

The Future of AI Training is Federated A Beginners Guide with Flower

Chong Shen Ng
PyCon DE & PyData · April 24th, 2025



Checkout:

github.com/chongshenng/pyconde2025

Create account:

flower.ai

Install Docker

Hello! Prep and Warm-up

- Setup
 - Checkout tutorial GitHub repo  github.com/chongshenng/pyconde2025
 - Create Flower account  flower.ai
 - Install Docker



Hello! Prep and Warm-up

- Setup
 - Checkout tutorial GitHub repo  github.com/chongshenng/pyconde2025
 - Create Flower account  flower.ai
 - Install Docker
- Chong Shen - Research Engineer @ Flower Labs
 - linkedin.com/in/chongshenng
 - github.com/adap/flower

Hello! Prep and Warm-up

- Setup
 - Checkout tutorial GitHub repo  github.com/chongshenng/pyconde2025
 - Create Flower account  flower.ai
 - Install Docker
- Chong Shen - Research Engineer @ Flower Labs
 - linkedin.com/in/chongshenng
 - github.com/adap/flower
- Merch!
 -  T-shirts
 -  Tote bags
 -  Socks
 -  Stickers!

Why Federated AI?

Data is naturally decentralized



AI can't use most of it



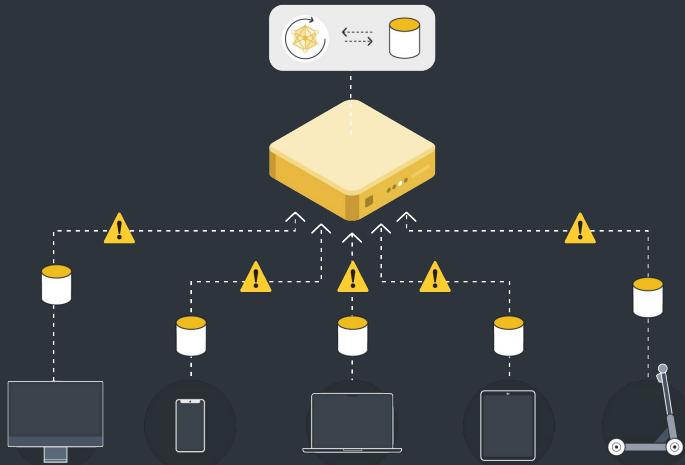
Collecting data doesn't solve this



Federated AI



Centralized AI can't access most data

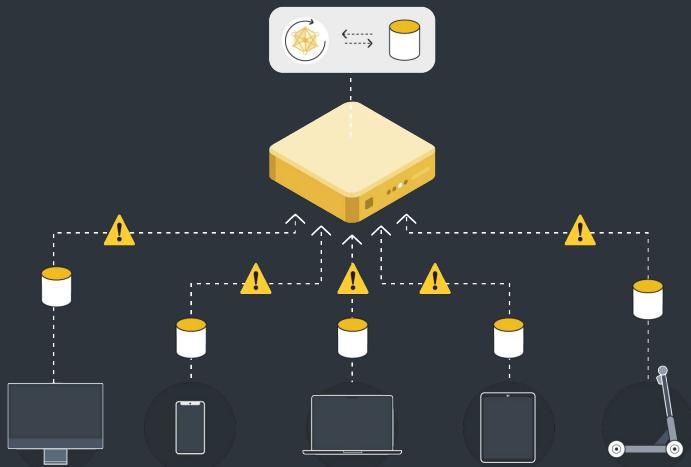


Move the data to the computation



Data needs to move

Federated AI can access more data



Move the data to the computation

⚠ Data needs to move

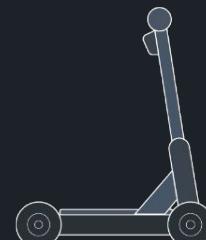
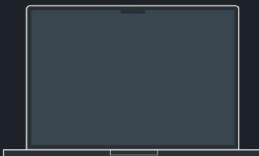
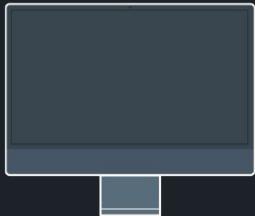


Move the computation to the data

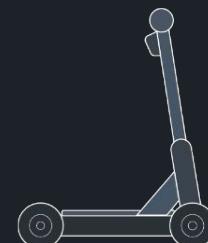
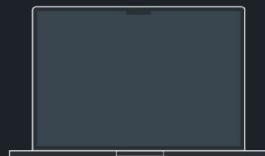
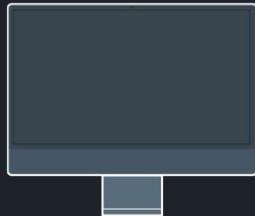
✓ Data stays where it originates

How does Federated Learning work?

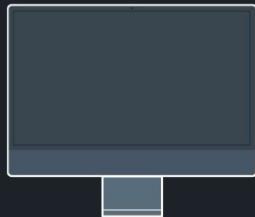
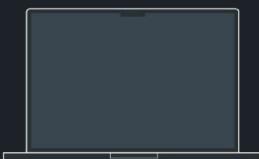
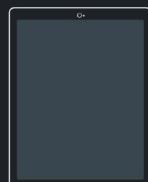
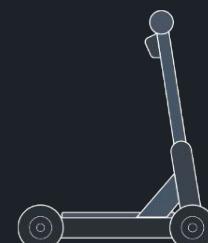
Step 0 · Initialize global model



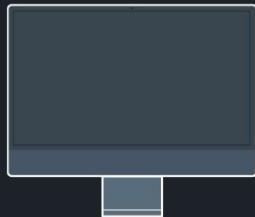
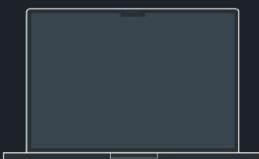
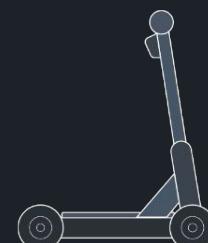
Step 0 · Initialize global model



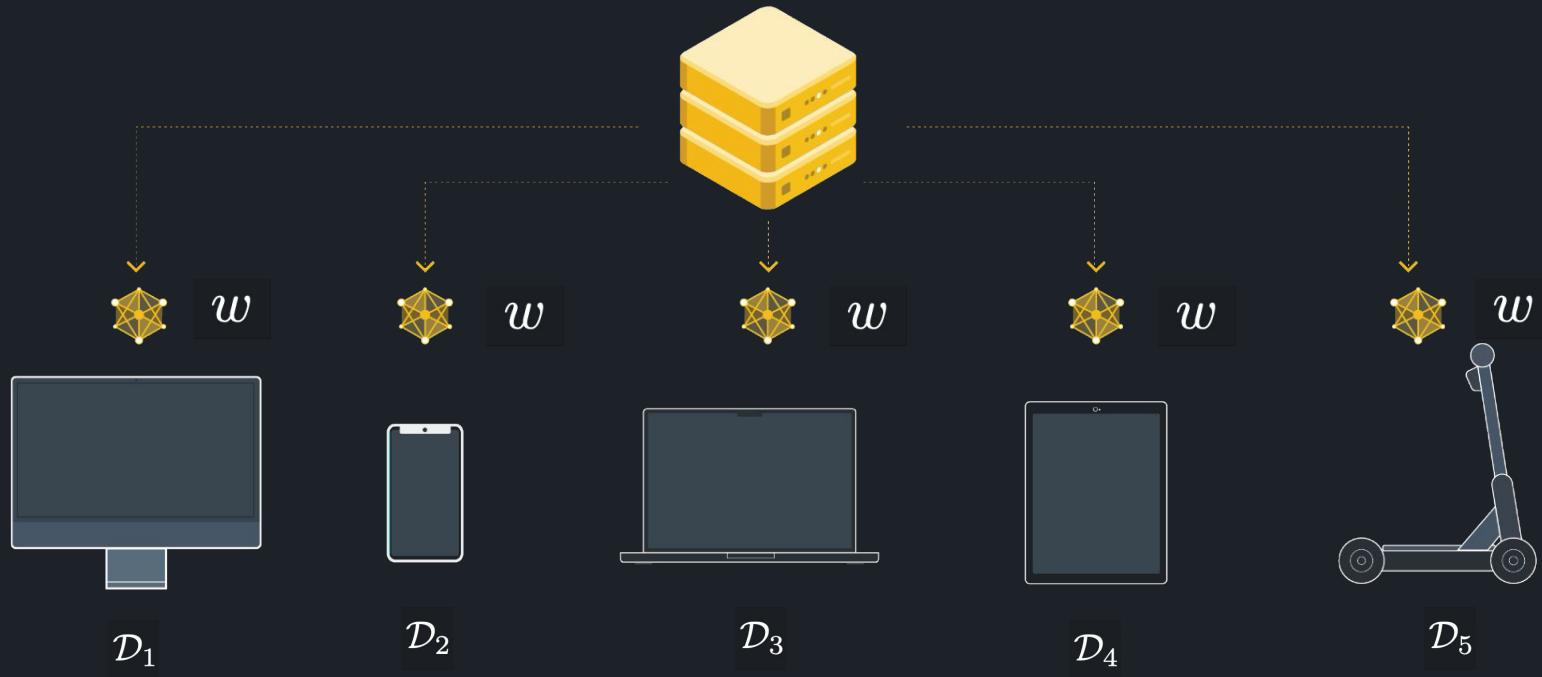
Step 0 · Initialize global model

 \mathcal{D}_1  \mathcal{D}_2  \mathcal{D}_3  \mathcal{D}_4  \mathcal{D}_5

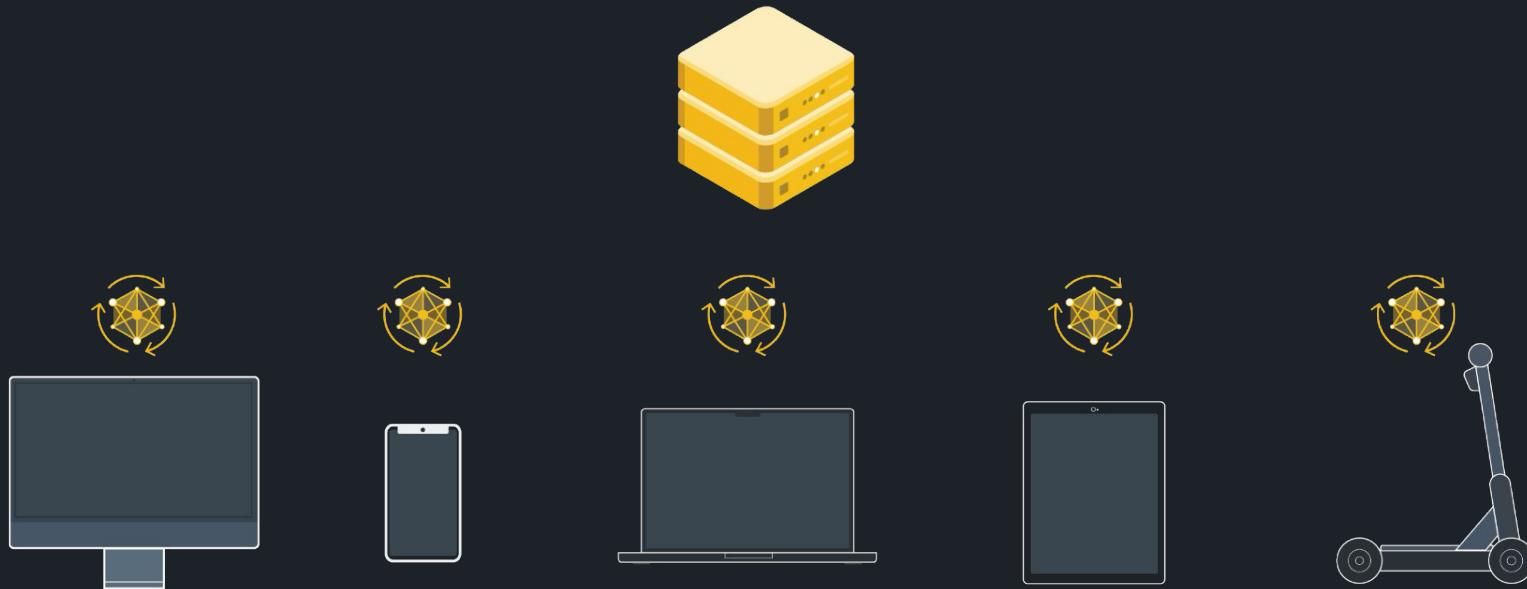
Step 0 · Initialize global model

 \mathcal{D}_1  \mathcal{D}_2  \mathcal{D}_3  \mathcal{D}_4  \mathcal{D}_5

Step 1 · Send model to connected orgs/devices



Step 2 • Train model on local data on each org/device



$$w_1 \leftarrow w - \eta \cdot \ell(w; \mathcal{D}_1)$$

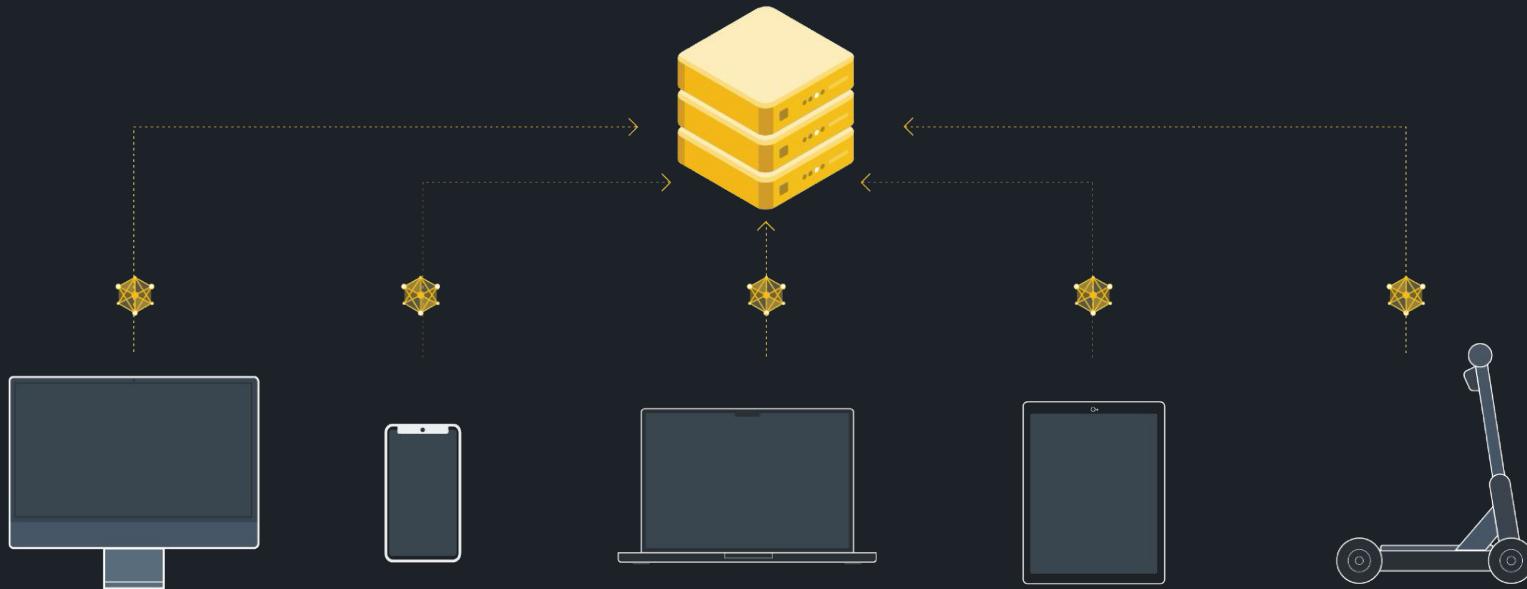
$$w_2 \leftarrow w - \eta \cdot \ell(w; \mathcal{D}_2)$$

$$w_3 \leftarrow w - \eta \cdot \ell(w; \mathcal{D}_3)$$

$$w_4 \leftarrow w - \eta \cdot \ell(w; \mathcal{D}_4)$$

$$w_5 \leftarrow w - \eta \cdot \ell(w; \mathcal{D}_5)$$

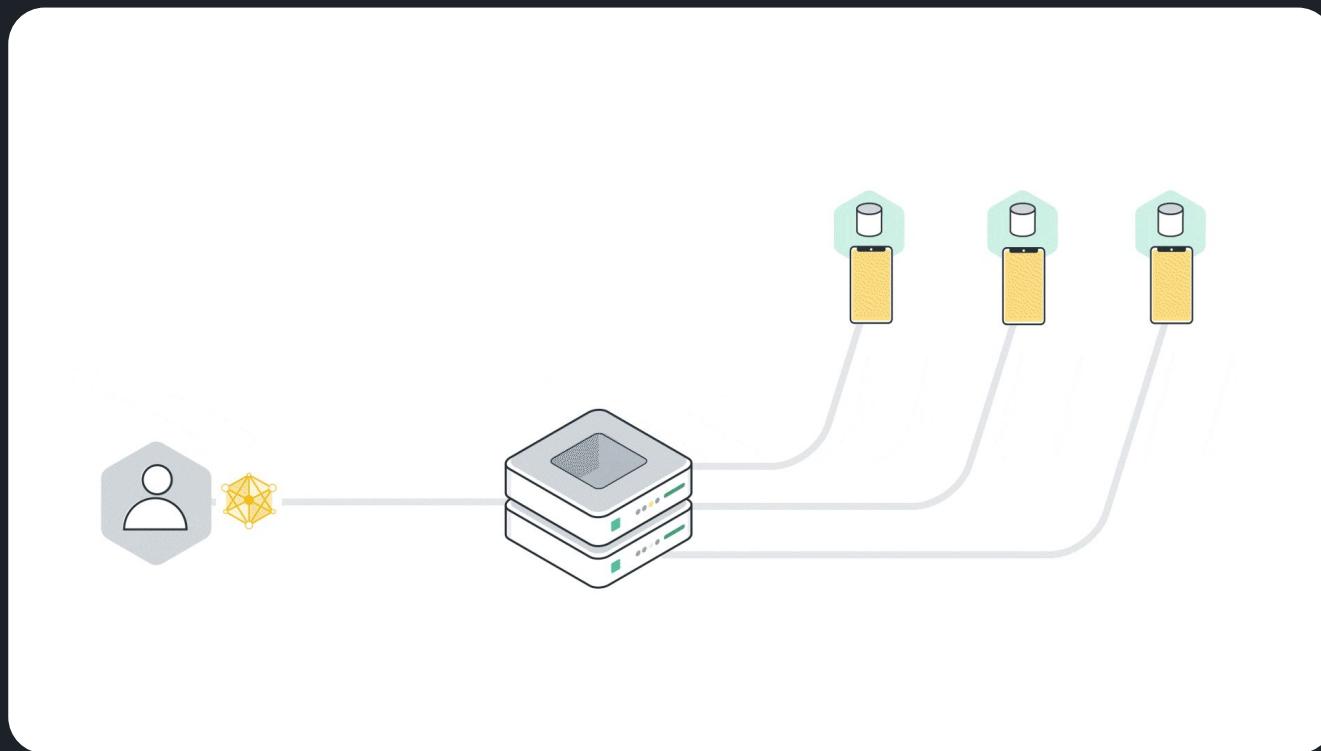
Step 3 • Send updated model back to the server



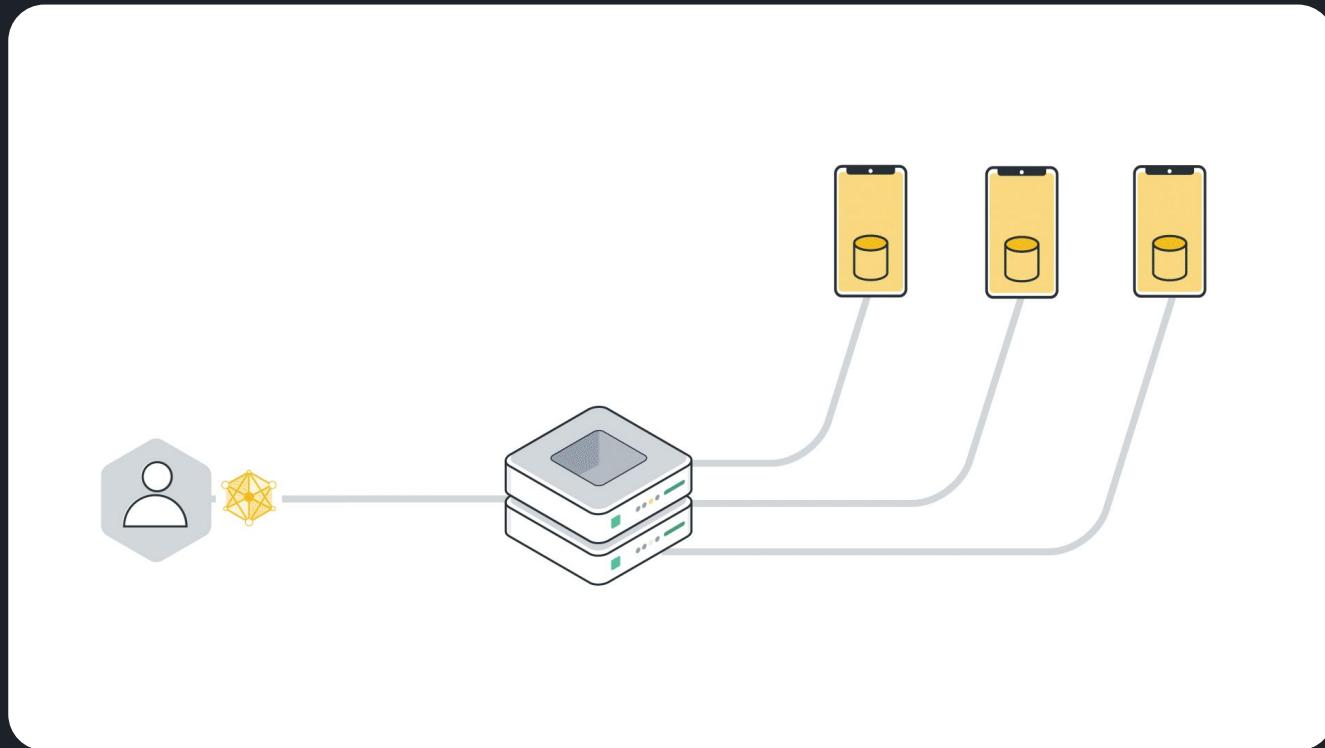
Step 4 · Aggregate updates into a new global model



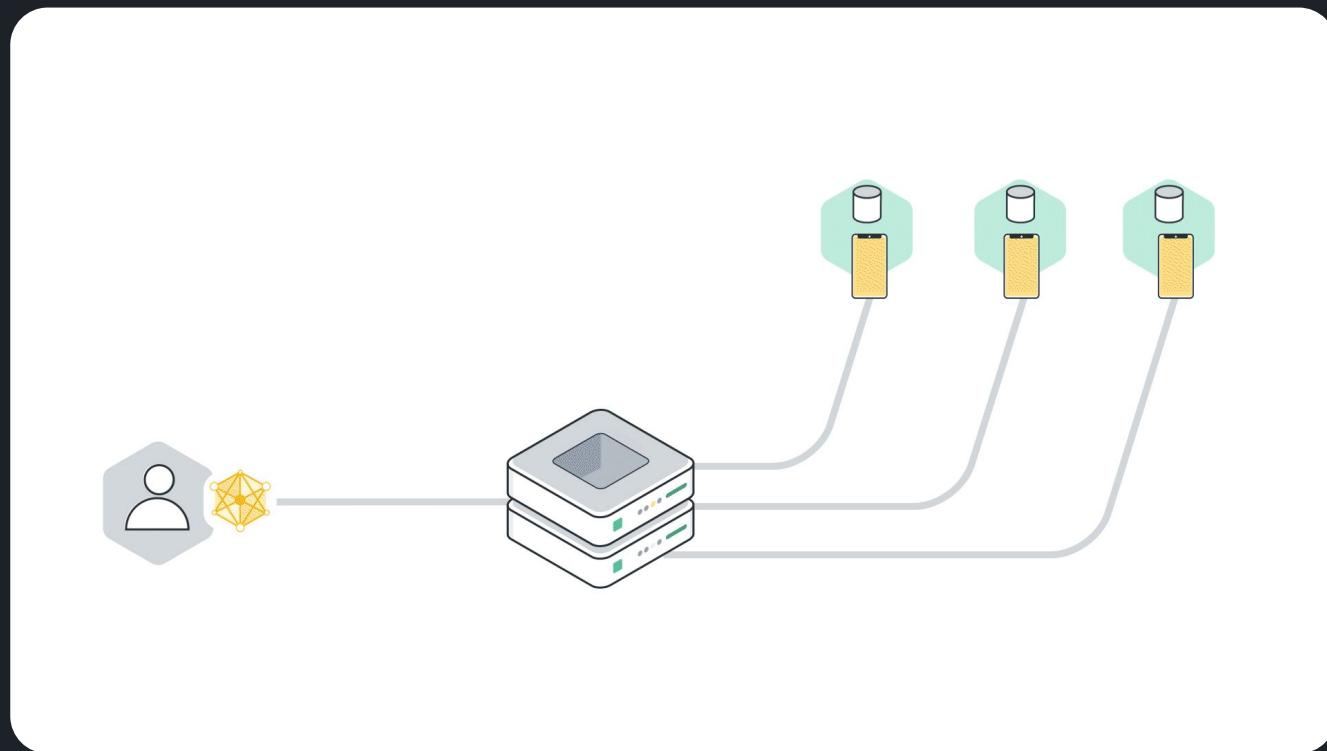
Repeat Steps 1 to 4



Federated Evaluation



Federated Analytics



Federate with Flower



Flower Platform

A unified approach to
Federated Learning,
Analytics, and Evaluation.



LLMs Neural nets

Linear regression XGB

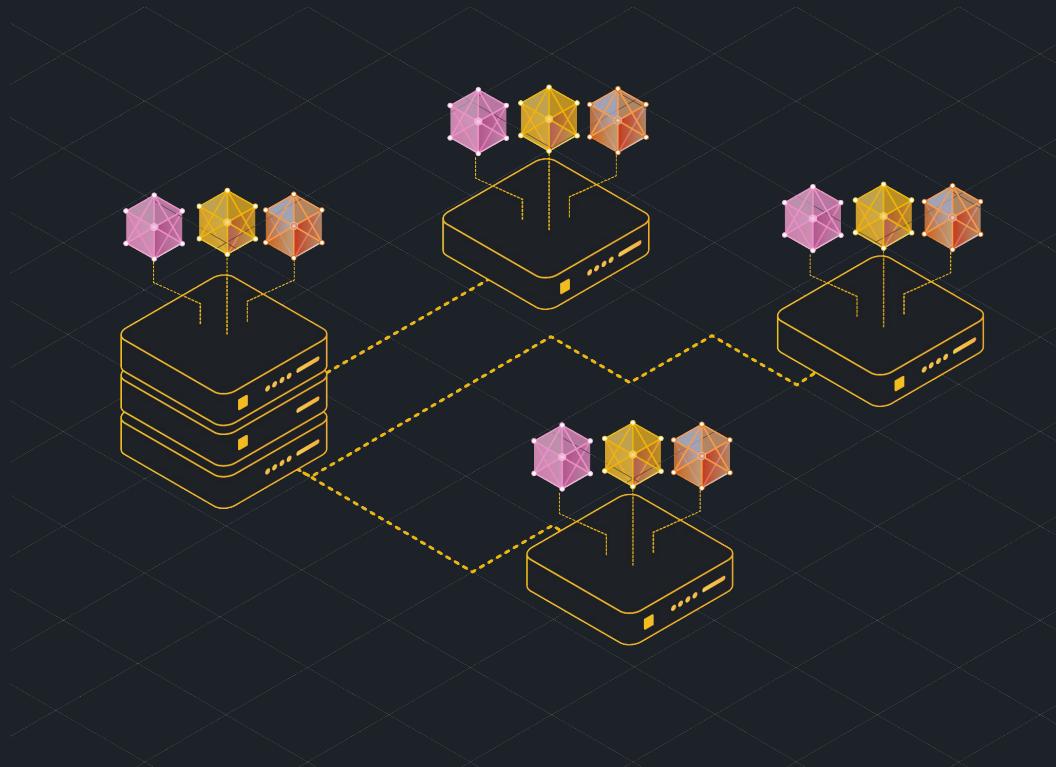
Diffusion models RL

RLHF self-supervision
pre-training fine-tuning

adapter-swapping

Yes.

Flower Deployment Runtime



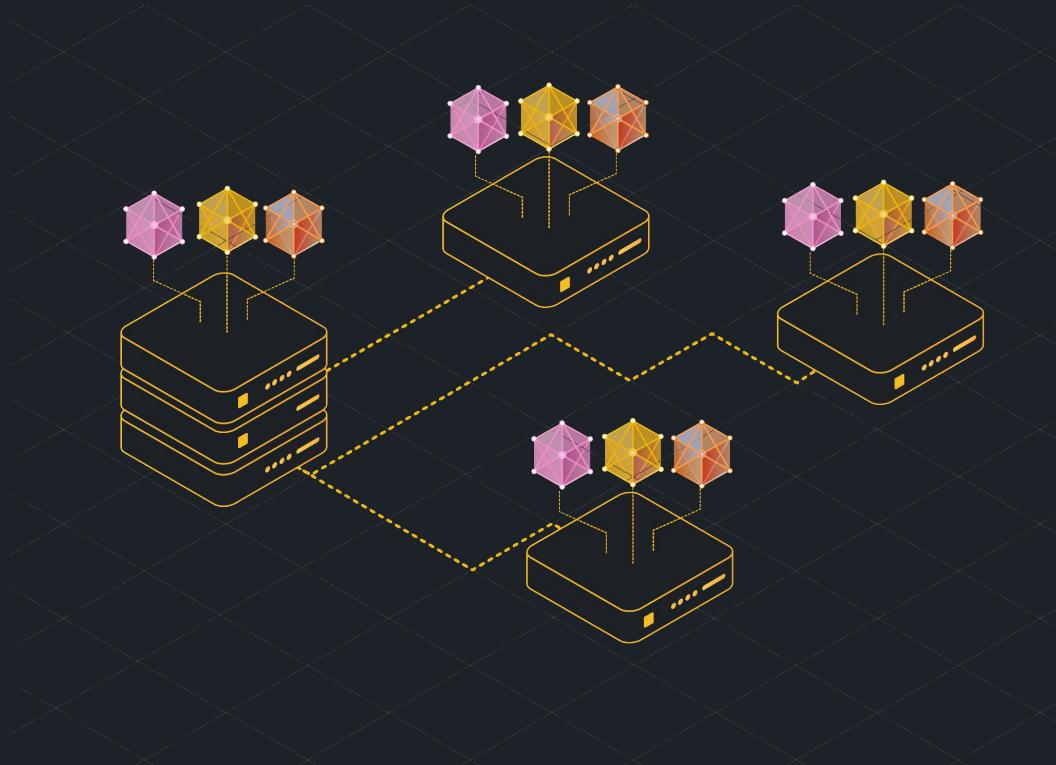
Multi-Run, flwr
CLI

User auth,
Node auth

Docker, K8s

Message API,
PETs

Flower PETs



Differential
Privacy

Secure
Aggregation

Homomorphic
Encryption

Custom SMPC,
Flower Mods

Adopted across different industries

Finance



Health

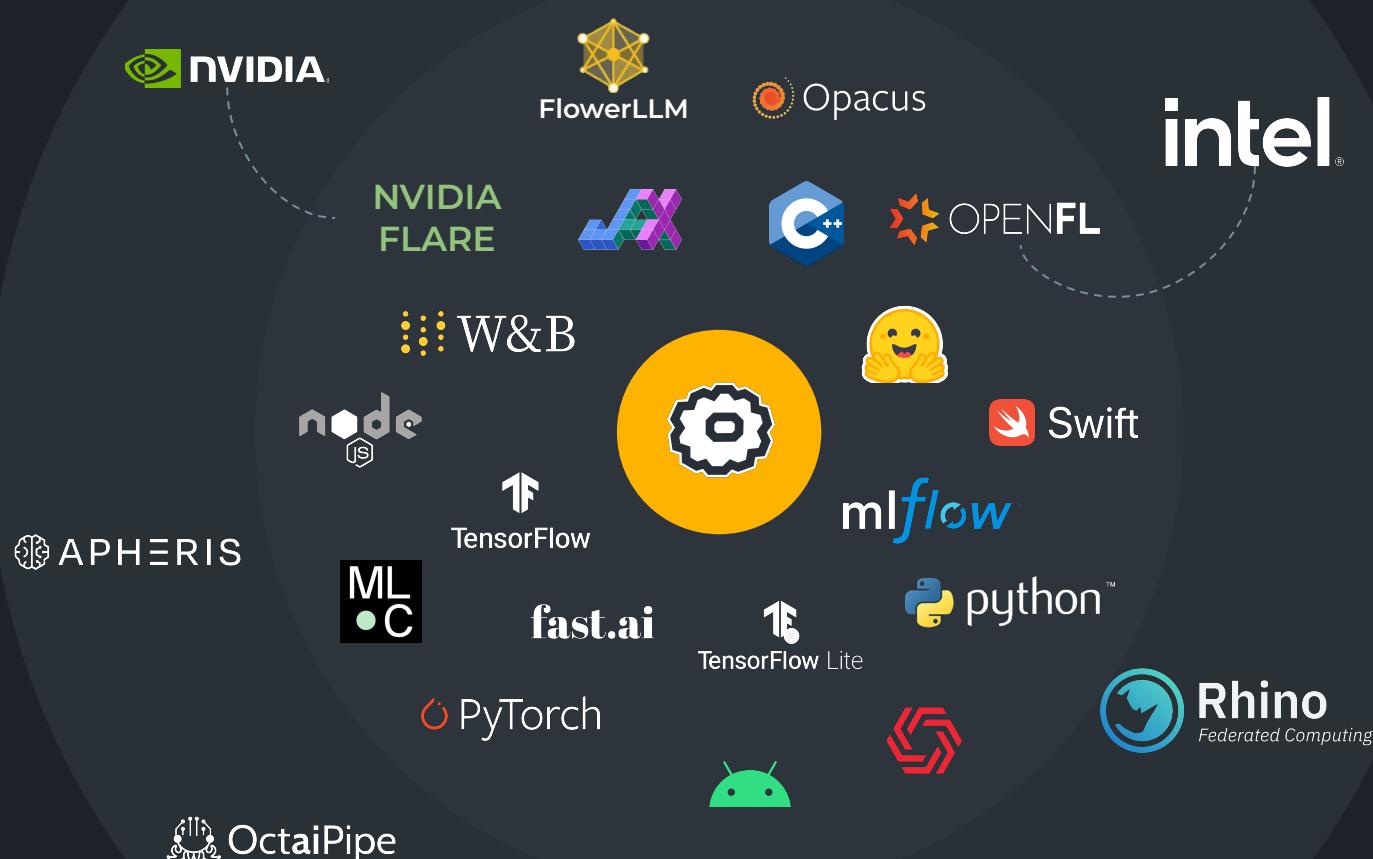


Consumer



Flower Ecosystem

Flower Ecosystem



Flower Runtimes



Flower Runtimes



Flower App

Flower
Simulation
Runtime

Flower Runtimes

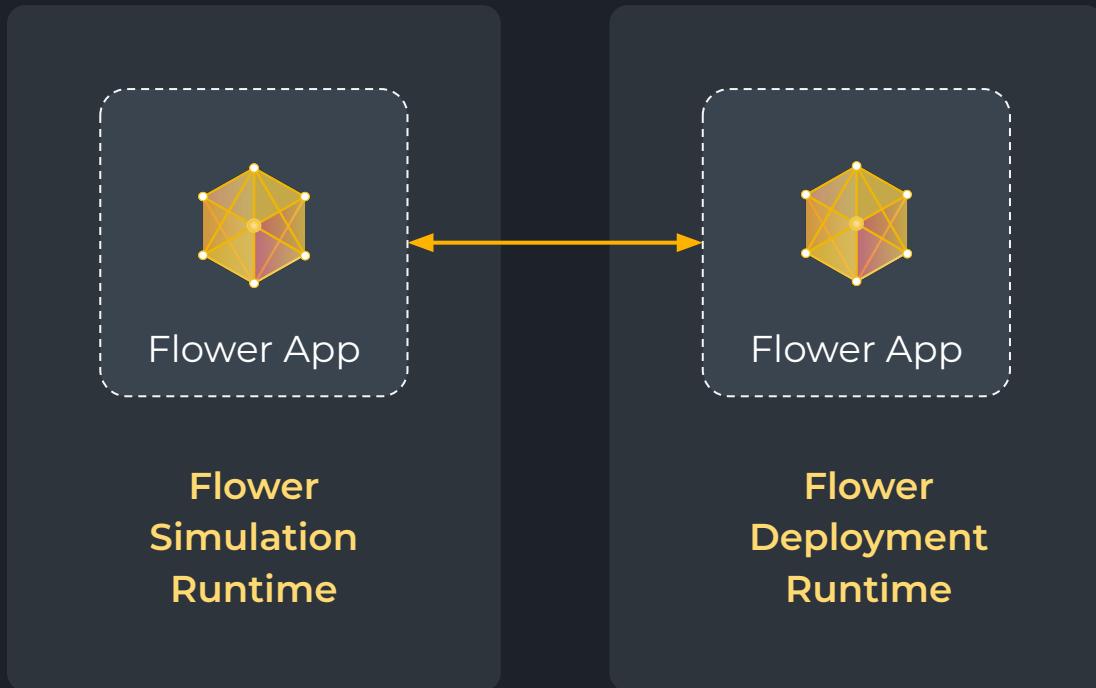


Flower App

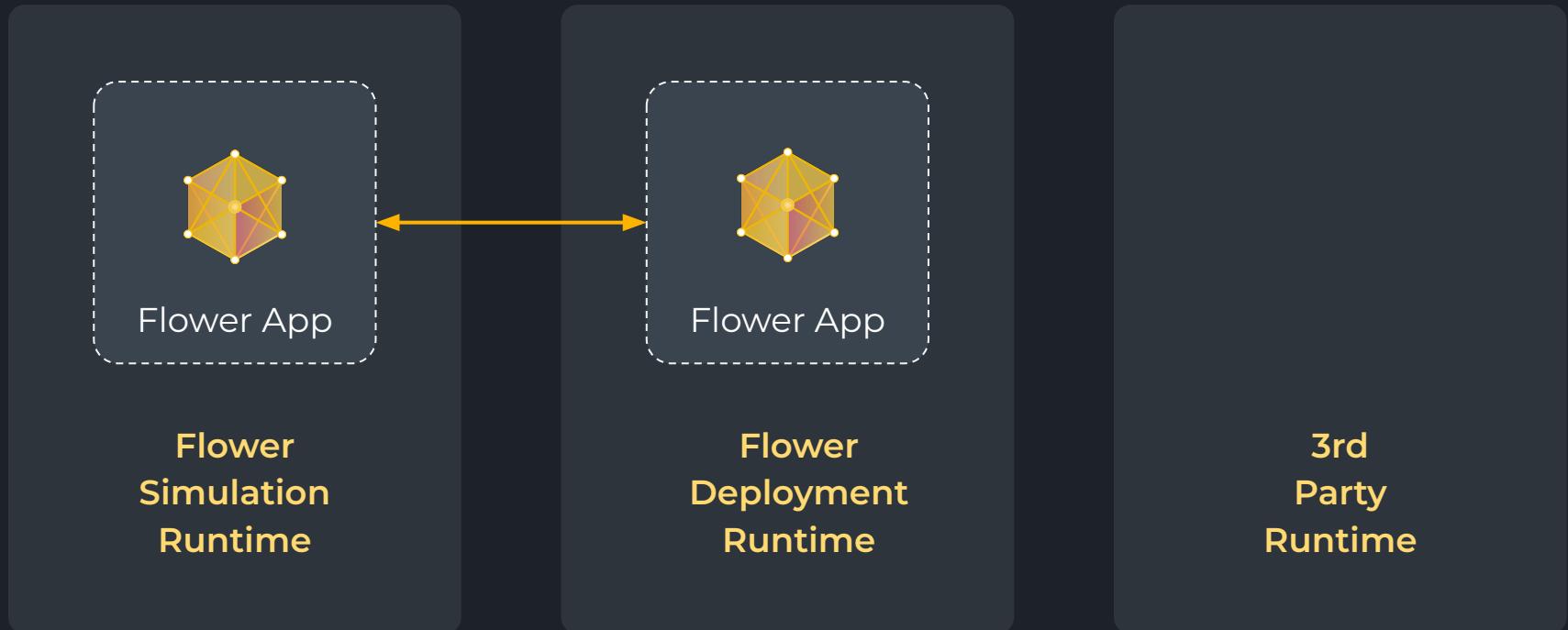
**Flower
Simulation
Runtime**

**Flower
Deployment
Runtime**

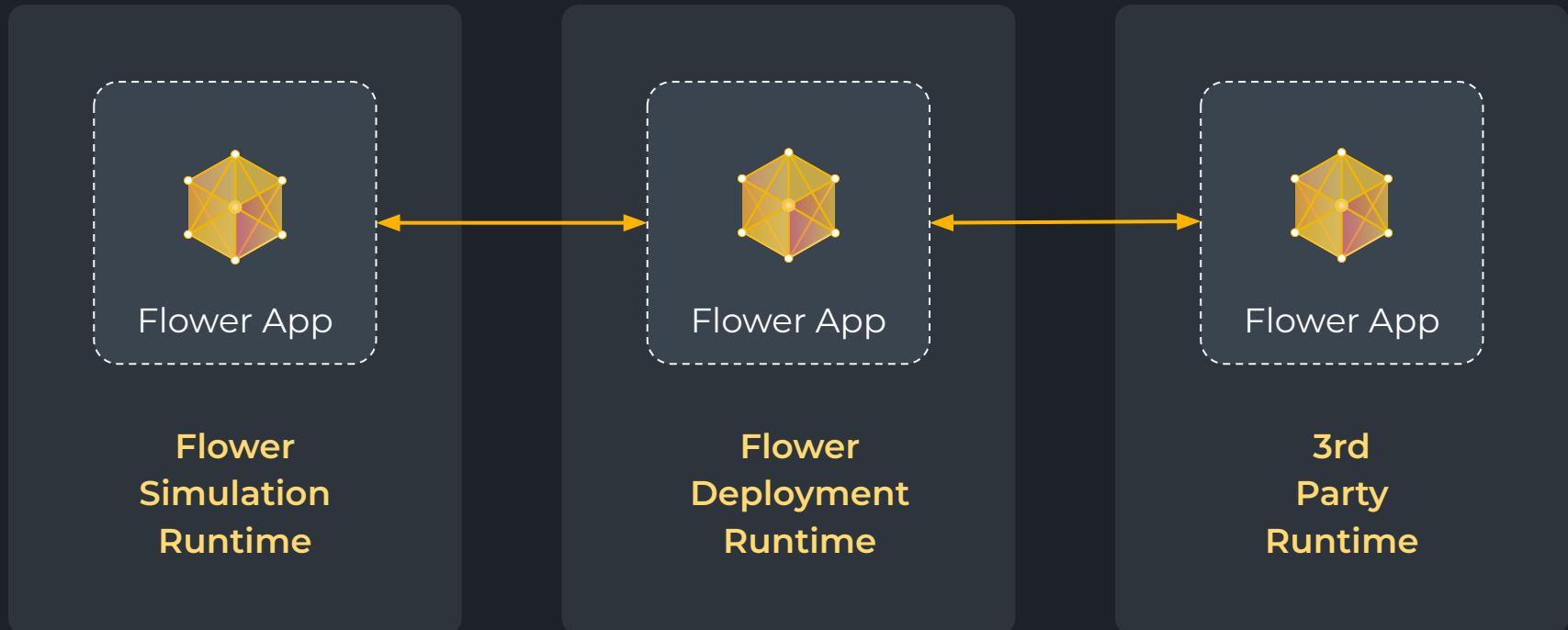
Flower Runtimes



Flower Runtimes



Flower Runtimes



Open Source Development with Flower

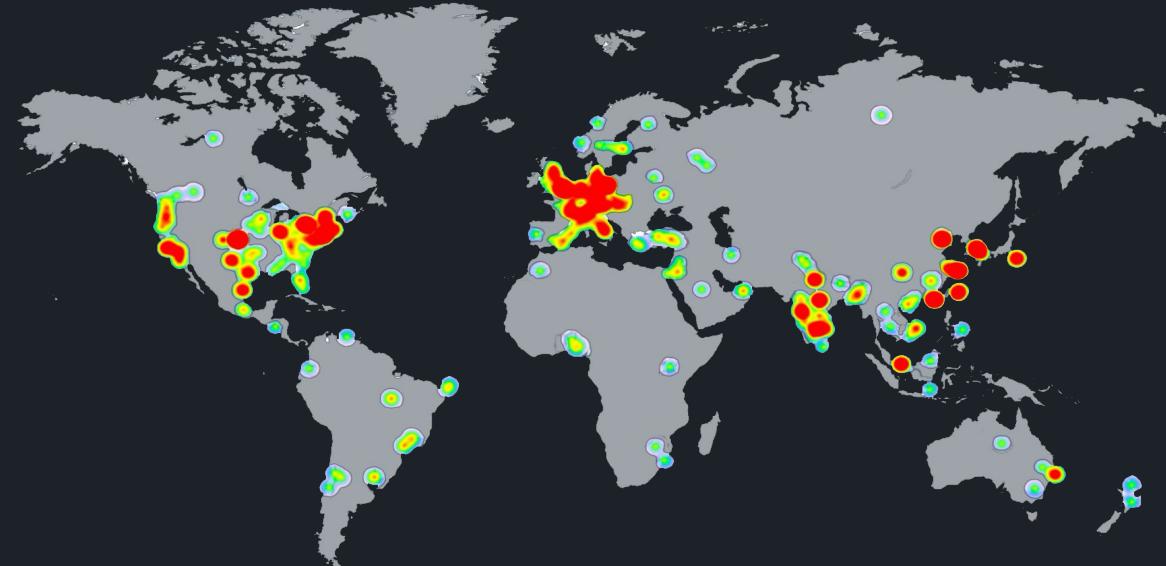
AI Devs ❤️ Flower

5,700+
Developers

5,700+
Stars

2,000+
Dependents

150+
Contributors



AI Devs ❤️ Flower

5,700+
Developers

5,700+
Stars

2,000+
Dependents

150+
Contributors



Flower

A Friendly Federated AI Framework

AI Devs ❤️ Flower

5,700+
Developers

5,700+
Stars

2,000+
Dependents

150+
Contributors

The screenshot shows the discuss.flower.ai forum interface. The left sidebar contains navigation links for Topics, My Posts, Flower Leaderboard, Review, Admin, Invite, More, Categories (with General, Flower Help - Beginners, Flower Help - Intermediate, and Flower Framework), Tags (with flower, faq, metrics, release, datasets, and All tags), Messages (Inbox and moderators), Channels (General and Staff), and DMS. The main content area displays a list of categories: General, Flower Help - Beginners, Flower Help - Intermediate, Flower Framework, Introduction, Research, Industry, Contributions, and Flower Client SDKs. Each category has a brief description and a list of topics. For example, the 'General' category has 37 topics, including 'Is it possible to attach (long running) business logic in...' and 'Announcing Flower 1.7.0'. The 'Flower Help - Beginner' category has 63 topics, including 'Client not getting selected? How to debug?' and 'How to replace Parameter with Gradient? Instead of ...'. The 'Flower Help - Intermediate' category has 35 topics, including 'Not sure how to implement SecAgg() in this FL' and 'How can I implement a YOLO model using the Flower...'. The 'Flower Framework' category has 23 topics, including 'ONNX Runtime Support?' and 'How do i use the get_properties method on the se...'. The 'Introduction' category has 0 topics. The 'Research' category has 4 topics, including 'Feedback on Federated Learning Uni-seminar Pr...' and 'Aggregation algorithms'. The 'Industry' category has 0 topics. The 'Contributions' category has 1 topic, 'Hierarchical Federated Learning'. The 'Flower Client SDKs' category has 3 topics, including 'How to import Flower iOS SDK to my project?'. The top right corner of the interface features the Flower logo and the text 'A Friendly Federated AI Framework'.

discuss.flower.ai

AI Devs ❤️ Flower

5,700+
Developers

5,700+
Stars

2,000+
Dependents

150+
Contributors

The screenshot shows a Slack interface for the 'Flower' team. The left sidebar lists various channels and direct messages. The main window is focused on the '#questions' channel, which has 5,739 members. A message from 'Abderezak B' asks about aggregators in tree-based models. Alberto (Jesus) Zancanaro responds with a link to a pyproject.toml file and suggests a tutorial. A message from 'Victor Hidalgo' asks about getting client properties faster. The interface includes standard Slack features like messages, canvas, files, and pins.



AI Devs ❤️ Flower

5,700+
Developers

5,700+
Stars

2,000+
Dependents

150+
Contributors

The screenshot shows the 'Flower Baselines Documentation' page. At the top, there's a sidebar with the Flower logo and the text 'Flower Baselines 1.18.0'. Below the sidebar is a search bar and a navigation menu with links to 'HOW-TO GUIDES', 'Use Baselines', and 'Contribute Baselines'. The main content area has a heading 'Join the Flower Community' followed by a paragraph about the community and a 'Join us on Slack' button. Below this is a section titled 'Flower Baselines' with a sub-section 'REFERENCES'. A table lists various federated learning baselines, each with a 'Method', 'Dataset', and 'Tags' column.

Method	Dataset	Tags
dasha	cifar10, mushrooms, libsvm	compression, heterogeneous setting, variance reduction, image classification
depthfl	CIFAR-100	image classification, system heterogeneity, cross-device, knowledge distillation
fedavgm	CIFAR-10, Fashion-MNIST	non-iid, image classification
fedbn	MNIST, MNIST-M, SVHN, USPS, SynthDigits	data heterogeneity, feature shift, cross-silo
feddebug	cifar10, mnist	malicious client, debugging, fault localization, image classification, data poisoning
fedmeta	FEMNIST, SHAKESPEARE	meta learning, maml, meta-sgd, personalization
fedmlb	CIFAR-100, Tiny-ImageNet	data heterogeneity, knowledge distillation, image classification
fednova	CIFAR-10	normalized averaging, heterogeneous optimization, image classification
fedpara	CIFAR-10, CIFAR-100, MNIST	image classification, personalization, low-rank training, tensor decomposition
	CIFAR-10, FLICKR-	system heterogeneity, image classification, personalization



ICASSP

Build Federated Learning Apps with Flower

Flower CLI

```
flwr new    # Create a new Flower  
            app from a template  
  
flwr run    # Run your app  
  
flwr log    # View run logs  
  
flwr ls     # List run statuses  
  
flwr stop   # Stop a run
```

Flower CLI

- Build with templates

```
flwr new    # Create a new Flower  
            app from a template  
  
flwr run    # Run your app  
  
flwr log    # View run logs  
  
flwr ls     # List run statuses  
  
flwr stop   # Stop a run
```

Flower CLI

- Build with templates
- Research code = production code

```
flwr new      # Create a new Flower  
               app from a template  
  
flwr run      # Run your app  
  
flwr log      # View run logs  
  
flwr ls       # List run statuses  
  
flwr stop     # Stop a run
```

Flower CLI

- Build with templates
- Research code = production code
- Simulate ↔ Deploy

```
flwr new    # Create a new Flower  
            app from a template  
  
flwr run    # Run your app  
  
flwr log    # View run logs  
  
flwr ls     # List run statuses  
  
flwr stop   # Stop a run
```

Run Flower Simulations on a Single Machine

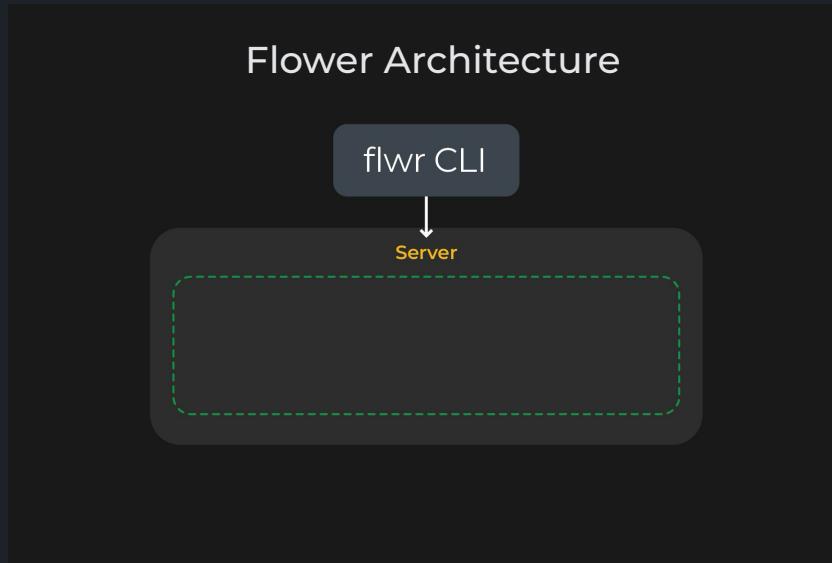


Tutorial Part #1

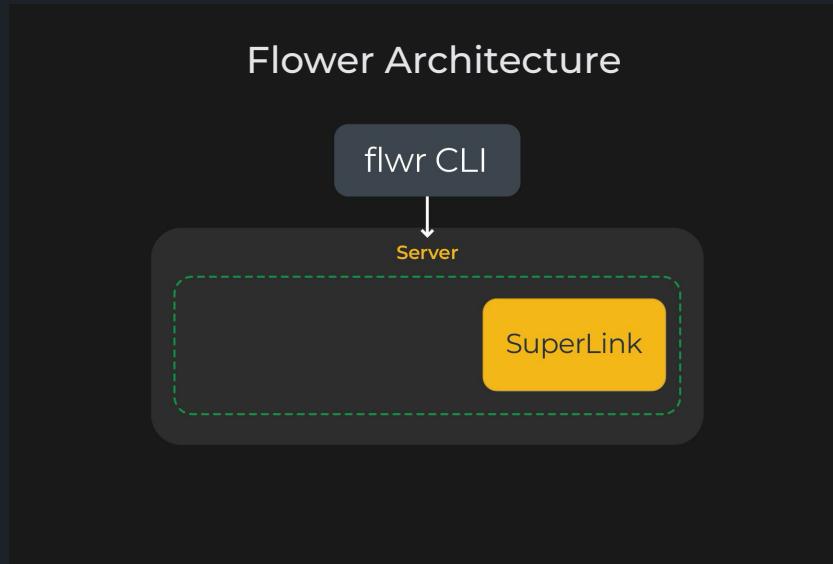
Run Flower Simulations on a Remote Machine



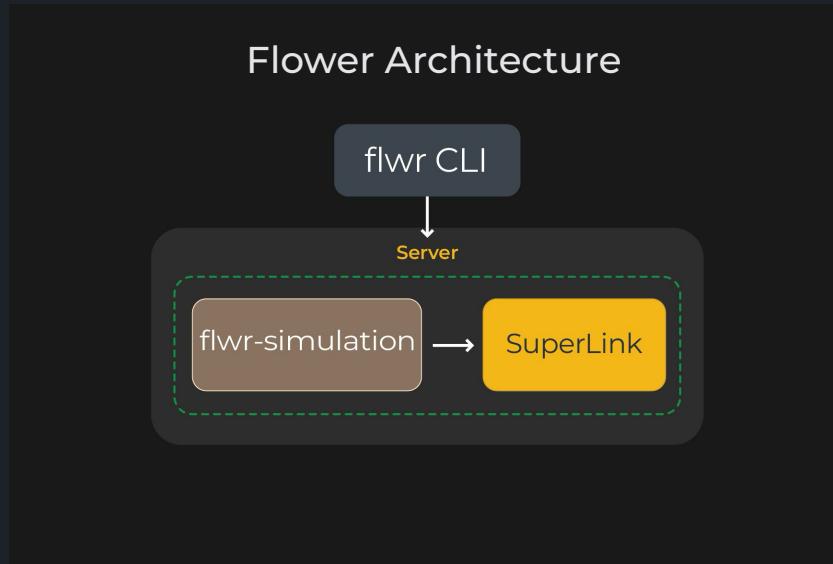
Run Flower Simulations on a Remote Machine



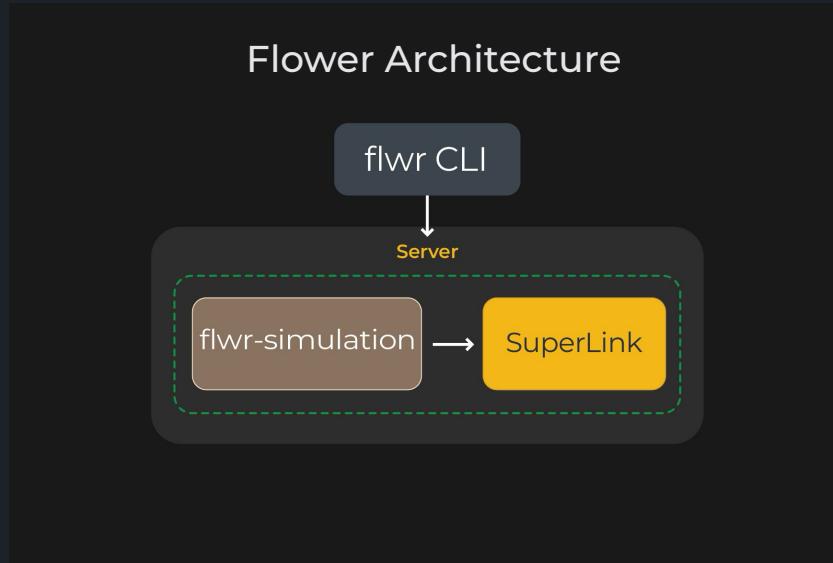
Run Flower Simulations on a Remote Machine



Run Flower Simulations on a Remote Machine



Run Flower Simulations on a Remote Machine



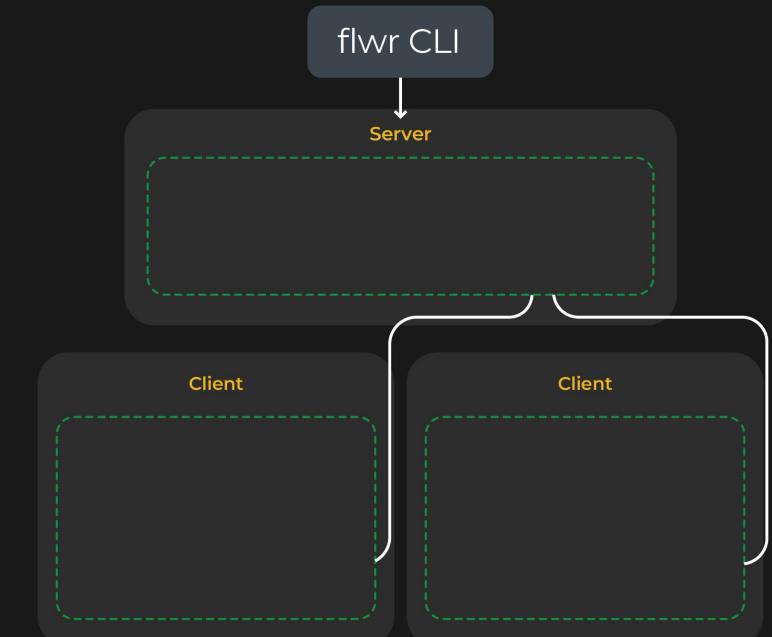
Let's try running a Flower Simulation at

👉 pyconde25.flower.ai

Tutorial Part #2

Deploy Flower on Distributed Systems

Flower Architecture

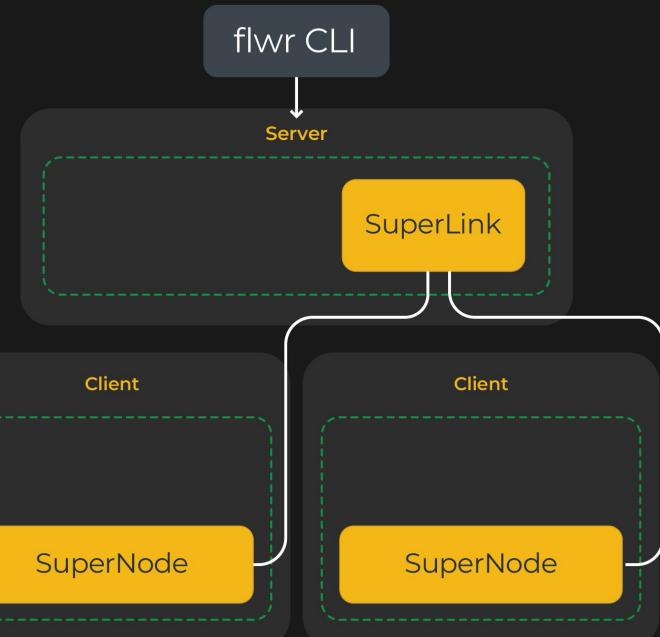


Flower

A Friendly Federated AI Framework

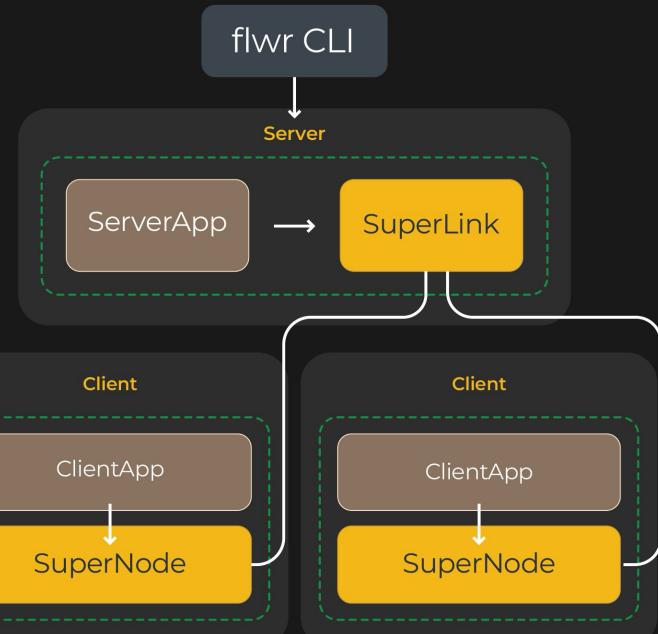
Deploy Flower on Distributed Systems

Flower Architecture



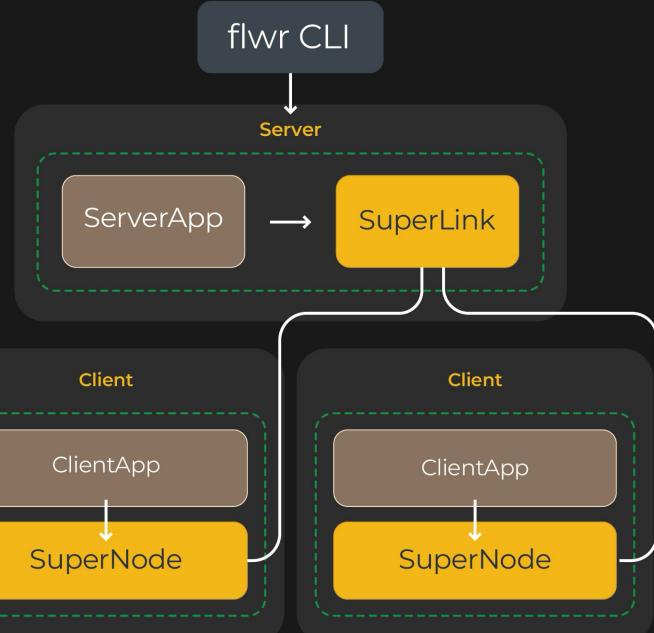
Deploy Flower on Distributed Systems

Flower Architecture



Deploy Flower on Distributed Systems

Flower Architecture *Subprocess Mode*

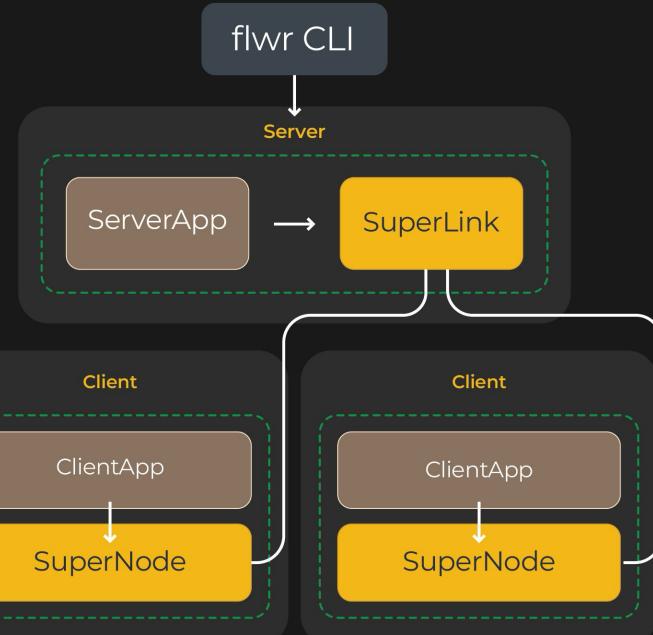


Flower

A Friendly Federated AI Framework

Deploy Flower on Distributed Systems in 3 steps

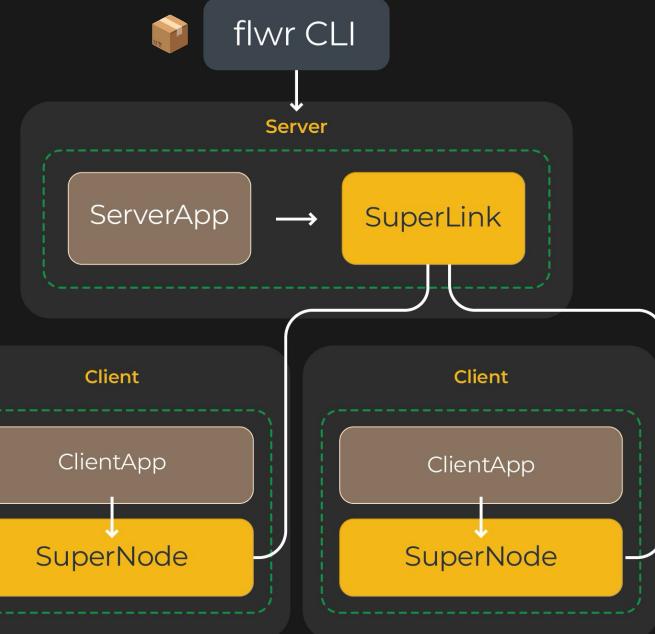
Flower Architecture *Subprocess Mode*



Deploy Flower on Distributed Systems in 3 steps

```
# 📦 Install Flower on machine  
$ pip install flwr
```

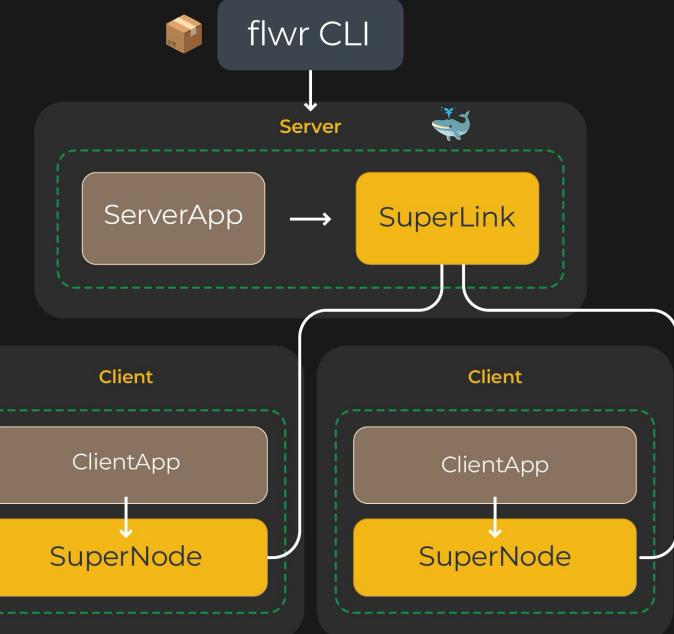
Flower Architecture *Subprocess Mode*



Deploy Flower on Distributed Systems in 3 steps

```
# 📦 Install Flower on machine  
$ pip install flwr  
  
# 🐳 Start SuperLink container  
$ docker run flwr/superlink*
```

Flower Architecture Subprocess Mode

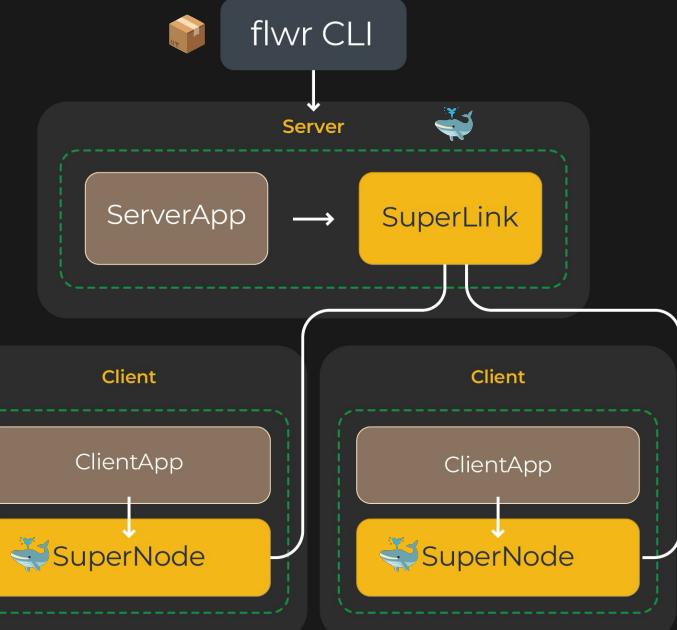


*Plus additional arguments for ports, certificates, volumes, etc ...

Deploy Flower on Distributed Systems in 3 steps

```
# 📦 Install Flower on machine  
$ pip install flwr  
  
# 🐳 Start SuperLink container  
$ docker run flwr/superlink*  
  
# 🐳 Start SuperNode container  
$ docker run flwr/supernode*
```

Flower Architecture Subprocess Mode

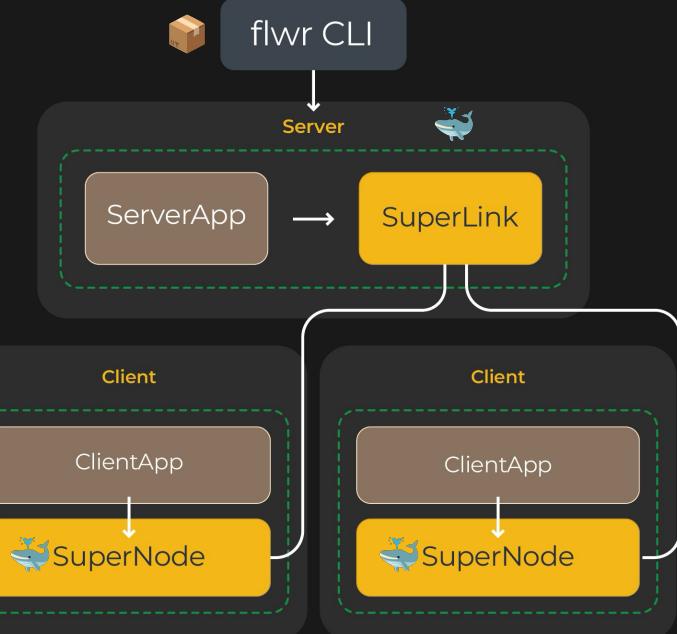


*Plus additional arguments for ports, certificates, volumes, etc ...

Deploy Flower on Distributed Systems in 3 steps

```
# 📦 Install Flower on machine  
$ pip install flwr  
  
# 🐳 Start SuperLink container  
$ docker run flwr/superlink*  
  
# 🐳 Start SuperNode container  
$ docker run flwr/supernode*
```

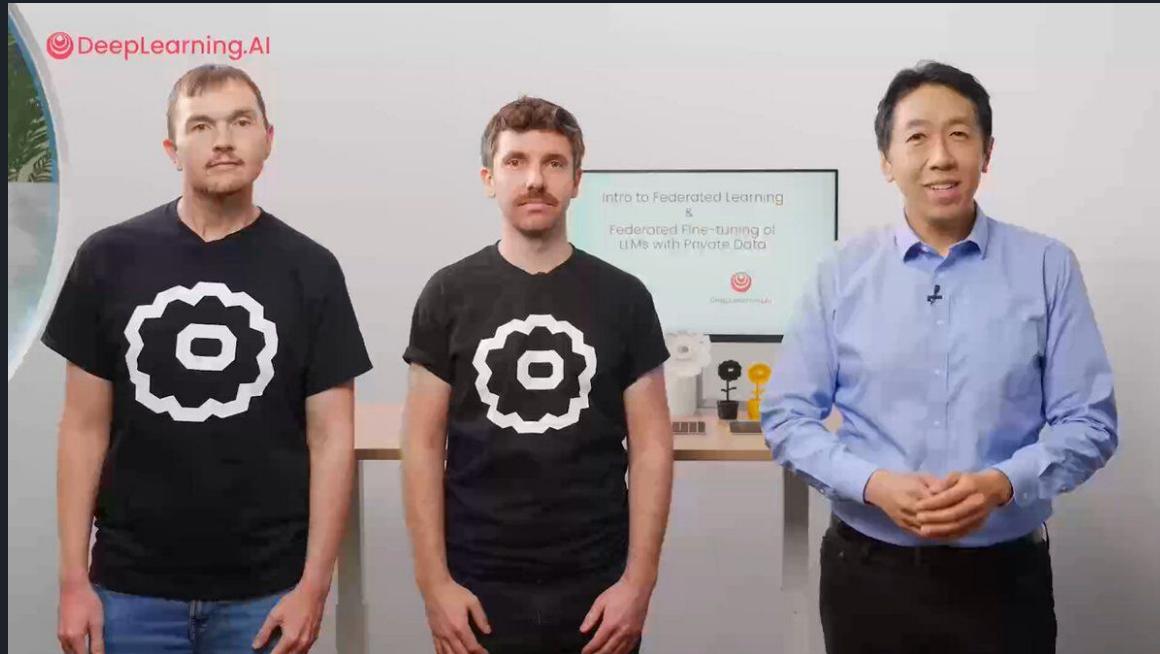
Flower Architecture Subprocess Mode



*Plus additional arguments for ports, certificates, volumes, etc ...

Tutorial Part #3

What's next?



deeplearning.ai/short-courses/intro-to-federated-learning/

Come join us at the Flower Monthly !

Building a Federation of Robots
LeRobot x Flower

Flower Monthly

LeRobot and Flower: Scaling Data and Compute for Real World Robotics

Ivelin Ivanov
8 Jan 16:00 GMT+0

Federating Medical Foundation Models

Flower Monthly

Federated Fine-Tuning a Foundation Model for Disease Detection

Eden Ruffell
8 Jan 16:00 GMT+0

Flower Monthly
2024-08-07

Achieving Debugging and Interpretability in Federated Learning Systems

Waris Gill
PhD Student at Virginia Tech

VT

Achieving Debugging and Interpretability in Federated Learning Systems

Waris Gill
7 Aug 17:00 GMT+1

PUFFLE: Balancing Privacy, Utility, and Fairness in FL
Luca Corbucci

Flower Monthly

PUFFLE: Balancing Privacy, Utility, and Fairness in Federated Learning

Luca Corbucci
6 Nov 16:00 GMT+0

Towards Active Participant-Centric Vertical Federated Learning
Jon Irureta Barrena

Flower Monthly

Towards Active Participant-Centric Vertical Federated Learning

Jon Irureta Barrena
4 Dec 16:00 GMT+0

Serverless and Asynchronous FL for Medical Imaging
ZZ Si

Flower Monthly

lvr-serverless: A Serverless and Asynchronous Federated Learning Recipe for Medical Imaging

ZZ Si
6 Nov 16:00 GMT+0

Flower Monthly
2024-09-04

EncCluster: Scalable Functional Encryption in Federated Learning

Samaneh Mohammadi
Industrial PhD Student

RISE Research Institutes of Sweden

EncCluster: Scalable Functional Encryption in Federated Learning

Samaneh Mohammadi
4 Sep 17:00 GMT+1

chong.shen@flower.ai



A Friendly Federated AI Framework



