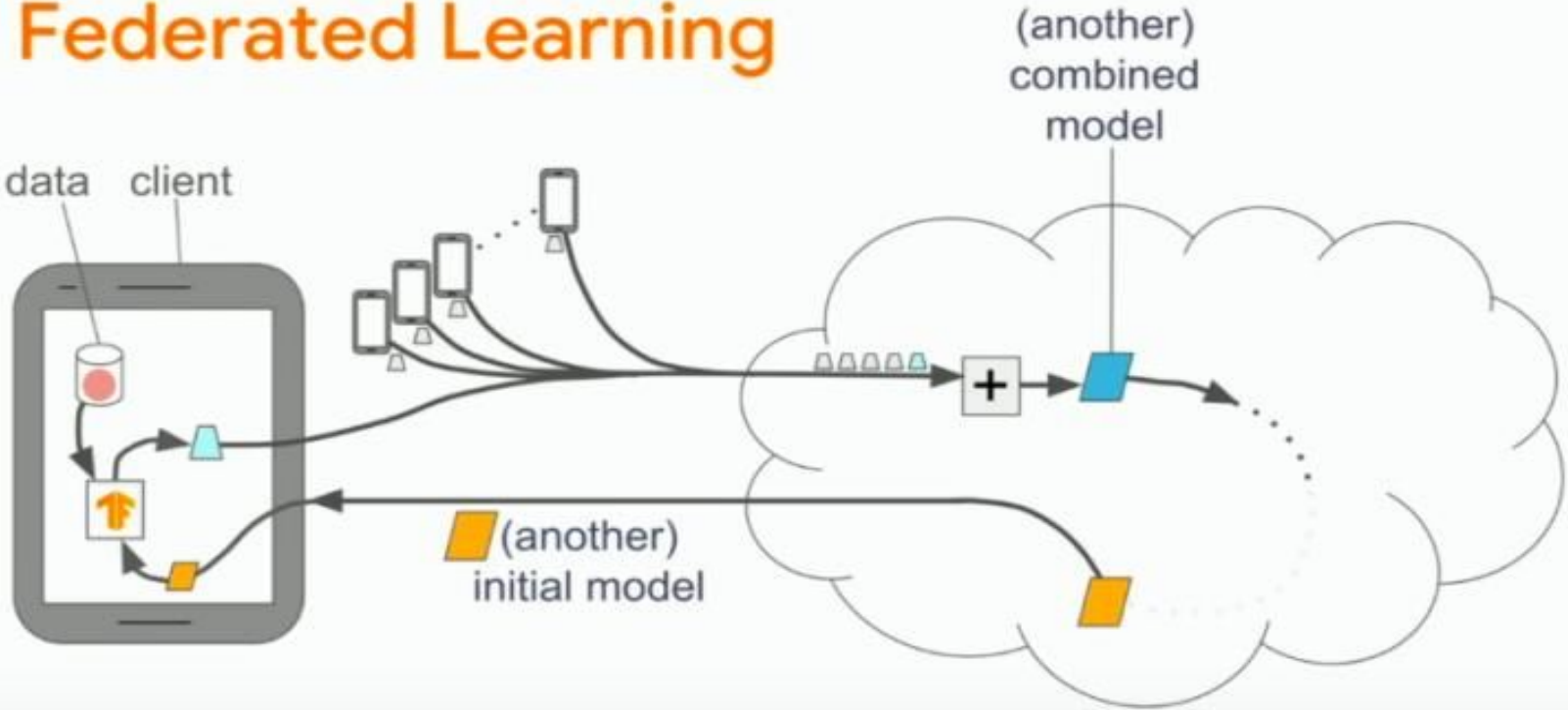
FEDERATED MACHINE LEARNING AND APPLICATIONS

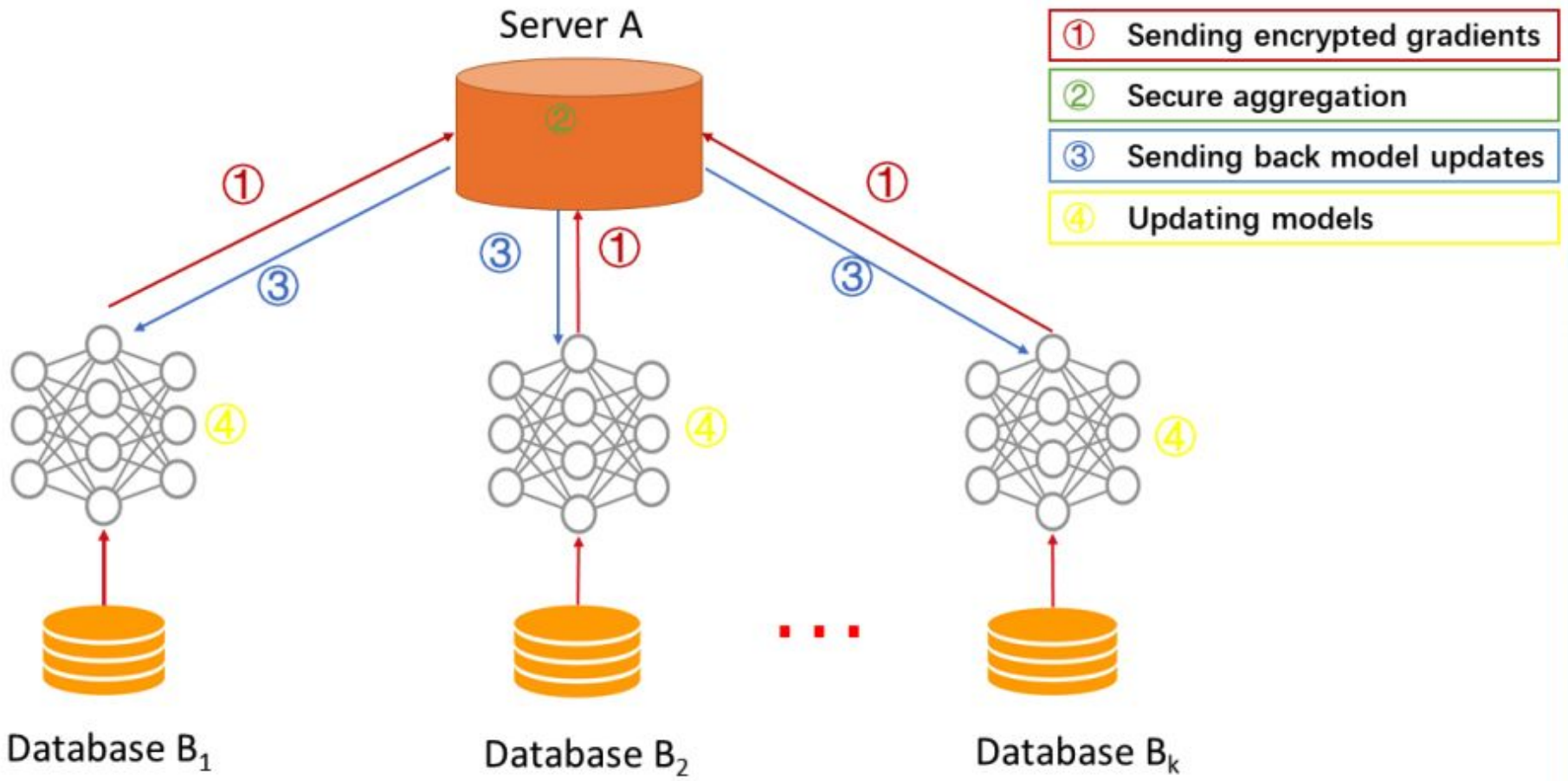
—

Dhruv Singhal

What is Federated Machine Learning?

Federated Learning

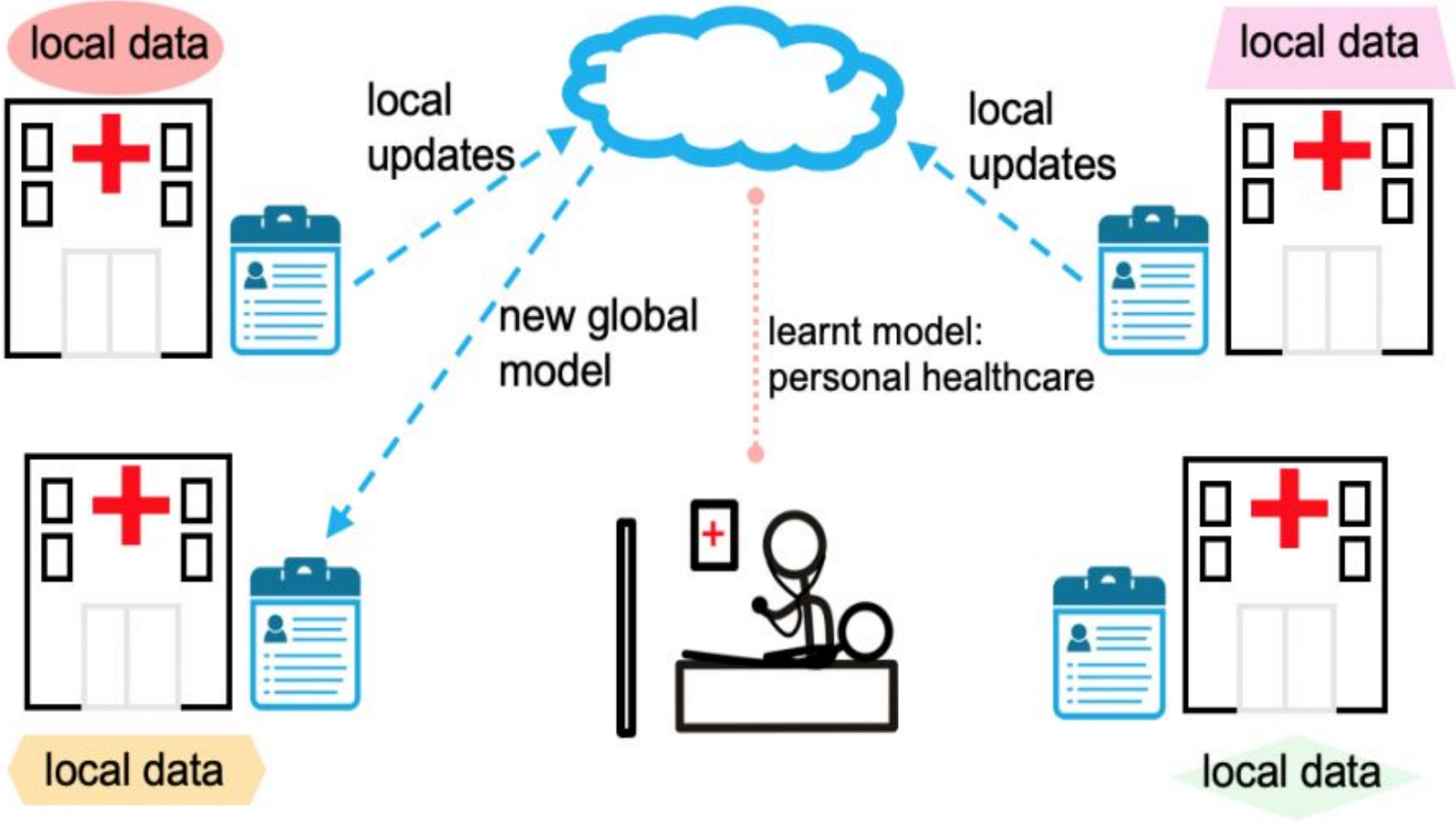


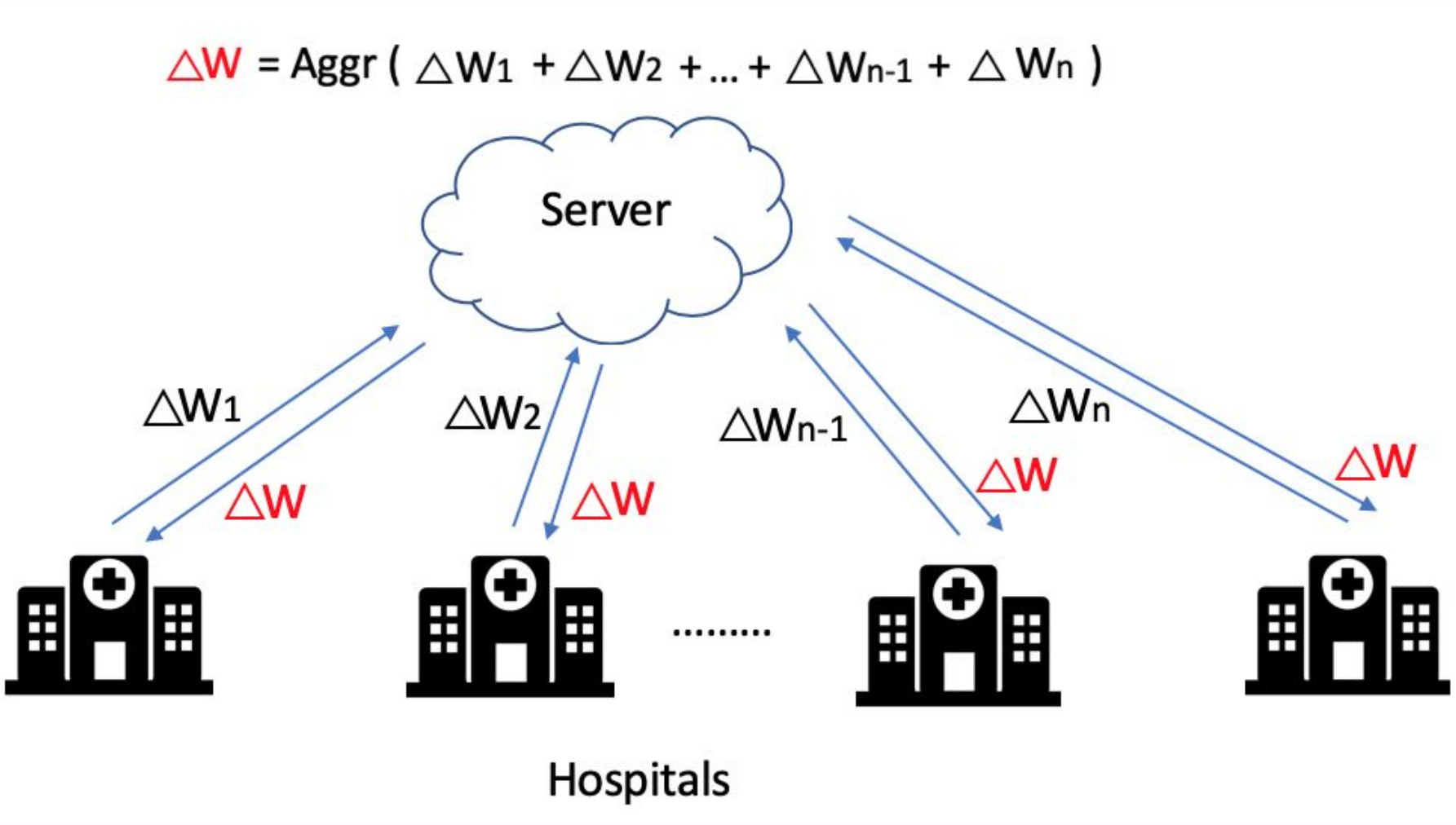

Server A
② Secure aggregation
① Sending encrypted gradients
③ Sending back model updates
④ Updating models
①
③
①
③
①
③
④
④
④
Database B_1
Database B_2
Database B_k

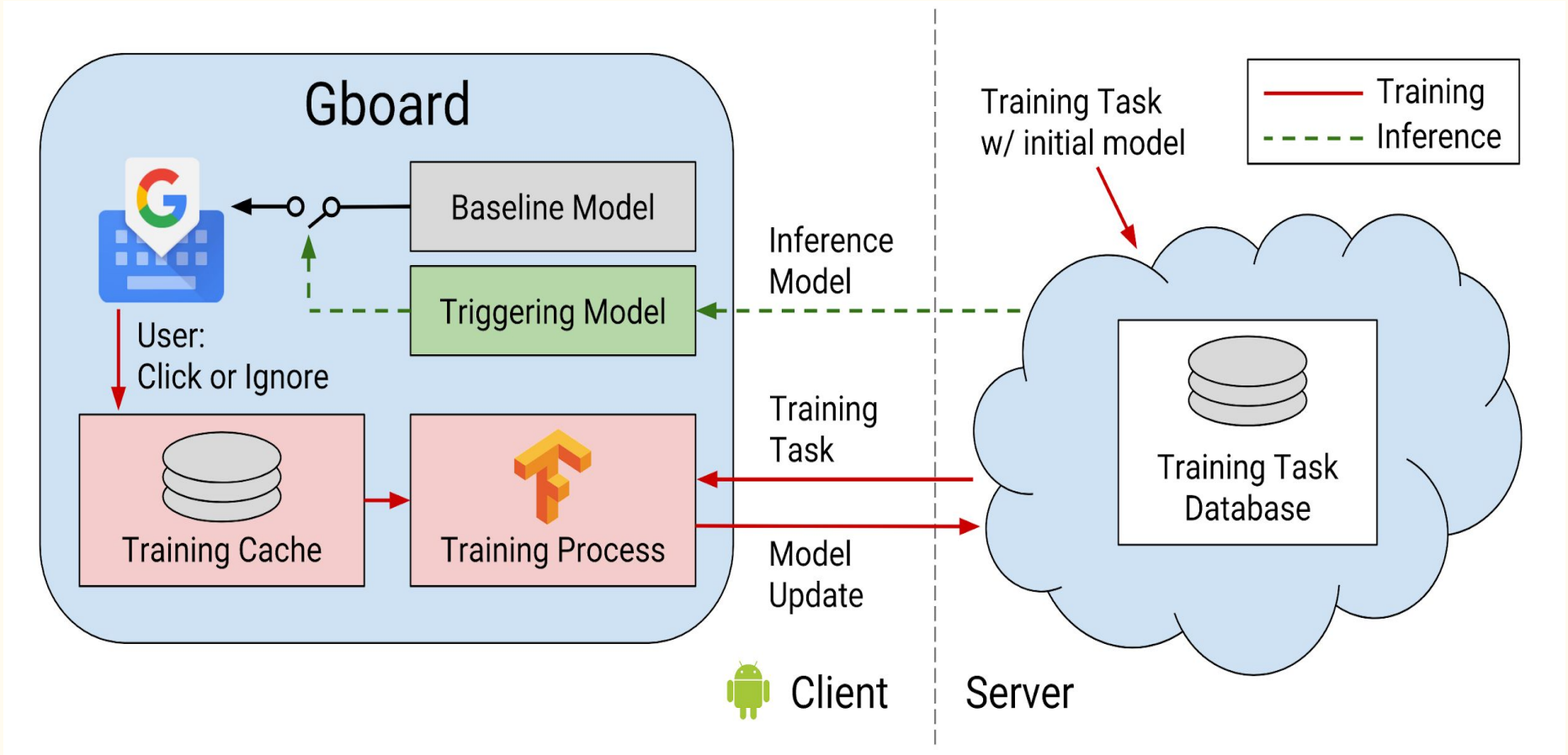
Need of FedML?

- Emerged as privacy preserving tool in the field of ML
- Enhances Data Privacy and Security
- Dataset can remain in the hands of workers(or the user)
- As not based on training dataset on centralized server
- Distributed Computing Power

Applications


local data
local updates
local updates
local data
new global model
learnt model: personal healthcare
local data
local data


ΔW = Aggr (ΔW1 + ΔW2 + … + ΔWn-1 + ΔWn)
Server
ΔW1
ΔW
ΔW2
ΔW
ΔWn-1
ΔW
ΔWn
ΔW
.....
Hospitals


Gboard
Baseline Model
Triggering Model
User:
Click or Ignore
Training Cache
Training Process
Inference
Model
Training
Task
Model
Update
Training Task
w/ initial model
Training
Inference
Training Task
Database
Client
Server

Alternatives

- Homomorphic Encryption
- Secure Multiparty Computation
- Trusted Execution Environment

Test Project

Implementing Handwriting Recognition using FedML

1. CNN on MNIST database for handwriting recognition

2. PySyft framework for federated machine learning

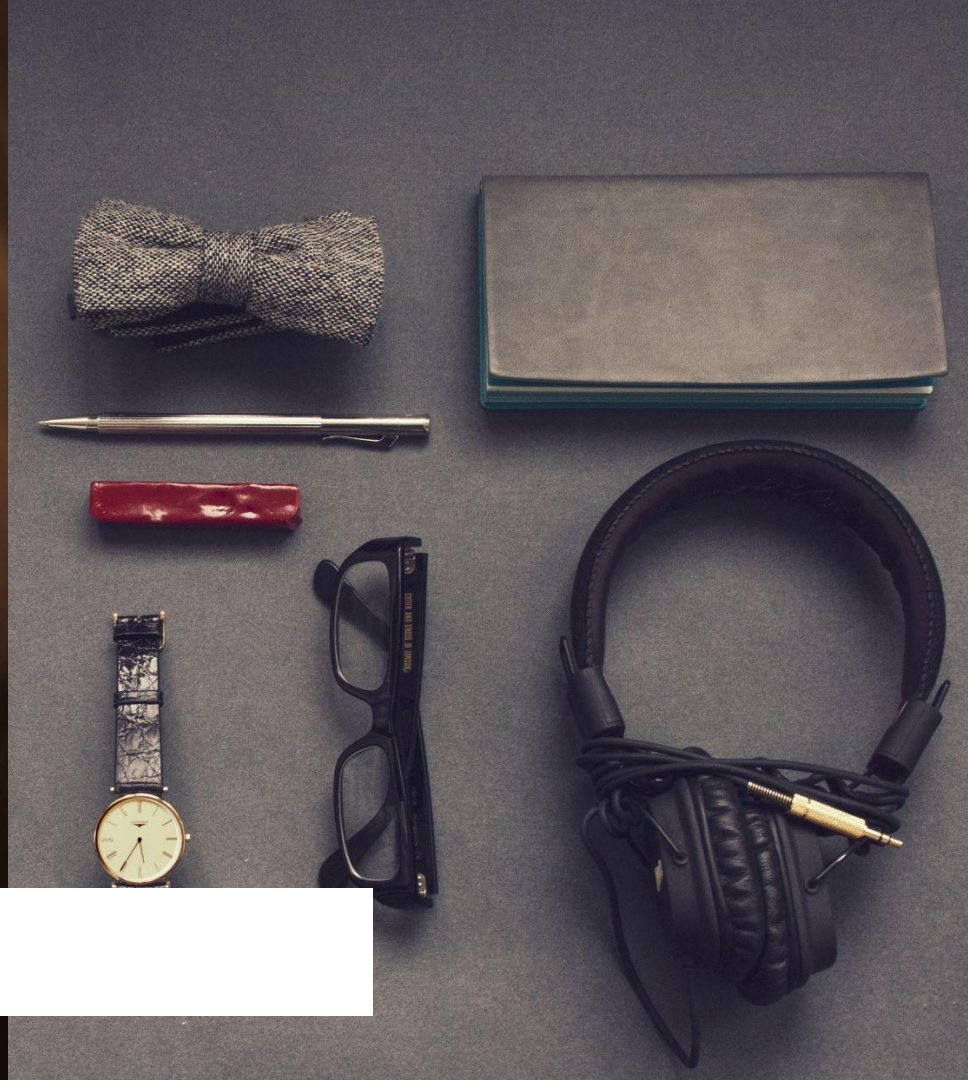
3. We will use virtual workers: these workers behave exactly like normal remote workers except that they live in the same Python program rather than on remote servers

4. We first load the data and transform the training Dataset into a Federated Dataset and send them to remote workers

5. Then we train normal CNN for handwriting recognition for each worker individually

6. And then test the returned model locally only

- **MNSIT dataset**
- **PySyft**
- **VirtualWorker**



THANK YOU