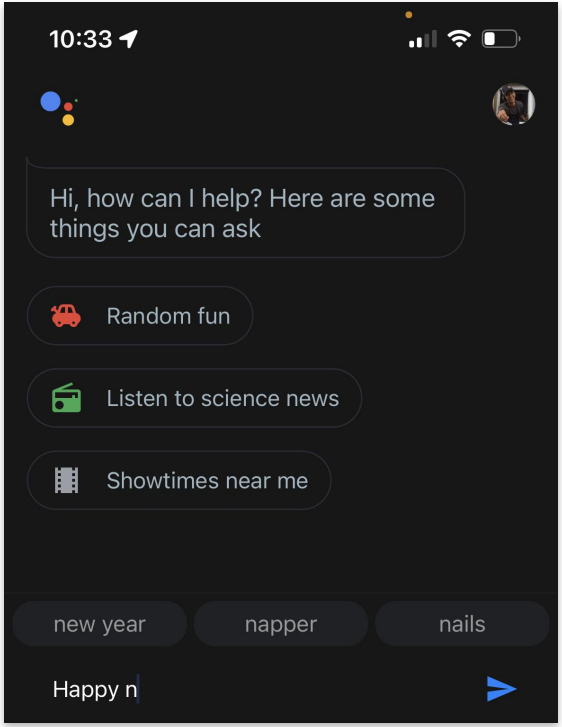
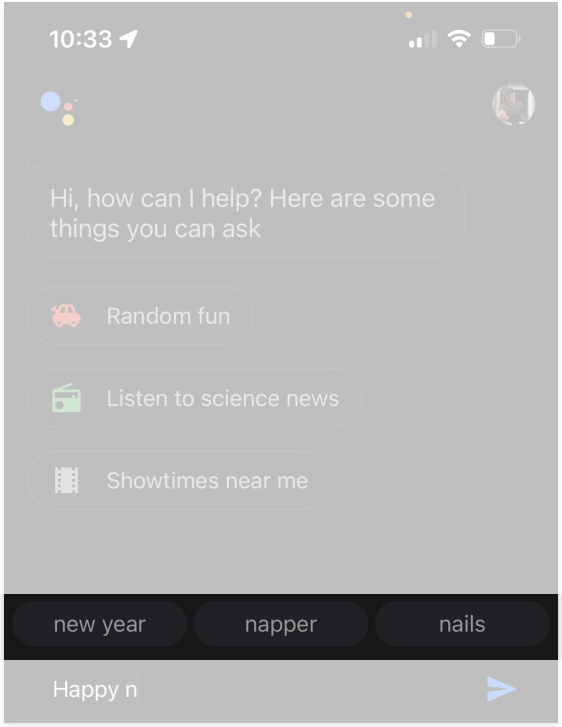
Continuous Monitoring with Federated ML

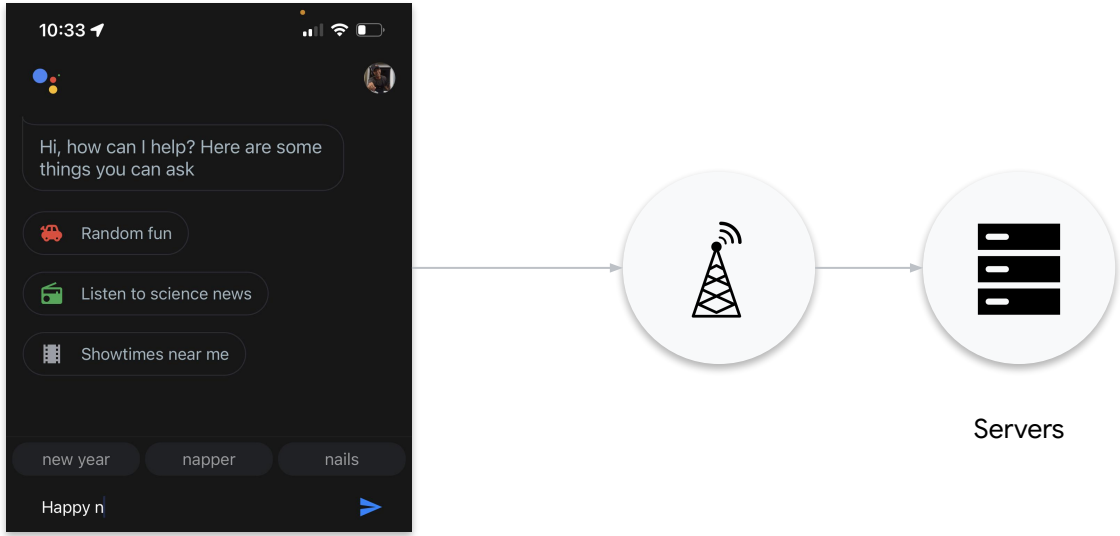
GBoard Example



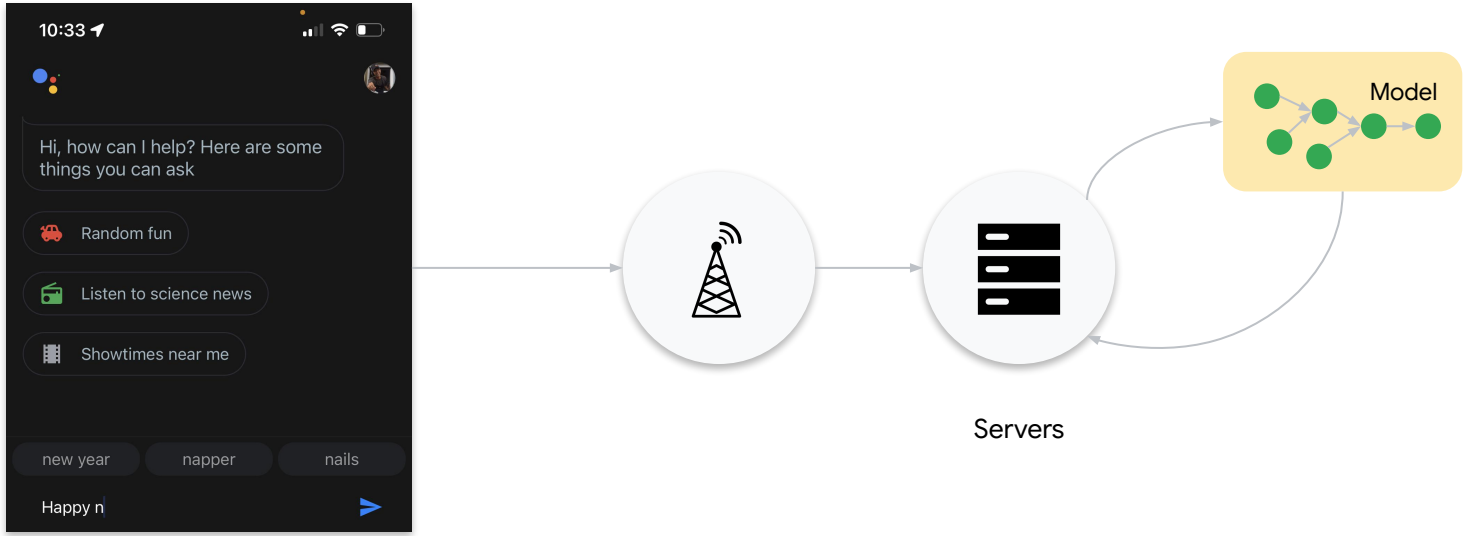
GBoard Example



How do we realize GBoard with MLaaS



How do we realize GBoard with MLaaS



How can **privacy be preserved**?

- **Minimize**
  - Avoid collecting unnecessary data, and dispose or delete data periodically
- **Protect**
  - Use encryption techniques to protect data
- **Map the flow of information**
  - Context, the type of information, and who has access
- **Informed consent**
  - Be transparent with users about how their data is being collected and used

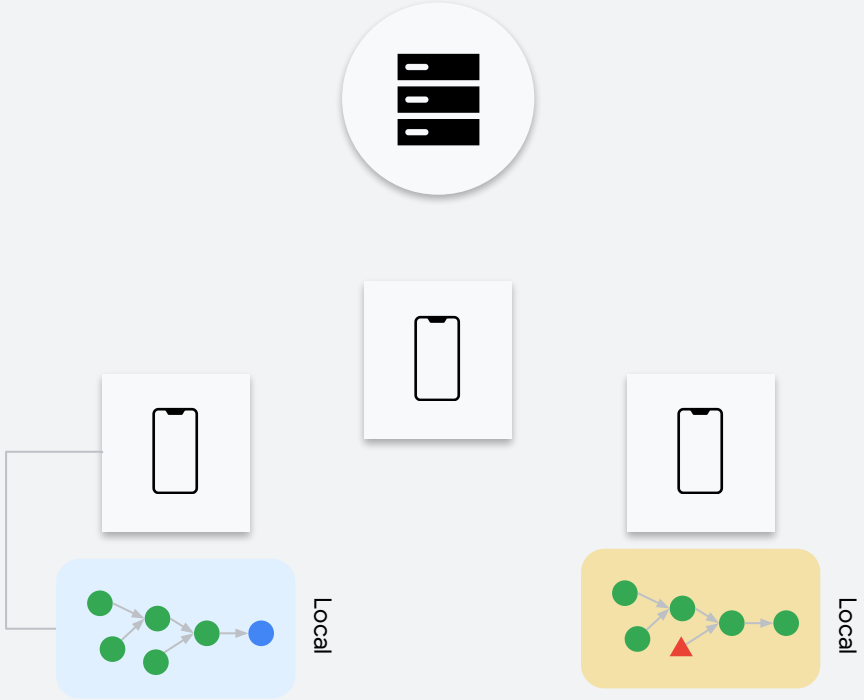
Alternative approach?

- The key flaw behind the prior approach to training is the sending of **raw client data** to central server
  - Server has access to raw client data, exposing clients to intrusion of privacy by central server

Central Servers

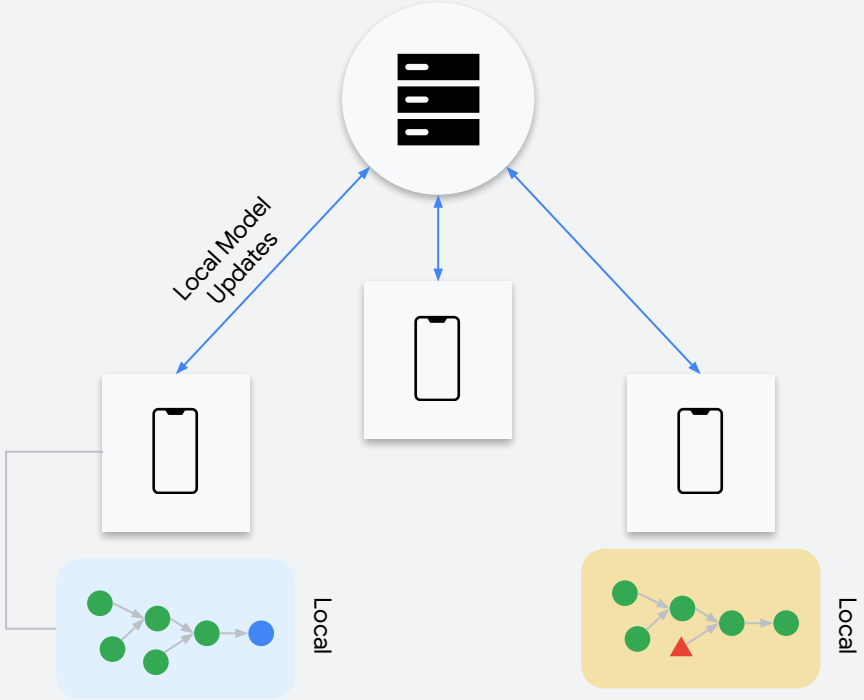
Federated ML

- **Data is kept local** to the endpoint device (data does not ever leave the device)



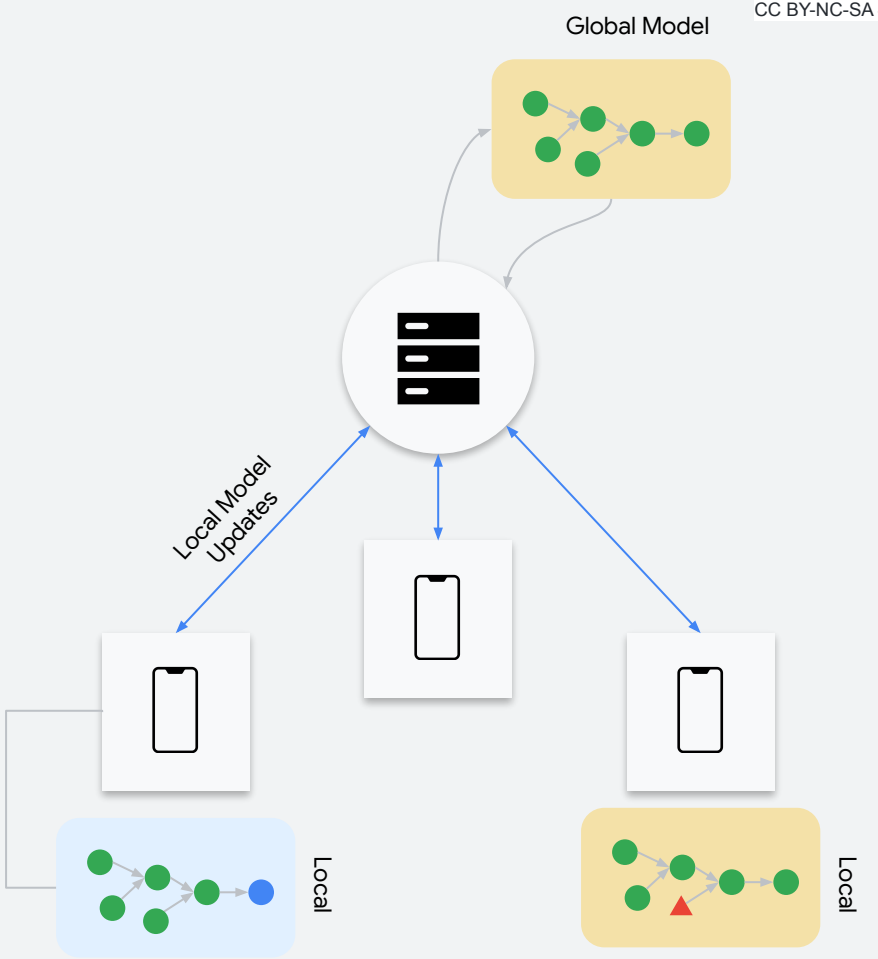
Federated ML

- **Data is kept local** to the endpoint device (data does not ever leave the device)
- **Only** local **model updates** are set to the central server



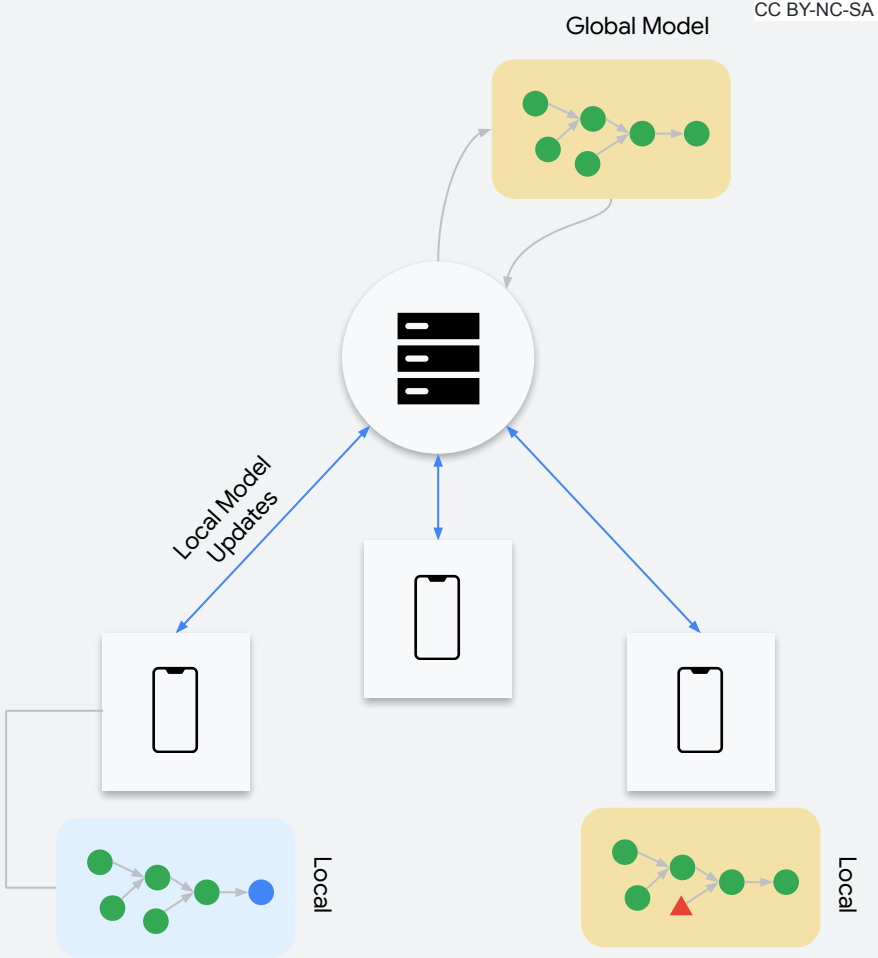
Federated ML

- **Data is kept local** to the endpoint device (data does not ever leave the device)
- **Only** local **model updates** are set to the central server
- Server creates a **global model** and sends it to the endpoints



Why is Federated ML **Useful**?

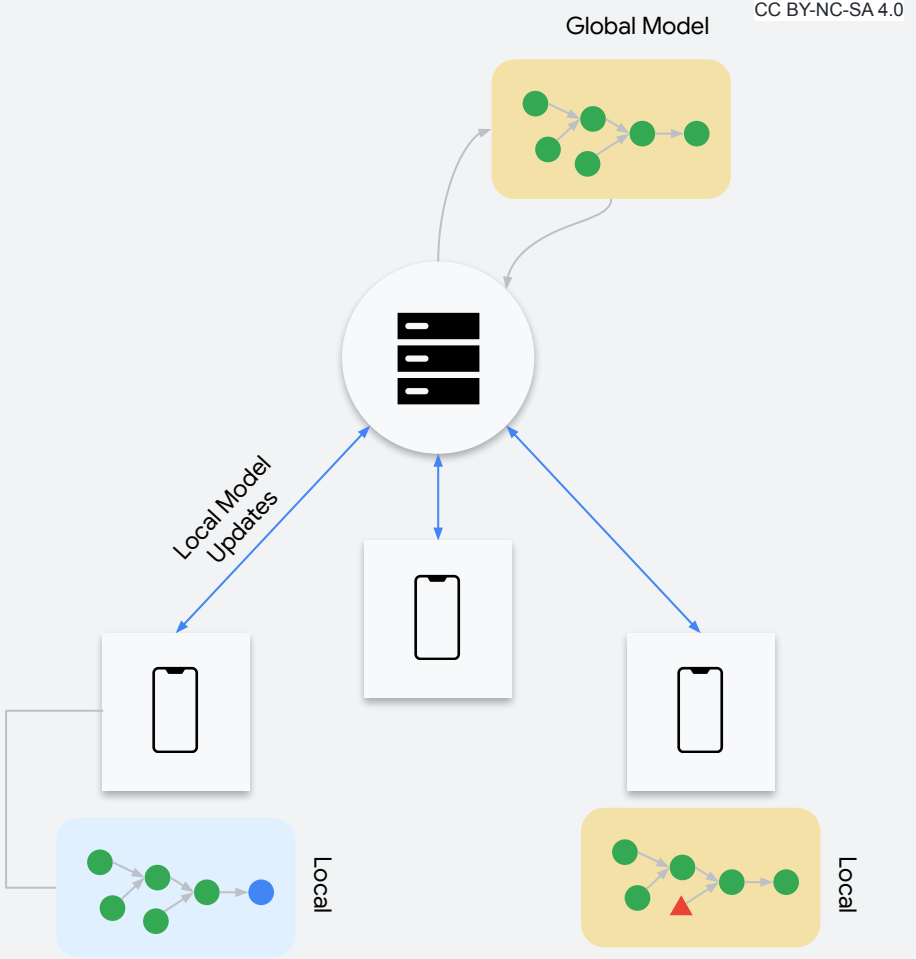
Hyper-Personalized



Why is Federated ML **Useful**?

Hyper-Personalized

Minimum Latencies

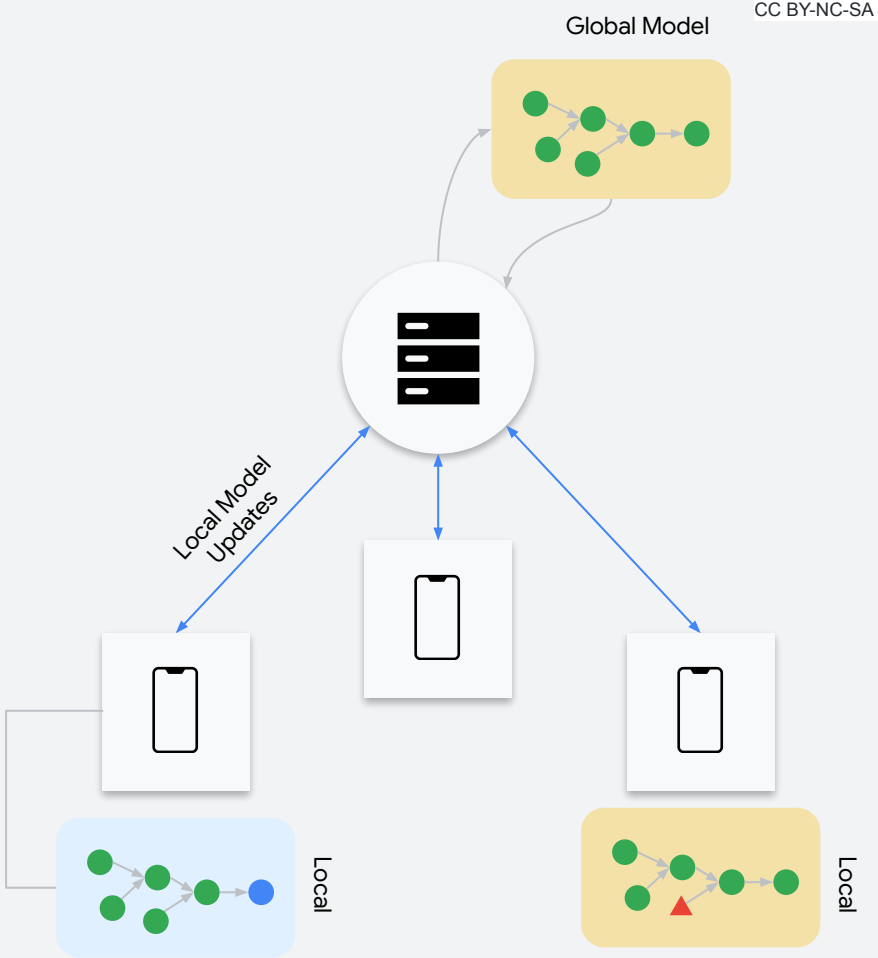


Why is Federated ML **Useful**?

Hyper-Personalized

Low Cloud Infra Overheads

Minimum Latencies



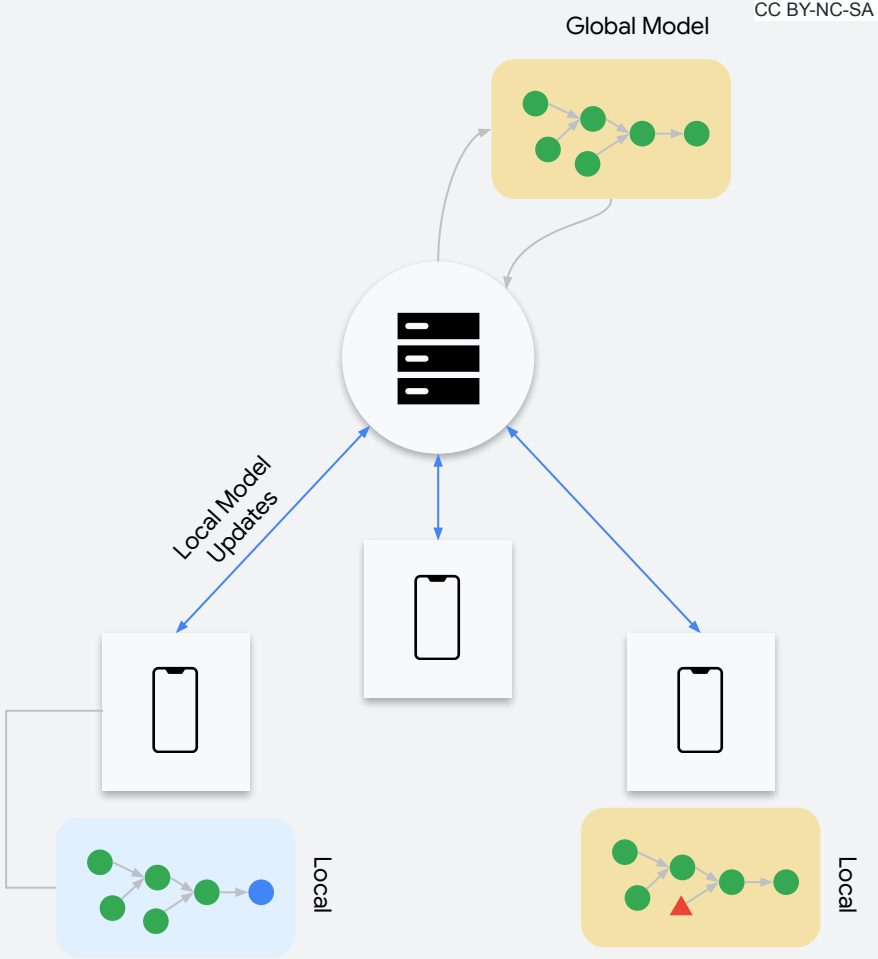
Why is Federated ML **Useful**?

Hyper-Personalized

Low Cloud Infra Overheads

Minimum Latencies

Privacy Preserving



No Free Lunch

Unbalanced training samples

Need a high # of endpoint devices/clients

Slow and unreliable network connections

No Free Lunch

Unbalanced training samples

Need a high # of endpoint devices/clients

Slow and unreliable network connections

No Free Lunch

Unbalanced training samples

Need a high # of endpoint devices/clients

Slow and unreliable network connections

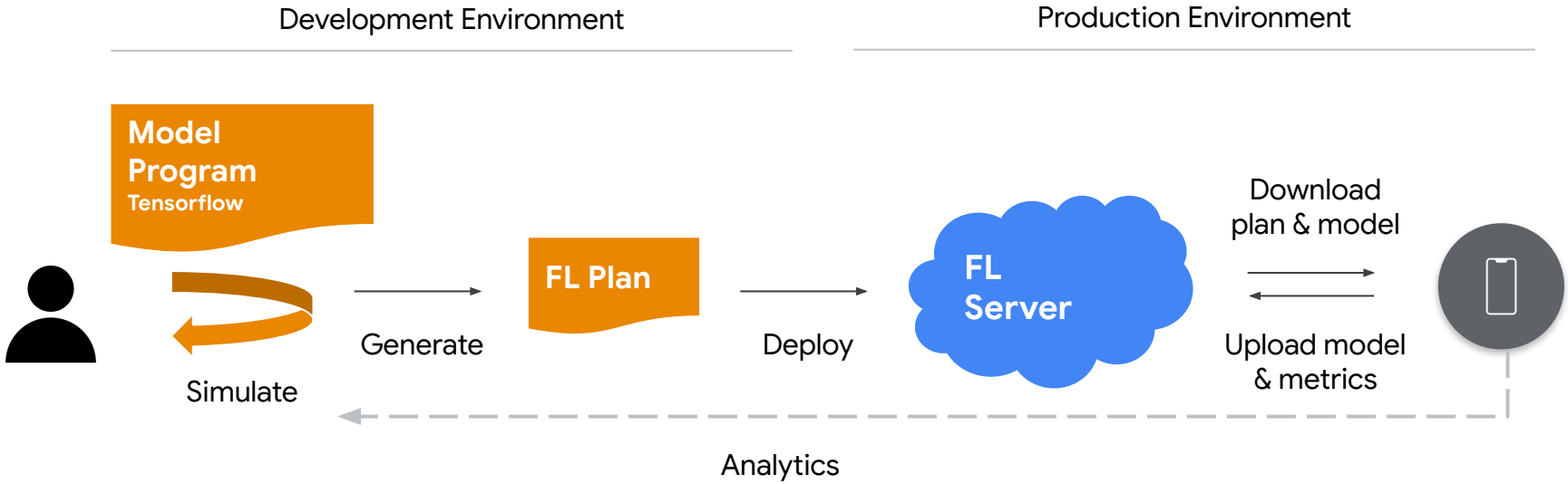
No Free Lunch

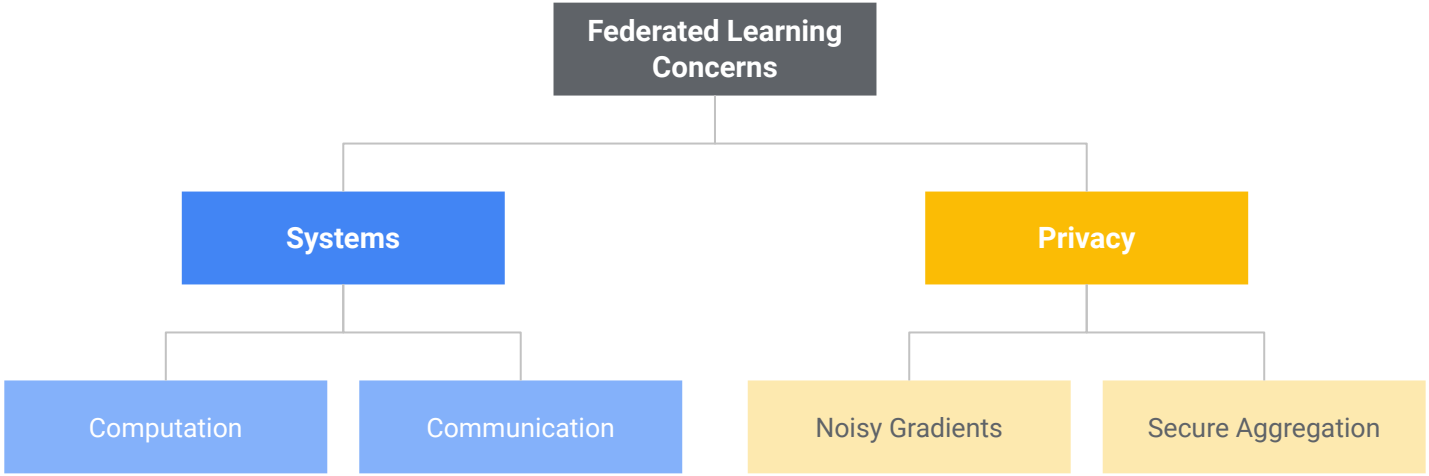
Unbalanced training samples

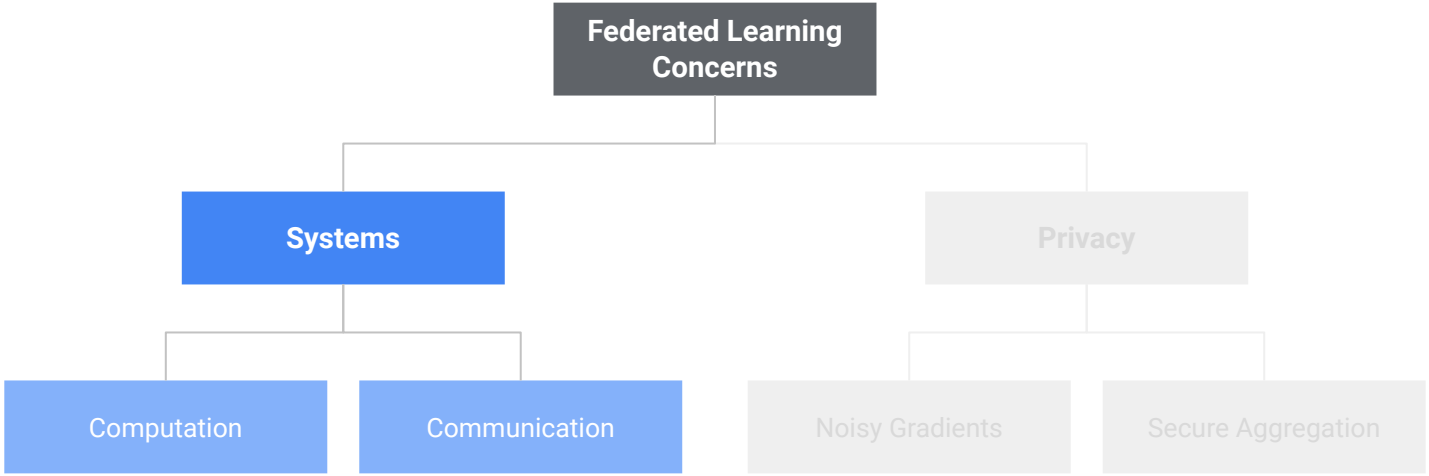
Need a high # of endpoint devices/clients

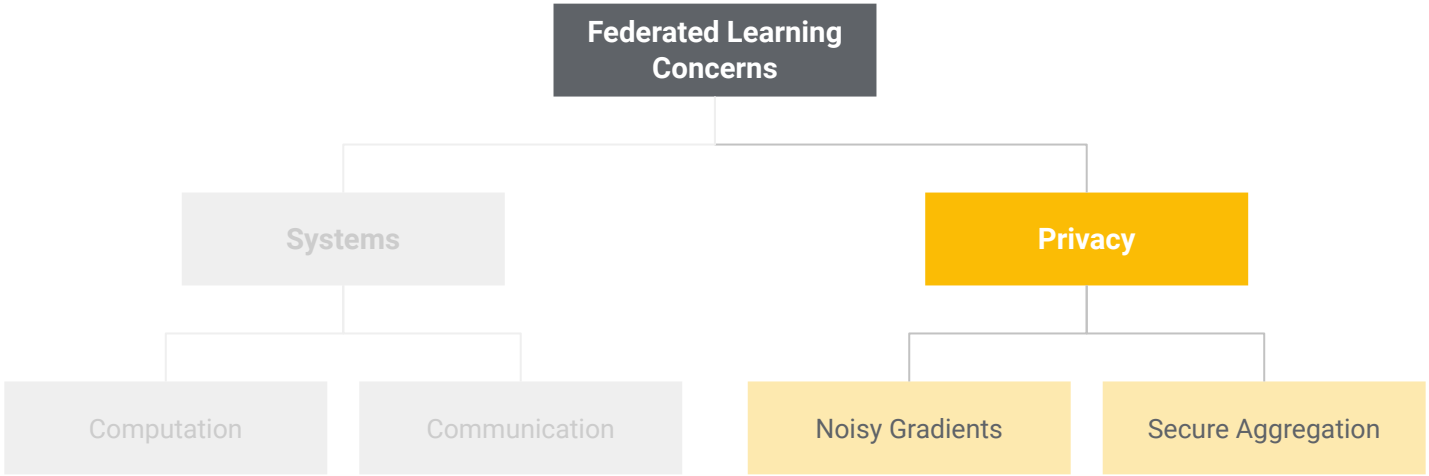
Slow and unreliable network connections

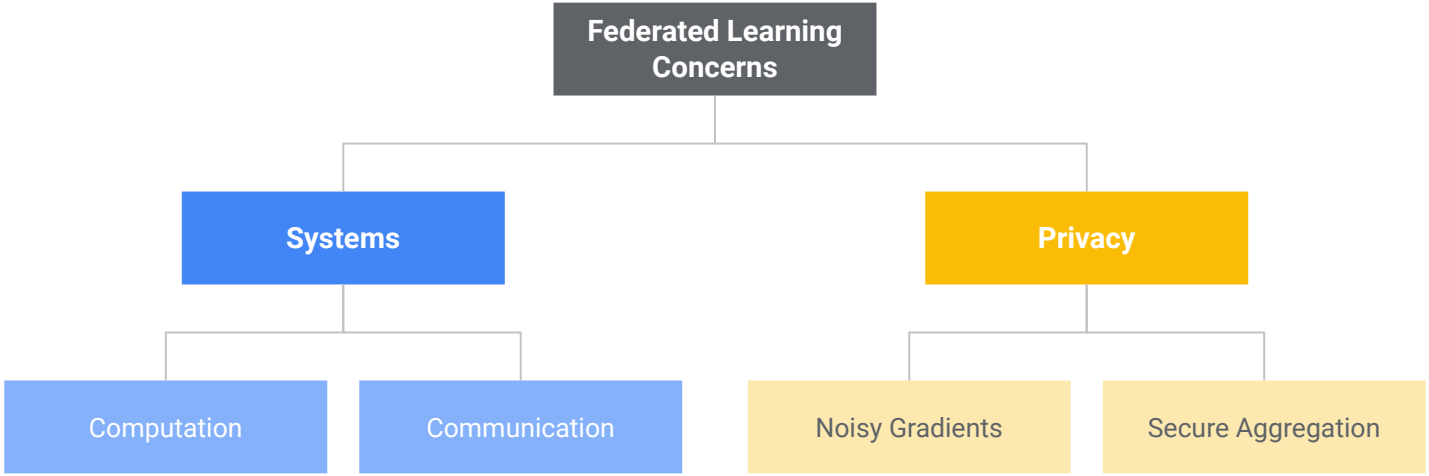
Federated Learning & MLOps











The MLOps **Personas**

ML Engineer

ML Researcher

Data Scientist

Data Engineer

Software Engineer

DevOps

Business Analyst

Traditional Machine Learning as a Service

- Send all of the raw clients' **user data** to the server

Servers

Traditional Machine Learning as a Service

- Send all of the raw clients' **user data** to the server

- All the ML model training is done in the remote cloud ***datacenters***

Servers

Traditional Machine Learning as a Service

- Key concern with sending raw data to the server: **Privacy**

Servers

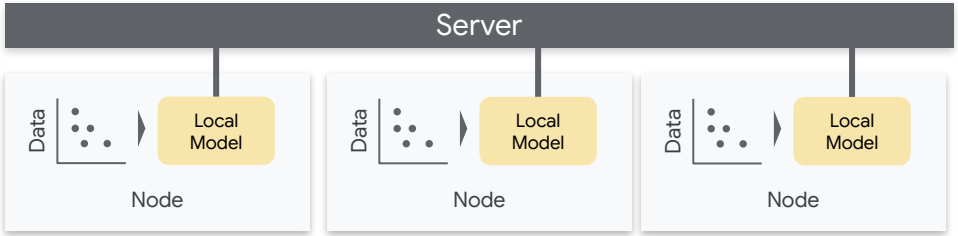
Traditional Machine Learning as a Service

- Key concern with sending raw data to the server: **Privacy**

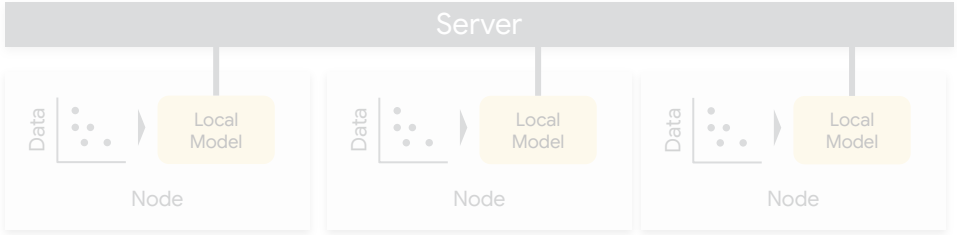
- Exposes user's raw data to the central server, which may potentially be compromised - risking the loss of private data

Servers

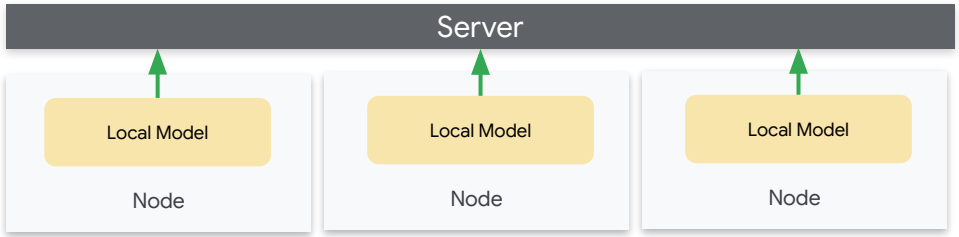
Receive model from server, start training.



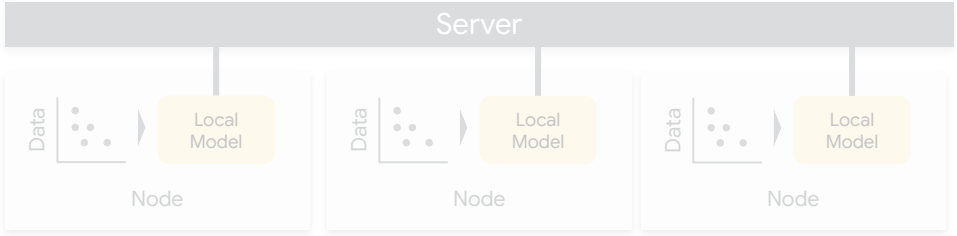
Receive model from server, start training.



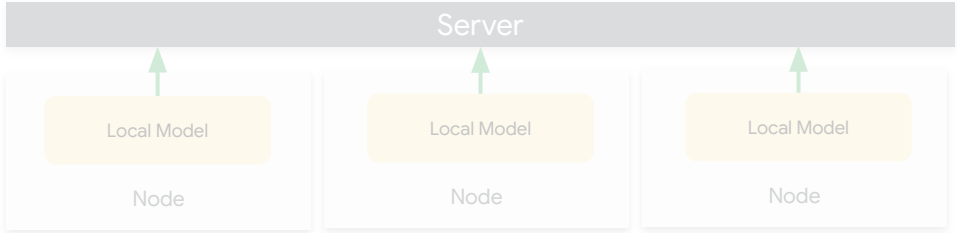
Partially trained models → server



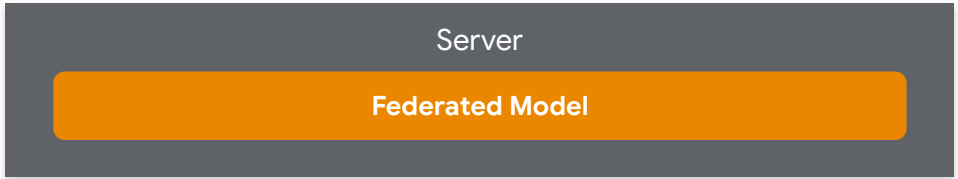
Receive model from server, start training.

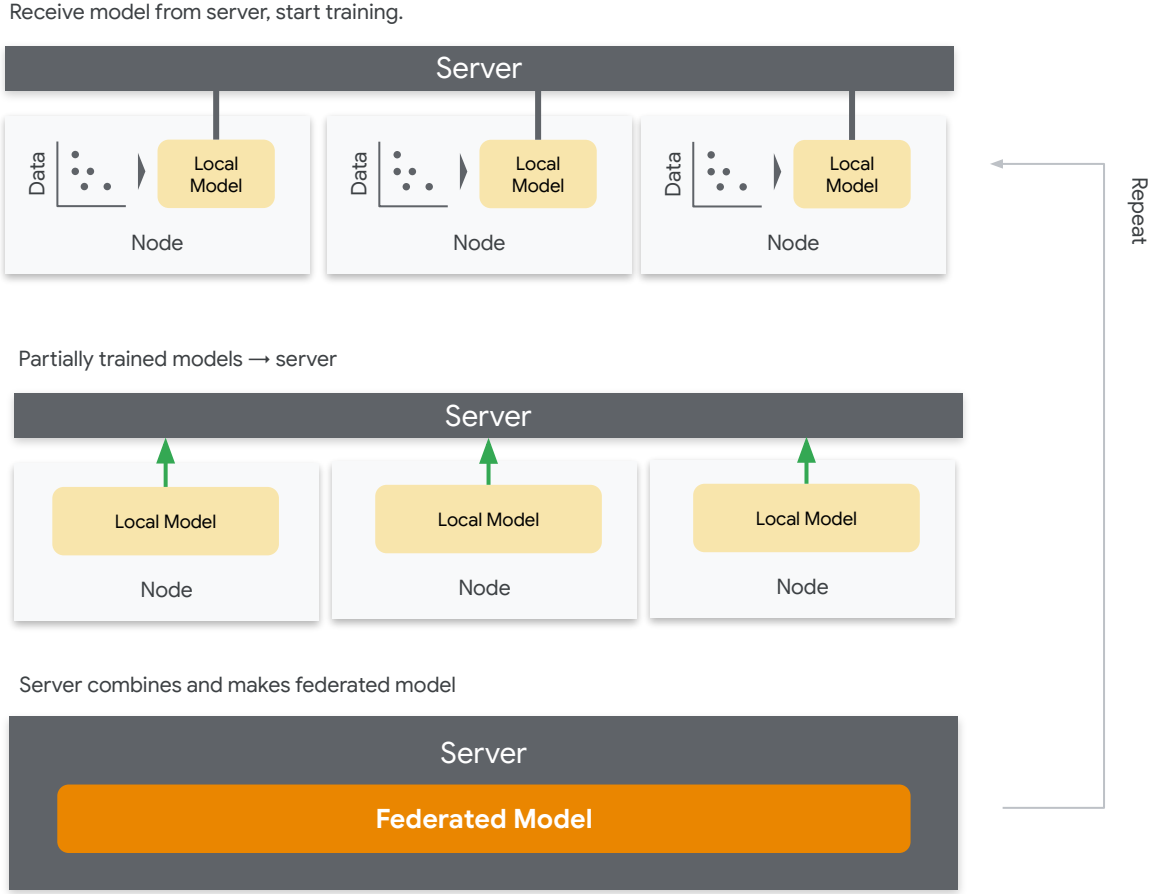


Partially trained models → server



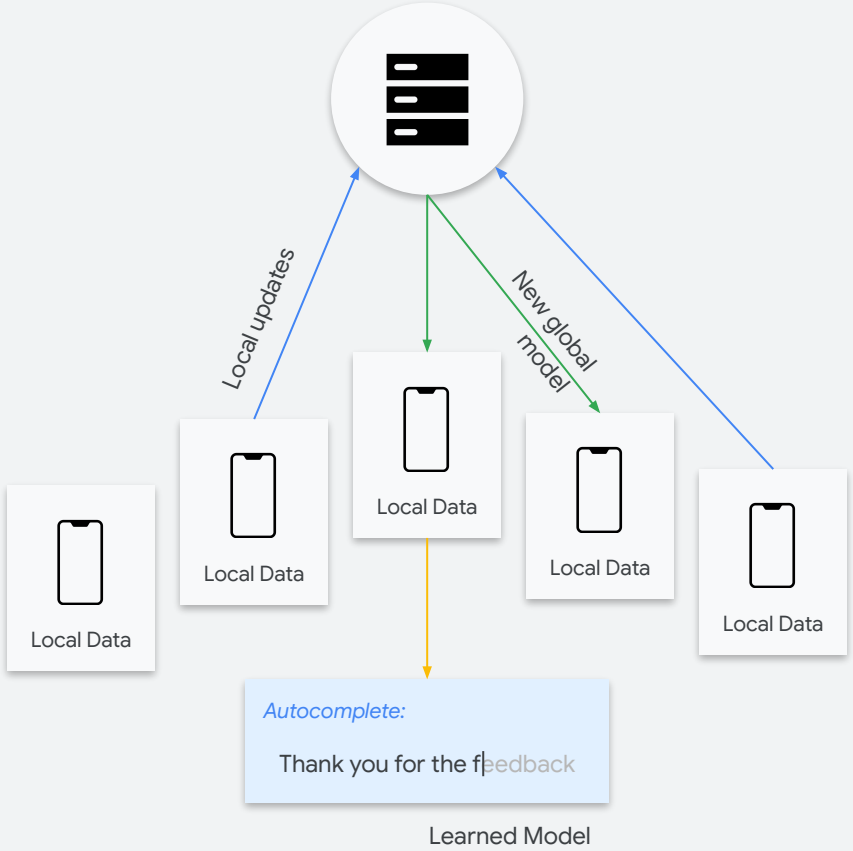
Server combines and makes federated model





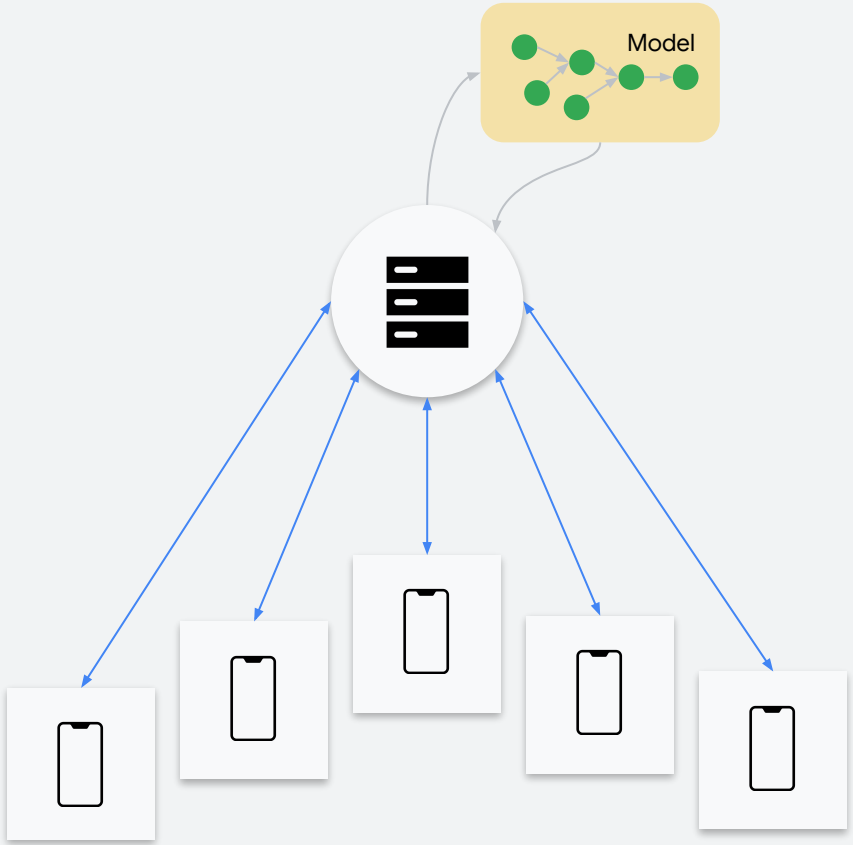
Federated Learning

- A method to ***collaboratively*** learn a shared model while keeping data on device



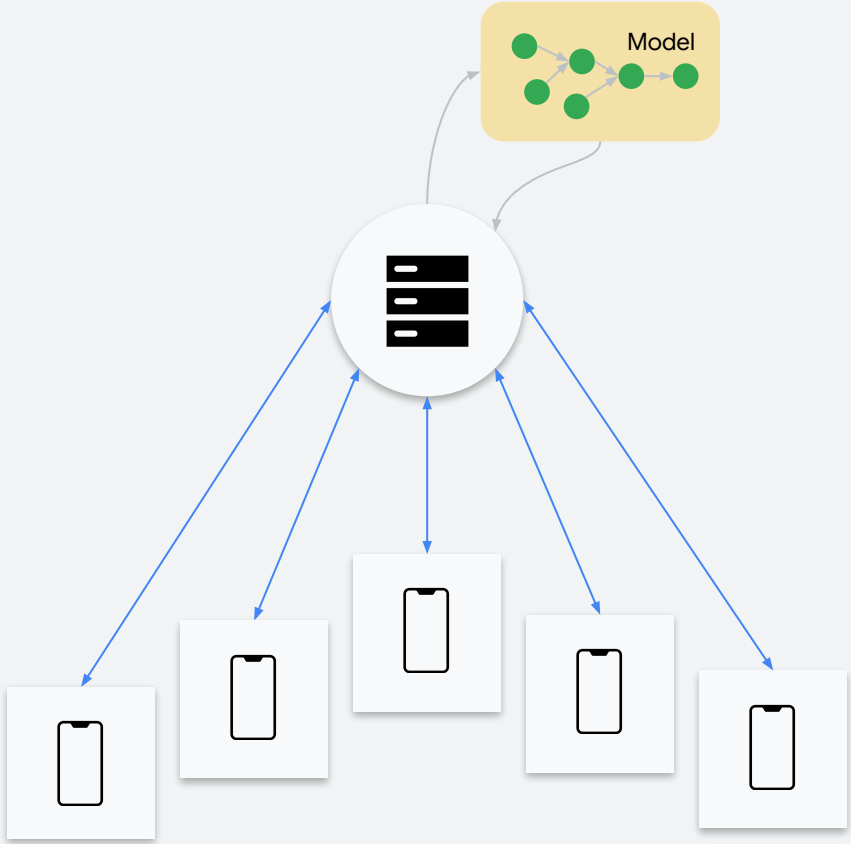
Traditional ML

- Data is **aggregated** from different sources at the server



Traditional ML

- Data is **aggregated** from different sources at the server
- **Central server** builds the machine learning model



Traditional ML

- Data is **aggregated** from different sources at the server
- **Central server** builds the machine learning model
- Central server **distributes the global model** to everyone