

UNIVERSITY OF OSLO

BedreFlyt

Improving Patient Flows through
Hospital Wards with Digital Twins

Riccardo Sieve

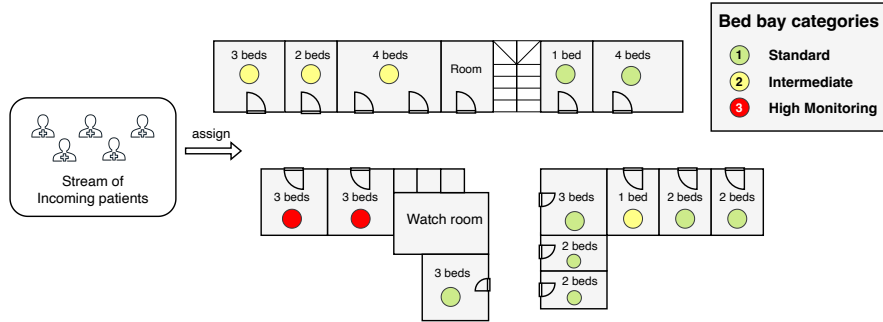
riccasi@ifi.uio.no

ASQAP

May 4, 2025

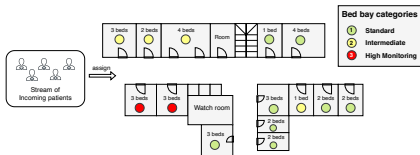


Why do we need it?



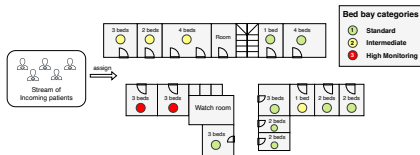
Why do we need it?

- Bed allocation is complex



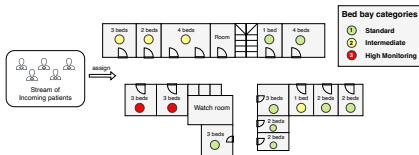
Why do we need it?

- Bed allocation is complex
- Admission is often handled manually



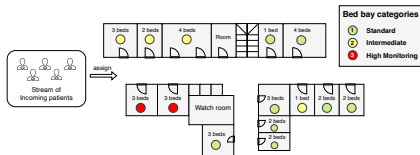
Why do we need it?

- Bed allocation is complex
- Admission is often handled manually
- Depends on several factors

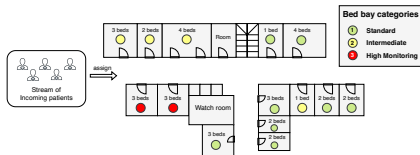


Why do we need it?

- Bed allocation is complex
- Admission is often handled manually
- Depends on several factors
- Different rooms have different monitoring categories



Why do we need it?



- Bed allocation is complex
- Admission is often handled manually
- Depends on several factors
- Different rooms have different monitoring categories
- With different simulations we can prepare for what-if scenarios

What is the goal

- Allocate patients into rooms

What is the goal

- Allocate patients into rooms
- Save intermediate steps

What is the goal

- Allocate patients into rooms
- Save intermediate steps
- Minimise changes for patients already allocated

What is the goal

- Allocate patients into rooms
- Save intermediate steps
- Minimise changes for patients already allocated
- Ensure the constraints are met

What is a Digital Twin

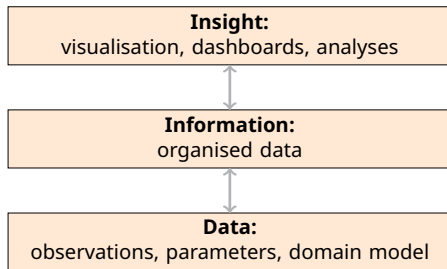
Digital twins are, as defined by NASEM

NASEM Definition of DT (2024)

A digital twin is a set of virtual information constructs that mimics the structure, context, and behavior of a natural, engineered, or **social system** (or system-of-systems), is dynamically updated with data from its physical twin, has a predictive capability, and **informs decisions** that realize value. The bidirectional interaction between the virtual and the physical is central to the digital twin^a.

^aNational Academies of Sciences, Engineering, and Medicine (NASEM) (2024): Foundational Research Gaps and Future Directions for Digital Twins. The National Academies Press

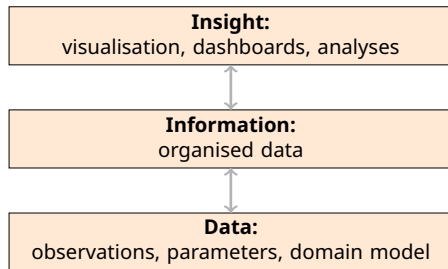
Digital Twins components



Capabilities for different insights:

- **Descriptive:** Insight into the past (“what happened”)

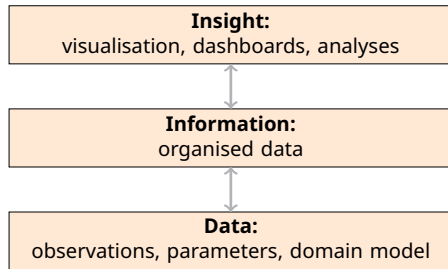
Digital Twins components



Capabilities for different insights:

- **Descriptive:** Insight into the past ("what happened")
- **Predictive:** Understand the future ("what may happen")

Digital Twins components



Capabilities for different insights:

- **Descriptive:** Insight into the past ("what happened")
- **Predictive:** Understand the future ("what may happen")
- **Prescriptive:** Advise on possible outcomes ("what if / what should")

Digital Twins components

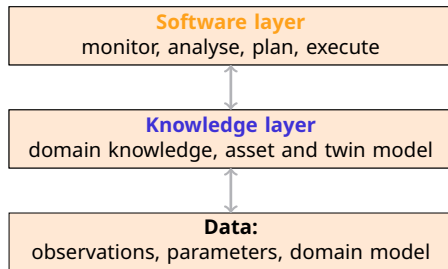
Capabilities for different insights:

- **Descriptive:** Insight into the past ("what happened")
- **Predictive:** Understand the future ("what may happen")
- **Prescriptive:** Advise on possible outcomes ("what if")

What next:

- **Reactive:** Automated decision making

Digital Twins components



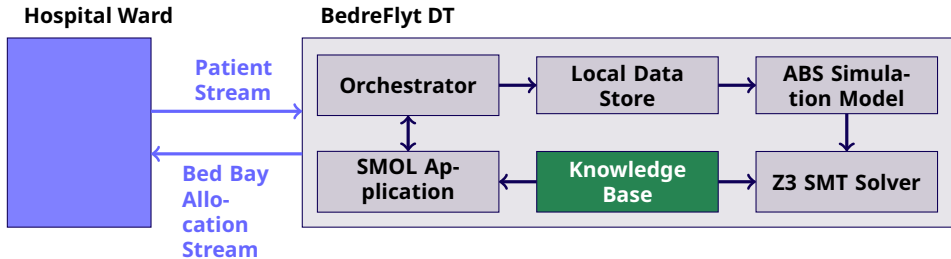
Capabilities for different insights:

- **Descriptive:** Insight into the past ("what happened")
- **Predictive:** Understand the future ("what may happen")
- **Prescriptive:** Advise on possible outcomes ("what if")

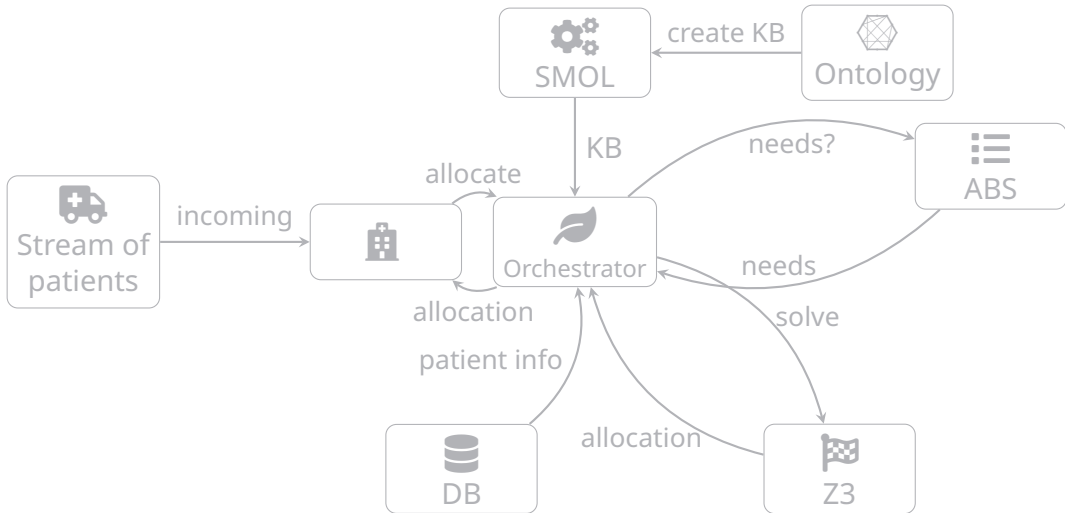
What next:

- **Reactive:** Automated decision making

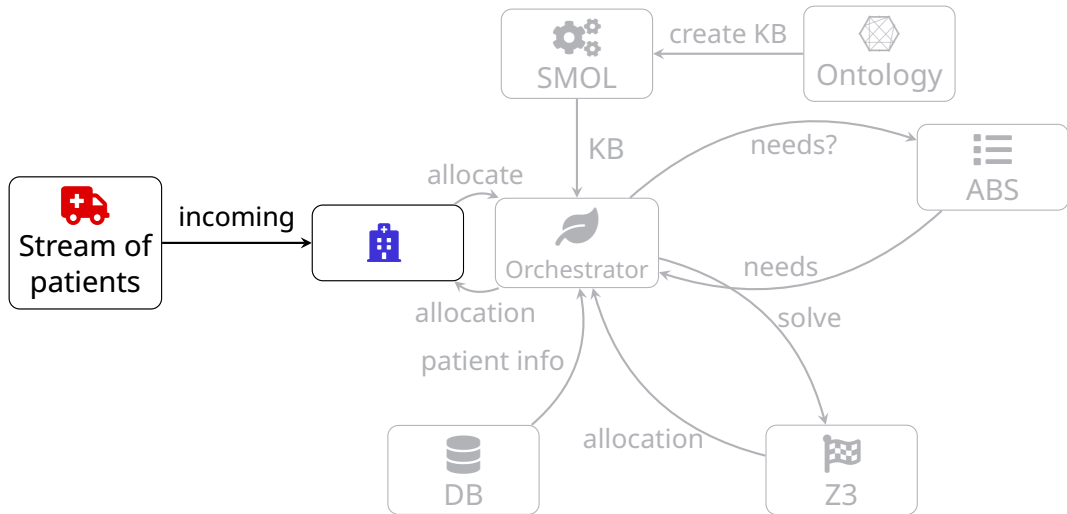
Architecture



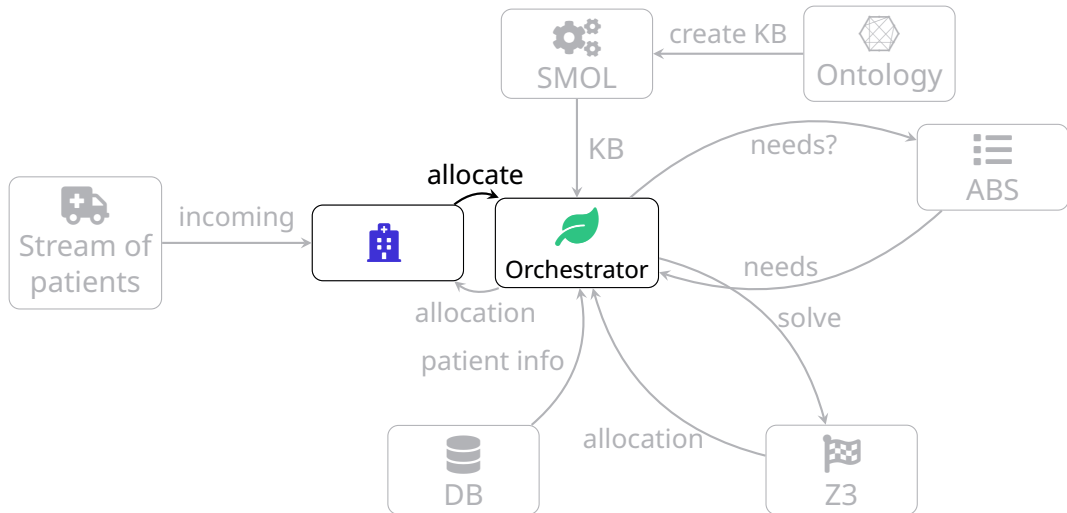
Bedreflyt DT in action



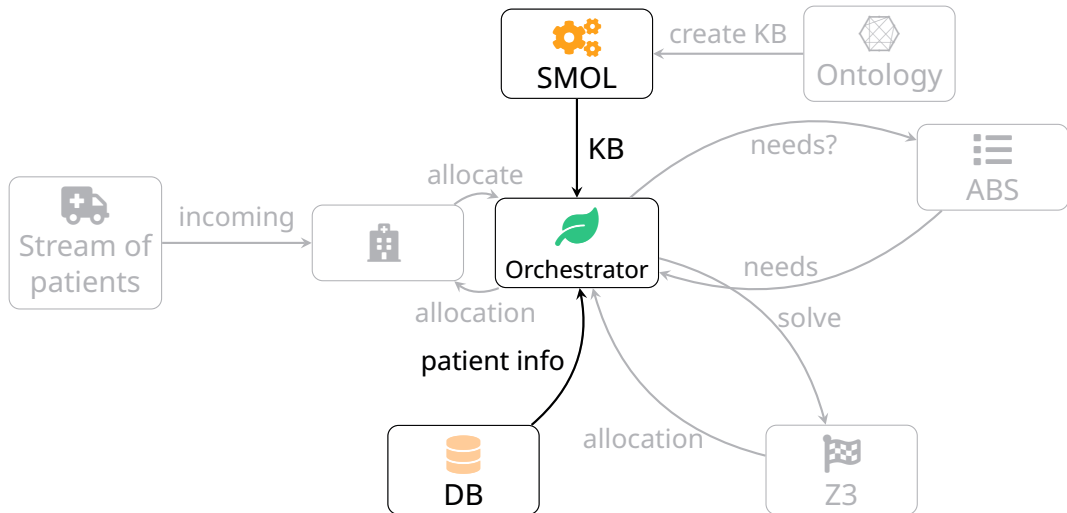
Bedreflyt DT in action



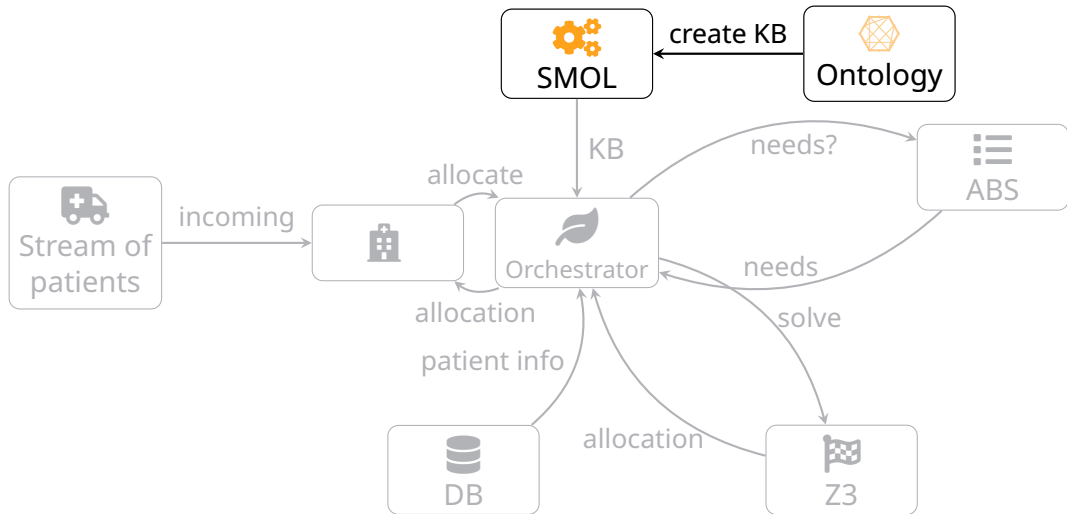
Bedreflyt DT in action



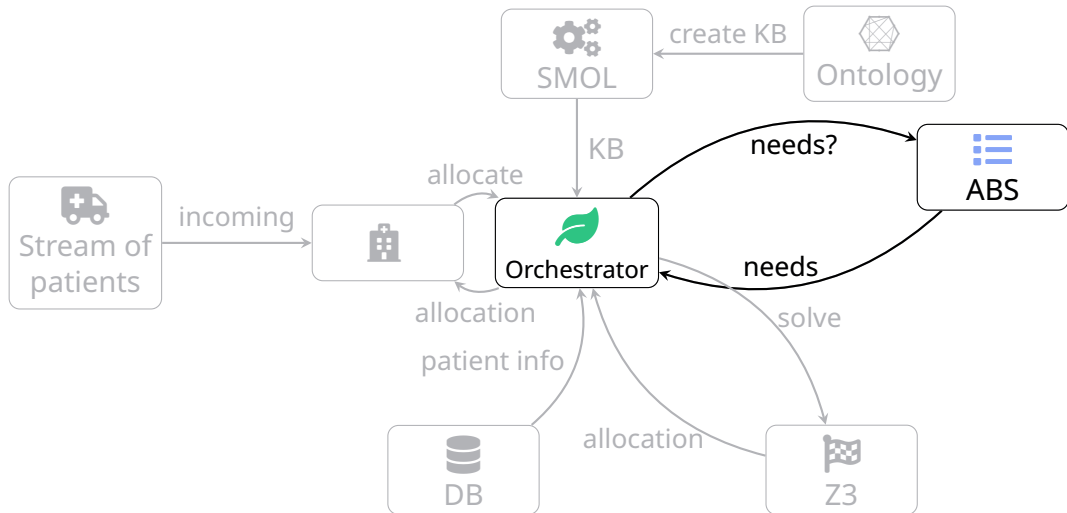
Bedreflyt DT in action



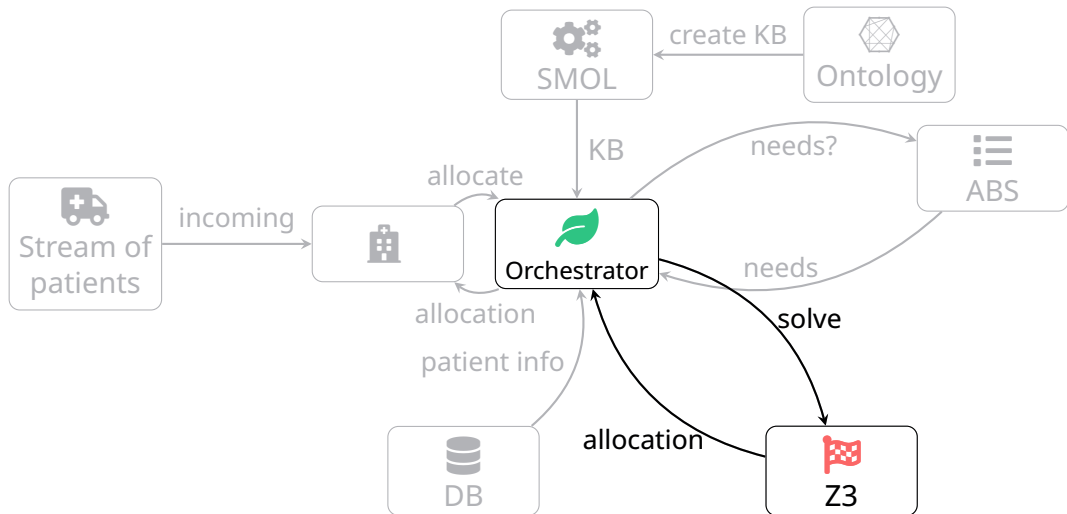
Bedreflyt DT in action



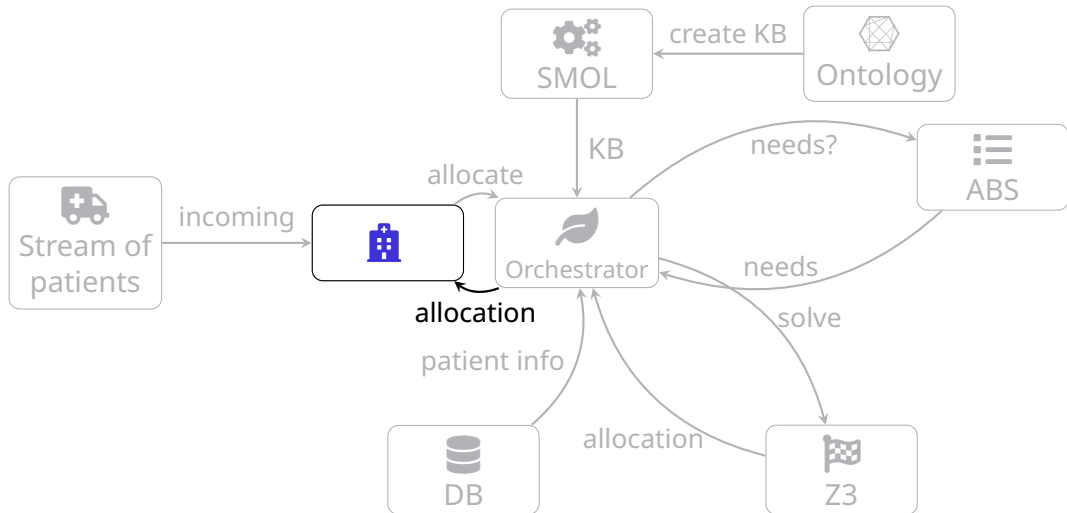
Bedreflyt DT in action



Bedreflyt DT in action



Bedreflyt DT in action



Output

Simulation Results		
Day 3		
Total changes: 0		
Room	Gender	Patients
302		
303		
304		
305		
306	Male	patient-1
307	Female	patient-2 patient-3

« < 1 2 3 4 5 6 7 8 > »

Technical components

- The simulation driver is implemented through a **Backend API** to handle the requests

Technical components

- The simulation driver is implemented through a **Backend API** to handle the requests
- The dynamic information for the allocation are saved in a **PostgreSQL** database

Technical components

- The simulation driver is implemented through a **Backend API** to handle the requests
- The dynamic information for the allocation are saved in a **PostgreSQL** database
- The data of the treatments and hospital structure is retrieved by **SMOL** to create a runtime Knowledge-Graph

Technical components

- The simulation driver is implemented through a **Backend API** to handle the requests
- The dynamic information for the allocation are saved in a **PostgreSQL** database
- The data of the treatments and hospital structure is retrieved by **SMOL** to create a runtime Knowledge-Graph
- The patient needs are computed by **ABS** on a daily basis

Technical components

- The simulation driver is implemented through a **Backend API** to handle the requests
- The dynamic information for the allocation are saved in a **PostgreSQL** database
- The data of the treatments and hospital structure is retrieved by **SMOL** to create a runtime Knowledge-Graph
- The patient needs are computed by **ABS** on a daily basis
- We solve the allocation by solving the constraints with **Z3**

Twinning in SMOL

Programming support for twins with a behavioural and a structural layer
smolang.org



Twinning in SMOL

Programming support for twins with a behavioural and a structural layer

SMOL: Semantic Model Object Language

- Smalll OO programming system
- Ontology reasoners allow querying the KB
- Used to create the digital model

smolang.org



Twinning in SMOL

Programming support for twins with a behavioural and a structural layer
smolang.org

Behavioral twins in SMOL

- SMOL can encapsulate (simulation) models based on the FMI standard
- Can automatically adapt the object states in the KG at runtime



Handling multiple scenarios

- We might want to make multiple simulations with various conditions

Handling multiple scenarios

- We might want to make multiple simulations with various conditions
- For the same diagnosis, different treatment could be selected

Handling multiple scenarios

- We might want to make multiple simulations with various conditions
- For the same diagnosis, different treatment could be selected
- We introduced the sampling into the simulation environment

Handling multiple scenarios

- We might want to make multiple simulations with various conditions
- For the same diagnosis, different treatment could be selected
- We introduced the sampling into the simulation environment
 - We set up a cost function for the different treatment

Handling multiple scenarios

- We might want to make multiple simulations with various conditions
- For the same diagnosis, different treatment could be selected
- We introduced the sampling into the simulation environment
 - We set up a cost function for the different treatment
 - For each treatment we also define the frequency with which they occur

Handling multiple scenarios

- We might want to make multiple simulations with various conditions
- For the same diagnosis, different treatment could be selected
- We introduced the sampling into the simulation environment
 - We set up a cost function for the different treatment
 - For each treatment we also define the frequency with which they occur
- When simulating we sample over that with different strategies (“worst case” and “common case”)

Future directions

- Extend the work to enable self-adaptive techniques for different wards

Future directions

- Extend the work to enable self-adaptive techniques for different wards
- Allow a more general optimization problem to add penalties for the allocation strategy

Future directions

- Extend the work to enable self-adaptive techniques for different wards
- Allow a more general optimization problem to add penalties for the allocation strategy
- Extend the architecture to use MDPs or Gurobi for optimisation

Future directions

- Extend the work to enable self-adaptive techniques for different wards
- Allow a more general optimization problem to add penalties for the allocation strategy
- Extend the architecture to use MDPs or Gurobi for optimisation
- Include different components to simulate (e.g. staff management)

Future directions

- Extend the work to enable self-adaptive techniques for different wards
- Allow a more general optimization problem to add penalties for the allocation strategy
- Extend the architecture to use MDPs or Gurobi for optimisation
- Include different components to simulate (e.g. staff management)
- Set up a live stream of patients from the hospital)