

ANESTIS DALGKITSIS
ALEXANDROS KOUFAKIS

ciena

 **prets** embedded
lending



UNIVERSITY OF AMSTERDAM

JORRIT STUTTERHEIM, ALEANDRO MIFSUD, PRIYANKA ATWANI, LEON
GOMMANS, CEES DE LAAT, CHRYSA PAPAGIANNI, ANA OPRESCU

**MICROSERVICE-BASED FABRIC
COLLABORATIVE MODEL TRAINING**

INTRODUCTION

- ▶ **Collaborative Model Training with Federated Learning**
 - ▶ An approach that enables multiple organizations to train machine learning models on decentralized data, while maintaining data privacy and sovereignty.
- ▶ **Key Benefits**
 - ▶ Enhanced privacy and security compared to traditional approaches, making it valuable in sectors with sensitive data (e.g. healthcare, building retrofitting financing).

CONTRIBUTIONS

- ▶ Demonstrate **Secure Collaborative Model Training with Vertical Federated Learning** for the building energy consumption prediction use case.
- ▶ Use **Microservices Architecture** to allow for digital data exchange while ensuring privacy and contractual agreements.
- ▶ Use **FABRIC** to create a realistic, multi-site networking slice and evaluate the implementation.

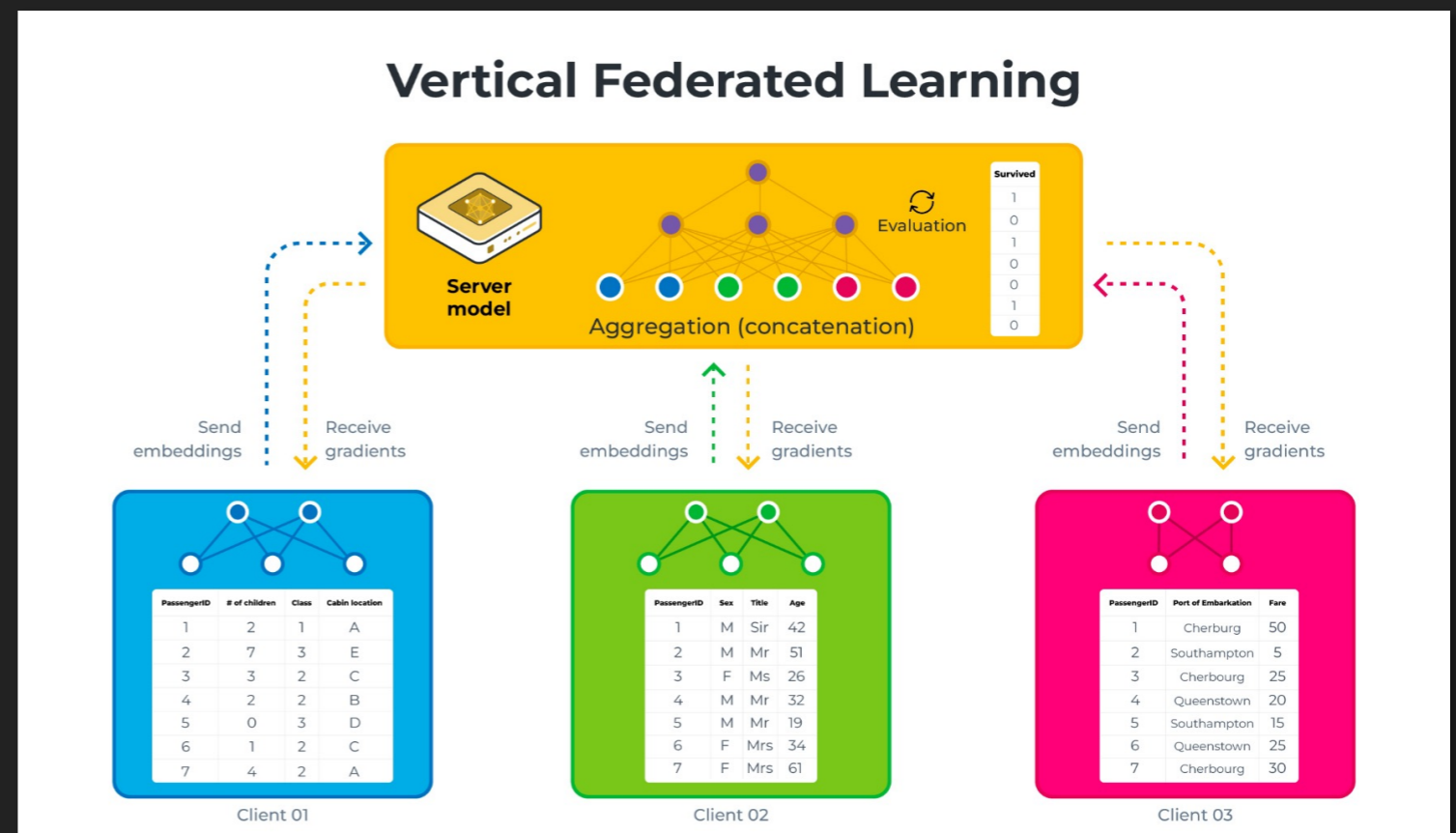
BACKGROUND

▶ DYNAMOS Microservices Orchestration

- ▶ Flexibility, scalability, and resilience by enabling independent deployment, management, and scaling of individual components

▶ Vertical Federated Learning

- ▶ Distributed Model
- ▶ Clients have heterogeneous features
- ▶ Common set of samples among the clients



USE CASE

- ▶ Vertical Federated Learning for **building energy consumption prediction** based on data from different sources.
 - ▶ EU Client 1: Building registry (area, floors, windows)
 - ▶ EU Client 2: Weather data (temperature, humidity)
 - ▶ US Client 1: Energy Provider (energy consumption)

CONSUMPTION IS THE TARGET VARIABLE

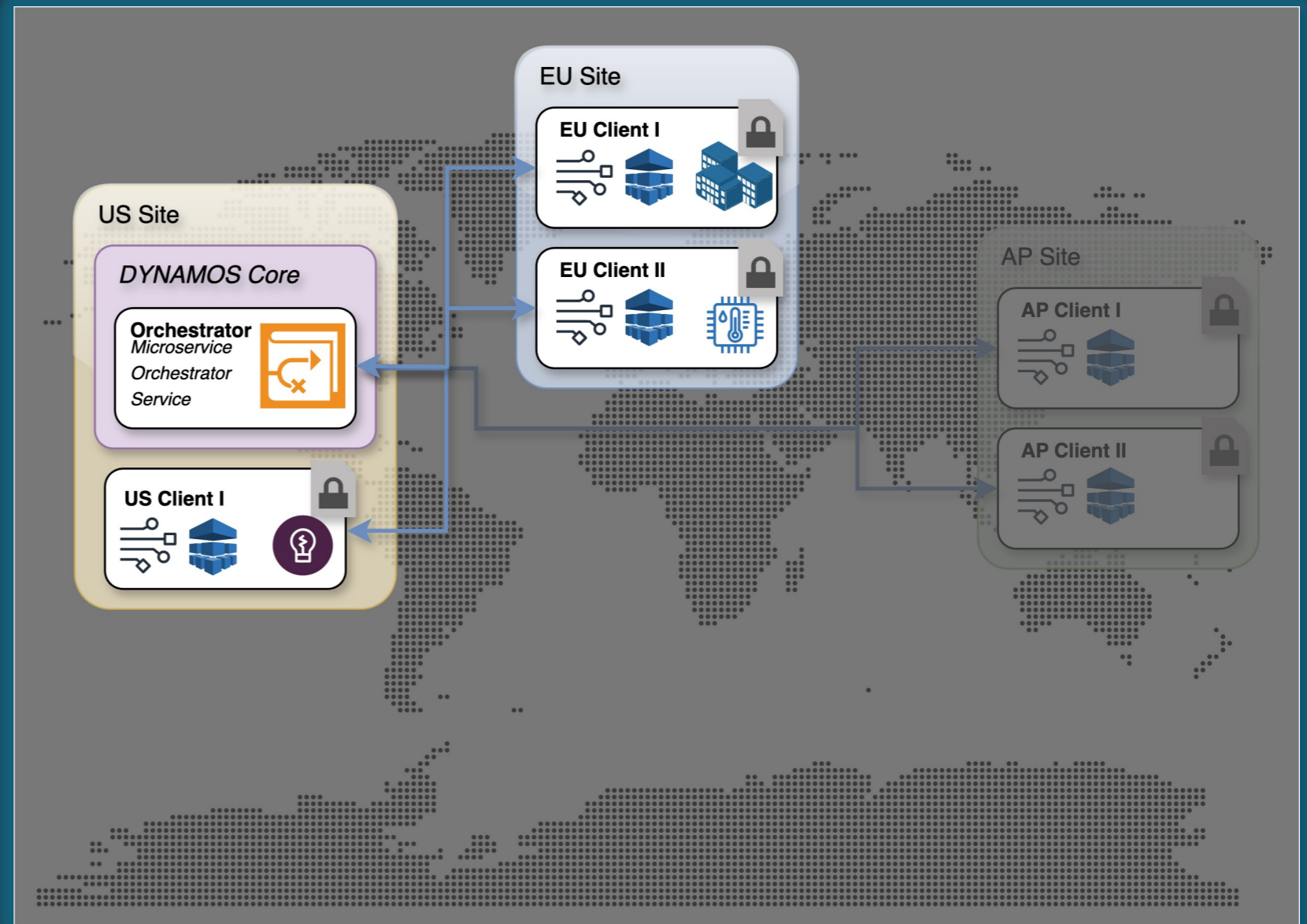
ID	Area	Floors	Windows
1	130	1	10
4	50	1	3
5	75	2	4

ID	Temp	Humidity	Date
1	20	80	2024-01-01
4	24	90	2024-01-01
5	16	90	2024-01-01

ID	Consumption	Date
1	200	2024-01-08
4	260	2024-01-08
5	180	2024-01-08

FABRIC SLICE

- ▶ Demo focused on two international sites.
- ▶ Emulating a scenario with 3 inter-site clients:
 - ▶ 2 clients in EU AMST
 - ▶ 1 client in US CIEN
- ▶ VFL Aggregator is deployed at the US site.
- ▶ L2STS Networking, Transatlantic data transfers between sites.



SLICE DEPLOYMENT

The screenshot displays the FABRIC web interface for a network slice deployment. The main area shows a topology diagram with the following components:

- DALL** (Datacenter Area):
 - dynamos-core**: Contains **dynamos-core-dcnic**.
 - client_2**: Contains **client_2-c2nic**.
- TRANSATLANTIC Network Service**: The central service connecting the two regions.
- AMST** (Atlantic Margin Service Territory):
 - client_1**: Contains **client_1-c1nic**.

Connections are shown between **dynamos-core-dcnic** and **client_2-c2nic** to the **TRANSATLANTIC Network Service**, and between the **TRANSATLANTIC Network Service** and **client_1-c1nic**.

Details sidebar information:

- Project: DESIRE6G experimentation
- Lease Start at: 2024-11-08 15:05:05
- Lease End at: 2024-11-09 15:05:05
- Buttons: [Extend](#)

Top navigation includes: FABRIC, Resources, Experiments, Knowledge Base, JupyterHub, Contact Us, About, Community, Search People/Project, and an AD button.

MICROSERVICES-BASED FABRIC VFL COLLABORATIVE MODEL TRAINING

DEMO

The screenshot displays the DYNAMOS web interface and several terminal windows. The web interface is split into a 'Request' panel on the left and a 'Response' panel on the right. The 'Request' panel contains the following fields:

- Request type:** VFL-demo
- Username:** research@uva.nl
- Password:** [masked]
- Aggregation node:** Utility Company
- Data providers:** Building Registry, Weather Station
- More options:** Bedroom windows, Build year

A blue 'Send' button is located at the bottom of the 'Request' panel. The 'Response' panel shows a diagram of the microservices architecture:

```
graph TD; DYNAMOS((DYNAMOS)) --- UC((Utility Company)); UC --- BR((Building Registry)); UC --- WS((Weather Station));
```

Below the diagram, the text 'Getting gradients...' is visible. To the right, three terminal windows are open, showing the output of a 'docker stats' command:

```
Every 1.0s: docker stats 0d3eb2dfde32 cefb7f2d8b39 89f0... UvA-PF4QQKR2: Fri Nov 15 15:03:04 2024
k8s_flwr-clientapp-3_prets-8d48658d4-ffbp4_prets_635629a9-73b0-4a77-9e29-364f1176f4a9_0: 60.40%
k8s_flwr-clientapp-2_prets-8d48658d4-ffbp4_prets_635629a9-73b0-4a77-9e29-364f1176f4a9_0: 1.35%
k8s_superexec_prets-8d48658d4-ffbp4_prets_635629a9-73b0-4a77-9e29-364f1176f4a9_0: 11.58%
```

Below the terminal windows, three more terminal windows are open, showing the output of a 'ls' command:

```
Every 1.0s: ls --color=auto pret_s-8d48658d4-ffbp4: Fri Nov 15 14:03:09 2024
checkpoint_1731679384.9665449.pth

Every 1.0s: ls --color=auto pret_s-8d48658d4-ffbp4: Fri Nov 15 14:03:08 2024
checkpoint_1731679384.9211307.pth

Every 1.0s: ls --color=auto pret_s-8d48658d4-ffbp4: Fri Nov 15 14:03:08 2024
checkpoint_1731679385.426571.pth
```

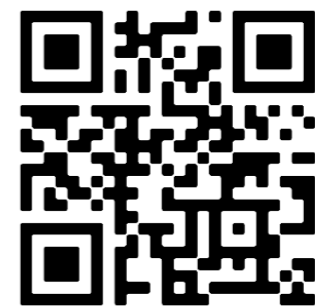

ANESTIS DALGKITSIS ALEXANDROS KOUFAKIS



JORRIT STUTTERHEIM, ALEANDRO MIFSUD, PRIYANKA ATWANI, LEON GOMMANS, CEES DE LAAT, CHRYSA PAPAGIANNI, ANA OPRESCU

GitHub

INFORMATION



FIND US: CIENA BOOTH 1940