Frontier Research of Federated Learning

马汝辉 副教授 博导
计算机科学与工程系
上海交通大学

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录

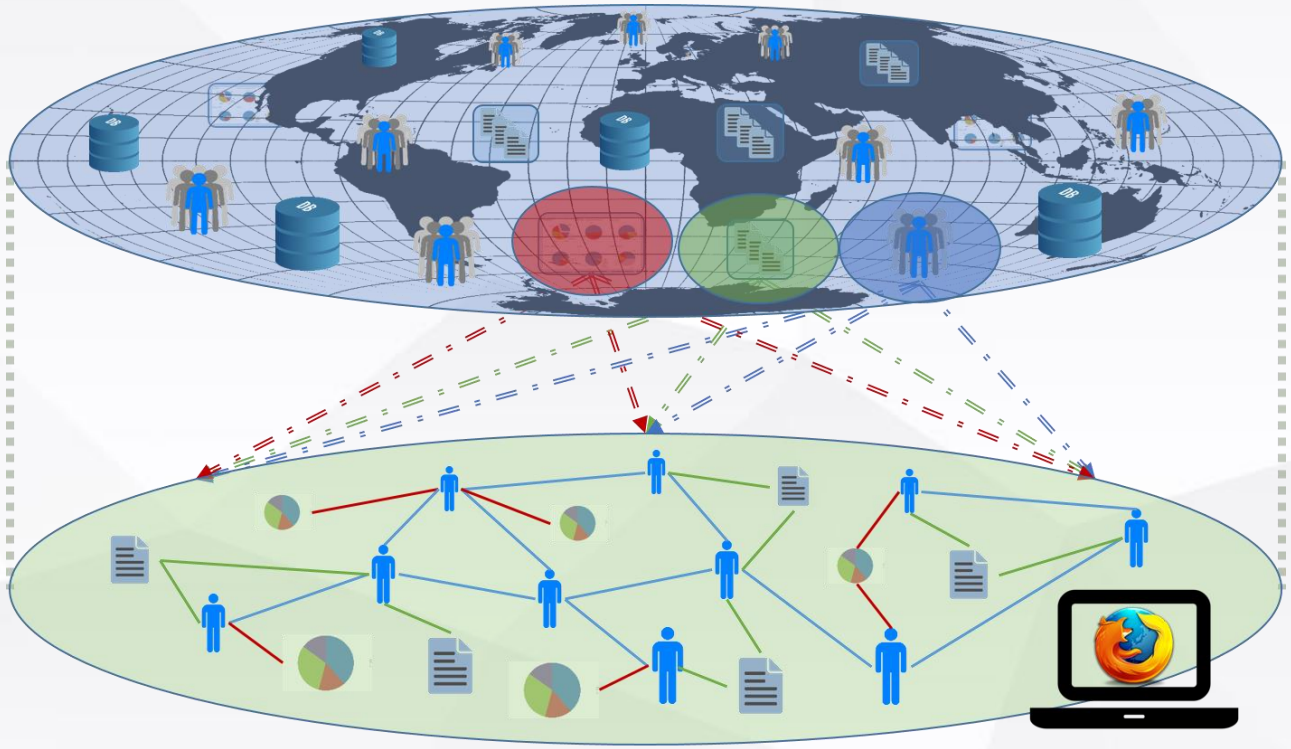
01

Challenges in FL

- Expensive communication
- Systems heterogeneity
- Statistical heterogeneity
- Privacy concerns

Challenges in FL

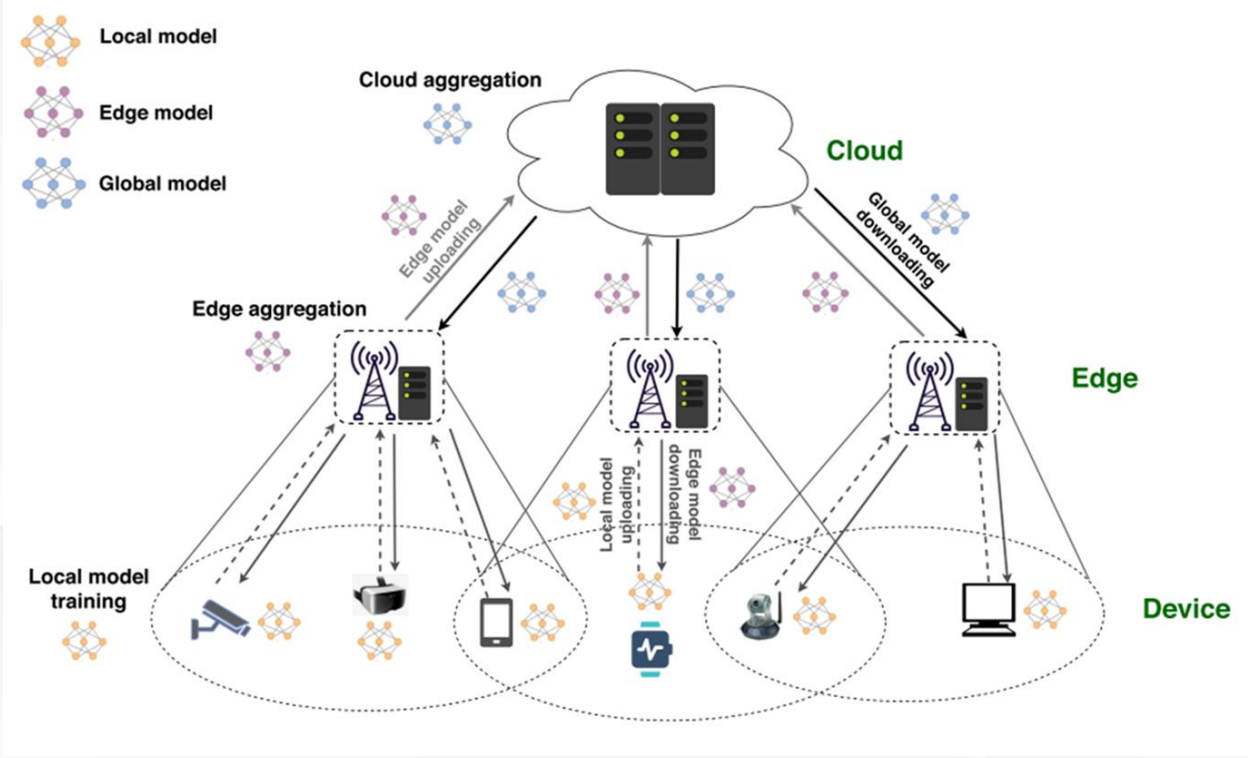
- **Expensive Communication.**
  - federated networks are potentially comprised of a massive number of devices, e.g., millions of smart phones, and communication in the network can be slower than local computation by many orders of magnitude.



Challenges in FL

Systems Heterogeneity.

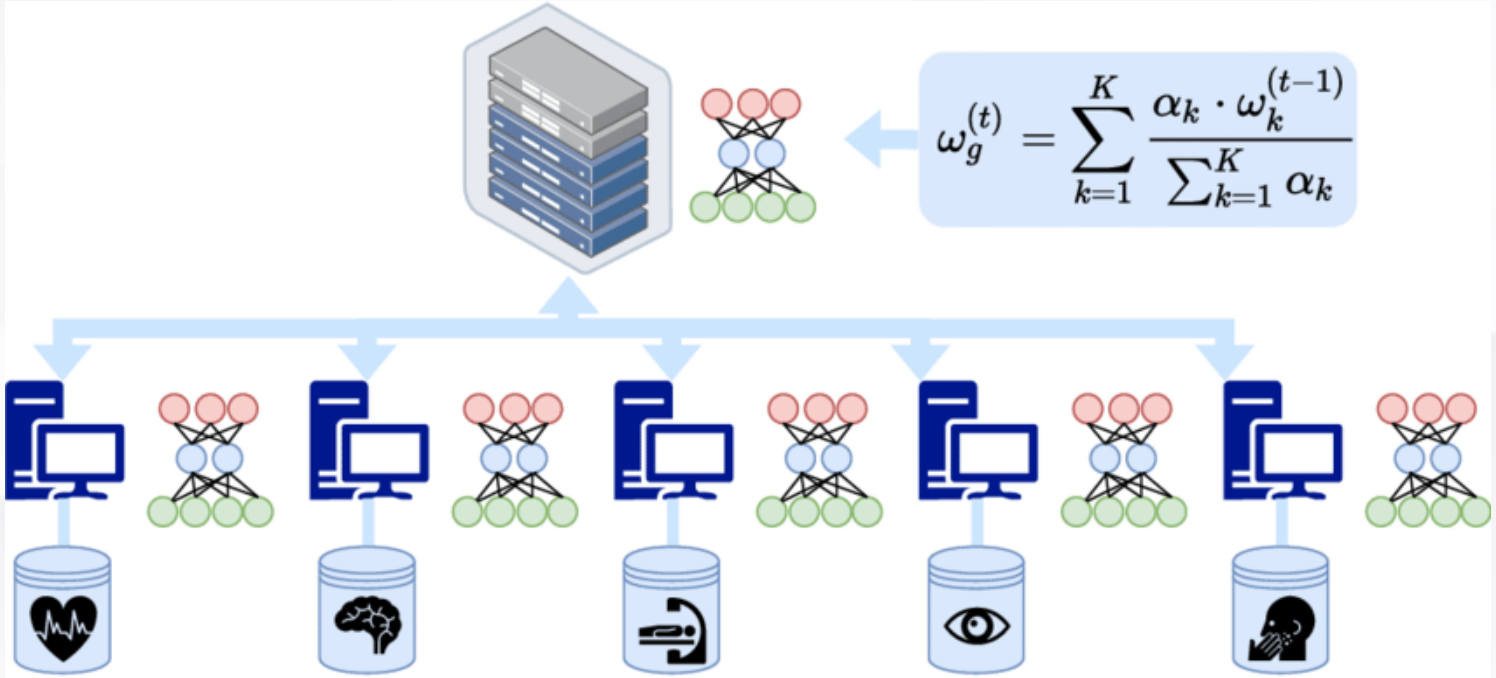
- The storage, computational, and communication capabilities of each device in federated networks may differ due to variability in hardware (CPU, memory), network connectivity (3G, 4G, 5G, wifi), and power (battery level).
- Each device may also be unreliable.



Challenges in FL

Statistical Heterogeneity.

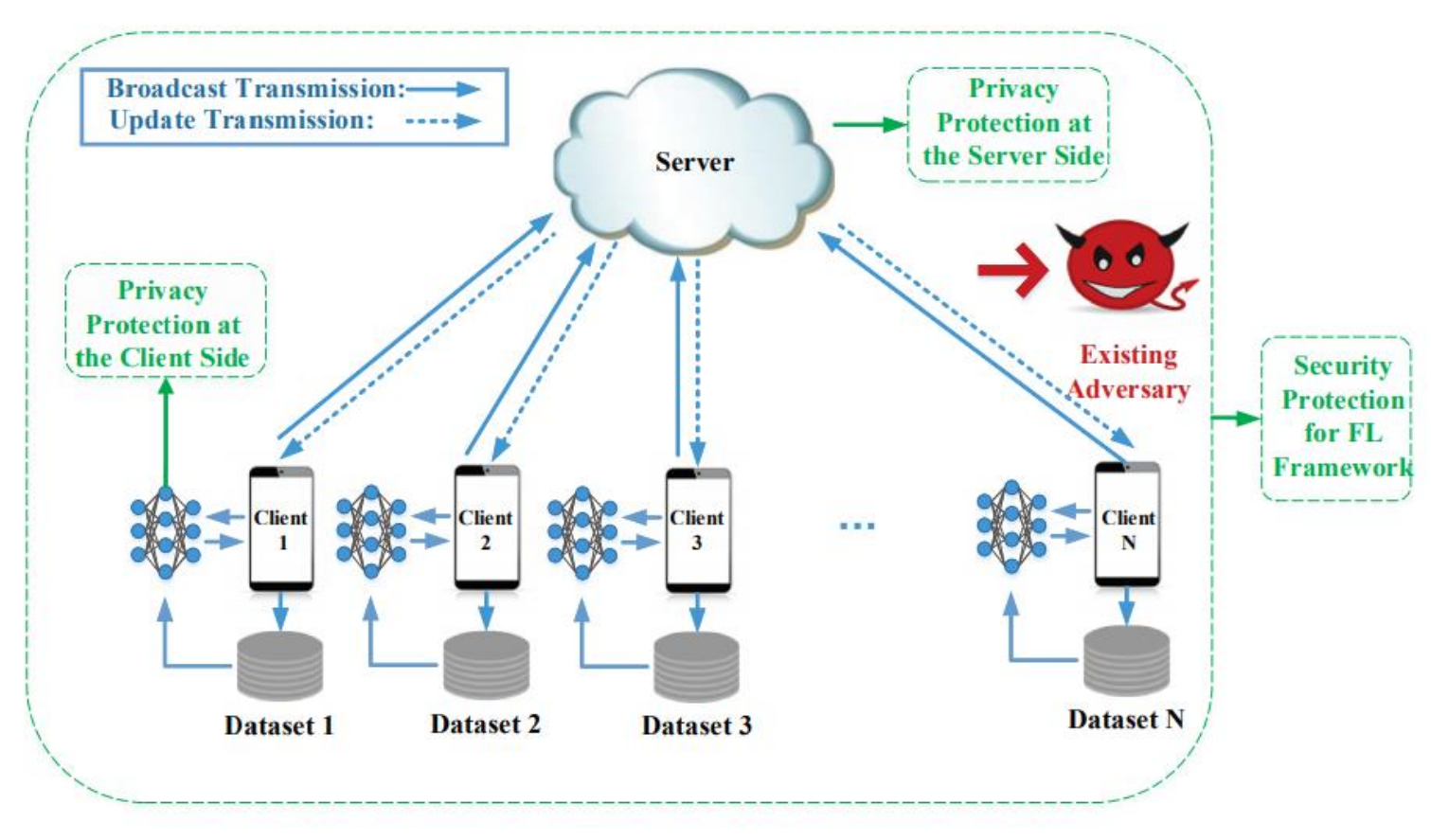
- Devices frequently generate and collect data in a non-identically distributed manner across the network, e.g., mobile phone users have varied use of language in the context of a next word prediction task.
- Increases the likelihood of stragglers.



Challenges in FL

Privacy Concerns.

- communicating model updates throughout the training process can nonetheless reveal sensitive information



02

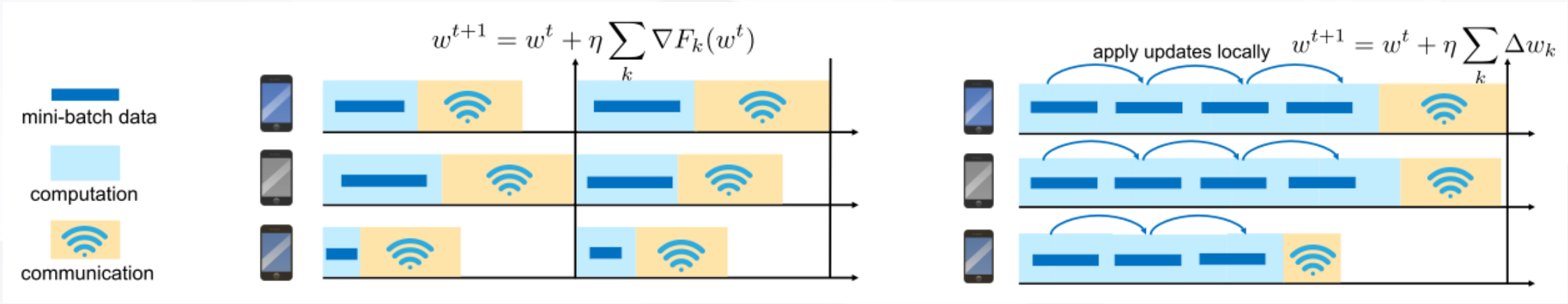
Communication-efficiency

- Local updating
- Compression schemes
- Decentralized training

Communication-efficiency

Local updating

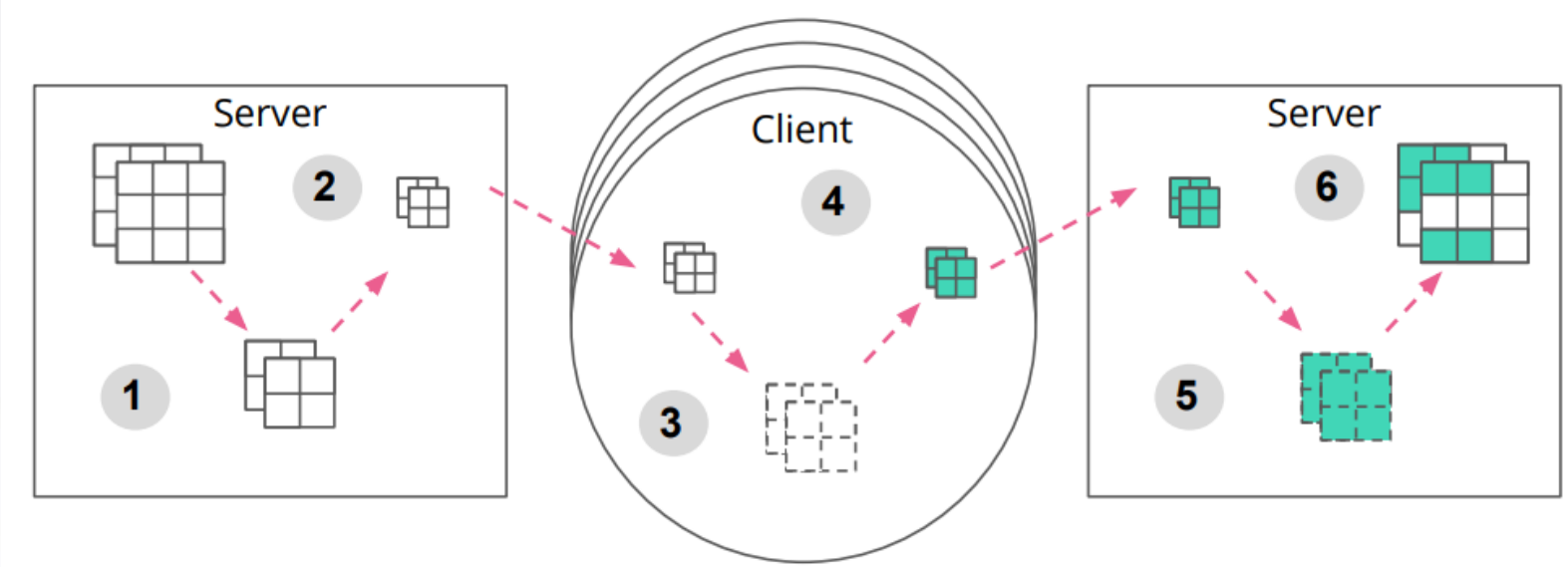
- Mini-batch optimization methods have been shown to have limited flexibility to adapt to communication-computation trade-offs that would maximally leverage distributed data processing.
- Allow for a variable number of local updates to be applied on each machine in parallel at each communication round
- For convex objectives, distributed local-updating primal-dual methods have emerged as a popular way to tackle such a problem.



Communication-efficiency

Compression schemes

- Use lossy compression and dropout to reduce server-to-device communication.

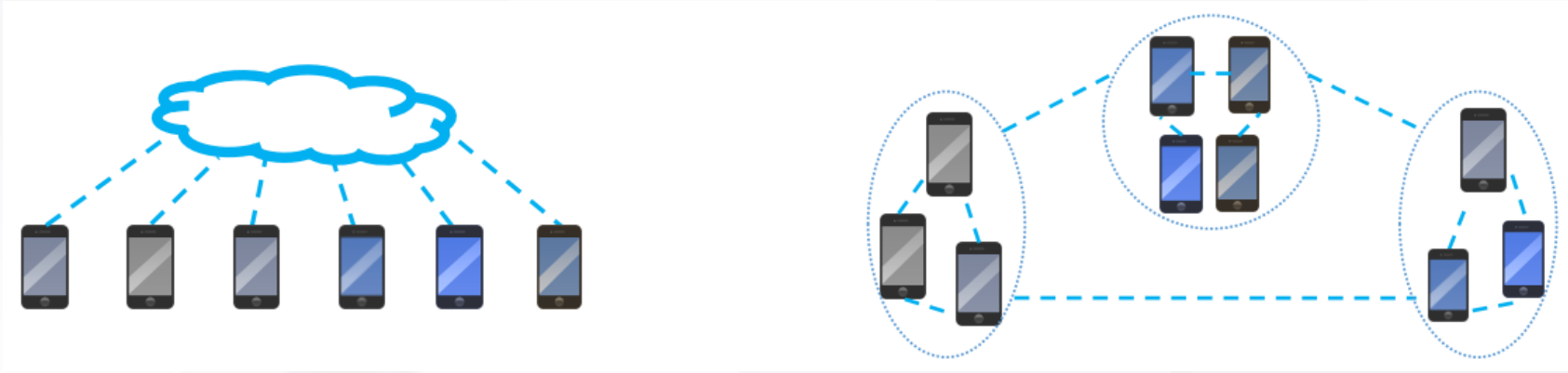


(1) constructing a sub-model via Federated Dropout, and by (2) lossily compressing the resulting object. This compressed model is then sent to the client, who (3) decompresses and trains it using local data, and (4) compresses the final update. This update is sent back to the server, where it is (5) decompressed and finally, (6) aggregated into the global model

Communication-efficiency

Decentralized Training

- In federated learning, a star network (where a central server is connected to a network of devices) is the predominant communication topology.
- Decentralized algorithms can in theory reduce the high communication cost on the central server.



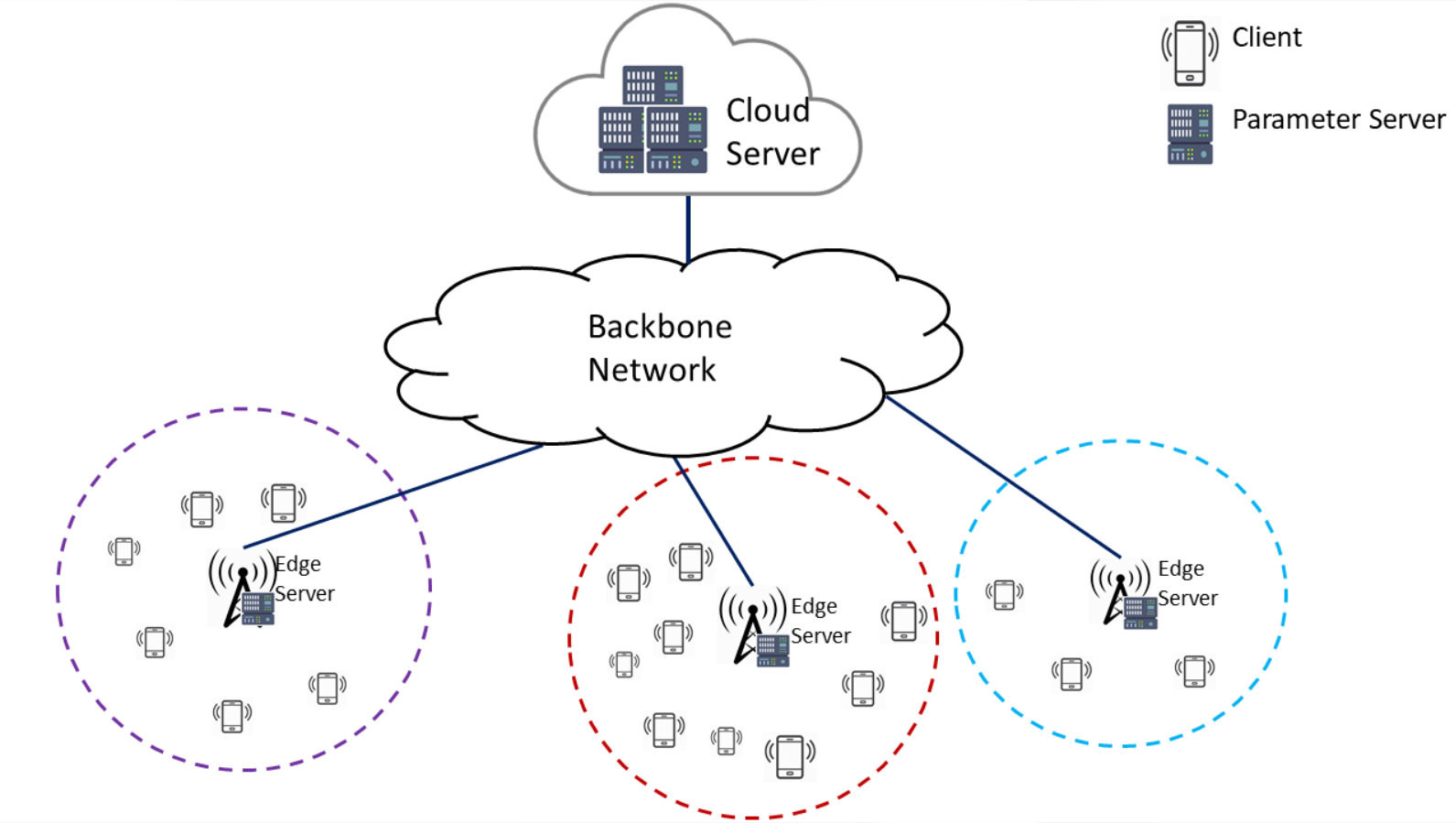
Centralized topology

Decentralized topology

Communication-efficiency

Decentralized Training

- Hierarchical communication patterns.



03

Privacy protection

- Privacy threats/attacks in federated learning (FL)
- Enhance the general privacy-preserving feature of FL
- Associated cost of the privacy-preserving techniques

Privacy protection

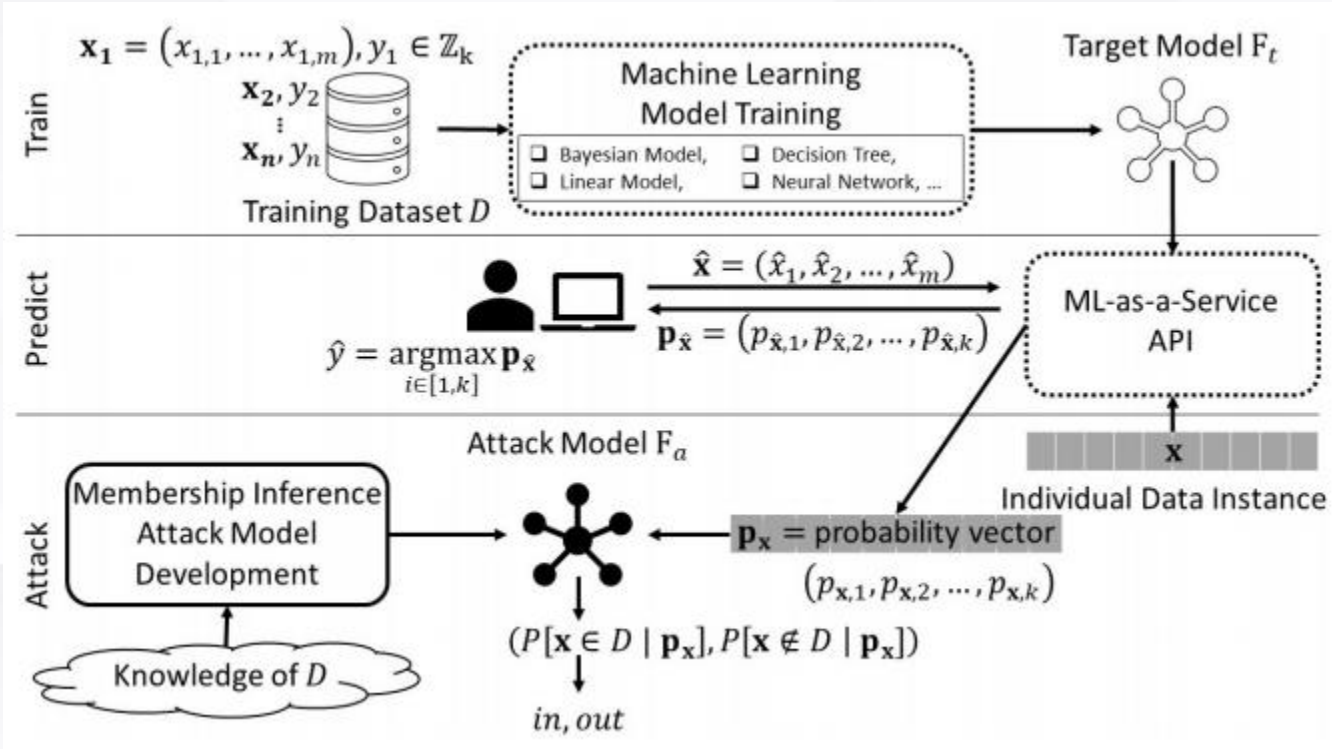
Privacy threats/attacks in FL

- Membership inference attacks
- Unintentional data leakage and reconstruction through inference
- GANs-based inference attacks

Privacy protection

Membership inference attacks

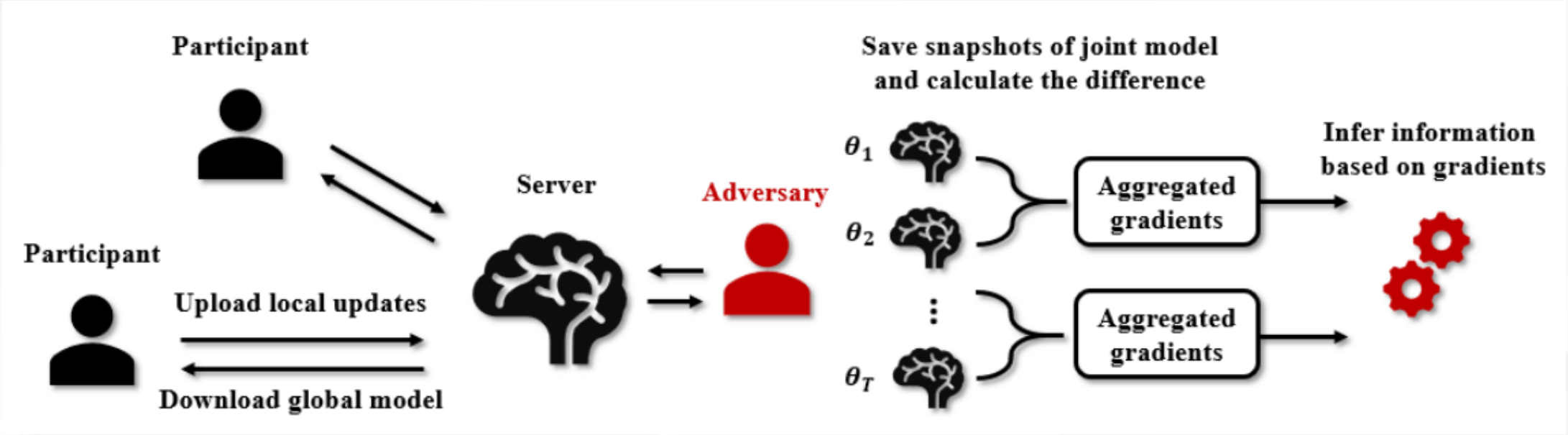
- The neural network is vulnerable to memorize their training data which is prone to passive and active inference attacks.
- The attacker misuses the global model to get information on the training data of the other users.



Privacy protection

Unintentional data leakage and reconstruction

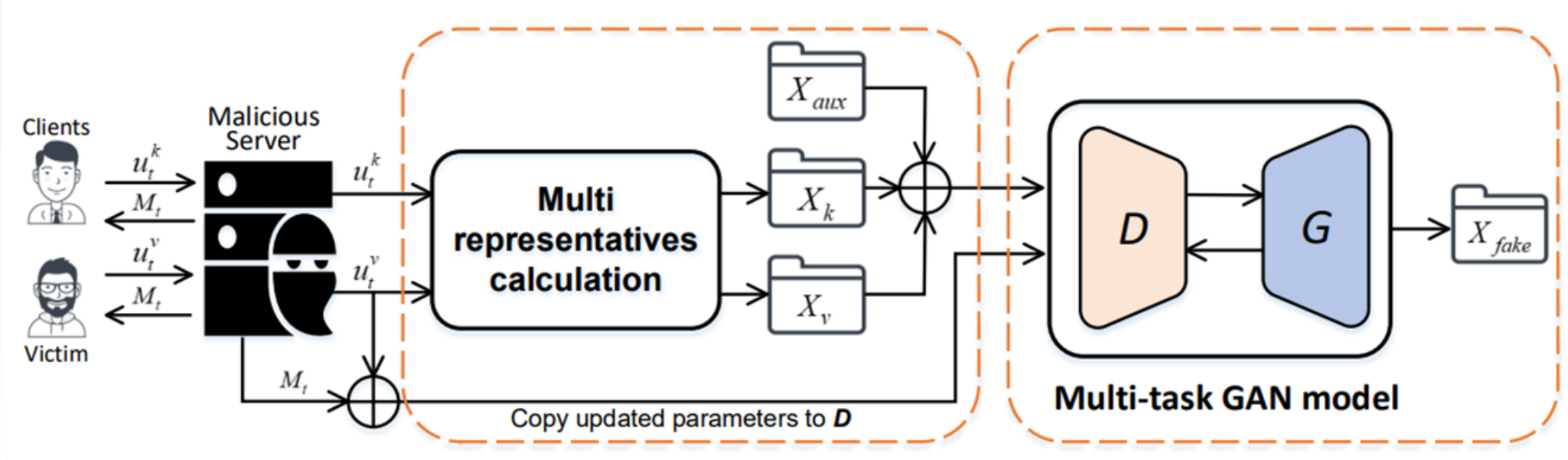
- Is a scenario where updates or gradients from clients leak unintended information at the central server.



Privacy protection

GANs-based inference attacks

- GANs are generative adversarial networks that have gained much popularity in big data domains.
- It is possible to have potential adversaries among FL clients.



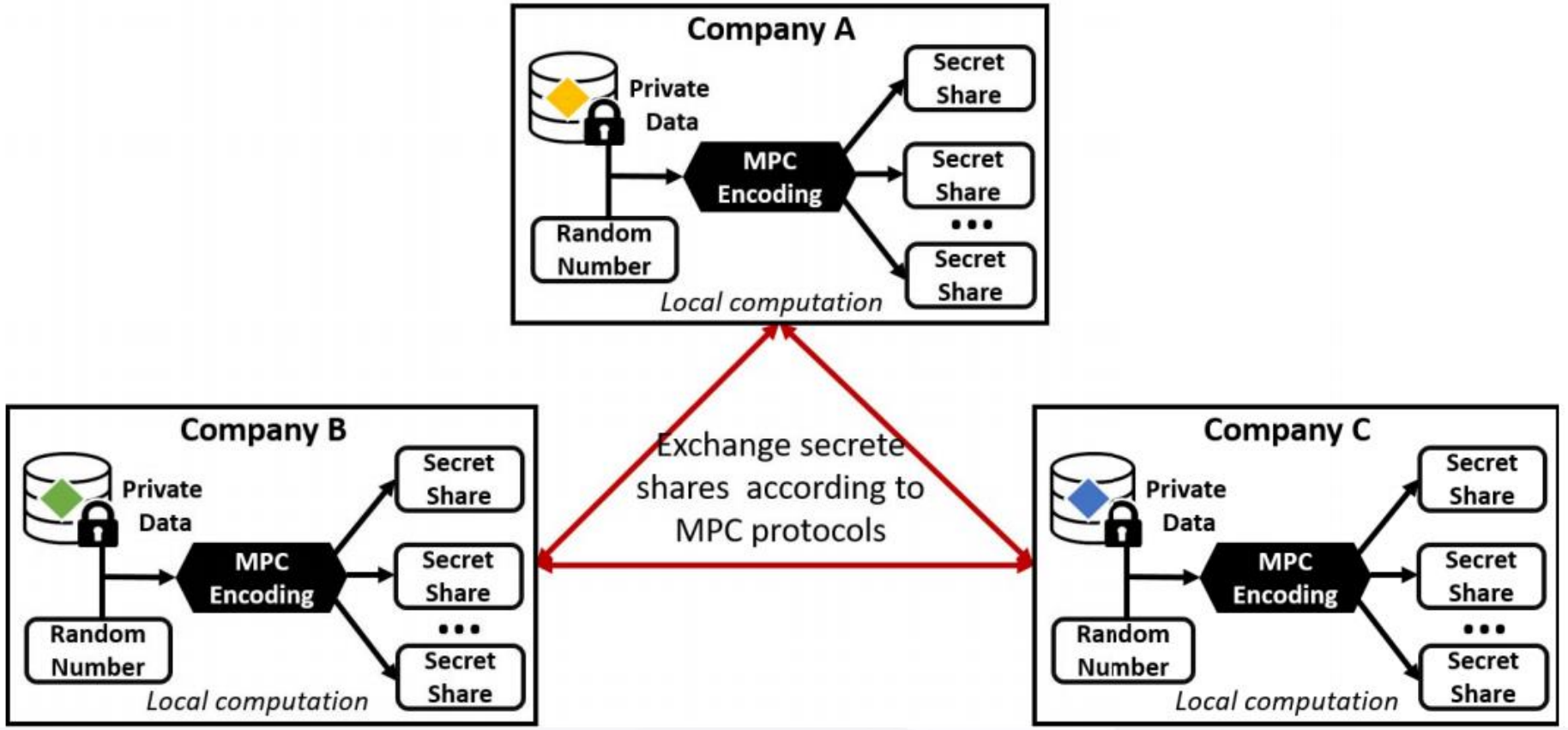
Privacy protection

- **Enhance privacy-preserving in FL**
  - **Secure multi-party computation**
  - **Differential privacy**
  - **VerifyNet**
  - **Adversarial training**

Privacy protection

Secure multi-party computation

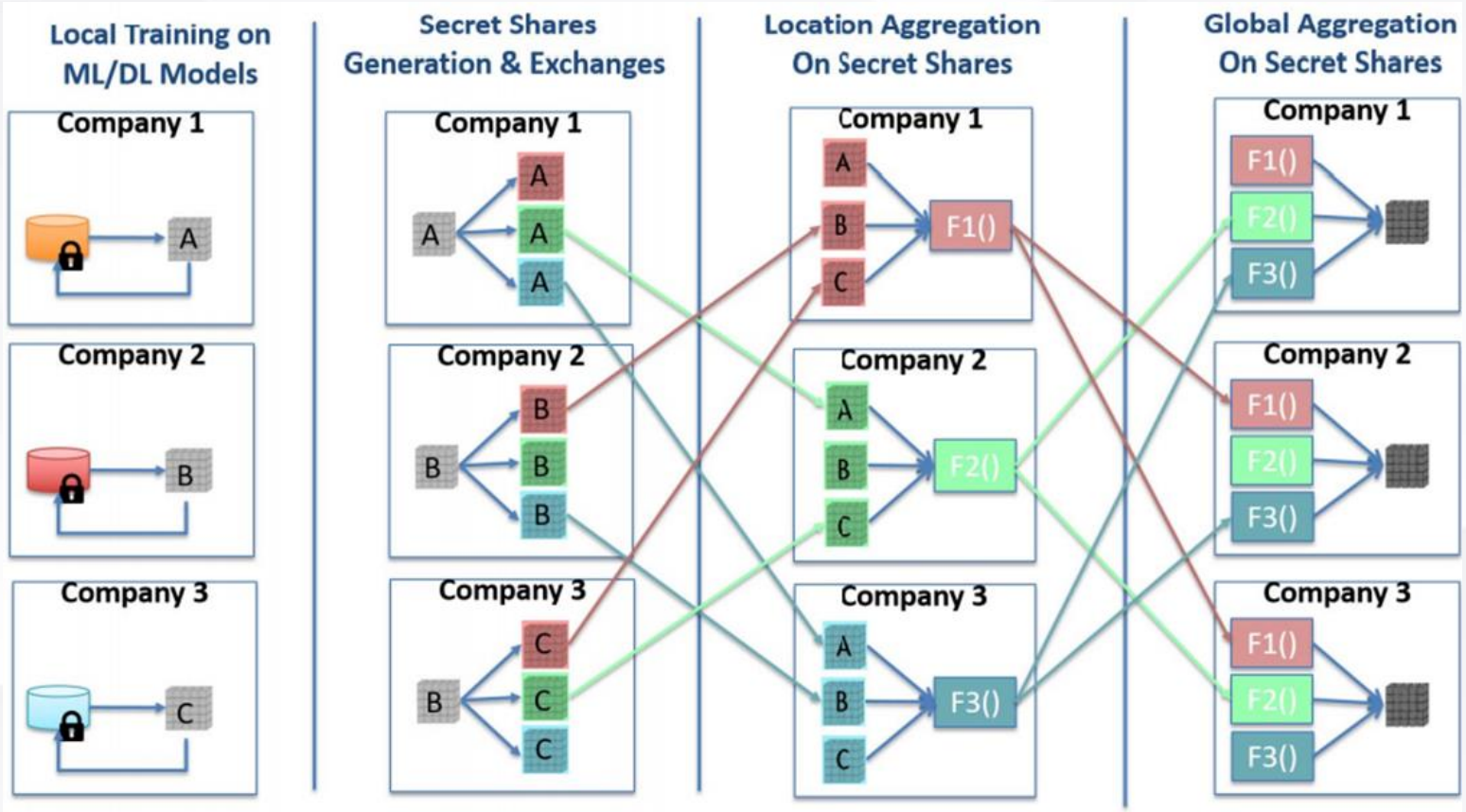
- Secure the inputs of multi-participant while they jointly compute a model or a function.



Privacy protection

Secure multi-party computation

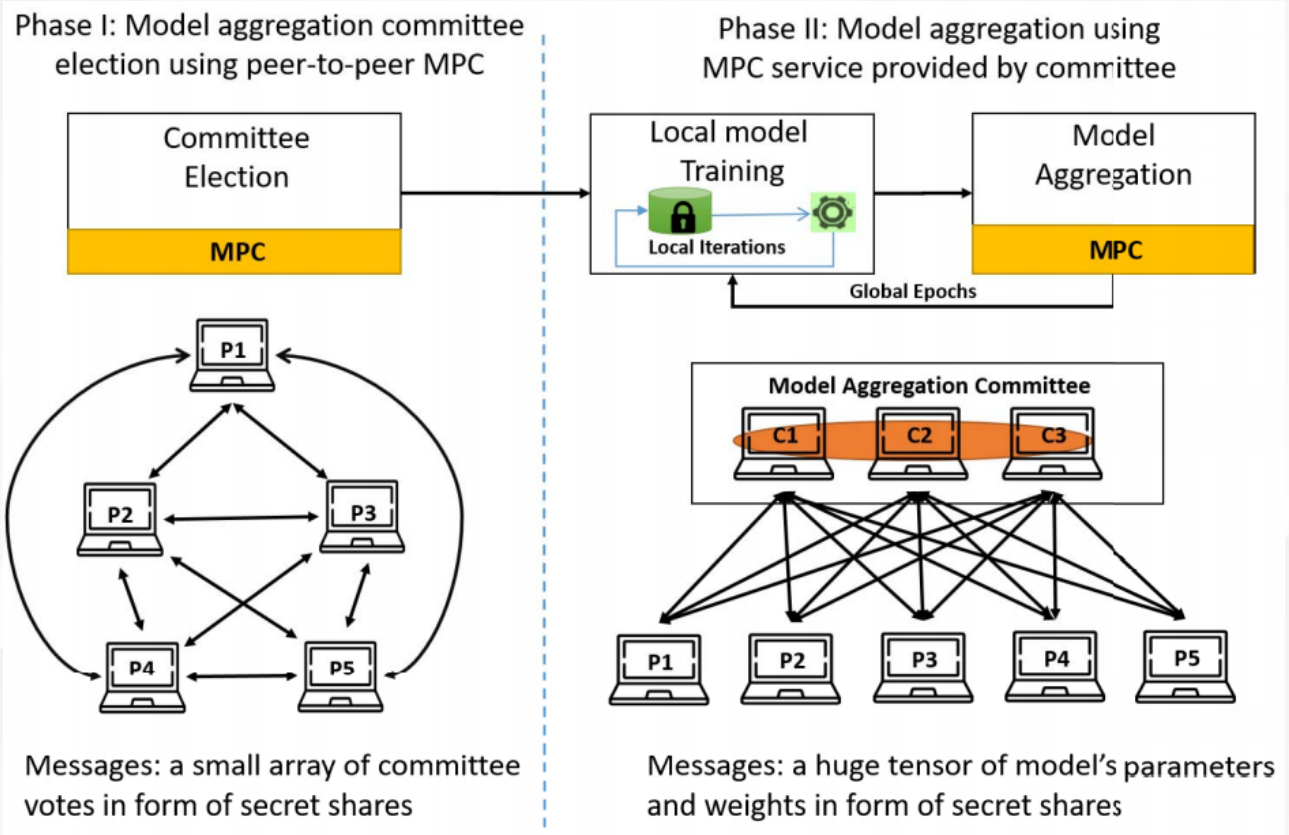
- Secure the inputs of multi-participant while they jointly compute a model or a function.



Privacy protection

Secure multi-party computation

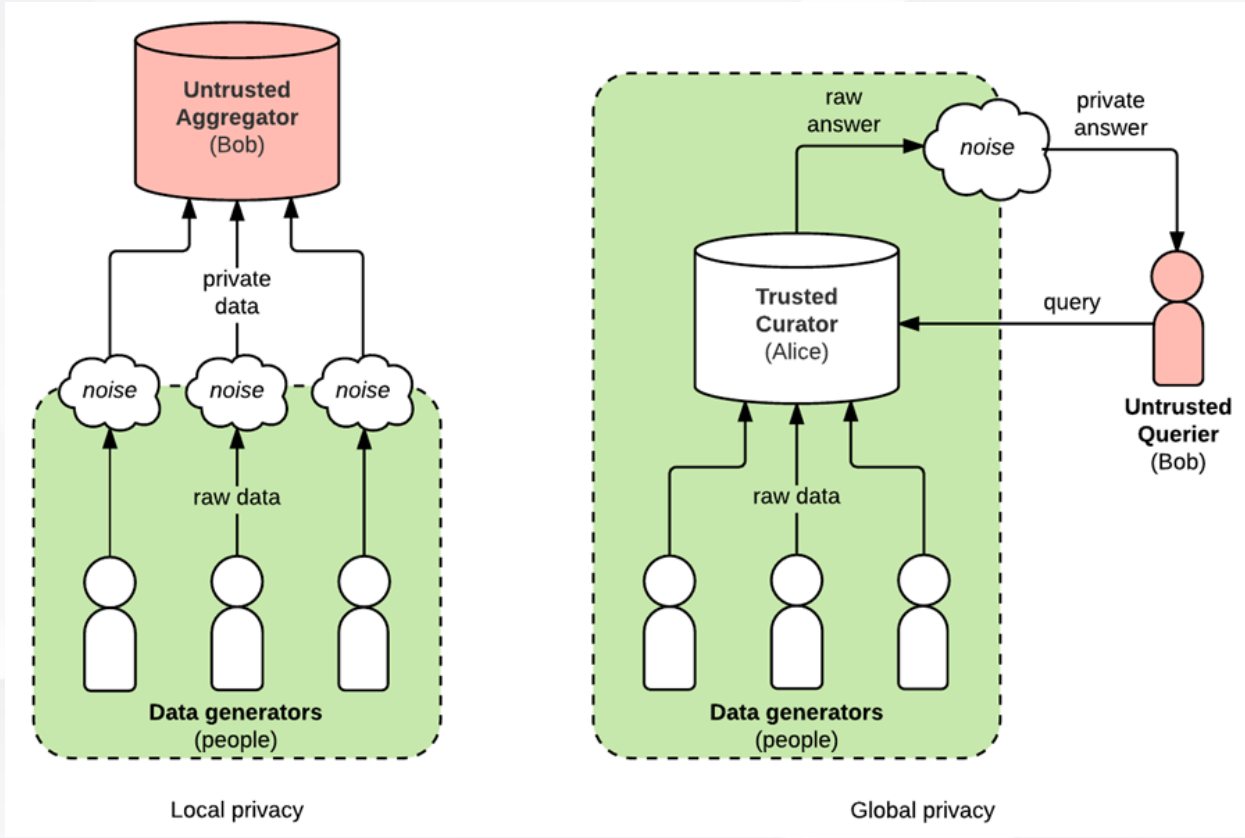
- In FL, the computing efficiency is increased immensely since it only needs to encrypt the parameters instead of the large volume of data inputs.



Privacy protection

Differential Privacy

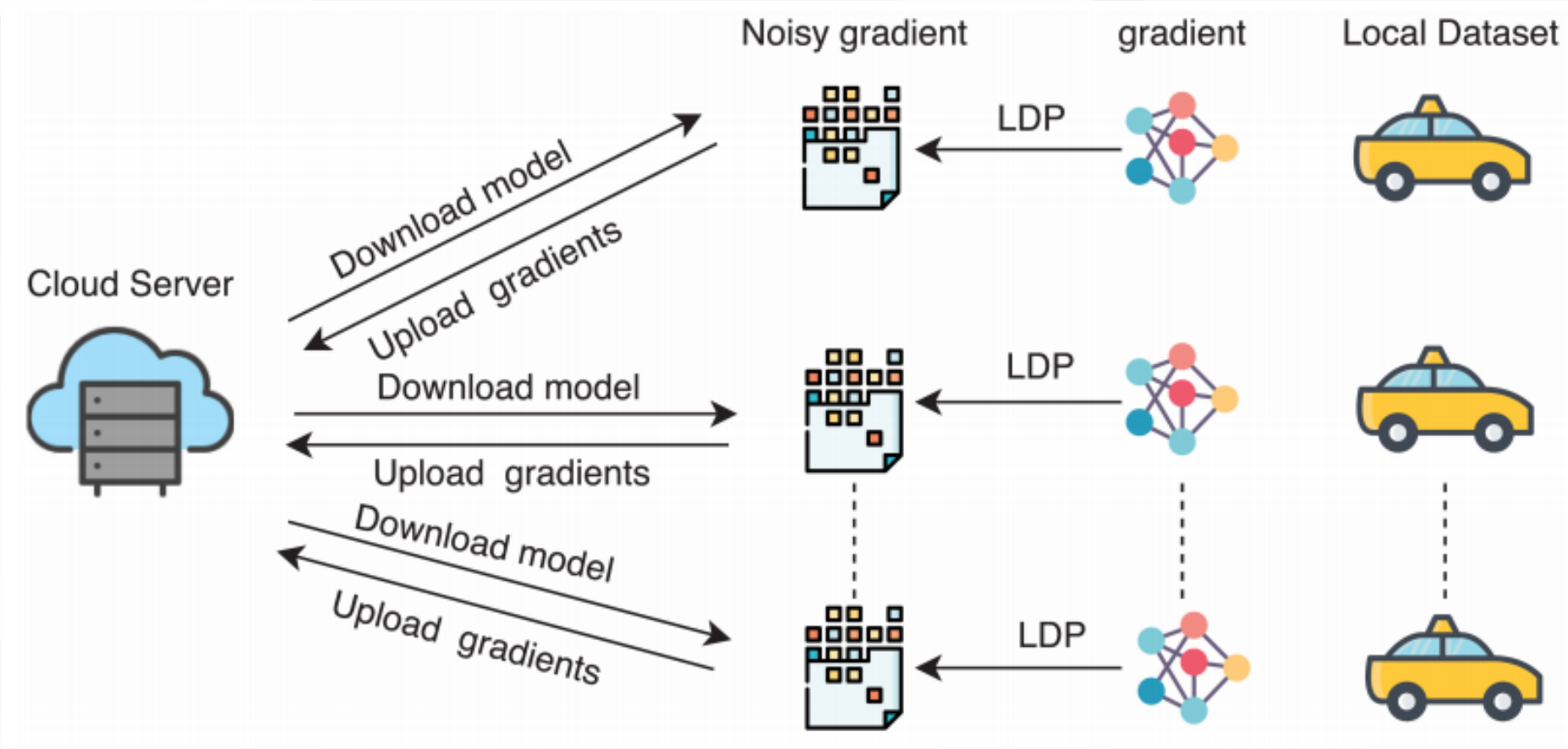
- Add noise to personal sensitive attributes



Privacy protection

Differential Privacy

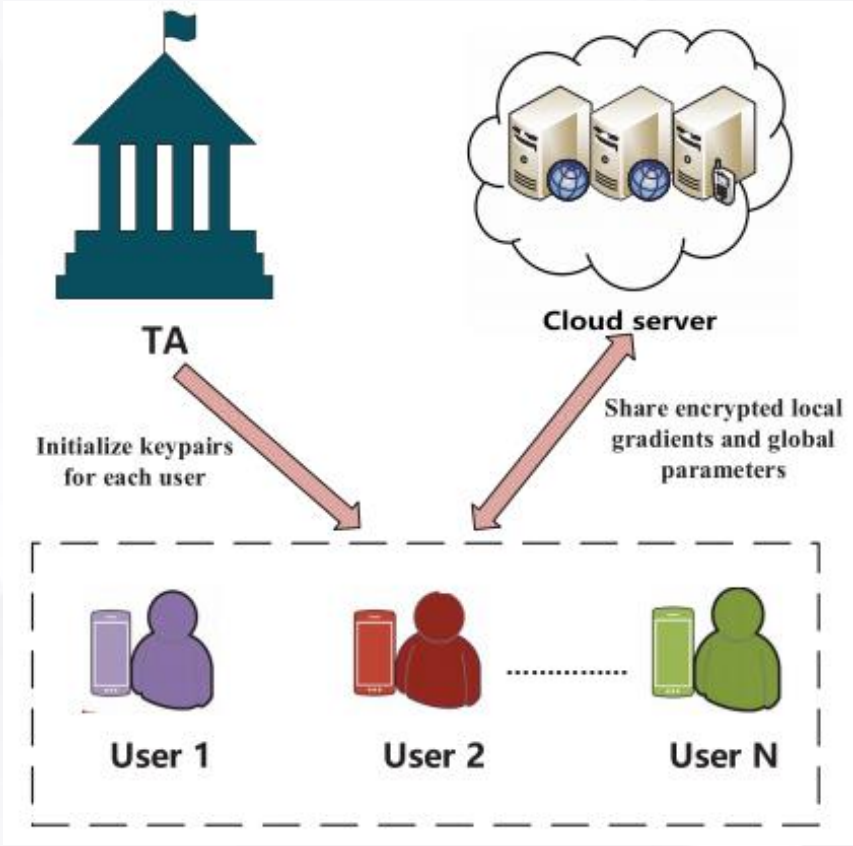
- DP is introduced to add noise to participants' uploaded parameters



Privacy protection

VerifyNet

- It gets listed as a preferred mitigation strategy to preserve privacy as it provides double-masking protocol which makes it difficult for attackers to infer training data.



Privacy protection

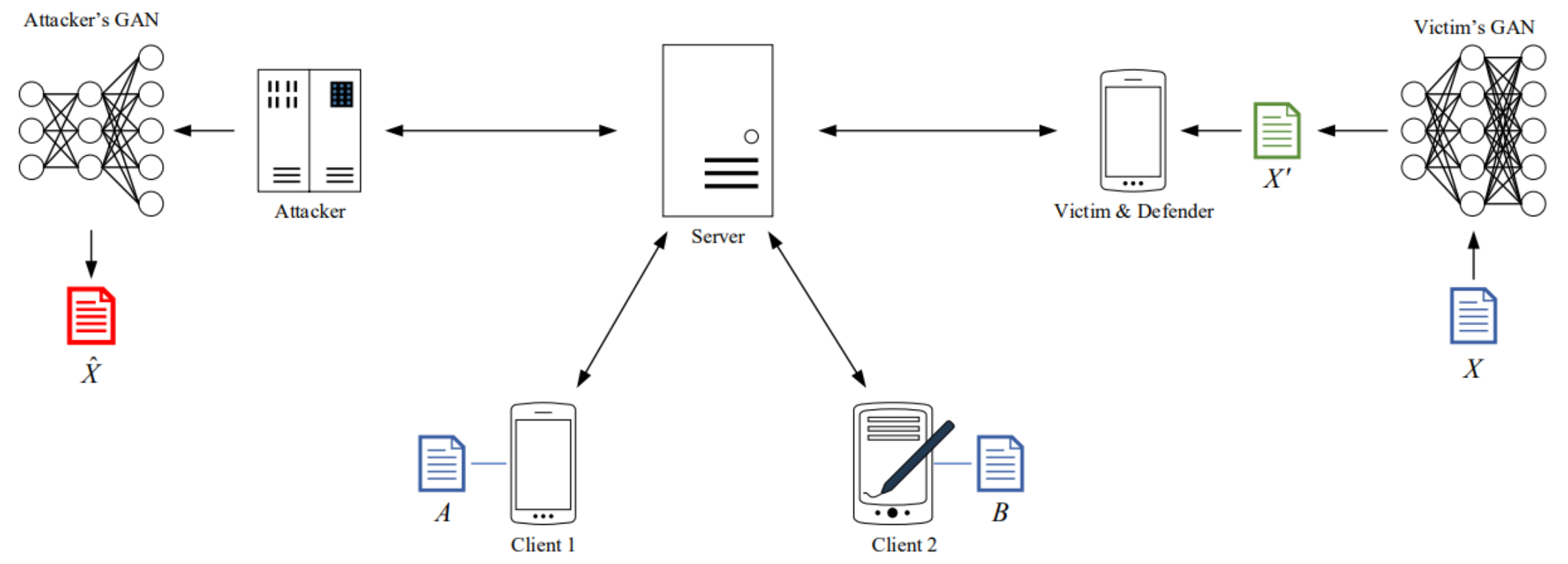
Adversarial training

- Evasion attacks from an adversarial user aims to fool ML models by injecting adversarial samples into the machine learning models.
- The attacker tries to impact the robustness of the FL model with perturbed data.
- Adversarial training, which is a proactive defense technique, tries all permutations of an attack from the beginning of the training phase to make the FL global model robust to known adversarial attacks.

Privacy protection

Adversarial training

- Use GAN to generate fake training data.



Privacy protection

Associated cost

Approach	Cost	Methodology
Secure Multi-party Computation	Efficiency loss due to encryption	Encrypt uploaded parameters
Differential Privacy	Accuracy loss due to added noise in client's model	Add random noise to uploaded parameters
Hybrid	Subdued cost on both efficiency and accuracy	Encrypt the manipulated parameter
VerifyNet	Communication overhead	Double-masking protocol Verifiable aggregation results
Adversarial Training	Computation power, training time for adversarial samples	Include adversarial samples in training data

04

Heterogeneity

- Addressing systems heterogeneity
- Addressing statistical heterogeneity

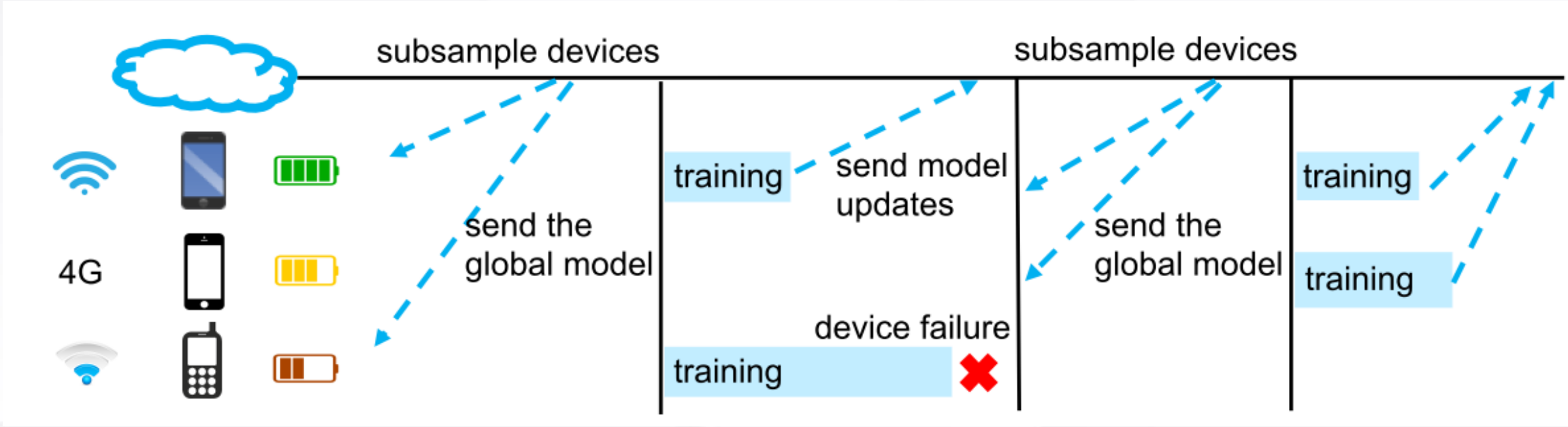
Heterogeneity

- Addressing systems heterogeneity
  - Asynchronous communication
  - Active sampling
  - Fault tolerance
  - Using client-specific model

Heterogeneity

Asynchronous communication

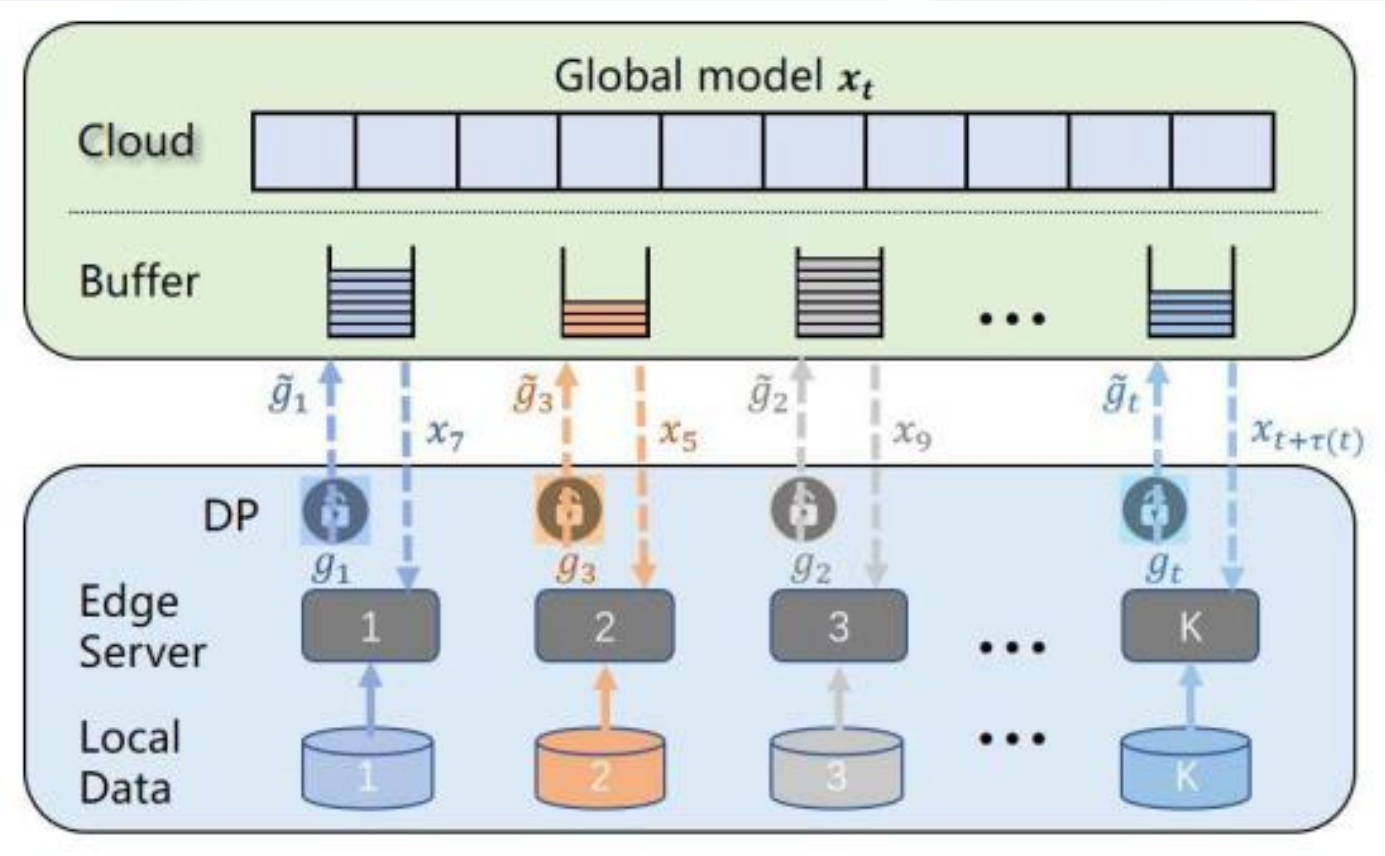
- Synchronous schemes are simple and guarantee a serial-equivalent computational model, but they are also more susceptible to stragglers in the face of device variability.



Heterogeneity

Asynchronous communication

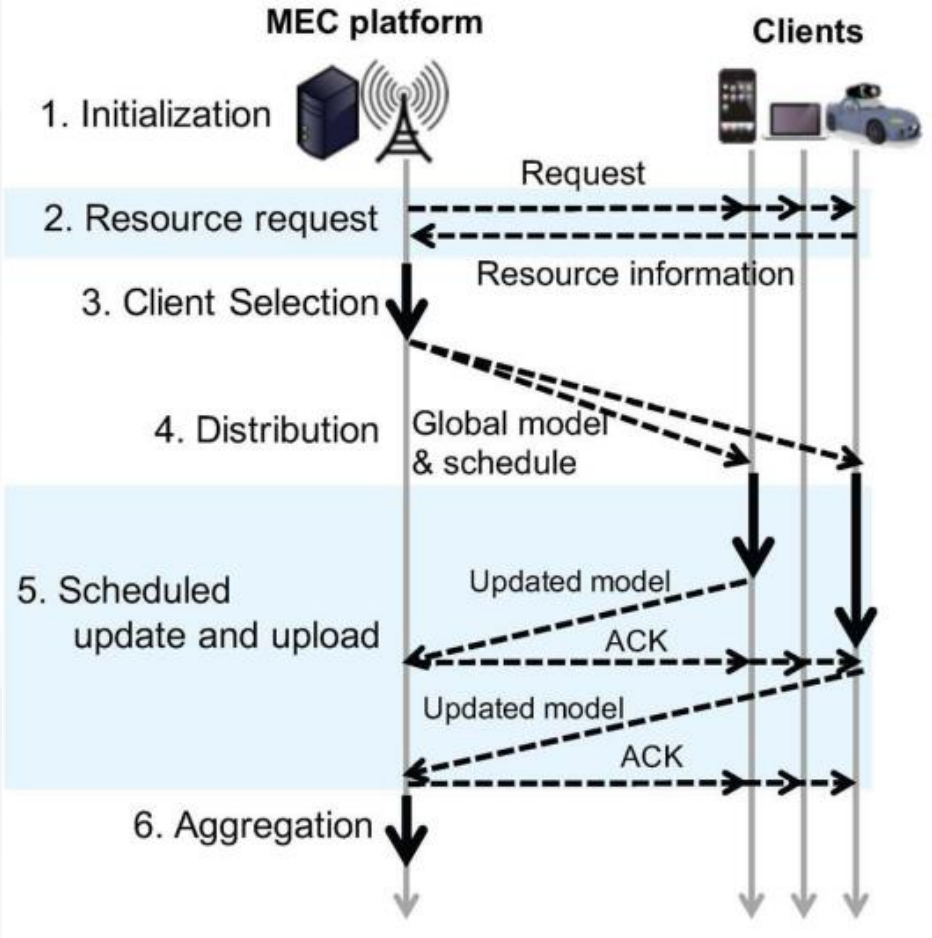
- Asynchronous schemes are an attractive approach to mitigate stragglers in heterogeneous environments.



Heterogeneity

Active sampling

- Actively selecting participating devices at each round.



Heterogeneity

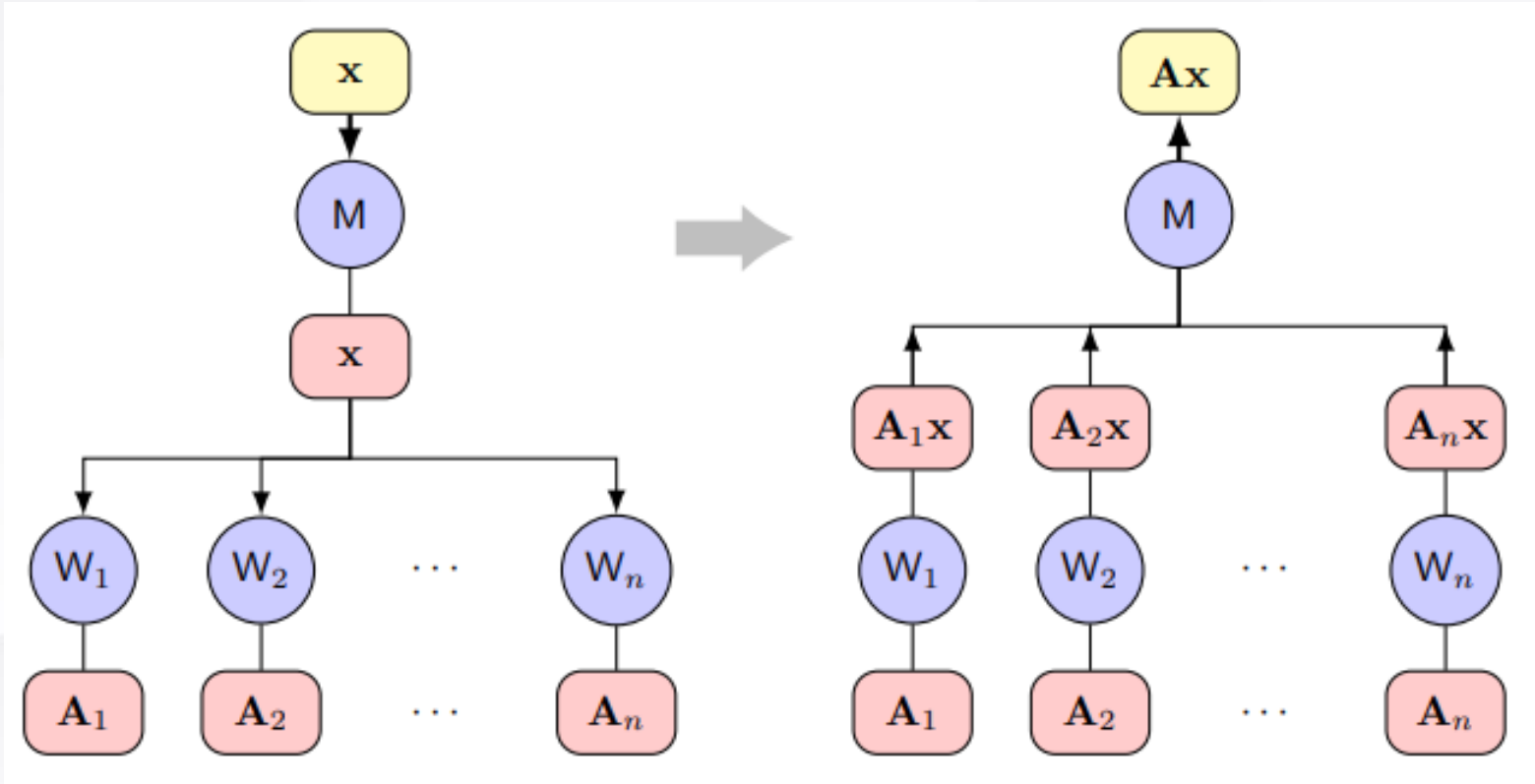
Fault tolerance

- Fault tolerance has been extensively studied in the systems community and is a fundamental consideration of classical distributed systems.
- When learning over remote devices, however, fault tolerance becomes more critical.
- One practical strategy is to simply ignore such device failure, which may introduce bias into the device sampling scheme if the failed devices have specific data characteristics.

Heterogeneity

Fault tolerance

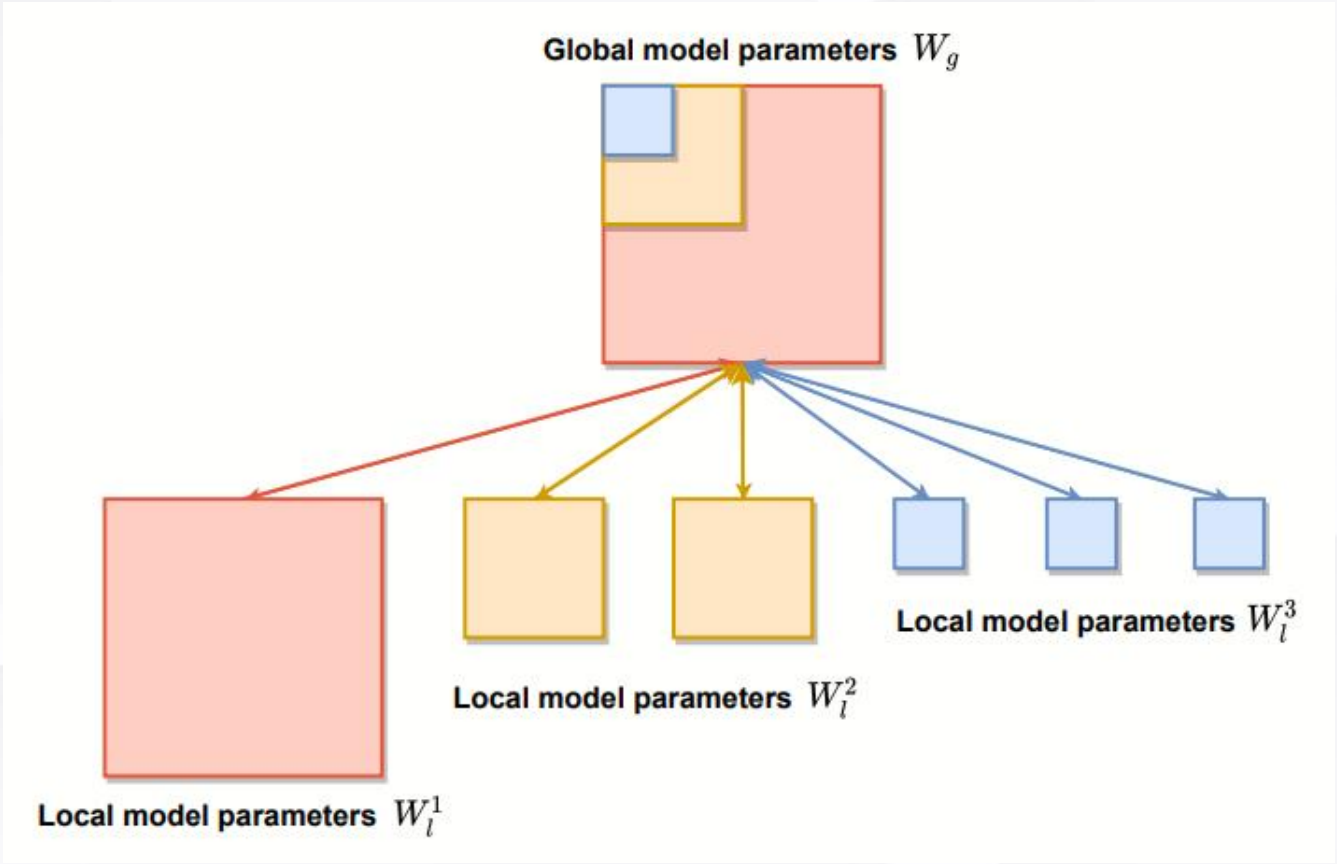
- Coded computation is another option to tolerate device failures by introducing algorithmic redundancy.



Heterogeneity

Using client-specific model

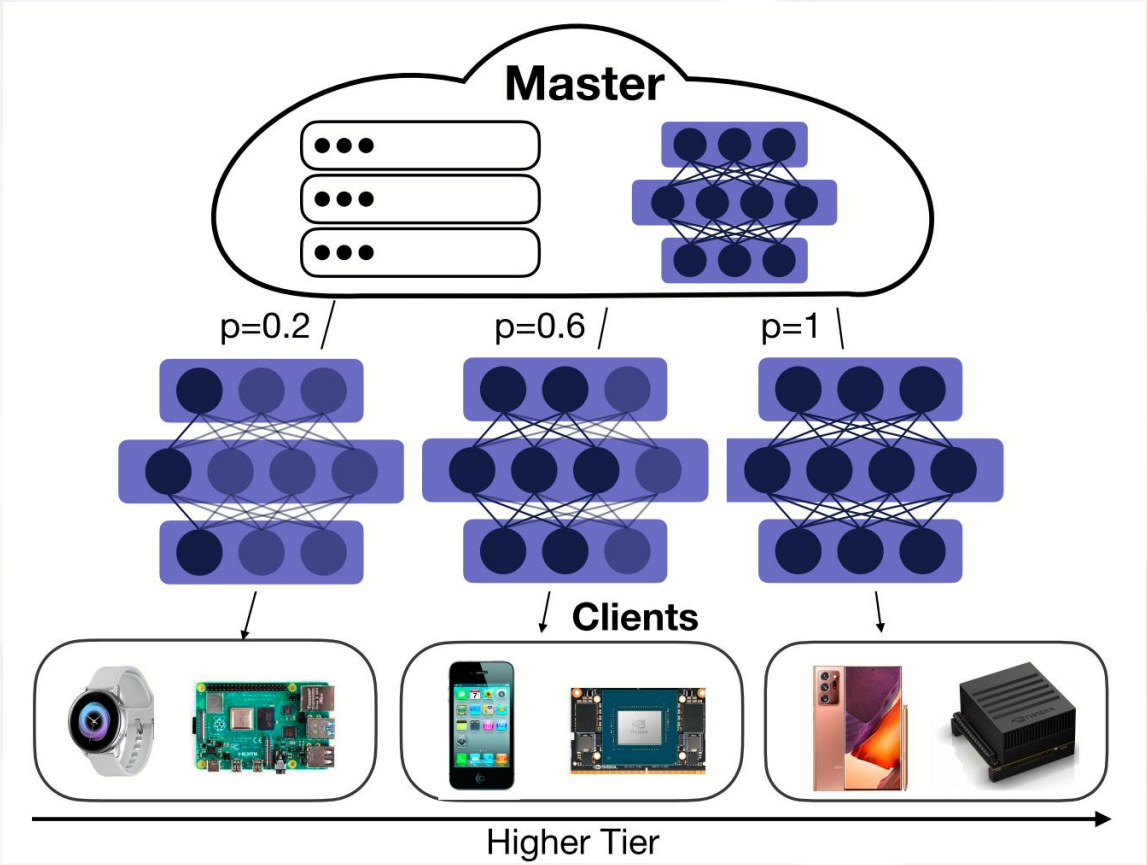
- HeteroFL trains heterogeneous local models and aggregate them stably and effectively into a single global inference model.



Heterogeneity

Using client-specific model

- FjORD employs Ordered Dropout (OD) to tailor the amount of computation to the capabilities of each participating device.



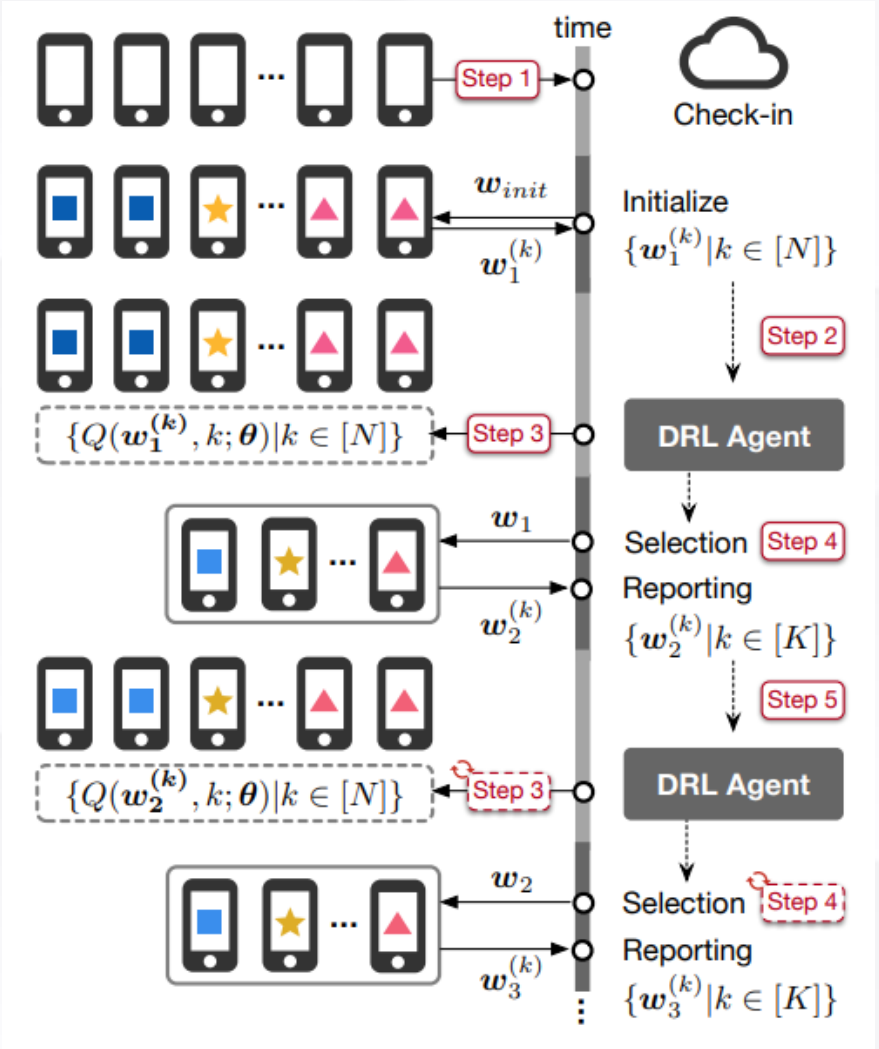
Heterogeneity

Statistical heterogeneity

- Overcome the non-IID and unbalanced issue
- Utilize the non-IID and unbalanced characteristic

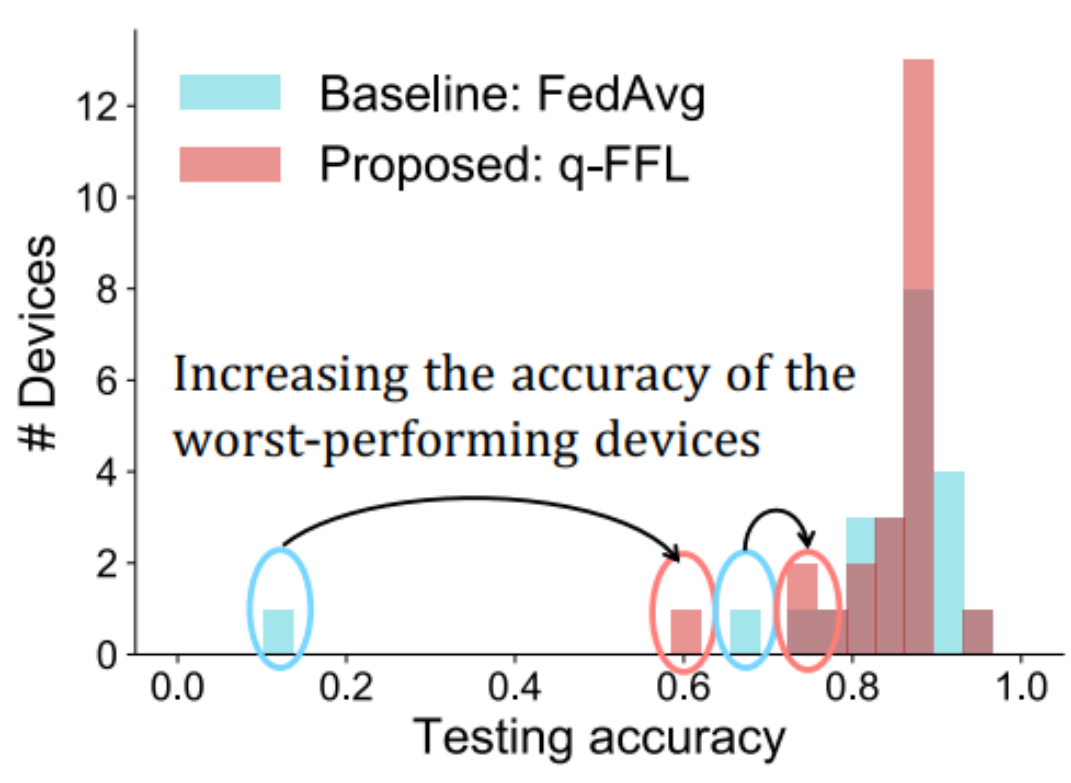
Heterogeneity

- **Overcome the non-IID and unbalanced issue**
  - Although the data is not independent and identically distributed among all the clients, we can relieve this issue by client selection.
  - Client selection can be formulated as a deep reinforcement learning problem in federated learning.
  - It solely relies on model weight information to determine which device may improve the global model the most —thus preserving the same level of privacy as the original FL does.



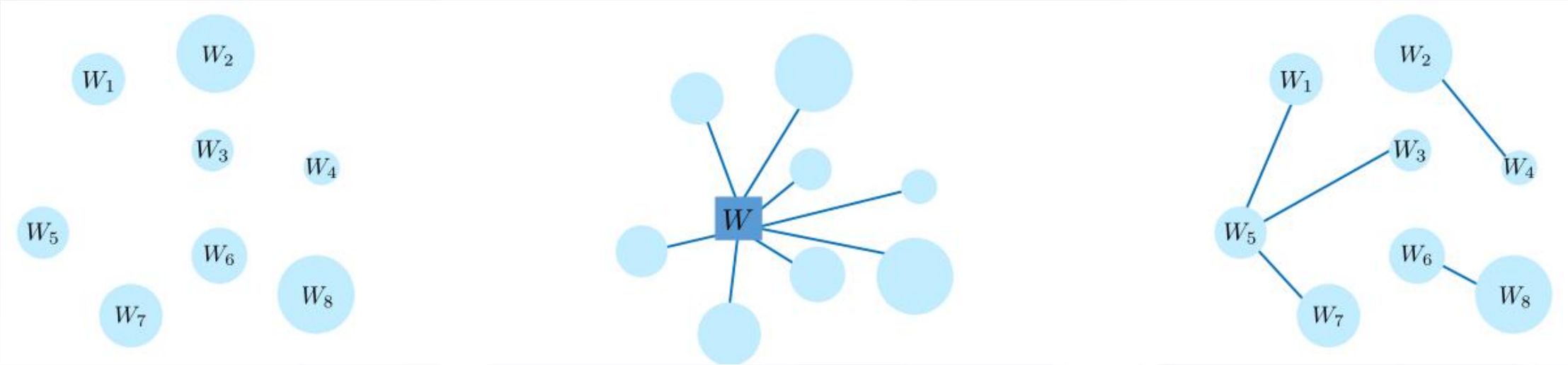
Heterogeneity

- **Overcome the non-IID and unbalanced issue**
  - **Devices with higher loss are given higher relative weight to encourage less variance in the final accuracy distribution.**



Heterogeneity

- **Utilize the non-IID and unbalanced characteristic**
  - Non-IID data is not just an issue for federated learning, but also a natural feature in this setting.
  - Personalized federated learning is welcomed.



Learn personalized models for each device; do not learn from peers.

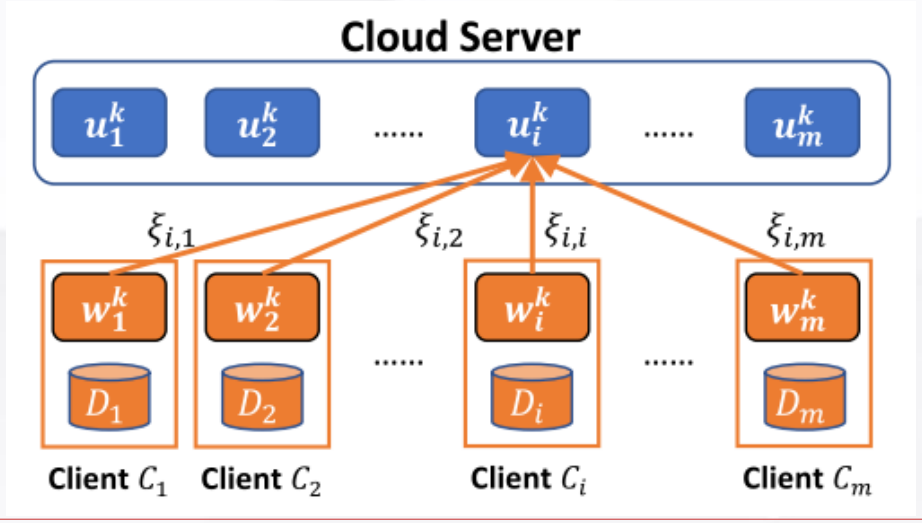
Learn a global model; learn from peers.

Learn personalized models for each device; learn from peers.

Heterogeneity

Personalized federated learning

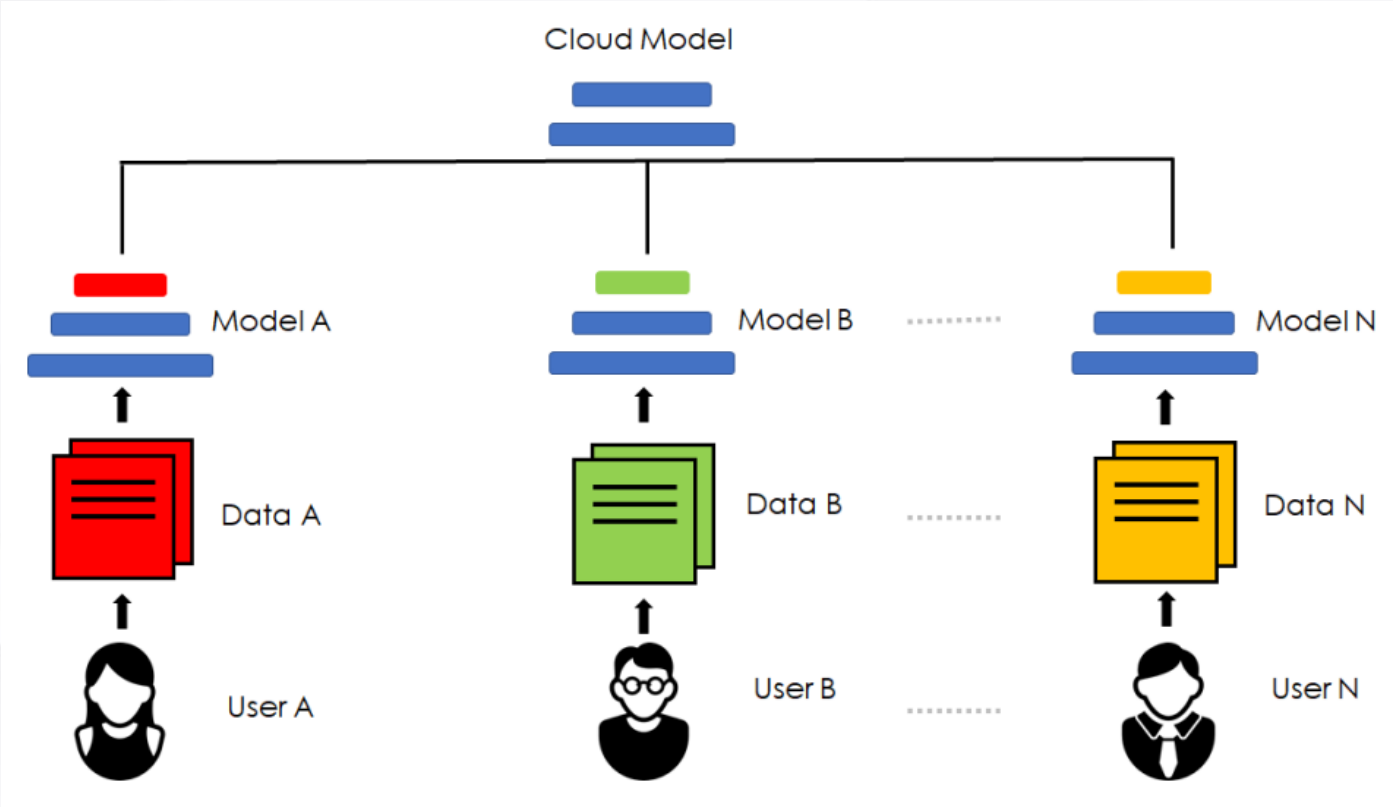
- FedAMP allows each client to own a local personalized model, it maintains a personalized cloud model on the cloud server for each client.
- FedAMP realizes the attentive message passing mechanism by attentively passing the personalized model of each client as a message to the personalized cloud models with similar model parameters.
- FedAMP updates the personalized cloud model of each client by a weighted convex combination of all the messages it receives.



Heterogeneity

Personalized federated learning

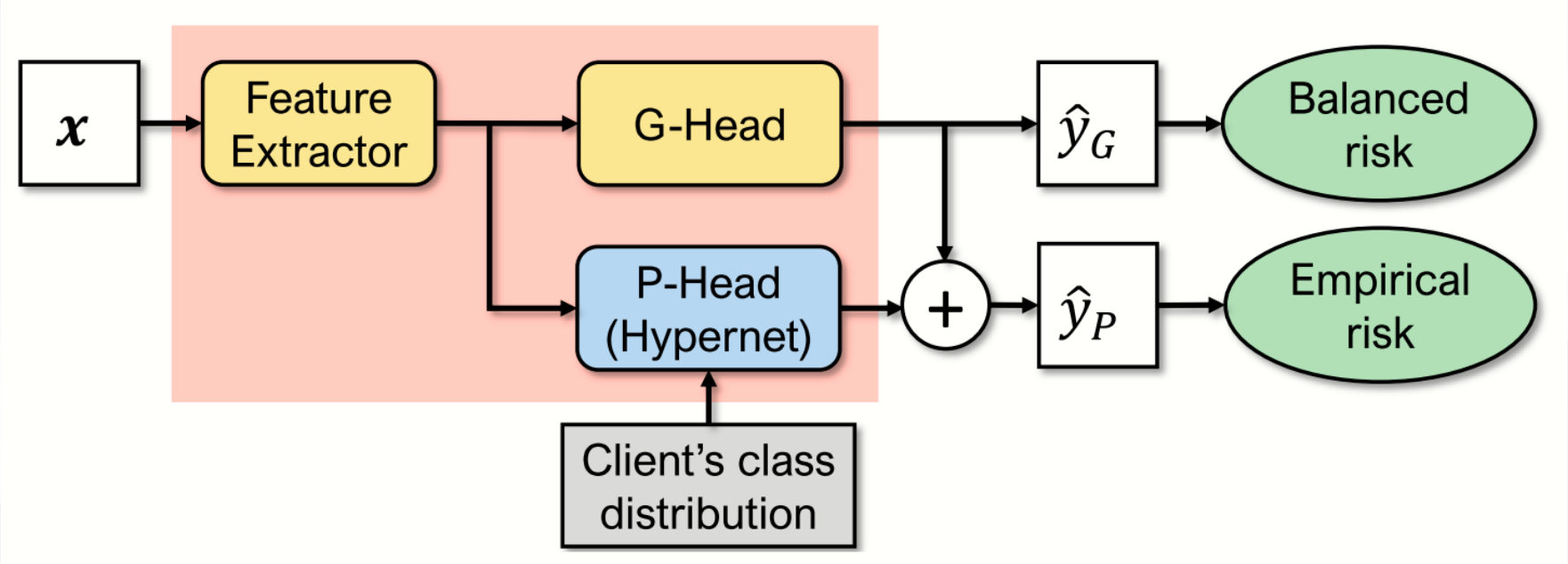
- The base layers are shared with the parameter server while the personalization layers are kept private by each device.



Heterogeneity

Personalized federated learning

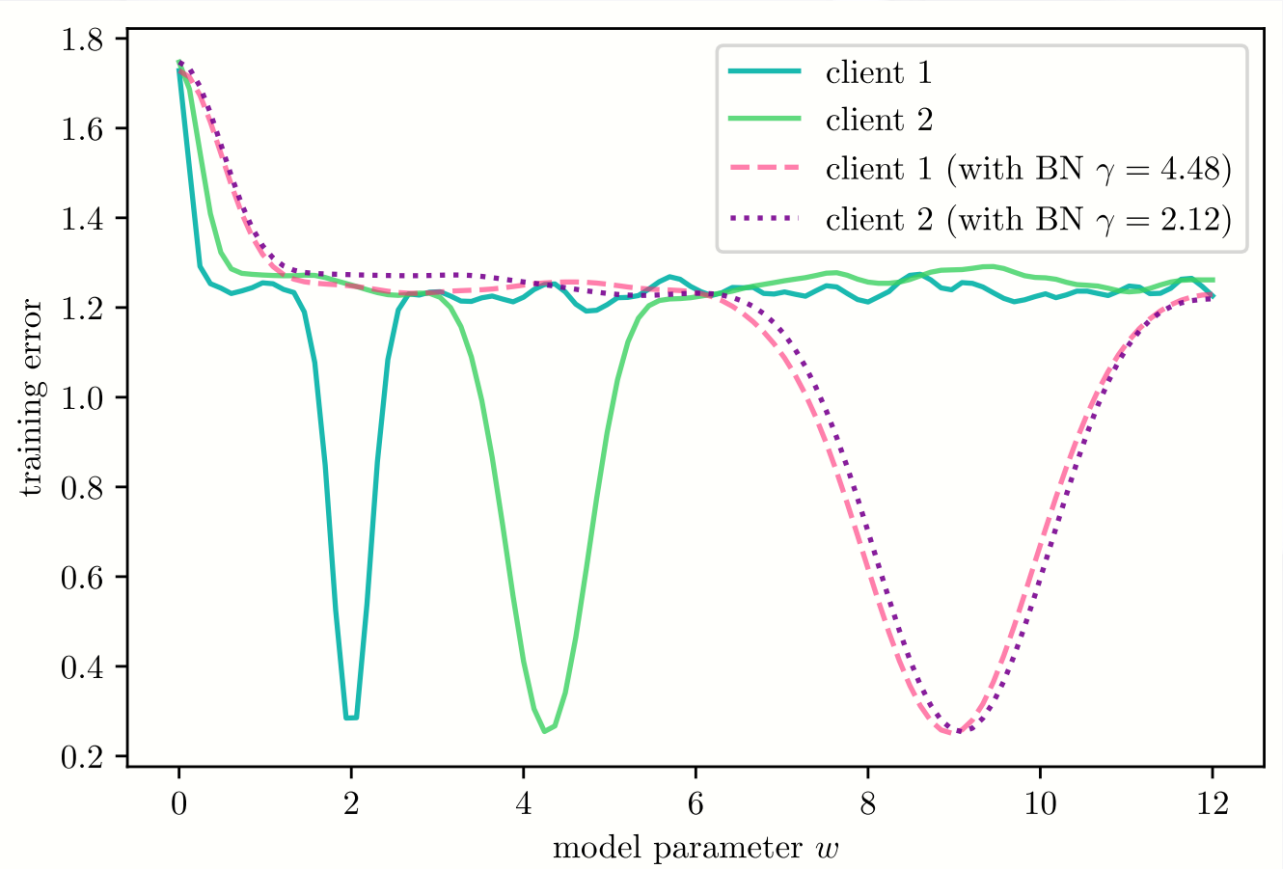
- In addition to only learning for the local objective, FedRoD also proposes to simultaneously learn the balanced objective and the local objective on each client.



Heterogeneity

Personalized federated learning

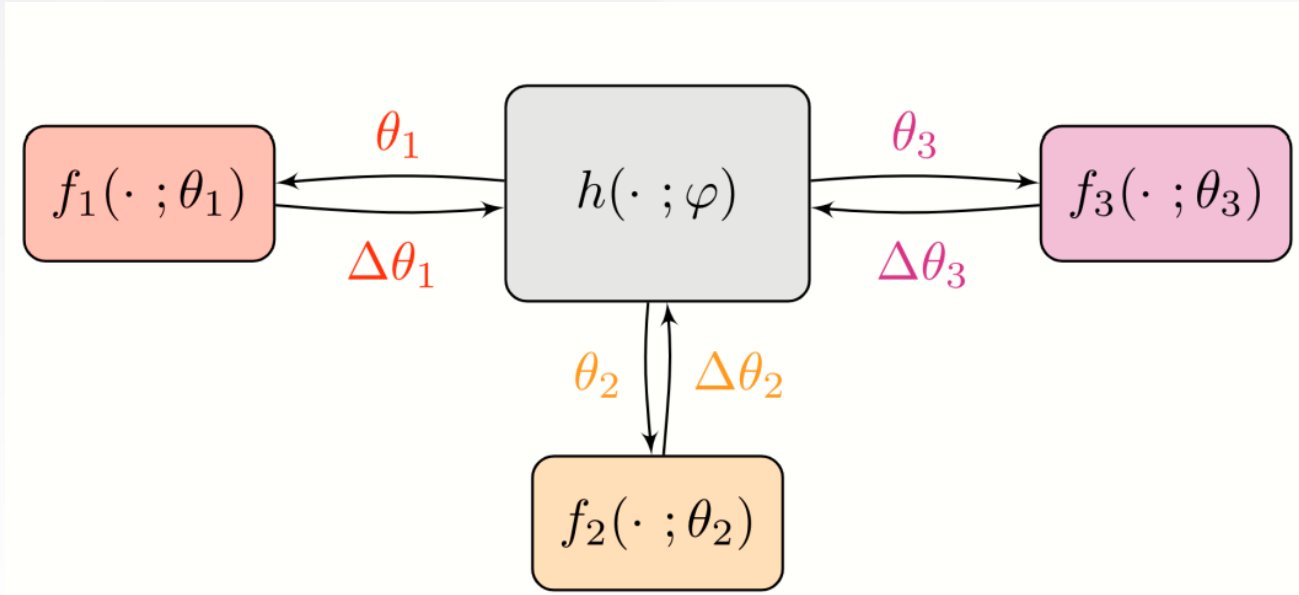
- Rather than localize only the higher layers, FedBN finds that the batch normalization (BN) layers in the ResNets are not beneficial for aggregation and proposes to localize all of them.



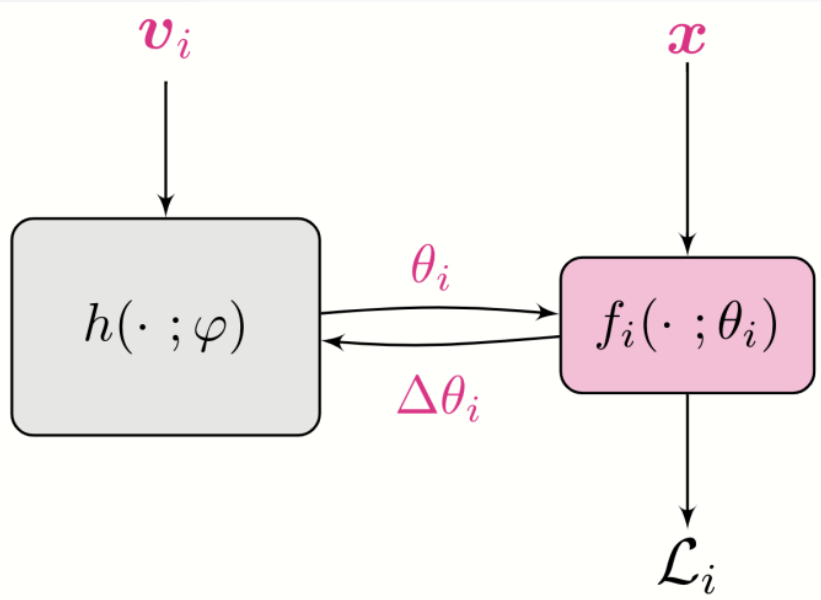
Heterogeneity

Personalized federated learning

- To further personalize models, FedHN assigns one client embedding for one client, and generate client model parameters through the hypernetwork on the server.



Updating the hypernetwork by client models

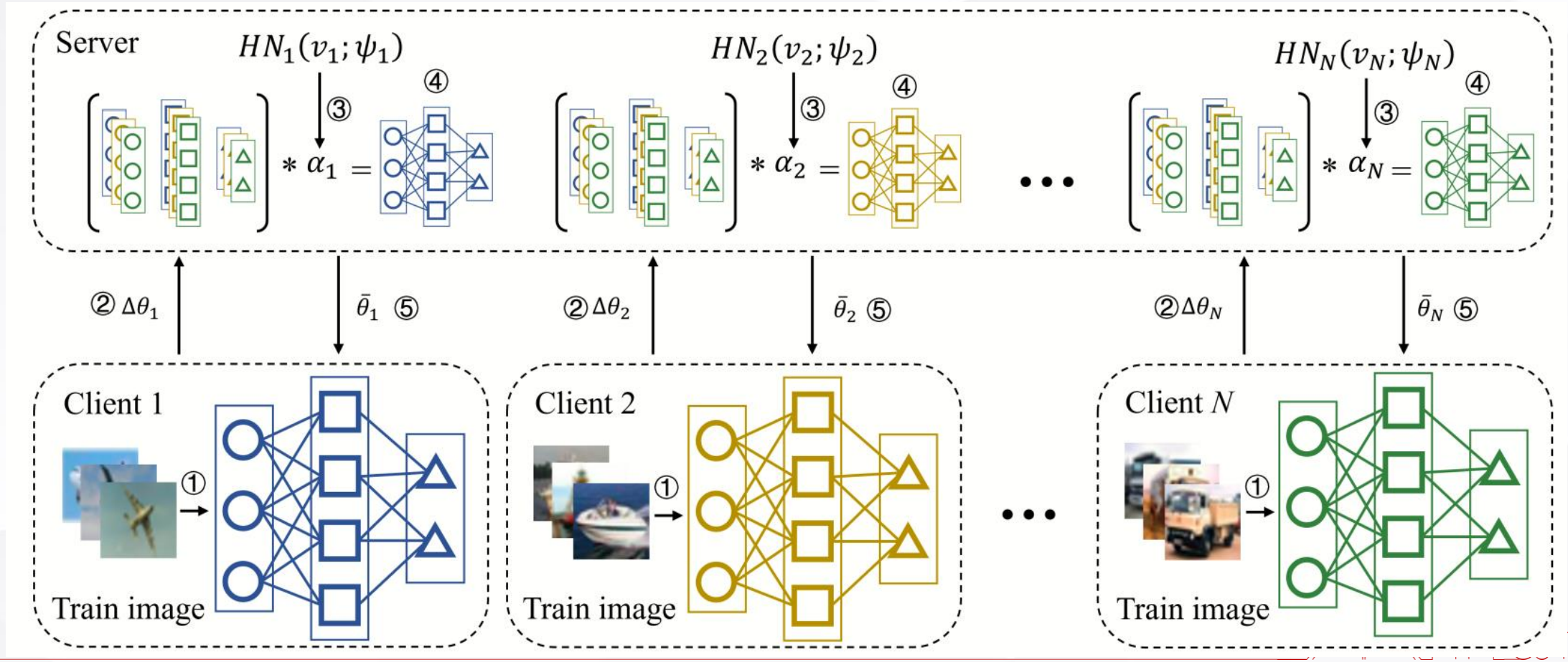


Generating client models

Heterogeneity

Personalized federated learning

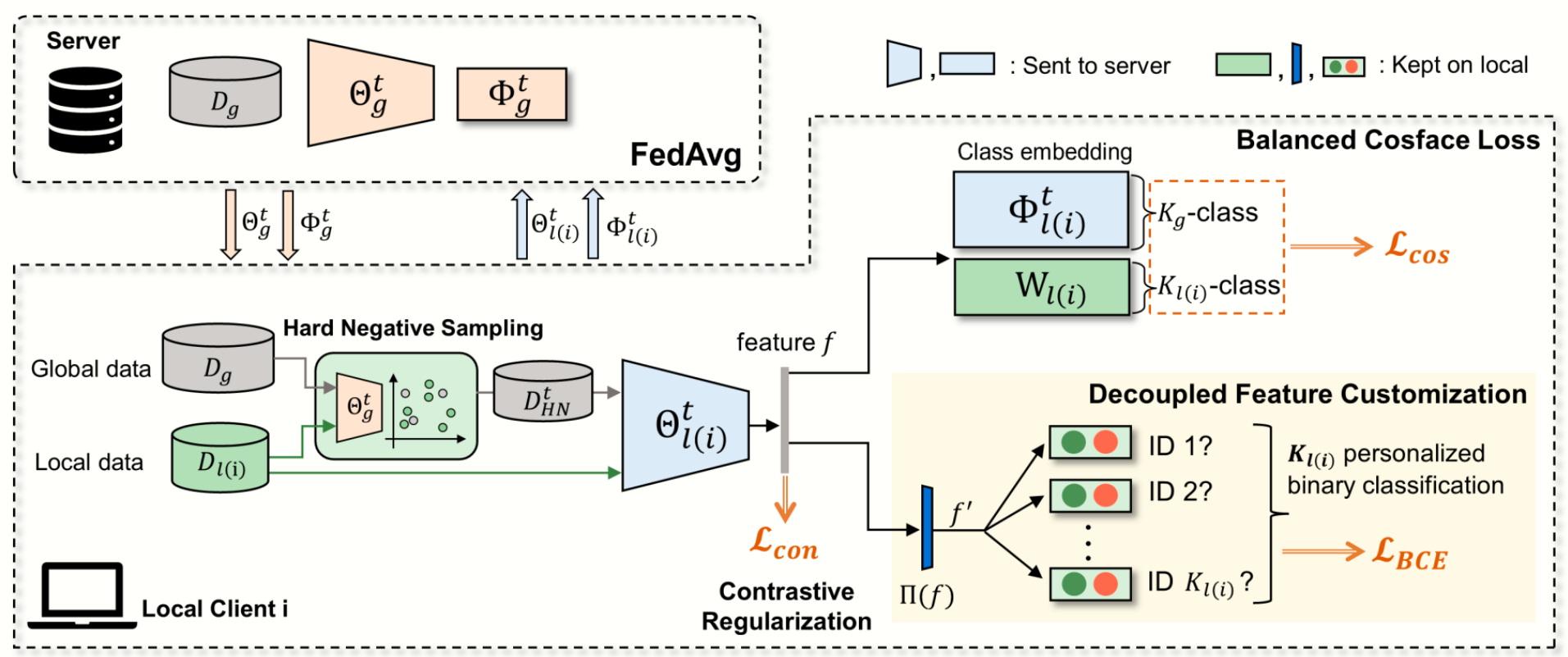
- With the client embeddings, pFedLA generate the layer-wise aggregation weights instead of directly generating the model parameters.



Heterogeneity

Personalized federated learning

- Public dataset also helps FL
- Like FedRoD , FedFR learns two objectives on the client, one of which is the balanced objective, in the face recognition task.



05

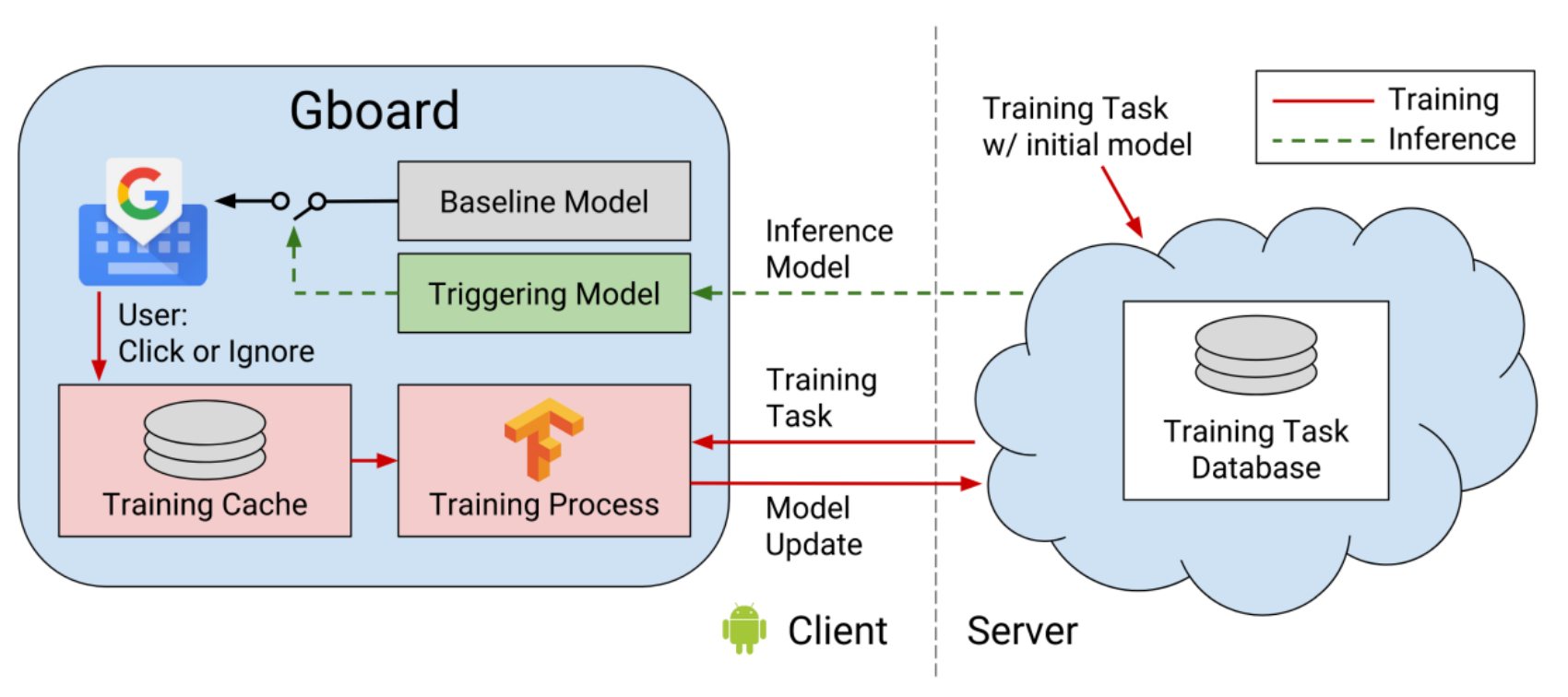
Application

- Gboard
- Recommender system
- Autonomous driving
- …

Application

Gboard

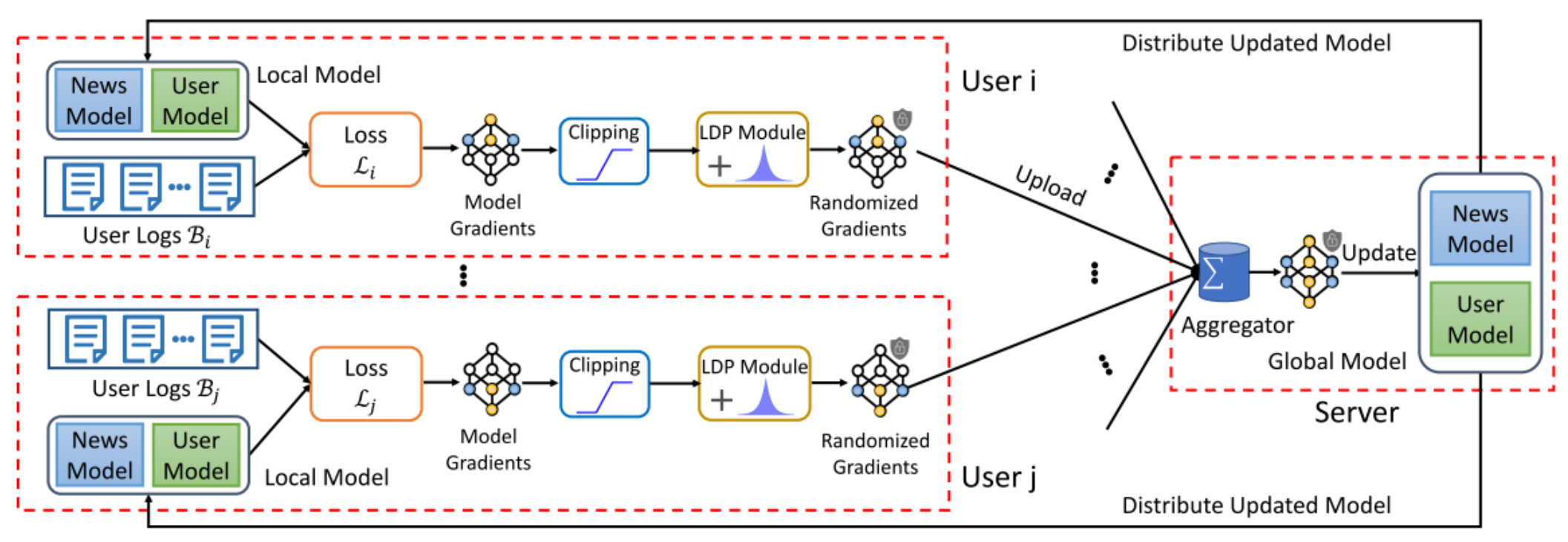
- Google's first implementation of federated learning.
- Triggering model is trained federated to tune the results of the pre-trained baseline model for better performance.



Application

Recommender system

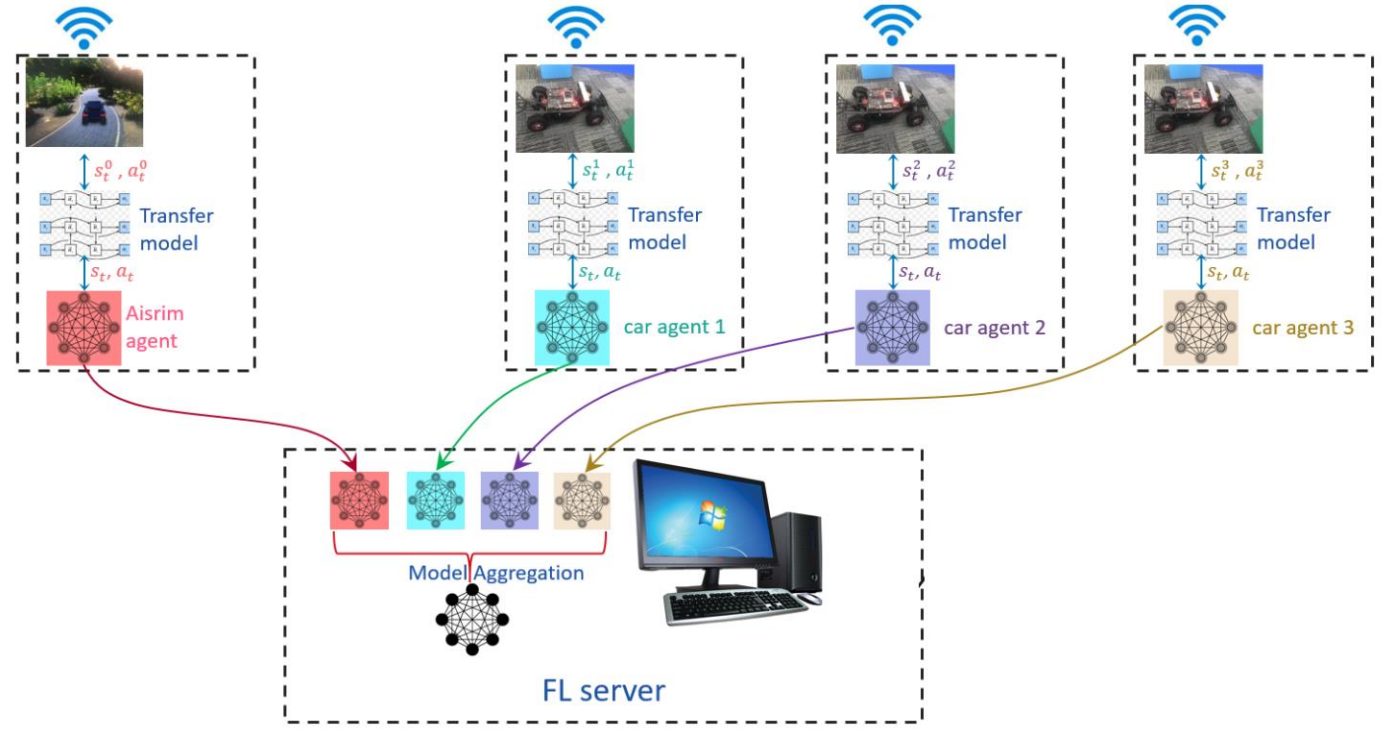
- The news model aims to learn news representations to model news content.
- The user model is used to learn user representations to model their personal interest.
- LDP denotes the local differential privacy



Application

Autonomous driving

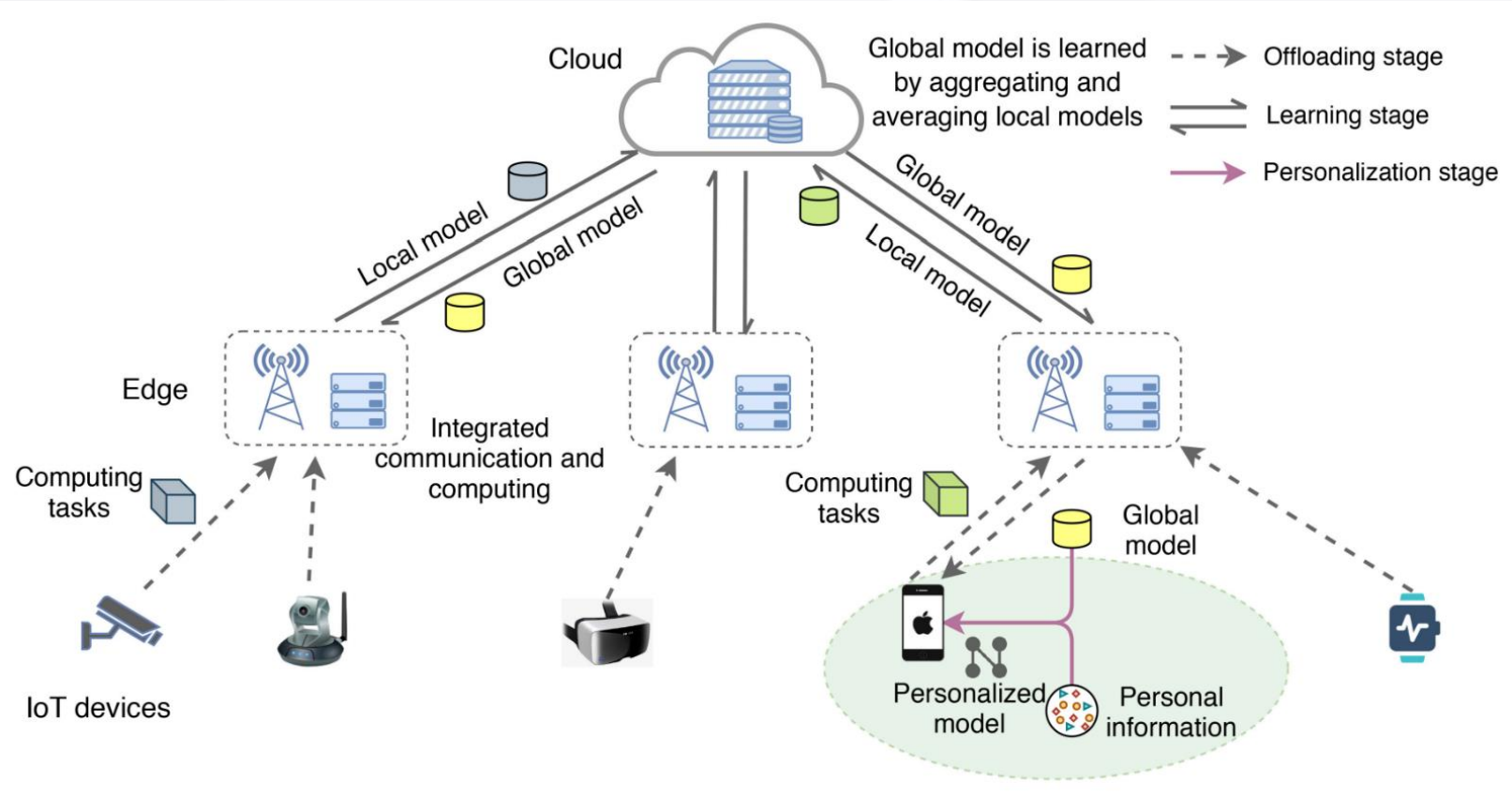
- The FTRL framework for collision avoidance RL tasks of autonomous driving cars
- Global model is asynchronously updated by different RL agents.
- Transfer knowledge from virtual world (Airsim platform) to real world



Application

IOT

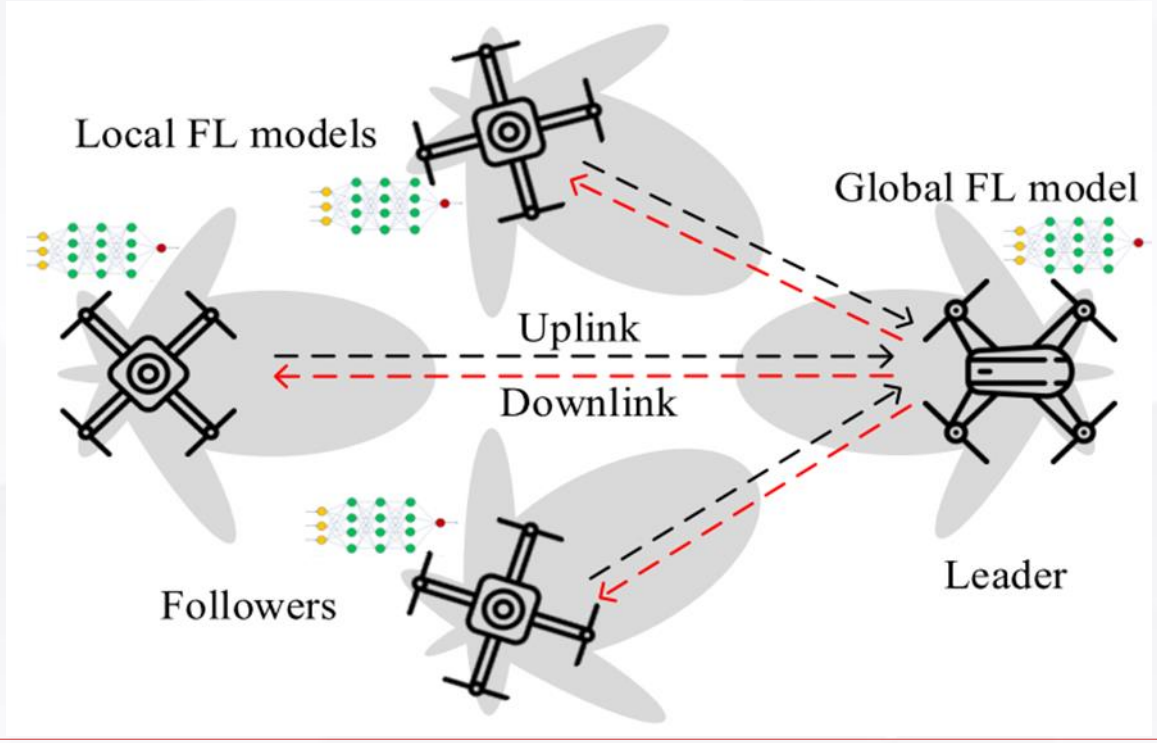
- Personalized federated learning framework for intelligent IoT applications.
- Supports flexible selection of personalized federated learning approaches.



Application

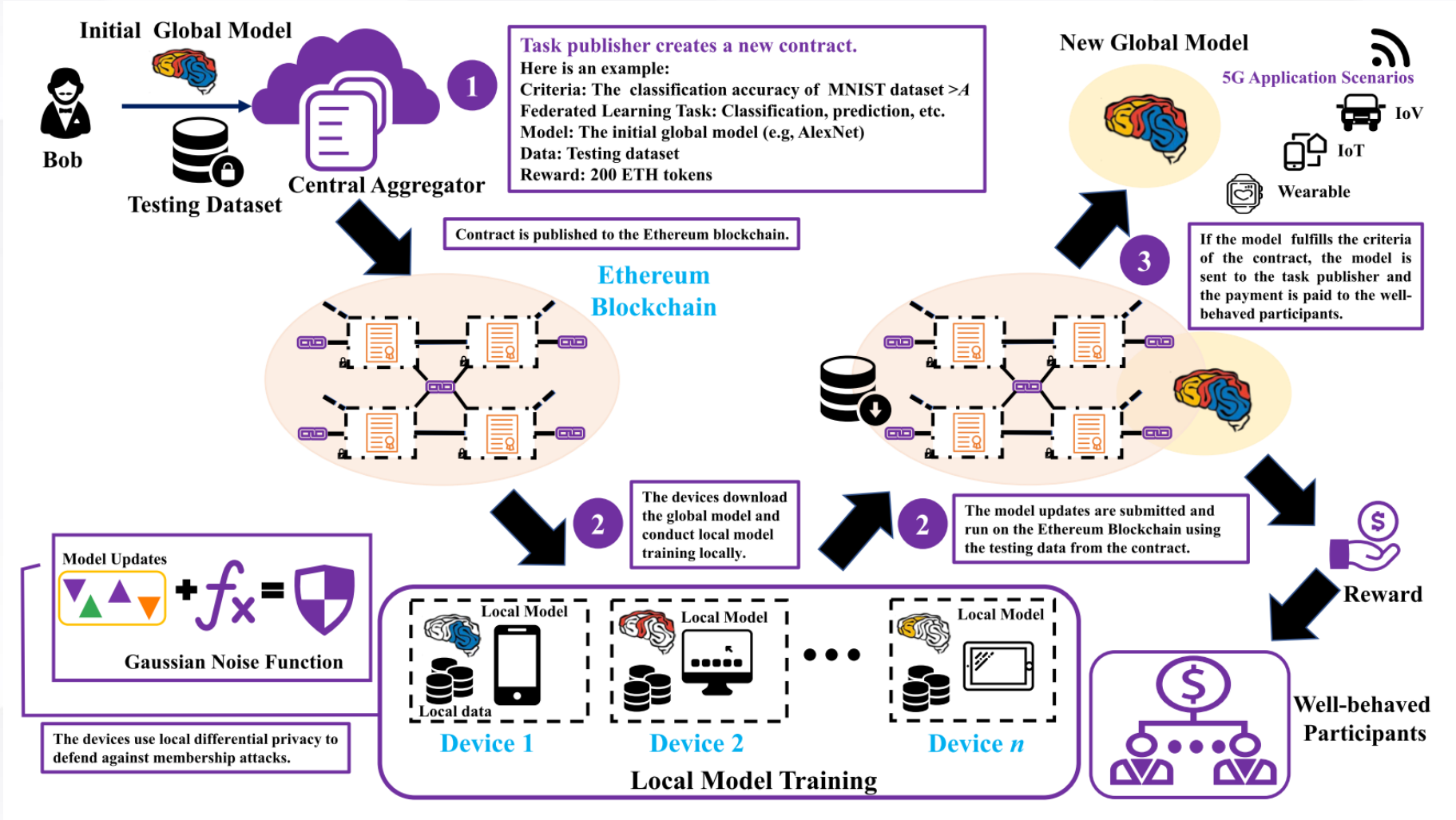
UAV (Unmanned aerial vehicle)

- Due to the high mobility of UAVs and their limited energy and stringent energy limitations, the analysis in previous federated learning work cannot be directly applied for UAV swarms.
- Use a sample average approximation approach from stochastic programming along with a dual method from convex optimization.



Application

Blockchain



谢谢！

饮 水 思 源　爱 国 荣 校