University of St.Gallen

TAPAS Project – Final Presentation

St.Gallen, 18th December 2023
Group 5

From insight to impact.

The Team

Kaan Aydin

Michael Brülisauer

Daniel Leal

Stephan Nef

Agenda

University of St.Gallen

- Key Architecture Changes
- Communication Flows
- Decoupling
- Demonstration
- Outlook & Contributions

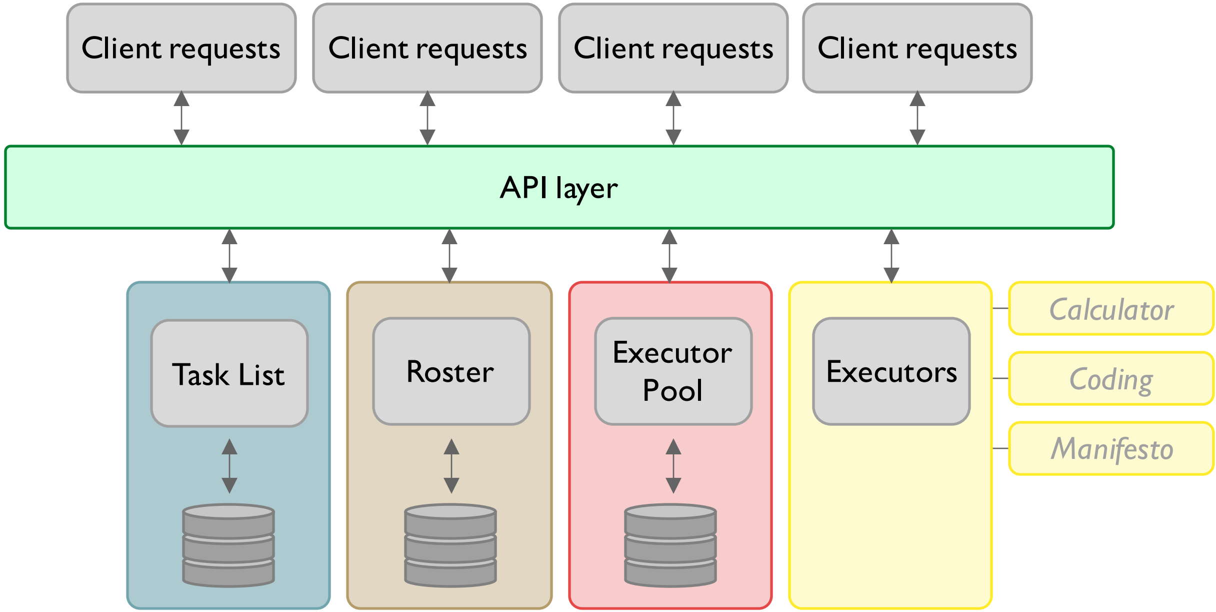
Agenda

- Key Architecture Changes
- Communication Flows
- Decoupling
- Demonstration
- Outlook & Contributions

Major improvements since last presentation

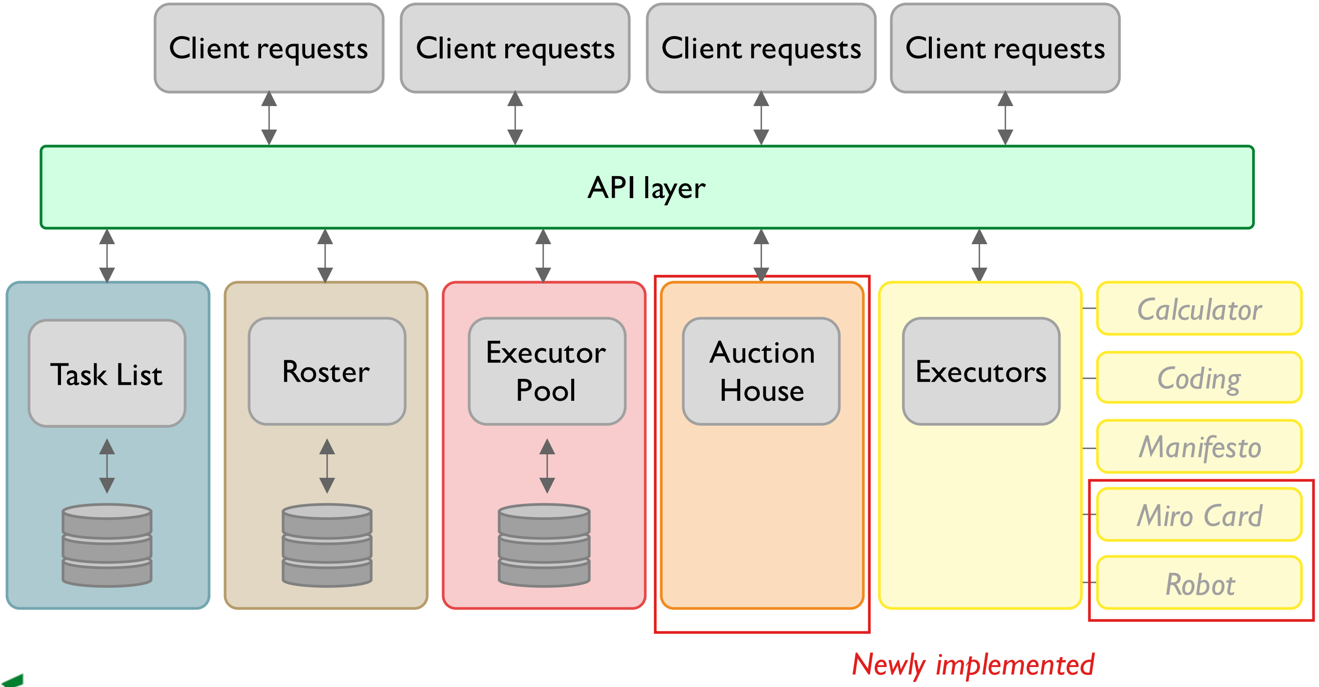
- **Implementation of the Auction House**
- **Implementation of WebSub & MQTT**
- **New Executors** *(Robot & MiroCard)*
- **Improvements to the Task List database**
- **Code Cleaning & Refactoring**

Architecture | What we implemented until our last mid-term presentation…



Architecture | … and now we added the Auction House to communicate with other groups

ADRs 15, 17



We implemented the **Auction House as a new service** to execute:

- Internal tasks for which we do not have an executor
- External tasks for which we have an executors

We also added **two new executors**:

- The Miro Card
- The Robot

Architecture | Deep-Dive into the Auction House

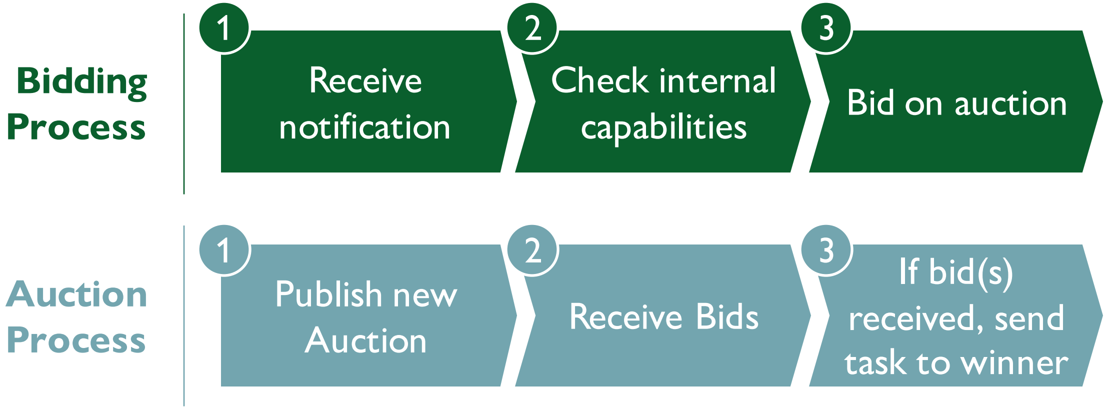
The Auction House acts as the **point of coordination and communication** with the other groups

Via the Auction House, we can either:

(1) launch an auction when no internal executor is found

(2) bid on an auction from another group only if a suitable executor exists

We implemented **two communication protocols**: WebSub and MQTT



Bidding Process

1. Receive notification
2. Check internal capabilities
3. Bid on auction

Auction Process

1. Publish new Auction
2. Receive Bids
3. If bid(s) received, send task to winner

Agenda

Key Architecture Changes

Communication Flows

Decoupling

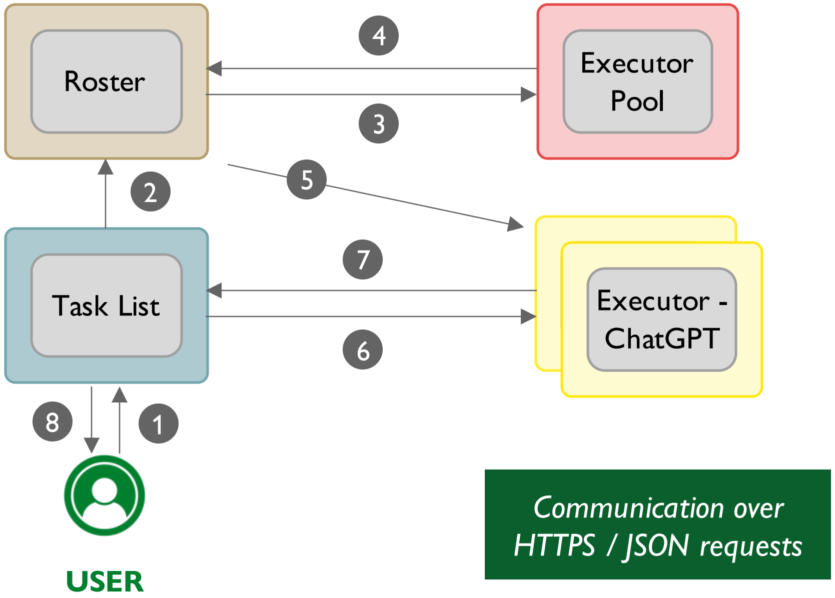
Demonstration

Outlook & Contributions

Communication Flows | Task Flow with Internal Assignment

Graphical Illustration

No databases



Communication over HTTPS / JSON requests

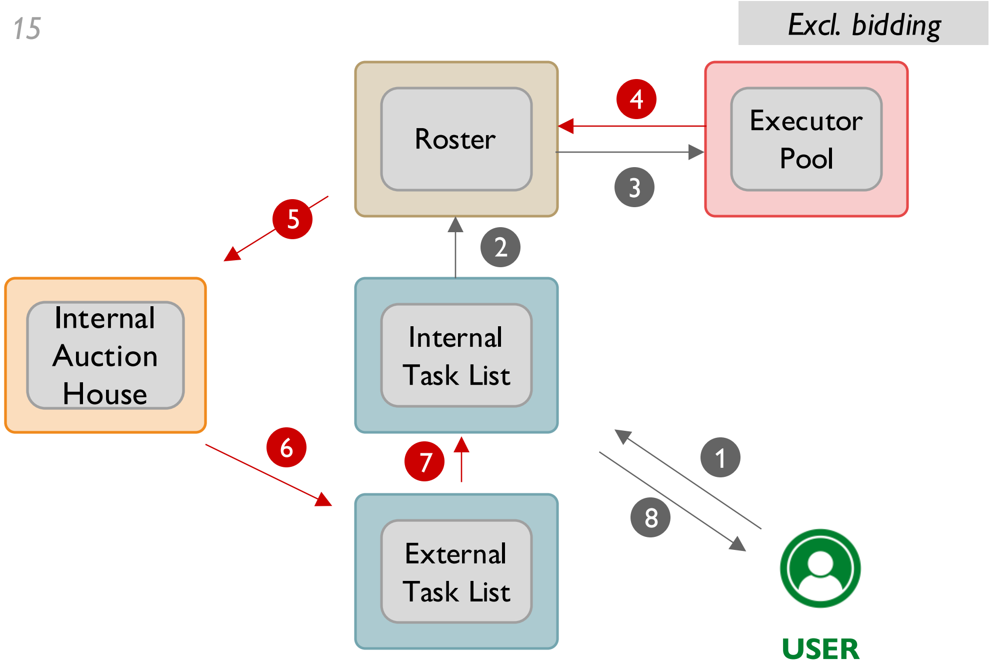
Explanation

1. User adds a task to the task list
2. Task List sends added task to Roster
3. Roster sends request to Pool with the Task Type
4. Pool returns to Roster a list of executors with the right type
5. Roster sends task location (i.e., URI) to the Executor, starting execution
6. Executor reads user input for task from Task List
7. Executor executes task and updates task output & status in Task list
8. User can view the task output

Communication Flows | Task Flow with External Executor

Graphical Illustration

ADRs 15



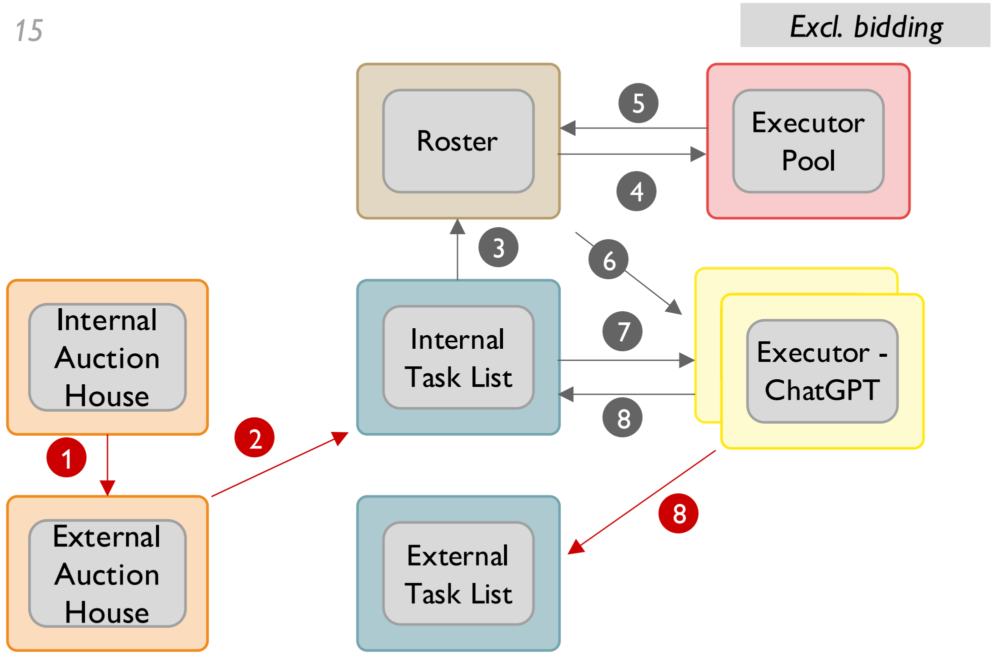
Explanation

1. User adds a task to the task list
2. Task List sends added task to Roster
3. Roster sends request to Executor Pool with the Task Type
4. Executor Pool CANNOT return an executor with the requested type
5. Roster sends task location (i.e., URI) to the auction house, starting auction
6. Auction house select winning bid and creates shadow task at ext. task list
7. External provider executes task and updates our internal task list
8. User can view the task output

Communication Flows | Task Flow with External Task

Graphical Illustration

ADRs 15



Explanation

1 Our auction house sends bid to another auction house, if capabilities match

2 External auction house creates shadow task in our task list

3 Task List sends shadow task to roster

4 Roster sends request to Pool with the Task Type

5 Pool returns to Roster a list of executors with the right type

6 Roster sends task location (i.e., URI) to the Executor, starting execution

7 Executor reads user input from Task List

8 External provider executes task and updates our internal / external task list

High-level process:

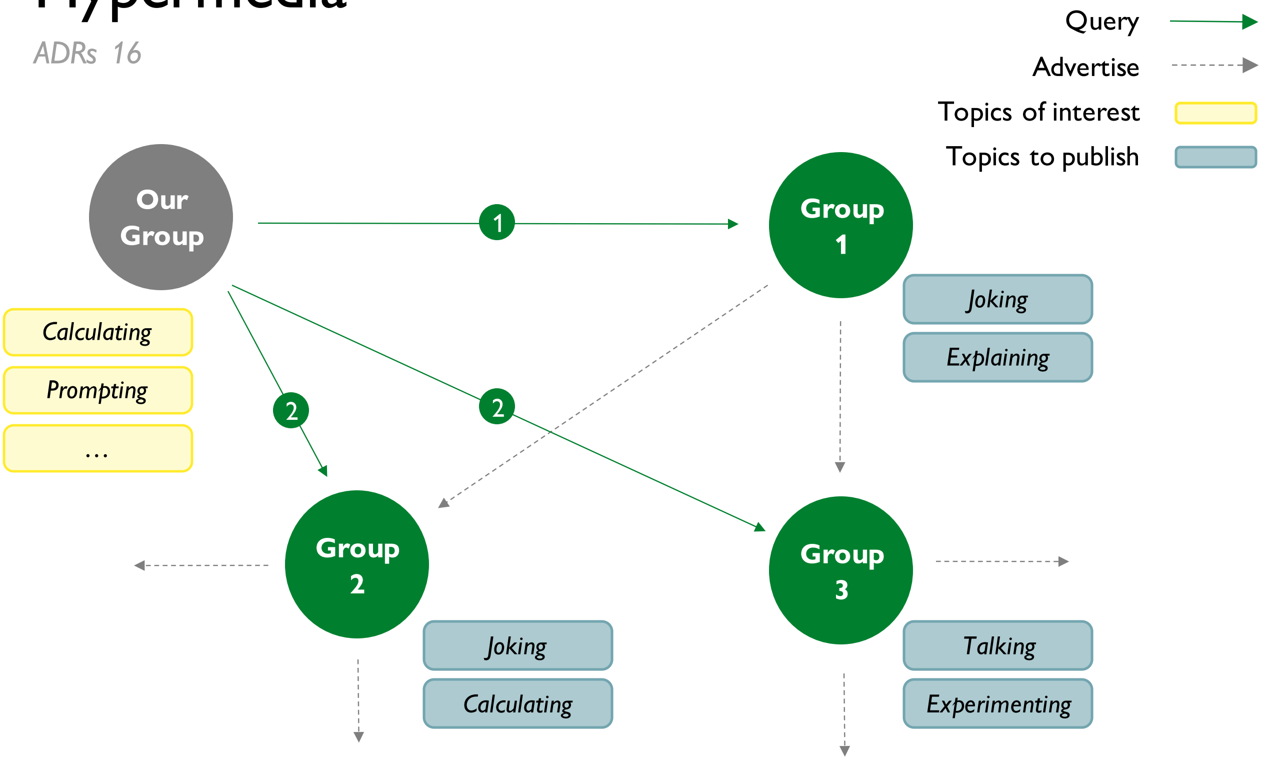
- Crawl through auction feeds
- Interpret the semantic content of headers
- Subscribe to topics of interest

HTTP link headers are used to understand the types of resources they reference

Decisions are made based on the semantic information extracted from these link header and whether it aligns with the client’s current interests

Communication Flows | Crawling via Semantic Hypermedia

ADRs 16



Agenda

Key Architecture Changes

Communication Flows

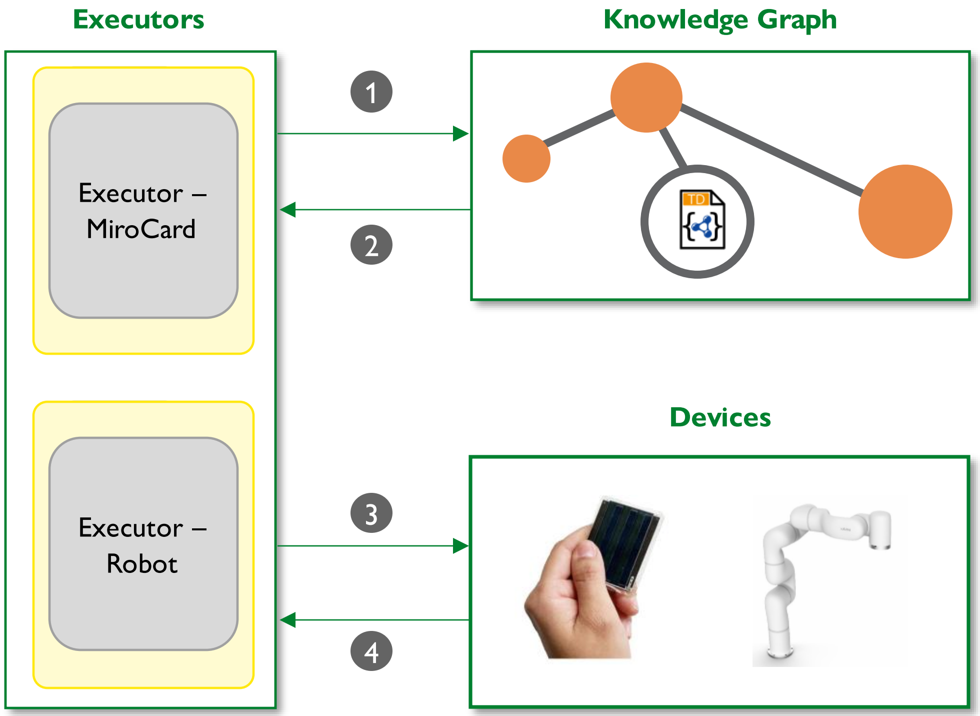
Decoupling

Demonstration

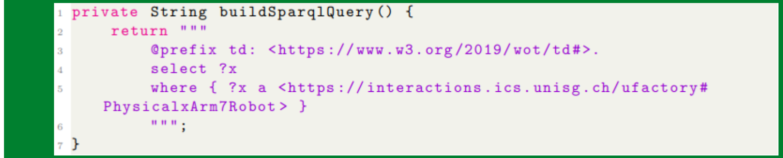
Outlook & Contributions

Decoupling | Hypermedia API

ADRs 17

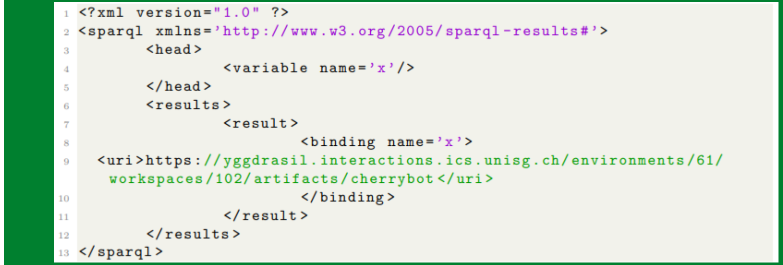


1 Send SPARQL Request



```
private String buildSparqlQuery() {
    return """
        @prefix td: <https://www.w3.org/2019/wot/td#>.
        select ?x
        where { ?x a <https://interactions.ics.unisg.ch/ufactory#
   PhysicalxArm7Robot> }
        """;
}
```

2 Receive URIs of Devices



```
<?xml version="1.0" ?>
<sparql xmlns='http://www.w3.org/2005/sparql-results#'>
        <head>
                <variable name='x'/>
        </head>
        <results>
                <result>
                        <binding name='x'>
  <uri>https://yggdrasil.interactions.ics.unisg.ch/environments/61/
   workspaces/102/artifacts/cherrybot</uri>
                        </binding>
                </result>
        </results>
</sparql>
```

3 Extract TD from Device URI and generate HTTP Post Request

4 Receive status response from device

Agenda

Key Architecture Changes

Communication Flows

Decoupling

Demonstration

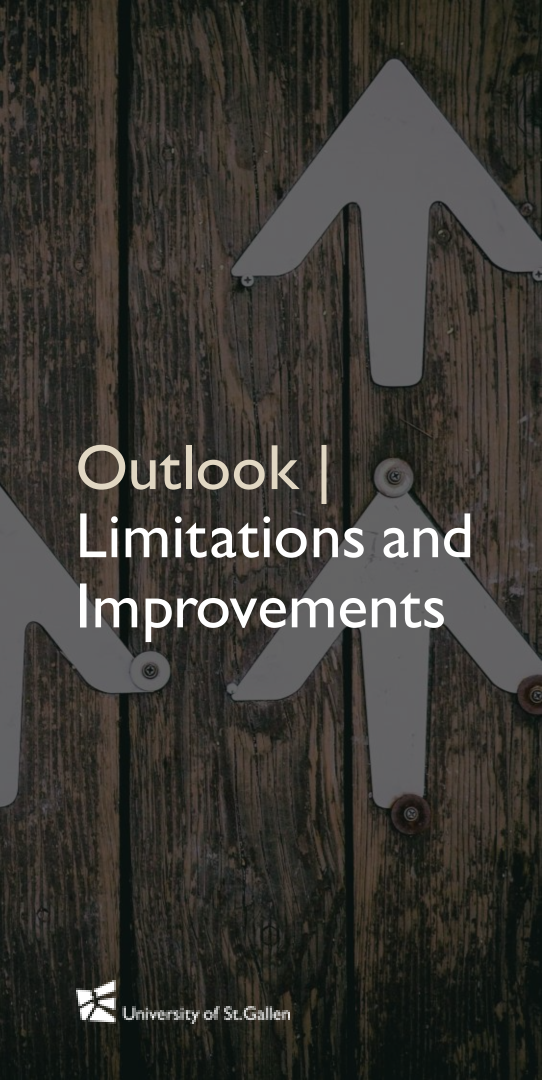
Outlook & Contributions

Demo Time

- Standard process for **internal task assignment** (i.e., with an executor that is not hard-coded)
- **Auction House discovery** at run time through semantic hypermedia
- **Auctioning process** (i.e., receive bid, select bid, delegate task, task execution)

Agenda

- Key Architecture Changes
- Communication Flows
- Decoupling
- Demonstration
- **Outlook & Contributions**



Outlook | Limitations and Improvements

Current limitations

- Potential data inconsistencies (e.g., database flags)
- Limited monitoring / logging
- No ingestion controls / bandwidth mgmt.
- …

Further improvements

- Implementation of async communication
- Code Library for executors / communication
- Develop front-end / SWAGGER UI
- …

Outlook | Troubles & Learning

Troubles

- MQTT implementation
- Alignment with the interoperability TF
- Memory issues with the VM
- Inefficient debugging over VM / deployment
- ….

Learning

- Use SSH to access to VM for efficient debugging
- Early developers define "path forward" for interoperability standard
- Methods & applications around decentralized communication and decoupling
- Plugfests are a lot of fun
- …

Contributions | We were able to distribute the group work quite fairly within our team

	Kaan	Michael	Daniel	Stephan
Auction House				
WebSub	✓	✓		✓
MQTT		✓	✓	
Other services	✓	✓		
Semantic HM	✓	✓	✓	✓
New Executors		✓	✓	✓
Documentation	✓		✓	✓

Any questions?