



NANYANG  
TECHNOLOGICAL  
UNIVERSITY  
SINGAPORE

# Privacy Preservation

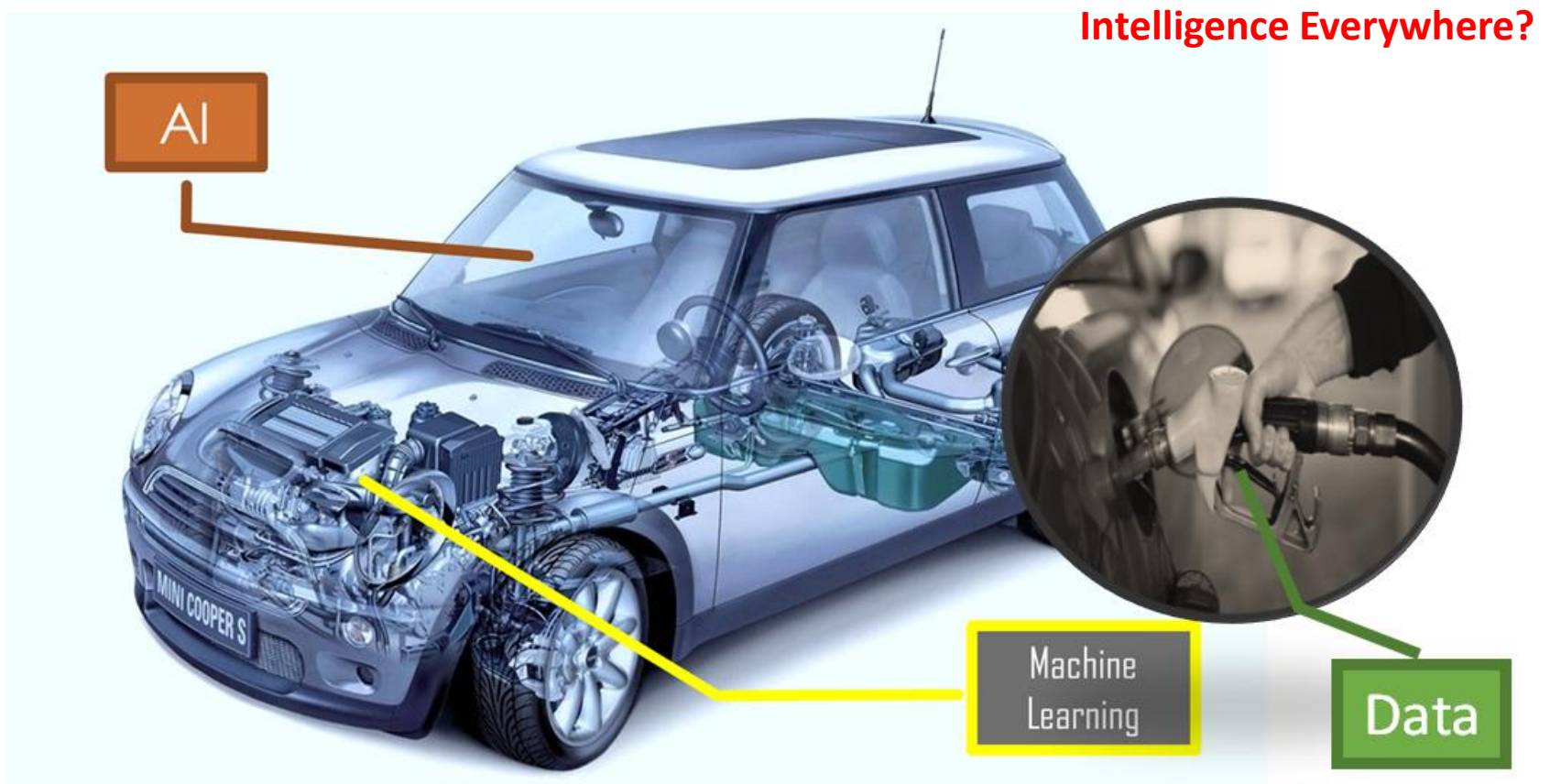
Yu Han

[han.yu@ntu.edu.sg](mailto:han.yu@ntu.edu.sg)

*Nanyang Assistant Professor  
School of Computer Science and Engineering  
Nanyang Technological University*

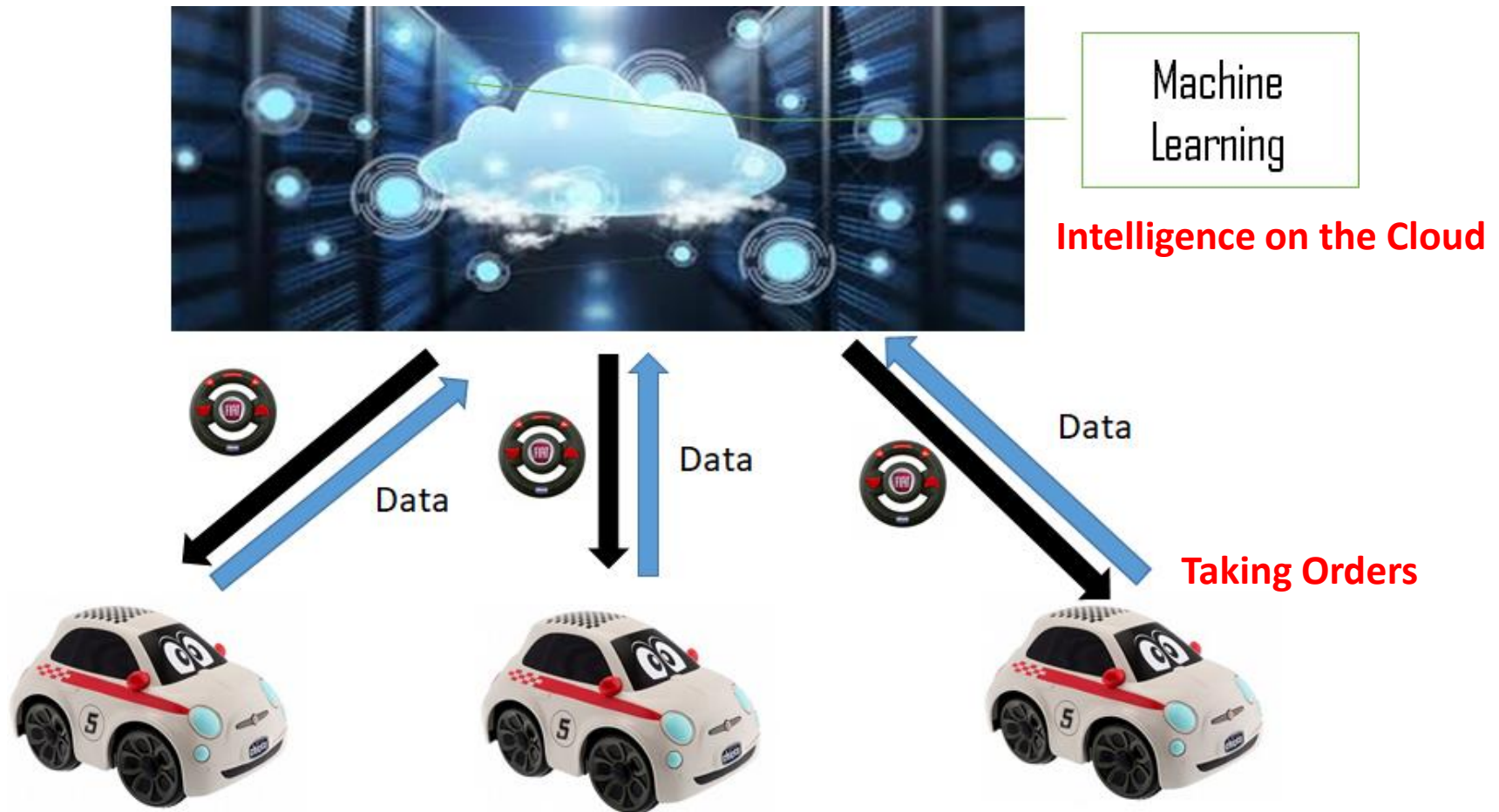


# Data, ML & AI (Ideally)





# Data, ML & AI (Reality)



# Data is the “New Oil”

computing power

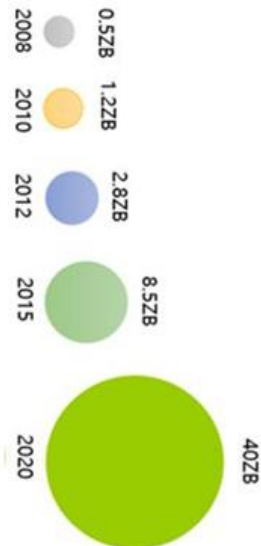
big data

1 ZB=  $10^{21}$ Byte



*Intel i386*  
*Intel i486*  
*Intel Pentium*  
*Intel Core*  
*nVidia GPU*  
*Google TPU*

来自互联网数据中心 (IDC)



The New Rich



# Challenge: Data Privacy Protection

Market summary > Facebook, Inc. Common Stock  
NASDAQ: FB - Mar 19, 2:21 PM EDT

172.32 USD ↓12.77 (6.90%)

1 day 5 day 1 month 3 month 1 year 5 year max



Open	177.01	Mkt cap	500.59B
High	177.17	P/E ratio	27.97
Low	170.06	Div yield	-

- More than **50** million people involved
- UK fined Facebook for **£500,000**
- **The worst single-day market value drop for a publicly listed company in the US, dropping \$120 billion, or 19%**

French regulator fines Google \$57 million for GDPR violations

[Share on Facebook](#) [Share on Twitter](#) [+](#)



# GDPR



- No Autonomous Modeling and Decision
- Interpretability of Model Decisions
- Users' Right for Data to be Forgotten
- Data Privacy By Design
- Explicit Consent for Data Usage

# Why Federated Learning?

---

- Traditional machine learning methods need all data to be gathered in a central entity
- In many real-world applications data are isolated across different organizations and data privacy is being emphasized
- Federated learning (FL) is well suited for these scenarios due to its distributed and privacy-preserving nature

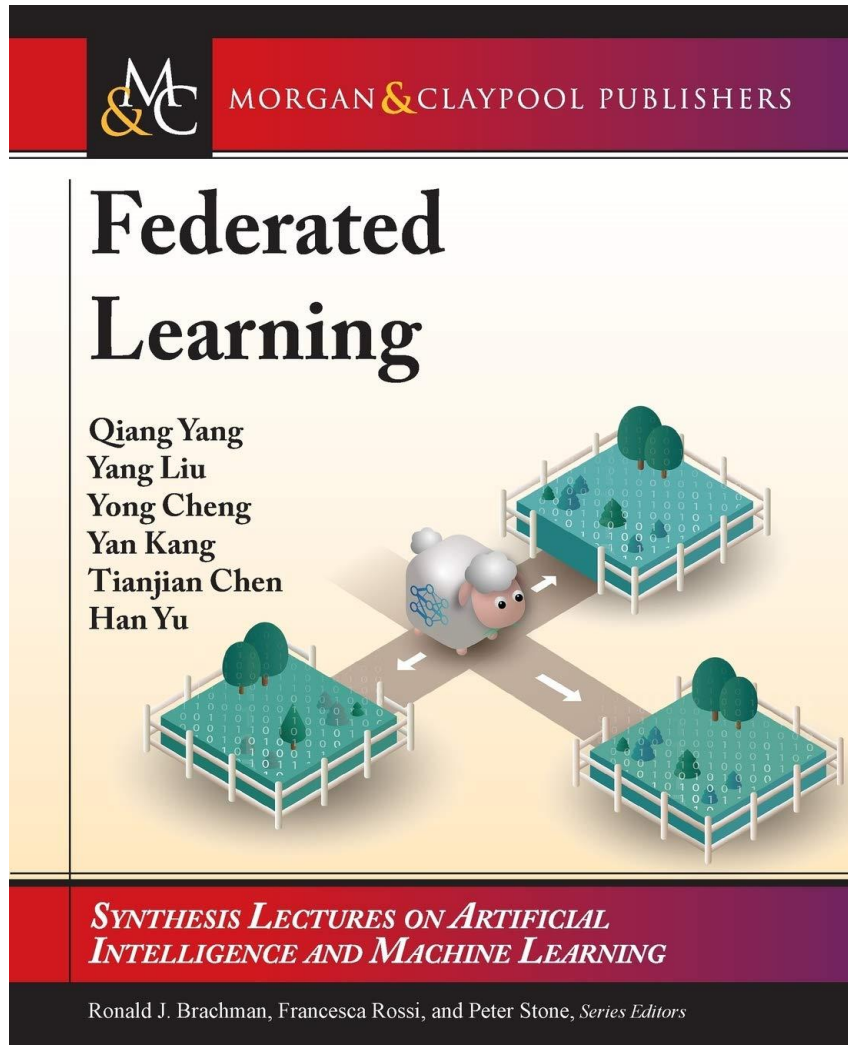
# What is Federated Learning?

---

- A new approach for models trained from user interaction with distributed devices.
  - **distributes** the machine learning process over to the edge.
  - enables devices to **collaboratively learn** a shared model using the training data on the device and **keeping the data on device**
  - decouples the **need for doing machine learning** with the **need to store the data** in the cloud



# Text Book



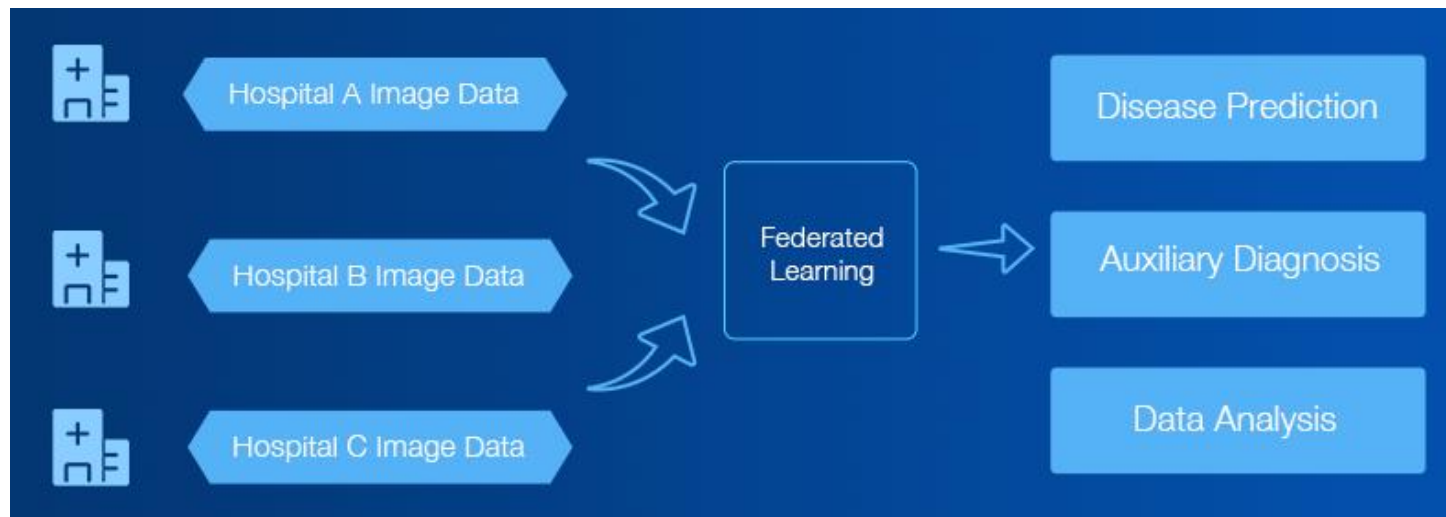
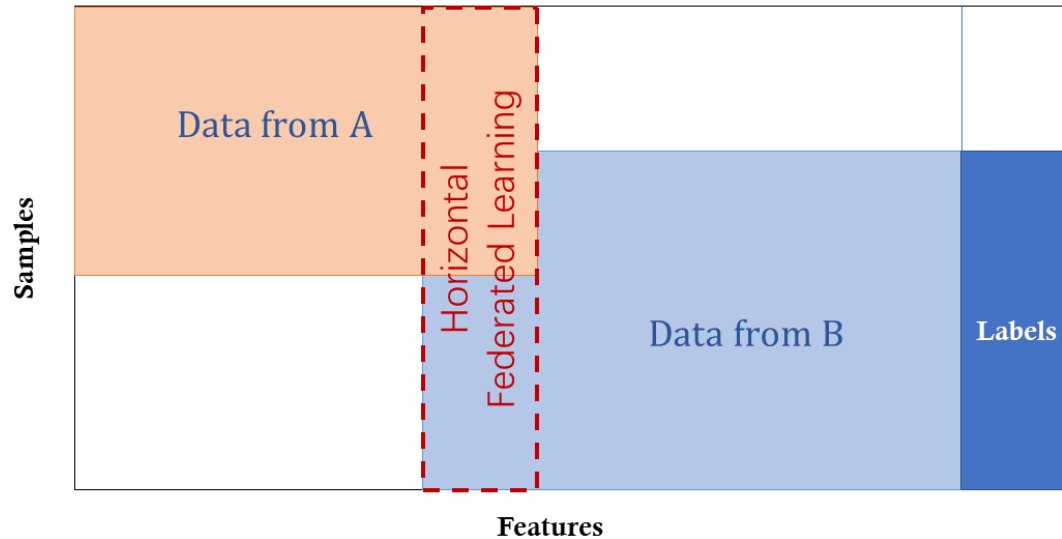
E-Book can be found from NTU Online Library:

[https://ntu-sp.primo.exlibrisgroup.com/discovery/search?vid=65NTU\\_INST:65NTU\\_INST&lang=en](https://ntu-sp.primo.exlibrisgroup.com/discovery/search?vid=65NTU_INST:65NTU_INST&lang=en)

Additional Resources can be found at:

<http://federated-learning.org/>

# Horizontal Federated Learning (HFL)



# Horizontal Federated Learning (HFL)

---

- HFL assumes that datasets from different participants **share the same feature space, but may not share the same sample ID space**
- Existing FL approaches mostly focus on HFL

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T. & Yu, H. (2019) *Federated Learning*. Morgan & Claypool Publishers, San Rafael, CA, USA, p. 207.

# HFL Architecture

---

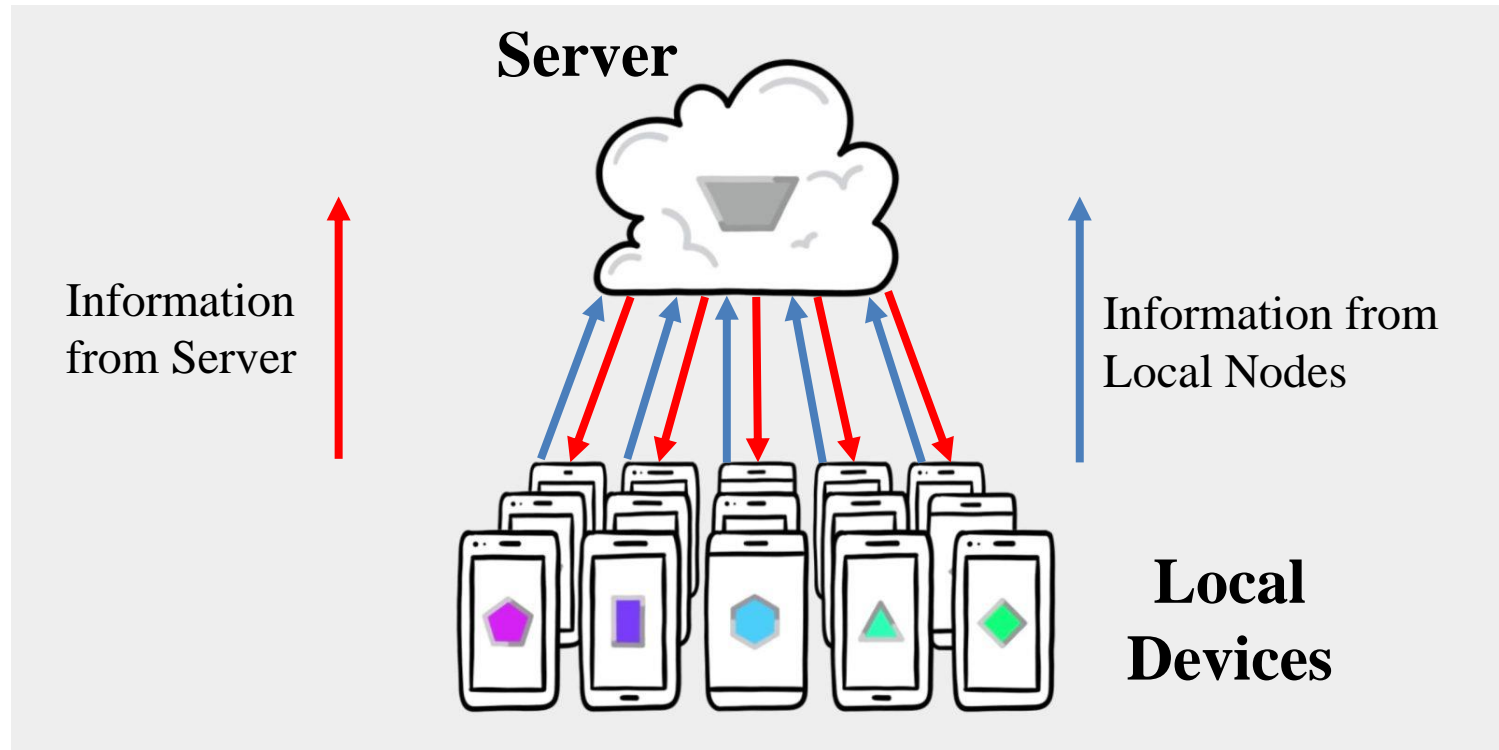
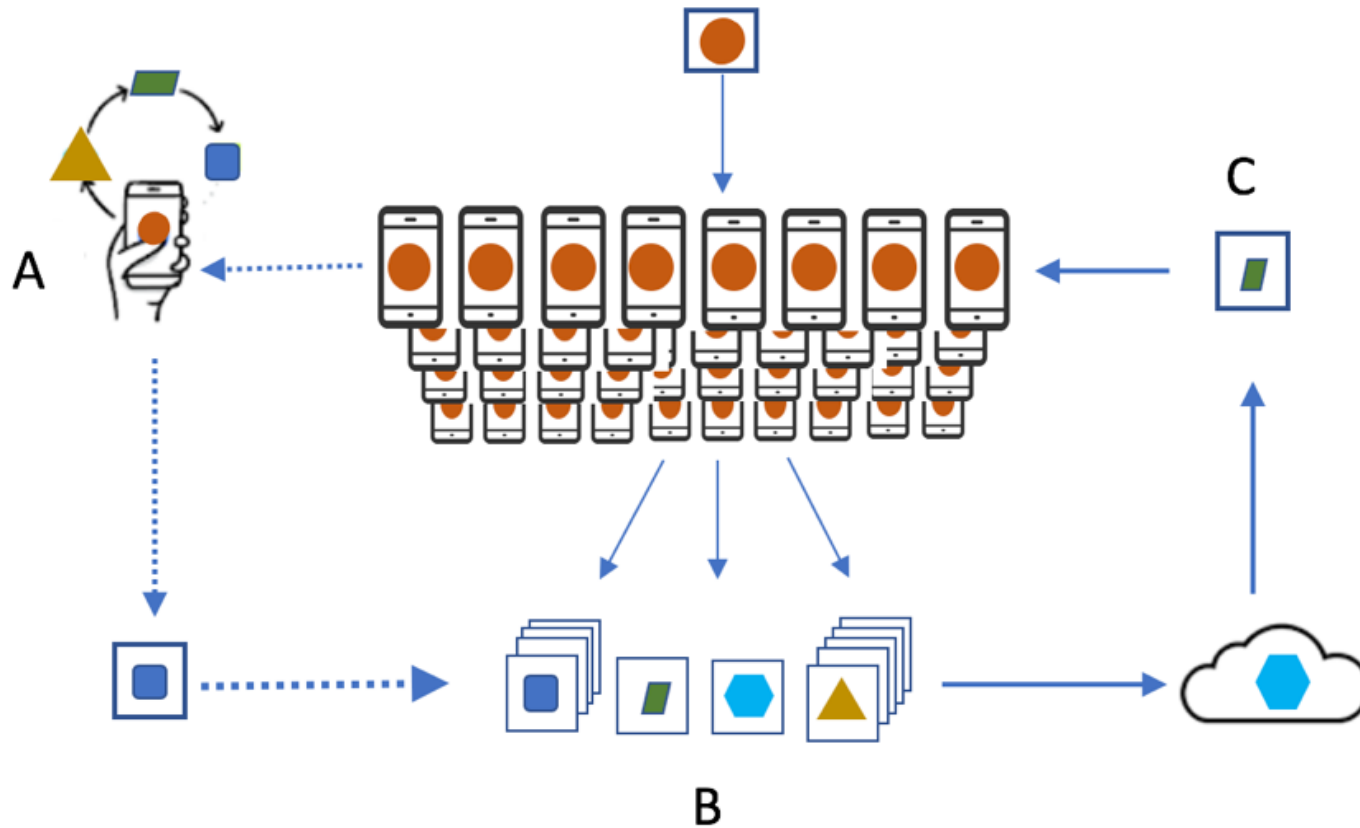


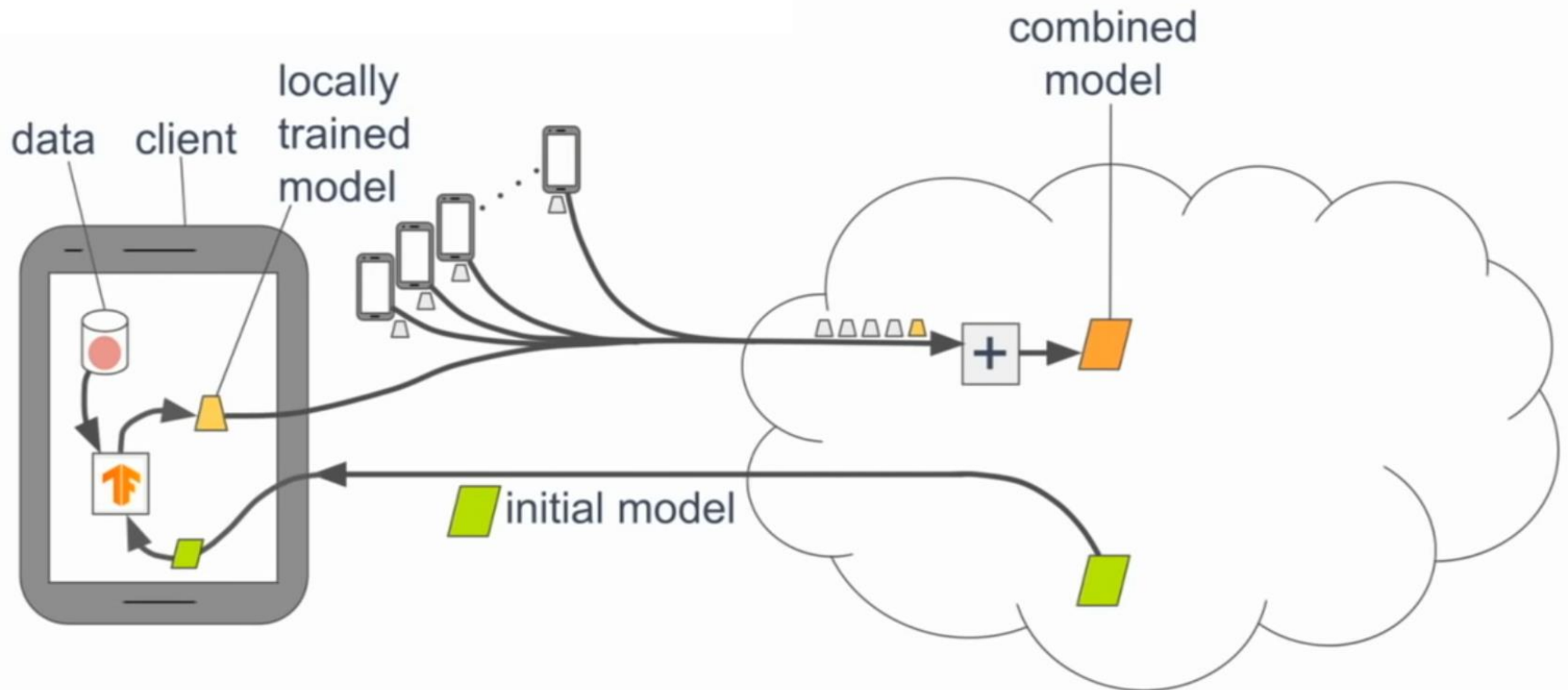
Figure 1: The general architecture of an HFL system



# Federated Learning

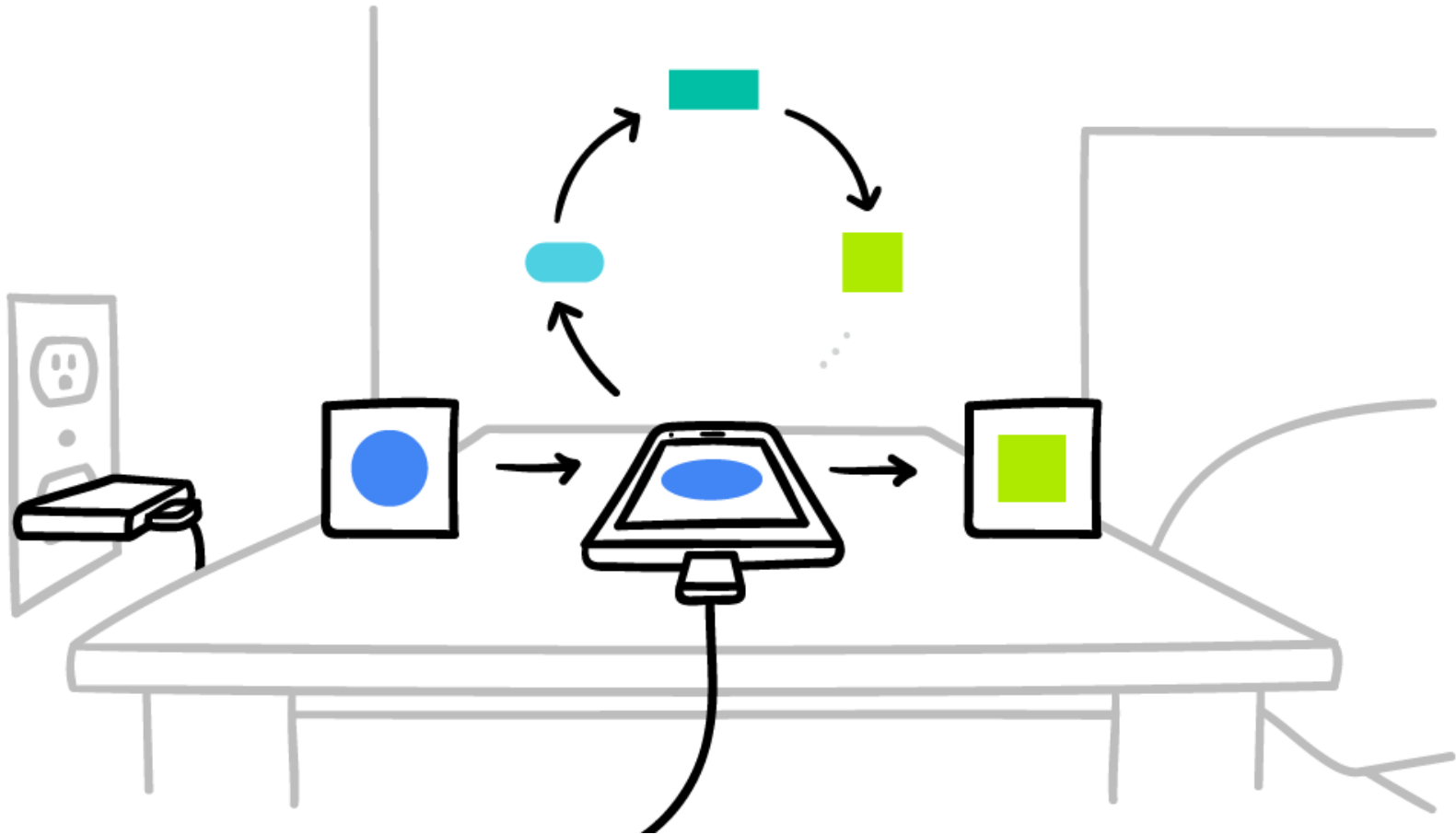


# Federated Learning (Google)



# Federated Learning (Google)

---



# How to Send Gradients to Server?

---

- Federated Stochastic Gradient Descent (FedSGD)
- Federated Averaging (FedAvg)

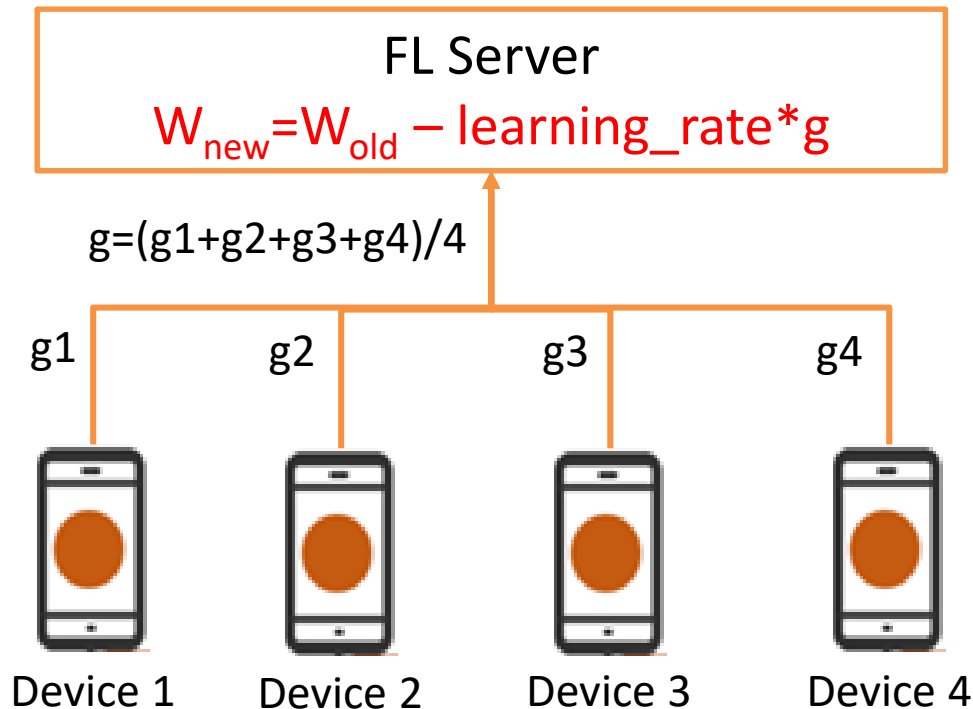


# FedSGD

---

- Devices send gradients/parameters to server
- Server averages these gradients/parameters to obtain a new model
- Server sends the new model back to devices
- High communication overhead

# FedSGD, C=1



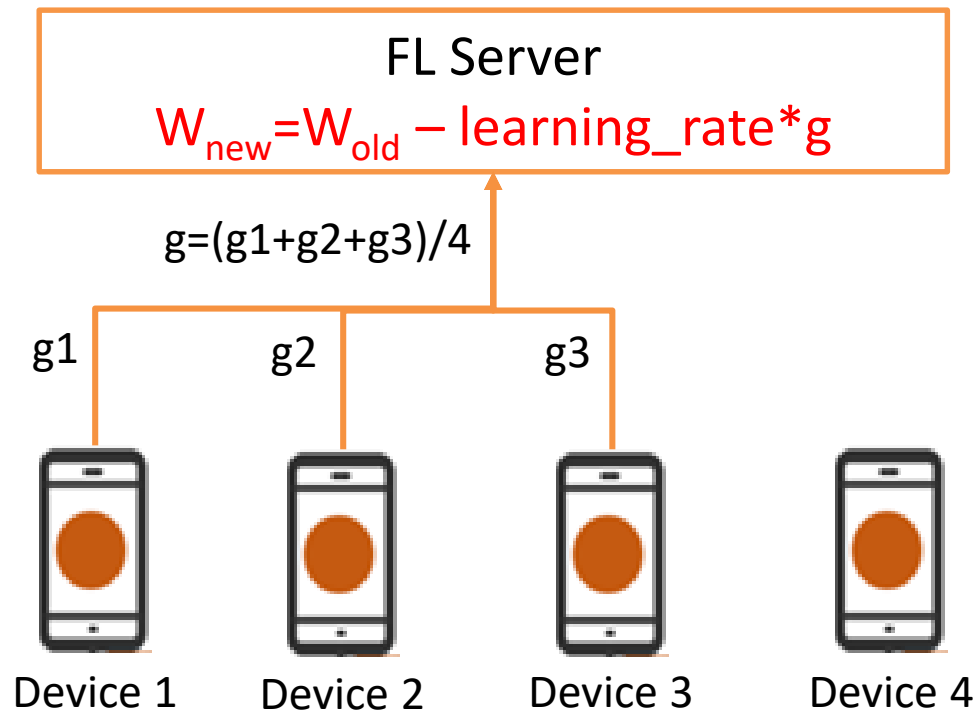
Version 1:

- Sending **gradients**
- The gradient descent operation happens on the FL server
- We set **C=1**, meaning 100% of the devices participate in FedSGD

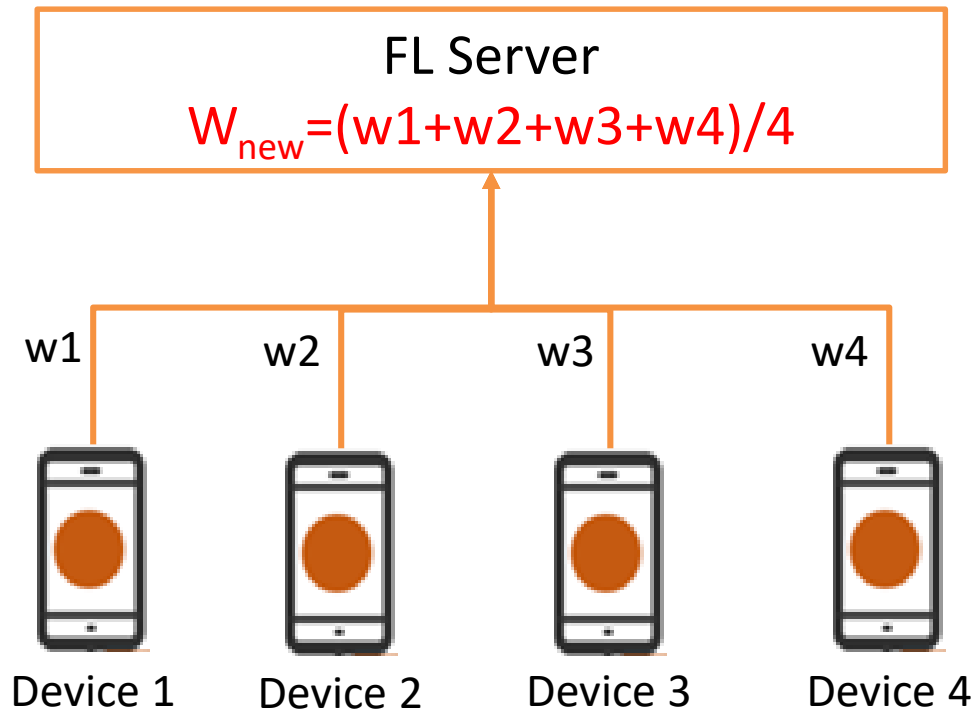
# FedSGD, C=0.75

Version 1:

- Sending **gradients**
- The gradient descent operation happens on the FL server
- We set **C=0.75**, meaning 75% of the devices participate in FedSGD



# FedSGD, C=1



Version 2:

- Sending **parameters** (i.e. weights)
- The gradient descent operation happens on the devices
- We set **C=1**, meaning 100% of the devices participate in FedSGD

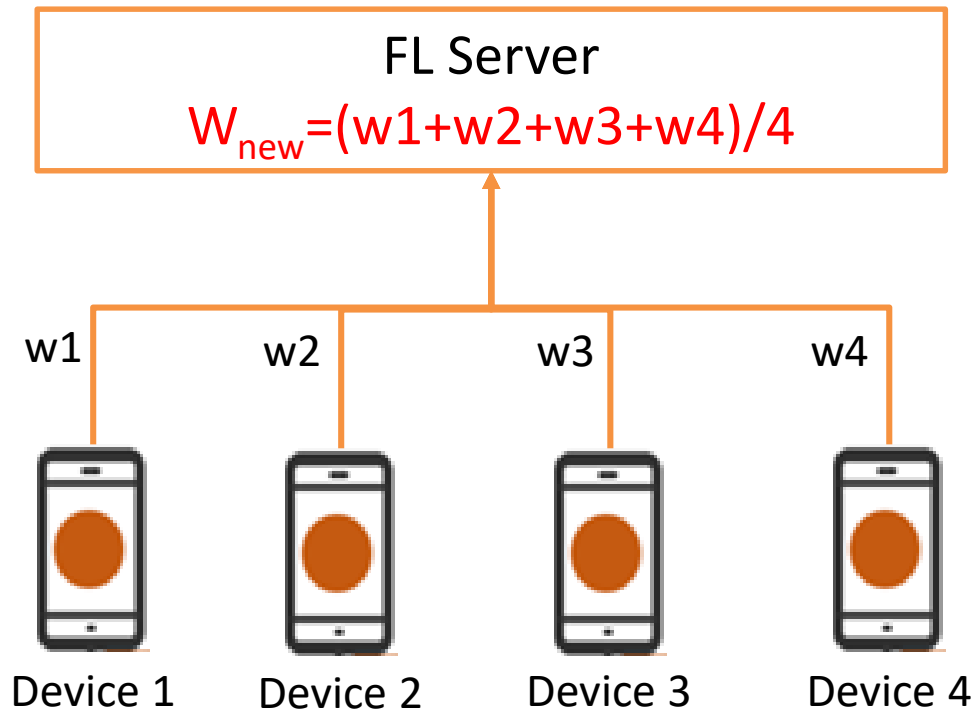


# FedAvg

---

- Devices perform mini-batch training locally, and update their local parameters using gradient descent
- Devices send parameters to server
- Server averages these parameters to obtain a new model
- Server sends the new model back to devices
- Less communication than FedSGD

# FedAvg, $C=1$ , $E=1$ , $B=\infty$



- We set  **$C=1$** , meaning 100% of the devices participate in FedAvg
- **$E=1$** , meaning the local SGD epoch=1
- **$B=\infty$** , meaning all local data are used for training. Setting it to a smaller means we have mini-batch training locally.

Under this setting, FedAvg = FedSGD

# FedAvg

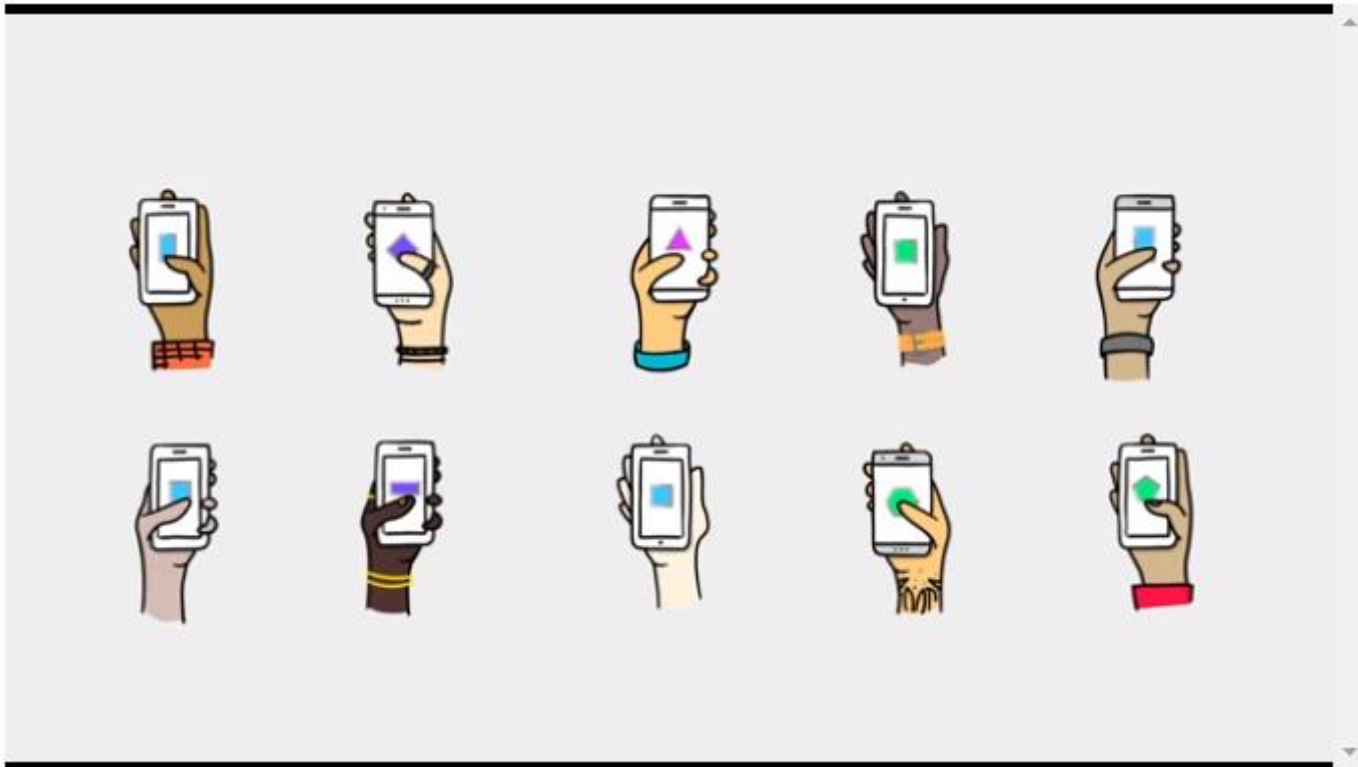
---

- You can increase **E** and reduce **B** to make more use of local device computing power to train the model and reduce communication overhead.
- FedAvg provides you with more flexibility to adjust local computing power utilization and communication overhead during FL model training compared to FedSGD.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agüera y Arcas. [Communication-Efficient Learning of Deep Networks from Decentralized Data](#). *CoRR*, arXiv:1602.05629, 2016.

# Federated Learning (Google)

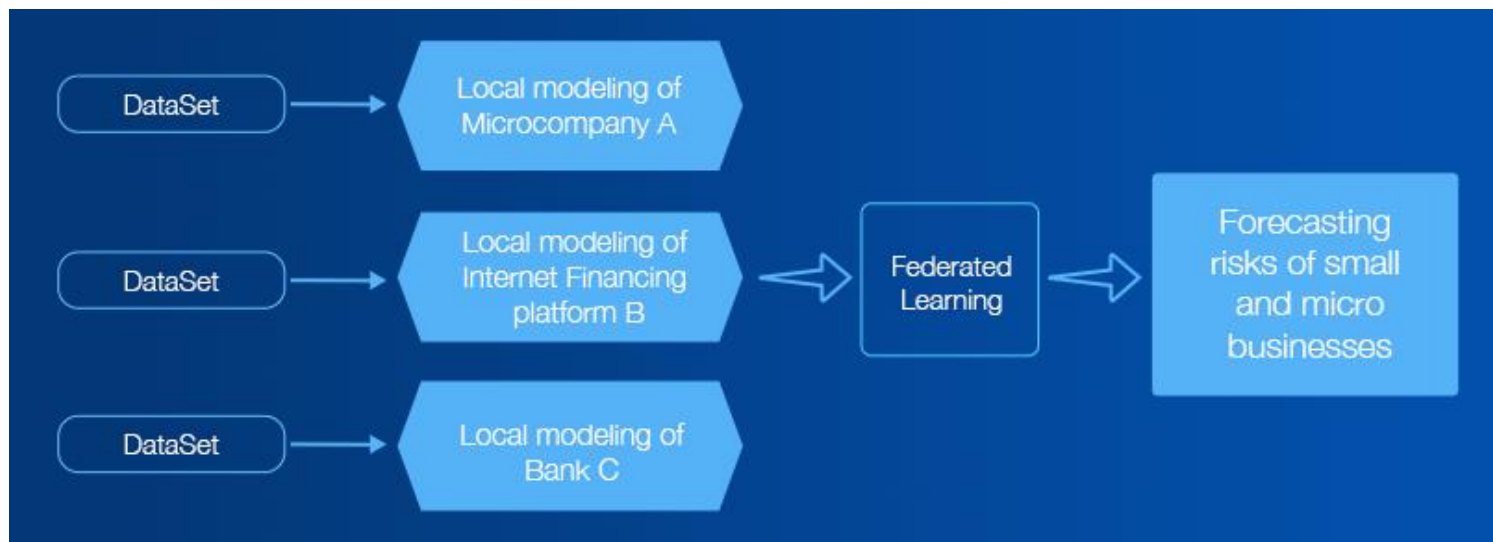
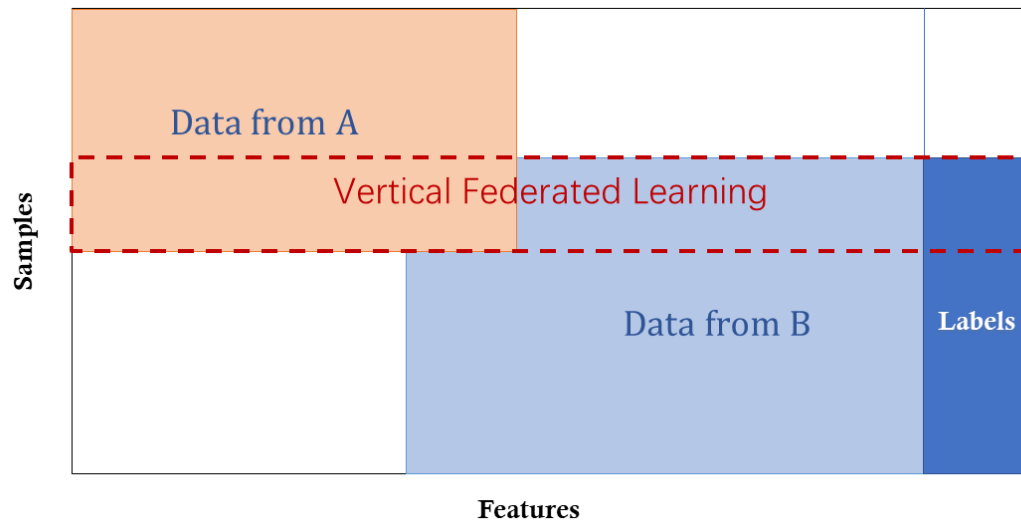
---



**Video Demo:** <https://youtu.be/gbRJPa9d-VU>



# Vertical Federated Learning (VFL)



# Vertical Federated Learning (VFL)

---

- VFL assumes that datasets from different participants **share the same sample ID space but may not share the same feature space**
- VFL assumes that label information is held by one participant
- VFL is less well explored at the moment

Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T. & Yu, H. (2019) *Federated Learning*. Morgan & Claypool Publishers, San Rafael, CA, USA, p. 207.

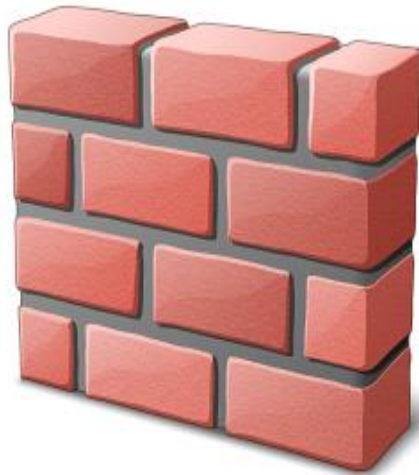
# A Practical Scenario for VFL

---



E-Commerce  
Company

**X1**



Bank

**(X2, Y)**

# Practical Scenarios for VFL

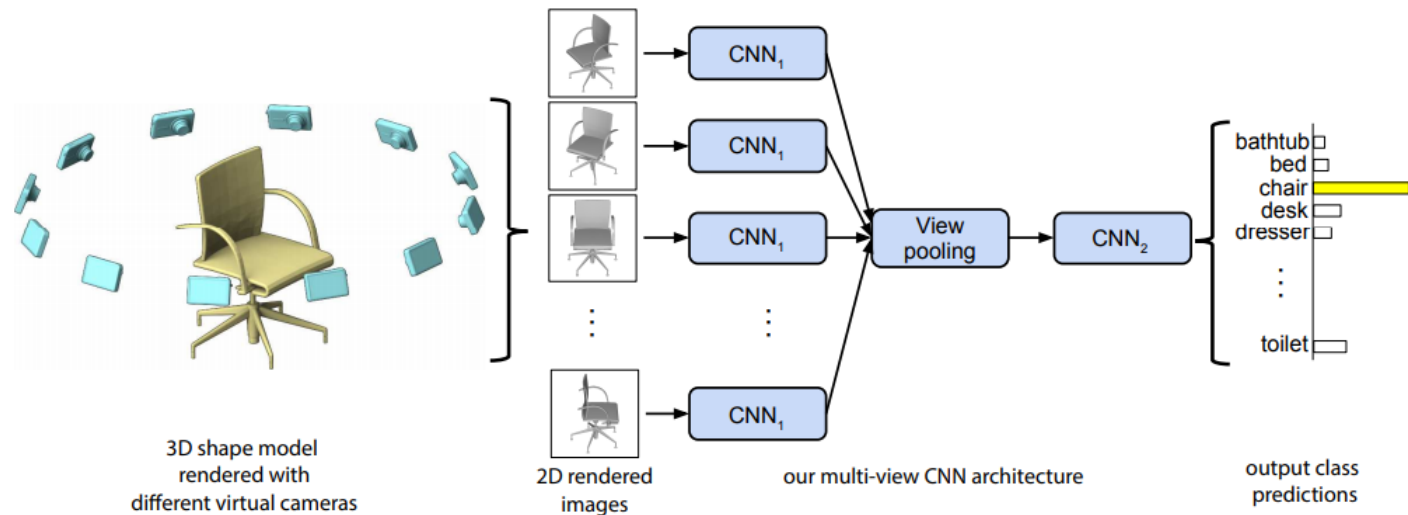
---

An example of VFL in practice:

- An e-commerce company and a bank that both serve users from the same city can train a model to recommend personalized loans for users based on their online shopping behaviors through VFL.
- In this case, only the bank holds label information for the intended VFL task.
- Due to the fact that both the e-commerce company and the bank are located in the same city, it is reasonable to assume that the data from both entities have large overlap of users.
- The challenge is to train a model collaboratively without exchanging the data and label information.

# From Multi-View Learning to VFL

- A Brief Introduction of Multi-View Learning (MVL)
  - MVL approaches aim to learn one function to model each view and jointly optimize all the functions to improve performance



An illustration of MVL in a 3D shape recognition research work. In this work, a 3D shape is rendered from multiple different views and finally a compact shape descriptor is obtained.

# From Multi-View Learning to VFL

---

- Similarity and Difference between MVL and VFL
  - **Similarity**
    - Both MVL and VFL assume that data from different views/nodes share the same sample ID space but different feature space.
    - Both MVL and VFL assume that data from different views/nodes share the same label space
  - **Difference**
    - MVL requires data from different views to interact
    - VFL forbids data exchange due to privacy concerns

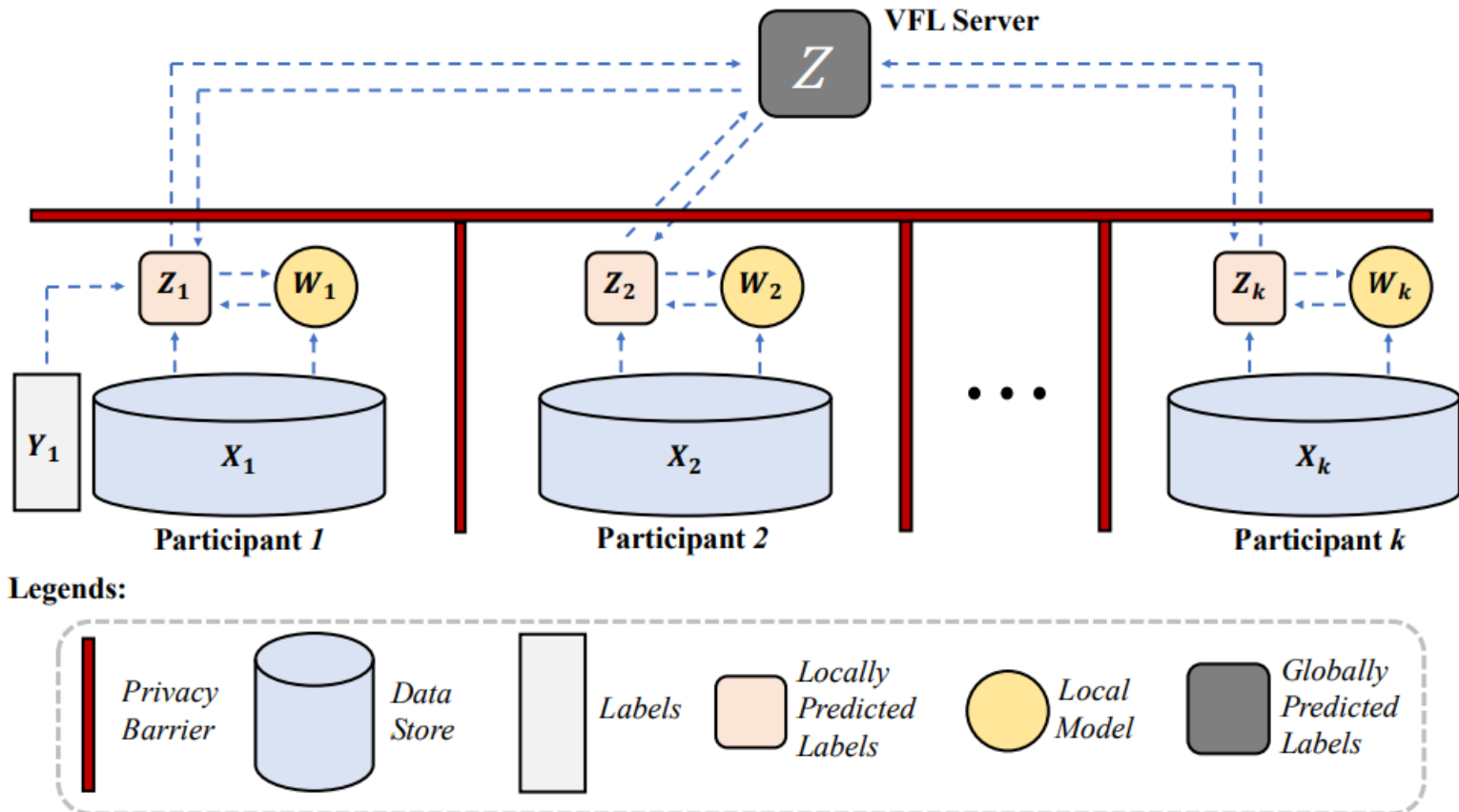
# From Multi-View Learning to VFL

---

- Advantage of MVL compared with existing VFL methods
  - Existing MVL approaches can handle multi-view-multi-class problems, instead of the binary-participant-binary-class problems that most existing VFL methods tackle with
- Goal
  - To build a VFL framework based on the methodology of MVL with data privacy preserved

Chang Xu, Dacheng Tao & Chao Xu. A survey on multi-view learning.  
*CoRR* arXiv:1304.5634, 2013

# From Multi-View Learning to VFL



By design, only the **locally predicted labels  $z_i$**  cross the privacy barriers to reach the VFL Server. The global FL model can be trained without raw data, labels or local models leaving their owners' machine.



# Feature Importance Evaluation

- Two advantages of feature importance evaluation:
  - It can quantify the contribution of different features from each participant to the FL model.
  - By discarding redundant and harmful features in initial training periods, the communication, computation and storage costs of a VFL system can be reduced for subsequent training under incremental learning settings.

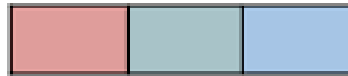
All Features



Feature Selection



Final Features



An illustration of feature selection

Siwei Feng & Han Yu, "Multi-Participant Multi-Class Vertical Federated Learning," *CoRR*, arXiv:2001.11154, 2020.

# Video Explanation

---



[https://www.youtube.com/watch?v=NPGf\\_OJrzOg&feature=youtu.be](https://www.youtube.com/watch?v=NPGf_OJrzOg&feature=youtu.be)

# Hands-on Practice

<https://colab.research.google.com/drive/1dRG3yNAIDar3tll4VOkmoU-aLslhUS8d>

Chrome File Edit View History Bookmarks People Tab Window Help

Federated Learning on MNIST using a CNN.ipynb

File Edit View Insert Runtime Tools Help Last saved at 9:23 PM

Table of contents

- Federated Learning on MNIST using a CNN with PyTorch & PySyft
  - Imports and model specifications
  - Data loading and sending to workers
  - CNN specification
  - Define the train and test functions
  - Launch the training !
- Section

### Federated Learning on MNIST using a CNN with PyTorch & PySyft

#### Context

Federated learning is a machine learning technique that trains an algorithm across multiple decentralized edge devices or servers holding local data samples, without exchanging their data samples.

In this tutorial, we'll use directly PySyft library based on PyTorch library. Only 10 line codes are modified for PySyft to upgrade a traditional CNN classification into a federated mode.

We use Google Colaboratory to execute code which is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

related reference:

- PyTorch (<https://github.com/pytorch/examples/blob/master/mnist/main.py>)
- PySyft (<https://github.com/OpenMined/PySyft/>)
- Colaboratory (<https://colab.research.google.com/>)

Ok, let's get started!

Colaboratory support for importing a library that's not in Colaboratory by default. In this tutorial, we just need install syft package by pip.

```
[ ] 1 ! pip install syft
```

```
[ ] Requirement already satisfied: syft in /usr/local/lib/python3.6/dist-packages (0.2.2a1)
```

**Video Guide:** [https://www.youtube.com/watch?v=NPGf\\_OJrzOg&feature=youtu.be](https://www.youtube.com/watch?v=NPGf_OJrzOg&feature=youtu.be)



NANYANG  
TECHNOLOGICAL  
UNIVERSITY  
SINGAPORE

# Privacy Preservation

Yu Han

[han.yu@ntu.edu.sg](mailto:han.yu@ntu.edu.sg)

*Nanyang Assistant Professor  
School of Computer Science and Engineering  
Nanyang Technological University*

