Private ML Track

UNECE Input Privacy Preserving Project

Introduction

- **Pilot's goal:**
  - Build a simulated environment to validate the concept of multi party privacy preserving Machine Learning (PPML) for both training and inference.

- **Project's scope:**
  - Investigate best practice and open source tools for distributed and collaborative ML training among multiple organisations in a low trust environment whilst mutually benefitting from the outcomes (the final model) or allowing safe 3rd party access.

- **Environment:**
  - Simulated multi organisational set-up with several NSOs gathering data from individuals (sensors) to predict their activities (time use and well-being surveys).

Introduction

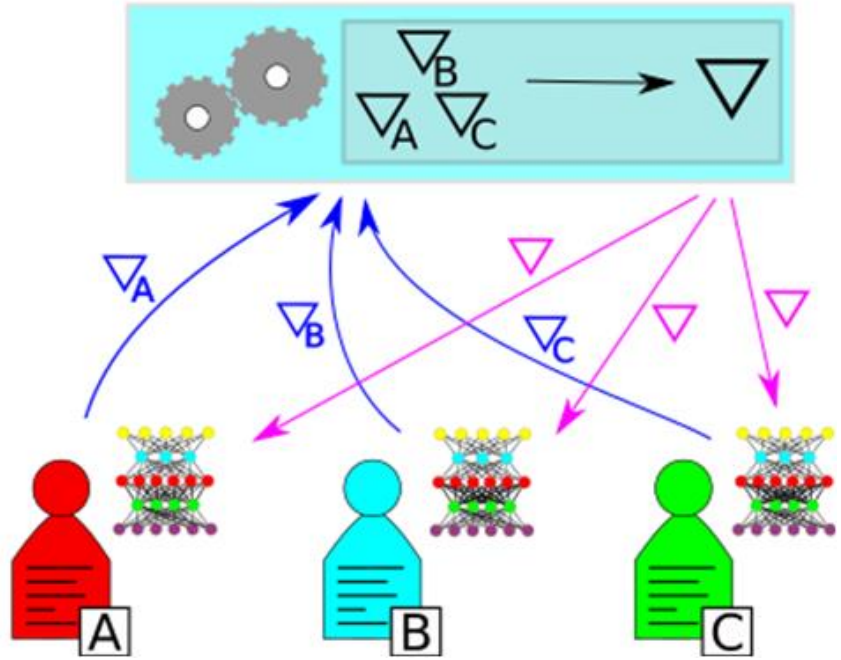
- **Architecture:**
  - Distributed containerized PPML architecture utilising Federated Learning to train a NN model and enable inference whilst protecting data security, privacy and confidentiality.

- **Data:**
  - Moderately sensitive - collected by wearable/smart devices using accelerometers, e.g. smart/sports watches. Open data used in the pilot.

- **Method:**
  - ML toolset - a typical ML classification task (i.e. to predict human activities starting from accelerometer data)

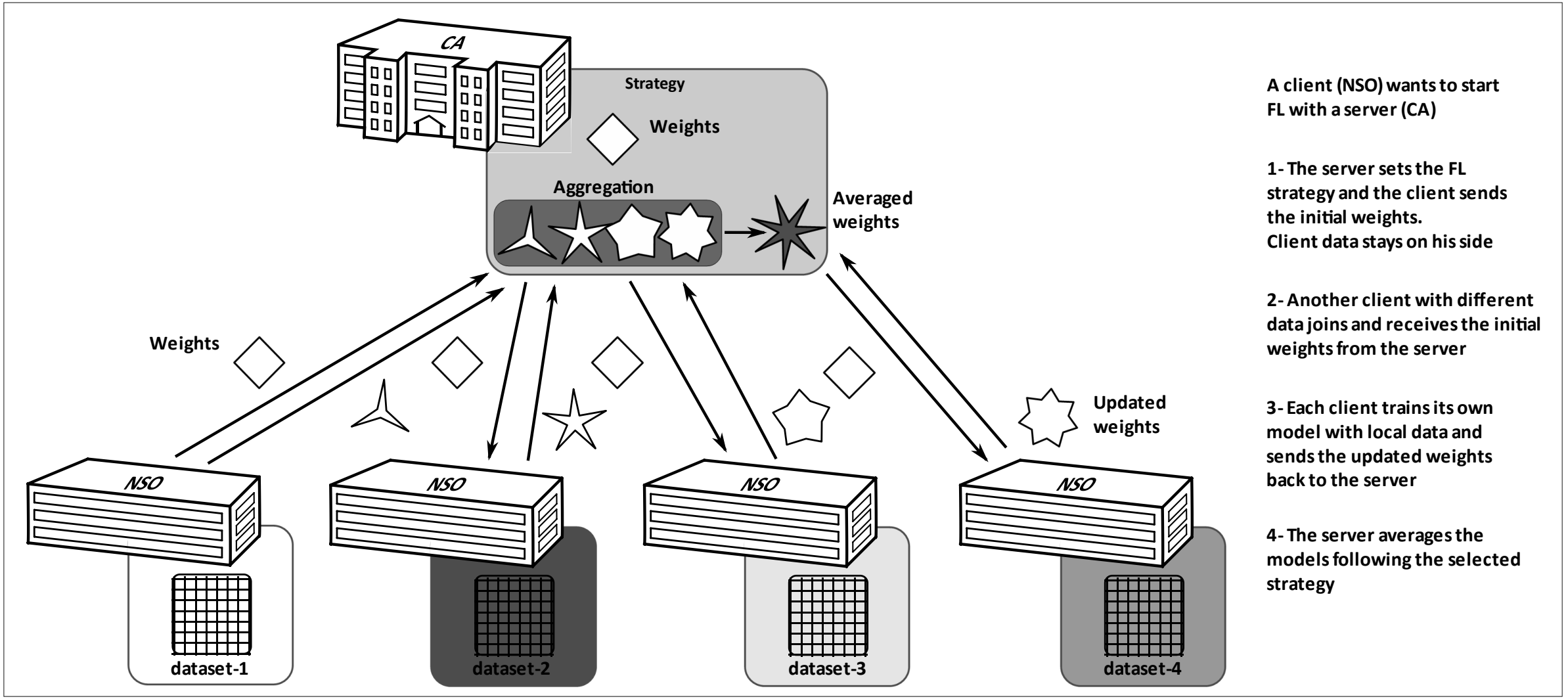
Federated Learning

- In FL, each party (e.g. NSO) holds a neural network that would like to train.
- After each round of the training, the parties send their weights (or parameters) to a central authority.
- Central authority aggregates the weights and send instructions to parties to update their local models.
- This process is repeated several times. Note that only the accumulated weights are shared among parties.
- The final model can be used locally by parties for inference on new data.
- FL protects the privacy of the input data by ensuring that the data never leaves the clients' devices.



https://www.statcan.gc.ca/eng/data-science/network/privacy-preserving

Simulated Environment (Scenario 1)



A client (NSO) wants to start FL with a server (CA)

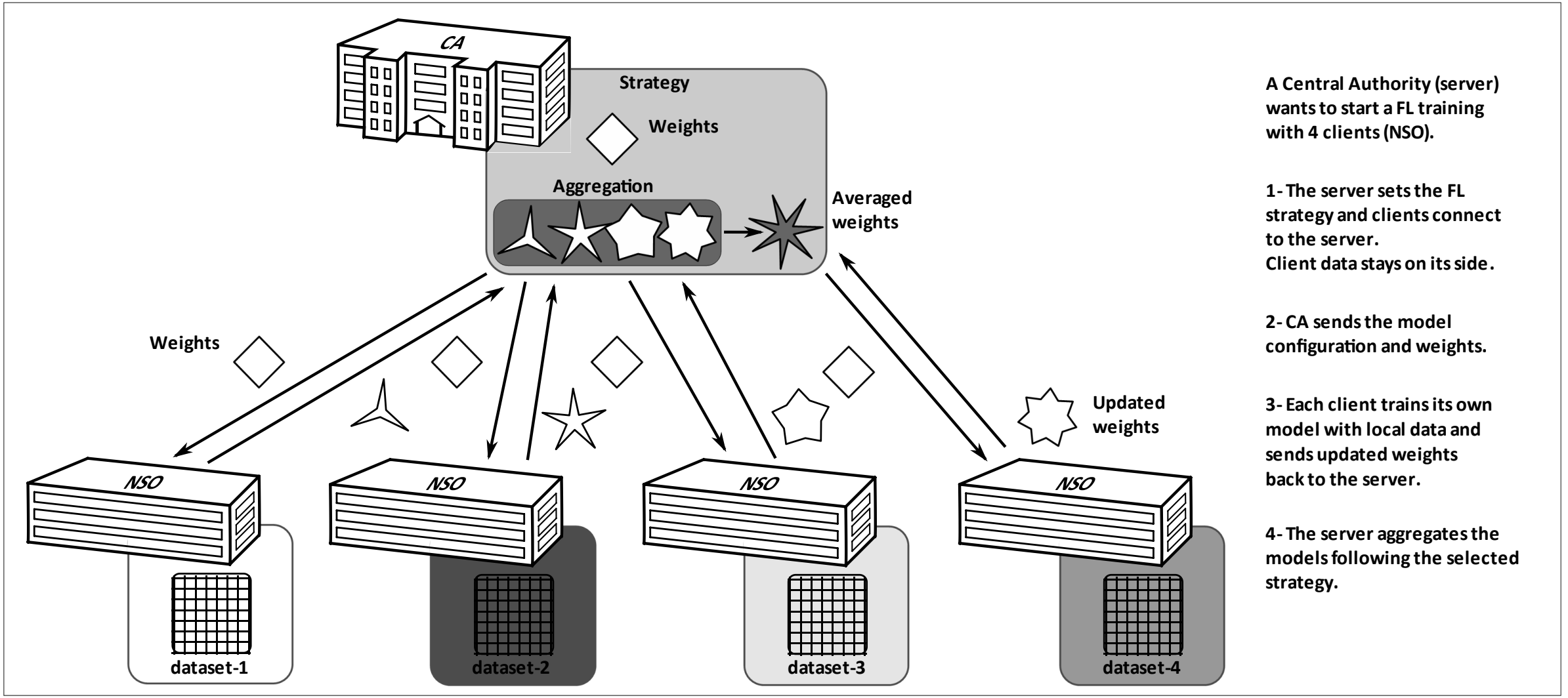
1- The server sets the FL strategy and the client sends the initial weights.
Client data stays on his side

2- Another client with different data joins and receives the initial weights from the server

3- Each client trains its own model with local data and sends the updated weights back to the server

4- The server averages the models following the selected strategy

Simulated Environment (Scenario 2)



A Central Authority (server) wants to start a FL training with 4 clients (NSO).

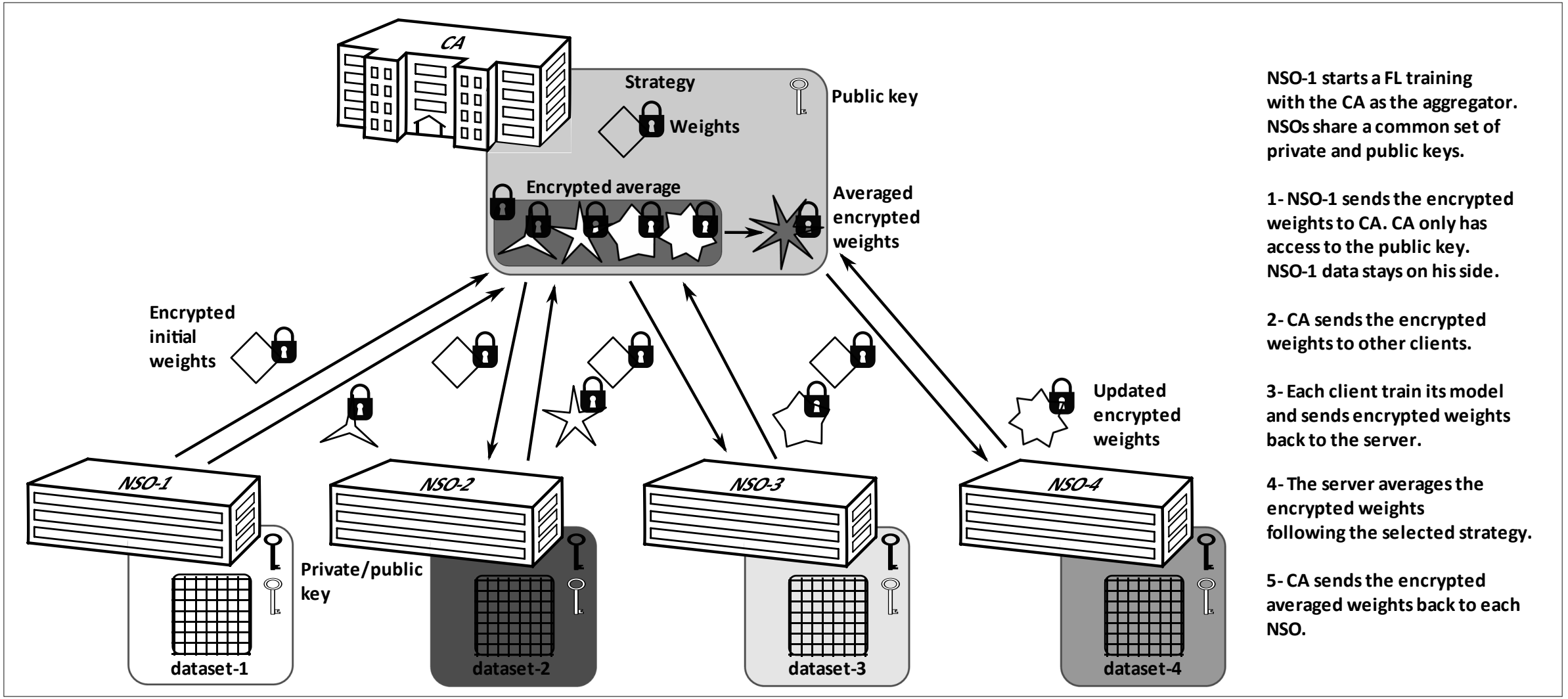
1- The server sets the FL strategy and clients connect to the server.
Client data stays on its side.

2- CA sends the model configuration and weights.

3- Each client trains its own model with local data and sends updated weights back to the server.

4- The server aggregates the models following the selected strategy.

Simulated Environment (Scenario 3)



NSO-1 starts a FL training with the CA as the aggregator. NSOs share a common set of private and public keys.

1- NSO-1 sends the encrypted weights to CA. CA only has access to the public key. NSO-1 data stays on his side.

2- CA sends the encrypted weights to other clients.

3- Each client train its model and sends encrypted weights back to the server.

4- The server averages the encrypted weights following the selected strategy.

5- CA sends the encrypted averaged weights back to each NSO.

Simulated Environment (Data & Model)

- Human activity recognition using smart devices' accelerometer and gyroscope data*, after pre-processing.
- The goal is to classify the data into 6 classes: WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING.
- The data was split into four subsets, one for each NSO (i.e. STATCAN, ONS, ISTAT and CBS), in the experiments.
- A neural network (Multi-Layer Perceptron with linear layers and ReLU activations) is used for the purpose of classification.

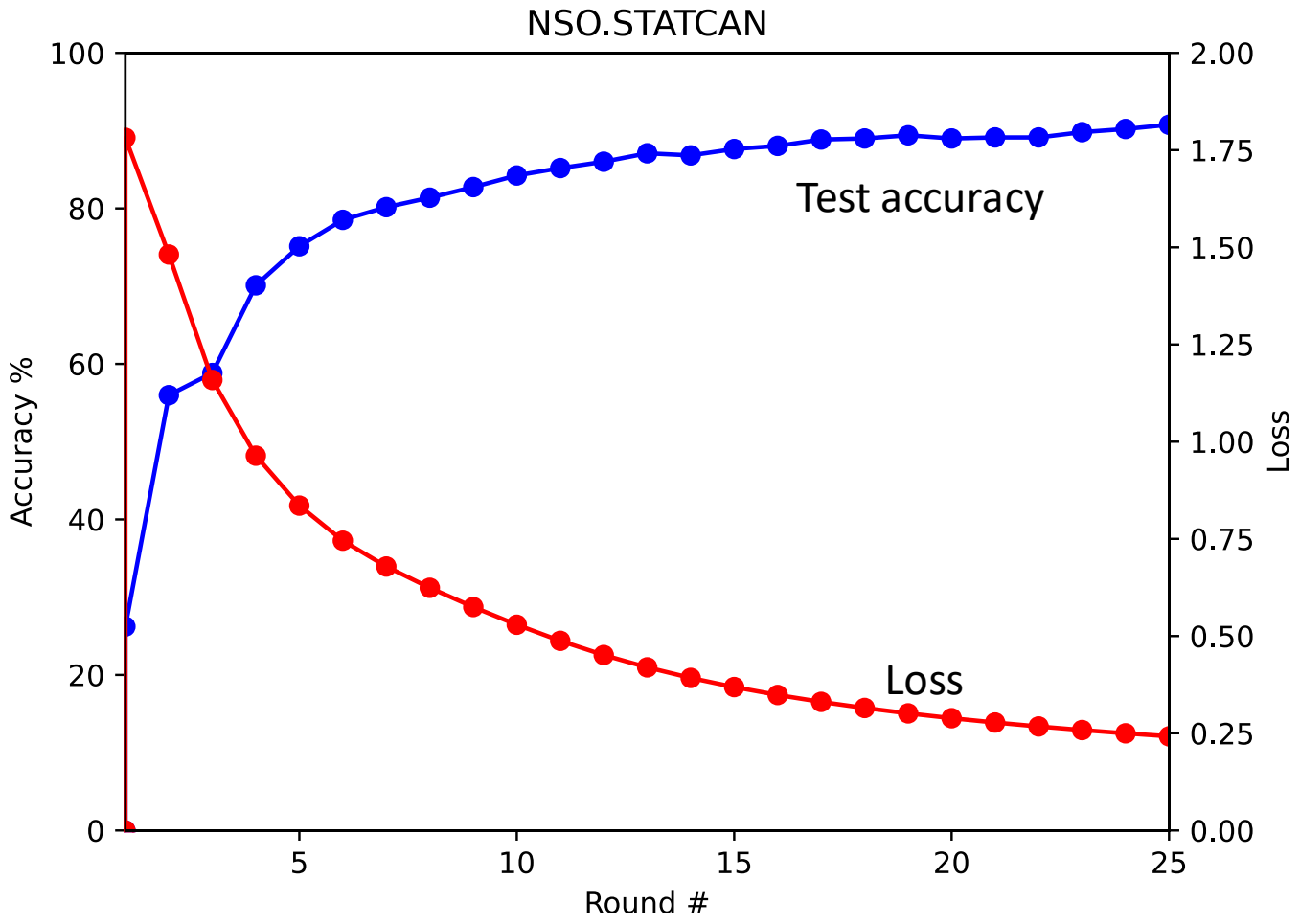
* D. Anguita, A. Ghio, L. Oneto, X. Parra and J. L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones. 21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.

Simulated Environment (Architecture)

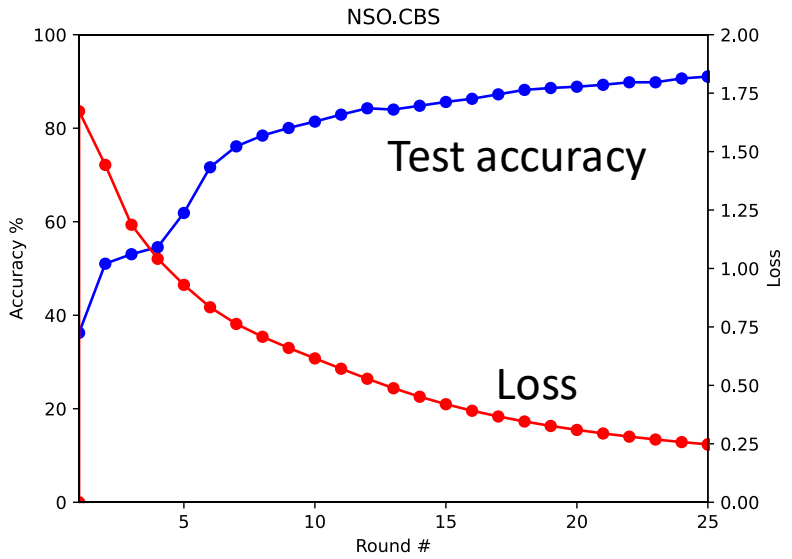
- A unified Federated Learning library called Flower* is used to simulate the environment.
- Only updated weights are transferred between the central authority and other NSOs during the training.
- Transferred weights are aggregated on server side after each round. The averaged weights are sent to clients (FedAvg). In the encrypted version, the average is computed on encrypted weights using the Paillier cryptosystem.
- This approach is very customizable: # training rounds, epochs, and network architecture can be changed.
- It is possible to use encryption at rest and in transit, using certificates with latest flower development for secure communication.

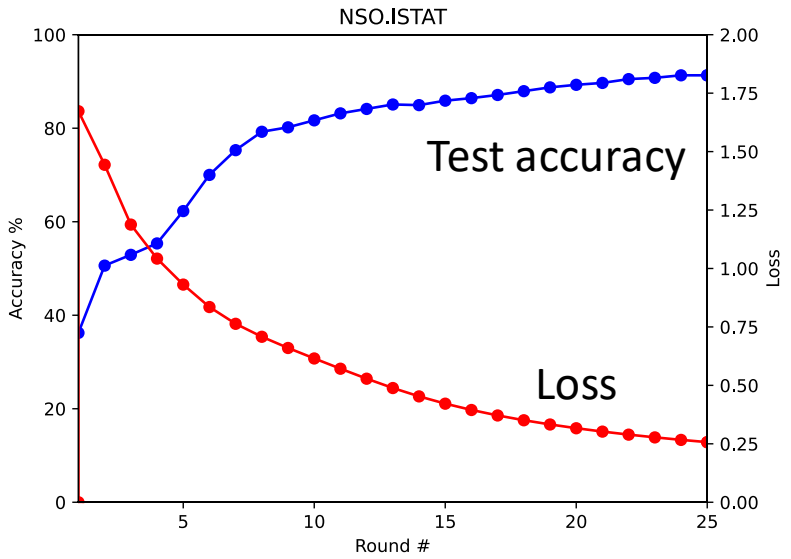
* https://flower.dev

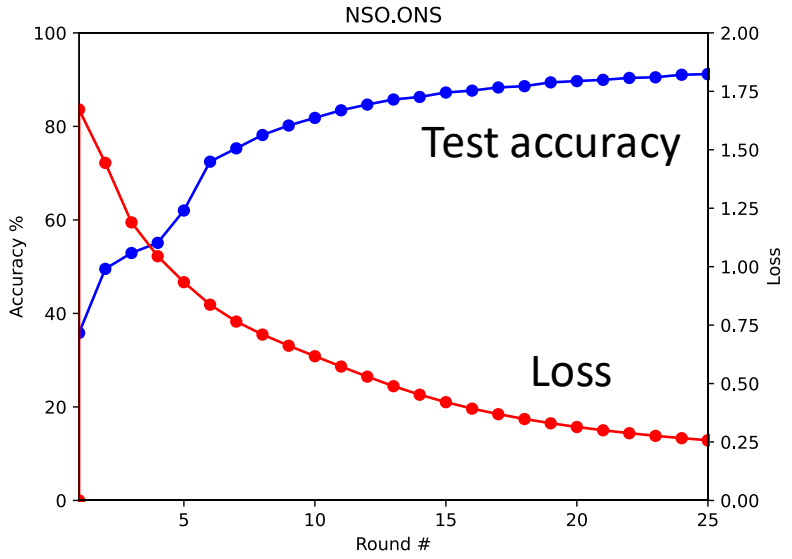
Federated averaged weights strategy results

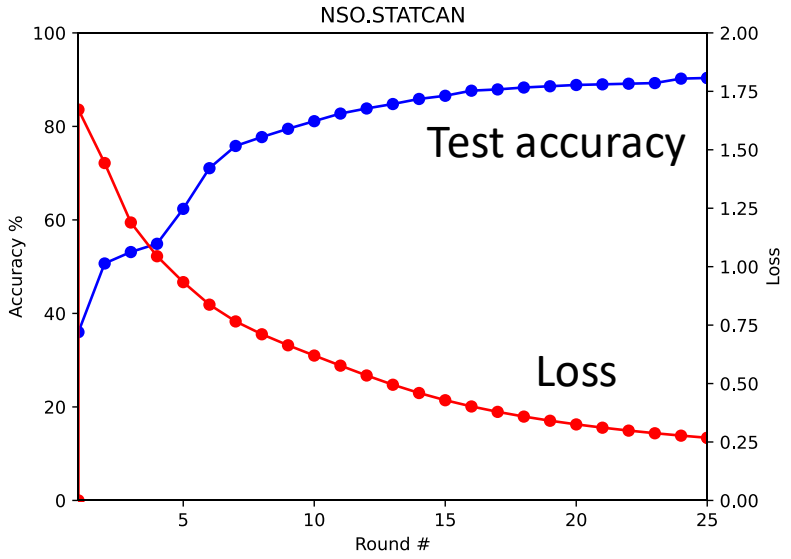


Encrypted Federated Learning









Conclusions and Results

- This experiment had a simplified scope and was performed in simulation environment.
- We have built a community of NSOs in the area of privacy enhancing technologies with link to open source community, industry and academia.
- There is a direct link to sustainability, when it comes to collaboration among NSOs, namely new ways of collaboration, driven by privacy requirements and technological constraints.

Challenges and Lessons Learned

- Open source software stack support for this particular scenarios.
- In reality, inconsistent data formats across multiple NSOs.
- Unbalanced and outlier data points and lack of sufficient and good-quality data. Different aggregation strategy can be tested and used to mitigate this.
- Pre-processing steps to take into account different international labelling and standards in distributed ML for deployment.

Next Steps

- Extend the scope to more complex models and other distributed data related to members of HLG-MOS, e.g. social media, border stats ...
- A systematic review of the open source tools and their maturity.
- Incorporate Secure Multi-party Computation for secure aggregation of weights during training, as well as inference.
- Integrate Differential Privacy as part of the protocol to protect output privacy.
- Collaborate with the OpenMined community to use their software stack, with requirements.
- Onboard the project to the UN PET-Lab infrastructure.