



UNIVERSITY OF ZAGREB  
Faculty of Electrical  
Engineering and  
Computing



TECHNISCHE  
UNIVERSITÄT  
WIEN



# AIoTwin

Twining action for spreading excellence in Artificial Intelligence of Things

# Introduction to Federated Learning

Jicheng Yuan, Duc Manh Nguyen  
Technical University of Berlin



Grant agreement ID: 101079214

# Content

---

- Introduction to Federated Learning
- Federated Learning in Five-Steps
- Quick View of the Wandb
- Let's hand on it!





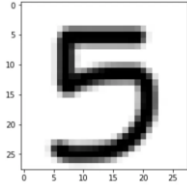

**AloTwin**

# Introduction to Federated Learning

---

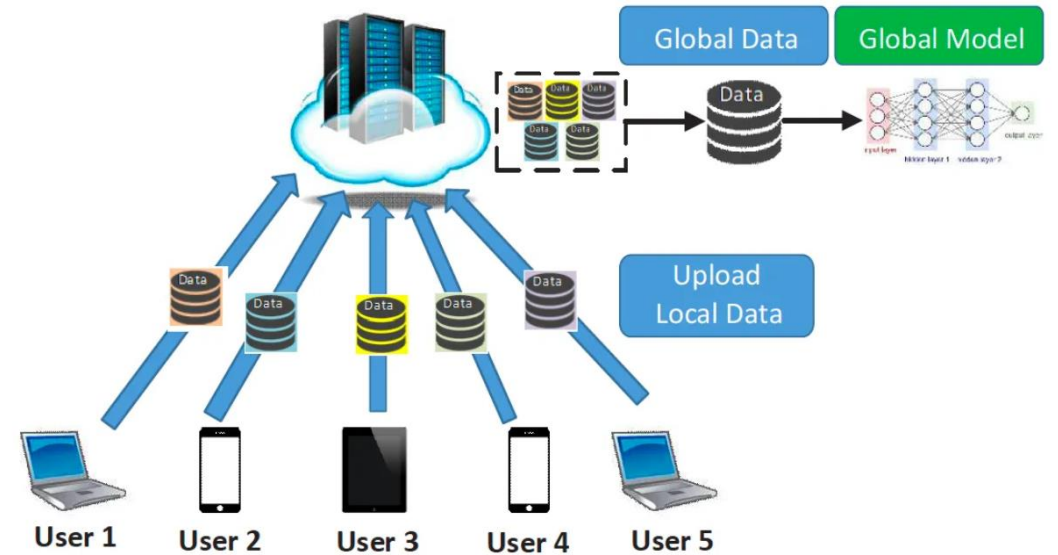
# The goal of Deep Learning

- Find a function, which gives a desired output given a particular input.

Tasks	Given Input	Desired Output
Image Recognition (CV)		5
Next Word Prediction (NLP)	Looking forward to your <u>?</u>	reply
Q-Learning (RL)		Next action

# Centralized Learning

- Centralized machine learning requires storing data on a single machine or in a datacenter
- The centralized entity trains and hosts the model
- Used in outsourced models (ML-as-a-service) e.g. Amazon AWS ML
- Central trainer has access to sensitive training data



# Why Federated Learning?

---



- The needs of IoT applications in deep learning.
  - Machine learning (Deep learning) needs more data.
  - Regulations: GDPR (Europe), CCPA (California), PIPEDA (Canada)...
  - User preference: users concern about privacy and security.
  - Benefit from Edge computing: Data volume, bandwidth...
- Federated Learning.



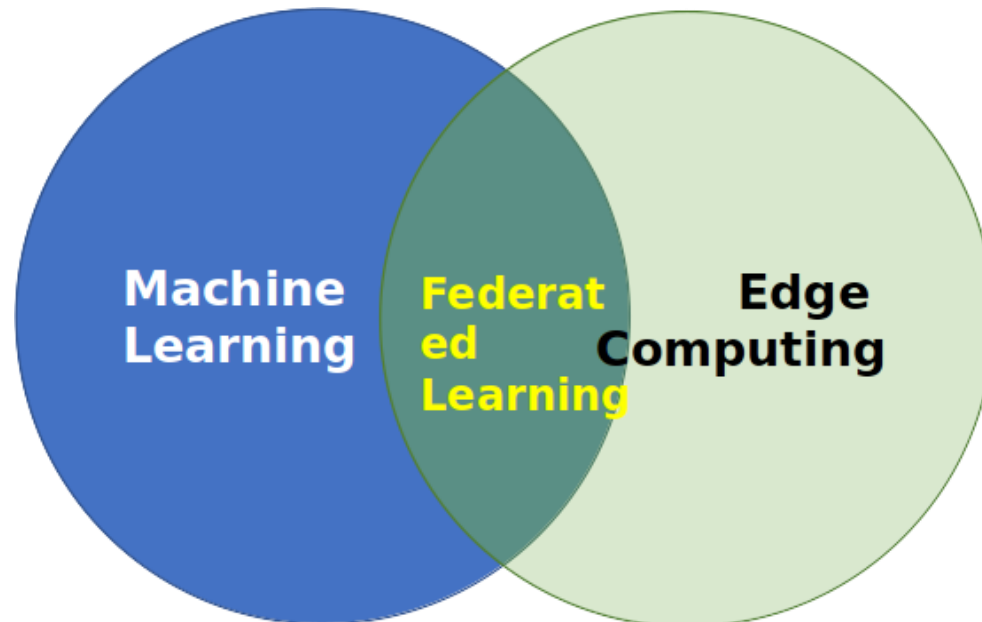
**AloTwin**

# Federated learning in five steps [2]

---

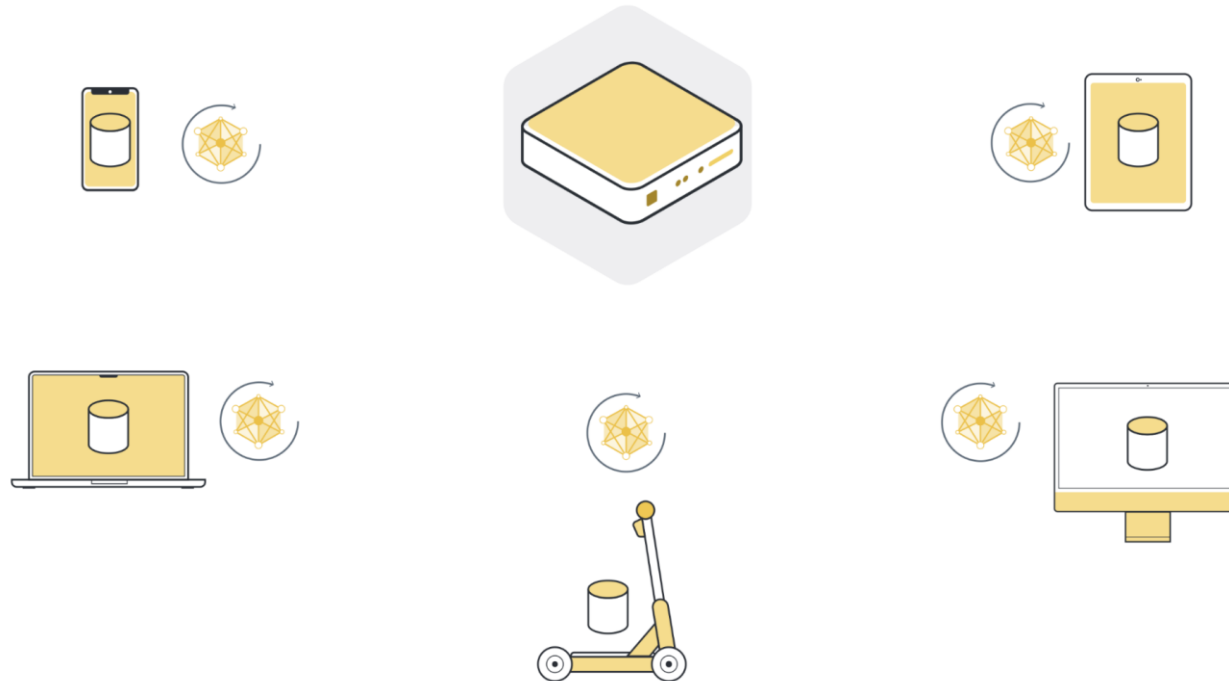
# What is Federated Learning?

- Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred. (\*arXiv:1912.04977)



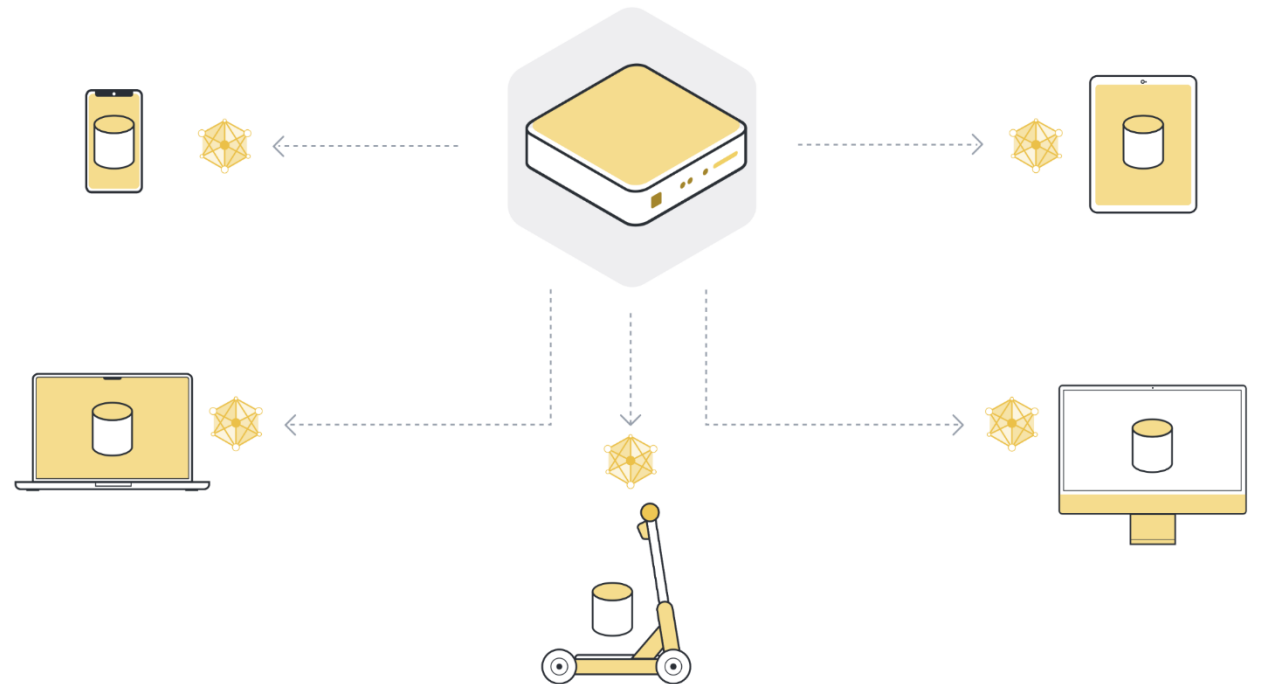
# 1. Initialization

- The devices will be given a training model which is initialized in the central server



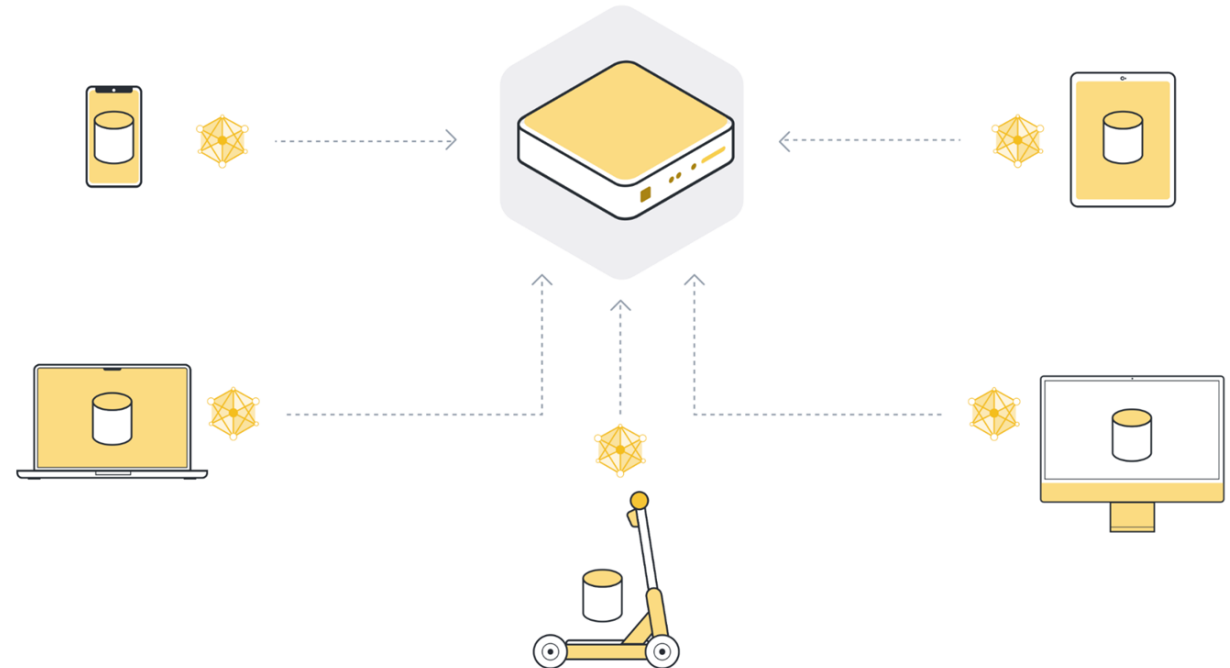
# 2. Model Distribution and Training

- The devices will be given a training model which is initialized in the central server
- Train model locally on the data of each organization/device



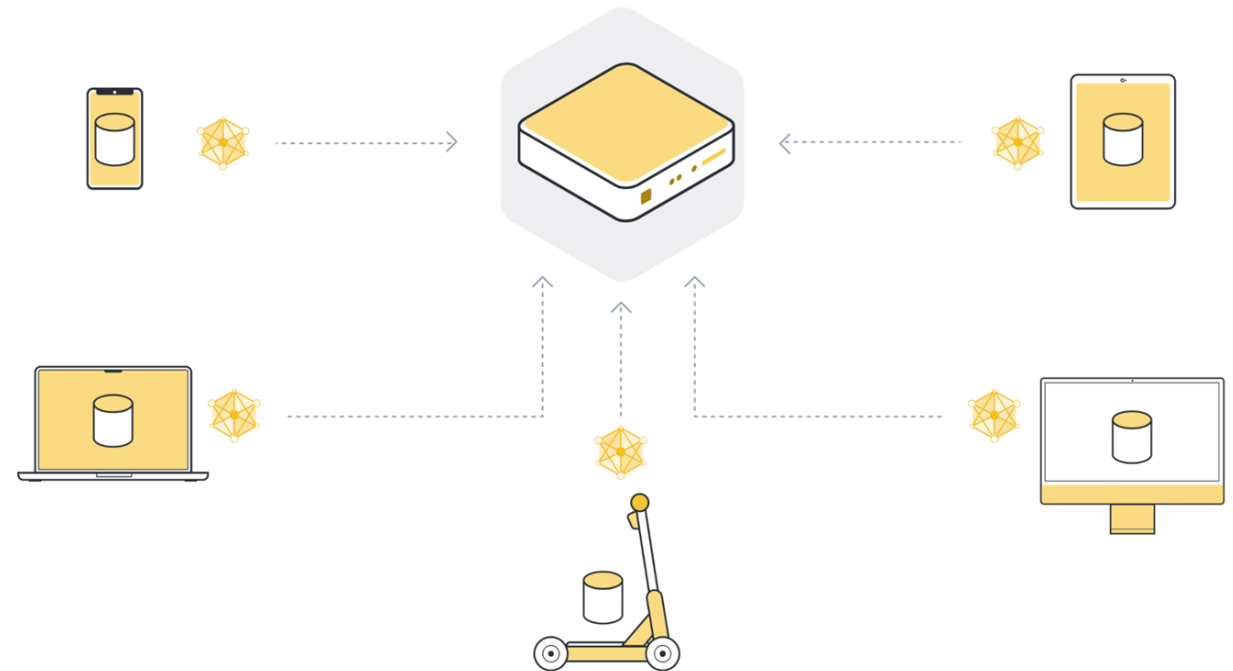
# 3. Model Aggregation

- The devices will be given a training model which is initialized in the central server
- Train model locally on the data of each organization/device
- Send the encrypted updates back to the server



# 4. Model Updates

- The devices will be given a training model which is initialized in the central server
- Train model locally on the data of each organization/device
- Send the encrypted updates back to the server and aggregate model updates
- Perform one update to the current global model



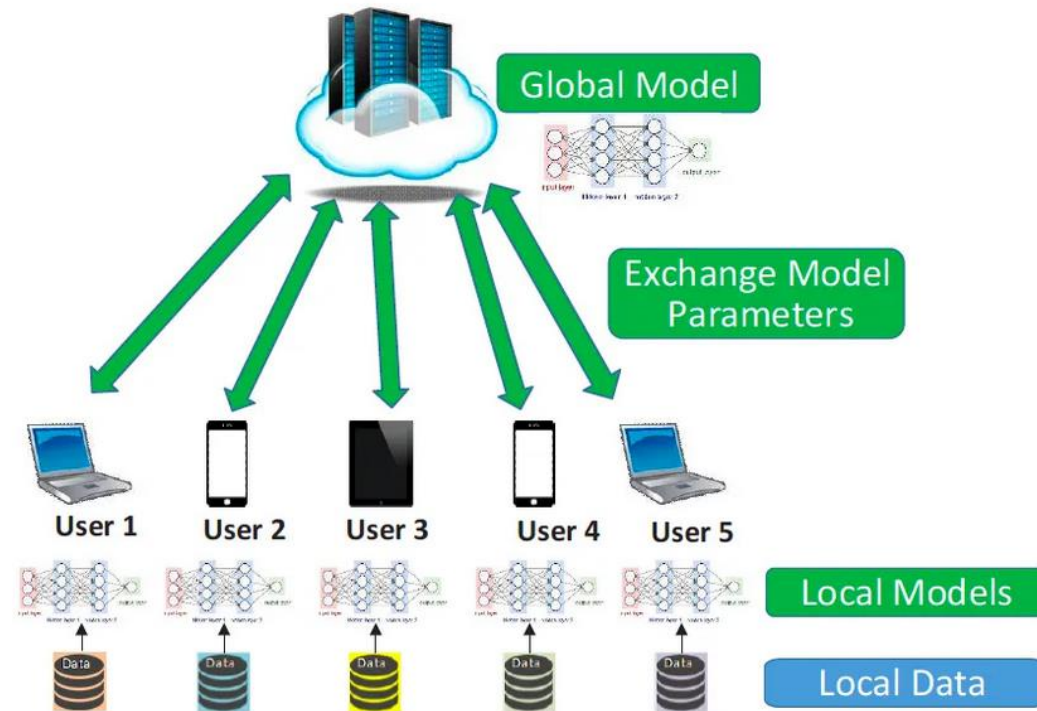
# Federated Learning in Five Steps



- i. The devices will be given a training model which is initialized in the central server
- ii. Train model locally on the data of each organization/device
- iii. Send the encrypted updates back to the server
- iv. Aggregate model updates into a new global model and perform one update to the current model
- v. After rounds of training, the new updated model will be sent to the devices for on-device testing and then start a new round of training

# Challenges in Federated Learning

- Non-IID: The data generated by each user are quite different
- Imbalanced: Some users produce significantly more data than others
- Massively distributed: Heterogeneous devices. (Reliability of Edges Devices)
- Security and privacy (Poisoning Attack)



# Weights & Biases



```

0000 p1 DRZ 3.11 - not super eval
mag_threshold 0.21
2L 300N BLSTM (BasicLSTM)
20D
sigmoid
AdamOptimizer
100 frames
dropout 1.0
zero input and label
log(x+1.0)
103300 training, 2000 CV
model:
weights20170224-005946_v10
(p1, loss .1419, epoch 40
MEAN IBM SDR GAIN: 2.
with 0.15 thresh during cl
STD IBM SDR GAIN: 2.
MEAN IBM SDR GAIN: 2.
with 0.32 threshold during
STD IBM SDR GAIN: 2.

0001 p2 DRZ 3.23
mag_threshold 0.12
2L 300N BLSTM_clean (LSTM & many
reworks) - note, this was the
massive model rewrite
20D
sigmoid
AdamOptimizer
100 frames
dropout 1.0
zero input and label

```

Experiment Name	Created	train_loss	valid_loss	acc	traffic_acc	road_acc
best car acc (50% data)	2021-04-14	0.5375041962	0.442730248	0.8823291659	0.8663836718	0.9399003386
best traffic acc (50% data))	2021-04-14	0.4919361174	0.4202951491	0.8879730701	0.8718349934	0.9439761043
best overall IOU (20% data)	2021-04-14	0.5095784068	0.4658596516	0.8725891709	0.8592621684	0.9359762073
major-sweep-196	2021-01-31	0.5705417991	0.4875227213			
swept-sweep-164	2021-01-31	0.5535062551	0.4849829972			
silver-sweep-139	2021-01-31	0.563354373	0.5251165628			
laced-sweep-115	2021-01-31	0.5277443528	0.5124291778			
eager-sweep-97	2021-01-31	0.5488699675	0.5005864501			
rich-sweep-88	2021-01-31	0.5587444901	0.5211353302			
hopeful-sweep-33	2021-01-31	0.503461957	0.4650281966			
autumn-sweep-24	2021-01-31	0.5777919888	0.500880897			
decent-sweep-21	2021-01-31	0.5714729428	0.4979581237			
vague-sweep-5	2021-01-31	0.6230331063	0.473508656			
Second best acc	2021-01-31	0.4194990396	0.4509823024			

**TensorBoard** SCALARS IMAGES GRAPHS > INACTIVE

Show data download links  
 Ignore outliers in chart scaling

Tooltip sorting: default  
 method:

Smoothing:  0.6

Horizontal Axis: STEP RELATIVE WALL

Runs: Write a regex to filter runs

train  eval

TOGGLE ALL RUNS

/tmp/mnist-logs

Name	Smoothed Value	Step	Time	Relative
eval/mean	0.02591	170.0	Mon Sep 12, 15:40:41	8s
train/mean	0.02851	0.03362	166.0	Mon Sep 12, 15:40:40

# Weights & Biases



# Weights & Biases

Runs (398)

Search

Name (84 visualized) acc

good-cosmos-425	0.4031
logical-energy-420	0.626
laced-dust-419	0.5968
whole-music-418	0.6139
grateful-glitter-417	0.2367
clear-night-415	0.5403
glorious-night-414	0.7627
smart-sponge-413	0.6517
atomic-feather-412	0.6913
sunny-cloud-411	0.6291
fragrant-bee-410	0.346
soft-eon-408	0.3354

Search panels

Hyperparameter Optimization 2

Parameter importance with respect to traffic\_acc

Search Parameters 1-10 of 15

Config parameter	Importance	Correlation
Runtime	High	High
learning_rate	Medium	Low
num_train	High	High

Model Predictions 3

prediction

camera view


# Weights & Biases

---

- Rapidly iterate: continuously refine and optimize models
- Reproduce: to reduce key-person dependencies
- Collaborate: ensure knowledge transfer across teammates



# Weights & Biases



- Documents: <https://docs.wandb.ai/>
- Learn Wandb & MLOps: <https://wandb.courses/>
  
- Create account: <https://wandb.ai/>
- Get key: `WANDB_API_KEY=c8bb68d3bbc09627eb...`

# Let's play!

1. Access devices.
2. Part I: Centralized Learning.
3. Part II: Federated Learning.
4. Monitoring the Training Process using Wandb.



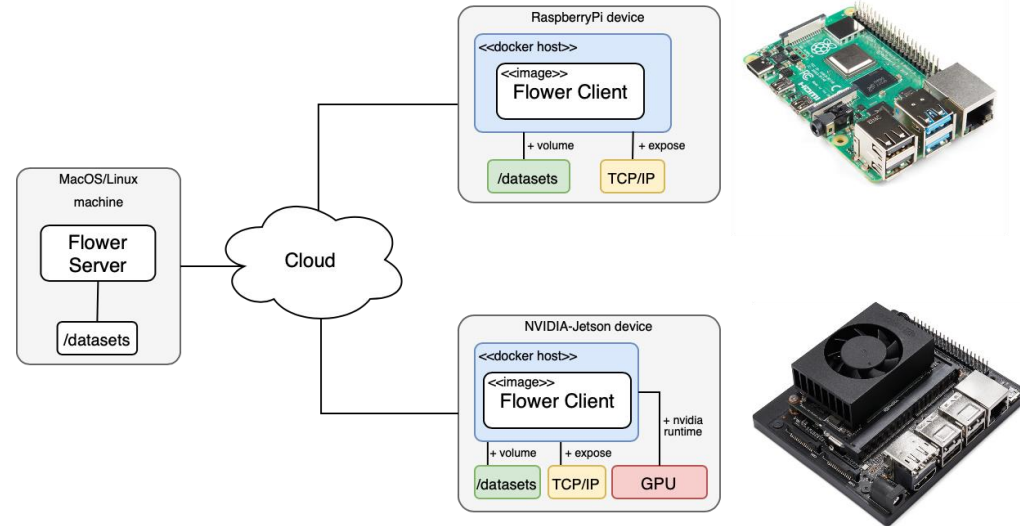
# Setup

## General:

- Data: CIFAR-10
- Deep learning Model: CNN

## Federated Learning:

- Data: chunks of CIFAR-10 in IID
- Algorithms: FedAvg



# References

---

- [1] <https://ai.googleblog.com/2017/04/federated-learning-collaborative.html>
- [2] <https://flower.dev/docs/tutorial/Flower-0-What-is-FL.html>

