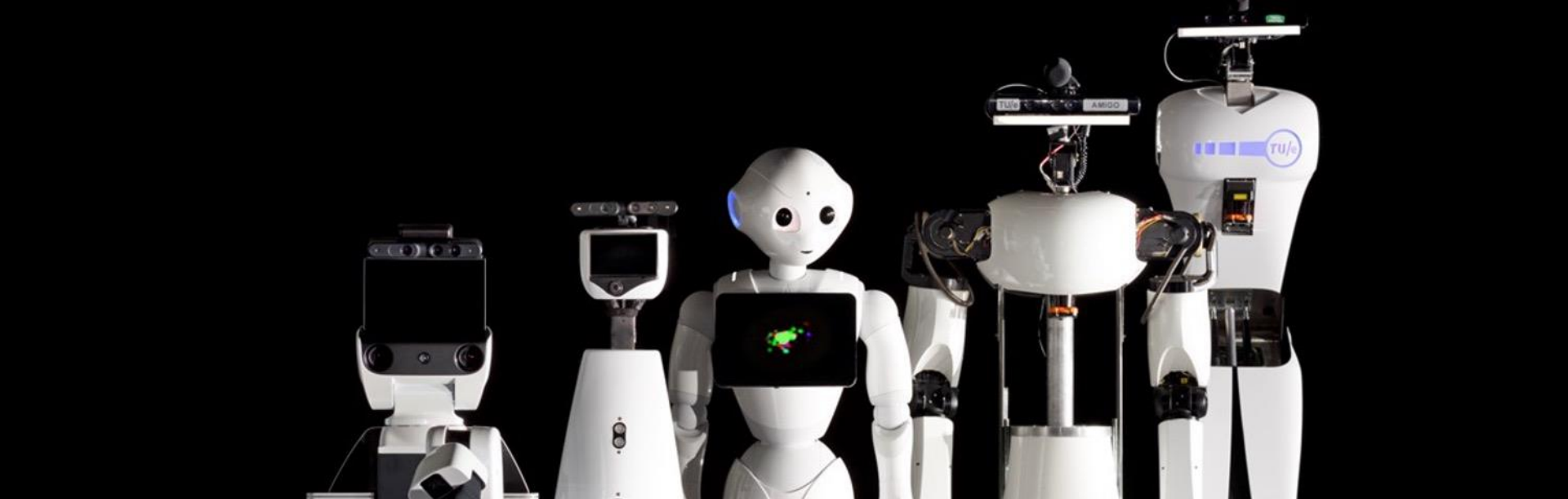


4SC020 Mobile Robot Control 2024: System Architecture

29TH OF MAY 2024

Jordy Senden
j.p.f.senden@tue.nl

Mechanical Engineering, Robotics

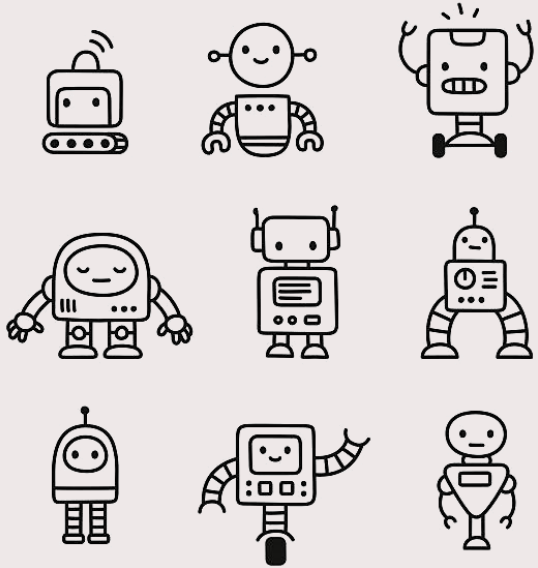
TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Recap

- Introduction
- Best practices for C++ and Git

- Local Navigation
- Global Navigation
- Localization

- System Architecture



Recap

- **Introduction**
- **Best practices for C++ and Git**

- Local Navigation
- Global Navigation
- Localization

- System Architecture

Recap

- Introduction
- Best practices for C++ and Git

- **Local Navigation**
- **Global Navigation**
- **Localization**

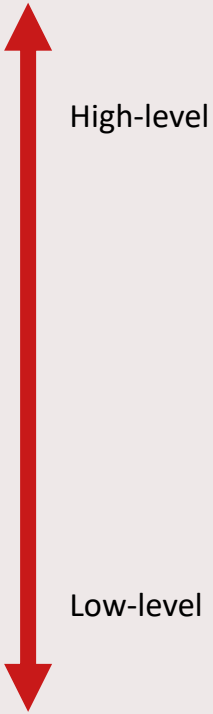
- System Architecture

Recap

- Introduction
- Best practices for C++ and Git

- Local Navigation
- Global Navigation
- Localization

- **System Architecture**



What is this lecture about?

Main question:

How to design a system that is able to solve a given task (*as a group*)?

Give you handles/input for the design presentation

What is this lecture about?

What do we mean with high-level system design?

Analogy with writing a report as a team:

What is this lecture about?

What do we mean with high-level?

Analogy with writing a report as a team:

Content:

What is the structure?

Which part contains what information?

Introduction

Problem definition
(Introduce terms)

Approach

Tests

Results

Conclusion

What is this lecture about?

What do we mean with high-level?

Analogy with writing a report as a team:

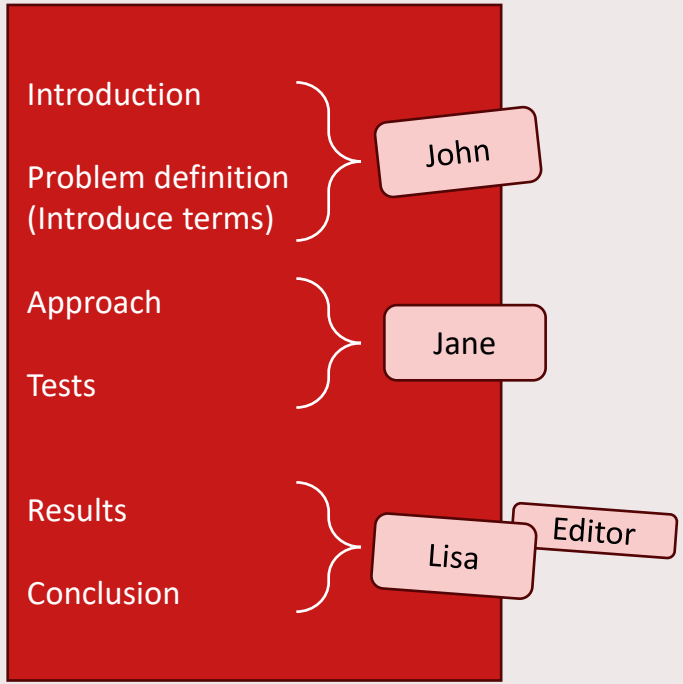
Content:

What is the structure?

Which part contains what information?

Process:

Who writes what part?



What is this lecture about?

What do we mean with high-level?

Analogy with writing a report as a team:

Content:

What is the structure?

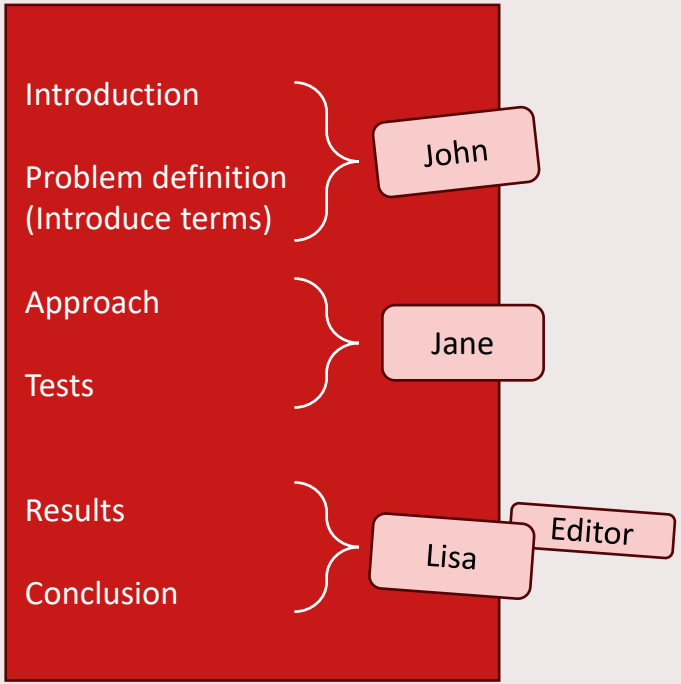
Which part contains what information?

Process:

Who writes what part?

Form:

Verbe tenses and aspects
(Consistency throughout)



Content

Introduction SA and SE

System Design process

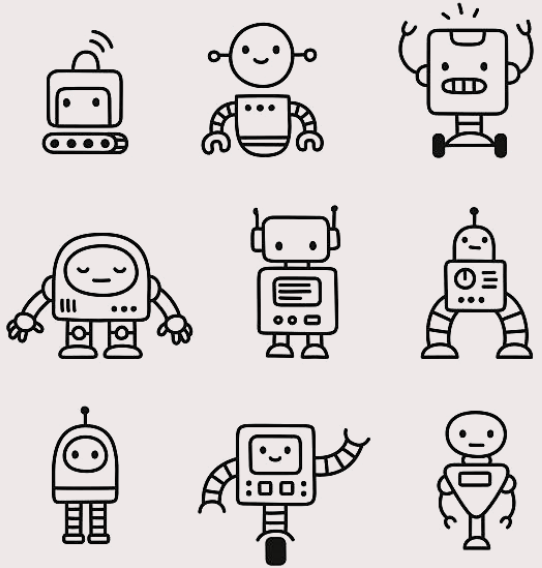
- V-model
- Requirements Engineering

Software Design

- Data Flow Diagram
- State Flow Diagram

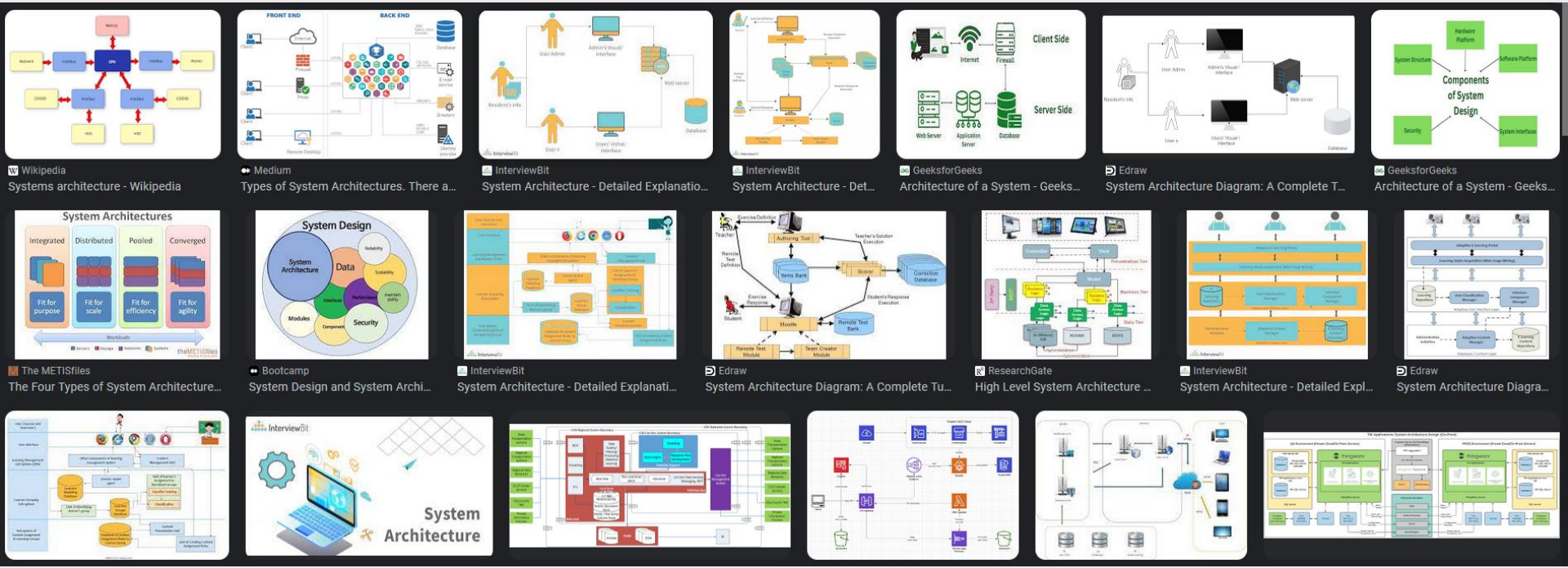
Discussion: examples previous years

Take-home



System Architecture

System architecture

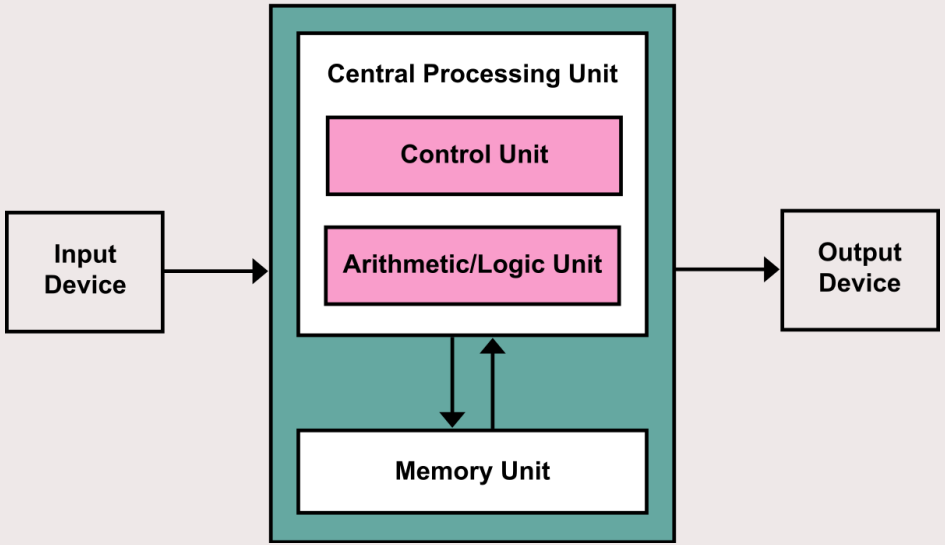


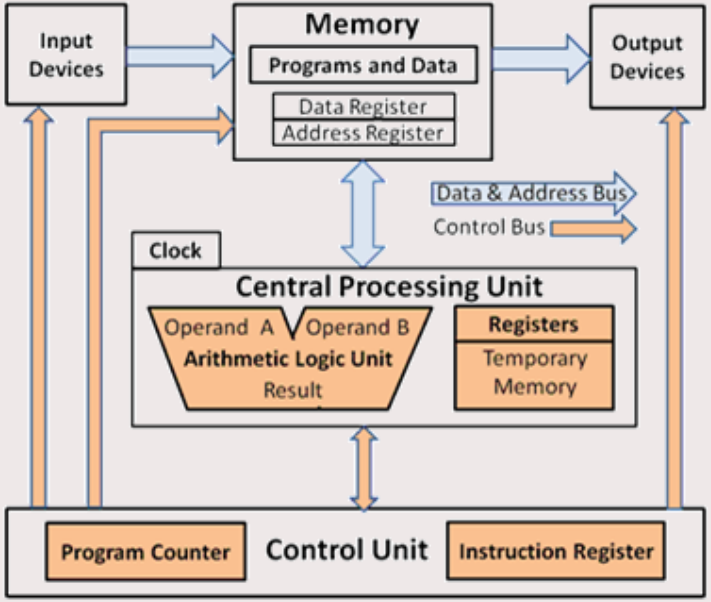
System architecture

Set of schematic drawings (models) that explain the structure and behavior
(A good model is worth a 1000 words)

Wiki:
A **system architecture** is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.
A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system.

Von Neumann Architecture – 2 views





Systems Engineering

(or Systems Architecting)

Systems engineering

Systematic process of creating a system (architecture)

Wiki:
Systems engineering is an interdisciplinary field of engineering and engineering management that focuses on how to design, integrate, and manage complex systems over their life cycles. At its core, systems engineering utilizes systems thinking principles to organize this body of knowledge. The individual outcome of such efforts, an **engineered system**, can be defined as a combination of components that work in synergy to collectively perform a useful function.

Questions to answer

RED = related to the customer/task
GREEN = related to the system
BLUE = related to group work

What is the task?
Who are the stakeholders?
What are the requirements imposed by these stakeholders?
…

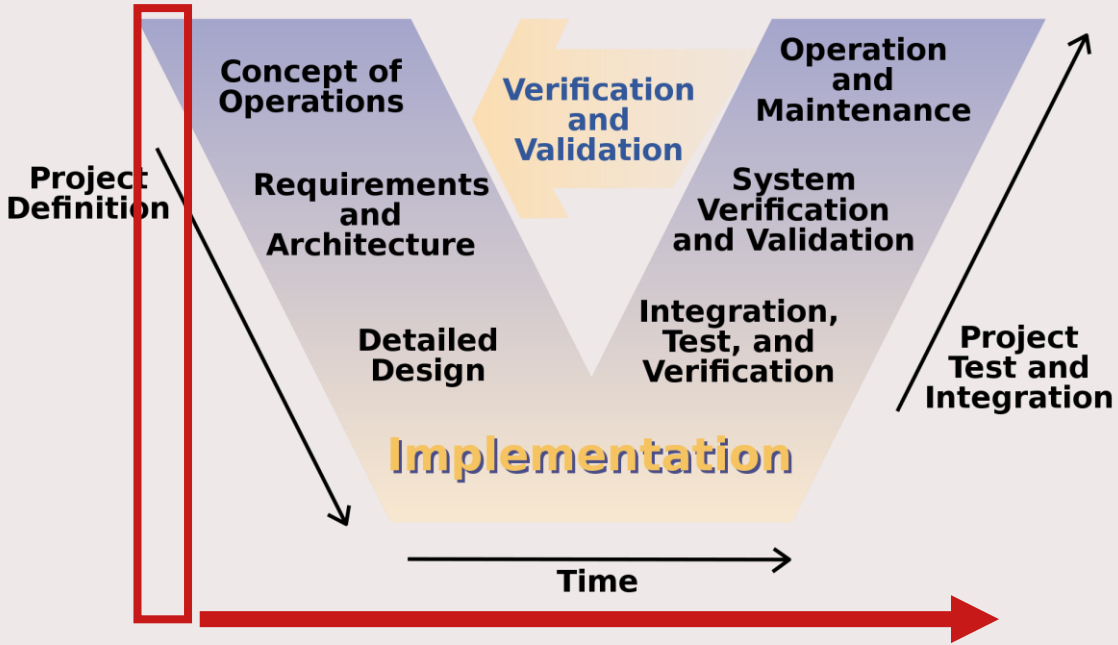
What system do/can we use to solve this task?
How do we (does the system) make sure to meet the stakeholder-requirements?
How does the characteristics of the system influence the system design/implementation.
…

How do we work together as a group?
How can we split up the tasks?
How do we integrate our separate pieces of code?
…

System Design

V-model

A model for software- and system design process



Requirements

What **should** the system do?

- Speed limits
- Wall clearance
- Driving lanes
- Driving heading
- …

- Design decisions
- Strict limitations
- What should be the value?

- What if people are in the way?
- What if a door is blocked?
- How long to ‘idle’?
- …

- Coming from stakeholder
- Often vague statements:
  “Safe”, “Robust”, “Easy-to-use”
- How to measure these?

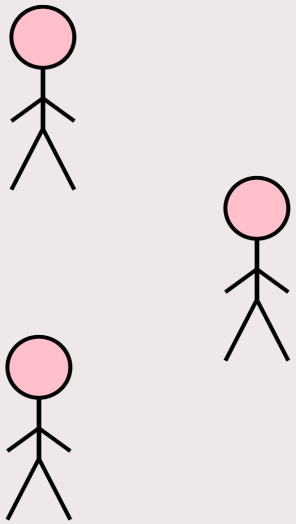
Specifications

What **can** the system do?

- 1 task, 2 systems
- Simularities and differences
- How does this effect your software?

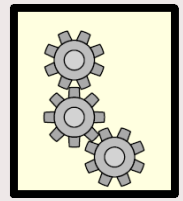
How to get from Desires to Specs?

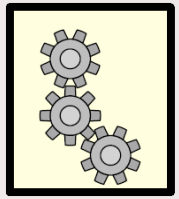
Stakeholders



The art of
Systems Engineering

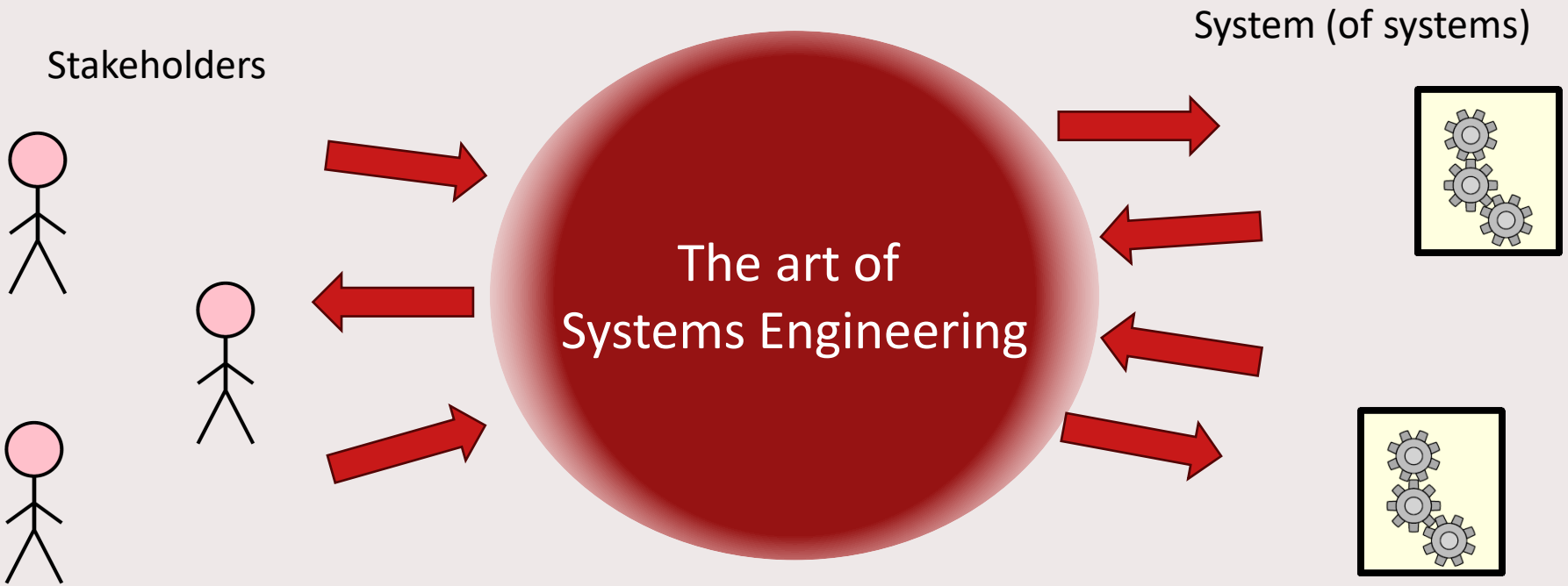
System (of systems)



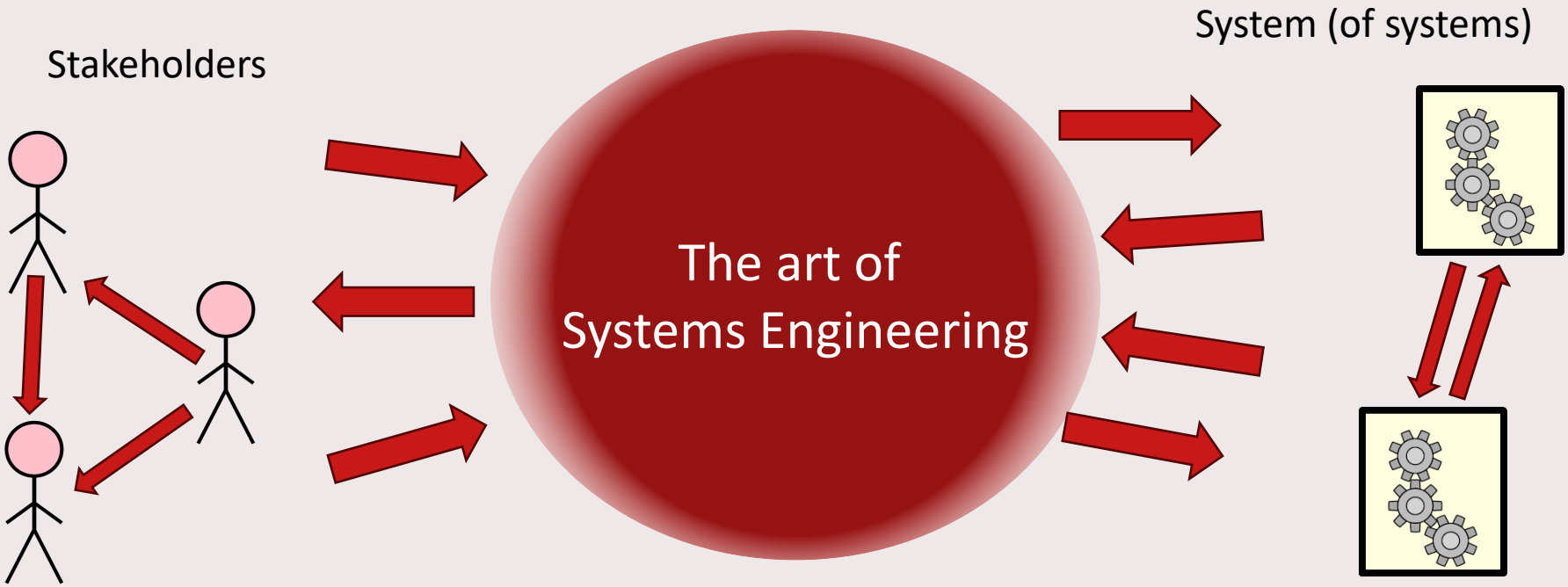


TU/e

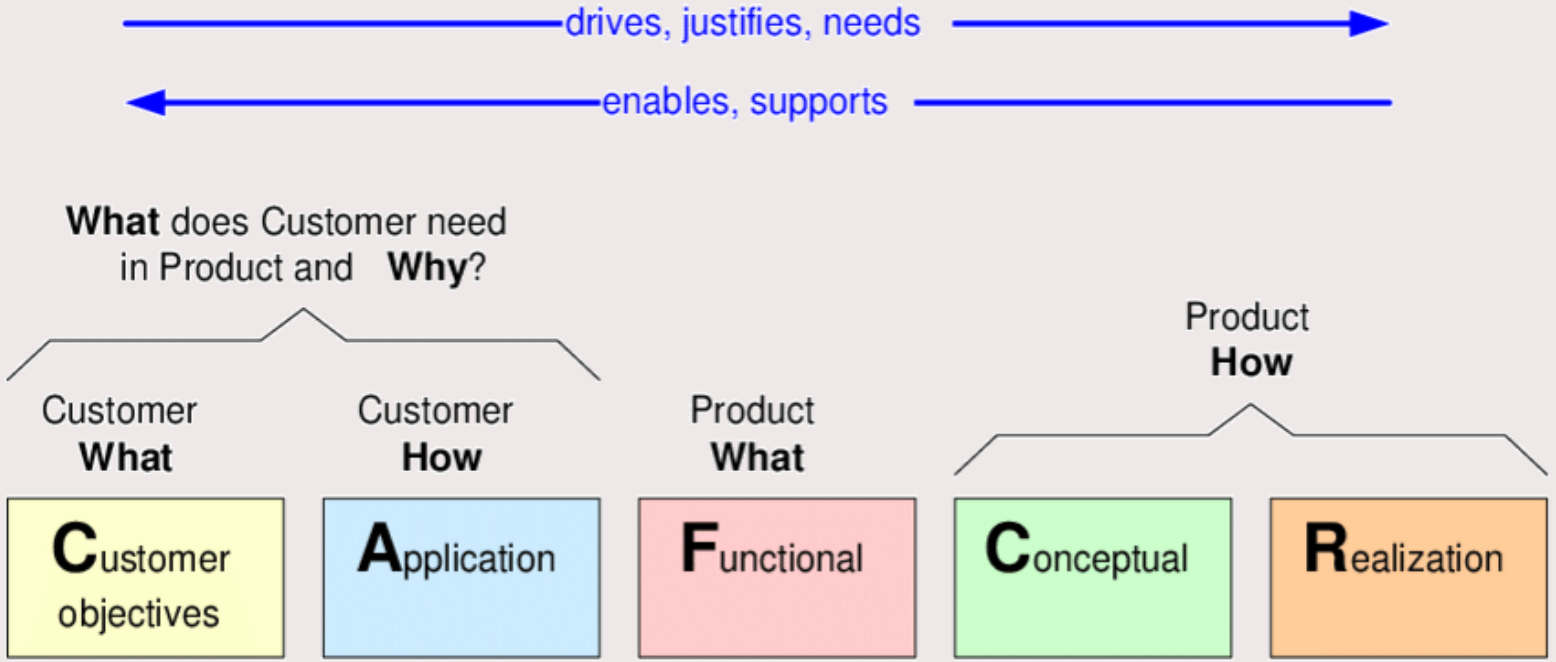
How to get from Desires to Specs?



How to get from Desires to Specs?



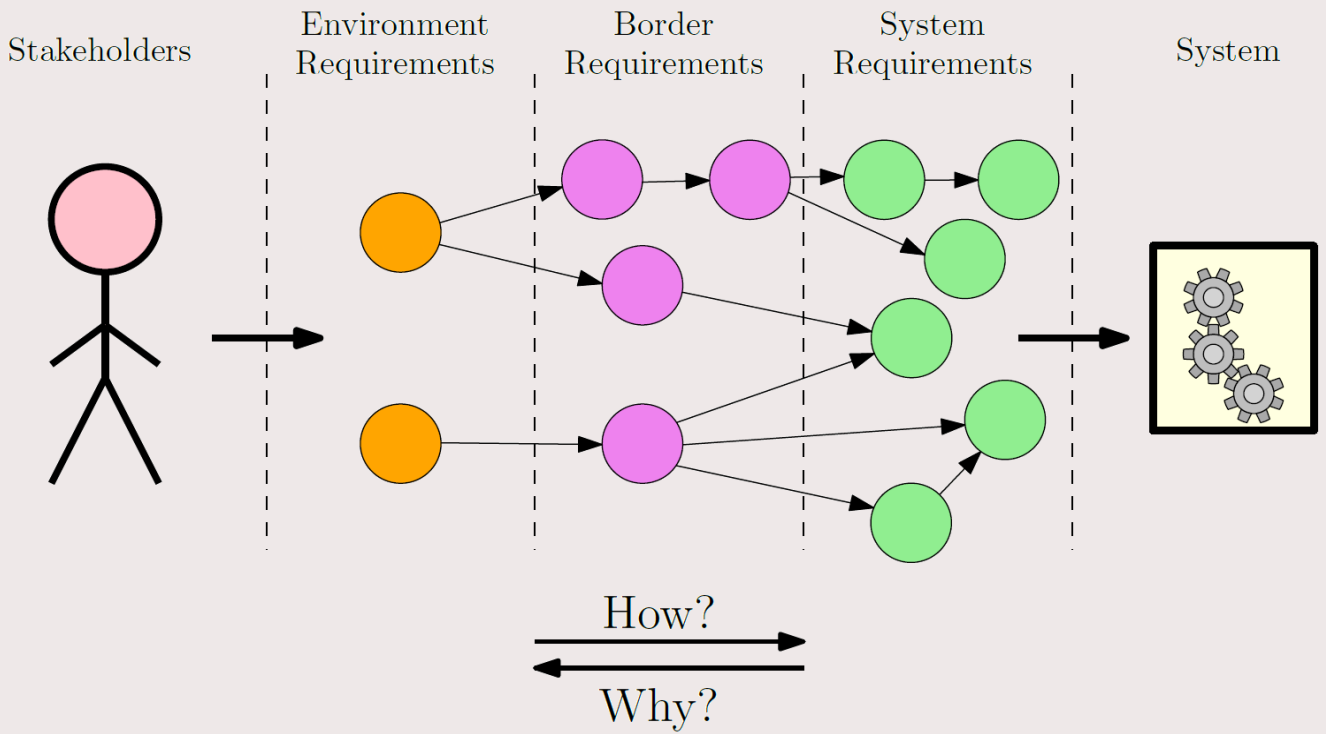
CAFCR model



Muller, G.. (2004). CAFCR: A Multi-view Method for Embedded Systems Architecting. Balancing Genericity and Specificity.

https://www.gaudisite.nl/

From Desires to Specs

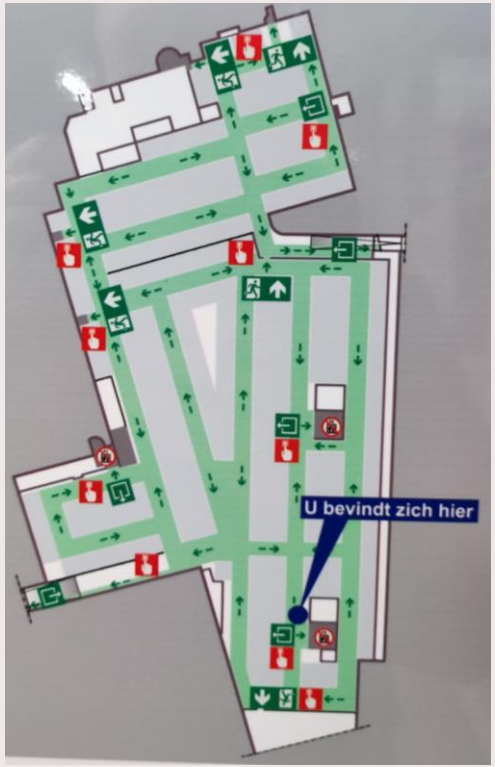


From Desires to Specs

Environment requirement	Border requirements	System requirements
Describe a customer-need without a solution in mind	Links system solution to the task	Describes a parameter (KPI) of a specific solution
Often vague and not measurable	Solution direction and type of models in mind	Simple and verifiable values
Come from stakeholders, directed to task	Design decisions	Come from models of the systems

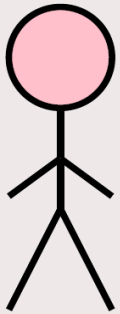
Example: Cleaning robot

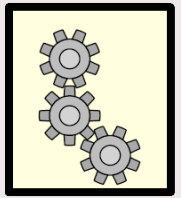
Data Flow Diagram – Cleaning robot



From Desires to Specs – Cleaning robot

Stakeholders | Environment Requirements | Border Requirements | System Requirements | System

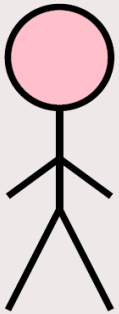


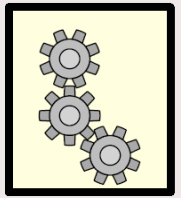


From Desires to Specs – Cleaning robot

Stakeholders | Environment Requirements | Border Requirements | System Requirements | System

Parking host



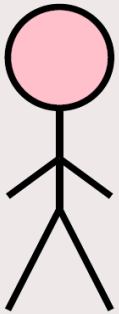


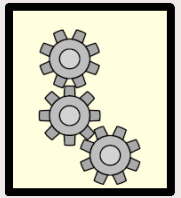
From Desires to Specs – Cleaning robot

Stakeholders | Environment Requirements | Border Requirements | System Requirements | System

Clean entire floor once a week

Parking host



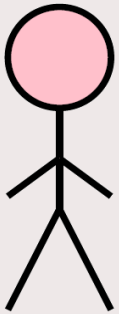


TU/e

From Desires to Specs – Cleaning robot

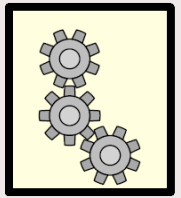
Stakeholders | Environment Requirements | Border Requirements | System Requirements | System

Parking host



Clean entire floor once a week

Safe around humans



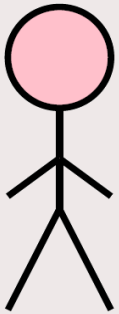
TU/e

From Desires to Specs – Cleaning robot

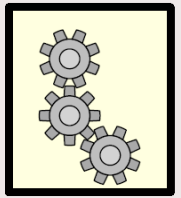
Stakeholders | Environment Requirements | Border Requirements | System Requirements | System

Clean entire floor once a week

Parking host

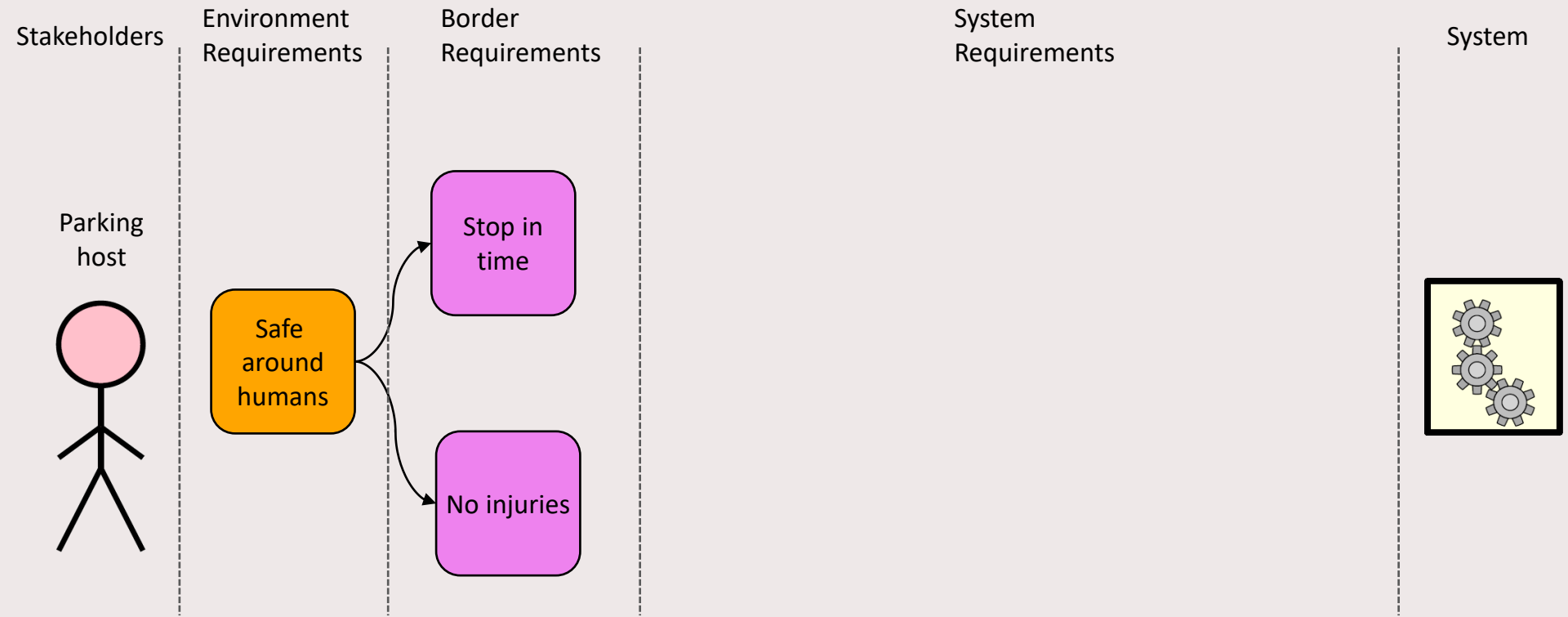


Safe around humans

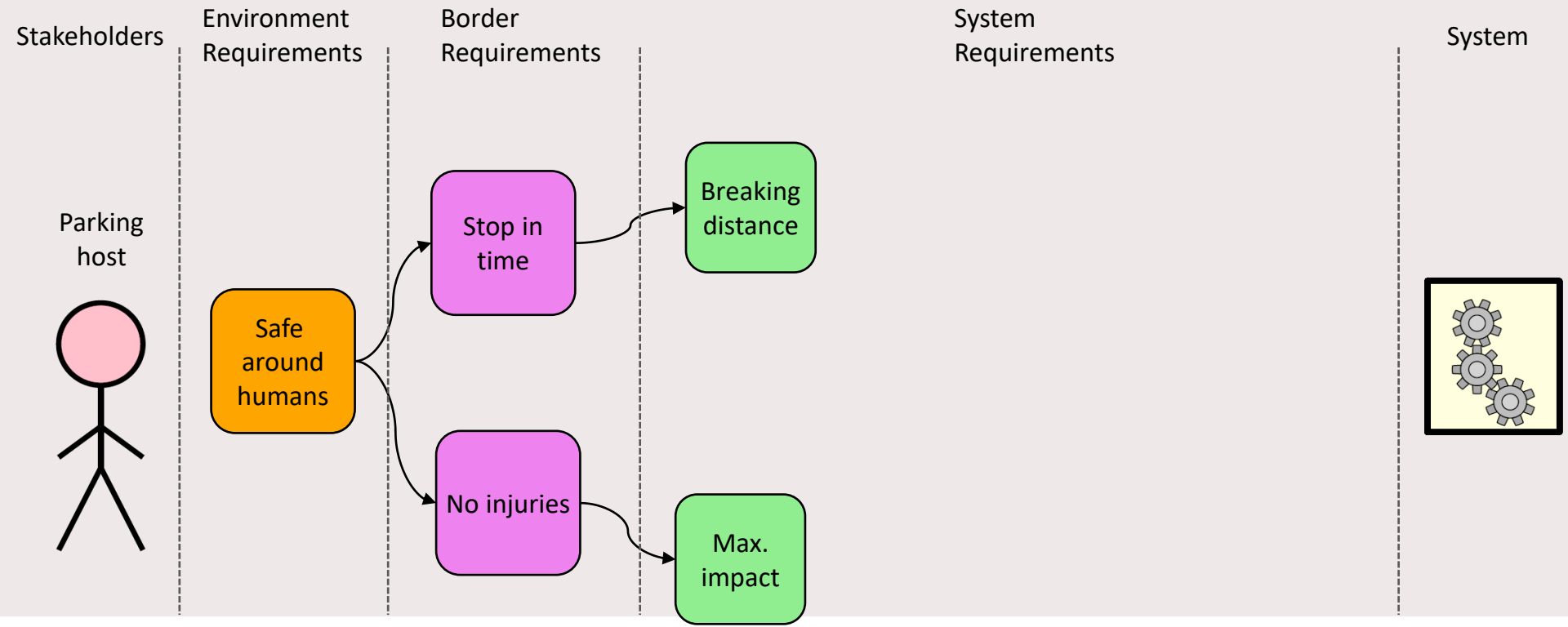


TU/e

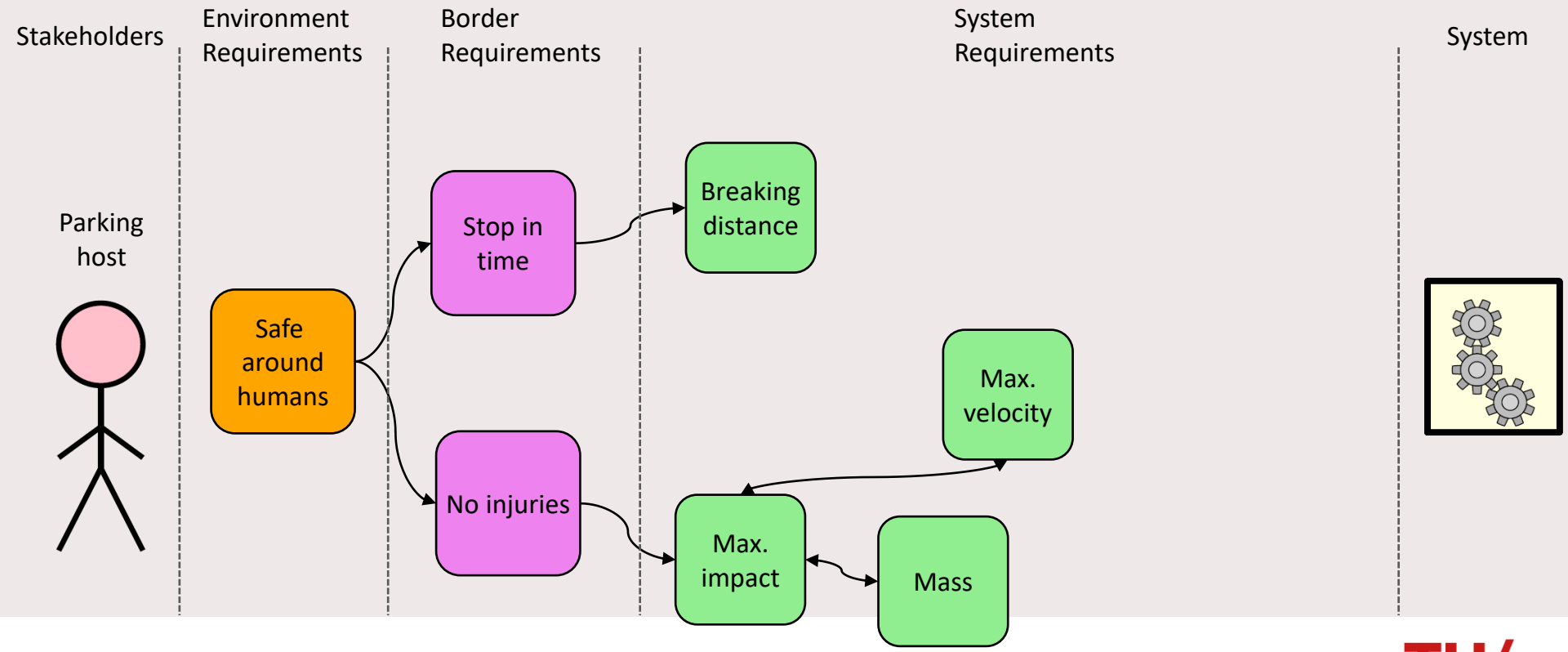
From Desires to Specs – Cleaning robot



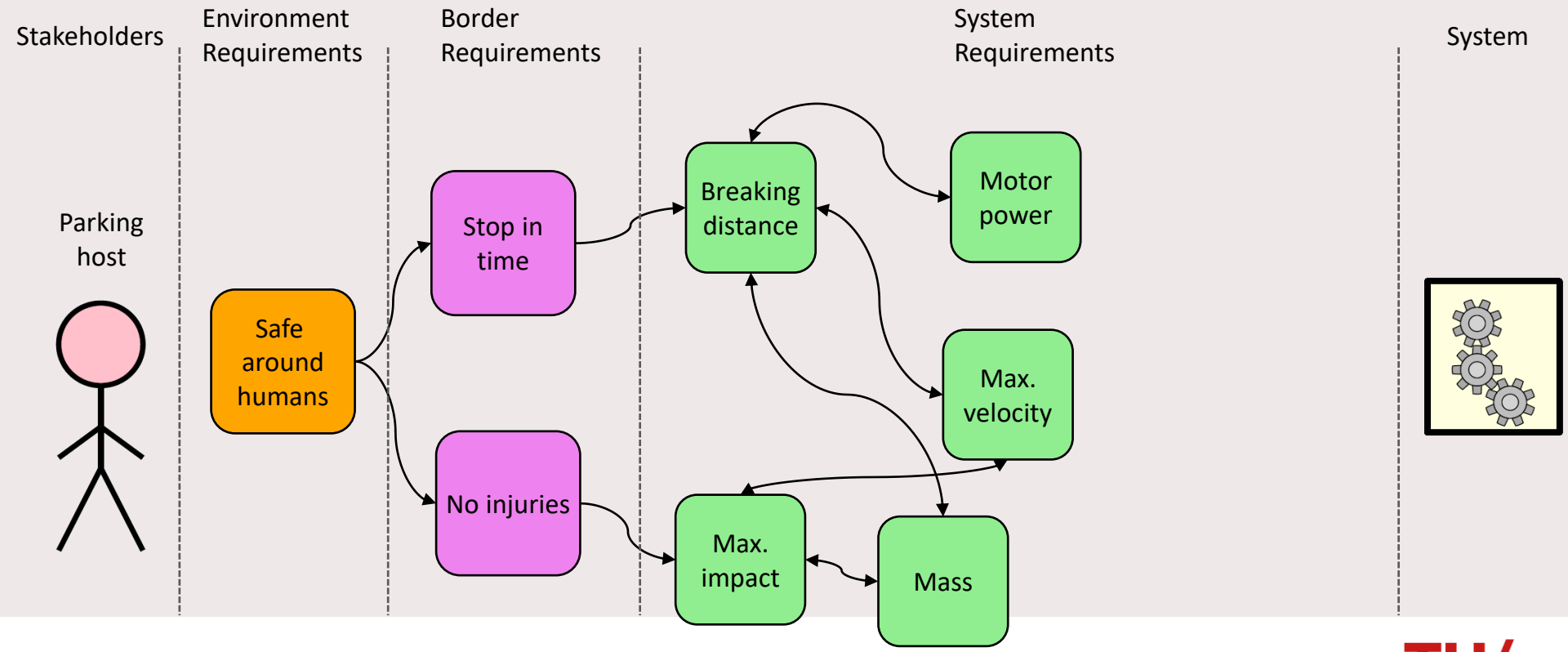
From Desires to Specs – Cleaning robot



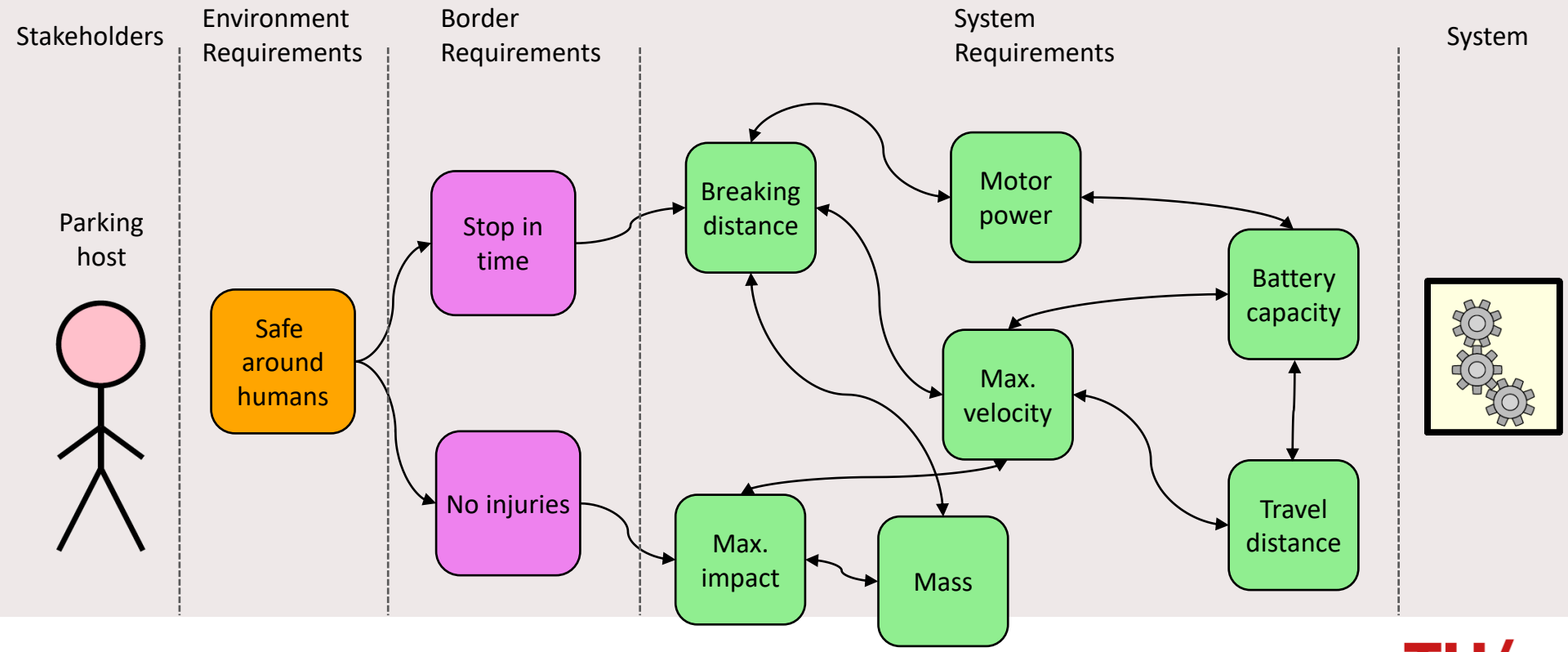
From Desires to Specs – Cleaning robot



From Desires to Specs – Cleaning robot



From Desires to Specs – Cleaning robot



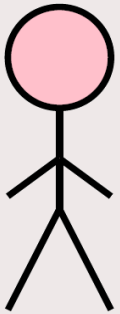
From Desires to Specs – Cleaning robot

Stakeholders

Environment Requirements

System

Parking host



Safe around humans

No damage to cars

No bumping walls+pillars

Cleaning quality

Mopping + vacuuming

Clean entire floor once a week

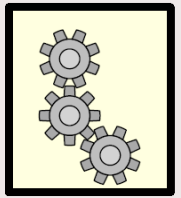
Slopes

Visibility

Autonomy level 4

Self docking

Easy to use



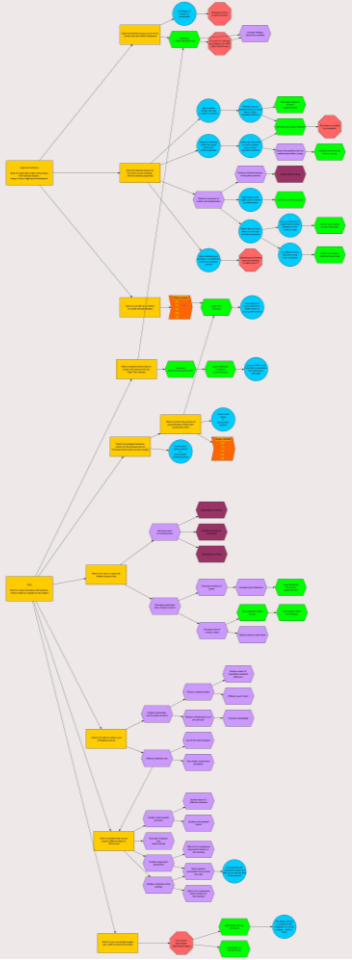
TU/e

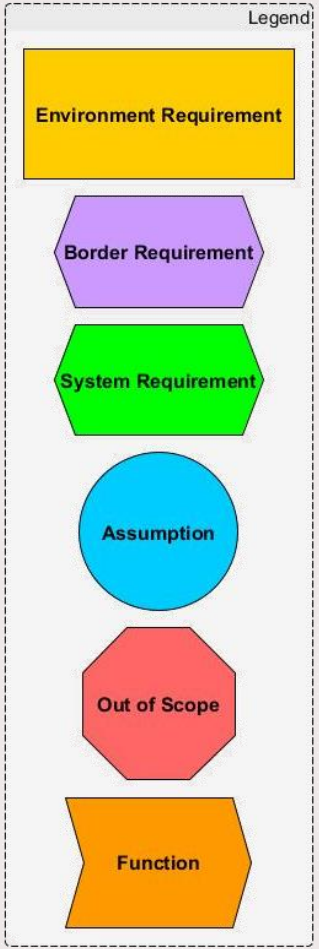
Why do this?

- Bookkeeping (explicit choices)
- Insight in conflicts
- Ordering of importance
- Back-traceability

- **NO MORE MAGIC NUMBERS!**
  Values are a result of models and/or explicit assumptions

- Coherence in group
- Discussion points
- Comparing system designs



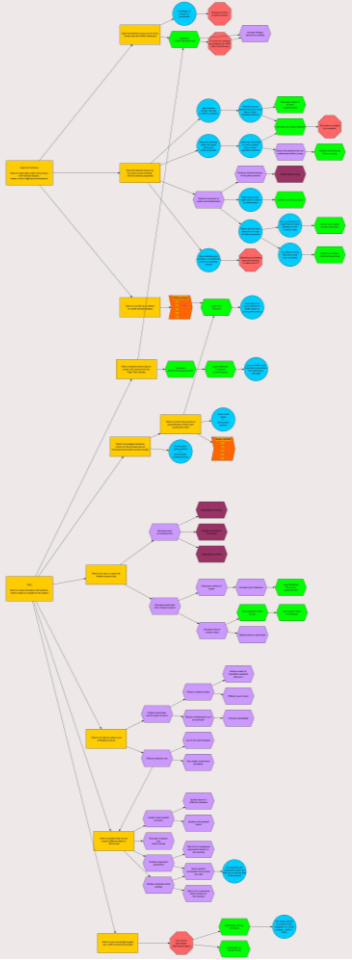


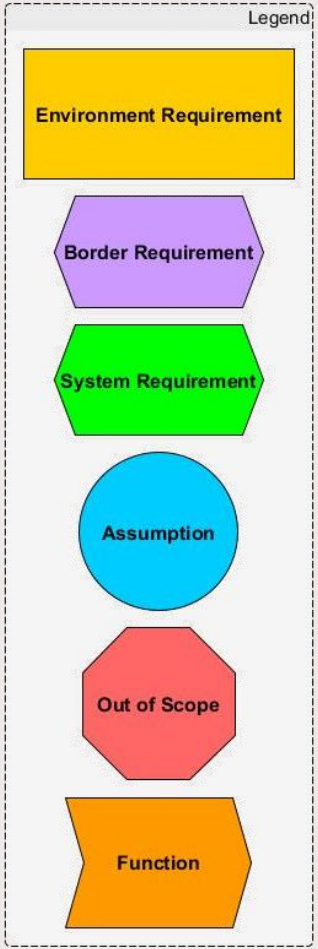
Challenges

- Many stakeholders
- Conflicting desires
- Design without system in mind
- No tangible output (at first)

- Boring (no it is not!)

- Not set in stone, fluid
- Not only top-down (left to right)
- Not an exact science!
- Use it as a tool





Software Design

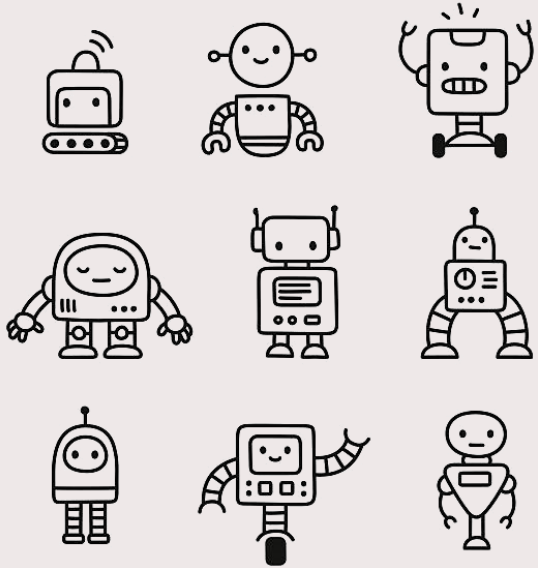
Software Design

Data Flow Diagram

- Functionalities
- Flow of data and information

State Diagram

- Behavior of the system



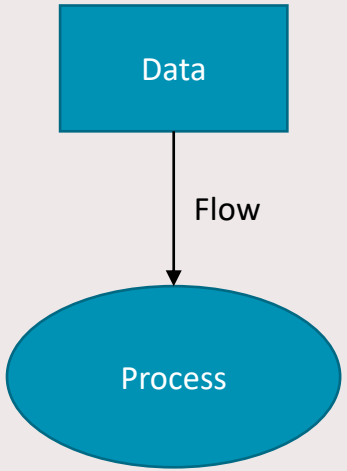
Data Flow Diagram

A form of structured analysis which gives insights into:

- Origin of data
- Interfaces between processes

Consist of:

- **Data**: information of the system and the world
- **Process**: a functional component with inputs and outputs
- **Flow**: specify an input/output relation between process and data sources



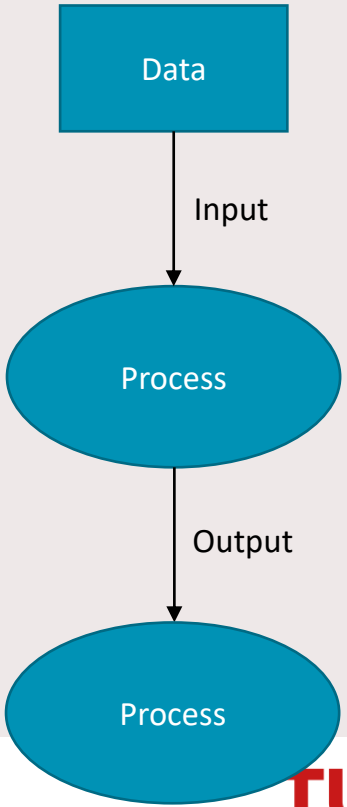
Data Flow Diagram

A form of structured analysis which gives insights into:

- Origin of data
- Interfaces between processes

Consist of:

- **Data**: information of the system and the world
- **Process**: a functional component with inputs and outputs
- **Flow**: specify an input/output relation between process and data sources



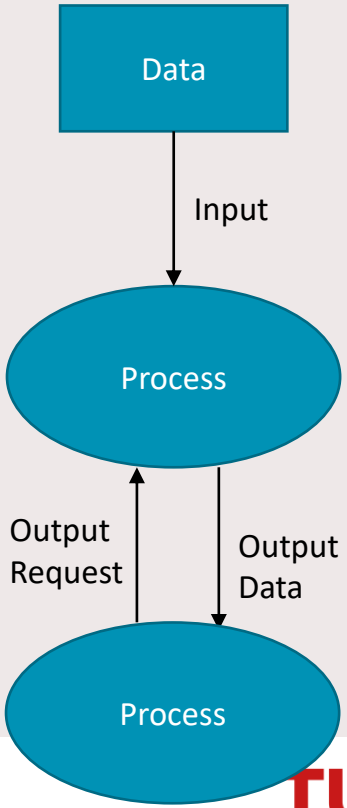
Data Flow Diagram

A form of structured analysis which gives insights into:

- Origin of data
- Interfaces between processes

Consist of:

- **Data**: information of the system and the world
- **Process**: a functional component with inputs and outputs
- **Flow**: specify an input/output relation between process and data sources



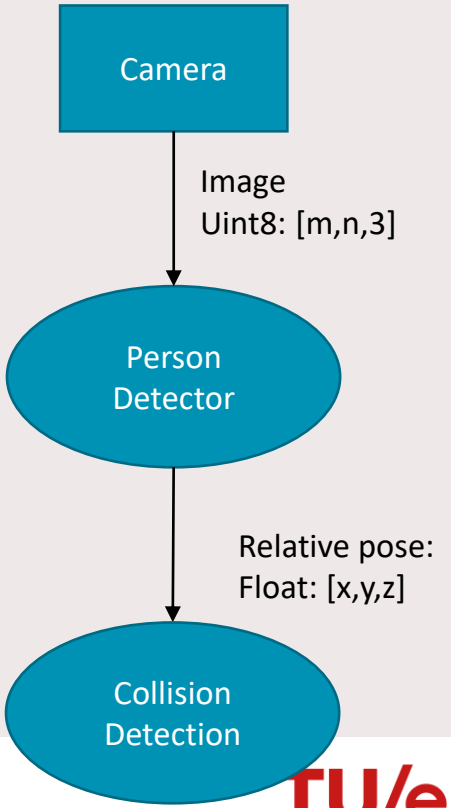
Data Flow Diagram

A form of structured analysis which gives insights into:

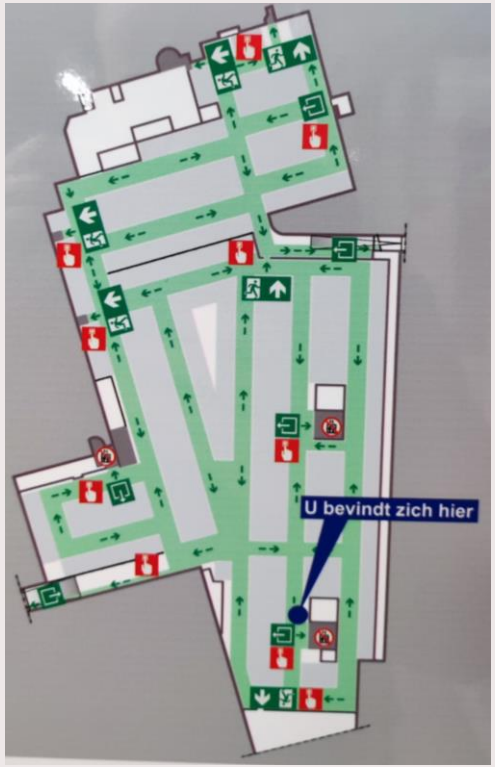
- Origin of data
- Interfaces between processes

Consist of:

- **Data**: information of the system and the world
- **Process**: a functional component with inputs and outputs
- **Flow**: specify an input/output relation between process and data sources



Data Flow Diagram – Cleaning robot

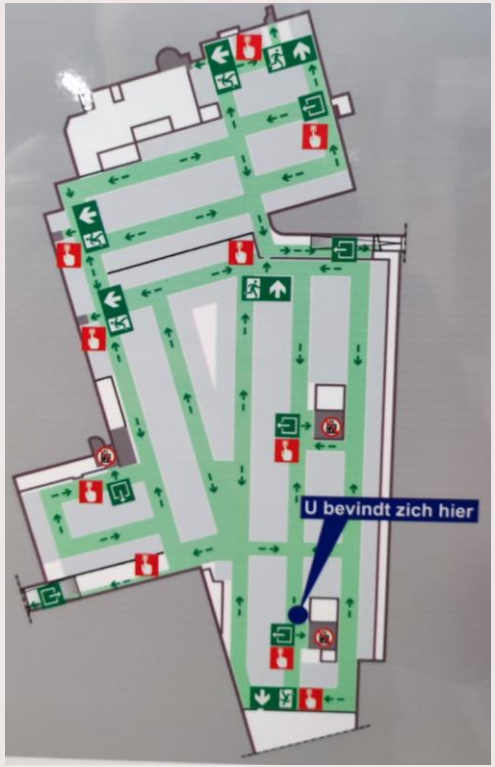


Data Flow Diagram – Cleaning robot

Camera

Odom

Map

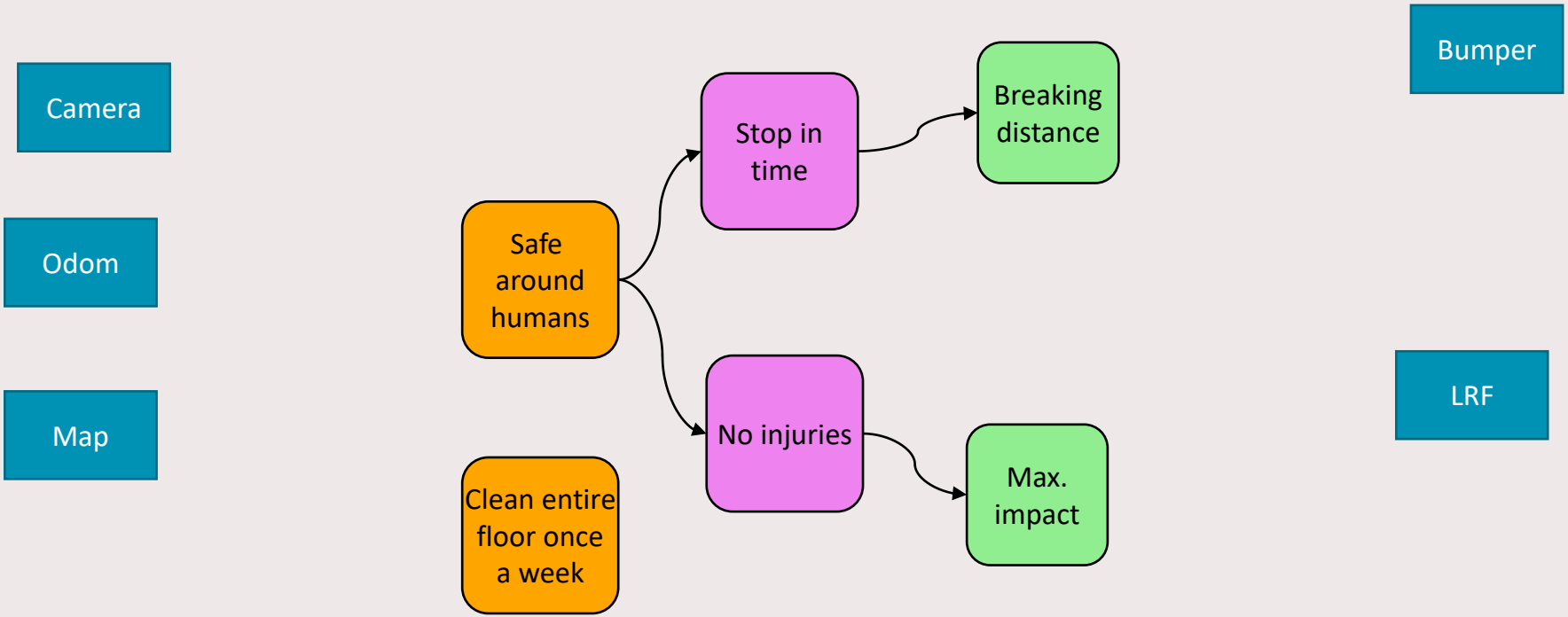


Bumper

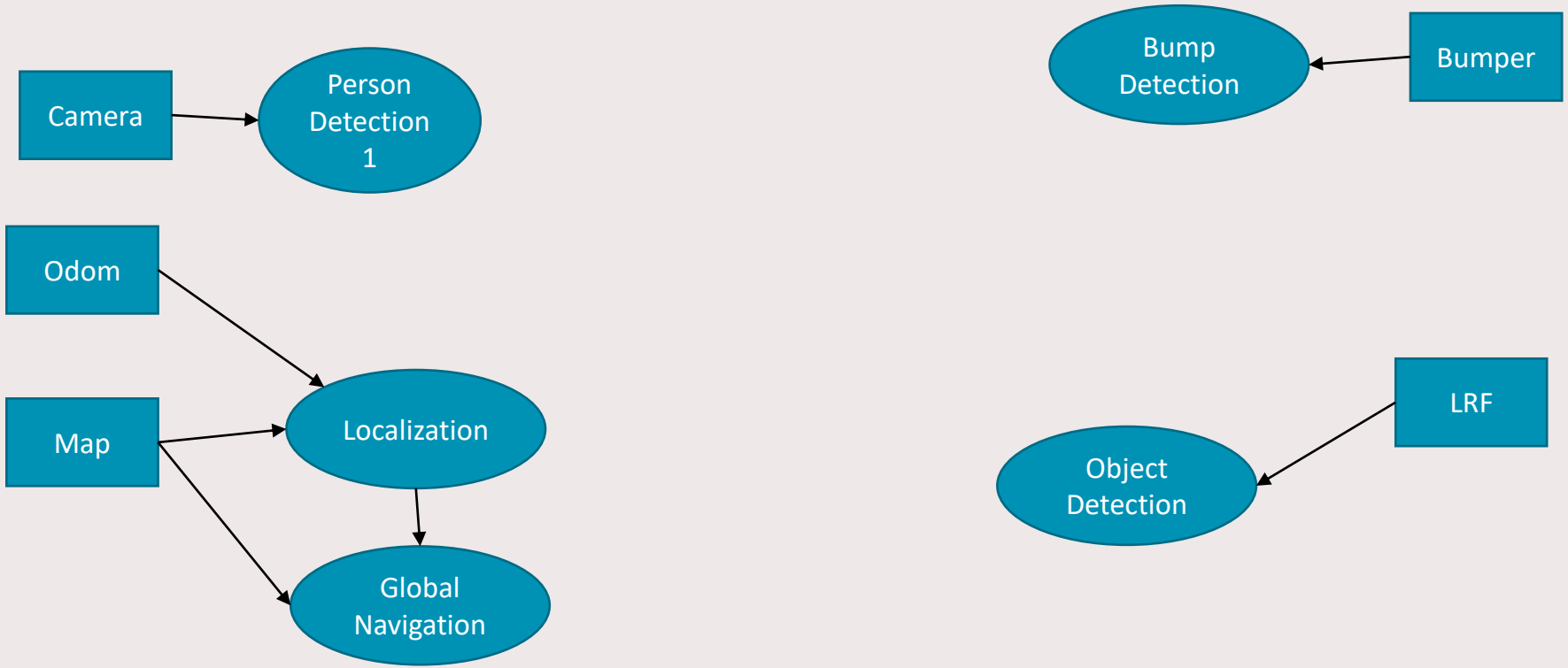
LRF

TU/e

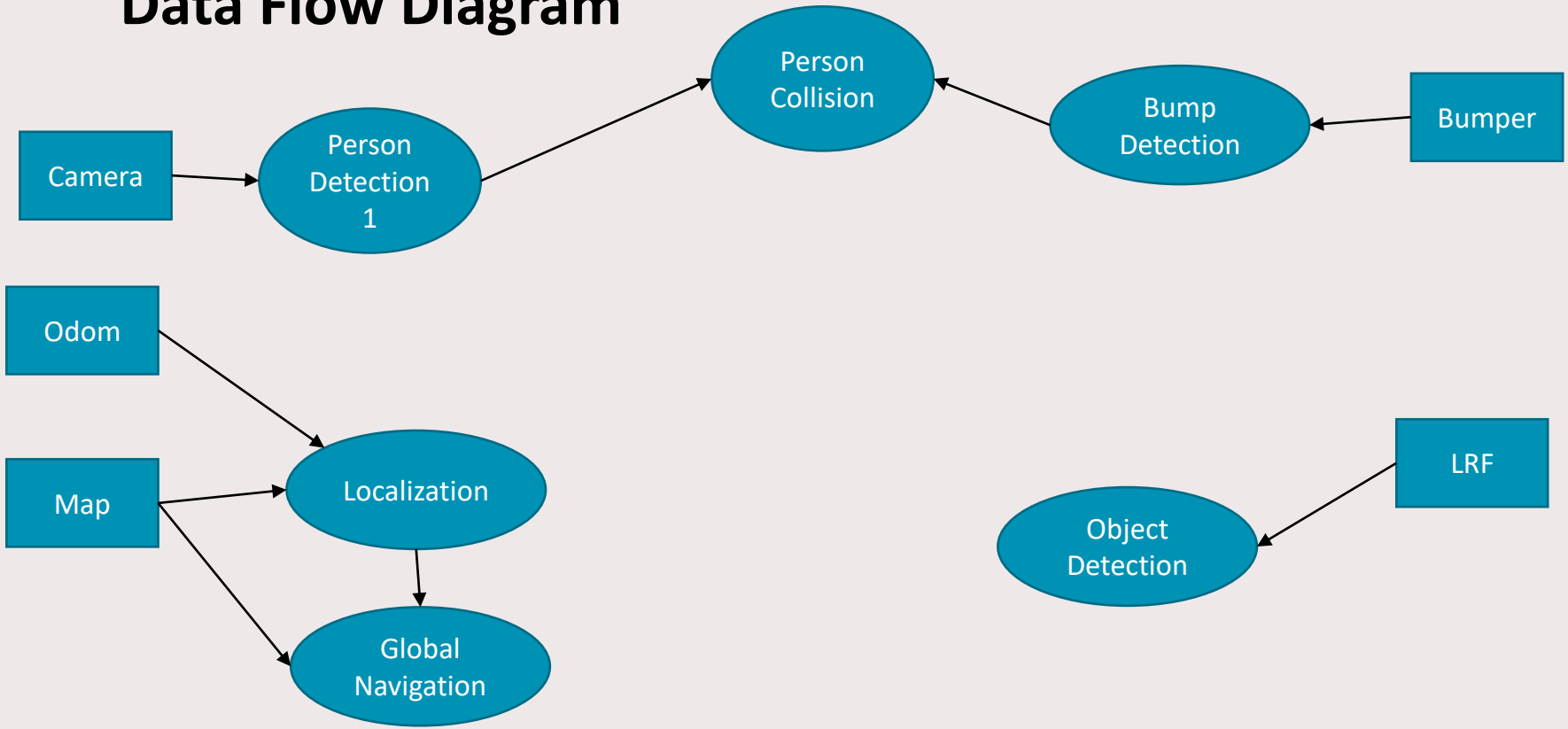
Data Flow Diagram – Cleaning robot



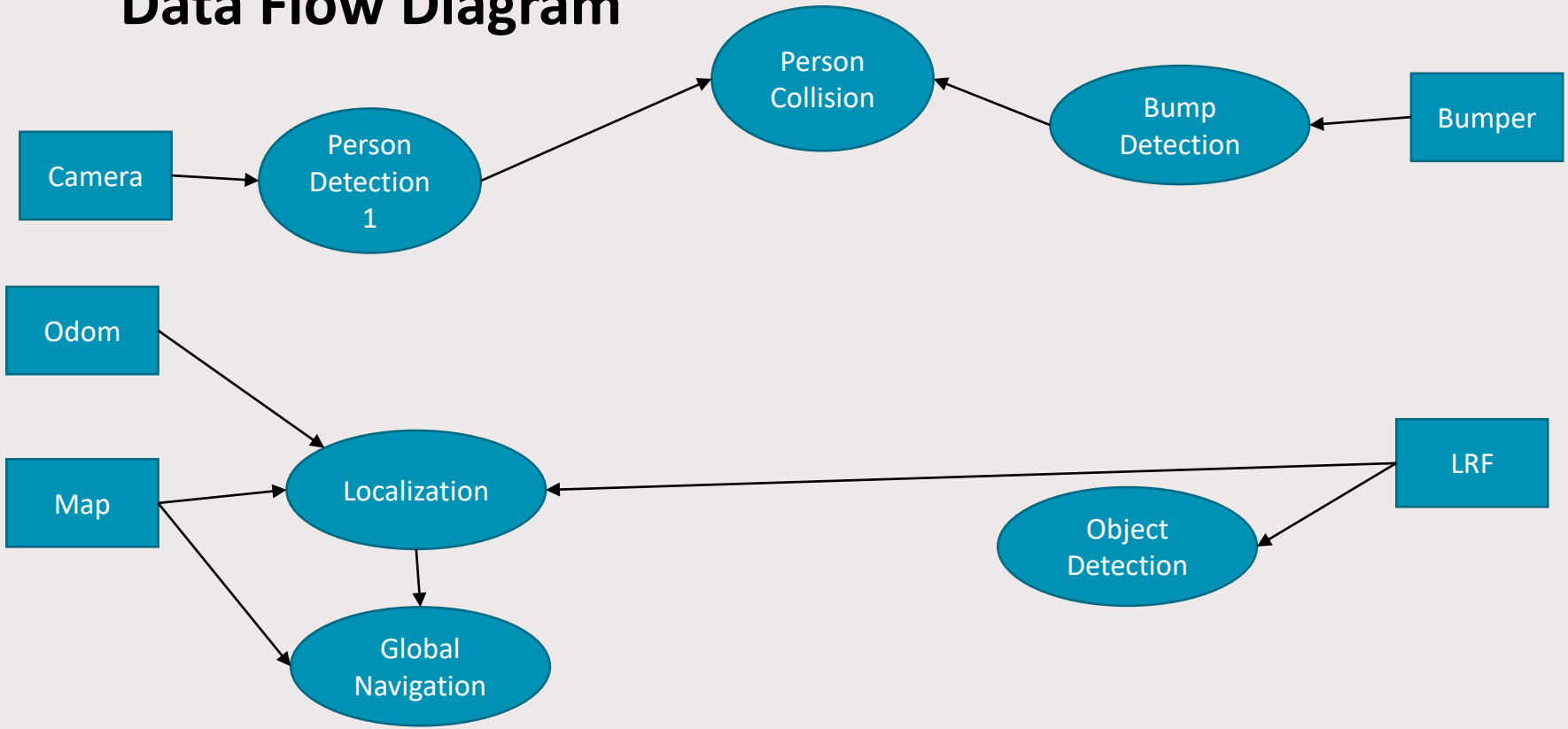
Data Flow Diagram – Cleaning robot



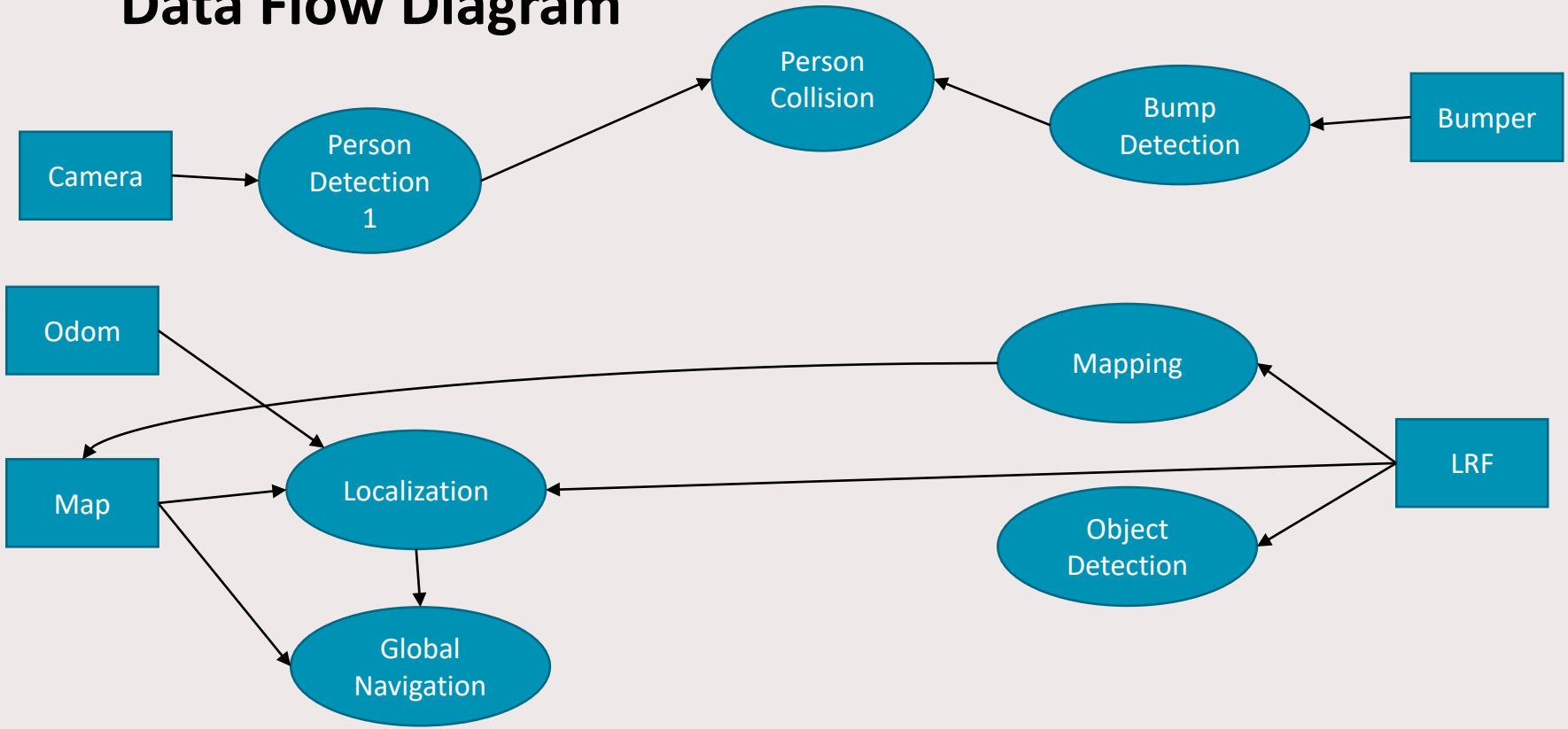
Data Flow Diagram



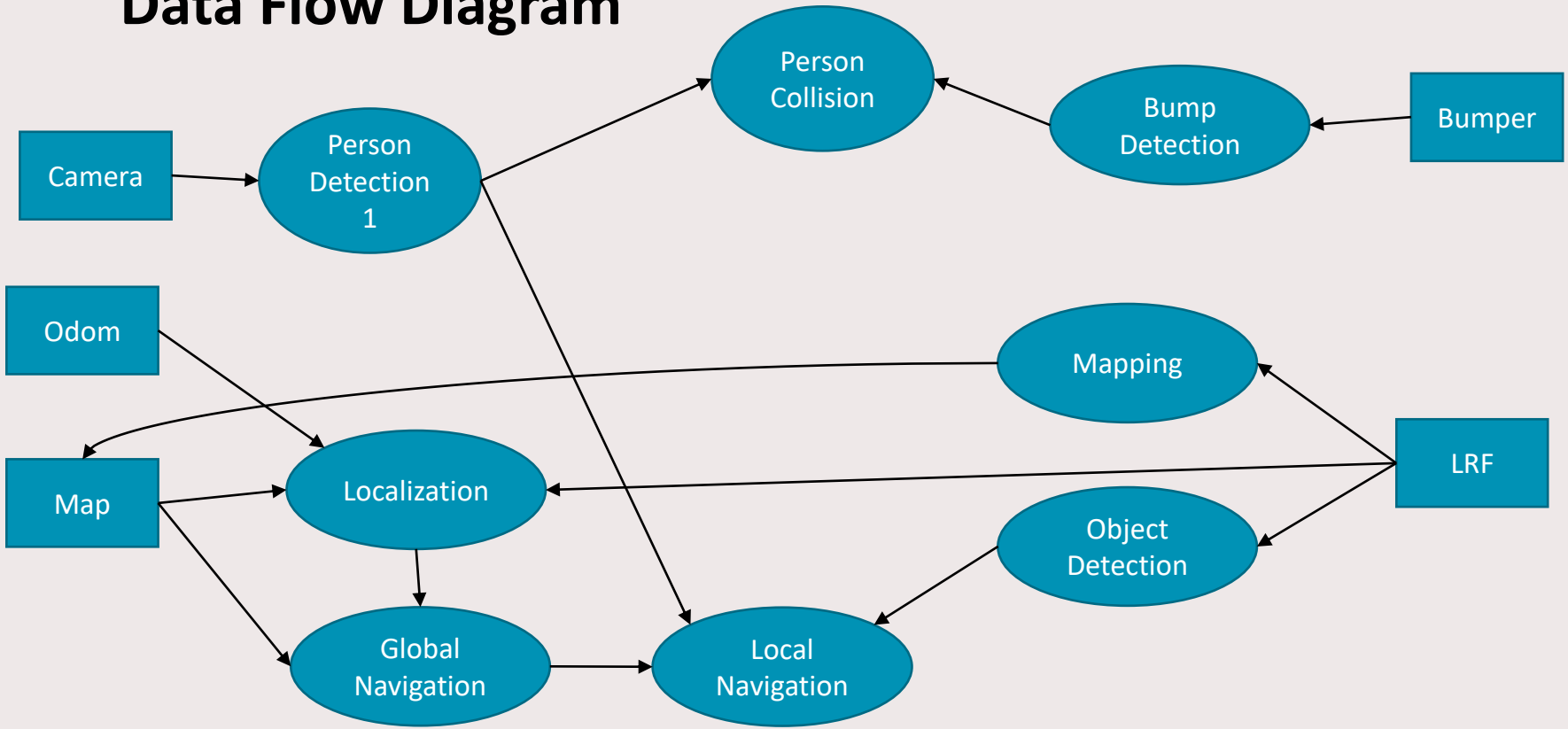
Data Flow Diagram



Data Flow Diagram

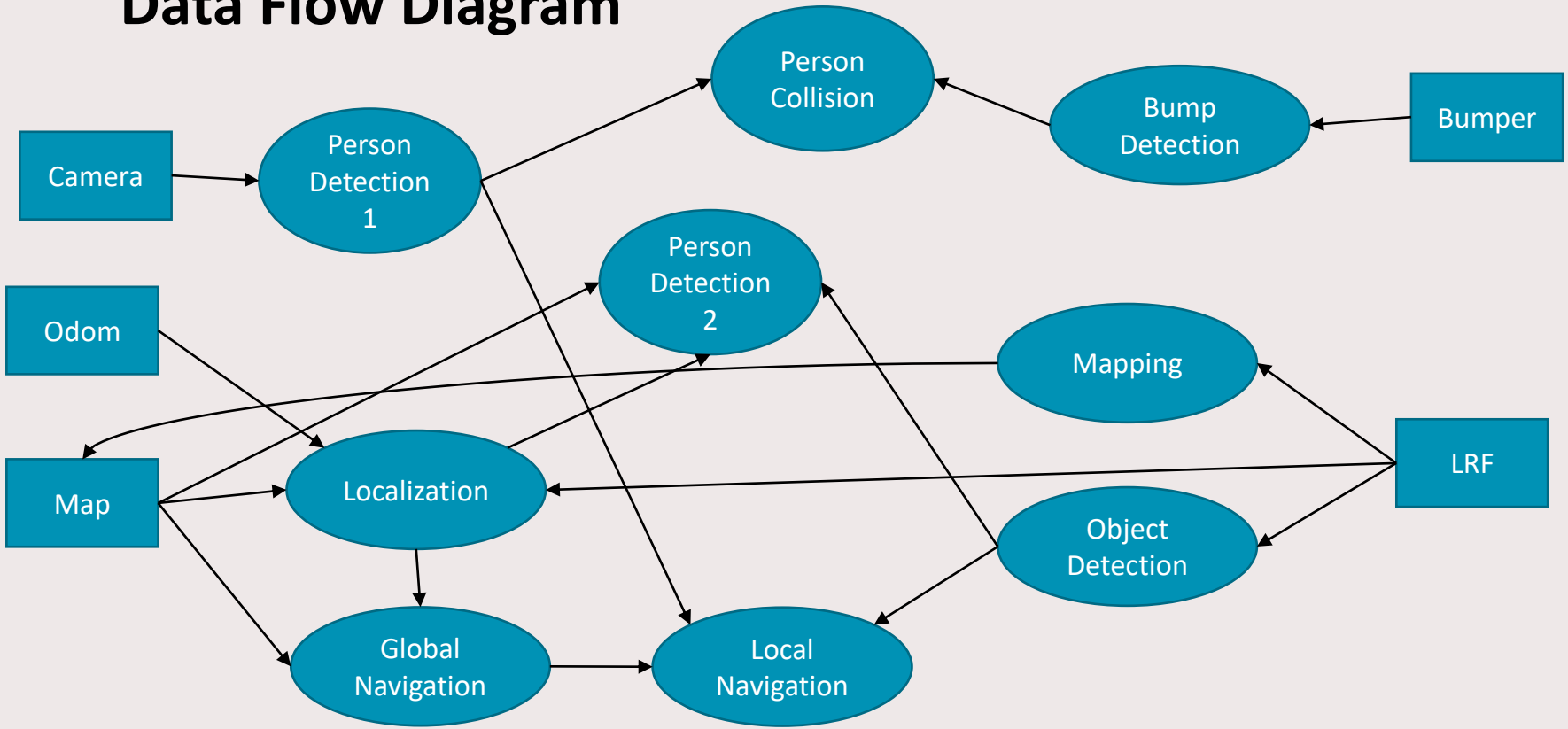


Data Flow Diagram

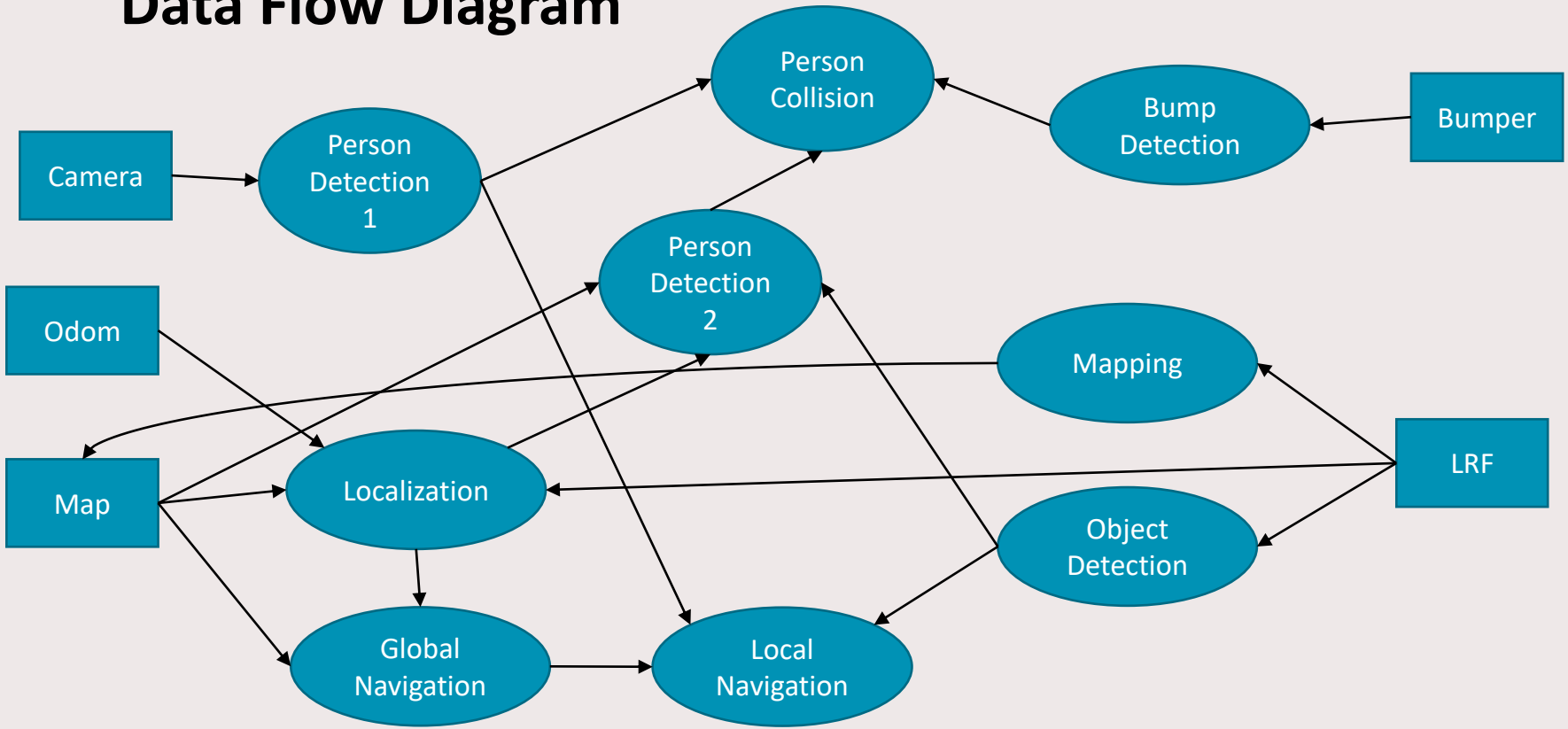


TU/e

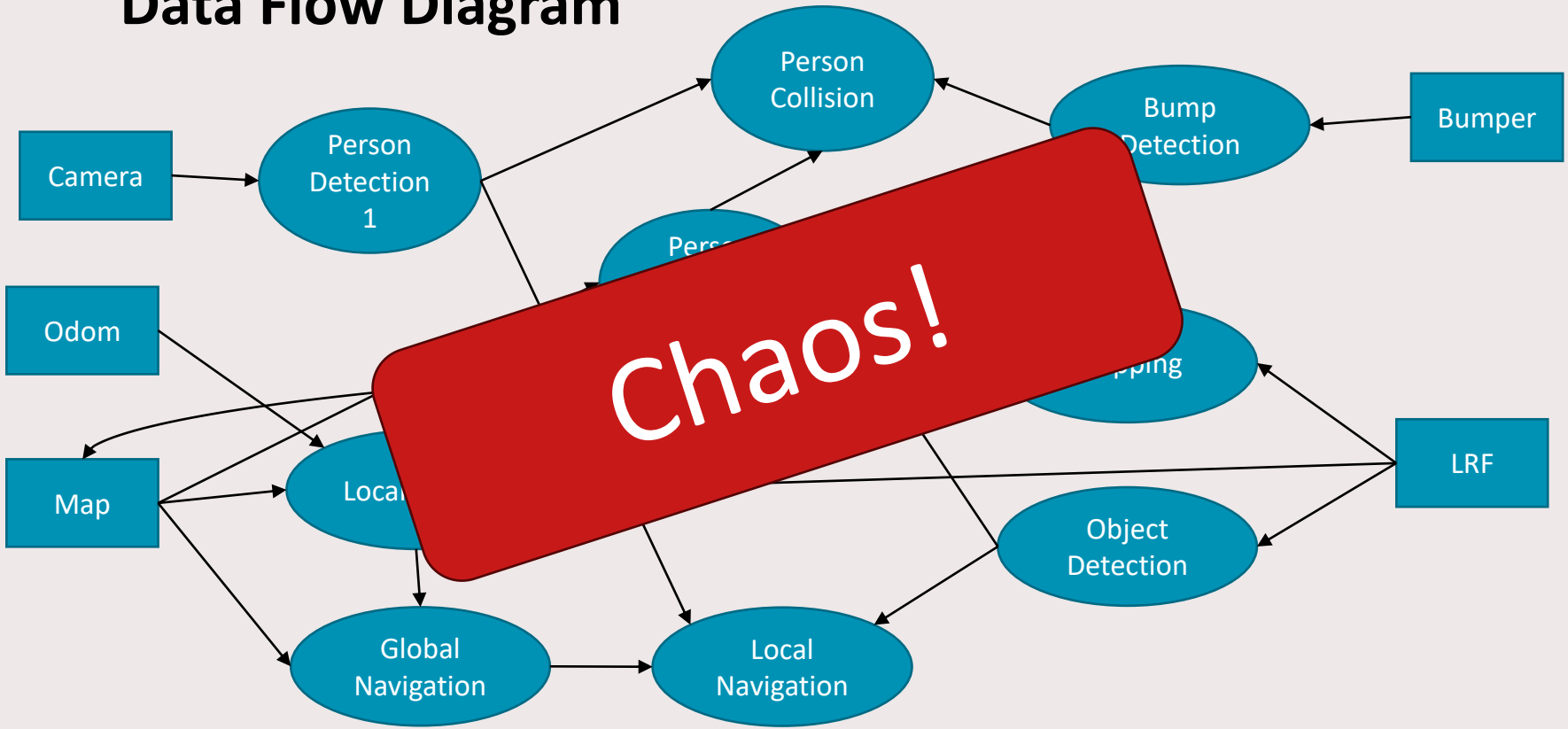
Data Flow Diagram



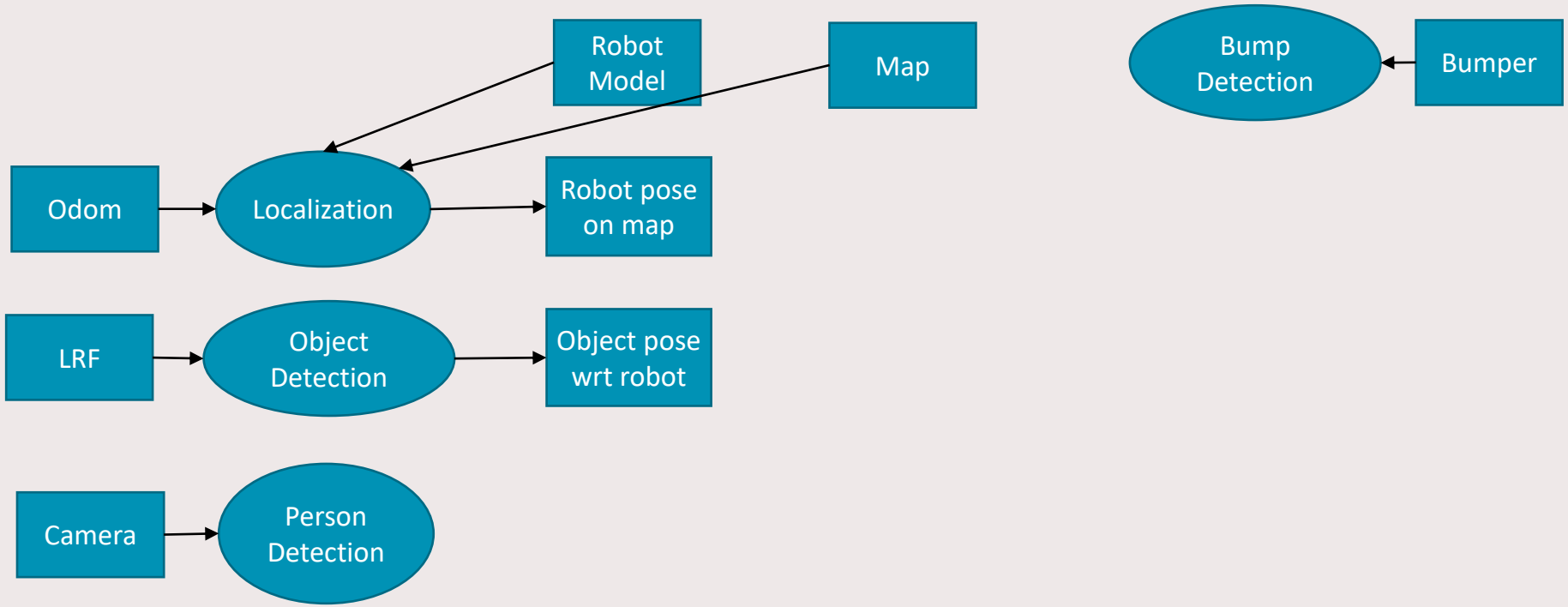
Data Flow Diagram



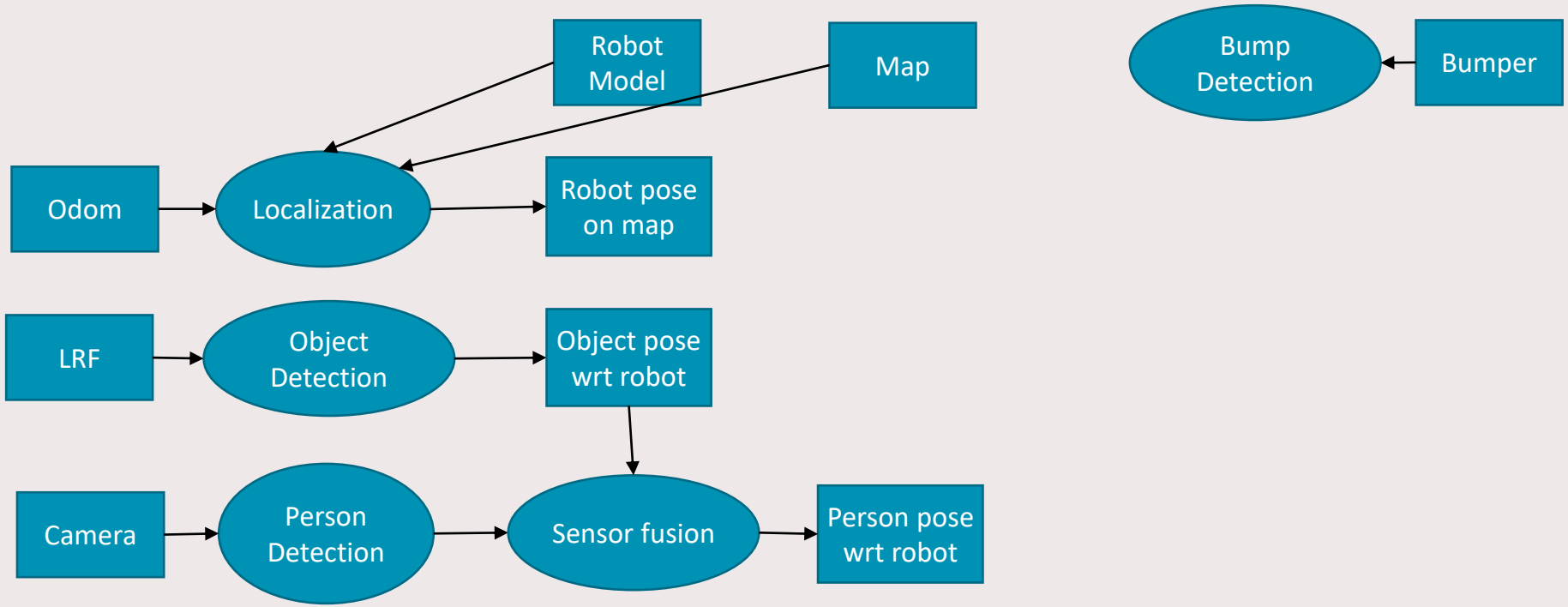
Data Flow Diagram



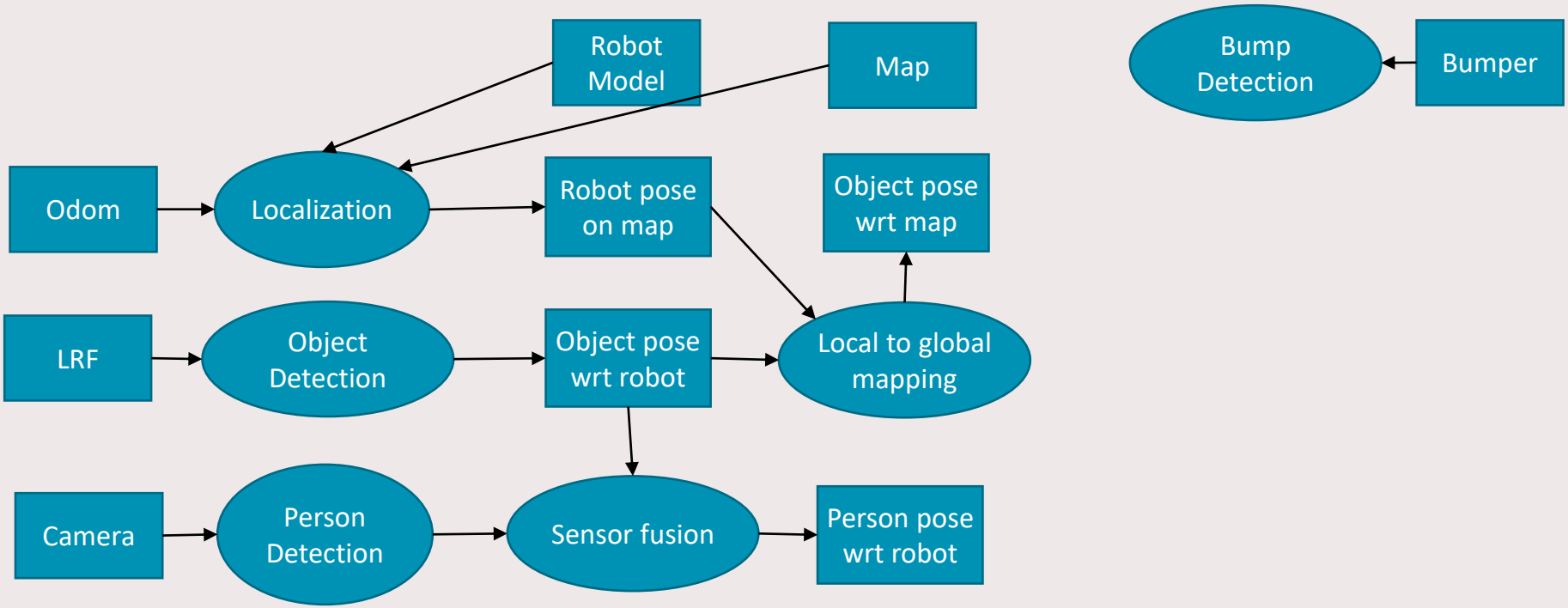
Data Flow Diagram



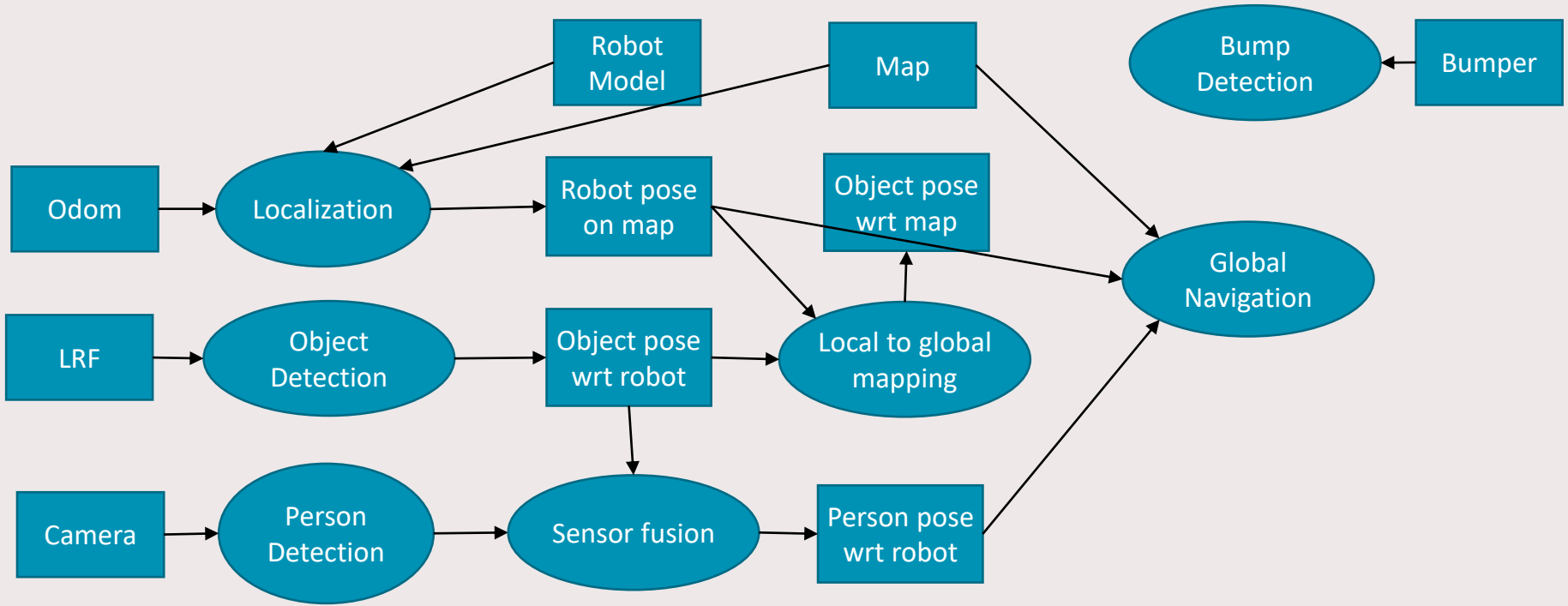
Data Flow Diagram



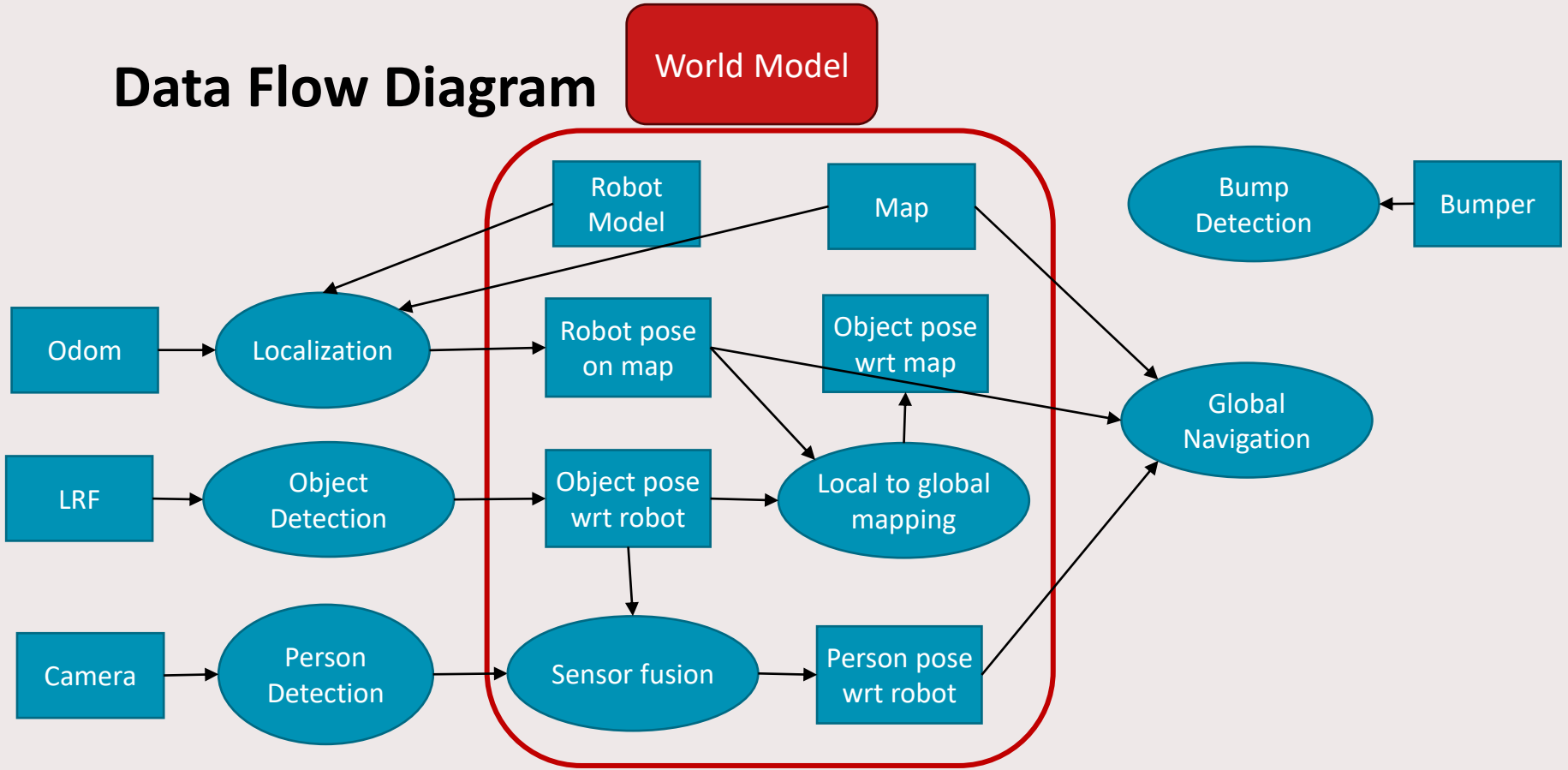
Data Flow Diagram



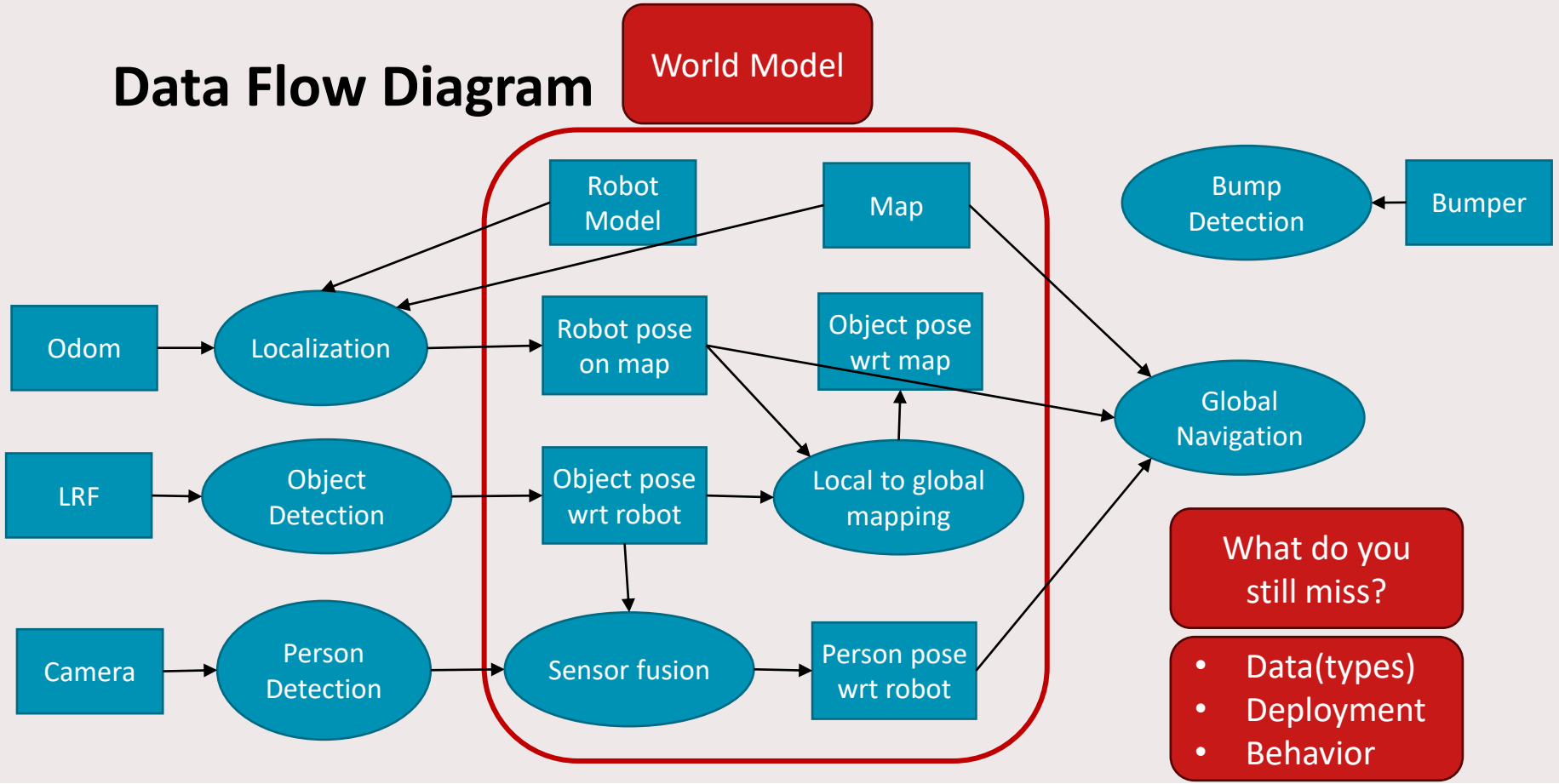
Data Flow Diagram



Data Flow Diagram



Data Flow Diagram

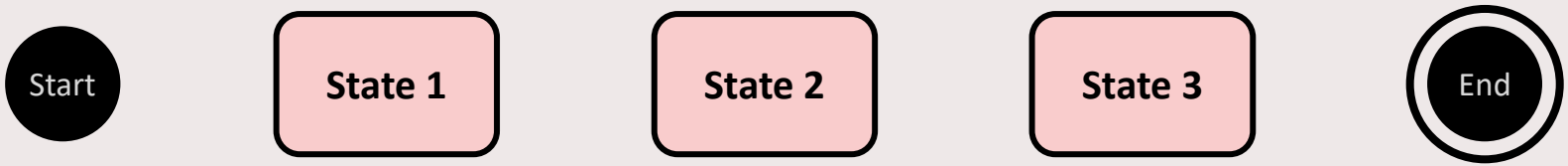


State diagram

State diagrams can be used to graphically represent finite state machines
They allow you to design the discrete control of your system

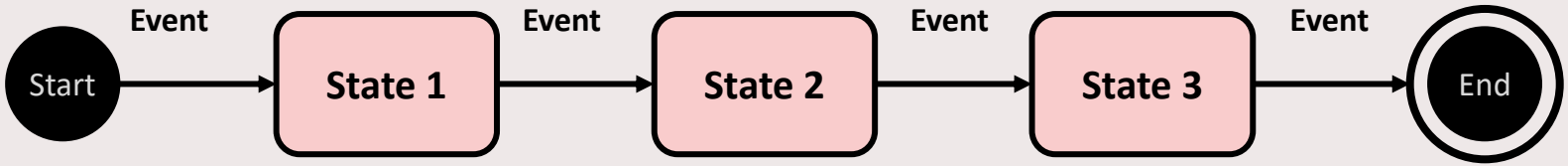
State diagram

State diagrams can be used to graphically represent finite state machines
They allow you to design the discrete control of your system



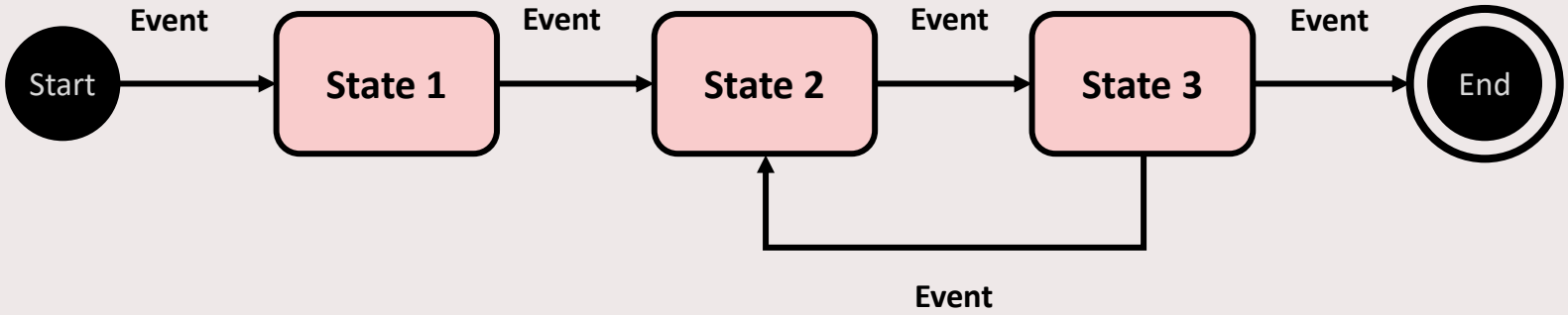
State diagram

State diagrams can be used to graphically represent finite state machines
They allow you to design the discrete control of your system



State diagram

State diagrams can be used to graphically represent finite state machines
They allow you to design the discrete control of your system



State diagram – vending machine

With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

Example from Computerphile

https://www.youtube.com/watch?v=vhiiia1_hC4

State diagram – vending machine

With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

Example from Computerphile
https://www.youtube.com/watch?v=vhiiia1_hC4)

State diagram – vending machine

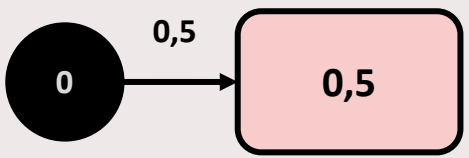
With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

Example from Computerphile
https://www.youtube.com/watch?v=vhiiia1_hC4)

State diagram – vending machine

With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

Example from Computerphile
https://www.youtube.com/watch?v=vhiiia1_hC4)



State diagram – vending machine

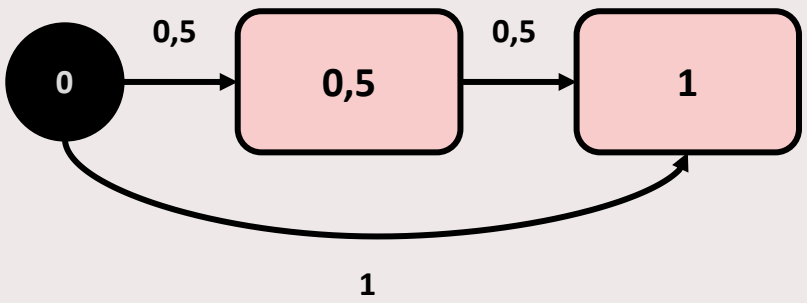
With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

Example from Computerphile
https://www.youtube.com/watch?v=vhiiia1_hC4)

State diagram – vending machine

With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

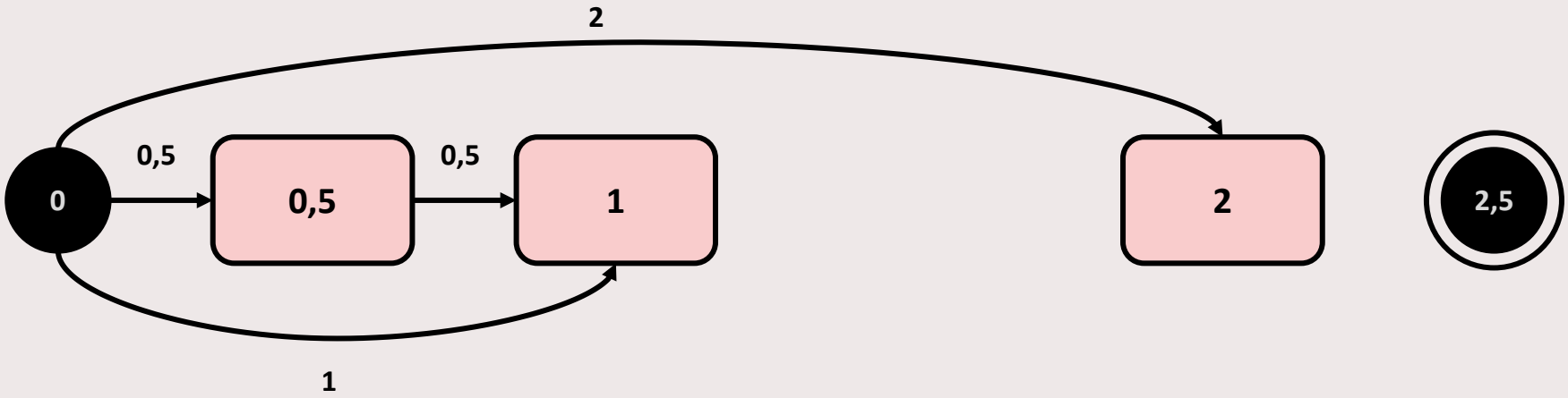
Example from Computerphile
https://www.youtube.com/watch?v=vhiiia1_hC4)



State diagram – vending machine

With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

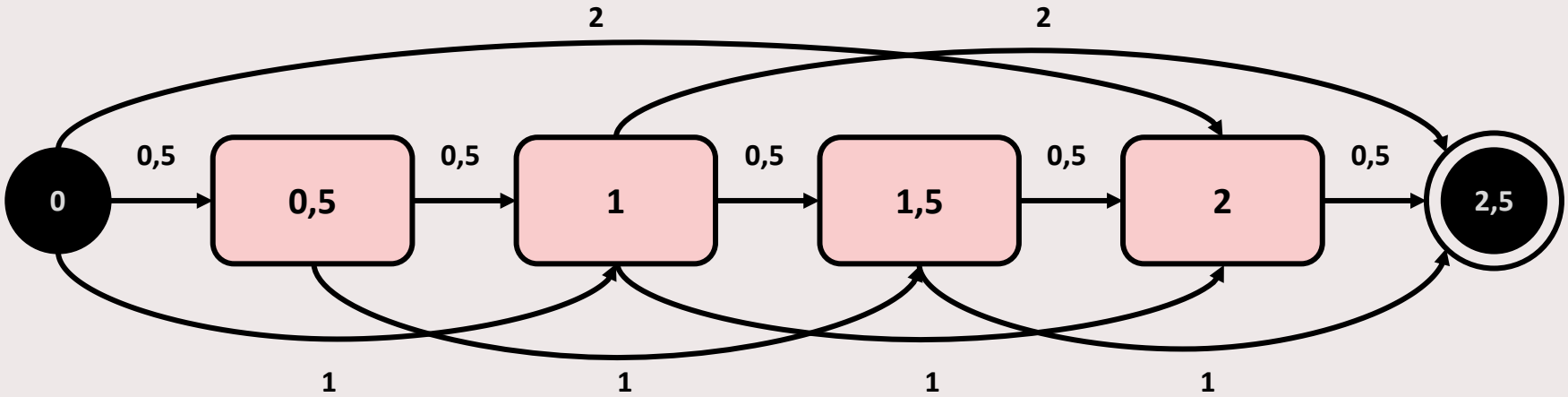
Example from Computerphile
https://www.youtube.com/watch?v=vhiiia1_hC4)



State diagram – vending machine

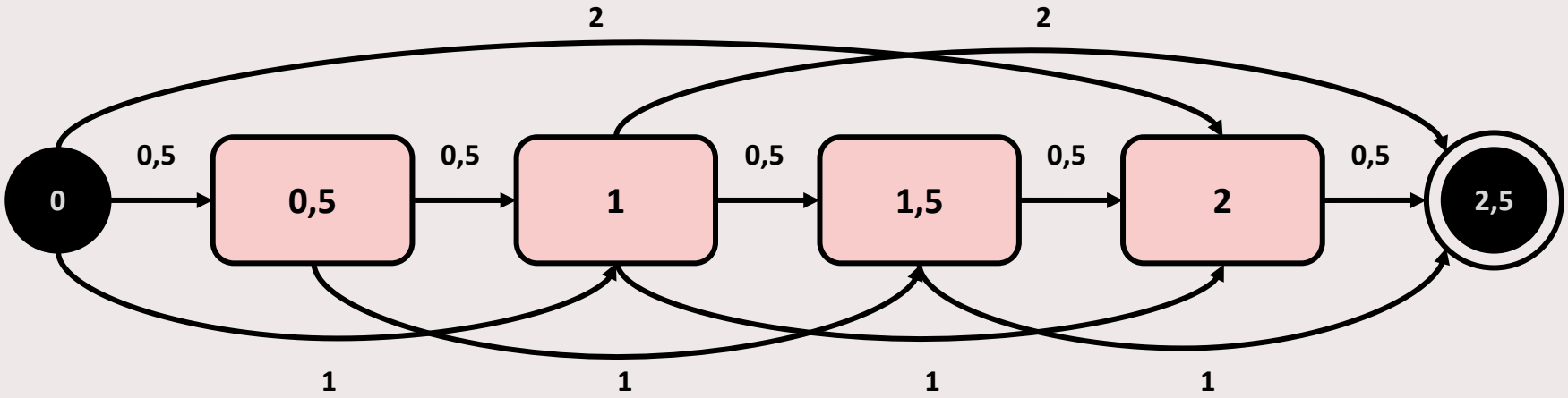
With pieces of €0,50, €1,- and €2,-.. Get to a total of €2,50

Example from Computerphile
https://www.youtube.com/watch?v=vhiiia1_hC4)



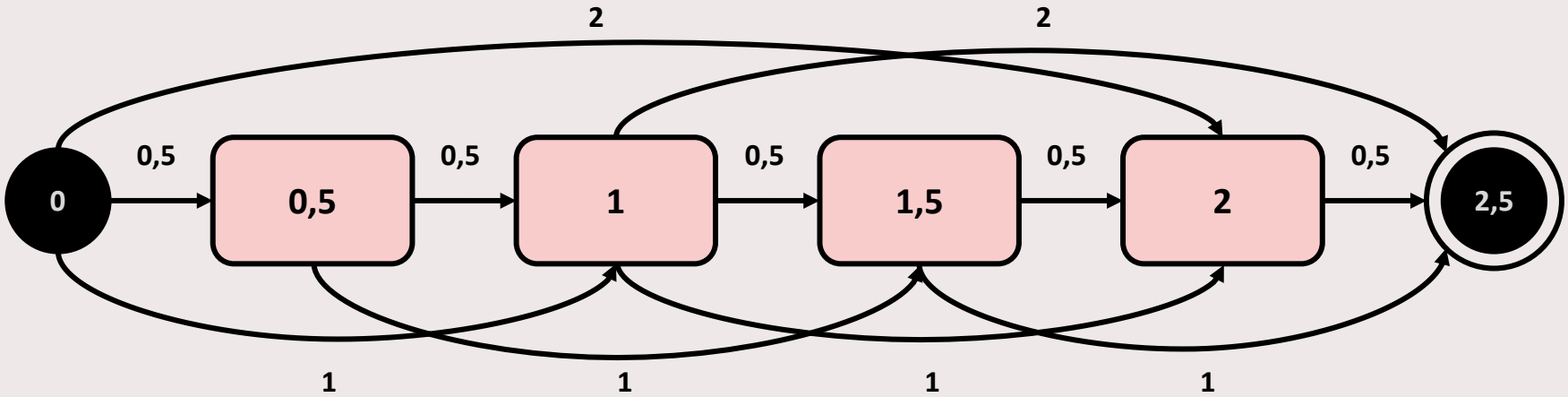
State diagram – vending machine

No memory



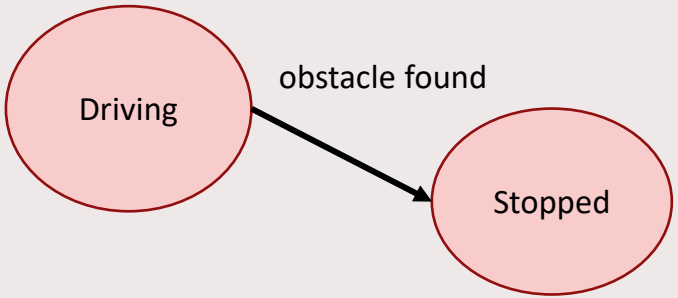
State diagram – vending machine

No memory
Different amount?
Overshoot?



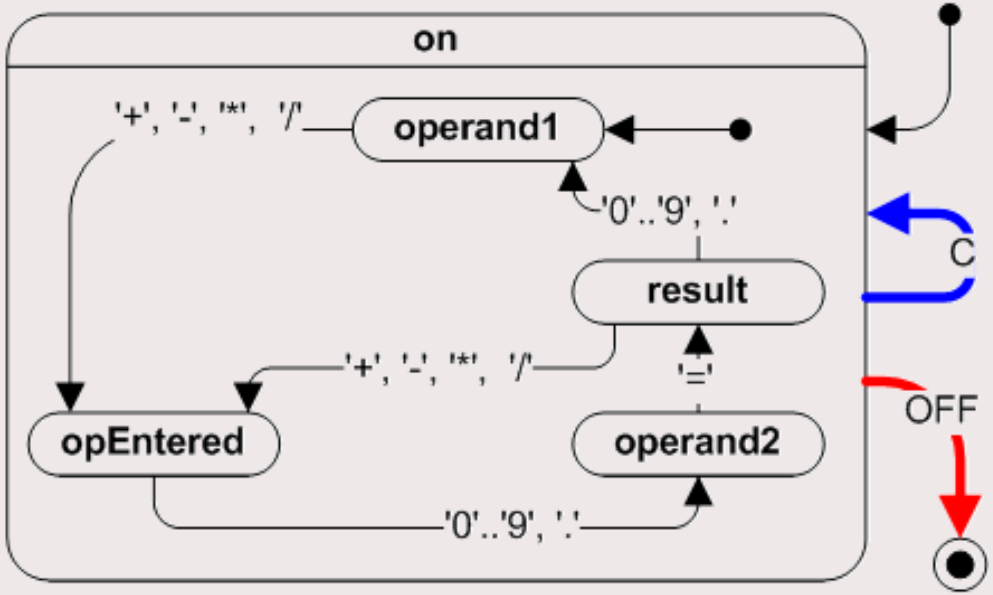
State diagram

Set of states: {State1, State2,… StateN}
Set of events: {Event1, Event2,… EventN}
Transition: triple(from_state, event_name, to_state)
Set of transitions: { (S1, E1, S2), (S2, E2, S1), … (…) }

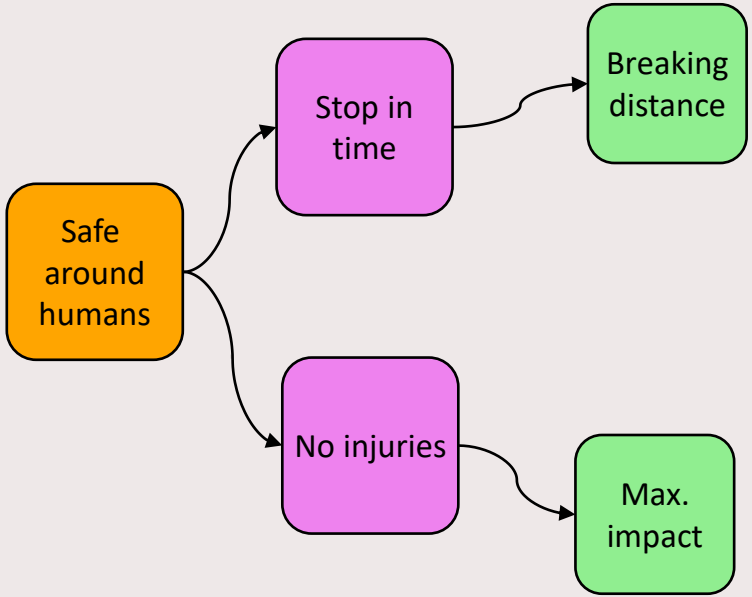


Set of states: {
Driving,
Stopped
}
Set of transitions: {
(Driving, obstacle found, Stopped)
}

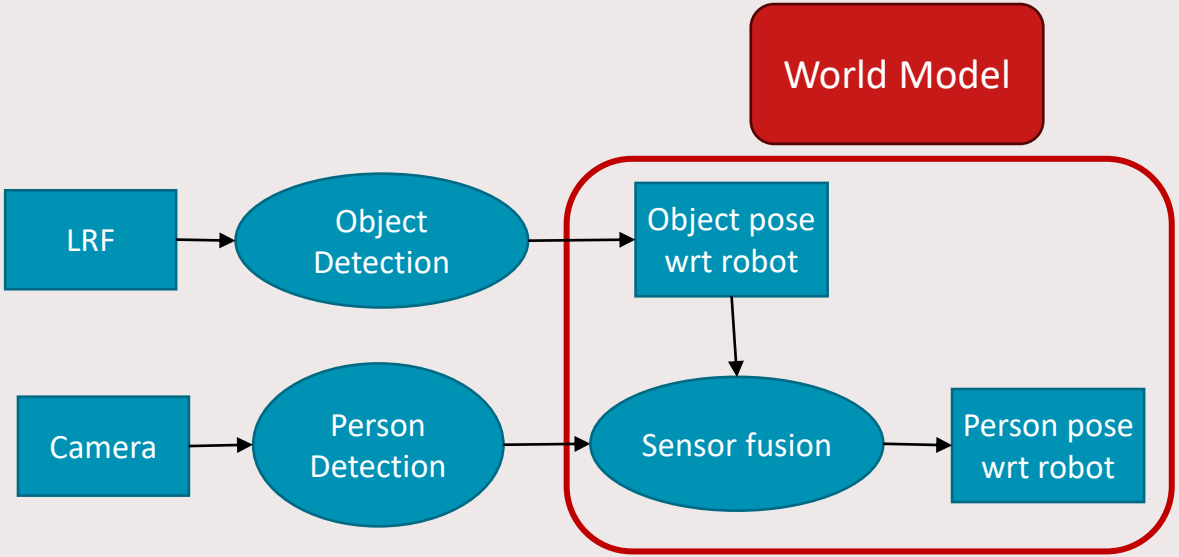
State diagram – nesting



State diagram – Cleaning robot

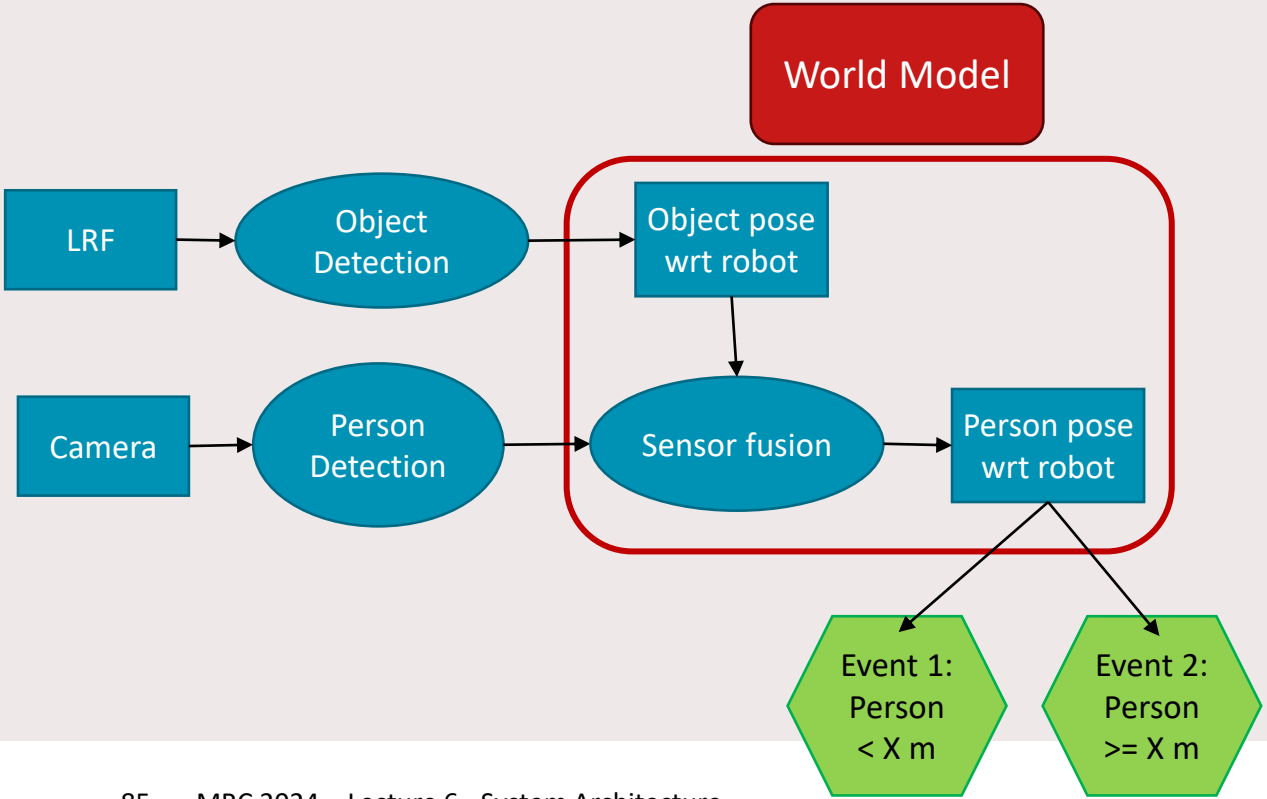


State diagram – Cleaning robot

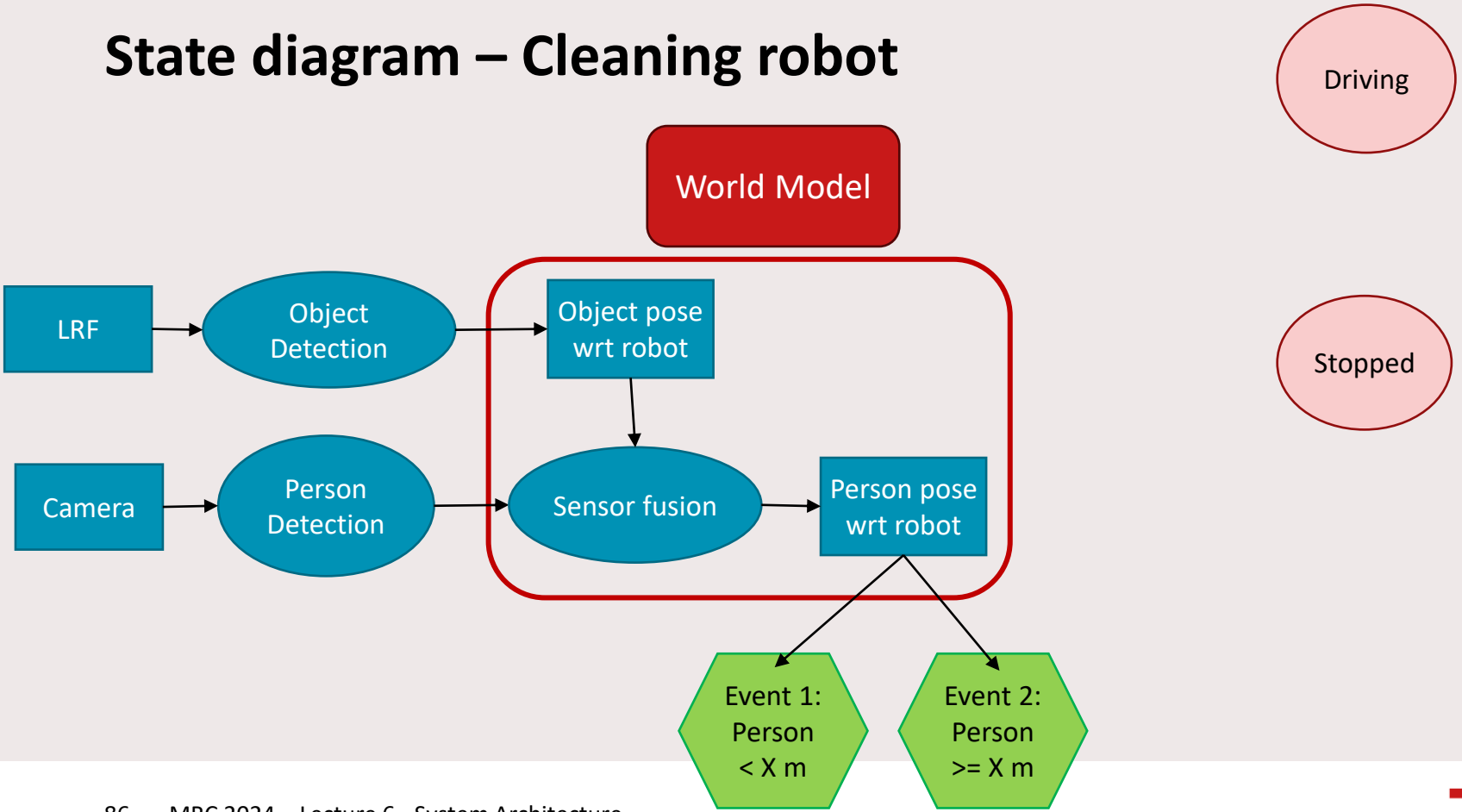


Event2

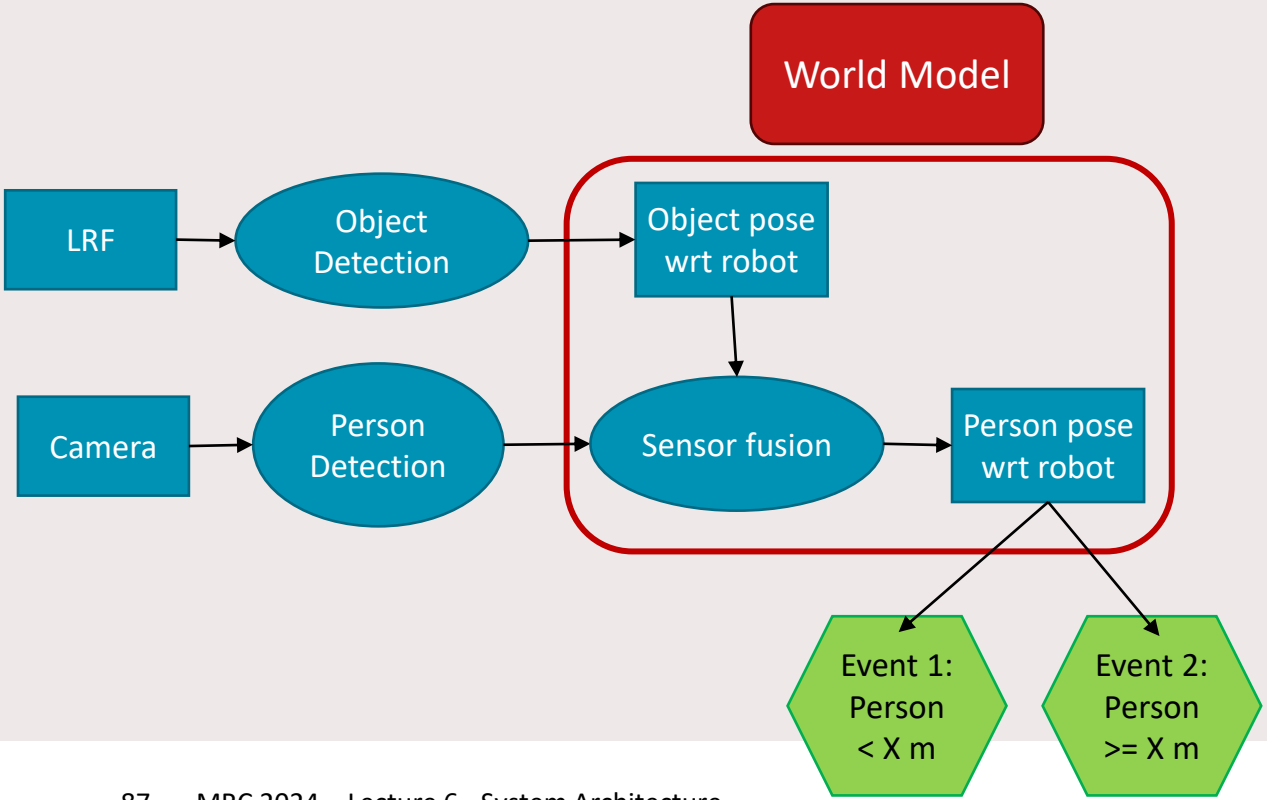
State diagram – Cleaning robot

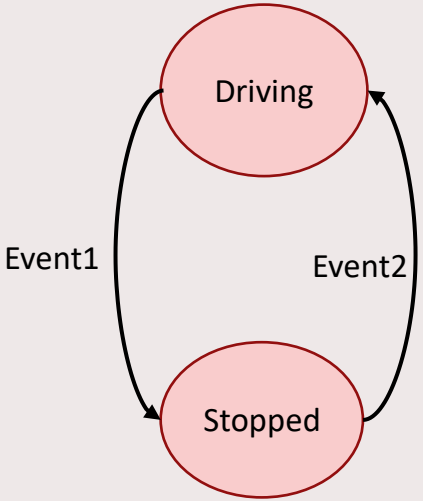


State diagram – Cleaning robot

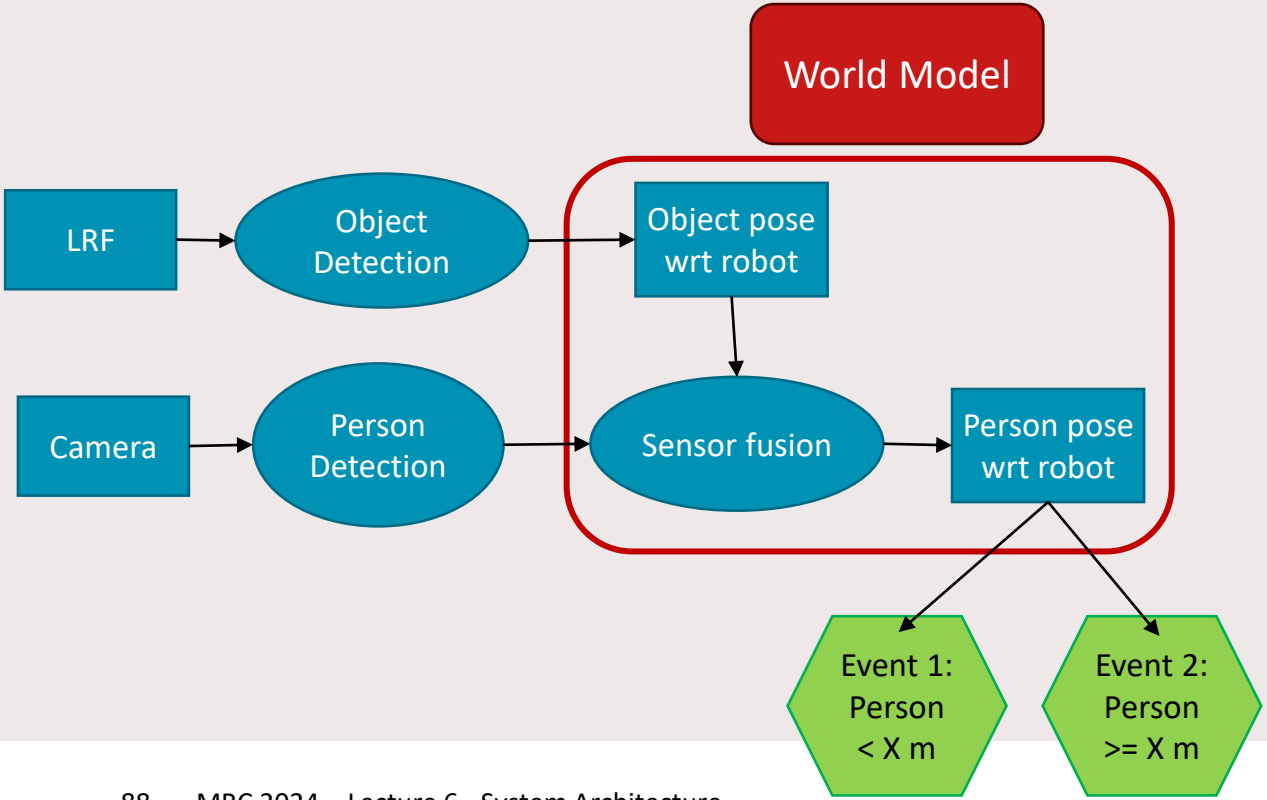


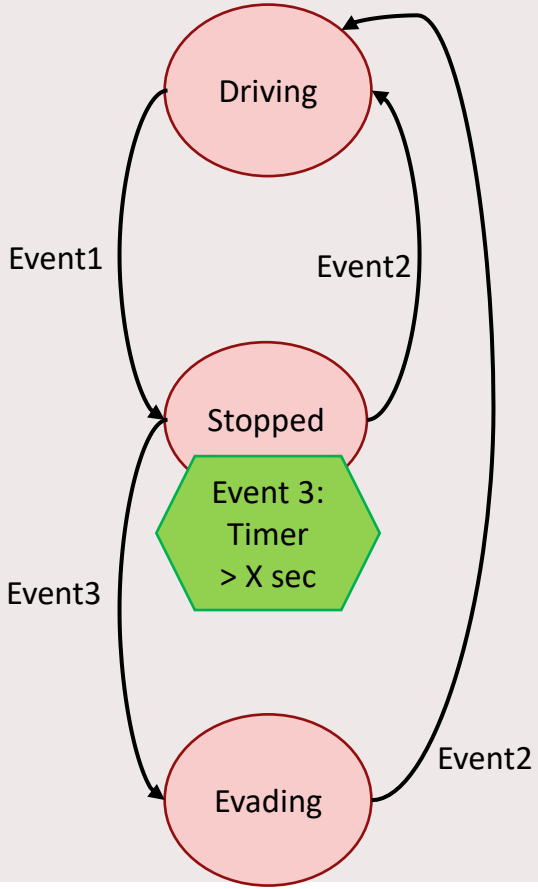
State diagram – Cleaning robot





State diagram – Cleaning robot

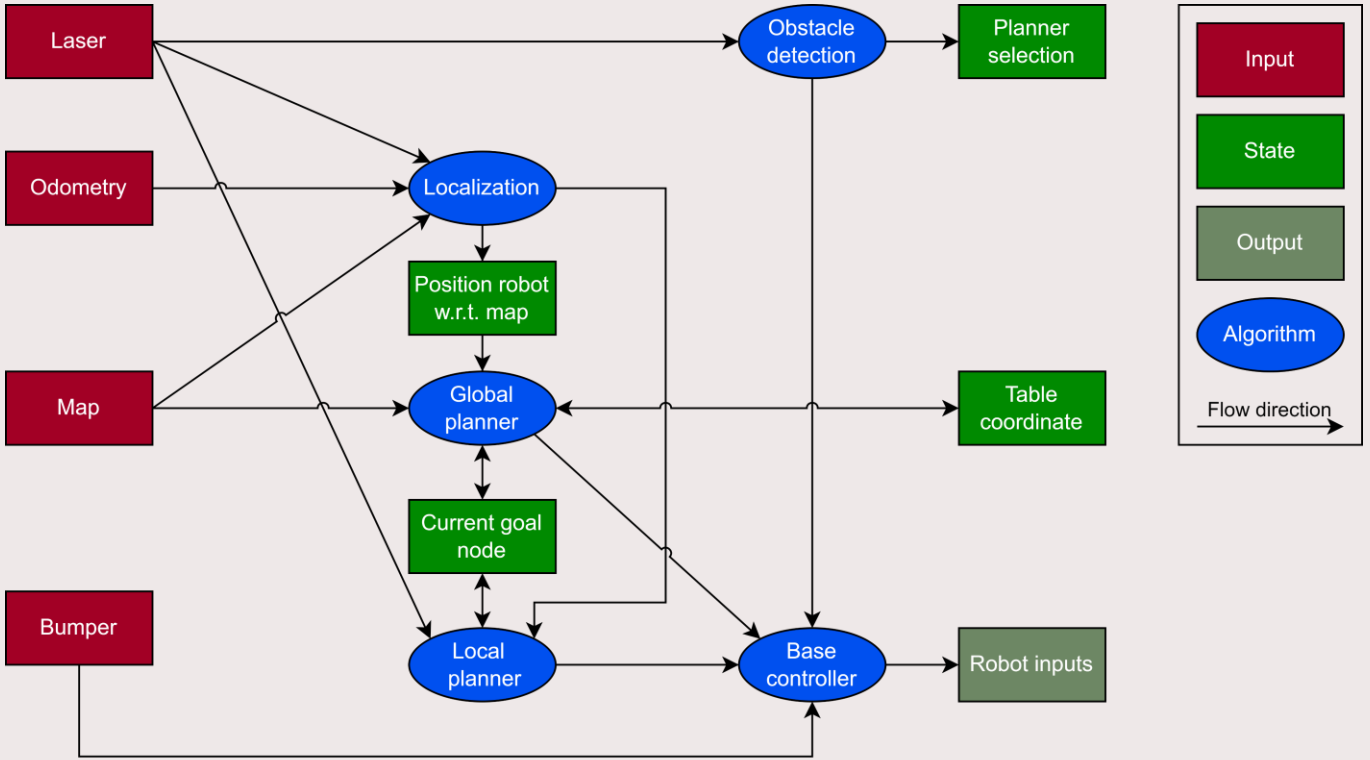




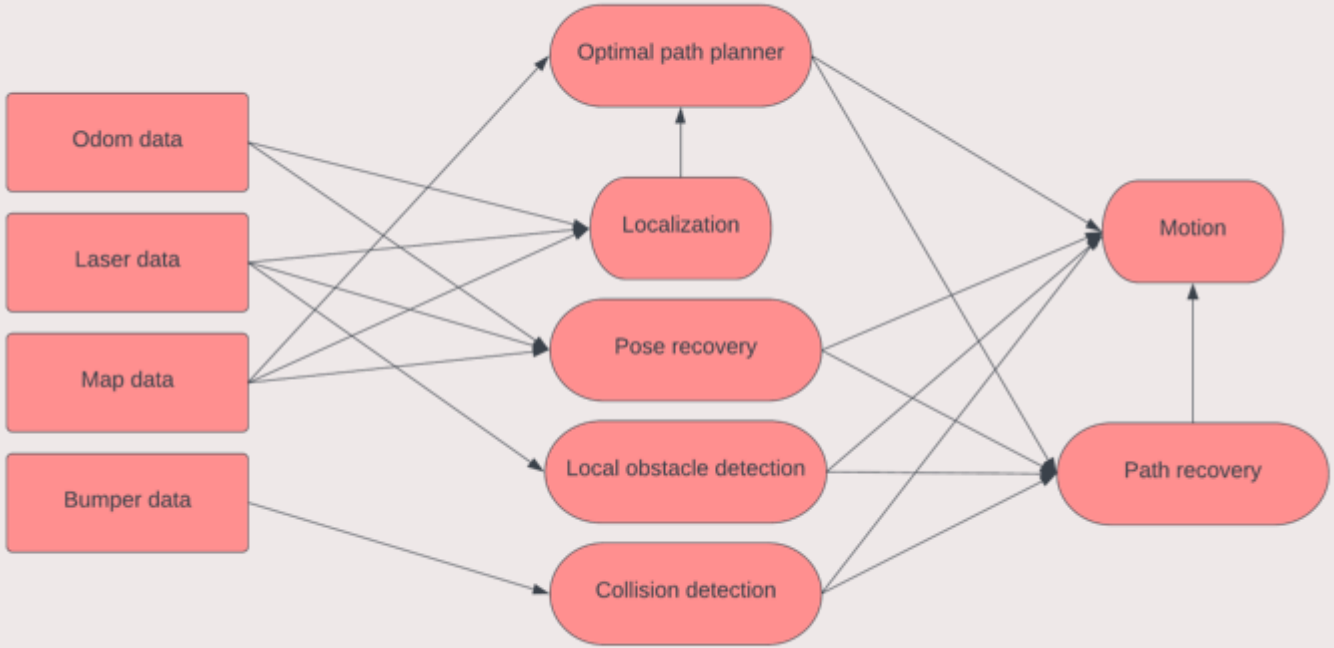
Examples previous years

Lessons to learn

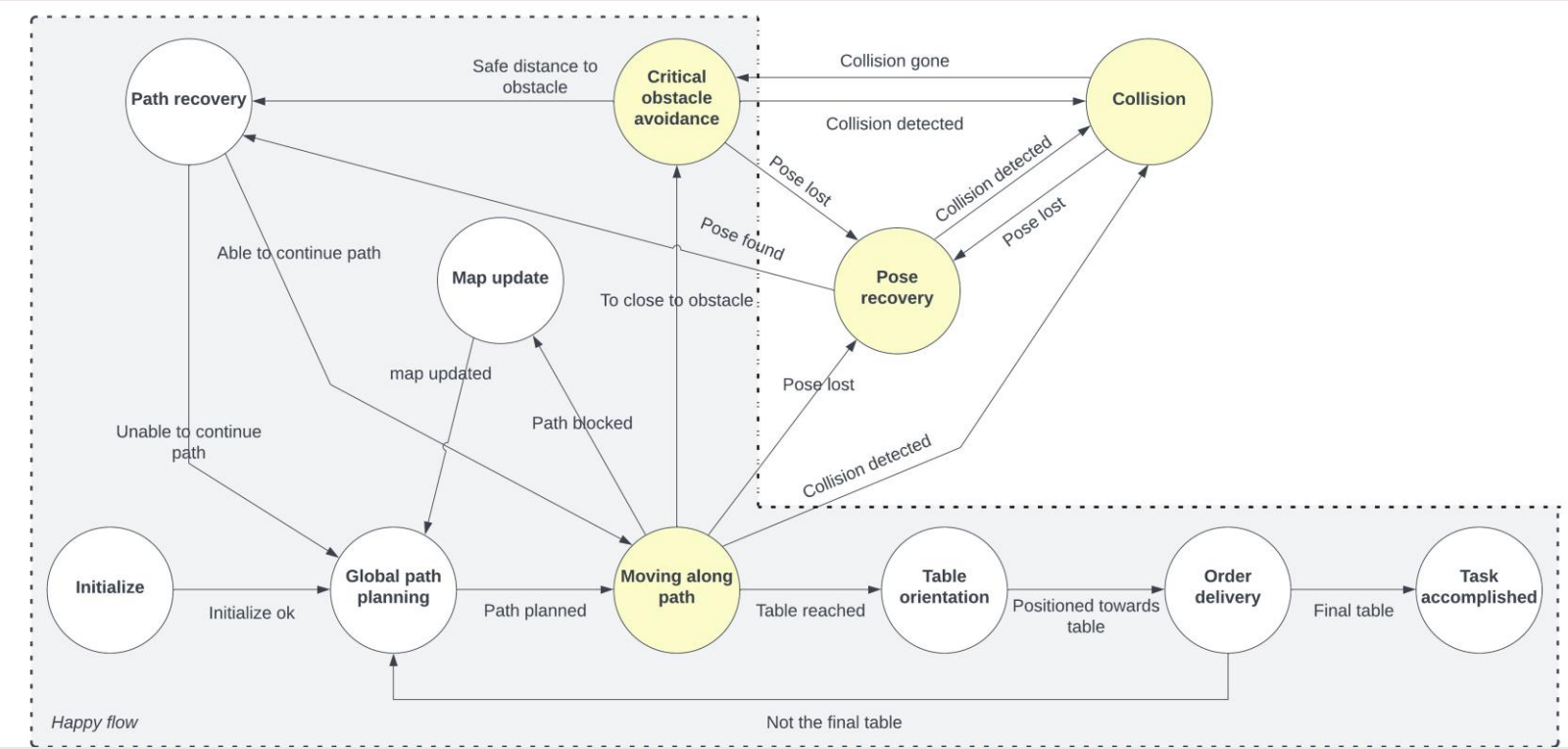
Data Flow


Laser
Odometry
Map
Bumper
Obstacle detection
Localization
Position robot w.r.t. map
Global planner
Current goal node
Local planner
Base controller
Planner selection
Table coordinate
Robot inputs
Input
State
Output
Algorithm
Flow direction

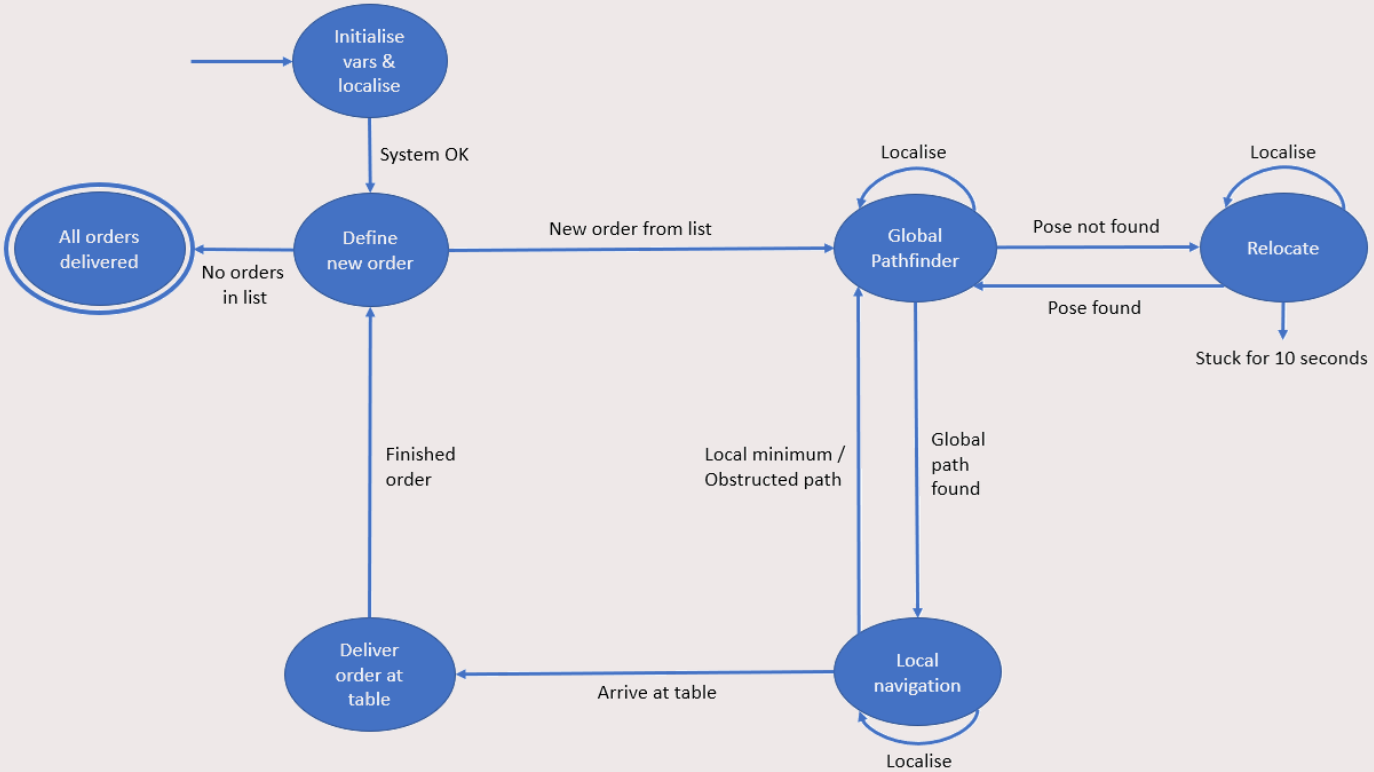
Data Flow


Optimal path planner
Odom data
Laser data
Map data
Bumper data
Localization
Pose recovery
Local obstacle detection
Collision detection
Motion
Path recovery

State Flow


Path recovery
Safe distance to obstacle
Critical obstacle avoidance
Collision gone
Collision
Collision detected
Pose lost
Collision detected
Pose lost
Pose found
Able to continue path
Map update
To close to obstacle
Pose recovery
map updated
Pose lost
Unable to continue path
Path blocked
Collision detected
Initialize
Initialize ok
Global path planning
Path planned
Moving along path
Table reached
Table orientation
Positioned towards table
Order delivery
Final table
Task accomplished
Happy flow
Not the final table

State Flow


Initialise vars & localise
System OK
All orders delivered
No orders in list
Define new order
New order from list
Localise
Global Pathfinder
Pose not found
Localise
Relocate
Pose found
Stuck for 10 seconds
Finished order
Local minimum / Obstructed path
Global path found
Deliver order at table
Arrive at table
Local navigation
Localise

Take-home messages

Take-home

Start thinking about *Integration*: decompose the system/software
Close your laptop and start drawing

What do you *magic numbers* represent?
Don’t argue about the value (outcome), discuss the models (how)

Link your diagrams, they represent 1 system

Explaining your system’s behavior, is better than not having a clue why you won.

Test, test, test…and re-itterate on your design

Design Presentations

Design Presentation

Structure:
7 minutes presentation + 4 minutes questions (STRICT!)

Short pitch:
Focus on essentials
“Sell” us your system

Be on time
All groups should (be) present

Design Presentation

Content:

- High level description of program
- Which components have you used and why?
- Analysis of the final challenge:
  What are the challenges?
  How are these reflected in your progam/components?
- System design models:
  - Requirements
  - Stateflow
  - Dataflow

RBT Promotion

If you like robotics and systems thinking, connect!

Master projects

- Mobile- and manipulation robots
- Soft robotics
- World Modelling

New paradigm: Task-Skill-Resource

- V-model and CAFCR as a functional system architecture (rather than a conceptual process model)

PhD in the RBT-group (4 years)
EngD (Engineering Doctorate) (2 years)
e.g. Mechatronic Systems Design

j.p.f.senden@tue.nl