Towards Federated Learning at Scale Google's FL System

Wolfgang Grieskamp et al.

Why do we need Federated Learning?

Data is born at the edge

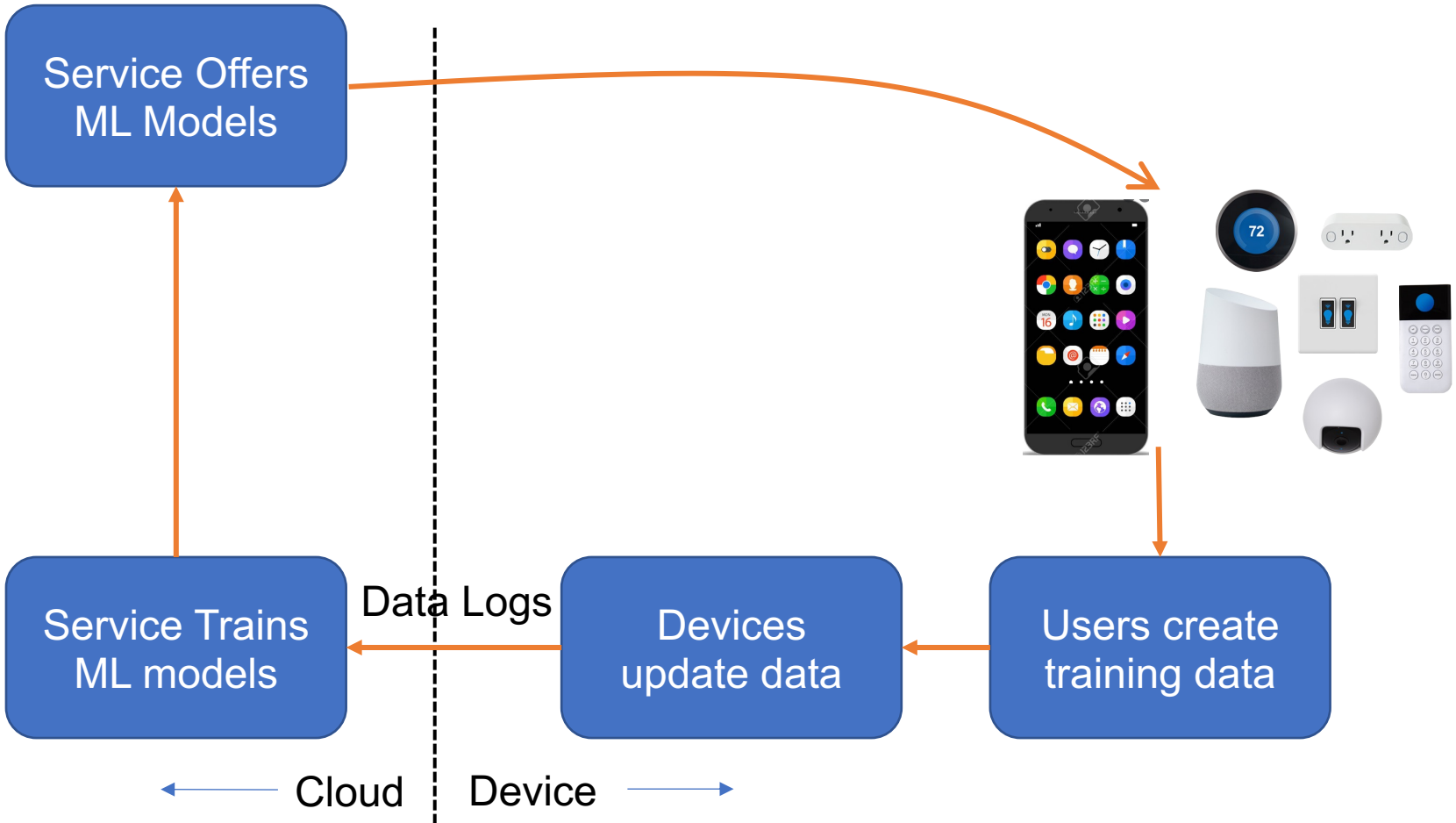
- Billions of phones and IoT devices constantly generating data
- Data can help improve products and services
- **How to leverage edge data?**

Smart Phones

IoT Devices

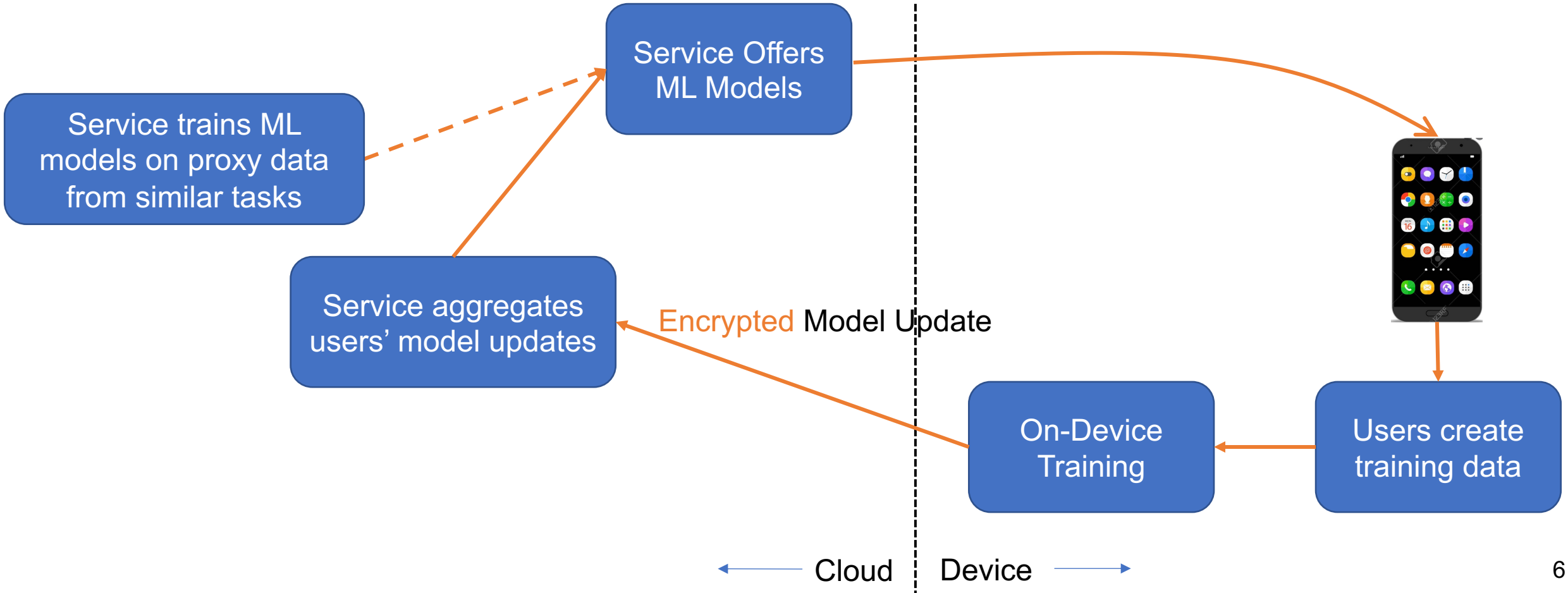
Traditional On-device Inference with Cloud-Trained Models

- Limitations: compromised privacy, sharing user data can be expensive (e.g., videos), centralized data collection



Federated Learning

- Key idea: allow cross-device models to be trained and evaluated without centralized data collection



Federated Learning: Advantages

- More representative data
- Protected privacy and ownership of data
- Preserved locality of data
- Enables edge devices to collaborate (more aggregated resource)

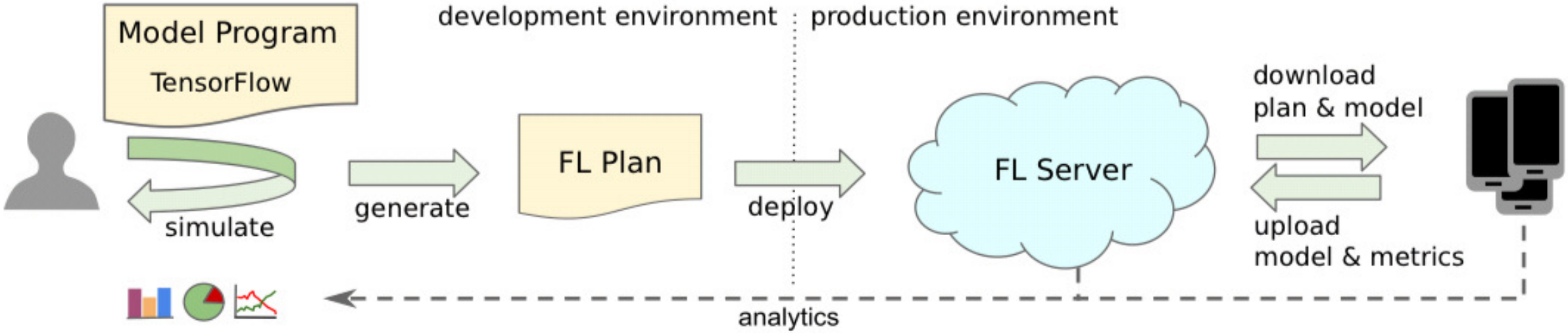
"Bring the code to the data and not the data to the code"

---Wolfgang Grieskamp, SysML19

Federated Learning is Hard

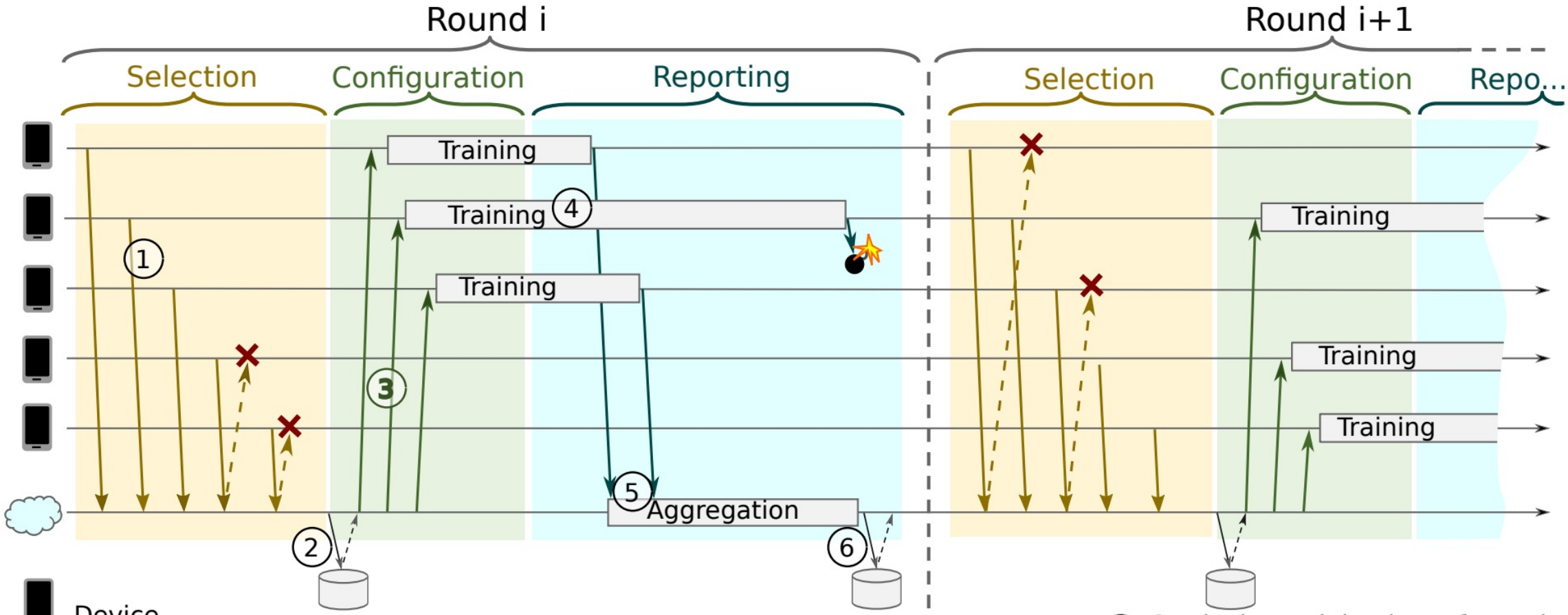
- Health of user devices must not be compromised
- User devices can drop any moment and at high rate
- No direct access to devices for diagnosis
- Large and uneven scale of populations on server side

Developer Workflow



- Model developers depend on the production system for experiments
  - Only have accesses to proxy data but not to real data
  - Develop on the cloud and then push the result to production and collect metrics
- Deployment must never affect the user experience on device
  - Training has no visible effect to the user
  - Device architecture ensures that device health is not compromised

Federated Learning Round Protocol



Device

Server

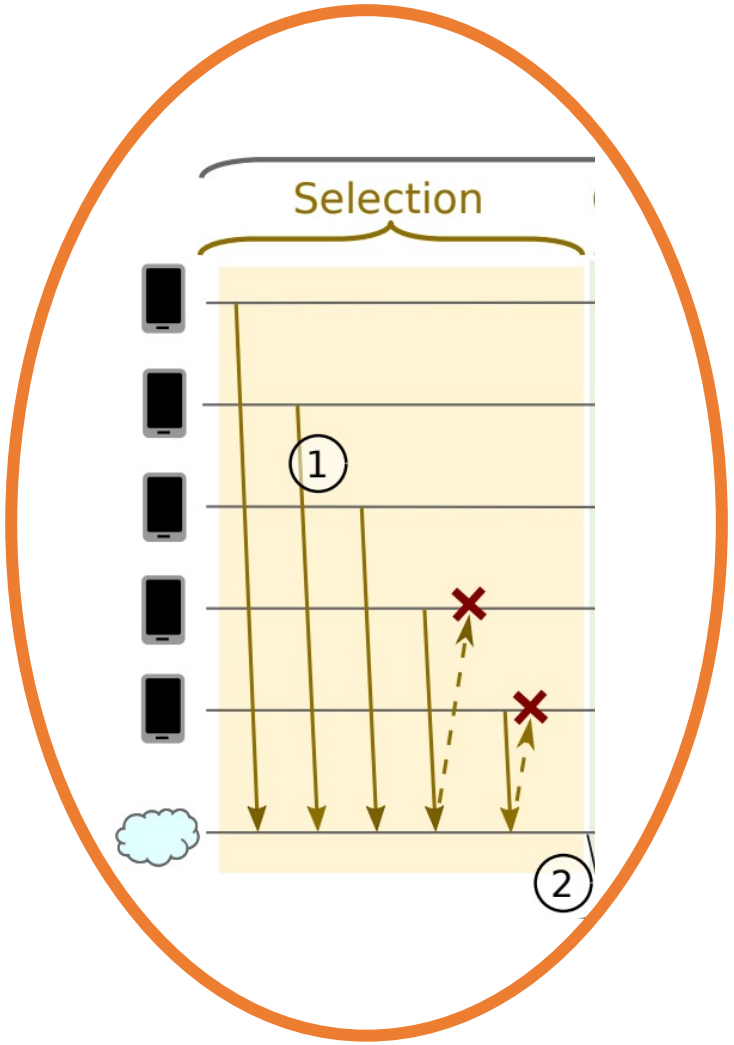
Persistent storage

Rejection ("come back later!")

Device or network failure

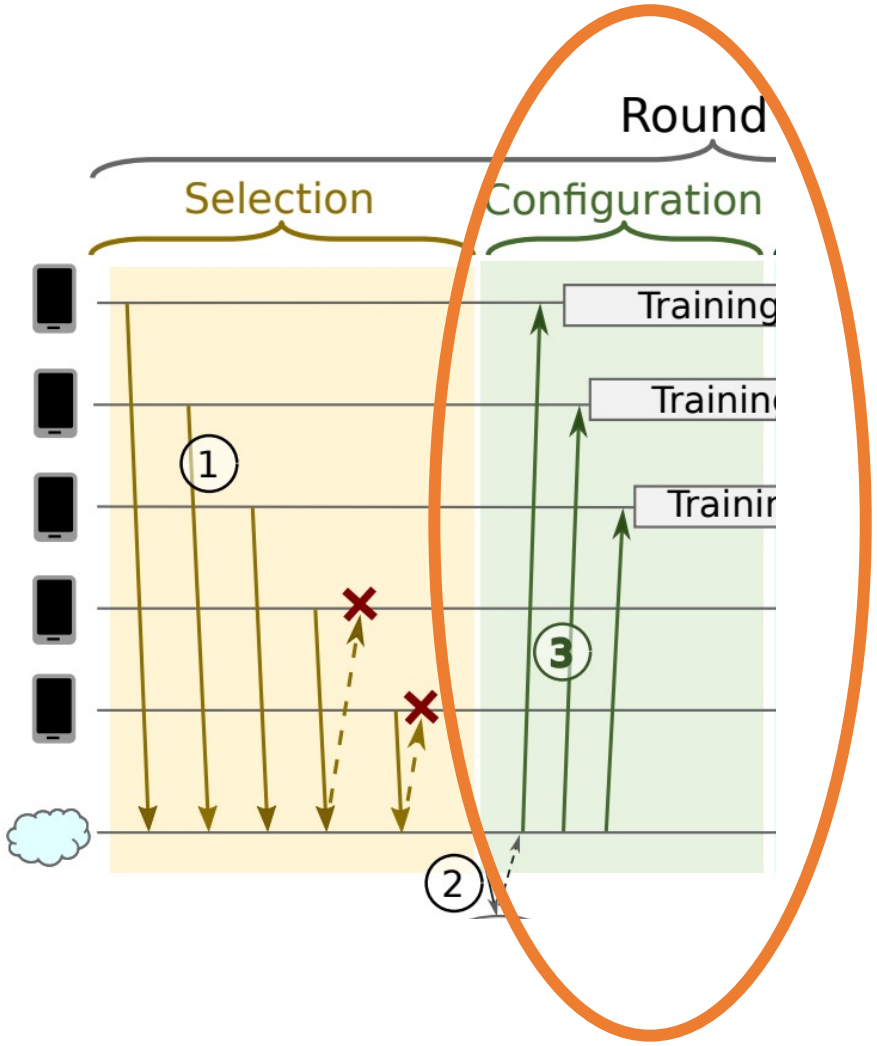
1. Devices check-in with the FL server, rejected ones are told to come back later
2. Server reads model checkpoint from persistent storage
3. Model and configuration are sent to selected devices
4. On-device training is performed, model update is reported back
5. Server aggregates updates into the global model as they arrive
6. Server writes global model checkpoint into persistent storage

Federated Learning Round Protocol: Selection



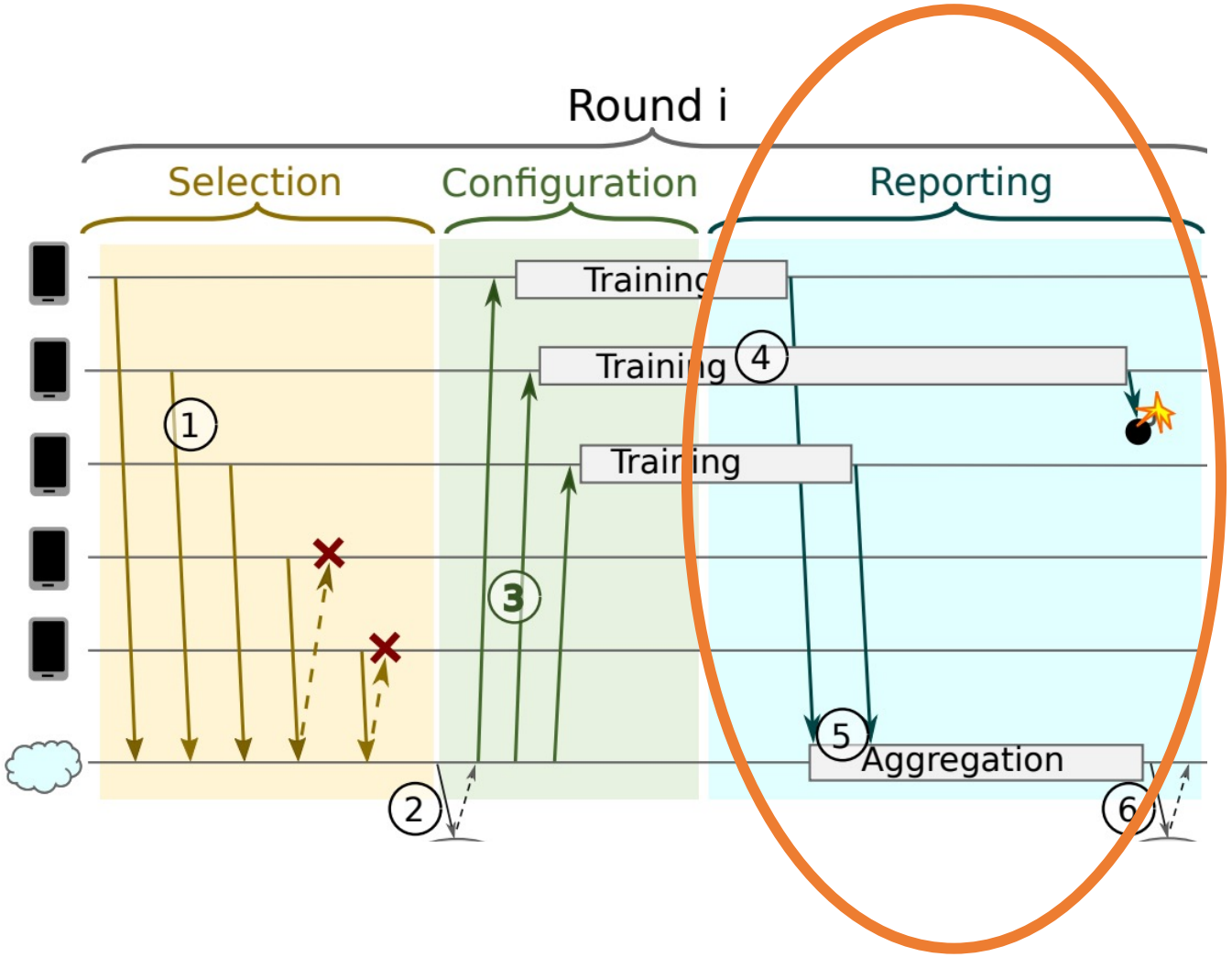
- Devices check in to server to announce availability for training, when device state allows
  - Charging, unmetered network, idle
- Server makes a random selection of participating devices – select a few hundred out of thousands
  - Quality doesn't increase by selecting more than few hundred devices
  - Server could apply strategies which clients to select

Federated Learning Round Protocol: Configuration



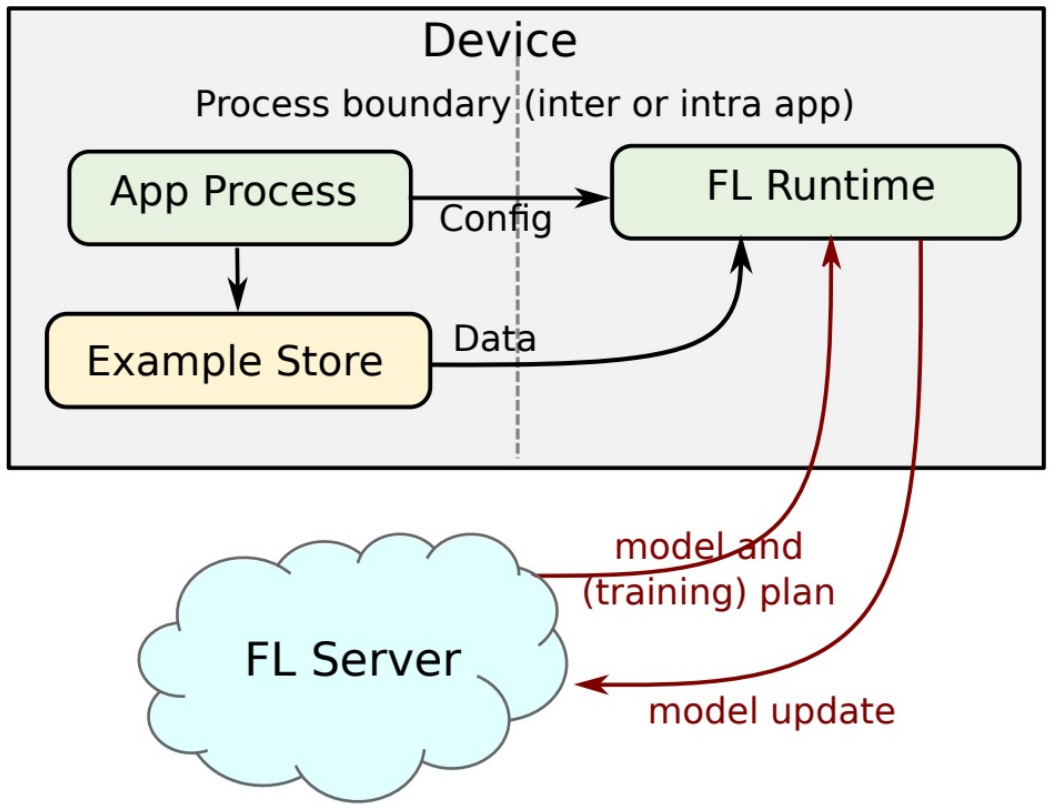
- Server locks population and reads current model weights from persistent storage.
- Server sends model and hyper-parameters to devices
- Devices start training as soon as they receive it
  - Apply model compression techniques to minimize communication overhead

Federated Learning Round Protocol: Reporting



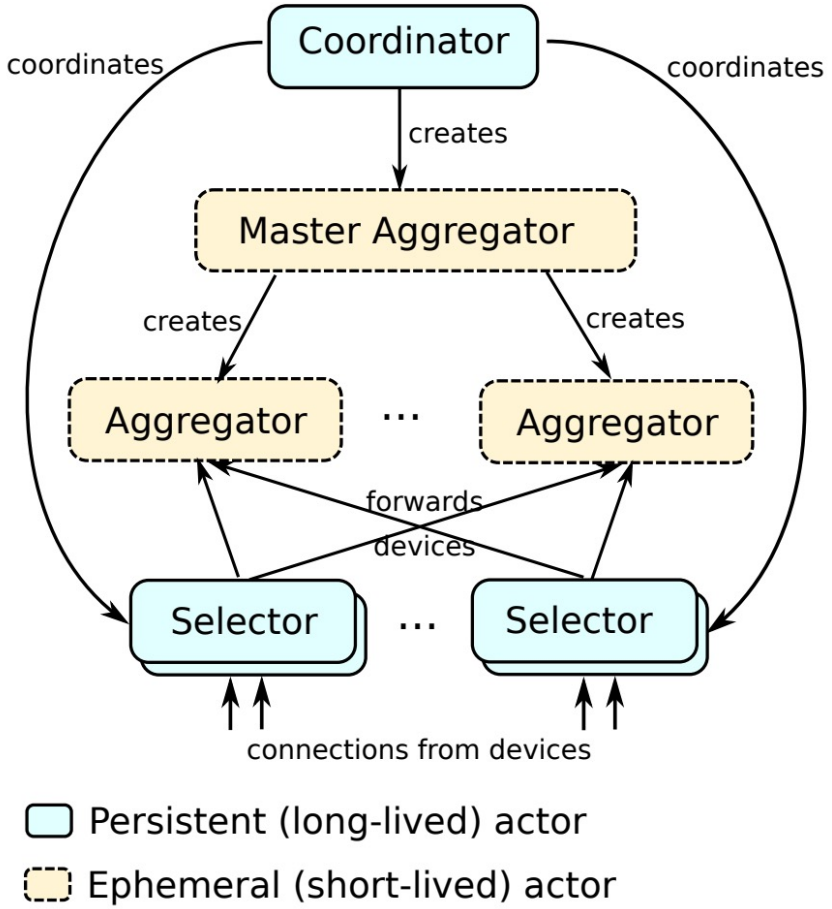
- When a device finishes training, it reports an encrypted model update back to the server
  - Apply gradient compression techniques
- Server aggregates updates as they arrive
- Server closes round once seen enough reports
  - Stragglers reporting after the timeout are ignored

Device Architecture



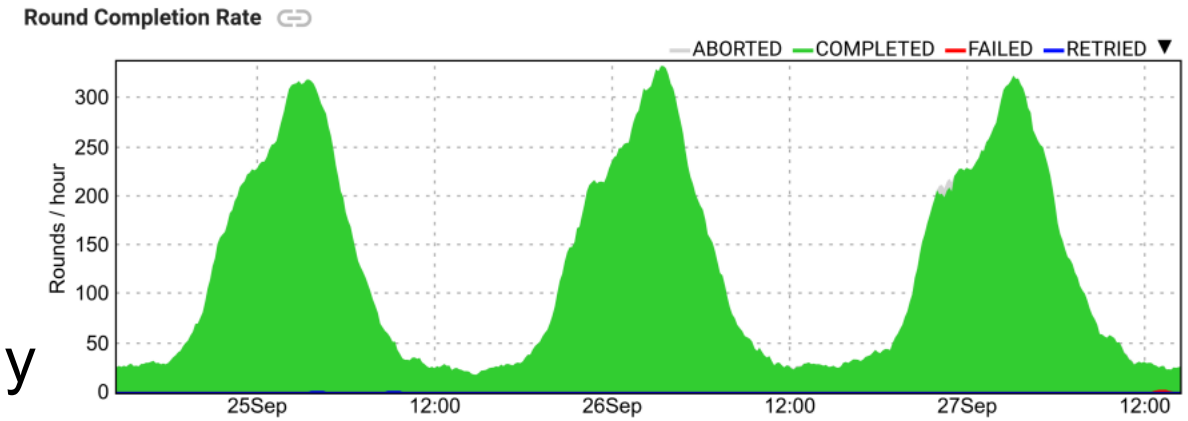
- Application collects and stores on device training examples
- Training is scheduled in separate process as a background job
  - When device state allows (charging, metered network, idle)
  - Training will be interrupted if device conditions change
- Once started, training checks into FL server to obtain model and parameters

Server Architecture



- Coordinator manages a training population (approx. a minibatch)
- Selectors are responsible for accepting and forwarding device connections
  - Dynamically configured by the coordinator based on operational profiles
- Aggregators are spawn when a training round is initiated and process model gradient updates

Pace Steering



Problem: Population size varies drastically

- Devices must be idle, plugged-in, on wi-fi to participate
- Device availability correlates with geo location

Solution: Pace Steering

- When a device is rejected for a training round, it gets back a suggested time window when to retry
- Server uses retry windows to steer the population pace

Questions to Discuss

- How to use FL to train large models that cannot fit on edge devices (e.g., GPT models)? (Hint: model distillation, mixture-of-experts, etc)

- Compare federated learning with traditional DNN training in the following aspects:
  - training data distribution
  - data availability
  - edge device reliability
  - primary bottleneck

Federated Learning v.s. Conventional DNN Training

Data locality and distribution

- Massively decentralized, naturally arising non-IID partition
- Centralized training, balanced data

Data availability

- Limited availability, time-of-day variations
- Almost all data nodes always available

Device statefulness

- Stateless (generally no repeat computation)
- Stateful

Device reliability

- Unreliable
- Reliable

Distribution scale

- Massively parallel (up to billions of devices)
- Single datacenter

Primary bottleneck

- Communication
- computation

* Slide adapted from Brendan McMahan