GÖTEBORGS UNIVERSITET

CHALMERS

GÖTEBORGS UNIVERSITET

CHALMERS

DAT 550 / DIT 978
Advanced Software Engineering for AI/ML-Enabled Systems

GÖTEBORGS UNIVERSITET
CHALMERS

Lecture 3: On AI/ML System Architecture

Your teachers:
Hans-Martin Heyn, Universitetslektor,
Eric Knauss, Docent

Computer Science and Engineering Department, Göteborg University

What will you learn?

Architectures and patterns for AI/ML-enabled systems

- How can we get from (prototyping) models to production systems
- Modularity and the problem of ML components in larger software systems
- An introduction to an architecture framework for distributed AI-enabled systems

- Explain how ML fits into the larger pictures of building and maintaining systems
- Explain the modularity implications
- Understand the need for architecture frameworks in AI system development

What will you learn?

Architectures and patterns for AI/ML-enabled systems

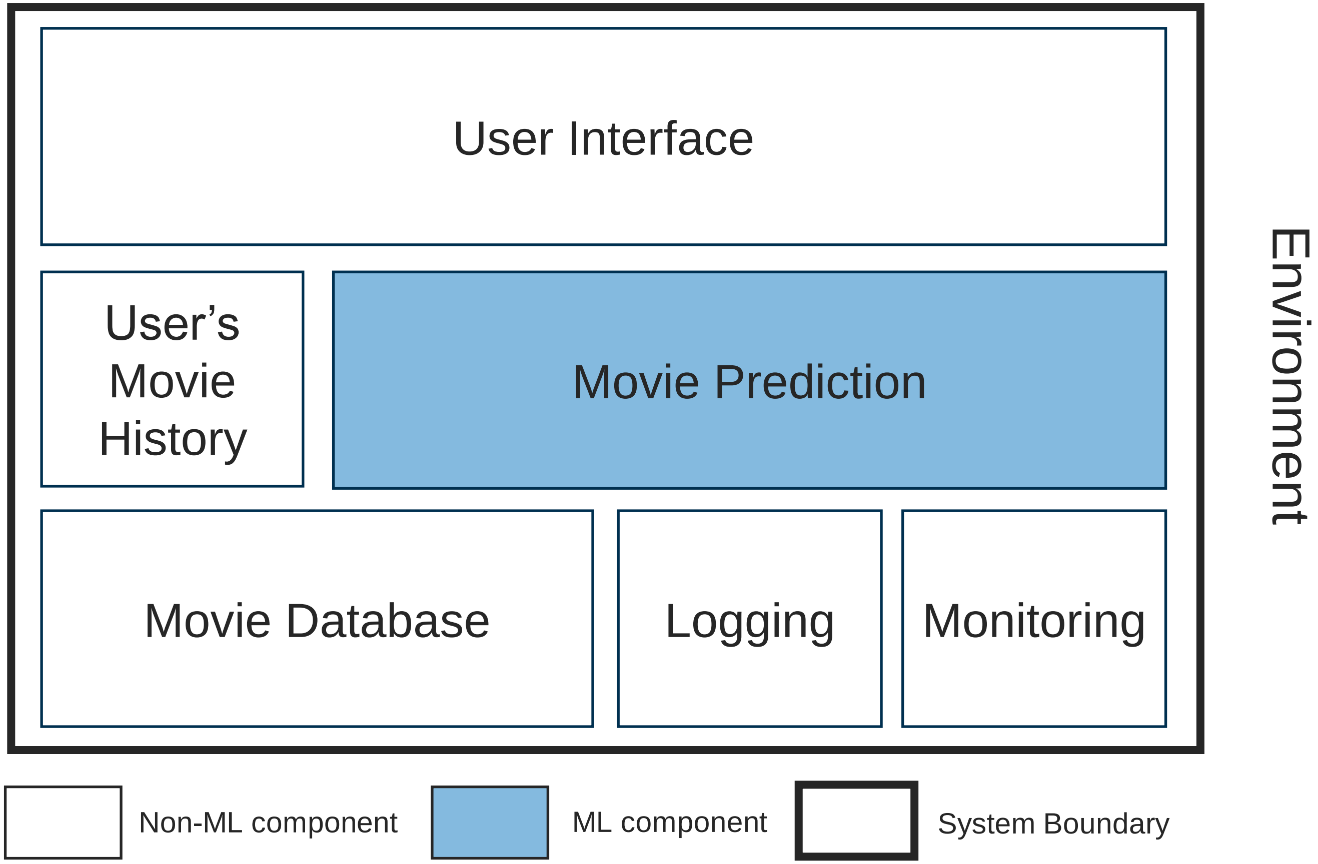
- How can we get from (prototyping) models to production systems
- Modularity and the problem of ML components in larger software systems
- An introduction to an architecture framework for distributed AI-enabled systems

- Explain how ML fits into the larger pictures of building and maintaining systems
- Explain the modularity implications
- Understand the need for architecture frameworks in AI system development

System: Movie Recommendation

?

System: Movie Recommendation



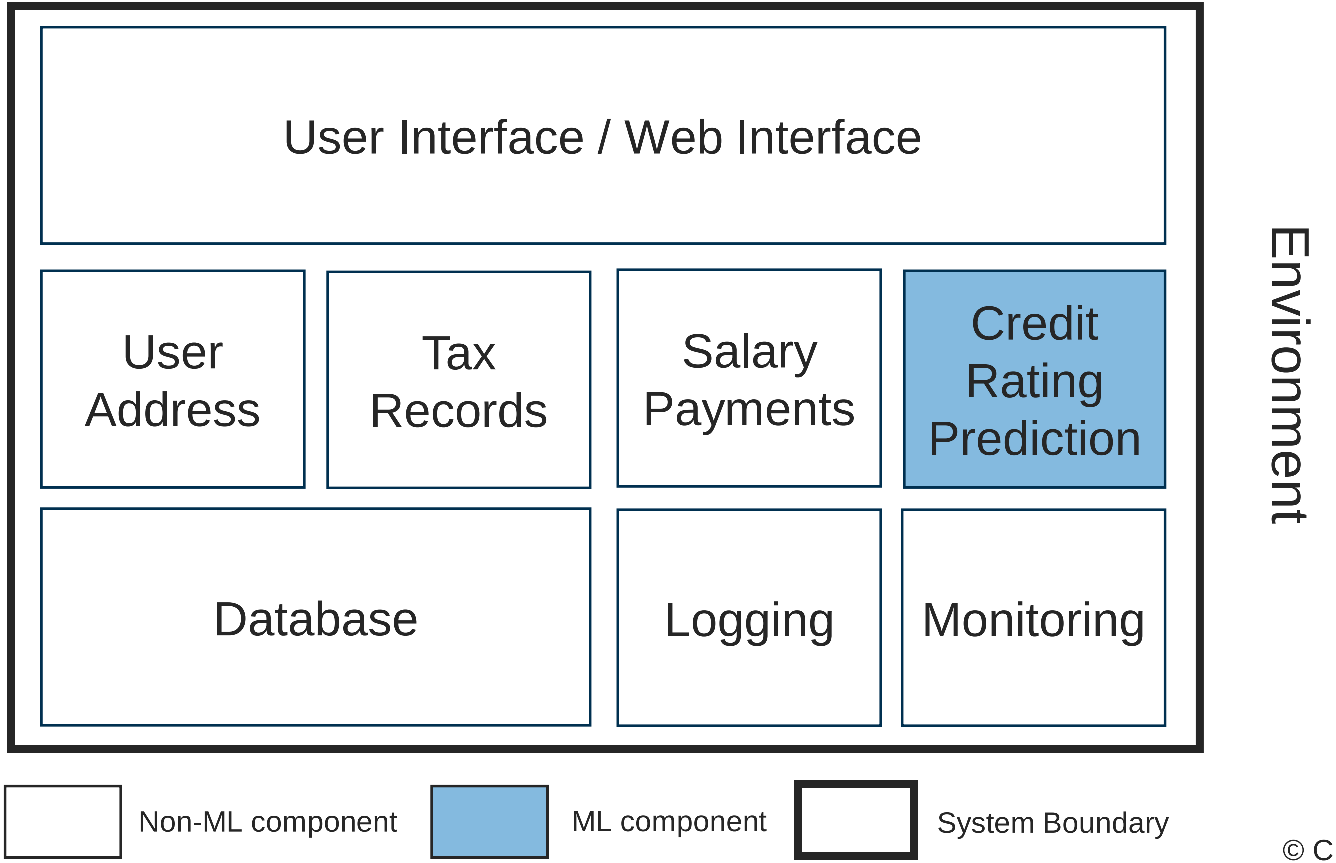
System: Credit Rating

Your credit rating is…

Great news! Our AI predicts a low probability of credit default based on your personal data.
We can offer you a loan for your house of 3.5 MSEK.
Print out your loan application here.

Picture by Christian Kaestner 2022

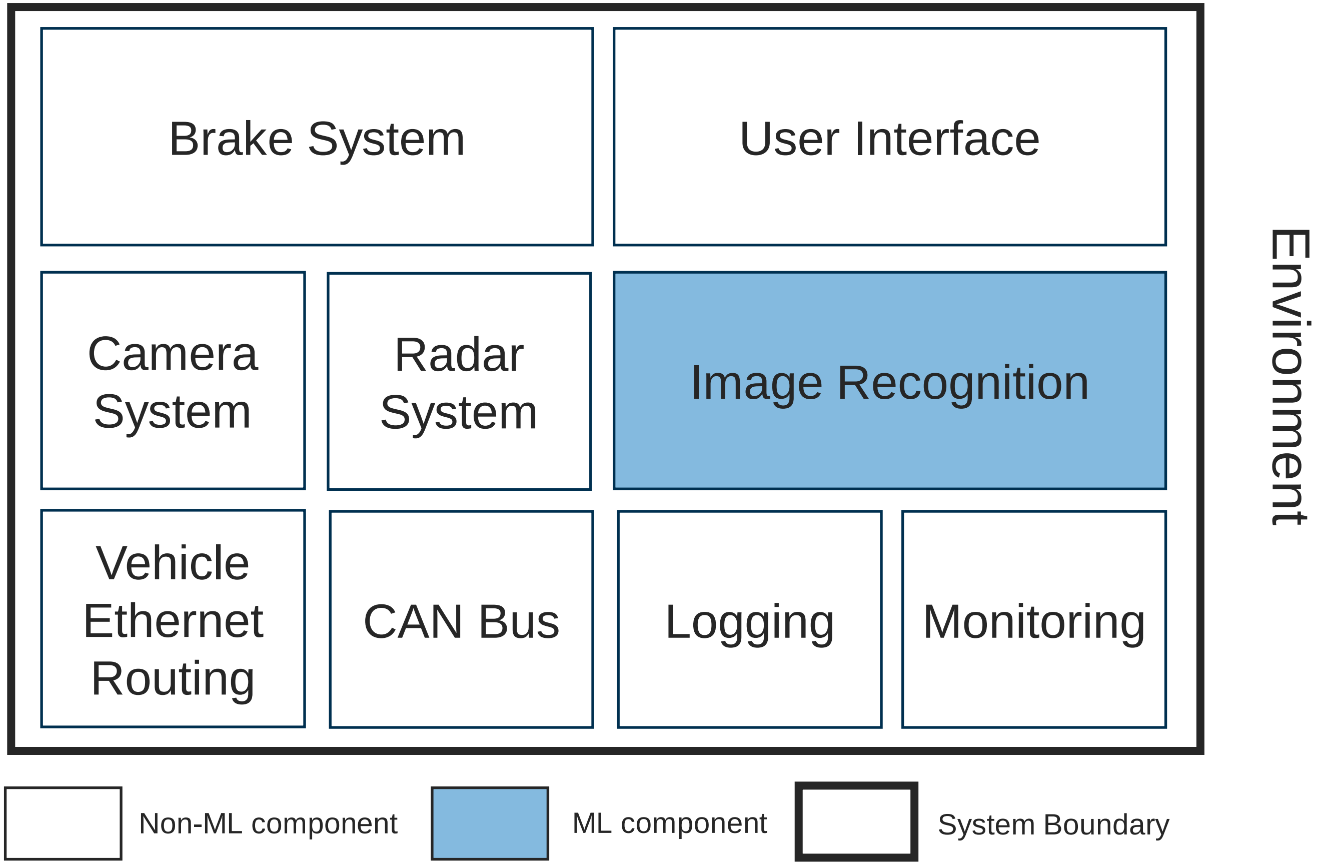
System: Credit Rating



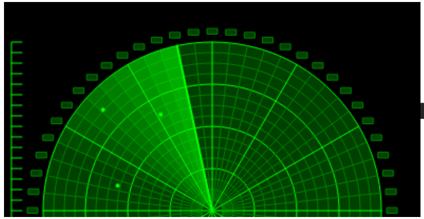
System: Obstacle Detection

Picture by AI Sweden

System: Obstacle Detection



Example: Obstacle Detection



Radar Echoes → Algorithm → ”No Obstacle Detected”

Result

Picture: https://commons.wikimedia.org/wiki/File:Radar_Graphic.svg

Example: Obstacle Detection

GÖTEBORGS UNIVERSITET
CHALMERS

Image

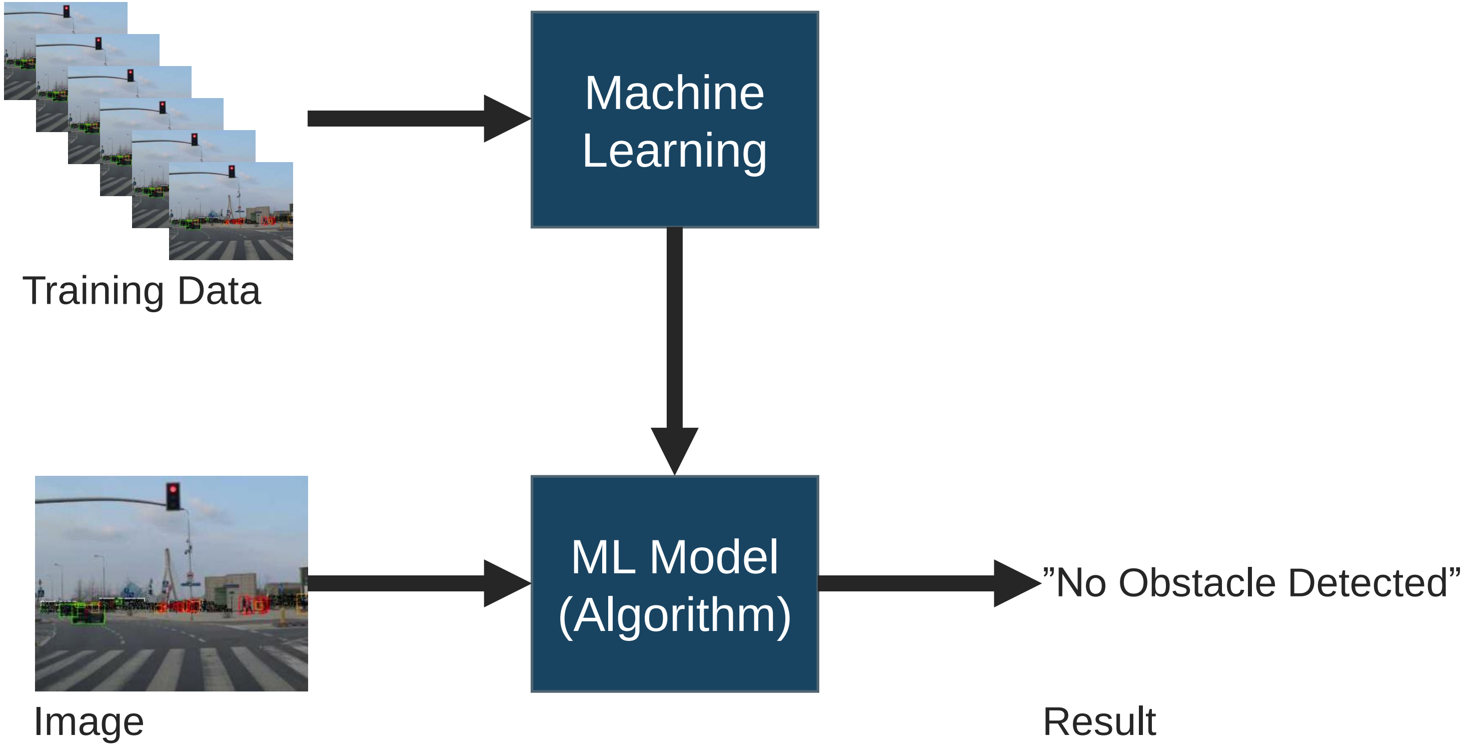
Algorithm

”No Obstacle Detected”

Result

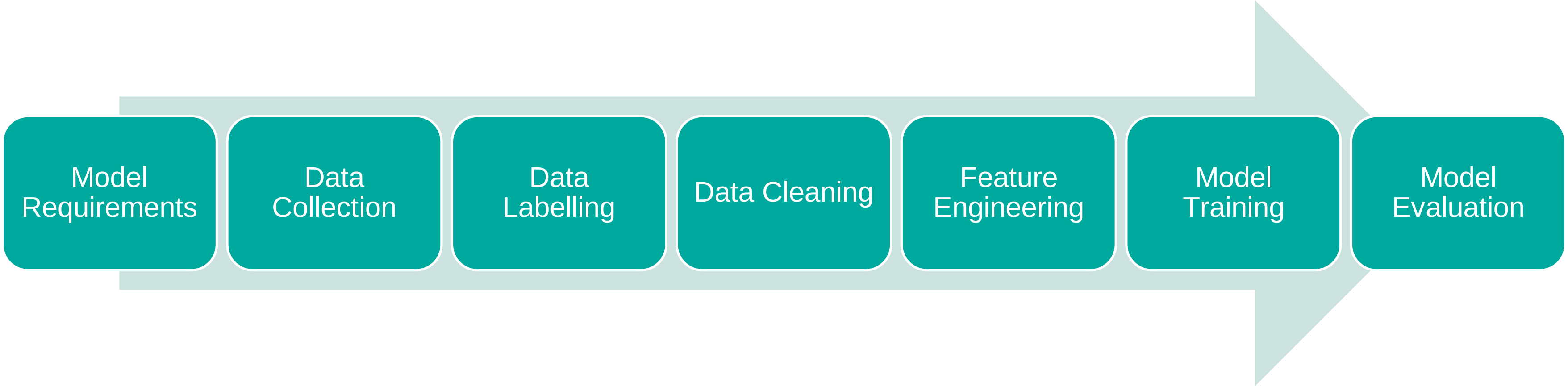
Picture by AI Sweden

Example: Obstacle Detection

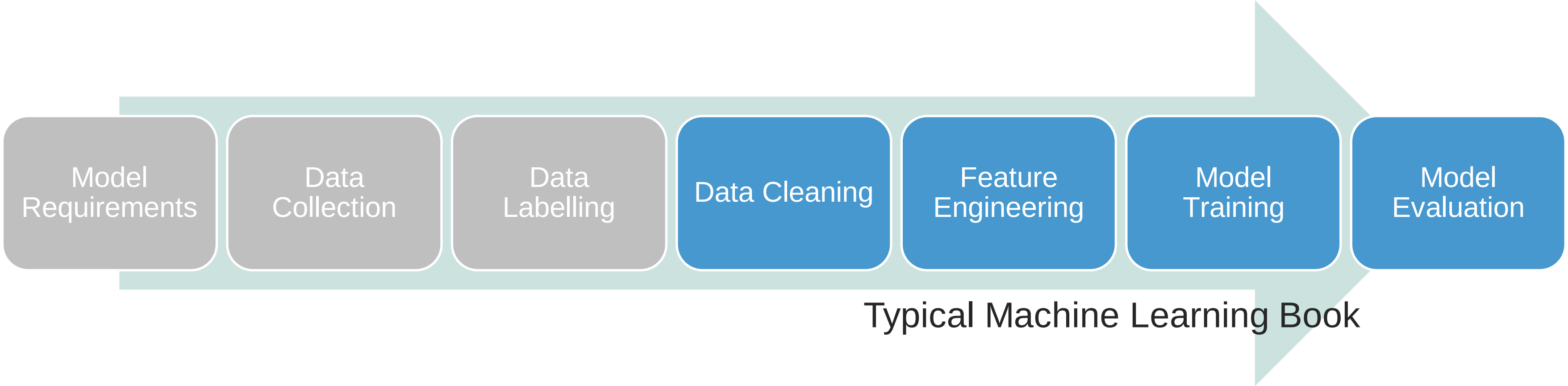


Picture by AI Sweden

Focus on the model vs. system-wide focus



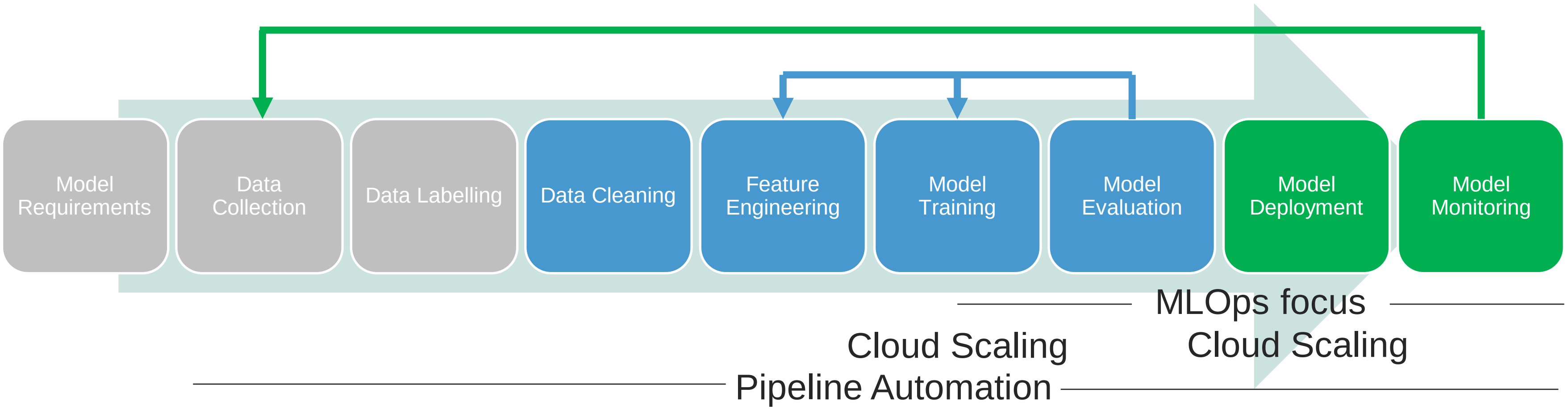
Model Focus (Data Science)



Typical Machine Learning Book

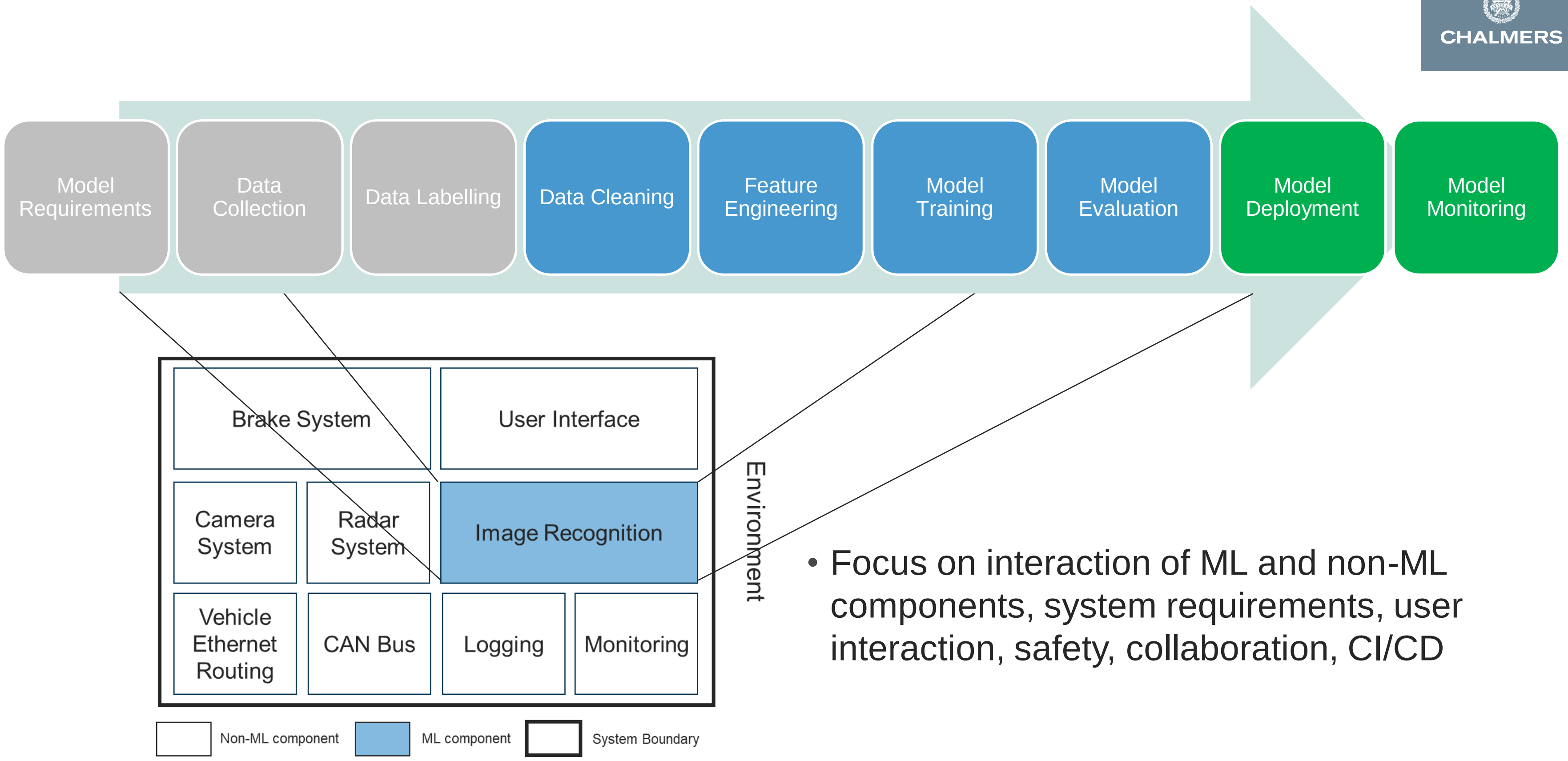
- The (traditional) focus in data science is building models from given data and evaluate the resulting accuracy. Small, prototype style of systems.

Let's automatise (ML Engineer)



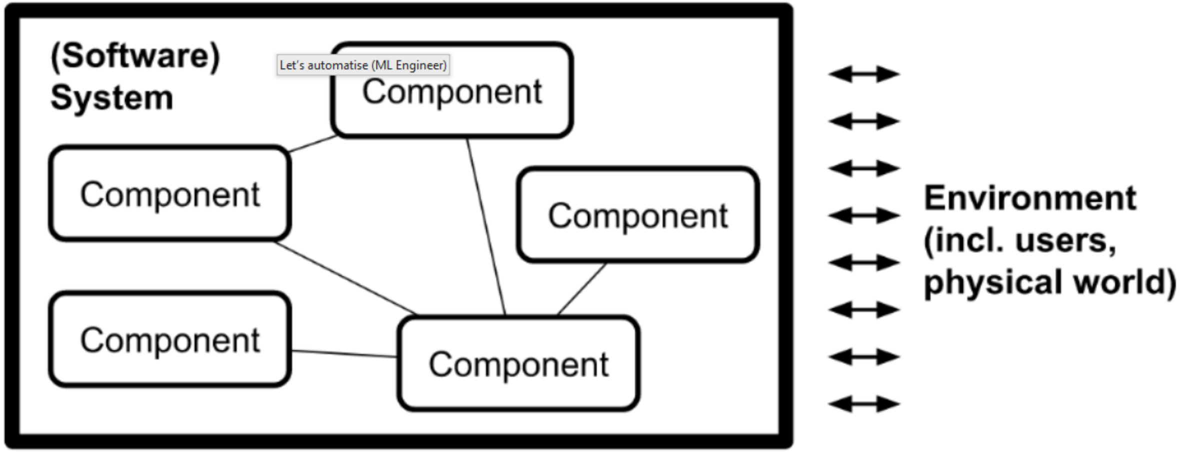
- ML Engineering focuses more on deploying, scaling training and deploying, model monitoring and updating (during operations)

ML in Production



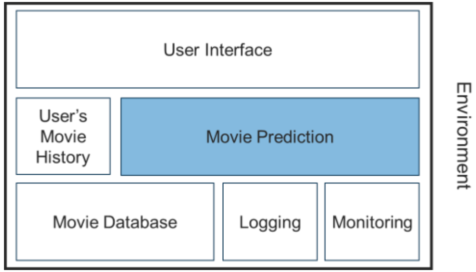
- Focus on interaction of ML and non-ML components, system requirements, user interaction, safety, collaboration, CI/CD

High level system requirements and goals



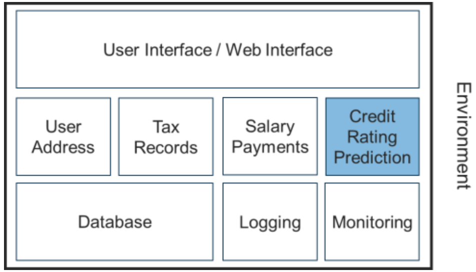
- Systems basically never consist of a single component.
- The ML model is only one of many other components in a system.
- How does it communicate with other components?
- A system architecture brings all components together.

3 Groups: System Goals



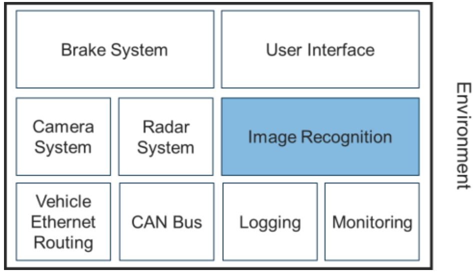
Movie Prediction

- Goal 1
- Goal 2



Credit Rating Prediction

- Goal 1
- Goal 2

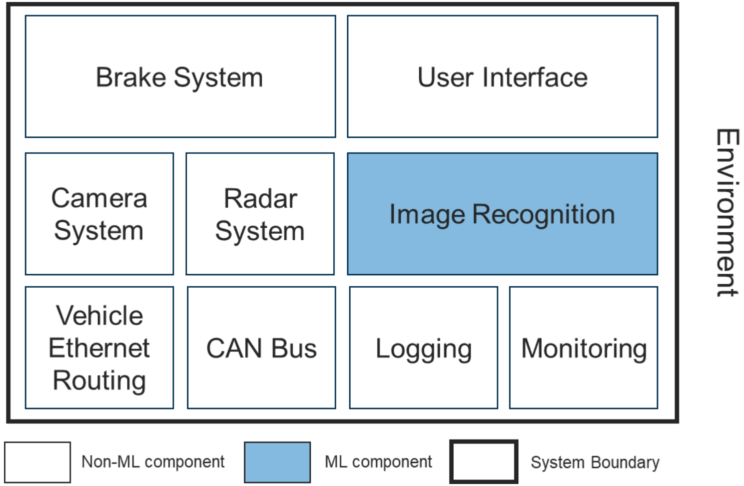


Automatic Emergency Braking

- Goal 1
- Goal 2

GÖTEBORGS UNIVERSITET
CHALMERS

We can define properties / non-functional goals for the system, e.g., safety is a system property



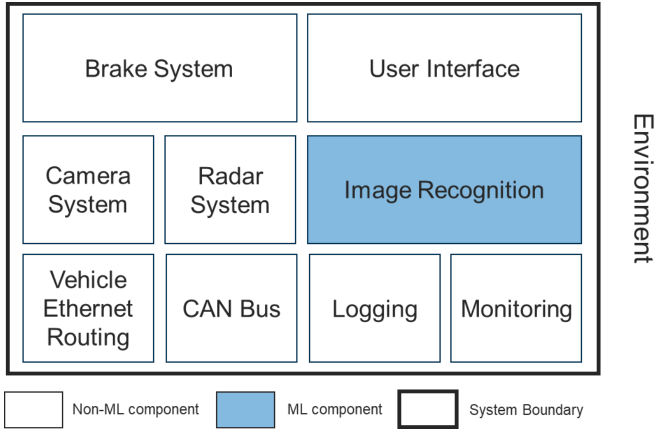
Example of Safety Goals:

- The system shall not trigger an unwanted braking request (ASIL C).
- The system shall not trigger a brake request too late (ASIL B).

How to translate this into model data requirements, accuracy needs, testing conditions?

System goals and model requirements

GÖTEBORGS UNIVERSITET
CHALMERS



Example of Safety Goals:

- The system shall not trigger an unwanted braking request (ASIL C).
- The system shall not trigger a brake request too late (ASIL B).

On one hand we want a “conservative” model that only triggers the brakes when it is *very certain* about an obstacle ahead.

On the other hand, we do not want to trigger the brakes too late. We also need to consider artifacts or noise in the image for example.

A narrow focus only on model accuracy that ignores how the model interacts with the rest of the system might compromise the ability to balance various desired quality aspects.

How to ensure the safety goals (Safety Assurance)

- For / in the model
  - Ensure correct / sufficient training data (how?)
  - Check that operational context (ODD) is as expected (how?)
  - Optimise prediction speed
  - Use confidence checks (what?)
- Outside the model
  - Add redundant non-ML system (radar)
  - Train a redundant independent second model (consequences?)
  - Move responsibility to the user

Picture by Forbes Magazine

Model vs. System Properties

- Similar to safety, many other (non-functuional) requirements / qualities should be discussed at **model and system level**

  - Security
  - Privacy
  - Transparency & Accountability
  - Maintainability
  - Scalability
  - Energy Efficiency

What data do we need?

- Often a model-centric view assumes we have (unlimited) pool of data available.
- In production systems, especially in industries outside of traditional software engineering (e.g., car industry, medical industry, environmental monitoring,…) creating data is actually expensive! Sometimes very very expensive!
- System designers therefore should also focus on how to collect data, label data, document data, plan experiments / data collection campaigns.
- However, defining clear data specifications for ML-enabled system development is not common:

"*Everyone wants to do the model work, not the data work*": Data Cascades in High-Stakes AI

Nithya Sambasivan, Shivani Kapania, Hannah Highfill, Diana Akrong, Praveen Paritosh, Lora Aroyo
[nithyasamba,kapania,hhighfill,dakrong,pkp,loraa]@google.com
Google Research
Mountain View, CA

Patterns

CellPress OPEN ACCESS

Review

Data and its (dis)contents: A survey of dataset development and use in machine learning research

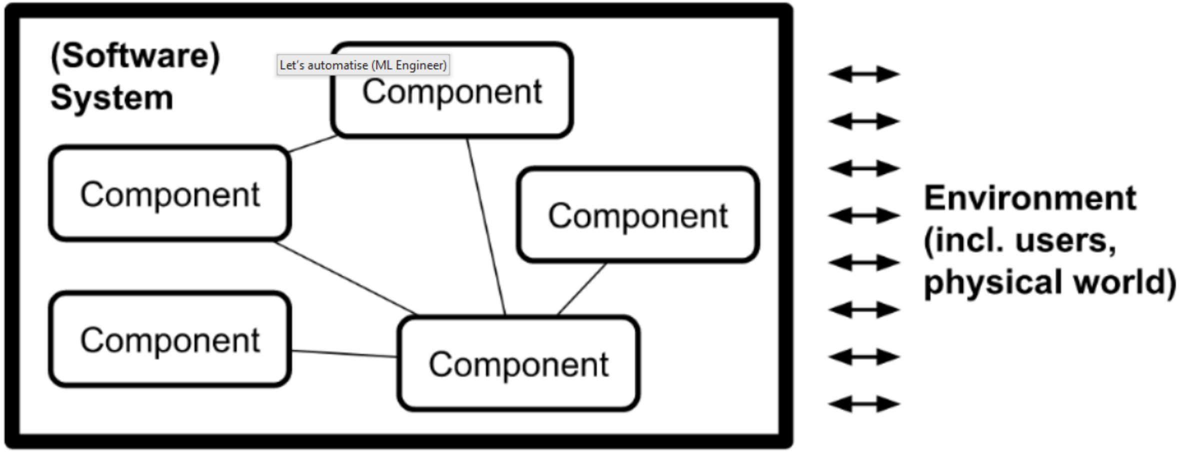
Amandalynne Paullada,[1,*] Inioluwa Deborah Raji,[3] Emily M. Bender,[1] Emily Denton,[2] and Alex Hanna[2,4]
[1]Department of Linguistics, University of Washington, Seattle, WA, USA
[2]Google Research, New York, NY, USA
[3]Mozilla Foundation, Mountain View, CA, USA
[4]Google Research, San Francisco, CA, USA
*Correspondence: paullada@uw.edu
https://doi.org/10.1016/j.patter.2021.100336

An investigation of challenges encountered when specifying training data and runtime monitors for safety critical ML applications*.

Hans-Martin Heyn[1,2[0000−0002−2427−6875]], Eric Knauss[1,2[0000−0002−6631−872X]], Iswarya Malleswaran[1], and Shruthi Dinakaran[1]

[1] Chalmers University of Technology, SE-412 96 Gothenburg, Sweden
[2] University of Gothenburg, SE-405 30 Gothenburg, Sweden

Managing complexity



- **Abstraction:** Focus first on high-level behaviour
- **Reuse:** Define small units / packages that are reusable and define interfaces (Divide & Conquer)
- **Composition:** Build larger components out of smaller ones

Time for a break

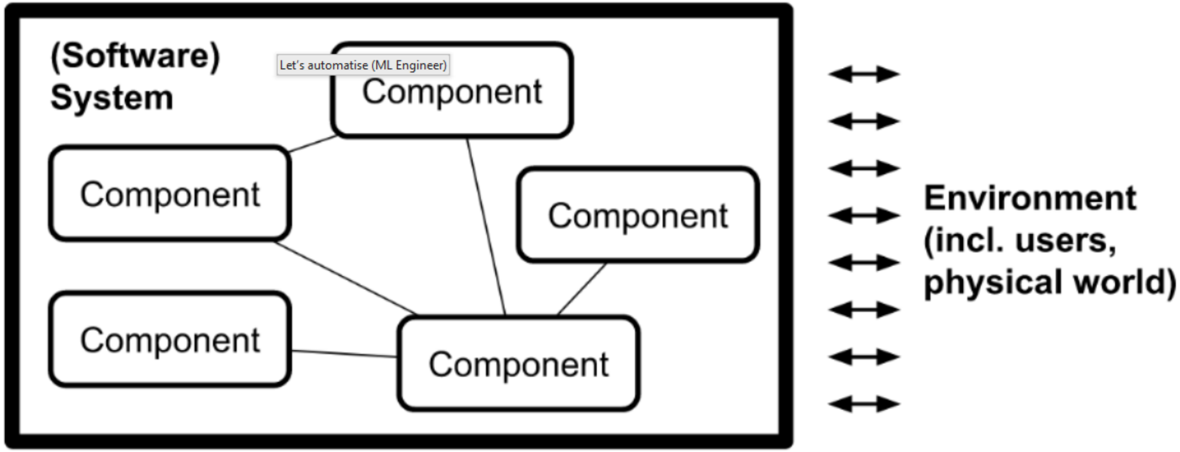
What will you learn?

Architectures and patterns for AI/ML-enabled systems

- How can we get from (prototyping) models to production systems
- Modularity and the problem of ML components in larger software systems
- An introduction to an architecture framework for distributed AI-enabled systems

- Explain how ML fits into the larger pictures of building and maintaining systems
- Explain the modularity implications
- Understand the need for architecture frameworks in AI system development

Managing complexity

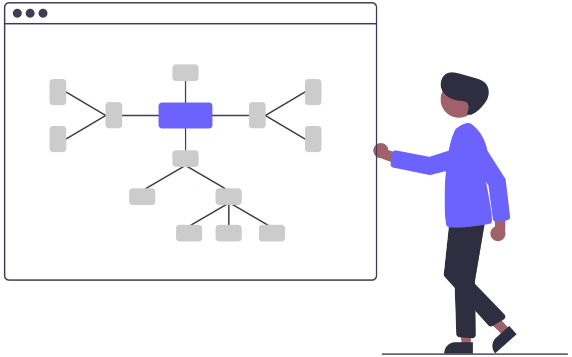


- **Abstraction:** Focus first on high-level behaviour
- **Reuse:** Define small units / packages that are reusable and define interfaces (Divide & Conquer)
- **Composition:** Build larger components out of smaller ones

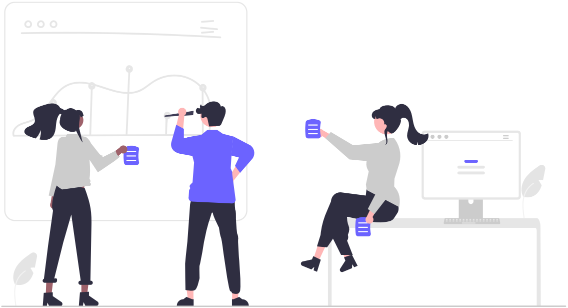
Co-Design of a system

GÖTEBORGS UNIVERSITET
CHALMERS

- Designing a complex and distributed system is a **hierarchical process of integration.**
  - Several, sometimes highly specialized views allow for decomposition of the design task.
  - Requirements and architecture often co-evolve (Twin Peaks).



- Developing complex system is a **highly collaborative** act between many stakeholders.

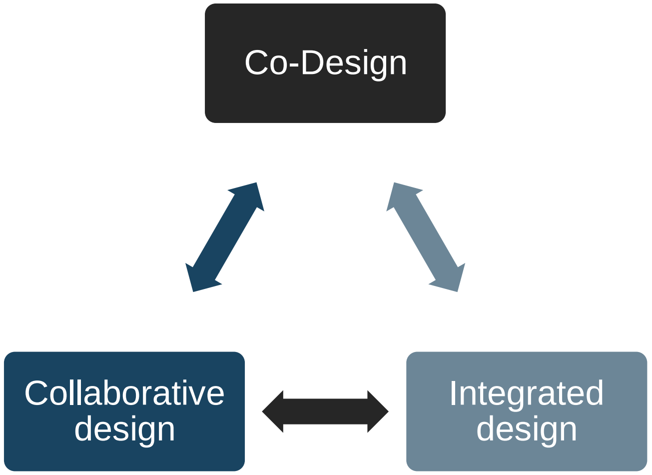


Problem Definition

Finding an architectural framework that works with AI/ML enabled systems

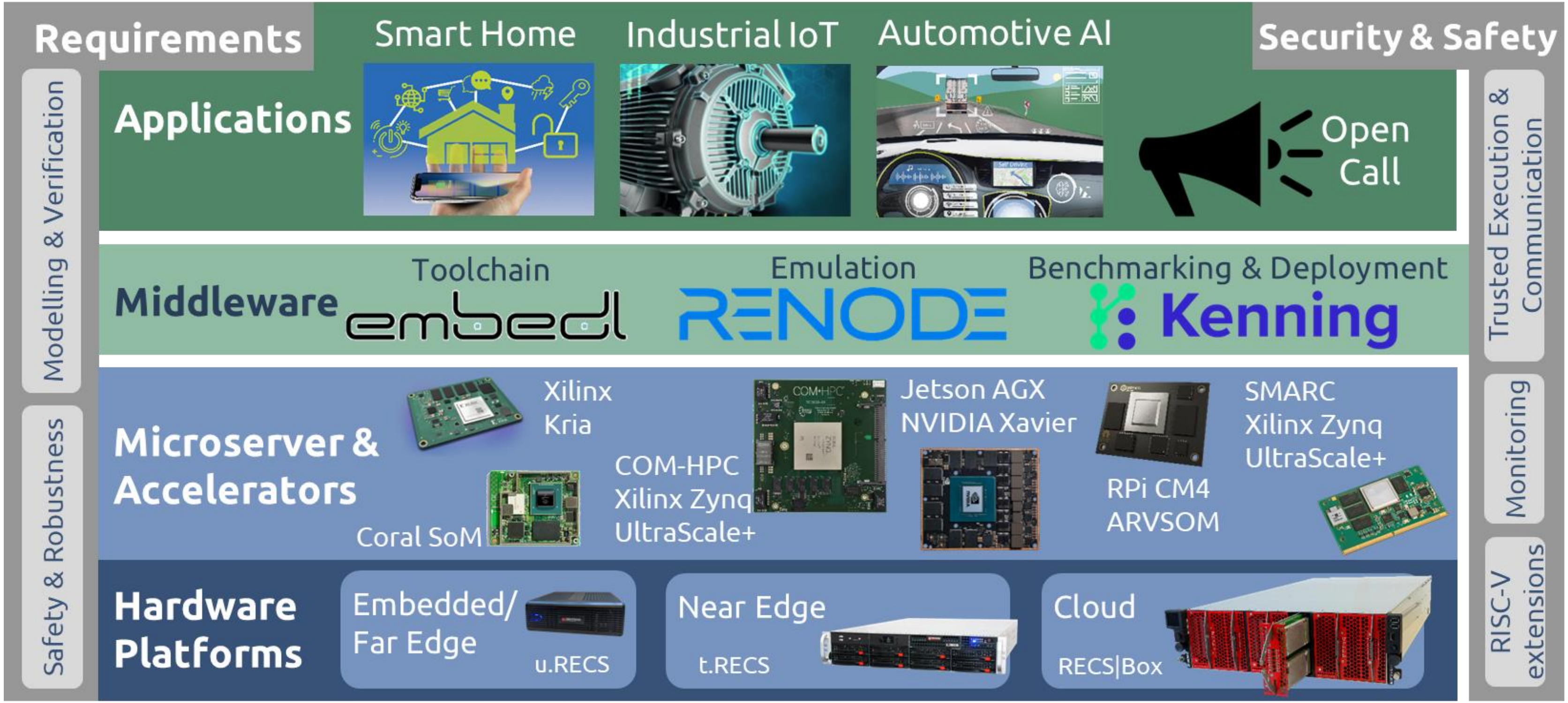
GÖTEBORGS UNIVERSITET
CHALMERS

- We needed to define an architectural framework, that supports both aspects of co-design.
- The framework must support explicitly aspects AI system development but also of other aspects such as connectivity
  - Learning and data
- A special focus lies on the support of non-functional requirements / quality views
- Traceability of design decisions



A bit of context

Building distributed systems with AI

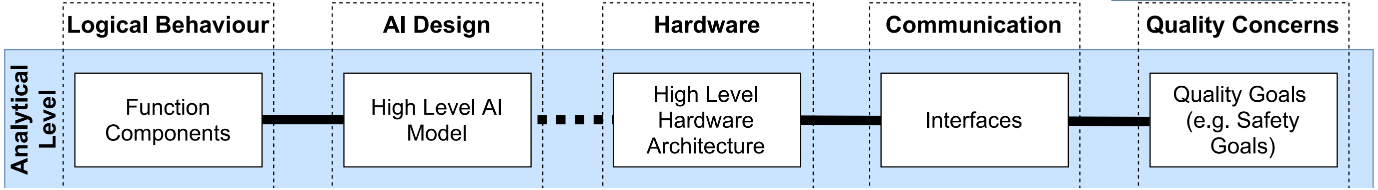


Current approaches to architecture did not help

- Providing the right learning setting / training data
  - No explicit views on the learning perspective of an AI system in common architecture approaches (Bosch et al., 2020, Muccini et al., 2021).
- Monitoring solutions must be represented explicitly in the architecture
  - Some flaws can only be detected after deployment
  - Therefore, monitoring is needed to ensure functional, and non-functional aspects of an AI system (Bernadri et al., 2019).
- New quality aspects arise, such as “explainability”, or “data privacy”
  - Depending on the use case certain a wide set of quality aspects can be relevant (Habibullah and Horkoff, 2019)
  - New stakeholders need to be included with their own views on the system (Vogelsang and Borg, 2019)

A compositional architecture framework

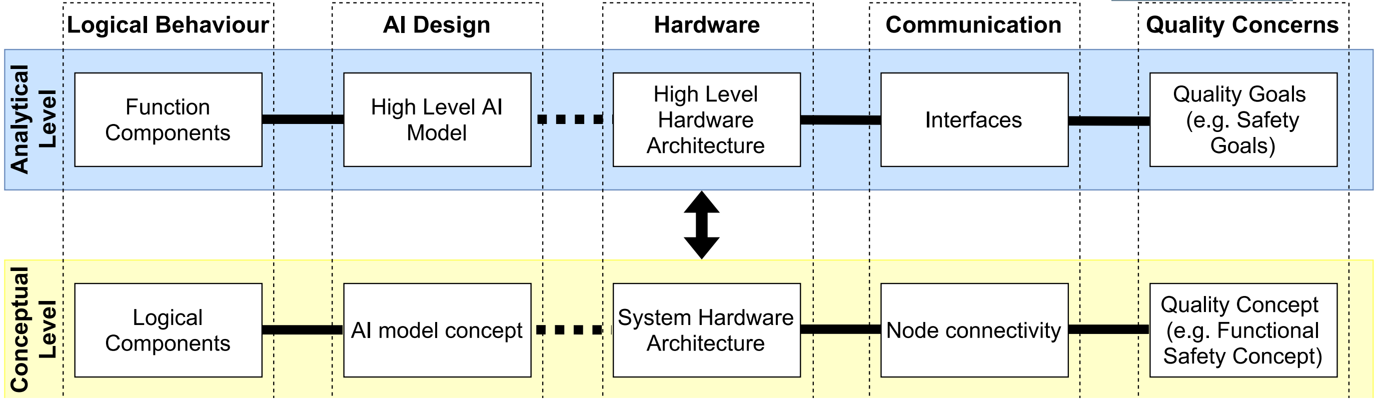
Knowledge creation (e.g. definition of safety goals).



A compositional architecture framework

Knowledge creation (e.g. definition of safety goals).

Concept design (e.g. introduction of redundancy to fulfil safety goals).

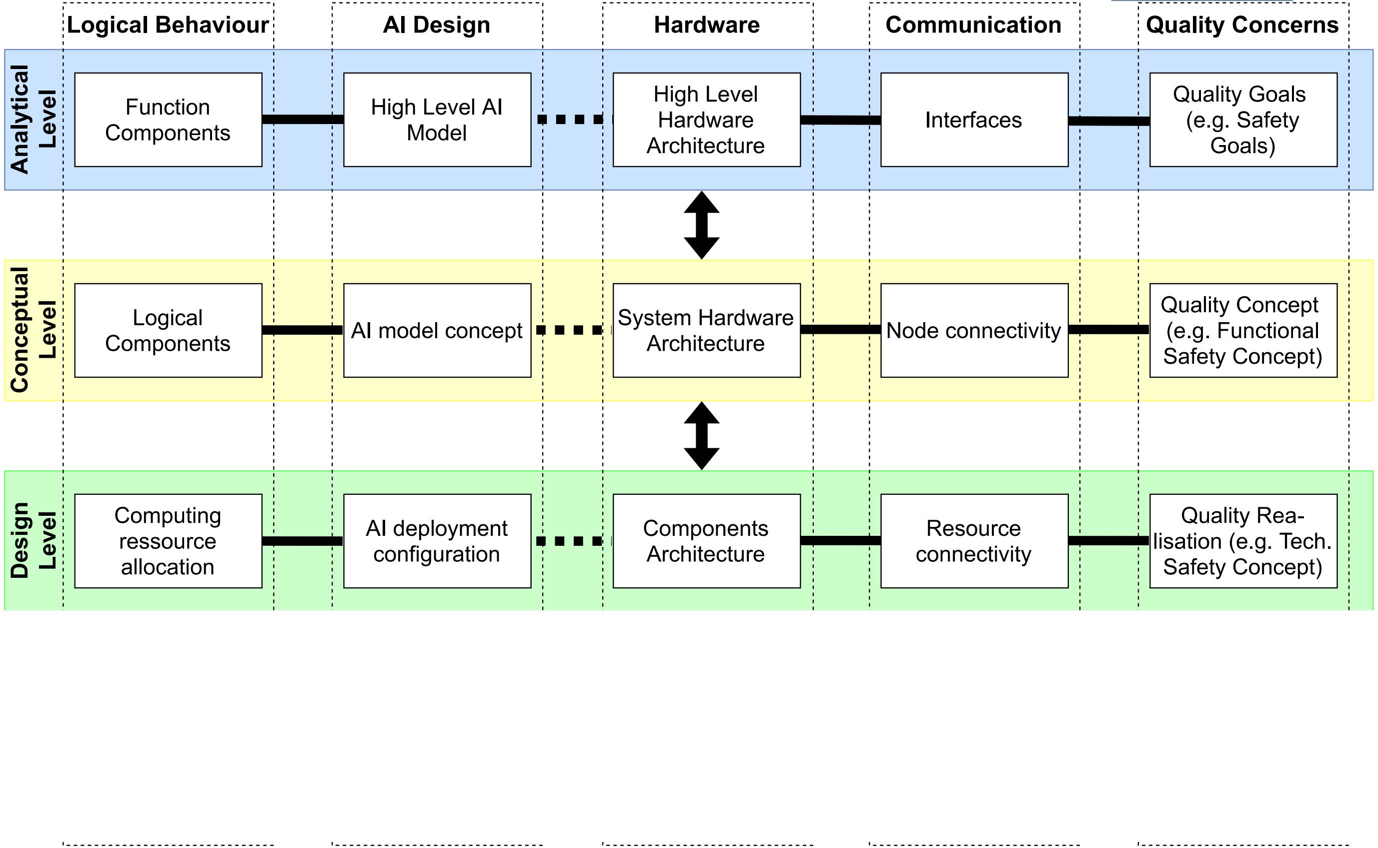


A compositional architecture framework

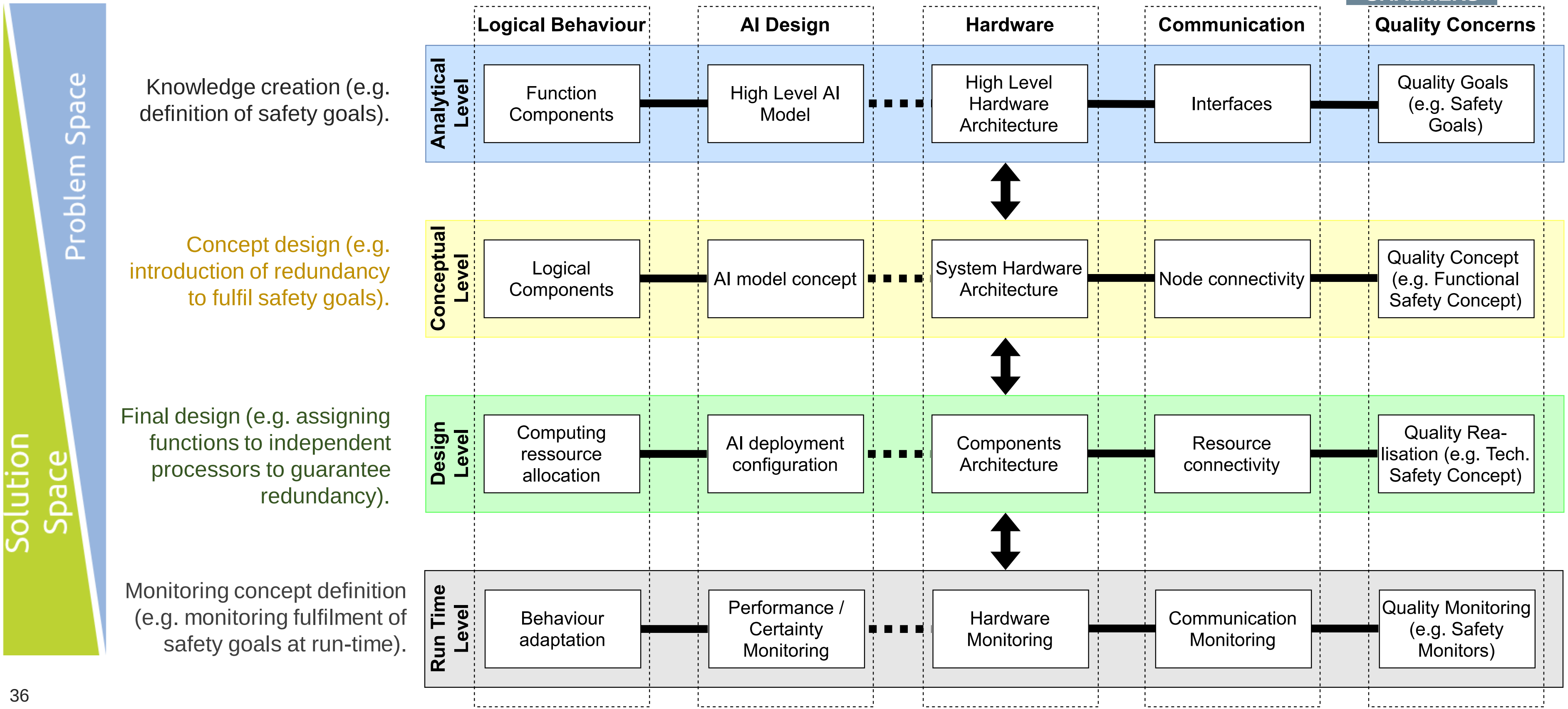
Knowledge creation (e.g. definition of safety goals).

Concept design (e.g. introduction of redundancy to fulfil safety goals).

Final design (e.g. assigning functions to independent processors to guarantee redundancy).



A compositional architecture framework



The full picture (for VEDLIoT)

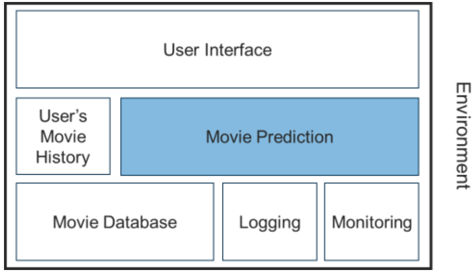
GÖTEBORGS UNIVERSITET

CHALMERS

Business Goals and Use Cases

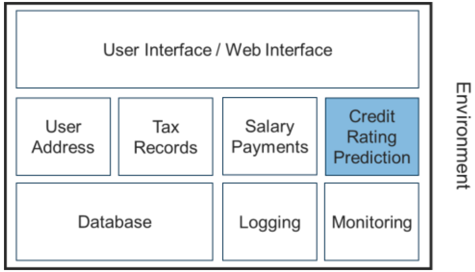
	Behaviour and Context			*Means and Resources*				*Communication*		*Quality Concerns*			
	Logical Behaviour	**Process Behaviour**	**Context & Constraints**	**Data Strategy**	**Learning**	**AI Model**	**Hardware**	**Information**	**Con-nectivity**	**Ethics**	**Privacy**	**Security**	**Safety**
Analytical Level	Function com-ponents	Interaction	Context assum-ptions	Data ingestion	Training objectives	High level AI model	High level hardware archi-tecture	Com-pilation	Interfaces	Ethic principles	Privacy impact analysis	Threat analysis (TARA)	Hazard analysis (HARA)
Conceptual Level	Logical com-ponents	Logical se-quences	Context definition	Data selection	Training concept	AI model concept	System hardware archi-tecture	Infor-mation model	Node con-nectivity	Ethic concept	Privacy concept	Cyber-security concept	Functional safety concept
Design Level	Com-puting ressource allocation	Resource se-quences	Con-straints / Design Domain	Data pre-paration / mani-pulation	Optimiser settings	AI model con-figuration	Com-ponent hardware archi-tecture	Comm-unication model	Resource con-nectivity	Ethic technical realisation	Technical solutions for privacy	Technical cyber-security concept	Technical safety concept
Run Time Level	Be-haviour monitoring	Adaptive behaviour	Context monitoring	Runtime data monitoring and collection	Manage continous improve-ments	AI model per-formance monitoring	Hardware per-formance monitoring	Data monitoring	Con-nectivity monitoring	Assess-ment / auditing of AI decisions	Assess-ment of privacy com-pliance	Security monitoring / threat response	Safety monitoring / safety de-gradition

3 Groups: Cluster of Concerns



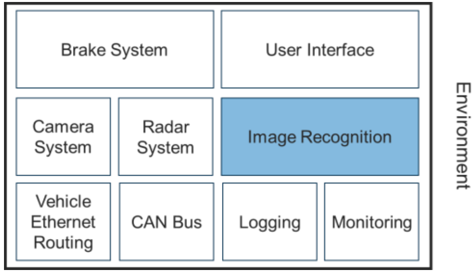
Movie Prediction

- Cluster 1
- Cluster 2
- …



Credit Rating Prediction

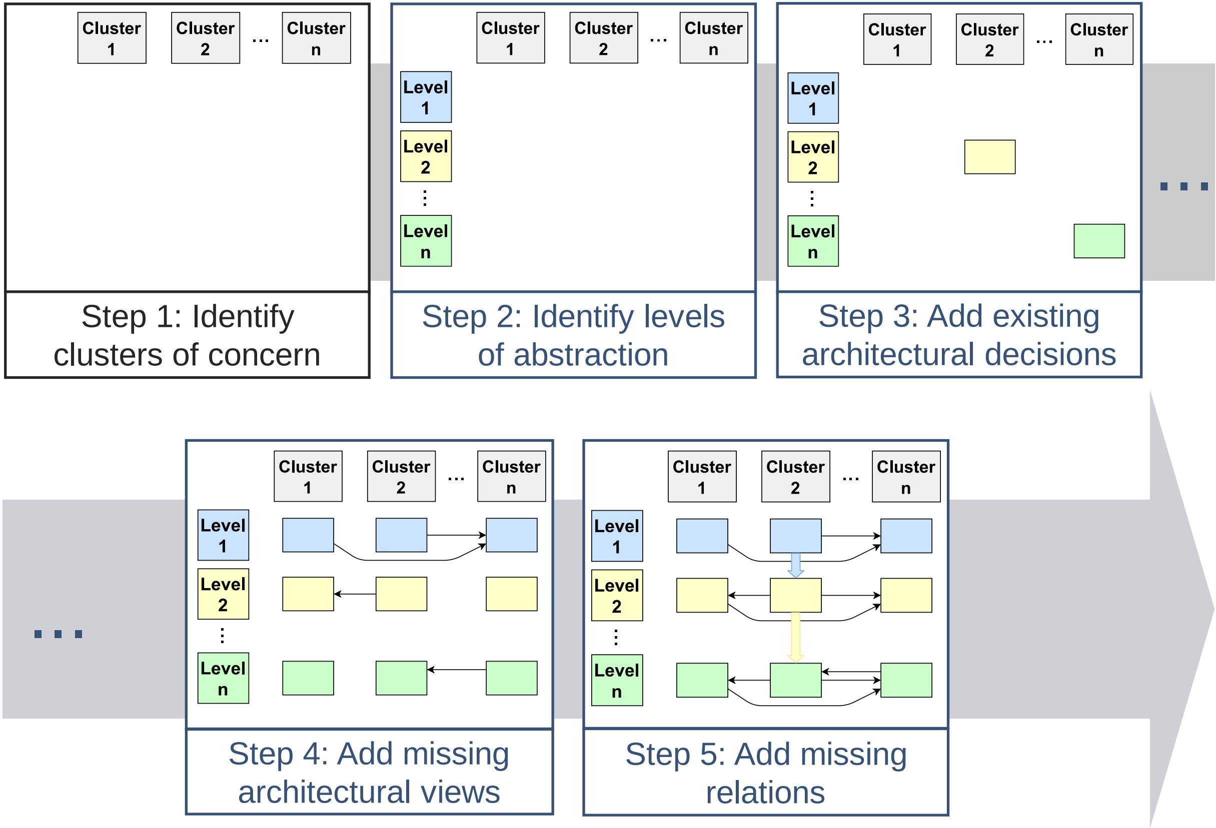
- Cluster 1
- Cluster 2
- …



Automatic Emergency Braking

- Cluster 1
- Cluster 2
- …

How to compositional architecture framework

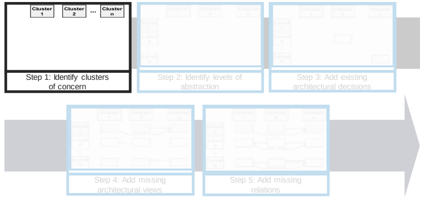


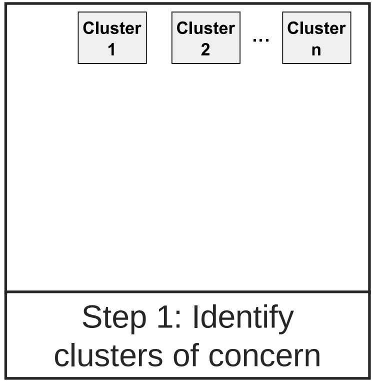
- Step 1: Identify clusters of concern
- Step 2: Identify levels of abstraction
- Step 3: Add existing architectural decisions.
- Step 4: Add missing architectural views.
- Step 5: Add missing relations.
- Step 6: Iterate if needed.

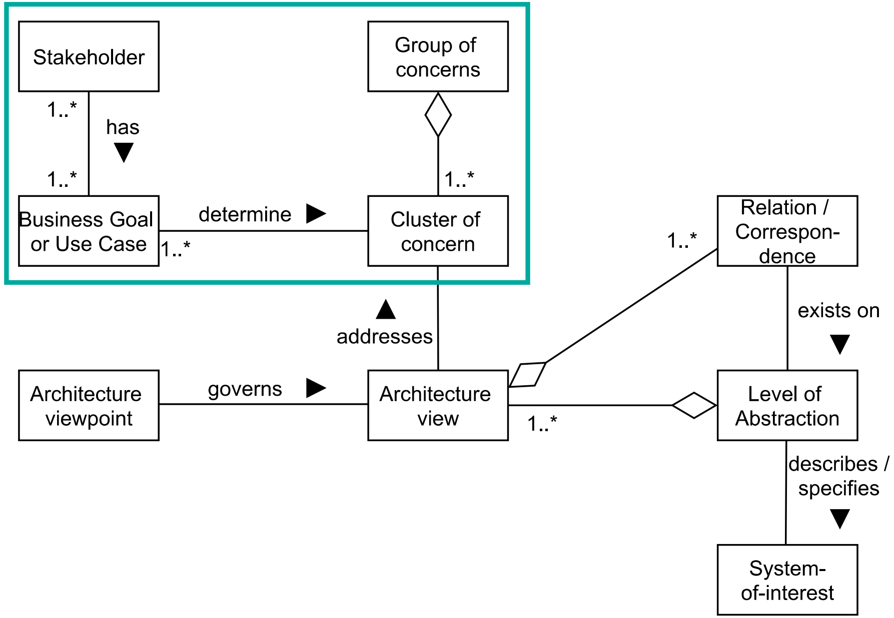
GÖTEBORGS UNIVERSITET
CHALMERS

How to compositional architecture framework

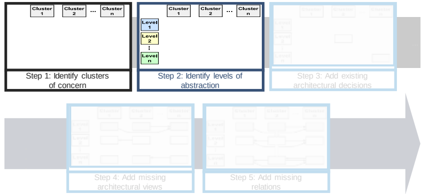
GÖTEBORGS UNIVERSITET
CHALMERS

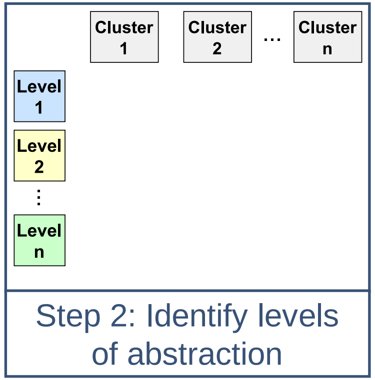


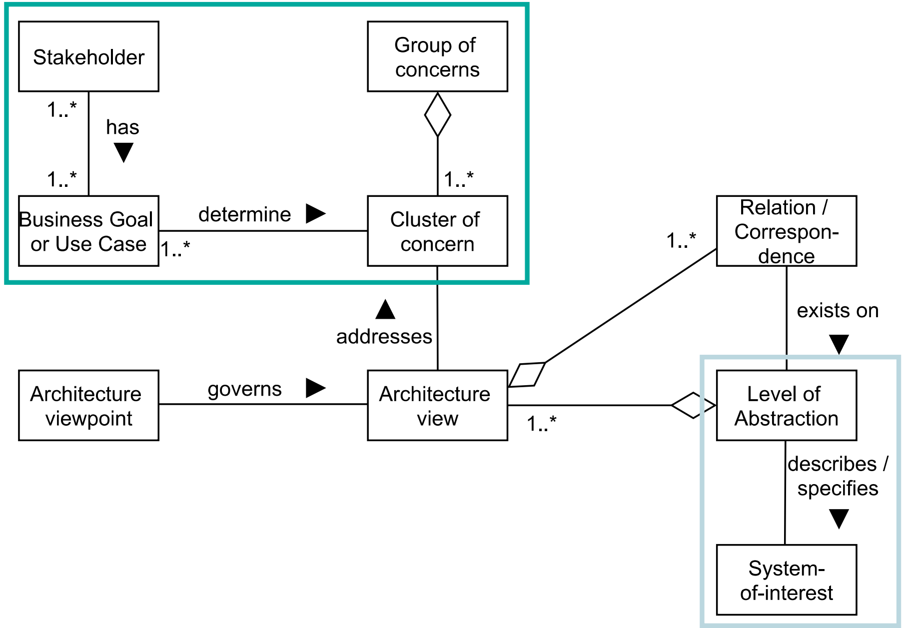




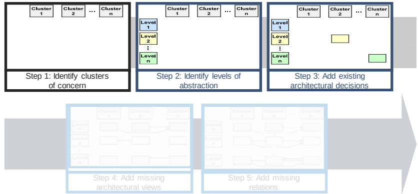
How to compositional architecture framework

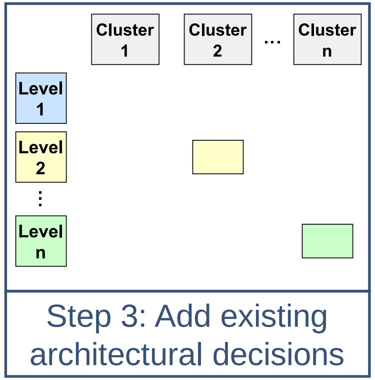


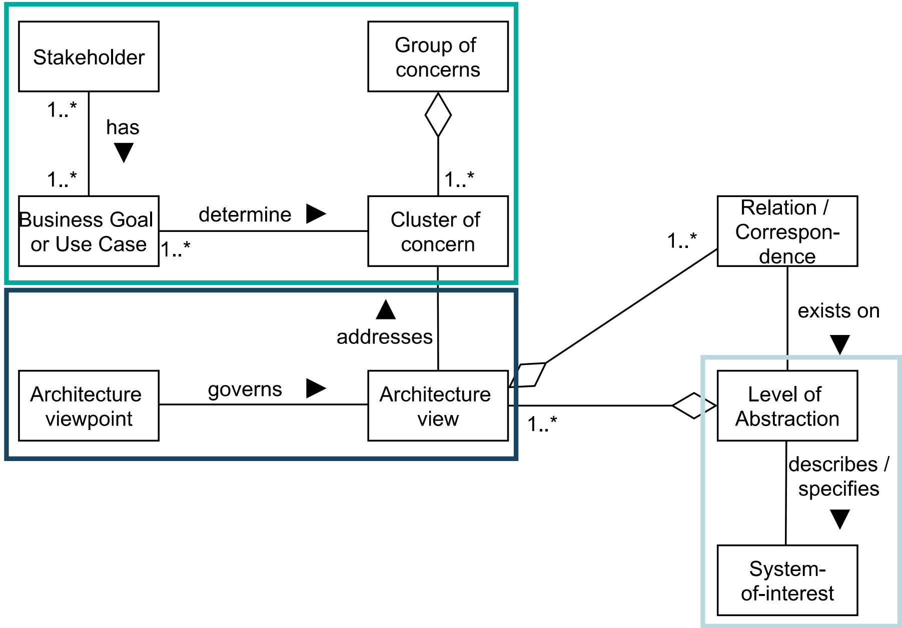




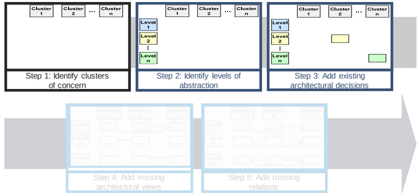
How to compositional architecture framework

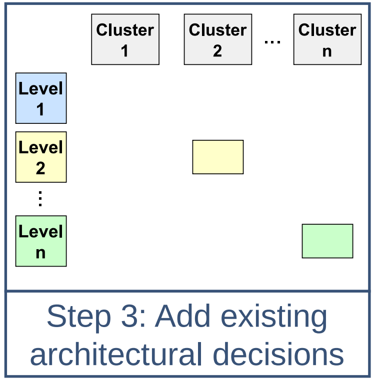


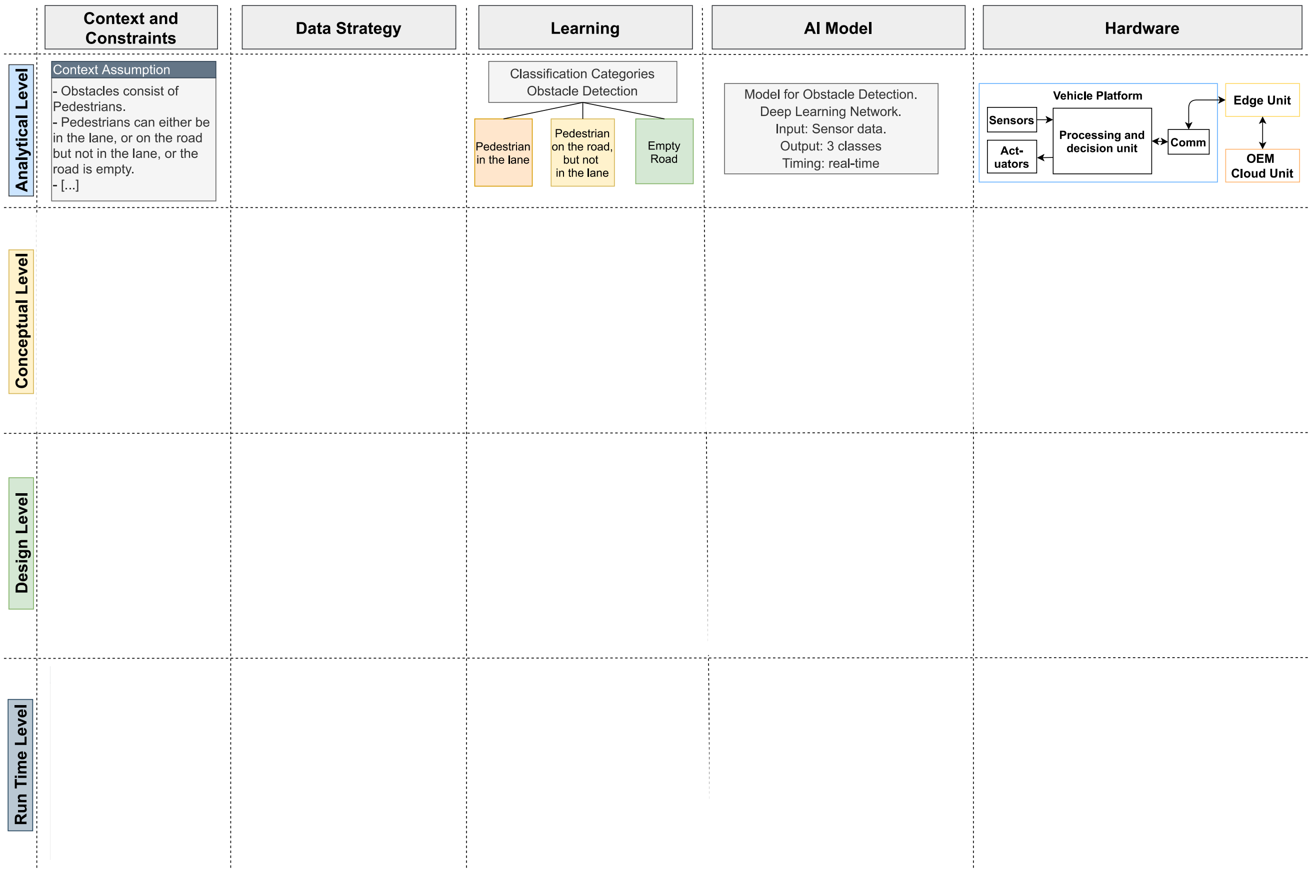




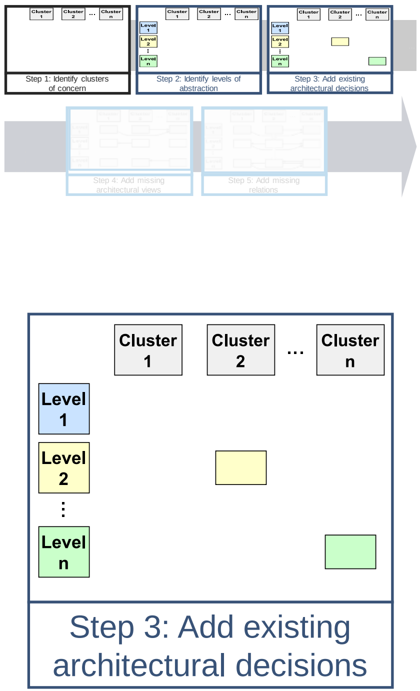
How to compositional architecture framework

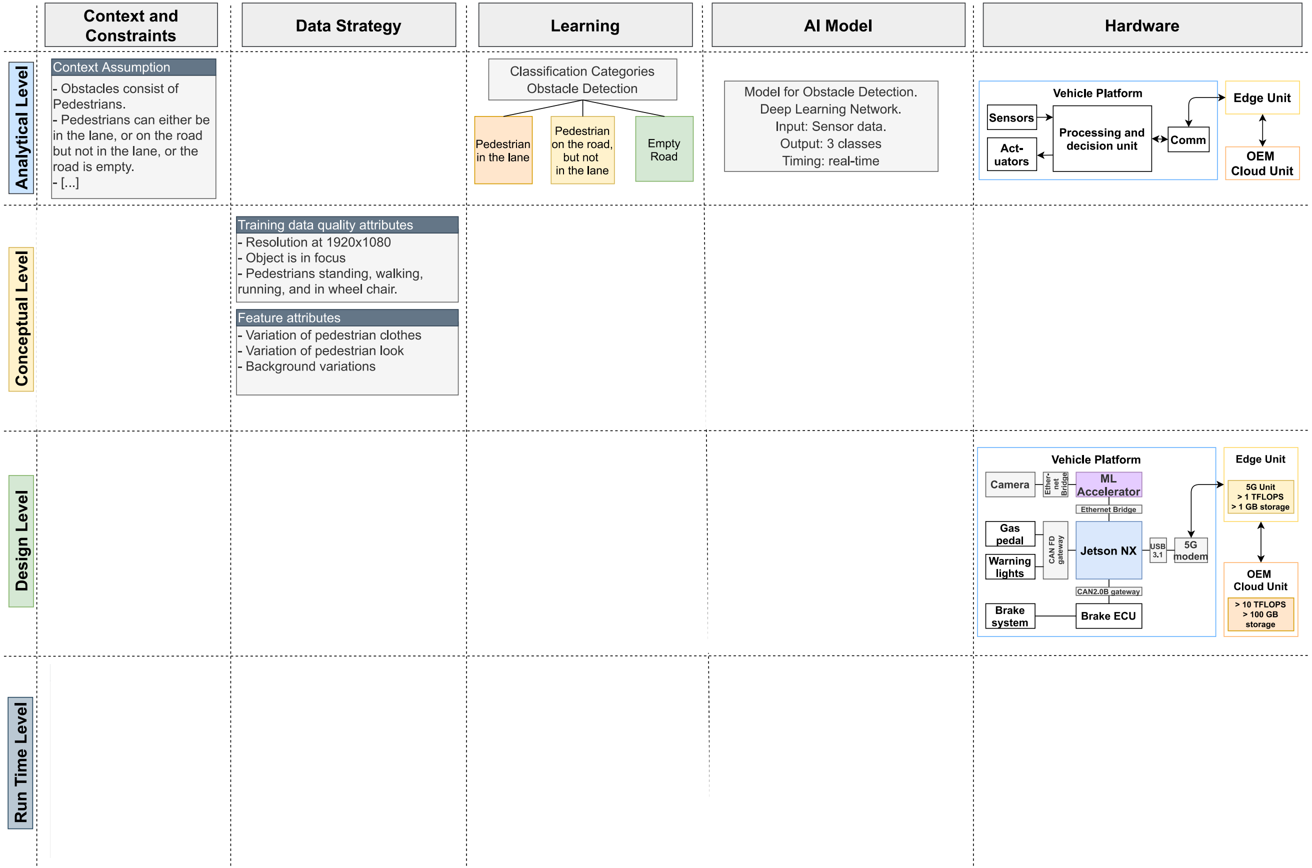




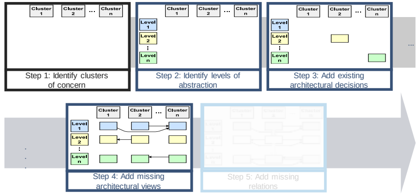


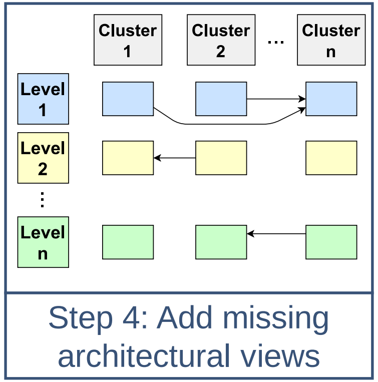
How to compositional architecture framework

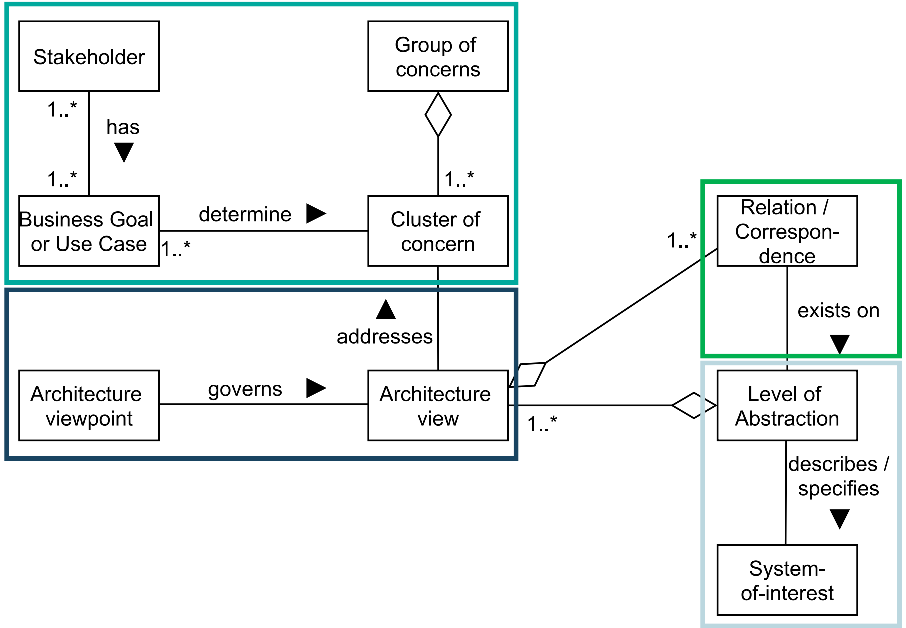




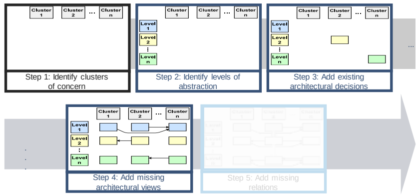
How to compositional architecture framework

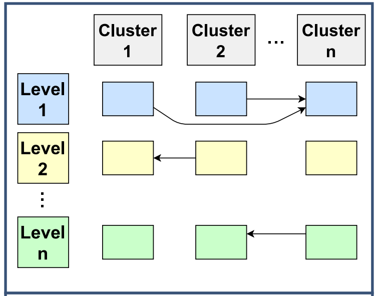




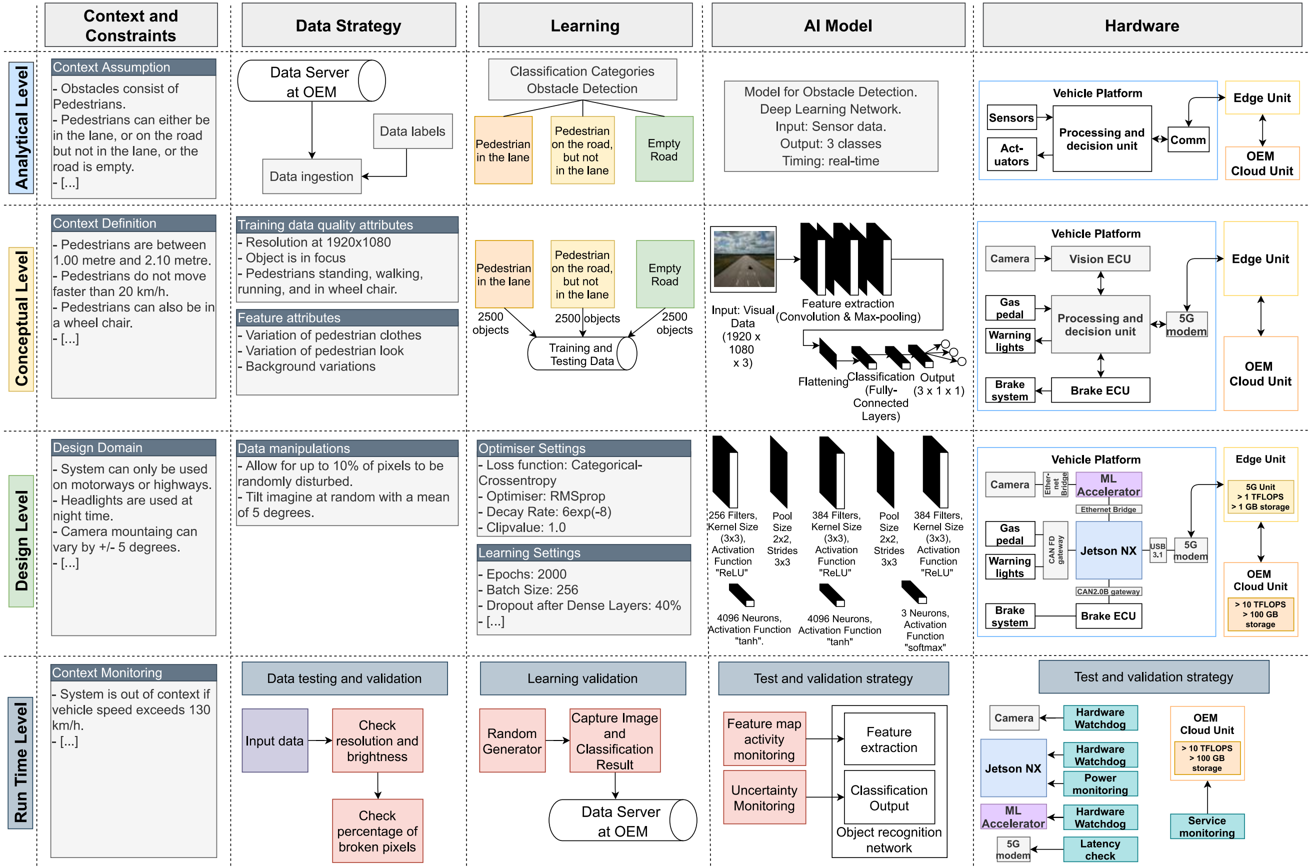


How to compositional architecture framework

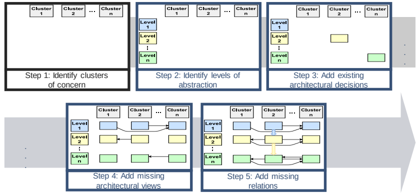


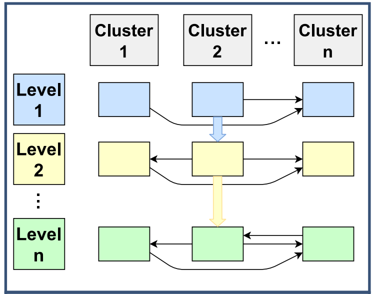


Step 4: Add missing architectural views

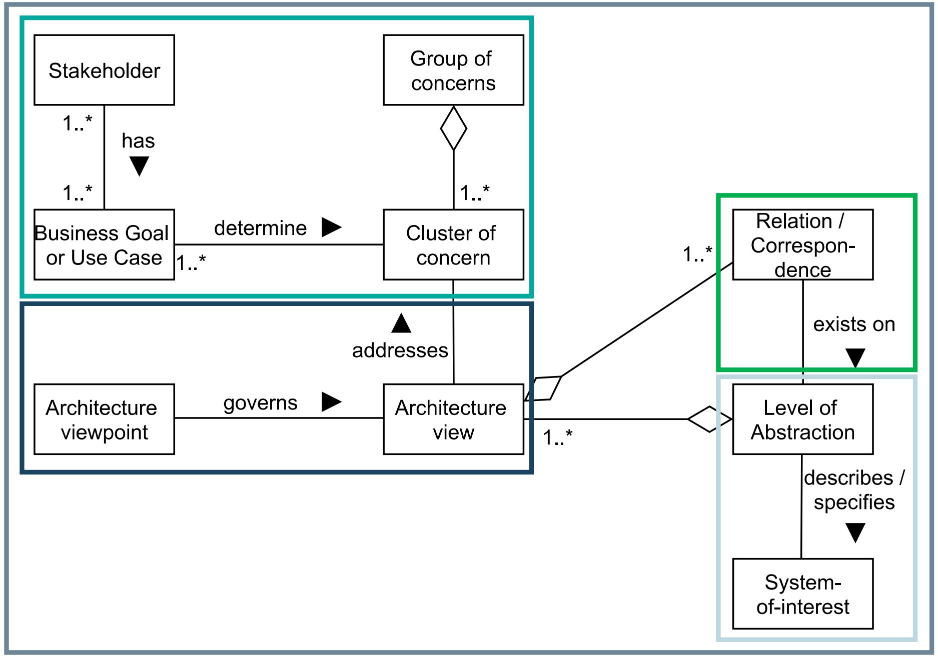


How to compositional architecture framework

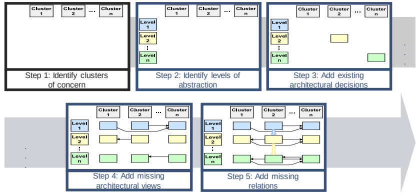


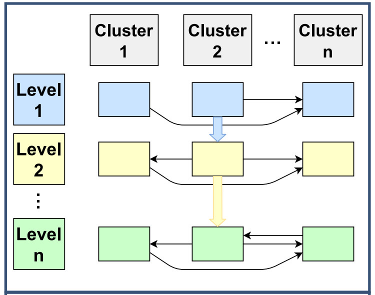


Step 5: Add missing relations

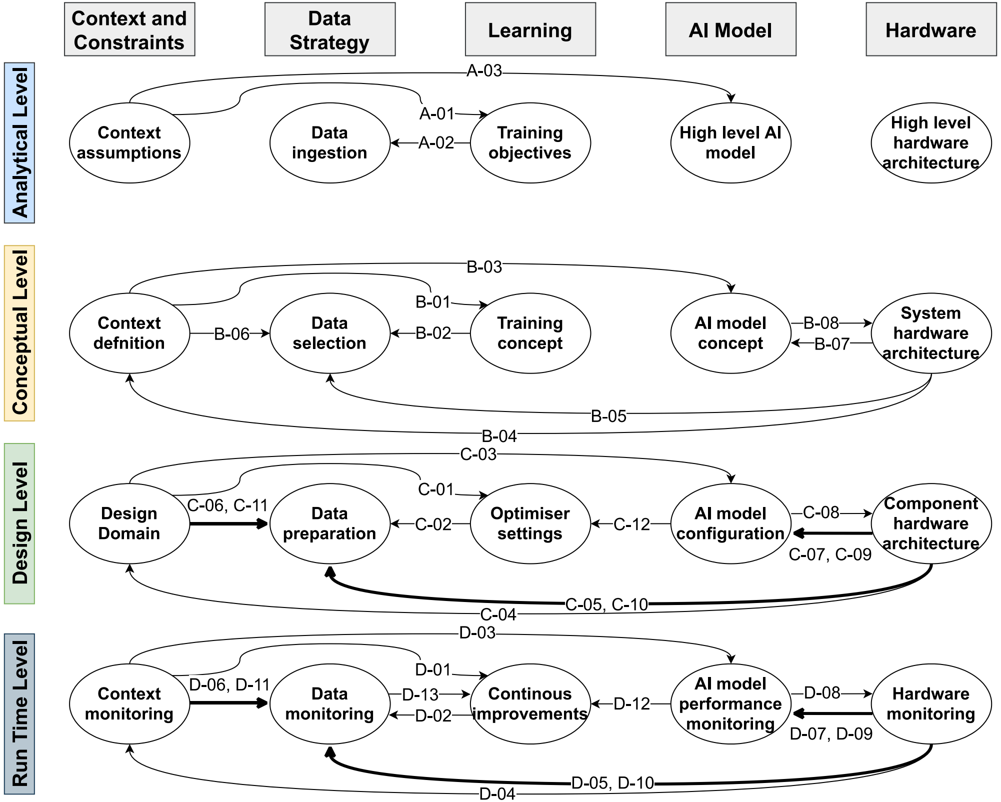


How to compositional architecture framework



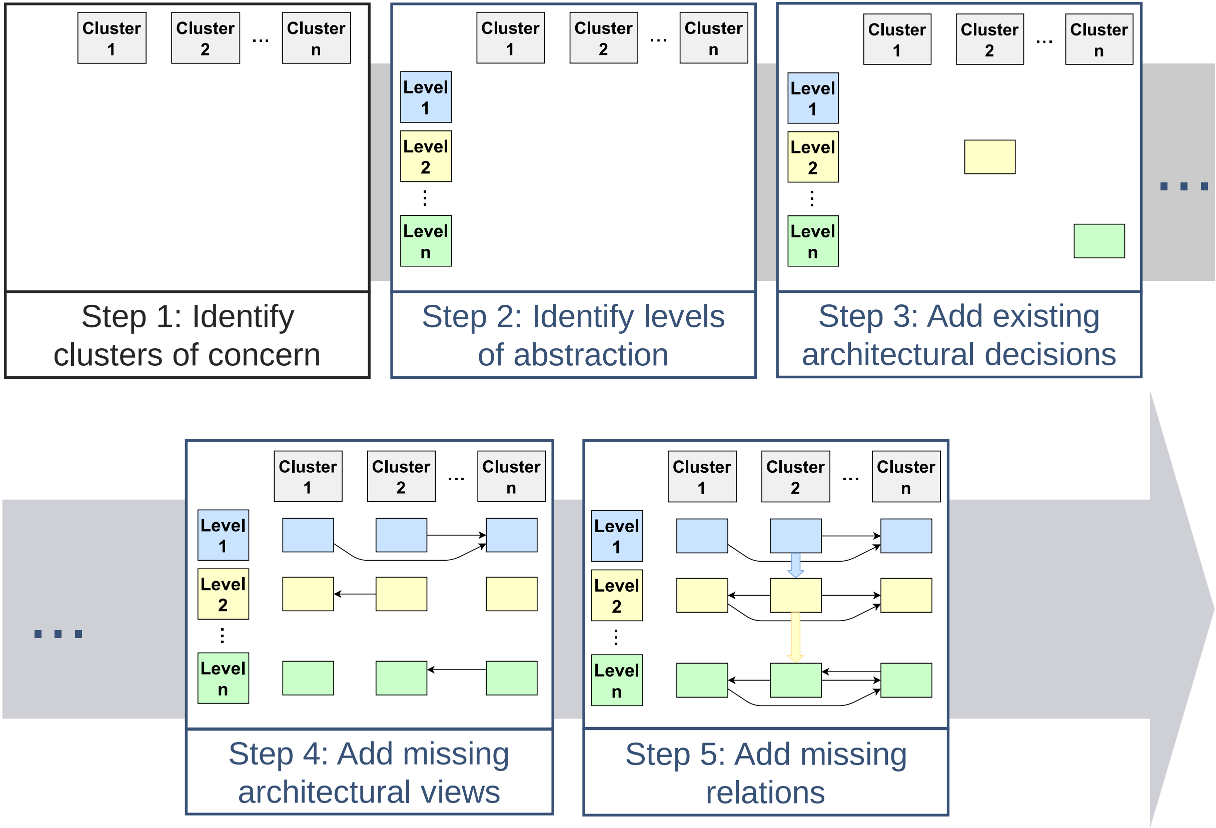


Step 5: Add missing relations



How to compositional architecture framework

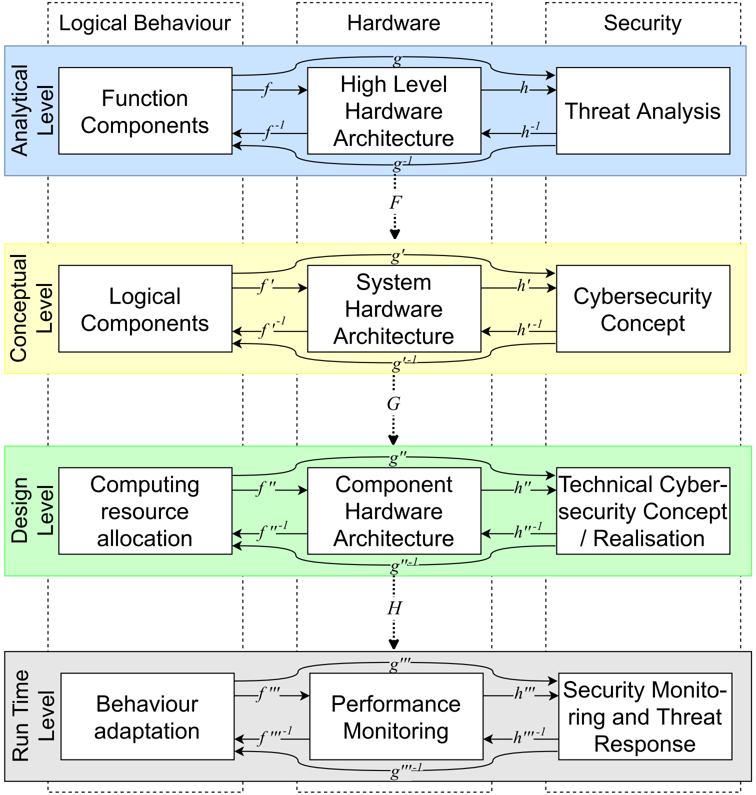
GÖTEBORGS UNIVERSITET
CHALMERS



- Step 1: Identify clusters of concern
- Step 2: Identify levels of abstraction
- Step 3: Add existing architectural decisions.
- Step 4: Add missing architectural views.
- Step 5: Add missing relations.
- Step 6: Iterate if needed.

A compositional approach to architecture framework

GÖTEBORGS UNIVERSITET
CHALMERS



Rule 1: Clusters of concern shall contain architectural views with different levels of details of a certain aspect of the VEDLIoT system.

Rule 2: Architectural views shall be sorted into levels of abstractions, according to their level of details about the VEDLIoT system.

Rule 3: By using correspondence rules, it shall be possible to arrive at different architectural views of the VEDLIoT system without encountering inconsistencies.

Rule 4: Architectural views, and relations between them, shall be mapped to the next lower level of abstraction.

Heyn, H. M., Knauss E., & Pelliccione P. (2023). A Compositional Approach to Architecture Frameworks for distributed AI Systems. *In Elsevier Journal of Systems and Software (JSS).*

Why?

- The architectural framework helps connecting different aspects of a larger system together.
- It allows for "middle-out" development, i.e., existing design decisions are explicitly considered.
- It allows to keep an overview over the necessary quality aspects, such as safety, security, ethical, or privacy aspects of the systems.
- The framework enforces a runtime concept for the system.
- The traceability of design decisions allows for compliance with upcoming AI regulations.

GÖTEBORGS UNIVERSITET
CHALMERS

EUROPEAN COMMISSION

Proposal for a

REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS

{SEC(2021) 167 final} - {SWD(2021) 84 final} - {SWD(2021) 85 final}

Some reading recommendations

Bosch, J., Olsson, H. H., & Crnkovic, I. (2021). Engineering ai systems: A research agenda. In Artificial Intelligence Paradigms for Smart Cyber-Physical Systems (pp. 1-19). IGI global.

Bernardi, L., Mavridis, T., & Estevez, P. (2019). 150 successful machine learning models: 6 lessons learned at booking. com. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 1743-1751).

Paullada, A., Raji, I. D., Bender, E. M., Denton, E., & Hanna, A. (2021). Data and its (dis) contents: A survey of dataset development and use in machine learning research. Patterns, 2(11)

Heyn, H. M., Knauss, E., & Pelliccione, P. (2023). A compositional approach to creating architecture frameworks with an application to distributed AI systems. Journal of Systems and Software, 111604.

Jackson, M. (1995). The world and the machine. In Proceedings of the 17th international conference on Software engineering (pp. 283-292).

Hulten, G. (2019). Building Intelligent Systems. Berkeley, CA: Apress., Chapter 5 and Chapter 7

Some reading recommendations

GÖTEBORGS UNIVERSITET
CHALMERS

Bosch, J., Olsson, H. H., & Crnkovic, I. (2021). Engineering ai systems: A research agenda. In Artificial Intelligence Paradigms for Smart Cyber-Physical Systems (pp. 1-19). IGI global.

Bernardi, L., Mavridis, T., & Estevez, P. (2019). 150 successful machine learning models: 6 lessons learned at booking. com. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 1743-1751).

Paullada, A., Raji, I. D., Bender, E. M., Denton, E., & Hanna, A. (2021). Data and its (dis) contents: A survey of dataset development and use in machine learning research. Patterns, 2(11)

Heyn, H. M., Knauss, E., & Pelliccione, P. (2023). A compositional approach to creating architecture frameworks with an application to distributed AI systems. Journal of Systems and Software, 111604.

Jackson, M. (1995). The world and the machine. In Proceedings of the 17th international conference on Software engineering (pp. 283-292).

Hulten, G. (2019). Building Intelligent Systems. Berkeley, CA: Apress., Chapter 5 and Chapter 7

More literature (if you are interested)

Muccini, H., & Vaidhyanathan, K. (2021, May). Software architecture for ml-based systems: what exists and what lies ahead. In 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN) (pp. 121-128). IEEE.

Habibullah, K. M., & Horkoff, J. (2021, September). Non-functional requirements for machine learning: understanding current use and challenges in industry. In 2021 IEEE 29th International Requirements Engineering Conference (RE) (pp. 13-23). IEEE.

Vogelsang, A., & Borg, M. (2019, September). Requirements engineering for machine learning: Perspectives from data scientists. In 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW) (pp. 245-251). IEEE.

What did you learn?

Architectures and patterns for AI/ML-enabled systems

GÖTEBORGS UNIVERSITET
CHALMERS

- How can we get from (prototyping) models to production systems
- Modularity and the problem of ML components in larger software systems
- An introduction to an architecture framework for distributed AI-enabled systems

- Explain how ML fits into the larger pictures of building and maintaining systems
- Explain the modularity implications
- Understand the need for architecture frameworks in AI system development

In the next lecture…

- Example of how the compositional architecture framework can be applied.
- Responsible Software Engineering for / with AI/ML-enabled systems

GÖTEBORGS UNIVERSITET

CHALMERS