



EclipseCon 2022

Building AI Pipelines with Eclipse Graphene

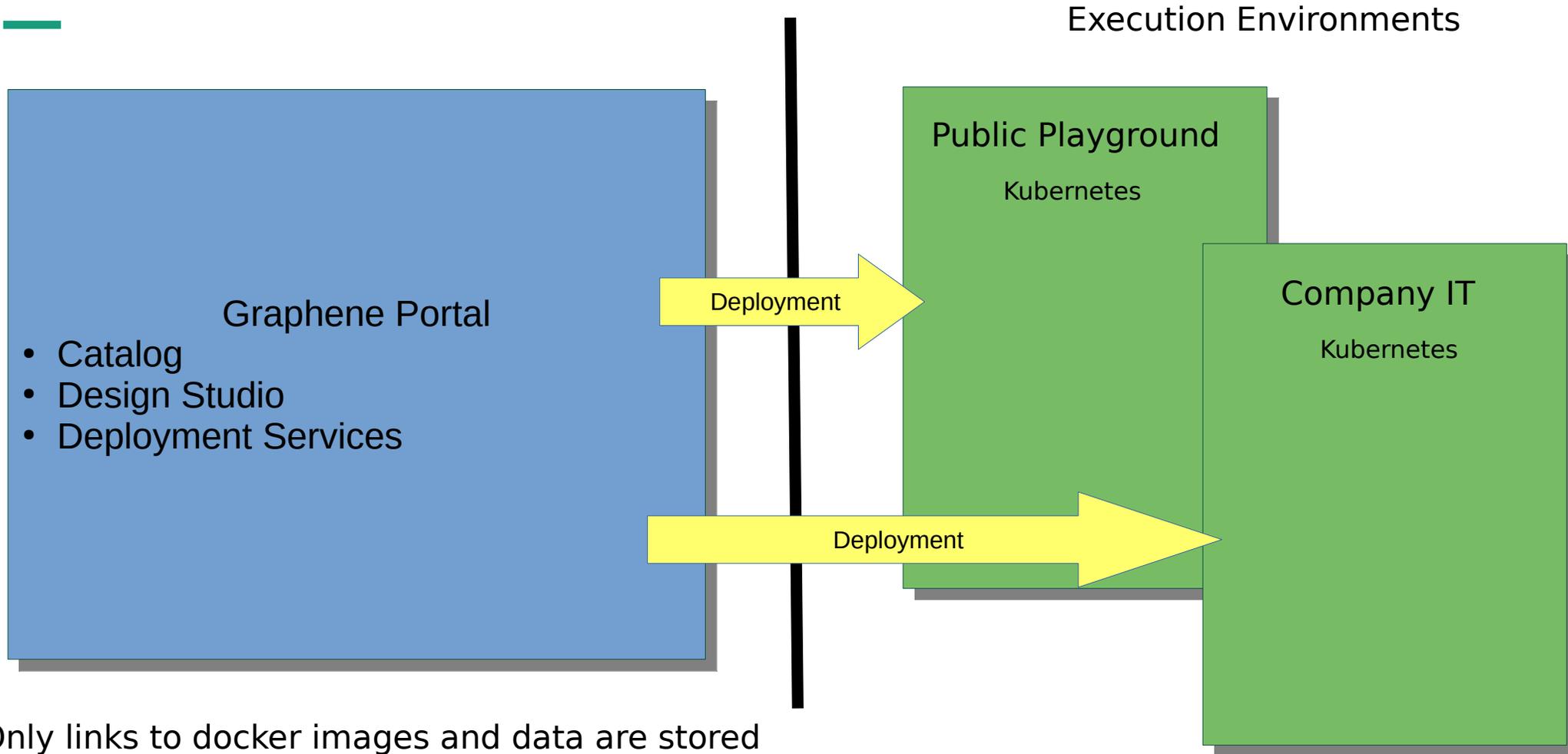
Martin Weiß

History



- In March 2018 the project *Acumos AI* was started at the Linux Foundation
- In 2019 the project *AI4EU* started to build the European AI on Demand Platform
- AI4EU decided to build the experimentation sub-system using Acumos AI
- Beginning of 2020, the *AI4EU Experiments* system was publicly available
- In the first half of 2021, the Linux Foundation decided to stop and archive Acumos AI
- AI4EU then decided to start a fork which is now *Eclipse Graphene* with many fundamental differences
- The initial release for Eclipse Graphene is now in the review process

System Overview



Only links to docker images and data are stored

AI Model Catalog

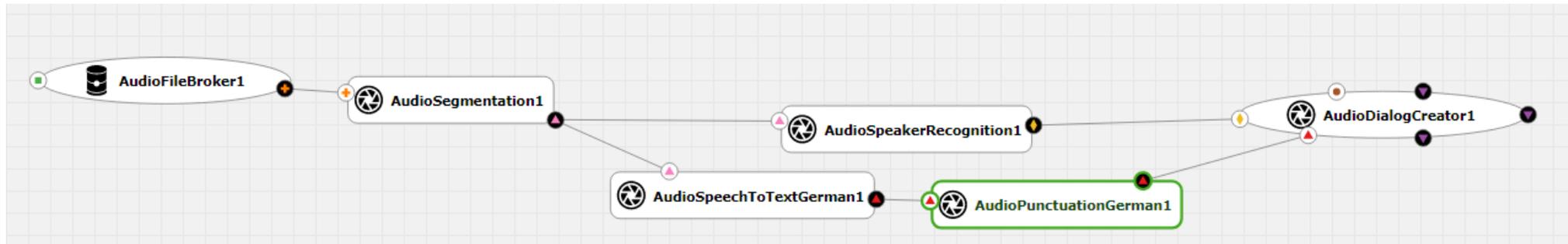
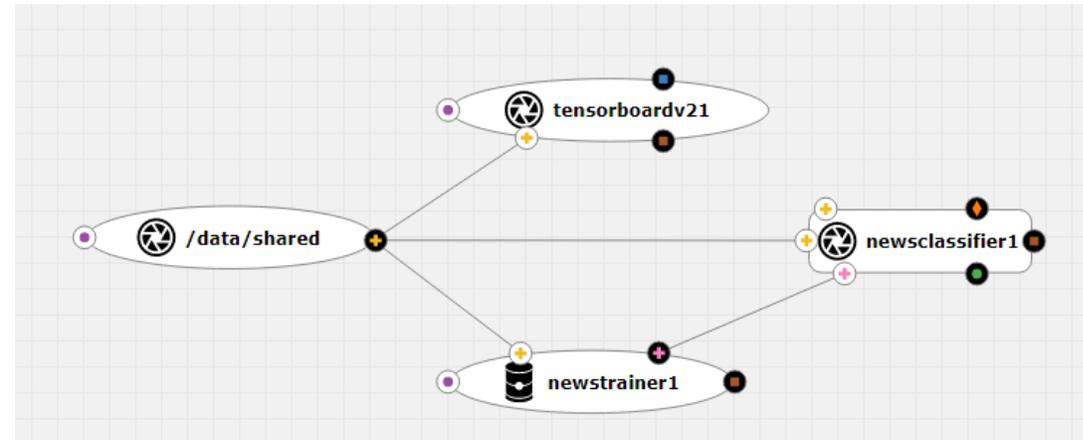
- Catalog of re-usable AI Modules
- Work in teams
- Support for commercial and open source licenses for models

The screenshot displays the 'Marketplace' interface for AI models. At the top, there are navigation tabs: 'All Catalogs' (selected), 'My Favorite Catalogs', and 'Select Favorite Catalogs'. Below this is a breadcrumb 'Home / Marketplace' and a 'BROWSE BY' section with a 'Show All' link. A search bar is present with the text 'Search here'. On the left, there are filter sections: 'Filter By Category' with checkboxes for Classification, Data Sources, Data Transformer, Prediction, and Regression; and 'Tags' with various model categories like Binary Classification, owl, Training, timeseries, electrical load forecasting, manufacturing, ai4agri, GUI, databroker, Tutorial, python, integrative ai, Air quality, hello world, Pose Estimation, VideoModels, classification, MultiClass Classification, housepriceprediction, connector, SensorThings, computer vision, Sentiment Analysis Model, Inspiration, A, CNN, covid predict, transformer, MNIST, Random Forests, and AudioMining. The main area shows a grid of model cards, each with a logo, title, author, date, and status (e.g., 'New'). The cards include: 'i-nergy-load-forecasting' (Energy Load Forecasting), 'VideoObjectRecognition', 'forWoT', 'Sudoku Tutorial', 'AWDrugsModel', 'covid predict', 'AI4Agri-qualitypredictor', 'FaceAI', 'OGC' (Making location count.), 'ITAINNOVA' (INSTITUTO TECNOLÓGICO DE ARAGÓN), and 'AI4Industry Pilot Solution'. Each card also features a star rating, a '1st Call for Solution' button, and icons for comments, views, downloads, and favorites.

Compose AI Pipelines in the Design Studio



- Training pipeline with Tensorboard Integration
- Shared-folder concept
- Show matching models
- Application Pipeline



AI Playground



- Execution Environment
- One-Click-Deployment
- Based on Kubernetes

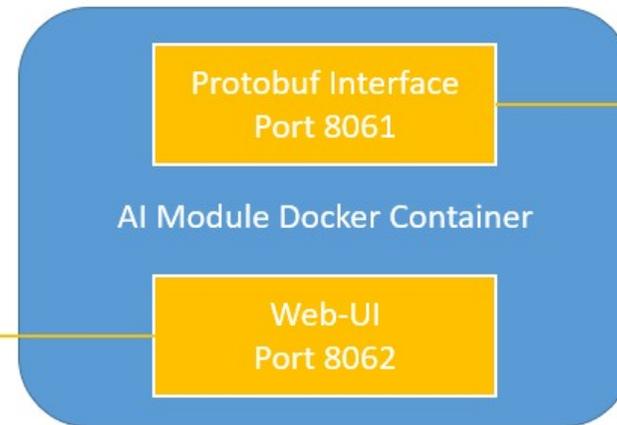
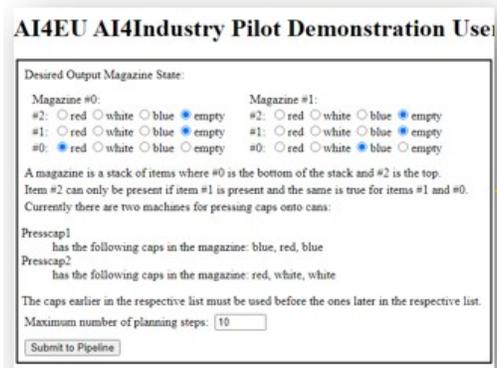
The screenshot displays the KI NRW AI Playground interface. At the top left is the KI NRW logo. Below it, a sidebar shows 'TrainingPipeline' and 'sudoku-stream'. The main content area is titled 'Deployment: TrainingPipeline' and features a control bar with 'Run', 'Reset', 'Delete', 'Logs', and 'Status: Ready' buttons. Below this is a table with columns for 'Status Check', 'Nodename', 'Status Details', 'Logs', and 'WebUI/Folder'. The table lists five components, all with a green checkmark status.

Status Check	Nodename	Status Details	Logs	WebUI/Folder
✓	newsclassifier1	show	view	
✓	newstrainer1	show	view	
✓	orchestrator	show	view	
✓	tensorboardv21	show	view	
✓	SharedFolder	show		

Container Specification for composable AI-Modules



- Docker Container with additional properties
- public interface in `protobuf` / `grpc`
- optional Web-UI (highly recommended)
- No Lock-In: it continues to be a normal Docker container
- If done right, the same Docker container can be used across different execution environments (e.g. GPU or no GPU)



```
// set used version of protobuf
syntax = "proto3";

// set unique package name
package fraunhofer_demo_cpp_iris.v1

// define input data structure
message IrisDataFrame {
  repeated double sepal_length = 1;
  repeated double sepal_width = 2;
  repeated double petal_length = 3;
  repeated double petal_width = 4;
}

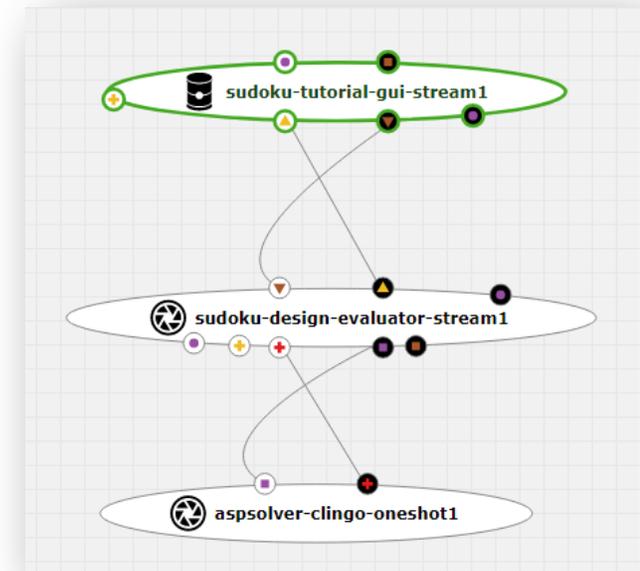
// define output data structure
message ClassifyOut {
  repeated int64 value = 1;
}

// define exposed service
service Model {
  rpc classify (IrisDataFrame) returns (ClassifyOut);
}
```

Orchestrator



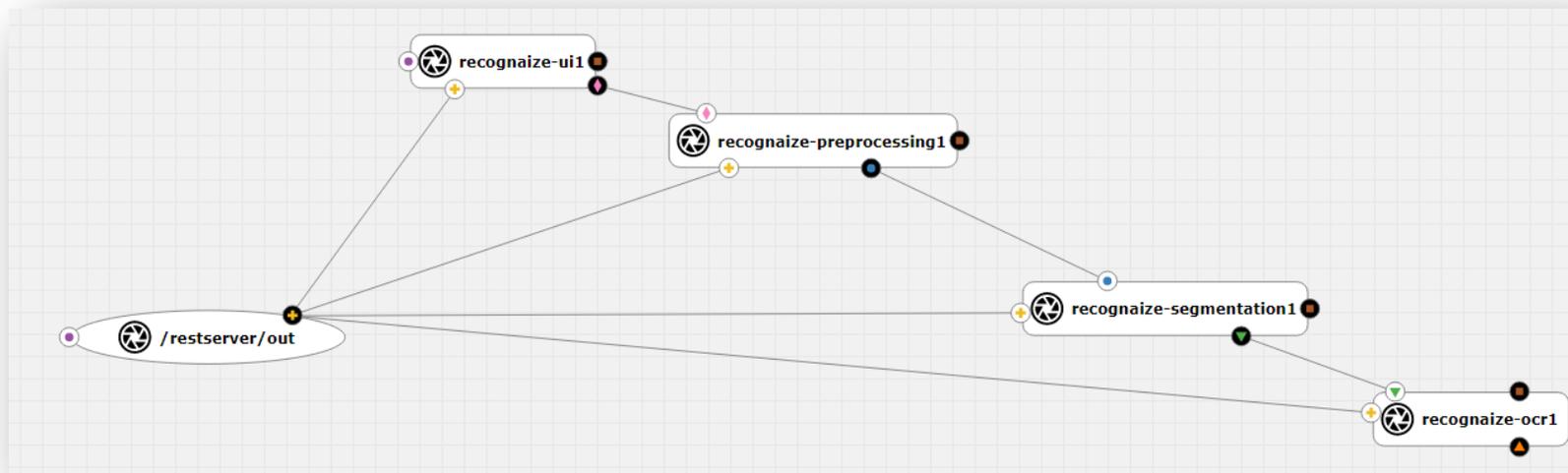
- The orchestrator executes the pipeline according to the topology file (generated by the design studio)
- The orchestrator dispatches the messages among the nodes following the edges
- Thanks to protobuf/grpc the communication stubs can be generated on the fly
- Support for grpc streaming:
 - Sensor-Input
 - UI-Input
 - Media-Streams
 - The edges can be served in parallel
- Different orchestrator implementations possible



Pipeline Definition (Topology)



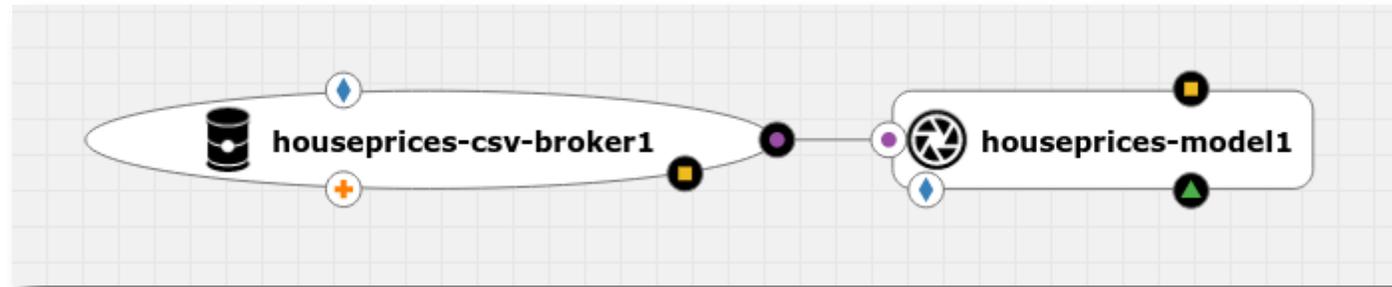
- A JSON-file which contains nodes and edges (technology neutral)
- Is generated by the Design Studio, but can be generated elsewhere (e.g. for AutoML)
- During deployment, the specific configurations for an execution environment are generated (MLOps)
- Different orchestrators are possible
- Shared Folder support



Databroker



- A Databroker is a shallow component that makes data available to the pipeline
- Specific for an execution environment (e.g. company IT)
- Can access the data source (credentials)
- Does not contain the dataset!
- We have already a prototype of an Eclipse Dataspace Connector



Graphene Roadmap



- Support for recursive datastructures
- Jupyter-Connect: one click deploy of a model along with a preconfigured Jupyterlab
- Benchmarking
- Reproducibility
- Hybrid AI Pipelines
- Deploy to HPC
- AutoML

Graphene Use Cases



- General: Building ai-pipelines from re-usable ai-modules
- AI Research: model training, benchmarking, jupyter-connect
- Education: practical live demonstrations and student excercises in the playground
- MLOps: simplify deployment, training pipelines, data cleaning pipelines, can support many tools
- Model showcase, catalog (internal, external)
- Application: let non ai-experts compose and deploy pipelines for thier business domains
- Collaboration: work in mixed teams on PoCs
- Automated certification

Useful Links



- AI4EU Experiments: <https://aiexp.ai4europe.eu/#/home>
- Eclipse Graphene Project: <https://projects.eclipse.org/projects/technology.graphene>
- Eclipse Graphene Gitlab: <https://gitlab.eclipse.org/eclipse/graphene>
- Container Spec: https://gitlab.eclipse.org/eclipse/graphene/tutorials/-/tree/main/Container_Specification
- Tutorials: <https://gitlab.eclipse.org/eclipse/graphene/tutorials>

Contact

Martin Weiß
Senior AI Solution Architect
martin.welss@iais.fraunhofer.de

Fraunhofer-Institut für intelligente Analyse- und
Informationssysteme IAIS
Schloss Birlinghoven 1
53757 Sankt Augustin
Germany

www.iais.fraunhofer.de