

# Integration mRUBiS with external Controller

Slides based on  
Paul Wullenweber  
Kim-Pascal Borchart  
August 3rd, 2021

Machine Learning-based Control of Dynamical Systems

# Requirements

---

- Launch mRUBiS
- Communicate from external process
- Observe mRubis states
- Assign the order of fix execution

**Controlling  
mRUBiS from  
Python**

# Communication – observing states

---

Issues handled individually

Encapsulation

Complex Architecture

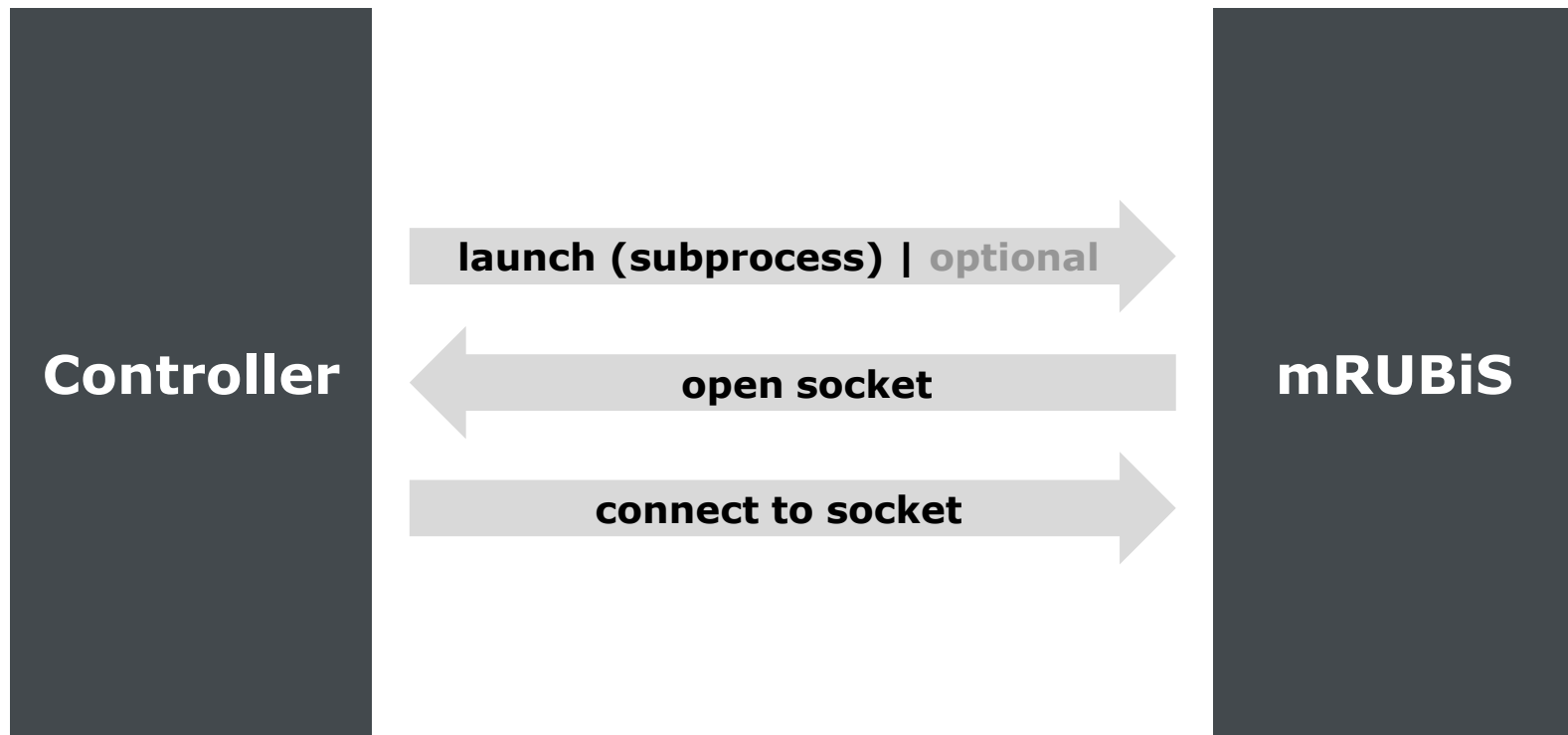
→ **quick solutions:**

- persist required data to disk (json)
- copy individual issue flow

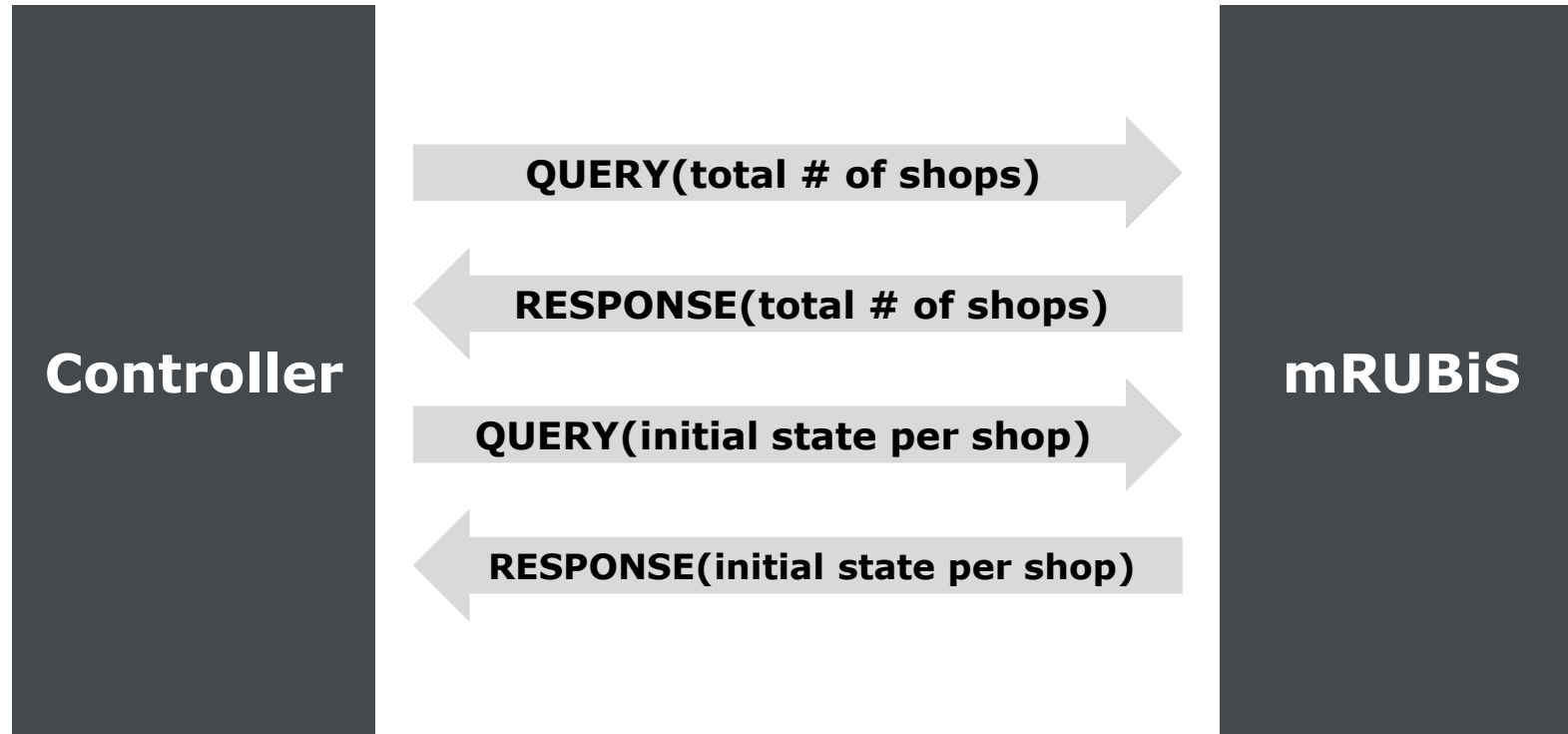
A large, dark gray vertical rectangle with the text "mRUBiS" in white, bold, sans-serif font centered within it.

**mRUBiS**

# Communication – initial connection



# Communication – initial states



# Communication – within run

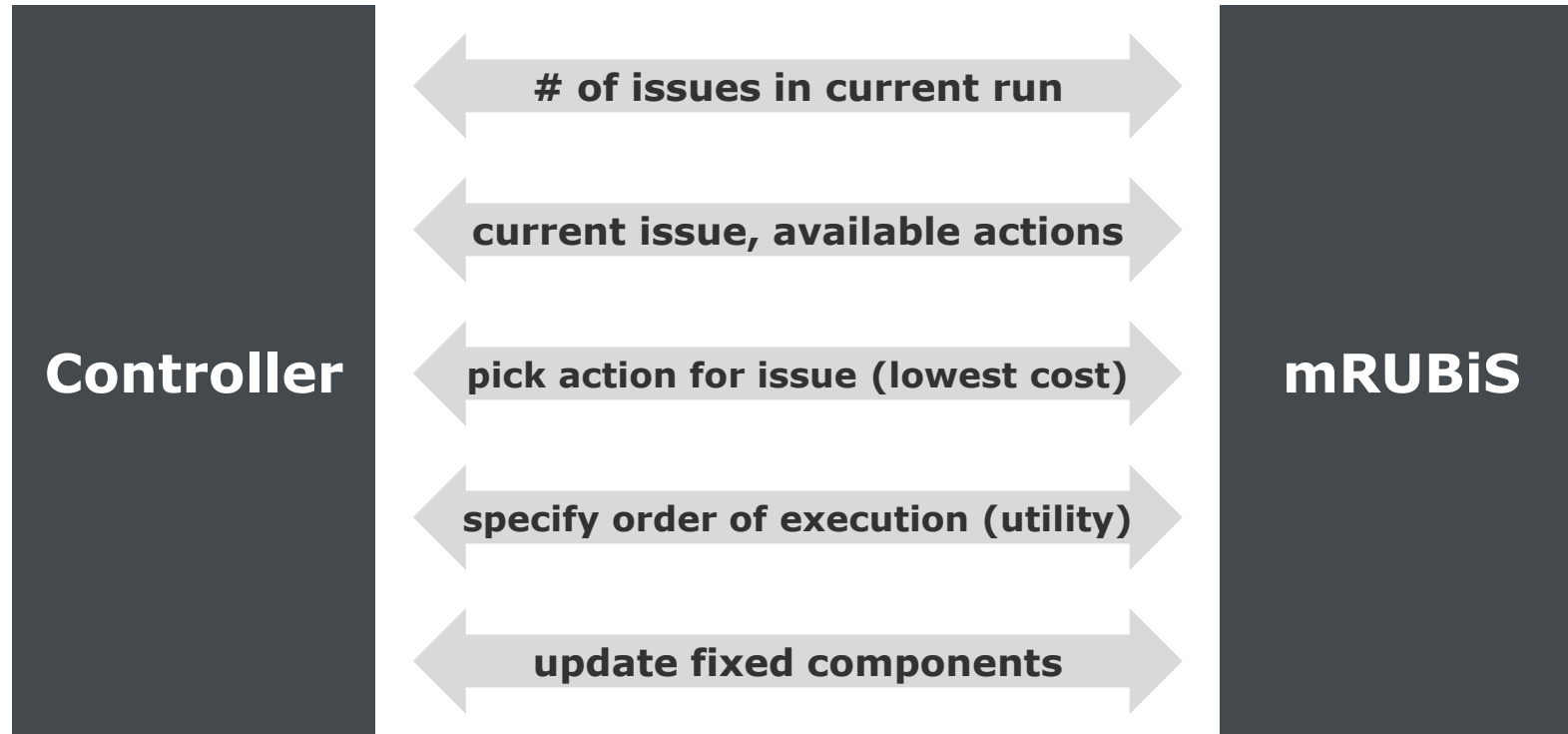
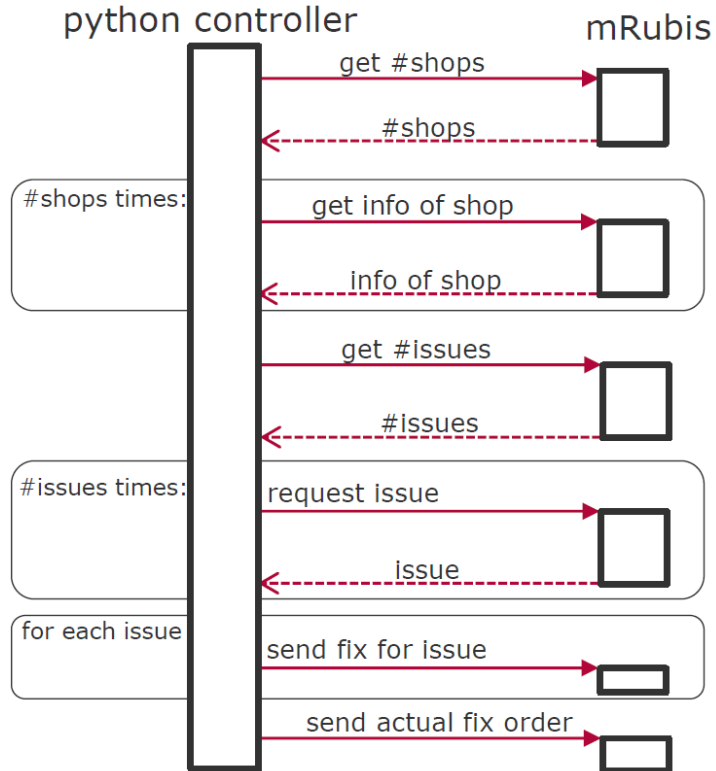


Chart 6

# Sequence Diagram



GitHub Repo:

<https://github.com/hpi-sam/MachineLearningControl>

**Presentation Title**

Kim-Pascal Bochart

Chart 7

# Executing Control Strategies

---

- Which action to take?
  - Compare available actions per issue
- Which issue to address first?
  - Compare components within and between shops

**Controlling  
mRUBiS from  
Python**



# Three Control Strategies

---

1. Random issue ordering
2. Cost-based issue ordering
3. Predicted-utility-based issue ordering

**Controlling  
mRUBiS from  
Python**

# Issue ranking – example run

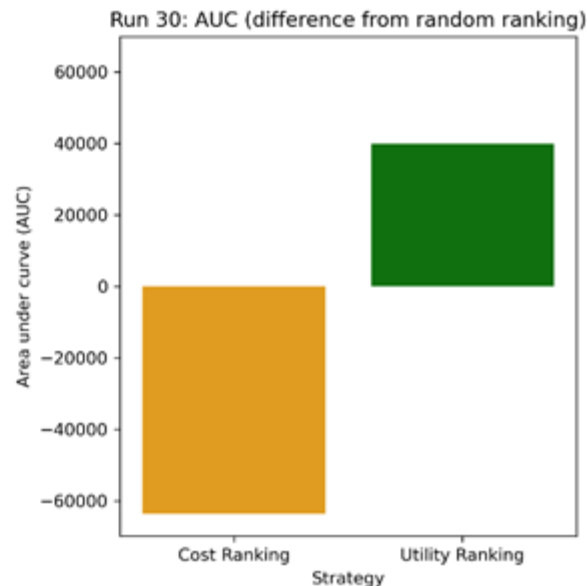
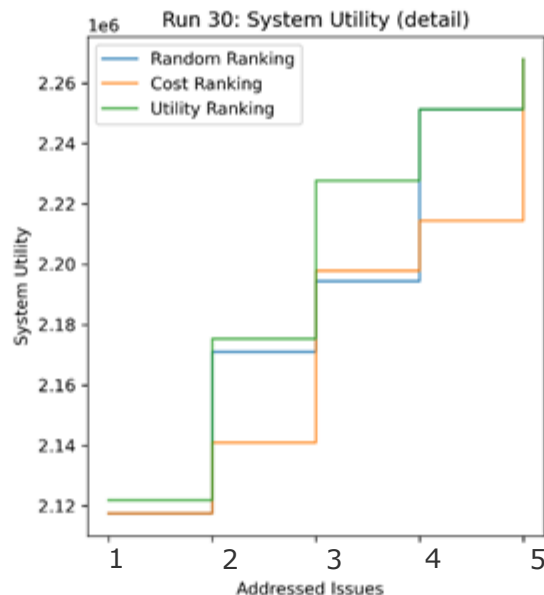
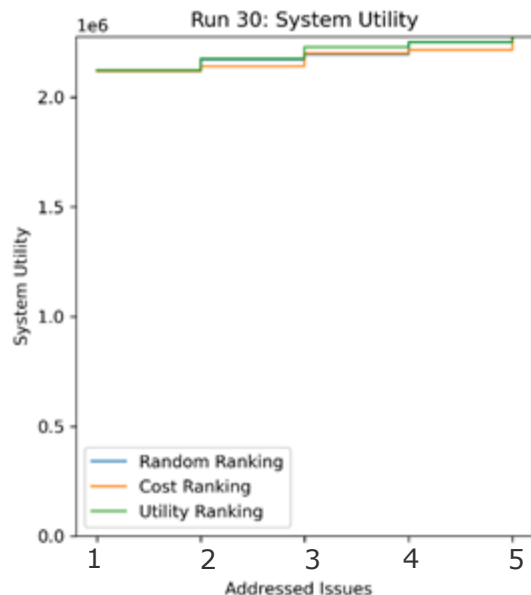


Chart 10

End