Create your Digital Twin in days, not months.

Use case from the Robotic applications and its IoT integrations

Hoa Nguyen, Østfold University College
Part of the Arrowhead Tools project

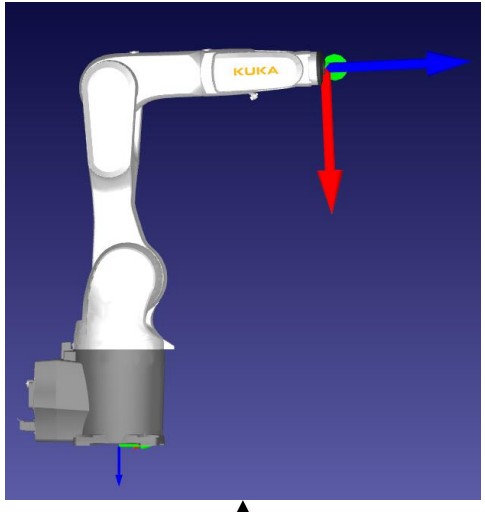
hoa.nguyen@hiof.no

ARROWHEAD TOOLS
KYKLOS 4.0
CHANGE2TWIN
DAIS
DRYADS
Supported by advanced EU H2020 projects

Introduction

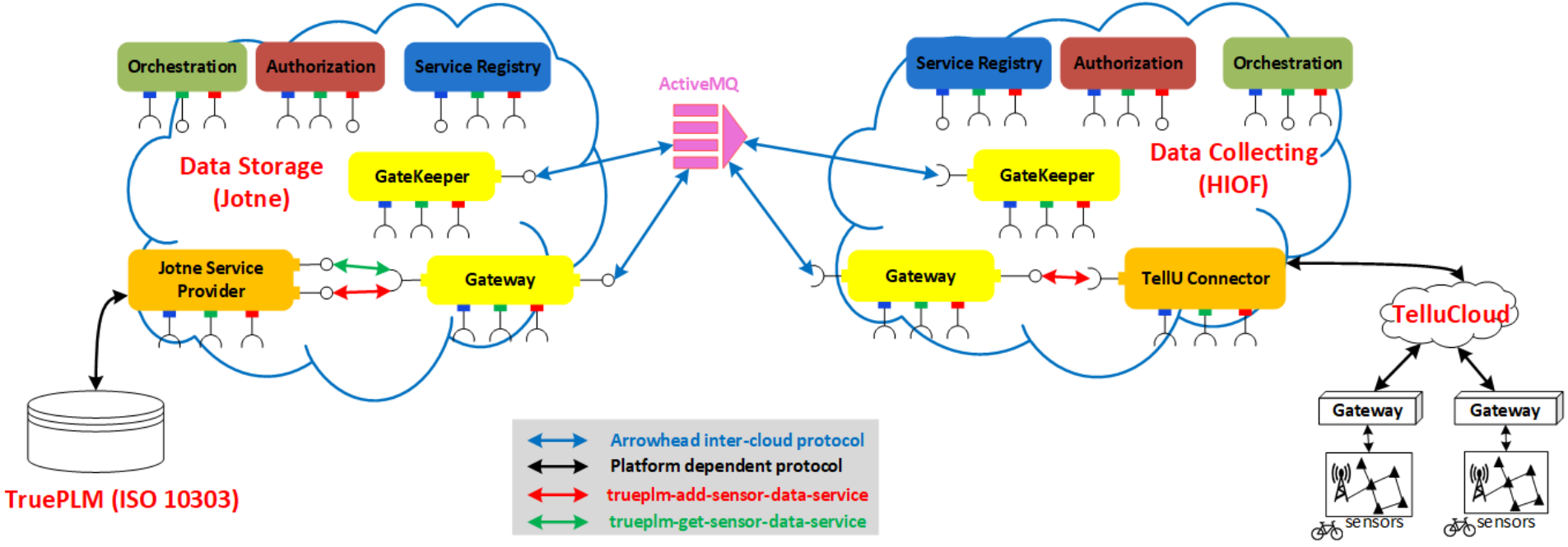
Data exchange challenge

- End-to-end security
- Interoperability support between systems



ARROWHEAD TOOLS

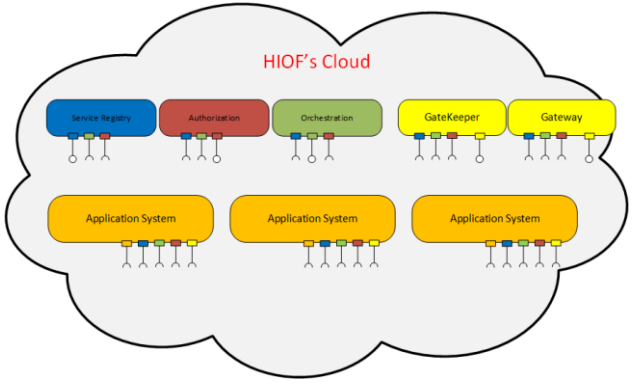
The local cloud solution

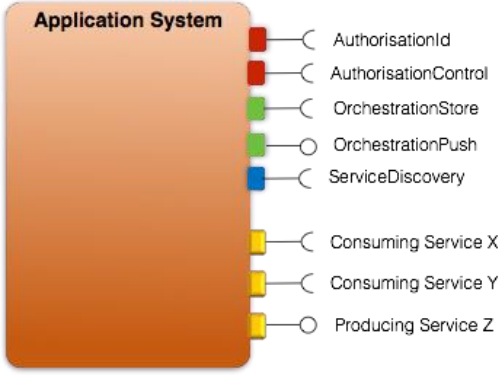


ARROWHEAD TOOLS

Setting up Arrowhead cloud

- A minimum local cloud requires:
  - ServiceDiscovery
  - AuthorisationControl
  - OrchestrationStore
- For inter-cloud communication:
  - Gatekeeper
  - Gateway



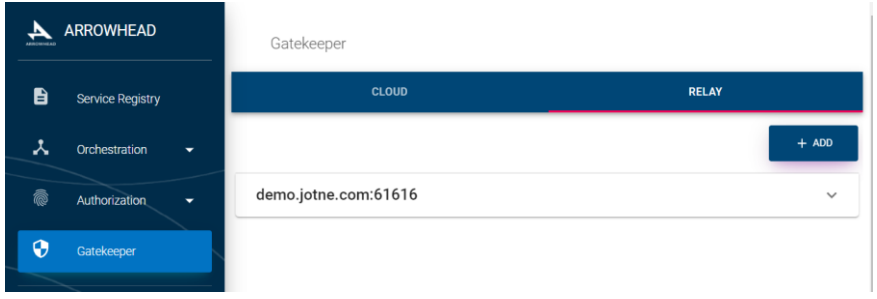


https://github.com/eclipse-arrowhead/core-java-spring#quickstart

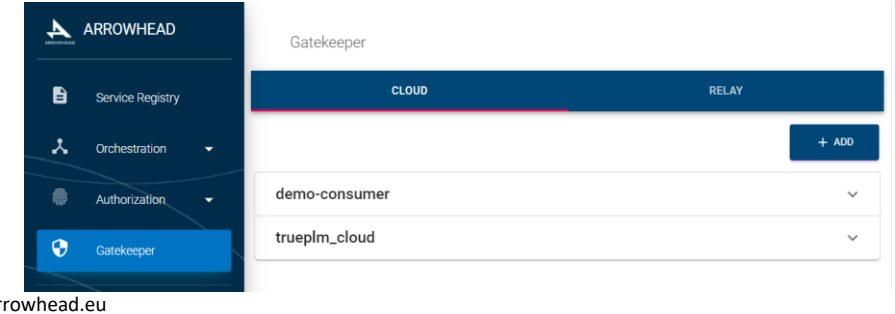
ARROWHEAD TOOLS

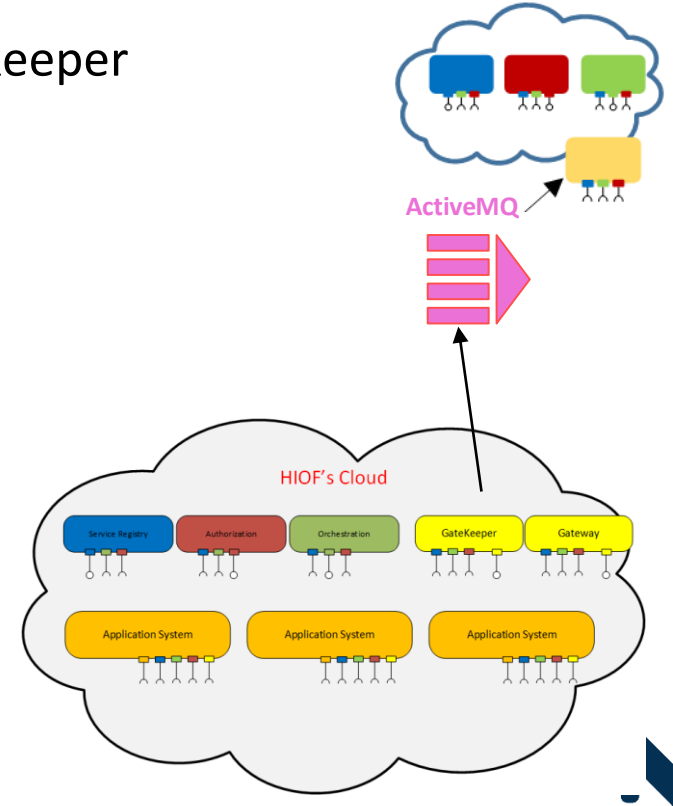
Setting up neighbour cloud

- Add Jotne's relay information to Gatekeeper



- Add Jotne's cloud information



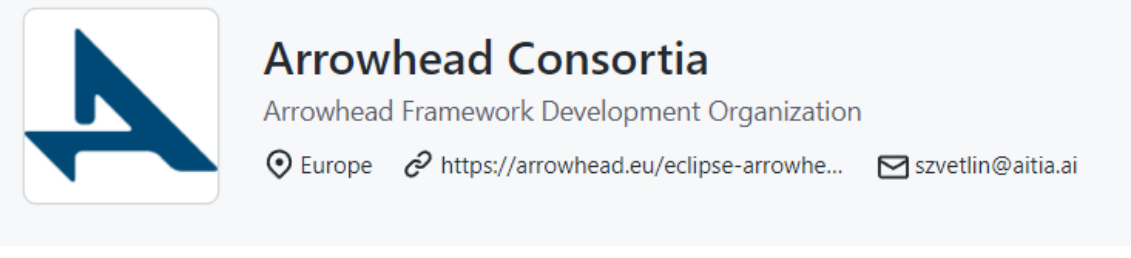


ARROWHEAD TOOLS

Create consumer system

Can use client-libraries from Arrowhead Consortia

- Java spring boot: https://github.com/arrowhead-f/client-skeleton-java-spring
- Kalix: https://github.com/arrowhead-f/arrowhead-kalix
- C sharp: https://github.com/arrowhead-f/application-library-csharp
- Python: https://github.com/arrowhead-f/client-library-python
- Nodejs: https://github.com/arrowhead-f/client-nodejs



ARROWHEAD TOOLS

An example of a java client

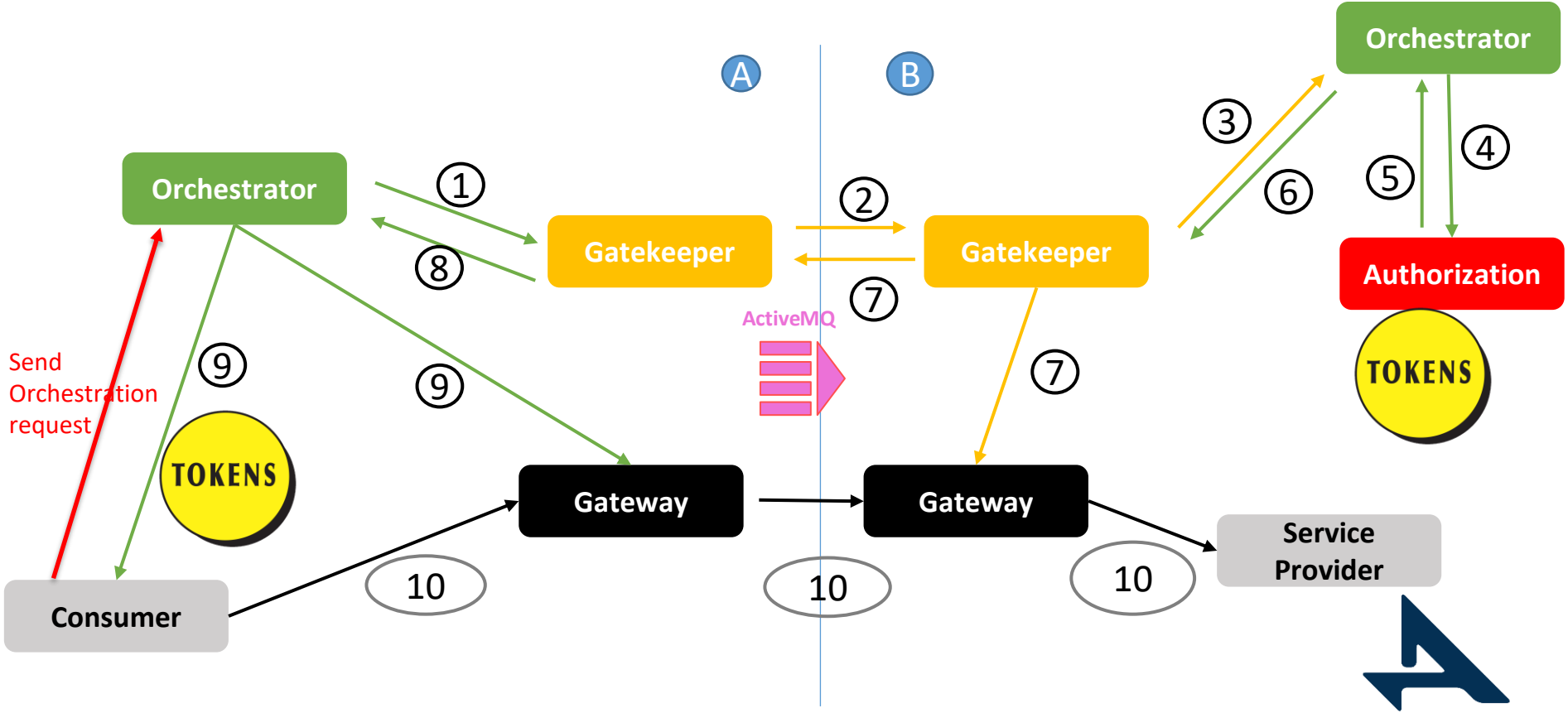
Initialize flags

Send Orchestration request

```
final ServiceQueryFormDTO serviceQueryForm = new ServiceQueryFormDTO.Builder(
        Constants.TRUEPLM_GET_SENSOR_SERVICE_DEFINITION).interfaces(getInterface()).build();

final Builder orchestrationFormBuilder = arrowheadService.getOrchestrationFormBuilder();
final OrchestrationFormRequestDTO orchestrationFormRequest = orchestrationFormBuilder
        .requestedService(serviceQueryForm).flag(Flag.TRIGGER_INTER_CLOUD, true).flag(Flag.OVERRIDE_STORE, true)
        .flag(Flag.ENABLE_INTER_CLOUD, true).build();

final OrchestrationResponseDTO orchestrationResponse = arrowheadService
        .proceedOrchestration(orchestrationFormRequest);

if (orchestrationResponse == null) {
    logger.info("No orchestration response received");
} else if (orchestrationResponse.getResponse().isEmpty()) {
    logger.info("No provider found during the orchestration");
} else {
    final OrchestrationResultDTO orchestrationResult = orchestrationResponse.getResponse().get(0);
    validateOrchestrationResult(orchestrationResult, Constants.TRUEPLM_GET_SENSOR_SERVICE_DEFINITION);
    final String token = orchestrationResult.getAuthorizationTokens() == null ? null
            : orchestrationResult.getAuthorizationTokens().get(getInterface());

    String path = "/Palfinger_Crane_Assembly_1/HA-KA-400-700/urn:rdl:Palfinger_Crane_Assembly_1:sensor values list";

    final String[] queryParamCraneData = { "fromDate", "01/03/2021 17:00:00", "toDate", "01/03/2021 17:30:00" };

    final JotneSensorDataDTO craneData = arrowheadService.consumeServiceHTTP(JotneSensorDataDTO.class,
            HttpMethod.valueOf(orchestrationResult.getMetadata().get(Constants.HTTP_METHOD)),
            orchestrationResult.getProvider().getAddress(), orchestrationResult.getProvider().getPort(),
            orchestrationResult.getServiceUri() + path, getInterface(), token, null, queryParamCraneData);
```


ARROWHEAD TOOLS

Inter-Cloud orchestration with the token



An example of a java client

Initialize flags

Send Orchestration request

Received token

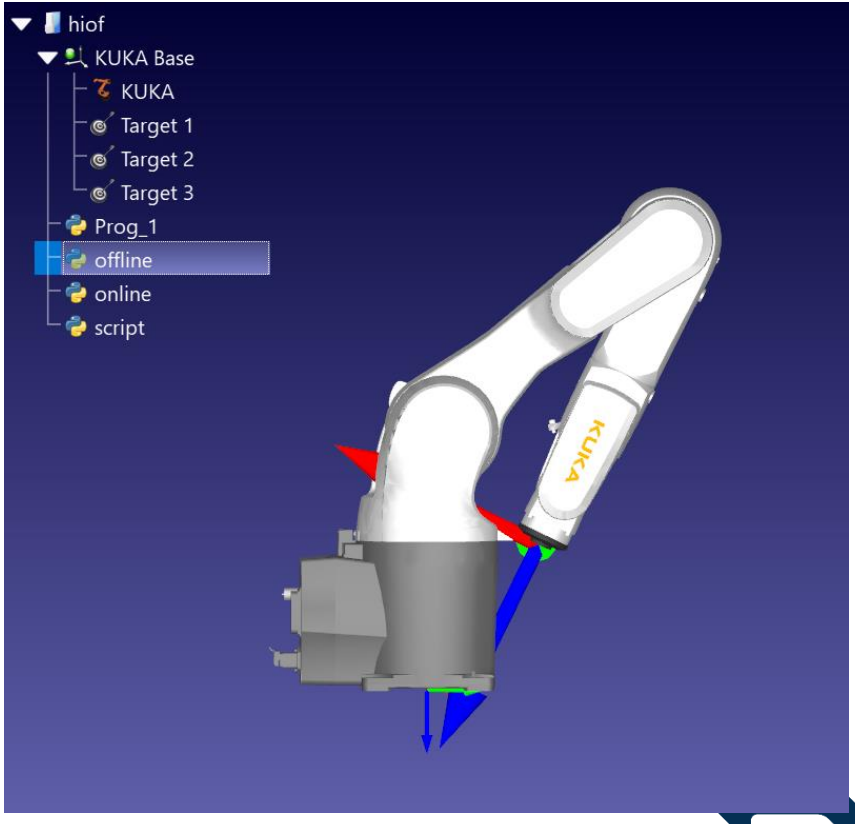
Consuming service

```java
final ServiceQueryFormDTO serviceQueryForm = new ServiceQueryFormDTO.Builder(
        Constants.TRUEPLM_GET_SENSOR_SERVICE_DEFINITION).interfaces(getInterface()).build();

final Builder orchestrationFormBuilder = arrowheadService.getOrchestrationFormBuilder();
final OrchestrationFormRequestDTO orchestrationFormRequest = orchestrationFormBuilder
        .requestedService(serviceQueryForm).flag(Flag.TRIGGER_INTER_CLOUD, true).flag(Flag.OVERRIDE_STORE, true)
        .flag(Flag.ENABLE_INTER_CLOUD, true).build();

final OrchestrationResponseDTO orchestrationResponse = arrowheadService
        .proceedOrchestration(orchestrationFormRequest);

if (orchestrationResponse == null) {
    logger.info("No orchestration response received");
} else if (orchestrationResponse.getResponse().isEmpty()) {
    logger.info("No provider found during the orchestration");
} else {
    final OrchestrationResultDTO orchestrationResult = orchestrationResponse.getResponse().get(0);
    validateOrchestrationResult(orchestrationResult, Constants.TRUEPLM_GET_SENSOR_SERVICE_DEFINITION);
    final String token = orchestrationResult.getAuthorizationTokens() == null ? null
            : orchestrationResult.getAuthorizationTokens().get(getInterface());

    String path = "/Palfinger_Crane_Assembly_1/HA-KA-400-700/urn:rdl:Palfinger_Crane_Assembly_1:sensor values list";

    final String[] queryParamCraneData = { "fromDate", "01/03/2021 17:00:00", "toDate", "01/03/2021 17:30:00" };

    final JotneSensorDataDTO craneData = arrowheadService.consumeServiceHTTP(JotneSensorDataDTO.class,
            HttpMethod.valueOf(orchestrationResult.getMetadata().get(Constants.HTTP_METHOD)),
            orchestrationResult.getProvider().getAddress(), orchestrationResult.getProvider().getPort(),
            orchestrationResult.getServiceUri() + path, getInterface(), token, null, queryParamCraneData);
```


ARROWHEAD TOOLS

Python client for robot simulation

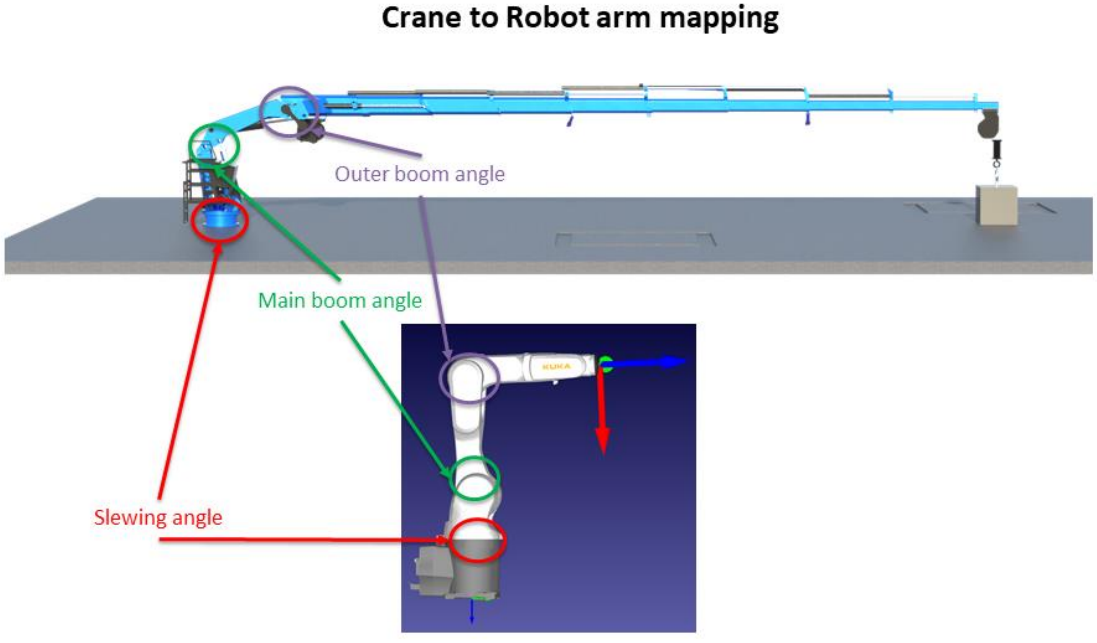
```python
import json
from client_python.arrowhead_client.client.implementations import SyncClient

robodk_client = SyncClient.create(
        system_name='robot_consumer',
        address='localhost',
        port=6000,
        keyfile='certificates/robot_consumer.key',
        certfile='certificates/robot_consumer.crt',
        cafile='certificates/sysop.ca',)

if __name__ == '__main__':

    robodk_client.setup()
    robodk_client.add_orchestration_rule('trueplm-get-sensor-data-service', 'GET',
                                         OrchestrationFlags.ENABLE_INTER_CLOUD)
    response = robodk_client.consume_service('trueplm-get-sensor-data-service',
                                             json={'fromDate', "01/03/2021 17:00:00",
                                                   'toDate', "01/03/2021 17:30:00"})

    data = parse_json(response.read_json())
```



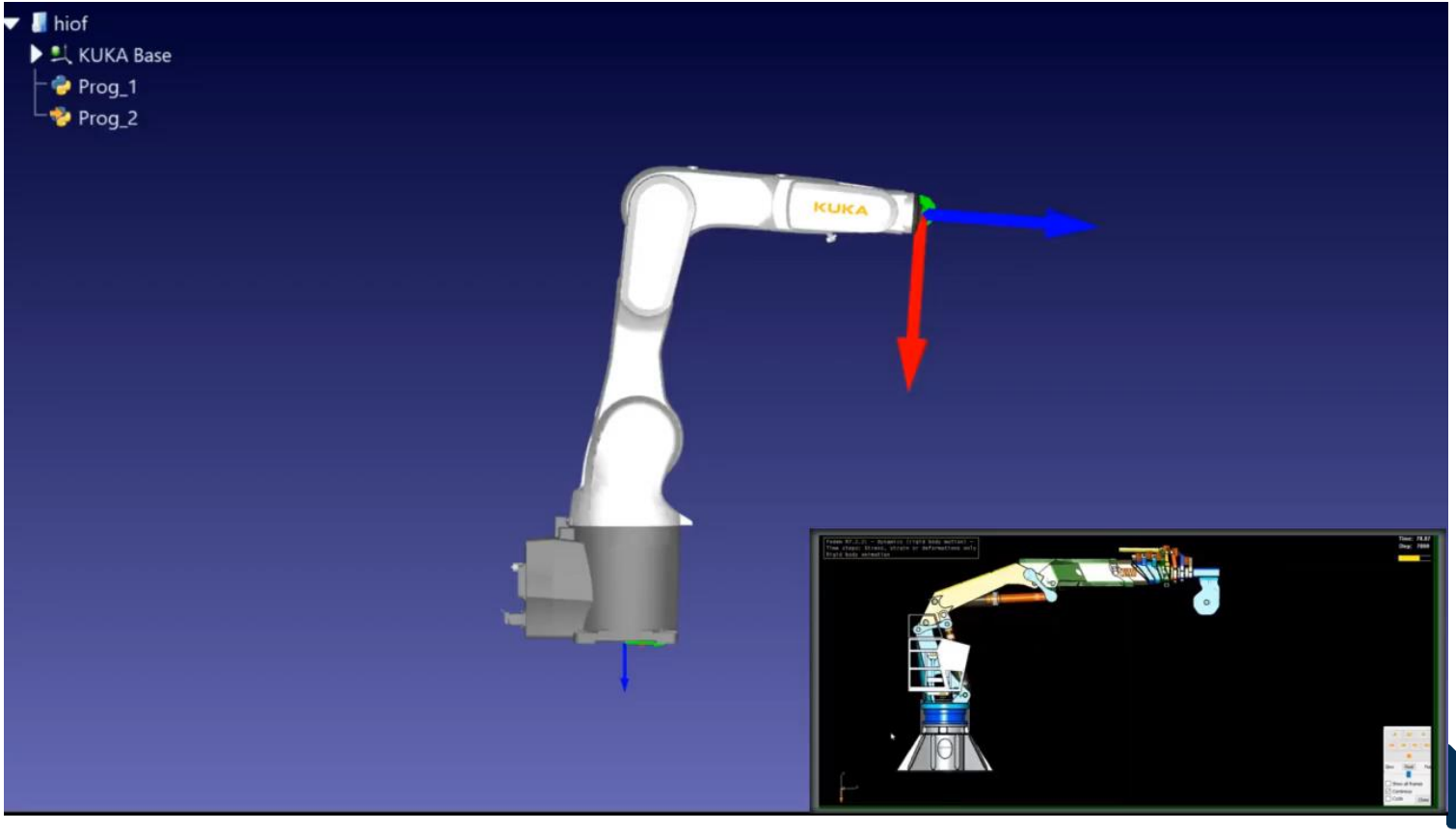
ARROWHEAD TOOLS

Data mapping

Replica of the digital crane with robot arm



ARROWHEAD TOOLS



ARROWHEAD TOOLS

Contacts

arrowhead-dev.slack.com

helpdesk@jotne.com

ARROWHEAD TOOLS