goKI

# Open Innovation Lab AI

## for the Realization of Public Good AI Applications

OPACA Tools Workshop, June 2025

Gefördert durch:

Bundesministerium
für Arbeit und Soziales

ZEKI
Zentrum für erlebbare
Künstliche Intelligenz
und Digitalisierung e.V.

GT ARC

# Agenda

- Introducing the Go-KI Project

- The “OPACA” Framework
- OPACA Development Tools
  - BPMN Editor
  - LLM Interface

- Live Demonstration
- Hands-on Session
- Feedback

# Motivation & Project Goals

Artificial intelligence (AI) will penetrate all aspects of life, work and private life

Public good-oriented AI requires an even broader range of competences

Many ideas cannot be implemented due to lack of individual competences

Projects such as the *Civic Innovation Platform* help realize specific ideas

## OUR APPROACH

Spaces, competences and technologies for the collaborative development of public good-oriented AI applications

Combining the technical feasibility with civil-societal responsibility

Information services and digital learning modules for AI technologies

Complementing existing structures for the networking of actors

# Four Pillars of the Project

## REAL LAB

- Modern Co-Working Space at ZEKI
- Experimental environment for data collection, modelling & validation
- Research into the future of work environment

## TANGIBLE AI

- Increase the visibility and explainability of AI applications
- Improve the trust in AI applications where appropriate
- Reduce fears, increase acceptance

## SOFTWARE-STACK

- Simplified and unified AI application development
- Scalable, distributed, interoperable, secure
- Open and modern technologies

## PLATFORM & ECOSYSTEM

- Libraries and tools for development, testing, deployment
- Directory services for reusable components and services
- Workshops and training events

# Four Pillars of the Project

## SOFTWARE-STACK

- AI Framework “OPACA”: API combining Multi-Agent Systems, Microservices and Container Technologies
- Language- and Platform-Independent, scalable, open-source
- Different Development Tools
- User applications, incl. BPMN Editor and LLM-Integration

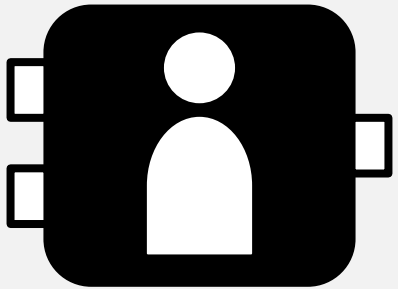
# OPACA Framework

App

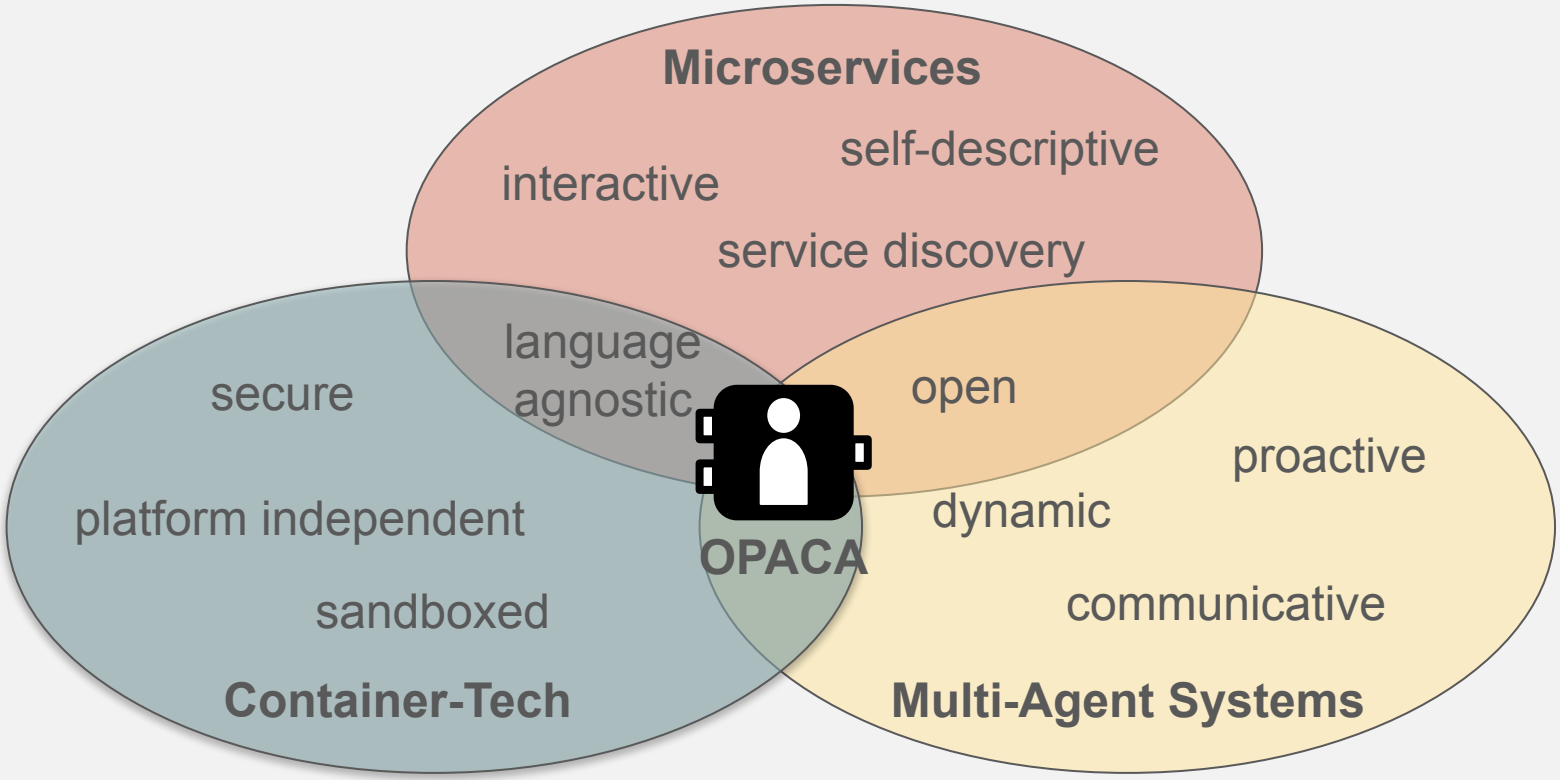
## OPACA - Open, Language- and Platform-Independent API for Containerized Agents

- Framework + API for containerized agents and AI applications
- Allow for heterogeneous applications written in different languages
- Dynamic environment, automated discovery and self-description
- Execution in Docker or Kubernetes

- https://github.com/gt-arc/opaca-core



# OPACA Framework

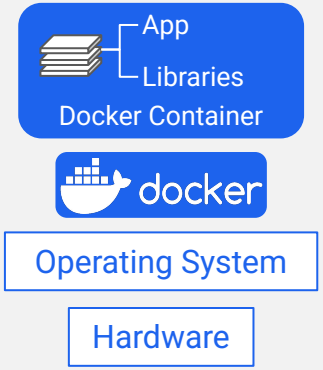


# OPACA Framework

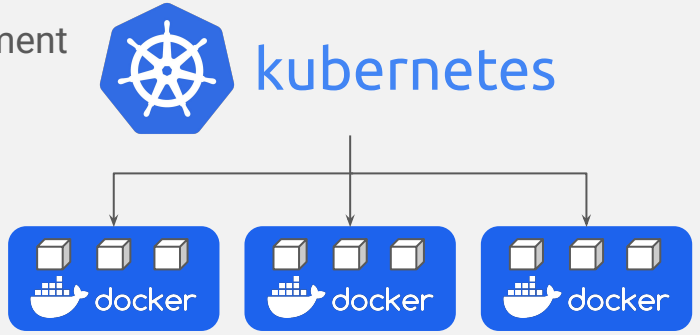
## Virtualization

- Encapsulating the program environment
- Deterministic behavior
- Security
- Docker for containerization
- Kubernetes for orchestration of container

Container Deployment



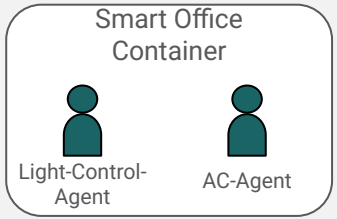

Kubernetes Deployment

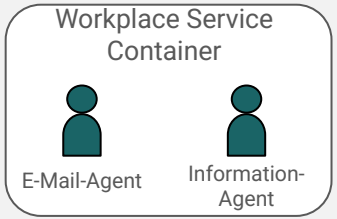


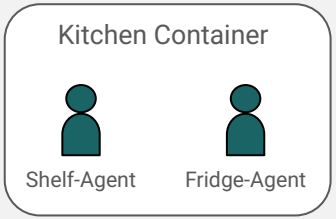
# OPACA Framework

## Agent Container

- Implemented in any language, packaged as Docker container
- Contain one or more *agents*, providing executable *actions*
- Self-descriptive models for container, agents, actions
- Uniform REST API



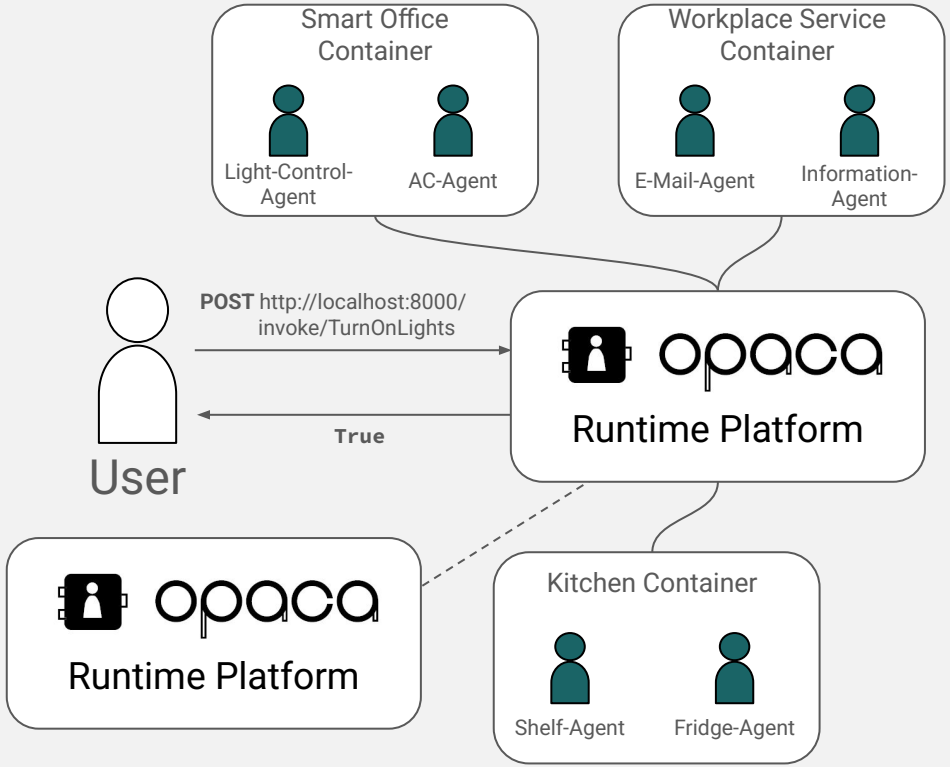



# OPACA Framework

## Runtime Platform

- Used to deploy, manage, and interact with Agent Containers
- Connect with other RPs
- Basic functions for e.g. auth, user management, monitoring
- Central point for interaction, forwards API calls to respective AC or connected RP



# OPACA: Applications

## Agent

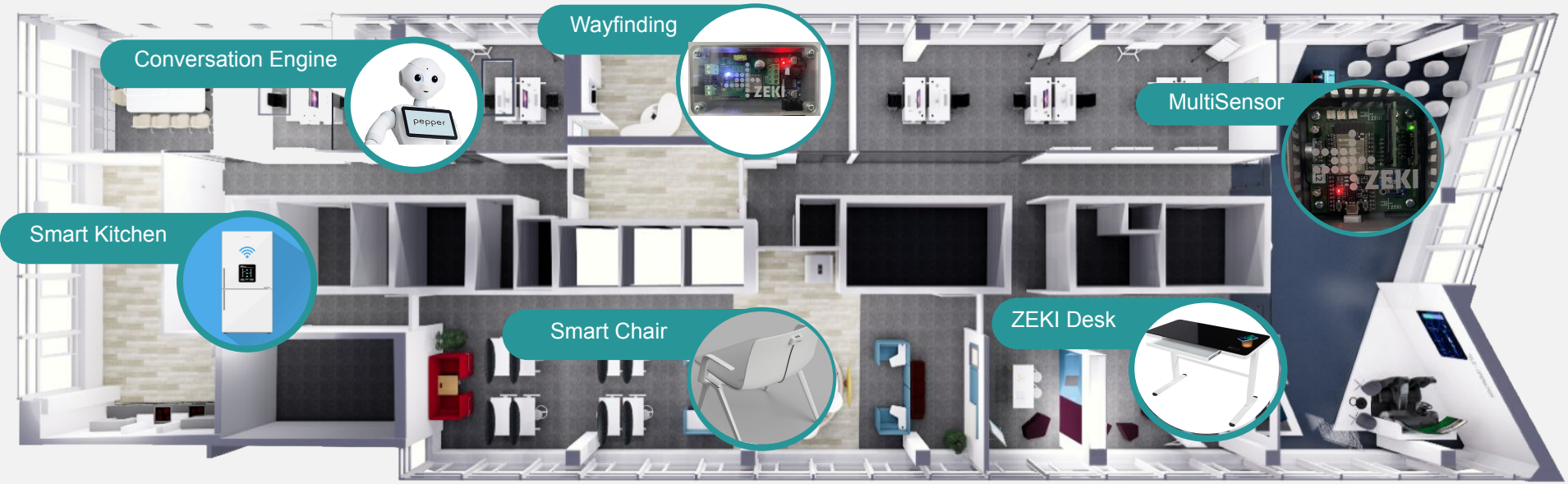
**Name:**
SensorAgent
**Description:**
Provides live sensor data collected in the ZEKI-Reallabor.
**Actions:**
- `GetTemperature`
- `GetNoise`
- `GetHumidty`
- `GetSensors`
- `RegisterSesnsor`
- `RemoveSensor`

## Action

**Name:**
GetTemperature
**Description:**
Retrieves the current temperature in Celsius from a sensor in the provided room.
**Parameters:**
```
{
 “room”: String
}
```
**Returns:**
`Number`

# OPACA: Applications

## ZEKI Reallabor Integration

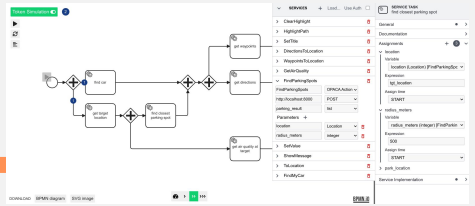


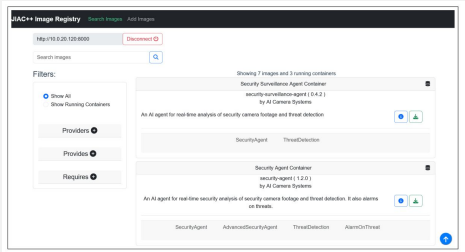
# OPACA Software Stack

Idea

Modelling

Low-threshold modelling of ideas as **BPMN processes**, alternatively to programming (**low-code and no-code**)



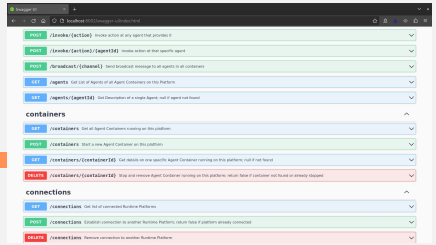


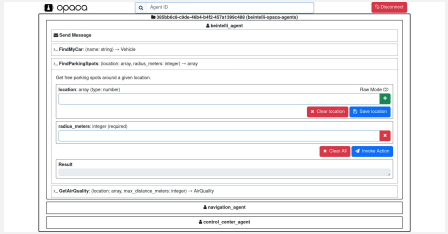
Enables **reusability** of functions and components from **registries**

Orchestration

Validation

**Uniform interfaces** for testing and integrating applications



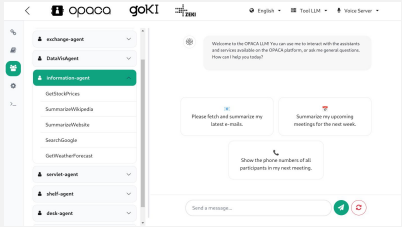


Tools for easy **inspection of and interaction** with running applications, agents and actions

Execution

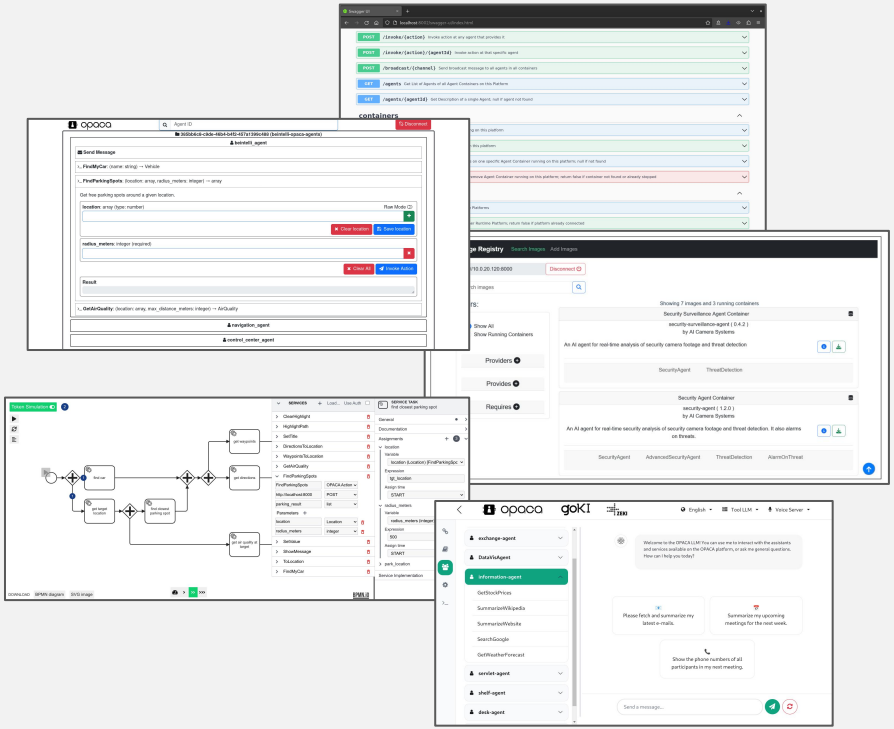
Interaction

**Intuitive interaction** via chat / LLM



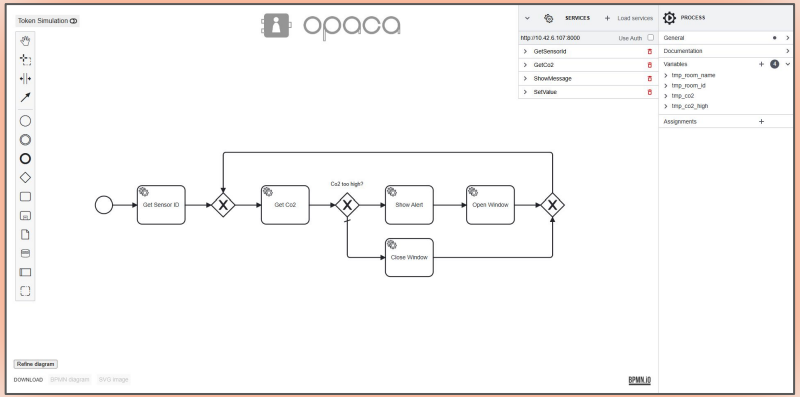
# OPACA Tools

- Integrated Swagger UI
- Command Line Interface
- Dynamic, Form-based Web UI
- Agent Container Registry
- **BPMN Editor & Interpreter**
- **LLM Web Interface**

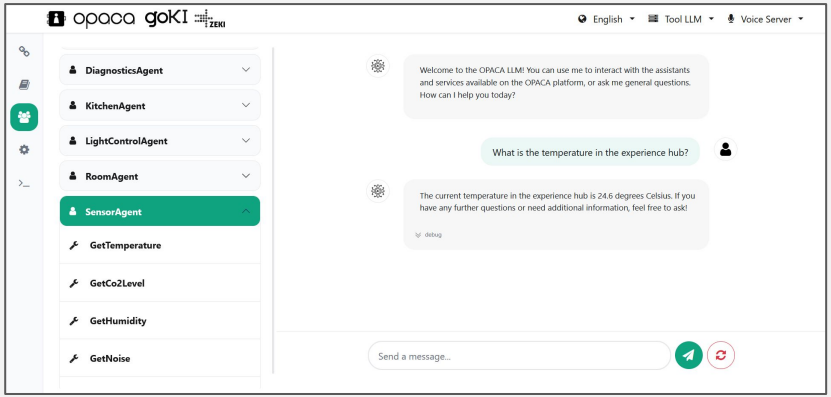


# OPACA Tools: Spotlight

## BPMN-Editor



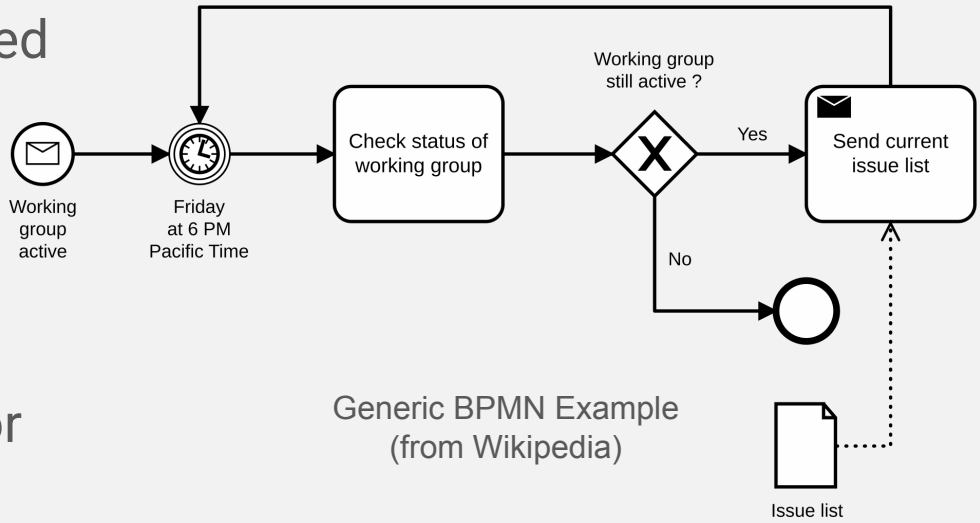

## OPACA-LLM



# OPACA BPMN Editor & Interpreter

## Business Process Model and Notation

- OMG-standardized graphical notation
- Intuitive to understand, widely used
- Bridging business and technical perspectives
- Can be used for modelling both, “real world” business processes and software-systems
- Especially suited for distributed or agent-based systems!



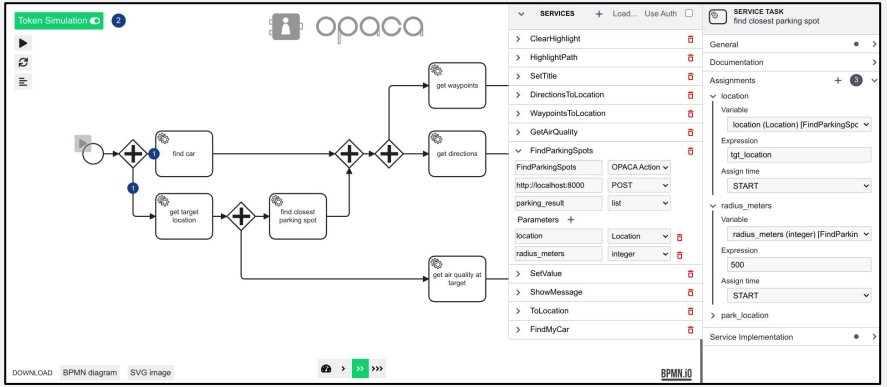

Generic BPMN Example (from Wikipedia)

# OPACA BPMN Editor & Interpreter

## Technical Foundations

- Based on popular **bpmn-js** framework
- Intuitive, web-based UI, extensible
- Basic editing features, import/export, and many more features
- Manual “Token Simulation”

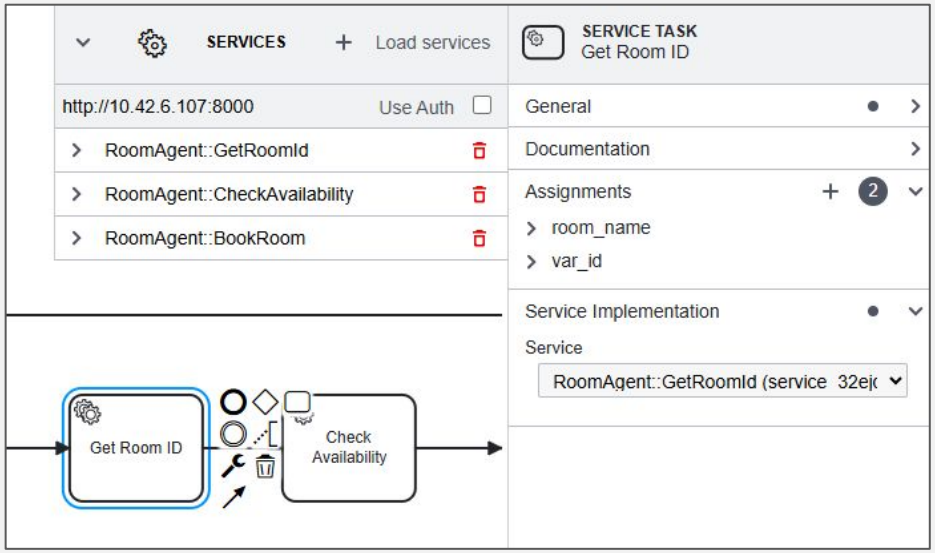
- and some extensions of our own…



# OPACA BPMN Editor & Interpreter

## Extensions & OPACA Integration

- Model Extensions for Executability: Variables, Assignments, …
- Import of OPACA Actions as Service implementations
- Extension of Simulation to fully-automated Process Interpreter
- In-editor or headless mode



# OPACA BPMN Editor & Interpreter

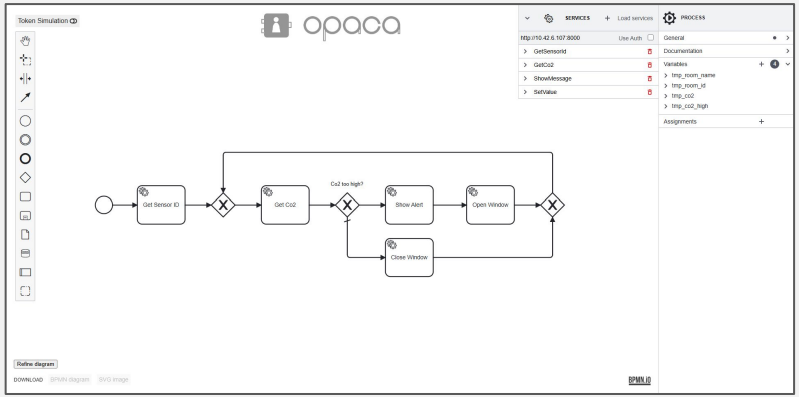
## LLM Integration (WIP)

- Based on ProMoAI [1] with some extensions
- Can be used to generate a first draft for “getting started”, or to ask the LLM for suggestions
- Currently works best with small/simple processes

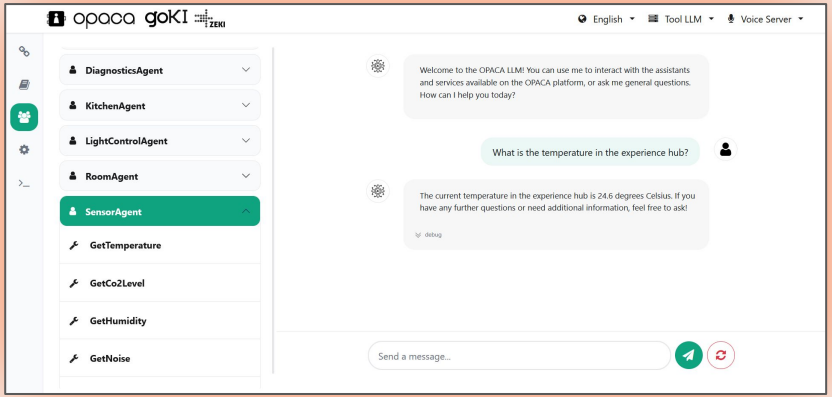

[1] https://github.com/humam-kourani/ProMoAI

# OPACA Tools: Spotlight

## BPMN-Editor



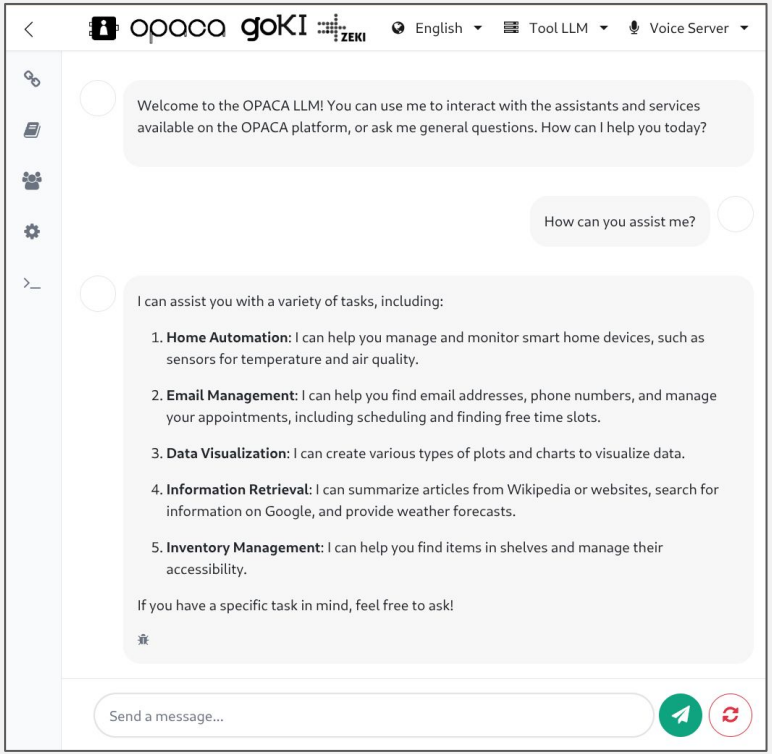

## OPACA-LLM



# OPACA LLM UI

## Overview & Motivation

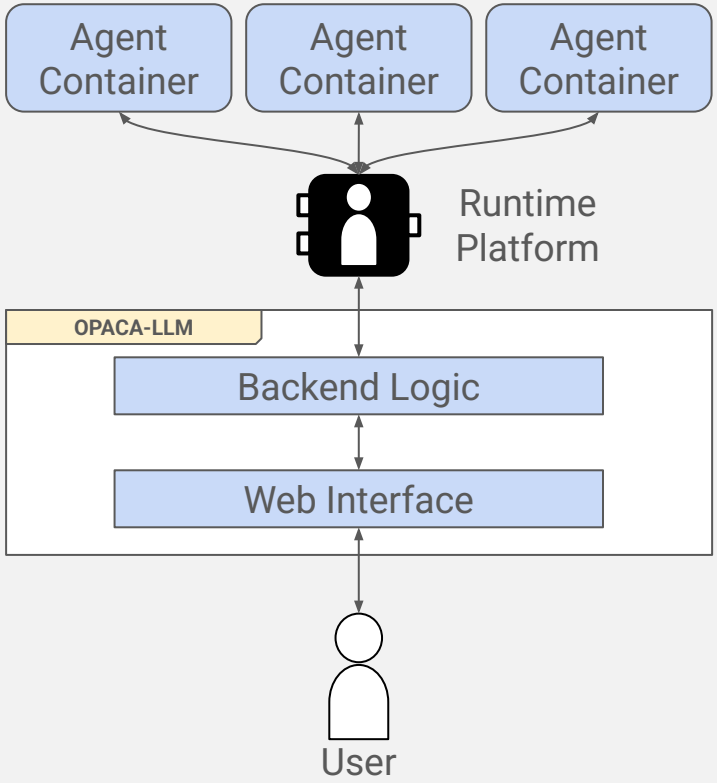
- Chat-Interface for OPACA
- Intuitive natural language input
- Select & invoke relevant actions based on user inputs
- Support for multiple prompting methods and LLMs
- No prior knowledge about available services necessary



# OPACA LLM UI

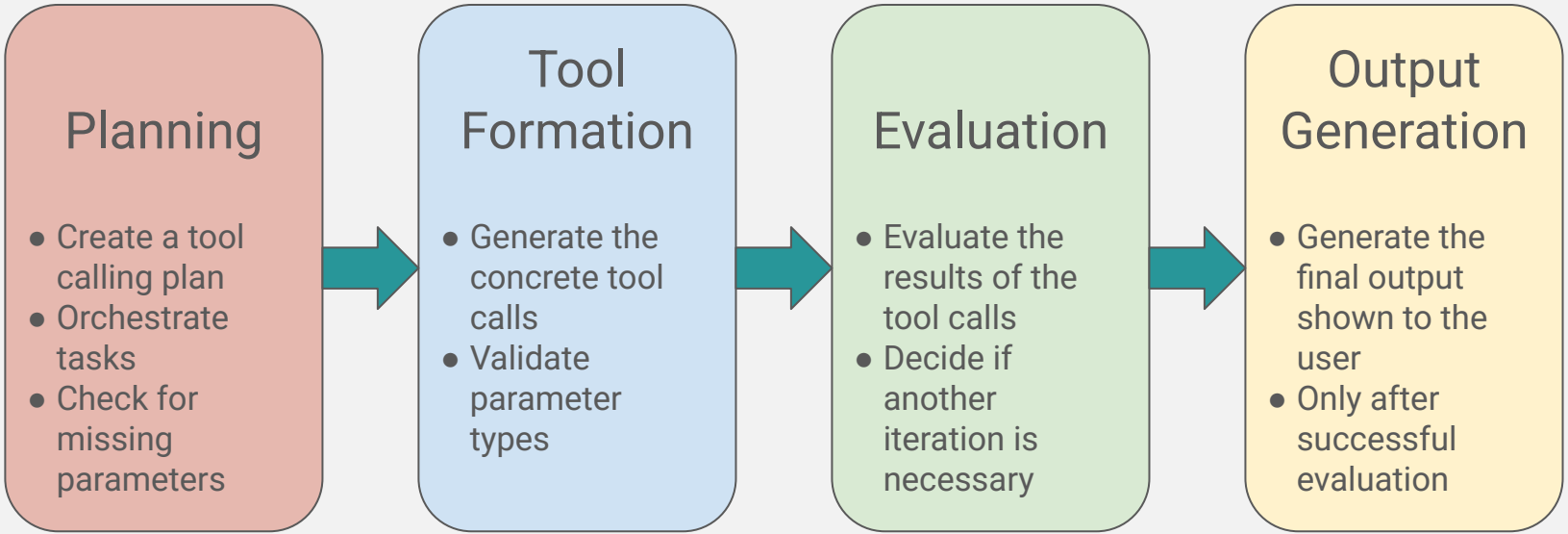
## Architecture

- Backend connected to OPACA Runtime Platform
- Get list of available agents and actions, parameters, types, …
- Can invoke multiple actions per query (in parallel or sequentially)
- Limited support for multiple users
- Rich “behind the scenes” logging for debugging & transparency



# OPACA LLM UI

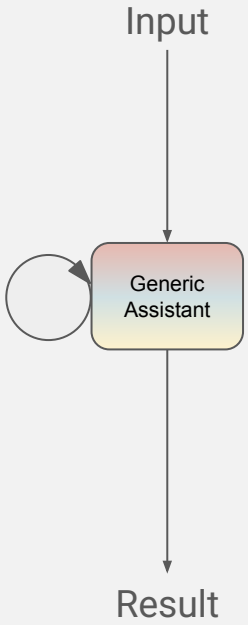
## Approach



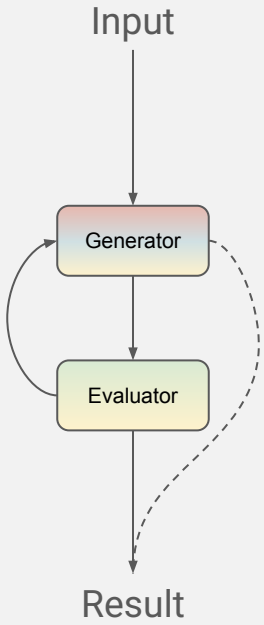
# OPACA LLM UI

## Methods Overview

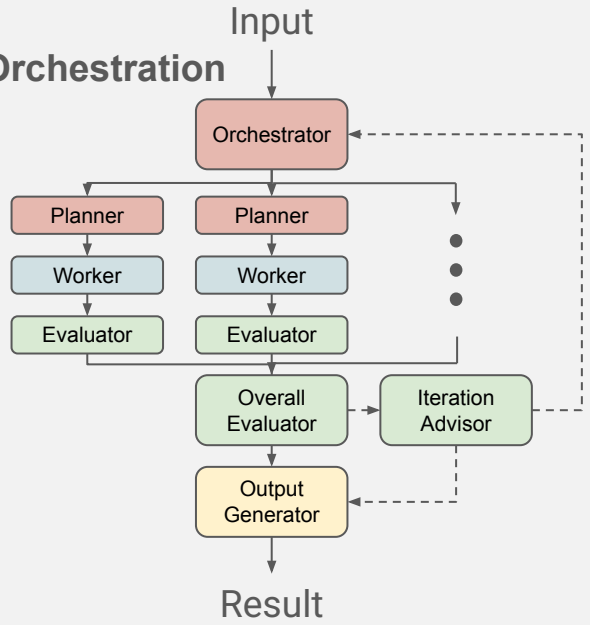
**Simple**




**Tool**




**Orchestration**



# OPACA LLM UI

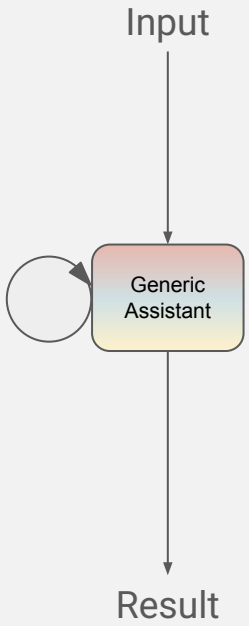
## Simple Method

### Pro

- Very fast
- Simple modification
- Moderate answer quality

### Contra

- Instruction overloading
- Dependent on base model
- No parallel action execution



# OPACA LLM UI

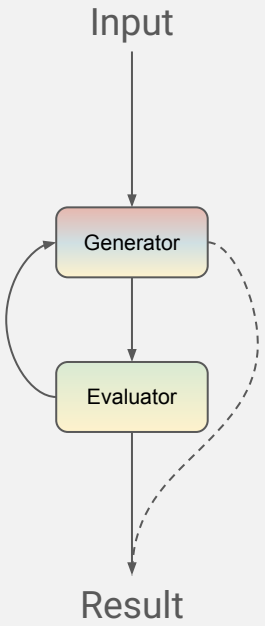
## Tool Method

### Pro

- Fast
- Parallel action generation
- Strong answer quality

### Contra

- Limited number of actions
- Dependent on base models



# OPACA LLM UI

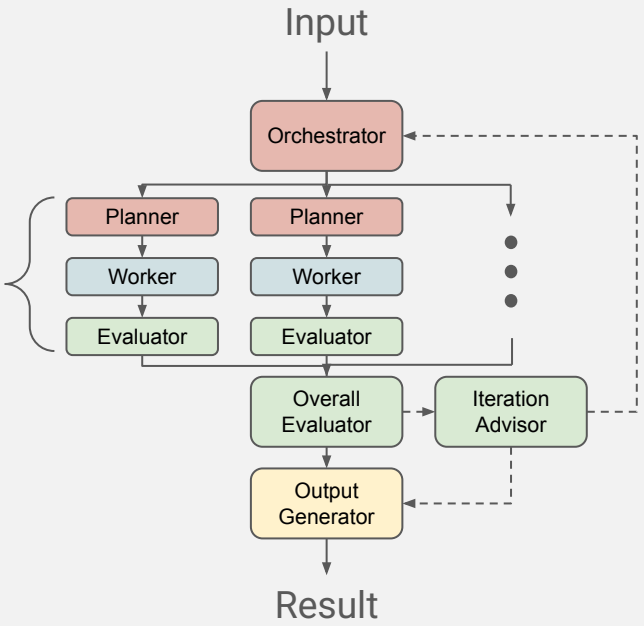
## Orchestration Method

### Pro

- Strong answer quality
- Uses different models for diff. agents
- LLM agents mirror OPACA platform
- Can handle large number of tools

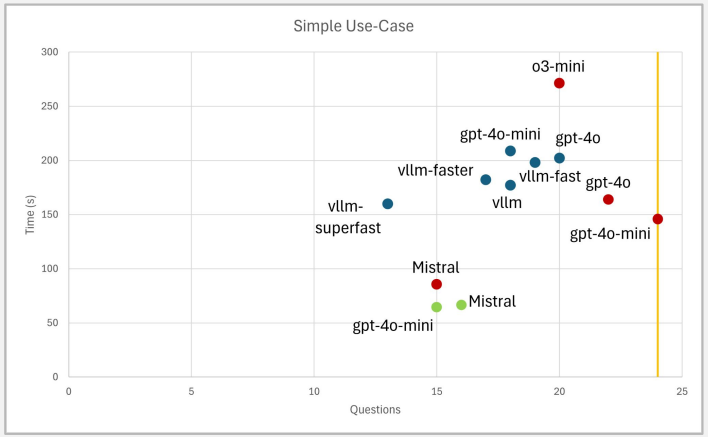
### Contra

- Slow, many LLM requests
- Very complex
- Multiple possible failure points

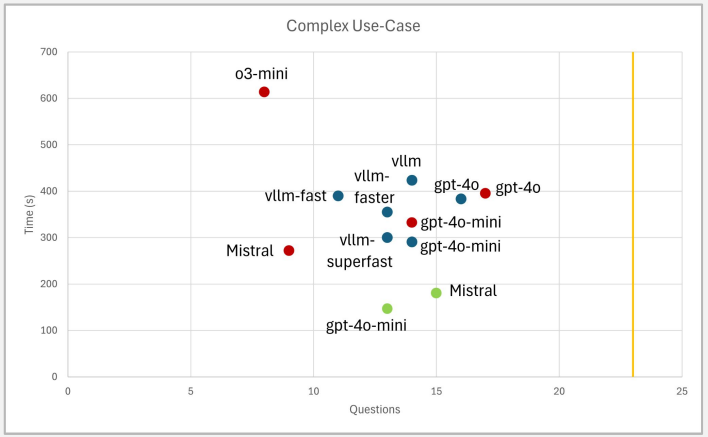


# OPACA LLM UI

## Method Comparison




Simple:
Questions will lead to exactly 1 tool call.




Complex:
Questions will lead to more than 1 tool call.

- Simple
- Tool
- Orchestration

# Live Demo

# Interactive Session

1. **Connect to the Guest-Wifi**
2. **Go to http://10.42.7.124/**, click on “BPMN Editor” and “LLM Prototype”
3. **Grab one of the Handouts** and try to follow the instructions.

**BPMN-Editor**

- Familiarize with the Editor
- Model a simple Example Process
- Make that Process Executable
- Execute the Process
- Try out the LLM-Integration

**OPACA-LLM**

- Connect with OPACA
- Familiarize with the System
- Configure the LLM
- Perform different sample queries
- Inspect Logging/Debugging Output

# Feedback

## Feedback?

### Please take our survey!

https://forms.gle/cnFMnwDowDzoNC218