

Stitching Together Innovation with FABRIC Users

***Federated Learning with microservices
on the FABRIC testbed***

Alexandros Koufakis
Jake Jongejans

20 May 2025



Welcome and Introduction





Alexandros Koufakis

University of Amsterdam
Scientific Programmer

Complex Cyber Infrastructure group





Jake Jongejans
University of Amsterdam
MSc Student

Agenda

1. Use Cases
2. Federated Learning
3. Implementation as Microservices
4. Deployment on FABRIC
5. Demonstration
6. Q&A



Use-Case 1: Prediction of Building Energy Consumption



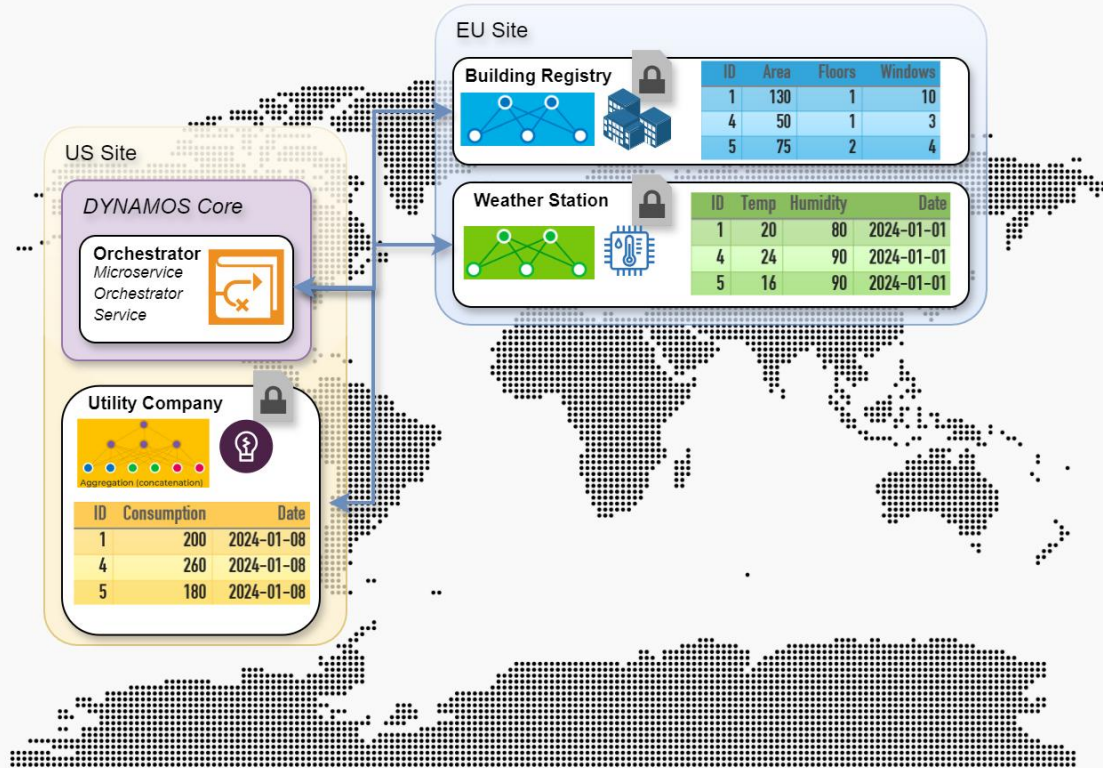
Prediction of Building Energy Consumption

- Multiple factors that affect building energy consumption (e.g. building characteristics, sustainability features, social factors, environmental factors)
- Understanding energy consumption can help with adoption of renewable energy sources, guide sustainability renovations, financial benefits etc.




Prediction of Building Energy Consumption

- Potentially geographically distributed
- Different agents hold separate contributing datasets



Prediction of Building Energy Consumption

- Building registry (building features)
- Weather station (climate data)
- Utility Company (**energy consumption**)



Target
Variable

ID	Area	Floors	Windows	ID	Temp	Humidity	Date	ID	Consumption	Date
1	130	1	10	1	20	80	2024-01-01	1	200	2024-01-08
4	50	1	3	4	24	90	2024-01-01	4	260	2024-01-08
5	75	2	4	5	16	90	2024-01-01	5	180	2024-01-08

Use-Case 2: Predictive Maintenance for Aircraft Engines



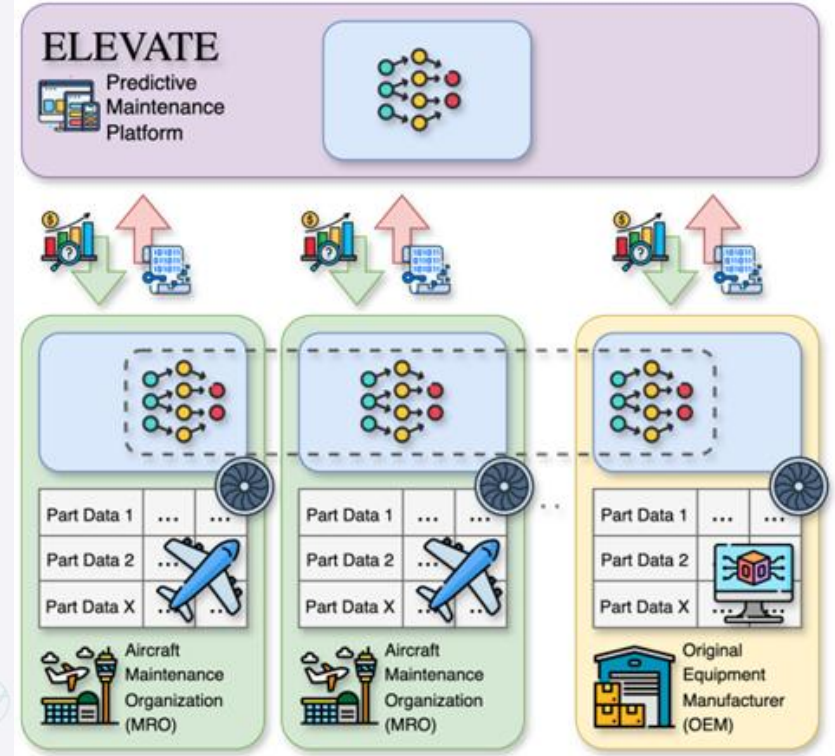
Predictive Maintenance for Aircraft Engines

- Multiple sensors continuously monitor the function of aircraft engines.
- Those measurements can predict when the engines might fail so that they can react proactively.
- Safer air travel, better service, fewer delays, etc



Predictive Maintenance for Aircraft Engines

- Different **airline companies** and **equipment manufacturing companies** can contribute to creating predictive maintenance ML solutions.
- Potentially geographically distributed
- Private datasets



Federated Learning



Federated Learning (FL)

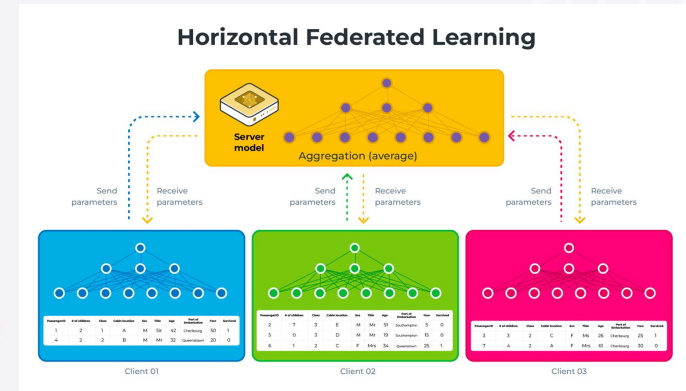
- FL is a machine learning approach where multiple participants collaboratively train a model without sharing their raw data
- Data stays local (on-premise)
- Only model updates (weights, gradients) are shared
- Ensures data privacy and regulatory compliance



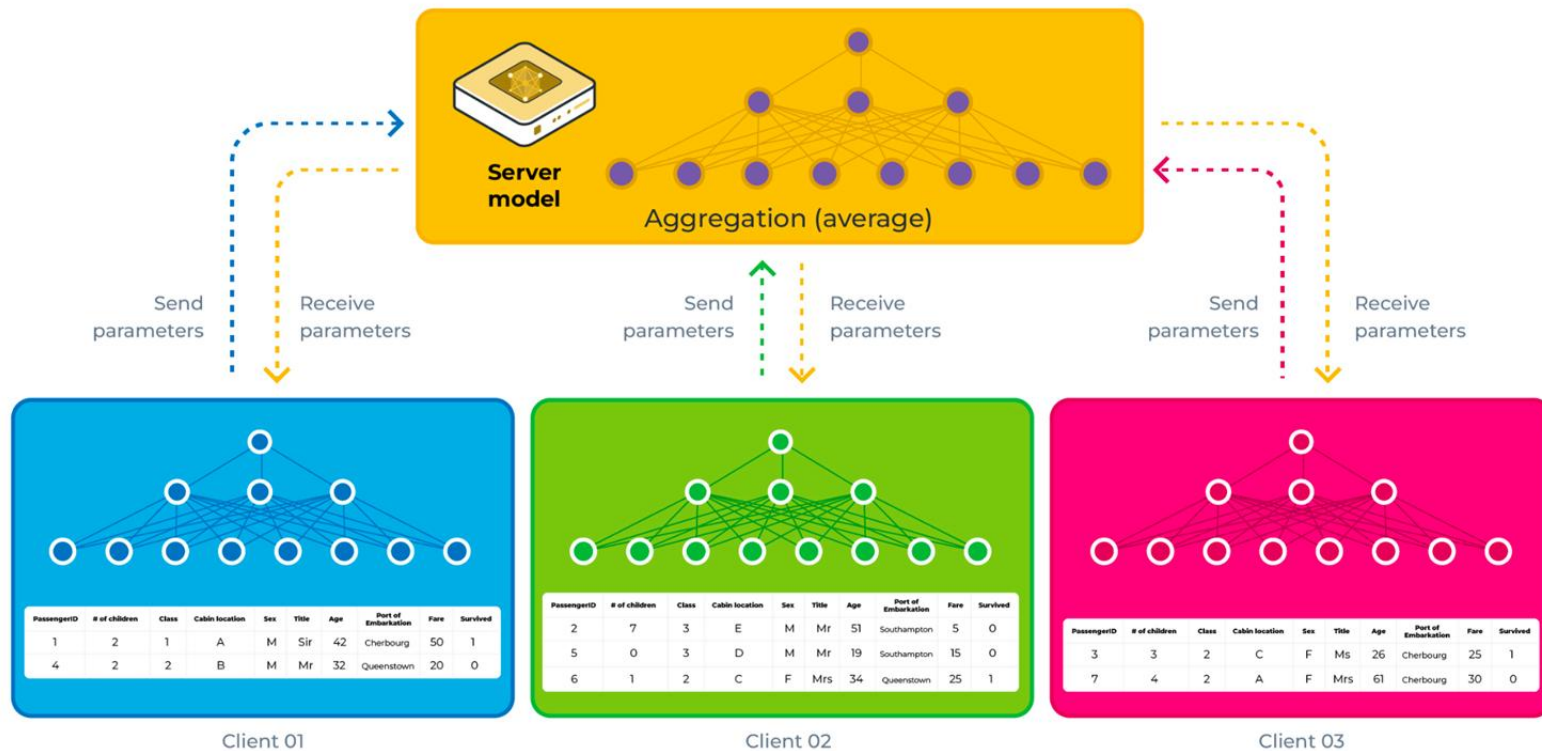
Horizontal Federated Learning

Participants share **the same feature space** but have data from different users or entities.

- Data has same structure, different instances
- Suitable for organizations with similar data schemas



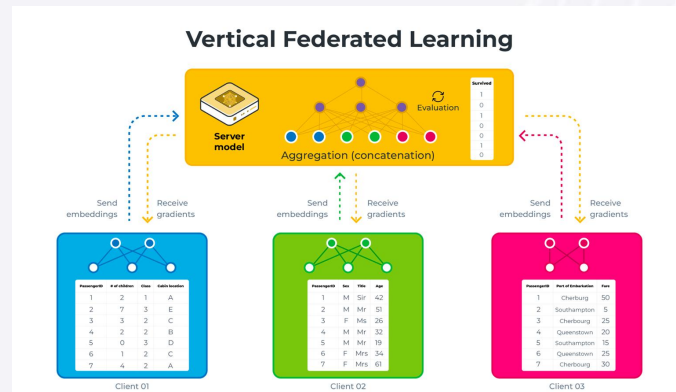
Horizontal Federated Learning



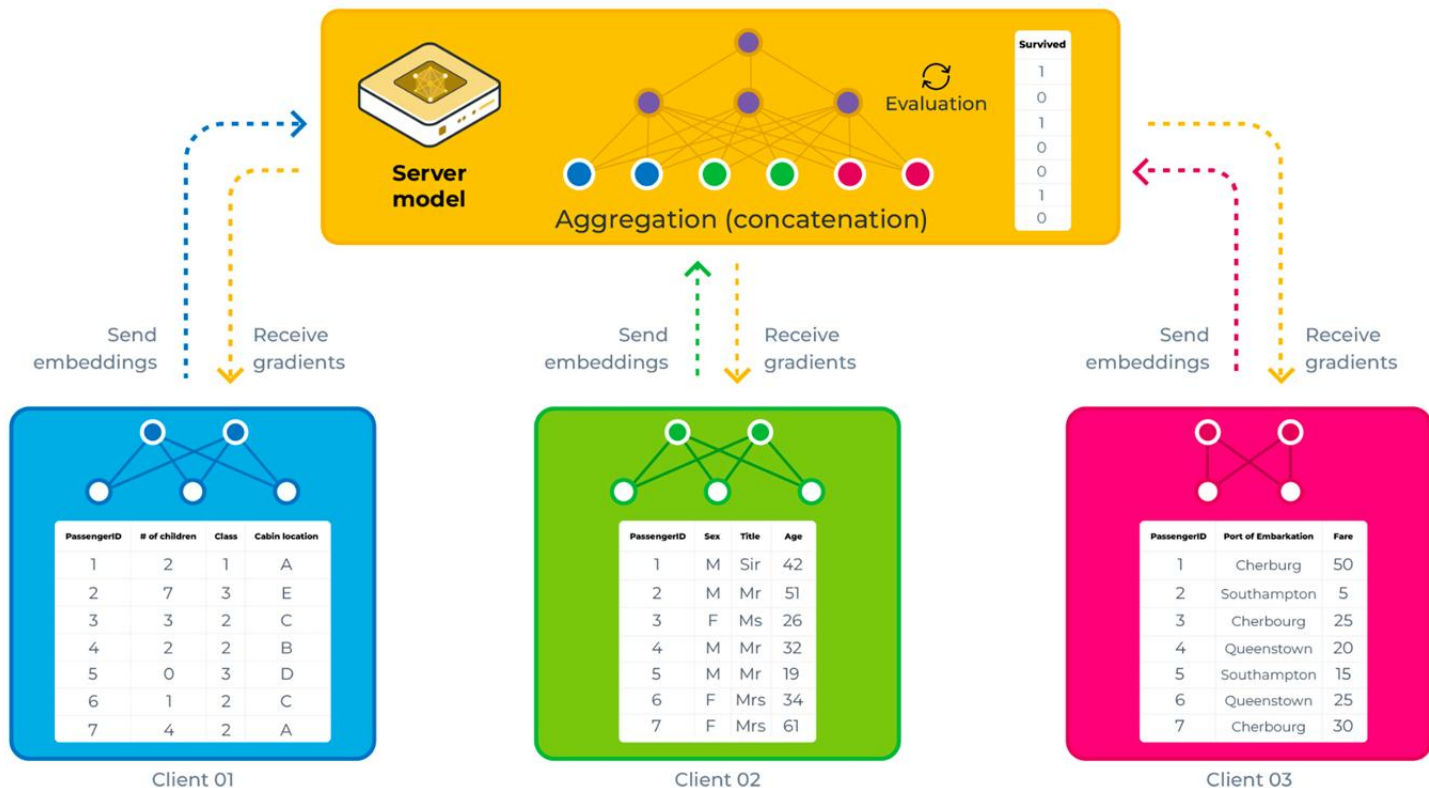
Vertical Federated Learning

Participants have **different features (data columns)** about the **same entities**.

- Datasets have complementary features
- Requires entity alignment



Vertical Federated Learning



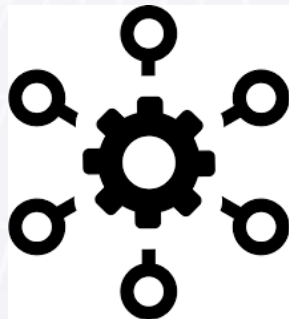
Microservices - DYNAMOS



Microservices

Microservice architecture is an architectural pattern that organizes an application into a collection of loosely coupled, fine-grained services that communicate through lightweight protocols

- Decentralized: Each service is developed, deployed, and scaled independently.
- Technology-agnostic: Services can use different languages or frameworks.
- Composable: Services communicate via APIs (often REST or gRPC).

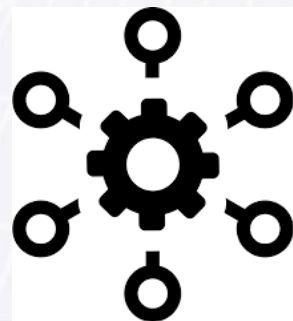


Microservices

Benefits:

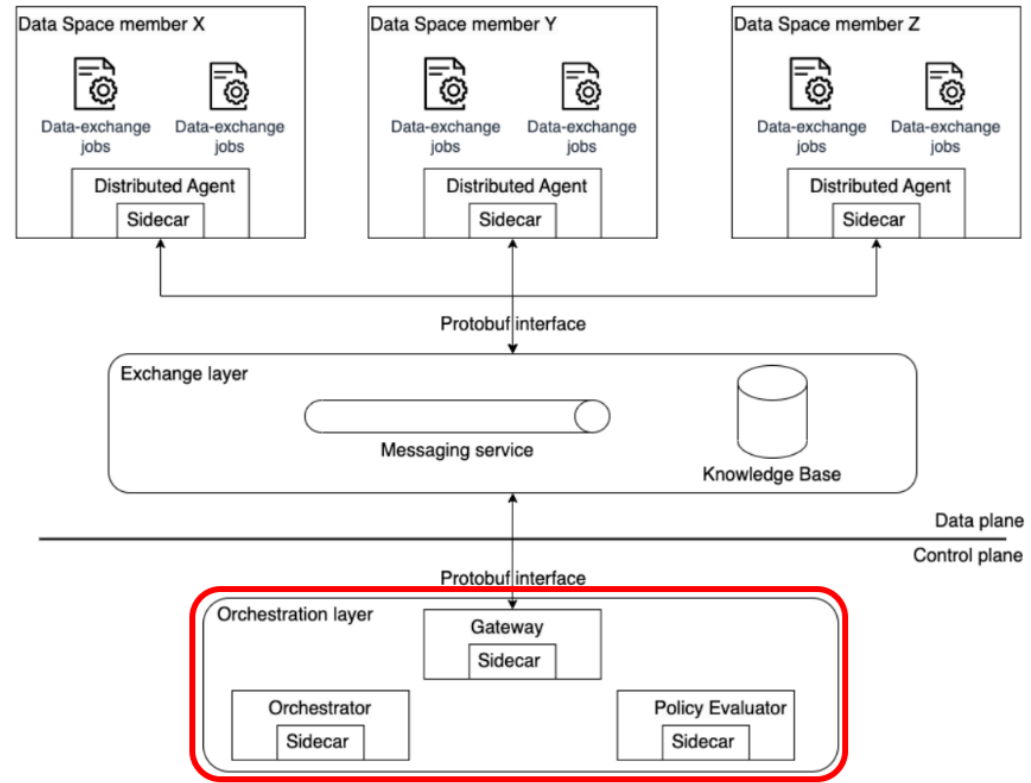
- Faster development & deployment
- Easier scalability and maintenance
- Better fault isolation and system resilience

Separate microservices for data preprocessing, model training, aggregation, and monitoring in a federated learning pipeline.

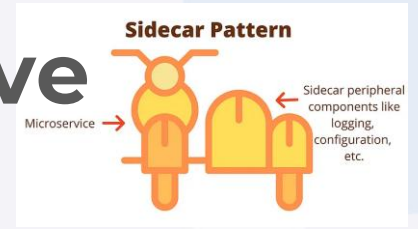


DYNAMOS: Dynamically Adaptive Microservice-based OS

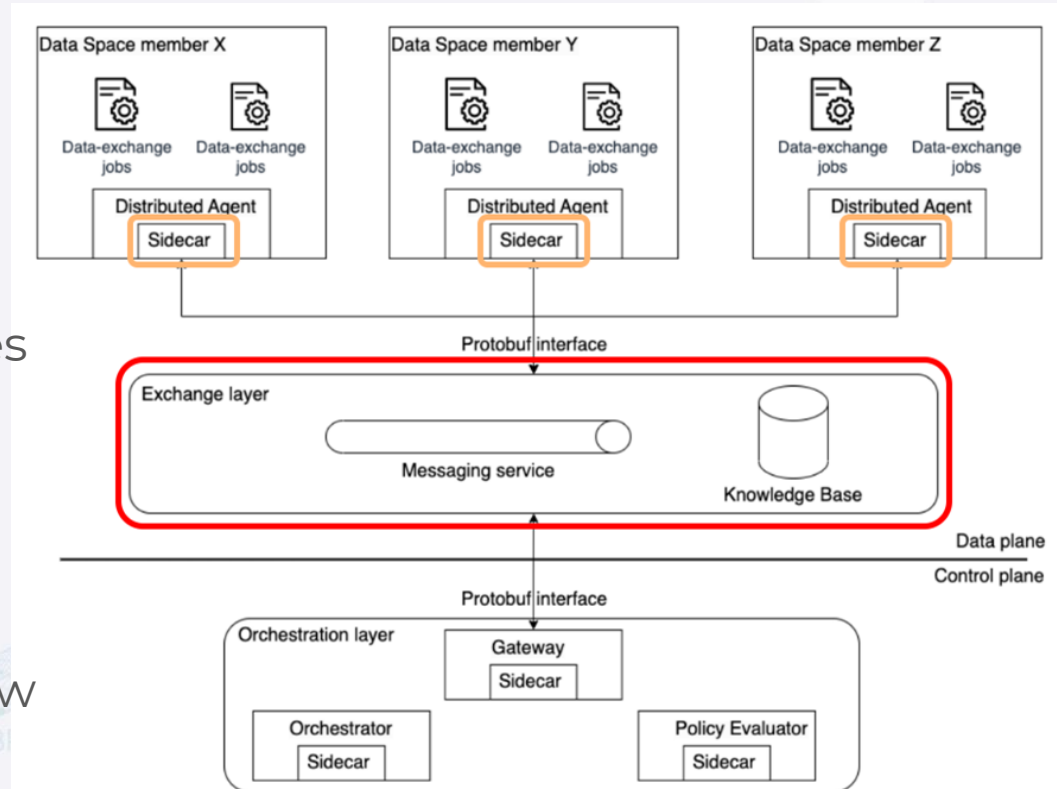
- **Gateway:** request handling
- **Policy Evaluator:** Accepted collaborations, data access, etc
- **Orchestrator:** microservice chain generation (what operations need to take place)



DYNAMOS: Dynamically Adaptive Microservice-based OS

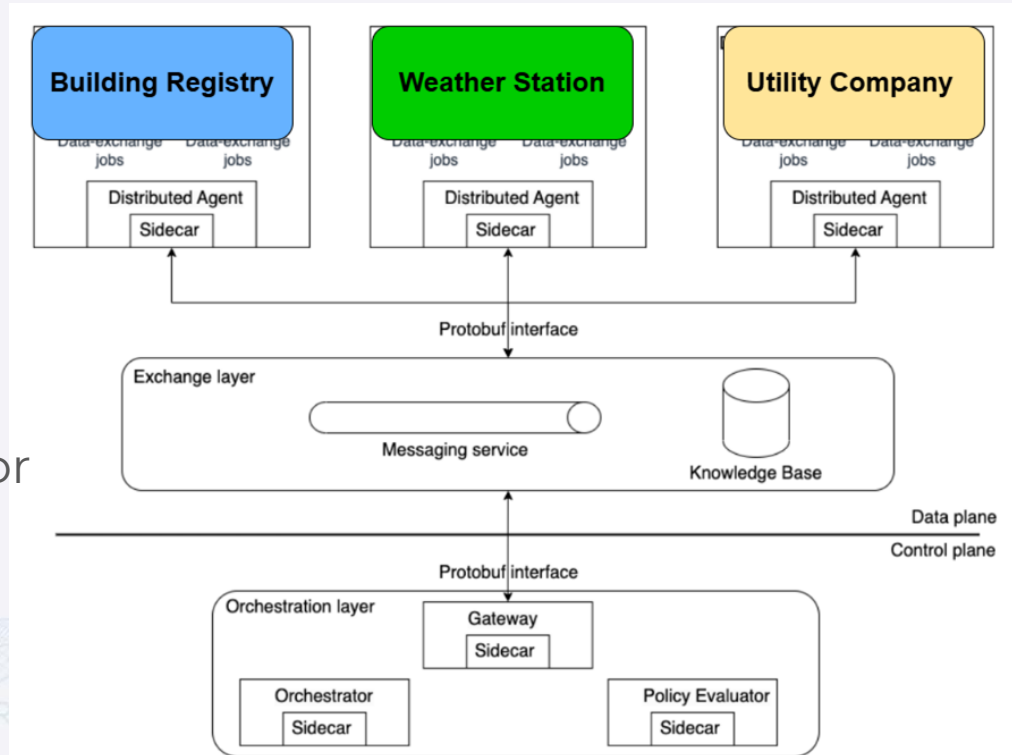


- **Messaging Queue:** Orchestration and management (RabbitMQ)
- **Sidecar Pattern:** Decouples the functionality from communication
- **Remote Procedure Calls:** Microservice chain data flow (gRPC)



DYNAMOS: Dynamically Adaptive Microservice-based OS

- **Agents:** represent different Federated Learning participants
 - Geographically distributed
 - One persistent service for requests
 - Ephemeral jobs for execution



Automated Deployment of DYNAMOS on FABRIC



FABRIC: The Basics

Why FABRIC?

- **Dedicated research infrastructure**
- **Intercontinental locations for nodes**
- **Dedicated high-bandwidth network connections**
- **Scriptability of deployments through the FABRIC API**



FABRIC: Deploying Kubernetes

We harnessed the power of FABRIC's infrastructure to create a Kubernetes cluster across borders.

- **Fully automated script**
- **Configurable nodes (resources, location)**
- **Seamless setup with Kubespray**
 - **Shaping the network topology with the FABRIC API**
 - **Automatically generate Kubespray configurations for this network topology**
- **Expose all FABRIC nodes to one another for an inter-site cluster**



Configuring DYNAMOS for FABRIC

DYNAMOS seamlessly integrates with Kubernetes clusters.

- **Automate the configuration of DYNAMOS**
 - Define agents based on the FABRIC node configuration
 - Constrain Kubernetes nodes to the respective FABRIC nodes
- Remove the DYNAMOS dependency on Docker for a smaller footprint
- The rest of the setup was seamless on FABRIC



The Benefits

Why are we utilising FABRIC as infrastructure backbone?

- **The stability of DYNAMOS on FABRIC is much better due to the network backbone**
- **Automated and reproducible deployments of the entire DYNAMOS system**
- **Seamless configuration of different DYNAMOS clusters, from single-site clusters to intercontinental clusters spanning Asia, USA and Europe**



DYNAMOS runs on FABRIC: What is next?

Our current research using FABRIC as our infrastructure is creating a DYNAMOS-native Vertical Federated Learning implementation that harnesses the power of policies



Demo Video





Optional: Live Code Showcase



Q&A



Acknowledgements

In addition to our presenter, we would like to acknowledge the behind-the-scenes team that diligently worked to bring this webinar to production:

- **KC Wang**, content
- **Chelsea Davis, Carolena Charalambous, Bryttany Todd**, project management
- **Jayasree Jaganatha**, social media specialist



Resources

Call to Action

FABRIC Matrix:
<https://bit.ly/FABRICmatrix>

Connect With Us

Newsletter Signup: bit.ly/FABRICnewsletter

Office Hour Sign Up: bit.ly/FABRIC-Office-Hours

Other Resources

Website: bit.ly/m/FABRICtestbed

YouTube: youtube.com/@fabrictestbed

FABRIC Account: portal.fabric-testbed.net

Ambassador Program: bit.ly/FABRIC-Ambassador-Program

FABRIC LinkedIn: linkedin.com/company/fabrictestbed

Citing FABRIC: bit.ly/citing-fabric



Thank You for Attending!

Join us for our upcoming webinars:

- **June 17** - Mastering FABRIC: Tips and Tricks Webinar

Visit our YouTube Channel: youtube.com/@fabrictestbed

