Federated Learning, from Research to Practice

Brendan McMahan
mcmahan@google.com

Presenting the work of many

CMU 2019.09.05

g.co/federated

Federated learning

Enable machine learning engineers and data scientists
to work productively with decentralized data
with privacy by default

Outline

1. Why do we need federated learning?
2. What is federated learning?
3. Federated learning at Google
4. Federated learning beyond Google
5. Federated learning and privacy
6. Open problems

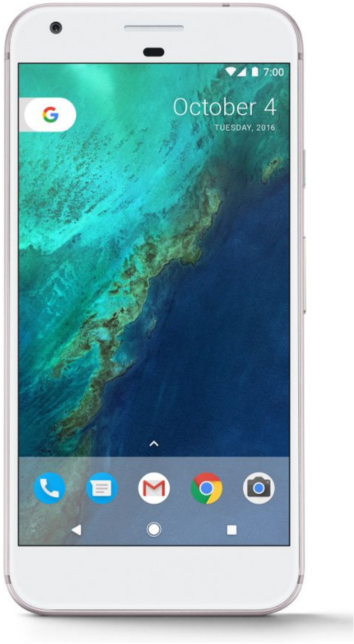
Why federated learning?

Google

Data is born at the edge

Billions of phones & IoT devices constantly generate data

Data enables better products and smarter models

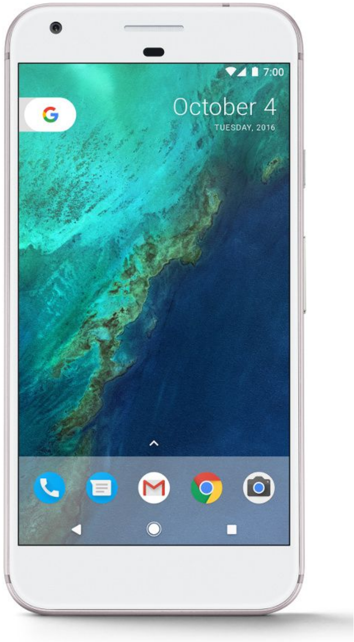


Can data live at the edge?

Data processing is moving on device:

- Improved latency
- Works offline
- Better battery life
- Privacy advantages

E.g., on-device inference for mobile keyboards and cameras.



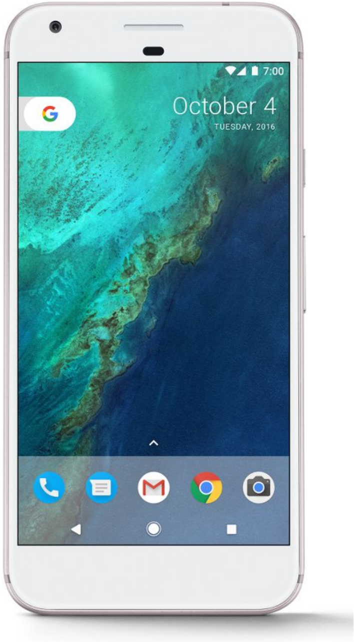
Can data live at the edge?

Data processing is moving on device:

- Improved latency
- Works offline
- Better battery life
- Privacy advantages

E.g., on-device inference for mobile keyboards and cameras.

What about analytics?
What about learning?



2014: Three choices

Don't use data to improve products and services

Log the data centrally *anyway*

Invent a new solution

Choose your poison □

2014: Three choices

Don’t use data to improve products and services

Log the data centrally *anyway*

Invent a new solution

Choose your poison □

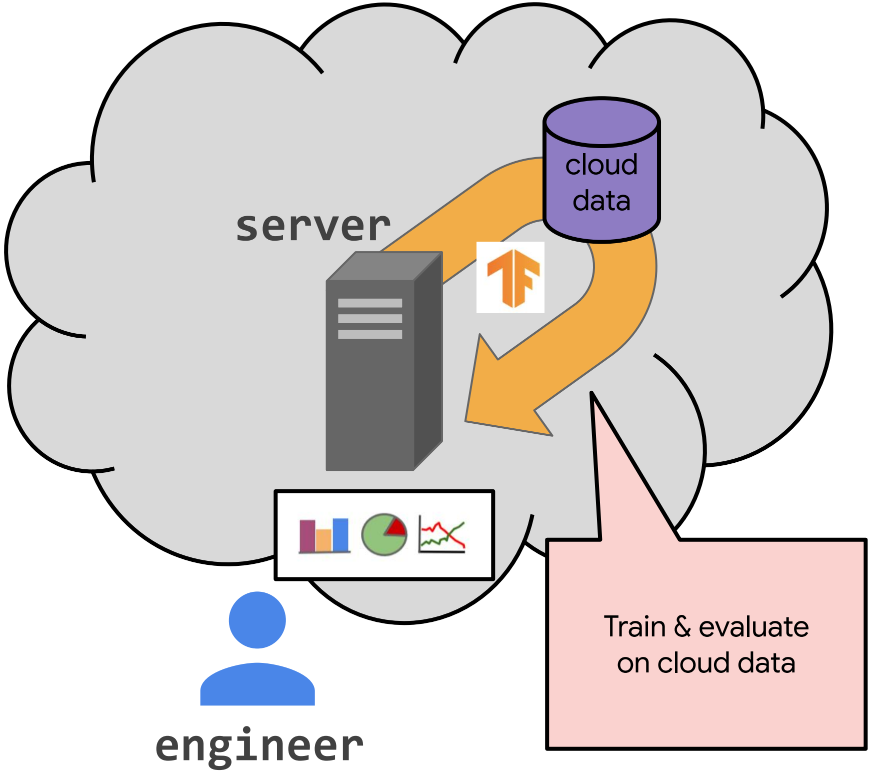
2019: Good reason to hope

Don’t use data to improve products and services	Log the data centrally *anyway*	Federated learning and analytics

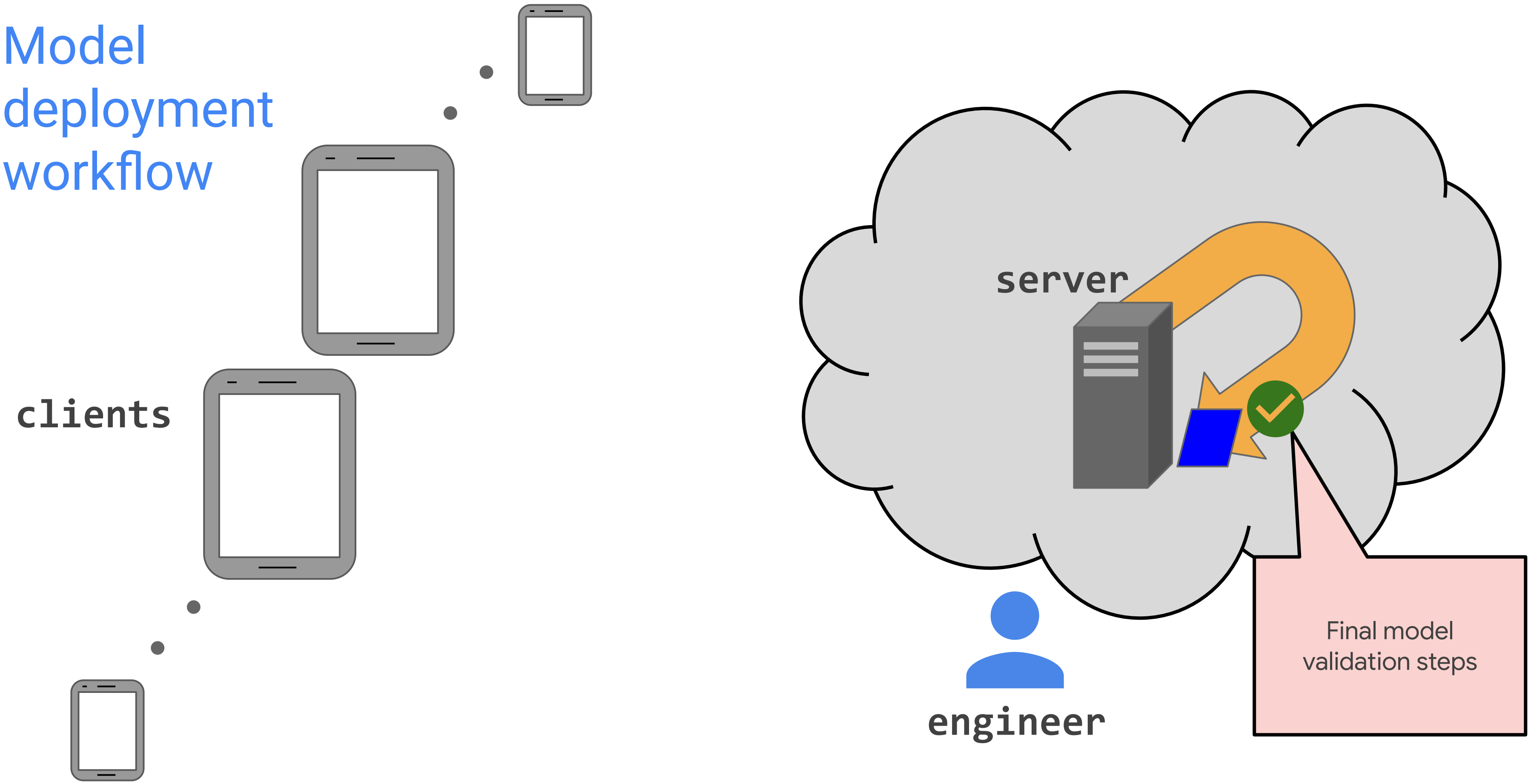
Google

What is federated learning?

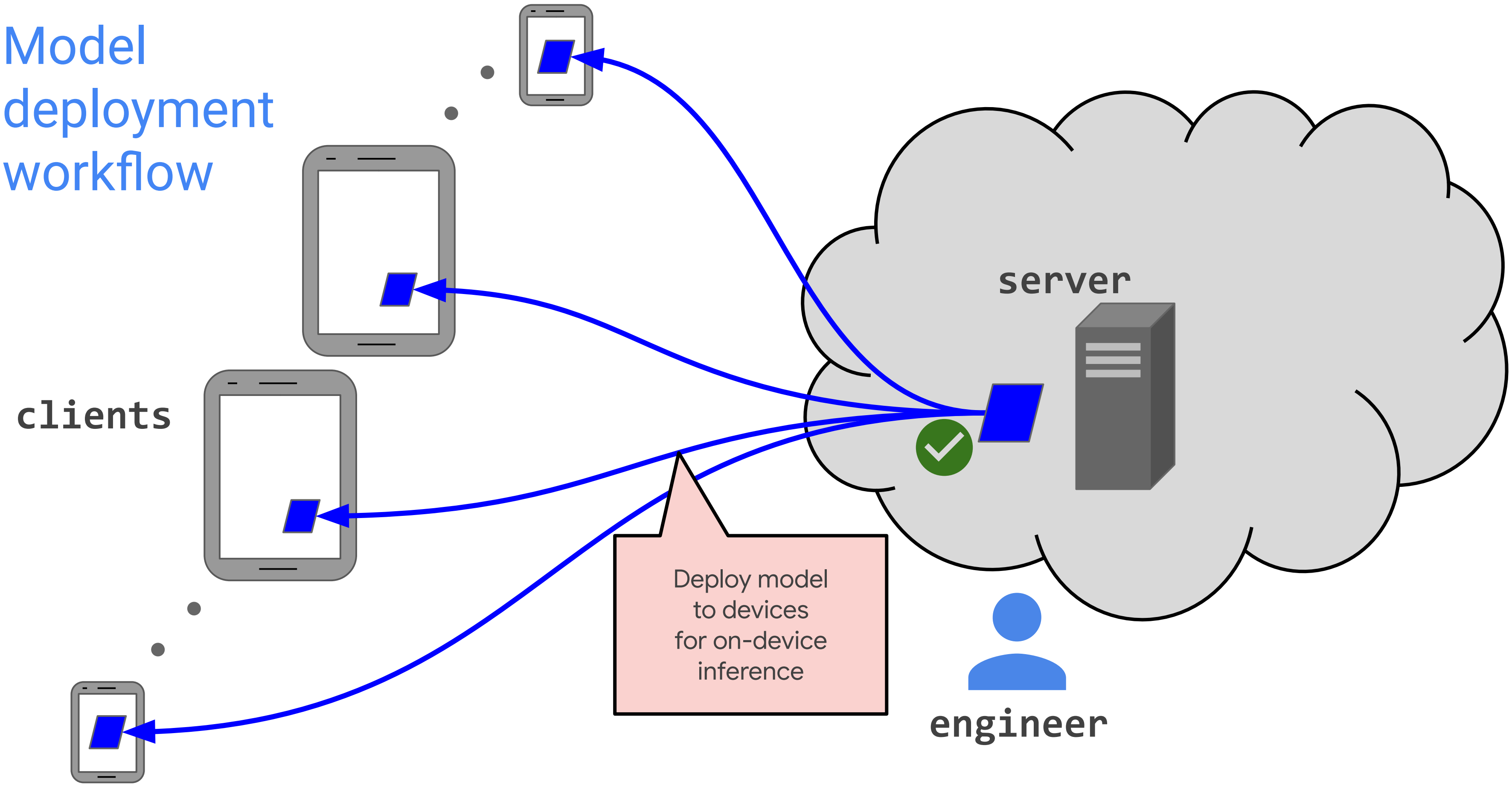
Model engineer workflow



Model deployment workflow

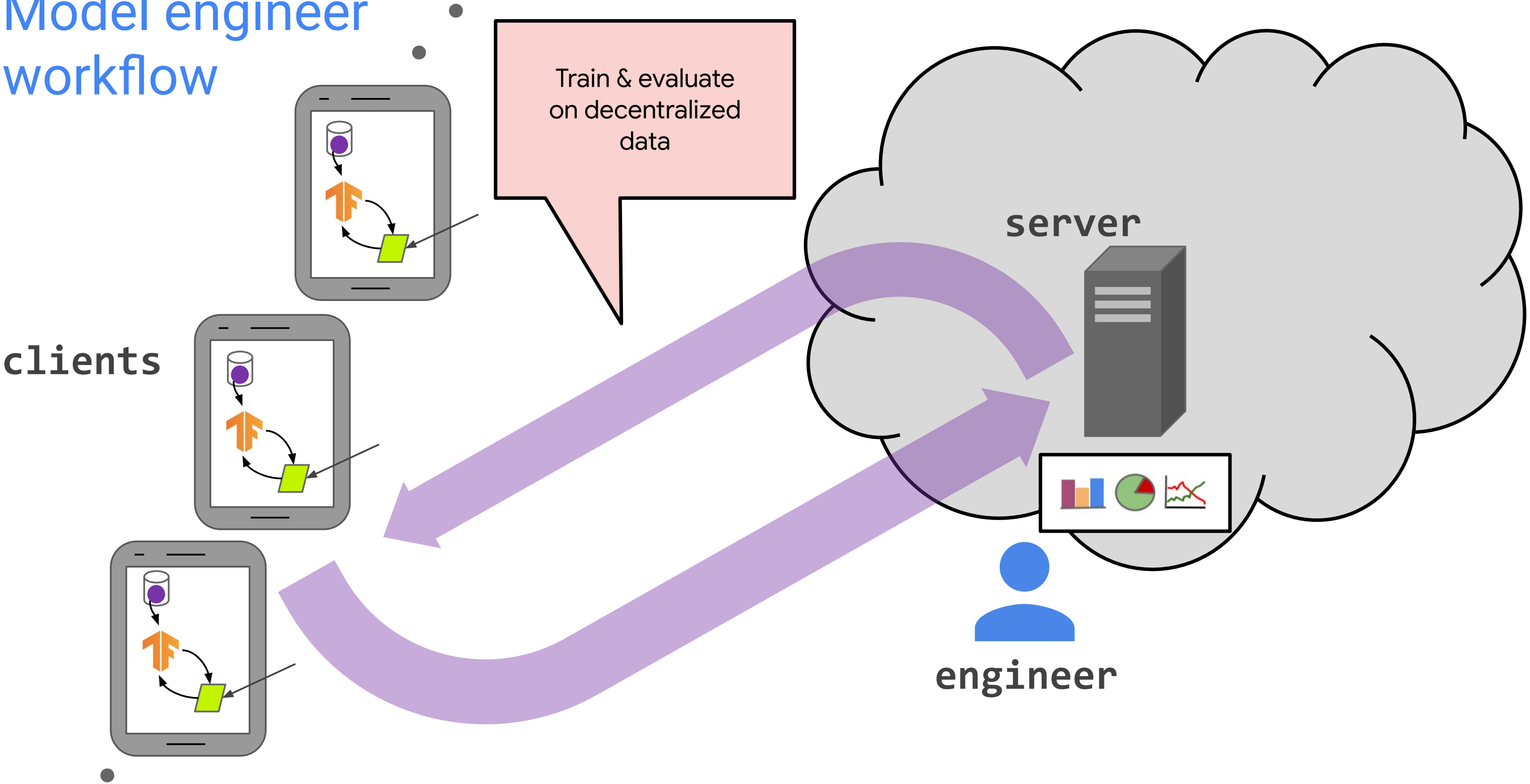


Model deployment workflow

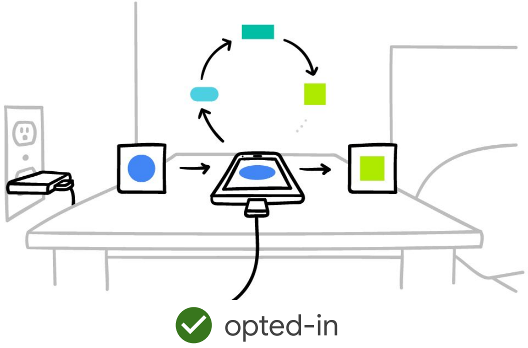


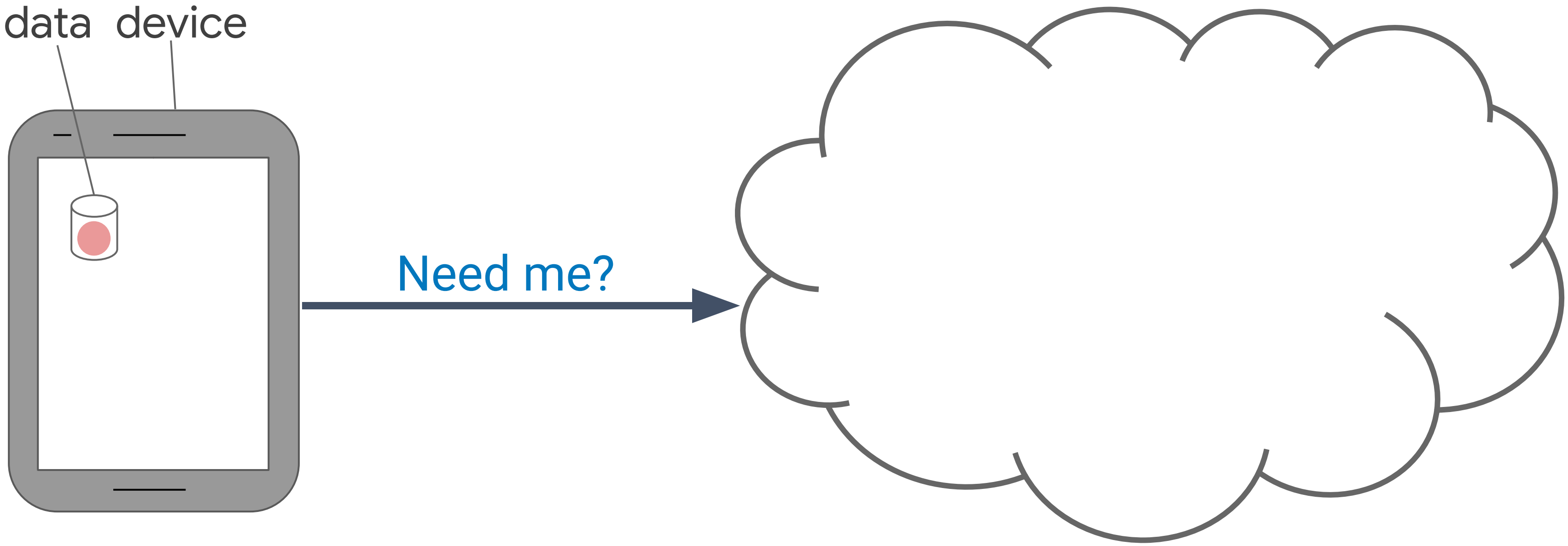
Google

Model engineer workflow

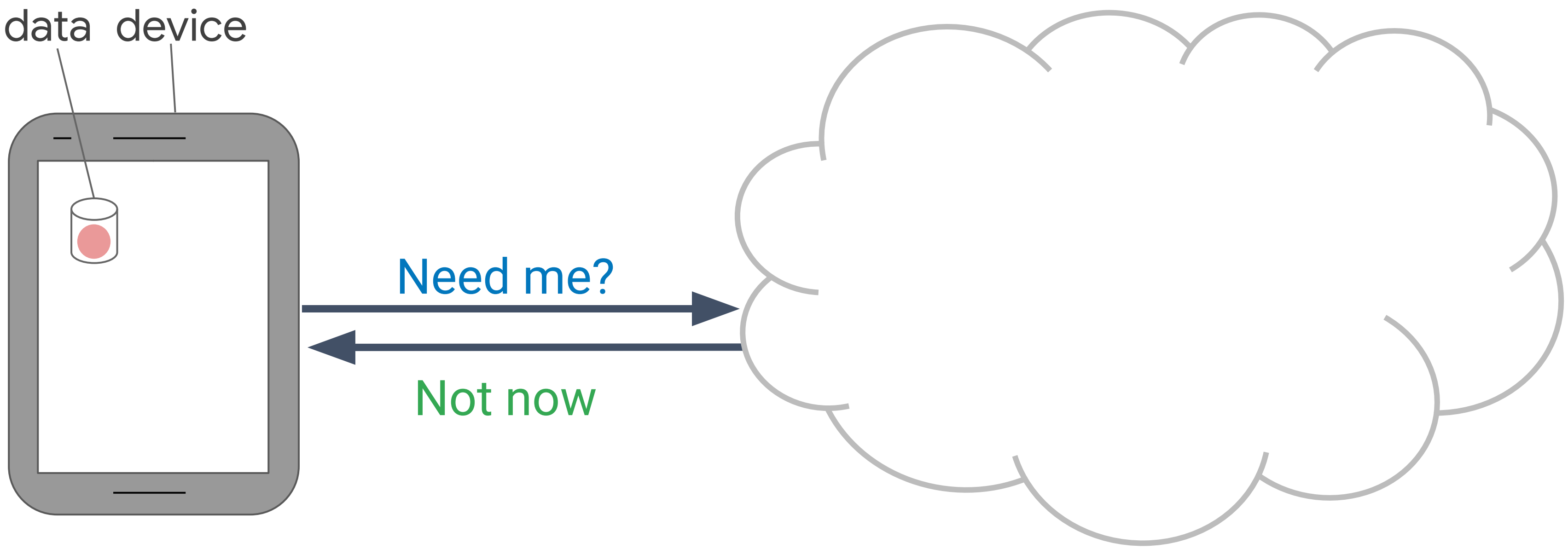


Federated learning



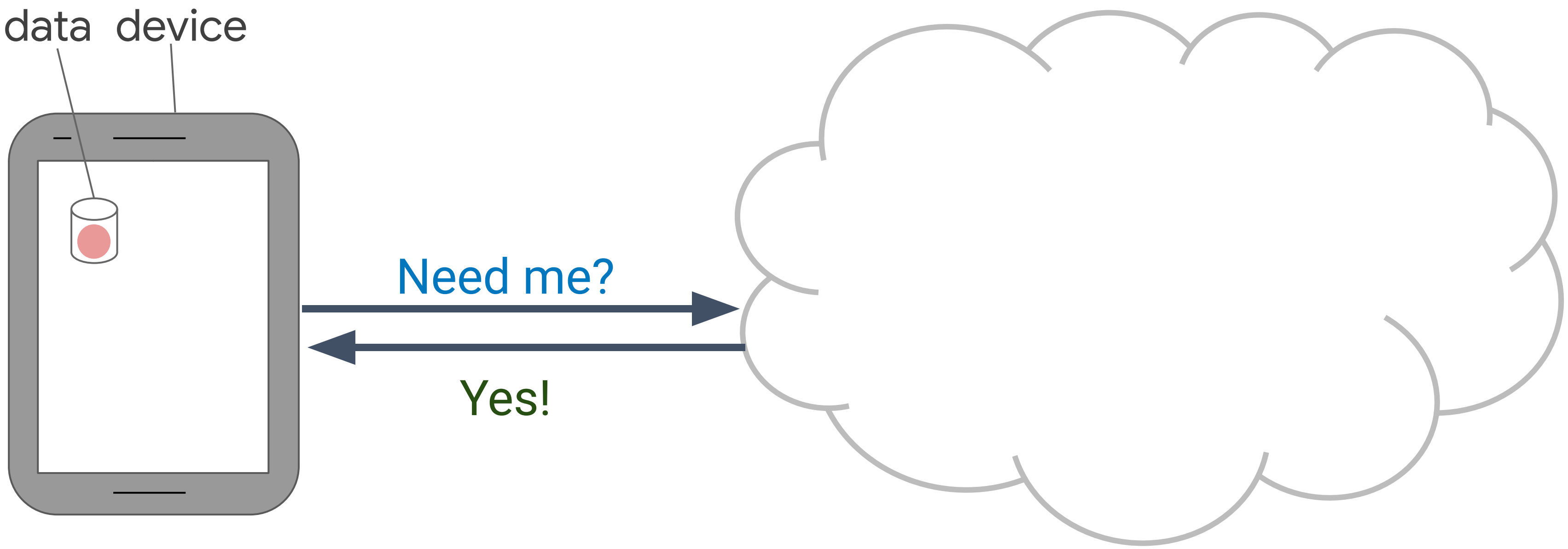


Federated learning



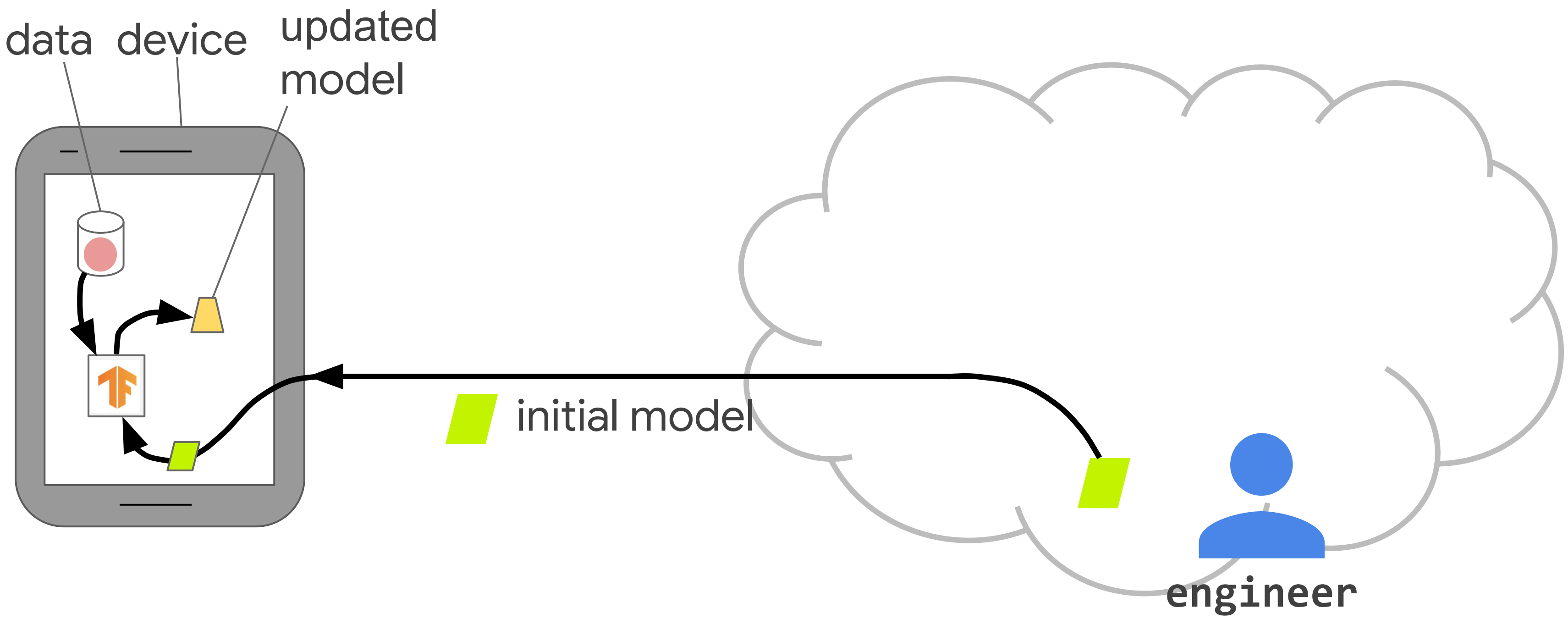
Google

Federated learning

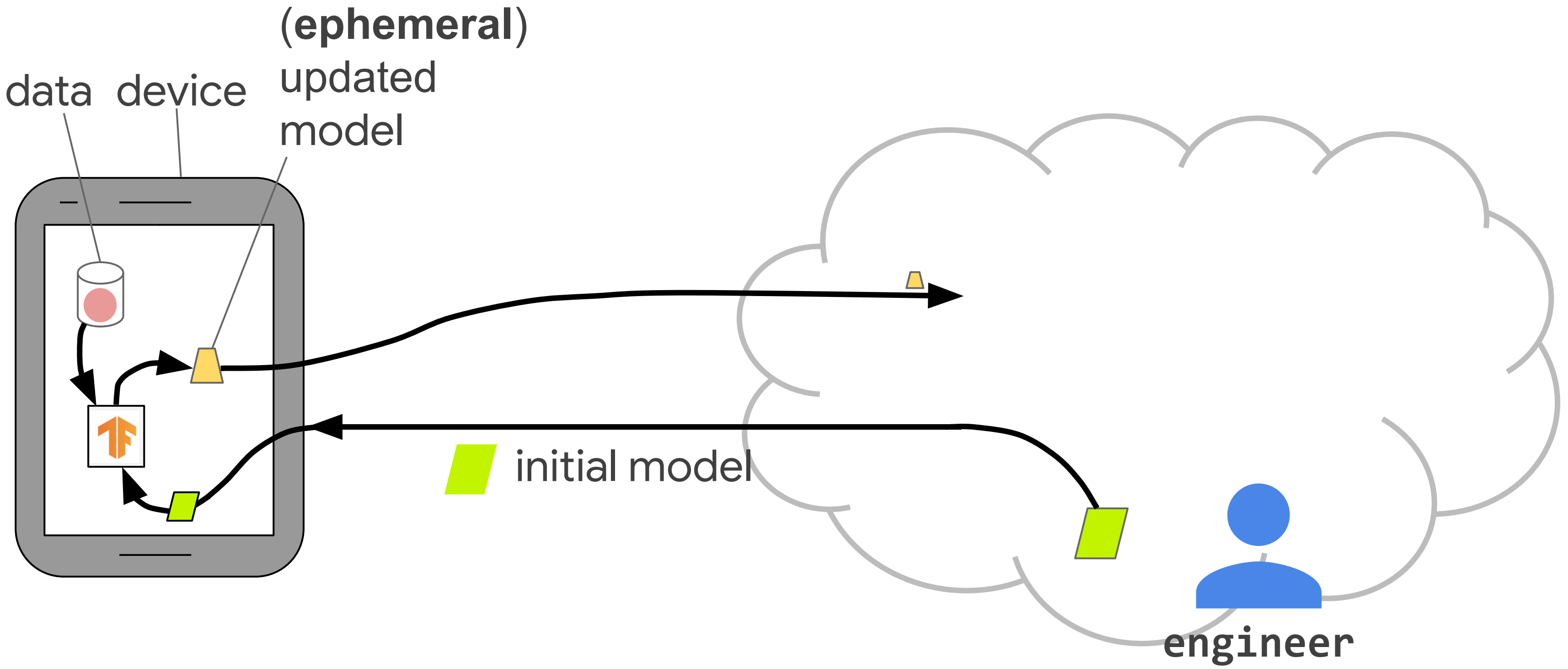


Google

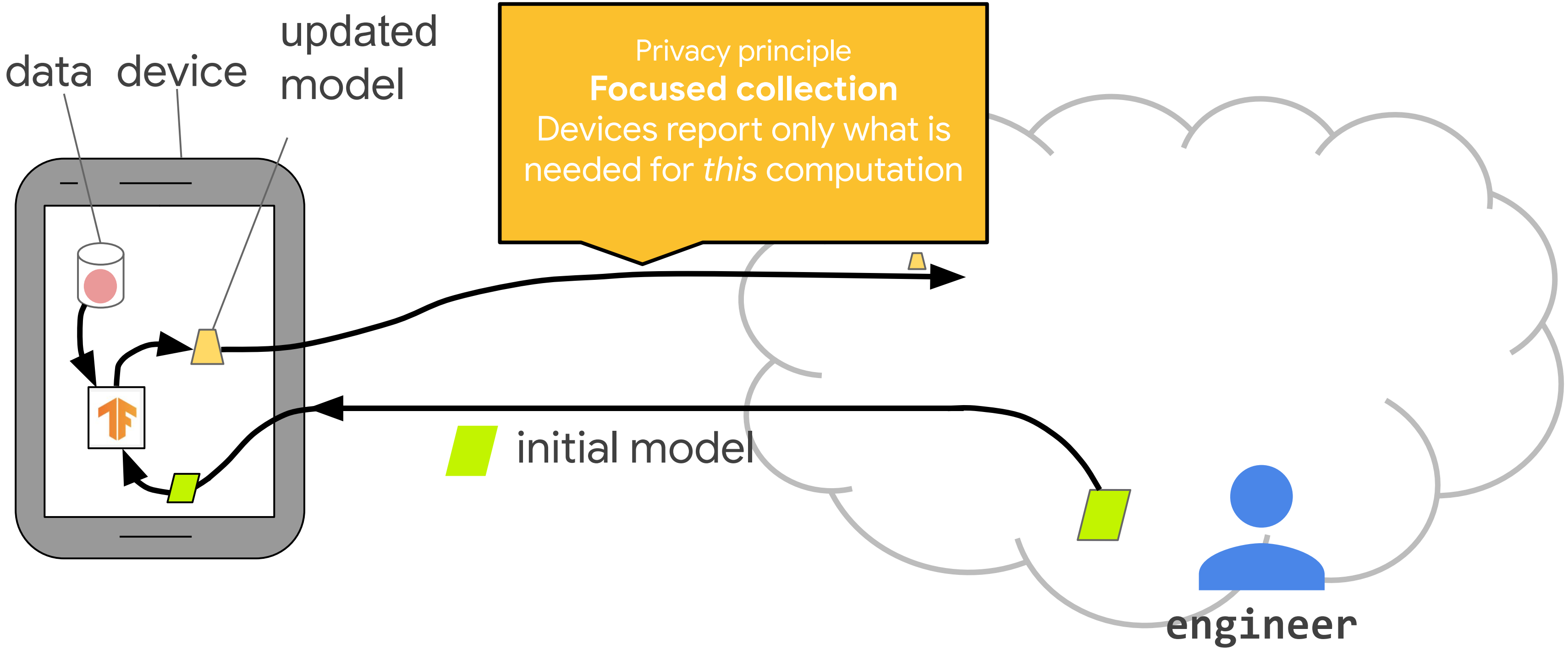
Federated learning



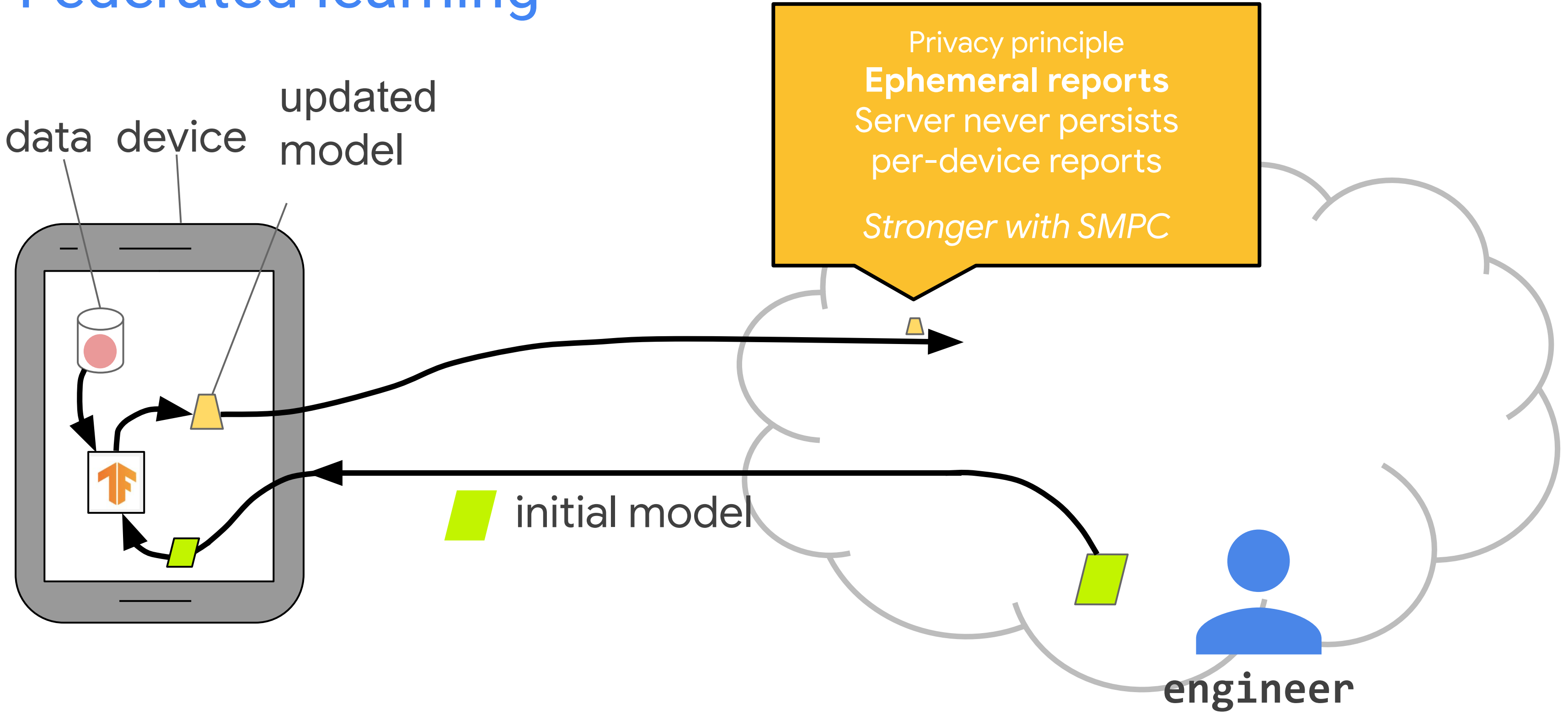
Federated learning



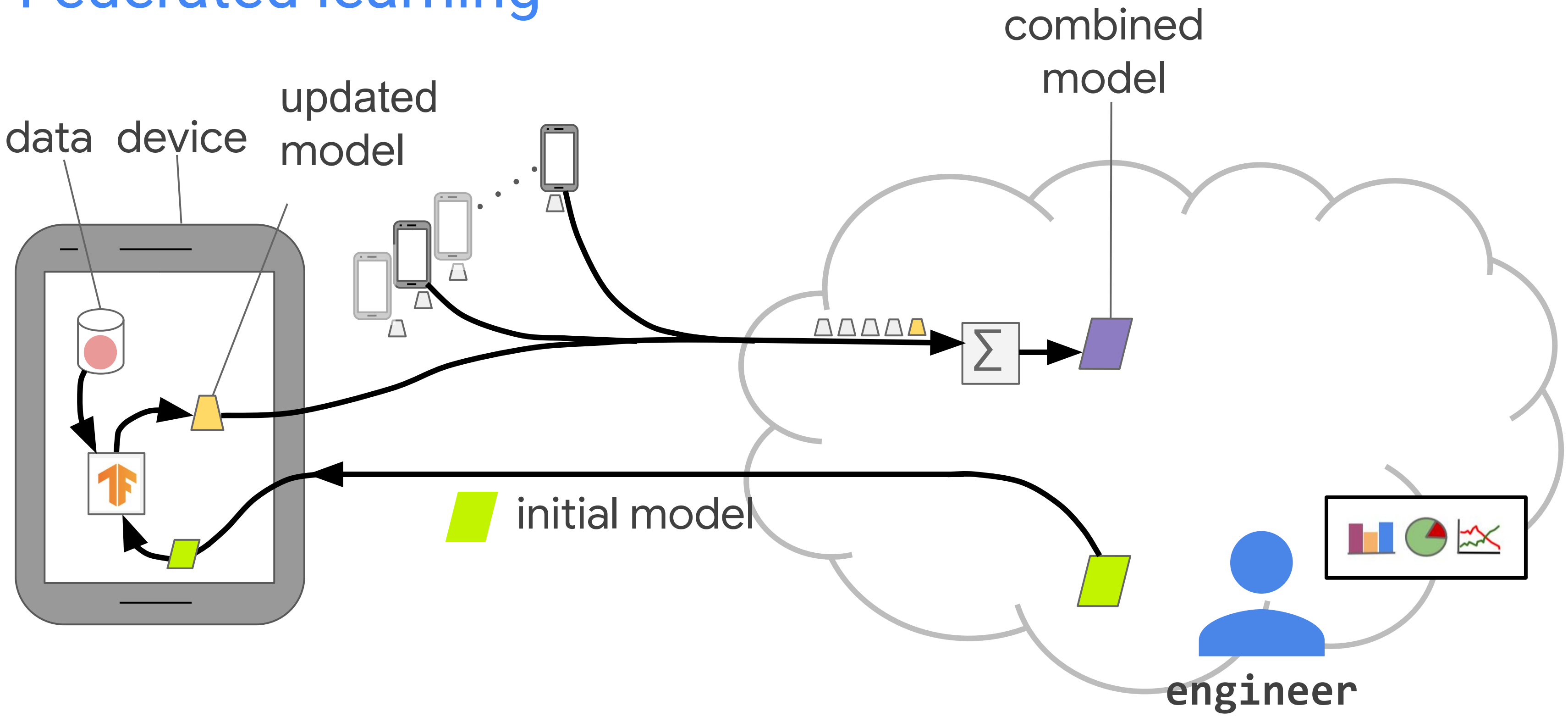
Federated learning



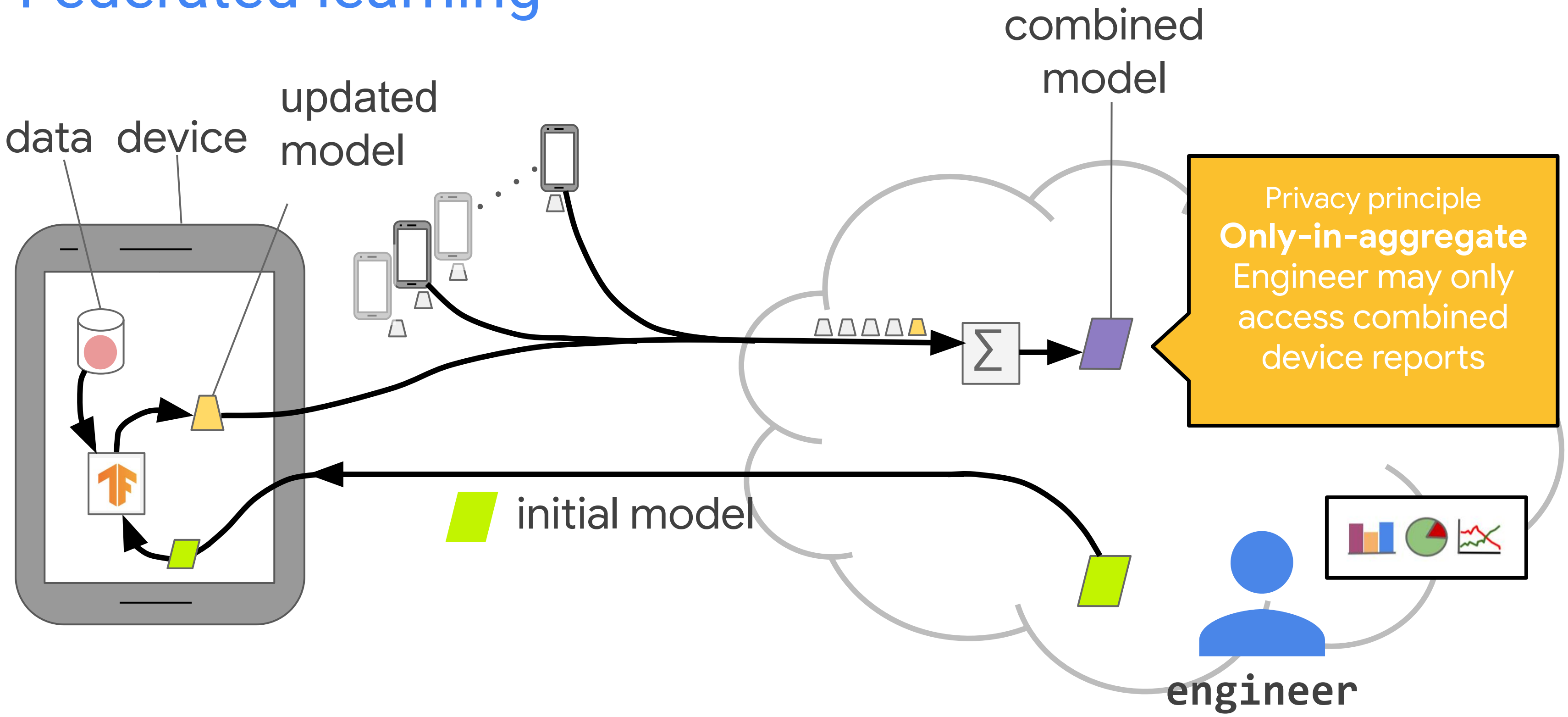
Federated learning



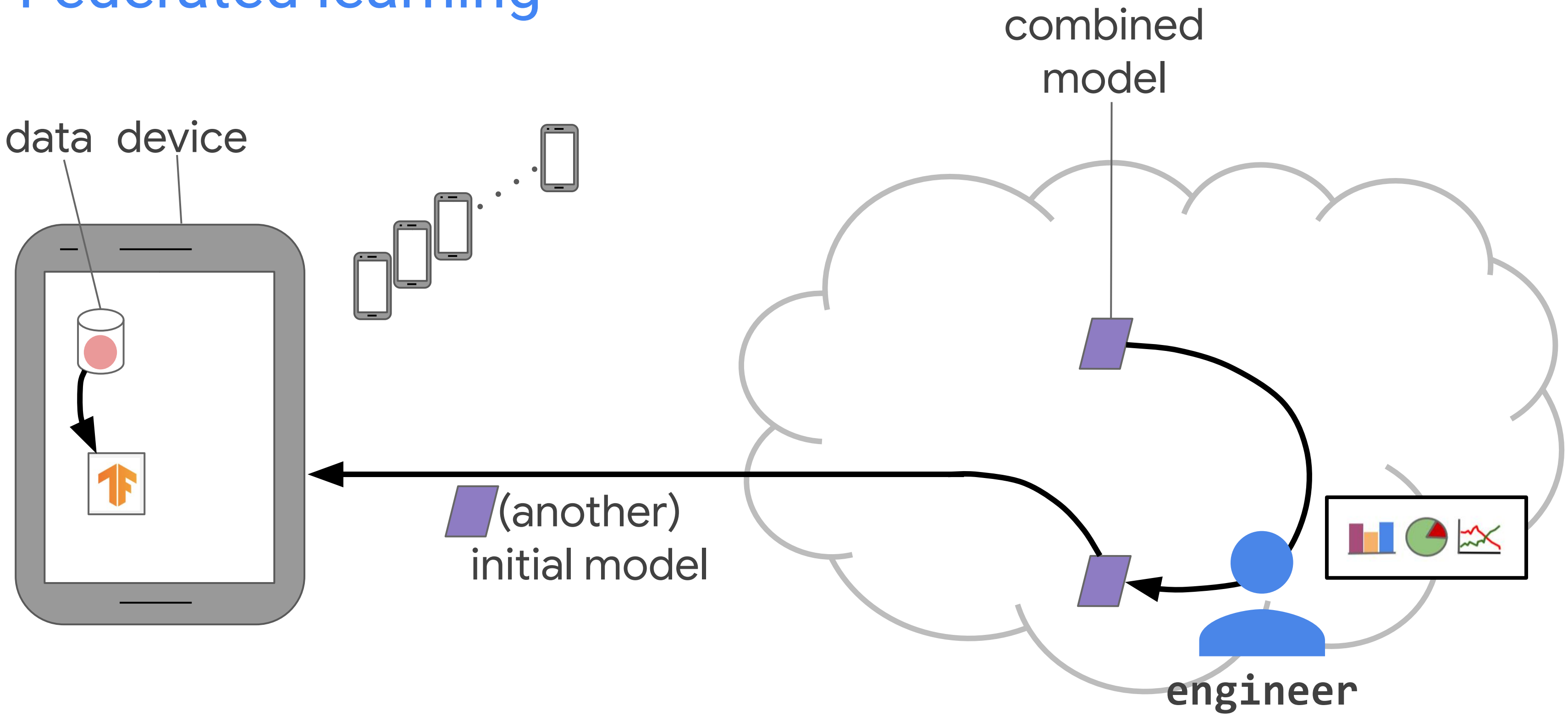
Federated learning



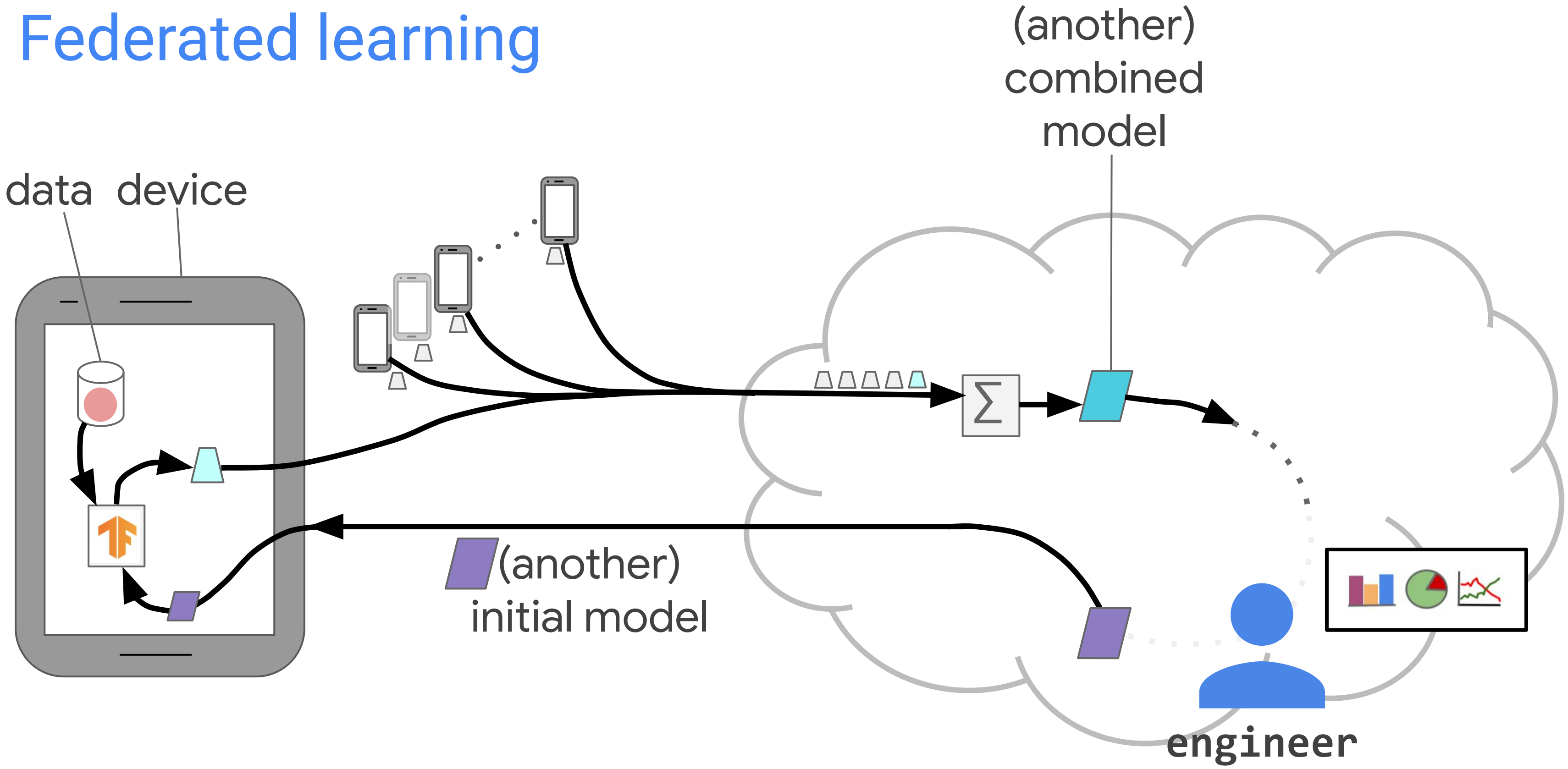
Federated learning



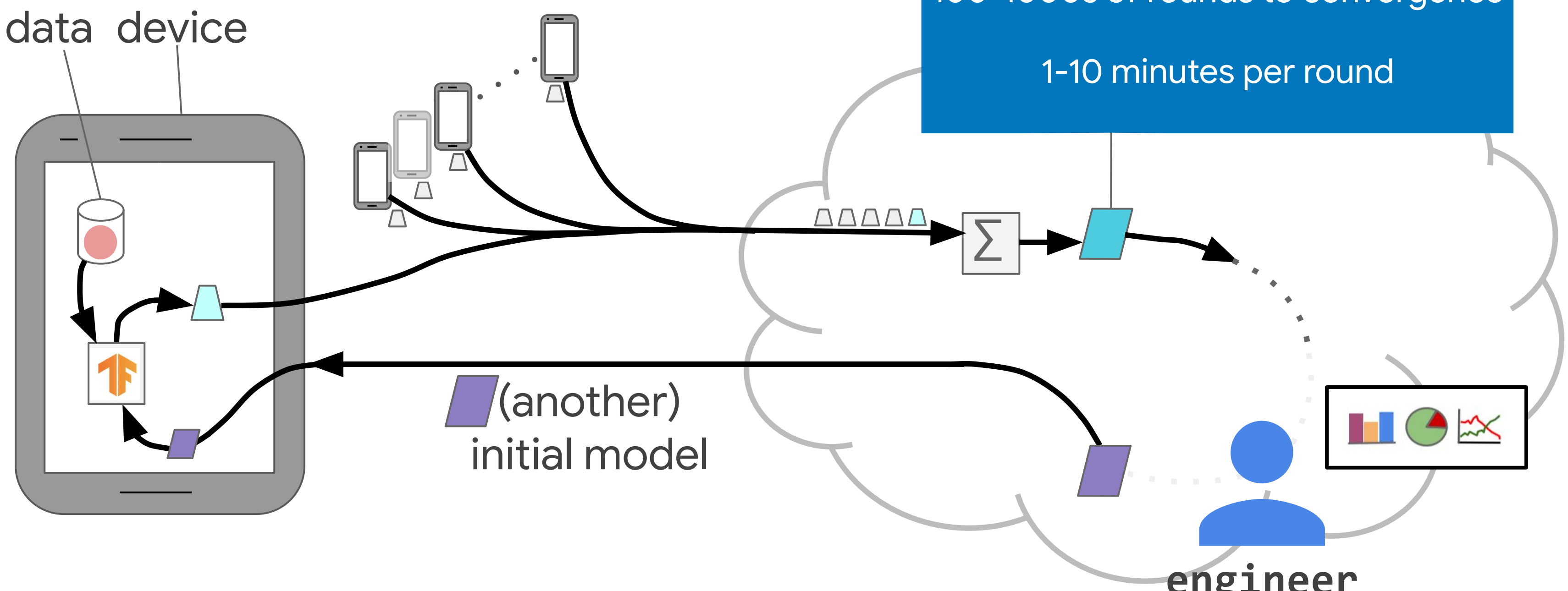
Federated learning



Federated learning



Federated learning



Typical orders-of-magnitude

100-1000s of users per round

100-1000s of rounds to convergence

1-10 minutes per round

Characteristics of **federated learning**

vs. traditional *distributed learning*

Data locality and distribution
- **massively decentralized, naturally arising (non-IID) partition**
- Data is siloed, held by a small number of coordinating entities
- *system-controlled* (e.g. shuffled, balanced)

Data availability
- **limited availability, time-of-day variations**
- *almost all data nodes always available*

Addressability
- **data nodes are anonymous and interchangeable**
- *data nodes are addressable*

Node statefulness
- **stateless (generally no repeat computation)**
- *stateful*

Node reliability
- **unreliable (~10% failures)**
- *reliable*

Wide-area communication pattern
- **hub-and-spoke topology**
- peer-to-peer topology (fully decentralized)
- *none (centralized to one datacenter)*

Distribution scale
- **massively parallel (1e9 data nodes)**
- *single datacenter*

Primary bottleneck
- **communication**
- *computation*

Constraints in the federated setting

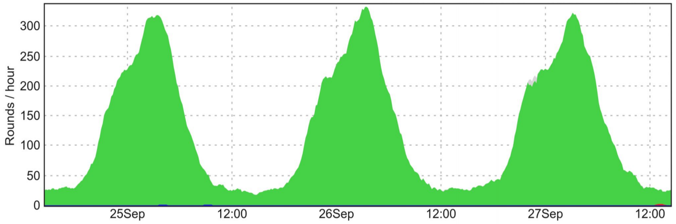
Each device reflects one users' data.

So no one device is representative of the whole population.

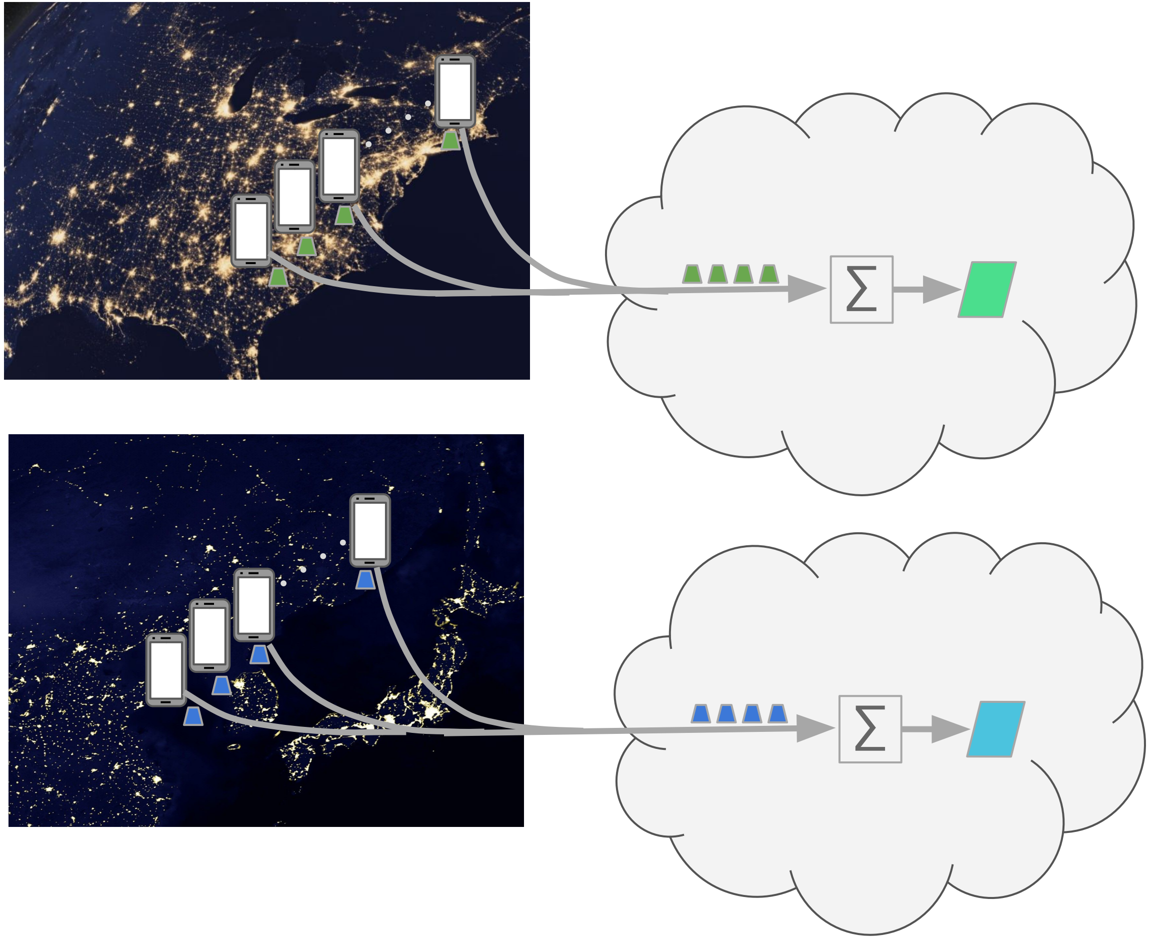
Devices must idle, plugged-in, on wi-fi to participate.

Device availability correlates with both geo location *and* data distribution.

Round completion rate by hour (US)

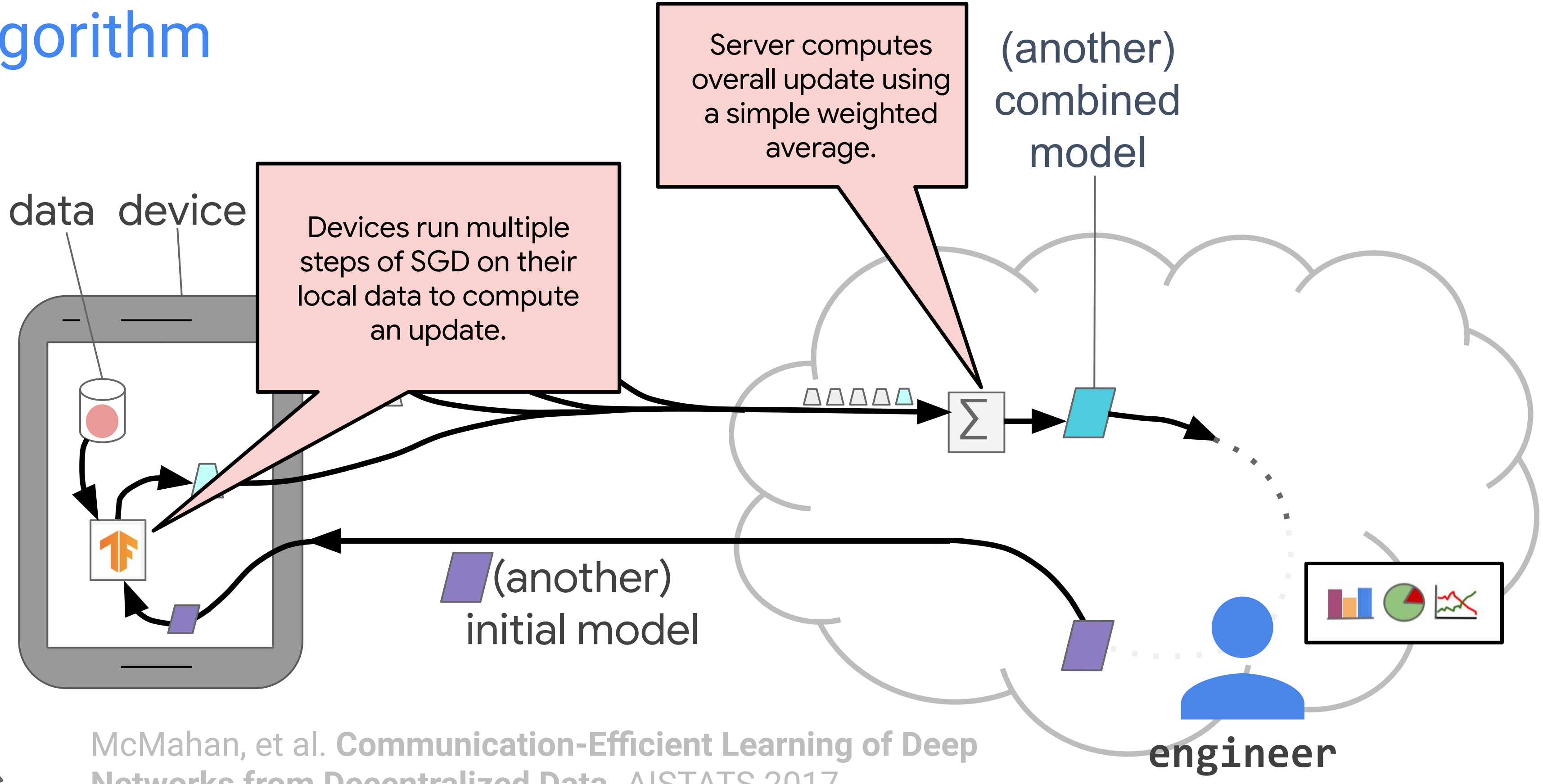


Hour (over three days)



H. Eichner, *et al*. **Semi-Cyclic Stochastic Gradient Descent**. *ICML 2019*.

Federated Averaging Algorithm



McMahan, et al. **Communication-Efficient Learning of Deep Networks from Decentralized Data.** AISTATS 2017.

The Federated Averaging algorithm

Server

Until Converged:

1. Select a random subset of clients

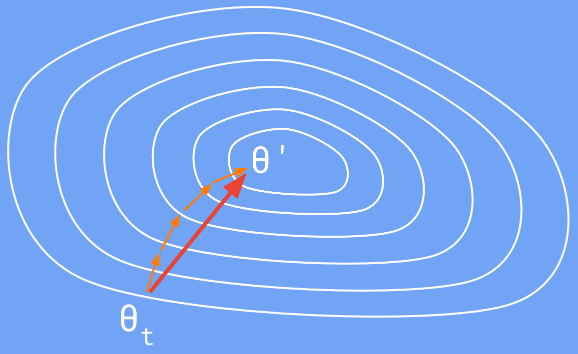
2. In parallel, send current parameters $\boldsymbol{\theta}_t$ to those clients

Selected Client k

1. Receive θ_t from server.

2. Run some number of minibatch SGD steps, producing θ'

3. Return $\theta' - \theta_t$ to server.

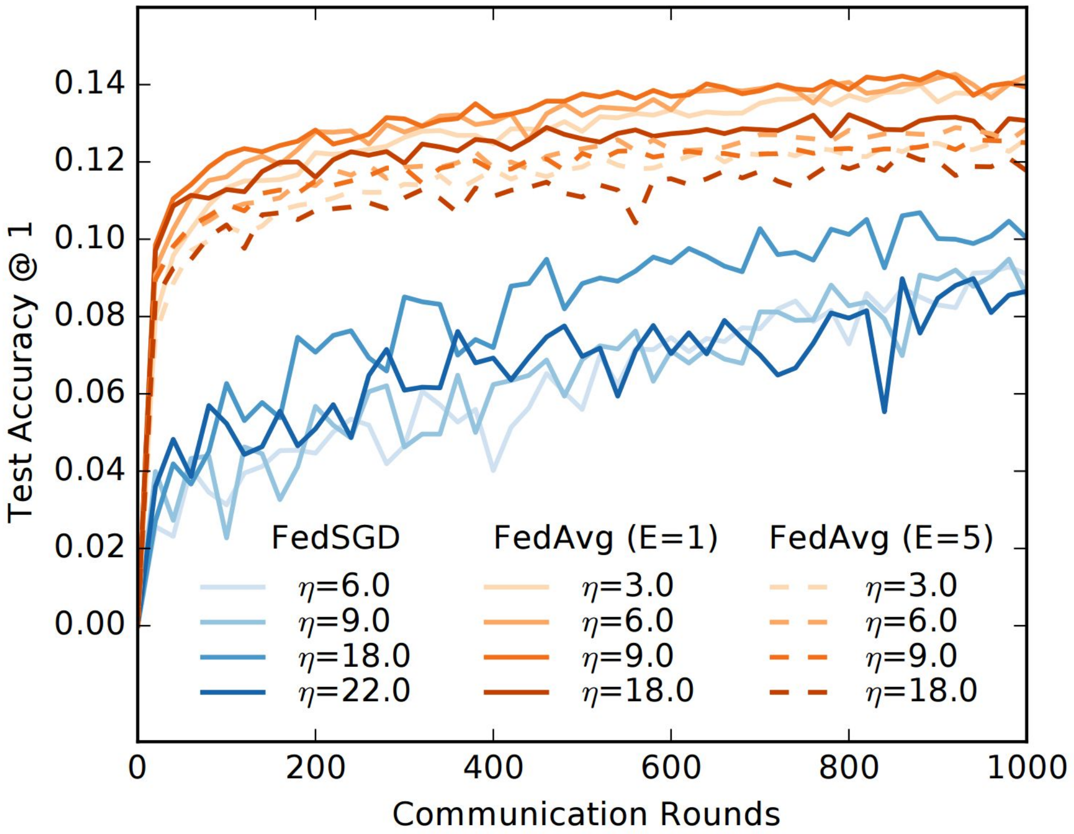


3. $\theta_{t+1} = \theta_t$ + data-weighted average of client updates

H. B. McMahan, *et al.* **Communication-Efficient Learning of Deep Networks from Decentralized Data.** AISTATS 2017

Large-scale LSTM for next-word prediction

Dataset: Large Social Network, 10m public posts, grouped by author.



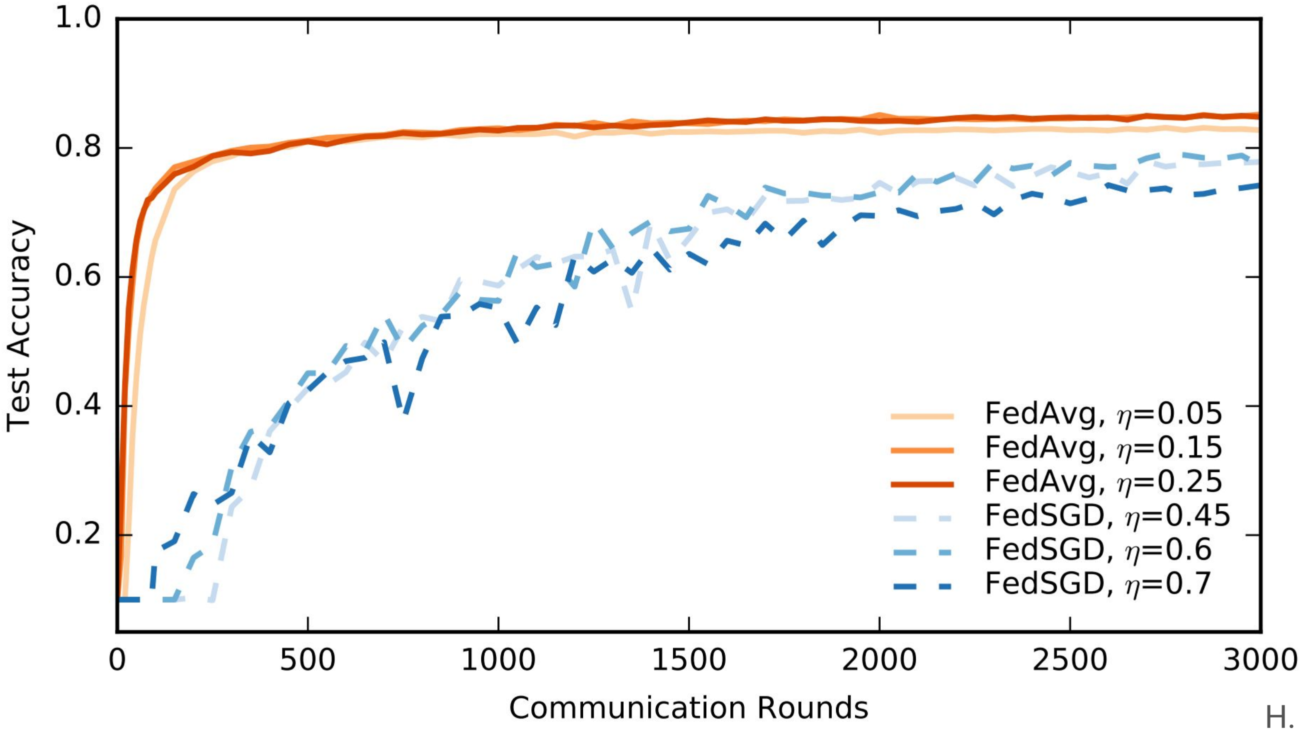
Rounds to reach 10.5% Accuracy

FedSGD	820
FedAvg	35

23x **decrease in communication rounds**

H. B. McMahan, *et al.* **Communication-Efficient Learning of Deep Networks from Decentralized Data.** AISTATS 2017

CIFAR-10 convolutional model



Updates to reach 82%	
SGD	31,000
FedSGD	6,600
FedAvg	630

49x decrease in communication (updates) vs SGD

(IID and balanced data)

H. B. McMahan, *et al*. **Communication-Efficient Learning of Deep Networks from Decentralized Data.** AISTATS 2017

Federated learning at Google

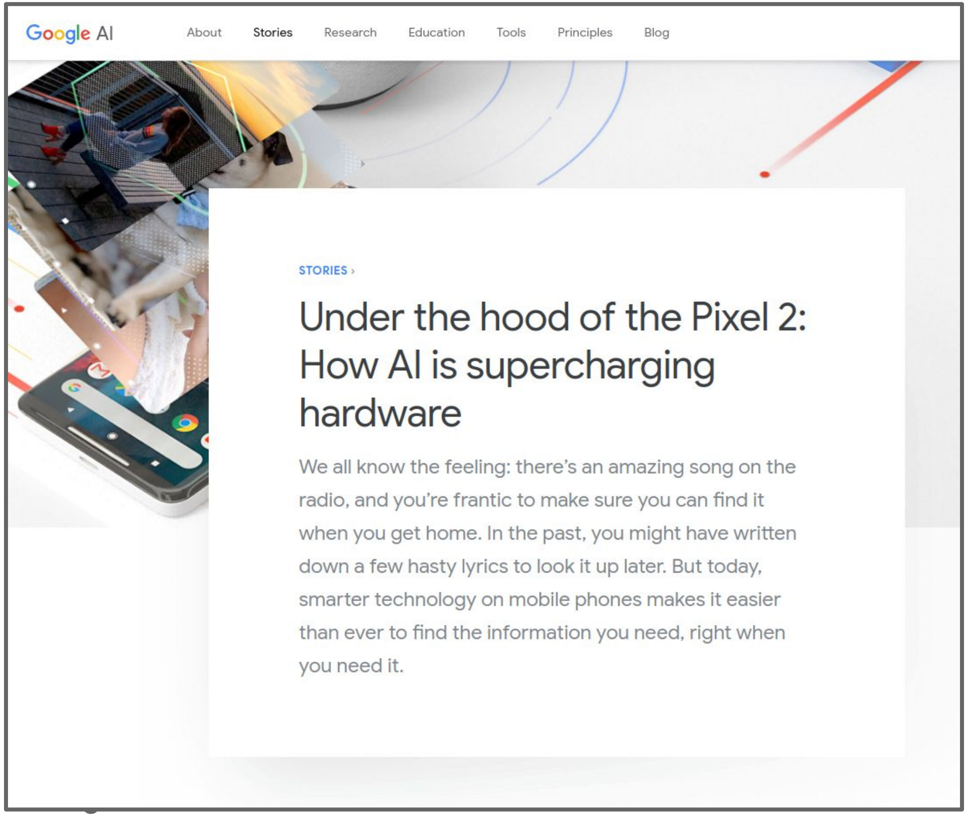
Federated learning at Google

0.5B installs

Daily use by multiple teams, dozens of ML engineers

Powering features in Gboard and on Pixel Devices

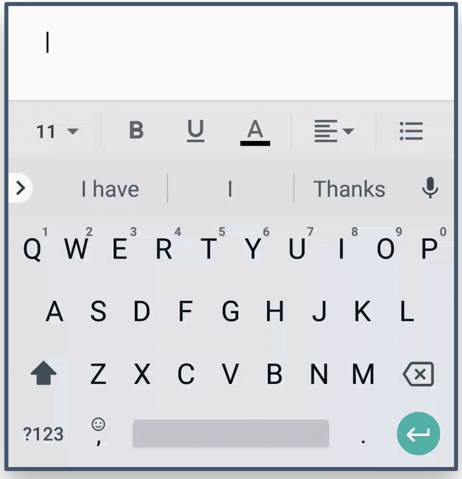
Federated Learning on Pixel Phones

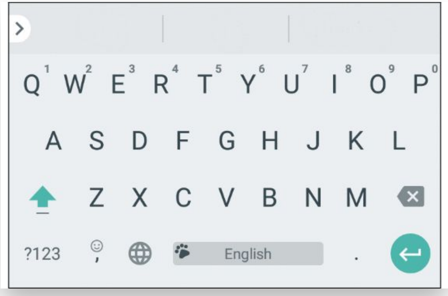


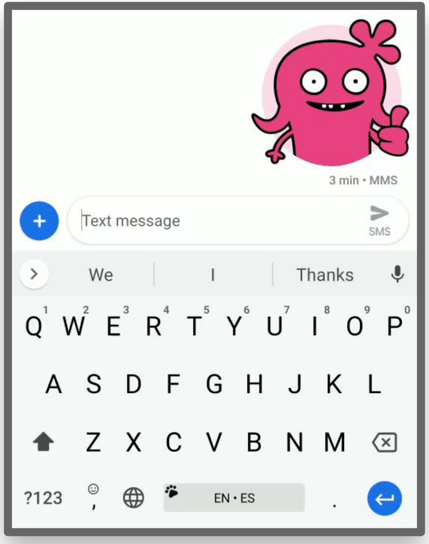
We're already using Federated Learning to improve several Google products. The Pixel first and second generation phones, for example, use Federated Learning to surface more accurate, useful settings search results so that people can find what that they're looking for faster. The Pixel has thousands of settings to adjust, from font size and brightness to app preferences and battery use. Different settings apply to different people and use cases, so personalizing users' experiences with machine learning can help people more easily find the one that they care about.

By using Federated Learning, the team replaced a hard-coded ranking system with a model that was trained on mobile phone usage. True to the Federated Learning model, each phone contributed improvements to the global model without sending any training data to Google's servers. "Federated learning helps us improve your experience using Pixel while keeping data from your interaction with your phone private," says Research Scientist Daniel Ramage.

Gboard: mobile keyboard





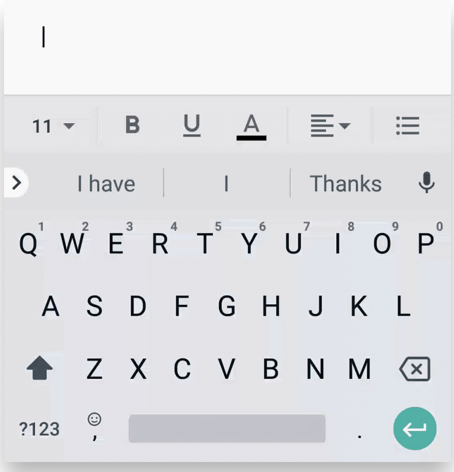


Gboard: language modeling

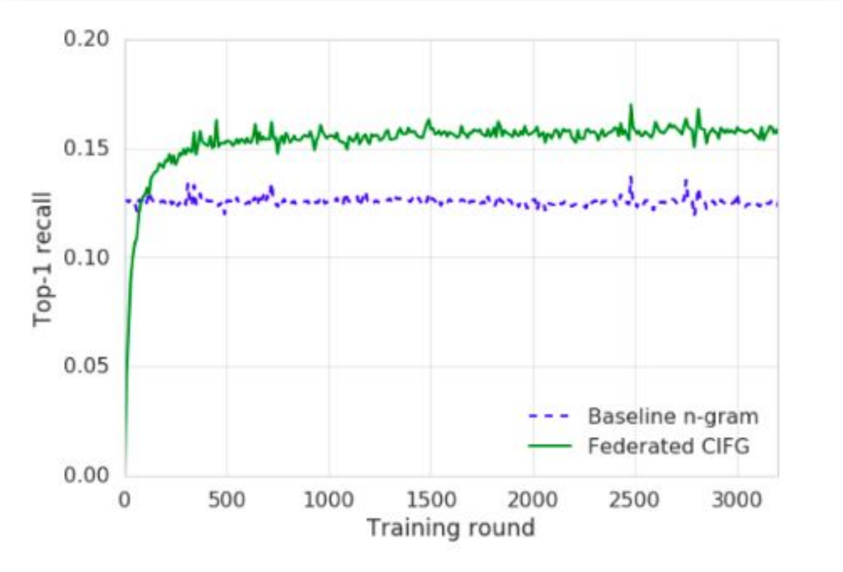
- Predict the next word based on typed text so far
- Powers the predictions strip

When should you consider federated learning?

- On-device data is more relevant than server-side proxy data
- On-device data is privacy sensitive or large
- Labels can be inferred naturally from user interaction



Gboard: language modeling



Federated model
compared to baseline

A. Hard, *et al*. **Federated Learning for Mobile Keyboard Prediction**. *arXiv:1811.03604*

Gboard: language modeling

Federated RNN (compared to prior n-gram model):

- Better next-word prediction accuracy: +24%
- More useful prediction strip: +10% more clicks

Other federated models in Gboard

Emoji prediction

- 7% more accurate emoji predictions
- prediction strip clicks +4% more
- 11% more users share emojis!

Ramaswamy, *et al*. **Federated Learning for Emoji Prediction in a Mobile Keyboard**. arXiv:1906.04329.

Action prediction

When is it useful to suggest a gif, sticker, or search query?

- 47% reduction in unhelpful suggestions
- increasing overall emoji, gif, and sticker shares

T. Yang, *et al*. **Applied Federated Learning: Improving Google Keyboard Query Suggestions**. *arXiv:1812.02903*

Discovering new words

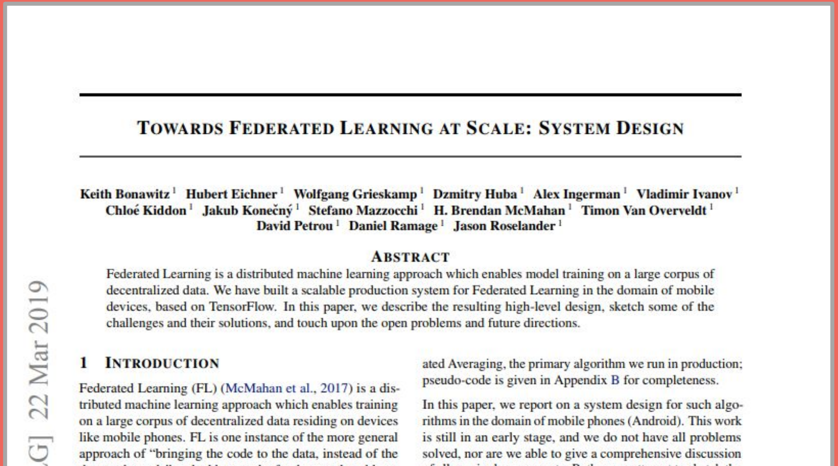
Federated discovery of what words people are typing that Gboard doesn't know.

M. Chen, *et al*. **Federated Learning Of Out-Of-Vocabulary Words**. *arXiv:1903.10635*

Google's federated system

Towards Federated Learning at Scale: System Design

ai.google/research/pubs/pub47976



TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN

Keith Bonawitz [1] Hubert Eichner [1] Wolfgang Grieskamp [1] Dzmitry Huba [1] Alex Ingerman [1] Vladimir Ivanov [1] Chloé Kiddon [1] Jakub Konečný [1] Stefano Mazzocchi [1] H. Brendan McMahan [1] Timon Van Overveldt [1] David Petrou [1] Daniel Ramage [1] Jason Roselander [1]

ABSTRACT

Federated Learning is a distributed machine learning approach which enables model training on a large corpus of decentralized data. We have built a scalable production system for Federated Learning in the domain of mobile devices, based on TensorFlow. In this paper, we describe the resulting high-level design, sketch some of the challenges and their solutions, and touch upon the open problems and future directions.

1 INTRODUCTION

Federated Learning (FL) (McMahan et al., 2017) is a distributed machine learning approach which enables training on a large corpus of decentralized data residing on devices like mobile phones. FL is one instance of the more general approach of "bringing the code to the data, instead of the

ated Averaging, the primary algorithm we run in production; pseudo-code is given in Appendix B for completeness.

In this paper, we report on a system design for such algorithms in the domain of mobile phones (Android). This work is still in an early stage, and we do not have all problems solved, nor are we able to give a comprehensive discussion

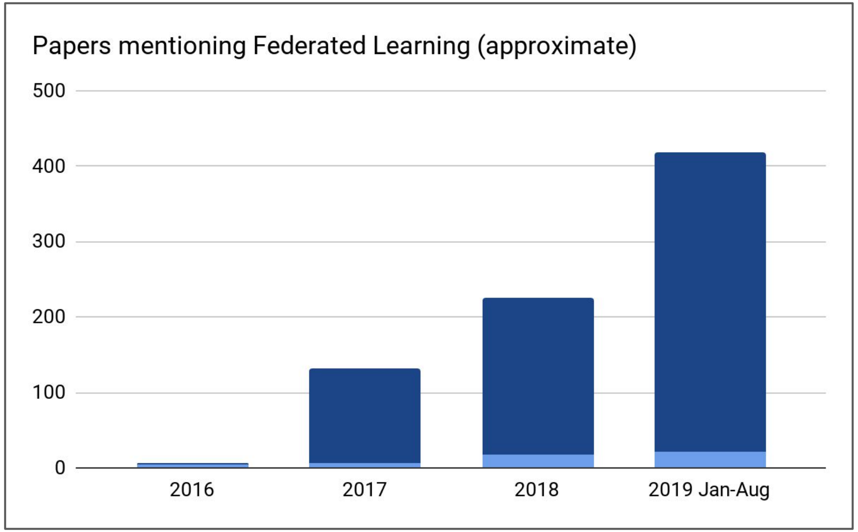
K. Bonawitz, *et al*. **Towards Federated Learning at Scale: System Design**. *SysML 2019.*

Federated learning beyond Google

Google

Federated Learning Research

FL Workshop in Seattle 6/17-18



Federated learning workshops:

- **NeurIPS 2019 (upcoming) Workshop on Federated Learning for Data Privacy and Confidentiality**
  Submission deadline Sept. 9
- IJCAI 2019 Workshop on Federated Machine Learning for User Privacy and Data Confidentiality (FML'19)
- Google-hosted June 2019
  ~40 faculty, 20 students, 40 Googlers

TensorFlow Federated

Experiment with federated technologies
in simulation

TensorFlow Federated
What's in the box?

Federated Learning (FL) API

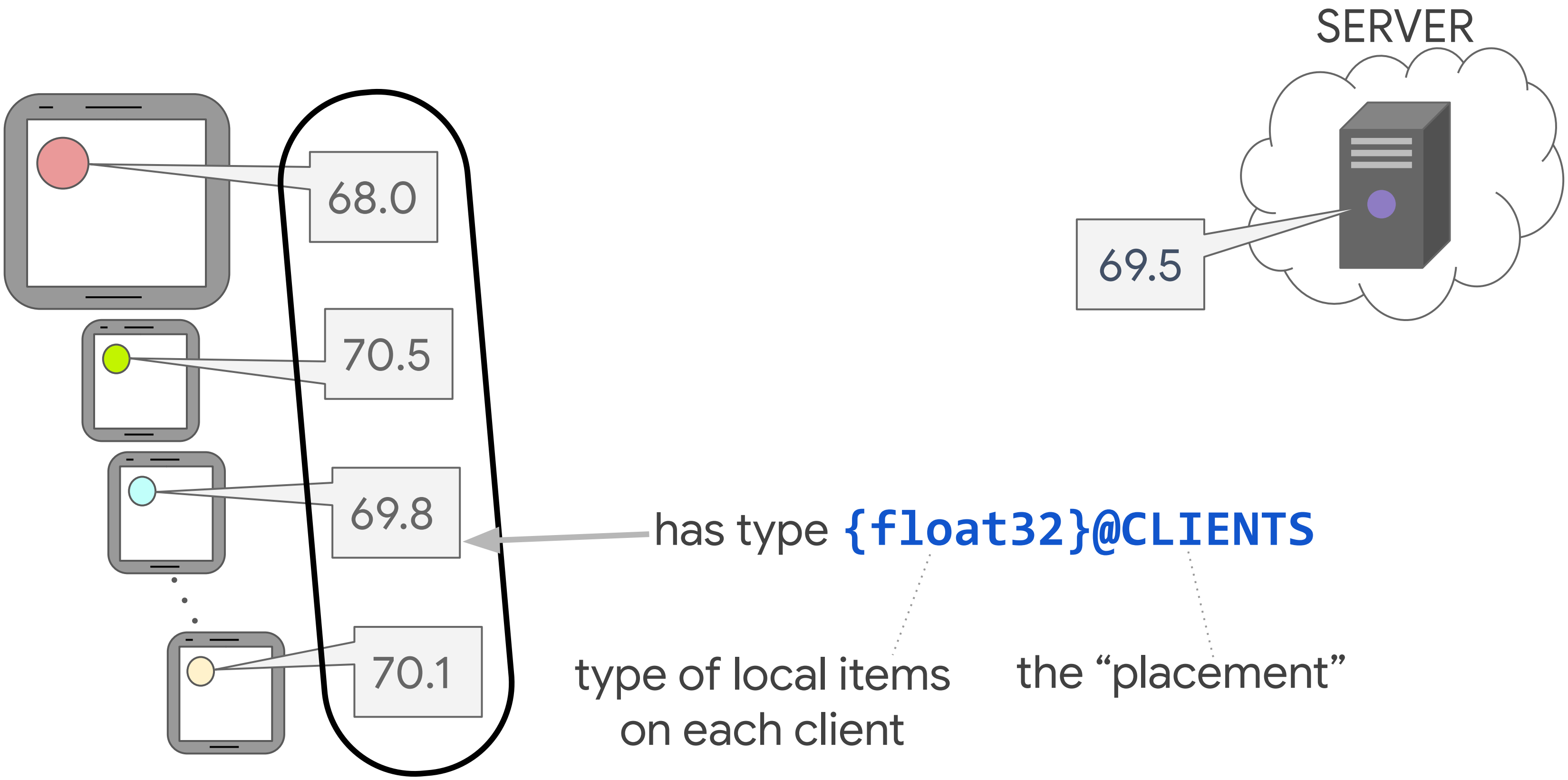
- Implementations of federated training/evaluation
- Can be immediately applied to existing TF models/data

Federated Core (FC) API

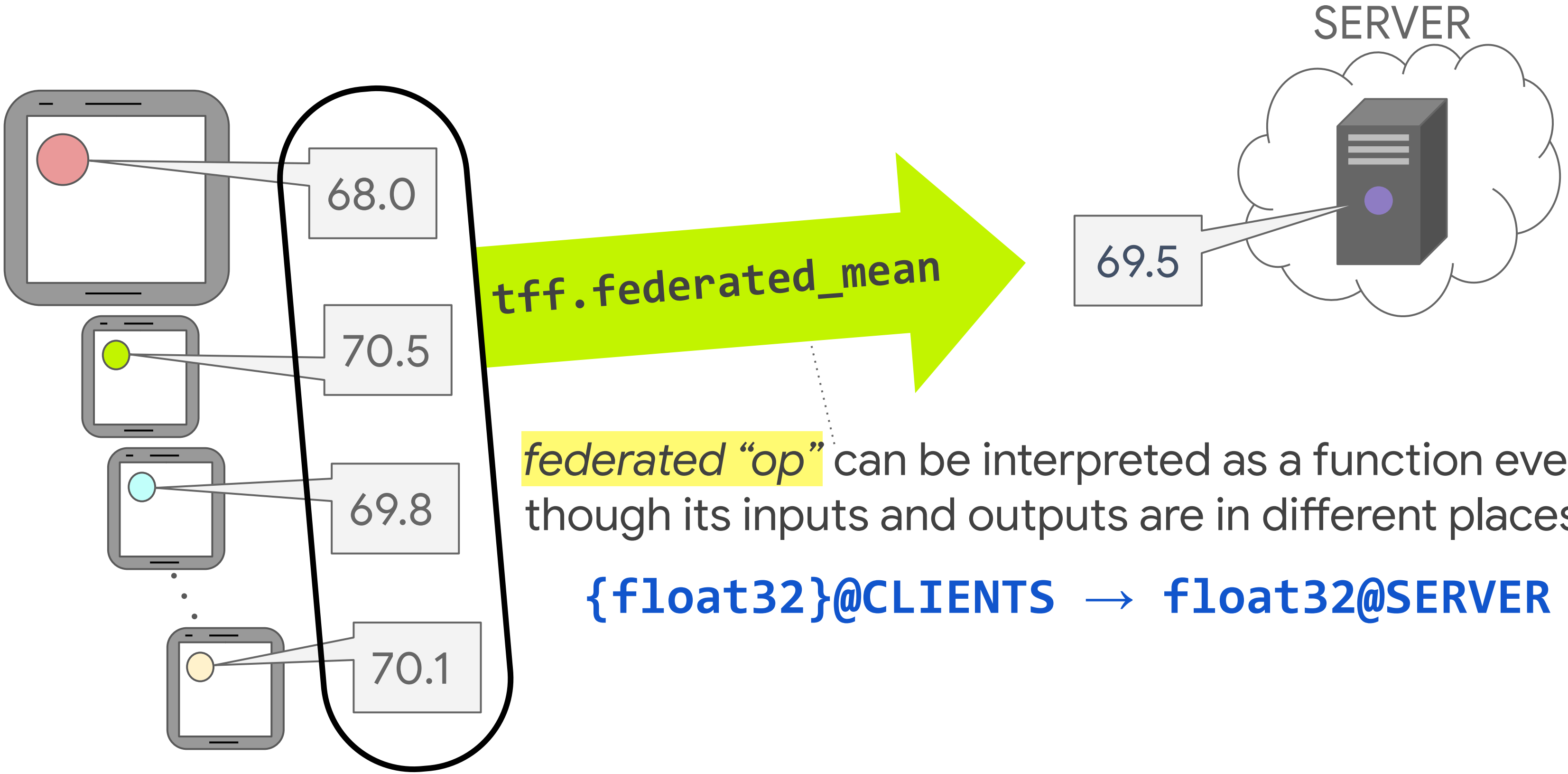
- Lower-level building blocks for expressing new federated algorithms

Local runtime for simulations

Federated computation in TFF



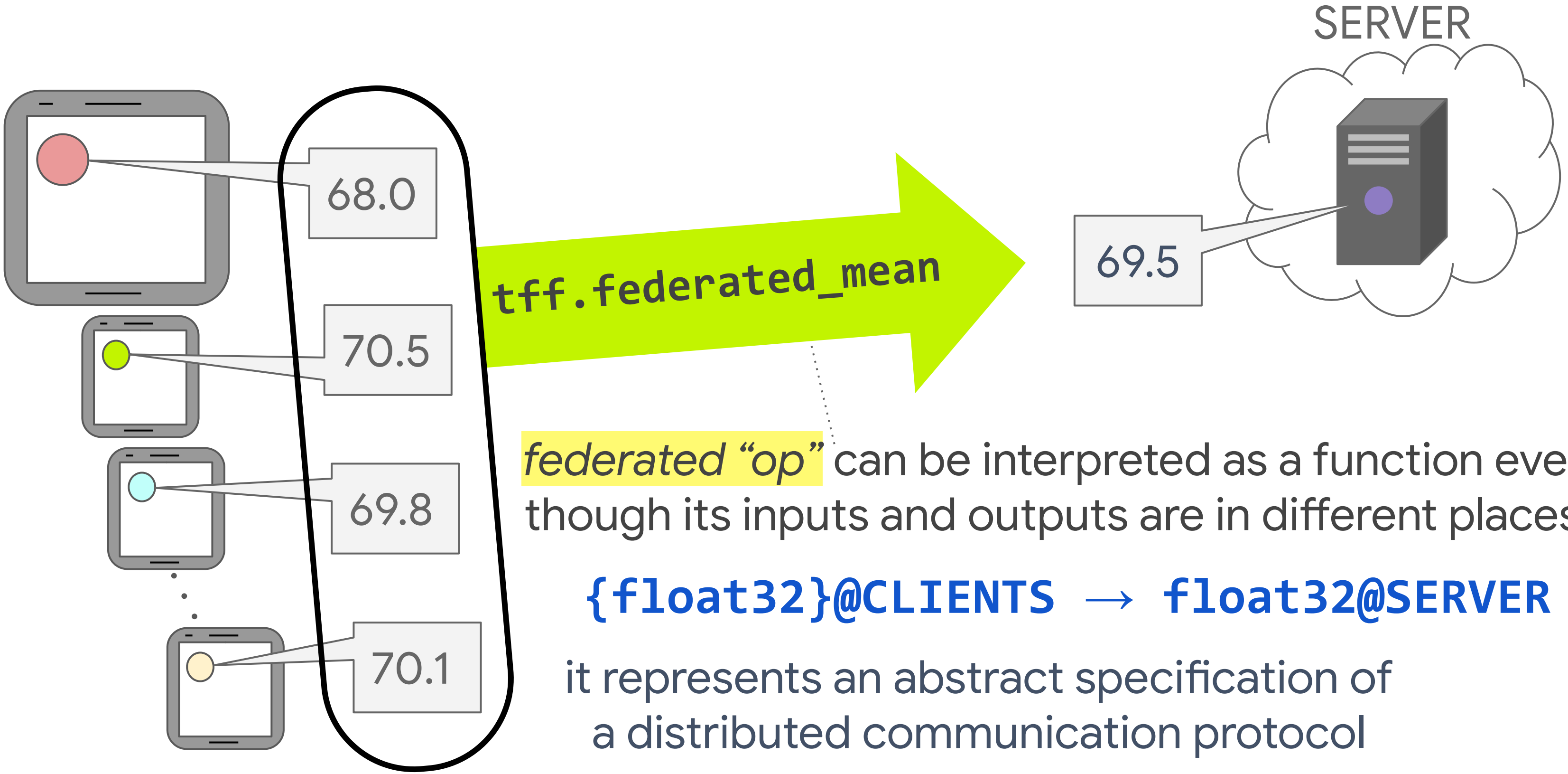
Federated computation in TFF



federated "op" can be interpreted as a function even though its inputs and outputs are in different places

`{float32}@CLIENTS → float32@SERVER`

Federated computation in TFF

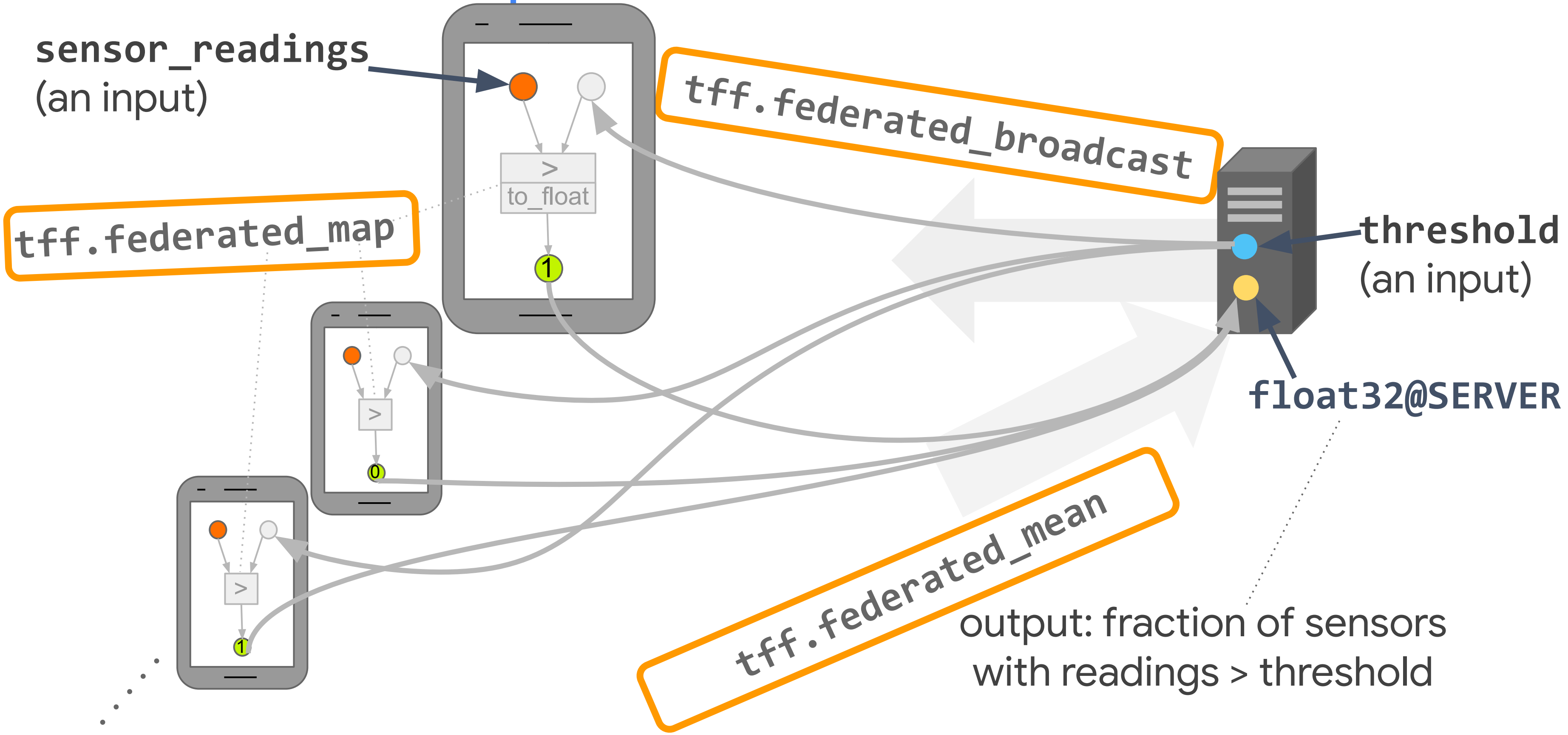


federated "op" can be interpreted as a function even though its inputs and outputs are in different places

`{float32}@CLIENTS → float32@SERVER`

it represents an abstract specification of a distributed communication protocol

Federated computation in TFF



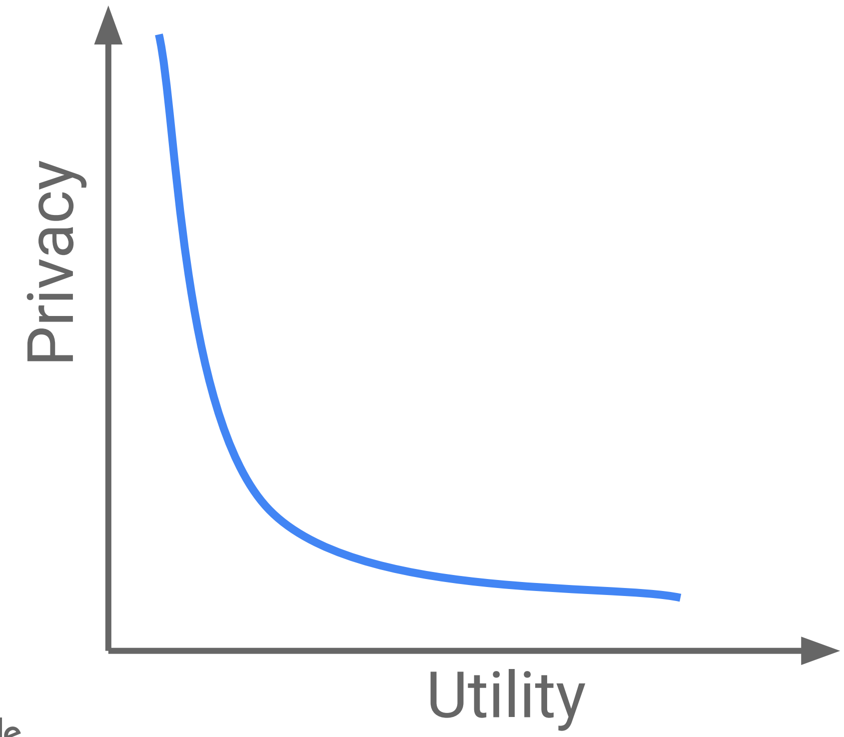
```python
THRESHOLD_TYPE = tff.FederatedType(tf.float32, tff.SERVER)

@tff.federated_computation(READINGS_TYPE, THRESHOLD_TYPE)
def get_fraction_over_threshold(readings, threshold):

  @tff.tf_computation(tf.float32, tf.float32)
  def _is_over_as_float(val, threshold):
    return tf.to_float(val > threshold)

  return tff.federated_mean(
      tff.federated_map(_is_over_as_float, [
          readings, tff.federated_broadcast(threshold)]))
```

For more TFF tutorials visit...

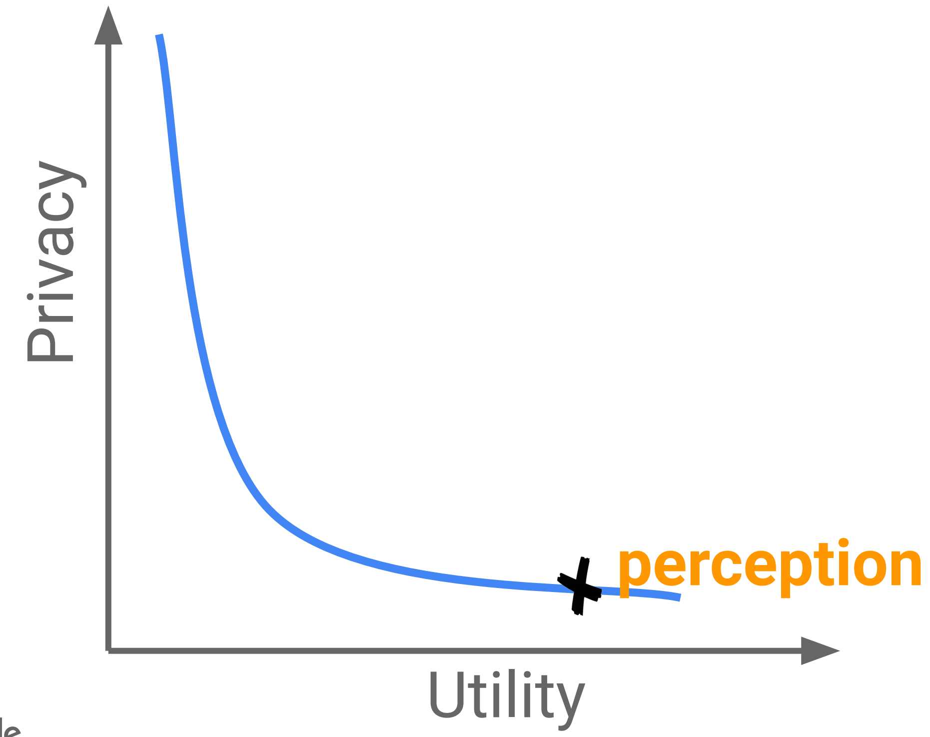
tensorflow.org/federated

Federated learning and privacy

ML on Sensitive Data: Privacy versus Utility

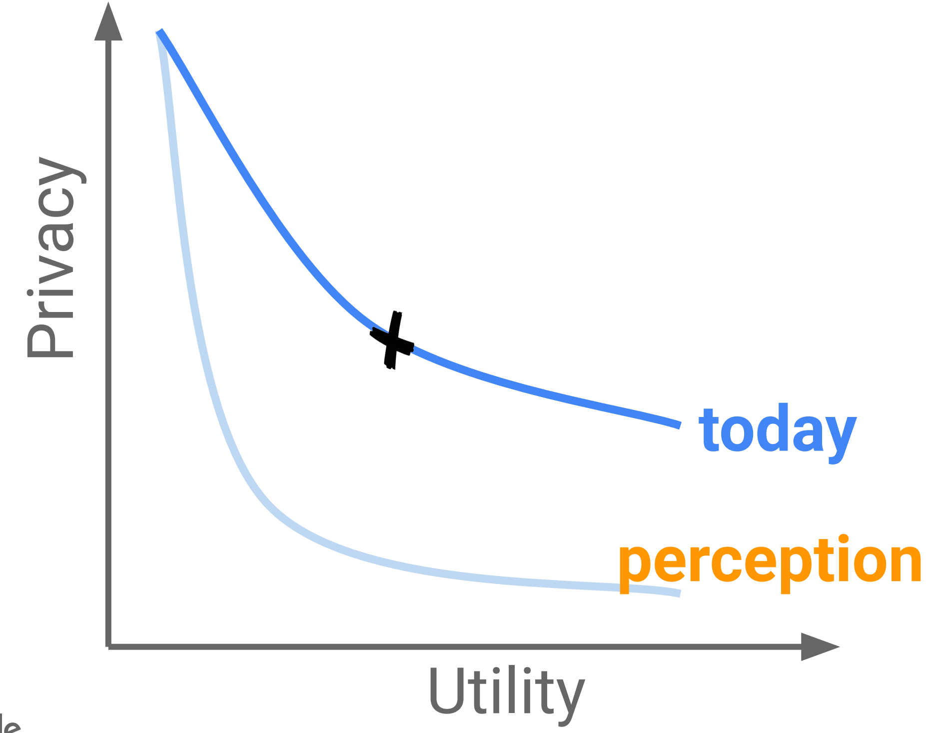


ML on Sensitive Data: Privacy versus Utility



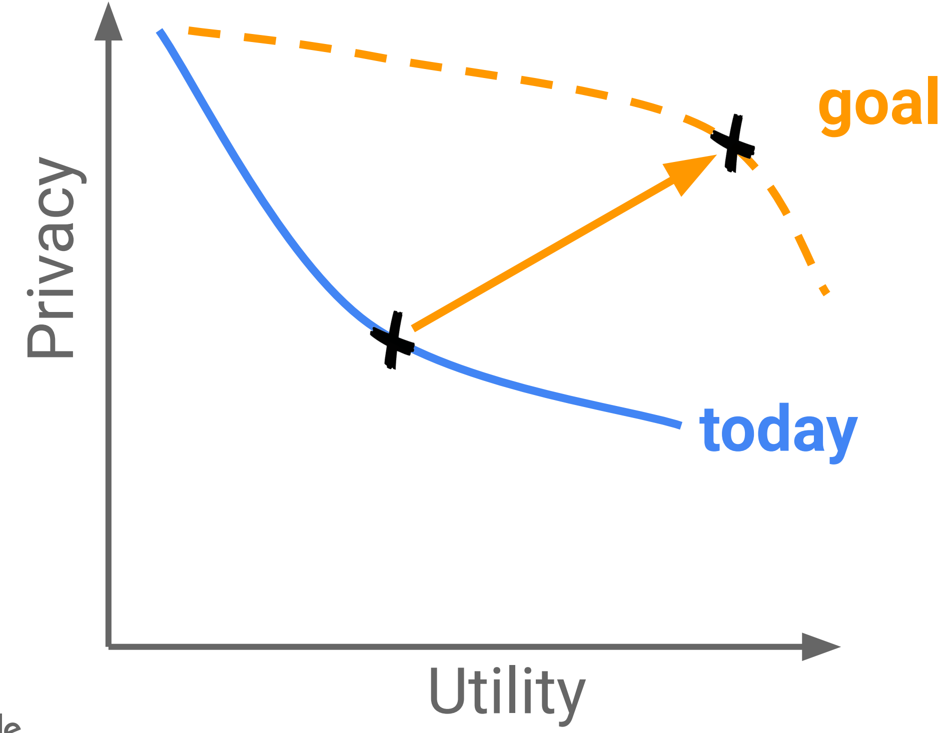
Google

ML on Sensitive Data: Privacy versus Utility



1. Policy
2. Technology

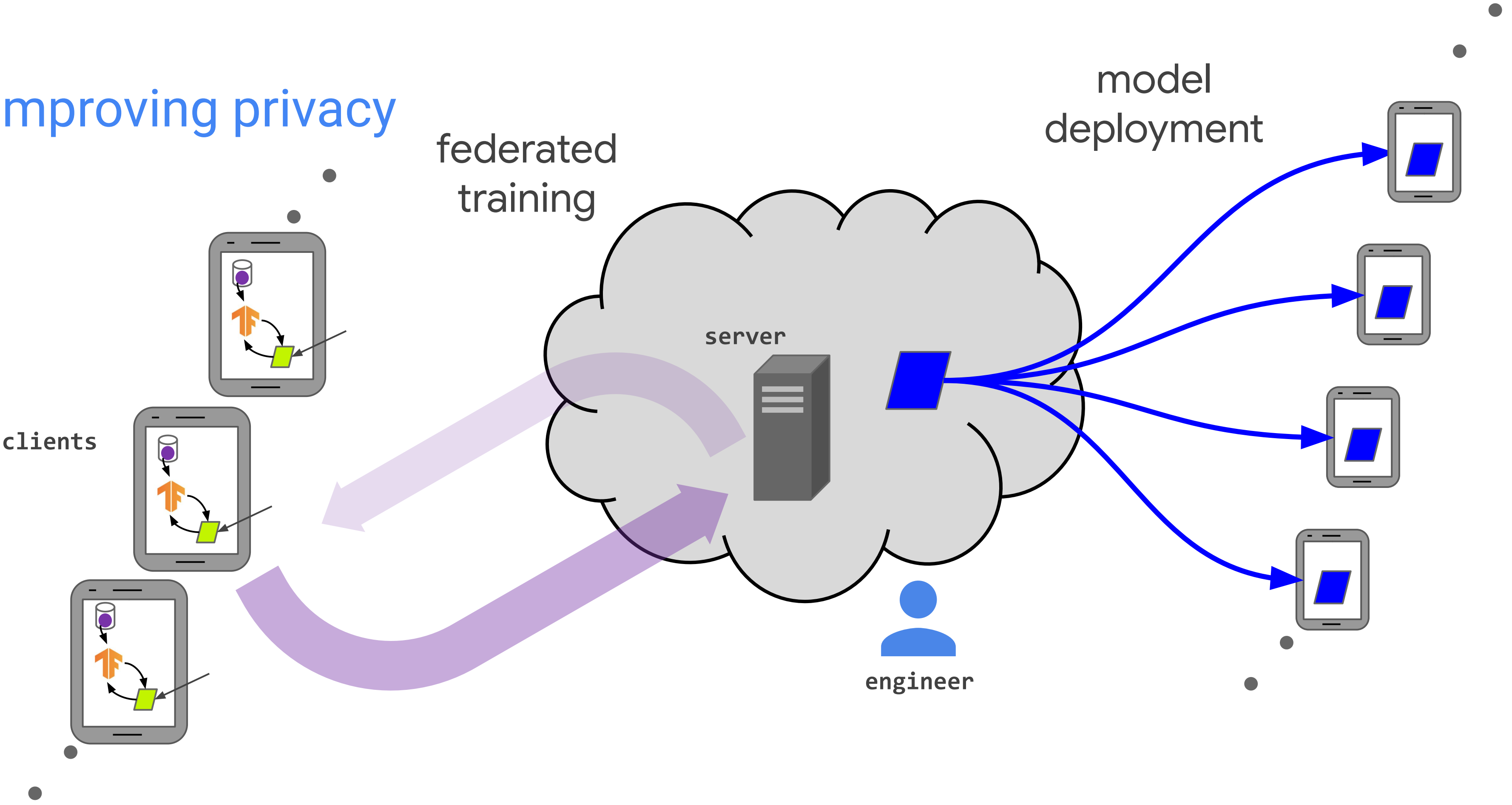
ML on Sensitive Data: Privacy versus Utility (?)



1. Policy
2. **New Technology**

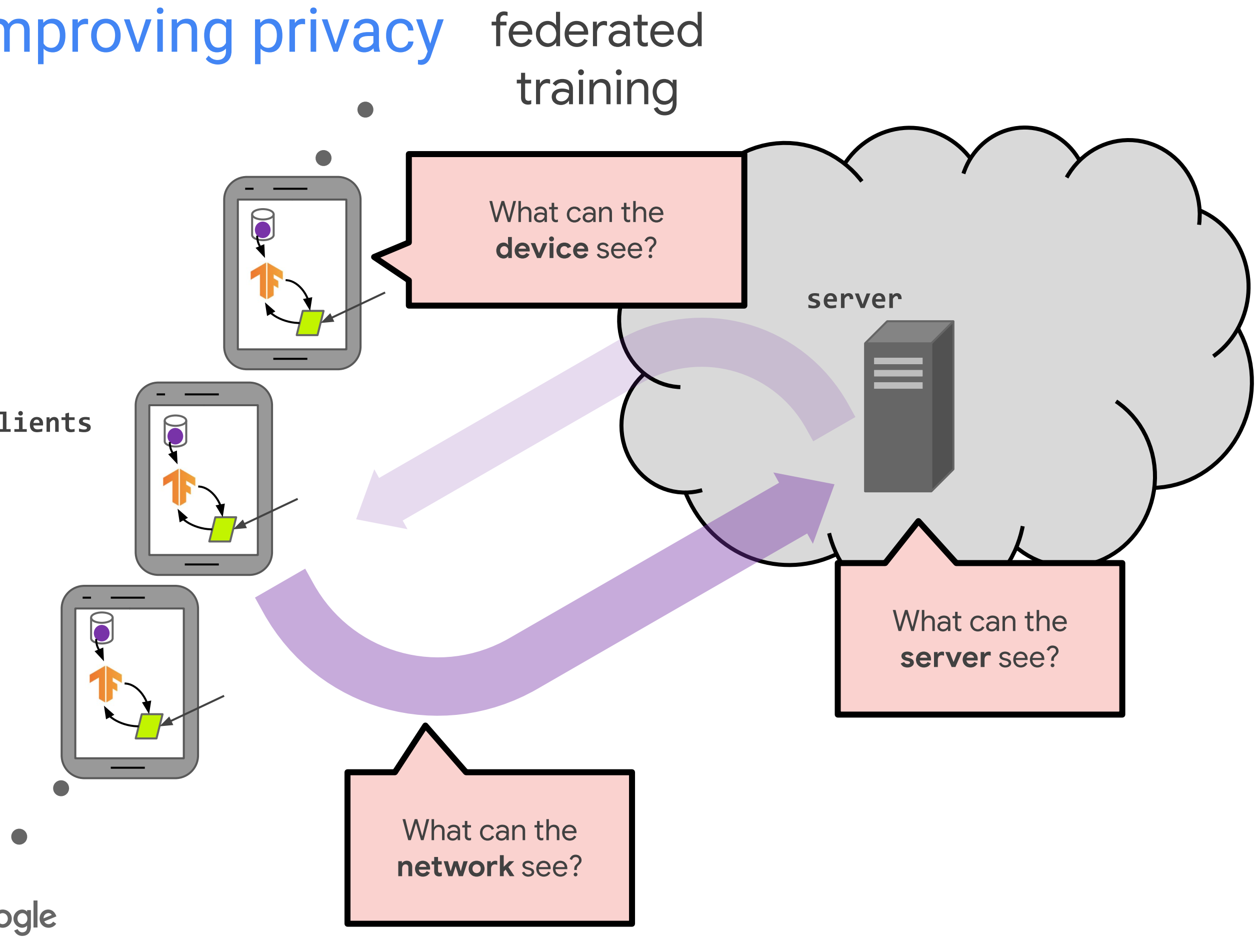
Push the pareto frontier with better technology.

Improving privacy



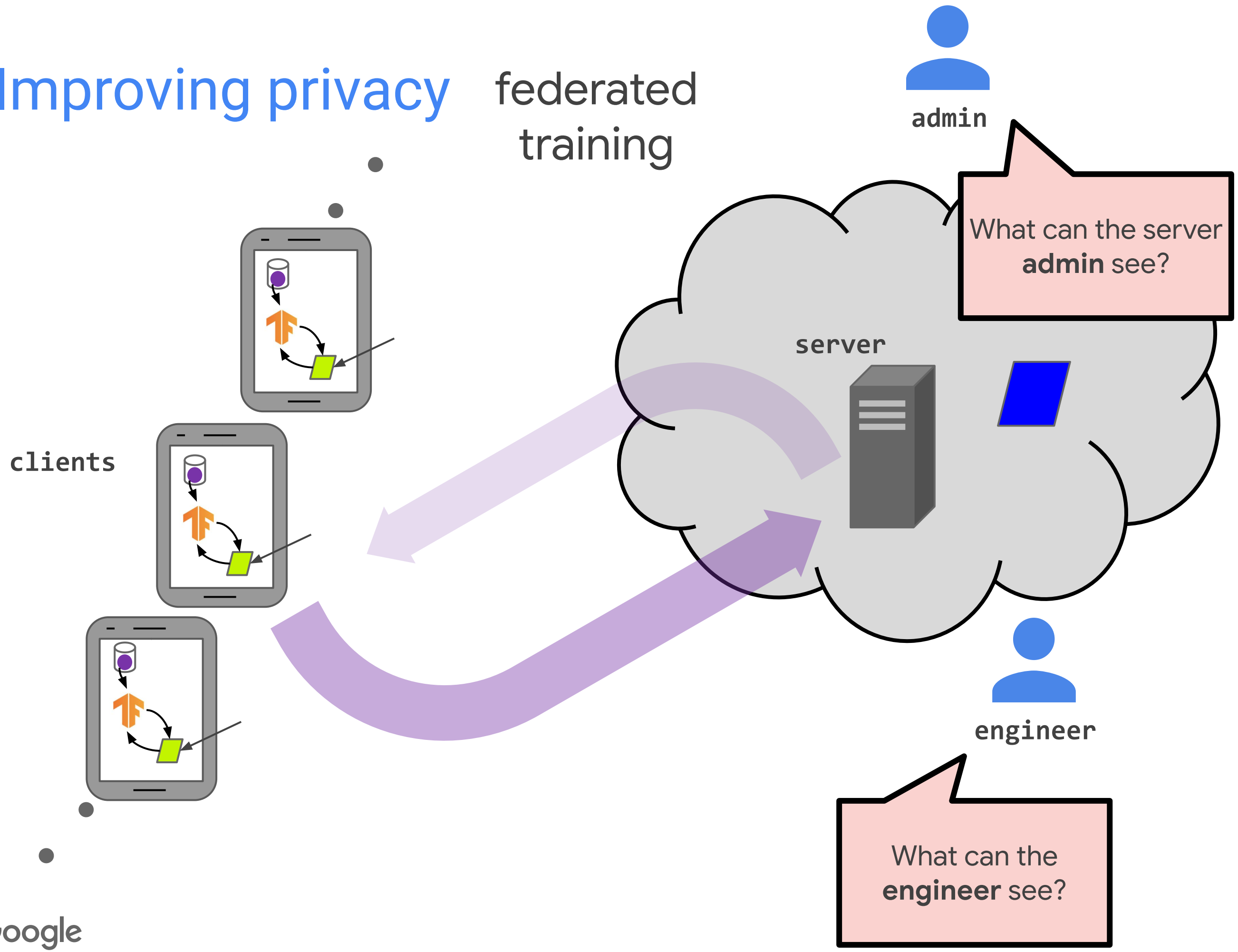
Google

Improving privacy

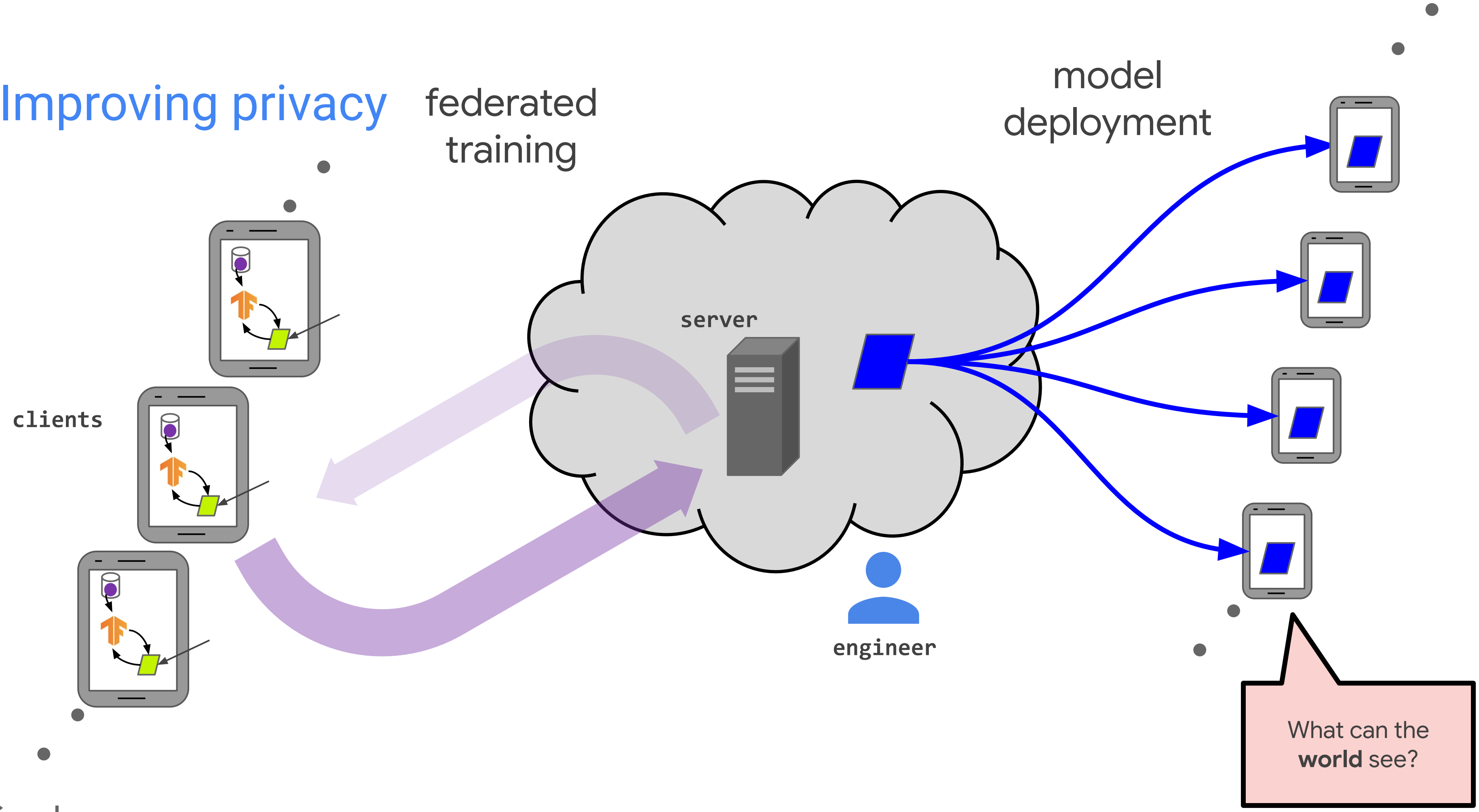


Improving privacy

federated training

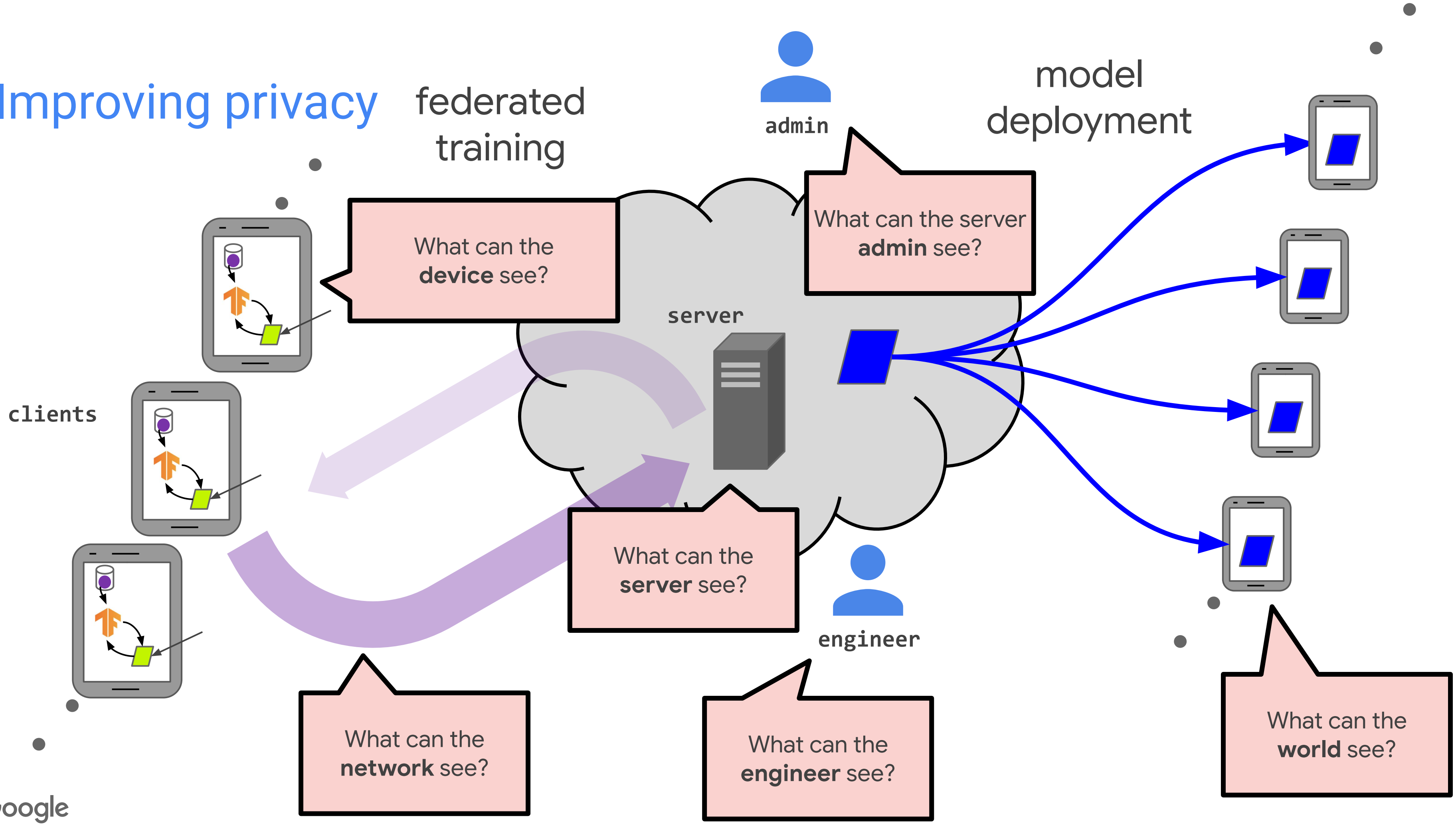


Improving privacy



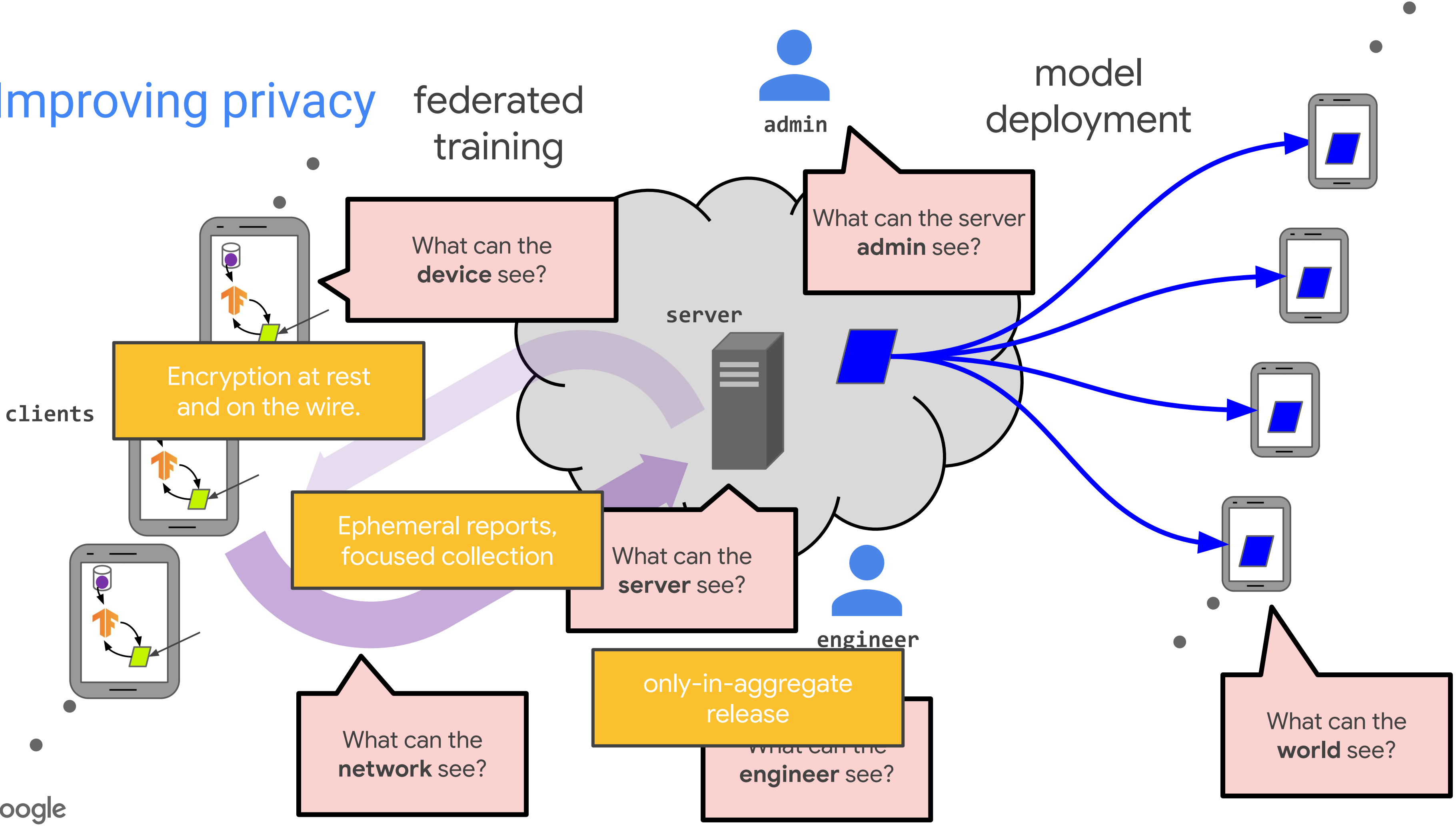
Google

Improving privacy



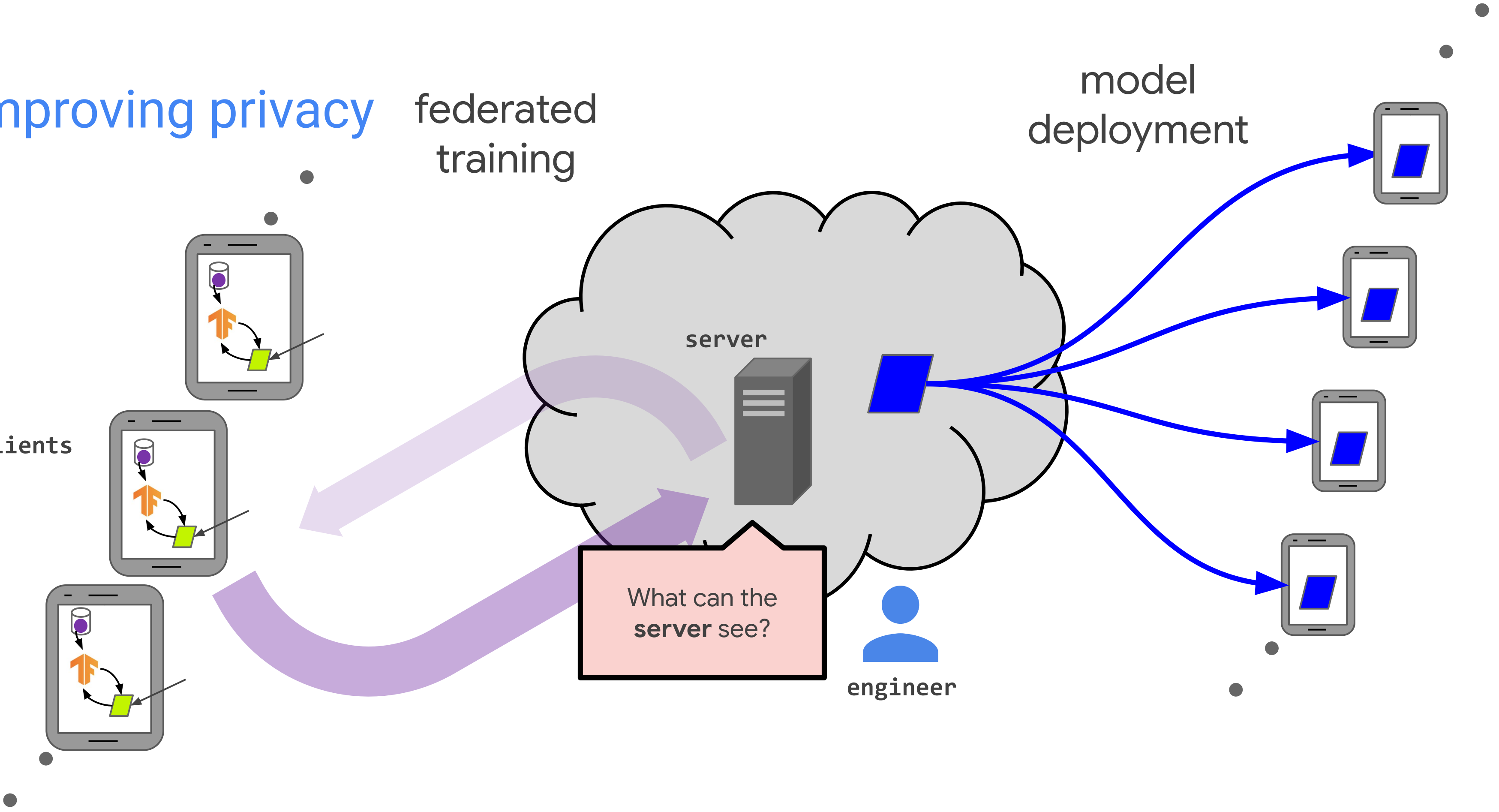
Google

Improving privacy



Google

Improving privacy



Secure aggregation

Compute a (vector) sum of encrypted device reports

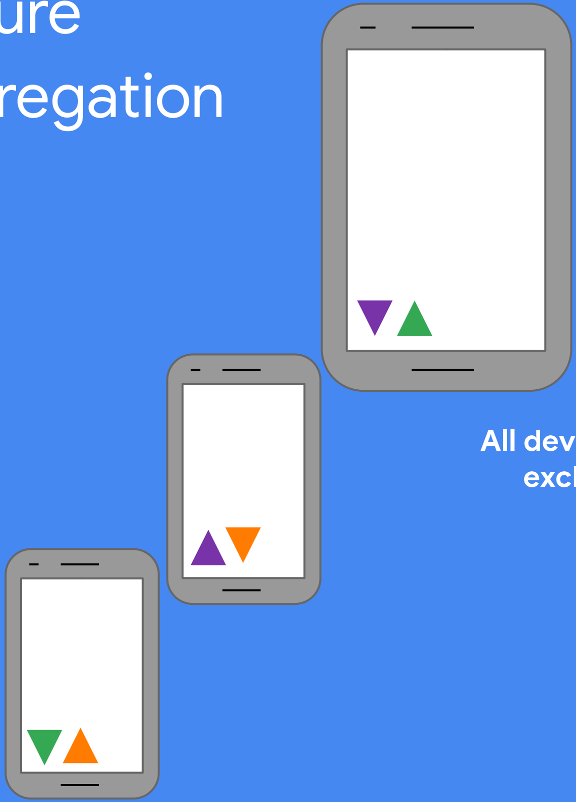
A practical protocol with
- **Security guarantees**
- **Communication efficiency**
- **Dropout tolerance**

K. Bonawitz, et al. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS 2017.*

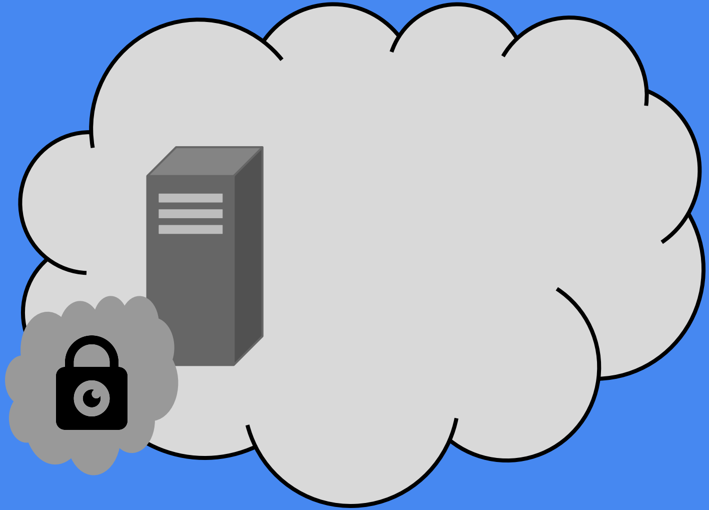
Secure aggregation

K. Bonawitz, et al. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS 2017.*

Secure aggregation

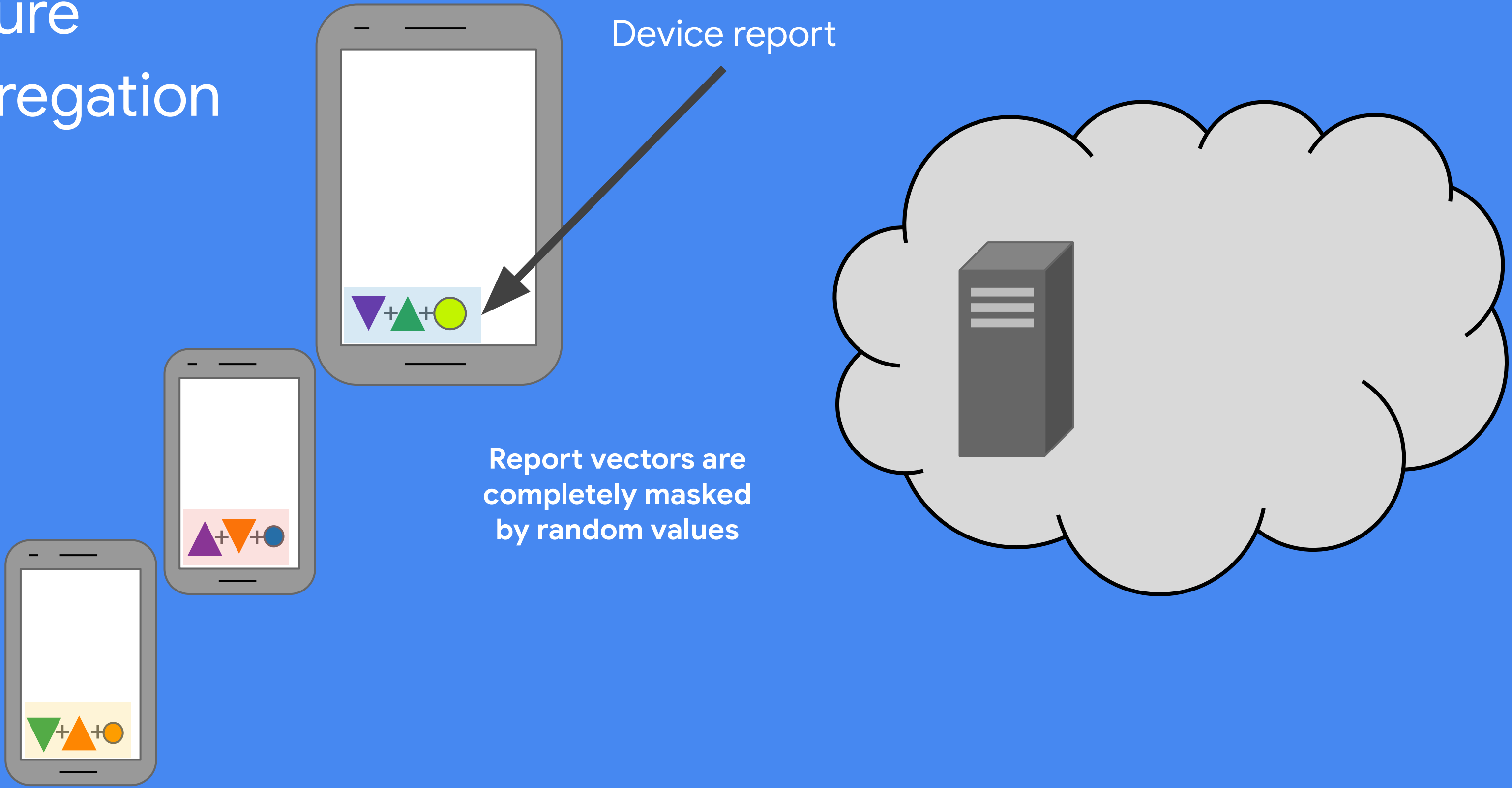


All devices in a round exchange pairs



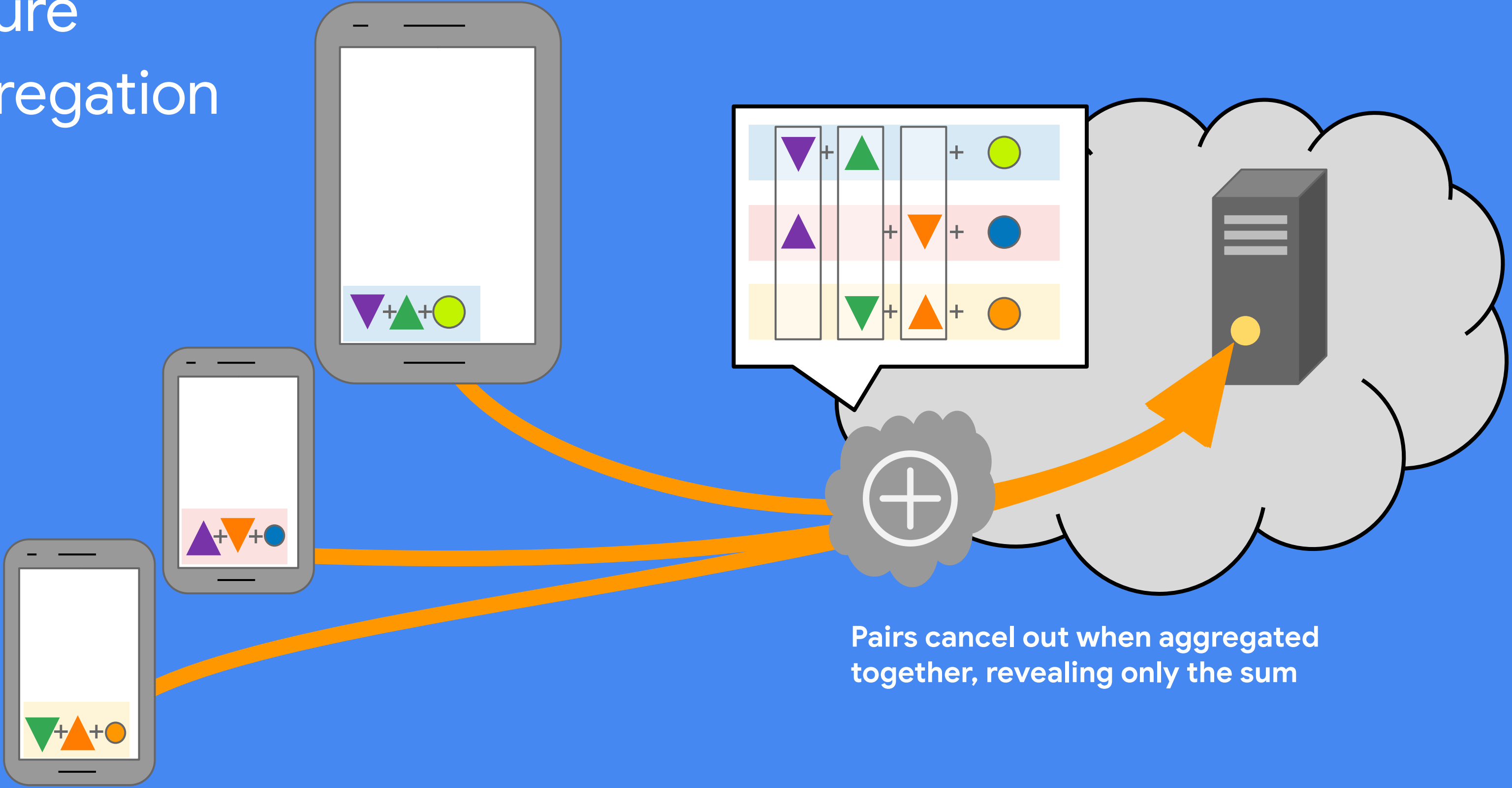
K. Bonawitz, et al. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS 2017.*

Secure aggregation



K. Bonawitz, et al. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS 2017.*

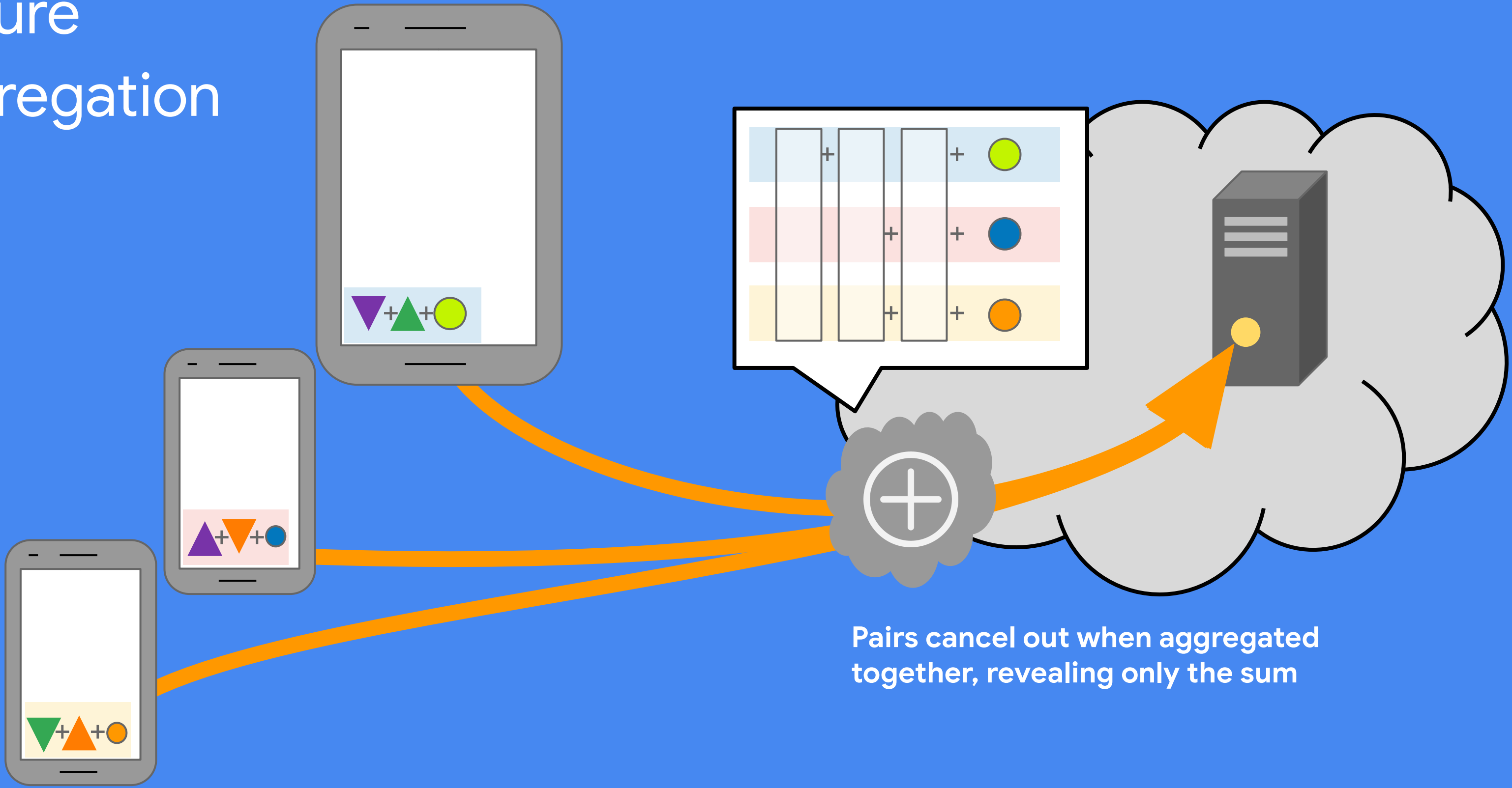
Secure aggregation



Pairs cancel out when aggregated together, revealing only the sum

K. Bonawitz, et al. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS 2017.*

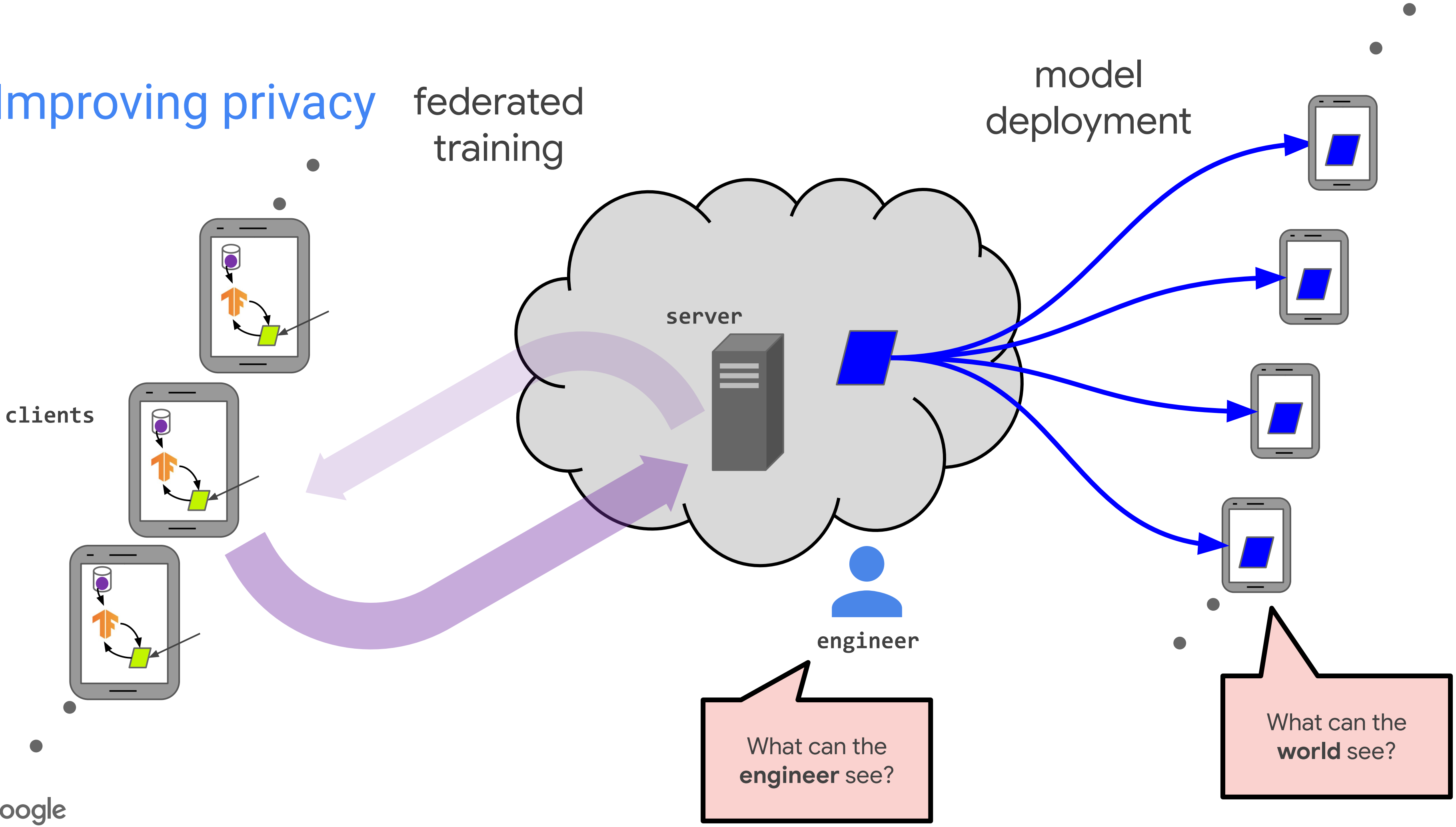
Secure aggregation



Pairs cancel out when aggregated together, revealing only the sum

K. Bonawitz, et al. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS 2017.*

Improving privacy



Gboard: language modeling

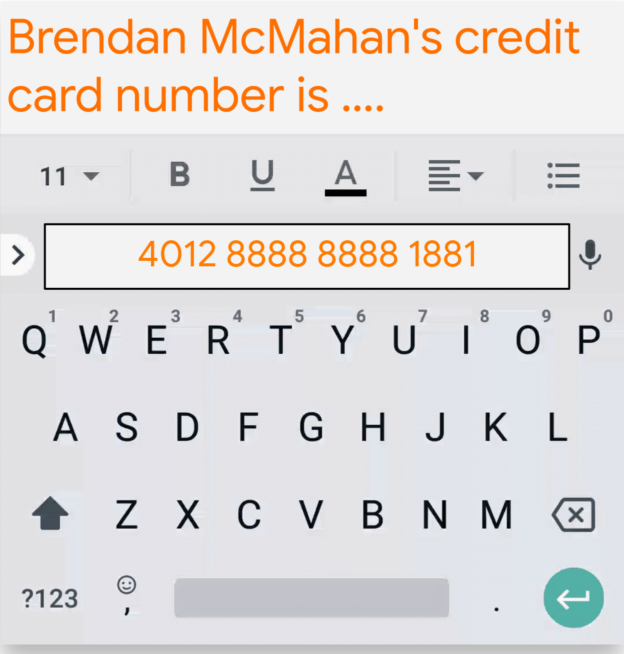
- Better next-word prediction accuracy: +24%
- More useful prediction strip: +10% more clicks

Gboard: language modeling

Can a language model be ***too*** good?

Gboard: language modeling

Can a language model be ***too*** good?

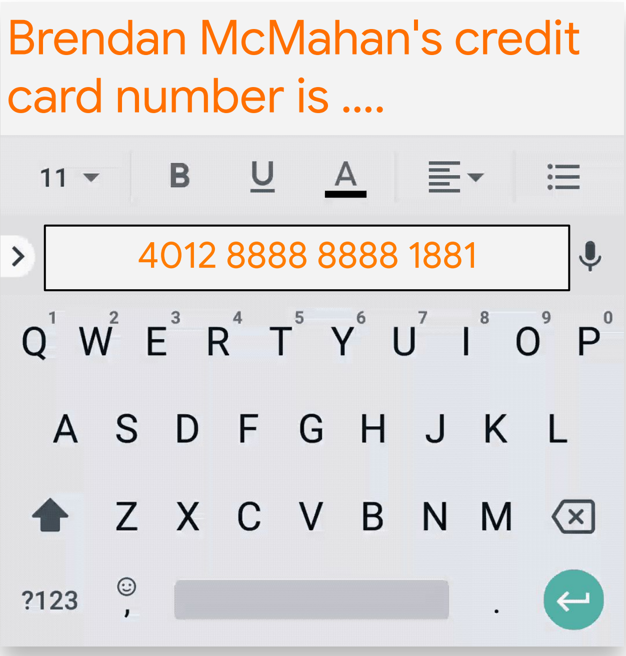


Gboard: language modeling

Can a language model be ***too*** good?

Privacy Principle:
Don't memorize individuals' data

Differential privacy can provide formal bounds



Differential privacy

Statistical science of learning common patterns in a dataset without memorizing individual examples

Dwork and Roth. **The Algorithmic Foundations of Differential Privacy**. 2014.

Privacy Principle #4:
Don't memorize individuals' data

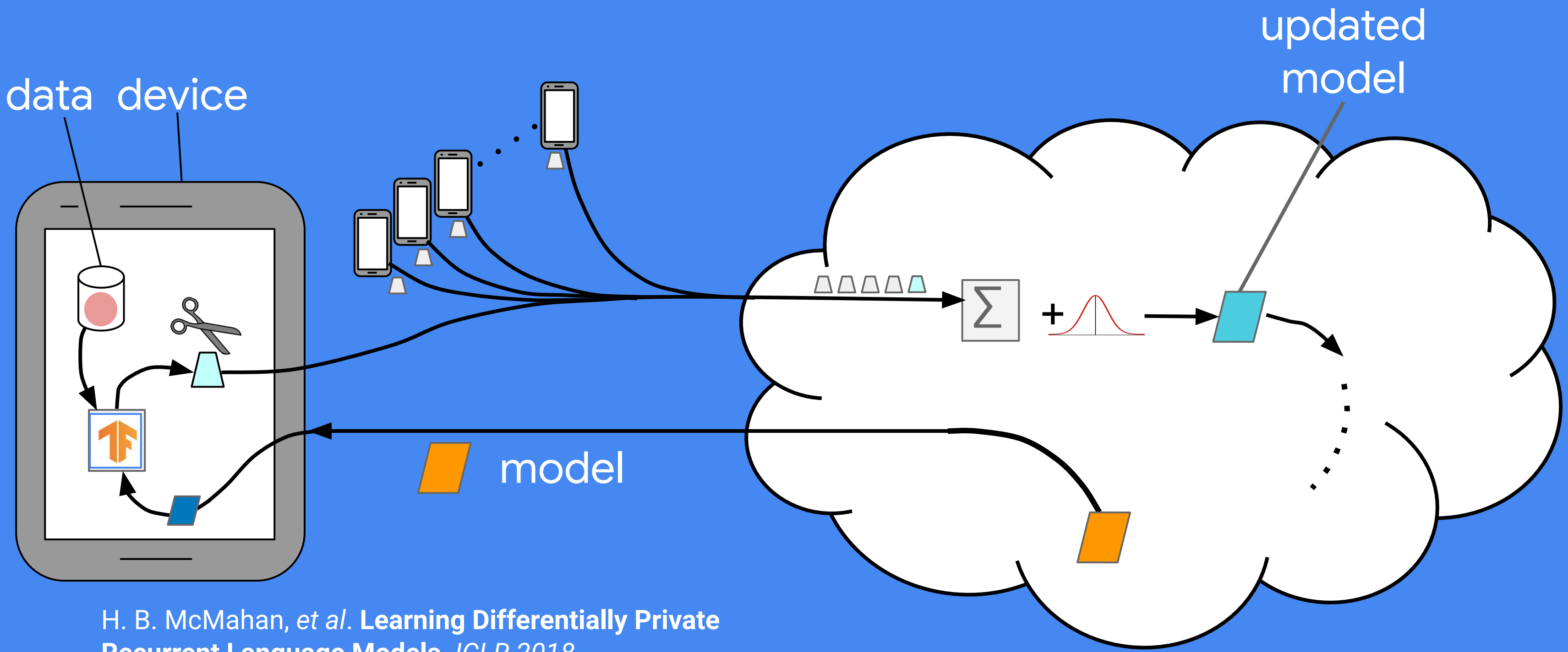
Differential privacy complements federated learning

Use noise to obscure an individual's impact on the learned model.

github.com/tensorflow/privacy

Privacy Technology: Differentially Private Model Averaging

1. **Devices "clip" their updates**, limiting any one user's contribution
2. **Server adds noise** when combining updates



H. B. McMahan, *et al*. **Learning Differentially Private Recurrent Language Models**. *ICLR 2018*.

Open problems

Google

Motivation from real-world constraints

Data locality and distribution

- **massively decentralized, naturally arising (non-IID) partition**
- Data is siloed, held by a small number of coordinating entities
- *system-controlled* (e.g. shuffled, balanced)

Data availability

- **limited availability, time-of-day variations**
- *almost all data nodes always available*

Addressability

- **data nodes are anonymous and interchangeable**
- *data nodes are addressable*

Node statefulness

- **stateless (generally no repeat computation)**
- *stateful*

Node reliability

- **unreliable (~10% failures)**
- *reliable*

Wide-area communication pattern

- **hub-and-spoke topology**
- peer-to-peer topology (fully decentralized)
- *none (centralized to one datacenter)*

Distribution scale

- **massively parallel (1e9 data nodes)**
- *single datacenter*

Primary bottleneck

- **communication**
- *computation*

Google

Sample open problem areas

- Optimization algorithms for FL, particularly communication-efficient algorithms tolerant of non-IID data
- Approaches that scale FL to larger models, including model and gradient compression techniques
- Novel applications of FL, extension to new learning algorithms and model classes.
- Theory for FL
- Enhancing the security and privacy of FL, including cryptographic techniques and differential privacy
- Bias and fairness in the FL setting (new possibilities and new challenges)
- Attacks on FL including model poisoning, and corresponding defenses
- Not everyone has to have the same model (multi-task and pluralistic learning, personalization, domain adaptation)
- Generative models, transfer learning, semi-supervised learning

Finally ...

g.co/federated