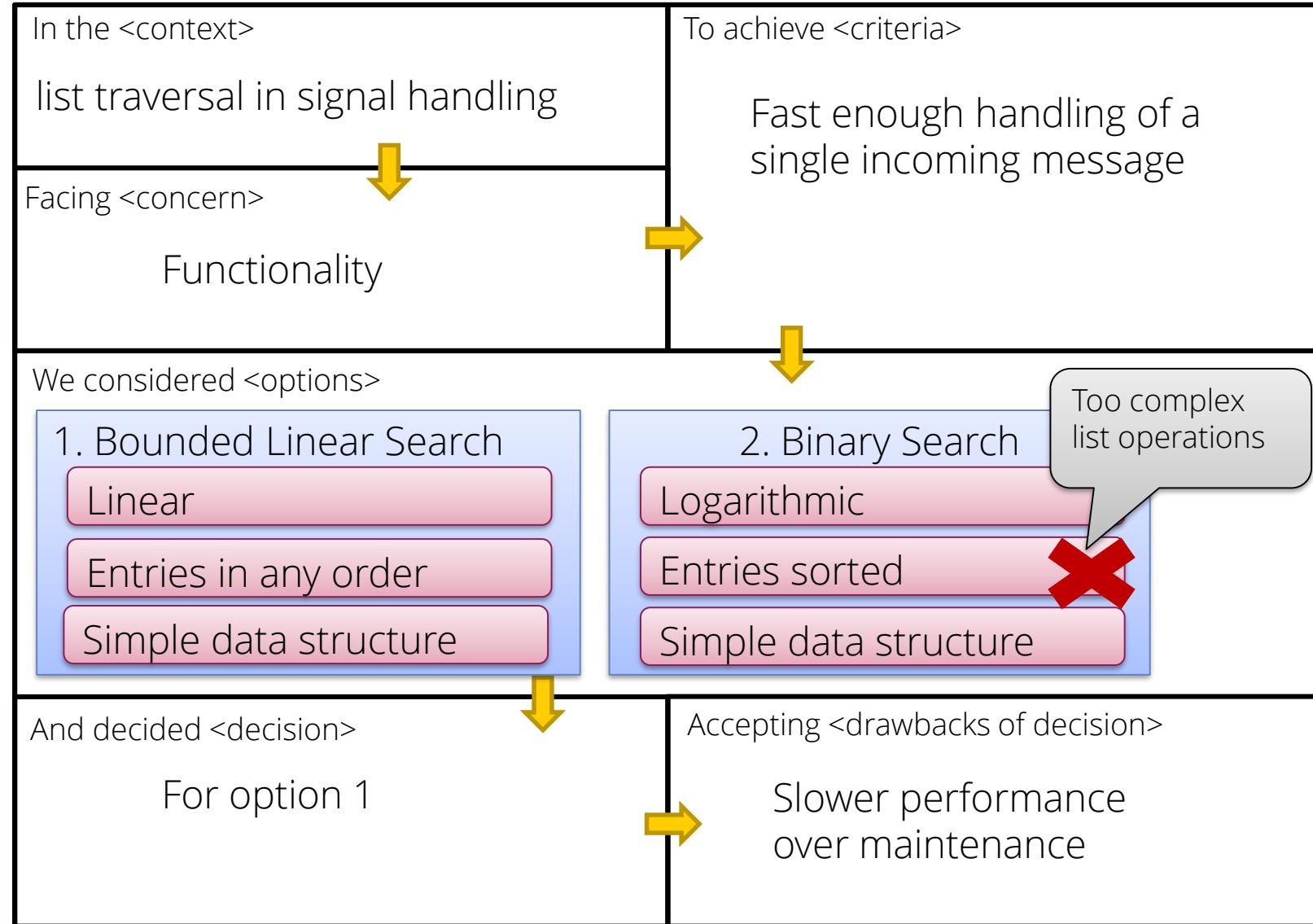
We've decided to adopt JUnit 5 because it handles testing for exceptions better and offers dynamic testing capabilities

Consequences

@Rule annotations we use are deprecated therefor some integration tests might need to be migrated.



Current research: decision stories





Current research: decision stories

In the <context>

list traversal in signal handling

To achieve <criteria>

Fast enough handling of a

Fac

Decision story: List traversal

In the context of
facing

We decided to

And not

To achieve

Accepting

list traversal in signal handling,
functionality,
use the Bounded Linear Search
the Binary Search
fast enough message handling
the drawback in performance
over maintenance.

And decided <decision>

For option 1

Accepting <drawbacks of decision>

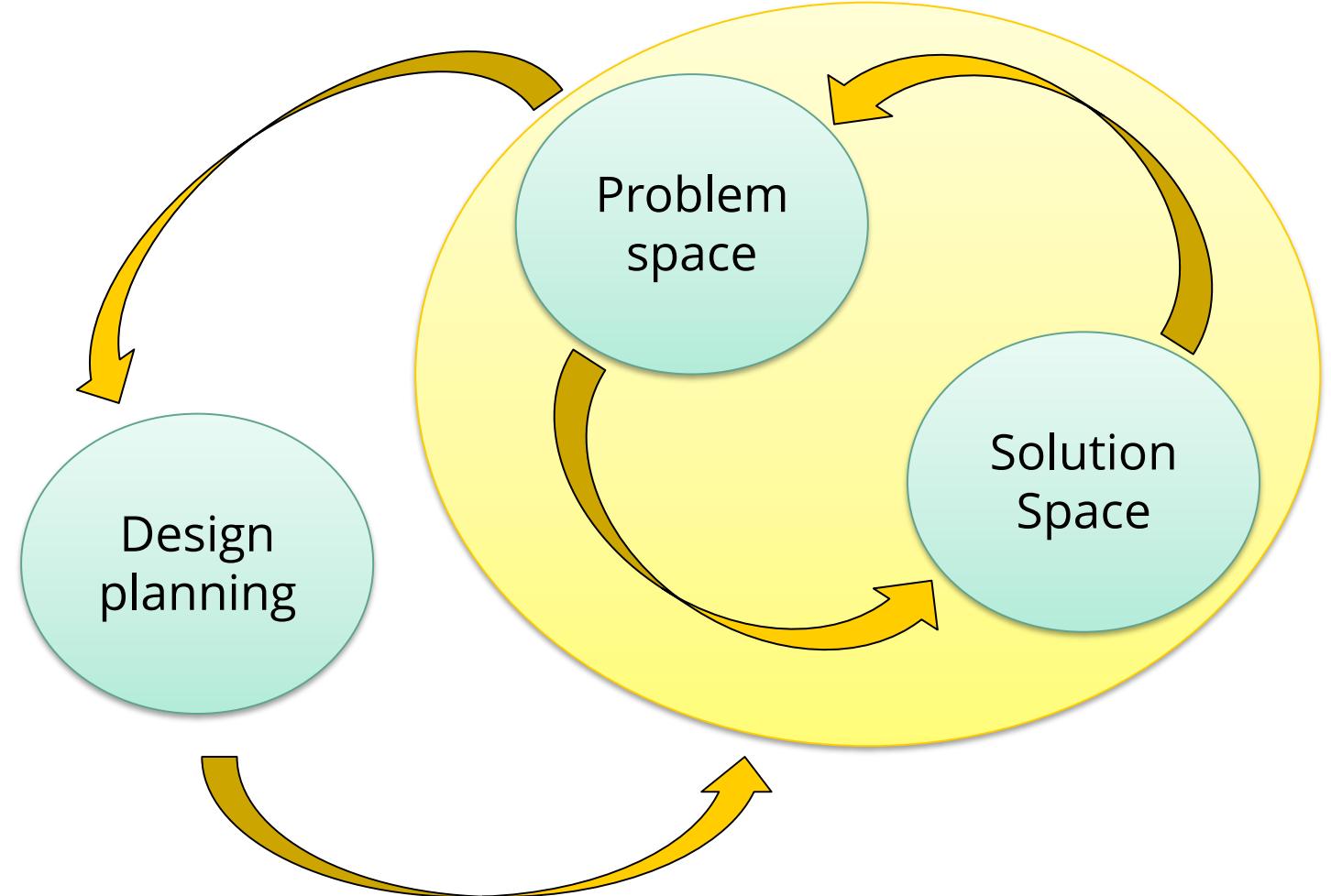
Slower performance
over maintenance



Decisions: what does psychology tell us?

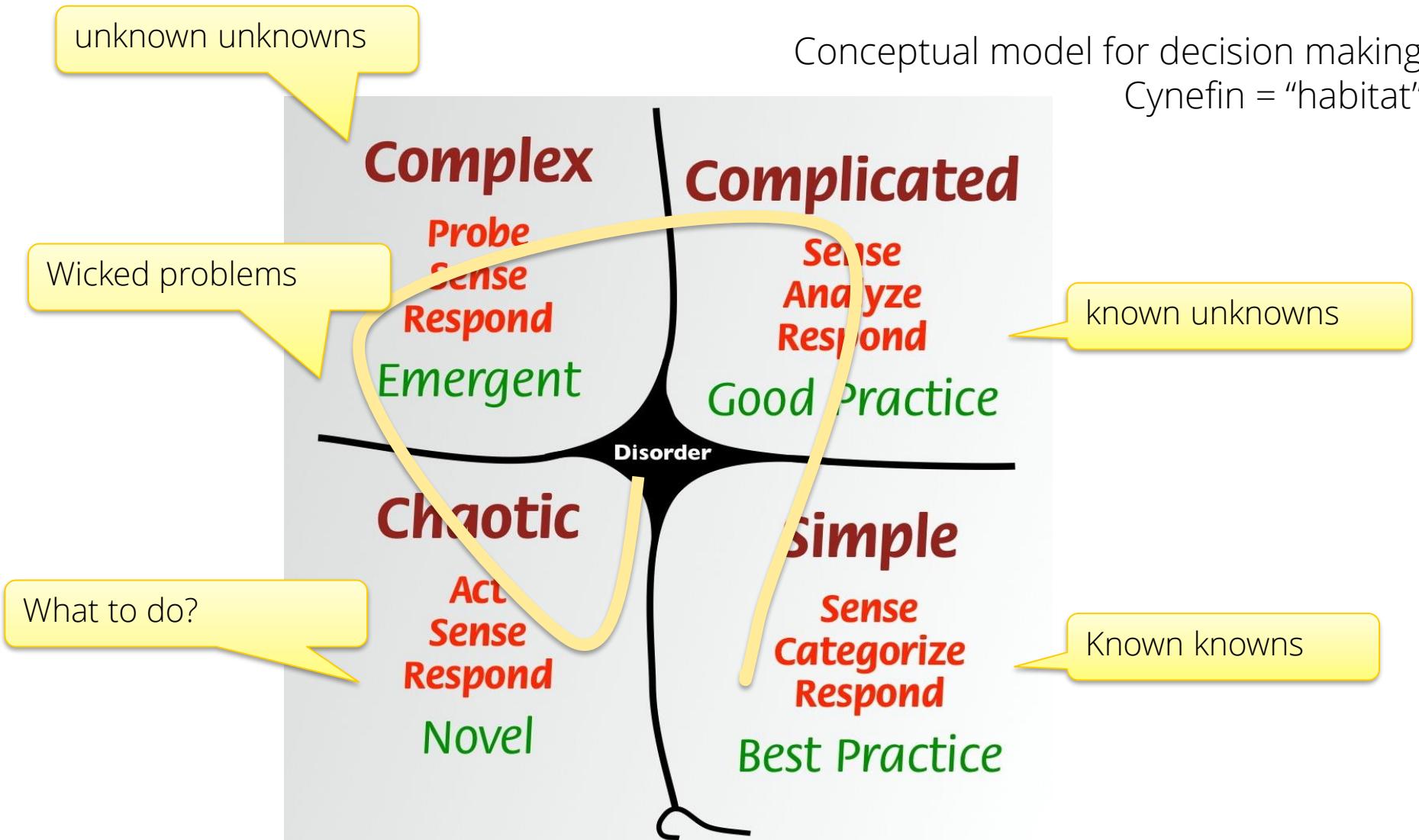


Making design decisions: moving in different spaces



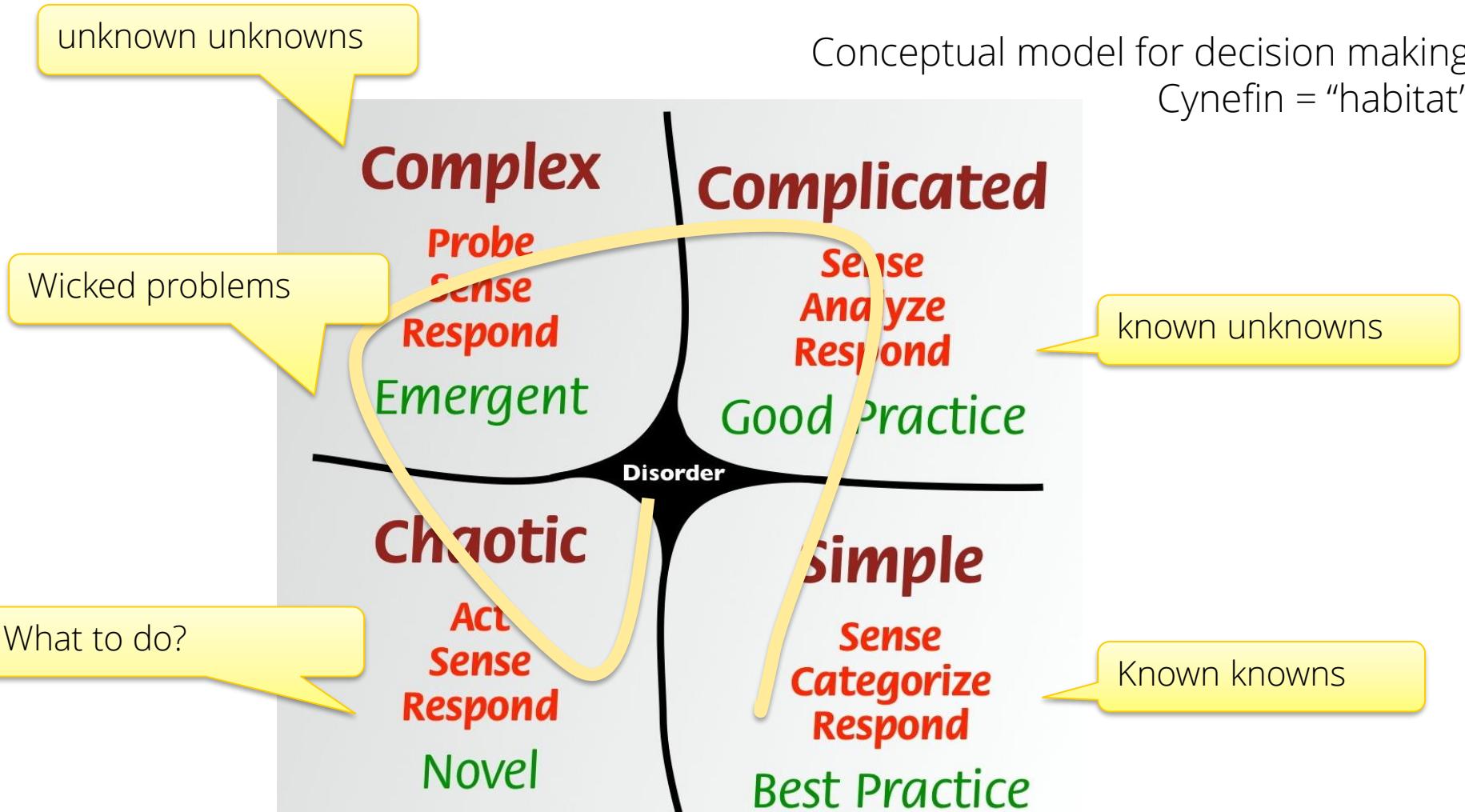


Cynefin model (Snowden, 1999)





Cynefin model (Snowden, 1999)

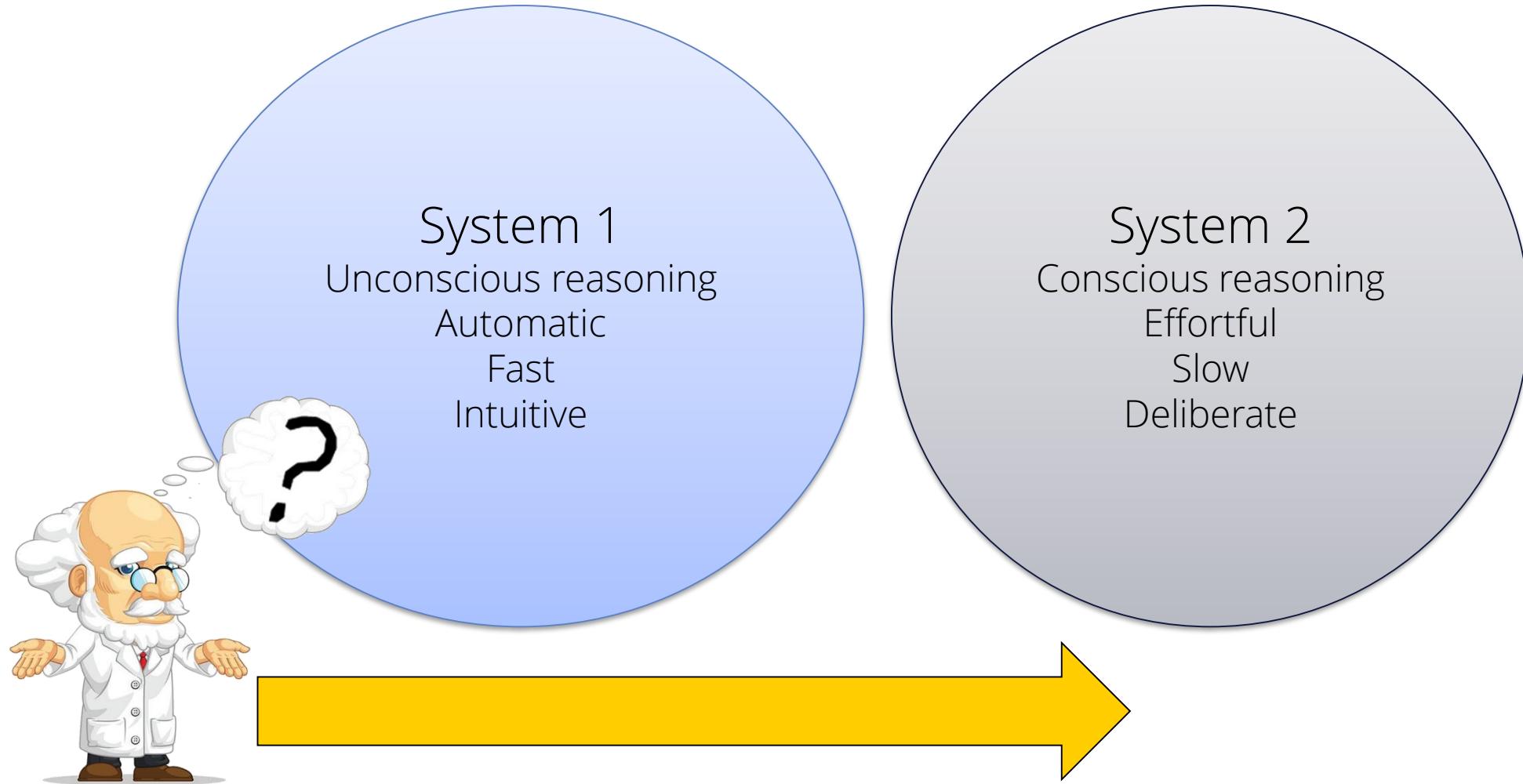


Architecture: from chaotic to complex to complicated...



Nice, but how do we make decisions?

D. Kahneman





Is system 2 always better?

(In software engineering disciplines)



What is intuition?

- Intuition in decision making:
Unsubstantiated attitude towards a decision alternative
Outcome of unconscious processing

Benefits

- Higher capacity (because unconscious)
- Access to implicit knowledge
- Openness to change
- Natural weighing options
- Making **new** associations
- Match complex patterns

Drawbacks

- Does not follow strict rules
- No process awareness
- No reasons available
- Can be tricked with numbers



Intuition vs. rational

- Emily's father has three daughters. The first two are named April and May. What is the third daughter's name?
- A farmer had 15 sheep and all but 8 died. How many are left?
- If you're running a race and you pass the person in second place, what place are you in?
- In a lake, there is a patch of lily pads. Every day, the patch doubles in size. If it takes 48 days for the patch to cover the entire lake, how long would it take for the patch to cover half of the lake?

This is called “Religious thinking”: be aware of system 1



Intuition vs. rational

- You are shown a set of four cards placed on a table, each of which has a number on one side and a colored patch on the other side. The visible faces of the cards show 3, 8, red and brown. Which card(s) must you turn over in order to test the truth of the proposition that if a card shows an even number on one face, then its opposite face is red?

Context matters!

- You are shown a set of four cards placed on a table, each of which has a number on one side and a drink on the other side. The visible faces of the cards show 16, 21, Coke and Beer. Which card(s) must you turn over to test the rule “You’re only allowed to drink alcohol if you’re 18 or older?”



Intuition vs. rational

- We tend to have **cognitive bias**
Representativeness bias
Availability bias
Anchoring

- We tend to do **illogical reasoning**

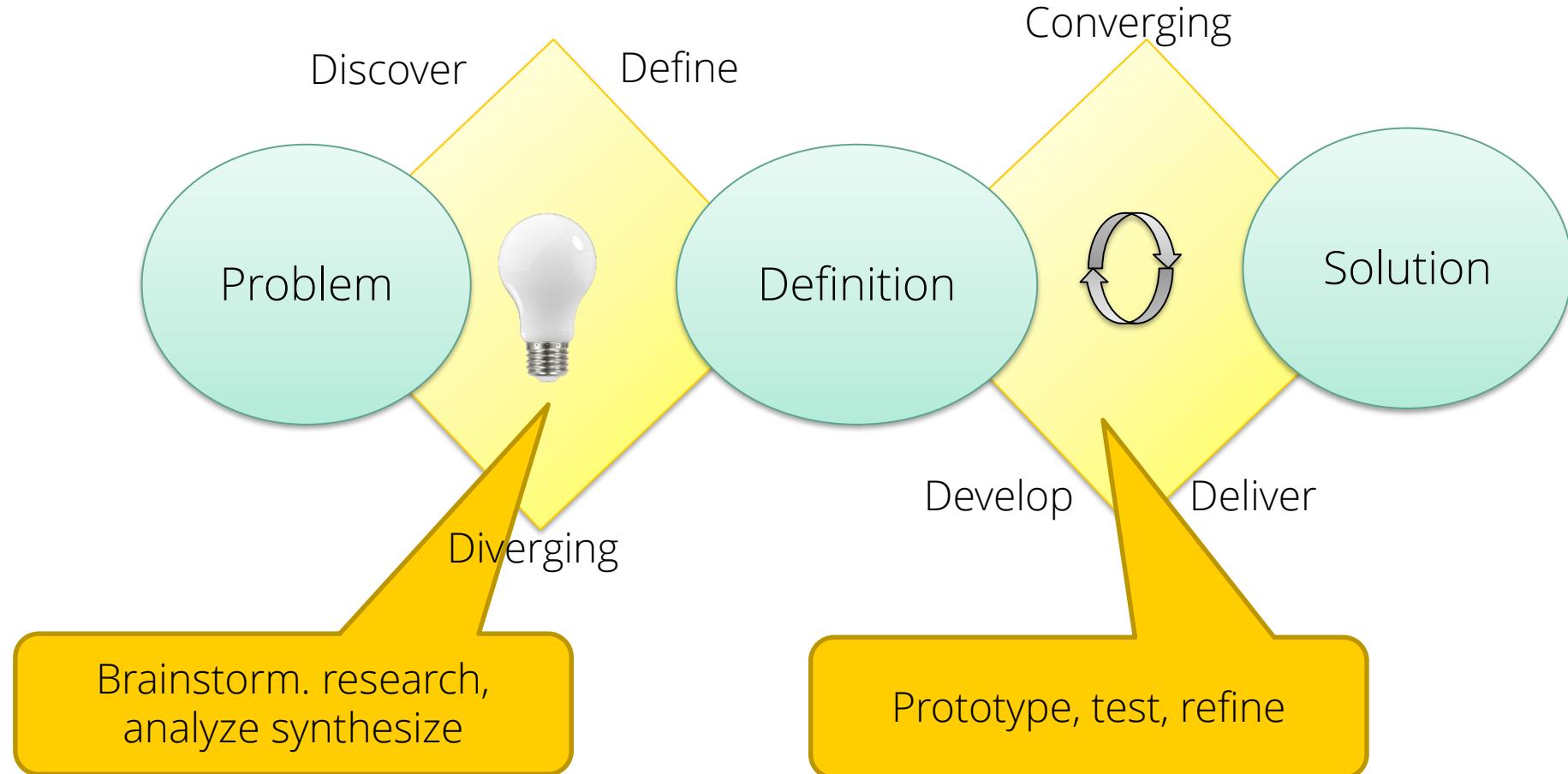
- We tend to use **low quality premises**



Important distinction: ask not IF but WHEN to use intuition



Intuition vs. rational





Group decision making

Group decision-making is a research area that aims to understand and develop methods to enhance the collective decision process.



Topic for SE conference

Bestand Bewerken Bekijken Invoegen Opmaak Gegevens Extra Uitbreidings Help

B7

	A	B	C
1	ID	Topic	Description (why interesting)
2		1	
3		2	
4		3	
5		4	
6		5	
7		6	
8		7	
9		8	
10		9	
11		10	
12		11	
13		12	
14		13	
15		14	
16		15	
17		16	
18		17	
19		18	
20		19	
21		20	
22		21	
23		22	
24		23	
25		24	
26		25	
27		26	
28		27	
29		28	



<https://bit.ly/2WHoAay>

A screenshot of a Google Forms survey titled "Vote for topic". The survey instructions state "Select 3 out of the topics". There is a list of 13 topics, each preceded by a checkbox. A purple sidebar on the right contains a pen icon and a small "0" indicating no responses. The URL https://docs.google.com/forms/d/e/1FAIpQLScc42_UzVllg1n_jXfiar2L0lrAeEKK6ejQcCnlpc9DM5Cbw/viewform is visible in the browser's address bar.

Vote for topic

Select 3 out of the topics

- Topic 1
- Topic 2
- Topic 3
- Topic 4
- Topic 5
- Topic 6
- Topic 7
- Topic 8
- Topic 9
- Topic 10
- Topic 11
- Topic 12
- Topic 13

0

https://docs.google.com/forms/d/e/1FAIpQLScc42_UzVllg1n_jXfiar2L0lrAeEKK6ejQcCnlpc9DM5Cbw/viewform



<https://bit.ly/2HVR6qm>

Screenshot of a Google Sheets document titled "Topic for exam". The document contains a table with columns for Topic ID, Name, Notes, Points, and Voters. The "Points" column is highlighted in green, and the "Voters" column is highlighted in orange. The table has 35 rows, numbered 1 to 35. The "Name" column contains names such as <name 1> through <name 11>, and the "Notes" column is empty. The "Points" and "Voters" columns both show values of 0 for all rows.

Topic ID	Name	Notes	Points	Voters
1	<name 1>		0	0
2	<name 2>		0	0
3	<name 3>		0	0
4	<name 4>		0	0
5	<name 5>		0	0
6	<name 6>		0	0
7	<name 7>		0	0
8	<name 8>		0	0
9	<name 9>		0	0
10	<name 10>		0	0
11	<name 11>		0	0
12			0	0
13			0	0
14			0	0
15			0	0
16			0	0
17			0	0
18			0	0
19			0	0
20			0	0
21			0	0
22			0	0
23			0	0
24			0	0
25			0	0
26			0	0
27			0	0
28			0	0
29			0	0
30			0	0
31			0	0
32			0	0
33			0	0
34			0	0
35			0	0



Group decision making

- Group decision-making
 - A research area that aims to understand and develop methods to enhance the collective decision process.**
- Main factors:
 - Social links among team members**
 - Context**
 - Group characteristics**
 - Selection strategy**
 - Conflict management**
 - Norms and regulations**



Group decision making in practice

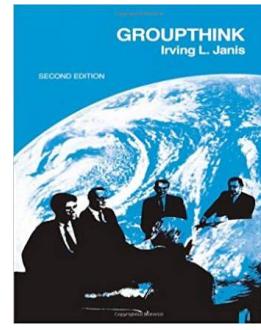
- Typically 3 – 5 people
- Homogeneous group
→ **groupthink!**
- Mostly discussion-based and democratic
→ **information asymmetry**



GroupThink

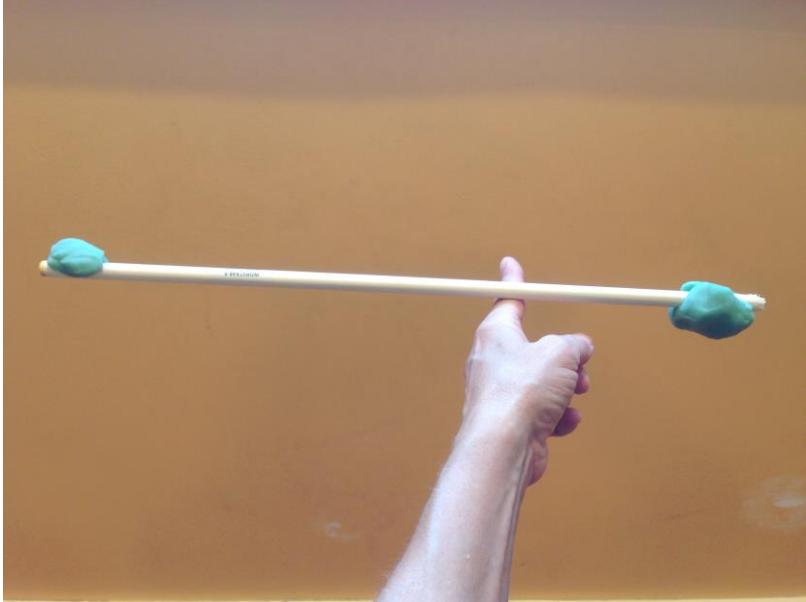
The tendency of groups to try to minimize conflict and reach consensus without sufficiently testing, analyzing and evaluating their ideas:

- The pressures for conformity
- Restrict the thinking of the group,
- Bias its analysis
- Promote simplistic and stereotyped thinking
- Stifle individual creative and independent thoughts





Information asymmetry



- One party knows more than the other parties
And able to take advantage of it!



Information asymmetry



- One party knows more than the other parties
And able to take advantage of it!
- Examples:
Lecturer who teaches material to students



Information asymmetry



- One party knows more than the other parties
And able to take advantage of it!
- Examples:
Lecturer who teaches material to students
Doctor who knows more than the patient
A seller, who has more background than the potential buyer
Politics: lobbyist who wants to get a certain decision

What would be an example in Software Architecture?



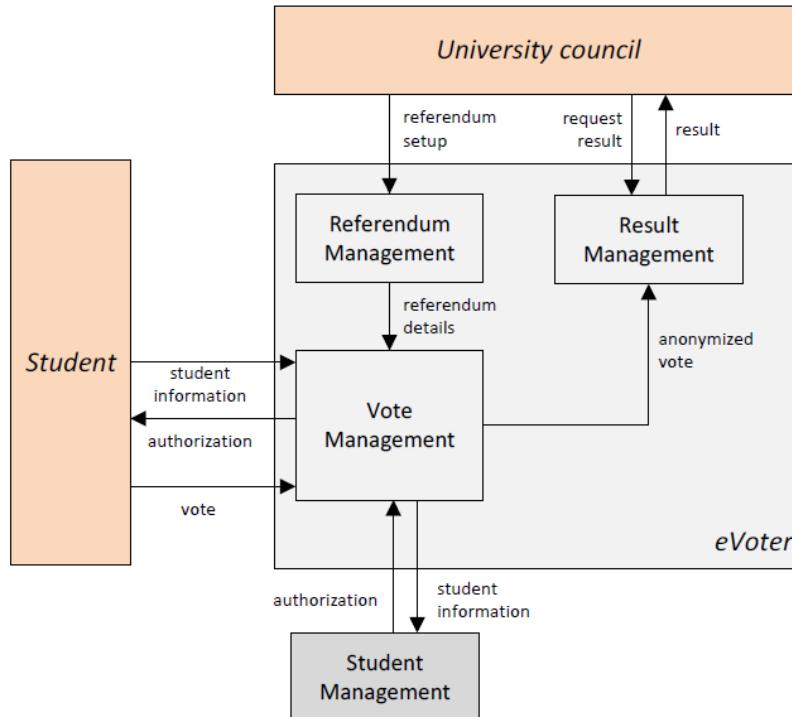
Group Decision Making (GDM) methods



- Brainstorming
Participants freely propose list of alternatives, then brainstorm to arrive at final decision
- Voting
Participants propose alternative, and then voted
- Delphi
**Experts answer in a distributed and anonymous way.
Facilitator provides anonymous summary**
- Consensus selection
Consensus: alternatives are listed, an effort is made to achieve maximum level of consensus
- Analytic Hierarchy Process (AHP)
Model as goals, alternatives and criteria. Participants do pairwise comparison, results are synthesized.

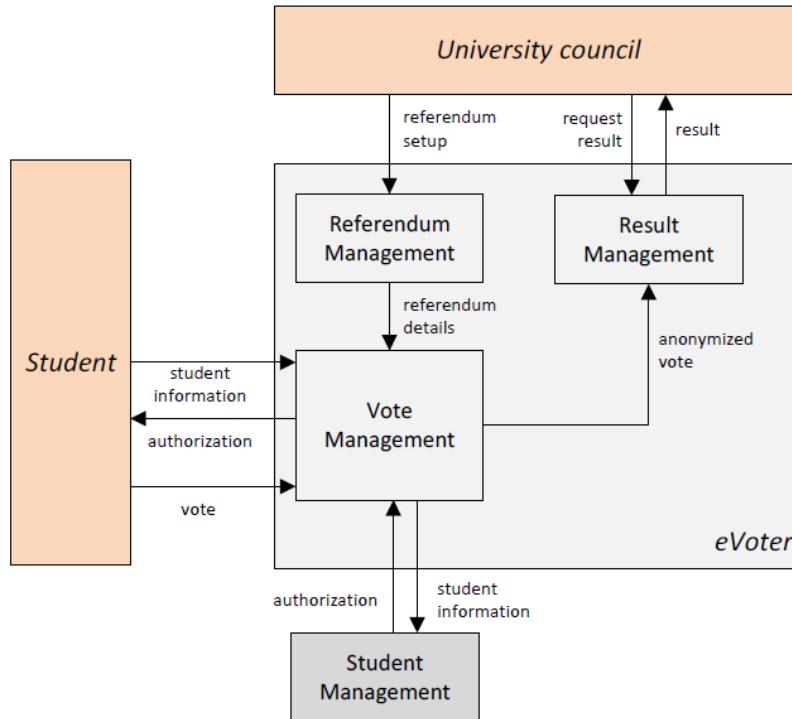


Assignment



Architecture decisions are hard

- “Architecture is the decisions that you wish you could get right early in a project.” – Ralph Johnson
- “Architecture is about the important stuff. Whatever that is.” – Martin Fowler
- Architecture is about “Things that people perceive as hard to change” – Martin Fowler



Architecture decisions are hard

- “Architecture is the decisions that you wish you could get right early in a project.” – Ralph Johnson
- “Architecture is about the important stuff. Whatever that is.” – Martin Fowler
- Architecture is about “Things that people perceive as hard to change” – Martin Fowler

What would be an architectural decision for the eVoter?



For now: your architecture!



- Assess your architecture
 - Which decisions did you take?**
 - Why?**
 - Can you document them using the template?**



Agenda for today



- 09:00 – 09:45: Patterns and styles
- 09:45 – 11:00: You: work on the assignment
- 11:00 – 11:30: Documenting architectures
- 11:30 – 12:40: You: work on the assignment
- 12:40 – 12:45: Wrap-up



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