Statistical Machine Learning

Privacy – Parts 05 & 06

(Version 1.0)

Hamid R. Rabiee

Spring 2023

AAAI Workshop of Privacy Preserving Artificial Intelligence (PPAI-21)
Brendan McMahan, Kallista Bonawitz, Peter Kairouz

Tutorial Outline

Part 1:
What is Federated Learning?

Part 2:

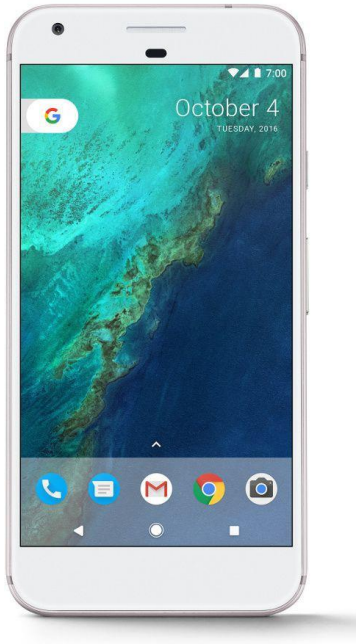
Privacy for Federated Technologies Part 2.1: Private Aggregation & Trust

Part 2.2: Differentially Private Federated Training

Part I: What is Federated Learning?

Data is born at the edge

- Billions of phones & IoT devices constantly generate data
- Data enables better products and smarter models

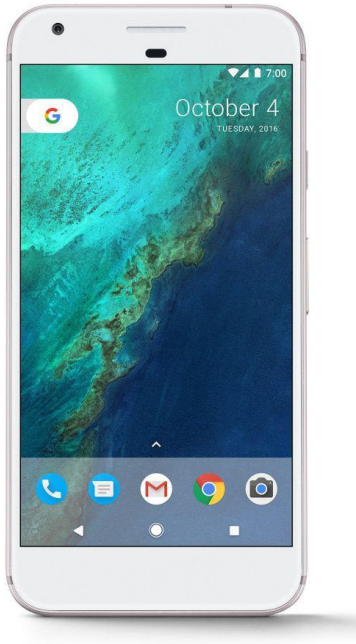


Can data live at the edge?

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?

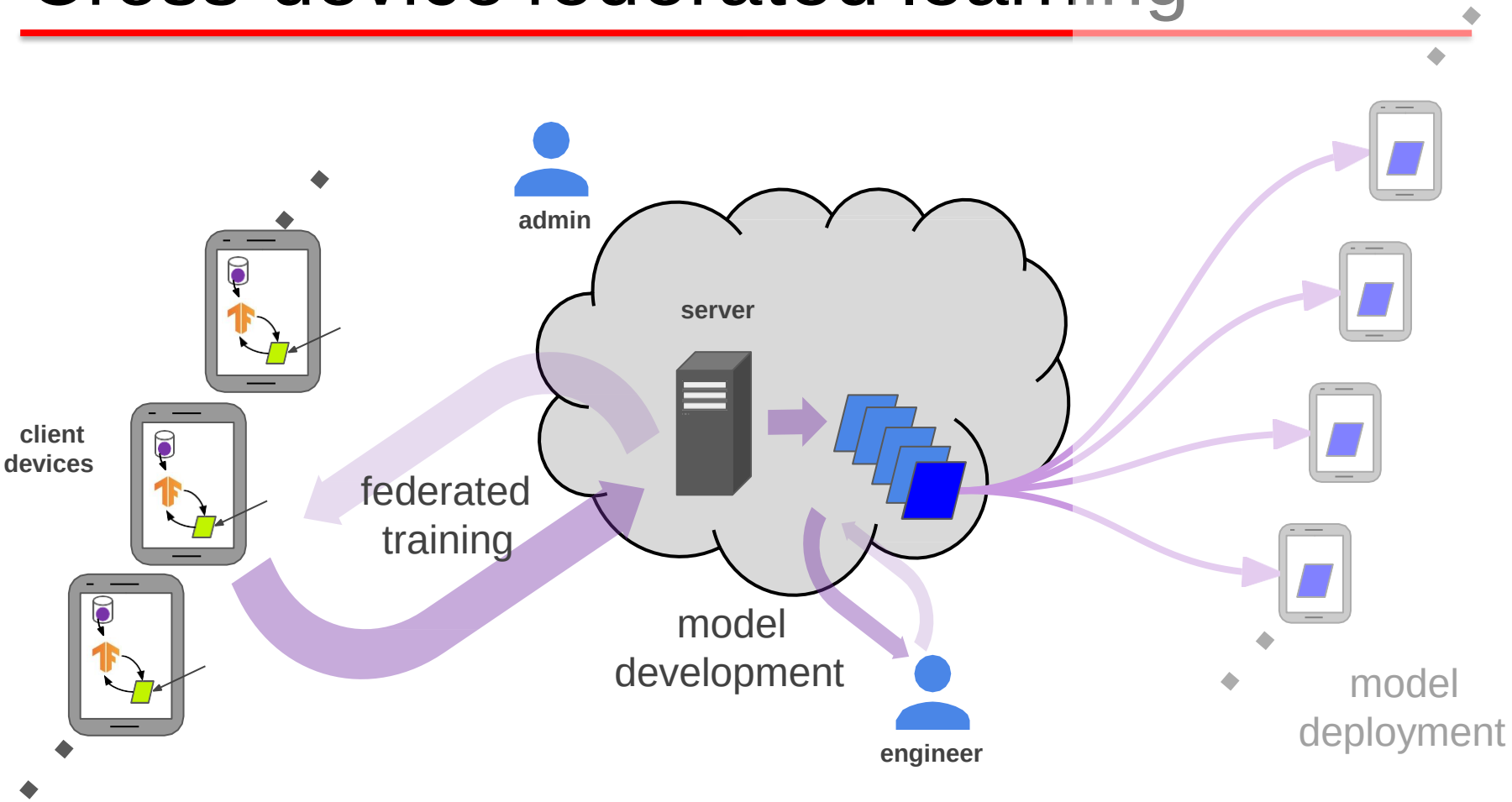
Data processing is moving on device:

- 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?

Cross-device federated learning



Applications of cross-device federating learning

What makes a good application?

- 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

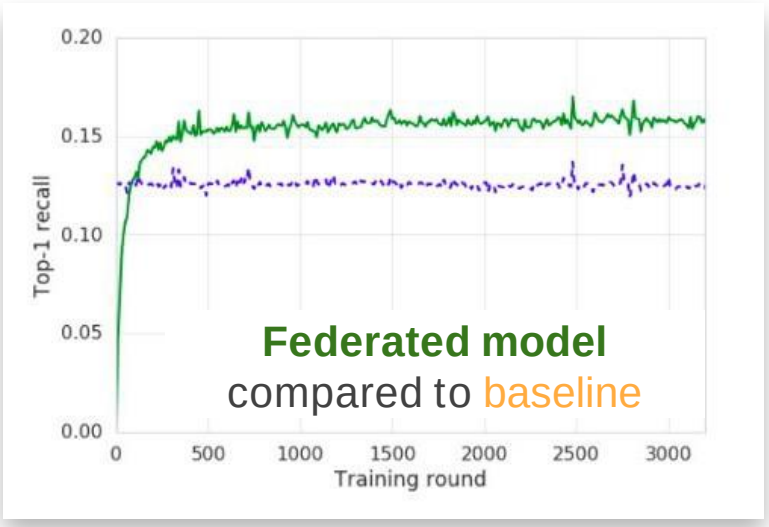
Example applications

- Language modeling for mobile keyboards and voice recognition
- Image classification for predicting which photos people will share
- ...

Gboard: next-word prediction

Federated RNN:

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



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

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*

Cross-device federated learning at Apple

MIT Technology Review

Sign in Subscribe

Artificial intelligence / Machine learning

How Apple personalizes Siri without hoovering up your data

The tech giant is using privacy-preserving machine learning to improve its voice assistant while keeping your data on your phone.

by **Karen Hao** December 11, 2019

*"Instead, **it relies primarily on a technique called federated learning**, Apple's head of privacy, Julien Freudiger, told an audience at the Neural Processing Information Systems conference on December 8. Federated learning is a privacy-preserving machine-learning method that was first introduced by Google in 2017. It allows Apple to train different copies of a speaker recognition model across all its users' devices, using only the audio data available locally. It then sends just the updated models back to a central server to be combined into a master model. In this way, raw audio of users' Siri requests never leaves their iPhones and iPads, but the assistant continuously gets better at identifying the right speaker."*

https://www.technologyreview.com/2019/12/11/131629/apple-ai-personalizes-siri-federated-learning/

Federated Learning

Federated learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective.

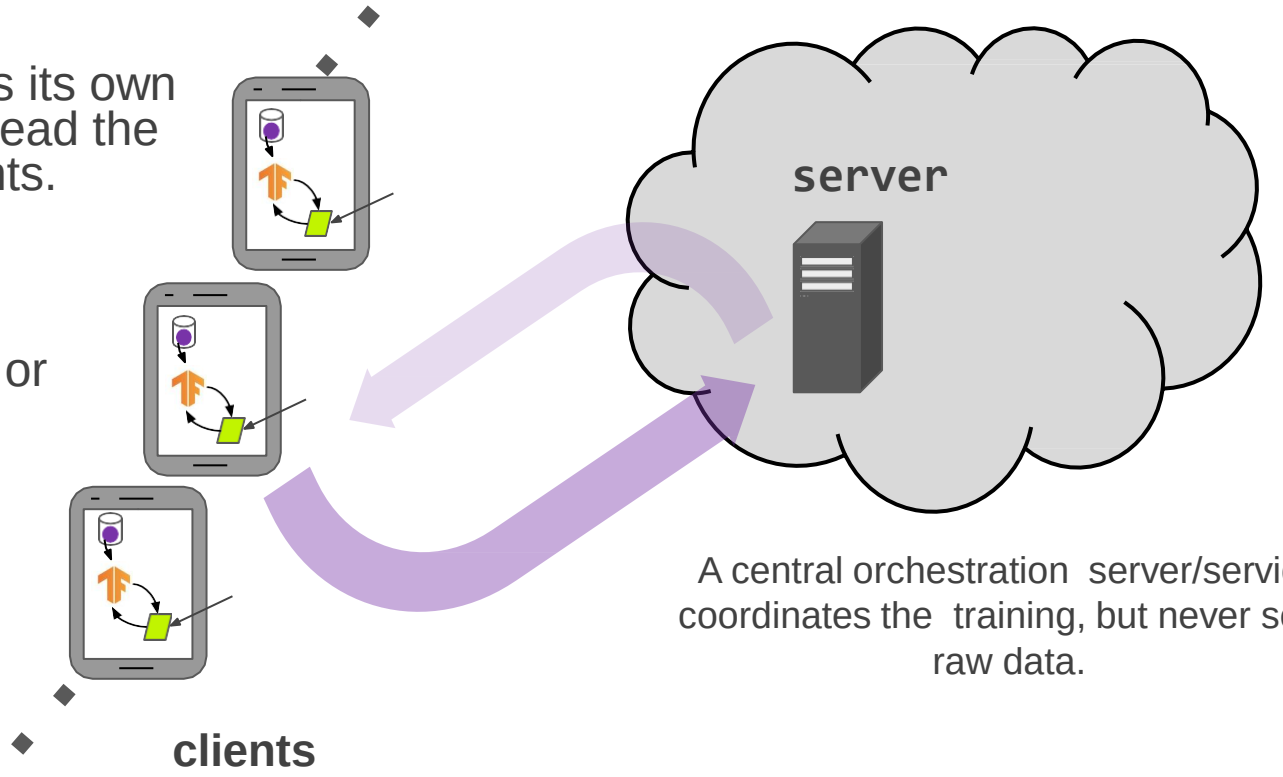
definition proposed in
Advances and Open Problems in Federated Learning (arxiv/1912.04977)

Federated learning - defining characteristics

Data is generated locally and remains decentralized.

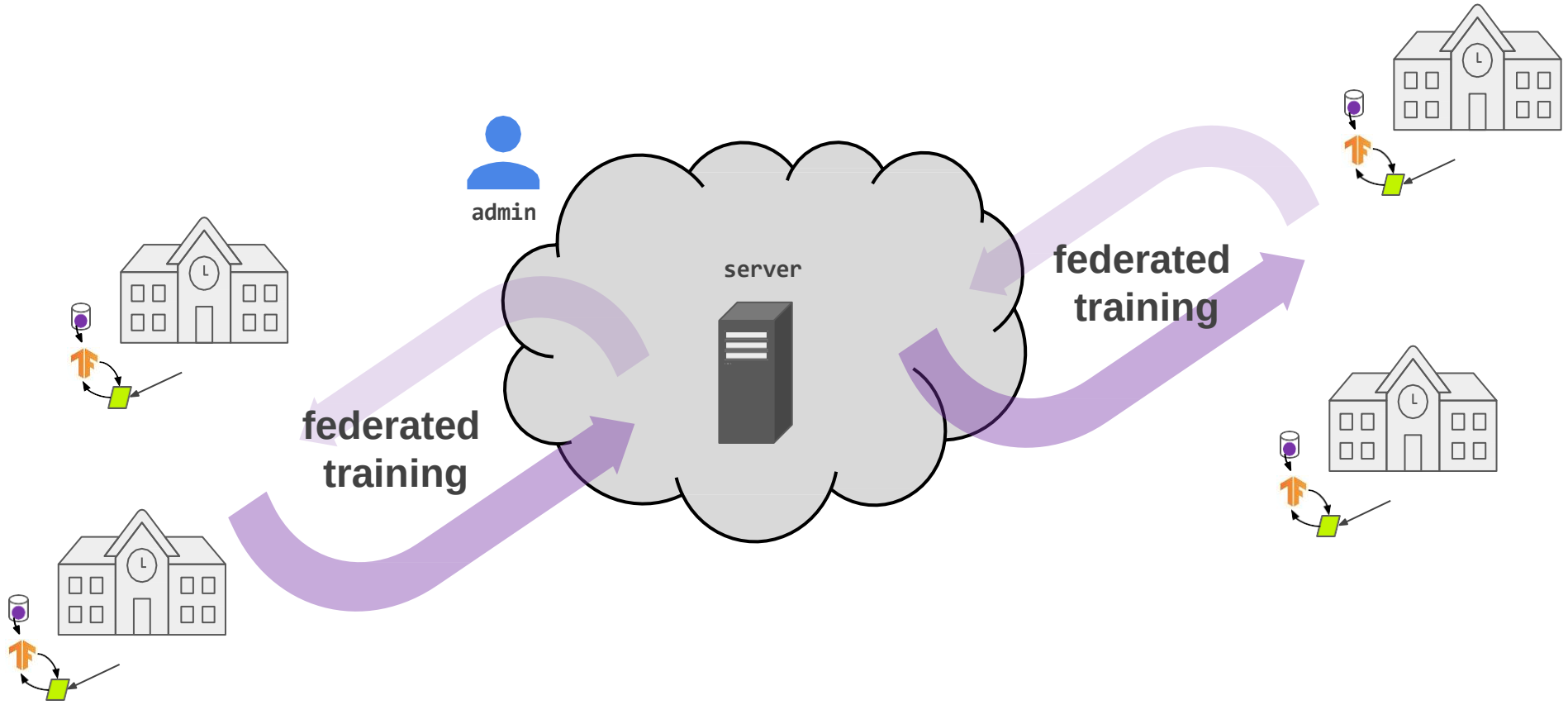
Each client stores its own data and cannot read the data of other clients.

Data is not independently or identically distributed.



A central orchestration server/service coordinates the training, but never sees raw data.

Cross-silo federated learning



Cross-silo federated learning from Intel

ARTIFICIAL INTELLIGENCE, DIAGNOSTICS

UPenn, Intel partner to use federated learning AI for early brain tumor detection

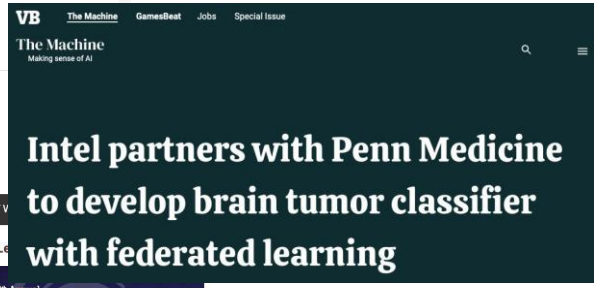
The project will bring in 29 institutions from North America, Europe and India and will use privacy-preserved data to train AI models. Federated learning has been described as being born at the intersection of AI, blockchain, edge computing and the Internet of Things.

By ALARIC DEARMENT

Post a comment / May 11, 2020 at 10:03 AM

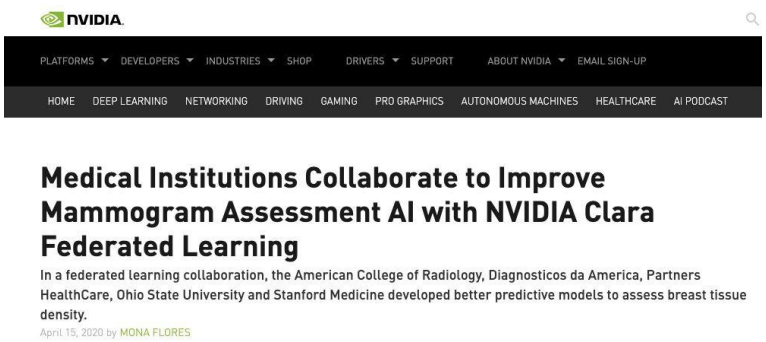
"The University of Pennsylvania and chipmaker Intel are forming a partnership to enable 29 heatlhcare and medical research institutions around the world to train artificial intelligence models to detect brain tumors early."

"The program will rely on a technique known as federated learning, which enables institutions to collaborate on deep learning projects without sharing patient data. The partnership will bring in institutions in the U.S., Canada, U.K., Germany, Switzerland and India. The centers – which include Washington University of St. Louis; Queen's University in Kingston, Ontario; University of Munich; Tata Memorial Hospital in Mumbai and others – will use Intel's federated learning hardware and software."

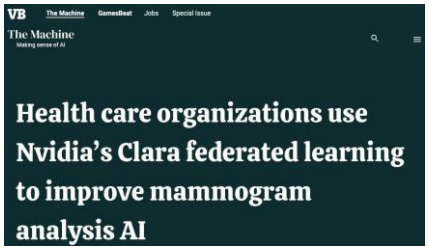


1 https://medcitynews.com/2020/05/upenn-intel-partner-to-use-federated-learning-ai-for-early-brain-tumor-detection/
2 https://www.allaboutcircuits.com/news/can-machine-learning-keep-patient-privacy-for-tumor-research-intel-says-yes-with-federated-learning/
3 https://venturebeat.com/2020/05/11/intel-partners-with-penn-medicine-to-develop-brain-tumor-classifier-with-federated-learning/
4 http://www.bio-itworld.com/2020/05/28/intel-penn-medicine-launch-federated-learning-model-for-brain-tumors.aspx

Cross-silo federated learning from NVIDIA



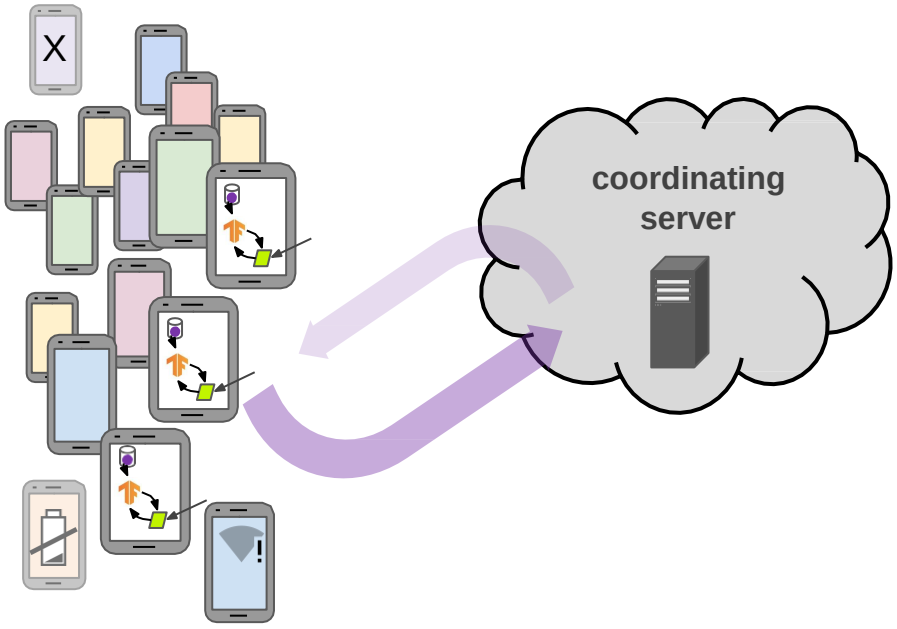
"Federated learning addresses this challenge, enabling different institutions to collaborate on AI model development without sharing sensitive clinical data with each other. The goal is to end up with more generalizable models that perform well on any dataset, instead of an AI biased by the patient demographics or imaging equipment of one specific radiology department."



1 https://blogs.nvidia.com/blog/2020/04/15/federated-learning-mammogram-assessment/
2 https://venturebeat.com/2020/04/15/healthcare-organizations-use-nvidias-clara-federated-learning-to-improve-mammogram-analysis-ai/
3 https://medcitynews.com/2020/01/nvidia-says-it-has-a-solution-for-healthcares-data-problems/
4 https://venturebeat.com/2020/06/23/nvidia-and-mercedes-benz-detail-self-driving-system-with-automated-routing-and-parking/

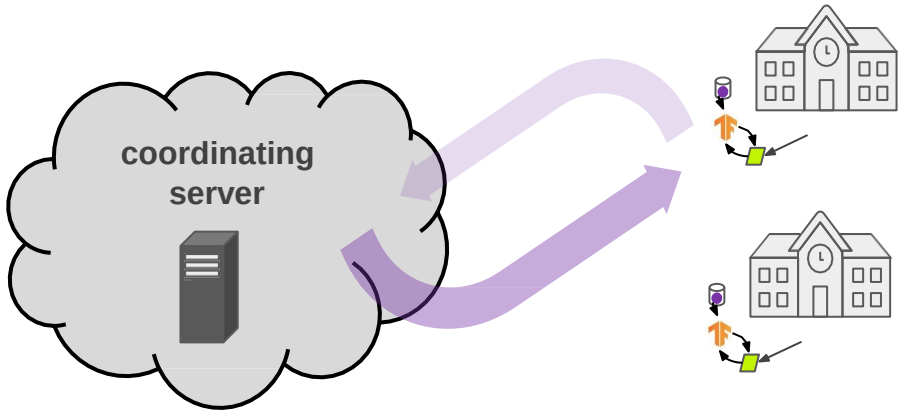
Cross-device federated learning

millions of intermittently available client devices



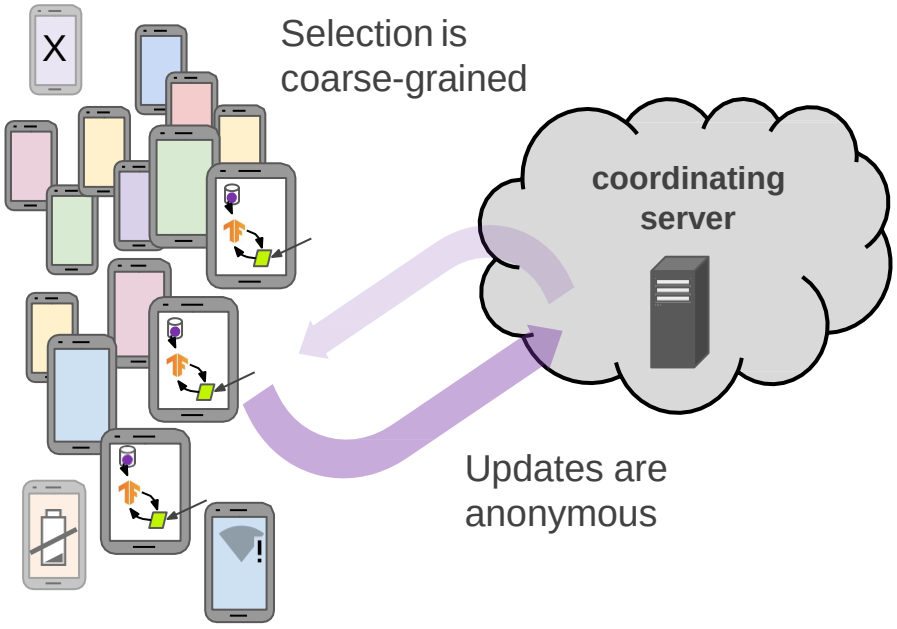
Cross-silo federated learning

small number of clients (institutions, data silos), high availability



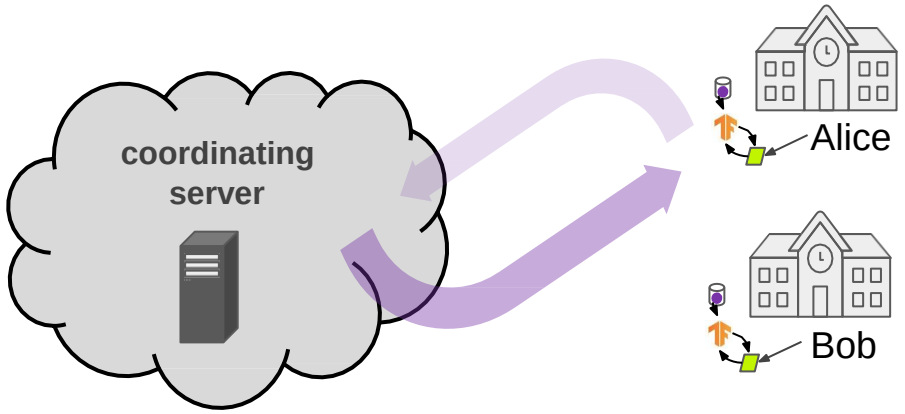
Cross-device federated learning

clients cannot be indexed directly (i.e., no use of client identifiers)



Cross-silo federated learning

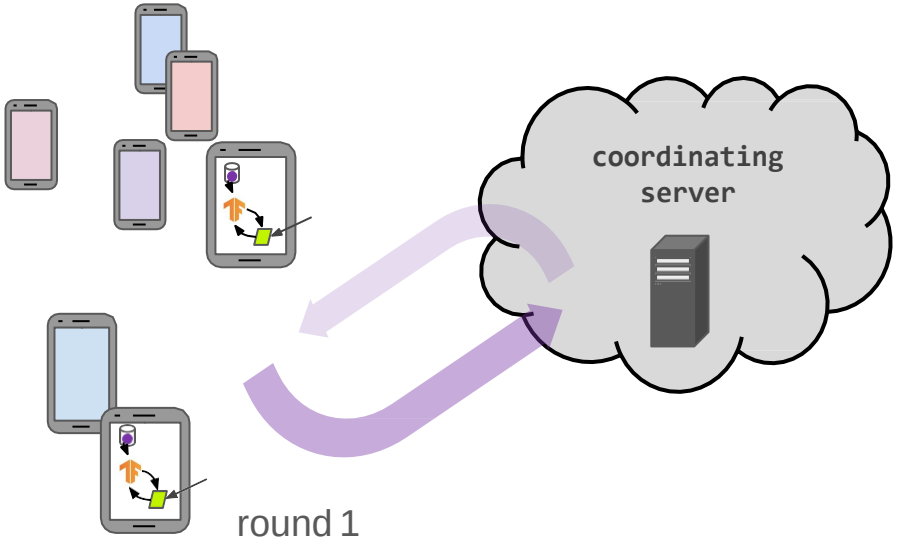
each client has an identity or name that allows the system to access it specifically



Cross-device federated learning

Server can only access a (possibly biased) random sample of clients on each round.

Large population => most clients only participate once.

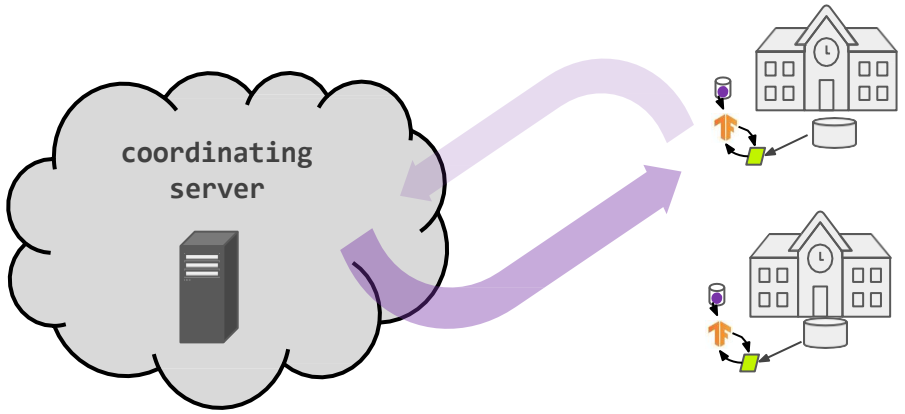


round 1

Cross-silo federated learning

Most clients participate in every round.

Clients can run algorithms that maintain local state across rounds.

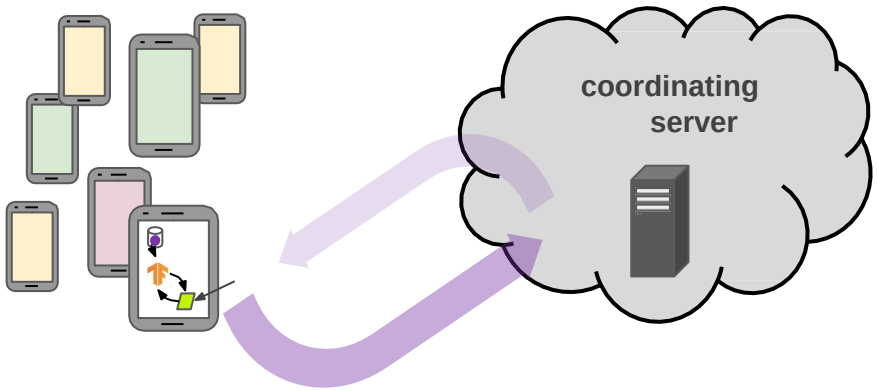


round 1

Cross-device federated learning

Server can only access a (possibly biased) random sample of clients on each round.

Large population => most clients only participate once.

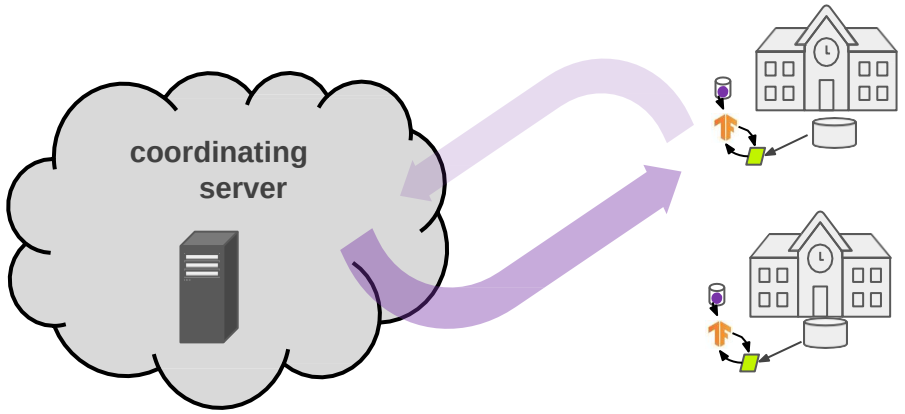


round 2
(completely new set of devices participate)

Cross-silo federated learning

Most clients participate in every round.

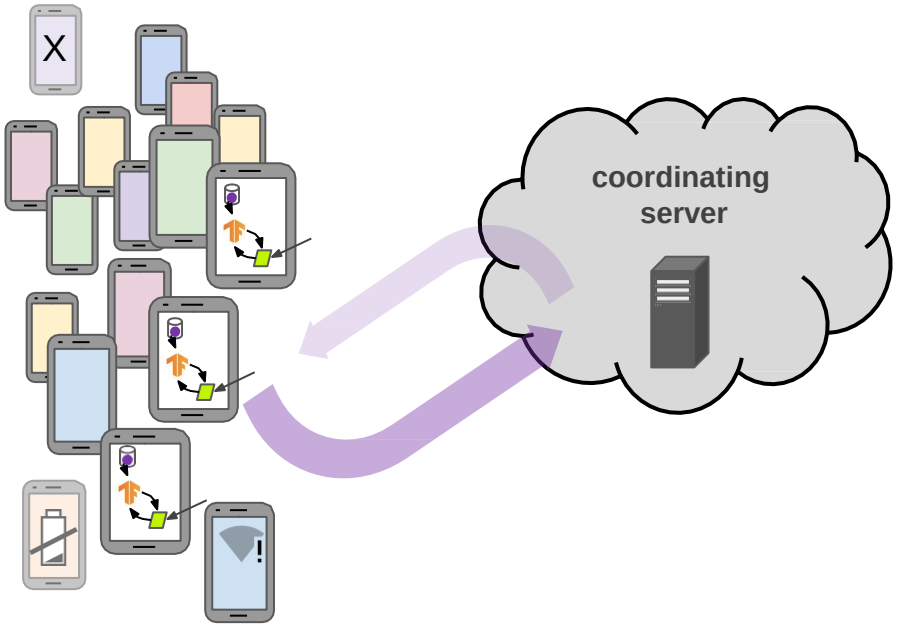
Clients can run algorithms that maintain local state across rounds.



round 2
(same clients)

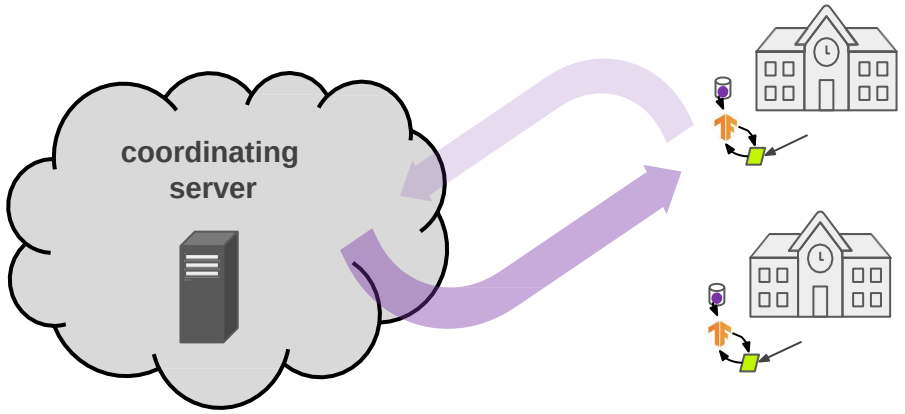
Cross-device federated learning

communication is often the primary bottleneck



Cross-silo federated learning

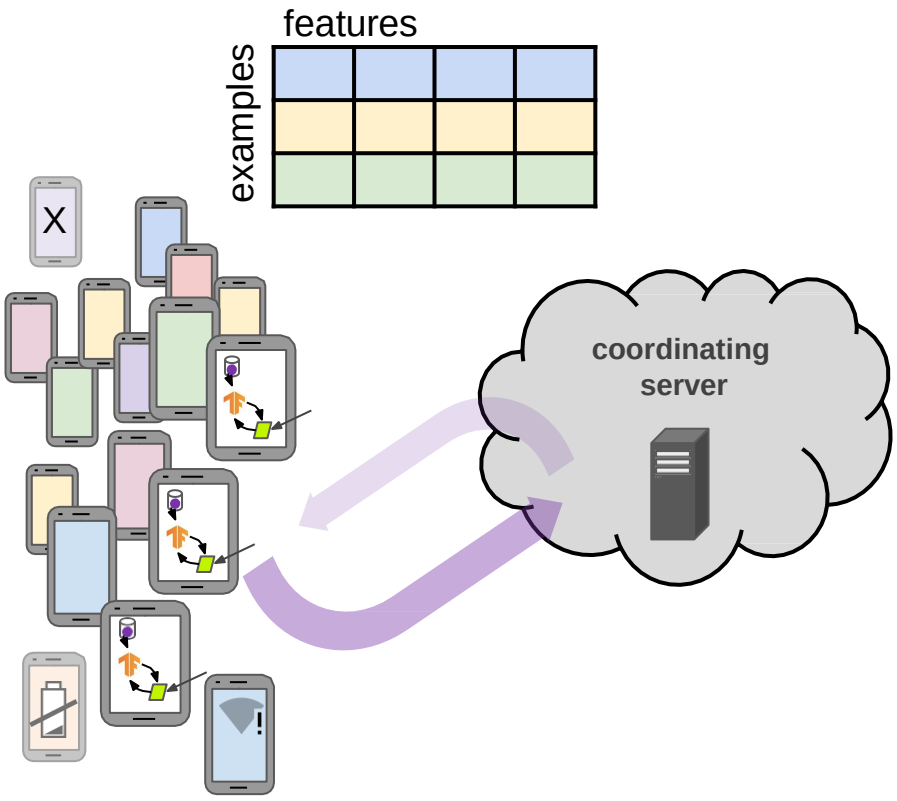
communication or computation might be the primary bottleneck

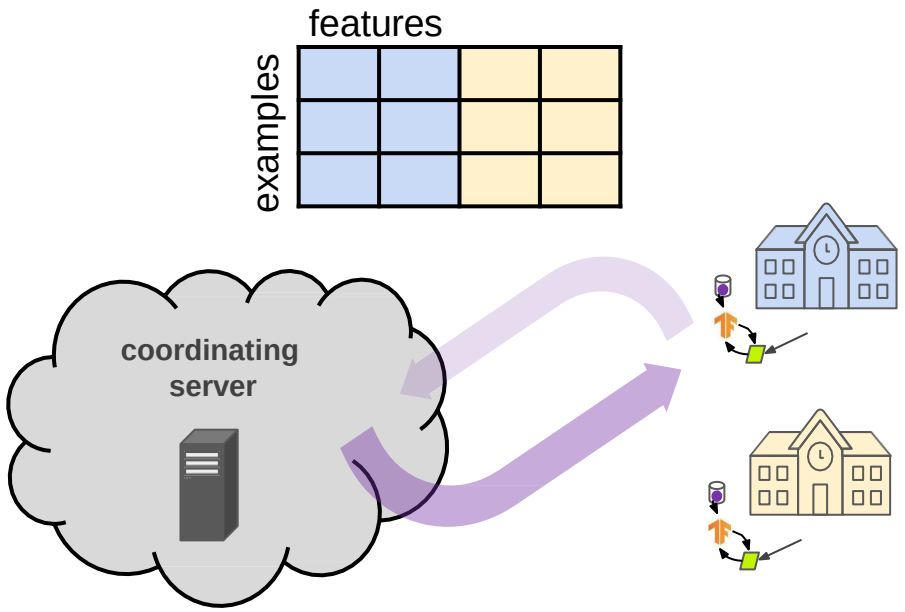


Cross-device federated learning

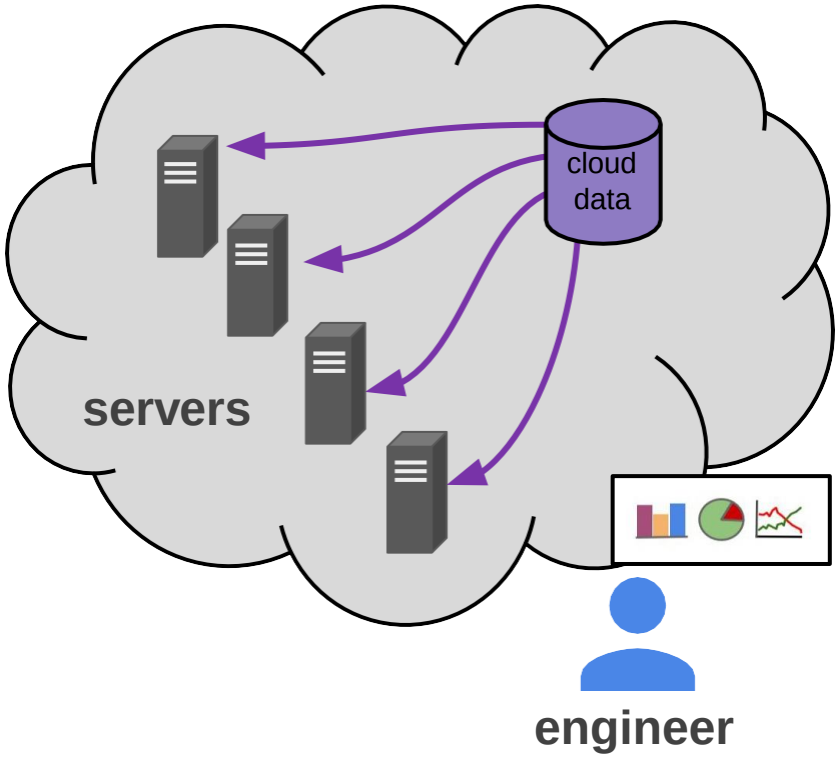
Cross-silo federated learning

horizontally partitioned data





Distributed datacenter machine learning



FL terminology

- **Clients** - Compute nodes also holding local data, usually belonging to one entity:
  - IoT devices
  - Mobile devices
  - Data silos
  - Data centers in different geographic regions

- **Server** - Additional compute nodes that coordinate the FL process but don't access raw data. Usually not a single physical machine.

Characteristics of the federated learning setting

	Datacenter distributed learning	Cross-silo federated learning	Cross-device federated learning
Setting	Training a model on a large but "flat" dataset. Clients are compute nodes in a single cluster or datacenter.	Training a model on siloed data. Clients are different organizations (e.g., medical or financial) or datacenters in different geographical regions.	The clients are a very large number of mobile or IoT devices.
Data distribution	Data is centrally stored, so it can be shuffled and balanced across clients. Any client can read any part of the dataset.	**Data is generated locally and remains decentralized.** Each client stores its own data and cannot read the data of other clients. Data is not independently or identically distributed.	
Orchestration	Centrally orchestrated.	A central orchestration server/service organizes the training, but never sees raw data.	
Wide-area communication	None (fully connected clients in one datacenter/cluster).	Typically hub-and-spoke topology, with the hub representing a coordinating service provider (typically without data) and the spokes connecting to clients.	
Data availability	All clients are almost always available.		Only a fraction of clients are available at any one time, often with diurnal and other variations.
Distribution scale	Typically 1 - 1000 clients.	Typically 2 - 100 clients.	Massively parallel, up to 10^{10} clients.

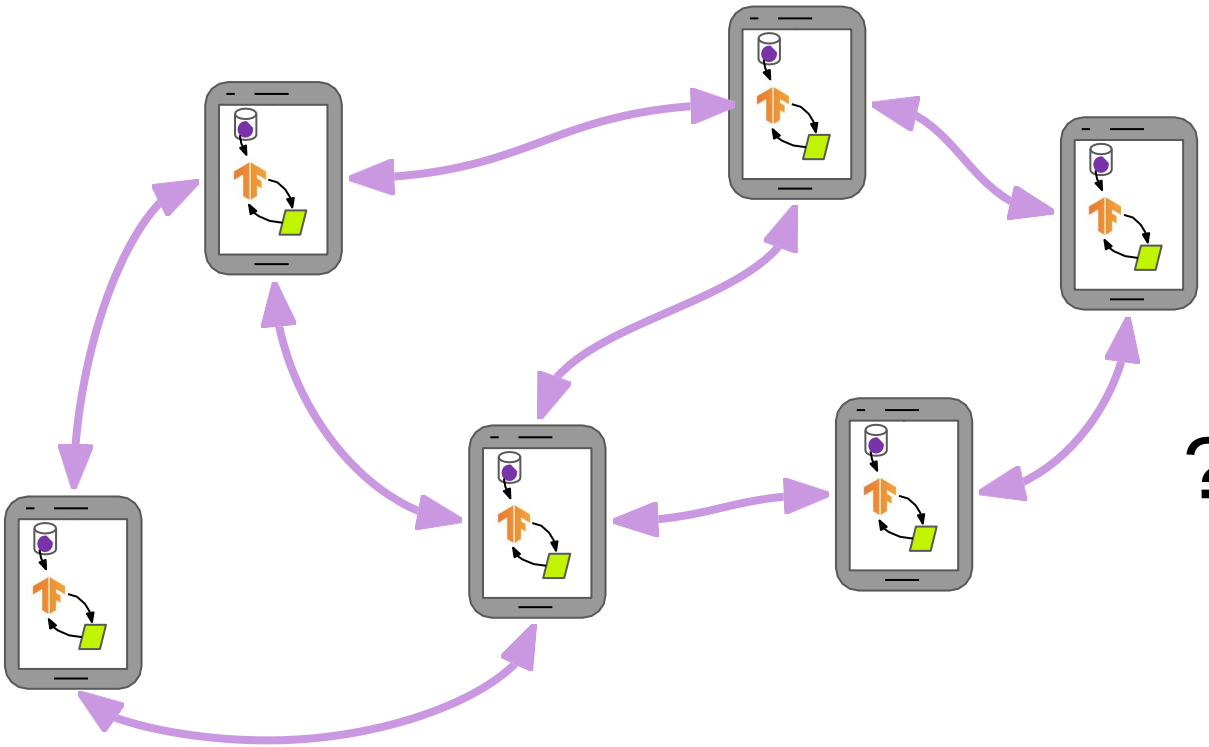
Adapted from Table 1 in *Advances and Open Problems in Federated Learning* (arxiv/1912.04977)

Characteristics of the federated learning setting

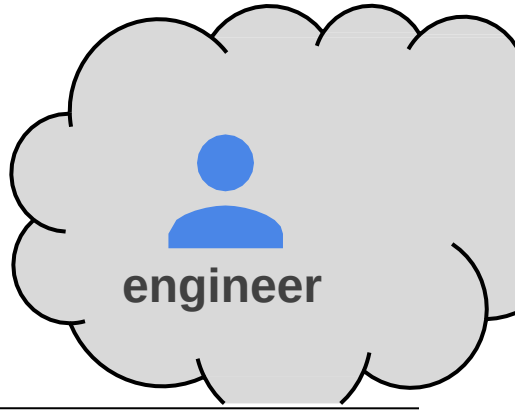
	Datacenter distributed learning	Cross-silo federated learning	Cross-device federated learning
Addressability	Each client has an identity or name that allows the system to access it specifically.		Clients cannot be indexed directly (i.e., no use of client identifiers)
Client statefulness	Stateful --- each client may participate in each round of the computation, carrying state from round to round.		Generally stateless --- each client will likely participate only once in a task, so generally we assume a fresh sample of never before seen clients in each round of computation.
Primary bottleneck	Computation is more often the bottleneck in the datacenter, where very fast networks can be assumed.	Might be computation or communication.	Communication is often the primary bottleneck, though it depends on the task. Generally, federated computations uses wi-fi or slower connections.
Reliability of clients	Relatively few failures.		Highly unreliable --- 5% or more of the clients participating in a round of computation are expected to fail or drop out (e.g., because the device becomes ineligible when battery, network, or idleness requirements for training/computation are violated).
Data partition axis	Data can be partitioned / re-partitioned arbitrarily across clients.	Partition is fixed. Could be example-partitioned (horizontal) or feature-partitioned (vertical).	Fixed partitioning by example (horizontal).

Adapted from Table 1 in *Advances and Open Problems in Federated Learning* (arxiv/1912.04977)

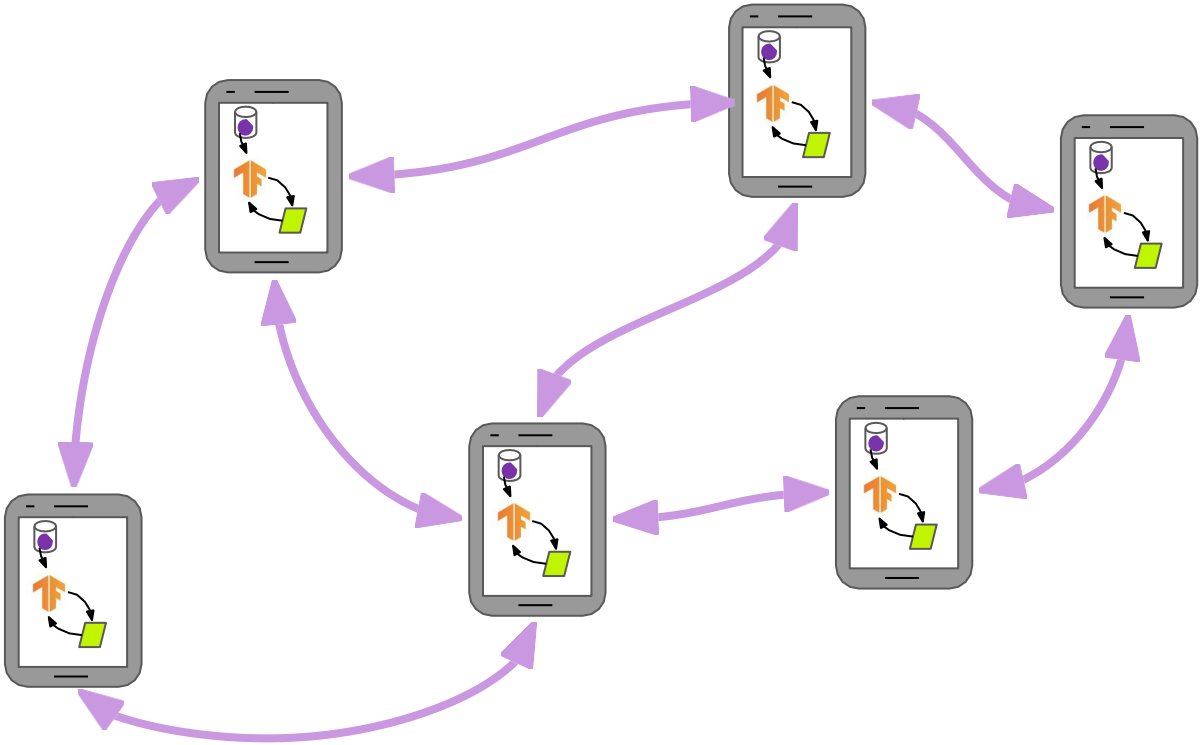
Fully decentralized (peer-to-peer) learning



?



Fully decentralized (peer-to-peer) learning



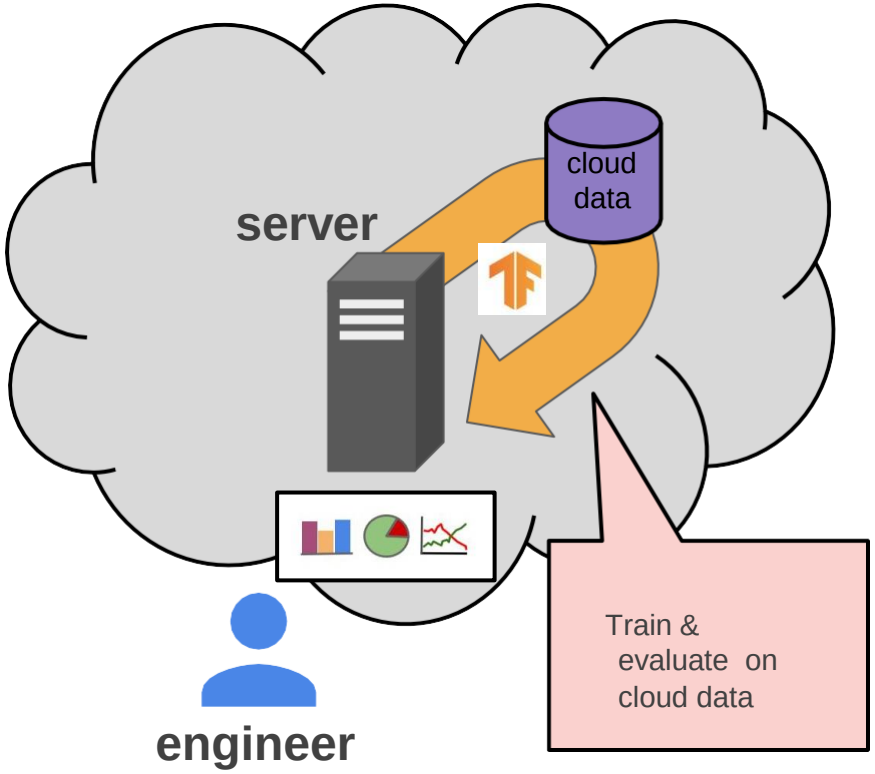
Characteristics of FL vs decentralized learning

	Federated learning	**Fully decentralized (peer-to-peer) learning**
Orchestration	A central orchestration server/service organizes the training, but never sees raw data.	No centralized orchestration.
Wide-area communication pattern	Typically hub-and-spoke topology, with the hub representing a coordinating service provider (typically without data) and the spokes connecting to clients.	Peer-to-peer topology.

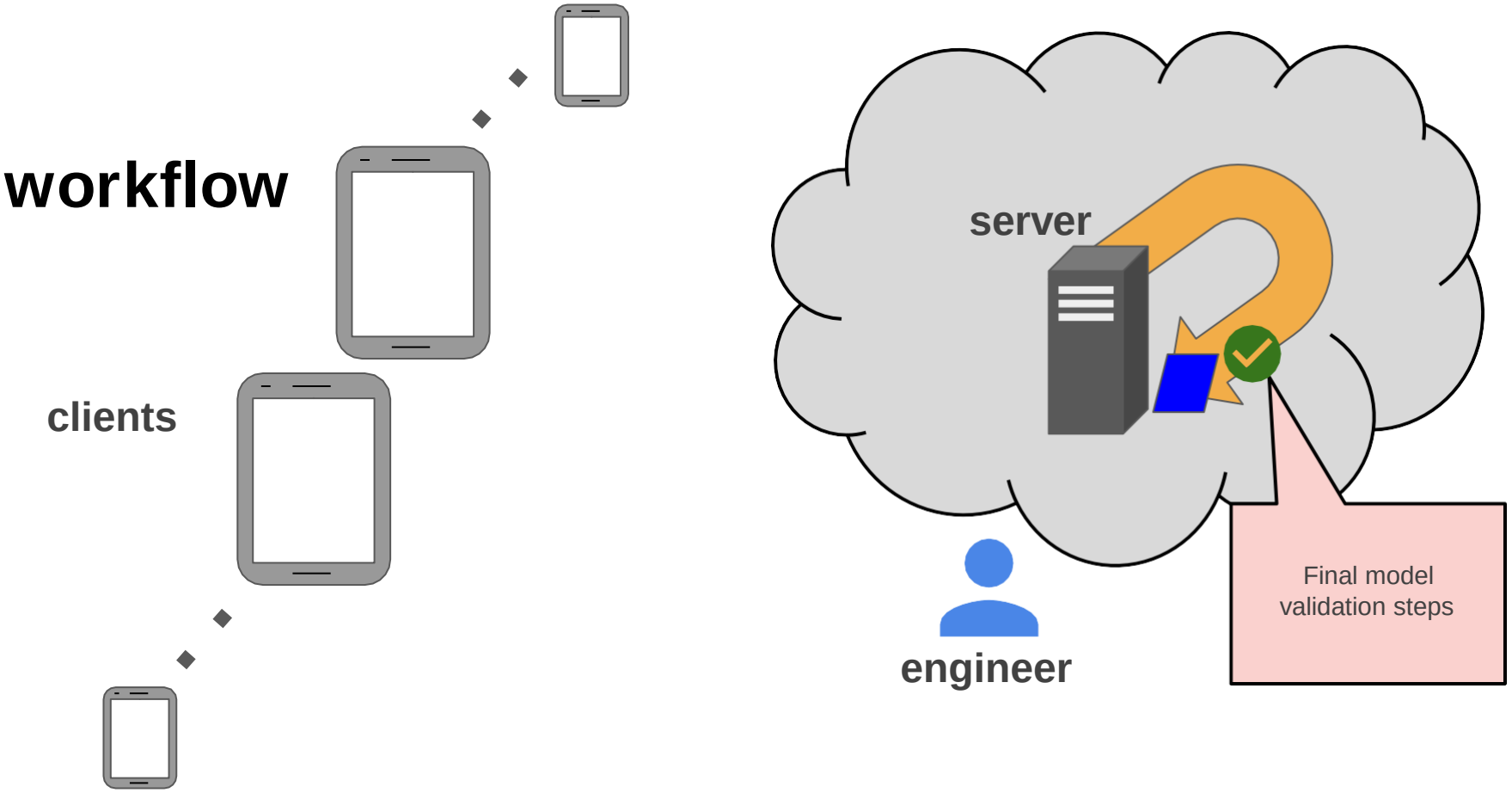
Adapted from Table 3 in *Advances and Open Problems in Federated Learning* (arxiv/1912.04977)

Cross-Device Federated Learning

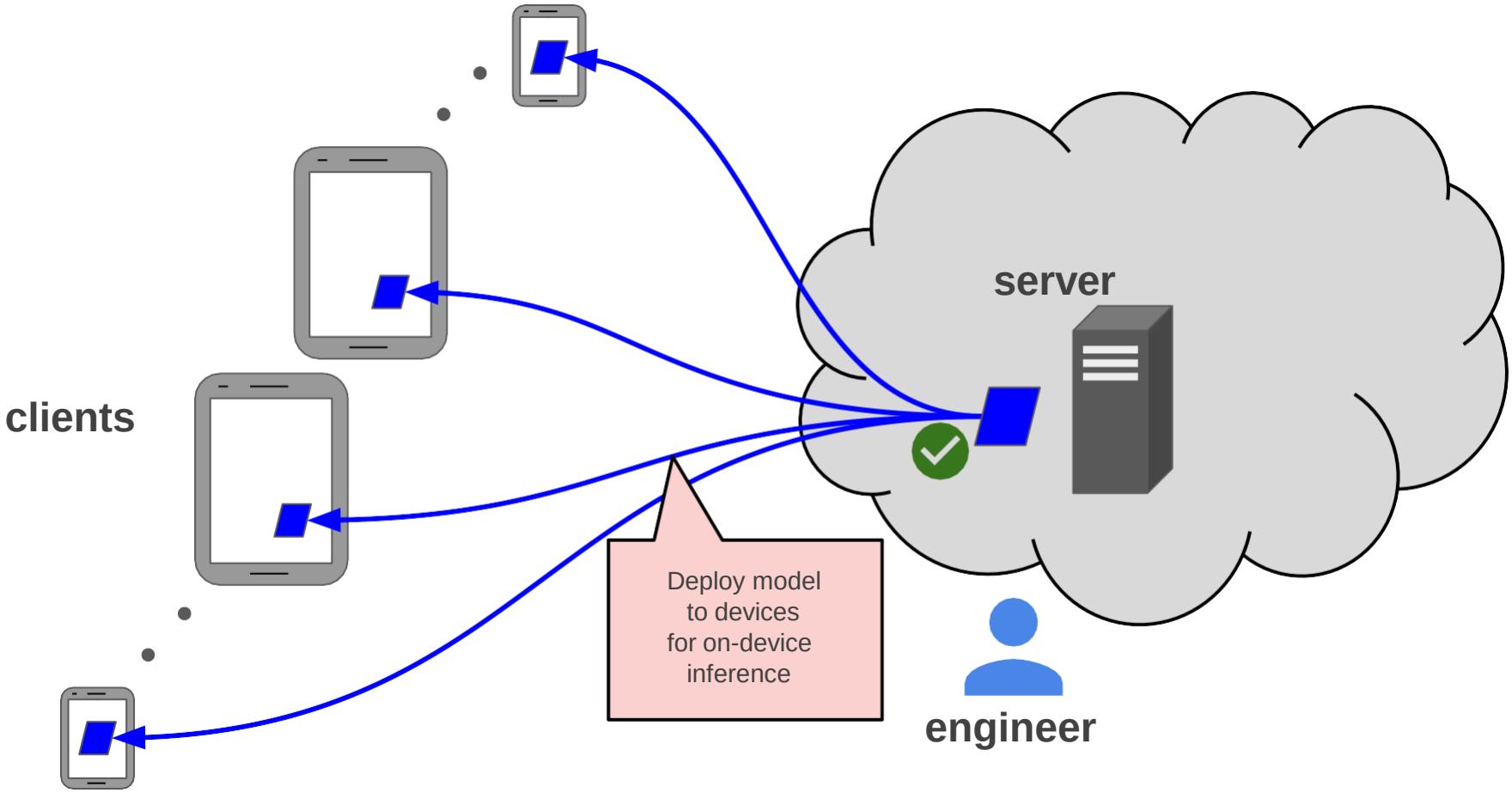
Model development workflow



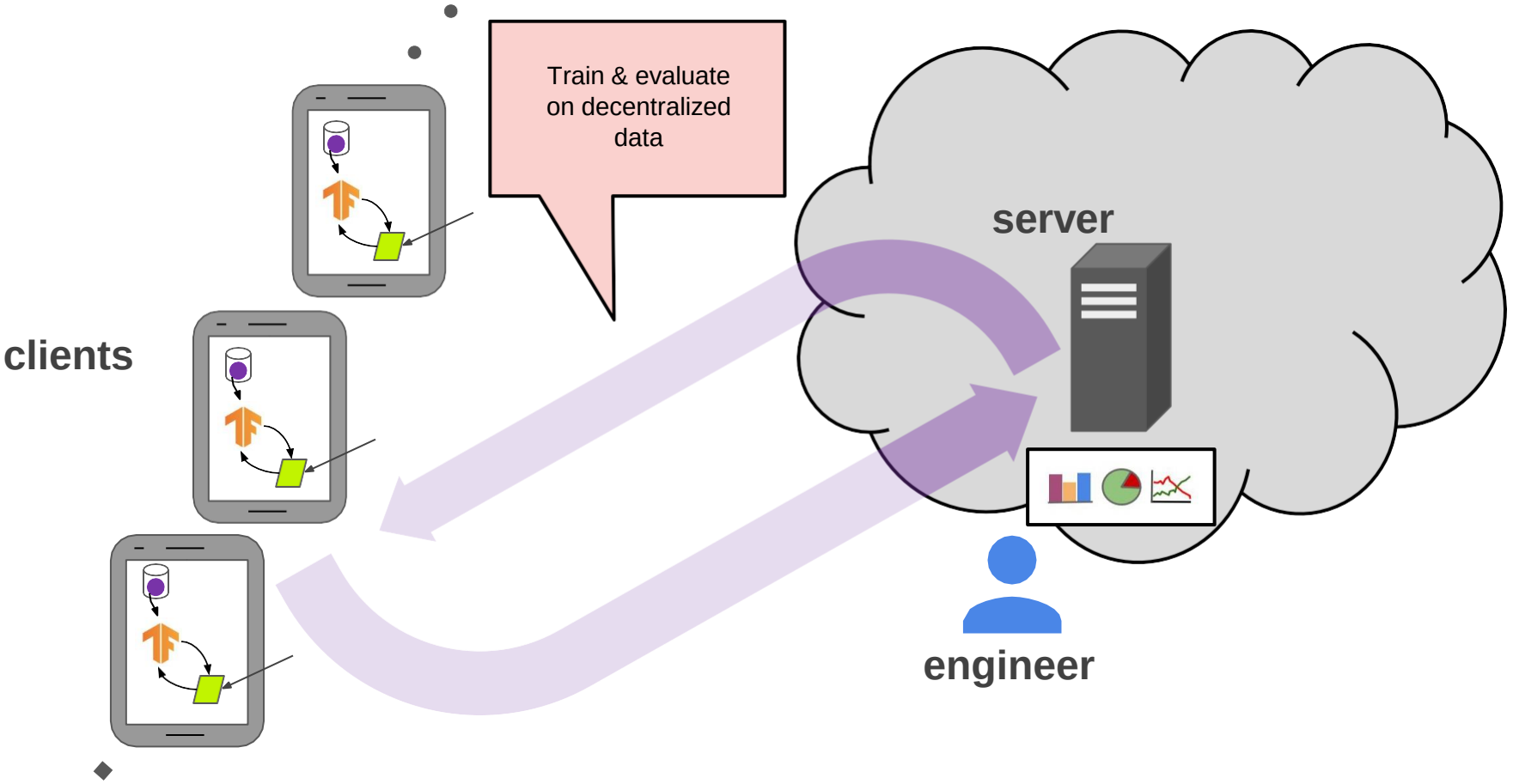
Model development



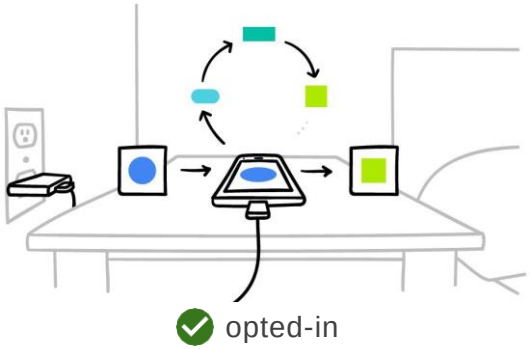
Model development workflow

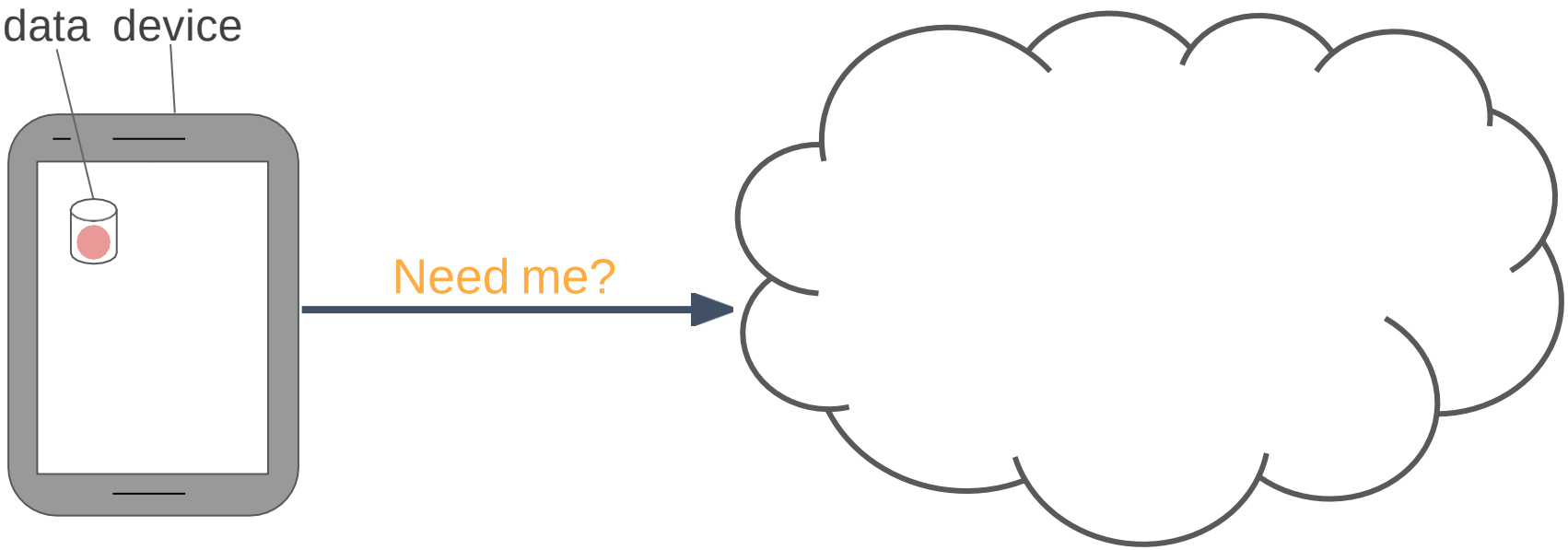


Federated training

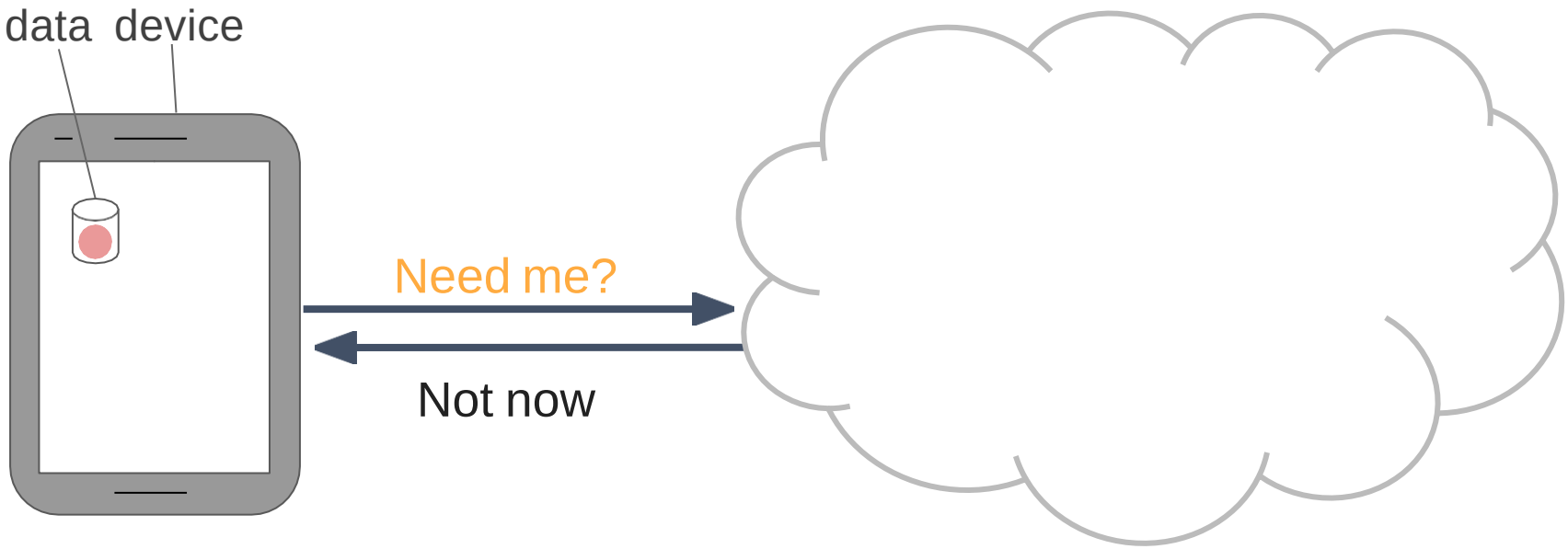


Federated learning

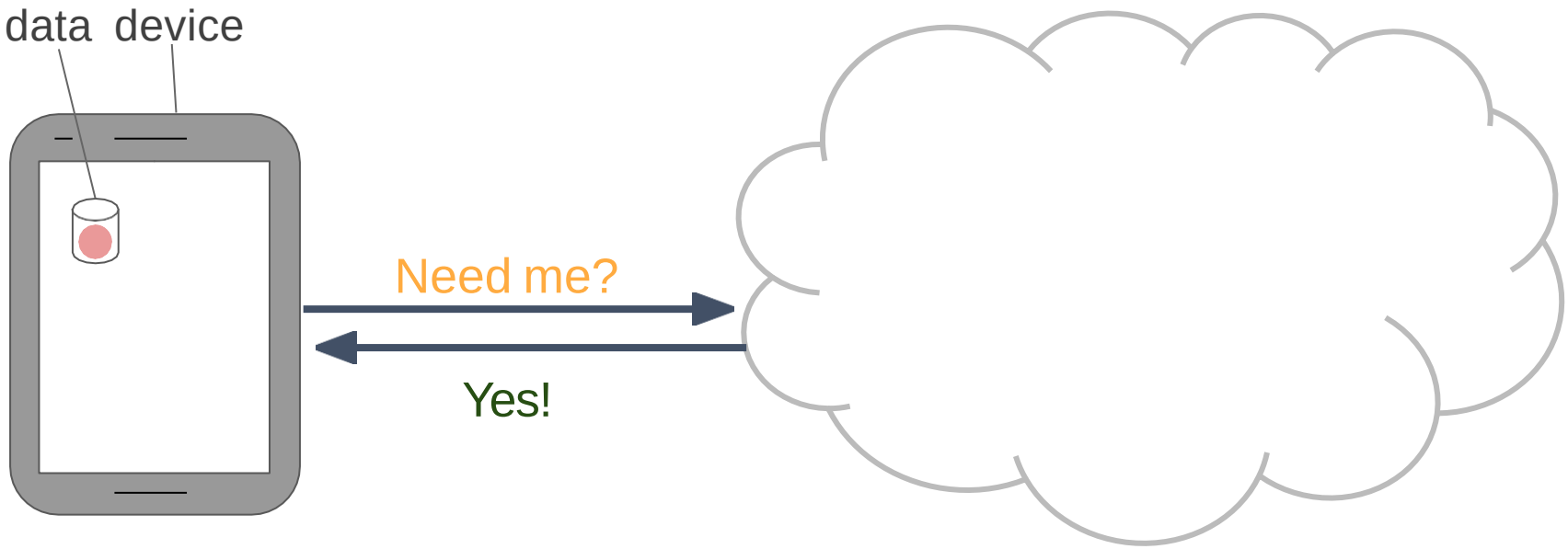




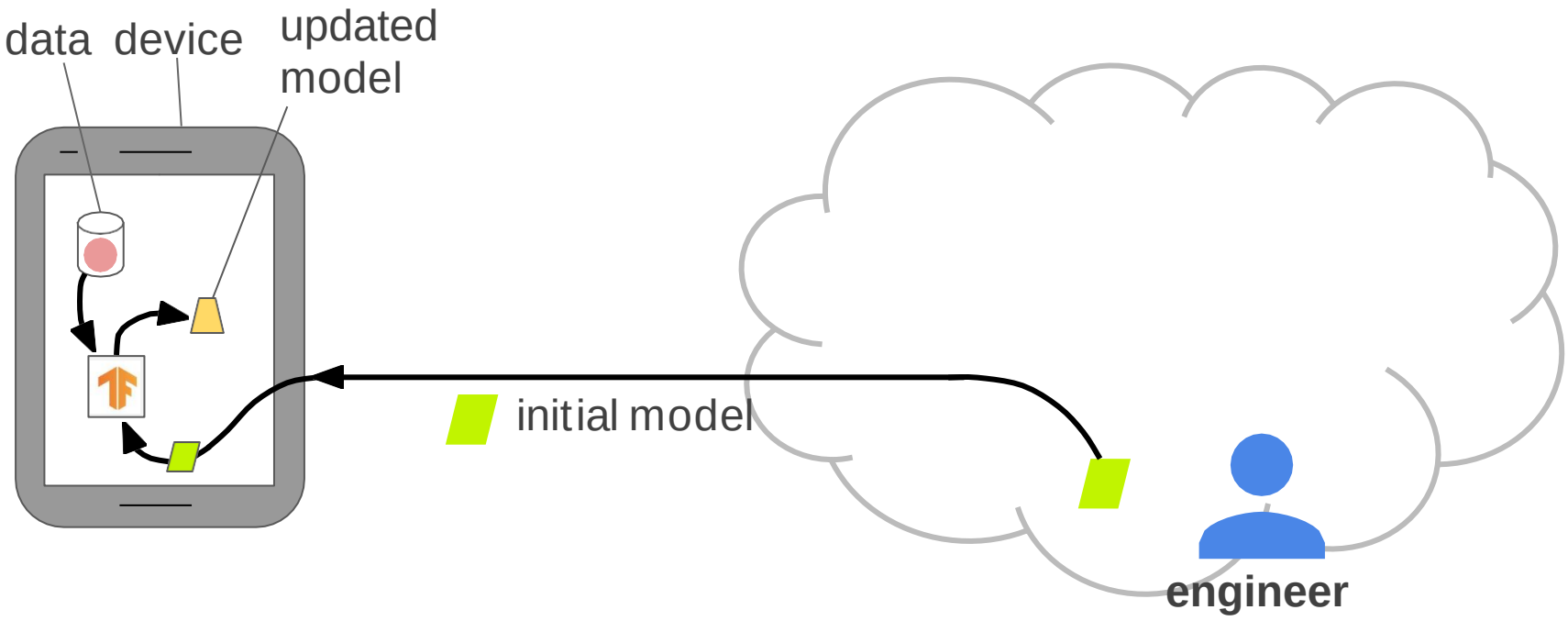
Federated learning



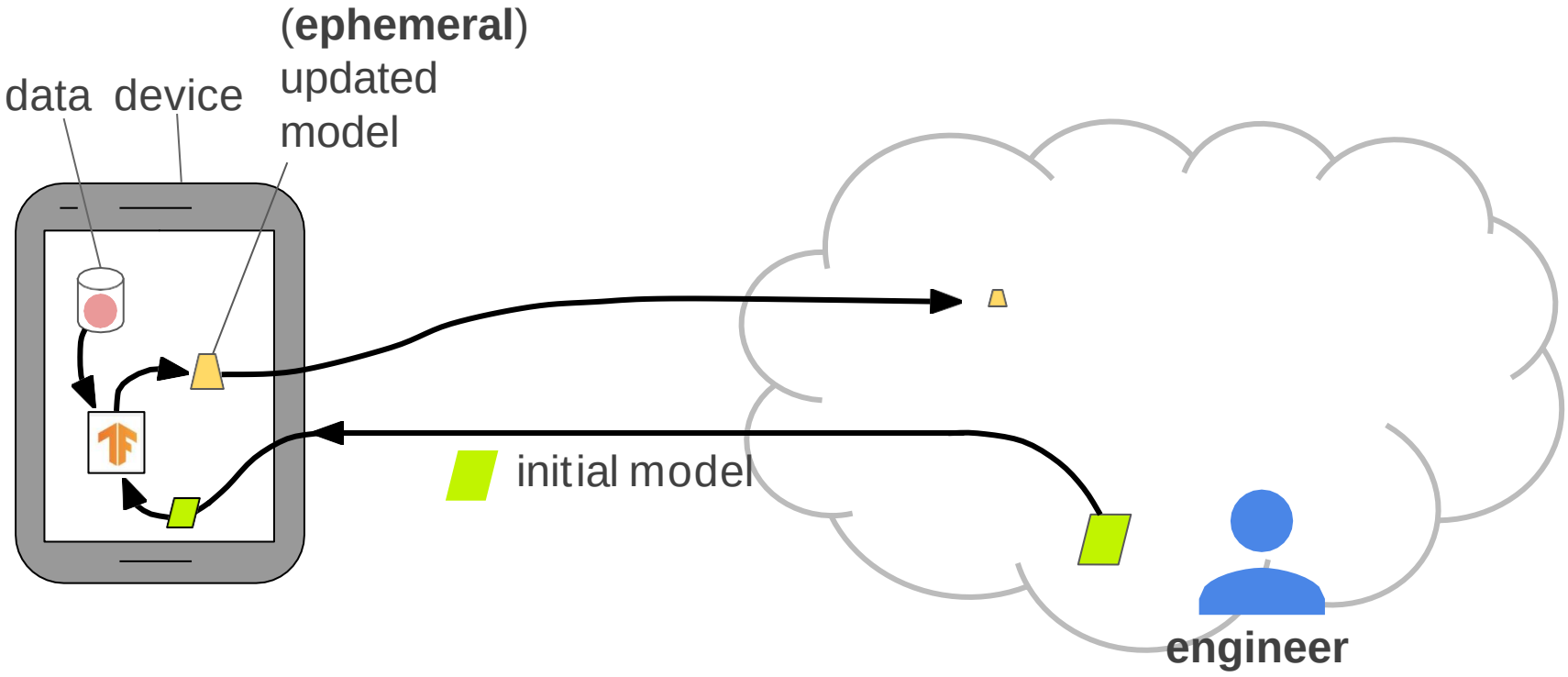
Federated learning



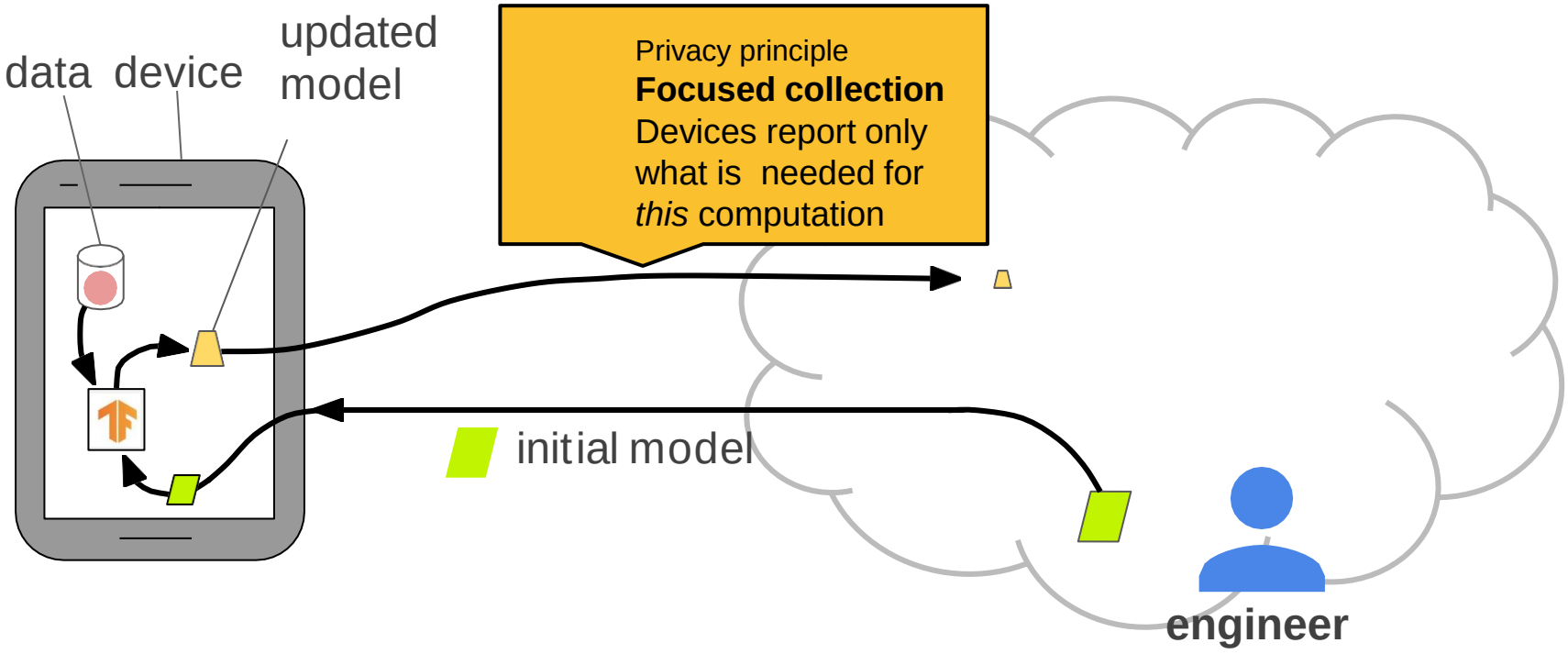
Federated learning



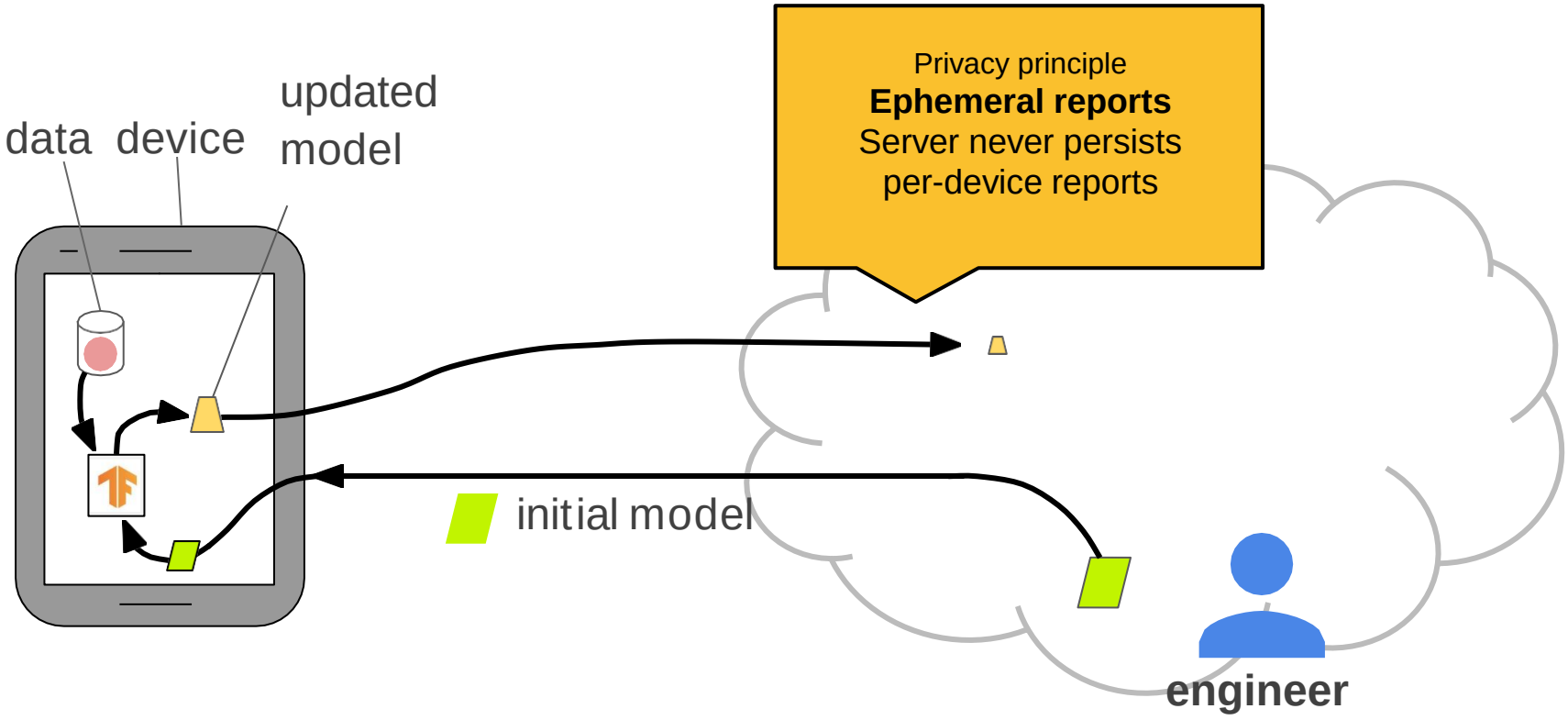
Federated learning



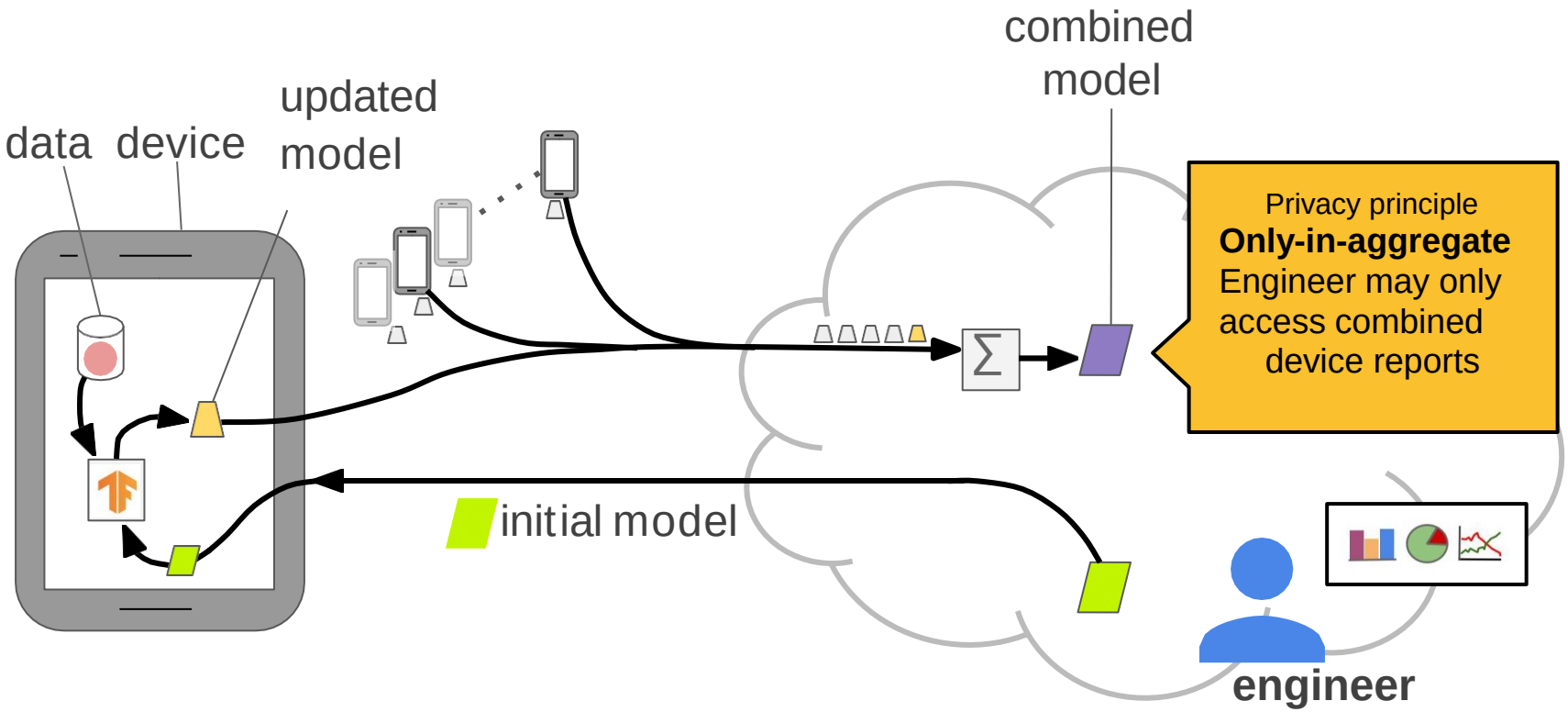
Federated learning



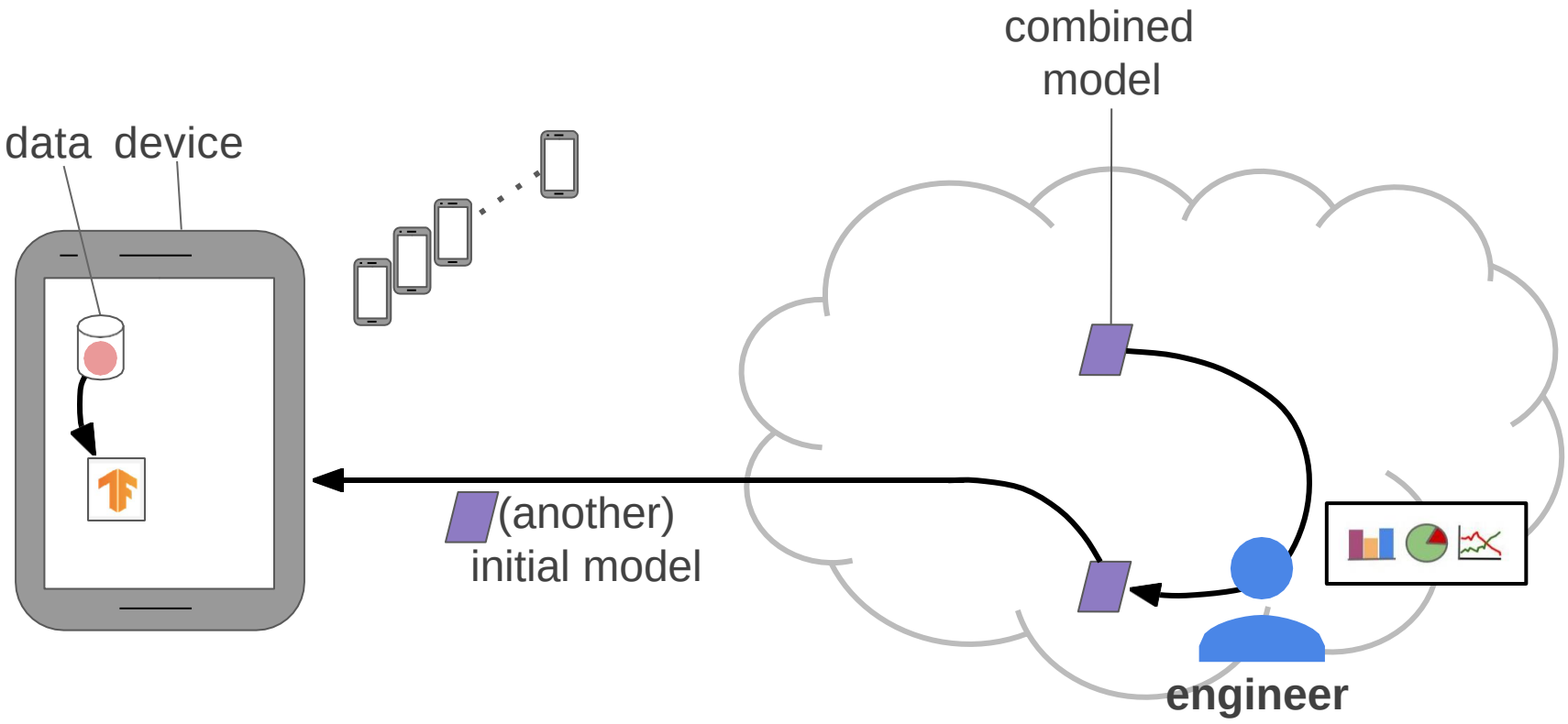
Federated learning



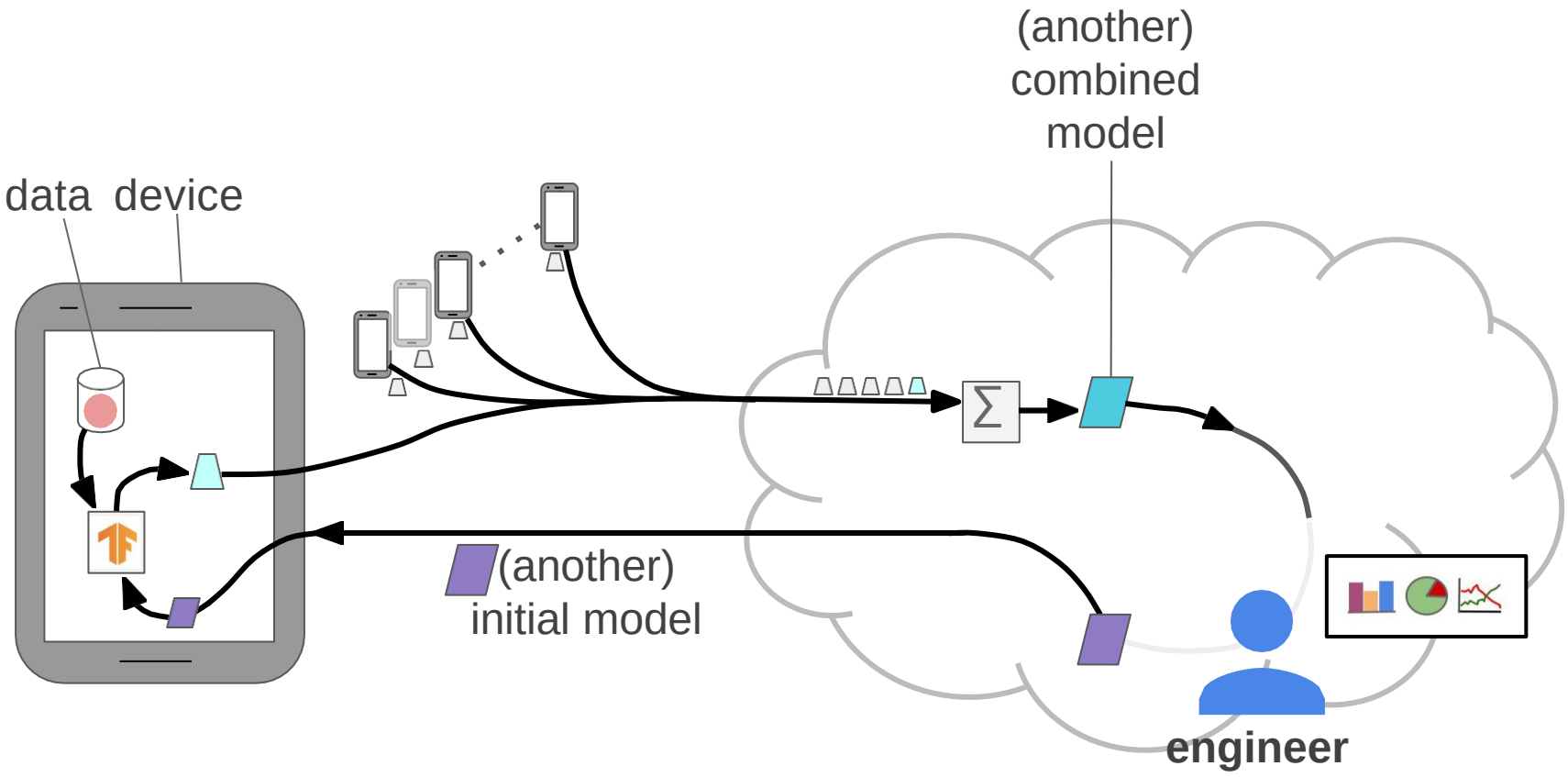
Federated learning



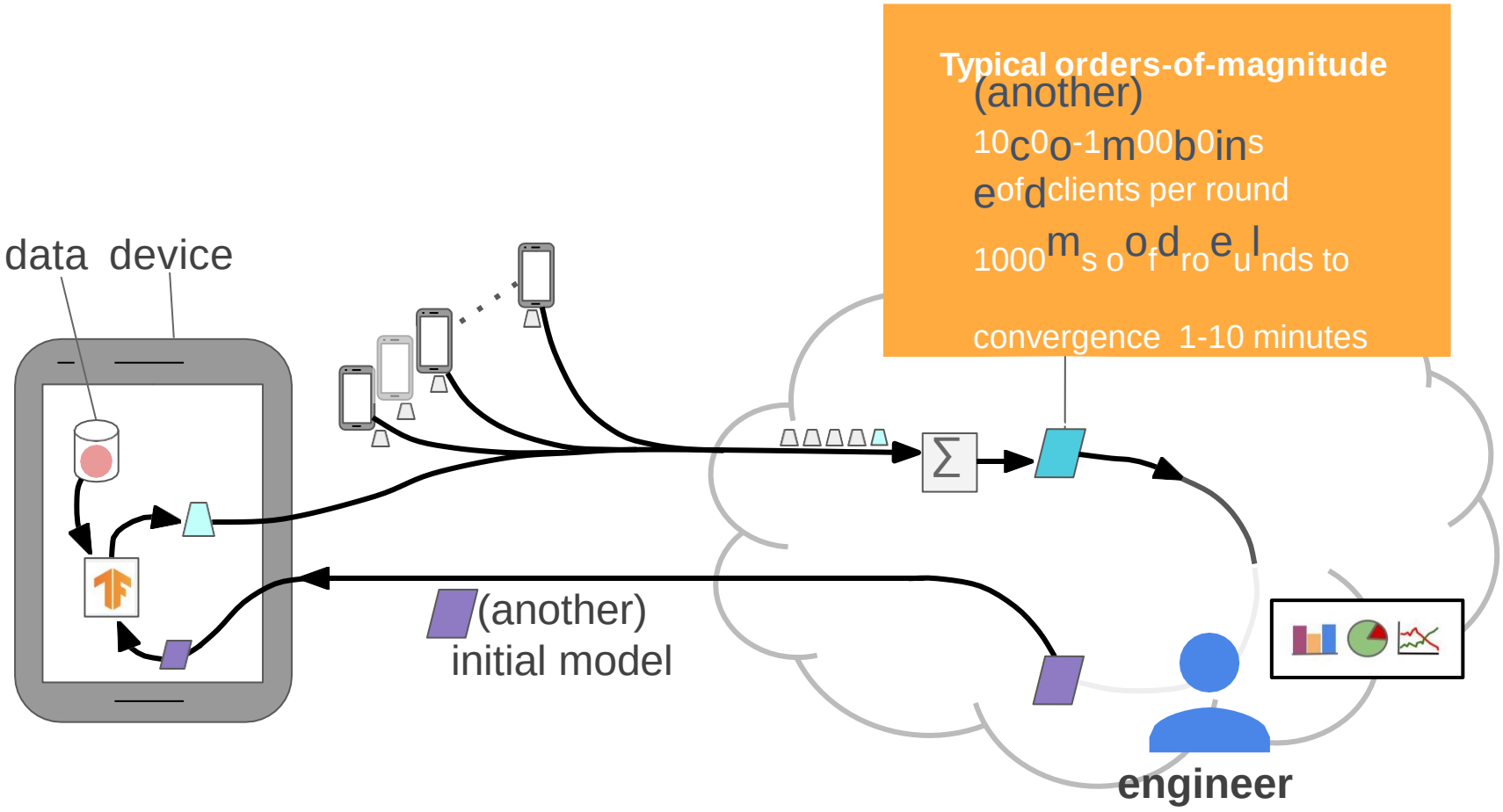
Federated learning



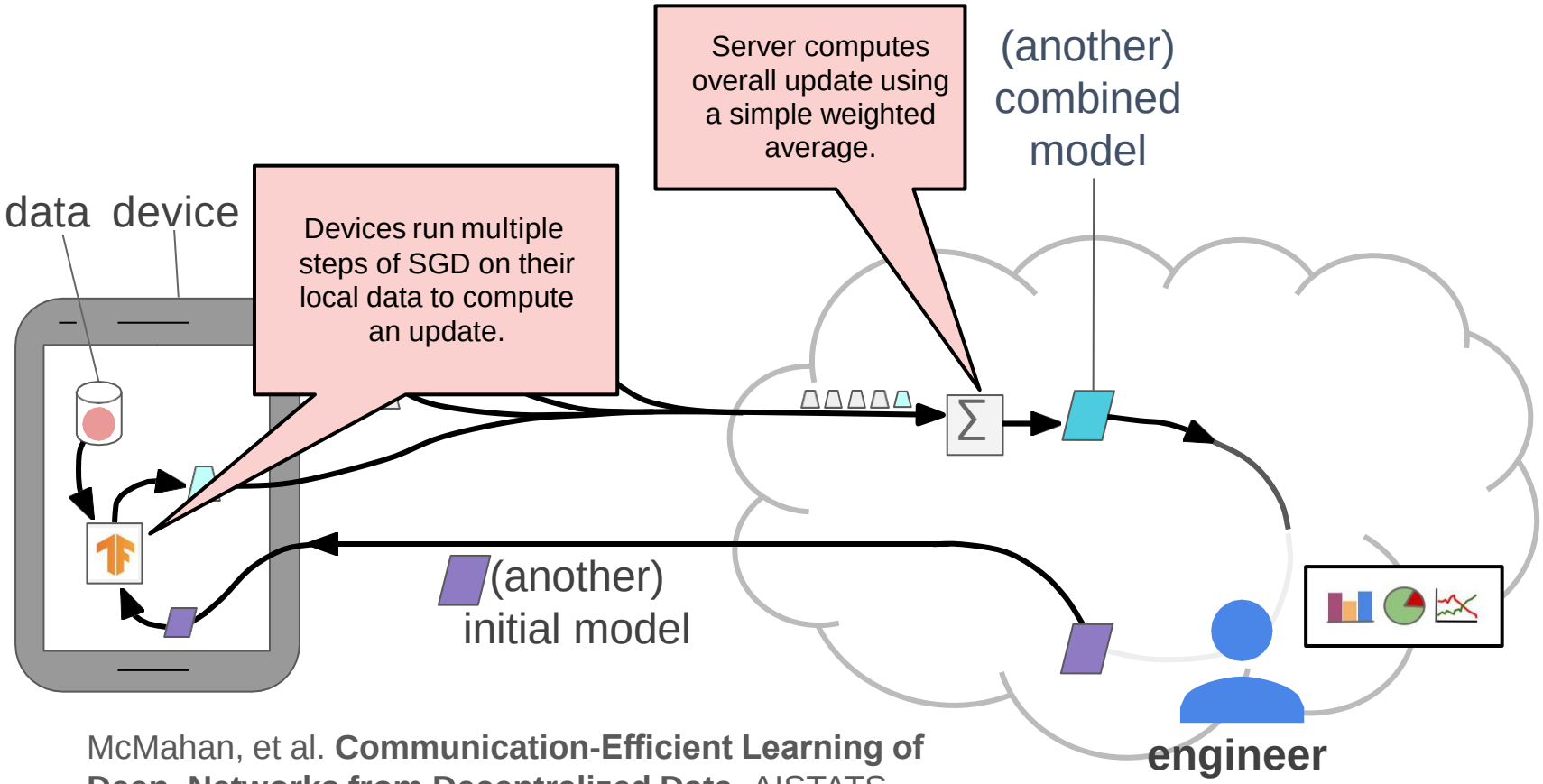
Federated learning



Federated learning



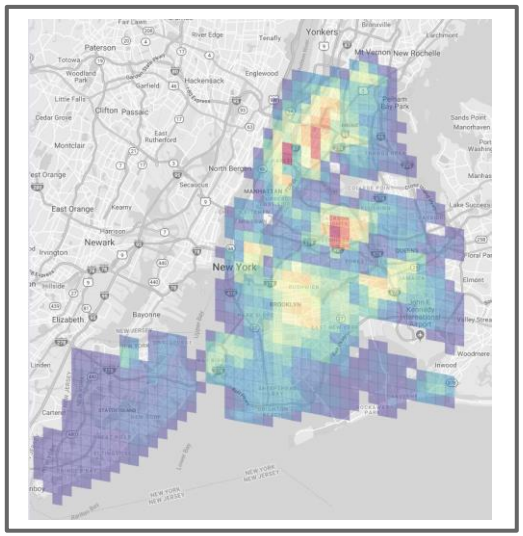
Federated Averaging (FedAvg) algorithm



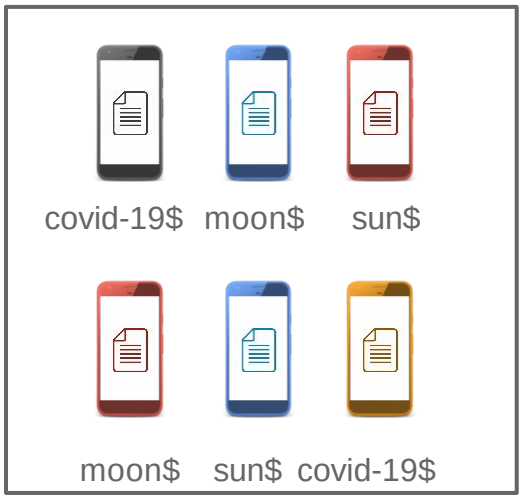
McMahan, et al. **Communication-Efficient Learning of Deep Networks from Decentralized Data.** AISTATS 2017.

Beyond Learning: Federated Analytics

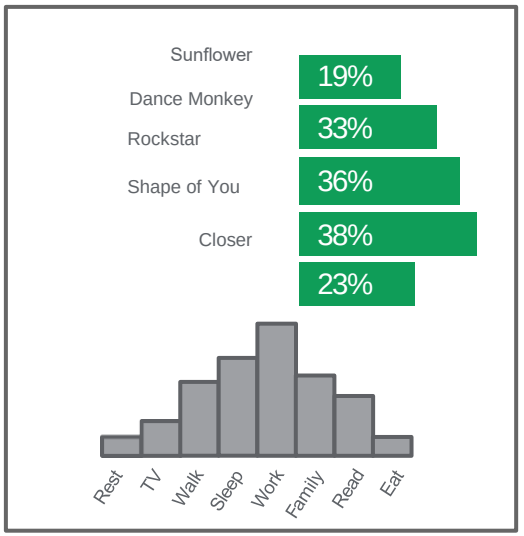
Beyond learning: federated analytics



Geo-location heatmaps



Frequently typed
out-of-dictionary words



Popular songs,
trends, and
activities

Federated analytics

Federated analytics is the practice of applying data science methods to the analysis of raw data that is stored locally on users' devices. Like federated learning, it works by running local computations over each device's data, and only making the aggregated results — and never any data from a particular device — available to product engineers. Unlike federated learning, however, federated analytics aims to support basic data science needs.

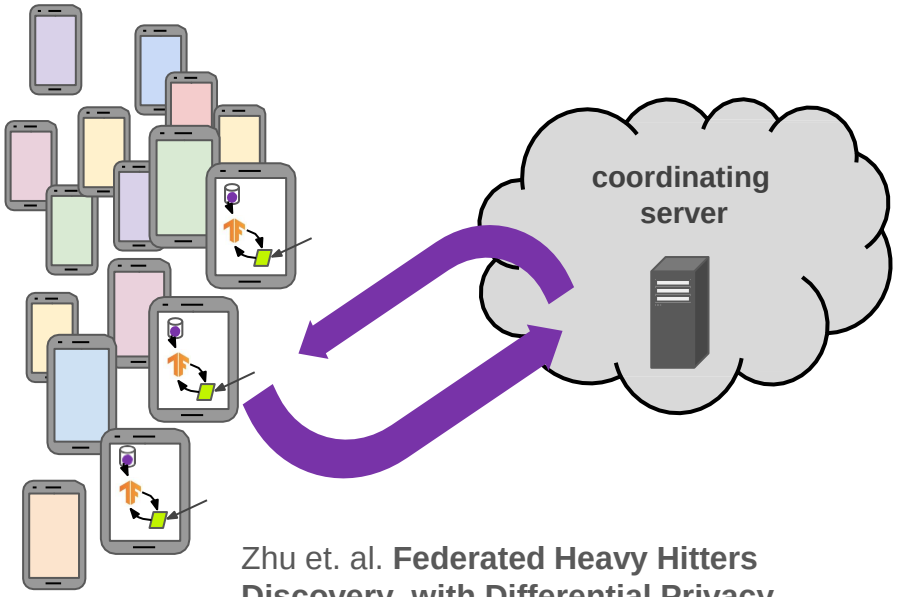
definition proposed in https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html

Federated analytics

- Federated histograms over closed sets
- Federated quantiles and distinct element counts
- Federated heavy hitters discovery over open sets
- Federated density of vector spaces
- Federated selection of random data subsets
- Federated SQL
- Federated computations?
- etc...

Interactive algorithms

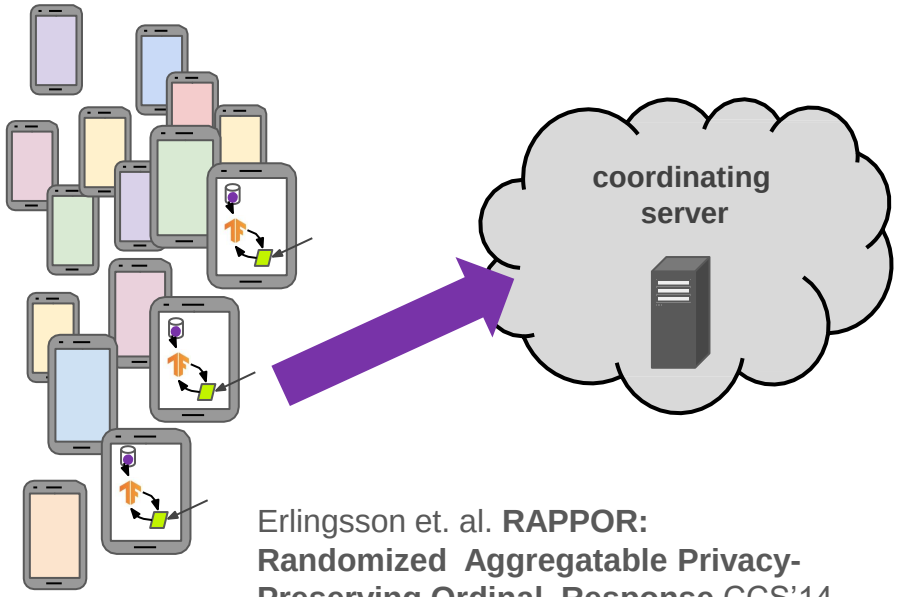
Similar to learning, the on-device computation is a function of a server state



Zhu et. al. **Federated Heavy Hitters Discovery with Differential Privacy** AISTATS'20.

Non-interactive algorithms

Unlike learning, the on device computation does not depend on a server state



Erlingsson et. al. **RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response** CCS'14.

Part II: Privacy for Federated Learning and Analytics

Aspects of Privacy

The why, what, and how of using data.

Why?
Transparency & consent

Why use this data? The user understands and supports the intended use of the data.

What?
Limited influence of any individual

What is computed? When data is released, ensure it does not reveal any user's private information.

How?
Security & data minimization

How and where does the computation happen? Release data to as few parties as possible. Minimize the attack surface where private information could be accessed.

Aspects of Privacy

The why, what, and how of using data.

Why?
Transparency & consent

Why use this data? The user understands and supports the intended use of the data.

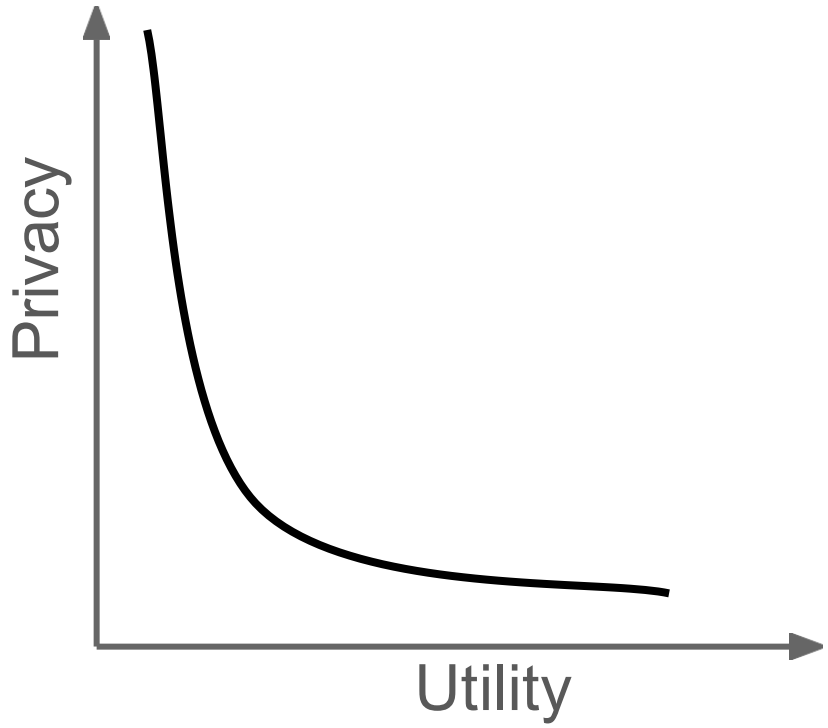
What?
Limited influence of any individual

What is computed? When data is released, ensure it does not reveal any user's private information.

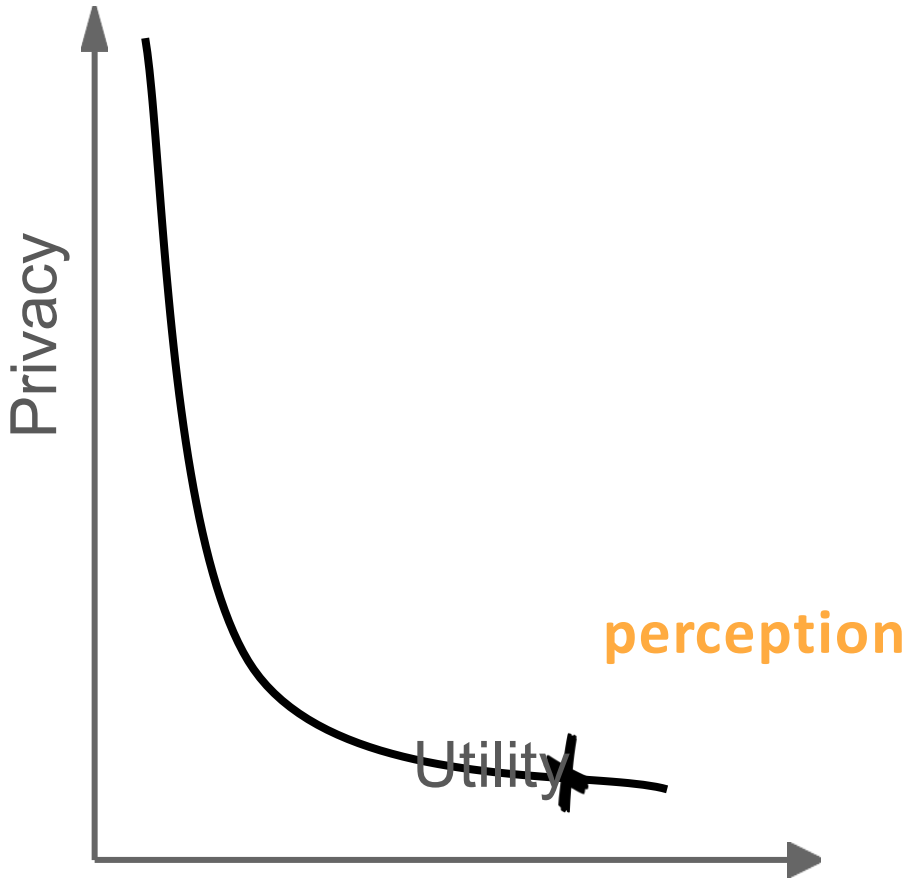
How?
Security & data minimization

How and where does the computation happen? Release data to as few parties as possible. Minimize the attack surface where private information could be accessed.

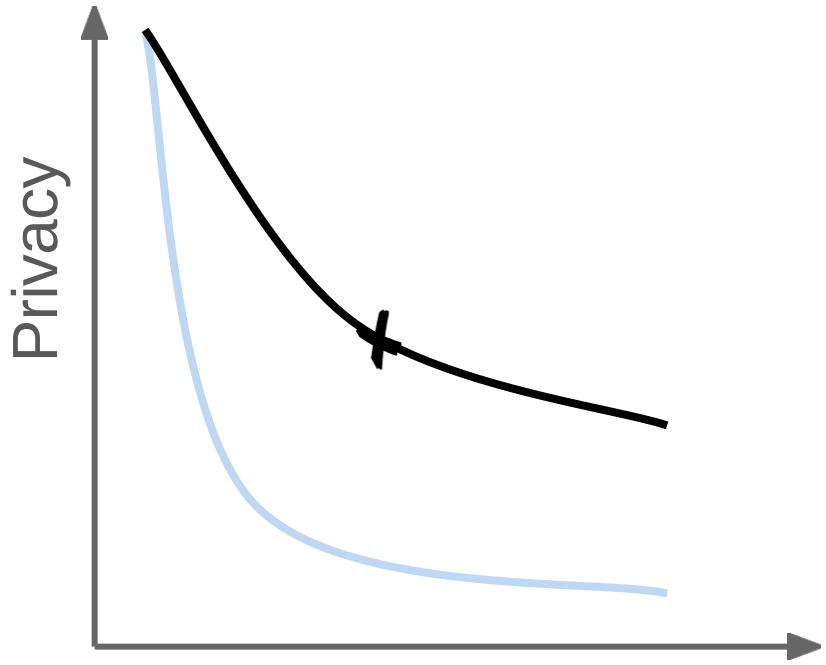
ML on sensitive data: privacy vs. utility



ML on sensitive data: privacy vs. utility



ML on sensitive data: privacy vs. utility

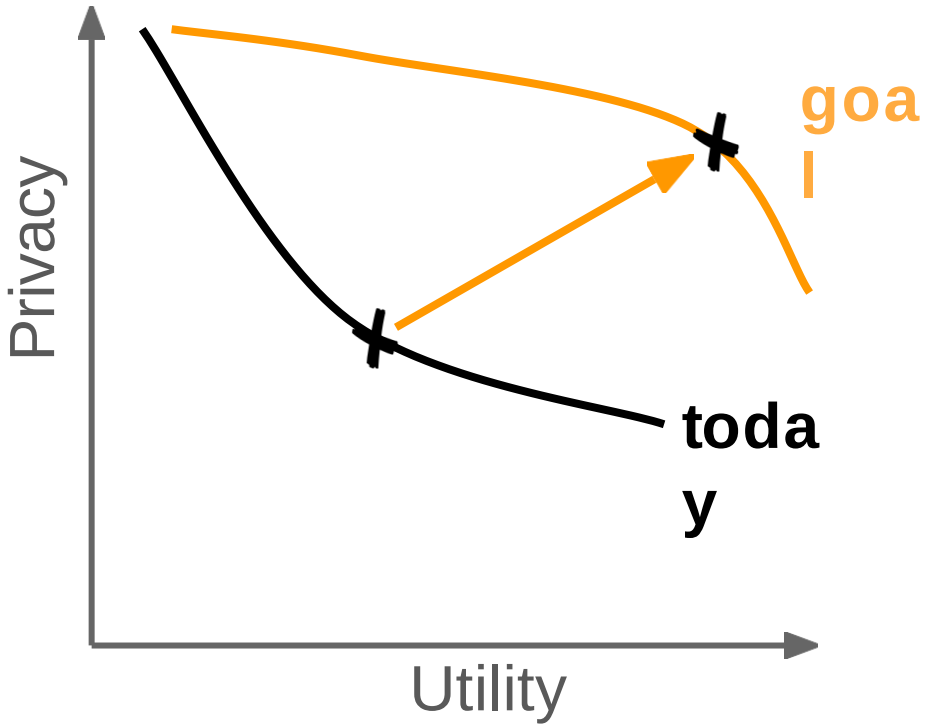


1. Policy
2. Technology

today

perception

ML on sensitive data: privacy vs. utility

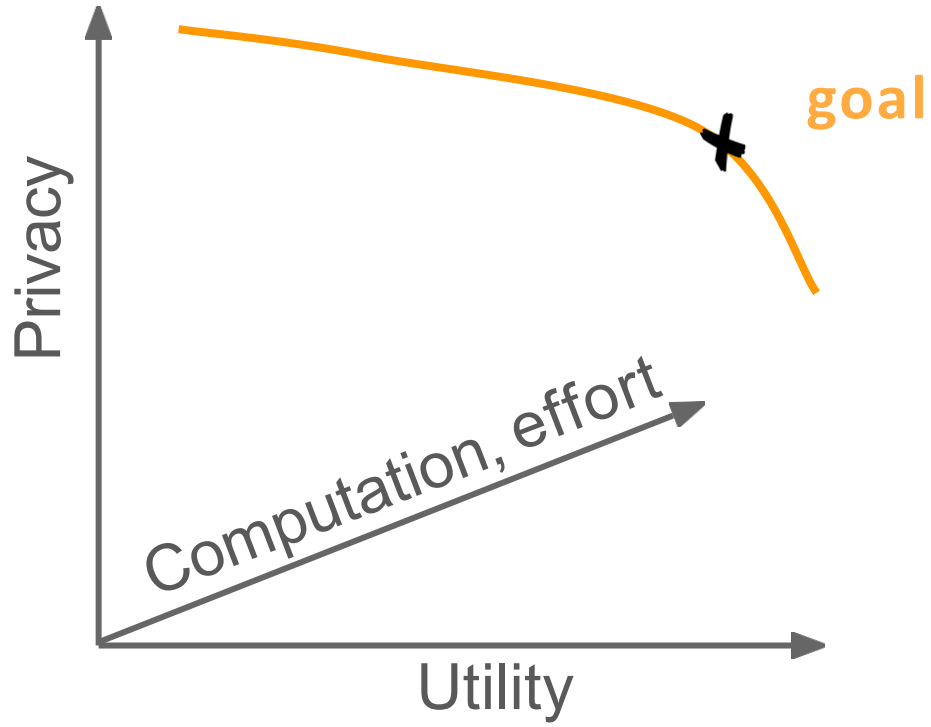


1. Policy

2. New Technology

Push the pareto frontier with better technology.
Make achieving high privacy and utility possible.

ML on sensitive data: privacy vs. utility (?)

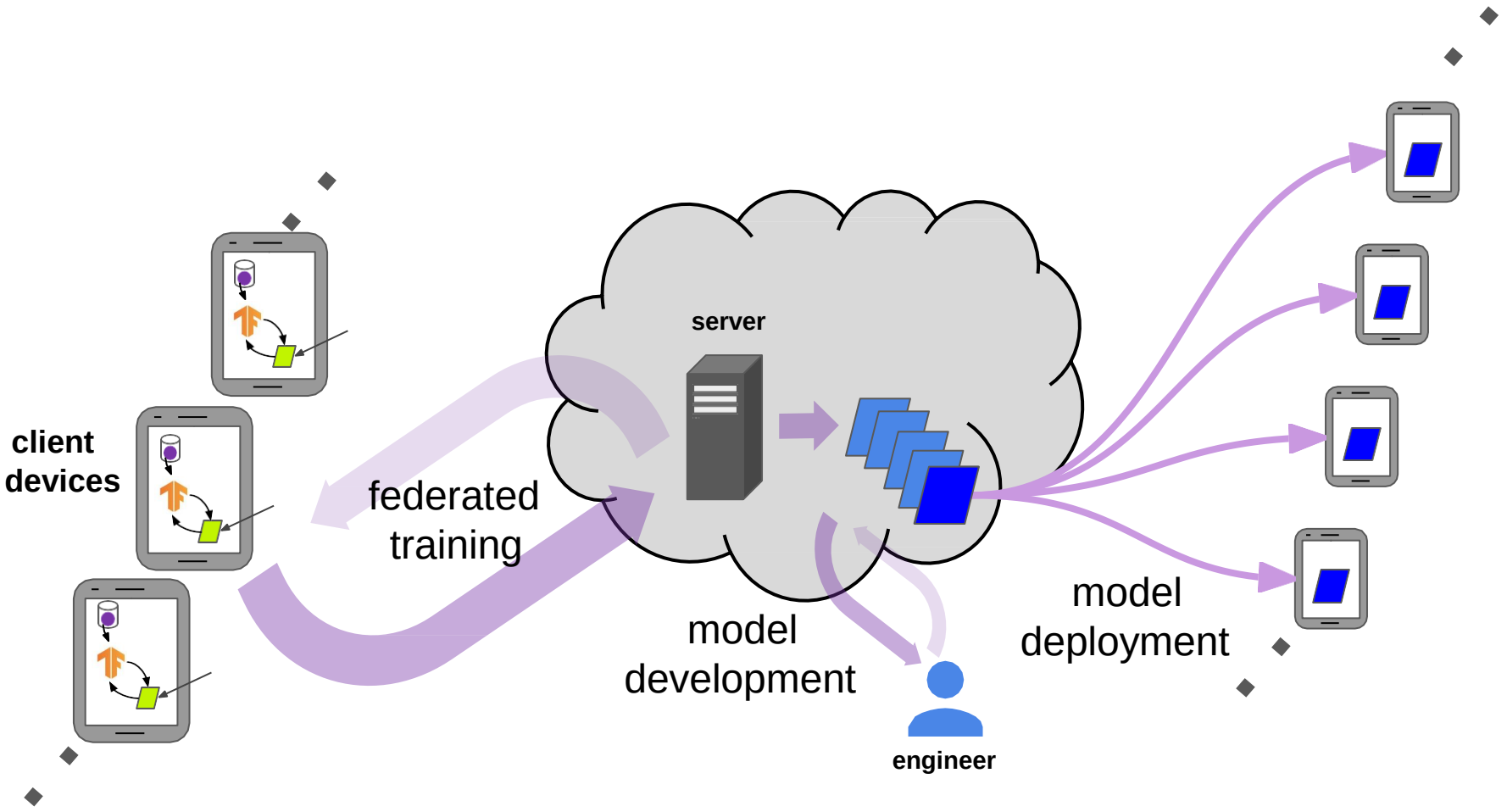


1. Policy
2. **New Technology**

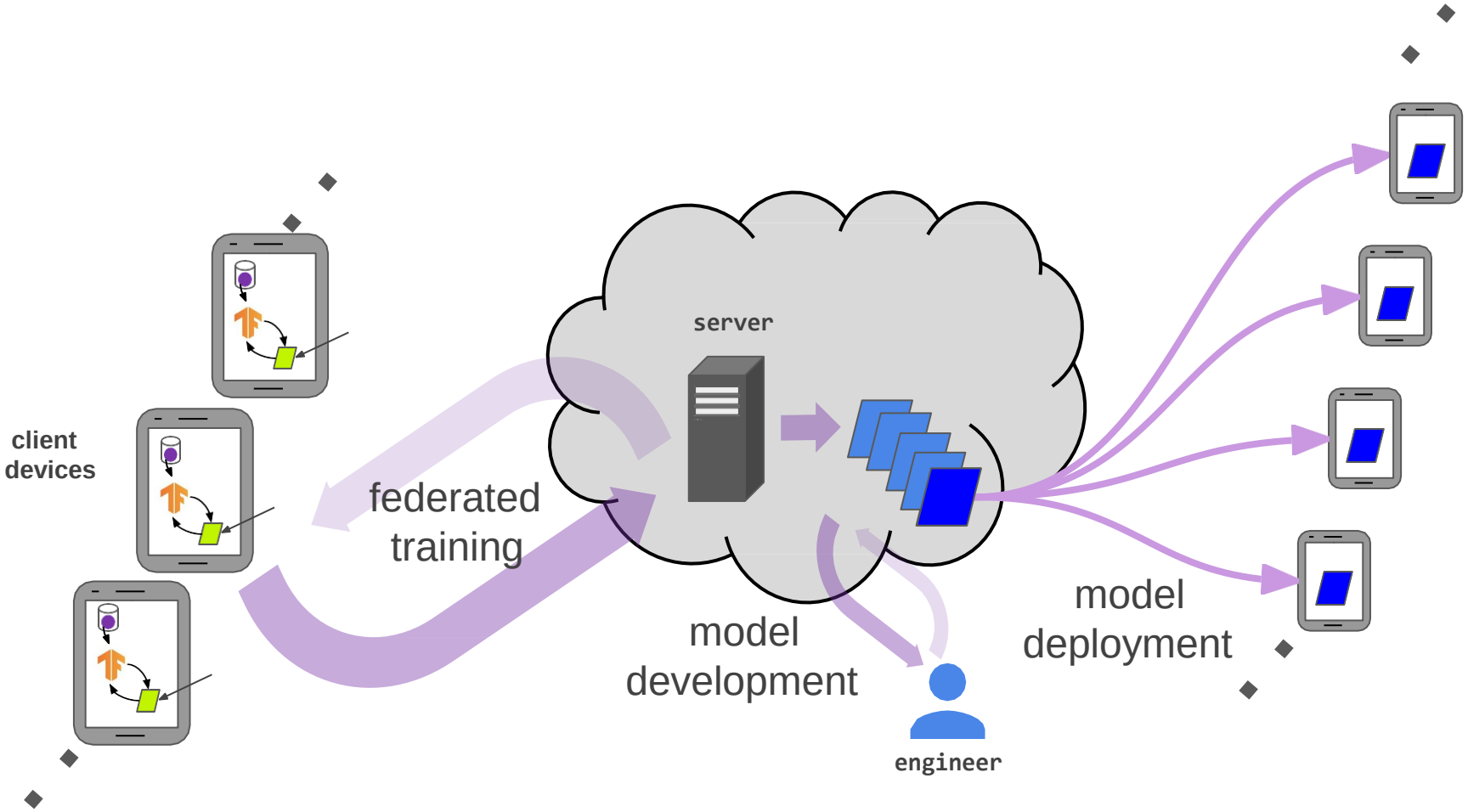
Push the pareto frontier with better technology.

Make achieving high privacy and utility possible **with less work**.

What private information might an actor learn?

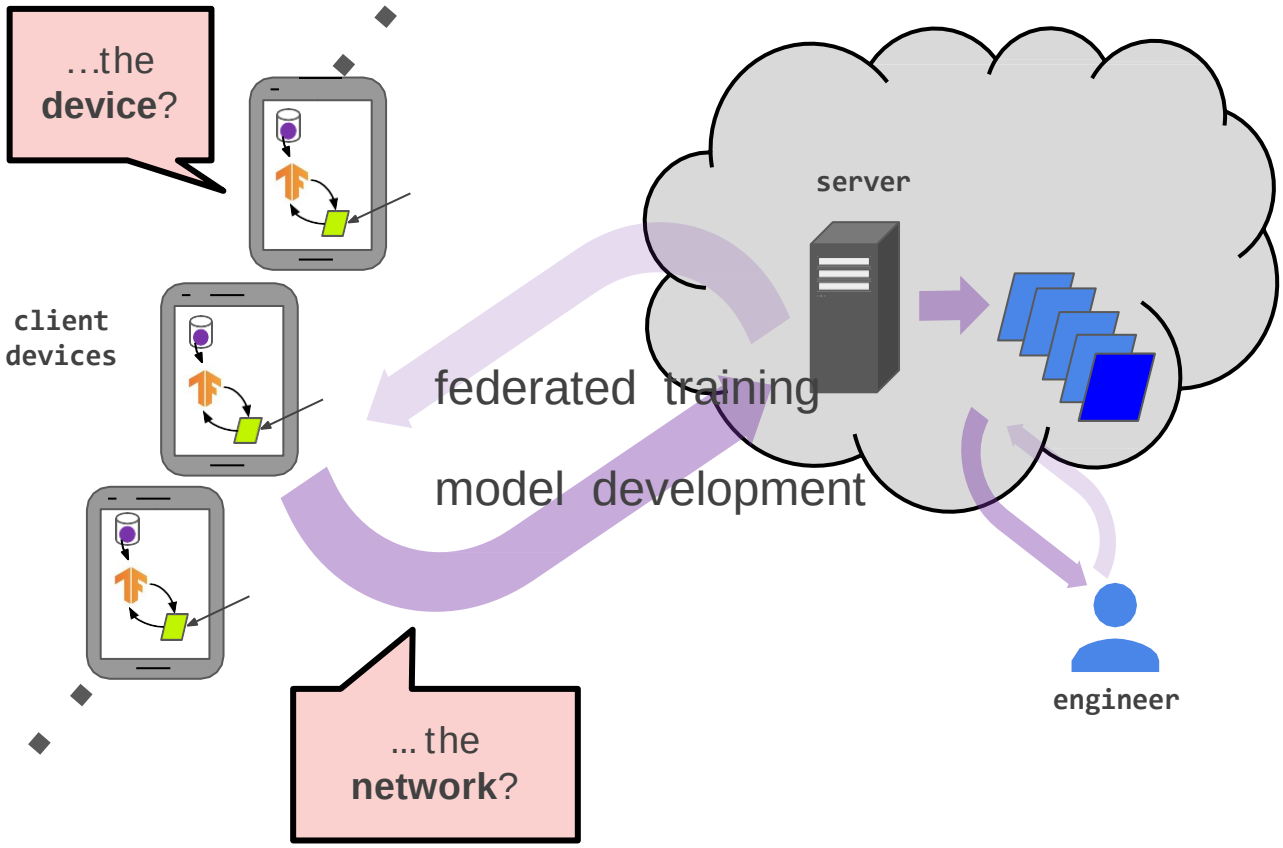


What private information might an actor learn?



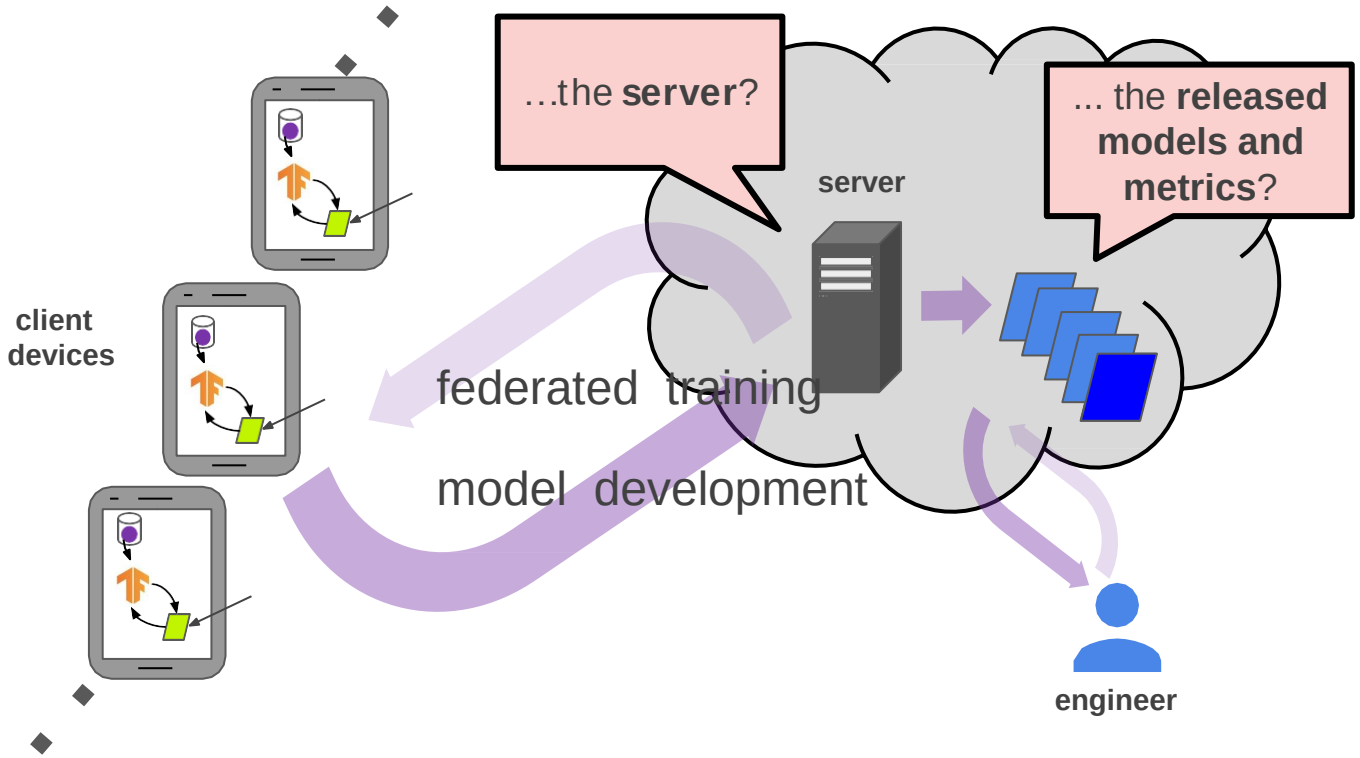
What private information might an actor learn

with access to ...

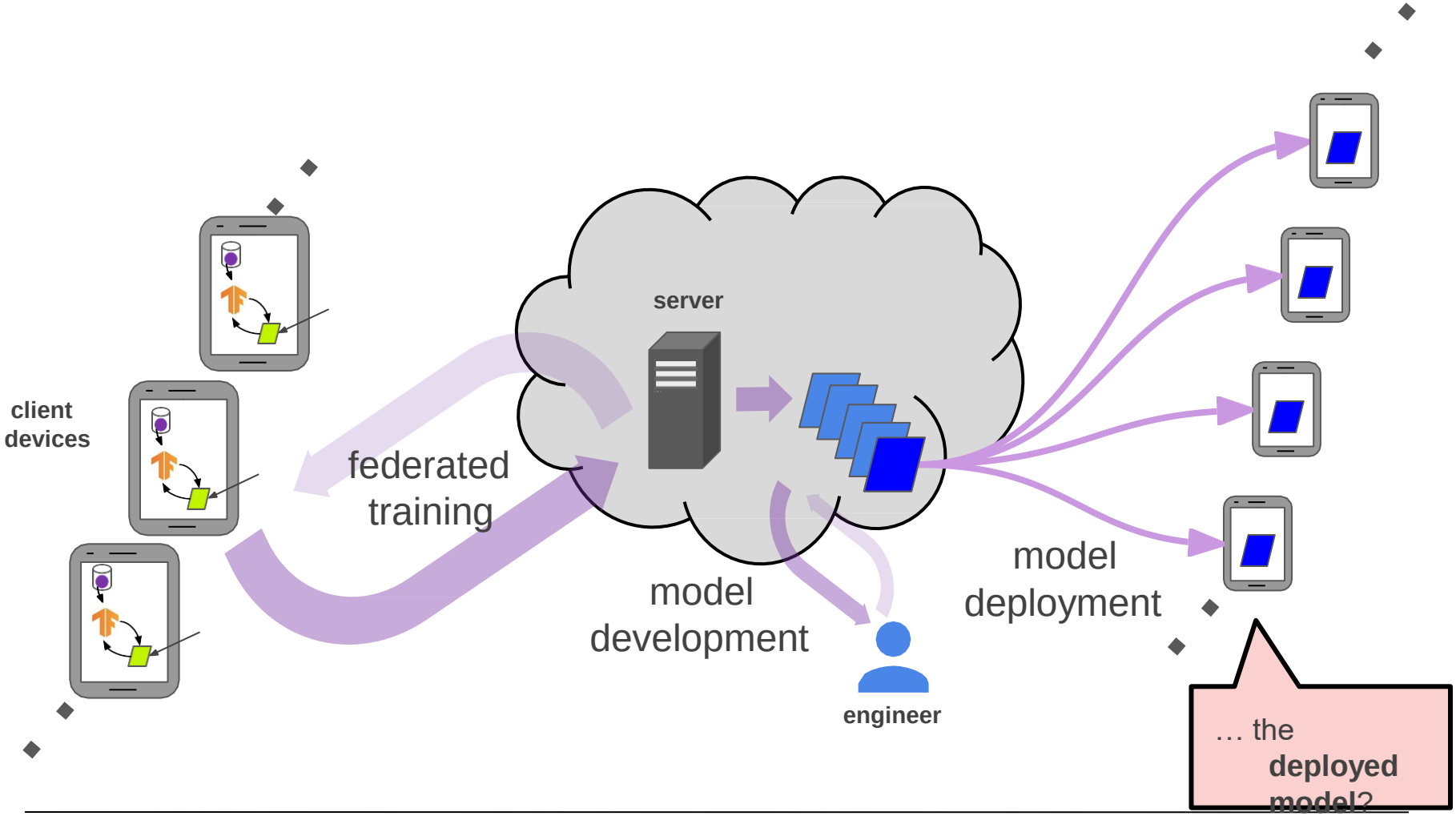


What private information might an actor learn

with access to ...

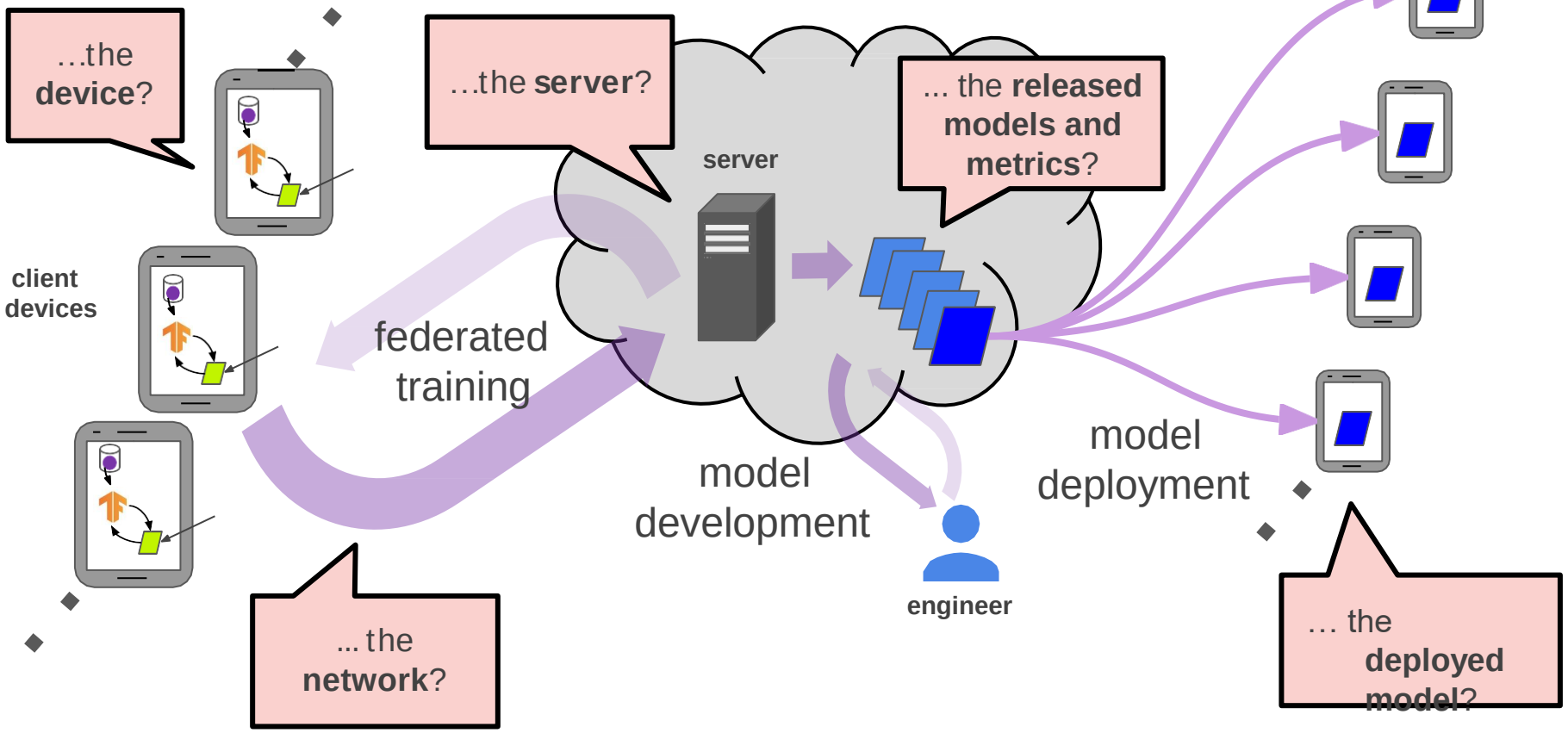


What private information might an actor learn
with access to ...

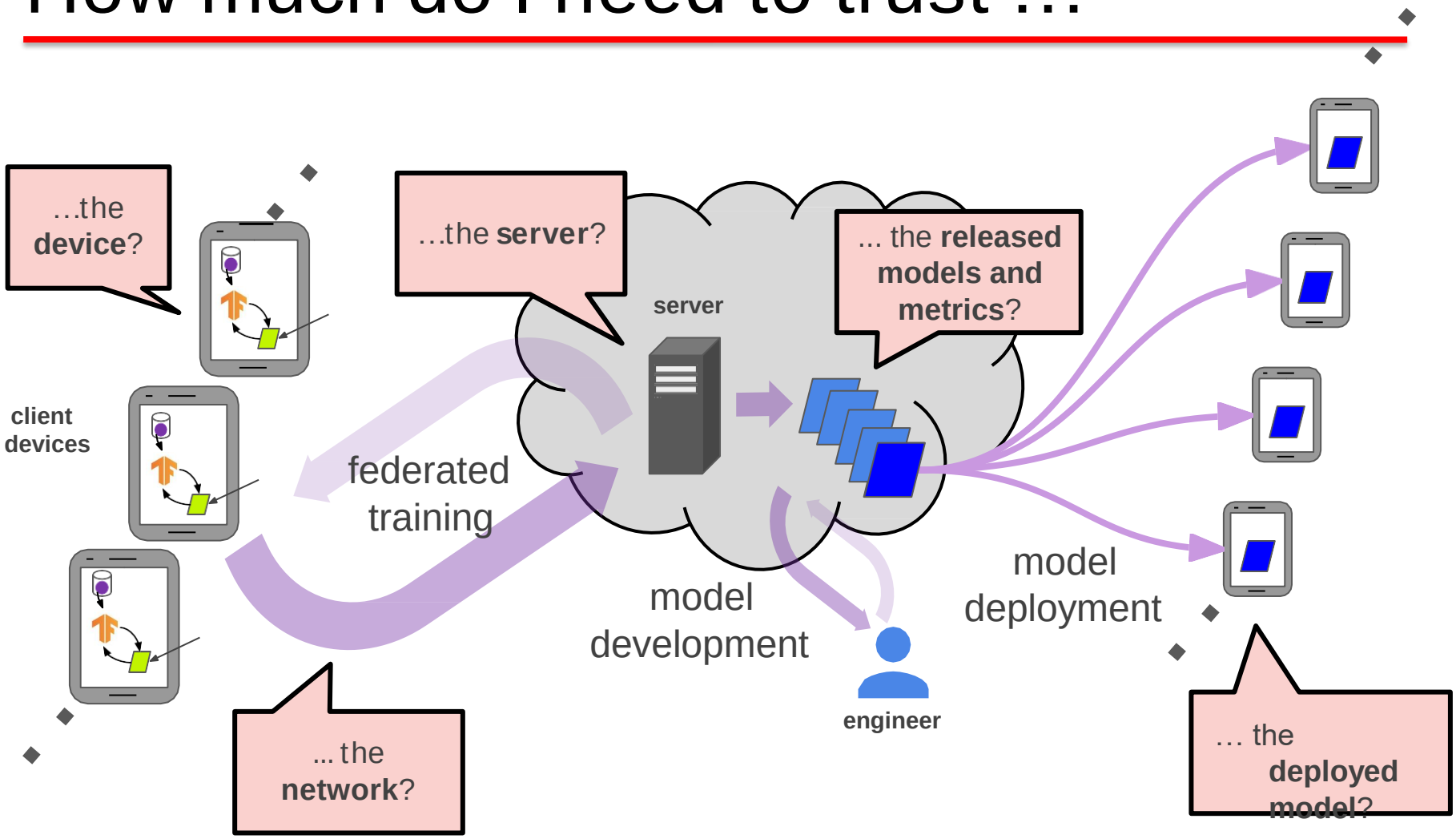


What private information might an actor learn

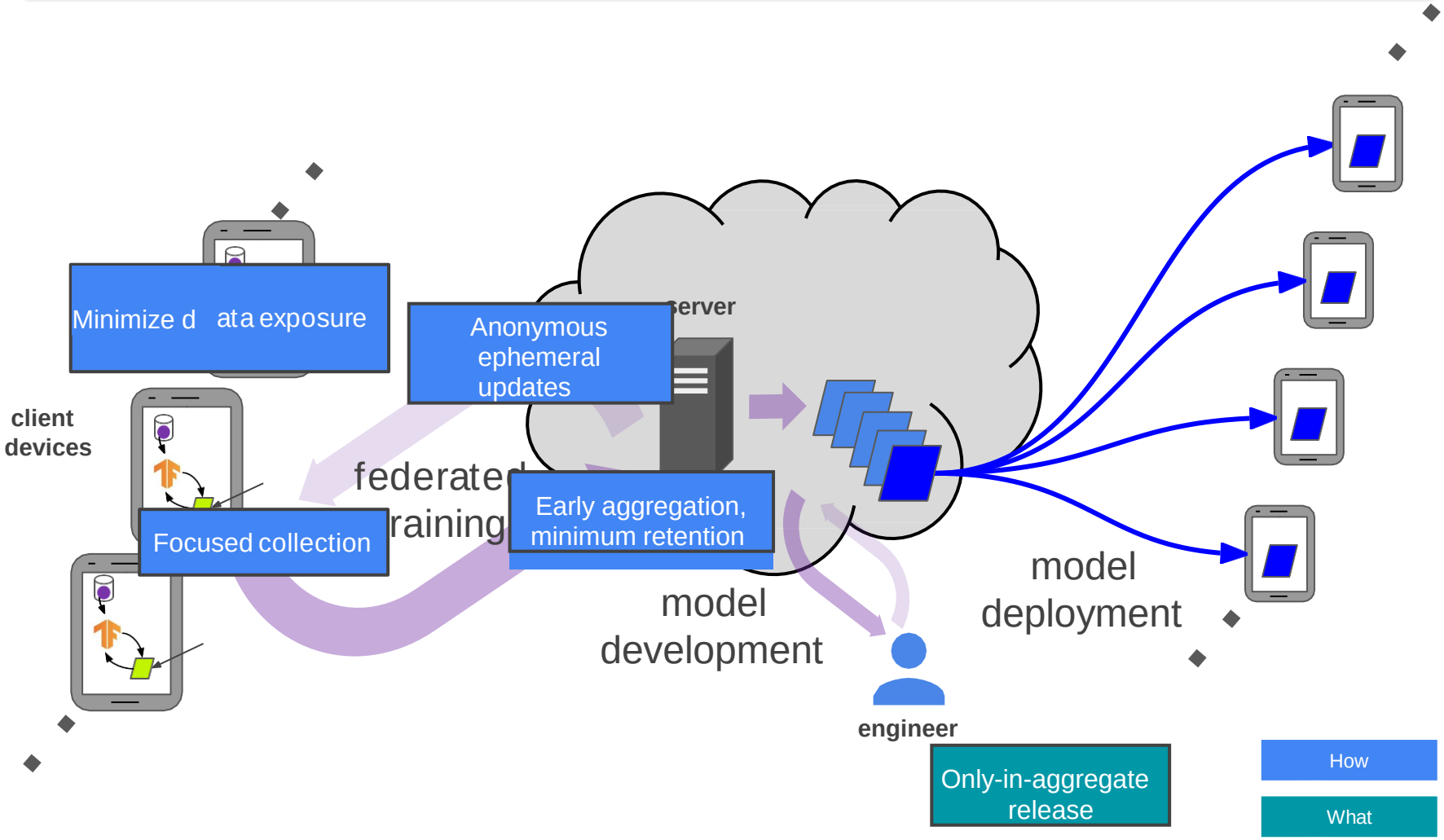
with access to ...



How much do I need to trust …

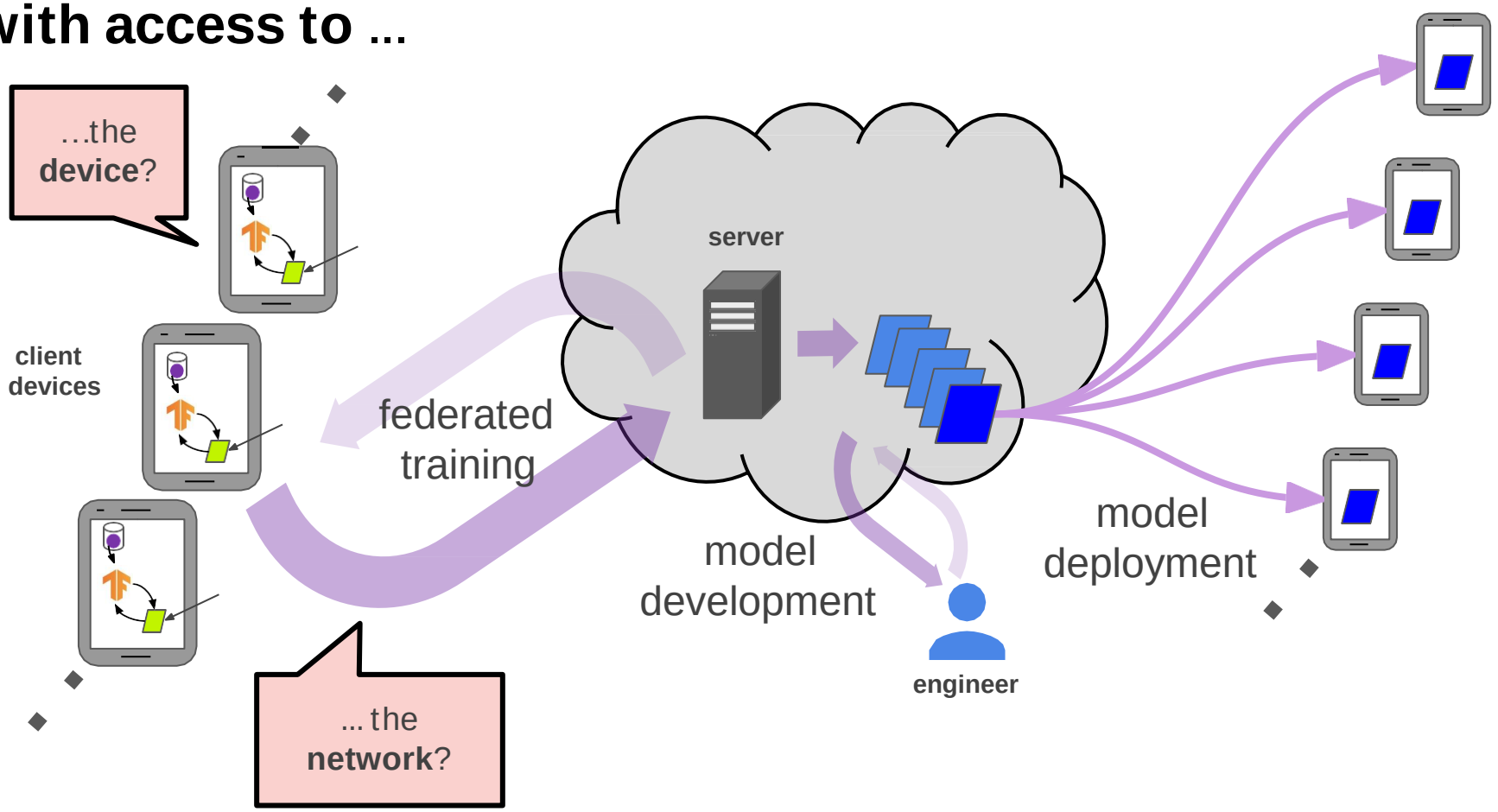


Privacy principles guiding FL

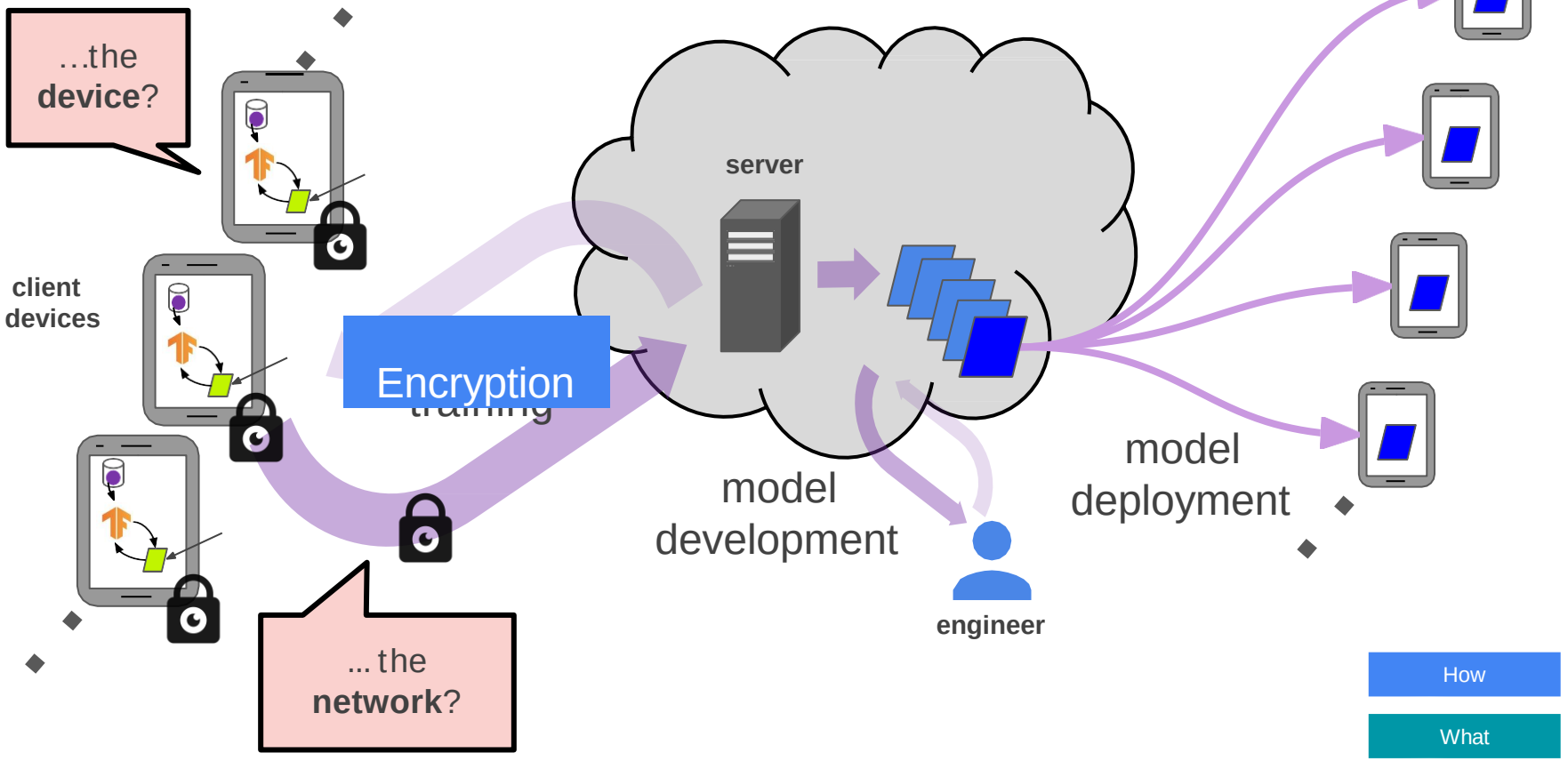


What private information might an actor learn

with access to ...

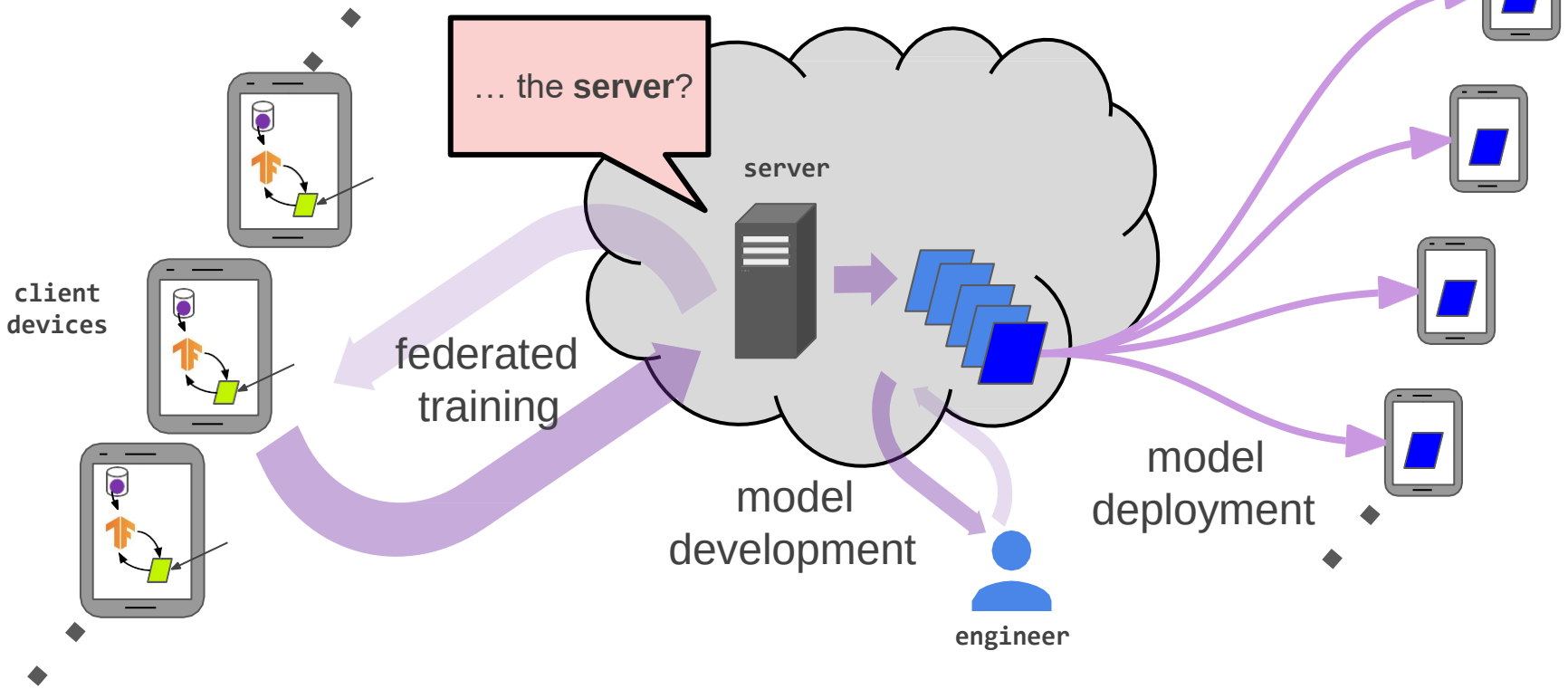


Encryption, at rest and on the wire



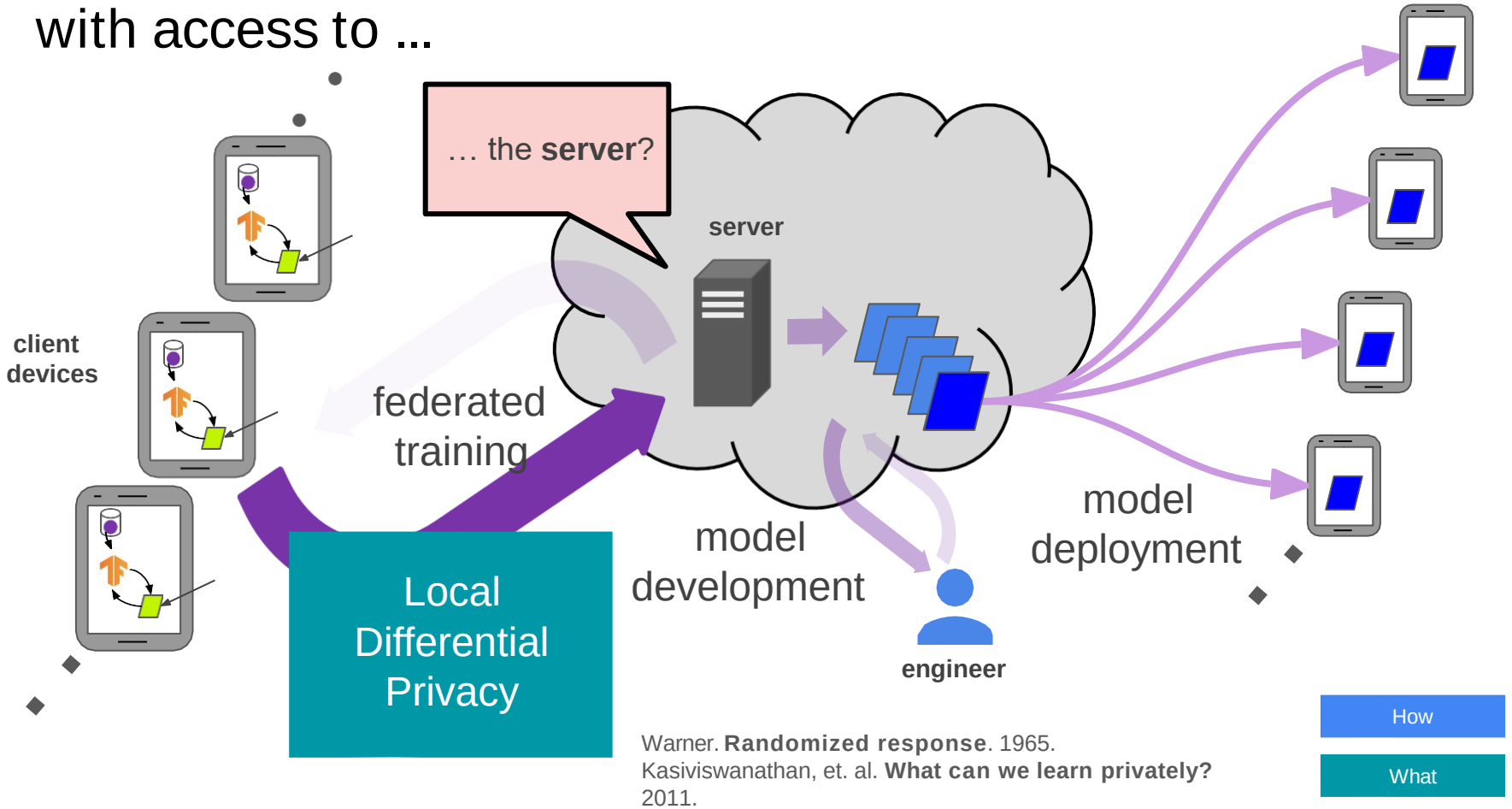
What private information might an actor learn

with access to ...



What private information might an actor learn,

with access to …



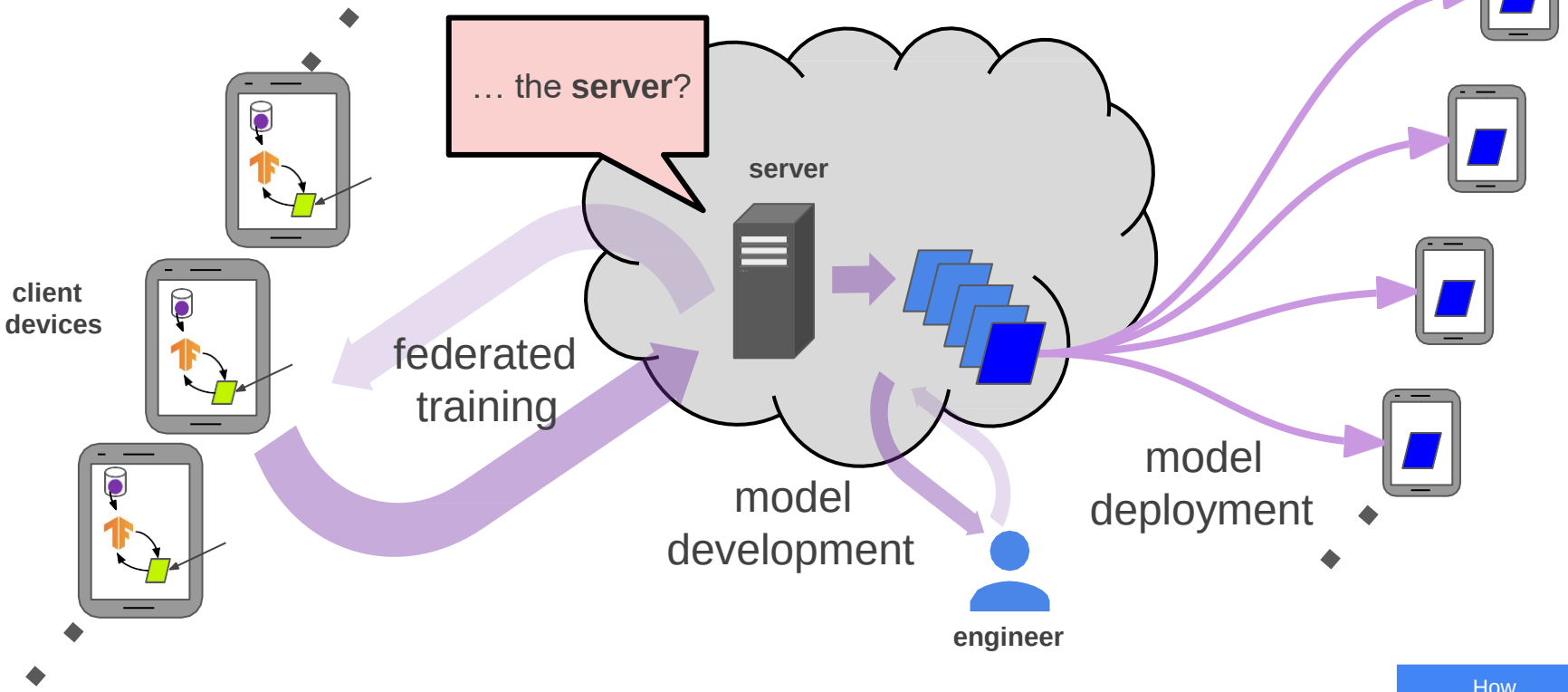
Warner. **Randomized response**. 1965.
Kasiviswanathan, et. al. **What can we learn privately?** 2011.

How

What

What private information might an actor learn

with access to ...

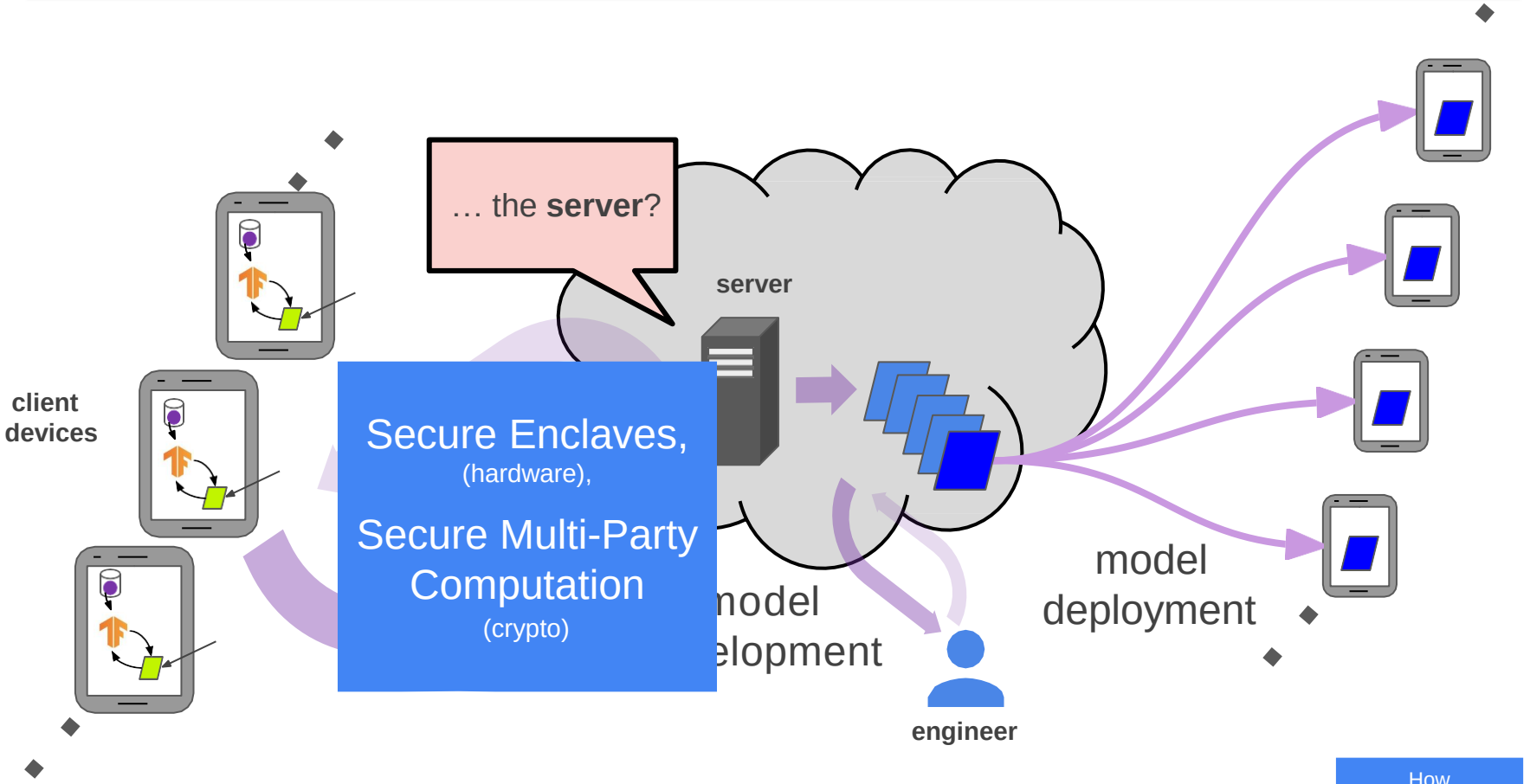


Ideally, **nothing**, even with root access.

How

What

What private information might an actor learn



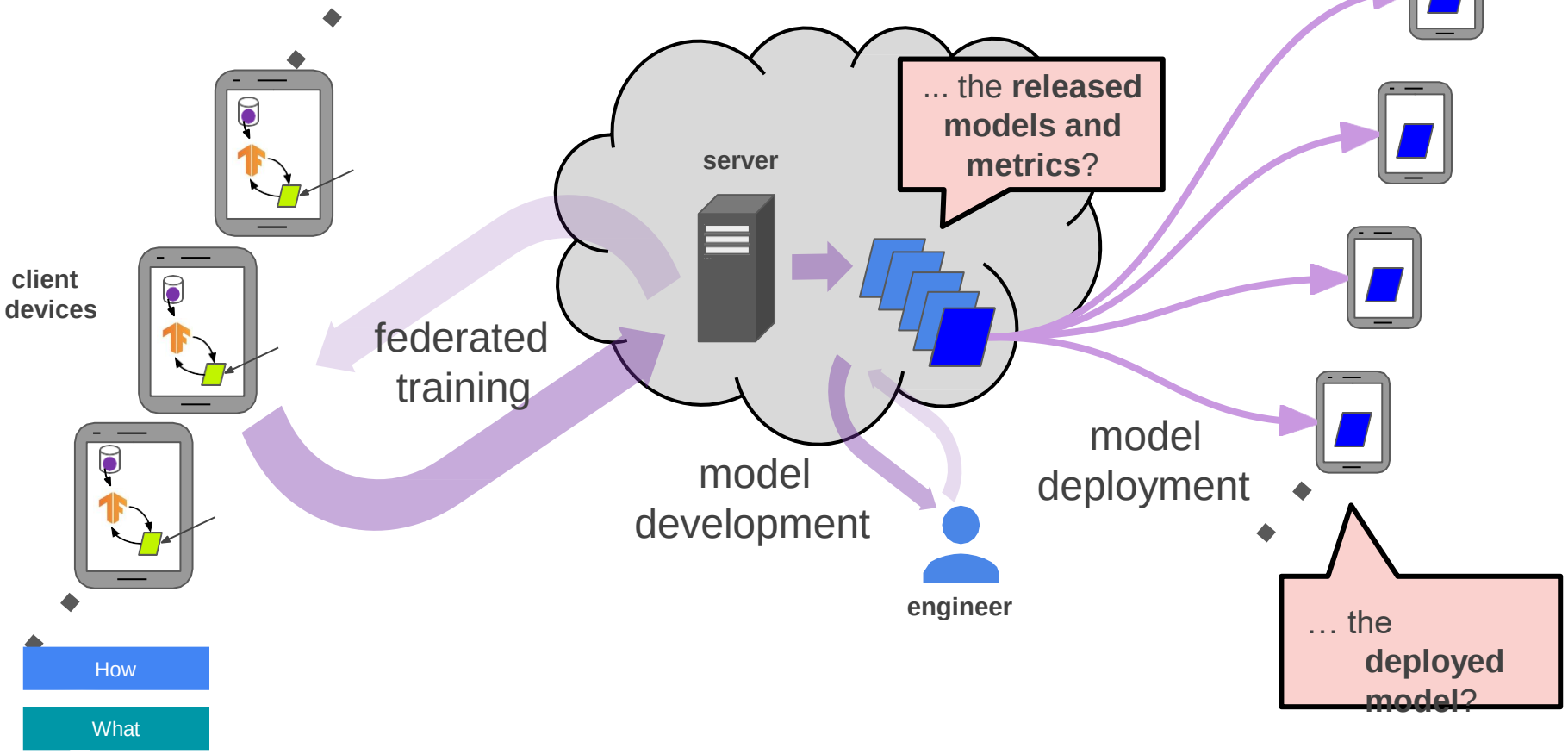
Ideally, **nothing**, even with root access.

How

What

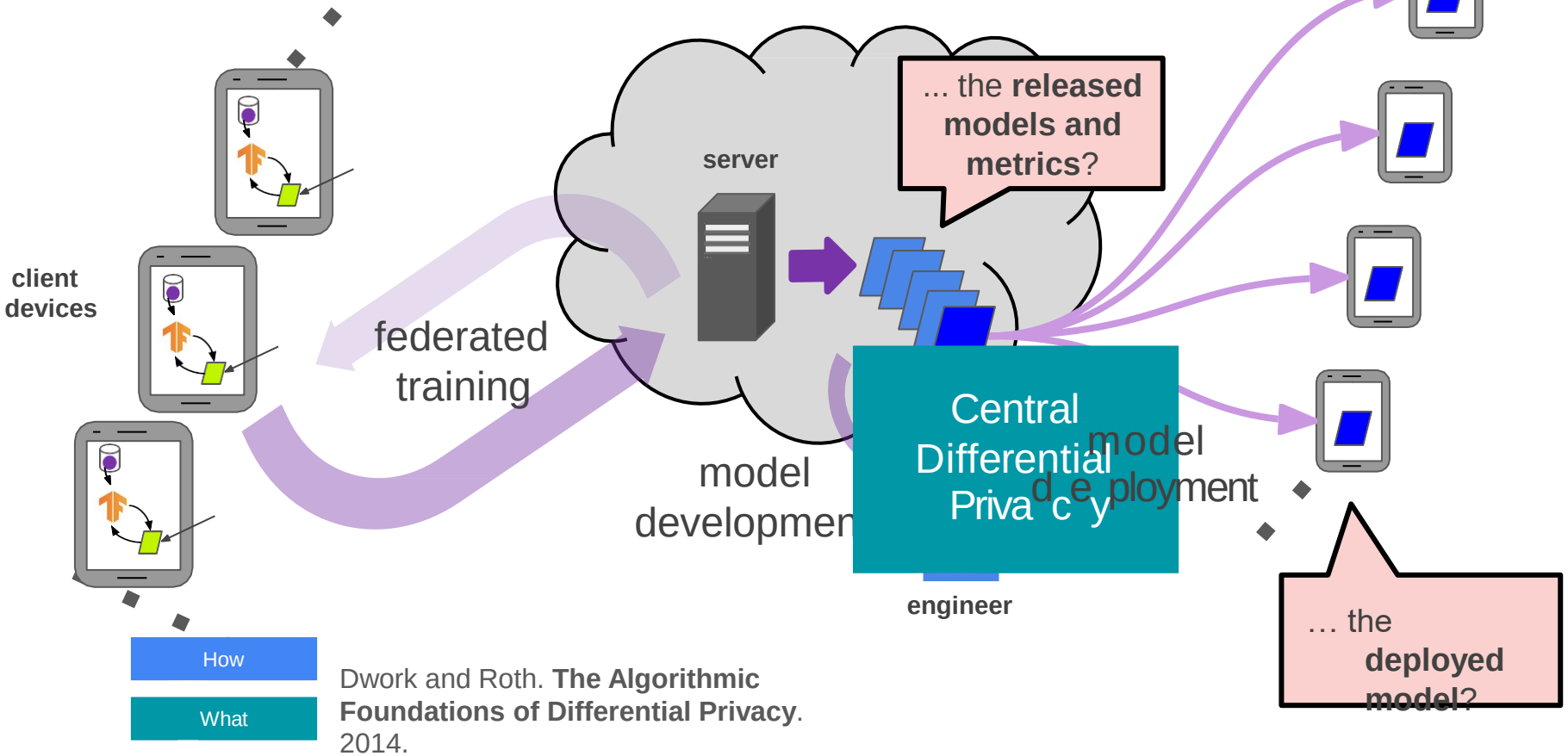
What private information might an actor learn

with access to ...



What private information might an actor learn

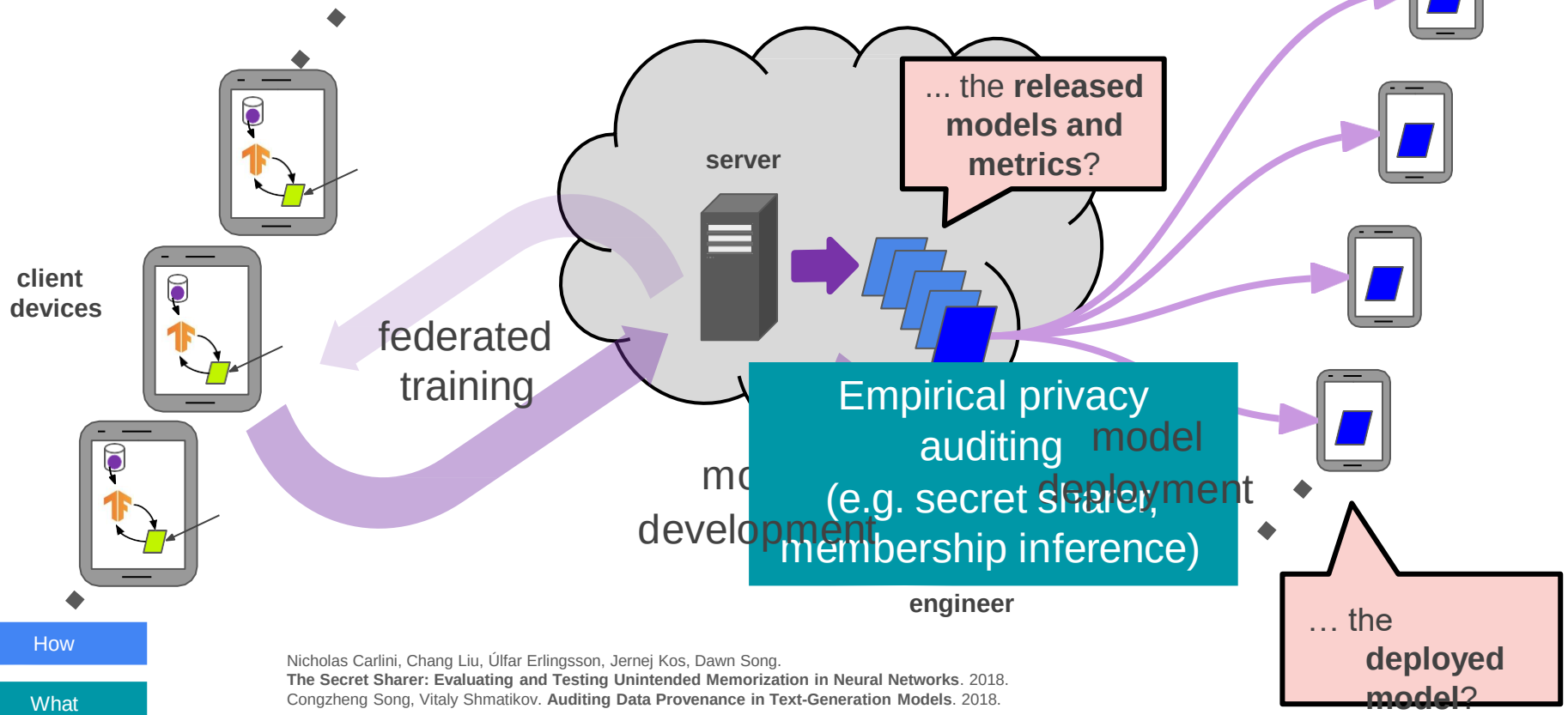
with access to ...



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

What private information might an actor learn

with access to ...

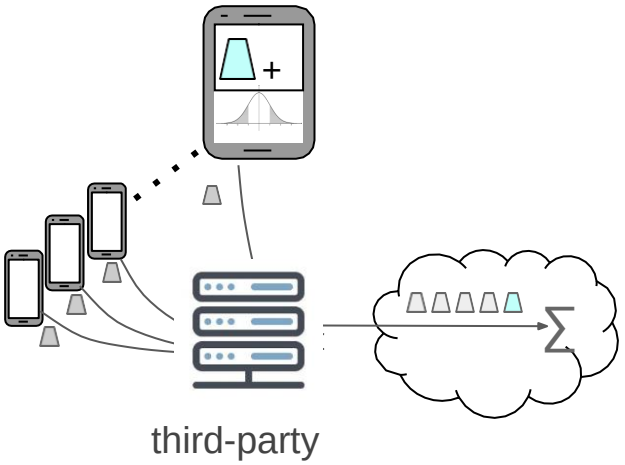


Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, Dawn Song.
The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. 2018.
Congzheng Song, Vitaly Shmatikov. **Auditing Data Provenance in Text-Generation Models**. 2018.
Matthew Jagielski, Jonathan Ullman, Alina Oprea. **Auditing Differentially Private Machine Learning: How Private is Private SGD?** 2020.

Private Aggregation & Trust

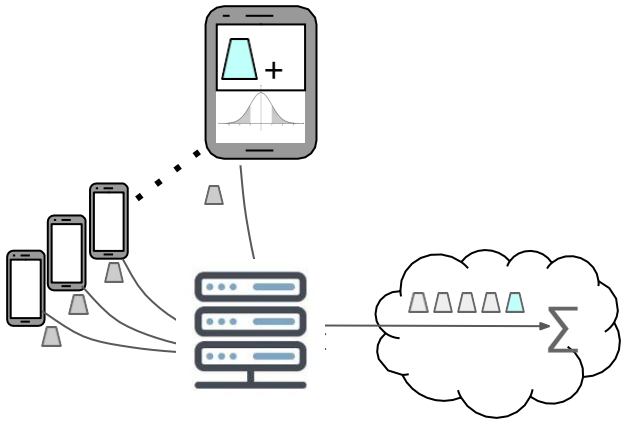
Distributing Trust for Private Aggregation

1. Trusted “third party”



Distributing Trust for Private Aggregation

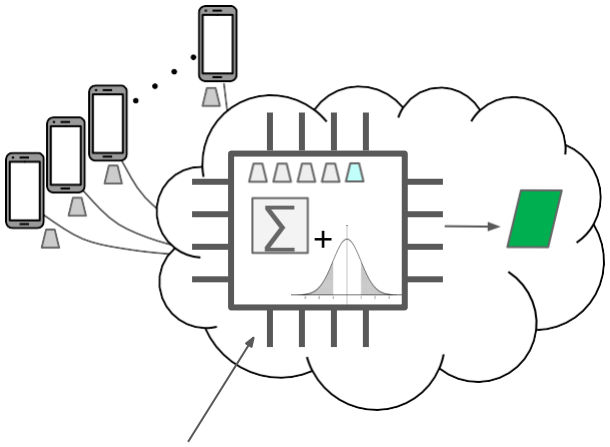
1 Trusted “third party”



third-party

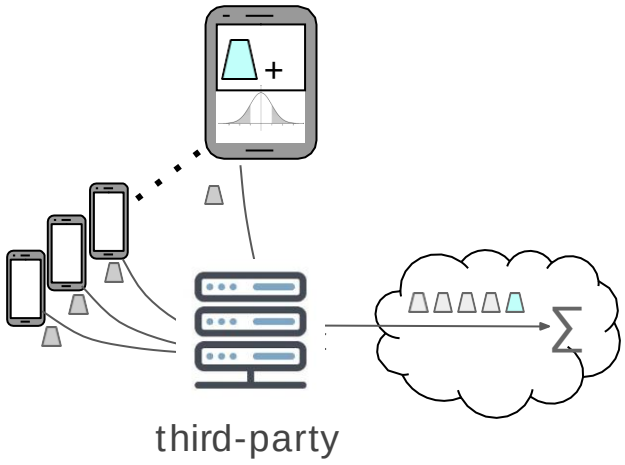
trusted hardware

2 Trusted Execution Environments

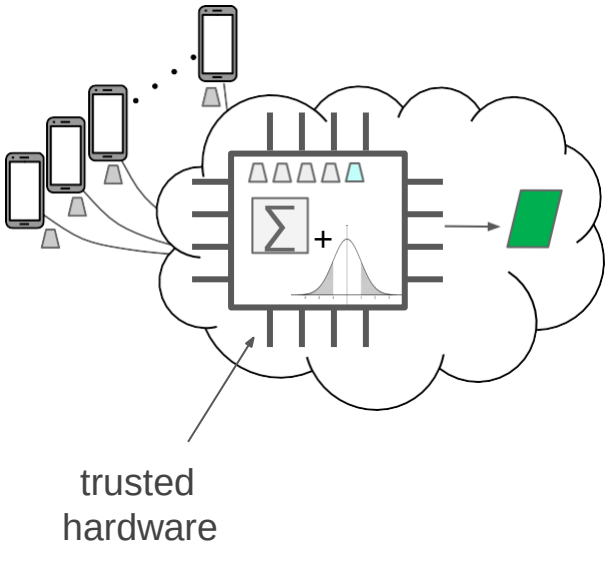


Distributing Trust for Private Aggregation

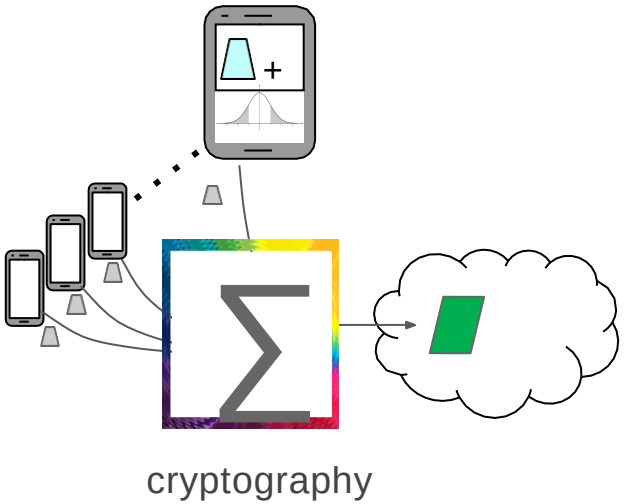
1 Trusted "third party"



2 Trusted Execution Environments



3 Trust via Cryptography



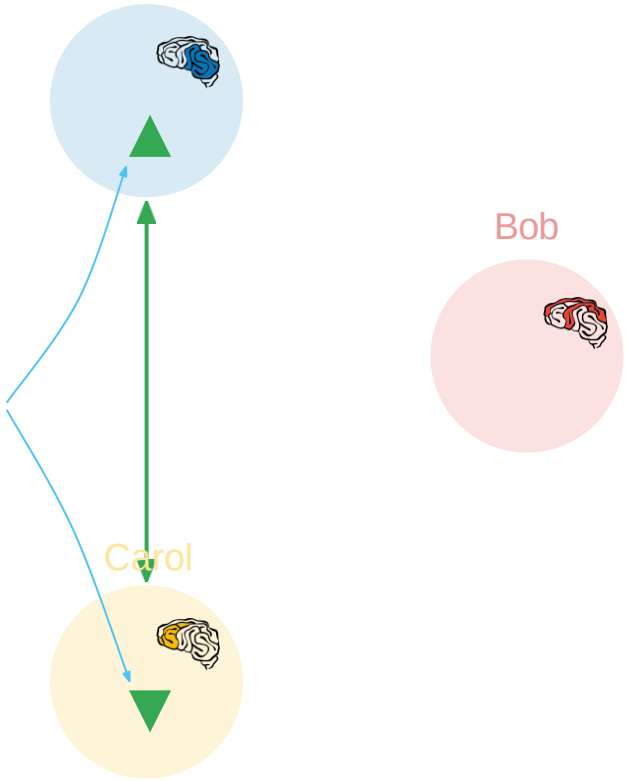
Secure Aggregation

Communication Efficient
for Vectors & Tensors

Robust
to Clients Going Offline

Random positive/negative pairs, *aka* antiparticles

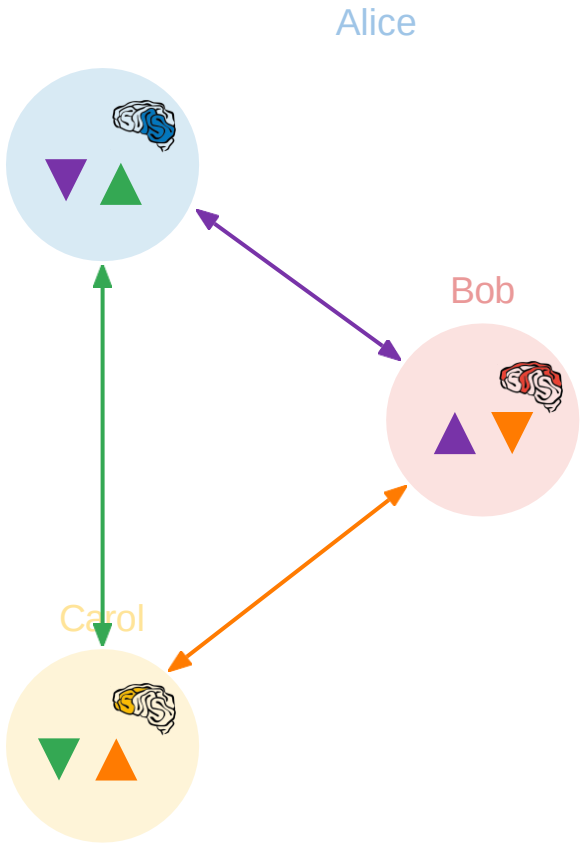
Devices cooperate to sample random pairs of 0-sum perturbations vectors.



Google

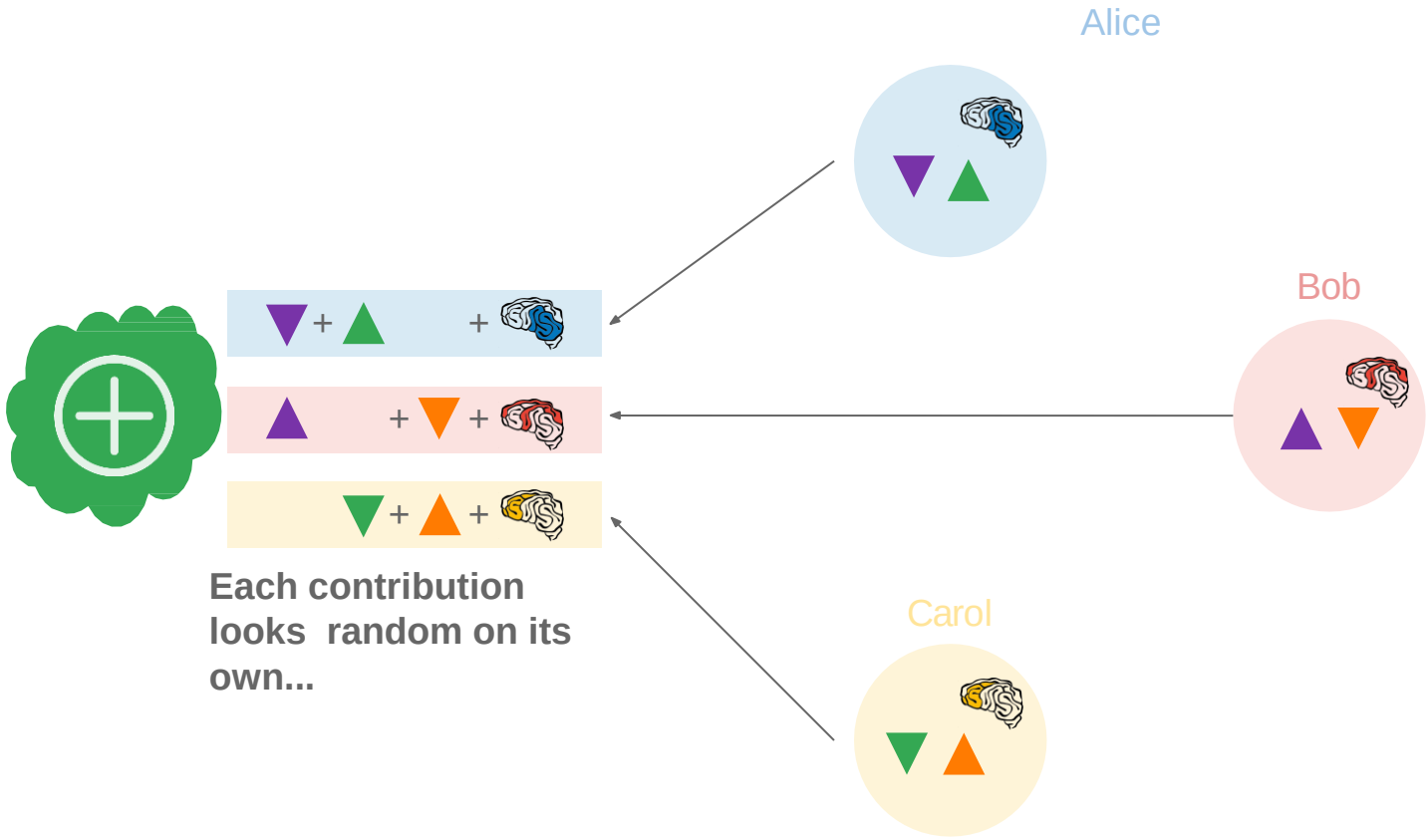
Random positive/negative pairs, *aka* antiparticles

Devices cooperate to sample random pairs of 0-sum perturbations vectors.



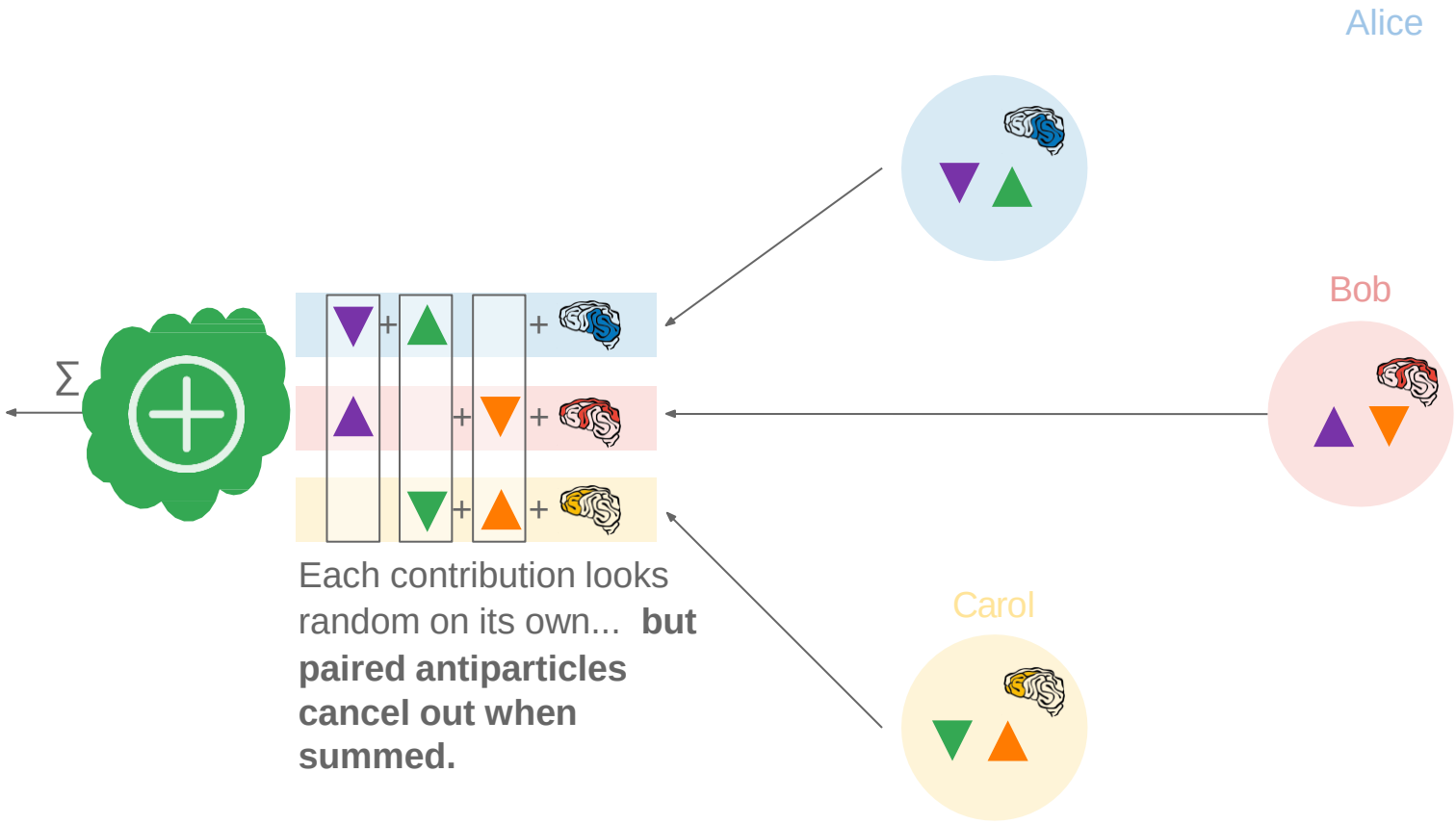
Google

Add antiparticles before sending to the server



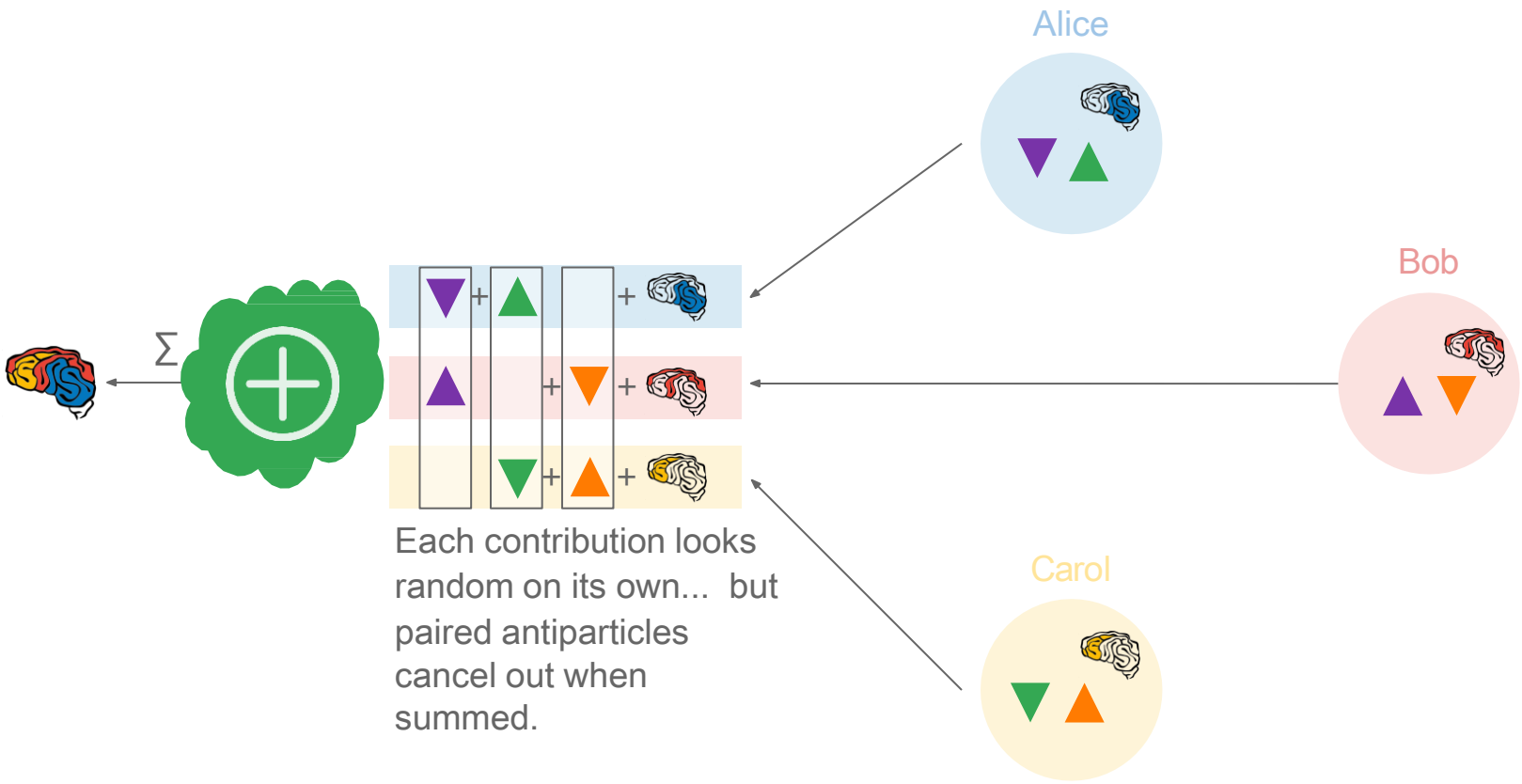
Google

The antiparticles cancel when summing contributions



Google

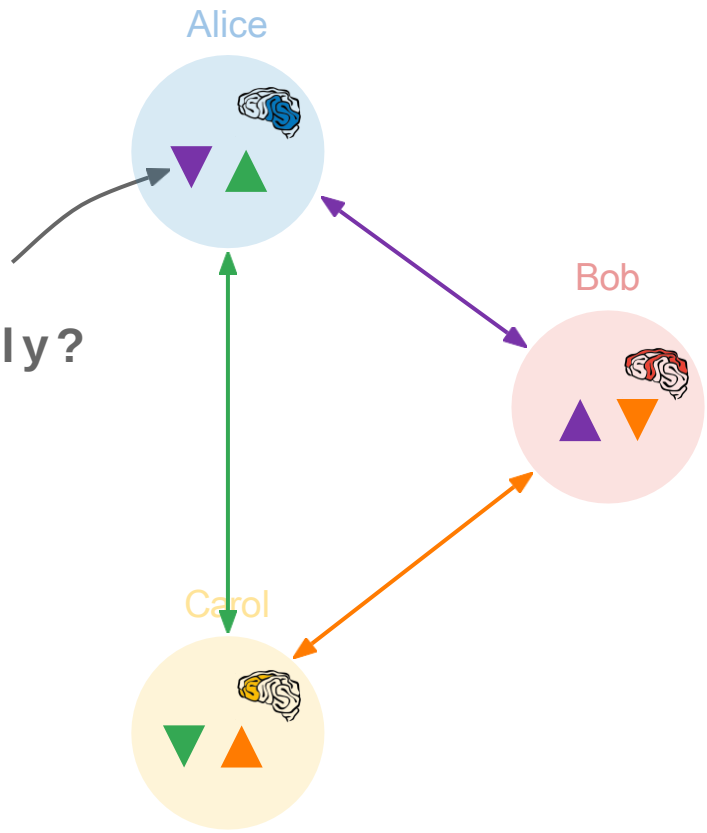
Revealing the sum.

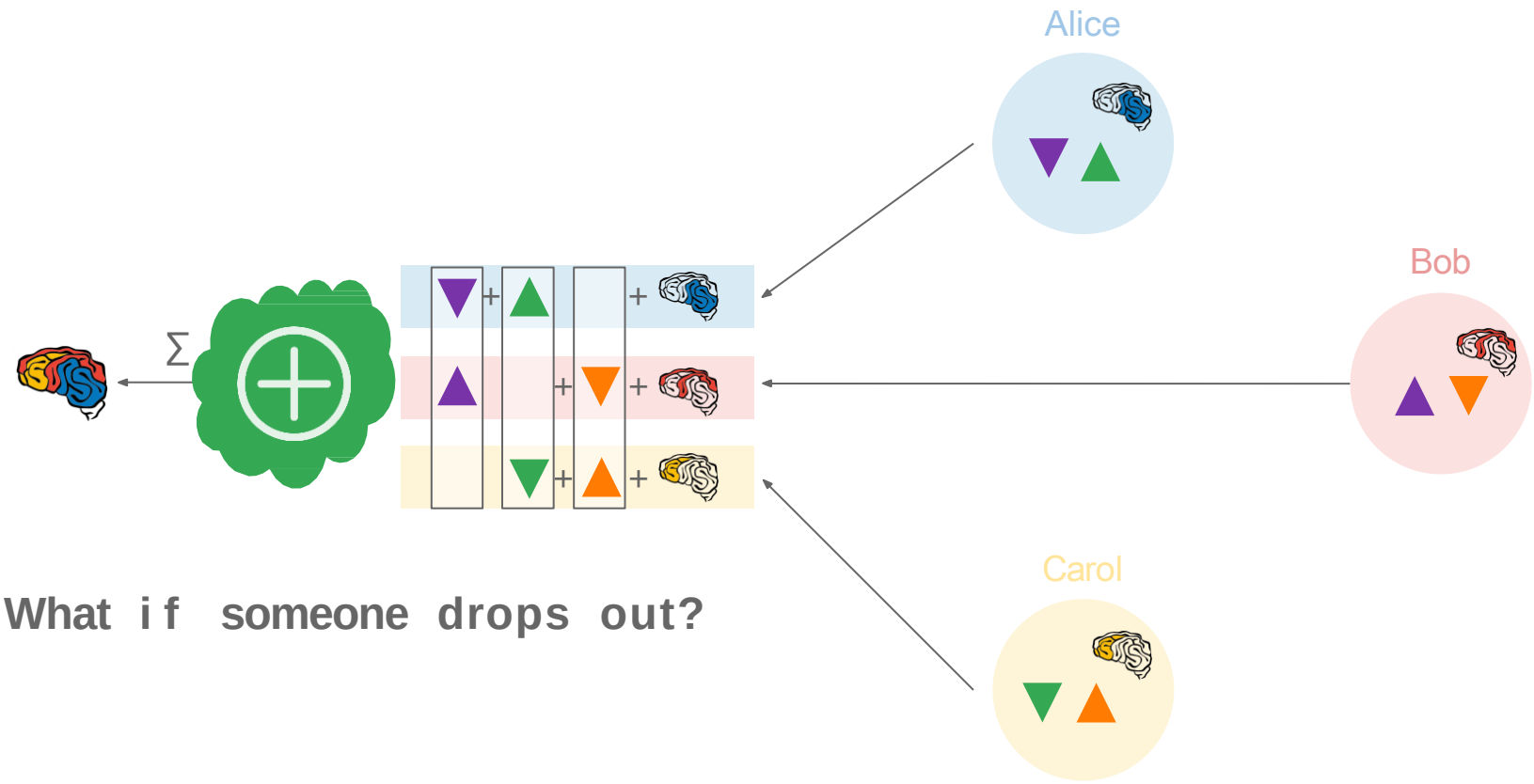


Google

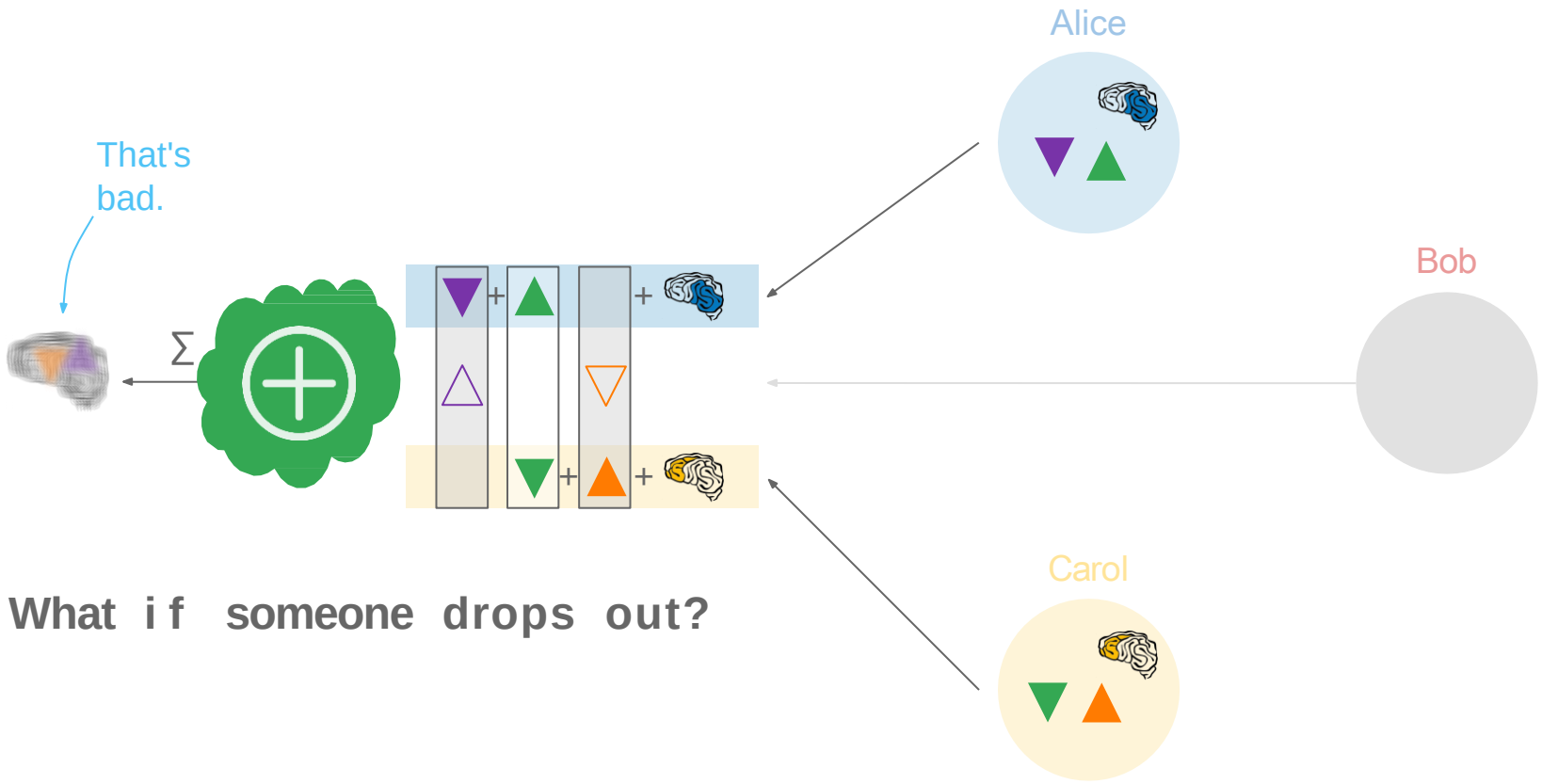
But there are two challenges...

1. These vectors are big!
   How do users agree efficiently?





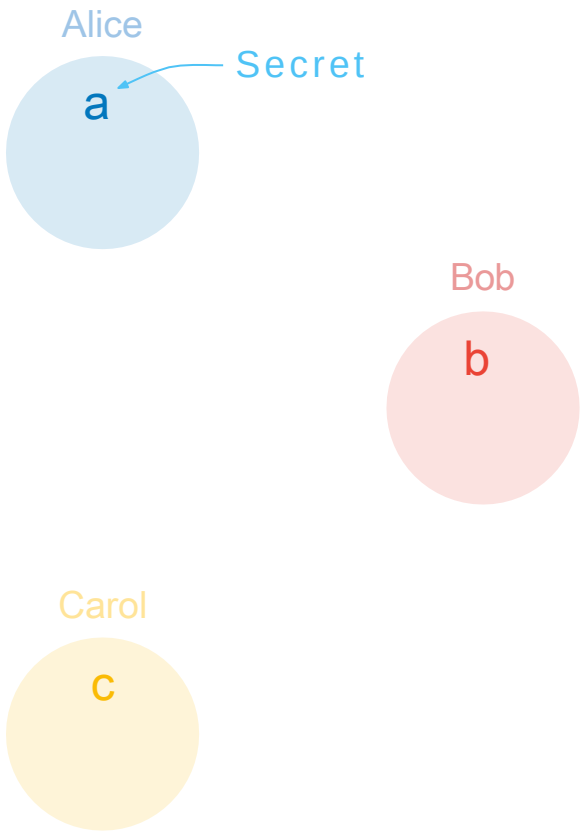
2. What if someone drops out?



2. What if someone drops out?

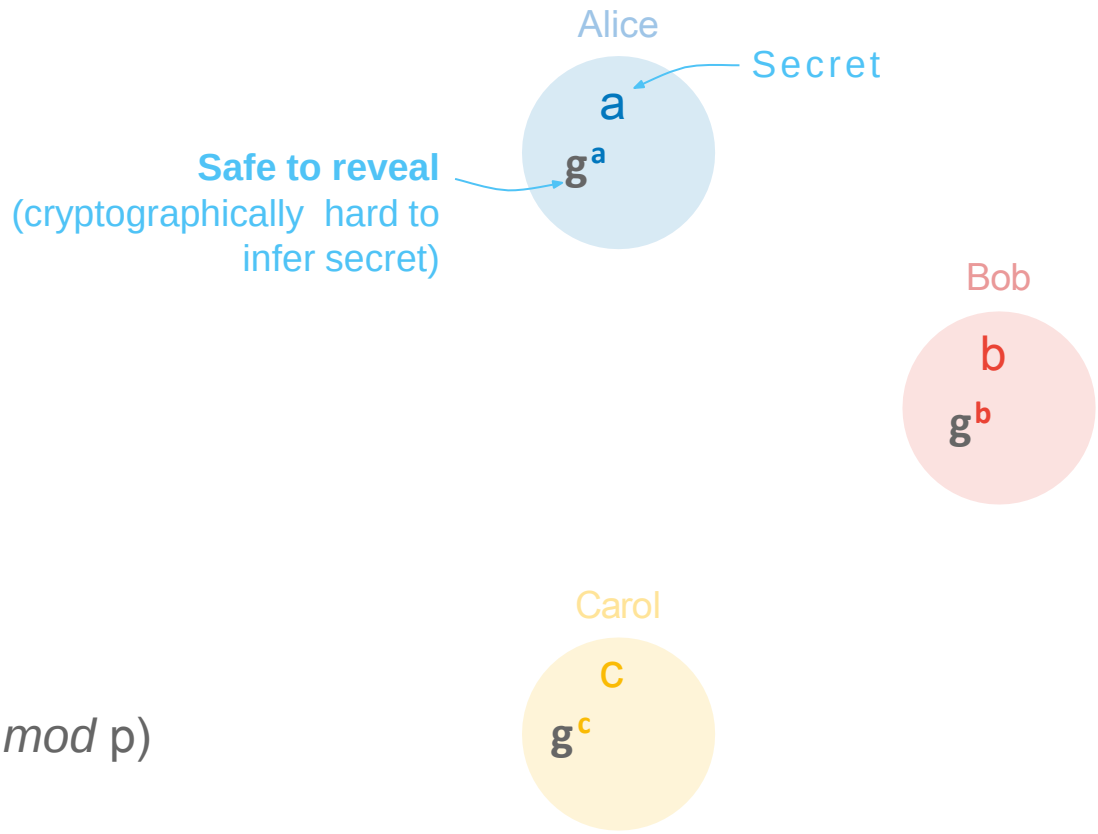
Google

Pairwise Diffie-Hellman Key Agreement



Google

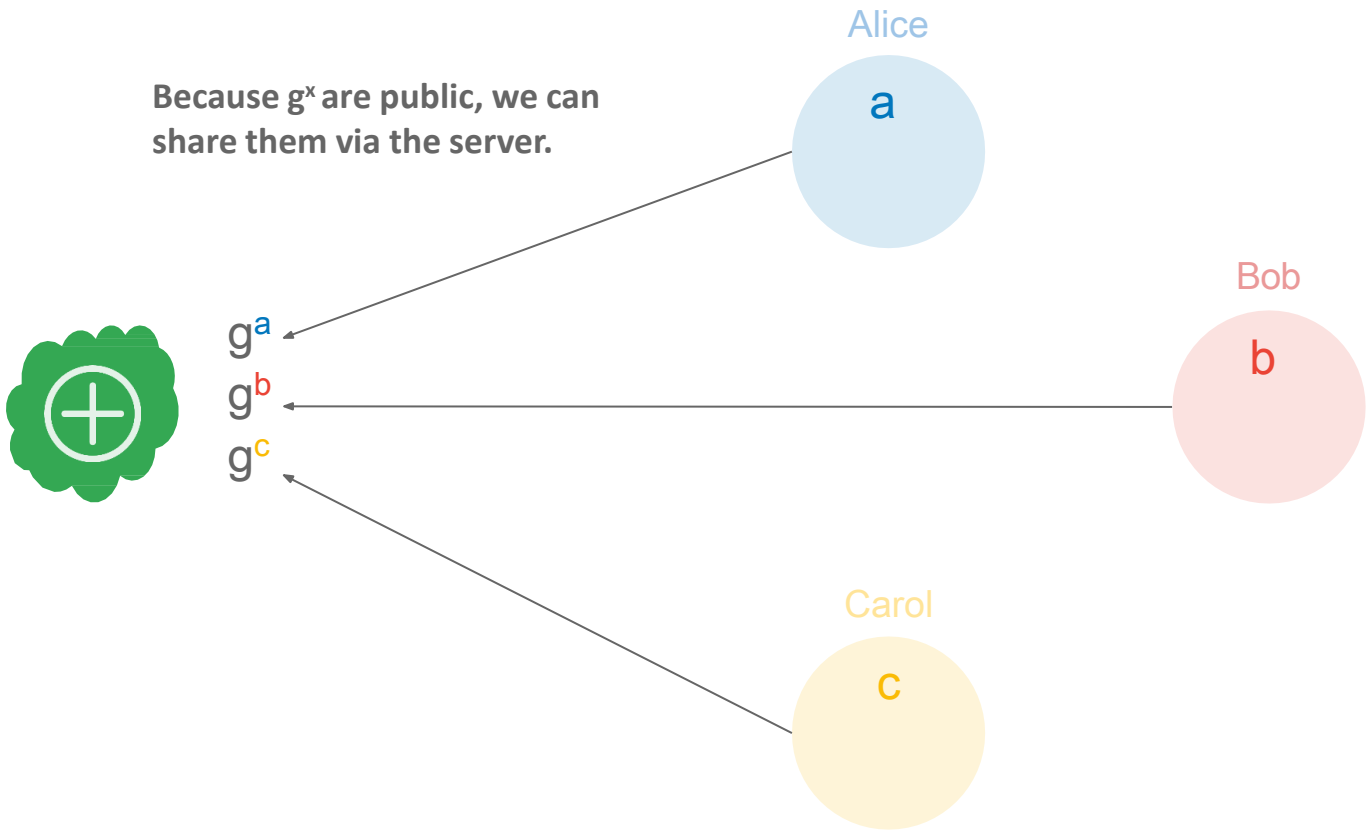
Pairwise Diffie-Hellman Key Agreement



Public parameters: g, (*mod* p)

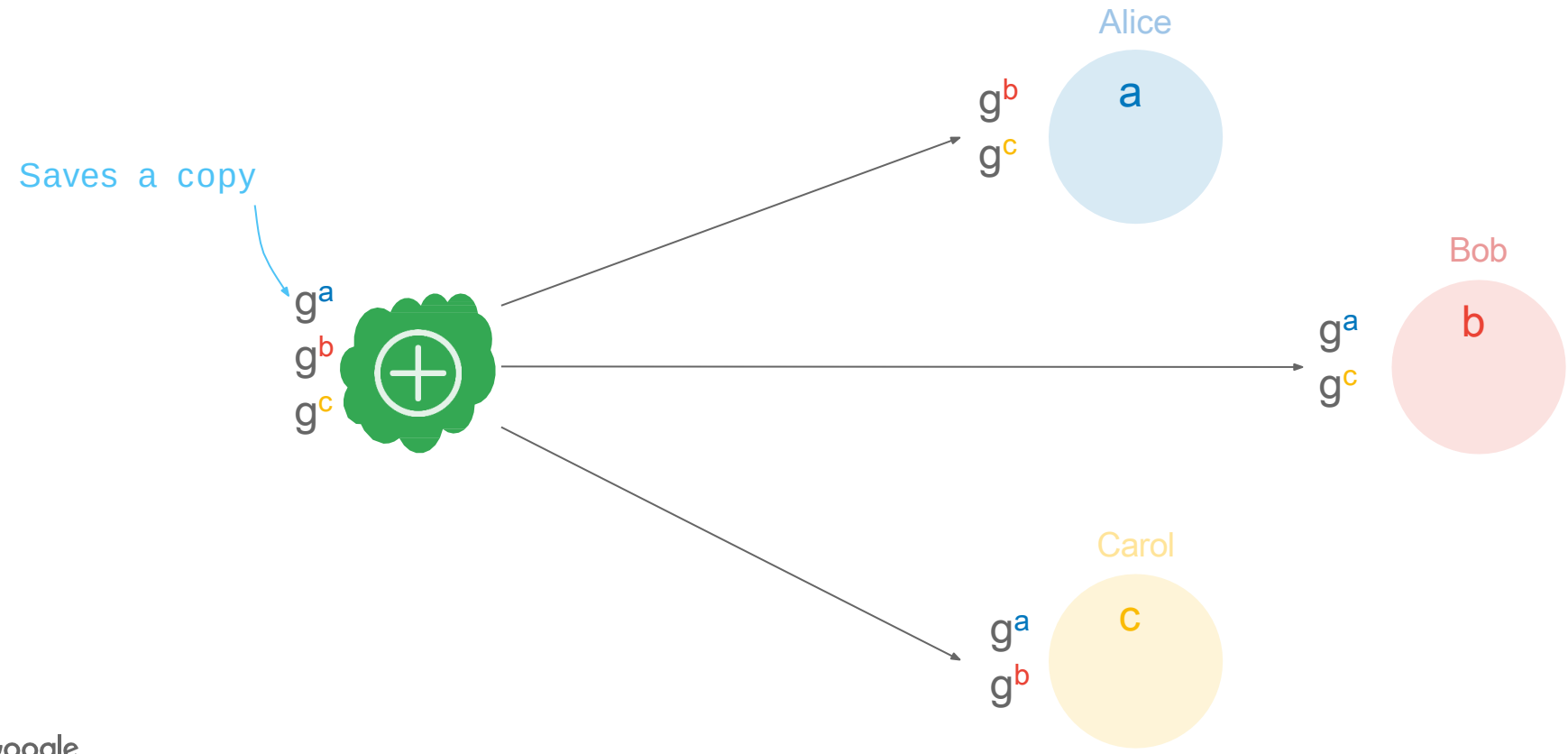
Google

Pairwise Diffie-Hellman Key Agreement



Google

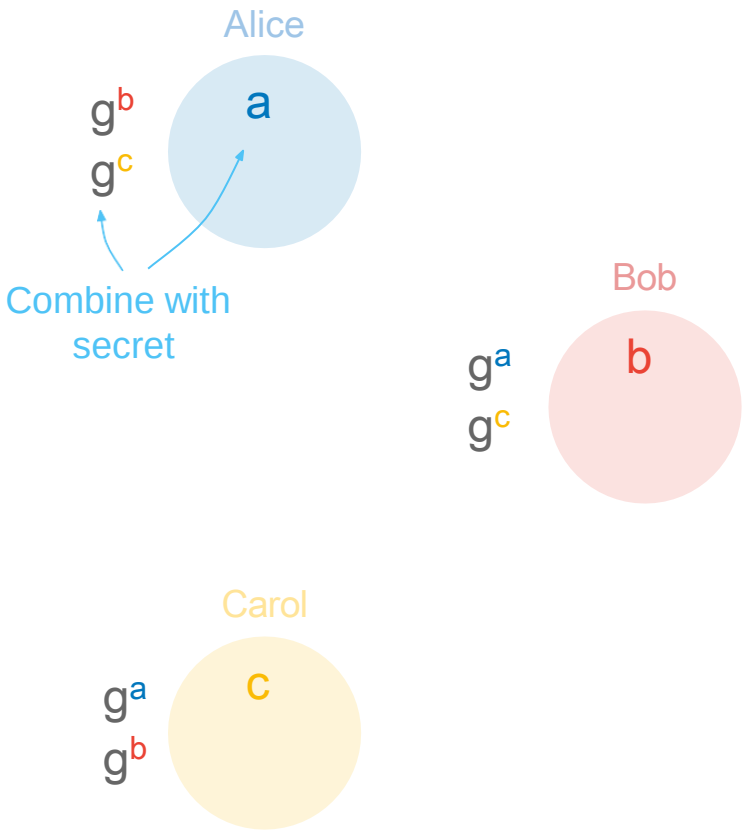
Pairwise Diffie-Hellman Key Agreement



Google

Pairwise Diffie-Hellman

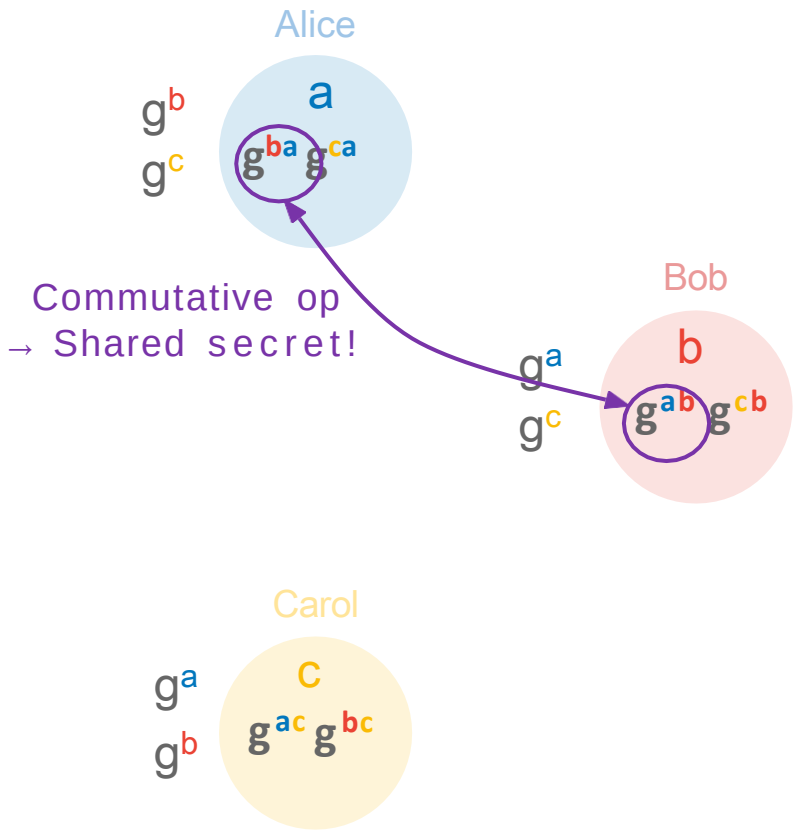
Key Agreement



Google

Pairwise Diffie-Hellman

Key Agreement

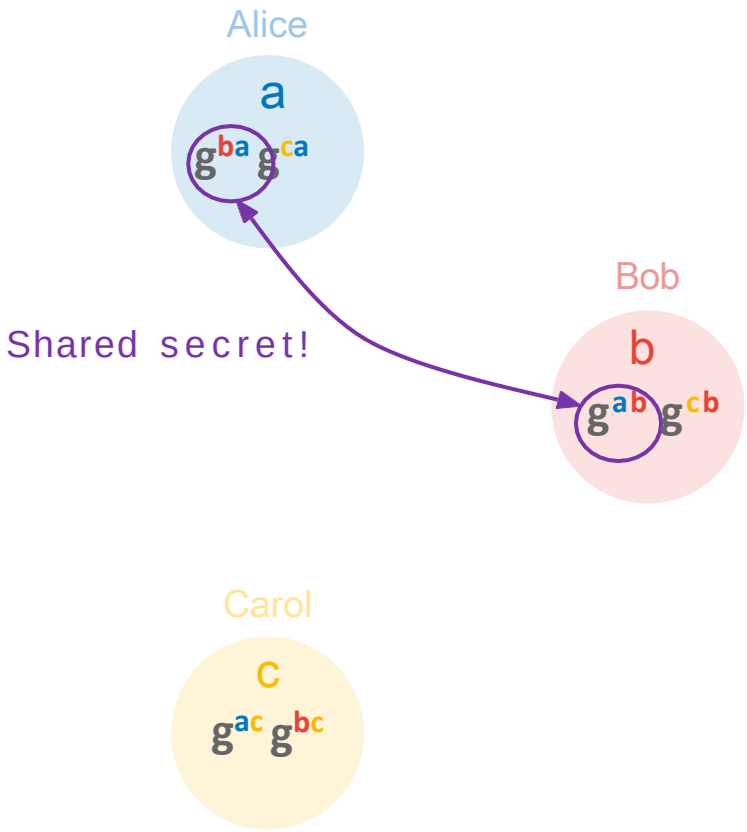


Google

Pairwise Diffie-Hellman

Key Agreement

Secrets are scalars, but….



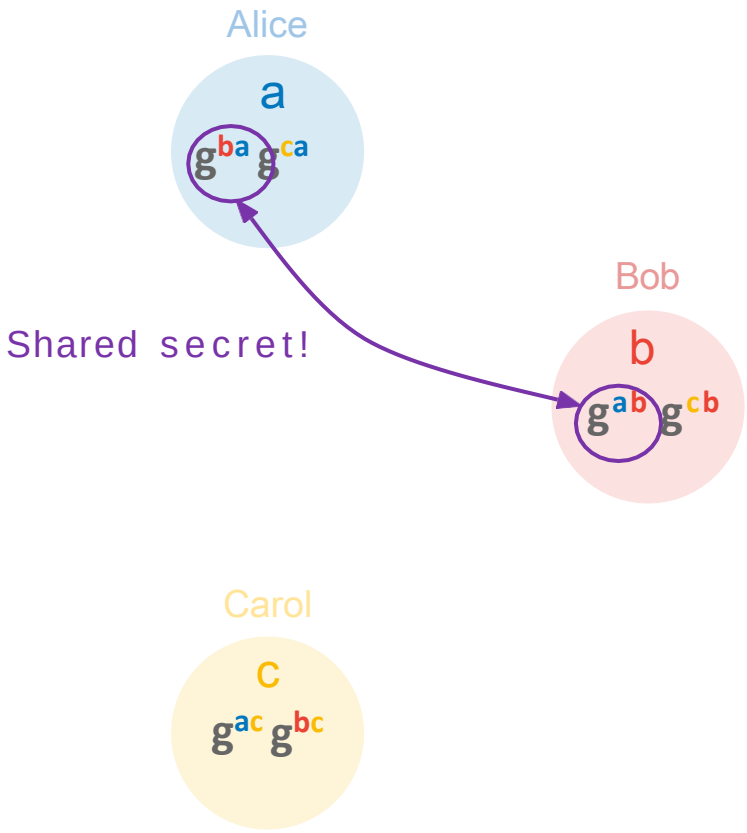
Google

Pairwise Diffie-Hellman Key Agreement + PRNG Expansion

Secrets are scalars, but….

Use each secret to seed a **pseudorandom number generator**, generate paired antiparticle vectors.

PRNG(g^{ba}) →  = -



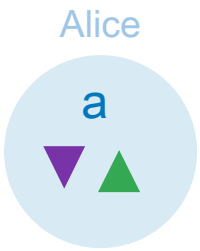
Google

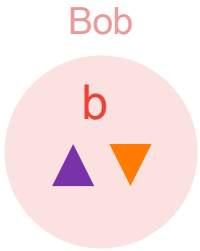
Pairwise Diffie-Hellman Key Agreement + PRNG Expansion

Secrets are scalars, but….

Use each secret to seed a pseudorandom number generator, generate paired antiparticle vectors.

PRNG(g^{ba}) →  = -





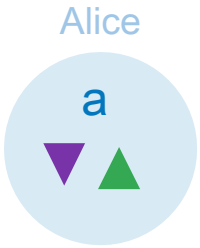
Google

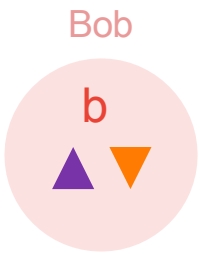
Pairwise Diffie-Hellman Key Agreement + PRNG Expansion

Secrets are scalars, but….

Use each secret to seed a pseudorandom number generator, generate paired antiparticle vectors.

PRNG(g^{ba}) →  = -





1. **Efficiency via pseudorandom generator**
2. **Mobile phones typically don't support peer-to-peer communication anyhow.**
3. **Fewer secrets = easier recovery.**

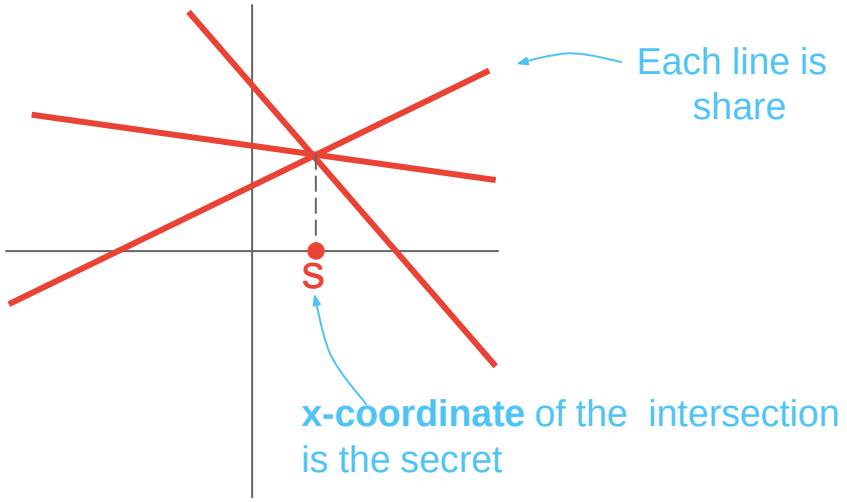
Google

k-out-of-*n* Threshold Secret Sharing

Goal: Break a secret into *n* pieces, called shares.

- **<k shares:** learn nothing
- **≥k shares**: recover *s* perfectly.

2-out-of-3 secret sharing:

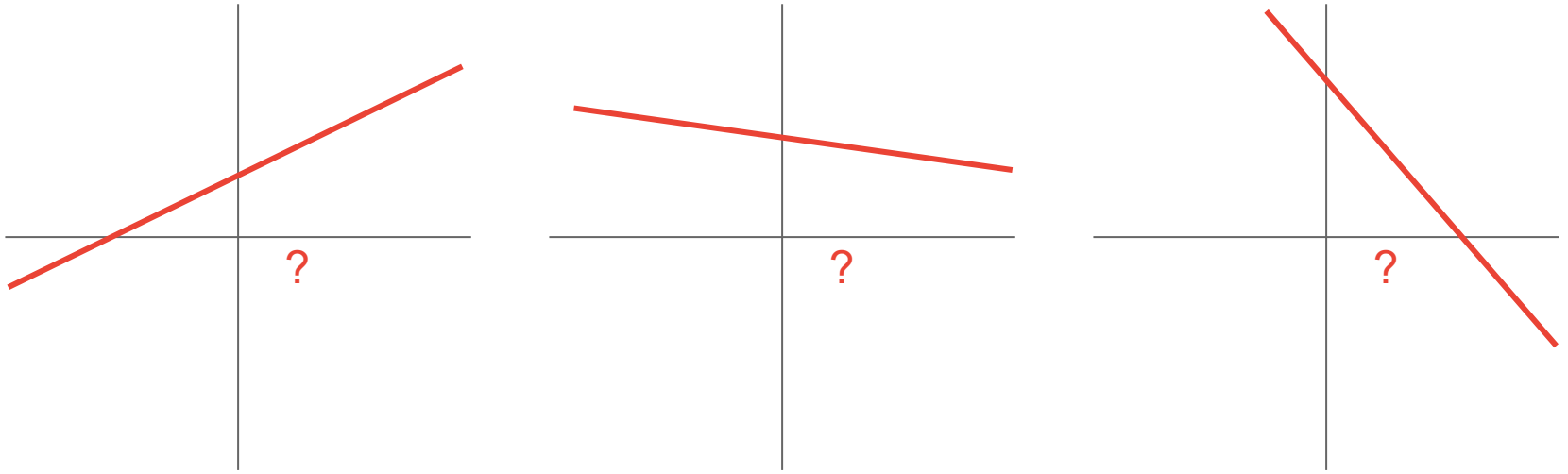


Google

k-out-of-*n* Threshold Secret Sharing

Goal: Break a secret into *n* pieces, called shares.

- **<k shares:** learn nothing
- **≥k shares**: recover *s* perfectly

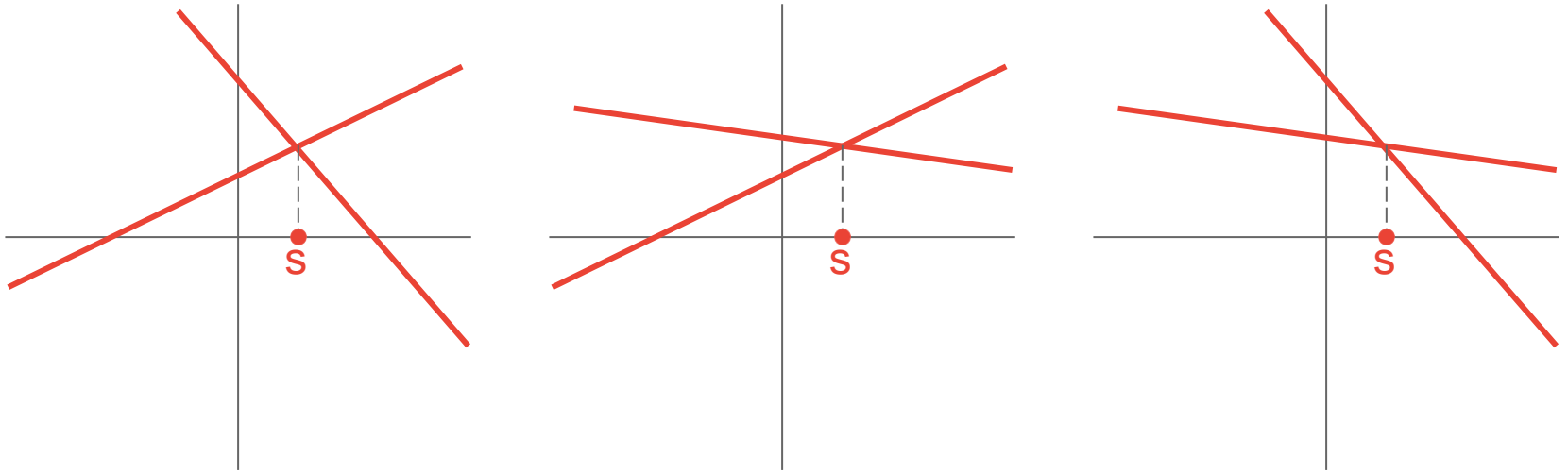


Google

k-out-of-*n* Threshold Secret Sharing

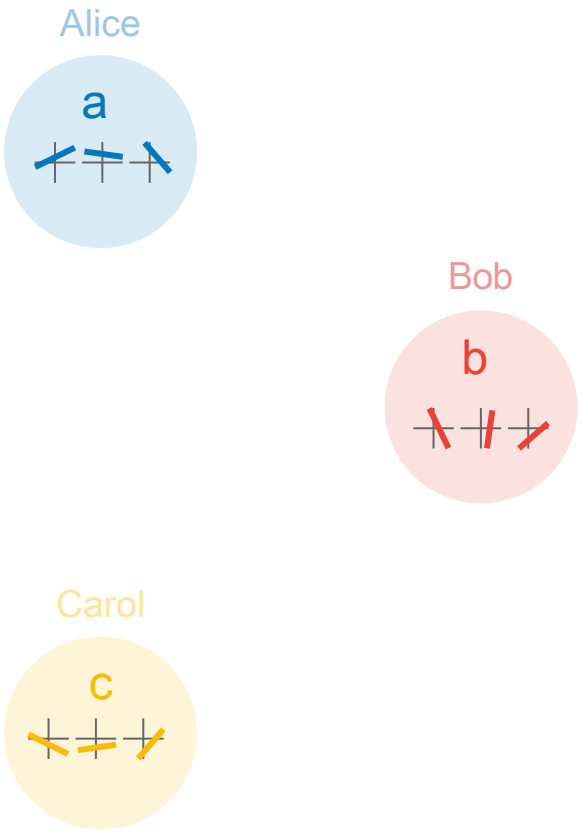
Goal: Break a secret into *n* pieces, called shares.

- **<k shares:** learn nothing
- **≥k shares**: recover *s* perfectly



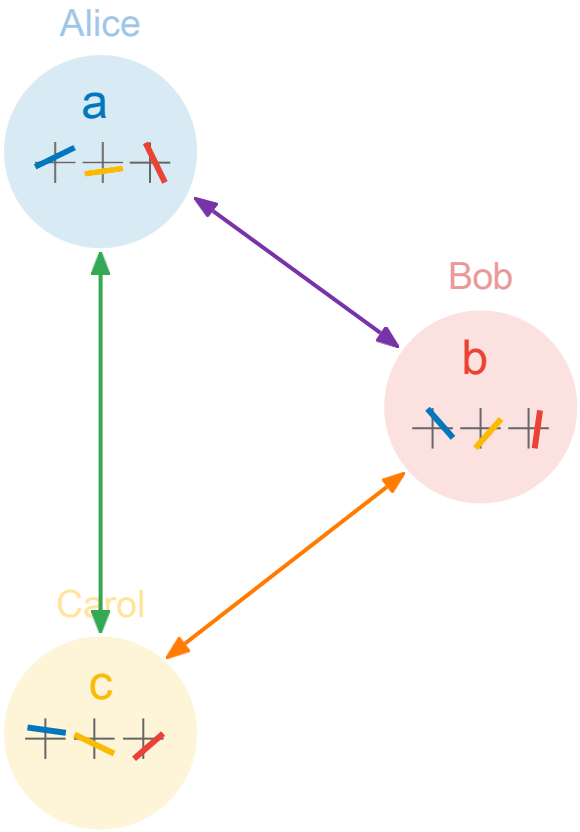
Google

Users make shares of their secrets

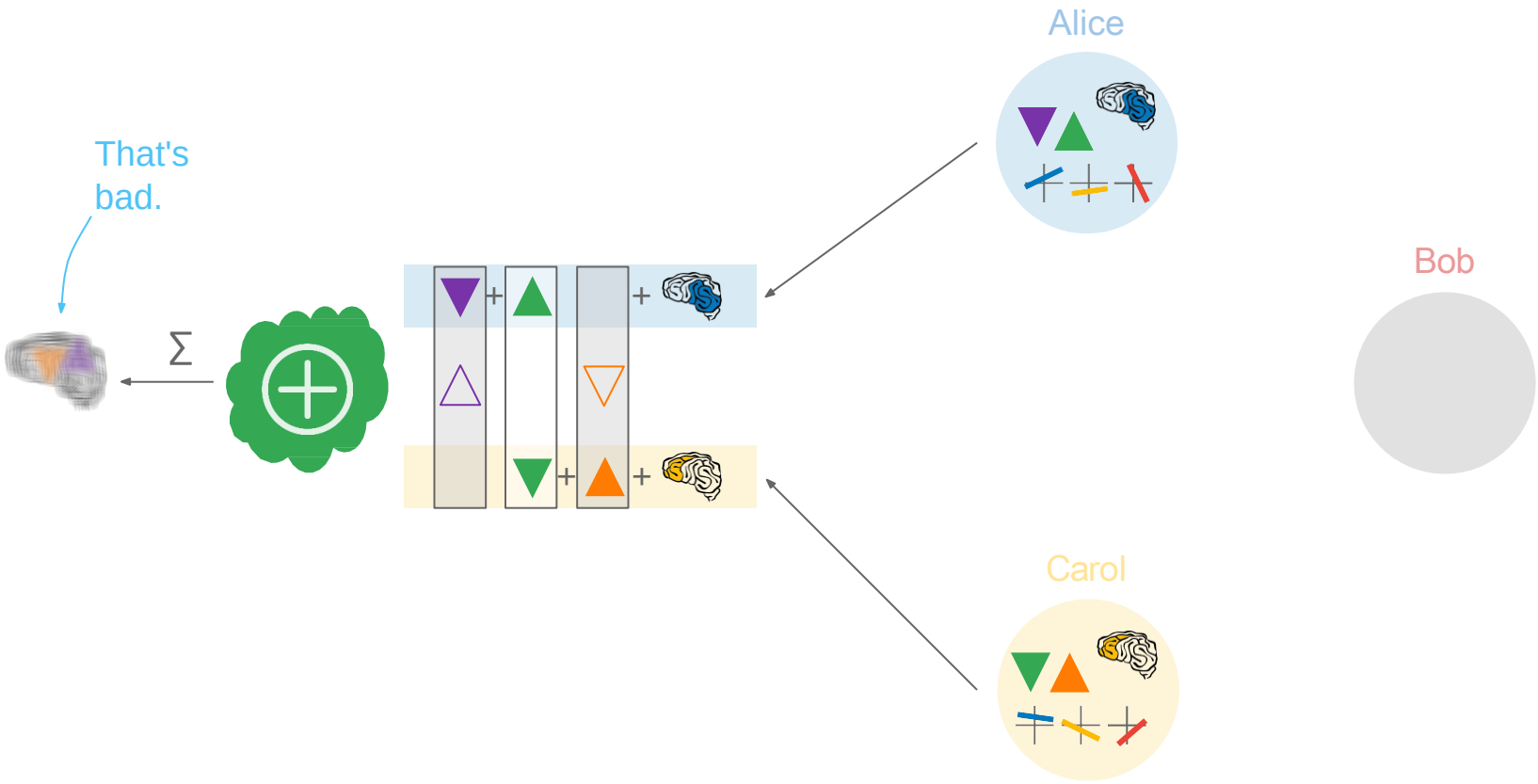


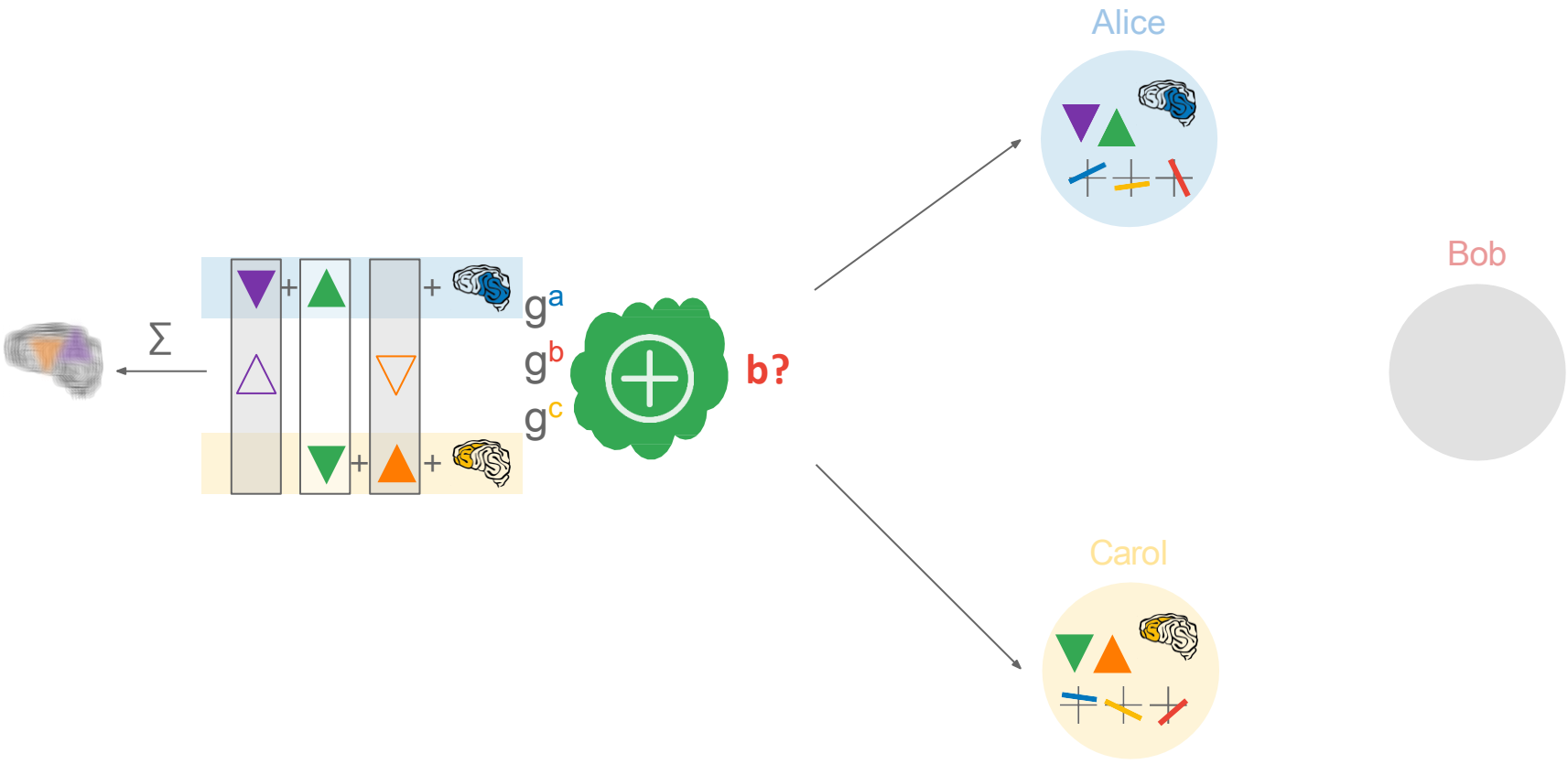
Google

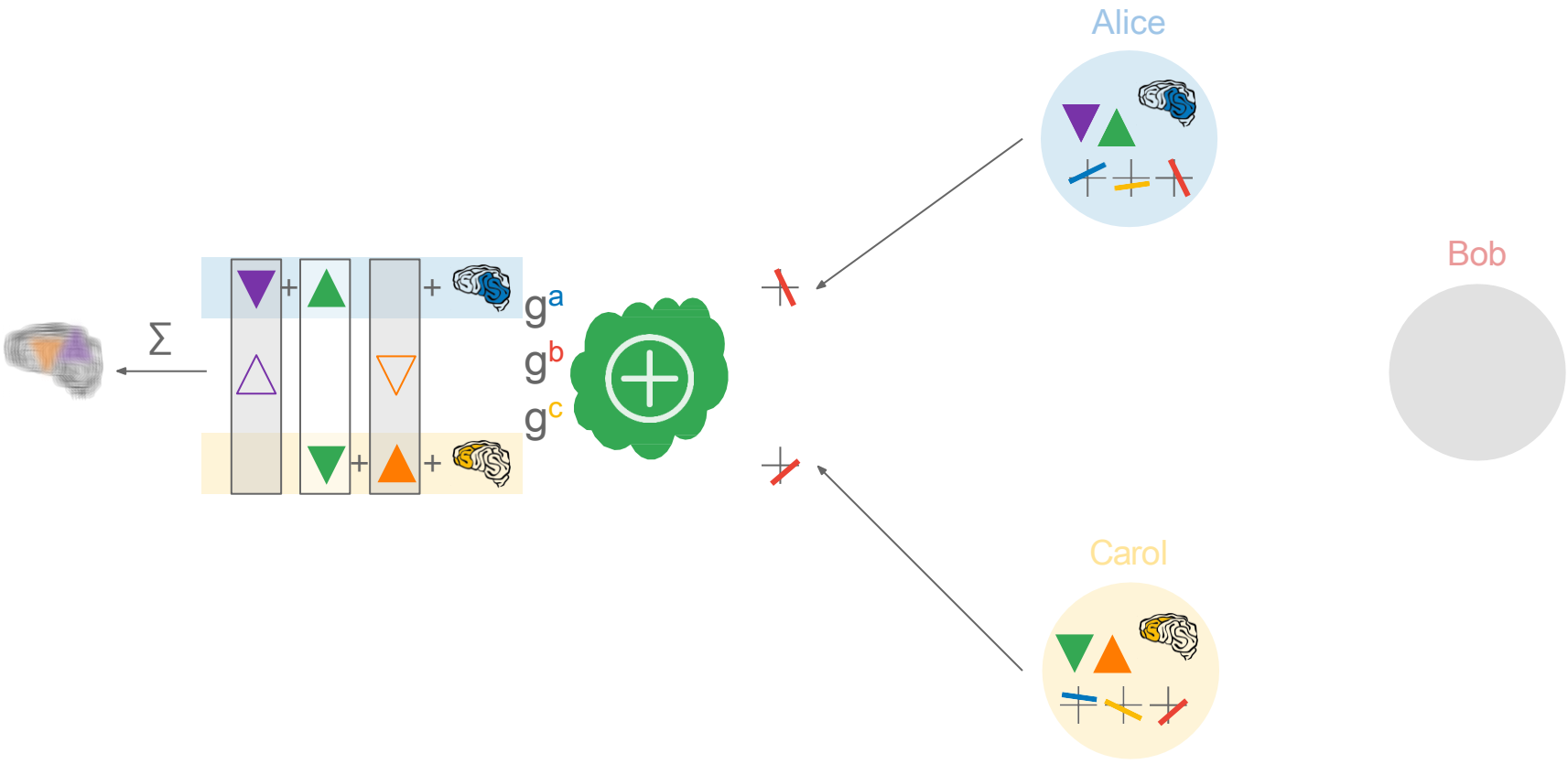
And exchange with their peers

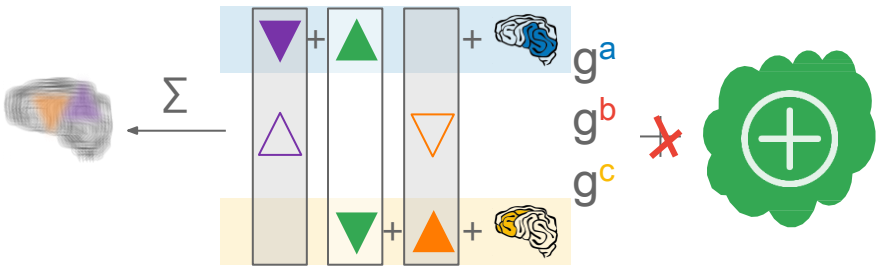


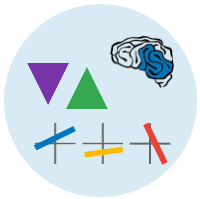
Google

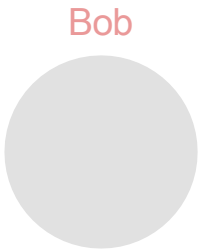

Alice
That's bad.
Σ
Bob
Carol

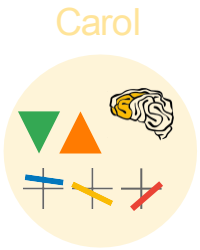

Alice
Bob
Carol
Σ
g^a
g^b
g^c
b?

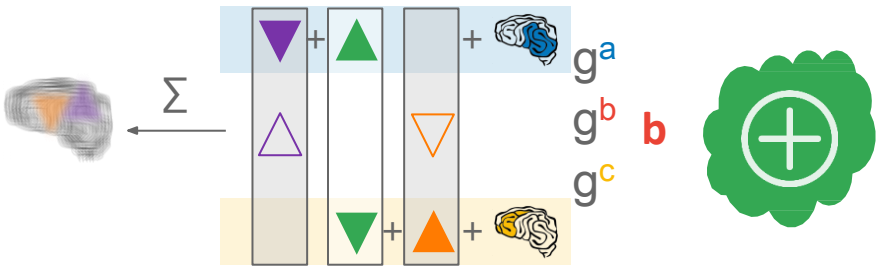

Alice
Bob
Carol
Σ
g^a
g^b
g^c

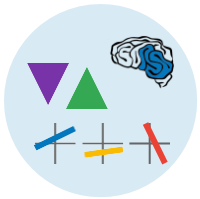

Σ
g^a
g^b
g^c

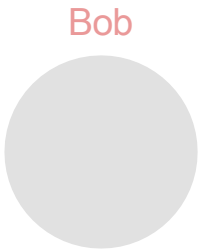

Alice

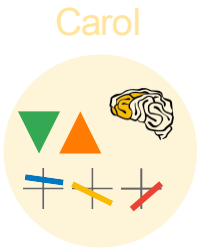

Bob

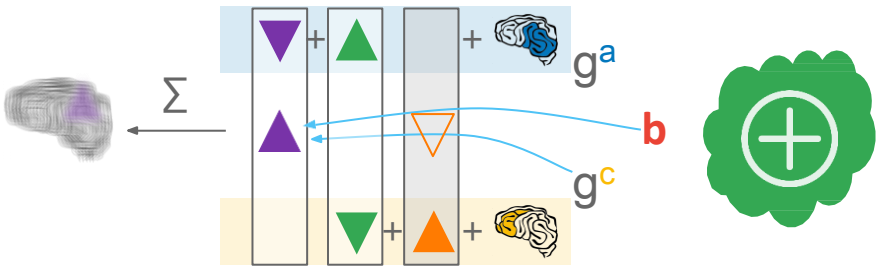

Carol

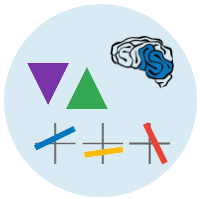

Σ
g^a
g^b
g^c
b

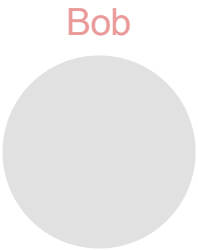

Alice


Bob

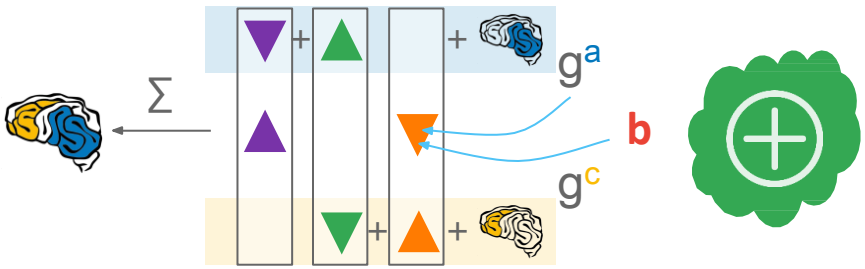

Carol

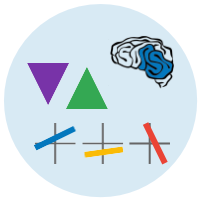

Σ
g^a
b
g^c

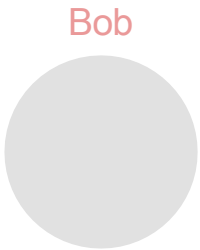

Alice

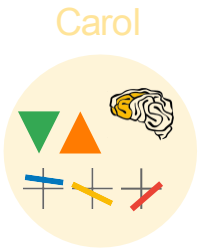

Bob

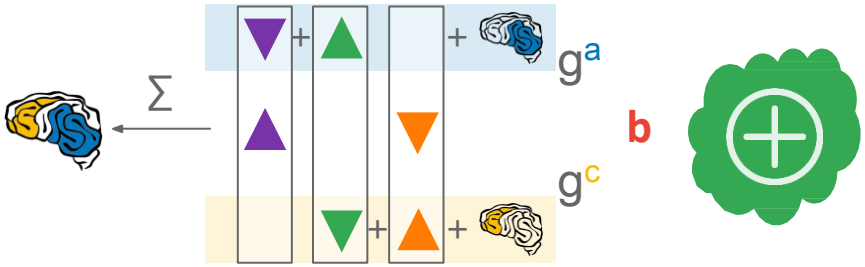

Carol


Σ
g^a
b
g^c

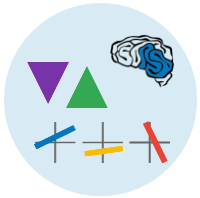

Alice


Bob


Carol



Alice

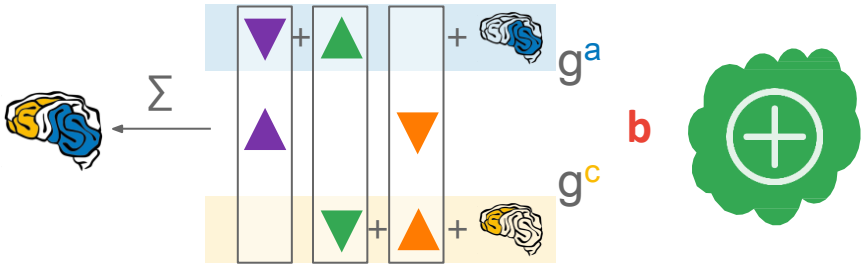


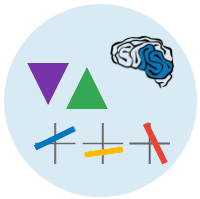
Bob

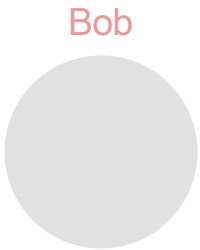
Carol

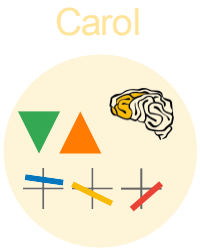
Enough honest users + a high enough threshold
⇒ dishonest users cannot reconstruct the secret.

Google





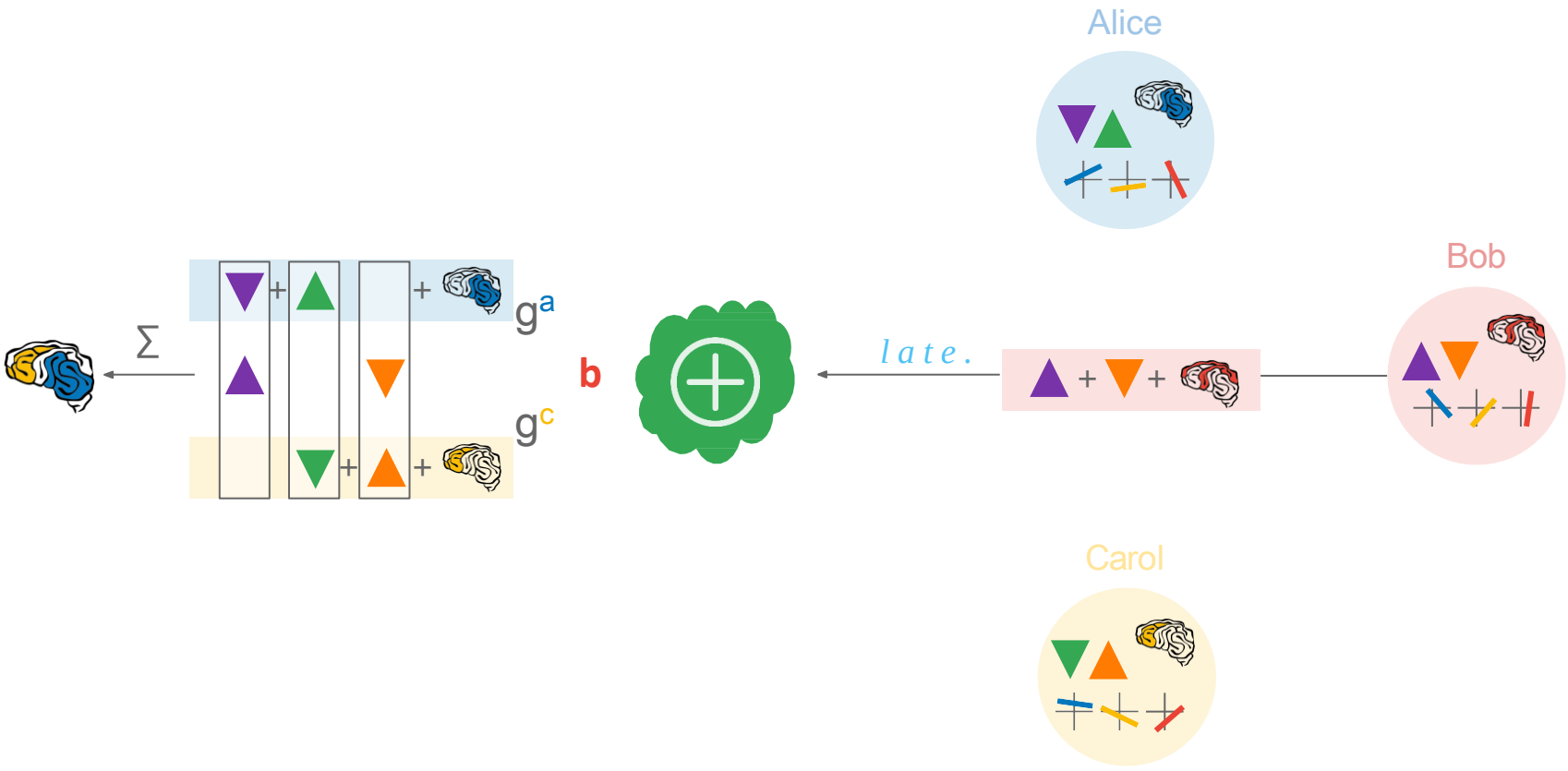


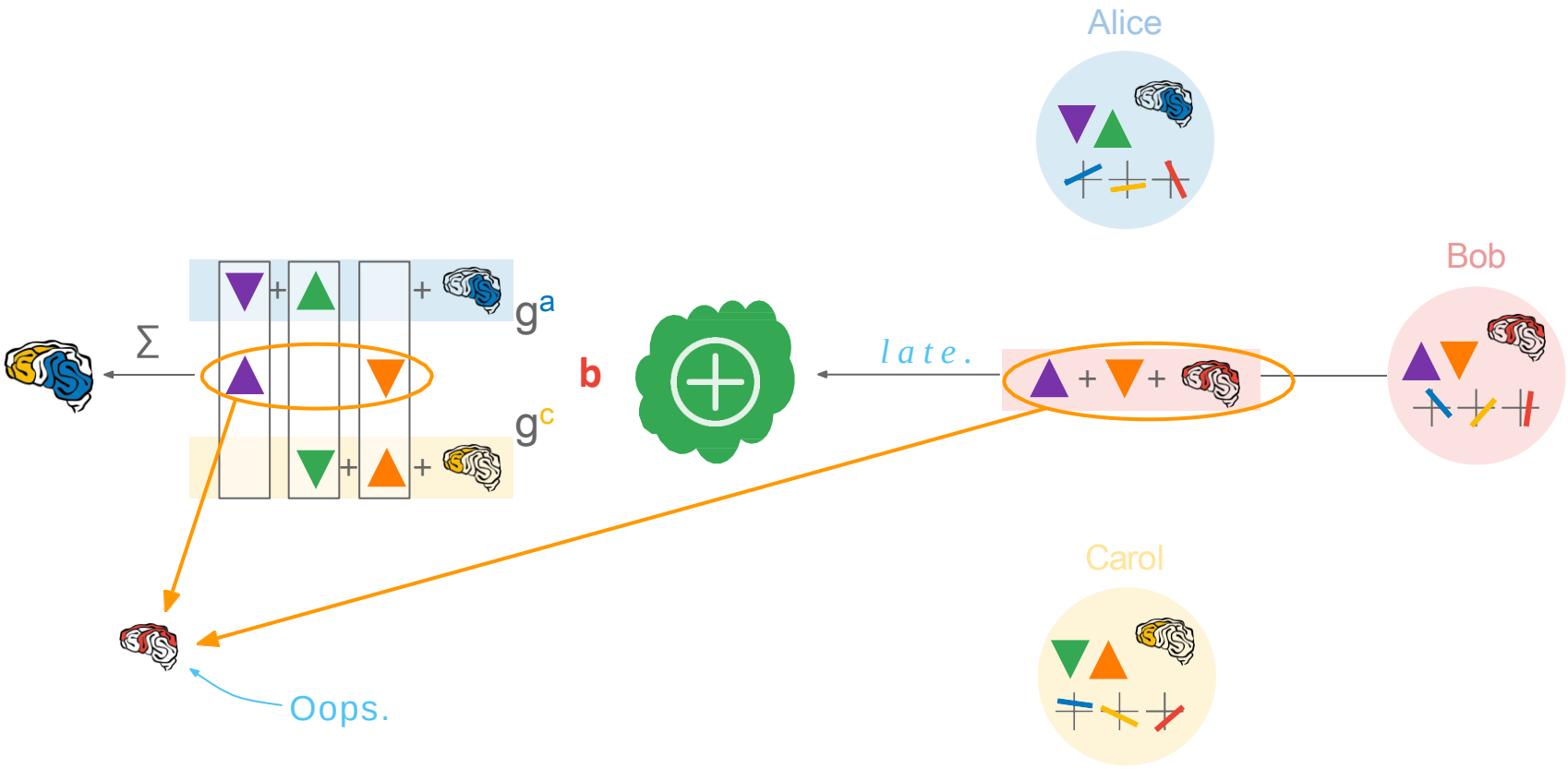


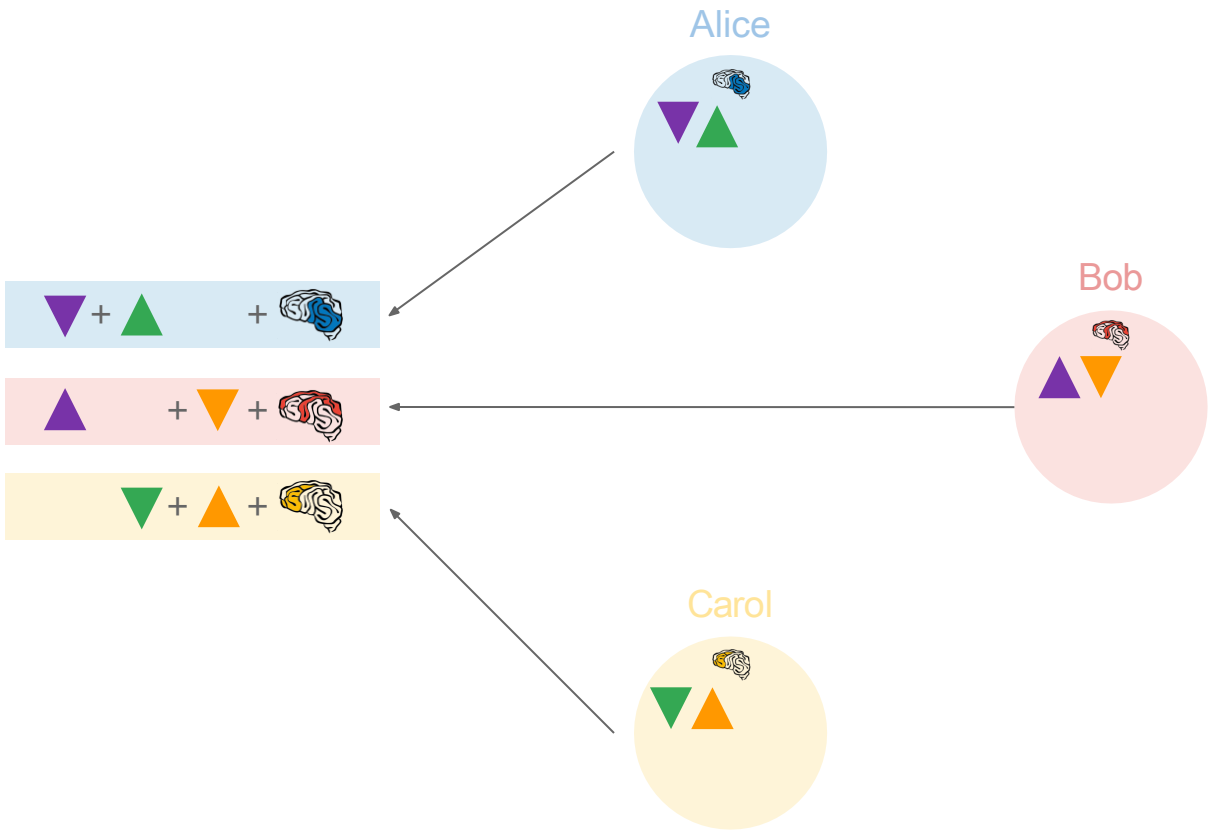
Enough honest users + a high enough threshold
⇒ dishonest users cannot reconstruct the secret.

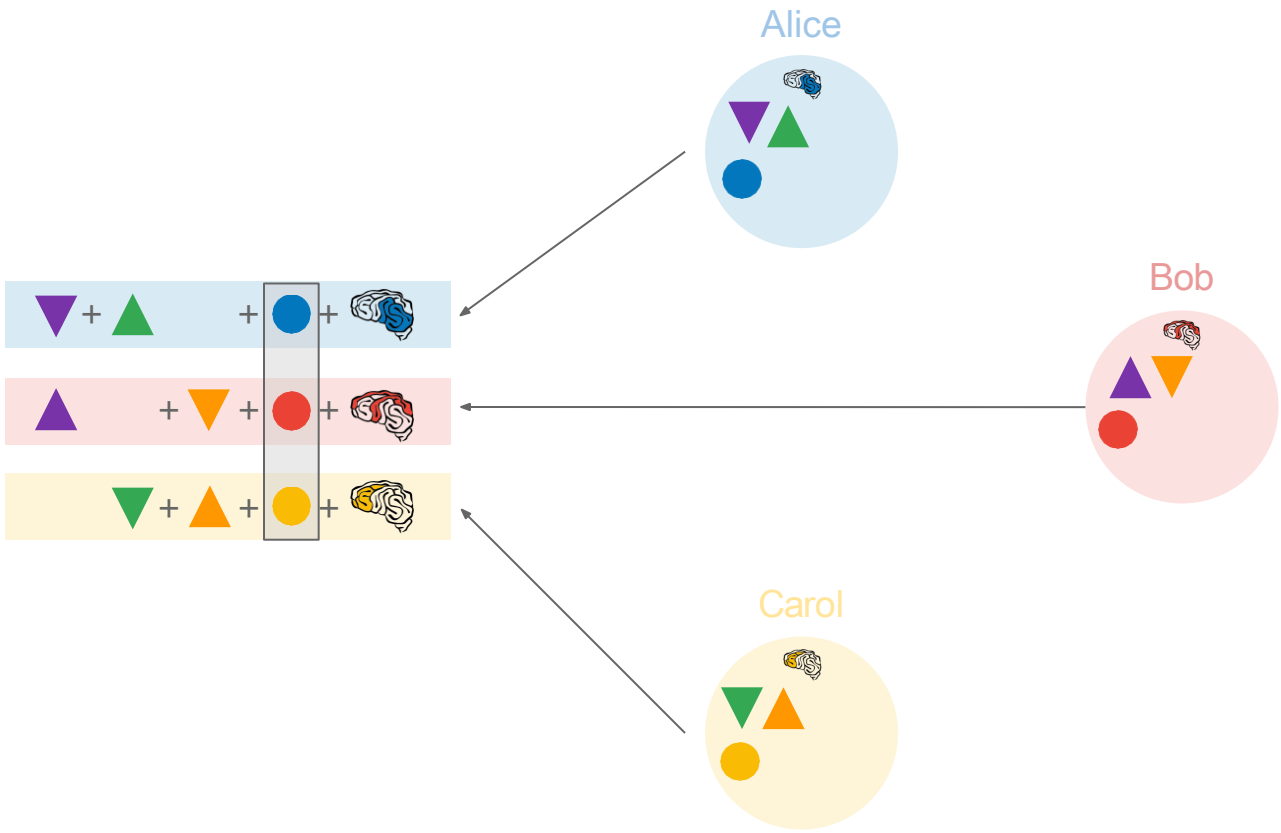
However….

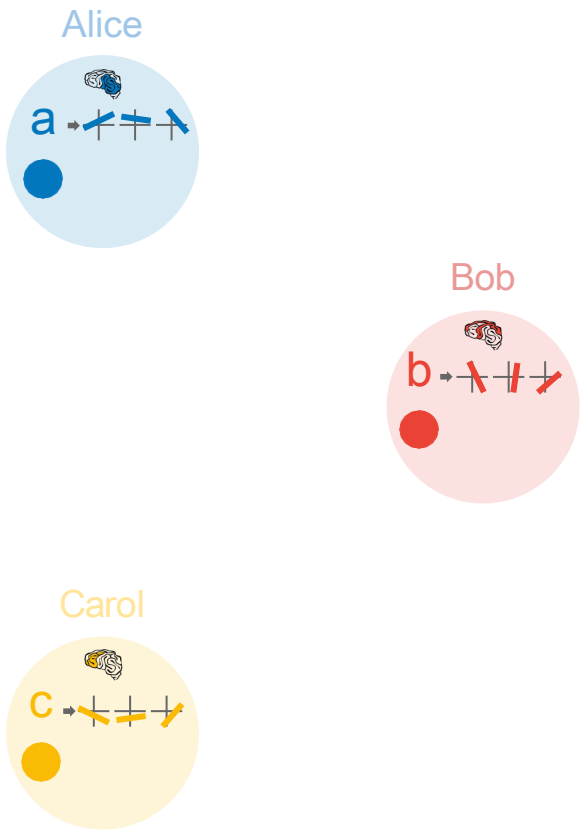
Google


Alice
Bob
Carol
late.
b
Σ
g^a
g^c

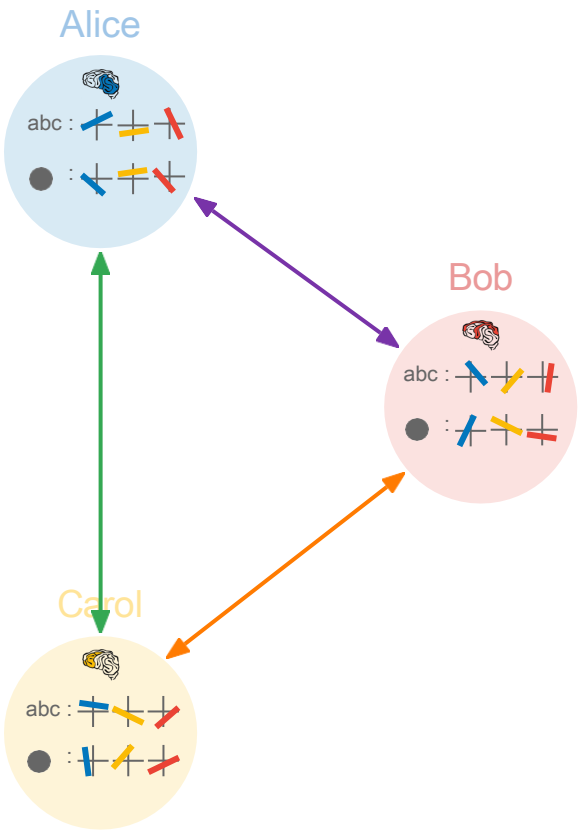

Alice
Bob
Carol
late.
b
Σ
g^a
g^c
Oops.

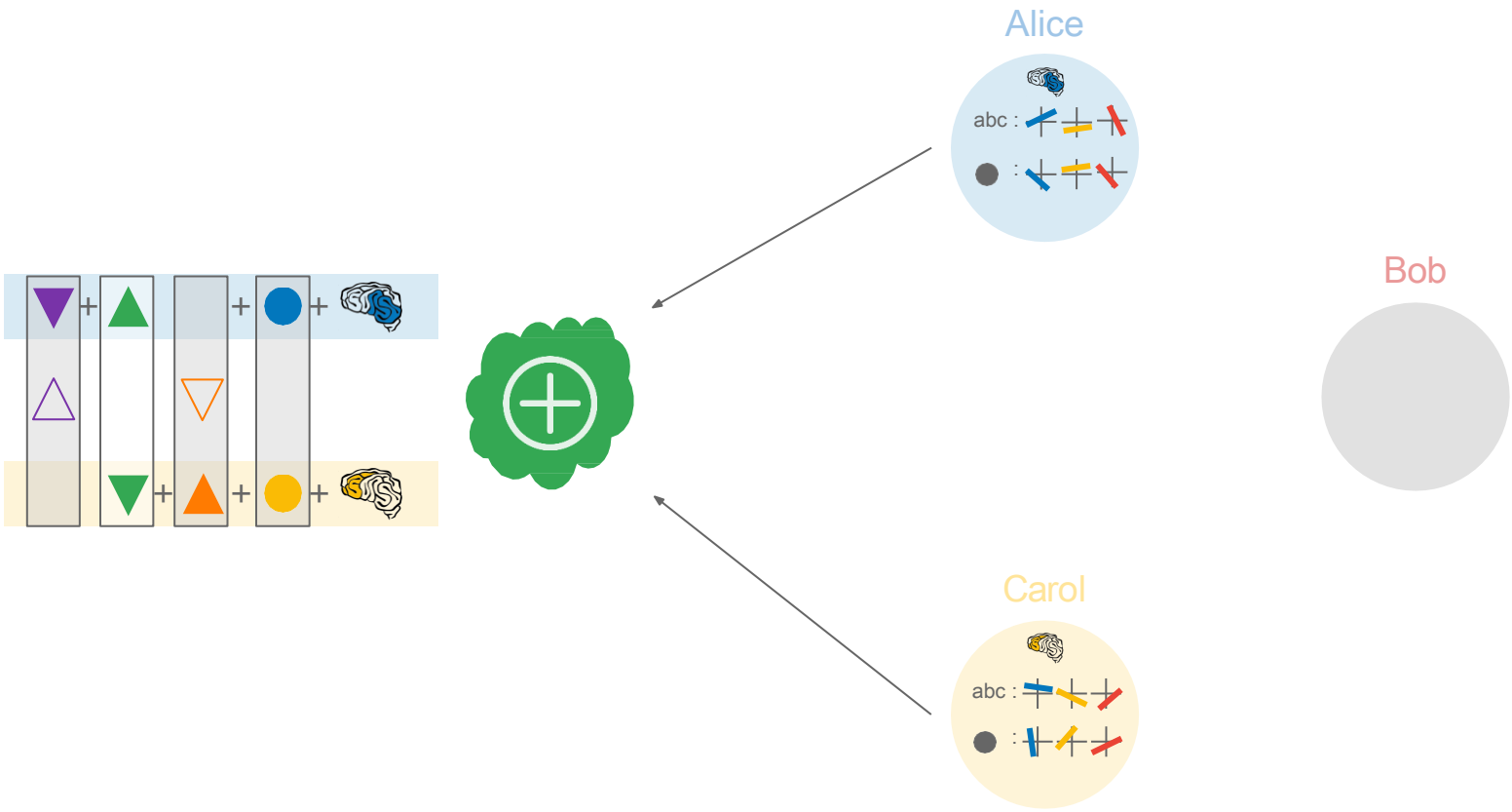

Alice
Bob
Carol

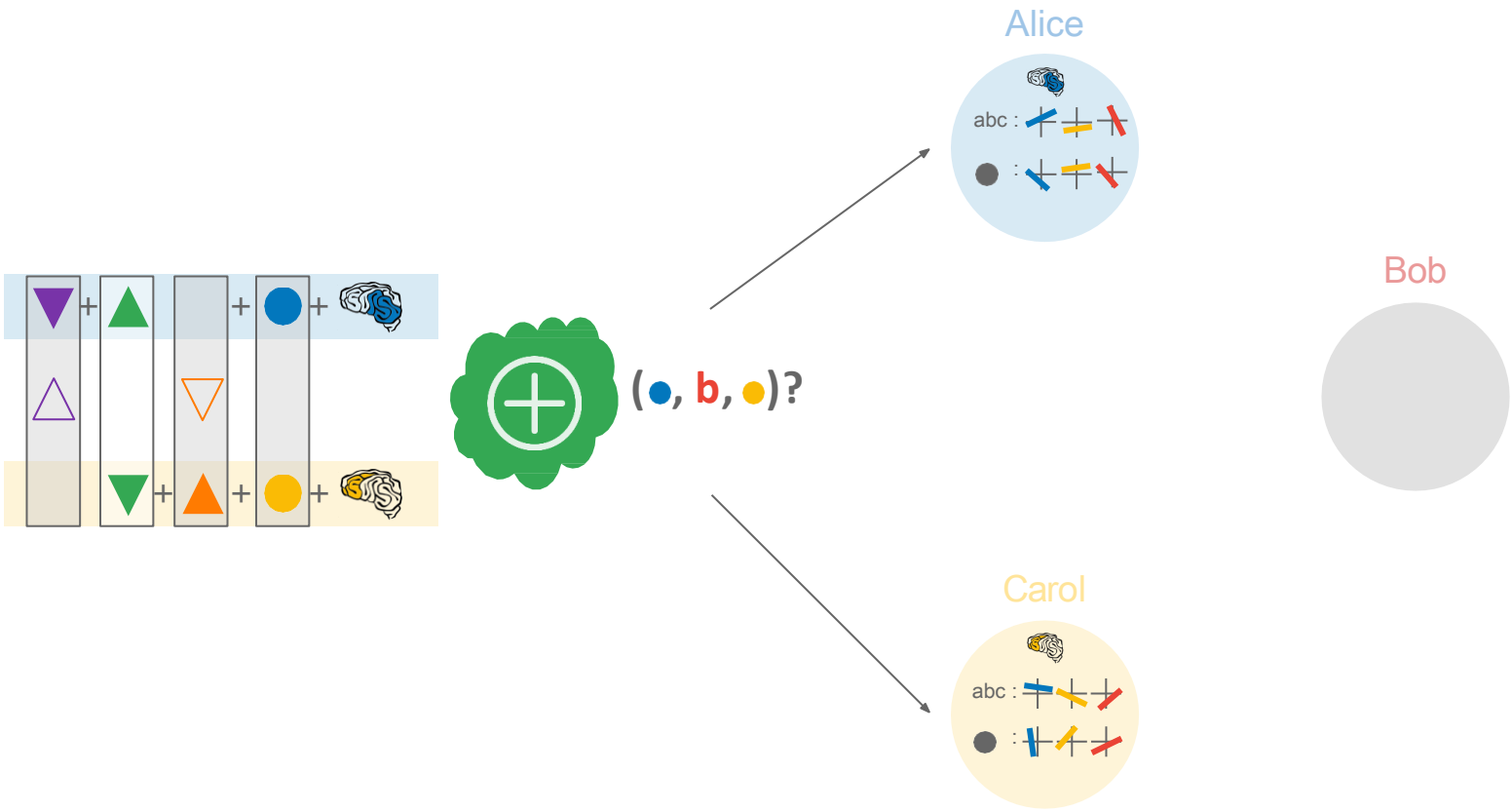

Alice
Bob
Carol

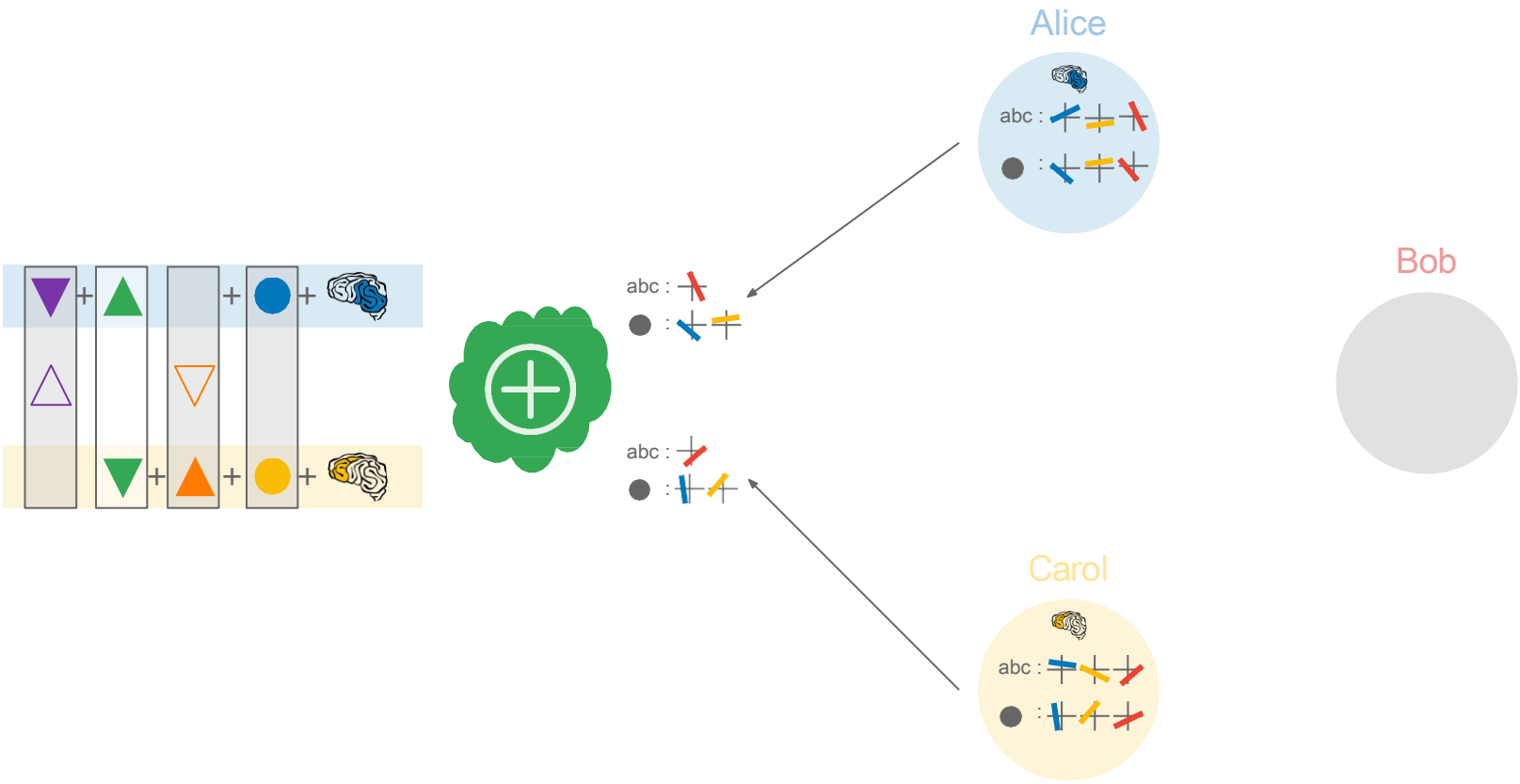

Alice
a
Bob
b
Carol
c

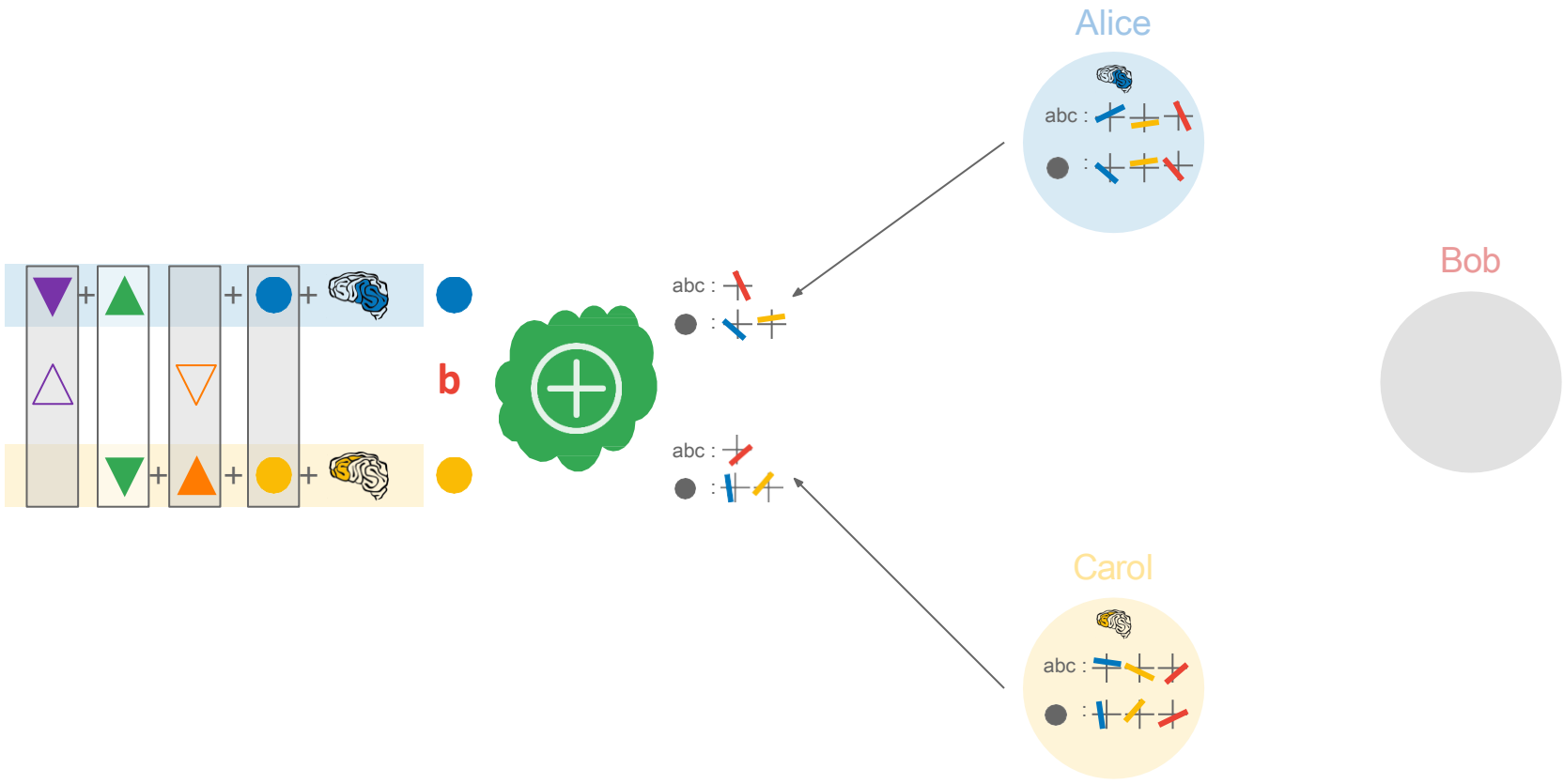

Alice
a
Bob
b
Carol
c

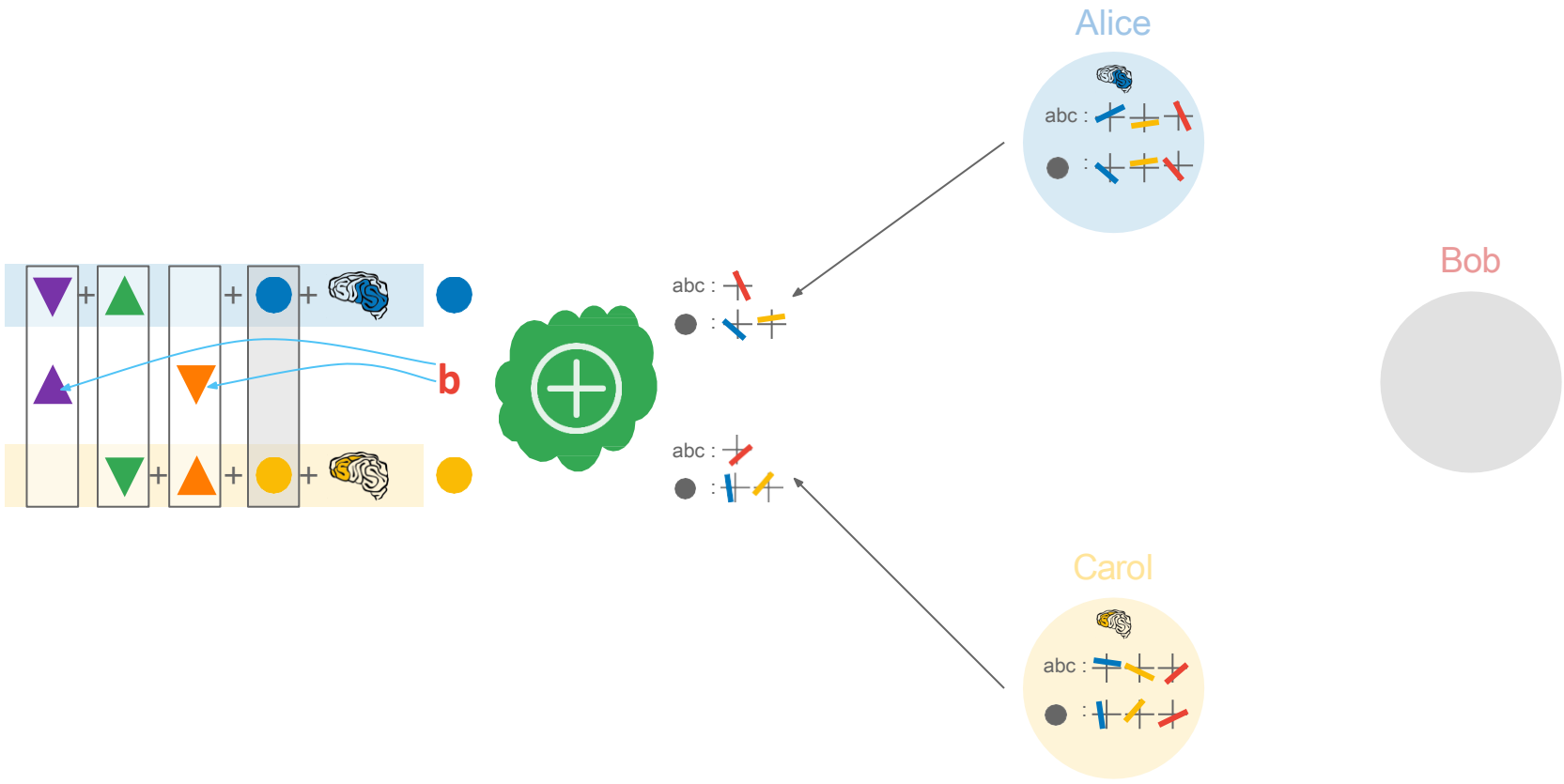

Alice
abc :
:
Bob
abc :
:
Carol
abc :
:

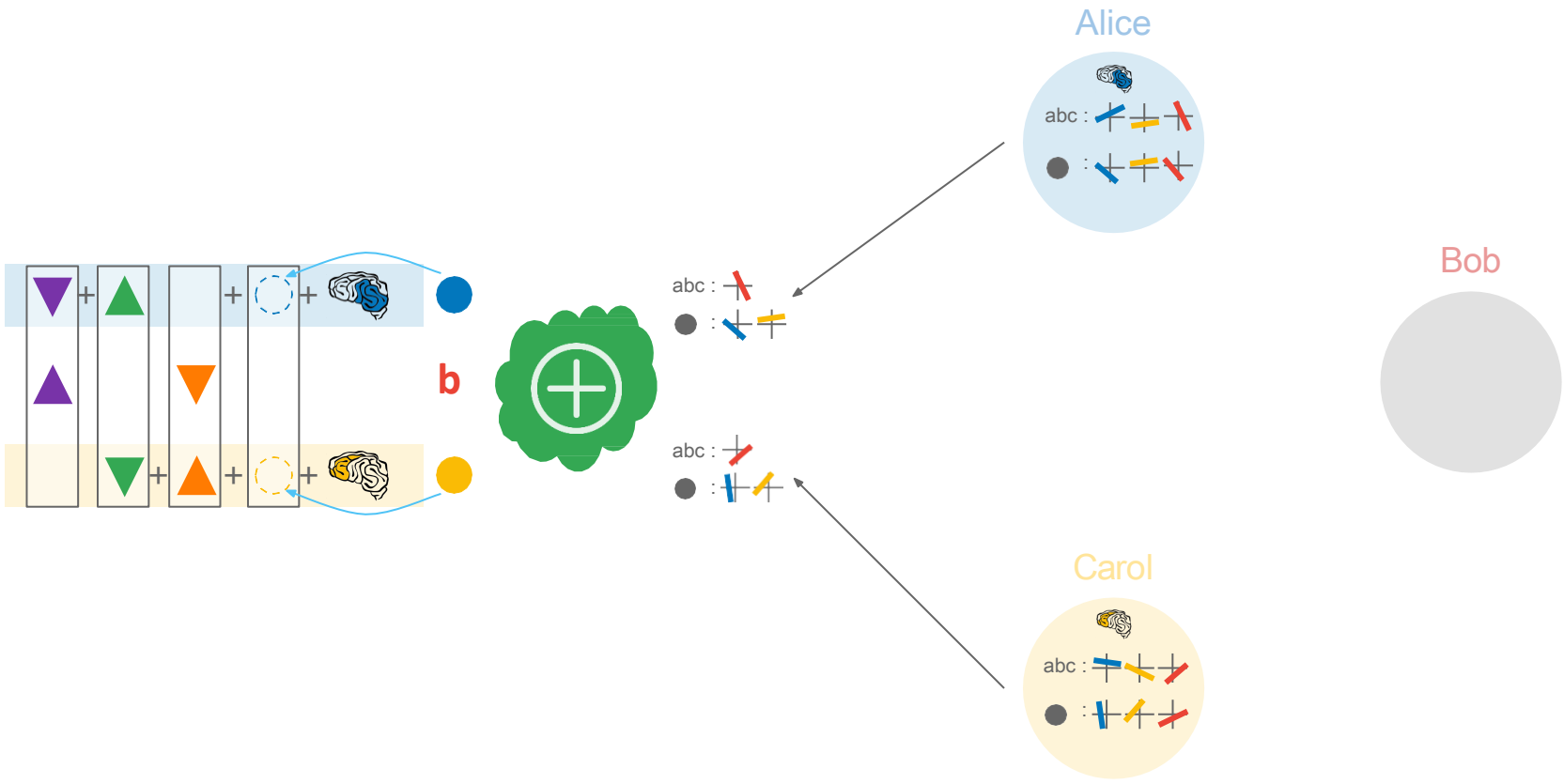

Alice
abc :
:
Bob
Carol
abc :
:

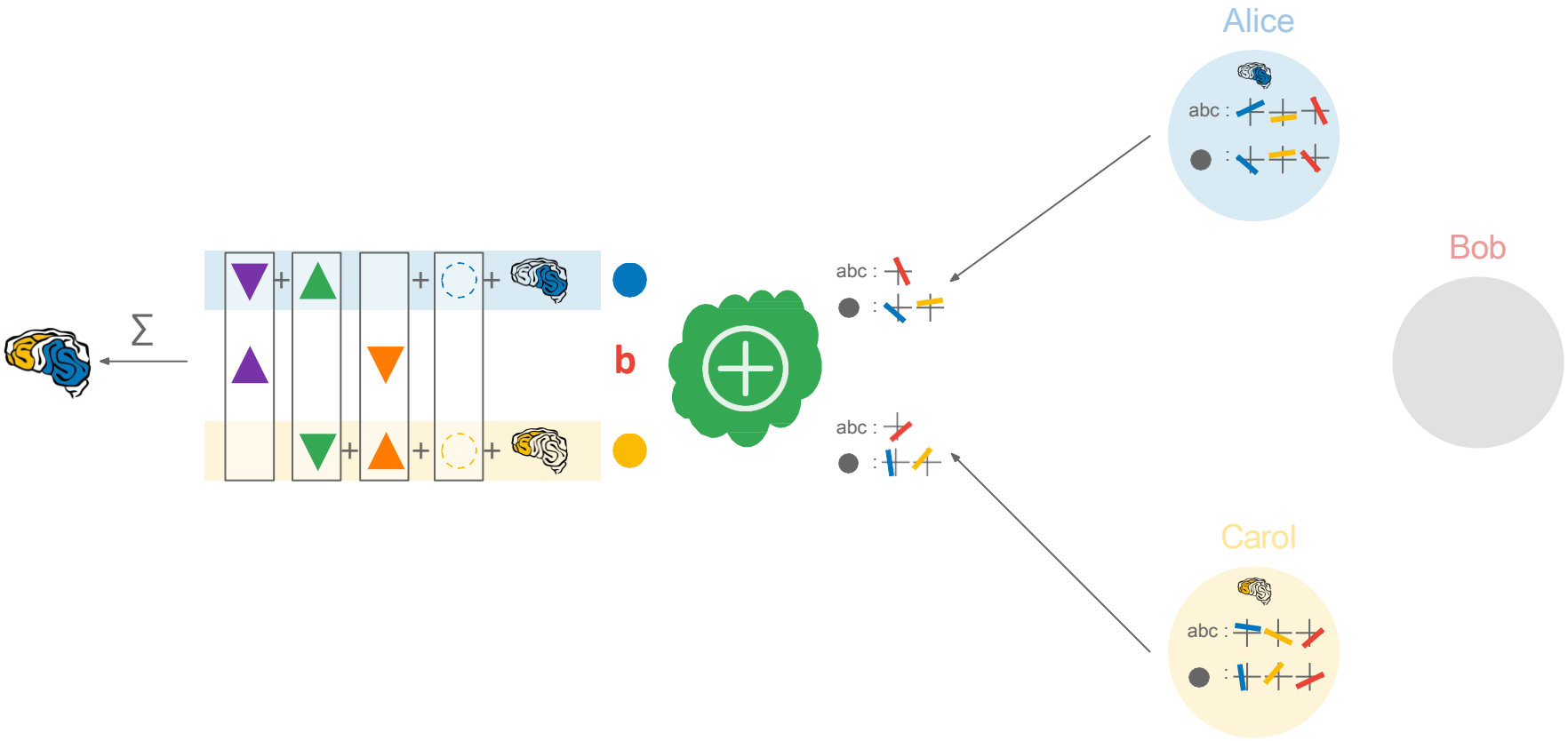

Alice
abc :
:
Bob
Carol
abc :
:
(•, b, •)?

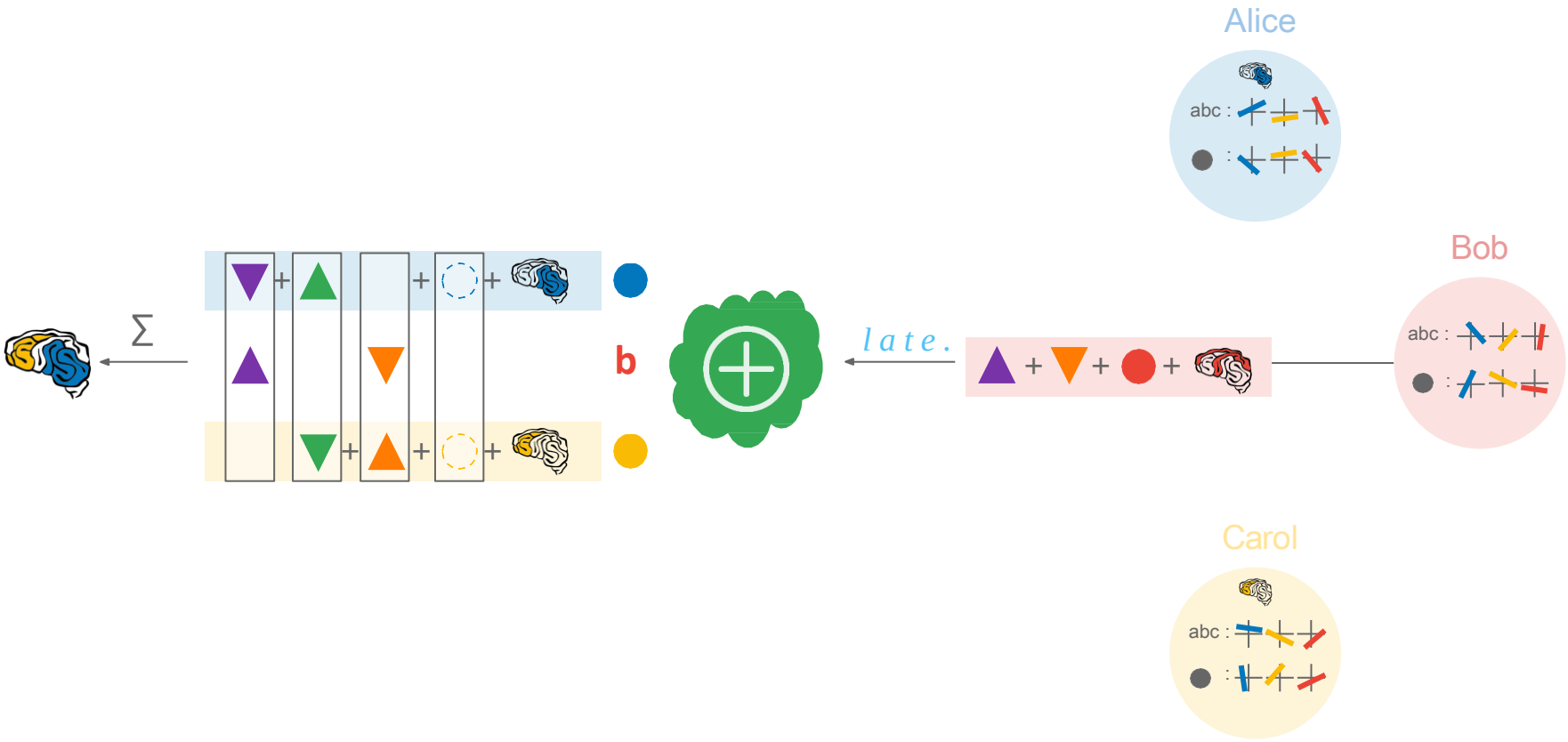

Alice
abc :
:
abc :
:
abc :
:
Carol
abc :
:
Bob

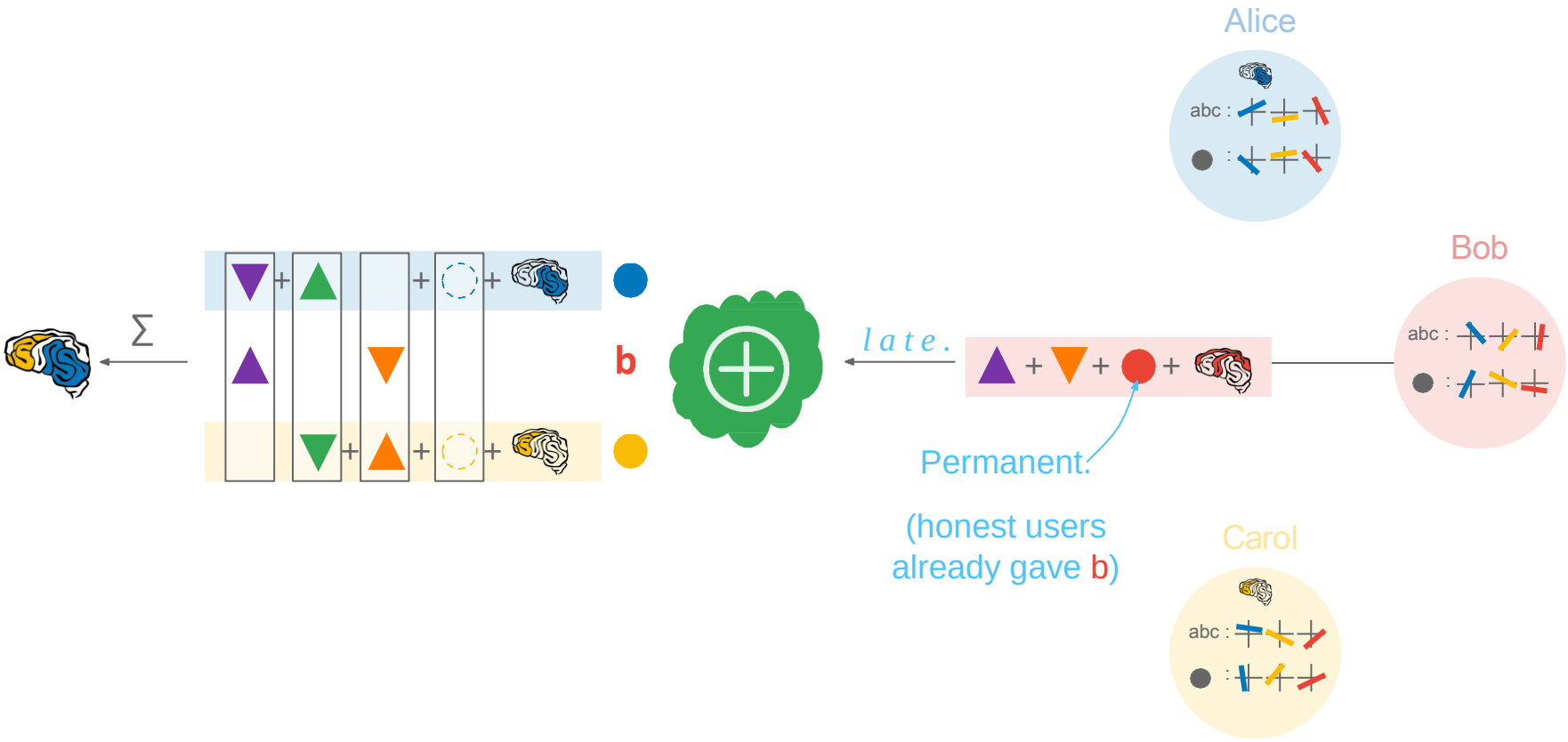

Alice
abc :
:
Bob
abc :
:
b
abc :
:
Carol
abc :
:


Alice
abc :
:
Bob
abc :
:
abc :
:
Carol
abc :
:
b

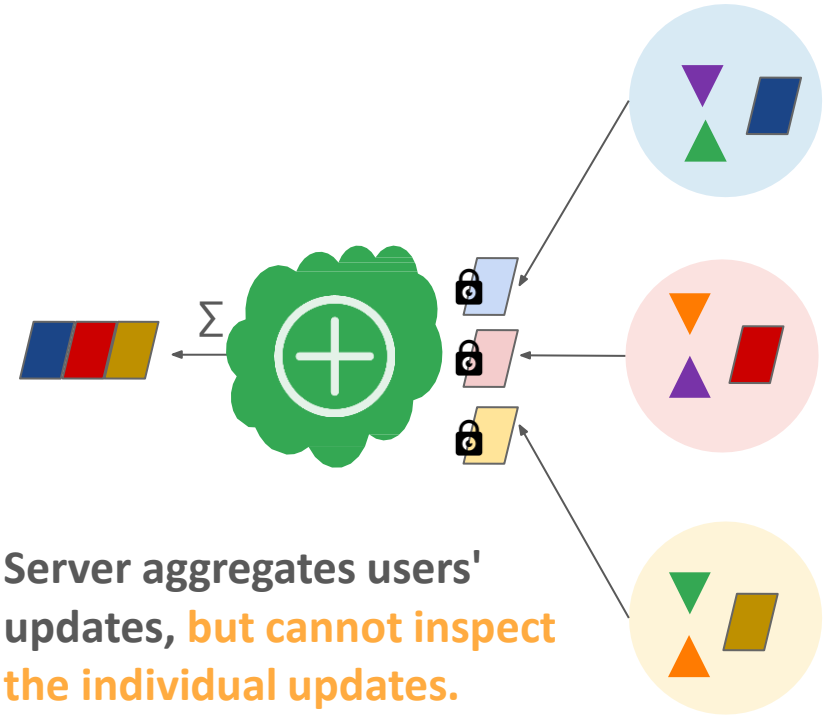

Alice
abc :
:
Bob
b
abc :
:
abc :
:
Carol
abc :
:


Alice
abc :
:
Bob
abc :
:
b
abc :
:
Carol
abc :
:
Σ


Alice
abc :
:
Bob
abc :
:
Carol
abc :
:
late.
Σ
b


Alice
abc :
:
Bob
abc :
:
Carol
abc :
:
Σ
b
late.
Permanent.
(honest users
already gave b)

Secure Aggregation



Server aggregates users' updates, but cannot inspect the individual updates.

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan,
S. Patel, D. Ramage, A. Segal, K. Seth. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS'17.*

Interactive Cryptographic Protocol

Each phase, 1000 clients + server interchange messages over 4 rounds of communication.

Secure
⅓ malicious clients + fully observed server

Robust
⅓ clients can drop out

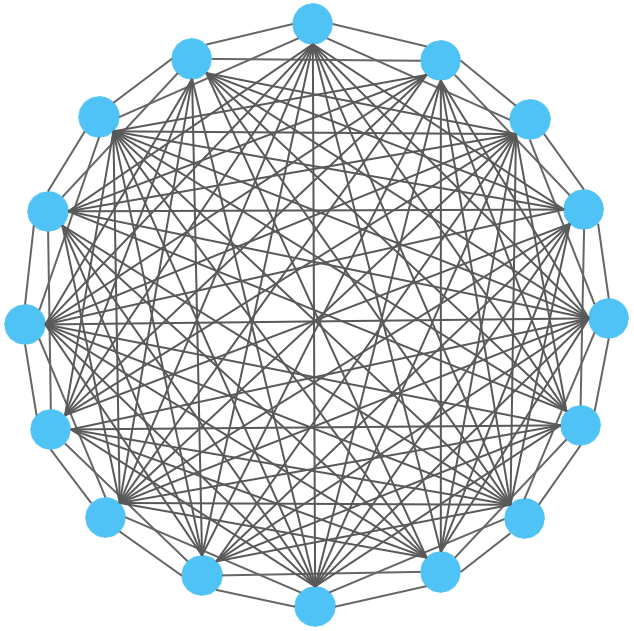
Communication Efficient

# Params	Bits/Param	# Users	Expansion
2^{20} = 1 m	16	2^{10} = 1 k	**1.73x**
2^{24} = 16 m	16	2^{14} = 16 k	**1.98x**

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan,
S. Patel, D. Ramage, A. Segal, K. Seth. ***Practical Secure Aggregation for Privacy-Preserving Machine Learning.***

Complete Graph

of pairwise masks, secret shares

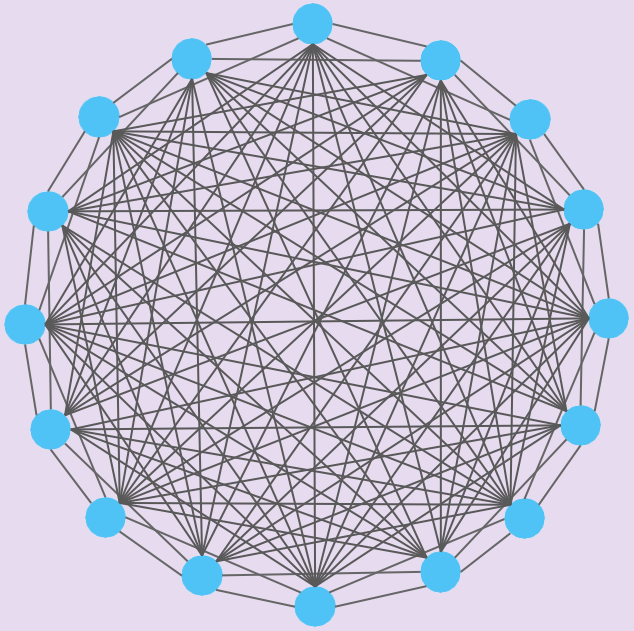


CCS 2017

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan,
S. Patel, D. Ramage, A. Segal, K. Seth. ***Practical Secure Aggregation for Privacy-Preserving Machine Learning.***

Complete Graph

of pairwise masks, secret shares

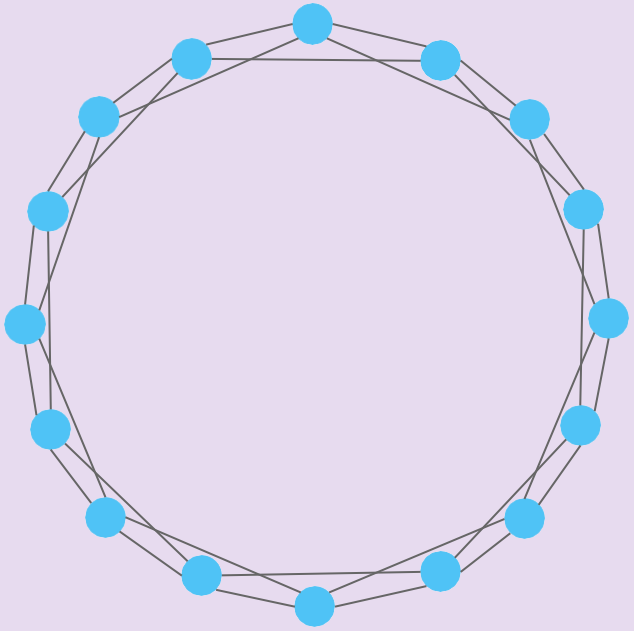


CCS 2020

J. Bell, K. Bonawitz, A. Gascon, T. Lepoint, M. Raykova
Secure Single-Server Aggregation with (Poly)Logarithmic Overhead.

Random Harary(n, k)

n clients, *k* neighbors, random node assignments
$k = O(\log n)$



Secure Aggregation

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, K. Seth. **Practical Secure Aggregation for Privacy-Preserving Machine Learning.** *CCS 2017.*

J. Bell, K. Bonawitz, A. Gascon, T. Lepoint, M. Raykova ***Secure Single-Server Aggregation with (Poly)Logarithmic Overhead.*** CCS 2020.

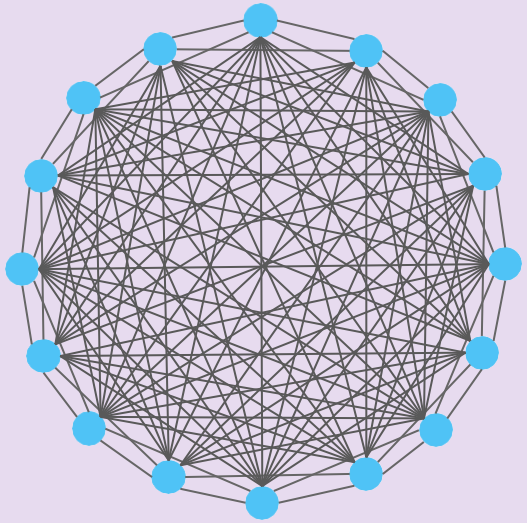
Protocol	Server		Client	
	Computation	Communication	Computation	Communication
Bonawitz et al. (CCS 2017)	$O(n^2 l)$	$O(n^2 + nl)$	$O(n^2 + nl)$	$O(n + l)$
Bell et al. (CCS 2020)	$O(n \log^2 n + nl \log n)$	$O(n \log n + nl)$	$O(\log^2 n + l \log n)$	$O(\log n + l)$
Insecure Solution	$O(nl)$	$O(nl)$	$O(l)$	$O(l)$

CCS 2017

K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan,
S. Patel, D. Ramage, A. Segal, K. Seth. ***Practical Secure Aggregation for Privacy-Preserving Machine Learning.***

Complete Graph

of pairwise masks, secret shares



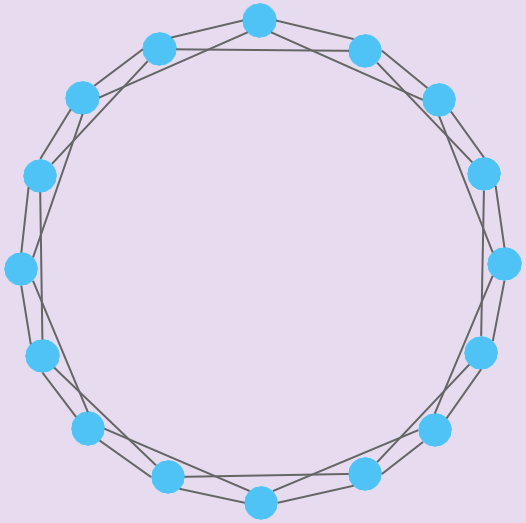
Cost of **1 million cohorts**, each of **1000 clients**

CCS 2020

J. Bell, K. Bonawitz, A. Gascon, T. Lepoint, M. Raykova
Secure Single-Server Aggregation with (Poly)Logarithmic Overhead.

Random Harary(n, k)

n clients, *k* neighbors, random node assignments
k = O(log *n*)



Cheaper!

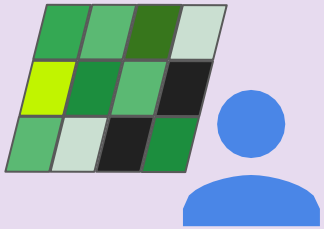
Cost of **a *single* cohort** of **1 billion clients**

Sparse Federated Learning & Analytics

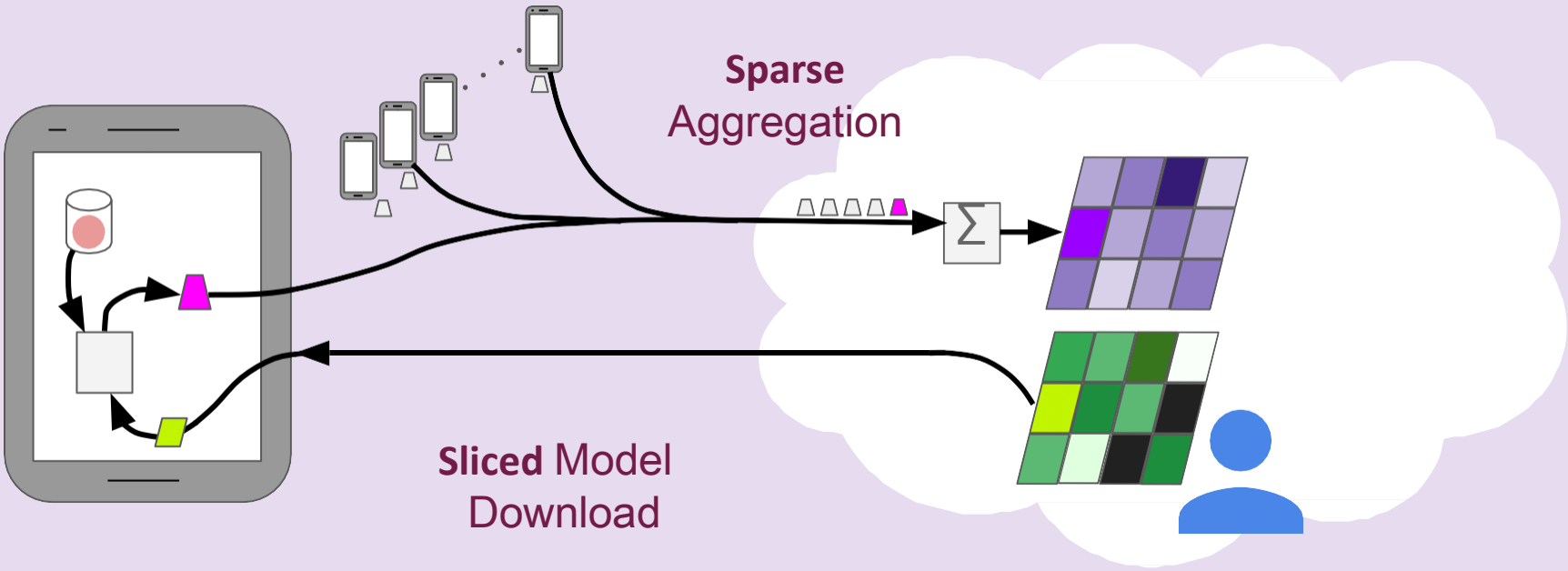
- **Embedding-based models**
  - Word-based Language Models
  - Object Recognition
- **Compound models**
  - Multiple fixed domains

e.g. locations, companies, etc

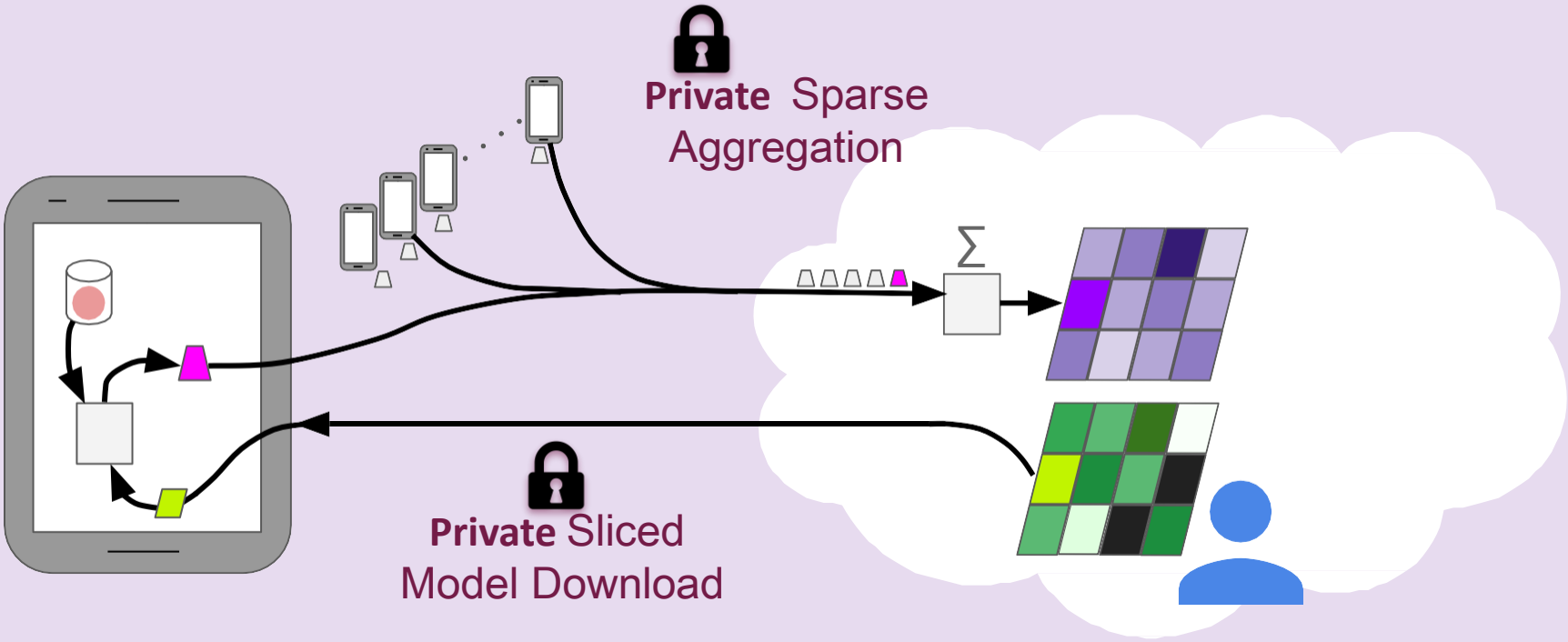
  - Genre/cluster models
  - Pluralistic models
- **Federated Analytics**



Sparse Federated Learning & Analytics



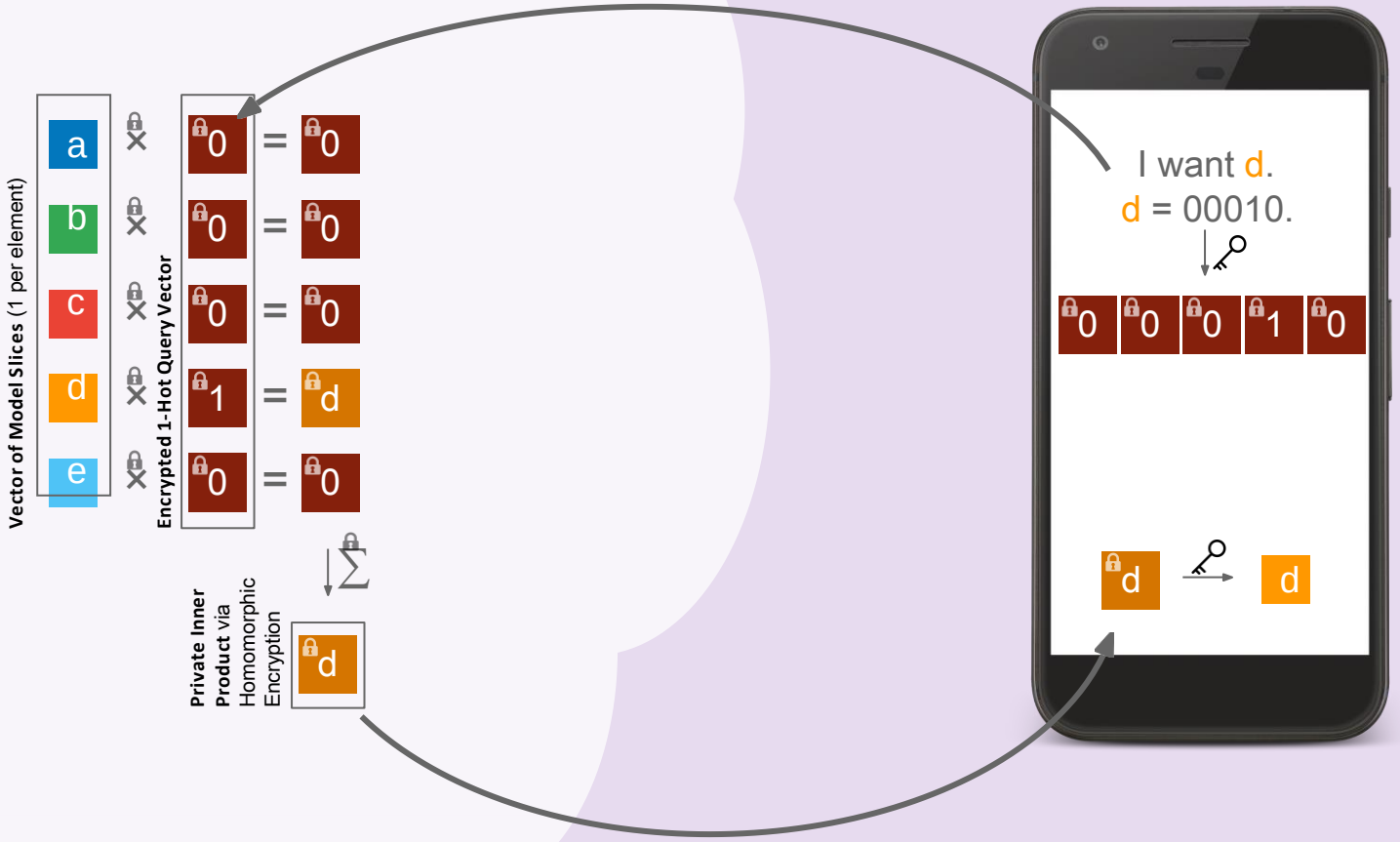
Sparse Federated Learning & Analytics



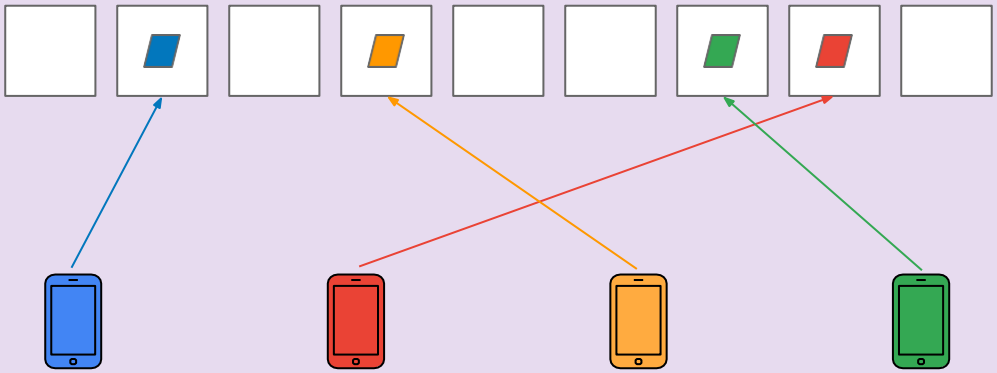
Private Sliced Model Download // Private Information Retrieval

Google

Private Sliced Model Download // Private Information Retrieval



Private Sparse Aggregation // Shuffling via Secure Aggregation



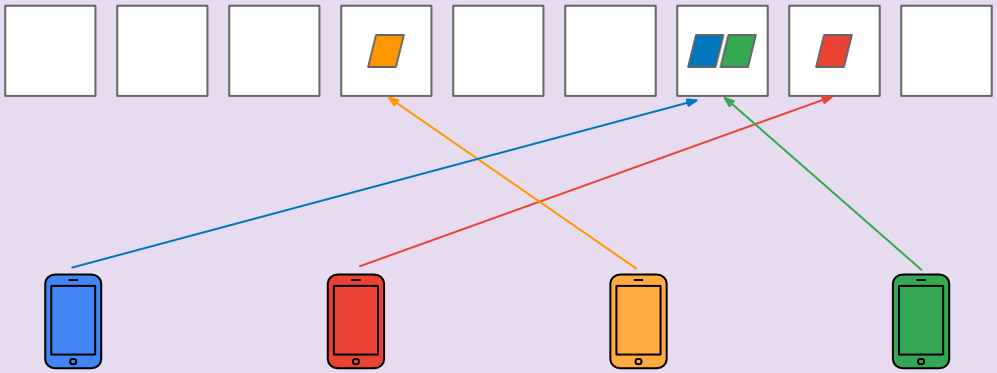
← Vector with message-sized slots

← Clients choose a random slot

Google

Private Sparse Aggregation // Shuffling via Secure Aggregation

Birthday "Paradox": conflicts are likely, even with quite large vectors



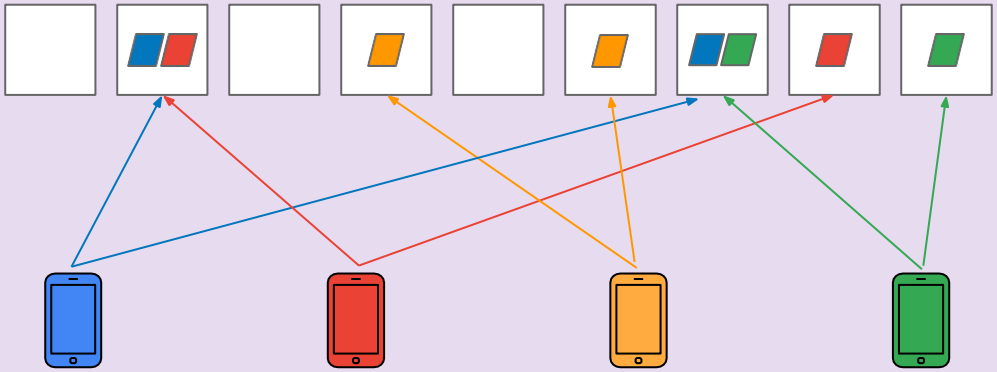
← Vector with message-sized slots

← Clients choose a random slot

Google

Private Sparse Aggregation // Shuffling via Secure Aggregation

Invertible Bloom Lookup Tables (IBLTs)



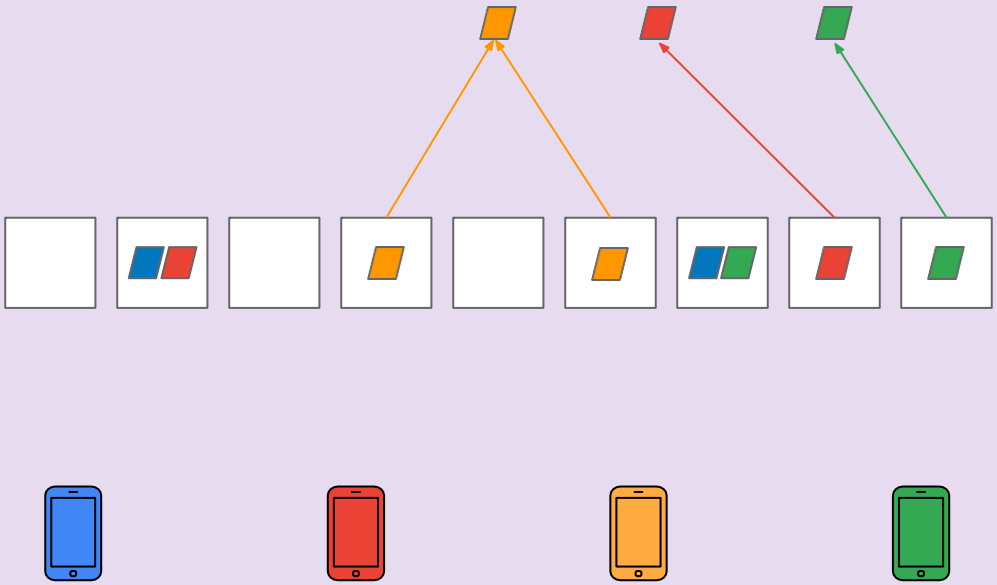
← **(pseudonym+message)**-sized slots

← Clients choose **random pseudonym,** map to **k slots** using hash functions

Google

Private Sparse Aggregation // Shuffling via Secure Aggregation

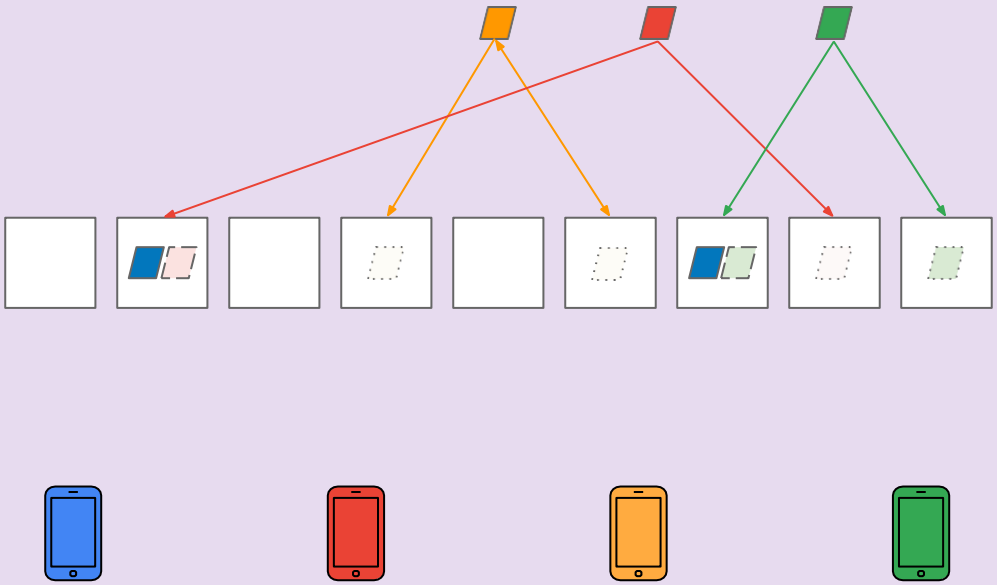
Invertible Bloom Lookup Tables (IBLTs)



← Server **recovers** (pseudonym+message) for all **non-conflict slots**

Private Sparse Aggregation // Shuffling via Secure Aggregation

Invertible Bloom Lookup Tables (IBLTs)

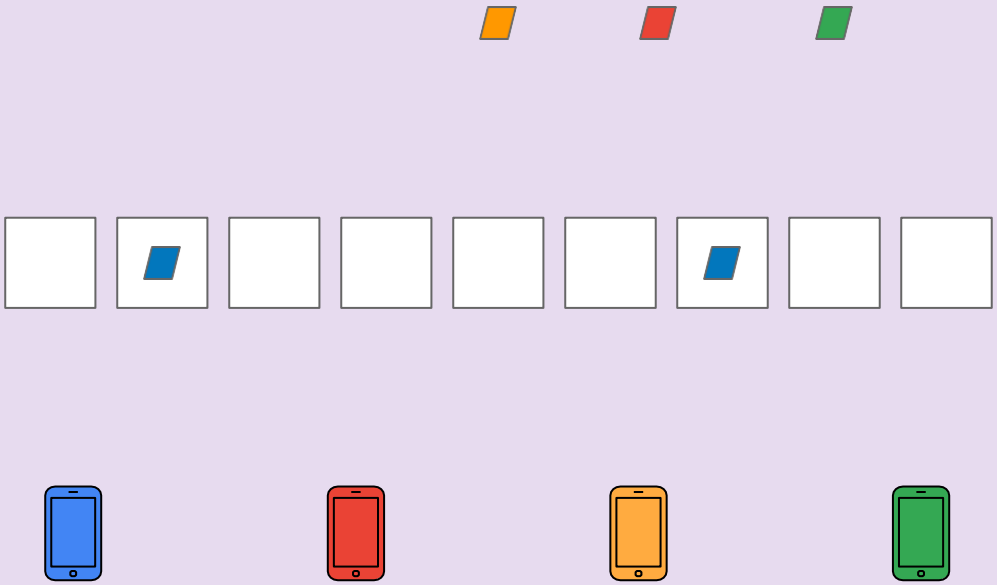


← Server recovers (pseudonym+message) for all non-conflict slots

Then **removes all copies** from vector

Private Sparse Aggregation // Shuffling via Secure Aggregation

Invertible Bloom Lookup Tables (IBLTs)



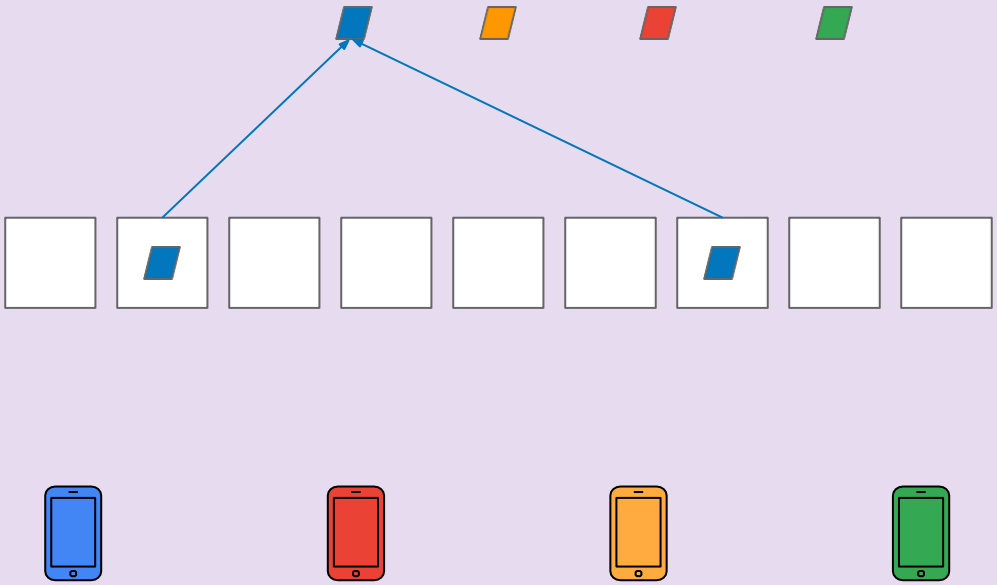
← Server recovers (pseudonym+message) for all non-conflict slots

Then removes all copies from vector

De-conflicting more slots

Private Sparse Aggregation // Shuffling via Secure Aggregation

Invertible Bloom Lookup Tables (IBLTs)



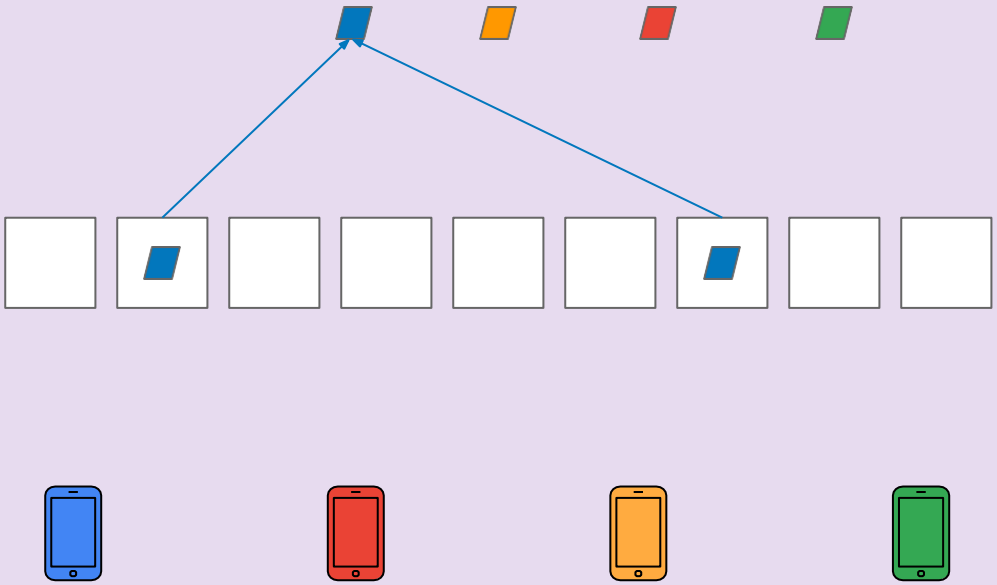
← Server recovers (pseudonym+message) for all non-conflict slots

Then removes all copies from vector De-conflicting more slots

Repeat until all messages extracted
(or no more progress)

Private Sparse Aggregation // Shuffling via Secure Aggregation

Invertible Bloom Lookup Tables (IBLTs)



J. Bell, K. Bonawitz, A. Gascon, T. Lepoint, M. Raykova

Secure Single-Server Aggregation with (Poly)Logarithmic Overhead. CCS 2020.

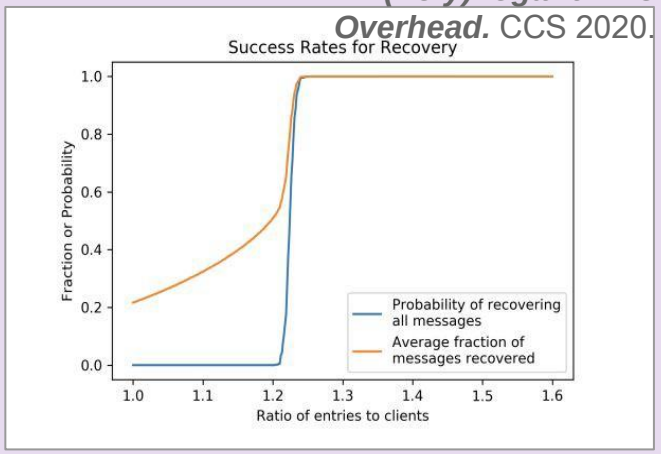


Figure 4. Expected fraction of messages recovered and probability of recovering all messages against the length l of the vectors used. For this the number of clients is n = 10000 and each inserts their message in c = 3 places.

Vector Length ~= 1.3x messages!

Google

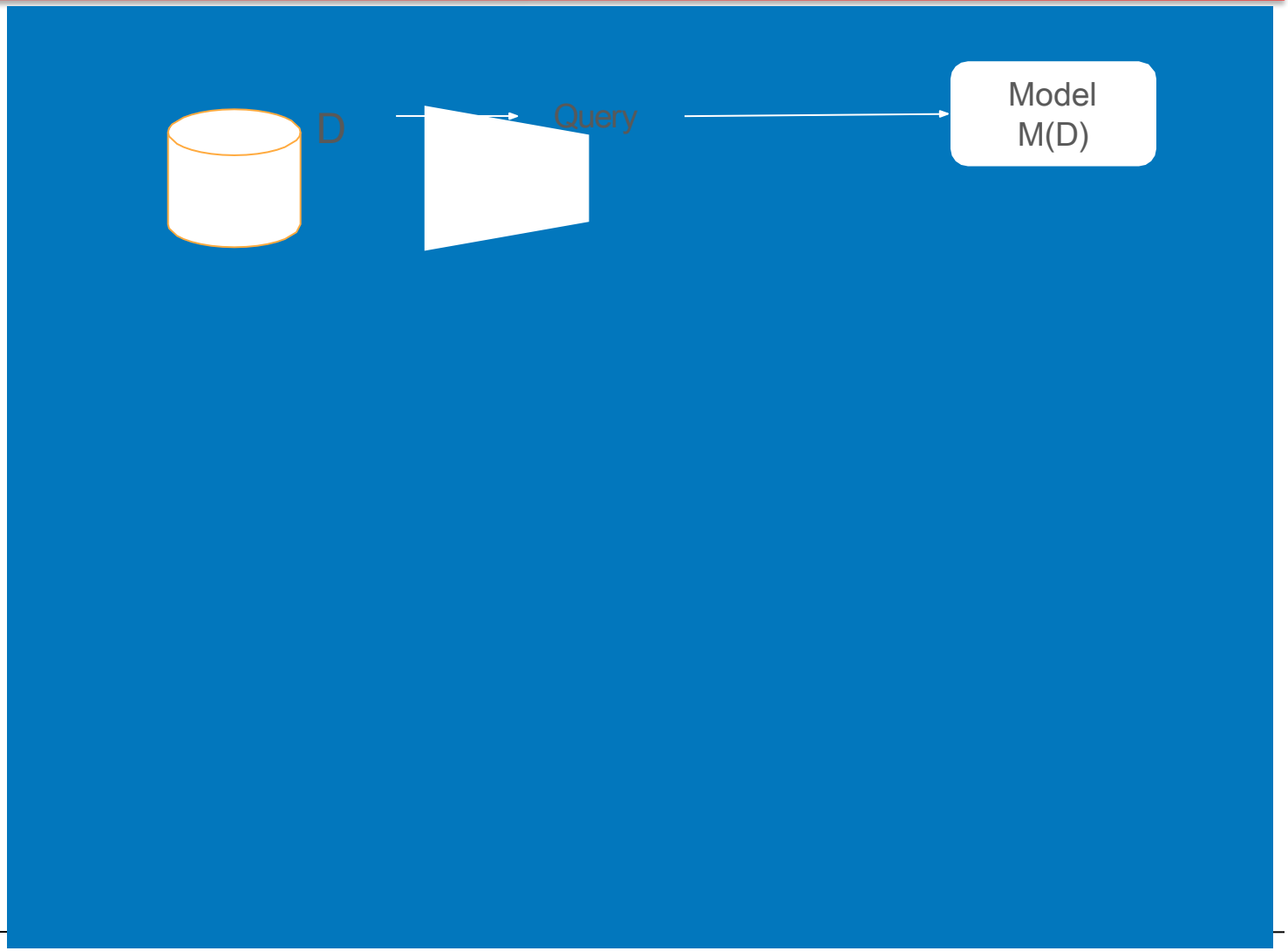
Differentially Private Federated Training

Differential Privacy

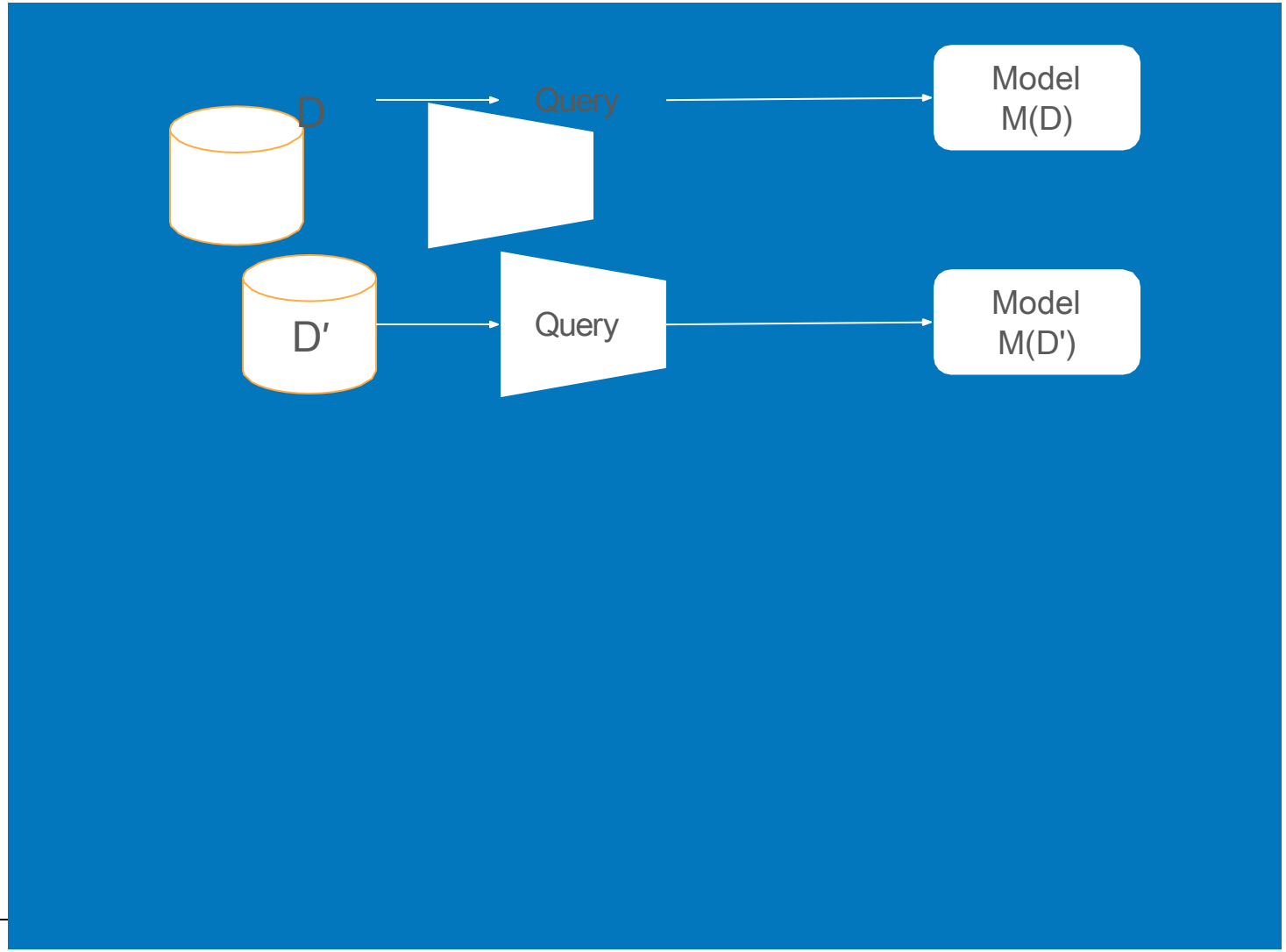
Differential privacy is the statistical science of trying to learn **as much as possible about a group** while learning **as little as possible about any individual in it.**

Andy Greenberg
Wired 2016.06.13

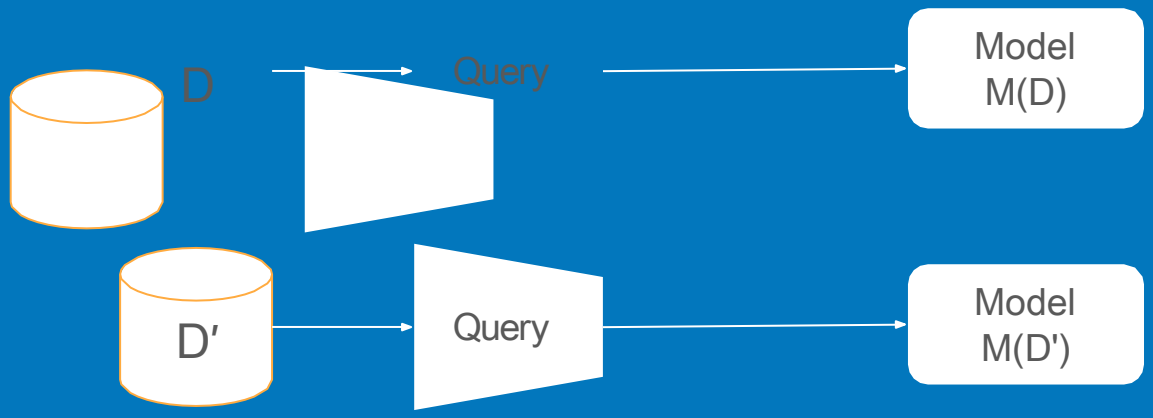
Differential Privacy



Differential Privacy



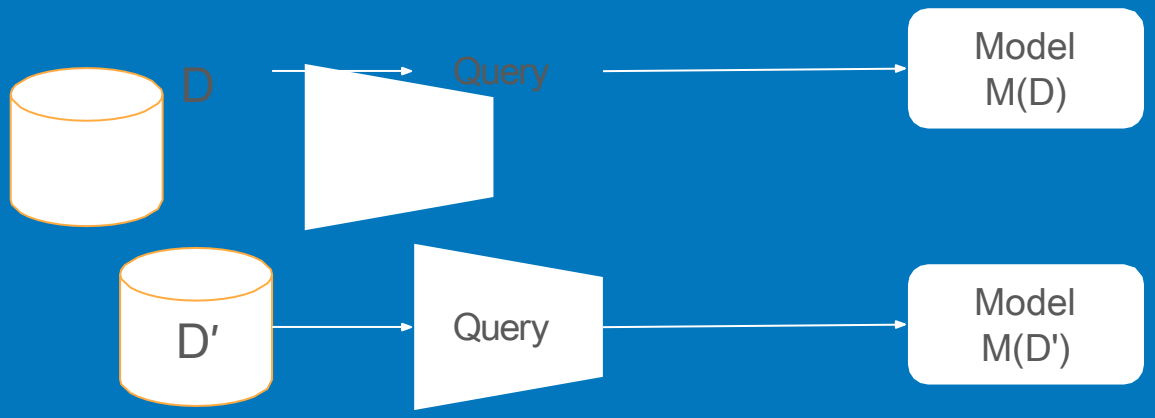
Differential Privacy



(ε, δ)-Differential Privacy: The distribution of the output $M(D)$ (a trained model) on database (training dataset) D is **nearly the same** as $M(D')$ for all adjacent databases D and D'

$\forall S$: $\Pr[M(D) \in S] \leq \exp(\varepsilon) \cdot \Pr[M(D') \in S] + \delta$

Record-level Differential Privacy



(ε, δ)-Differential Privacy: The distribution of the output $M(D)$ (a trained model) on database (training dataset) is nearly the same as $M(D')$ for all **adjacent** databases and D'

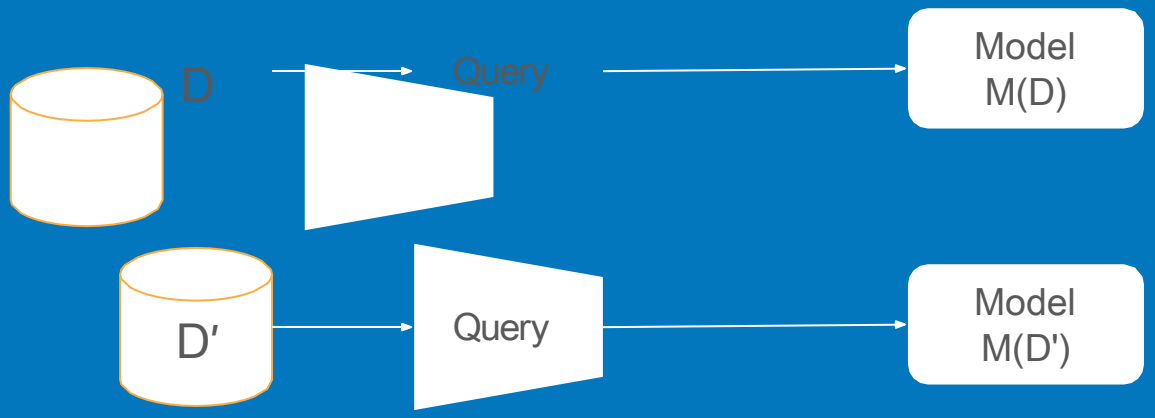
Goodfellow, H. B. McMahan, I. Mironov, K.

adjacent: Sets D and D' differ only by presence/absence of one **example**

M. Abadi, A. Cl

Talwar, & L. Zhang.
Deep Learning wi
Differential Privac
CCS 2016.

User-level Differential Privacy

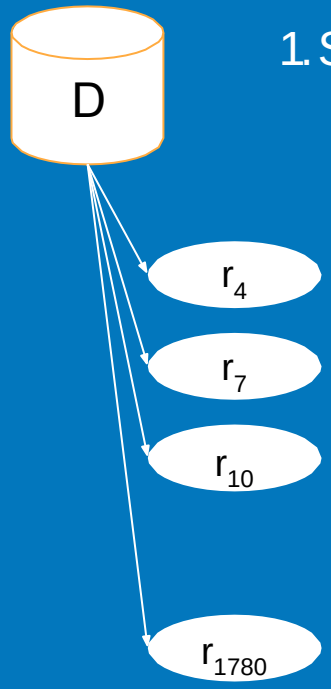


(ε, δ)-Differential Privacy: The distribution of the output $M(D)$ (a trained model) on database (training dataset) nearly the same as $M(D')$ for all **adjacent** databases D D'

adjacent: Sets D and D' differ only by presence/absence of one ~~**example**~~ **user**

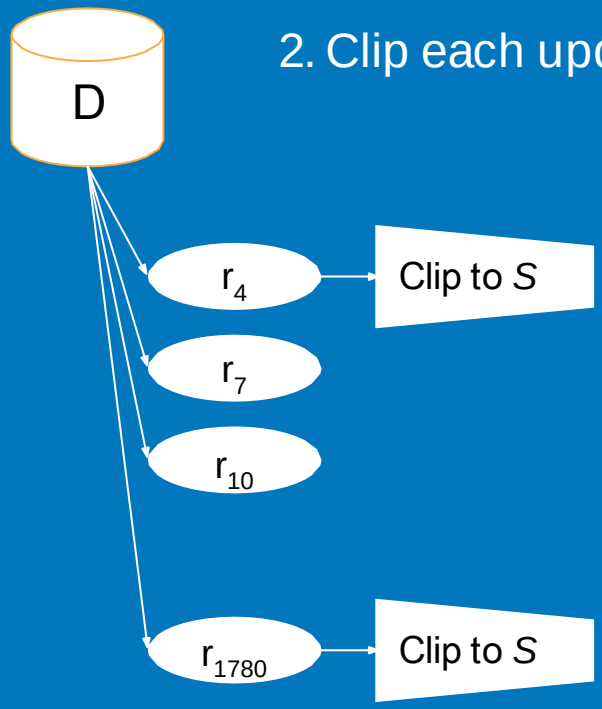
H. B. McMahan ***Learning Differe Private Recurrent Language Models****. ICLR 2018.*

Iterative training with differential privacy

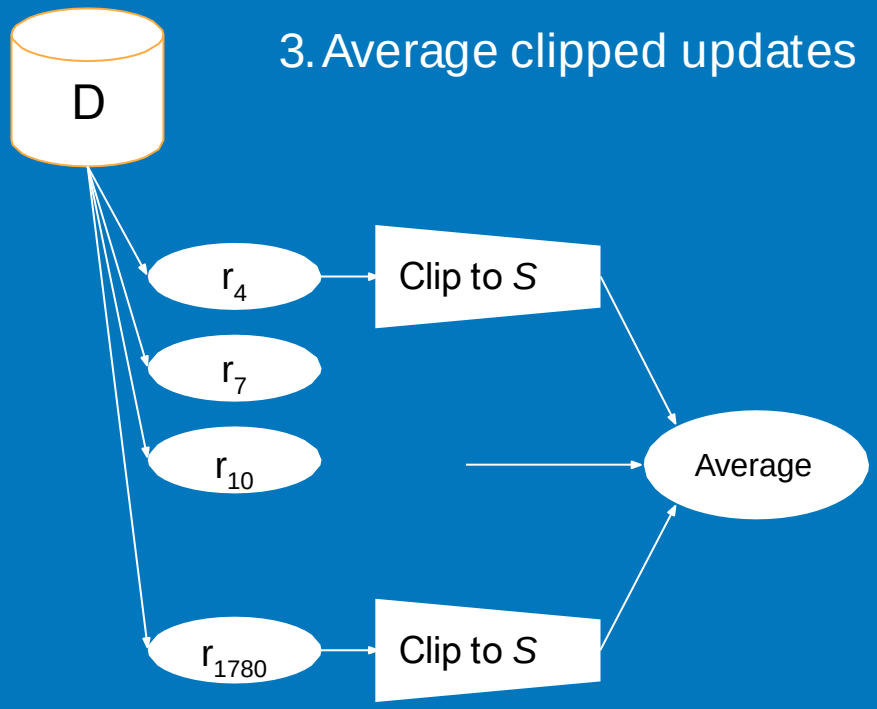


1. Sample a batch of clients ***uniformly at random***

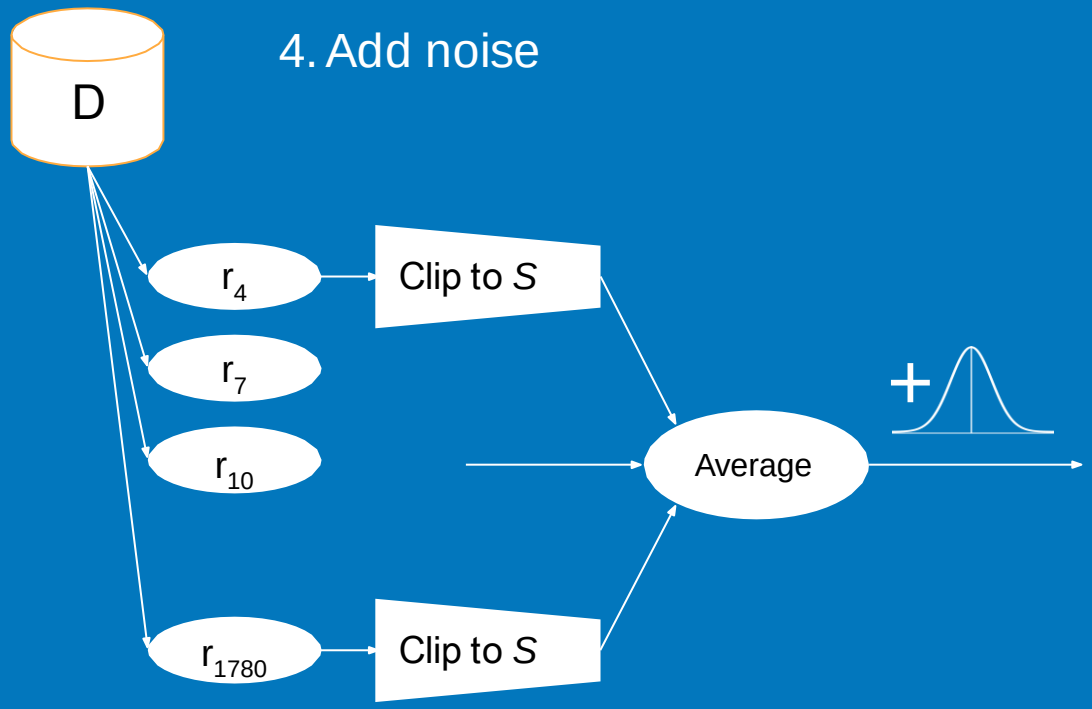
Iterative training with differential privacy



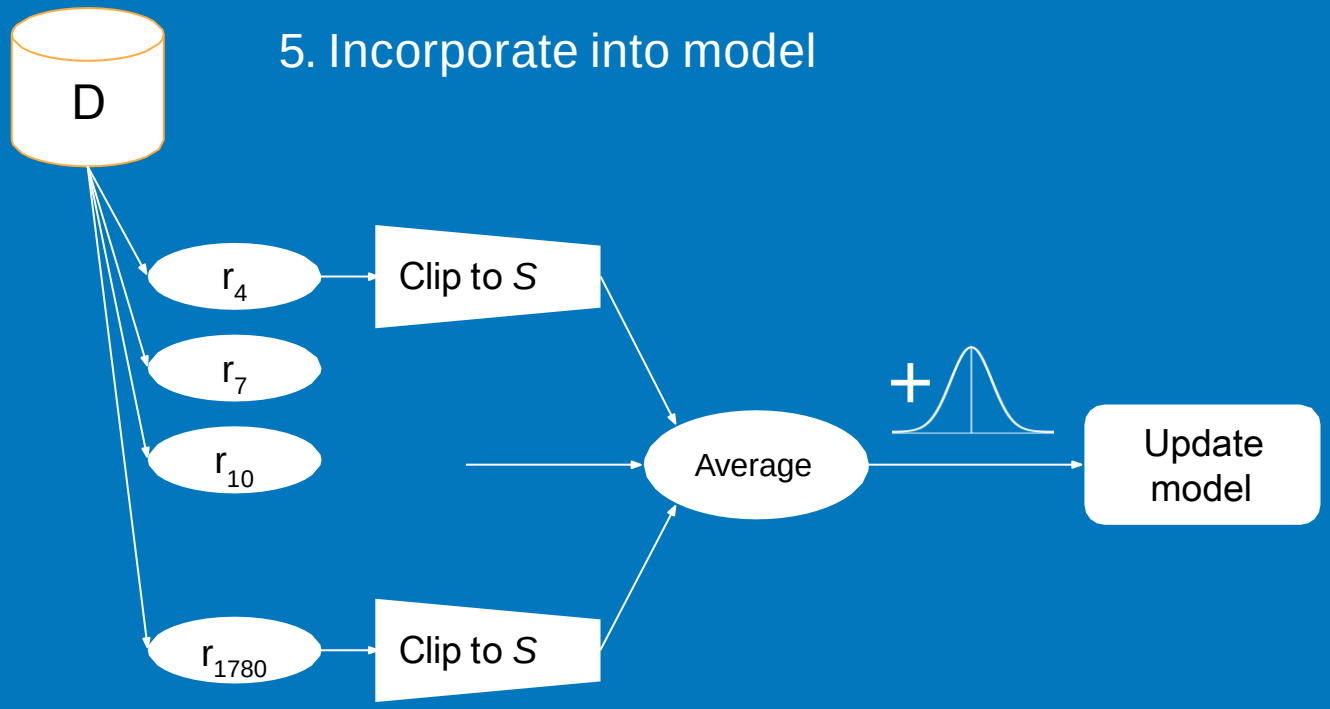
Iterative training with differential privacy



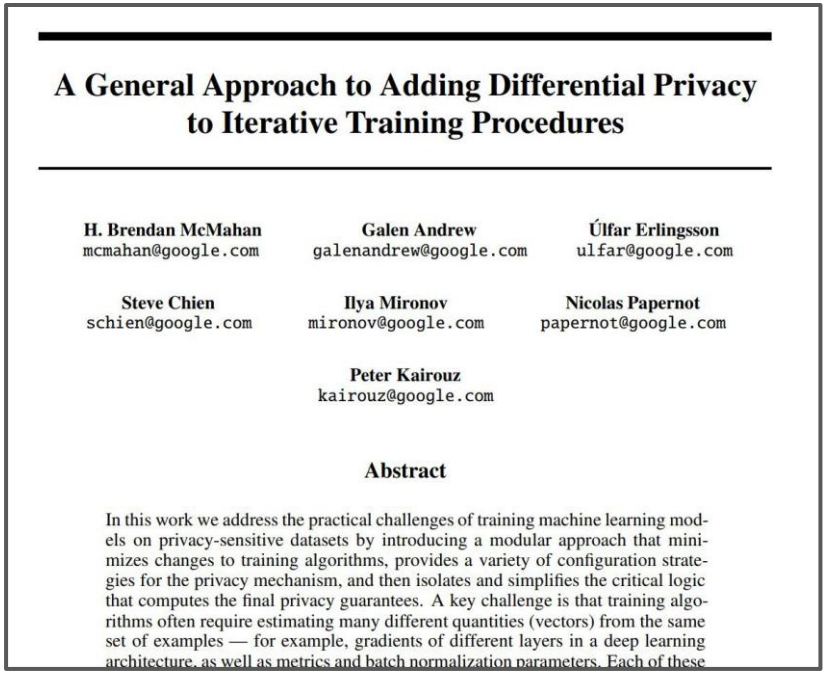
Iterative training with differential privacy



Iterative training with differential privacy



There are many details and possibilities



A General Approach to Adding Differential Privacy to Iterative Training Procedures

H. Brendan McMahan
mcmahan@google.com

Galen Andrew
galenandrew@google.com

Úlfar Erlingsson
ulfar@google.com

Steve Chien
schien@google.com

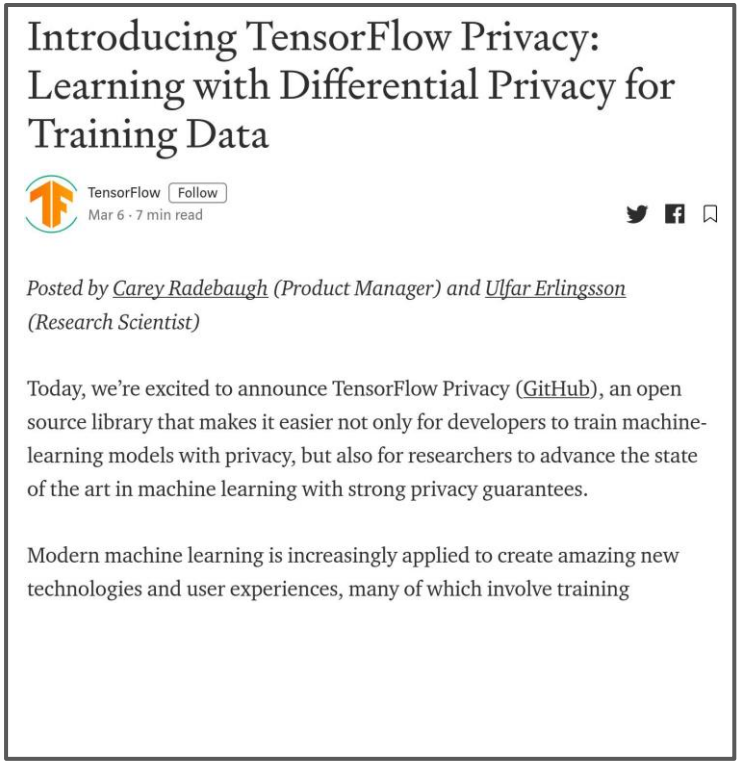
Ilya Mironov
mironov@google.com

Nicolas Papernot
papernot@google.com

Peter Kairouz
kairouz@google.com

Abstract

In this work we address the practical challenges of training machine learning models on privacy-sensitive datasets by introducing a modular approach that minimizes changes to training algorithms, provides a variety of configuration strategies for the privacy mechanism, and then isolates and simplifies the critical logic that computes the final privacy guarantees. A key challenge is that training algorithms often require estimating many different quantities (vectors) from the same set of examples — for example, gradients of different layers in a deep learning architecture, as well as metrics and batch normalization parameters. Each of these



Introducing TensorFlow Privacy: Learning with Differential Privacy for Training Data

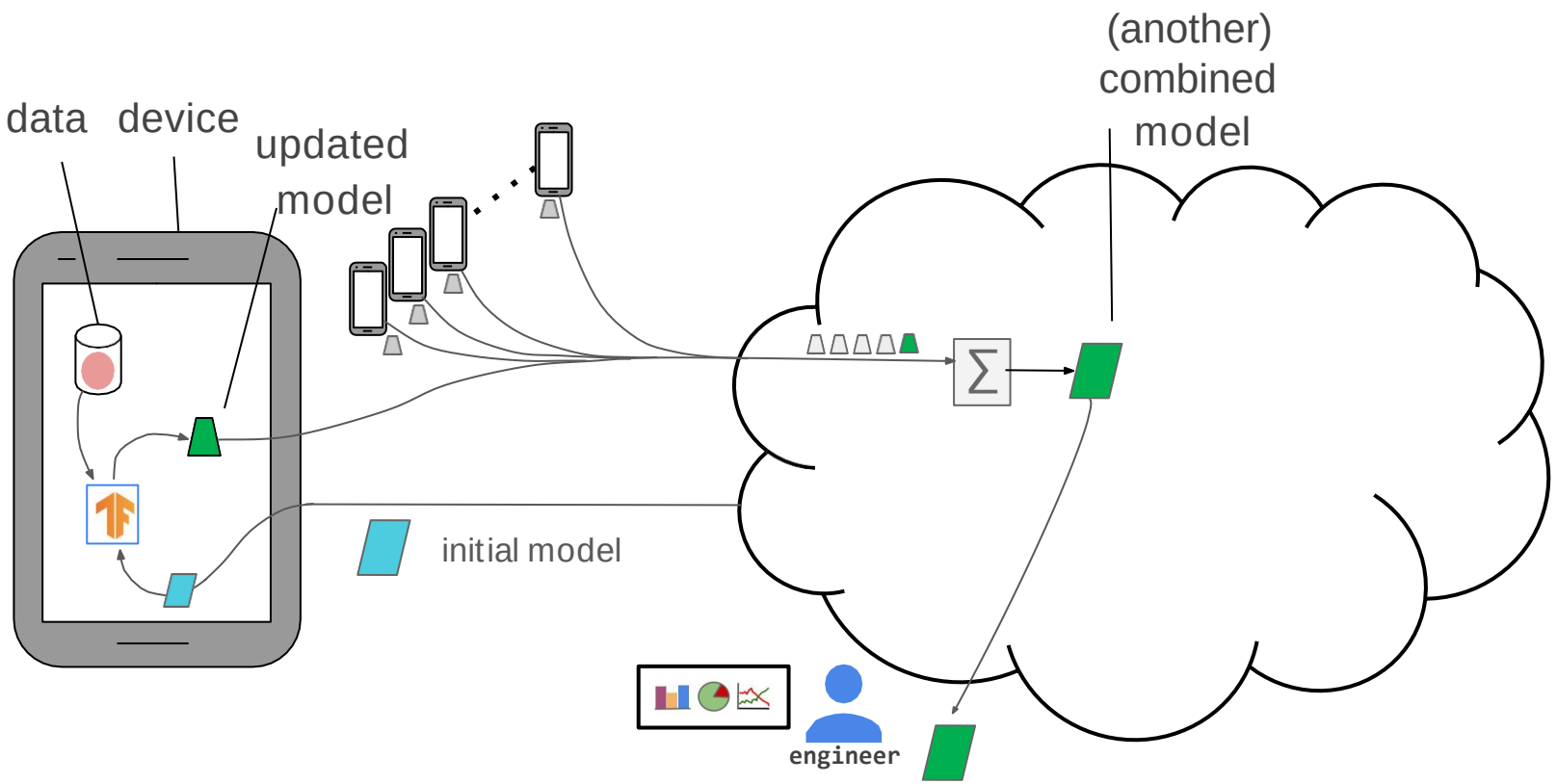
TensorFlow Follow
Mar 6 · 7 min read

Posted by Carey Radebaugh (Product Manager) and Ulfar Erlingsson (Research Scientist)

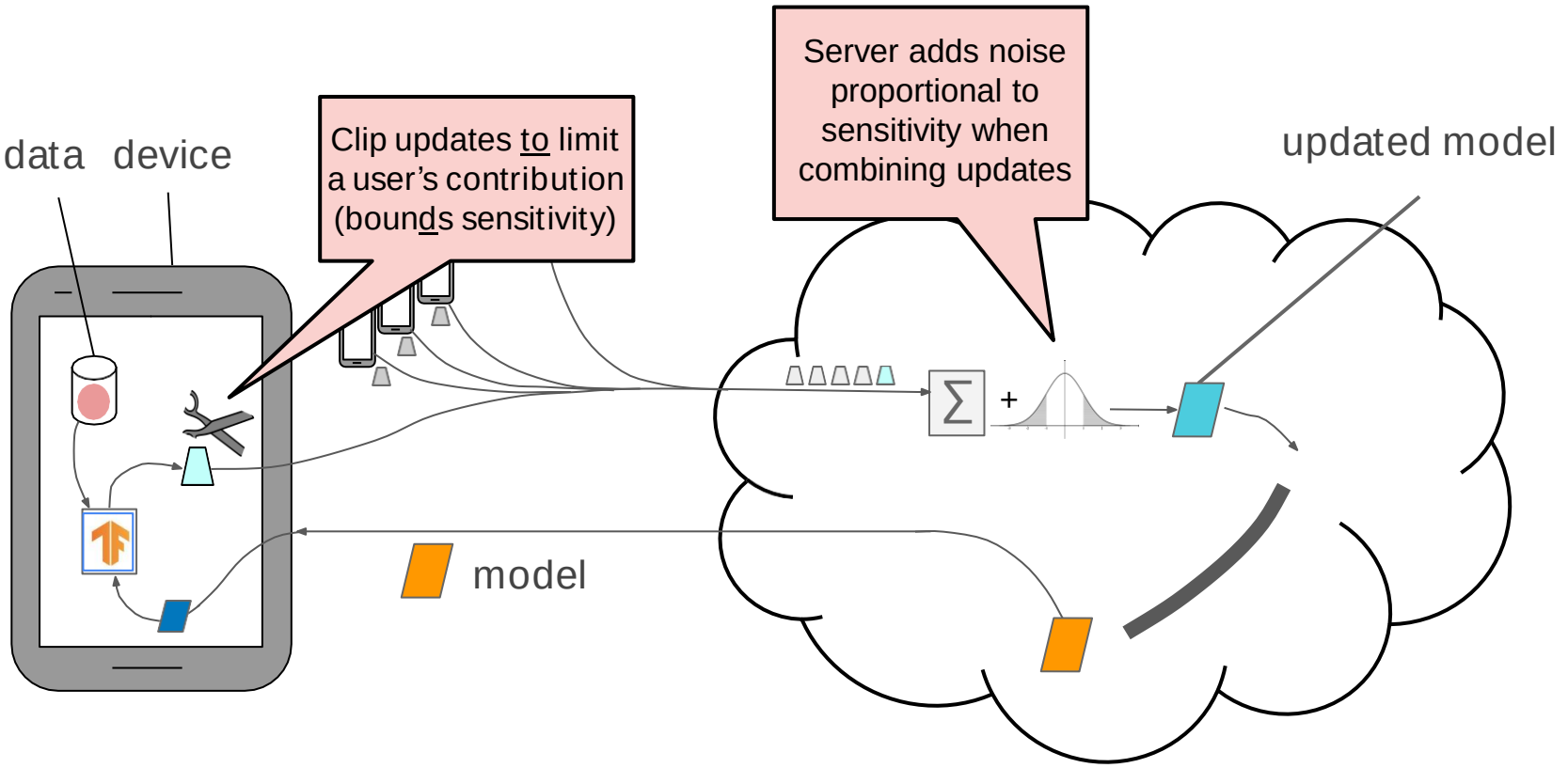
Today, we're excited to announce TensorFlow Privacy (GitHub), an open source library that makes it easier not only for developers to train machine-learning models with privacy, but also for researchers to advance the state of the art in machine learning with strong privacy guarantees.

Modern machine learning is increasingly applied to create amazing new technologies and user experiences, many of which involve training

Back to federated learning

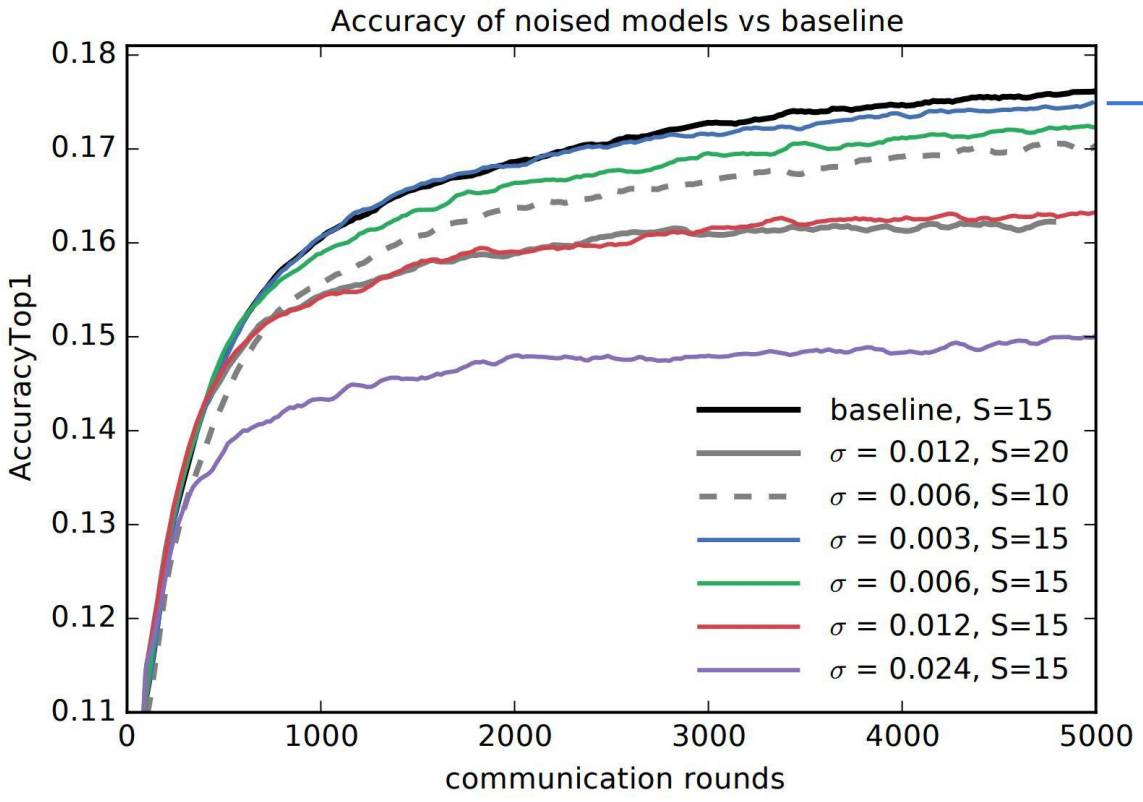


Differentially private federated learning



Differential privacy for language models

LSTM-based predictive language model.
10K word dictionary, word embeddings $\in \mathbb{R}^{96}$, state $\in \mathbb{R}^{256}$, parameters: 1.35M. Corpus=Reddit posts, by author.

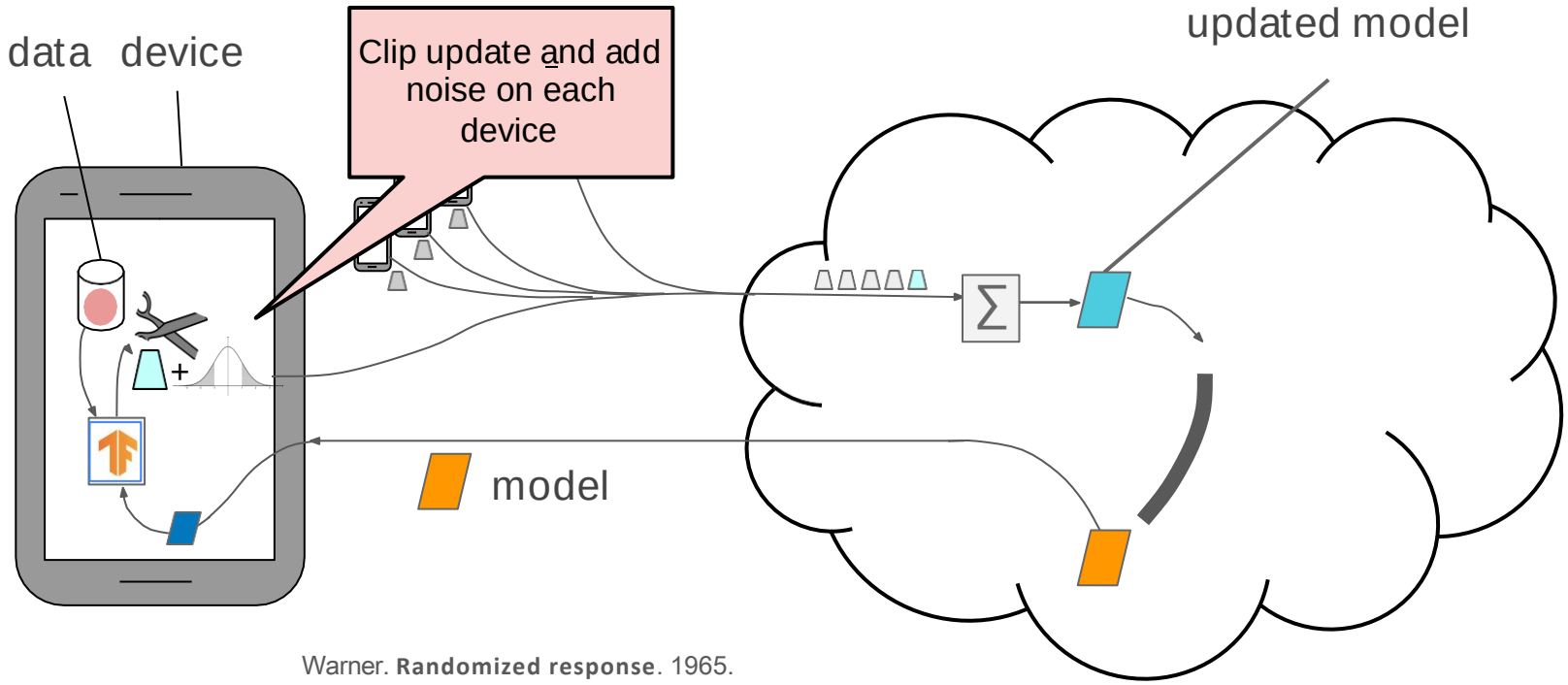


(4.634, 1e-9)-DP with 763k users
(1.152, 1e-9)-DP with 1e8 users

$\mathbb{E}$[users per minibatch] = 5k
$\mathbb{E}$[tokens per minibatch] = 8m

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

Locally differentially private federated learning



Warner. **Randomized response**. 1965.
Kasiviswanathan, et. al. **What can we learn privately?** 2011.

Central DP:

easier to get high utility with good privacy

Local DP:

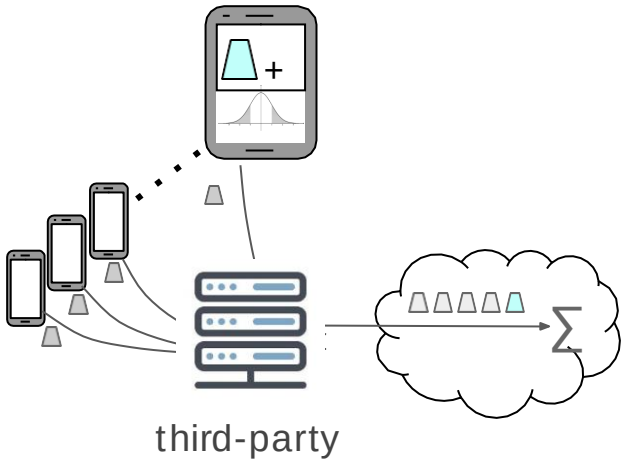
requires much weaker trust assumptions

Can we combine the best of both worlds?

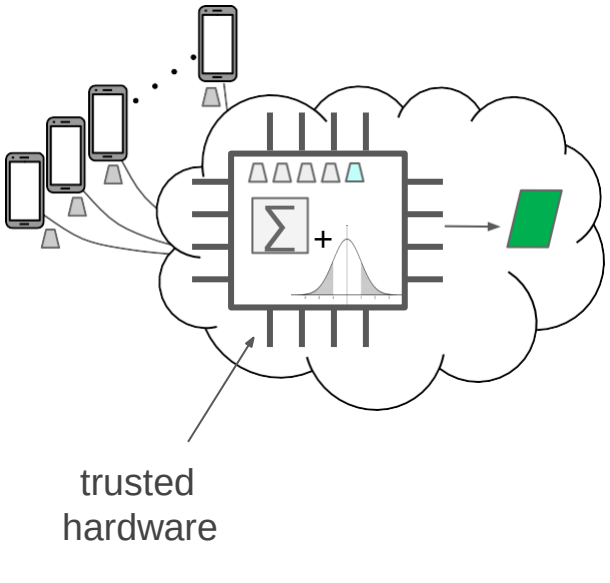
Distributed Differential Privacy

Distributing Trust for Private Aggregation

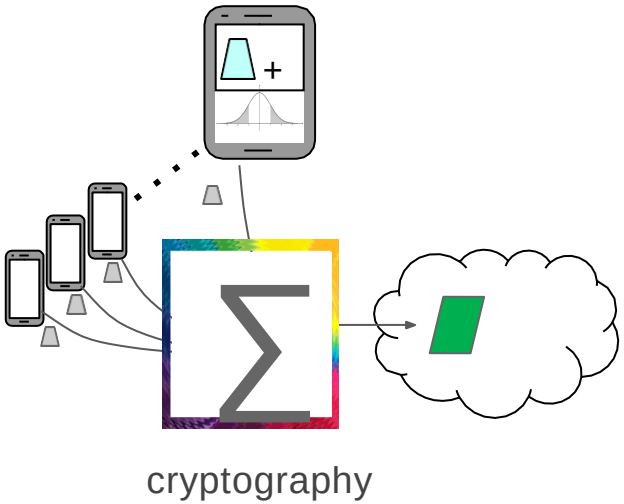
1 Trusted "third party"



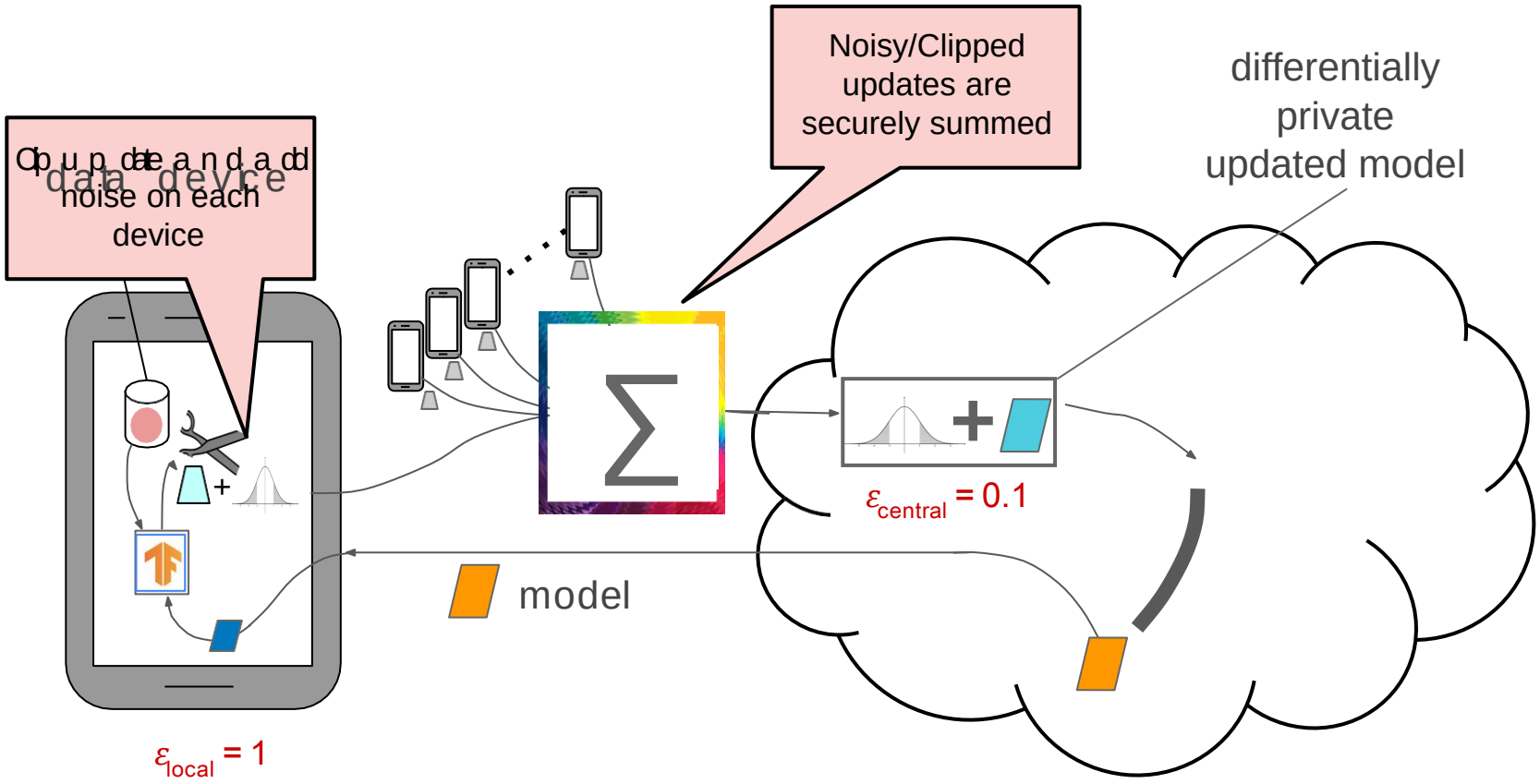
2 Trusted Execution Environments



3 Trust via Cryptography



Distributed DP via secure aggregation



Distributed DP via secure aggregation

Challenges faced

- SecAgg operates on a finite group (finite precision) with modulo arithmetic
- Discrete Gaussian random variables are not closed under summation
- Discrete distributions with finite tails lead to catastrophic privacy failures
- Tight DP accounting needs to be fundamentally rederived

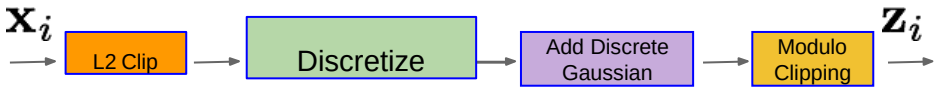
Solutions needed

- A family of discrete mechanisms that mesh well with SecAgg's modulo arithmetic
- Closed under summation or have tractable distributions upon summation
- Can be sampled from exactly and efficiently using random bits
- Exact DP guarantees with tight accounting and no catastrophic failures

The distributed discrete Gaussian mechanism

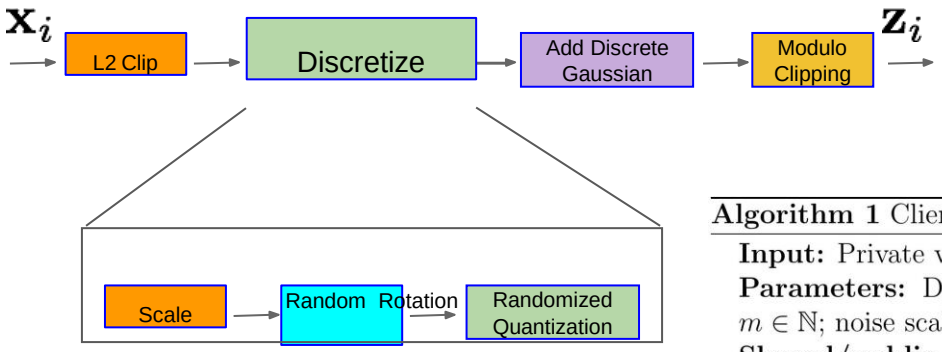
The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation (on arXiv)

The distributed discrete Gaussian mechanism



The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation (on arXiv)

The distributed discrete Gaussian mechanism



Algorithm 1 Client Procedure $\mathcal{A}_{\text{client}}$

Input: Private vector $x_i \in \mathbb{R}^d$. {Assume dimension d is a power of 2.}

Parameters: Dimension $d \in \mathbb{N}$; clipping threshold $c > 0$; granularity $\gamma > 0$; modulus $m \in \mathbb{N}$; noise scale $\sigma > 0$; bias $\beta \in [0, 1)$.

Shared/public randomness: Uniformly random sign vector $\xi \in \{-1, +1\}^d$.

Clip and rescale vector: $x_i' = \frac{1}{\gamma} \min\left\{1, \frac{c}{\|x_i\|_2}\right\} \cdot x_i \in \mathbb{R}^d$.

Flatten vector: $x_i'' = H_d D_\xi x_i' \in \mathbb{R}^d$ where $H \in \{-1/\sqrt{d}, +1/\sqrt{d}\}^{d \times d}$ is a Walsh-Hadamard matrix satisfying $H^T H = I$ and $D_\xi \in \{-1, 0, +1\}^{d \times d}$ is a diagonal matrix with ξ on the diagonal.

repeat

Let $\tilde{x}_i \in \mathbb{Z}^d$ be a randomized rounding of $x_i'' \in \mathbb{R}^d$. I.e., $\tilde{x}_i$ is a product distribution with $\mathbb{E}[\tilde{x}_i] = x_i''$ and $\|\tilde{x}_i - x_i''\|_\infty < 1$.

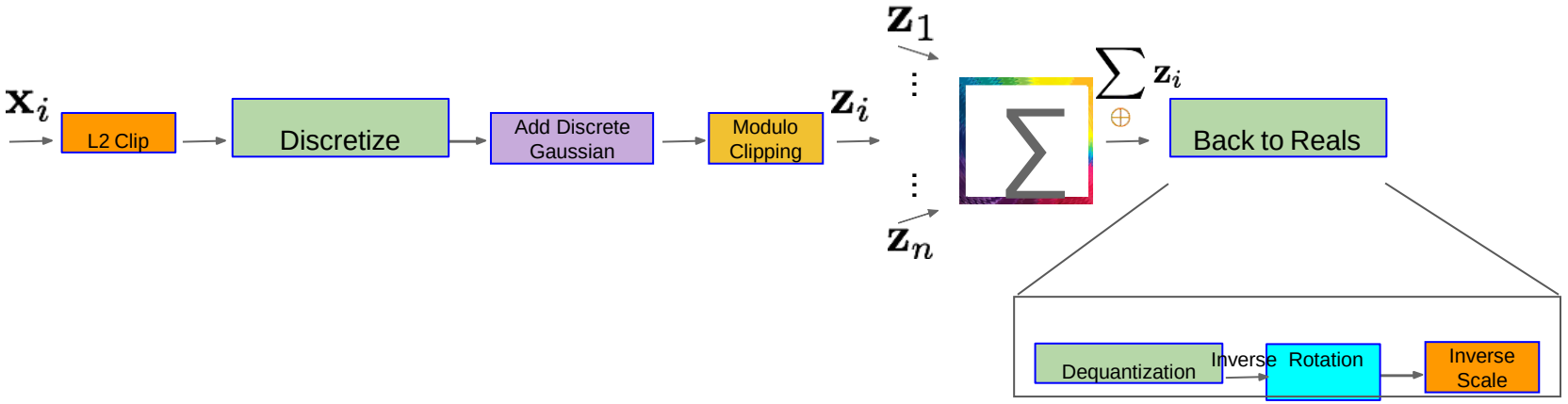
until $\|\tilde{x}_i\|_2 \leq \min\left\{c/\gamma + \sqrt{d}, \sqrt{c^2/\gamma^2 + \frac{1}{4}d + \sqrt{2\log(1/\beta)} \cdot \left(c/\gamma + \frac{1}{2}\sqrt{d}\right)}\right\}$.

Let $y_i \in \mathbb{Z}^d$ consist of d independent samples from the discrete Gaussian $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2/\gamma^2)$.

Let $z_i = (\tilde{x}_i + y_i) \mod m$.

Output: $z_i \in \mathbb{Z}_m^d$ is returned via secure aggregation protocol.

The distributed discrete Gaussian mechanism



Algorithm 2 Server Procedure $\mathcal{A}_{\text{server}}$

Input: Vector $\bar{z} = (\sum_i^n z_i \mod m) \in \mathbb{Z}_m^d$ via secure aggregation.

Parameters: Dimension $d \in \mathbb{N}$; number of clients $n \in \mathbb{N}$; clipping threshold $c > 0$; granularity $\gamma > 0$; modulus $m \in \mathbb{N}$; noise scale $\sigma > 0$; bias $\beta \in [0, 1)$.

Shared/public randomness: Uniformly random sign vector $\xi \in \{-1, +1\}^d$.

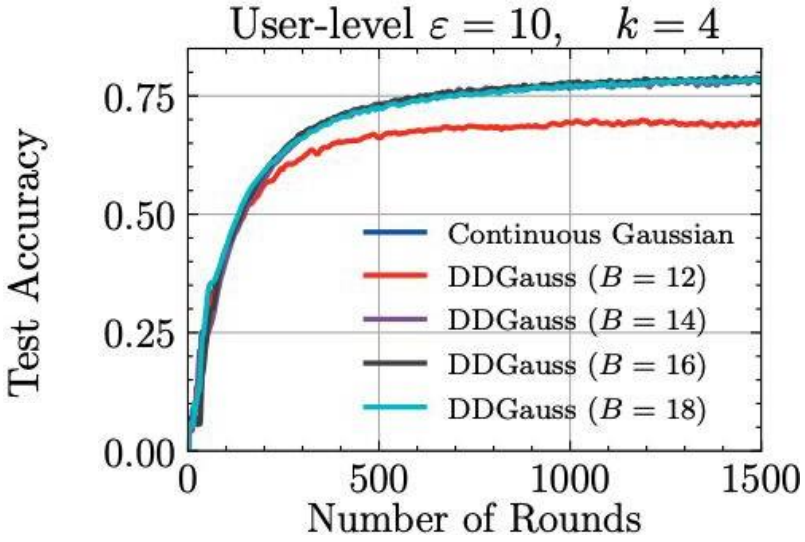
Map $\mathbb{Z}_m$ to $\{1 - m/2, 2 - m/2, \cdots, -1, 0, 1 \cdots, m/2 - 1, m/2\}$ so that $\bar{z}$ is mapped to $\bar{z}' \in [-m/2, m/2]^d \cap \mathbb{Z}^d$ (and we have $\bar{z}' \mod m = \bar{z}$).

Output: $y = \gamma D_\xi H_d^T \bar{z}' \in \mathbb{R}^d$. {Goal: $y \approx \bar{x} = \sum_i^n x_i$}

Federated EMNIST Classification

- Classifying handwritten digits/letters grouped by their writers
- Total writers/clients = 3400, number of clients per round = 100
- 671,585 training examples, 62 classes, model size = 1M parameters

Centralized Continuous Gaussian
Distributed Discrete Gaussian (18 bits)
Distributed Discrete Gaussian (16 bits)
Distributed Discrete Gaussian (14 bits)
Distributed Discrete Gaussian (12 bits)

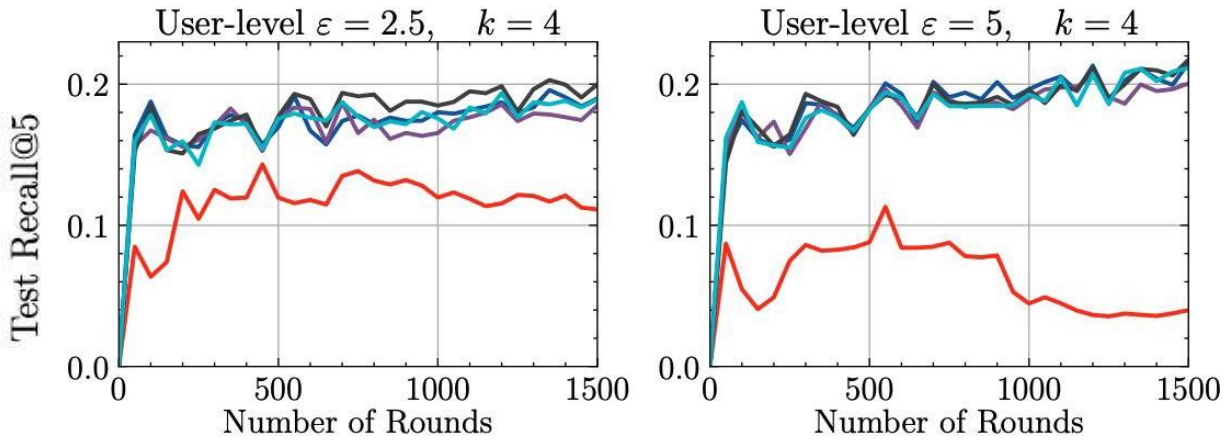


The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation (on arXiv)

StackOverflow Tag Prediction

- Predicting the tags of the sentences on Stack Overflow with Logistic Regression
- Total users/clients = 342477, number of clients per round = 60
- Tags vocab size = 500, Tokens vocab size = 10000, model size = 5M parameters

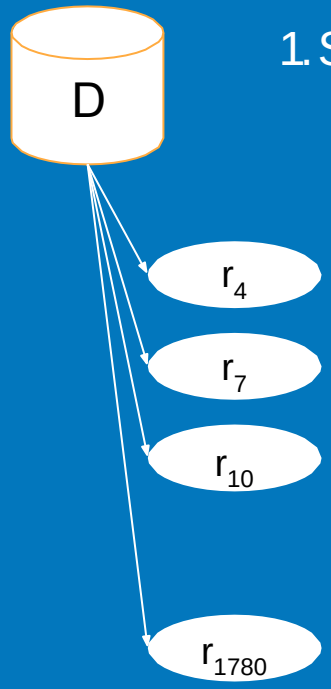
Centralized Continuous Gaussian
Distributed Discrete Gaussian (18 bits)
Distributed Discrete Gaussian (16 bits)
Distributed Discrete Gaussian (14 bits)
Distributed Discrete Gaussian (12 bits)



The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation (on arXiv)

Precise DP Guarantees for Real-World Cross-Device FL

Iterative training with differential privacy



1. Sample a batch of clients ***uniformly at random***

Challenges

- There is no fixed or known database / dataset / population size
- Client availability is dynamic due to multiple system layers and participation constraints
  - "Sample from the population" or "shuffle devices" don't work out-of-the-box
- Clients may drop out at any point of the protocol, with possible impacts on privacy and utility

For **privacy** purposes, model the environment (availability, dropout) as the choices of *Nature* (possibly malicious and adaptive to previous mechanism choices)

Goals

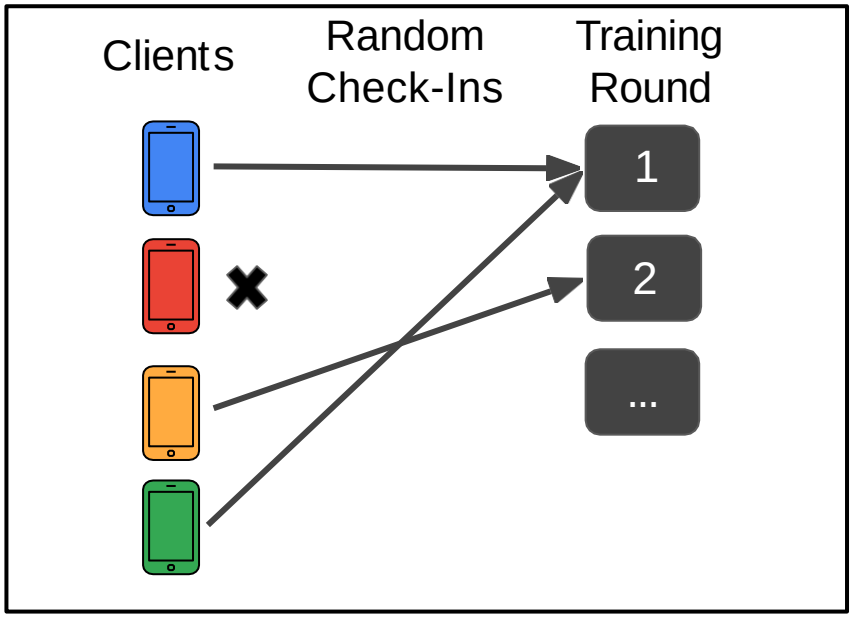
- **Robust** to Nature's choices (client availability, client dropout) in that privacy and utility are both preserved, possibly at the expense of forward progress.
- **Self-accounting**, in that the server can compute a precise upper bound on the (ε,δ) of the mechanism using only information available via the protocol.
- **Local selection**, so most participation decisions are made locally, and as few devices as possible check-in to the server
- **Good privacy vs. utility tradeoffs**

Algorithm 2 Protocol schema for DP in Cross-Device FL

$\theta_0 =$ (initialization)
for each protocol step $t = 1, 2, \dots$ **do**
  Nature (maybe malicious, adaptive) chooses $C^{\text{AVAILABLE}} \subseteq C^{\text{POPULTATION}}$
  # Clients in $C_t^{\text{AVAILABLE}}$ decide locally whether to check in to the server
  $C_t^{\text{CHECKEDIN}} = \{u \mid \texttt{ShouldCheckIn}(u, t, \dots) = 1, u \in C_t^{\text{AVAILABLE}}\}$
  $C_t^{\text{SELECTED}} = \texttt{SelectFrom}(C_t^{\text{CHECKEDIN}}, \dots)$
  Nature chooses the clients $C_t^{\text{REPORTED}} \subseteq C_t^{\text{SELECTED}}$ that report
  $X_t = \texttt{Aggregate}(\{\texttt{LocalUpdate}(u, \theta_t) \mid u \in C_t^{\text{REPORTED}}\})$
  # DP should allow the release of X_t
  $\theta_{t+1} = \texttt{ServerUpdate}(\theta_t, X_t)$
  Server outputs θ_{t+1} and (ϵ, δ)

`ShouldCheckIn`	Server	Runs locally on client, decides to connect to server
`SelectFrom`	Client	Selects devices to participate from CHECKEDIN
`LocalUpdate`	Client	Computes the value to report to the server
`Aggregate`	Server	Aggregates updates to produce DP output
`ServerUpdate`	Server	Update server state (DP post processing)

Random Check-ins



Privacy Amplification via Random Check-Ins

Borja Balle* Peter Kairouz† H. Brendan McMahan† Om Thakkar†
Abhradeep Thakurta‡

July 15, 2020

Abstract

Differentially Private Stochastic Gradient Descent (DP-SGD) forms a fundamental building block in many applications for learning over sensitive data. Two standard approaches, privacy amplification by subsampling, and privacy amplification by shuffling, permit adding lower noise in DP-SGD than via naïve schemes. A key assumption in both these approaches is that the elements in the data set can be uniformly sampled, or be uniformly permuted — constraints that may become prohibitive when the data is processed in a decentralized or distributed fashion. In this paper, we focus on conducting iterative methods like DP-SGD in the setting of federated learning (FL) wherein the data is distributed among many devices (clients). Our main contribution is the *random check-in* distributed protocol, which crucially relies only on randomized participation decisions made locally and independently by each client. It has privacy/accuracy trade-offs similar to privacy amplification by subsampling/shuffling. However, our method does not require server-initiated communication, or even knowledge of the population size. To our knowledge, this is the first privacy amplification tailored for a distributed learning framework, and it may have broader applicability beyond FL. Along the way, we extend privacy amplification by shuffling to incorporate (ε, δ)-DP local randomizers, and exponentially improve its guarantees. In practical regimes, this improvement allows for similar privacy and utility using data from an order of magnitude fewer users.

1 Introduction

Modern mobile devices and web services benefit significantly from large-scale machine learning, often involving training on user (client) data. When such data is sensitive, steps must be taken to ensure privacy, and a formal guarantee of differential privacy (DP) [15, 16] is the gold standard. For this reason, DP has been adopted by companies including Google [9, 18, 20], Apple [2], Microsoft [13], and LinkedIn [31], as well as the US Census Bureau [26].

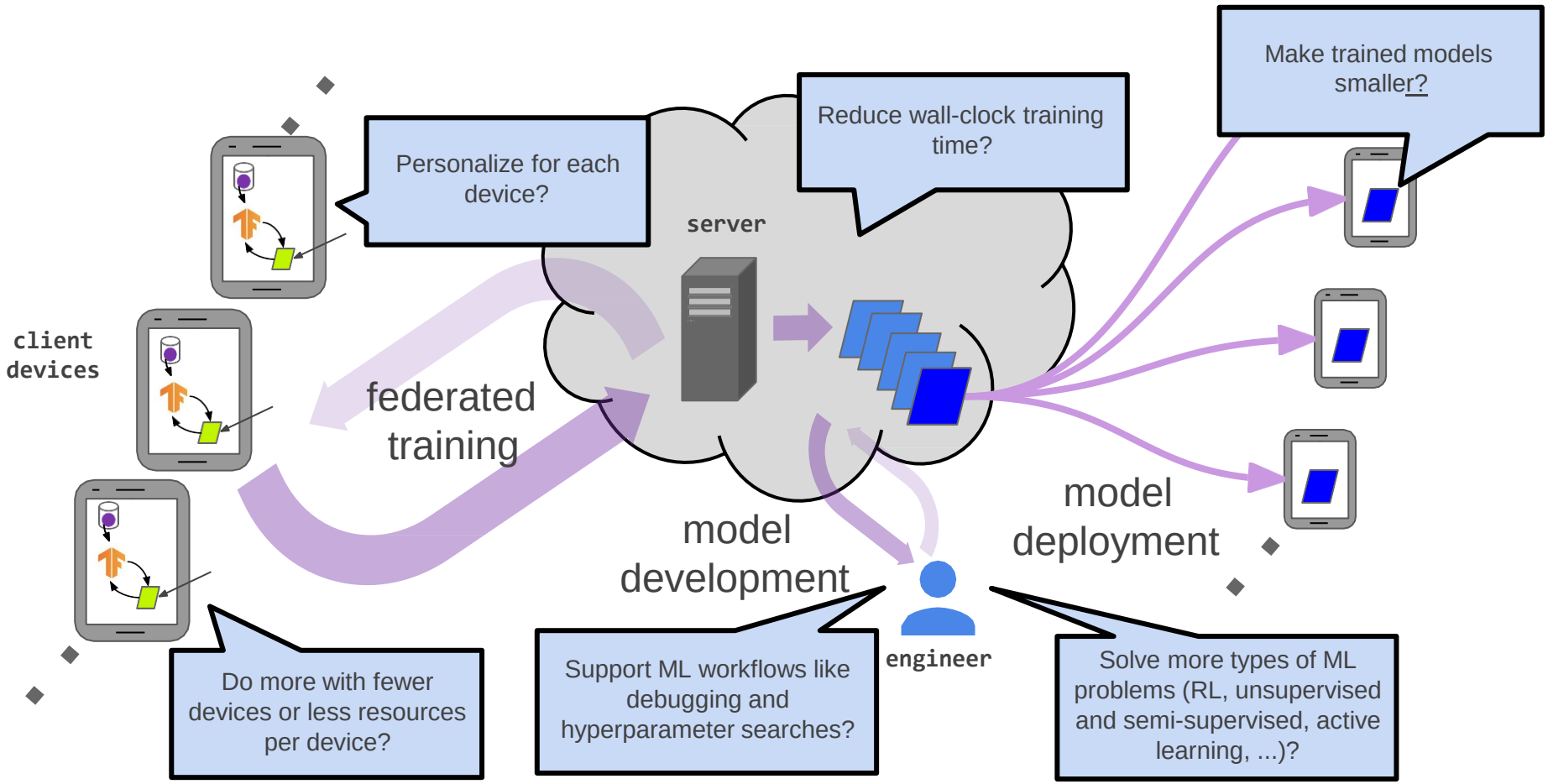
Other privacy-enhancing techniques can be combined with DP to obtain additional benefits. In particular, cross-device federated learning (FL) [27] allows model training while keeping client data decentralized (each participating device keeps its own local dataset, and only sends model updates or gradients to the coordinating server). However, existing approaches to combining FL and DP make a number of assumptions that are unrealistic in real-world FL deployments such as [10]. To highlight these challenges, we must first review the state-of-the-art in centralized DP training, where differentially private stochastic gradient descent (DP-SGD) [1, 8, 34] is ubiquitous. It achieves optimal error for convex problems [8], and can also be applied to non-convex problems, including deep learning, where the privacy amplification offered by randomly subsampling data to form batches is critical for obtaining meaningful DP guarantees [1, 5, 8, 25, 37].

Attempts to combine FL and the above lines of DP research have been made previously; notably, [3, 28] extended the approach of [1] to FL and user-level DP. However, these works and others in the area sidestep a critical issue: the DP guarantees require very specific sampling or shuffling schemes assuming, for example, that each client participates in each iteration with a fixed probability. While possible in theory, such schemes are incompatible with the practical

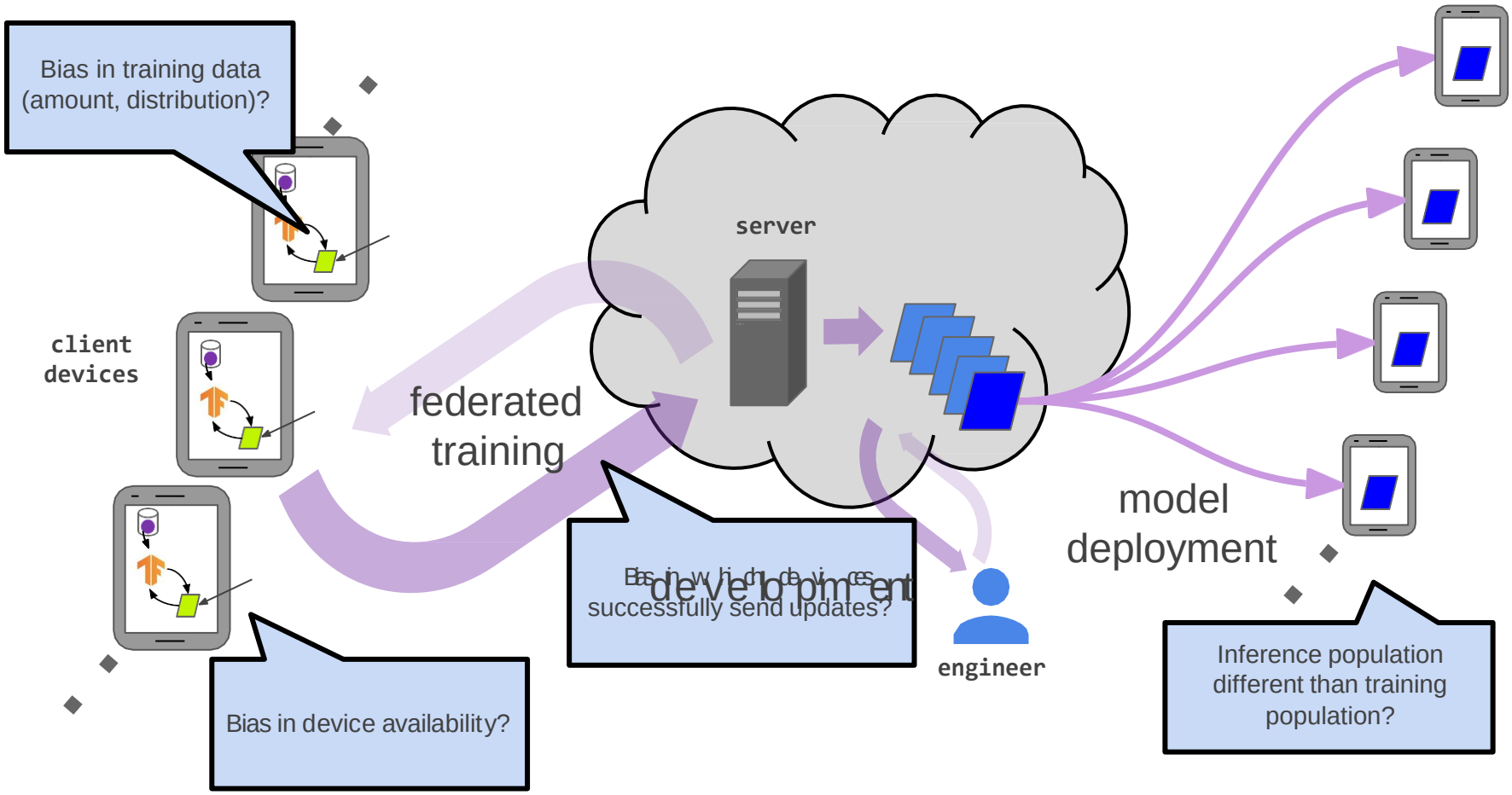
*DeepMind. bballe@google.com
†Google. {kairouz, mcmahan, omthkkr}@google.com
‡Google Research - Brain. {athakurta}@google.com

Part III: Other topics

Improving efficiency and effectiveness



Ensuring fairness and addressing sources of bias



Robustness to attacks and failures