



Federated Learning with NVFlare

K. Alibabaei, V. Kozlov, L. Berberi, B. Esteban



Co-funded by
the European Union

What is Federated Learning?

- Enables collaborative learning with private data by sharing trained weights instead of raw data.



What is NVFlare?

- **NVFlare (NVIDIA Federated Learning Application Runtime Environment):**
 - A domain-agnostic, open-source, extensible SDK for adapting ML/DL workflows to a federated paradigm.
 - Allows customization and experimentation with workflows and deployment scenarios.
- [NVFlare Official Website](#) and [github](#)

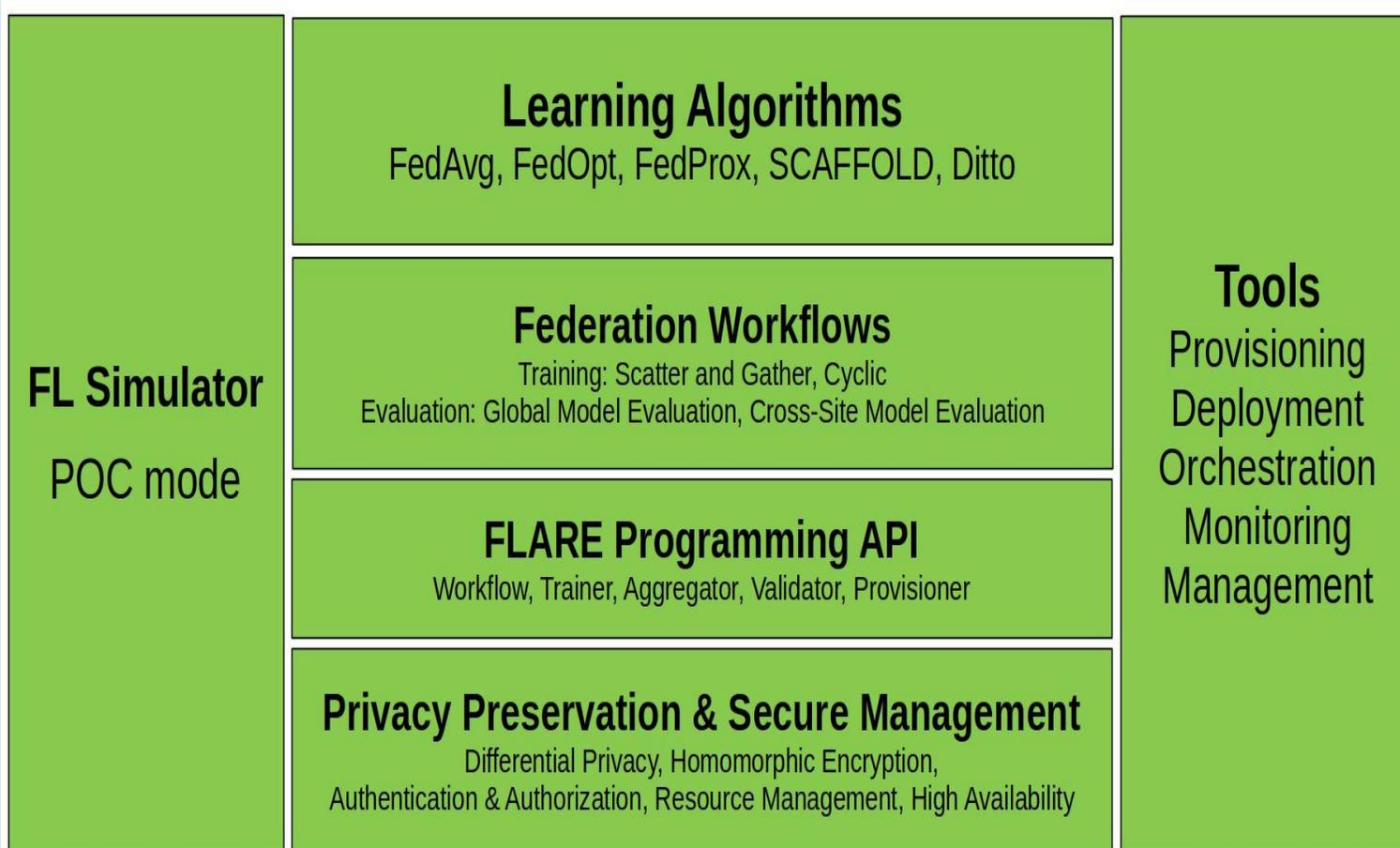
NVIDIA FLARE components

AI4

eosc



Co-funded by
the European Union

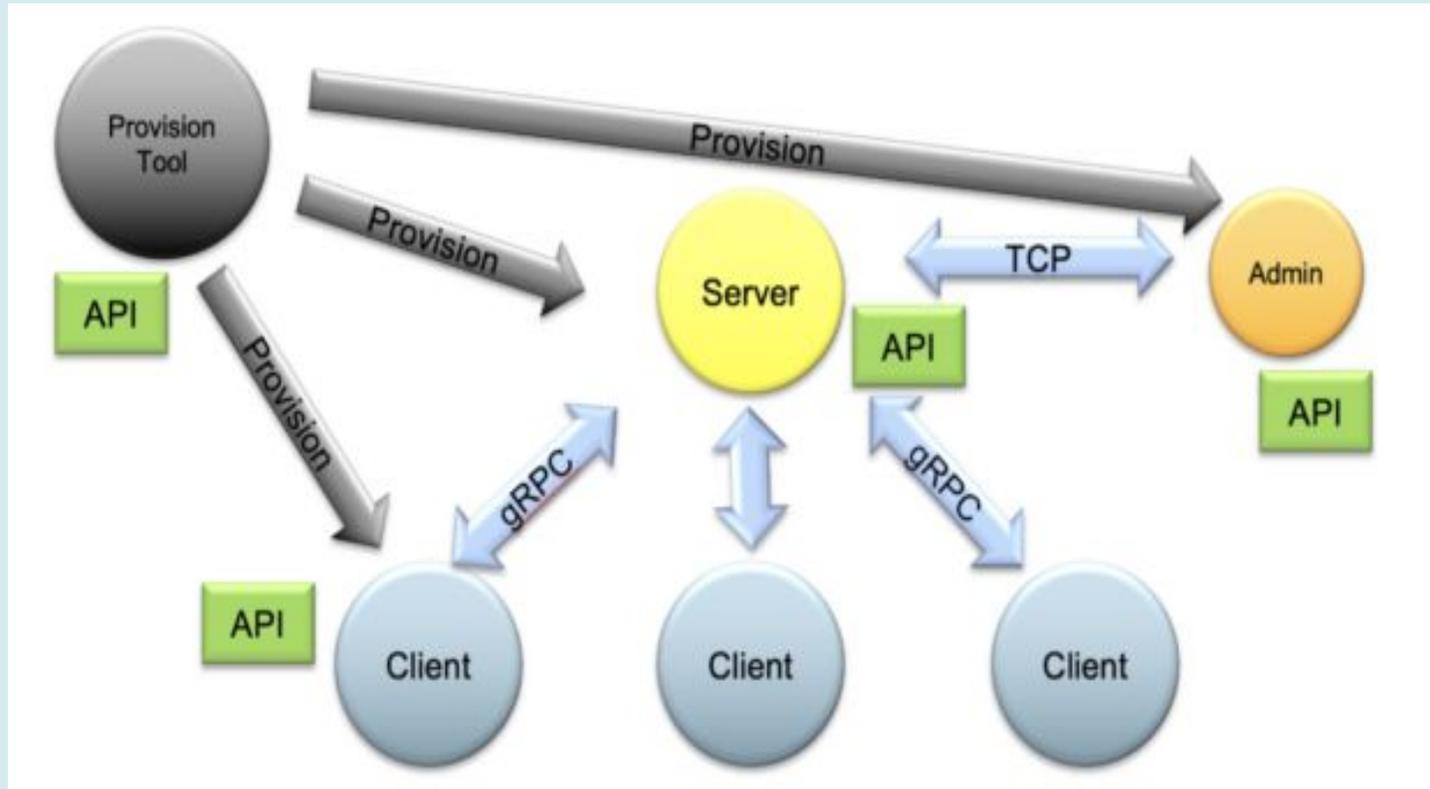


Terminologies

- **Org:** Participating organization.
- **Site:** The computing system running NVFLARE.
- **FL Server:** Coordinates clients on the server.
- **FL client:** Executes tasks and learns on local data.
- **User (Admin):** Human participant in FL project:
 - Roles
 - *Project Admin*
 - *Org Admin*
 - *Lead Researcher*
 - *Member Researcher*

Provisioning in NVIDIA FLARE

- A necessary first step in establishing a federation is **provisioning** to establish the identities of the server, clients, and admin clients.



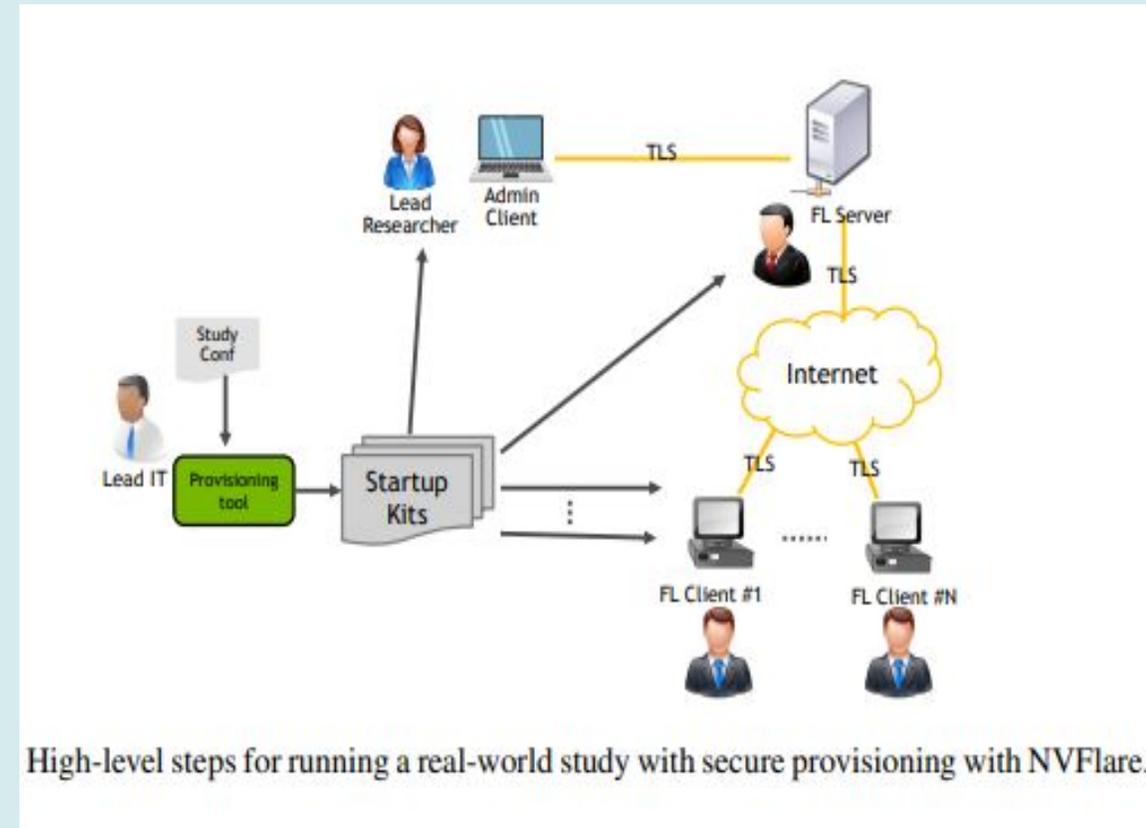
Provisioning in NVIDIA FLARE

AI4

eOSC

Co-funded by
the European Union

- Generate startup kits for each client, server and users which usually include:
 - Authentication Credentials.
 - Authorization Policies.
 - Tamper-Proof Mechanisms (Signatures).
 - Convenient commands (Shell scripts) for launching participants.



- Users can connect to the FL System with the
 - **Admin Console**
 - Each user's startup kit contains a shell script for launching a personal console via the command line.
 - **Admin APIs:**
 - USE the corresponding startup kit location to start a session from python script.
- **Console/APIs** Enables communication with the server and clients based on permissions and access levels.

High Availability: Scalability and Redundancy

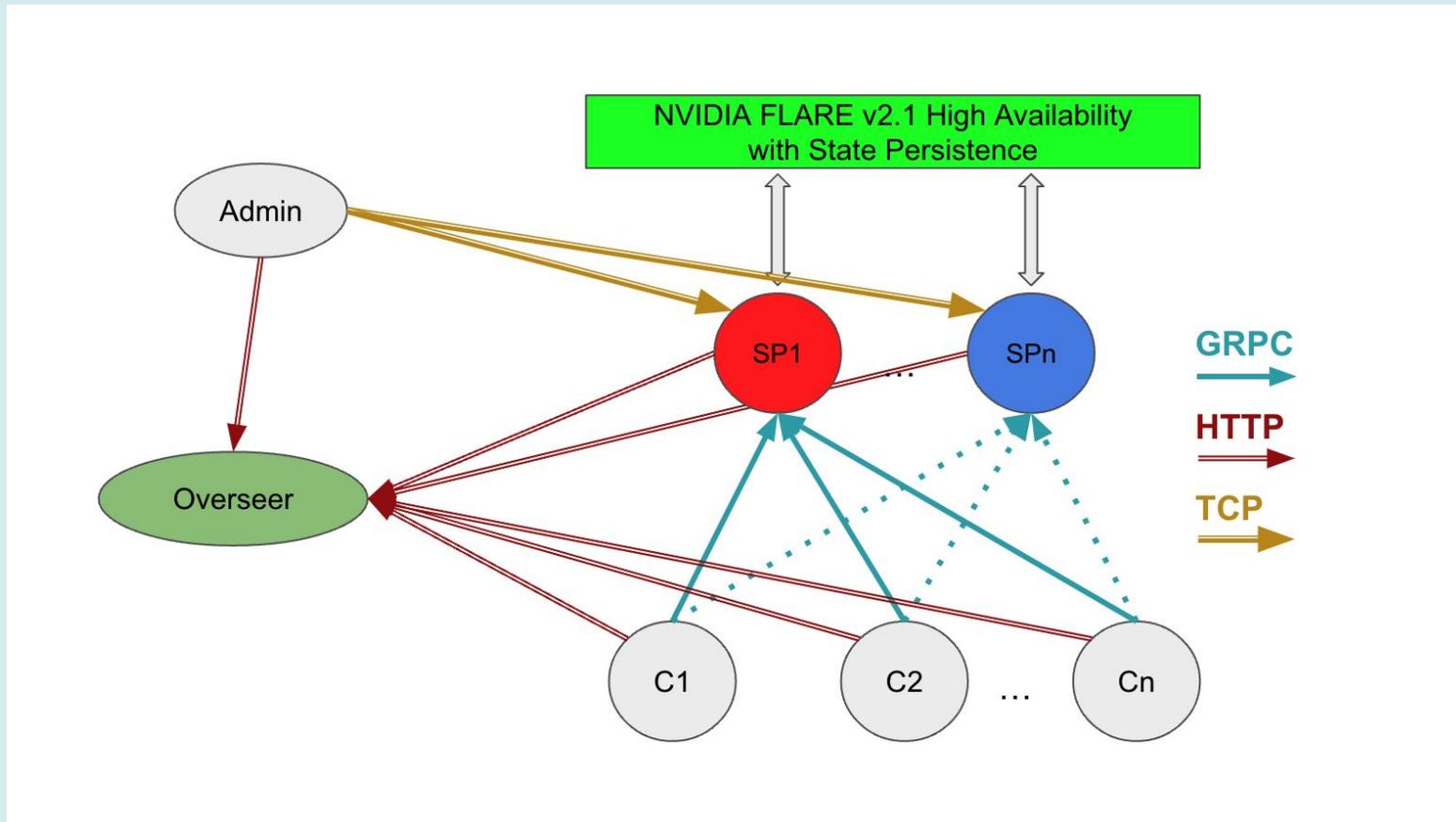
AI4

eOSC



Co-funded by
the European Union

- **High Availability:** There can now be any number of FL servers.



Authorization policy:

- Organizations can define strict policies for controlling access to resources and FL jobs.
 - Examples:
 - Limit Custom Code Access,
 - Access Source Restrictions,
 - Disable Remote Shell Commands,
 - job level authorization
- This locally defined policy is loaded by FL Clients owned by the organization

NVIDIA FLARE Application:

- The **NVIDIA FLARE application** defines how the server and client should run.
- Project structure:
 - project_folder/
 - config/
 - config_fed_client.config
 - config_fed_server.config
 - custom/
 - custom_code1.py
 - custom_code2.py
 - ...

Exploring Different Modes

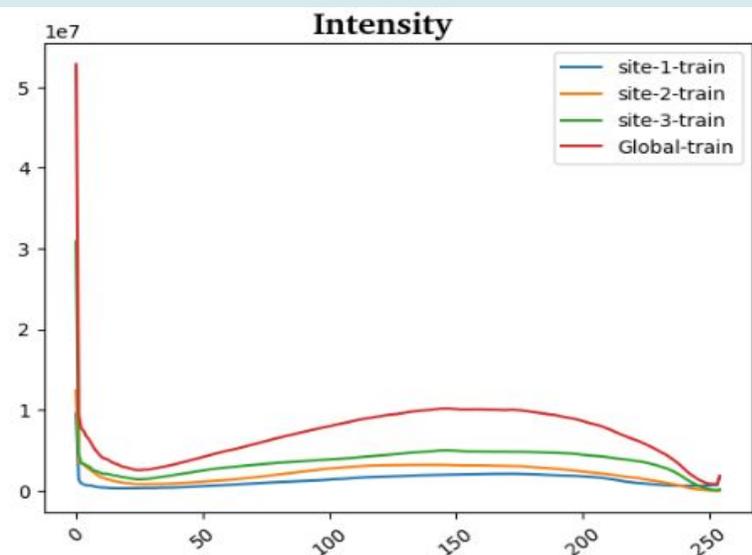
- **FL simulator:** Lightweight simulator for testing and debugging.
 - `nvflare simulator -w WORKSPACE [-n N_CLIENTS] job_folder`
- **Proof Of Concept (POC) mode:** For trying out FLARE features on a single machine.
 - `nvflare poc -h`
- **Production mode:** Safe mode and it start with provisioning

Federated statistics

- **Federated statistics**: Built-in operators for generating global statistics based on local client-side statistics.
- Common statistics include: count, sum, mean, std, histogram

	count	histogram
site-2-train	6012	[[0.0, 1.0, 12430030], [1.0, 2.0, 3511491], [2...
site-4-train	1345	NaN
site-3-train	10192	[[0.0, 1.0, 30867349], [1.0, 2.0, 4553187], [2...
site-1-train	3616	[[0.0, 1.0, 9512374], [1.0, 2.0, 1381654], [2....
Global-train	21165	[[0.0, 1.0, 52809753], [1.0, 2.0, 9446332], [2...

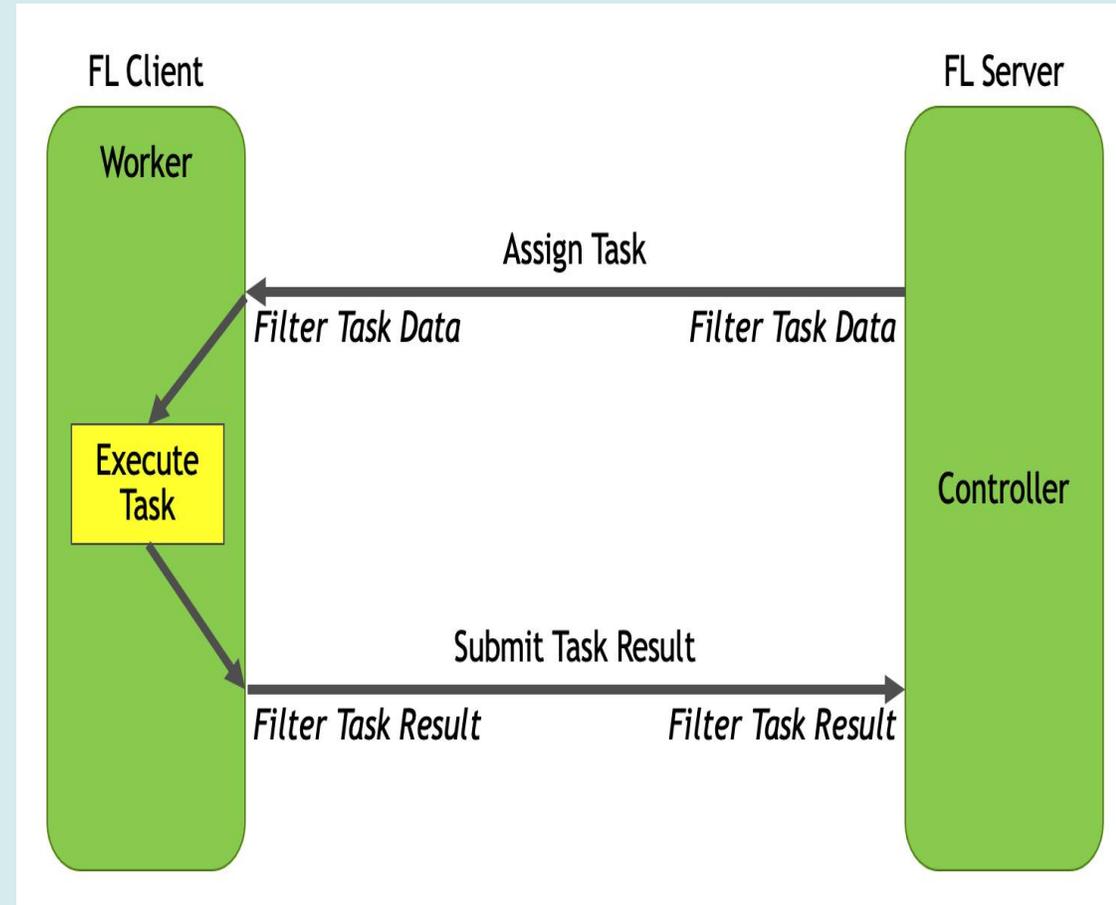
(a) Federated statistics. Note the data of “site-4” violates the client’s privacy policy and therefore does not share its statistics with the server.



(b) Histogram visualization.

Privacy Preservation

- NFlare addresses privacy concerns using data and result **filtering** mechanisms.
 - Methods:
 - Differential privacy
 - Homomorphic encryption
- local privacy policy: Each organizer can define a custom filter.



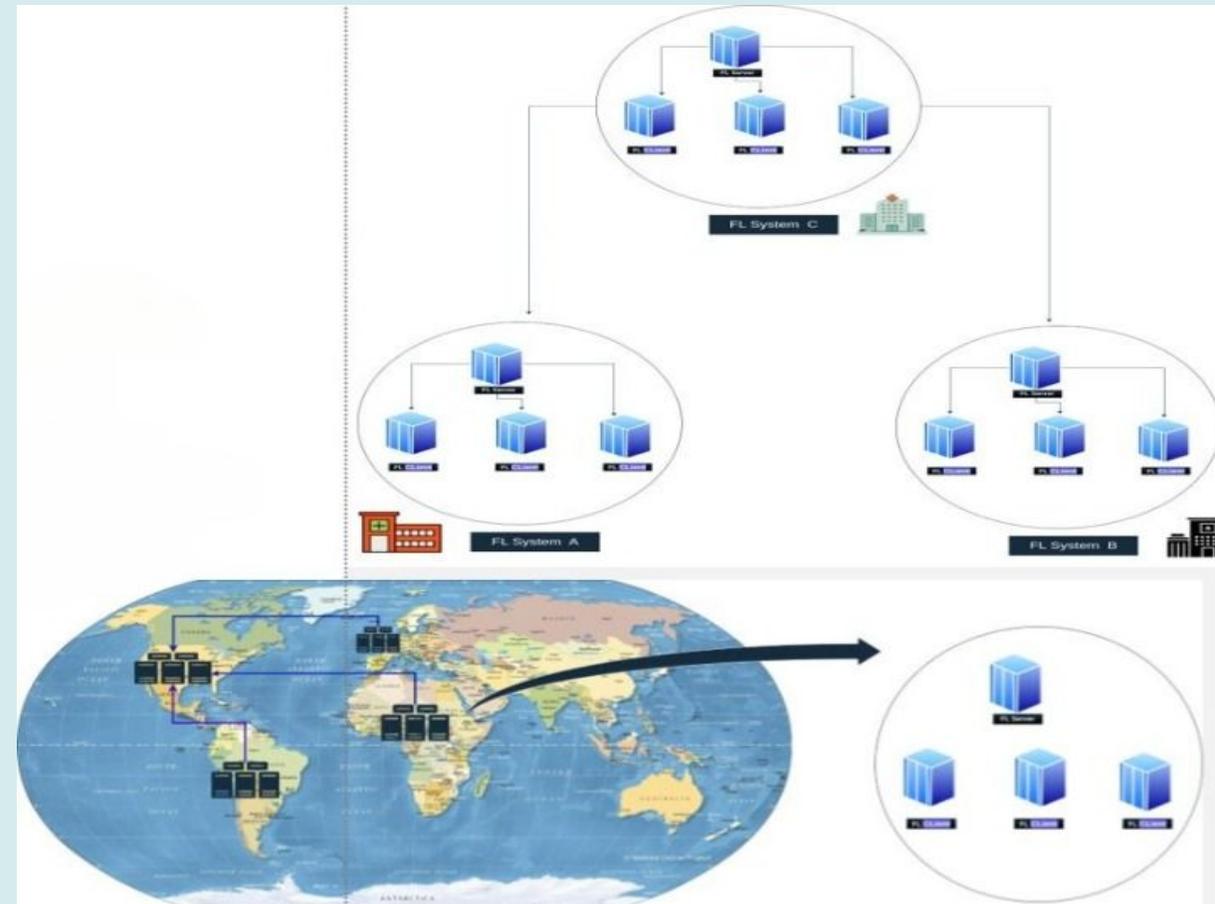
New Features added to NVFlare 2.4

- **ML2FL transition:** Turn existing ML/DL with few lines of code changes.
- **Experiment tracking:** support MLflow, weights & Biases, Monai auto-logging integration.
- **DISTRIBUTED DATA PARALLEL (DDP)** training for pytorch.
- **Workflow & algorithms:** [Swarm Learning](#), [GNN](#), [Vertical fed XGBoost](#).

New Features added to NVFlare 2.4

- **FL HUB (HIERARCHICAL UNIFICATION BRIDGE):**

- Allow two or more FL systems to connect together
- motivations:
 - overcome client limitations,
 - ensure local management of cross-region FL systems.



New Features added to NVFlare 2.4

- **Security Enhancements:**
 - Event-Based Security Plugin
 - enable site-specific authentication/authorization methods (such as each side has its own authentication mechanism)
 - Unsafe Component detection
 - enable the approval process to remove unsafe job
- **LLM support:** [LLM examples](#)

Conclusions

- NFlare advantages for federated learning:
 - Step-by-step examples
 - Support for vertical and horizontal FL
 - Rich Set of Programmable APIs
 - Authentication with Provisioning Tools
 - Local Authorization Policy
 - Local Privacy Policy
 - Federated Statistics Tools
 - High Availability Mode
 - Multi-Tool Monitoring

AI4

eOSC



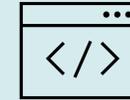
Co-funded by
the European Union



AI4EOSC



khadijeh.alibabaei@kit.edu
valentin.kozlov@kit.edu
lisana.berberi@kit.edu
borja.sanchis@kit.edu



ai4eosc.eu

Thank you for your attention