SiLA Rapid Integration

Pick & place labware transportation with benchtop and mobile robots
Lab robot use cases and how to address them – Technical session

4th bioSASH (BioLAGO – SiLA 2/AnIML Serial Hackathon), Konstanz, Germany

29.SEP.2022

Ádám Wolf

PhD Student, Takeda
Working Group Lead, SiLA Robotics

The SiLA Robotics Working Group

The SiLA Robotics Working Group

SiLA Rapid Integration

Organization

- Domain-specific working group
- Reports back to the core WG
- Open group
- Bi-weekly meetings

Mission

- Combine existing established technologies in a comprehensive framework
- SiLA as the central element of the tech stack
- Unify, scale-up and extend functionality
- Incorporate new concepts
- Facilitate exchange in the lab robotics community

Vision

- Foster the SiLA-based plug & play integration of lab robots
- Unify the communication standard
- Provide vendor-independent solutions

Milestones

- SiLA-ROS interface
  - Hackathons
- Unify feature definitions
  - Structure to incorporate present and future lab robot capabilities
  - Identify candidates as standard definitions for specific capabilities
- Reference implementations
  - TIAGo □ Panna
  - MIR + UR
  - **4th BioSASH Hackathon**
- Incorporate new concepts (LAPP)
  - Digital twin
  - Robotic action templates
  - Labware library
  - Advanced robotic technologies: Perception, manipulation, Human-machine collaboration

Bi-weekly meetings on-going

- Discussions: workflow representations, labware ontologies, etc.
- Contact adam.wolf@sila-standard.org to join

Preparation

Pre-read material

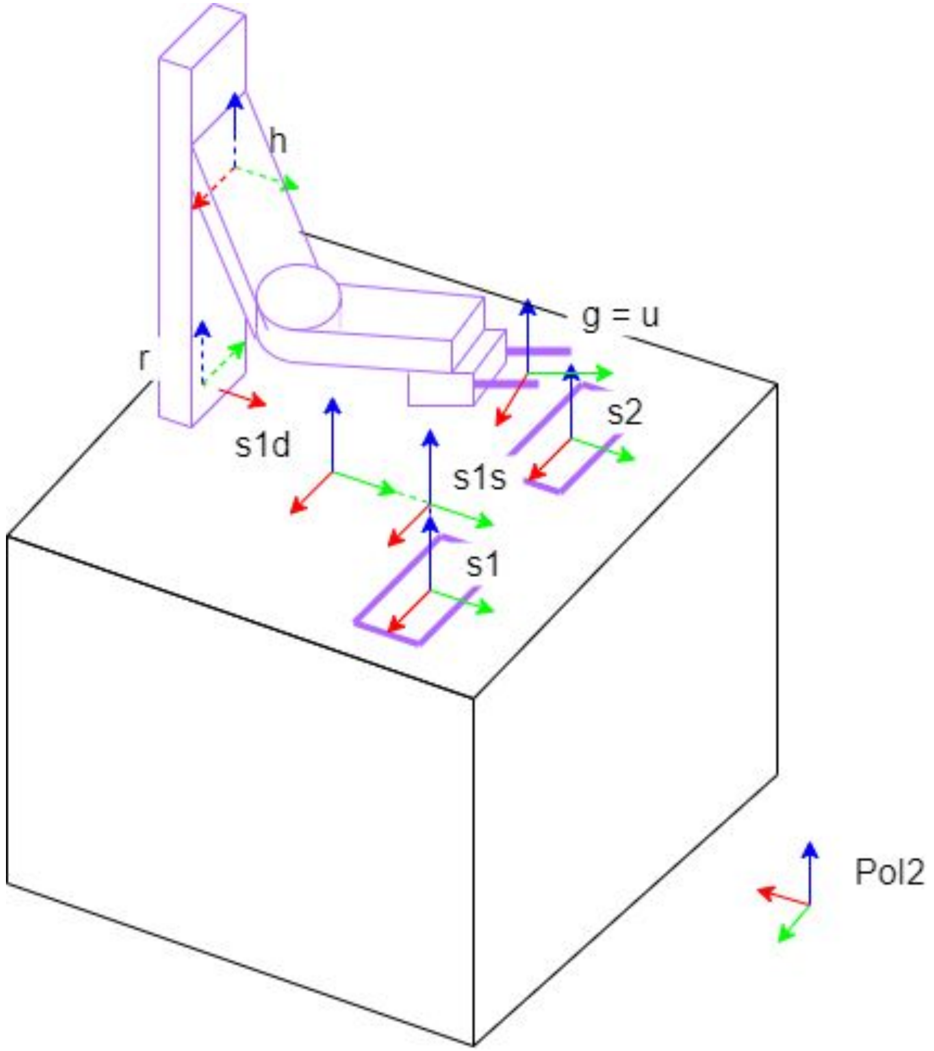
- Intro to SiLA (Presentation recording)
- SiLA Part (C), Chapter Robots (integration guide, draft)
- Feature definition and corresponding documentation
- Robot interface documentations
  - PreciseFlex
  - Universal Robots, RTDE

The scope of BioSASH #4

The “What?”

Goals of the Robotics hackathon working group

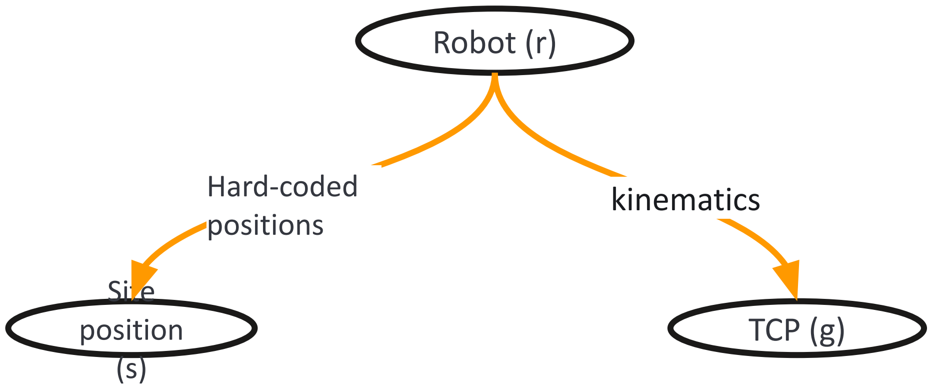
- Reference implementation of the **labware transfer** feature for different benchtop robot arms
  - Based on the LabwareTransferController feature definition
- Implement an exemplary labware transfer action, where the robot picks a plate from one device and places it in another device
  - Passive dummy devices with **fixed** site (aka nest) **positions**



Technical consideretions

The “How?”

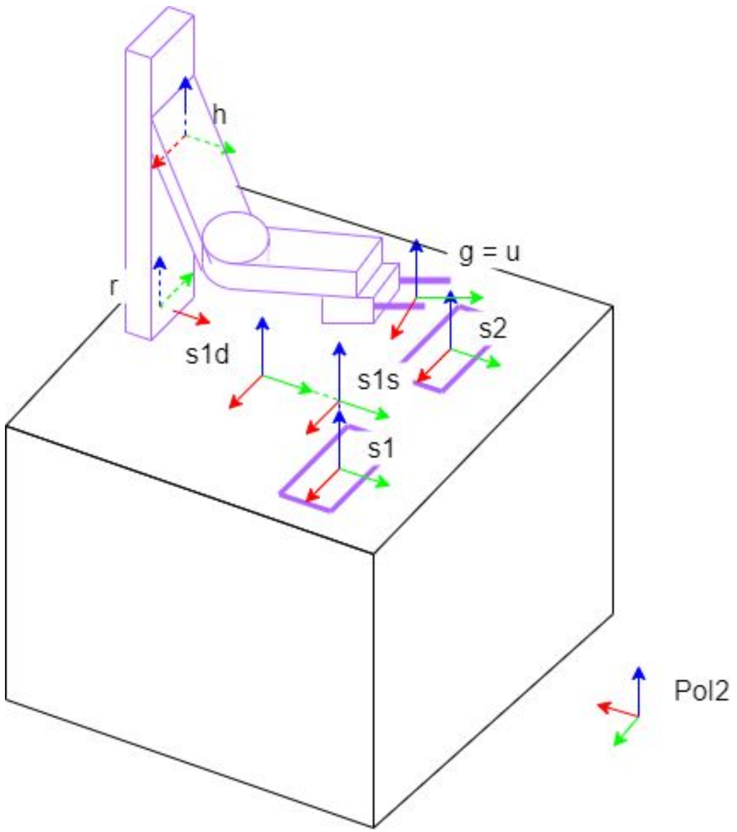
Position representation for stationary robots with the LAPP DT



Legend

Live, robot-level, not exposed towards SiLA

- Positions defined in robot coordinate system
- Defined by on-line programming (manual teaching)

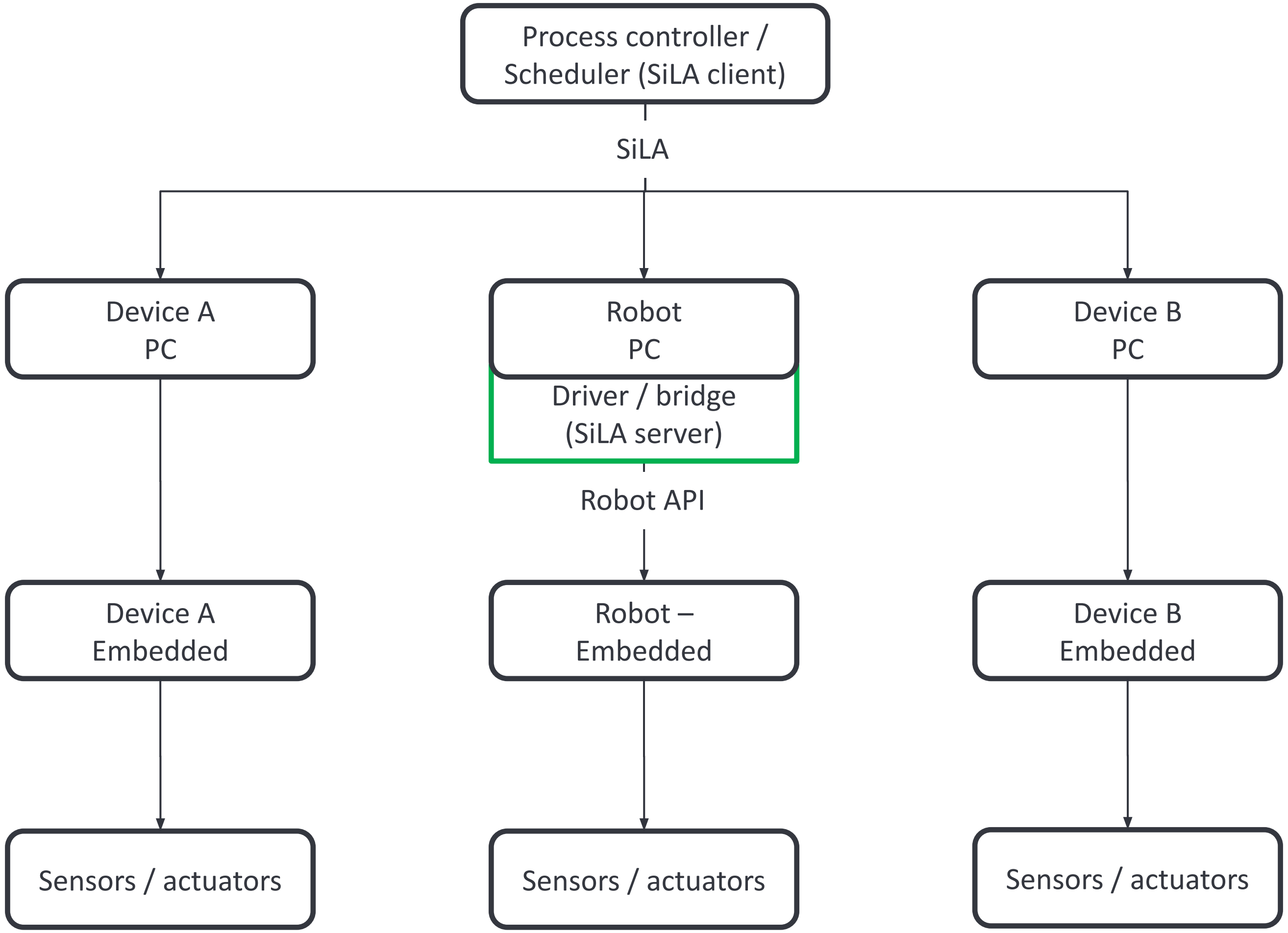


Hierarchical levels of workflow representation and control architecture

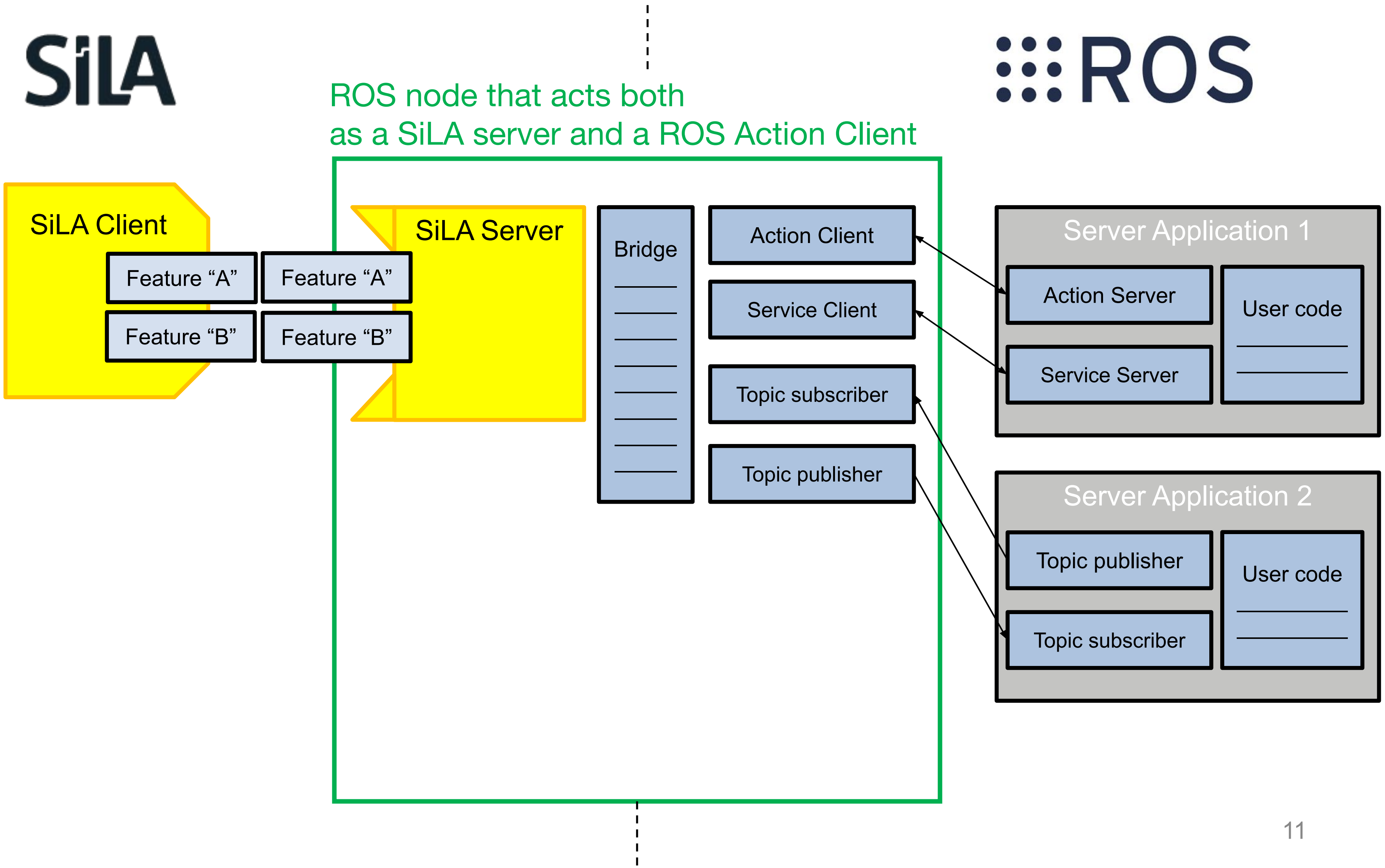
SiLA Rapid Integration

<table>
<tr><th colspan="2">Process</th><th colspan="2">Protocols and languages</th><th colspan="2">Control architecture</th></tr>
<tr><th>Level nr</th><th>Level name</th><th>Liquid handling</th><th>Robotics</th><th>Liquid handling</th><th>Robotics</th></tr>
<tr><td>7</td><td>Service</td><td colspan="2">Service protocol</td><td colspan="2">Lab management (LIMS, LES)</td></tr>
<tr><td>6</td><td>Procedure
(Experiment / assay)</td><td colspan="2">Experiment design language
Laboratory process language</td><td colspan="2">Automation Scheduler (E.g. GBG, niceLabs, PharmaMV)</td></tr>
<tr><td>5</td><td>Task</td><td>High-level liquid handling script</td><td rowspan="2">Modular robot program</td><td rowspan="4">EVO PC</td><td rowspan="4">Robot controller PC</td></tr>
<tr><td>4</td><td>Subtask</td><td rowspan="2">Low-level liquid handling script</td></tr>
<tr><td>3</td><td>Motion sequence</td><td rowspan="2">Low-level robot program</td></tr>
<tr><td>2</td><td>Motion primitive</td><td rowspan="2">Device firmware</td><td rowspan="2">Embedded controller</td><td rowspan="2">Robot controller</td></tr>
<tr><td>1</td><td>Actuator primitive</td><td>Joint trajectories, IO control</td></tr>
</table>

System architecture



Recap – SiLA ROS bridge



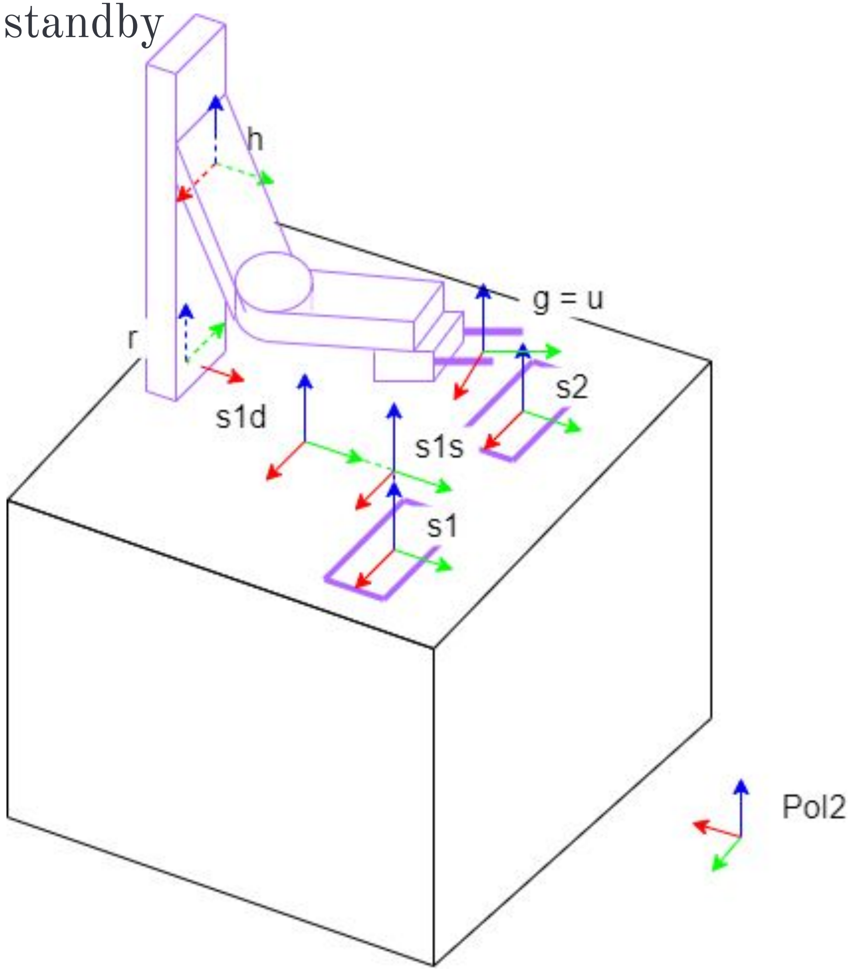
The program sequence (presented in detail by Stefan Koch)

The sequence (primary focus set in **bold**)

- Device1: PrepareForOutput
- Robot: PrepareForInput
- **Robot: GetLabware**
- Device1: LabwareRemoved
- Device2: PrepareForInput
- Robot: PrepareForOutput
- **Robot: PutLabware**
- Device2: LabwareDelivered

The motion sequence

- The robot starts in its home position (h)
- Performs a MoveJ-type movement to "untangle" itself and arrive to the standby conguration at the position u)
- Moves linearly (MoveL) to a device-approach position (s1d)
- MoveL to a site-approach position (s1s)
- MoveL to the nal site hand-over position (s1)
- Grips the plate
- MoveL back to s1s
- MoveL back to s1d
- MoveL to s2d (not displayed)
- MoveL to s2s (not displayed)
- MoveL to s2 (not displayed)
- Releases the plate
- MoveL back to s2s
- MoveL back to s2d
- MoveL to u and returns to standby/ready state



Time plan – Day one

29.09. – Get acquainted with the framework together

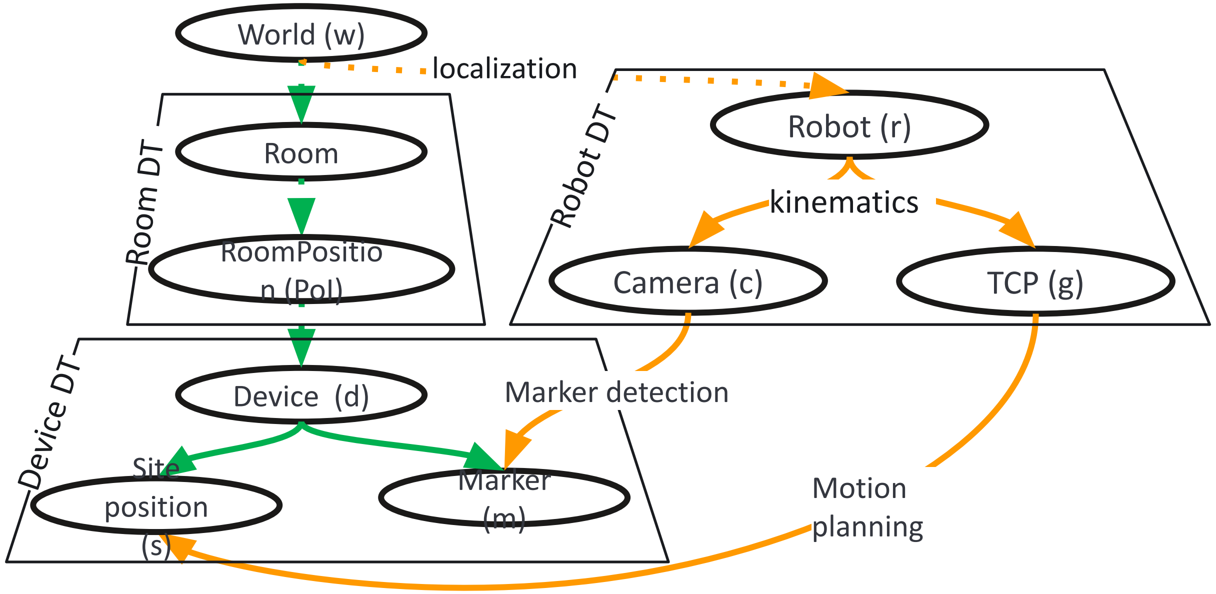
- 1h – Introduction
    - 15 min –Challenge (Ádám Wolf)
    - 30 min –The feature definition, the sequence and the logic (Stefan Koch)
    - 15 min –Development environment (Johannes Waidner, Ádám Wolf)
- 10 min – Personal introductions
- 1 h – Setup of personal environments and/or SSH access
- 1 h – SiLA Hello world
- 1 h – Get familiar with the robots and their API
- Open till 11pm – Proceed with next steps (listed on next slide)

Time plan – Day two

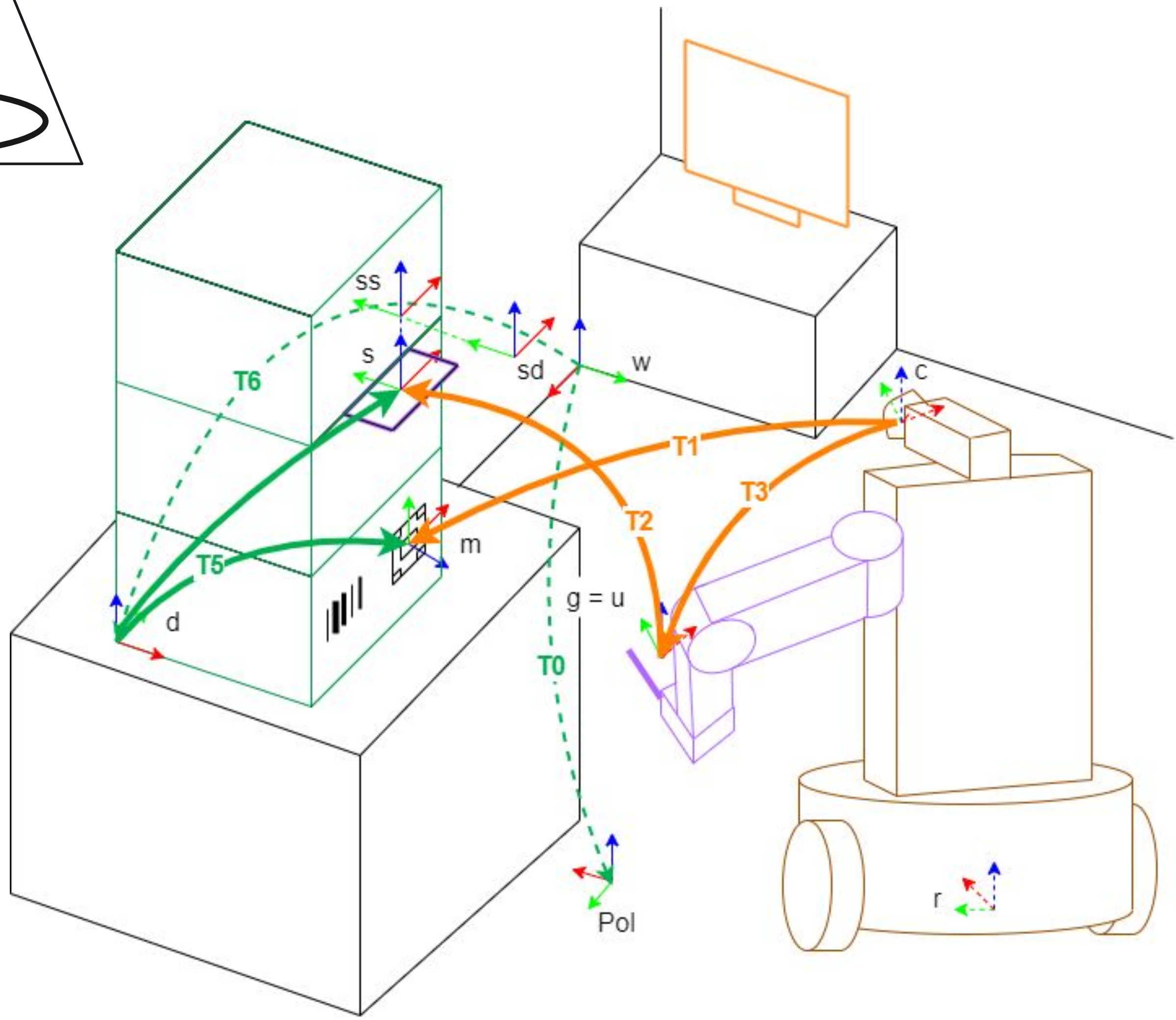
30.09. – Break down to smaller groups of 4-5 people to work on a specific robot

- Teach and store robot positions for the (dummy) devices and sites
  - SiLA positions
    - Named site positions: HandoverPosition
  - Intermediary robot positions
    - Site
    - Site-approach
    - Device-approach
- Generate the skeleton based on the feature definition, using the sila_python code generator or a similar tool (e.g. for C#)
- Implement the SiLA driver on the robot-dedicated PC
  - SiLA server communication based on the generated skeleton
  - Communication with the robot via specific API
- Implement SiLA commands
  - GetLabware
  - PutLabware
- Test the labware transfer action
- Optional: store the positions in a sharable format/database
   preparation for LAPP PoC

An outlook:
Position representation for mobile robots with the LAPP DT



- Top-down position definitions
- Stored in parent



Legend

Live, robot-level, not exposed towards SiLA

Stored in the LAPP DT
Represented as high-level SiLA properties (references)

Transformation originates from inaccurate base odometry

Transformation originates from accurate sources
- robot kinematics
- marker detection
- positions stored in the digital twin