CS249r: Learning on the Edge

Oct 16th

Goals for today

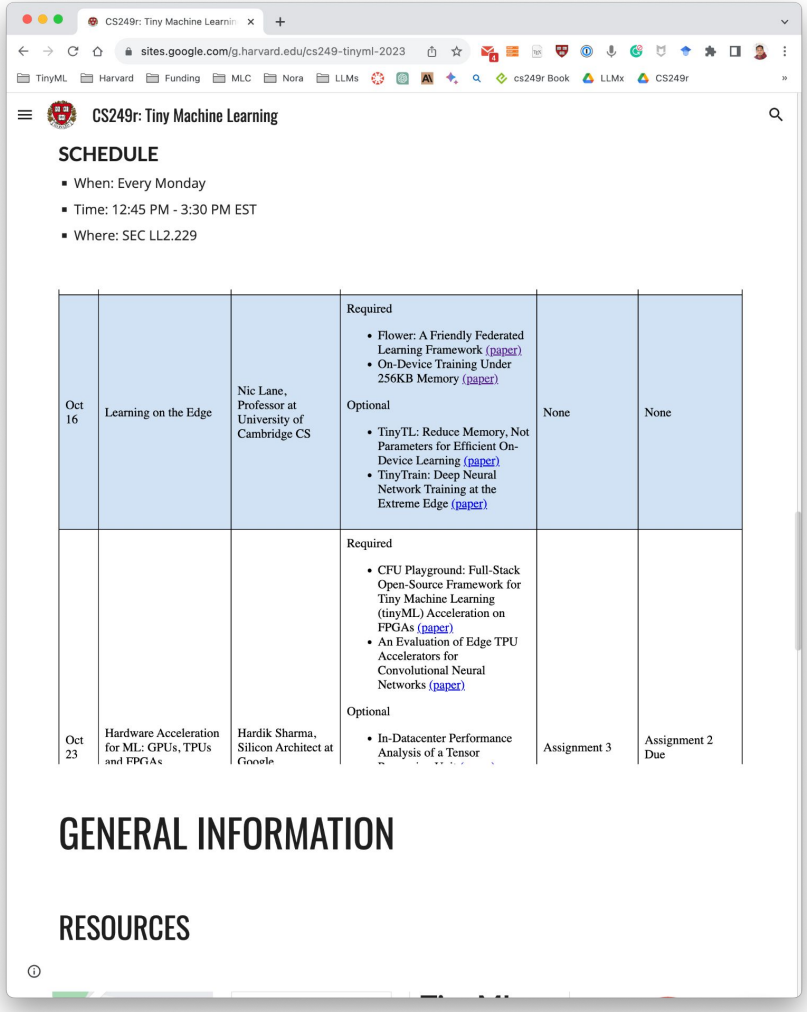
1. Course logistics
2. Lecture - Challenges with on-device training/learning/updates
3. Paper discussions
4. Guest speaker

Course Logistics

Schedule

Please check the website for the latest and most up to date information

A	B	C
	Paper Reading Table Leads	
	MLOps	Hidden Technical Debt
Table 1		Vijay Edupuganti
Table 2		Annie Landefeld
Table 3		Andrew Bass
Table 4		Aghyad Deeb
Table 5		Jared Ni



Assignments Goals

Four assignments

- Assignment 1
- Assignment 2
- Assignment 3
- Assignment 4

Each assignment focuses on different aspects

Want to connect the dots on **why** we picked **what** we picked and **how** you do it

Assignment - Why, What, How

- Assignment 1
  - **Why**: Vision (Camera) use cases + Dataset generation
  - **What**: End-to-end ML workflow
  - **How**: Frameworks (Edge Impulse) + Embedded Hardware (Nicla)
- Assignment 2
  - **Why**: Audio (Microphone) use cases
  - **What**: ML Pipeline (Data preprocessing + Model optimizations)
  - **How**: NAS (Edge Impulse) + Quantization/Pruning (TFLite)
- Assignment 3 - Model development
  - **Why**: Time series (IMU) use cases
  - **What**: ML frameworks programming (Tensorflow/TFLite/TFLM)
  - **How**: Bare metal implementation of end-to-end ML workflow
- Assignment 4 - Responsible AI
  - **Why**: Sustainability + Responsibility
  - **What**: Sustainability-aware system design and safe AI
  - **How**: TinyML Footprint + Adversarial Nibbler

Assignment Schedule Updates

- Assignment 2
  - Due: October 23rd (Monday)
- Mid-Project Review
  - Due: October 30th (Monday)
- Assignment 3
  - Due: November 6th (Monday)
- Assignment 4 Part 1
  - Due: November 20th (Monday)
- Assignment 4 Part 2
  - Due: November 27th (Monday)
- Project Presentations
  - Due: December 4th (Monday)
- Final Report
  - Due: December 11th (Monday)

Project Logistics

Deadlines

- Project Proposal – Oct 10th
- **Mid-Project Presentation (5 slides/3 mins) – Oct 30th**
- Final Project Presentation – Dec 4th
- Final Project Report – Dec 11th

Reviewed the projects

- More application centric, than TinyML engineering centric - that's OK!
- Generally, groups of 1 ~ 3/4

Project Topics

1. Cache 4 Trash
2. Coloring our Thoughts
3. Context Dependent Notification System
4. Decoding Sleep States through Neural Models in Wearable Devices
5. Does regularization during training improve performance of quantized models?
6. Early Wildfire Mapping via TFLite and Image Detection
7. Hand-Tracking for BrilliantLabs Monocle
8. Improving waste disposal practices through smart trash classification
9. Quant-Eyes: Fast and Efficient Quantized Models for On-Device Real-Time Eye Tracking
10. Quantization and Fine-Tuning in IMU-based Gesture Recognition
11. Real-time Visual Narration with tinyCLIP for the Visually Impaired
12. Saving power in ultrasound sensing for tracking neuromuscular function
13. WickWatcher

Waste Management and Environmental Sustainability

- Cache 4 Trash
- Improving waste disposal practices through smart trash classification

Visualization and Digital Art

- Coloring our Thoughts
- WickWatcher

Notifications and Context-aware Systems

- Context Dependent Notification System

Health and Biometric Monitoring

- Decoding Sleep States through Neural Models in Wearable Devices
- Saving power in ultrasound sensing for tracking neuromuscular function
- Hand-Tracking for BrilliantLabs Monocle

Quantization and Model Efficiency

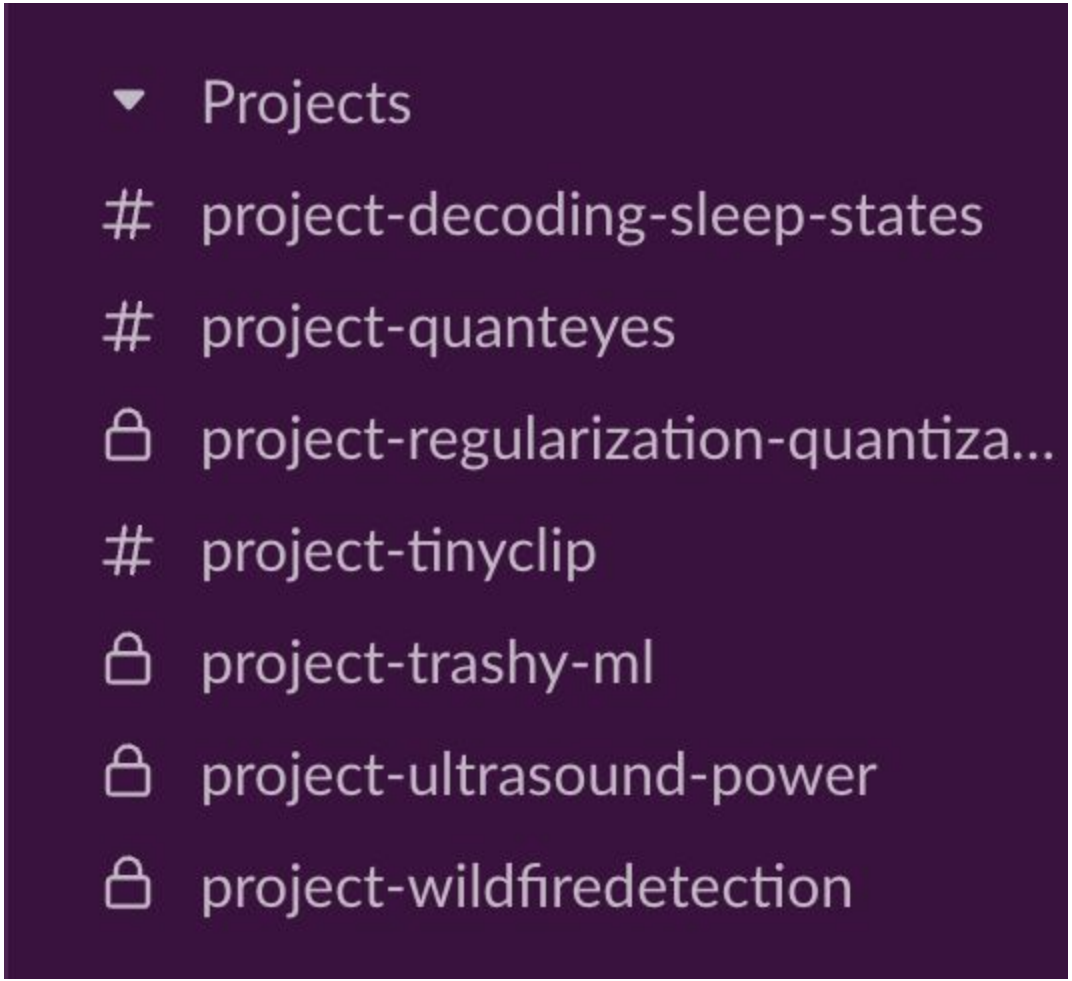
- Does regularization during training improve performance of quantized models
- Quant-Eyes: Fast and Efficient Quantized Models for On-Device Real-Time Eye Tracking:
- Quantization and Fine-Tuning in IMU-based Gesture Recognition

Image Recognition and Processing

- Early Wildfire Mapping via TFLite and Image Detection
- Real-time Visual Narration with tinyCLIP for the Visually Impaired

Projects

- Create a project channel
- Add all staff to the channel
- Update description of the channel w/ project summary



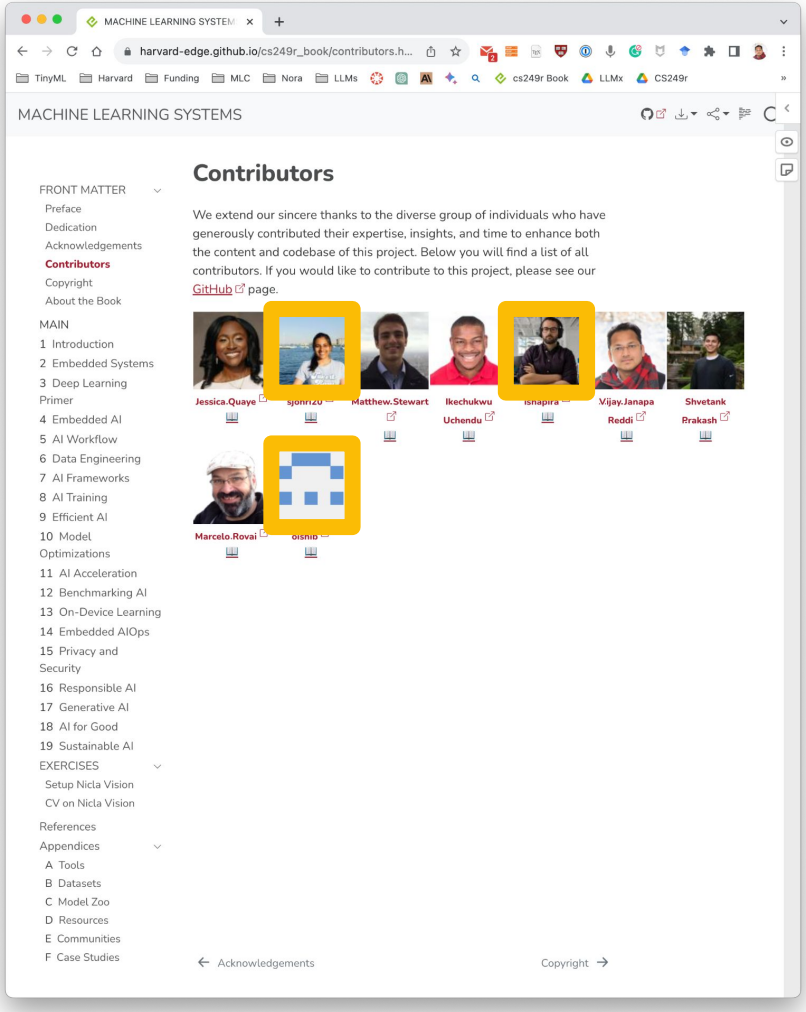
Scribing Updates

Chapters

- Data engineering ✅
- ML frameworks ✅
- Model optimizations
- On-device learning (today!)
    - Come say hi! 👋

New Contributors

- Itai, Oishi, Shreya (Data Engineering) 🥳
- Marcelo - NICLA



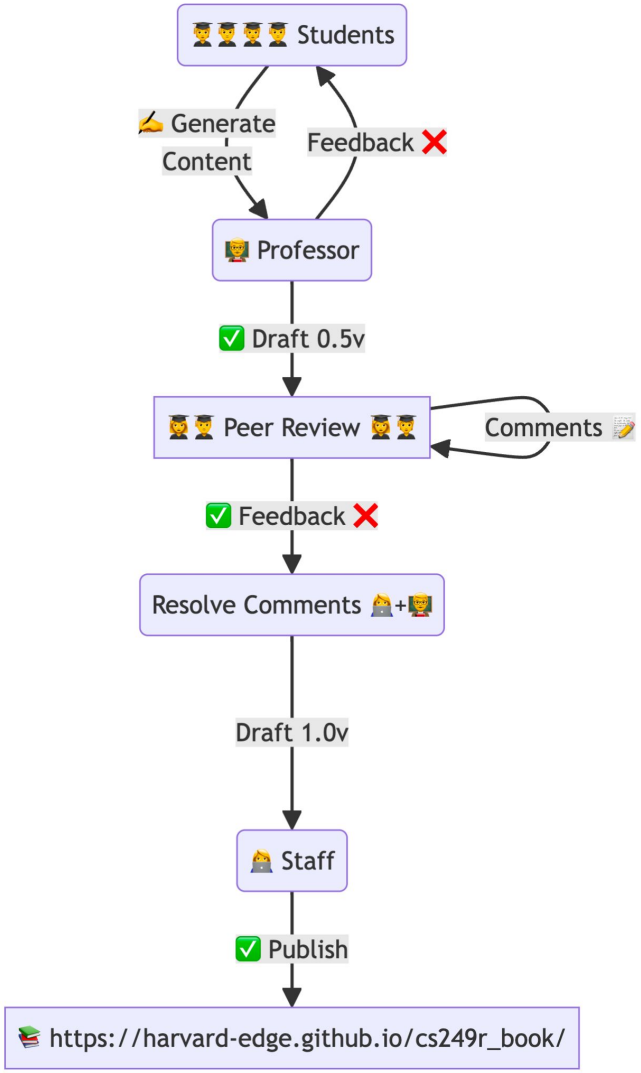
Scribing Updates

General flow for how things are working →

- Collaboratively generate content
- Peer review + feedback + comments
- Resolve comments w/ staff
- Staff merge update

Comments on Implementation

- Put links to figures in Google Doc
- Make sure chapter renders correctly on quarto
- Check figure caption format is the same as other book chapters
- Make sure references render correctly and update refs.bib file—do not have a separate references section at the end of your chapter



Course Topics

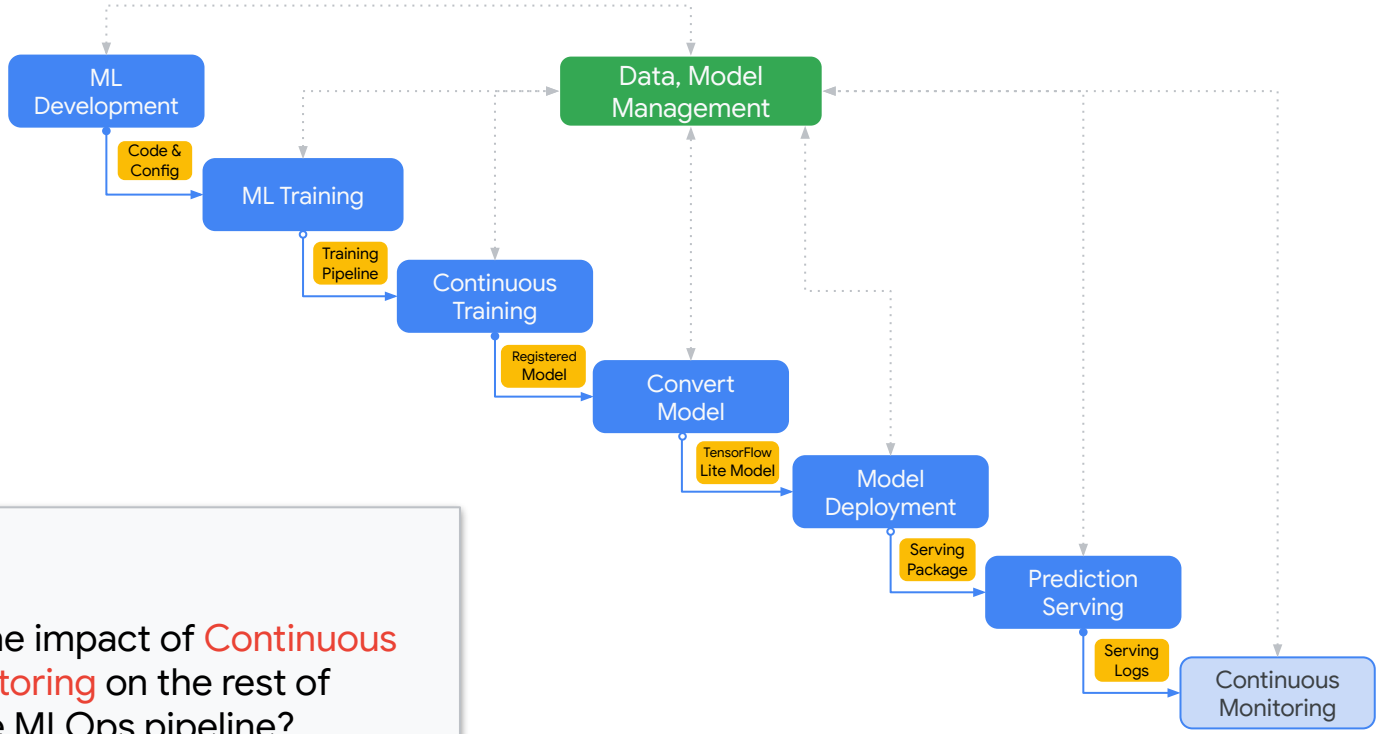
1. Overview and Introduction to Embedded Machine Learning
2. Data Engineering
3. Embedded Machine Learning Frameworks
4. Efficient Model Representation and Compression
5. ~~Performance Metrics and Benchmarking of ML Systems~~
6. **Learning on the Edge**
7. Hardware Acceleration for Edge ML: GPUs, TPUs and FPGAs
8. Embedded MLOps
9. Secure and Privacy-Preserving On-Device ML
10. Responsible AI
11. Sustainability at the Edge
12. Generative AI at the Edge

Attendance Sheet

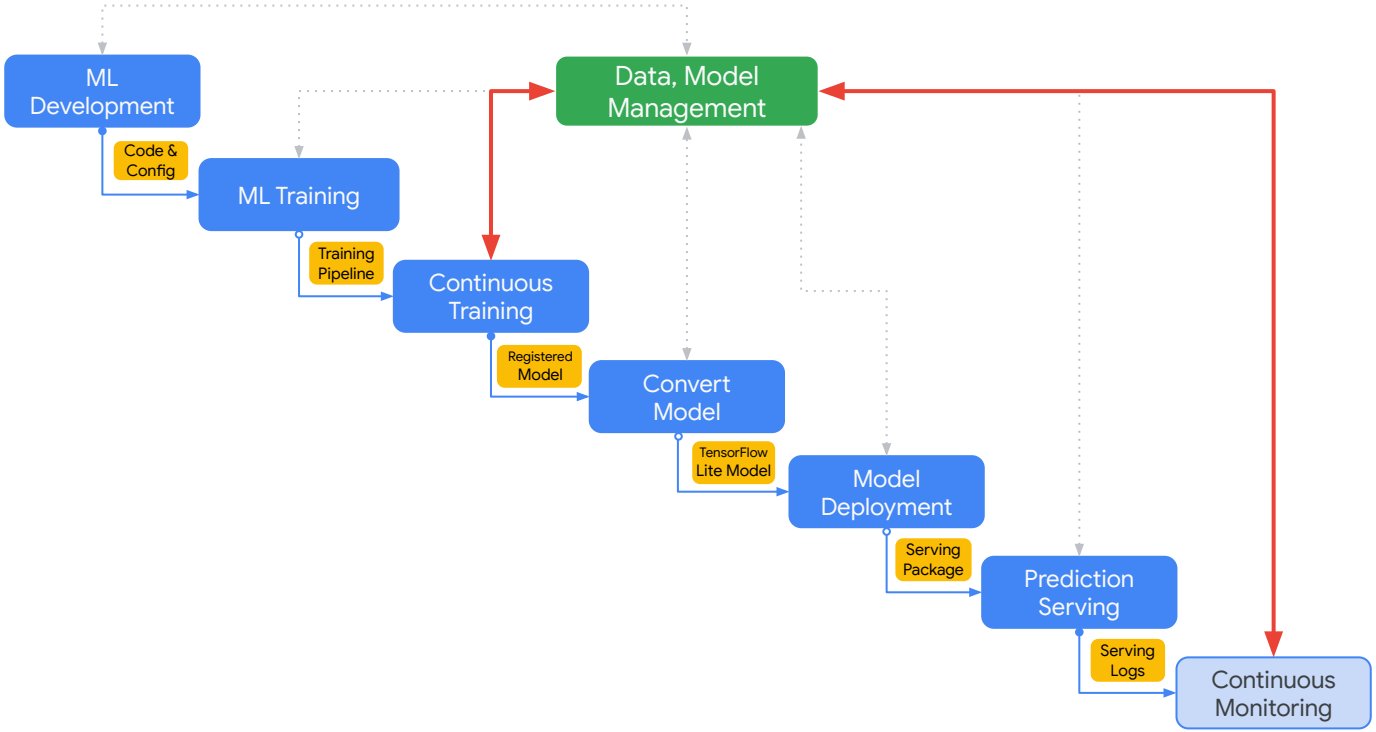
To be filled in during class.

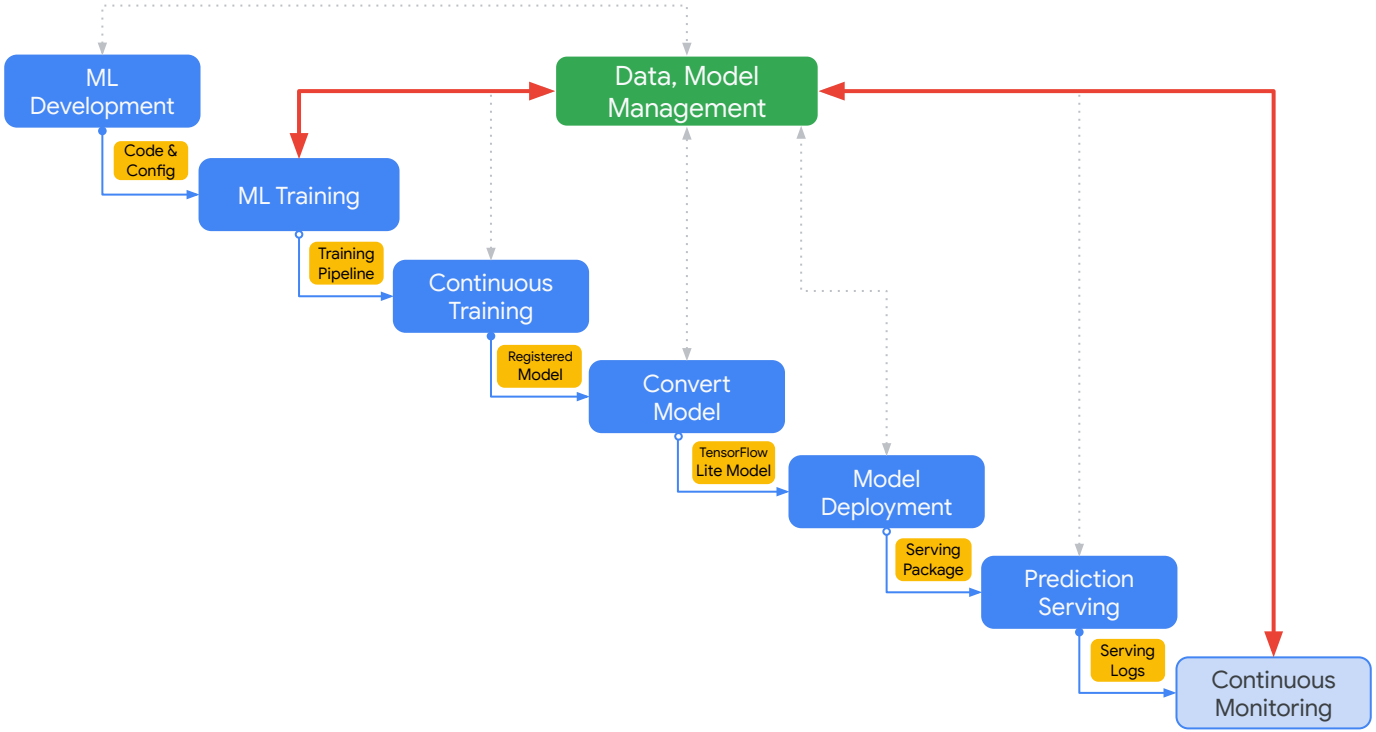
Will be passed around each class.

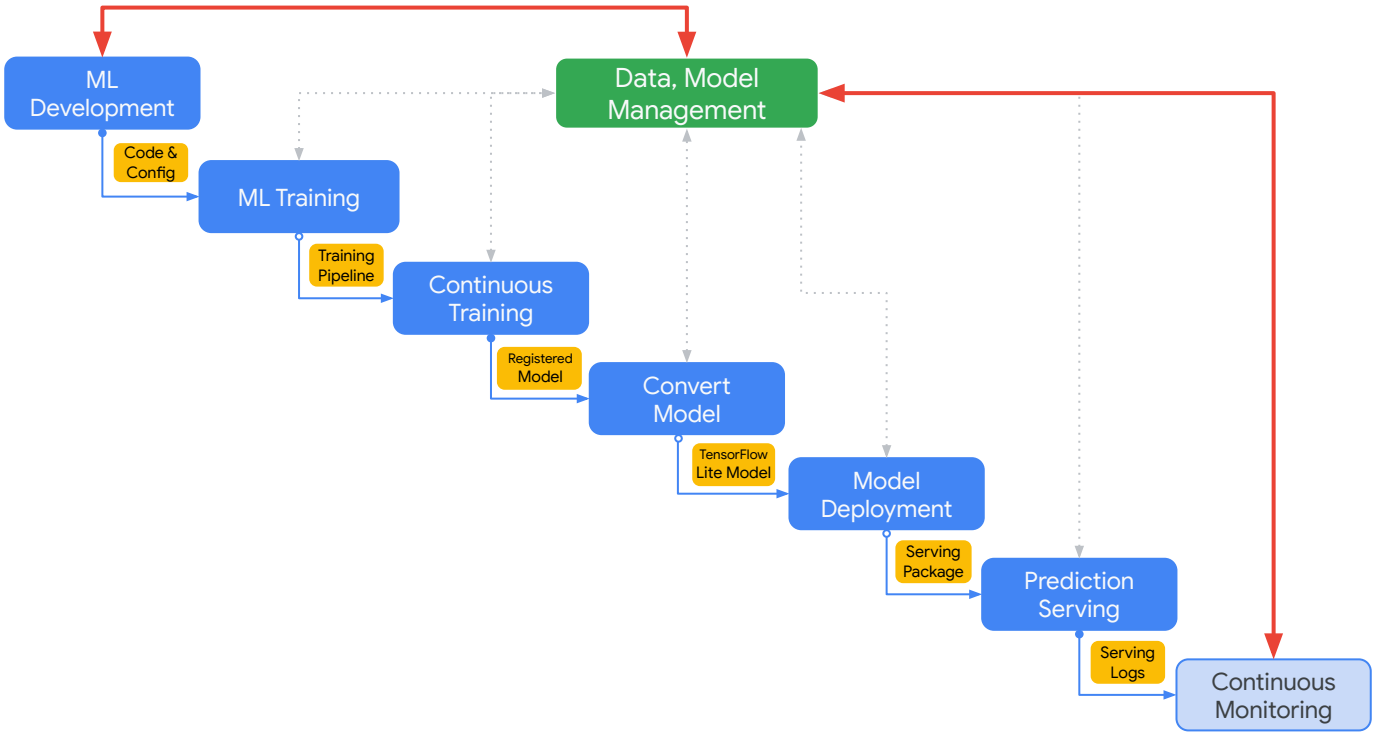
Lecture



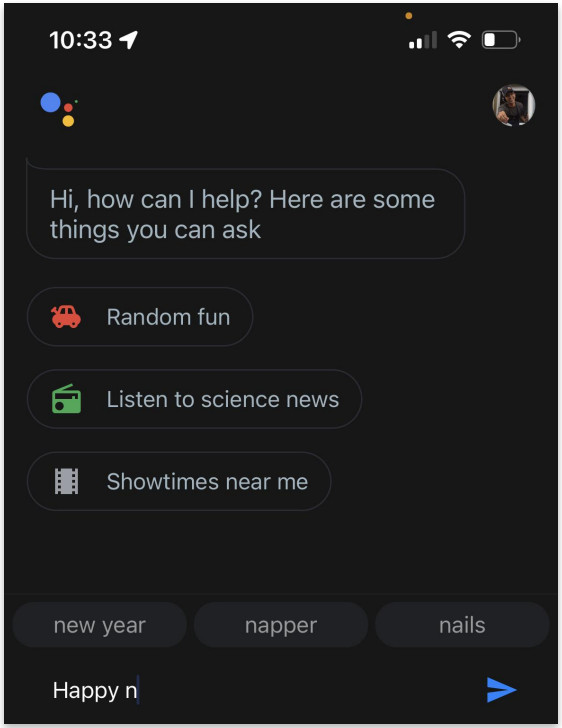
What's the impact of Continuous Monitoring on the rest of the MLOps pipeline?


ML Development
Code & Config
ML Training
Training Pipeline
Continuous Training
Registered Model
Convert Model
TensorFlow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring
Data, Model Management

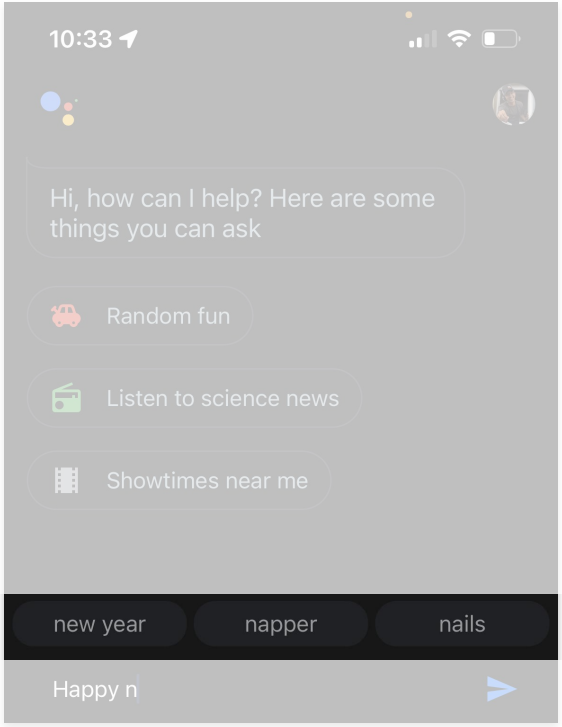

ML Development
Code & Config
ML Training
Training Pipeline
Continuous Training
Registered Model
Convert Model
TensorFlow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring
Data, Model Management


ML Development
Code & Config
ML Training
Training Pipeline
Continuous Training
Registered Model
Convert Model
TensorFlow Lite Model
Model Deployment
Serving Package
Prediction Serving
Serving Logs
Continuous Monitoring
Data, Model Management

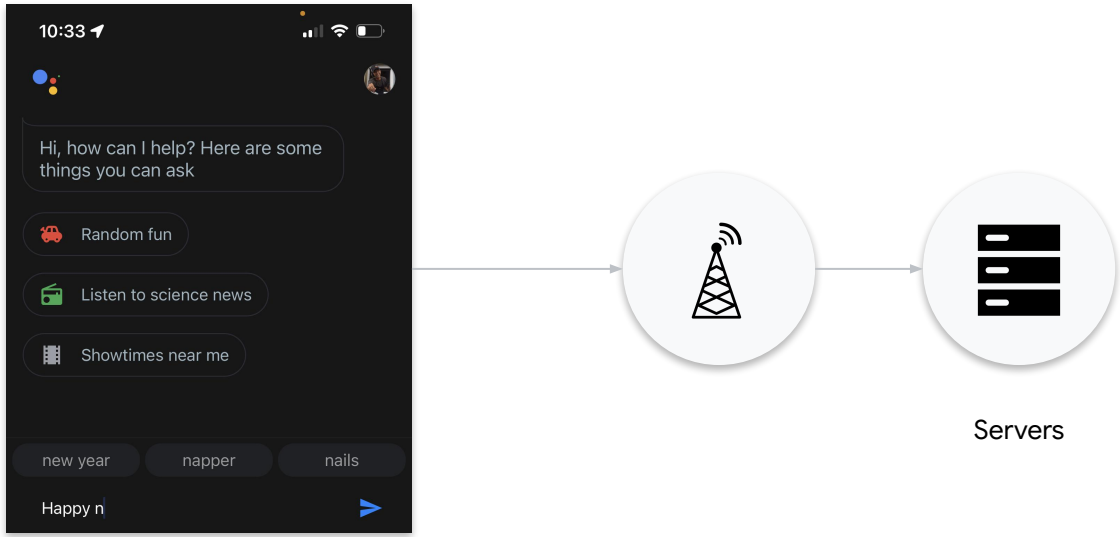
GBoard Example



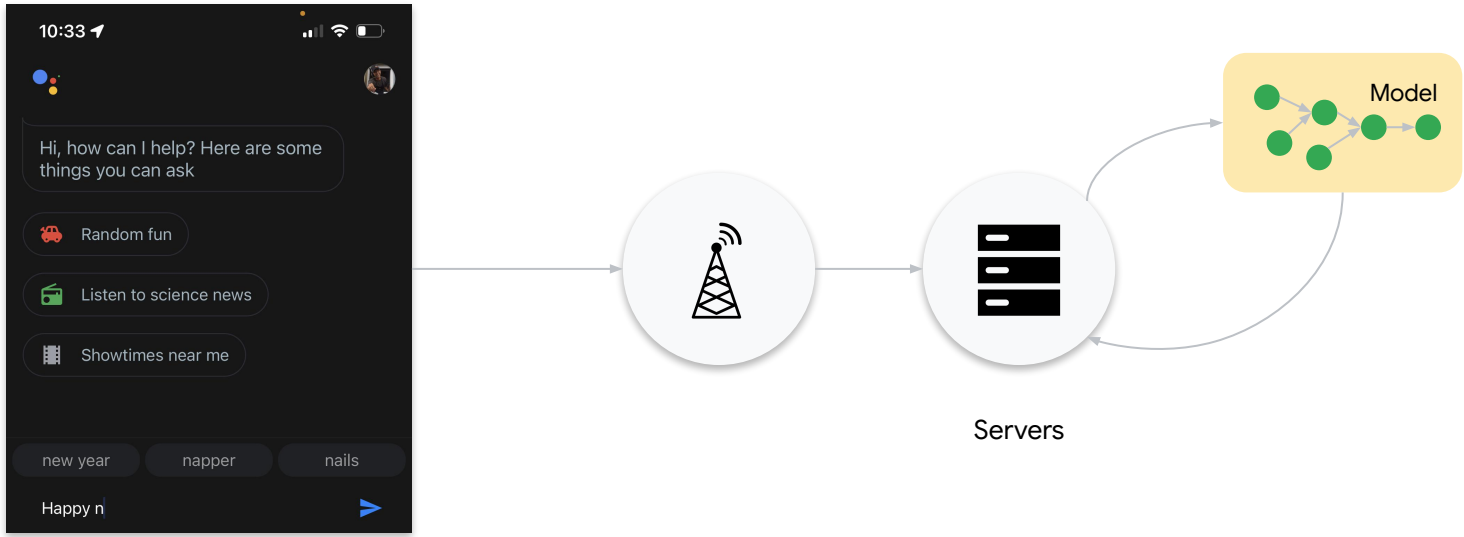
GBoard Example



How do we realize GBoard with MLaaS

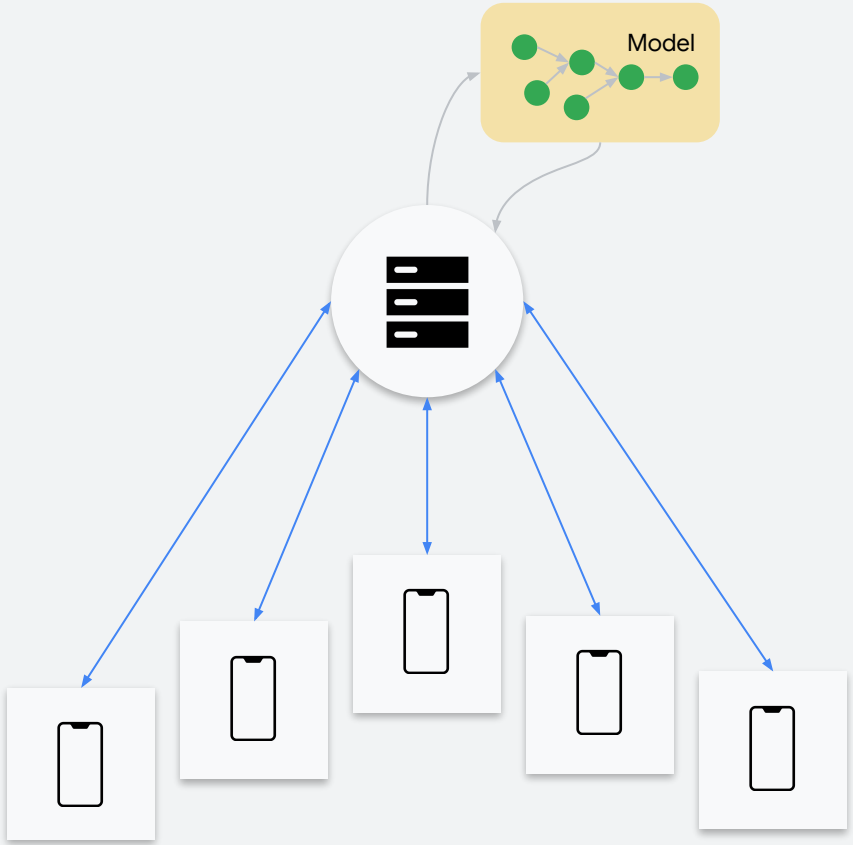


How do we realize GBoard with MLaaS



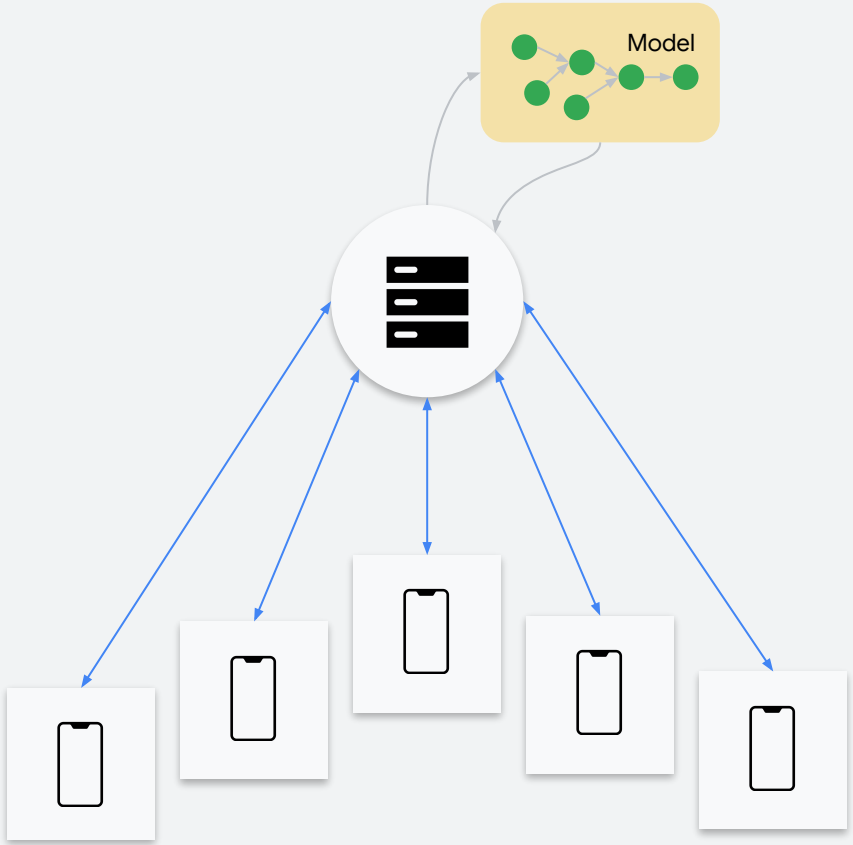
Traditional ML

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



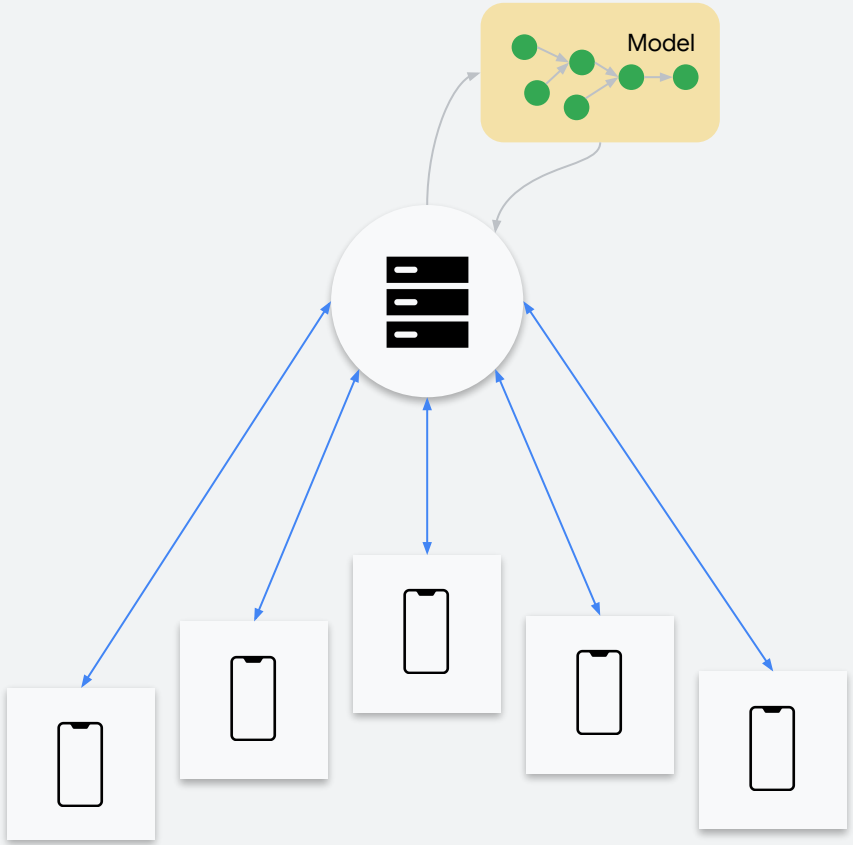
Traditional ML

- 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



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**

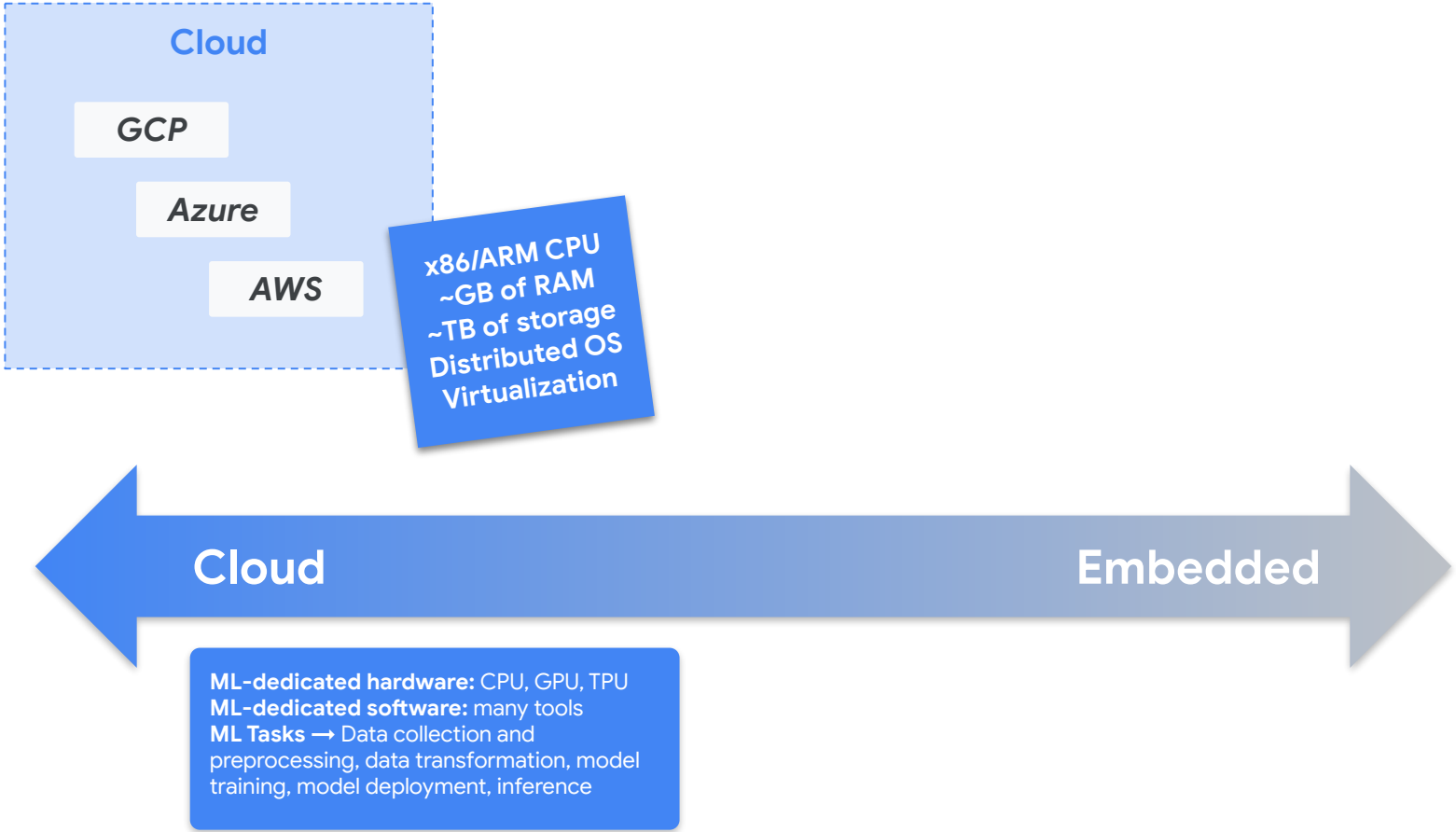
- Exposes user's raw data to the central server, which may potentially be compromised - risking the loss of private data

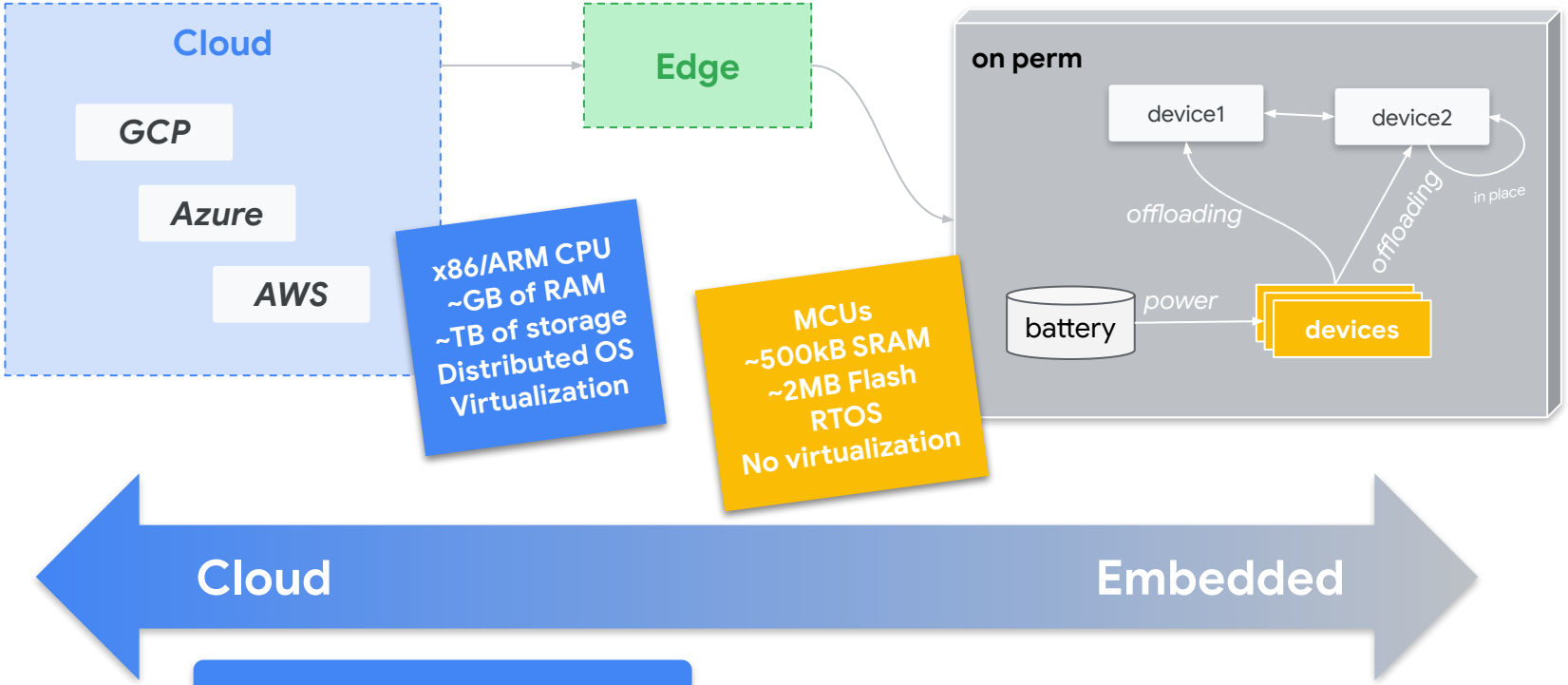
Servers

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


Cloud
Embedded


Cloud
GCP
Azure
AWS
x86/ARM CPU
~GB of RAM
~TB of storage
Distributed OS
Virtualization
Cloud
Embedded
ML-dedicated hardware: CPU, GPU, TPU
ML-dedicated software: many tools
ML Tasks → Data collection and preprocessing, data transformation, model training, model deployment, inference

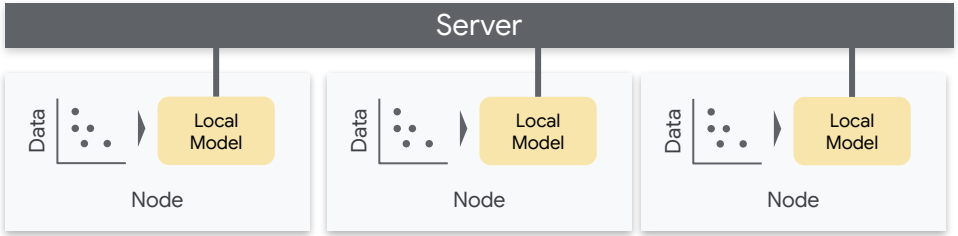

Cloud
GCP
Azure
AWS
Edge
on perm
device1
device2
offloading
offloading
in place
battery
power
devices
x86/ARM CPU
~GB of RAM
~TB of storage
Distributed OS
Virtualization
MCUs
~500kB SRAM
~2MB Flash
RTOS
No virtualization
Cloud
Embedded
ML-dedicated hardware: CPU, GPU, TPU
ML-dedicated software: many tools
ML Tasks ⟶ Data collection and preprocessing, data transformation, model training, model deployment, inference
No ML-dedicated Hardware
No ML-dedicated software
ML Tasks ⟶ Inference

Alternative approach?

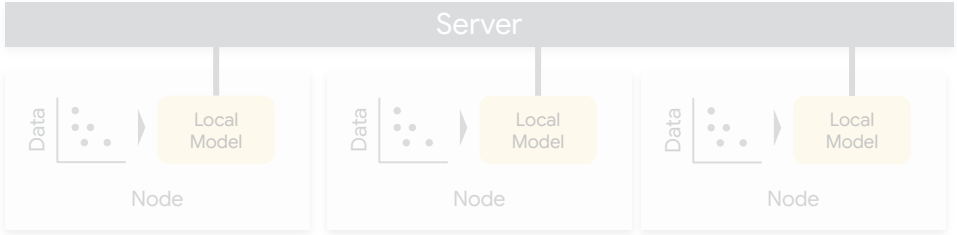
- The key flaw behind the prior approach to training is the sending of **raw client data** to central server
  - Server has access to raw client data, exposing clients to intrusion of privacy by central server

Central Servers

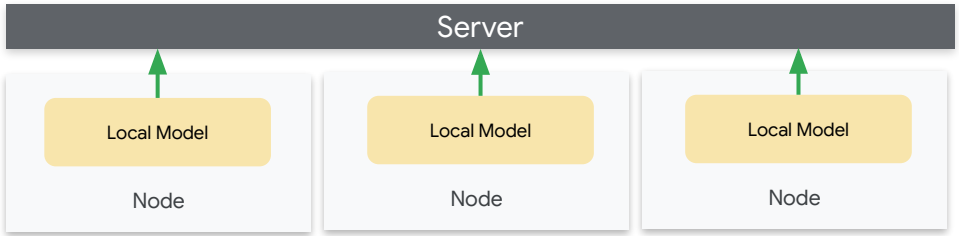
Receive model from server, start training.



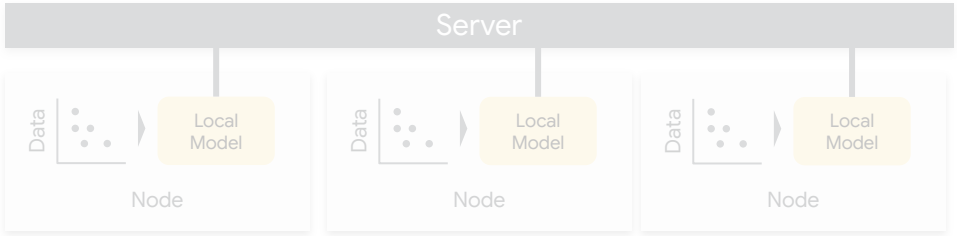
Receive model from server, start training.



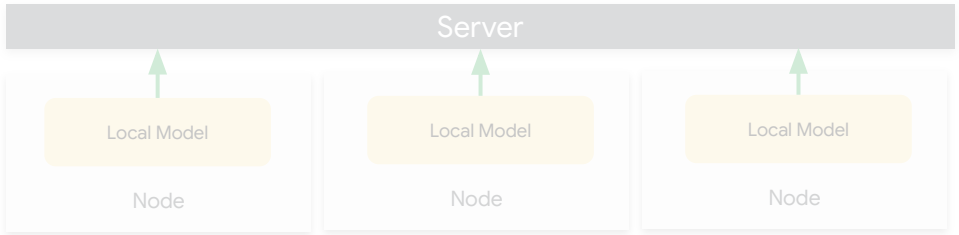
Partially trained models → server



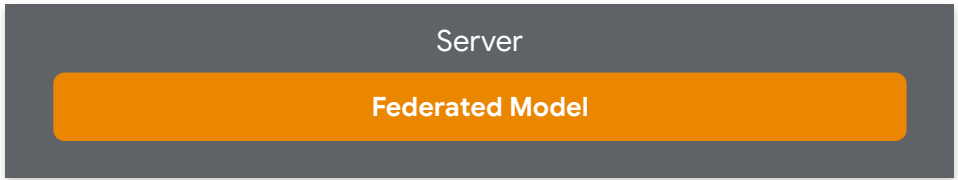
Receive model from server, start training.

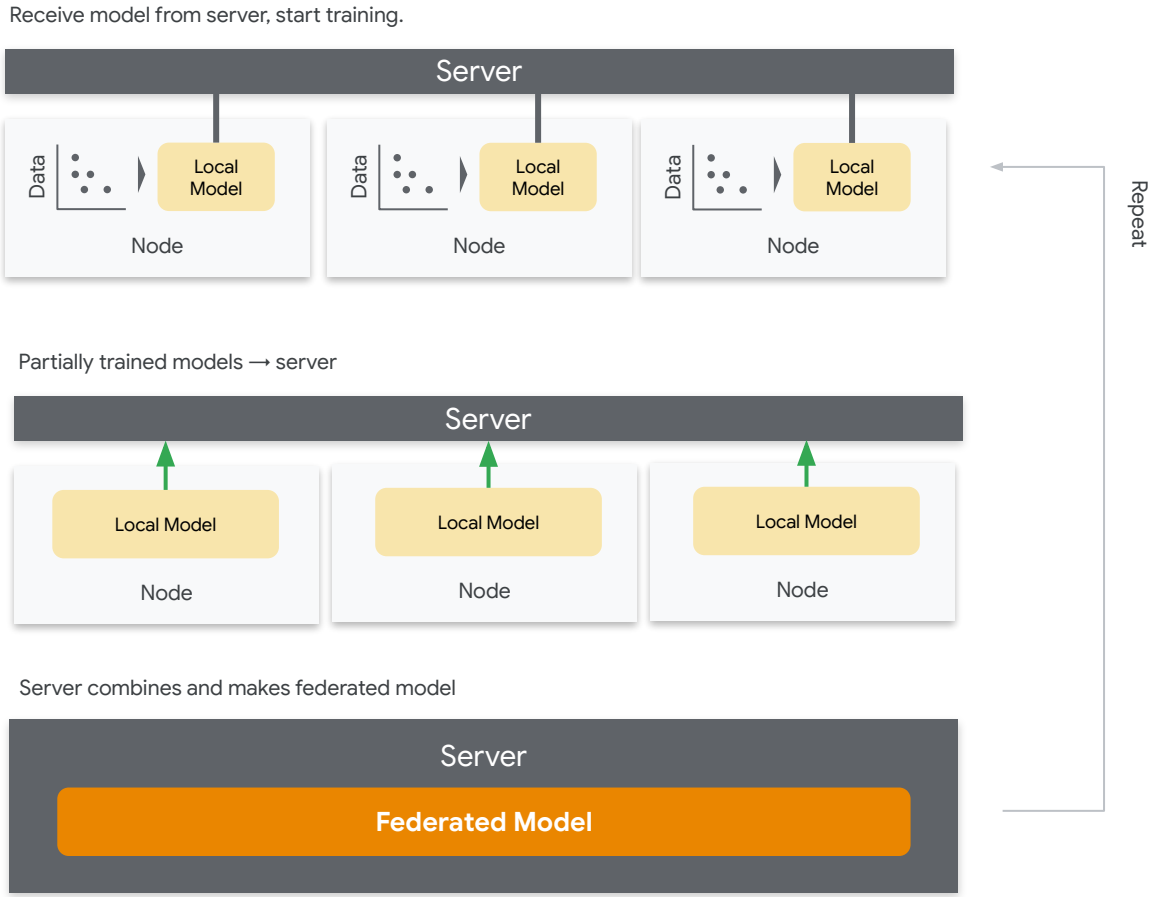


Partially trained models → server



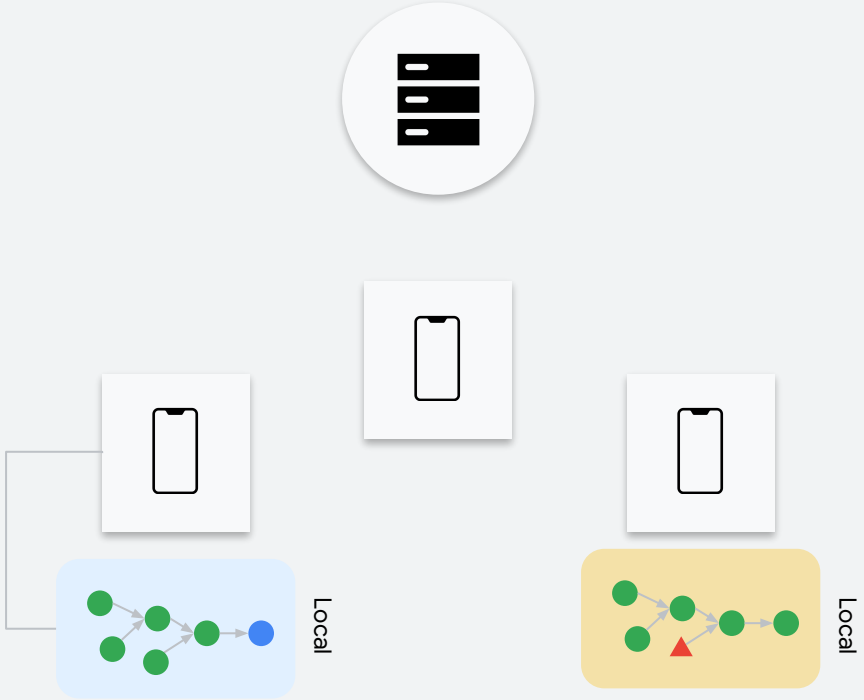
Server combines and makes federated model




Receive model from server, start training.
Server
Data
Local Model
Node
Data
Local Model
Node
Data
Local Model
Node
Repeat
Partially trained models → server
Server
Local Model
Node
Local Model
Node
Local Model
Node
Server combines and makes federated model
Server
Federated Model

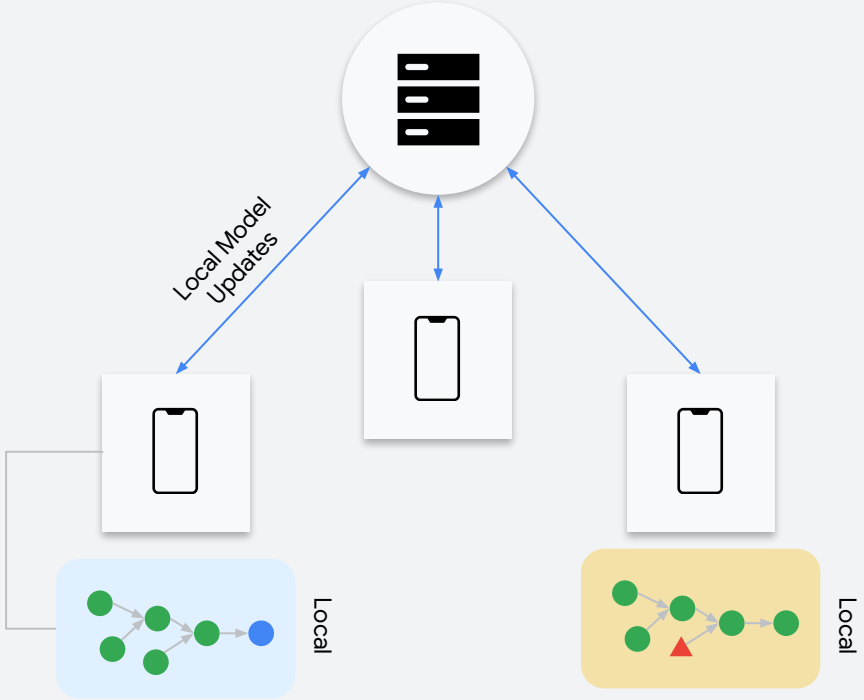
Federated ML

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



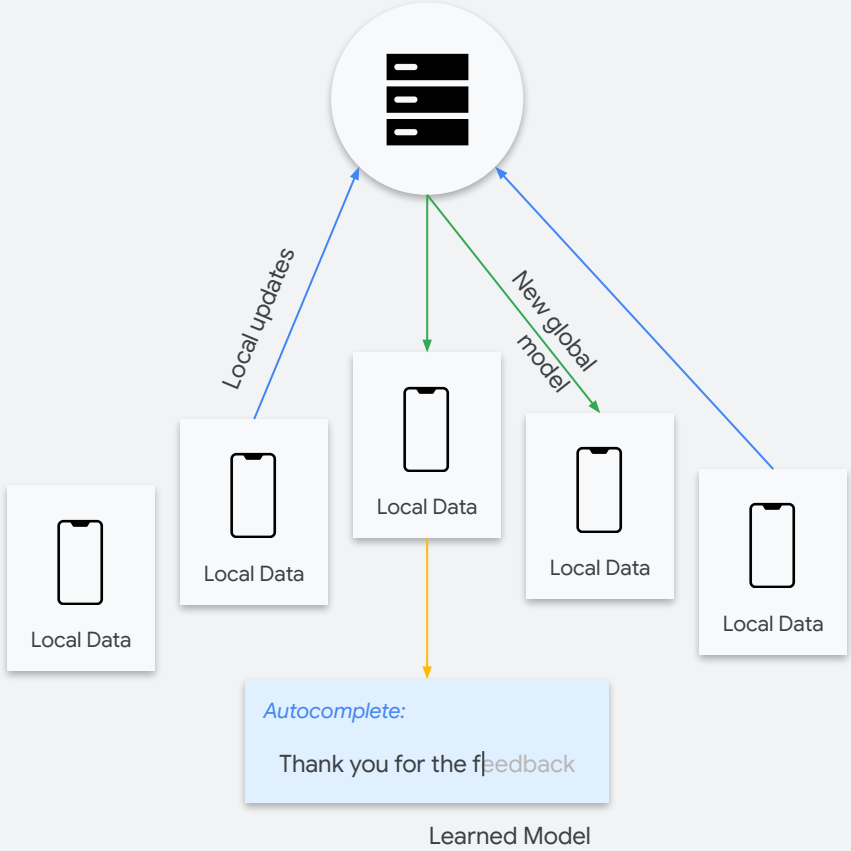
Federated ML

- **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



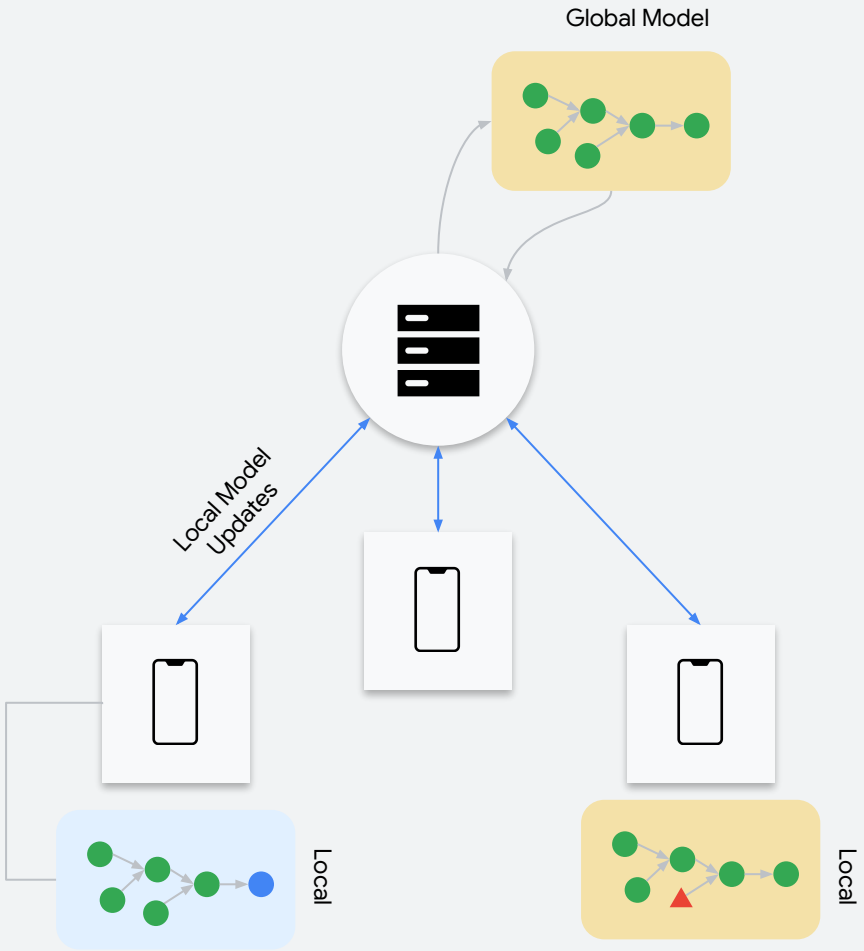
Federated Learning

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



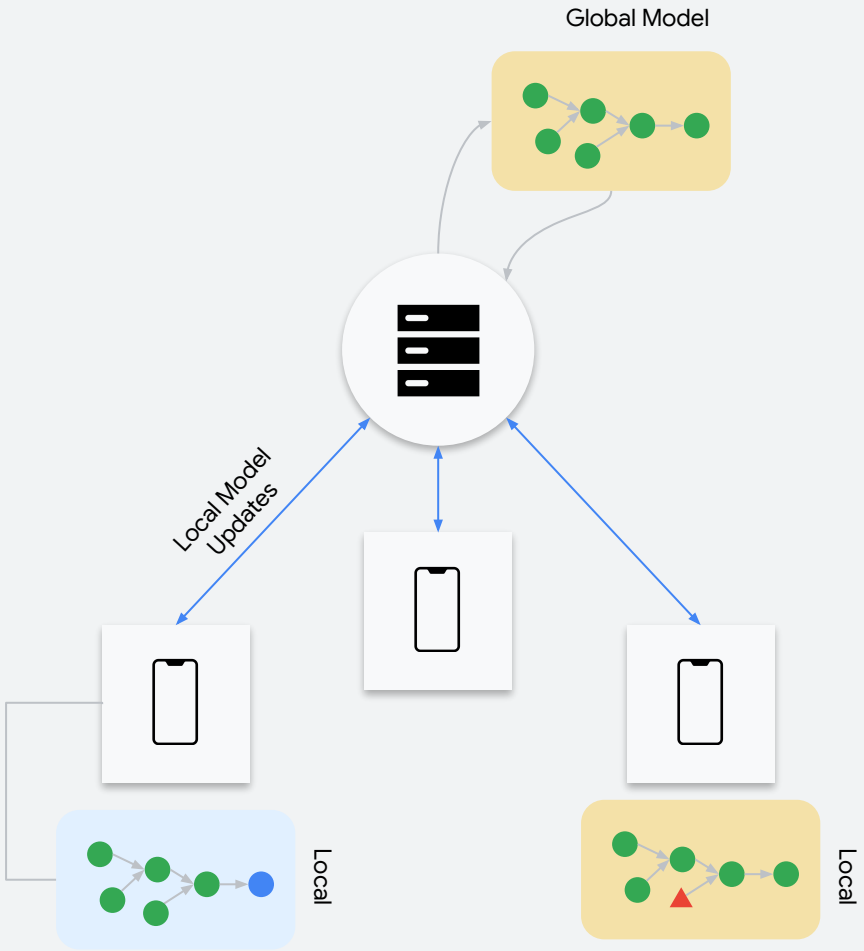
Federated ML

- **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
- Server creates a **global model** and sends it to the endpoints



Why is Federated ML **Useful**?

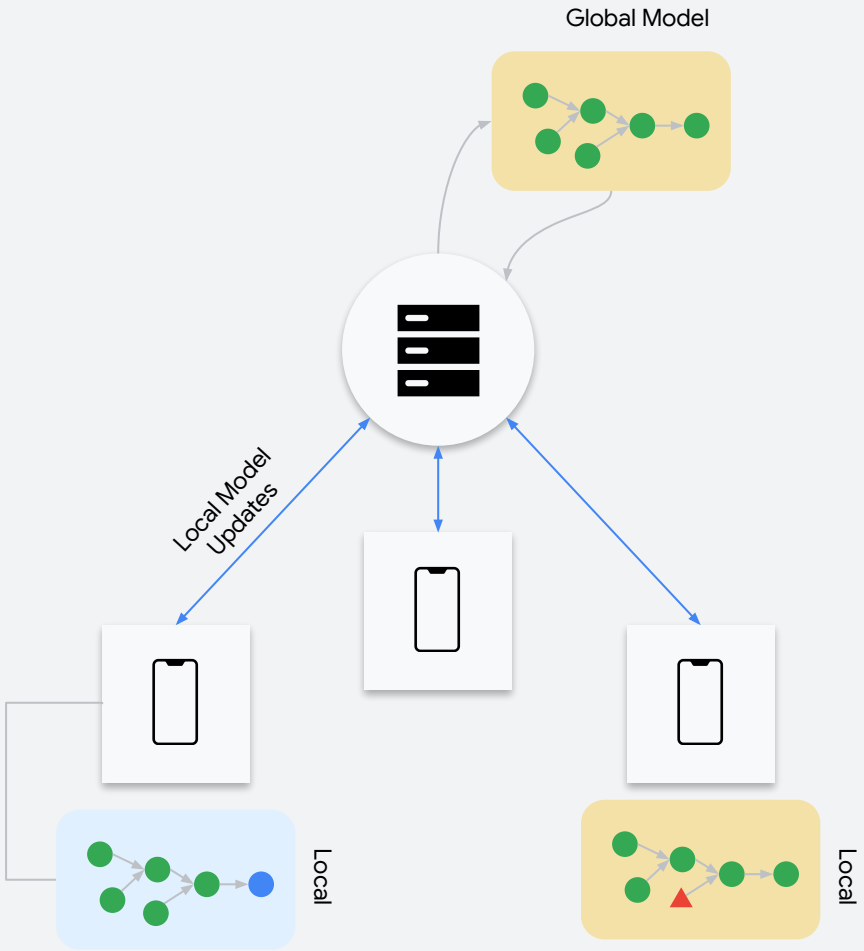
Hyper-Personalized



Why is Federated ML **Useful**?

Hyper-Personalized

Minimum Latencies

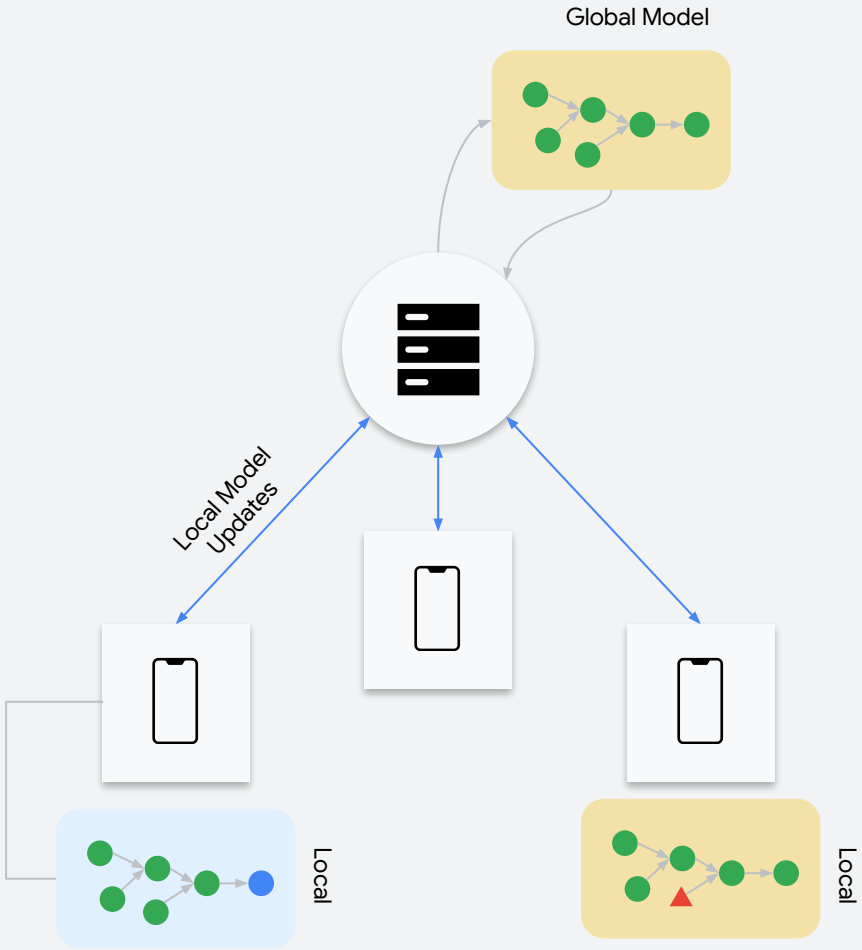


Why is Federated ML **Useful**?

Hyper-Personalized

Low Cloud Infra Overheads

Minimum Latencies



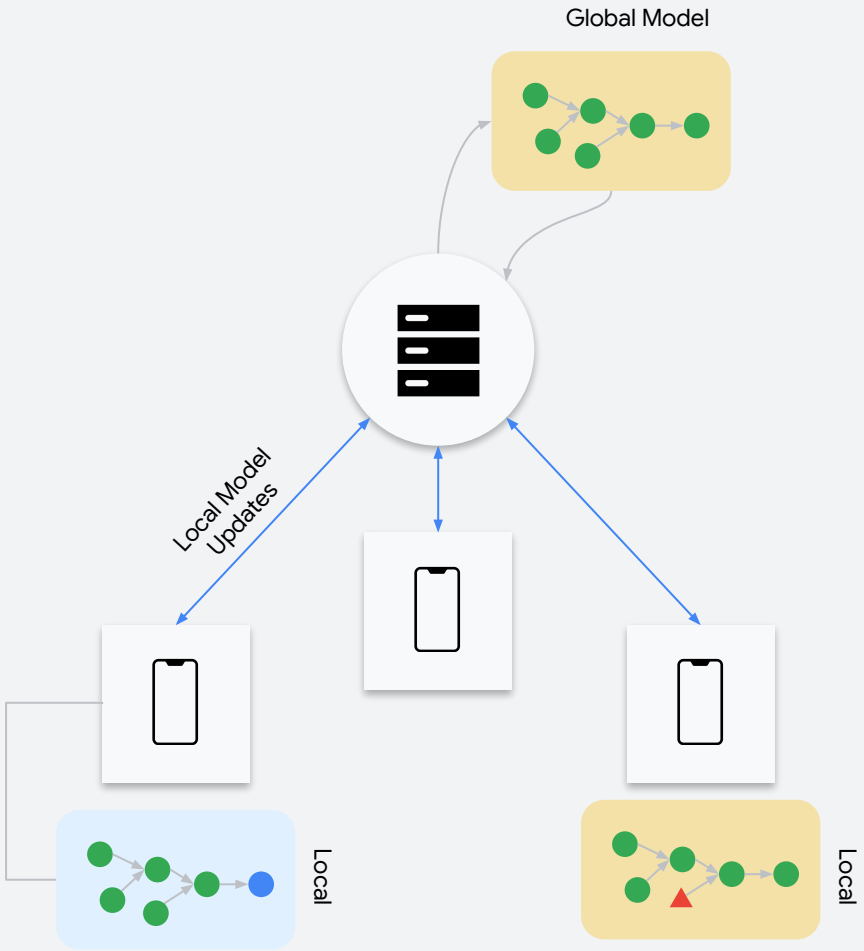
Why is Federated ML **Useful**?

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

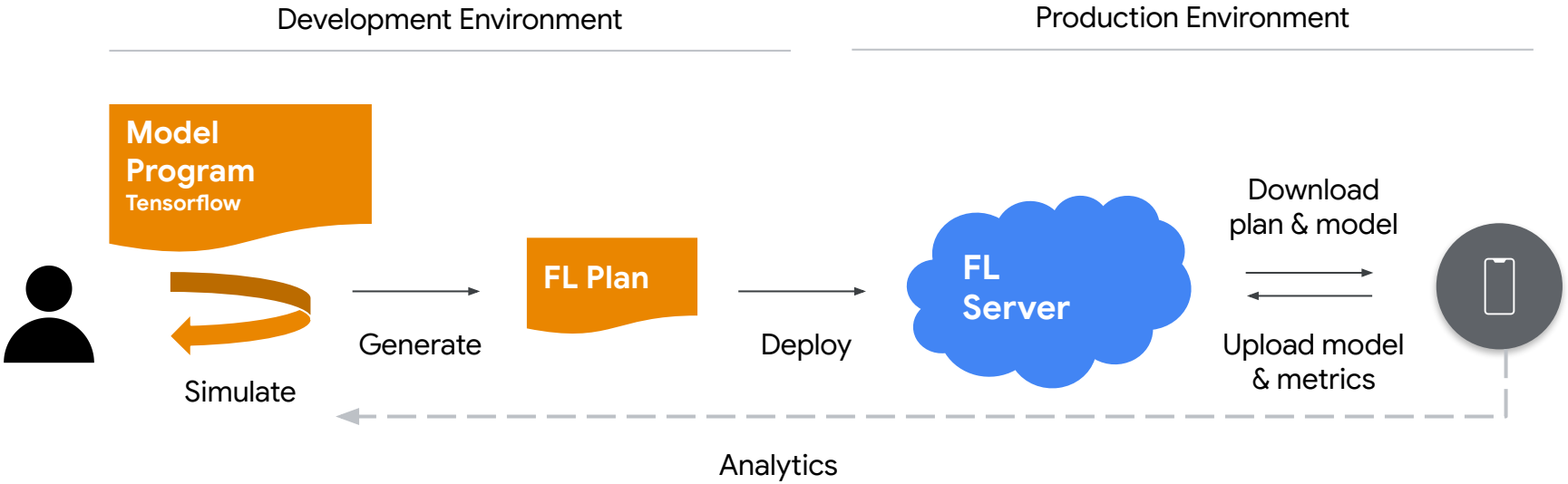
No Free Lunch

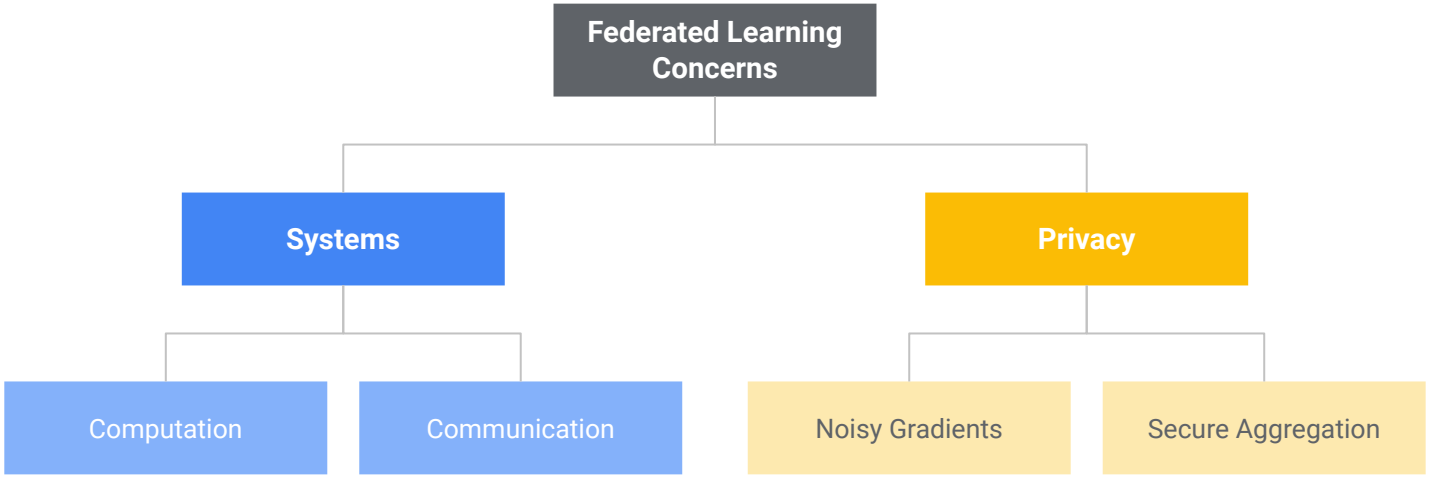
Unbalanced training samples

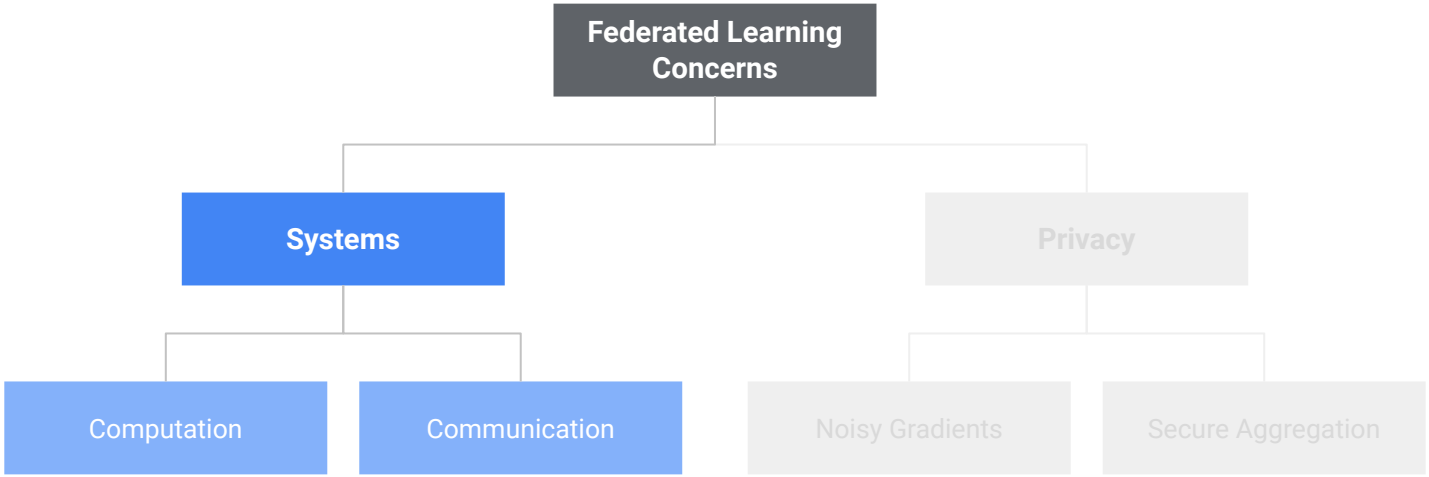
Need a high # of endpoint devices/clients

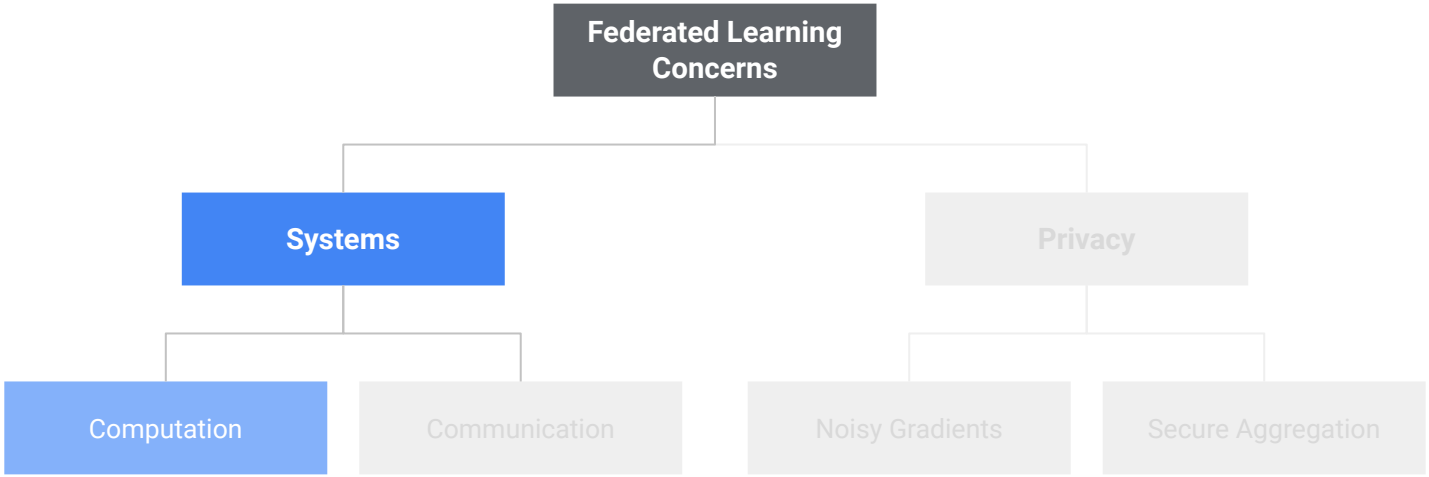
Slow and unreliable network connections

Federated Learning



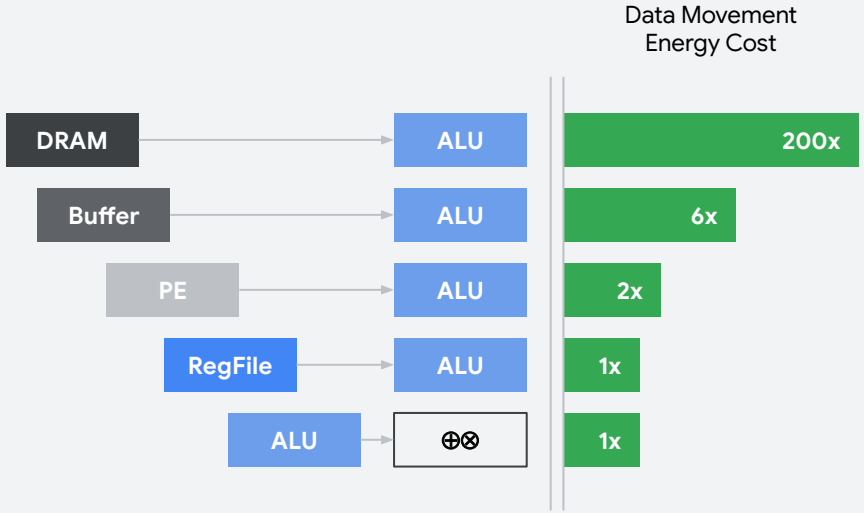

Federated Learning Concerns
Systems
Privacy
Computation
Communication
Noisy Gradients
Secure Aggregation

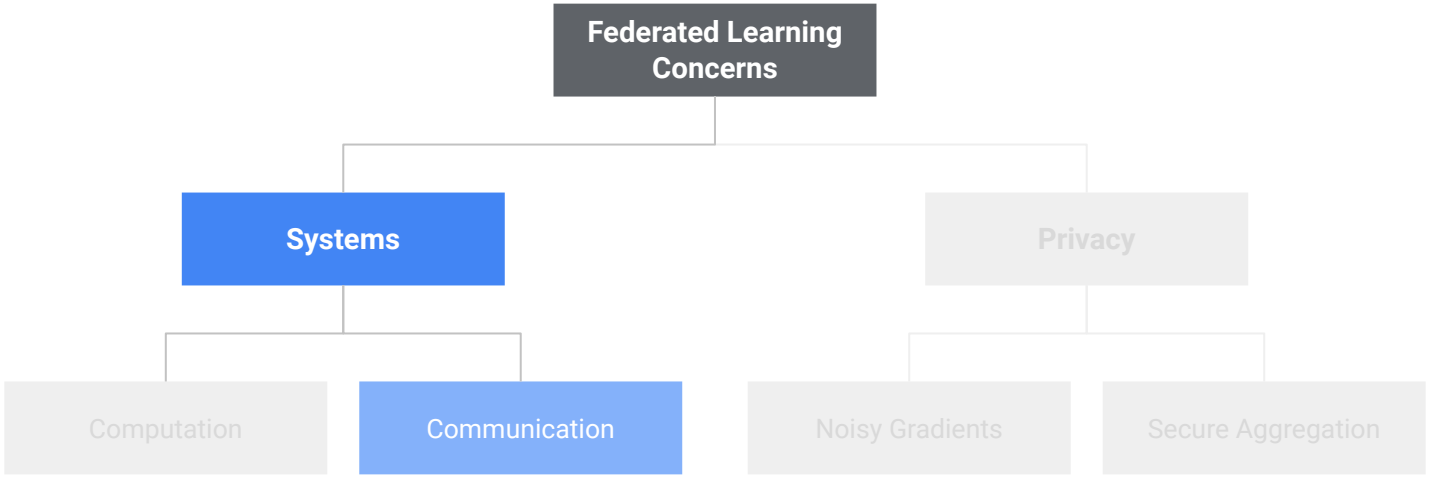

Federated Learning Concerns
Systems
Privacy
Computation
Communication
Noisy Gradients
Secure Aggregation


Federated Learning Concerns
Systems
Privacy
Computation
Communication
Noisy Gradients
Secure Aggregation

Computation vs. Communication

- Data movement is extremely important to keep in mind for efficient system engineering
- Data movement is more costly than computation itself




Federated Learning Concerns
Systems
Privacy
Computation
Communication
Noisy Gradients
Secure Aggregation

Continuous Monitoring for **TinyML**

- Monitoring may **not always** be a **feasible** option
  - Low power communication protocol
  - Device isn't wifi-enabled

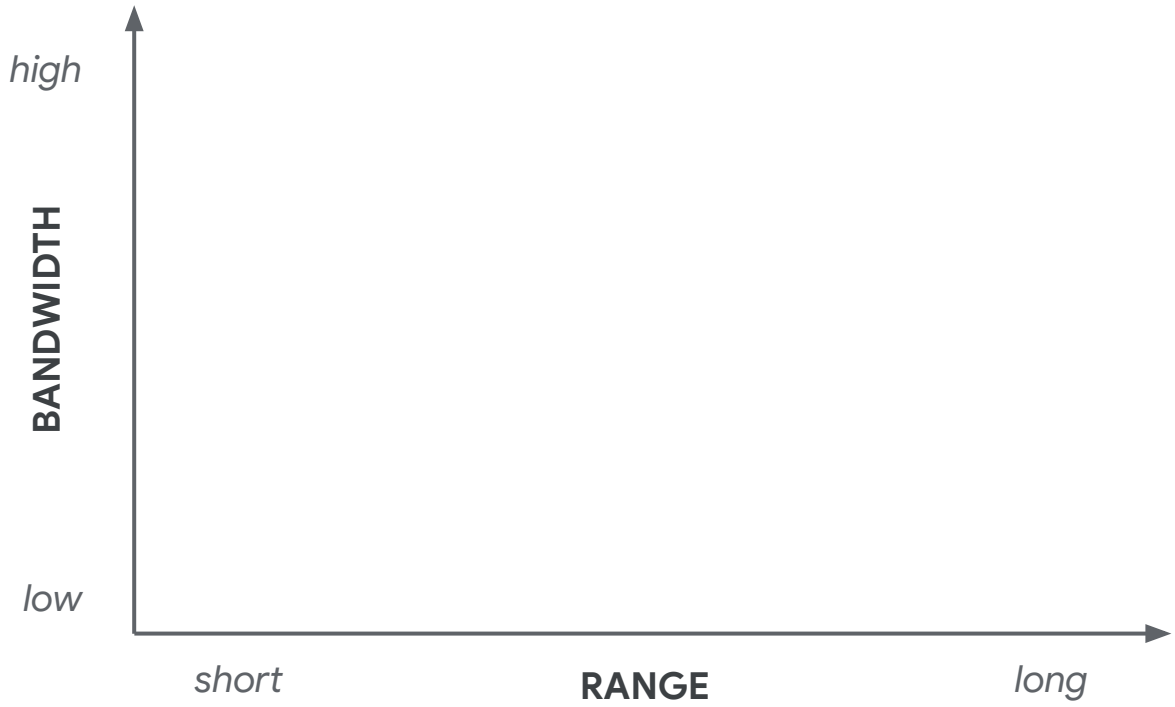
- Monitoring opens up **security and privacy risks**

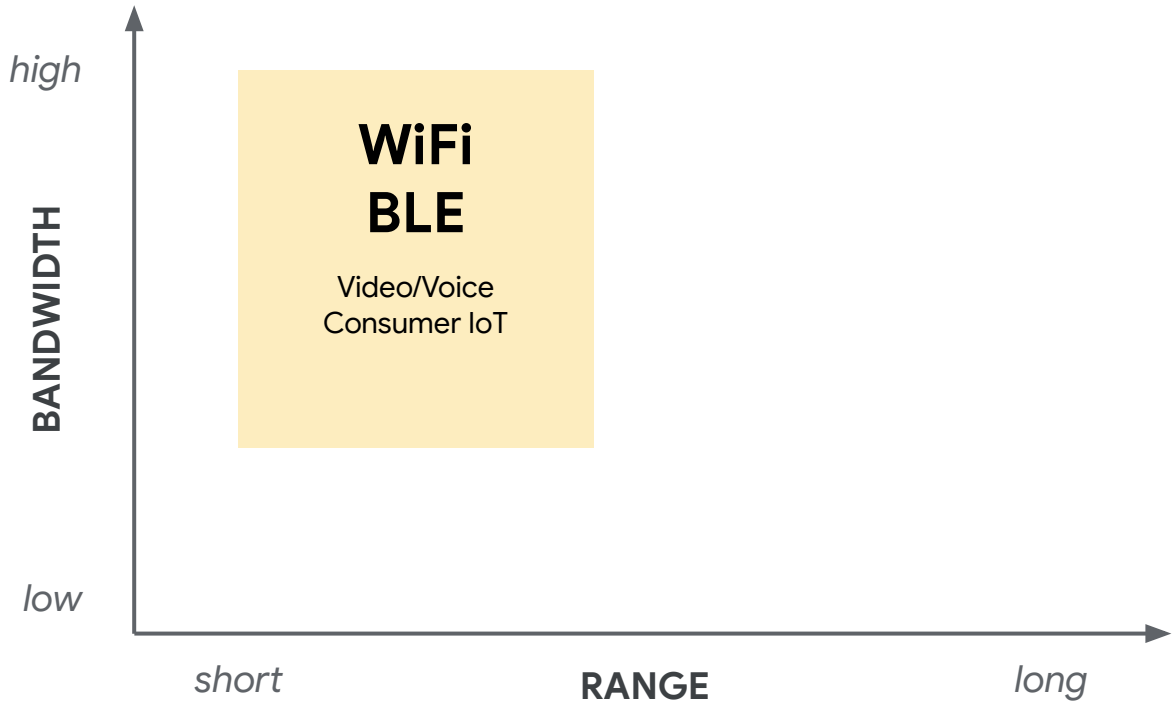
Continuous Monitoring for **TinyML**

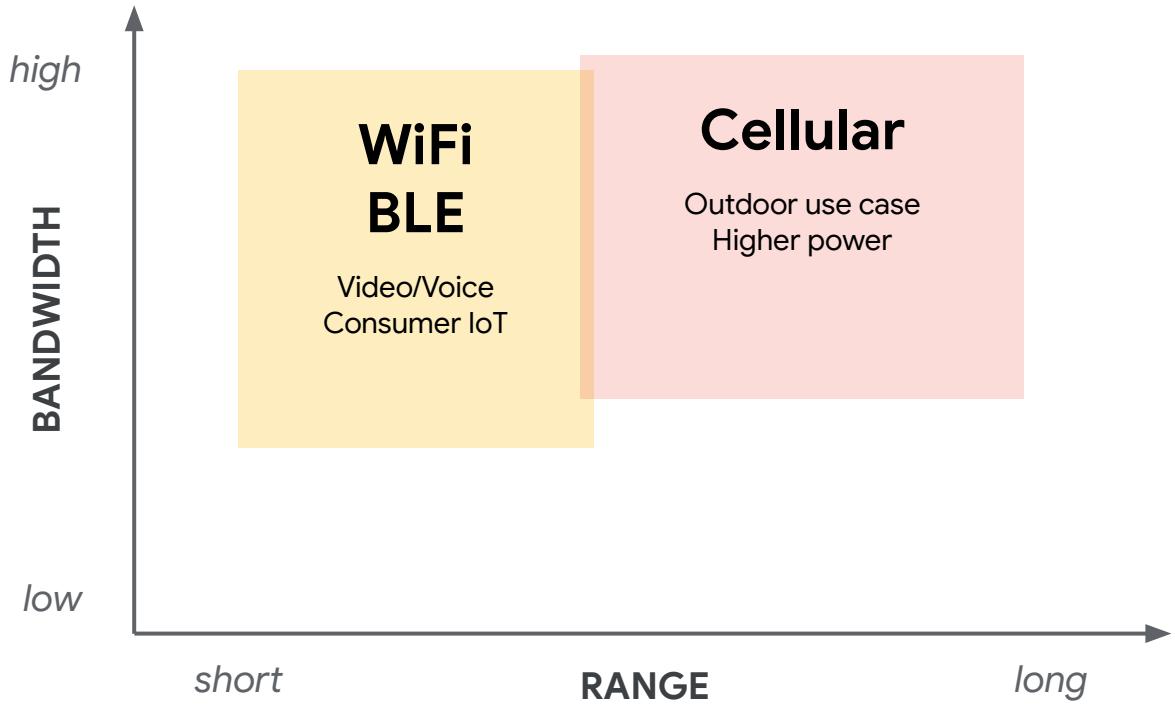
- Monitoring may **not always** be a **feasible** option
  - Low power communication protocol
  - Device isn't wifi-enabled

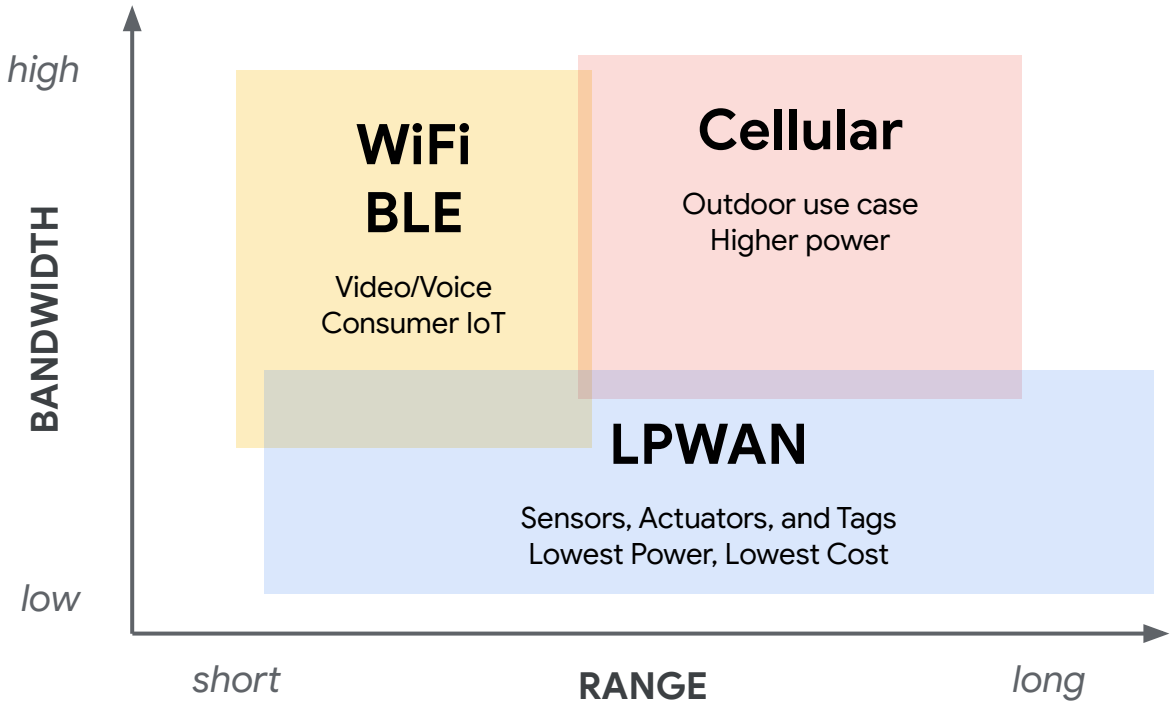
- Monitoring opens up **security and privacy risks**

- How can we enable **Continuous Monitoring** to enable **Continuous Training** without moving the data off the endpoint tiny ML device?


high
BANDWIDTH
low
short
RANGE
long


high
BANDWIDTH
low
WiFi
BLE
Video/Voice
Consumer IoT
short
RANGE
long


high
BANDWIDTH
low
WiFi
BLE
Video/Voice
Consumer IoT
Cellular
Outdoor use case
Higher power
short
RANGE
long


high
BANDWIDTH
low
WiFi
BLE
Video/Voice
Consumer IoT
Cellular
Outdoor use case
Higher power
LPWAN
Sensors, Actuators, and Tags
Lowest Power, Lowest Cost
short
RANGE
long

	LAN Bluetooth, ZigBee, WiFi	Cellular 2G, 3G, 4G	LPWAN LoRa, Sigfox, NB-IoT
Data Rate	~100kbps - 100mbps	~100kbps - 100mbps	10kbps
Range	Short	Long	Long Range (10 miles)
Battery Life	Varies	Medium	Long Battery Life (10 yrs)
Cost	Expensive	Very Expensive	Best Price
Use Cases	*Smart TV, WiFi Network, Bluetooth Speakers*	*Smart Grid, CCTV, Personal Communication (smartphones)*	*Monitoring, Metering, Temperature, Asset Tracking, Weather, Location*

Long Range

Deep indoor coverage (multi-floor buildings)

Star topology network design

Long Battery Life

Low-power optimized

Up to 10 year lifetime

Up to 10x versus Cellular M2M

High Capacity

Millions of messages per gateway

Multi-tenant interoperability

Public/Private network deployments

Low Cost

Minimal infrastructure

Low cost end-node

Open source software

Geolocation

Indoor/outdoor

Accurate without GPS

No battery life impact

Updates

Firmware updates over-the-air for apps and LoRaWAN stack

Roaming

Seamless handoffs from one network to another

Security

Embedded end-to-end AES-128 encryption

Unique ID

Application

Network

Example Use Cases

- Vending machines could alert distributors when a product requires **maintenance**

Example Use Cases

- Vending machines could alert distributors when a product requires **maintenance**
- Animal lovers can **track** pets or study migration patterns over longer distances

Example Use Cases

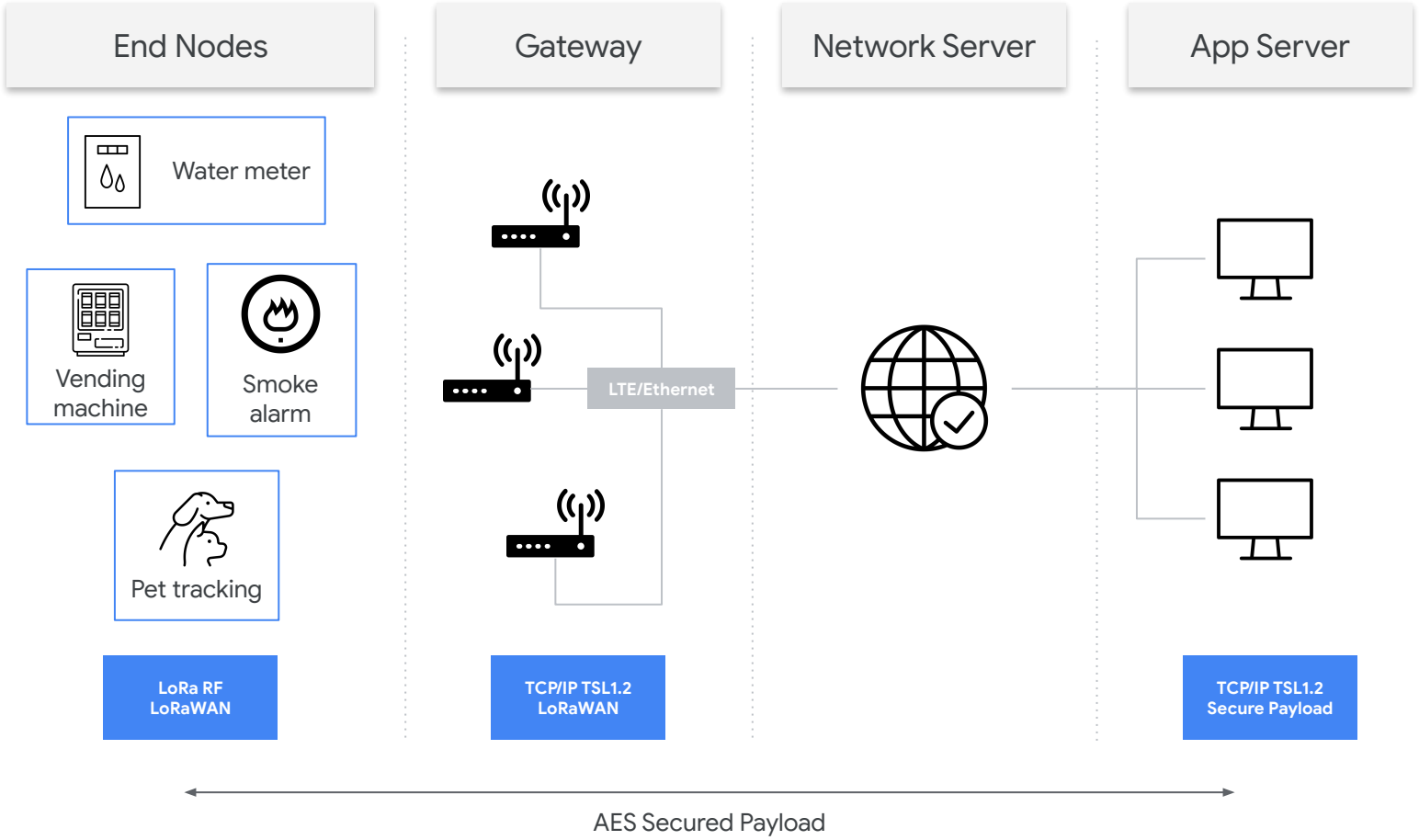
- Vending machines could alert distributors when a product requires **maintenance**
- Animal lovers can **track** pets or study migration patterns over longer distances
- Oil companies could **receive alerts** when home oil tanks are running low

Example Use Cases

- Vending machines could alert distributors when a product requires **maintenance**
- Animal lovers can **track** pets or study migration patterns over longer distances
- Oil companies could **receive alerts** when home oil tanks are running low
- Logistics providers could **track** cargo containers on trucks, ships and trains

Example Use Cases

- Natural **Disaster Prevention**
- Smart **Agriculture Monitoring**
- Animal **Production Monitoring**
- Endangered Species **Protection**
- **Smart** Industry **Control**
- **Smart** Cities, Homes, Buildings & Offices
- Supply Chain **Logistics**, **Asset Tracking** & **Quality Management**
- **Smart Metering** Facilities (Water, Gas, Electricity)


End Nodes
Gateway
Network Server
App Server
Water meter
Vending machine
Smoke alarm
Pet tracking
LTE/Ethernet
LoRa RF
LoRaWAN
TCP/IP TSL1.2
LoRaWAN
TCP/IP TSL1.2
Secure Payload
AES Secured Payload

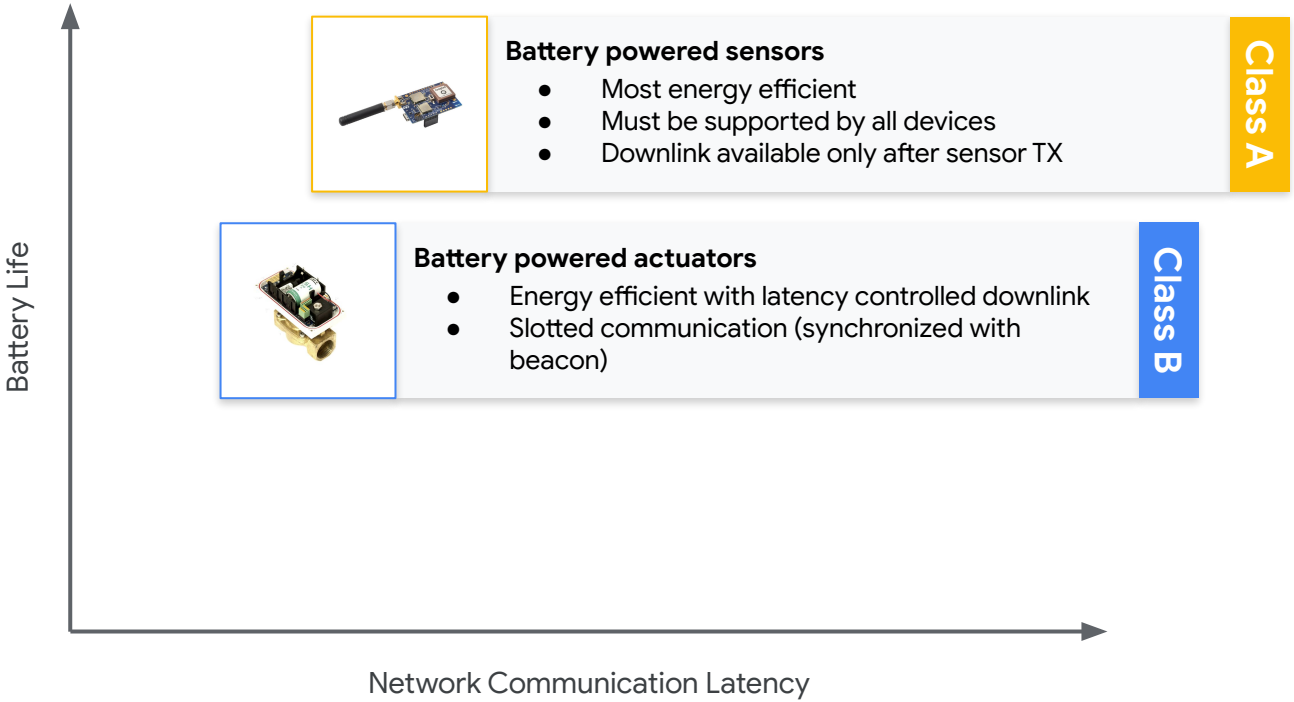
Battery powered sensors

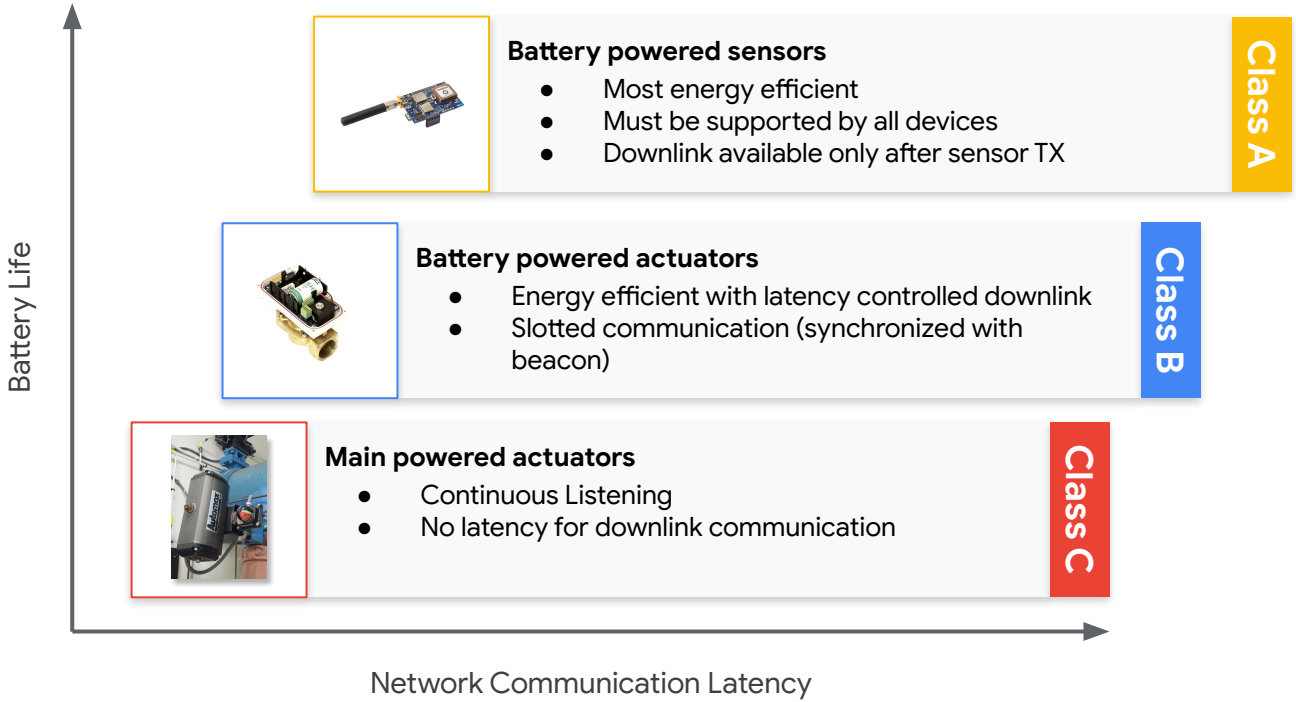
- Most energy efficient
- Must be supported by all devices
- Downlink available only after sensor TX

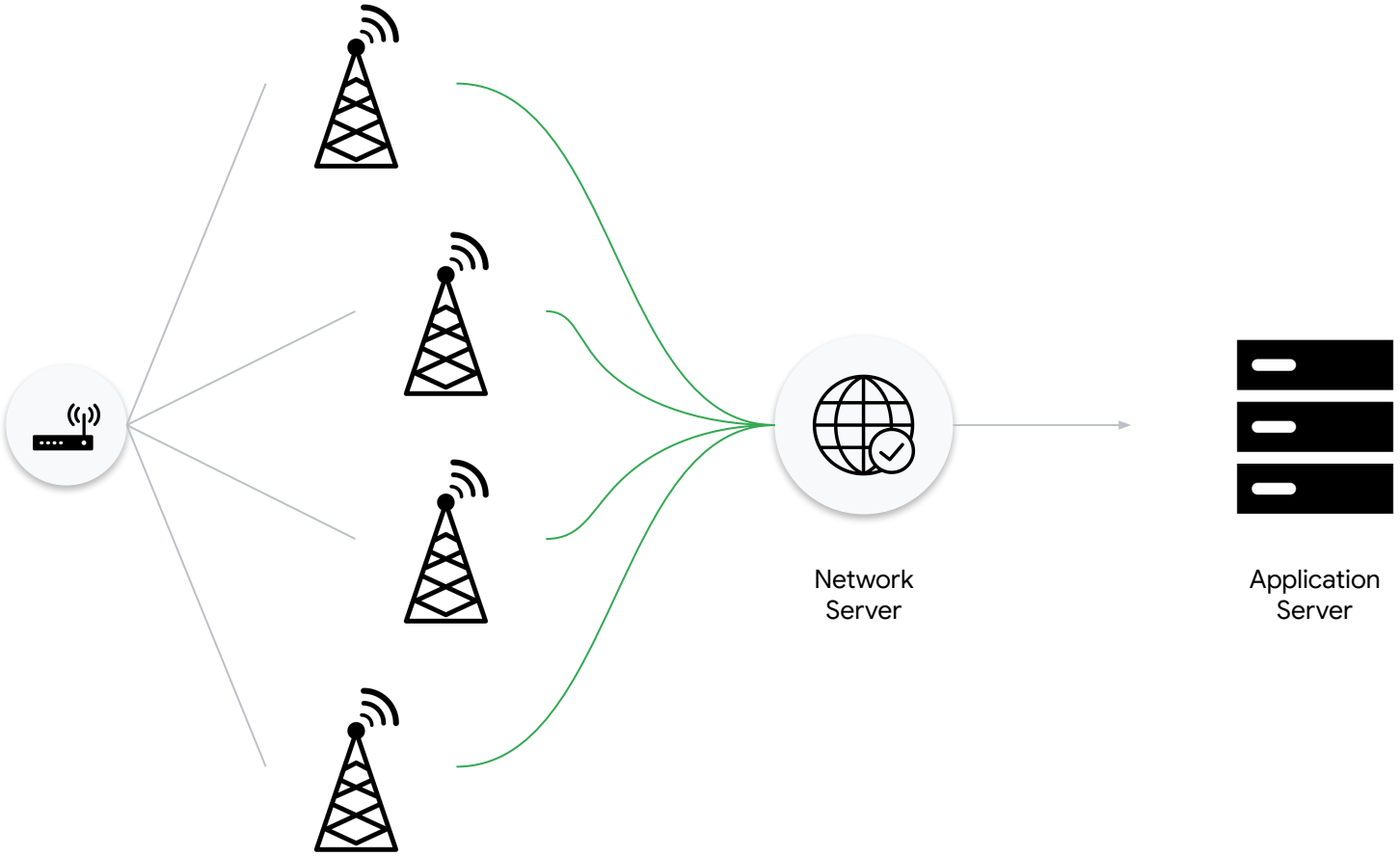
Class A

Battery Life

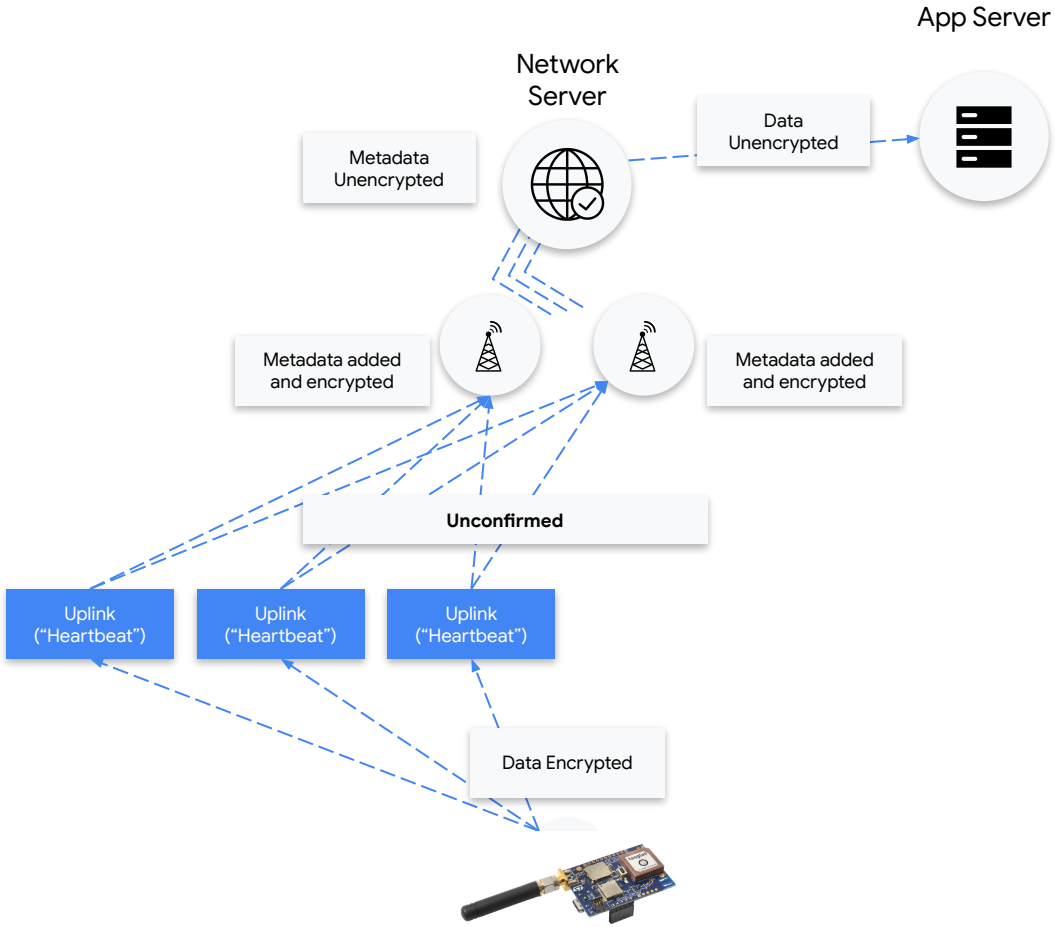
Network Communication Latency


Battery powered sensors
Most energy efficient
Must be supported by all devices
Downlink available only after sensor TX
Class A
Battery powered actuators
Energy efficient with latency controlled downlink
Slotted communication (synchronized with beacon)
Class B
Battery Life
Network Communication Latency

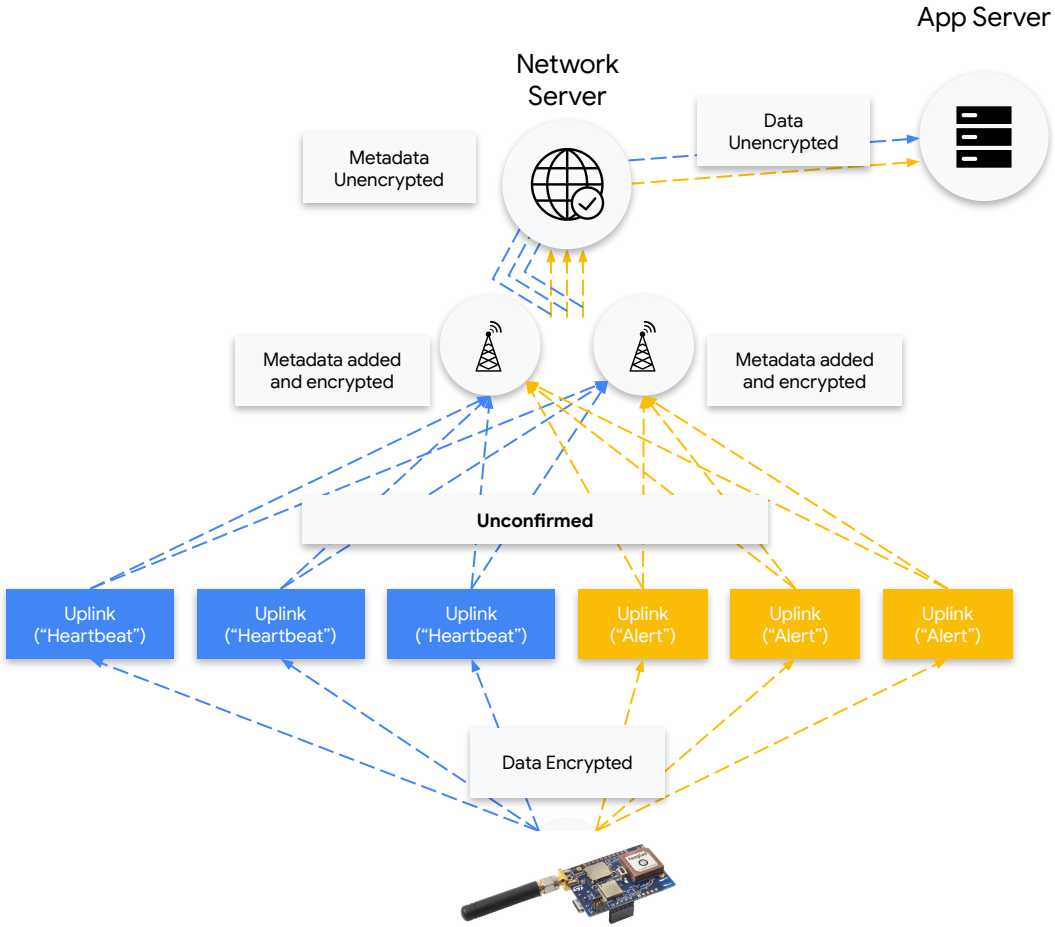

Battery powered sensors
Most energy efficient
Must be supported by all devices
Downlink available only after sensor TX
Class A
Battery powered actuators
Energy efficient with latency controlled downlink
Slotted communication (synchronized with beacon)
Class B
Main powered actuators
Continuous Listening
No latency for downlink communication
Class C
Battery Life
Network Communication Latency


Network
Server
Application
Server

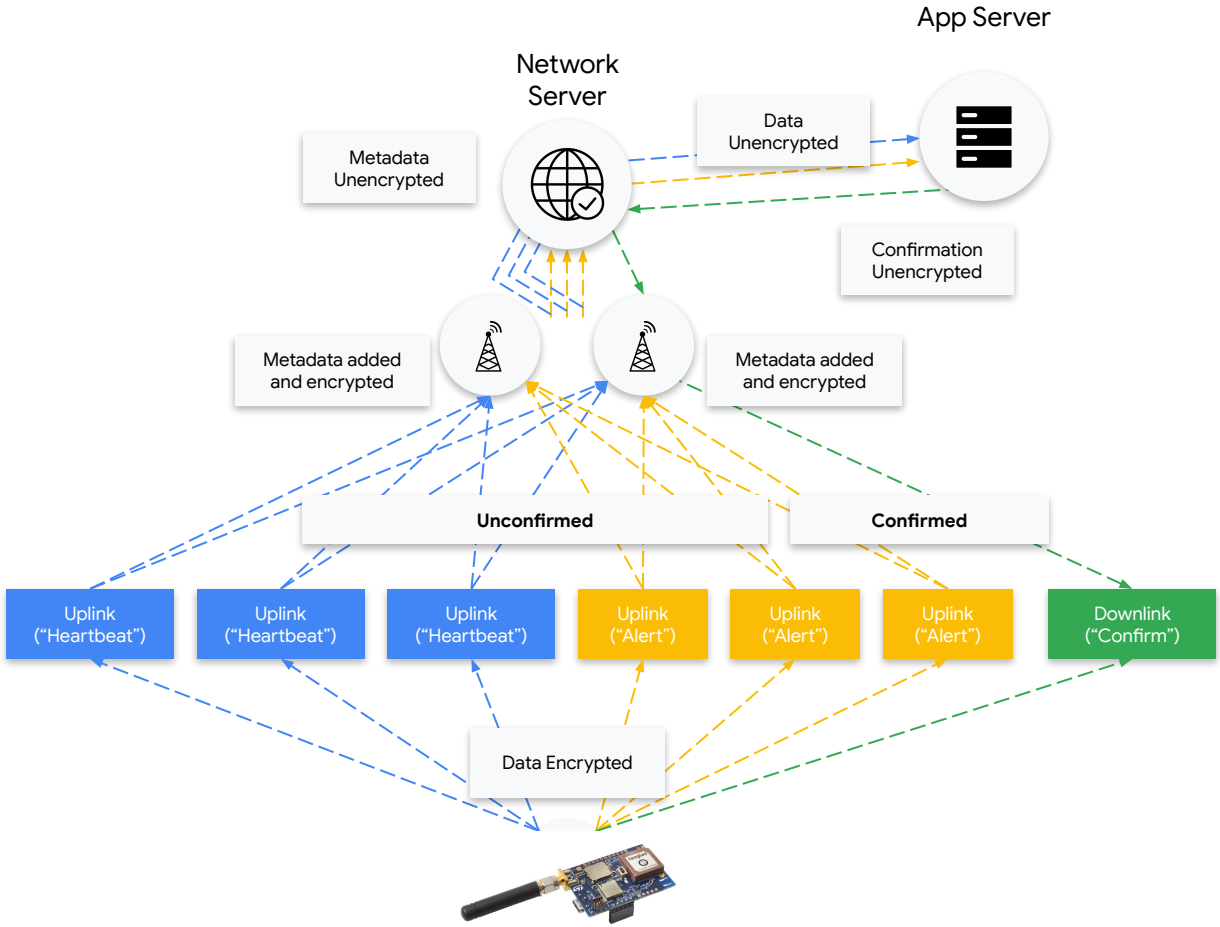
Example Use Case



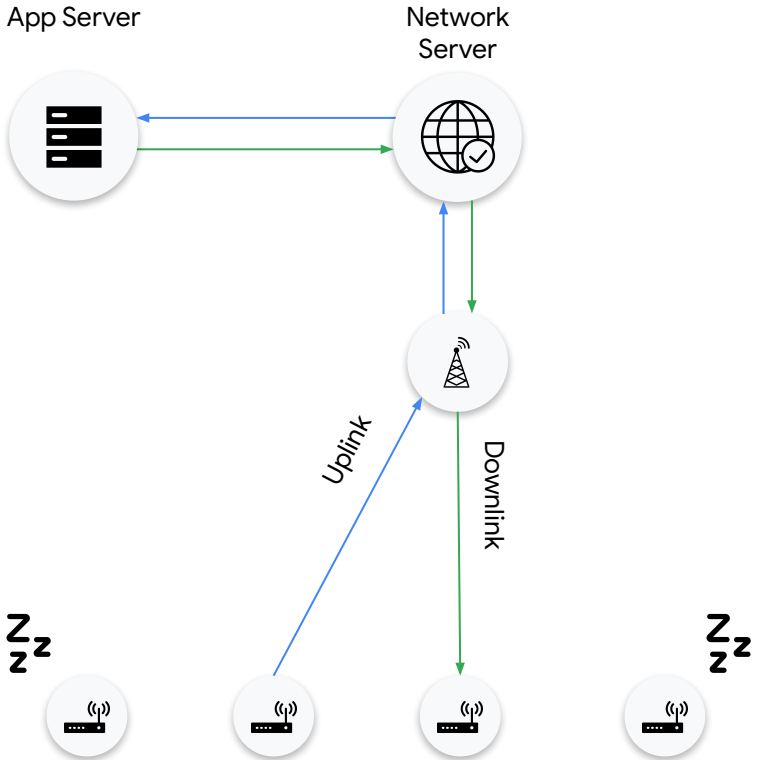
Example Use Case



Example Use Case

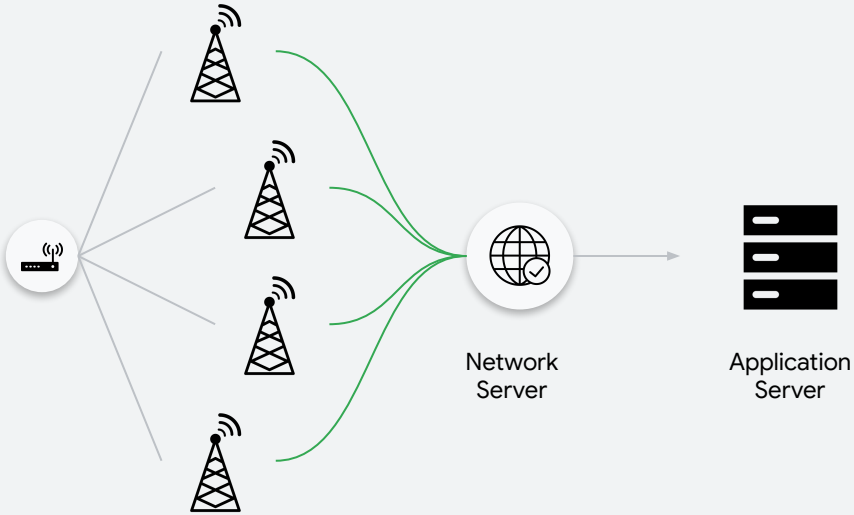


Example Use Case



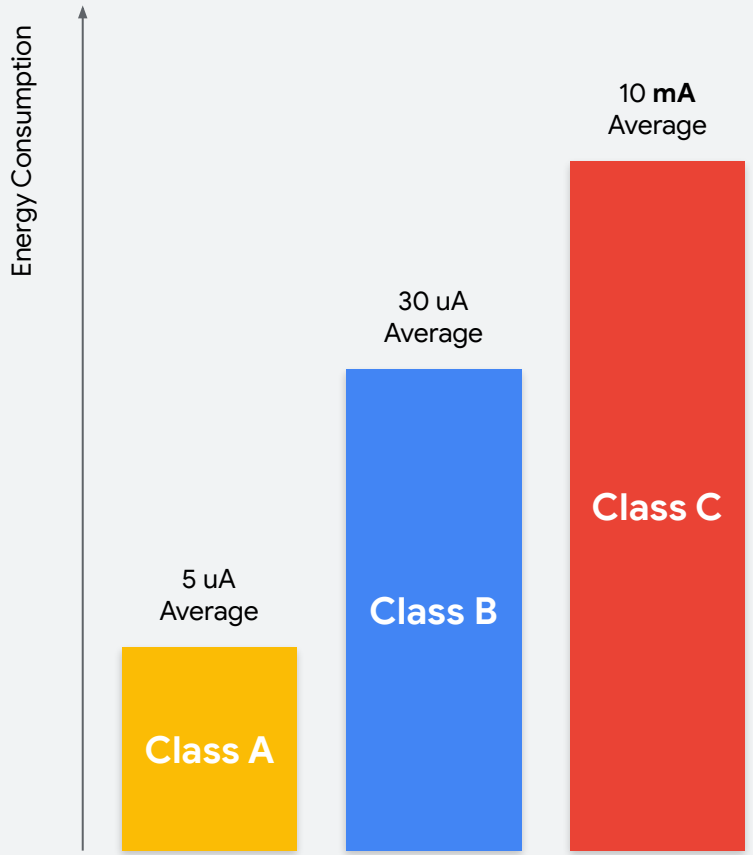
Network Advantages for Class A Devices

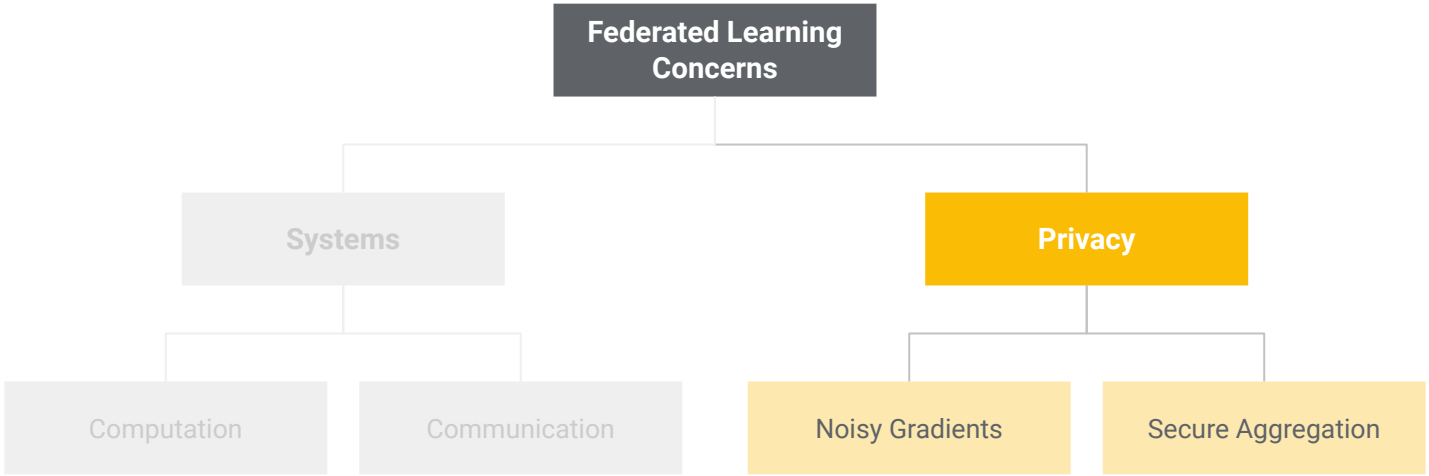
- **No complicated network** planning is required. Gateways can be added anywhere at any time
- **Accurate message delivery is robust**, since multiple gateways receive the same data packet during each uplink. This is called uplink spatial diversity
- There is **no need to plan for different frequencies for each gateway**, or to reallocate frequencies when the number of gateways change. All gateways are constantly listening to all frequencies of the network
- Mobile devices can **operate at ultra low power** thanks to the fact that any gateway can receive messages from any device at any time

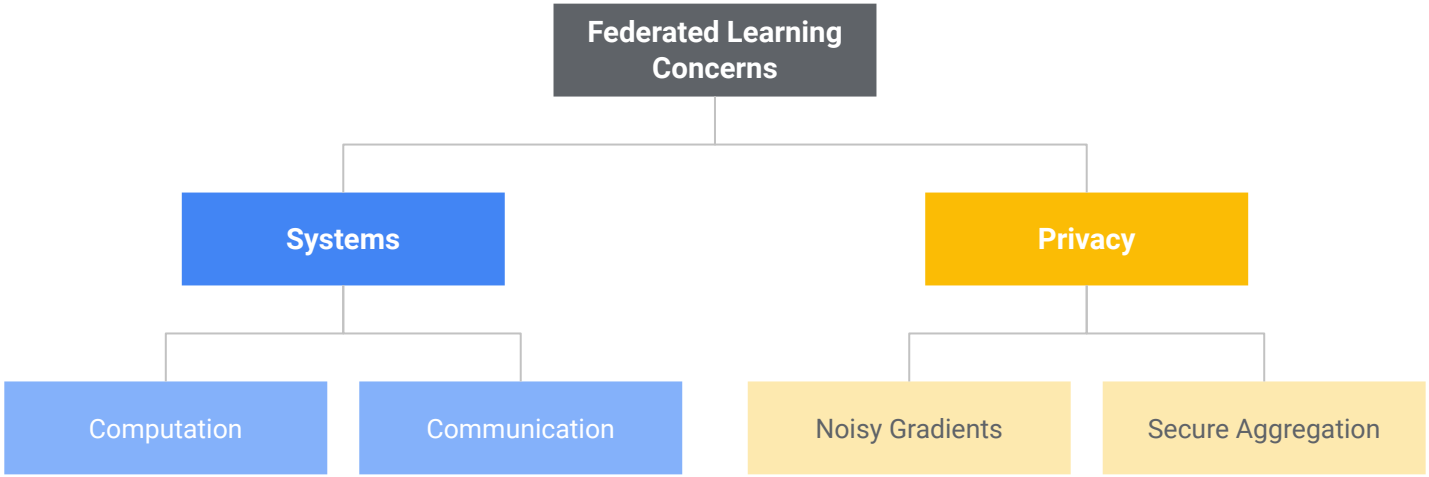


Cross Comparison

- **Class A** device communication is initiated only by the end device
- **Class B** devices have regularly-scheduled windows, in addition to those that open when a Class A-style uplink is sent to server
- **Class C** devices achieve the lowest latency among all operating modes




Federated Learning Concerns
Systems
Privacy
Computation
Communication
Noisy Gradients
Secure Aggregation


Federated Learning Concerns
Systems
Privacy
Computation
Communication
Noisy Gradients
Secure Aggregation

Paper Discussions

Discussion Questions

Discuss

- Why did we read the two papers?
- What are the differences between them?
- <lead discussion>

Summary

- Discuss as a group

Guest Speaker

Nicholas Lane

Nic is a professor of machine learning systems at Cambridge University. His research focuses on the design, architecture and algorithms of scalable and robust end-to-end machine learning systems. He is also the co-founder and Chief Scientific Officer of Flower Labs, a venture-backed AI company behind the Flower federated learning framework.

Personal Website

Google Scholar