Federated Learning, Diff Privacy, and Generative Models

Sean Augenstein
saugenst@google.com

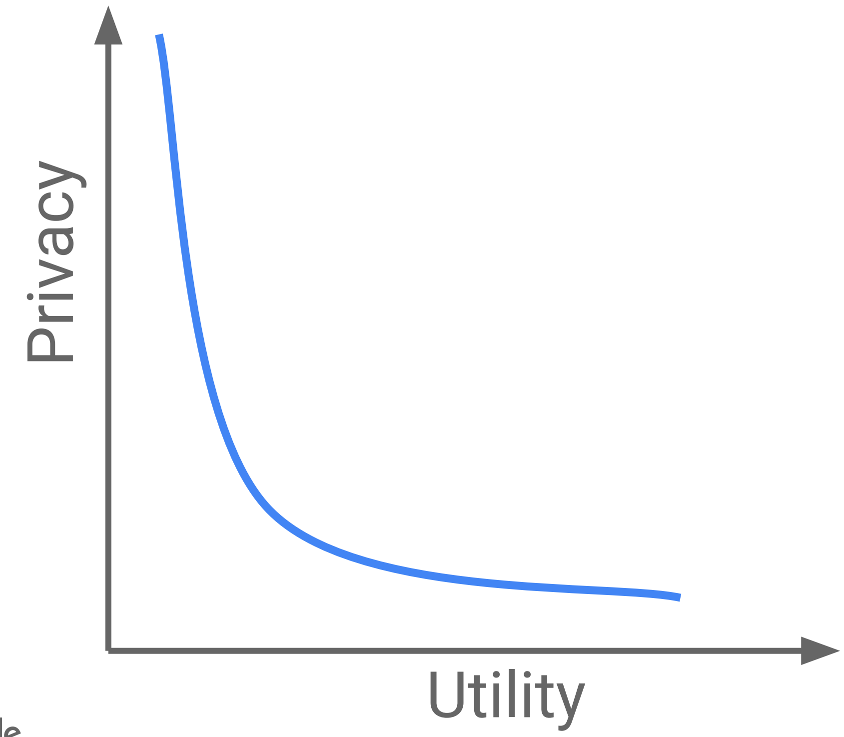
Presenting the work of many

UC Berkeley 2019.09.26

g.co/federated

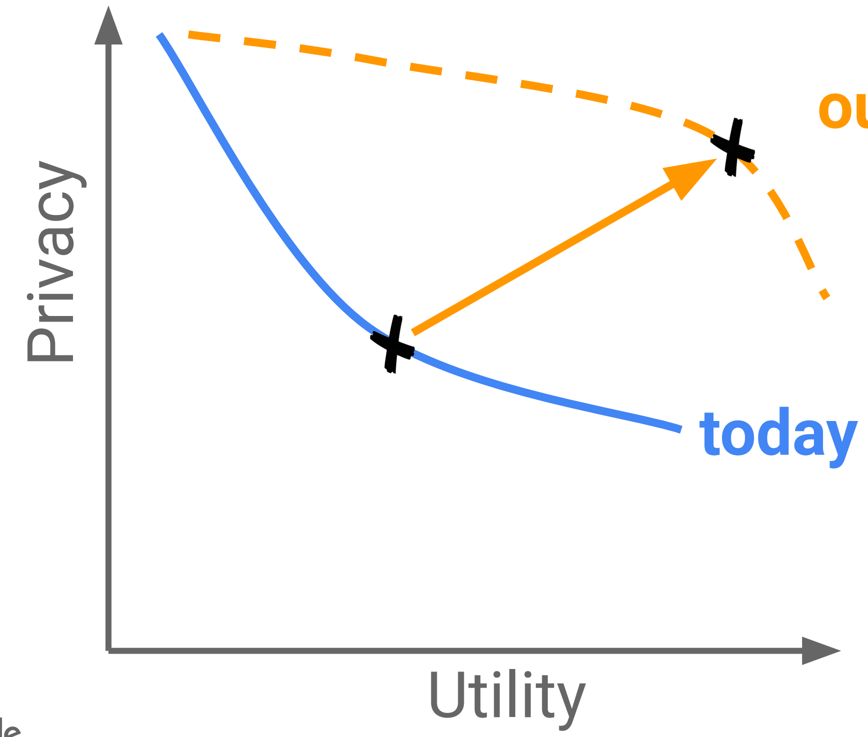
Context

ML on Sensitive Data: Privacy versus Utility



Google

ML on Sensitive Data: Privacy versus Utility (?)



1. Policy
2. **New Technology**

Push the pareto frontier with better technology.

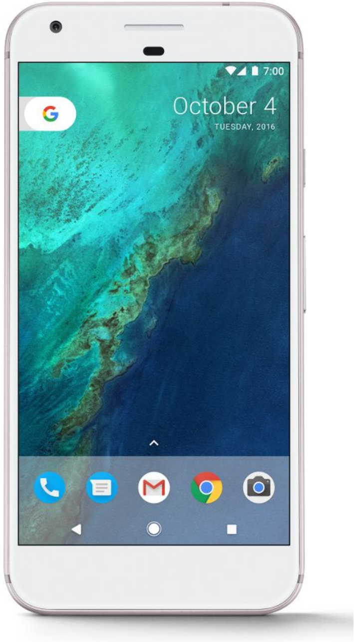
Why federated learning?

Google

Data is born at the edge

Billions of phones & IoT devices constantly generate data

Data enables better products and smarter models

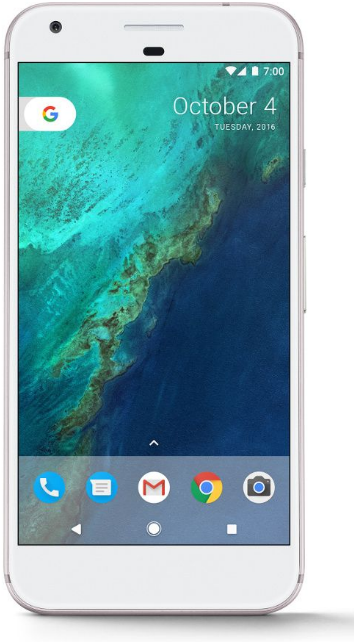


Can data live at the edge?

Data processing is moving on device:

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

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



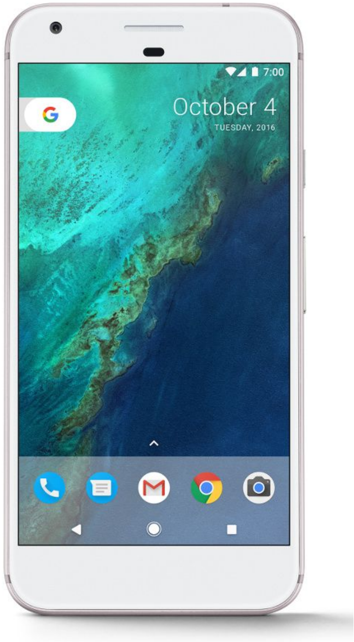
Can data live at the edge?

Data processing is moving on device:

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

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

What about analytics?
What about learning?



2014: Three choices

Don't use data to improve products and services

Log the data centrally *anyway*

Invent a new solution

Choose your poison □

Google

2014: Three choices

Don't use data to improve products and services

Log the data centrally *anyway*

Invent a new solution

Choose your poison □

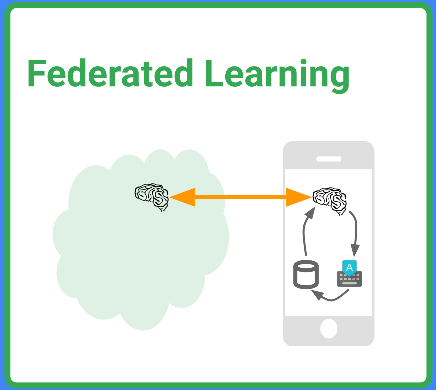
2019: Good reason to hope

Don't use data to improve products and services

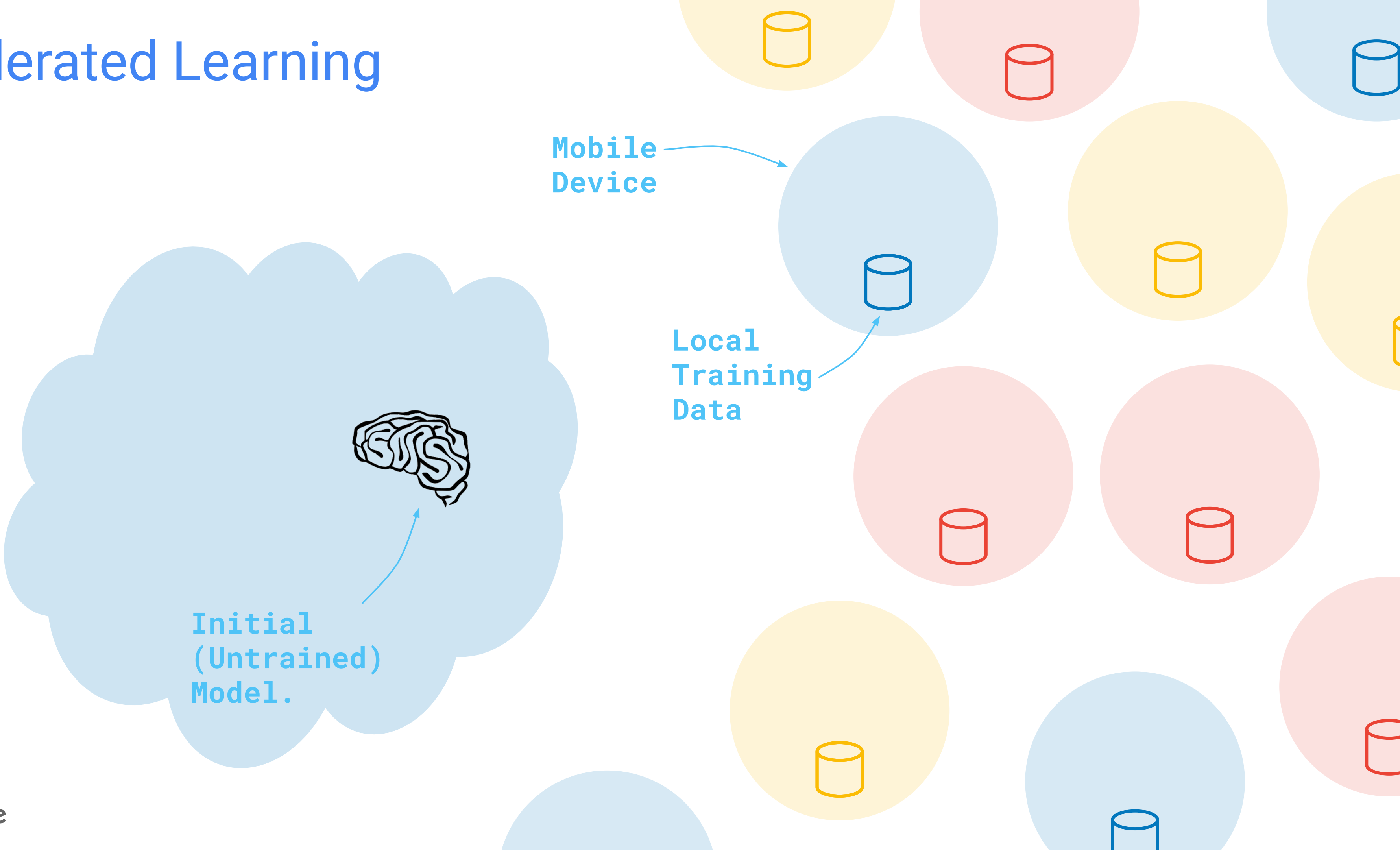
Log the data centrally *anyway*

Federated learning and analytics

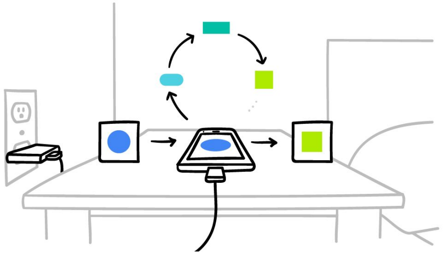
Federated Learning

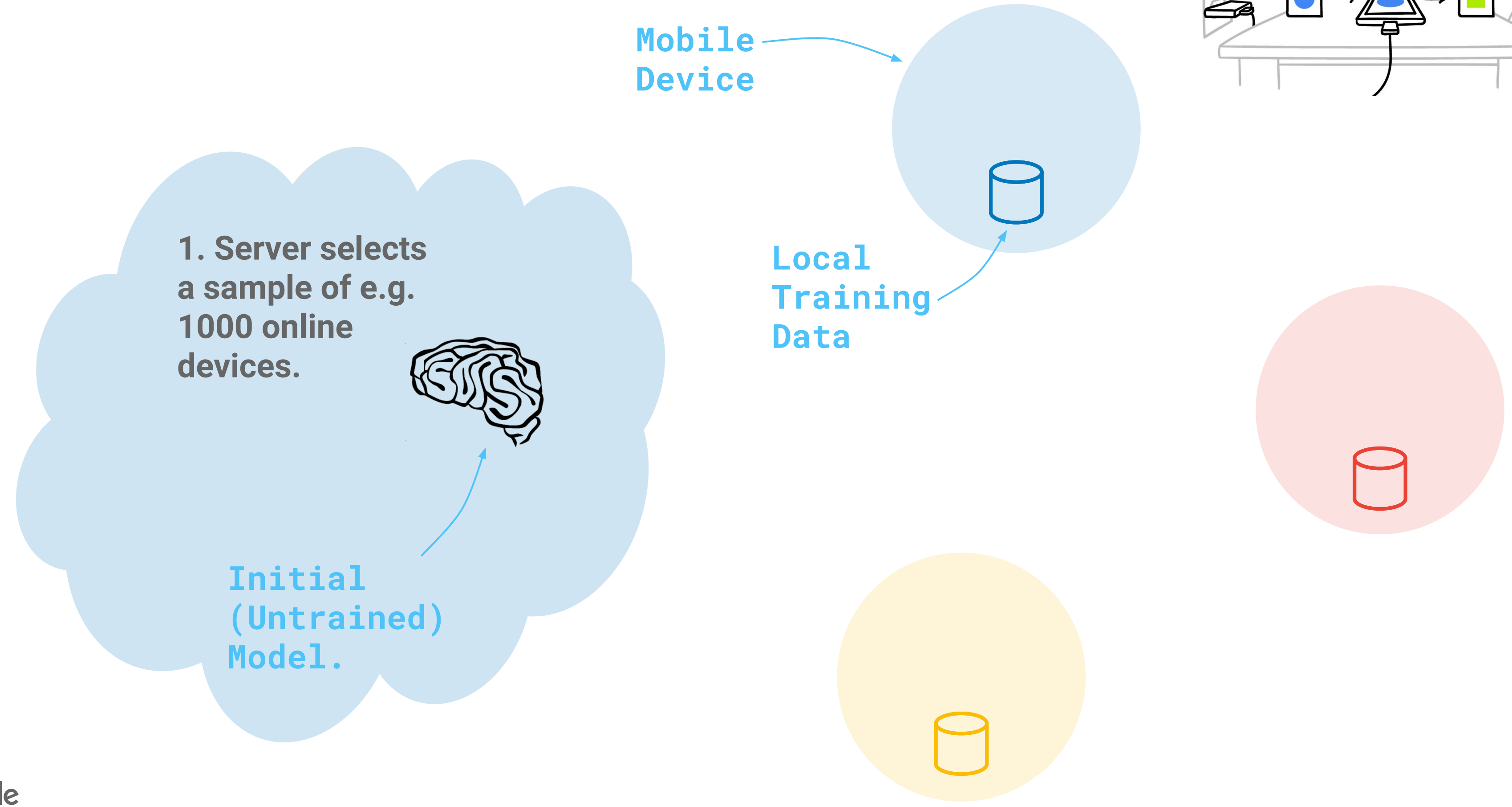


Federated Learning

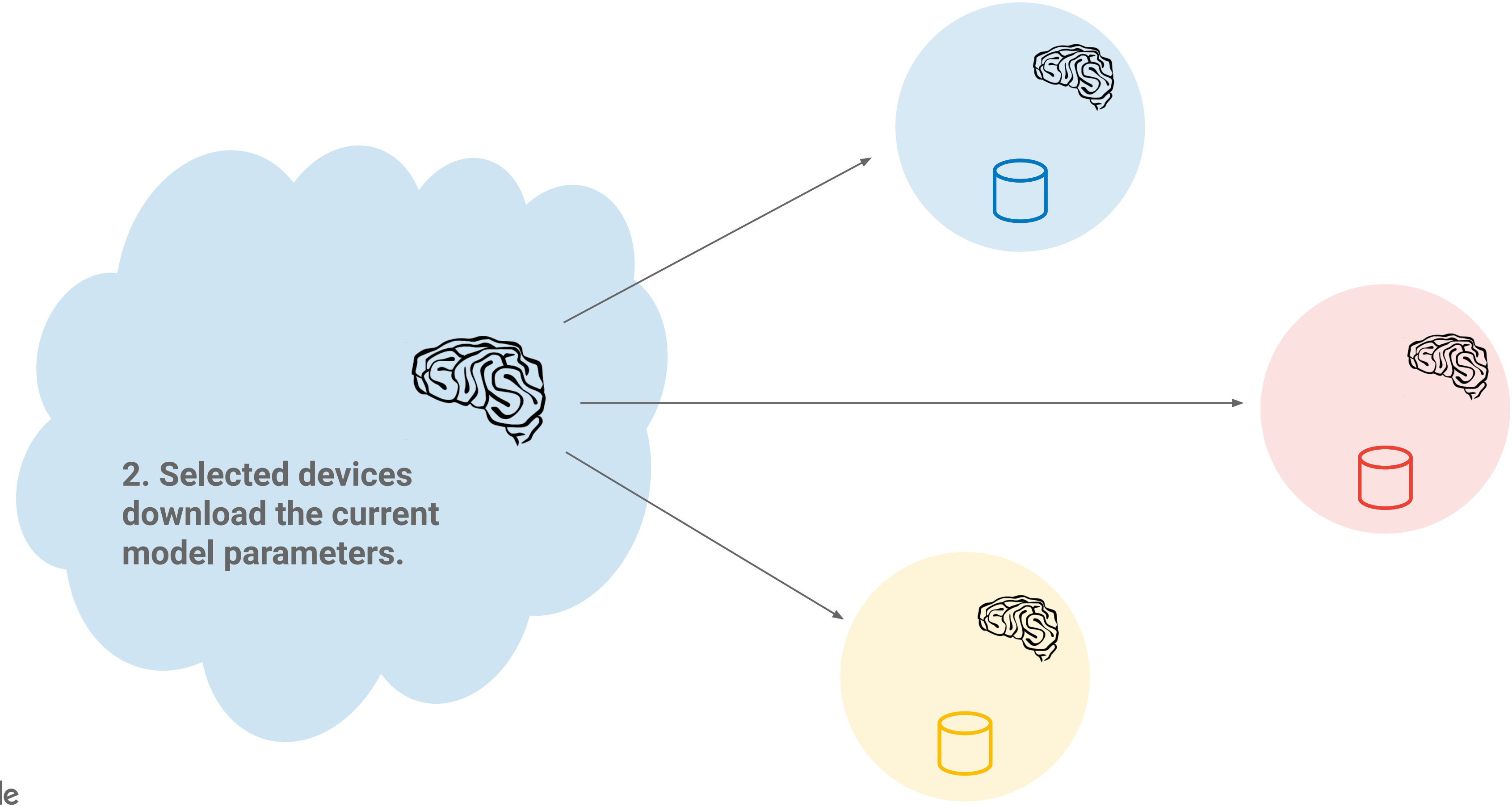


Federated Learning

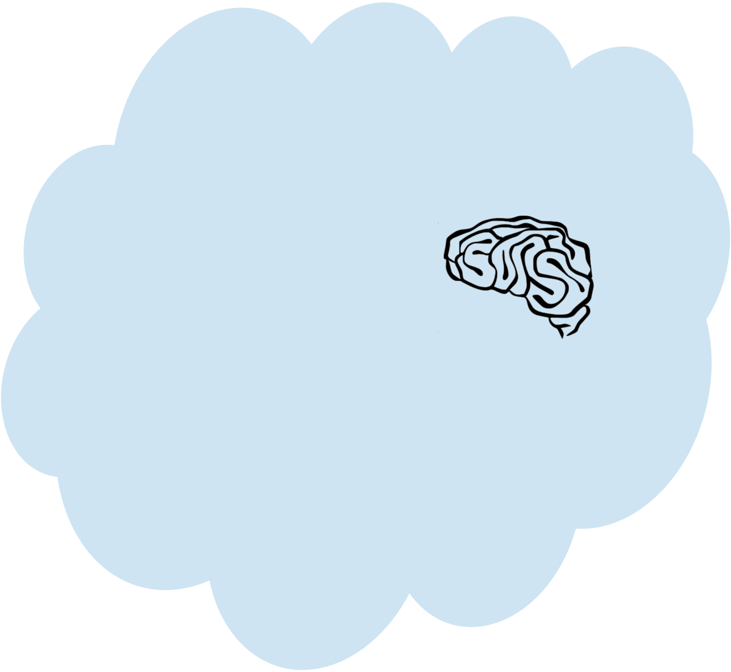




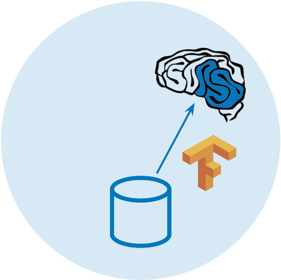
Federated Learning

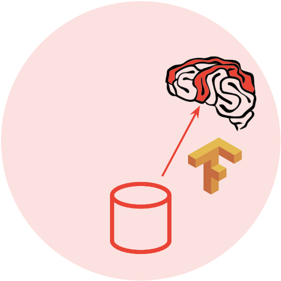


Federated Learning

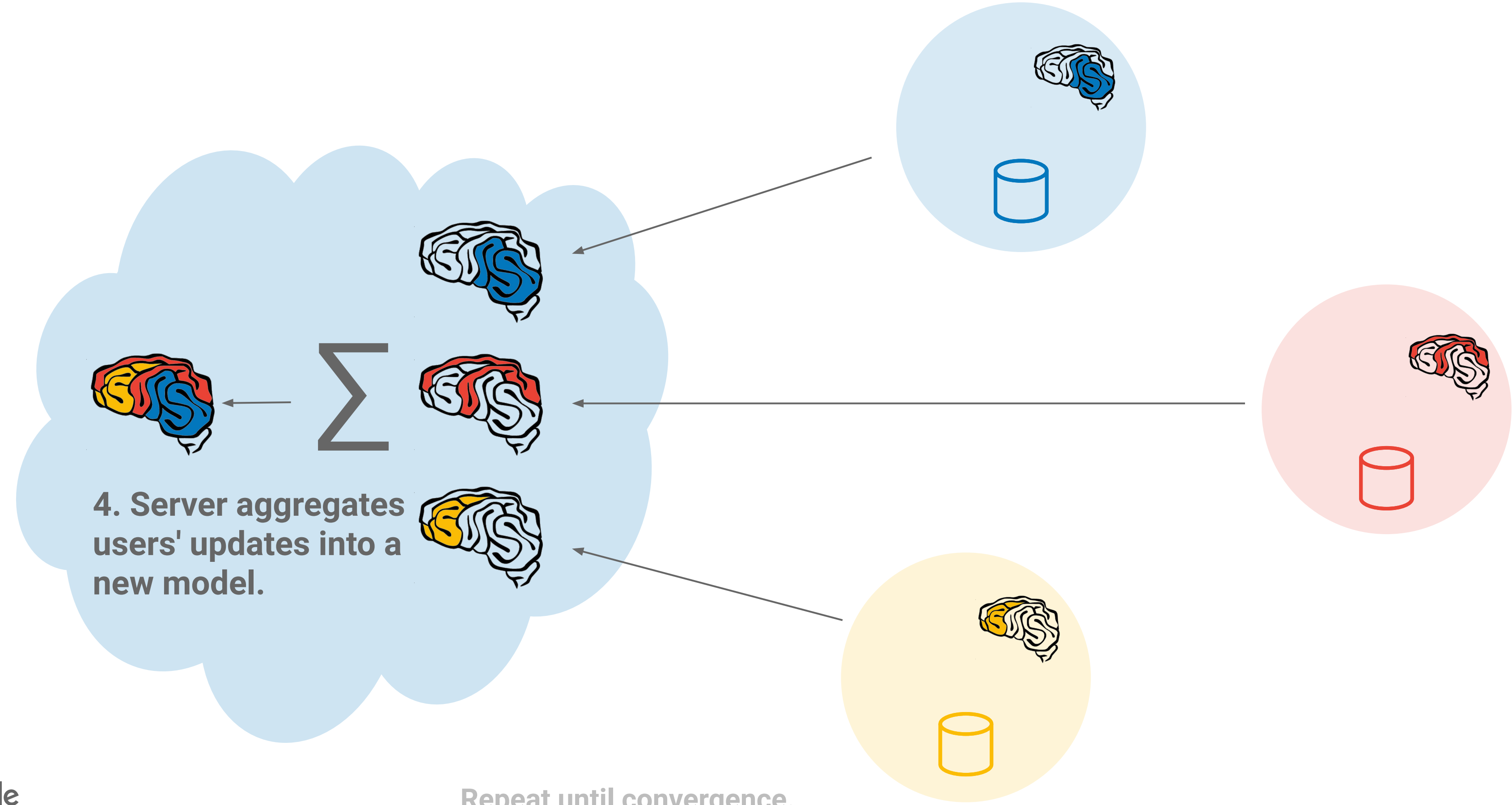


3. Users compute an update using local training data





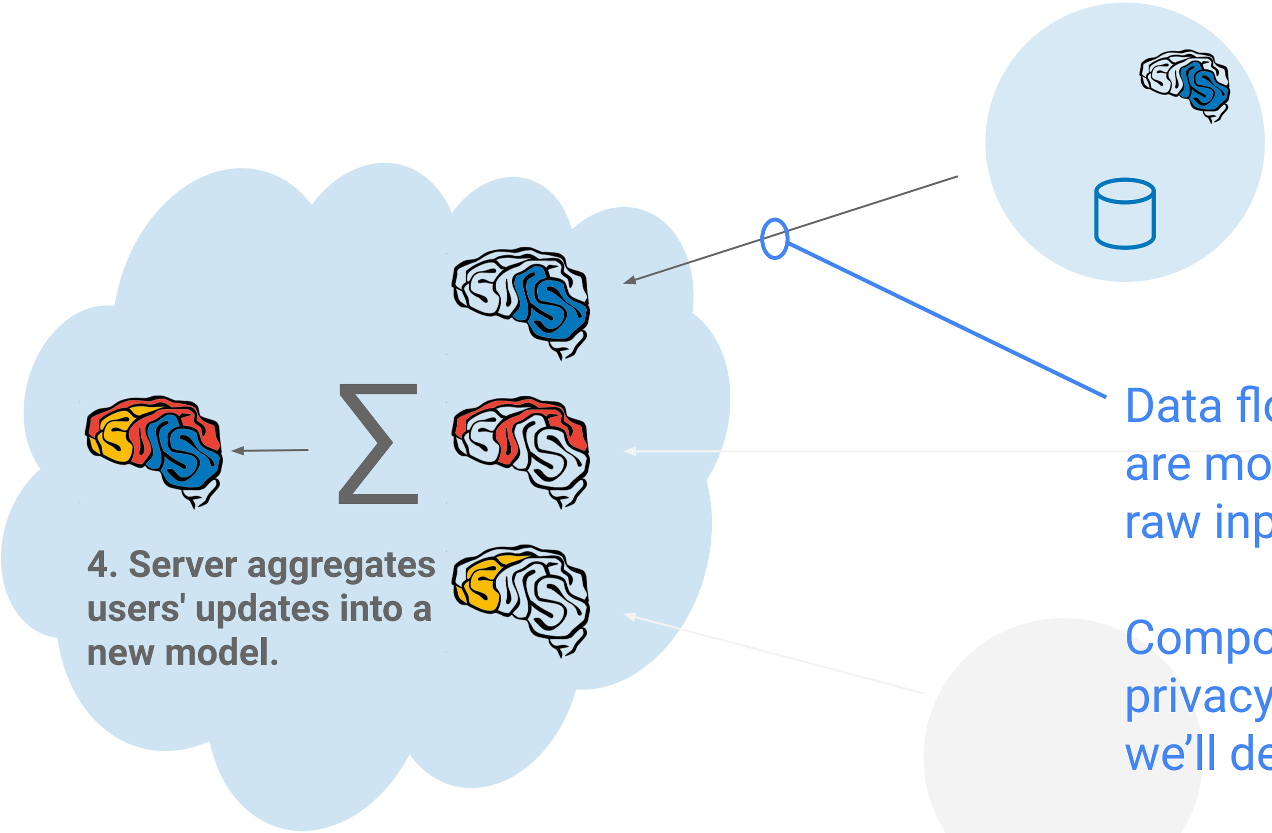
Federated Learning



Google

Repeat until convergence.

Federated Learning

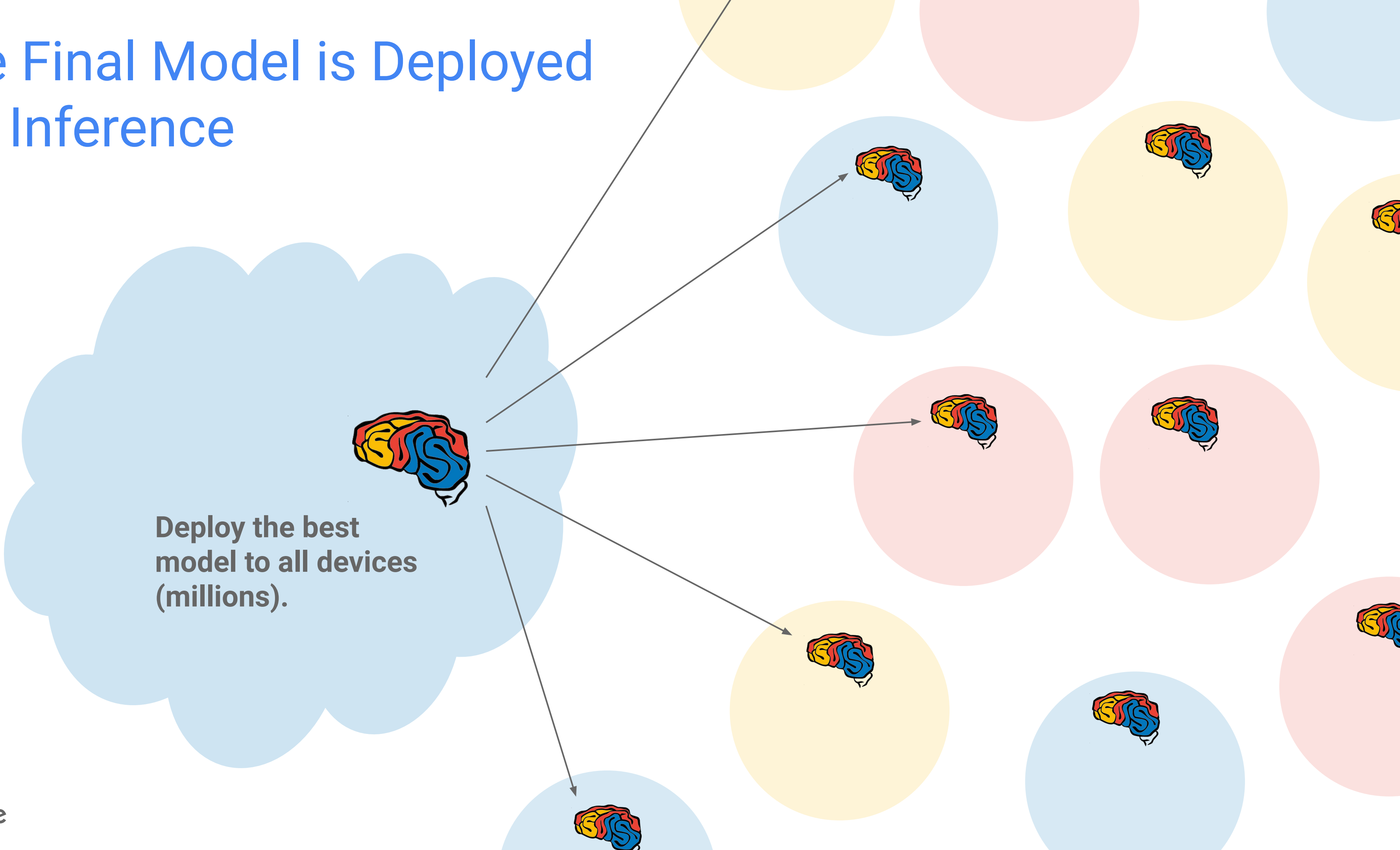


Data flowing back to server are model parameters, NOT raw input data

Composable with strong privacy guarantees, which we'll describe in a bit

Repeat until convergence.

The Final Model is Deployed For Inference



Google

The Federated Averaging algorithm

Server

Until Converged:

1. Select a random subset of clients

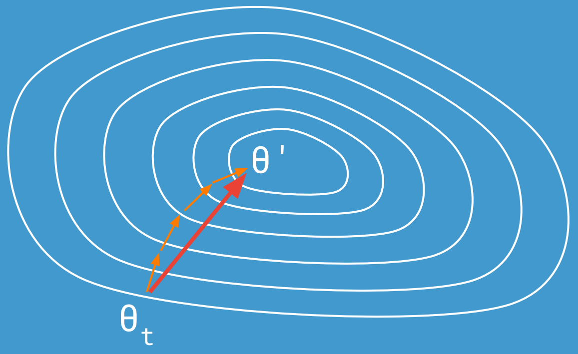
2. In parallel, send current parameters $\boldsymbol{\theta_t}$ to those clients

Selected Client k

1. Receive θ_t from server.

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

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

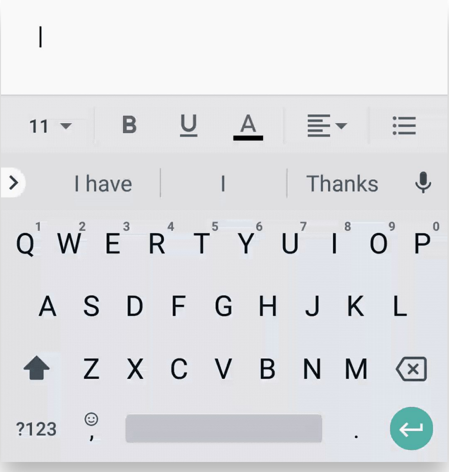


3. $\theta_{t+1} = \theta_t$ + data-weighted average of client updates

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

Gboard: language modeling

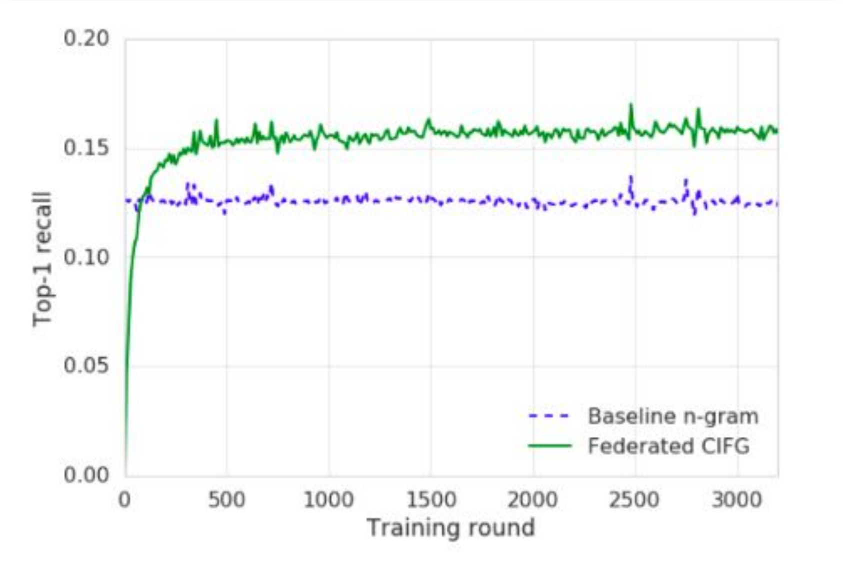
- Predict the next word based on typed text so far
- Powers the predictions strip



When should you consider federated learning?

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

Gboard: language modeling



Federated model
compared to baseline

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

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*

Is federated computation just distributed computing?

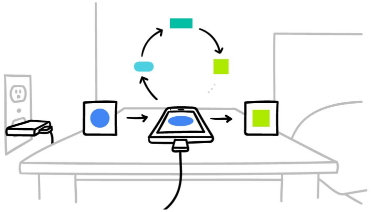
Semi-cyclic data availability

Each device reflects one users data.

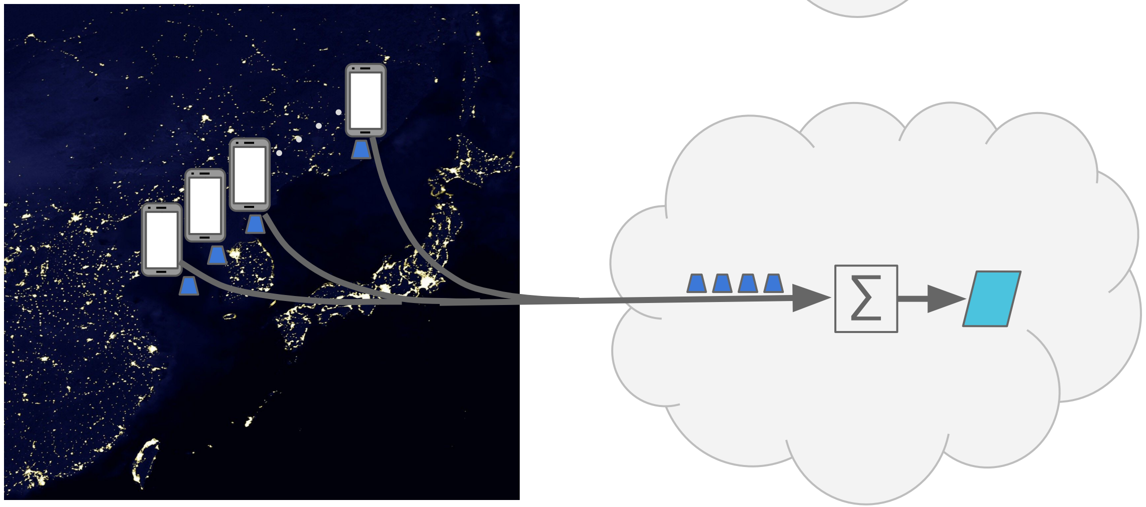
So no one device is representative of the whole population.

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

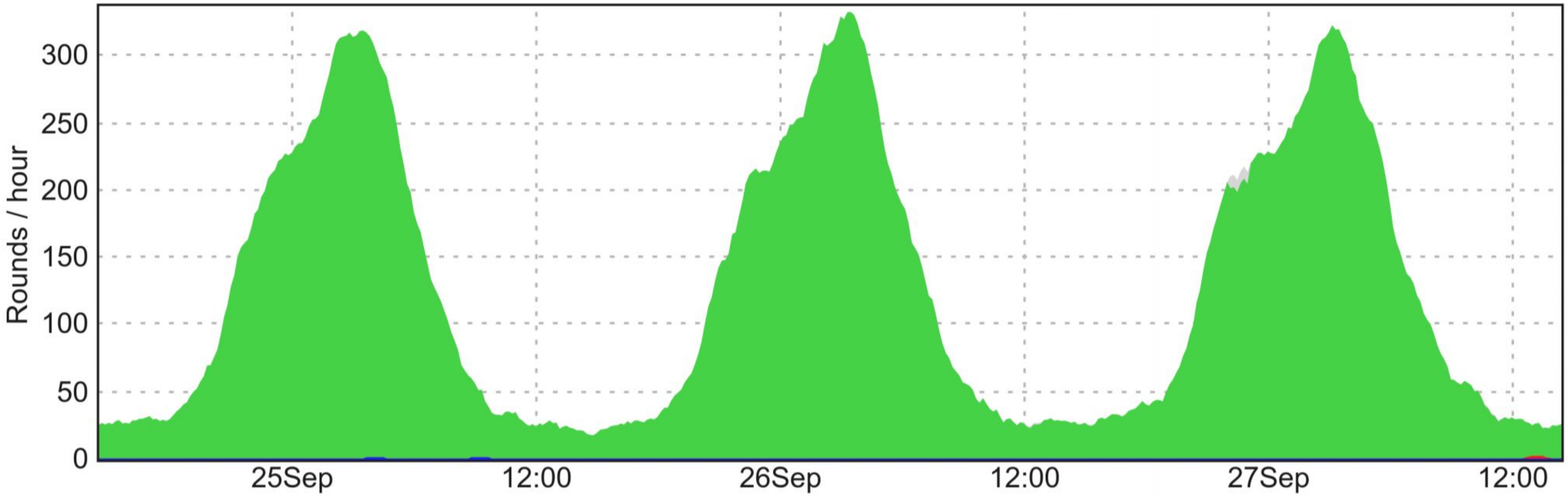
Device availability correlates with both geo location *and* data distribution.



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



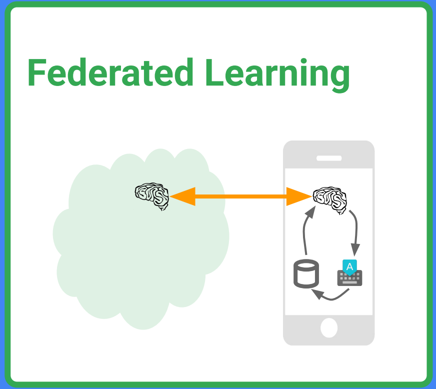
Round completion rate by hour (US)



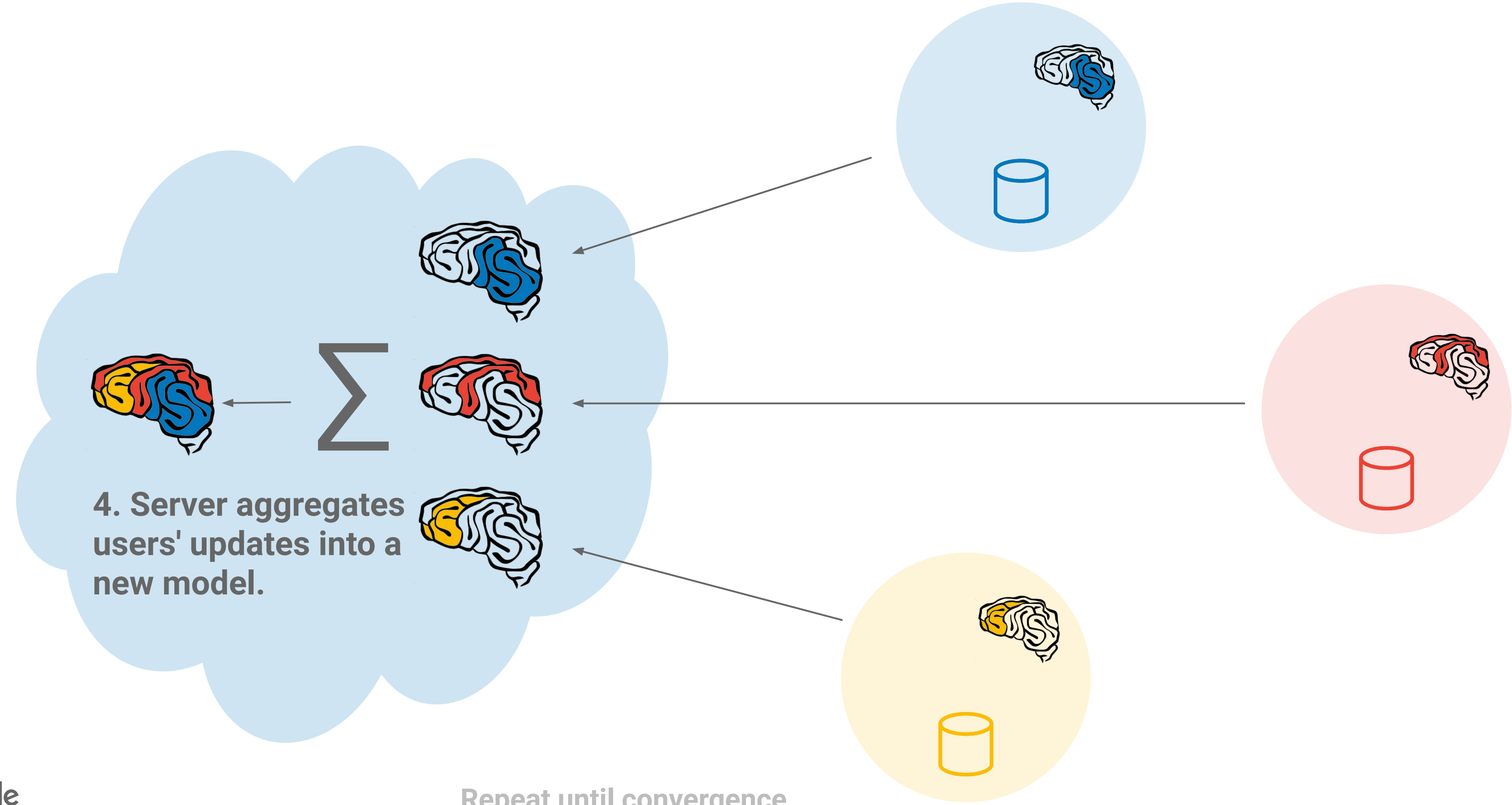
Hour (over three days)

- Rounds complete faster when more devices available
- Device availability changes over the course of a day

FL and Privacy



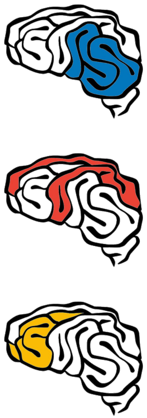
Federated Learning



Repeat until convergence.

Google

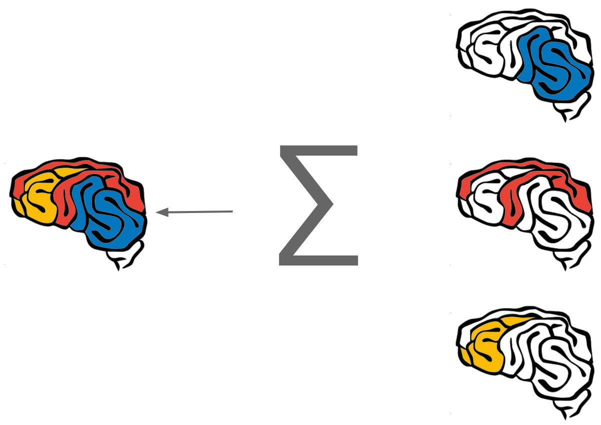
Might these updates contain privacy-sensitive data?



1. Ephemeral
2. Focused

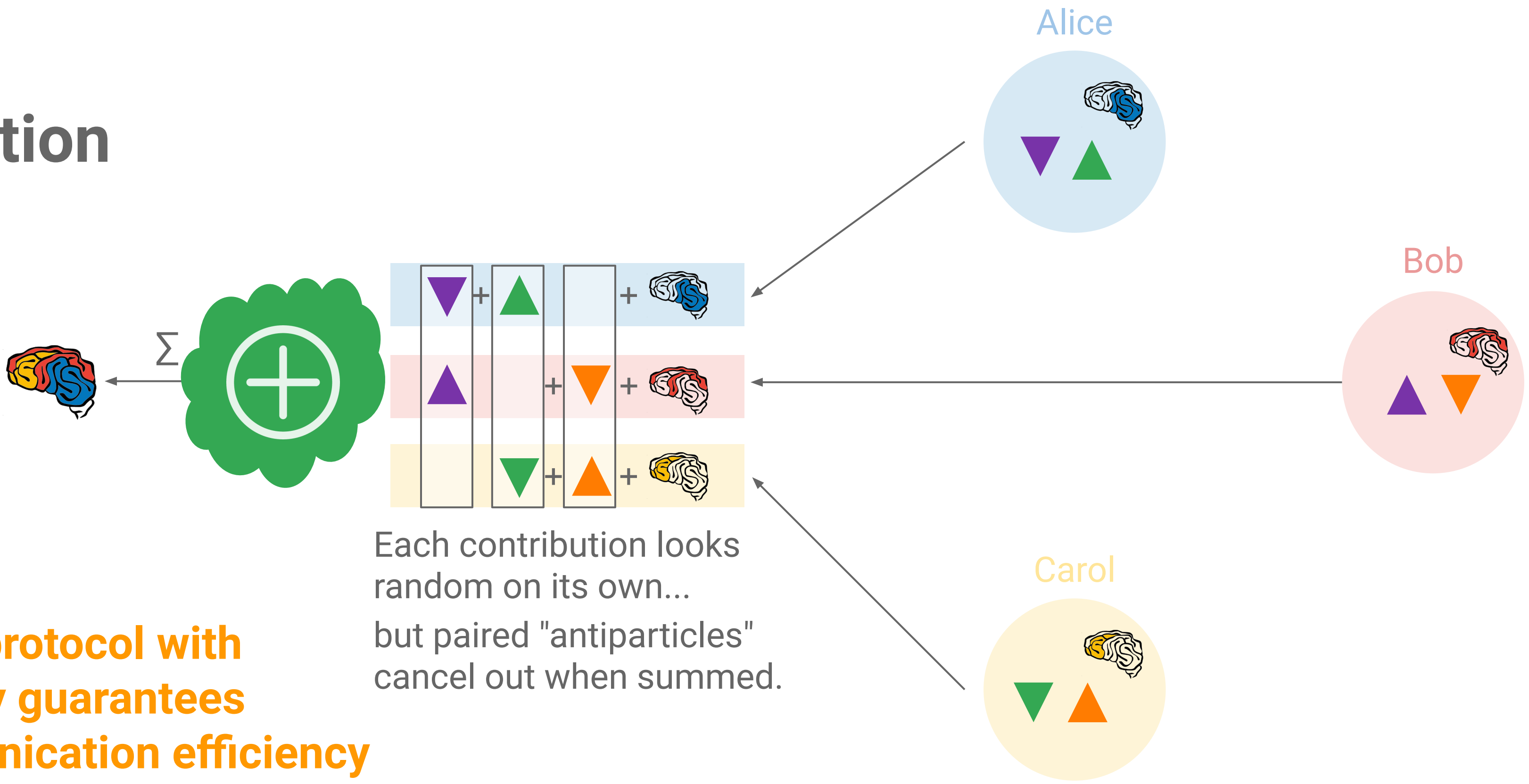
Improve privacy & security by minimizing the "attack surface"

Might these updates contain privacy-sensitive data?



1. Ephemeral
2. Focused
3. **Only in aggregate**

Secure Aggregation



Each contribution looks random on its own...

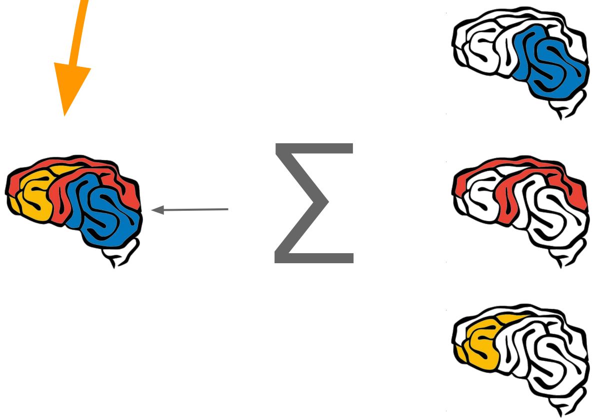
but paired "antiparticles" cancel out when summed.

A practical protocol with

- **Security guarantees**
- **Communication efficiency**
- **Dropout tolerance**

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.*

Might the final model memorize a user's data? (e.g, B. McMahan's credit card #)



1. Ephemeral
2. Focused
3. Only in aggregate
4. **Differentially private**

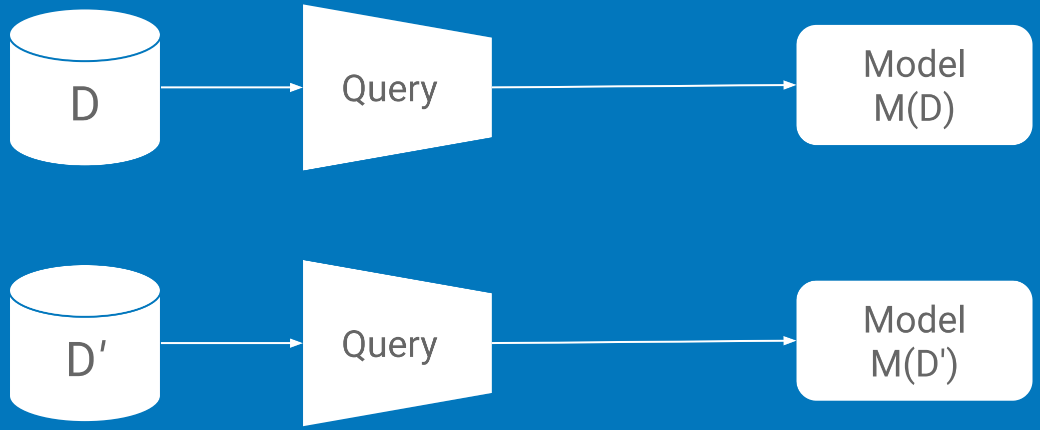
Differential Privacy

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]

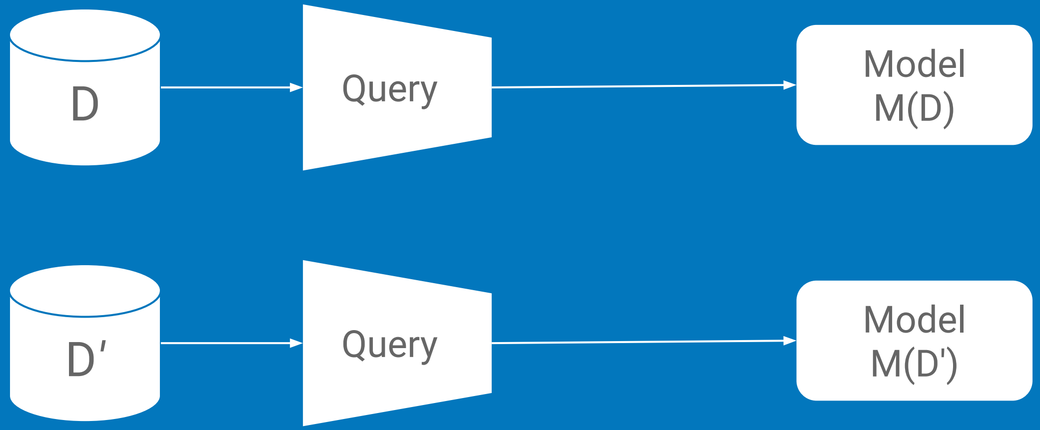
Google

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'

Differential Privacy

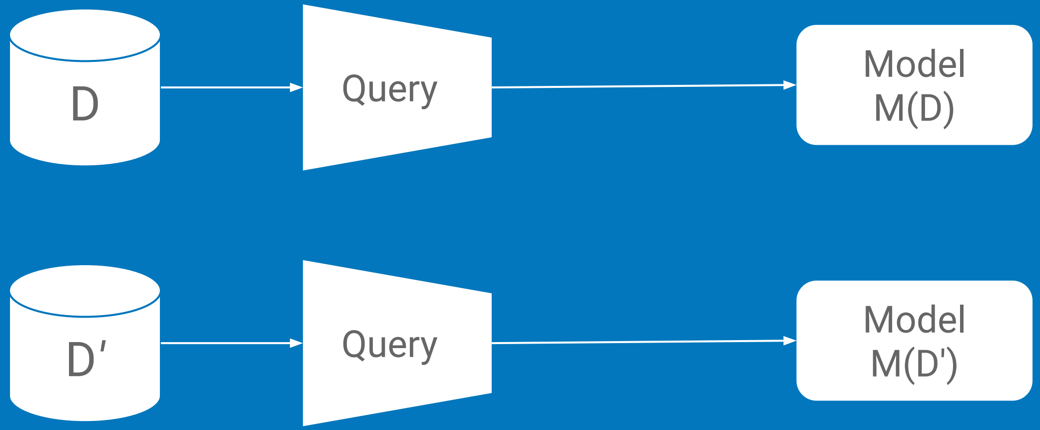


(ε, δ)-Differential Privacy: The distribution of the output $M(D)$ (a trained model) on database (training dataset) D is **nearly the same** as $M(D')$ for all adjacent databases D and D'

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

Google

Differential Privacy



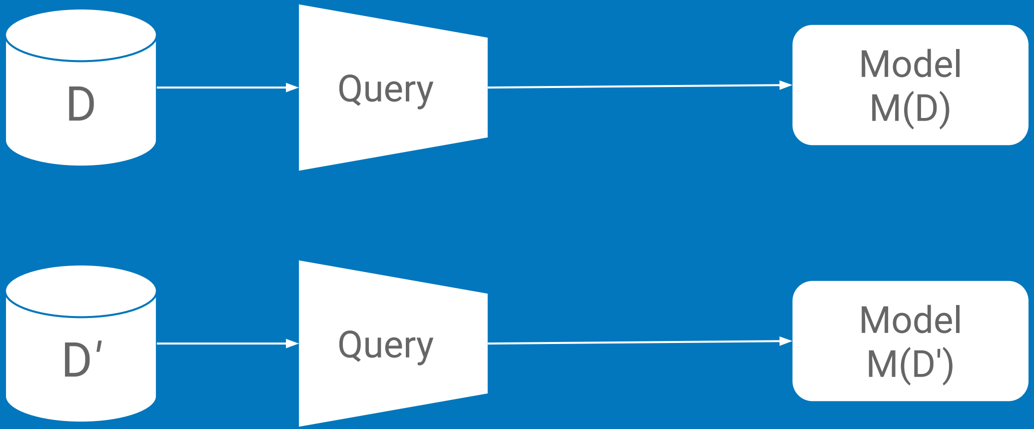
(ε, δ)-Differential Privacy: The distribution of the output $M(D)$ (a trained model) on database (training dataset) D is nearly the same as $M(D')$ for all **adjacent** databases D and D'

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

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, & L. Zhang. ***Deep Learning with Differential Privacy.*** *CCS 2016.*

Google

Differential Privacy (in Federated Context)

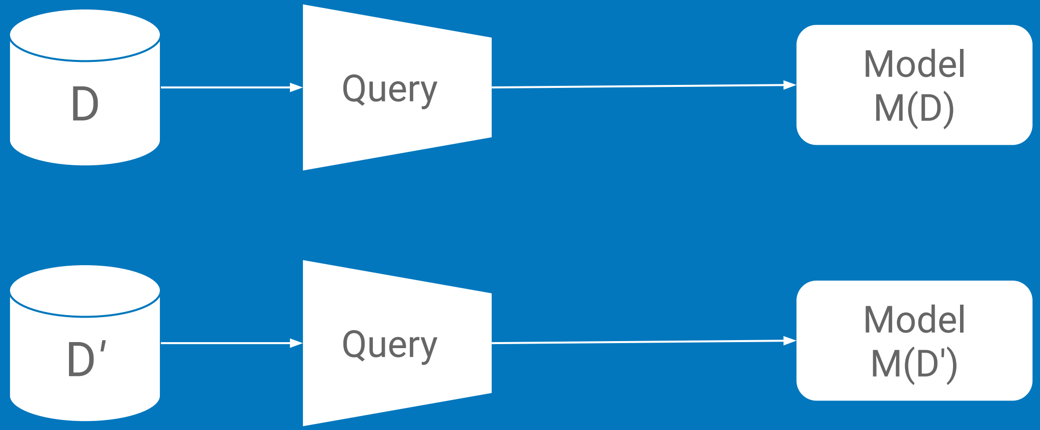


(ε, δ)-Differential Privacy: The distribution of the output $M(D)$ (a trained model) on database (training dataset) D is nearly the same as $M(D')$ for all **adjacent** databases D and D'

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

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

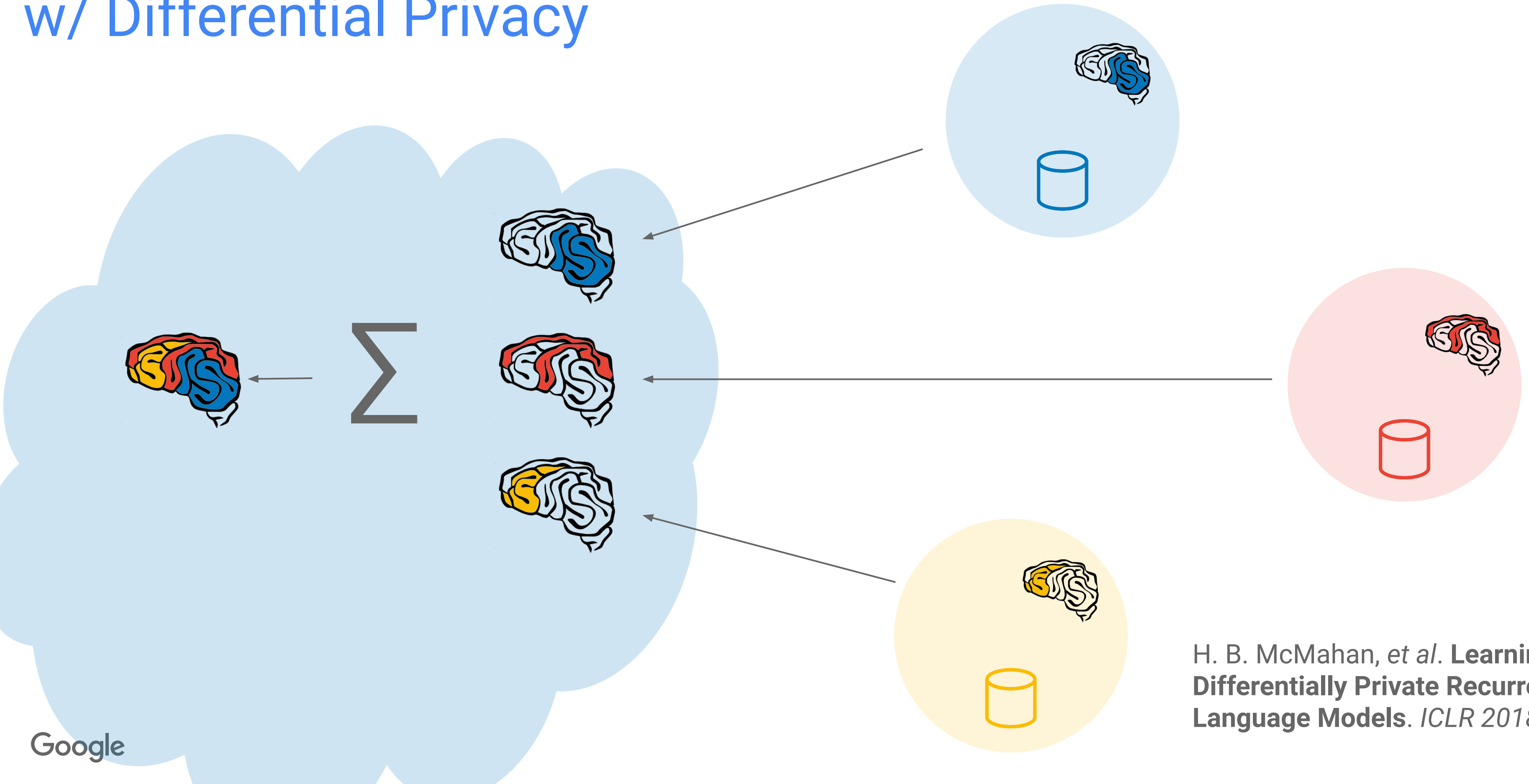
Differential Privacy



(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'

Sensitivity: How much Query(D) and Query(D') differ

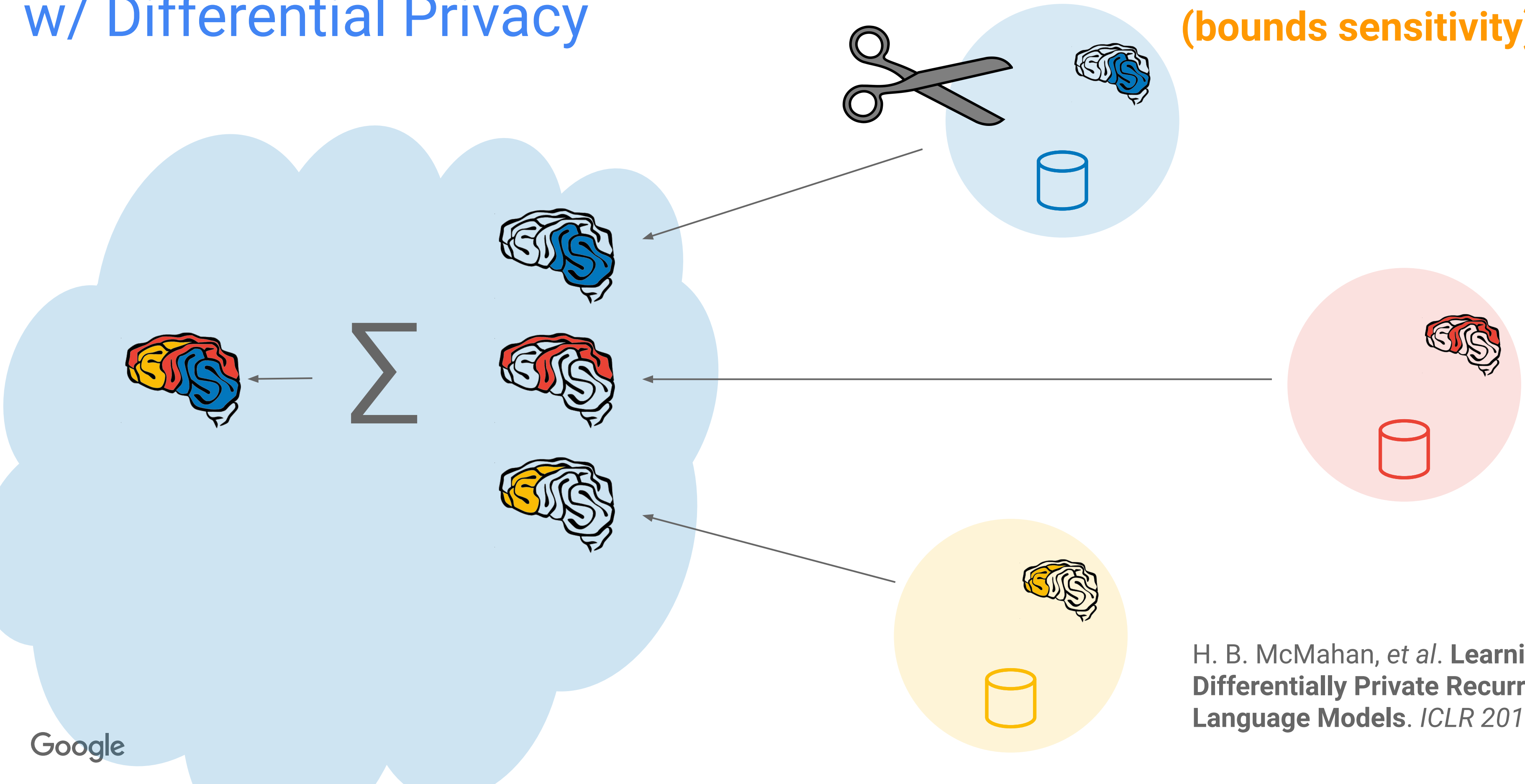
Federated Learning w/ Differential Privacy



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

Federated Learning w/ Differential Privacy

Clip updates to limit a user's contribution (bounds sensitivity).

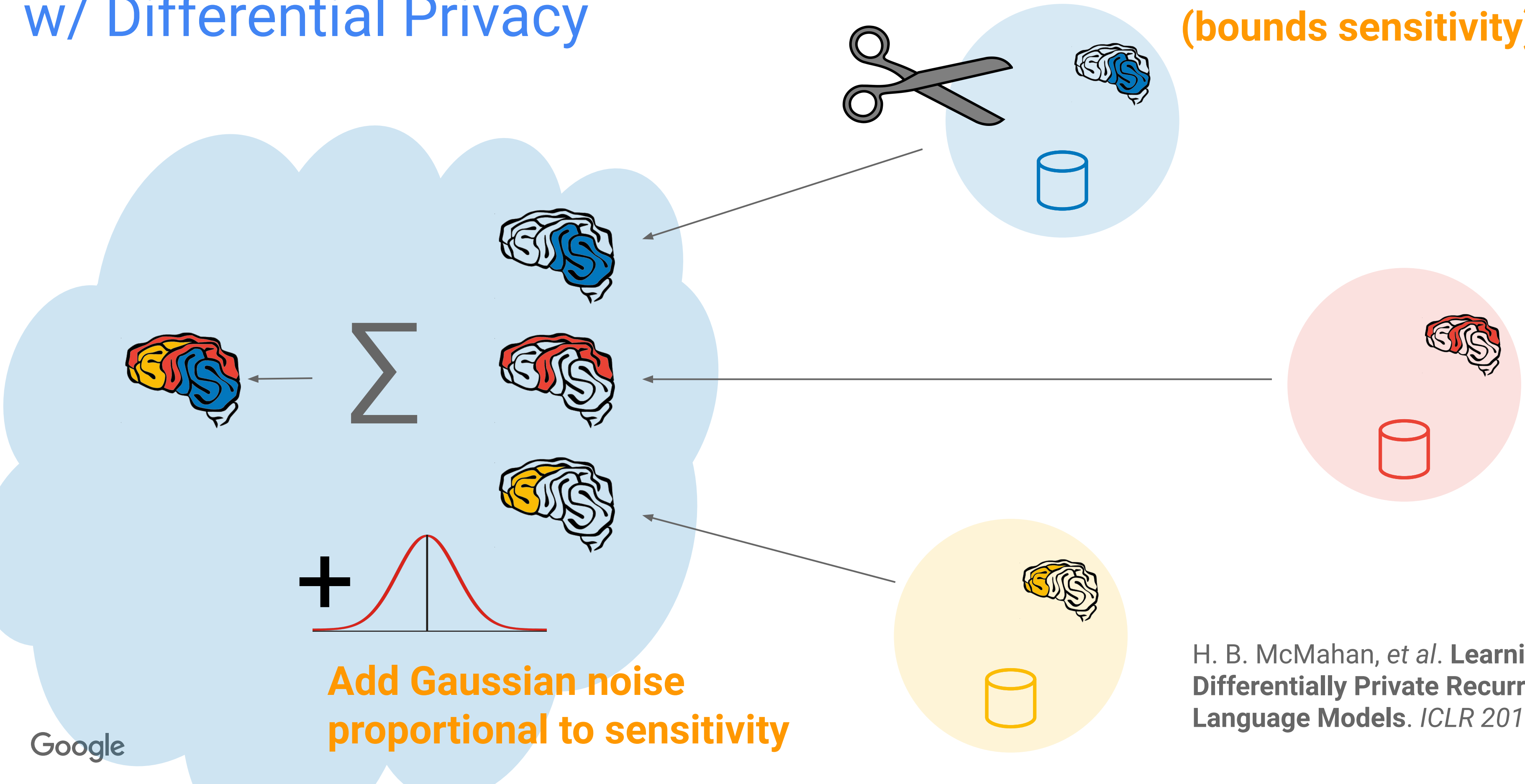


H. B. McMahan, *et al*. **Learning Differentially Private Recurrent Language Models**. *ICLR 2018*.

Google

Federated Learning w/ Differential Privacy

Clip updates to limit a user's contribution (bounds sensitivity).



Add Gaussian noise proportional to sensitivity

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

Google

Differentially-Private Federated Averaging

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

Server

Until Converged:

1. Select each user **independently** with **probability q**, for say E[C]=1000 clients

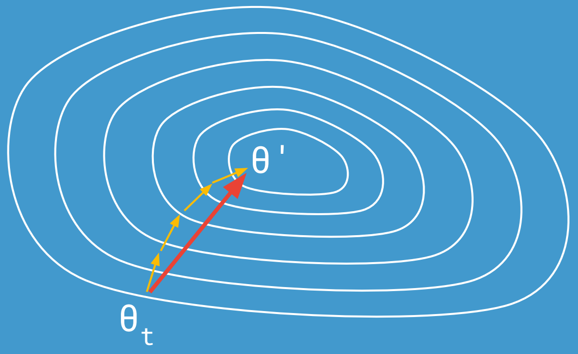
2. In parallel, send current parameters $\boldsymbol{\theta_t}$ to those clients

Selected Client k

1. Receive θ_t from server.

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

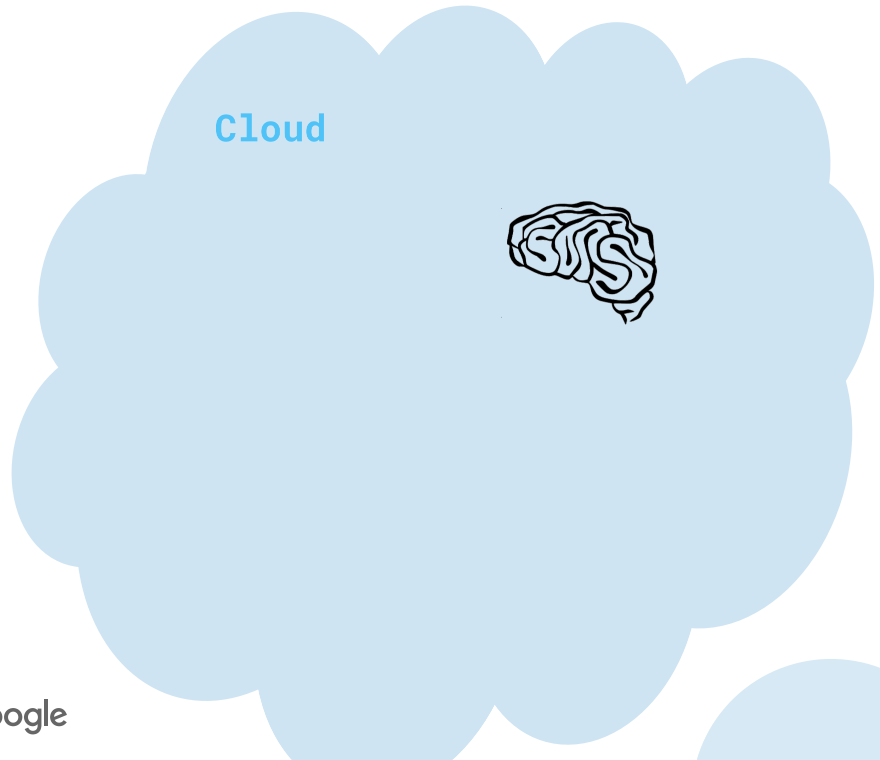
3. Return **Clip($\theta'-\theta_t$)** to server.

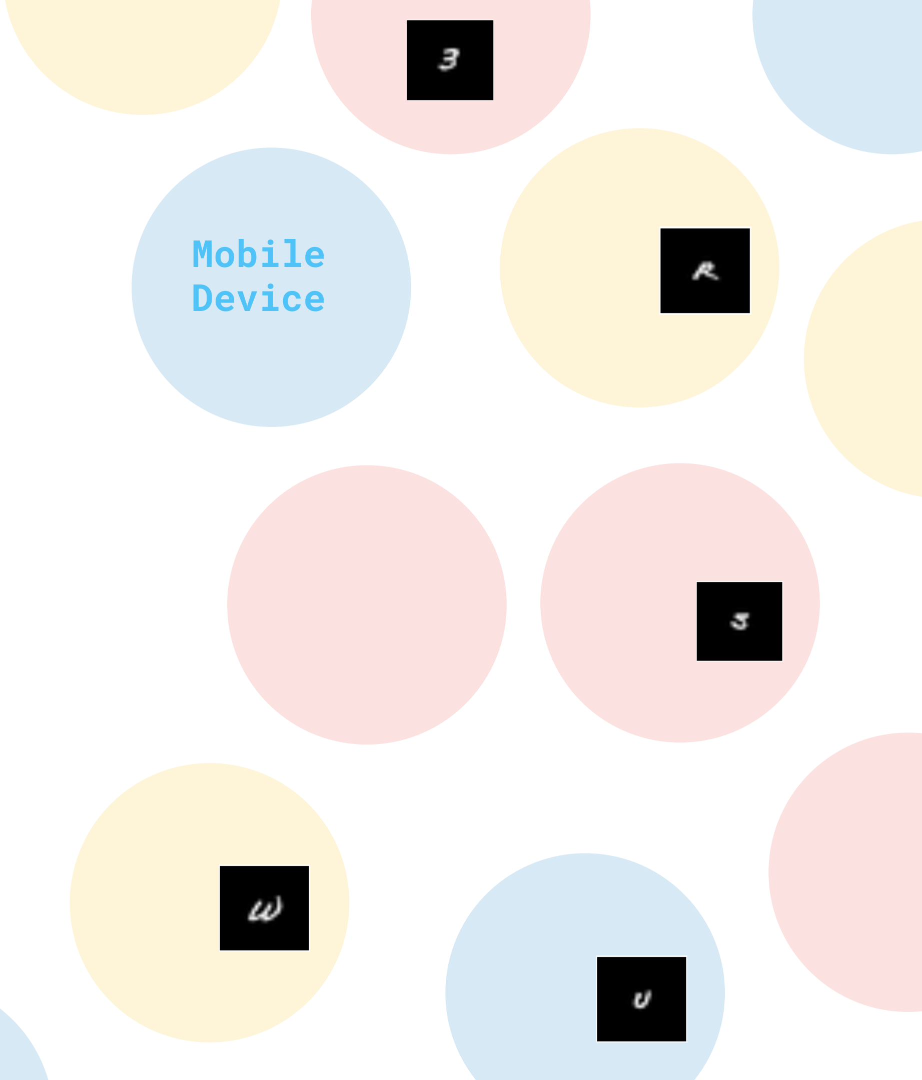


3. $\theta_{t+1} = \theta_t$ + **bounded sensitivity** data-weighted average of client updates + Gaussian noise **N(0, Iσ²)**

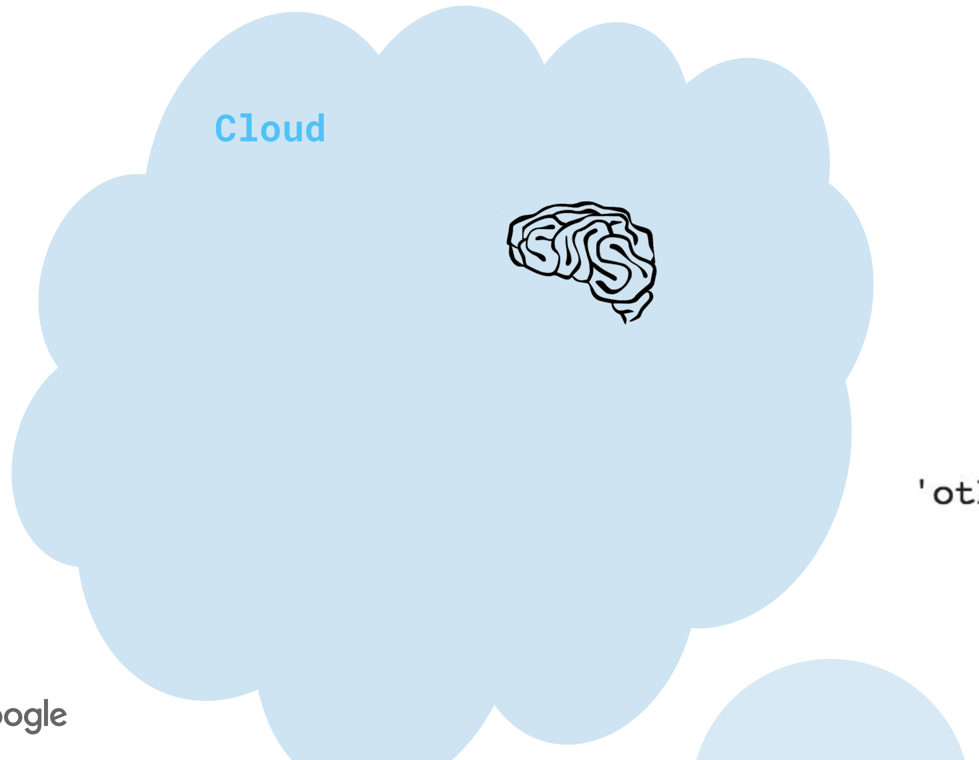
Challenges to private, decentralized learning/analytics

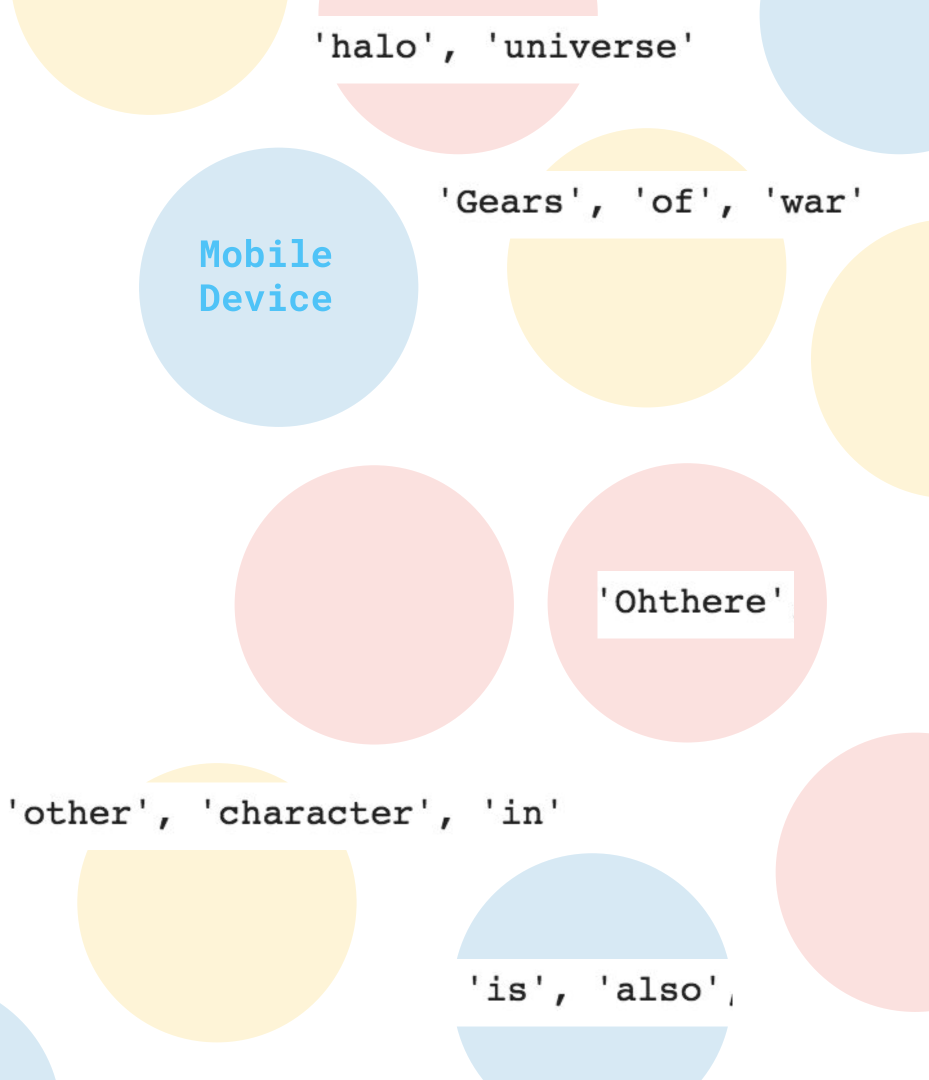
Example: Local Data Caches store Images





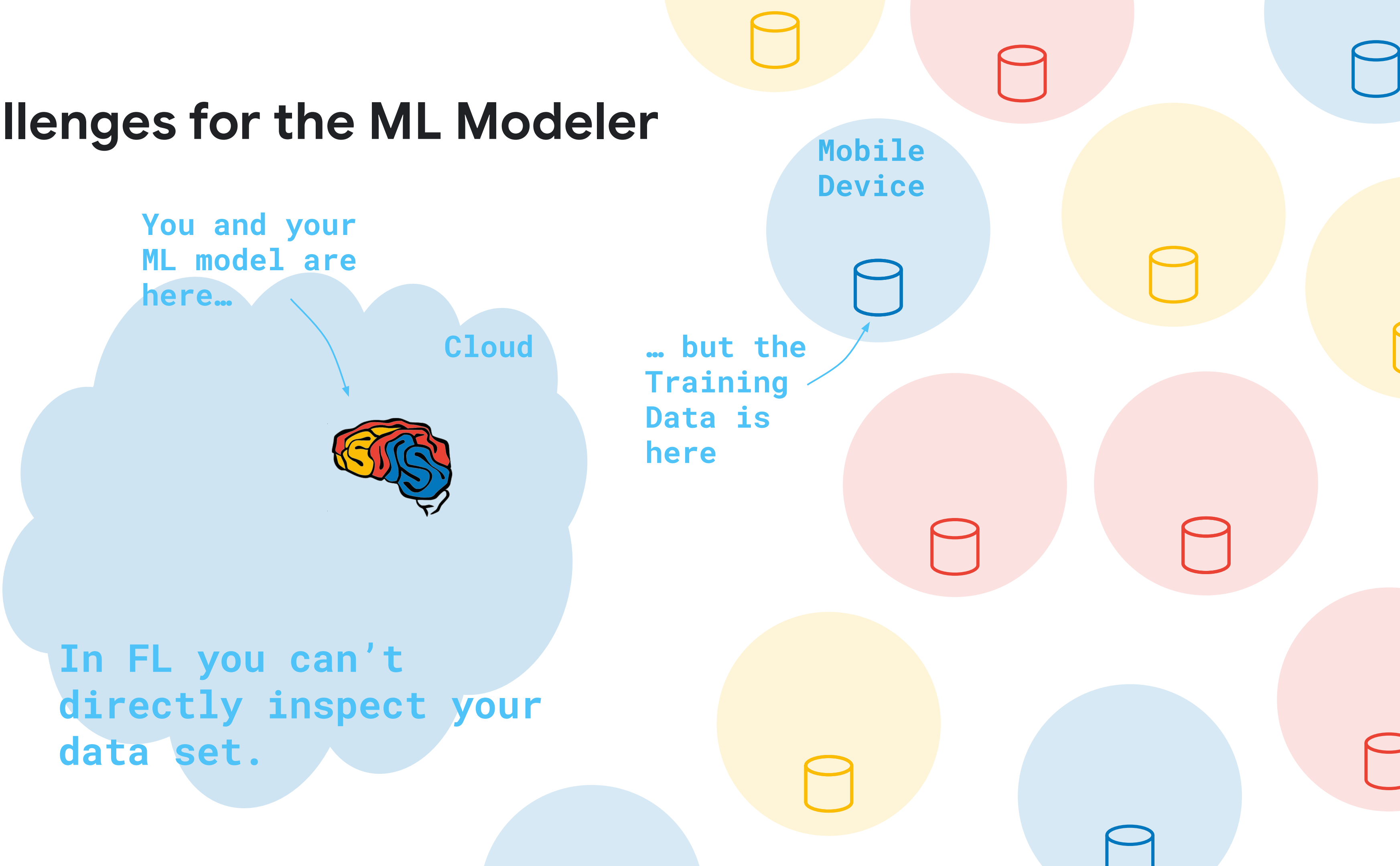
Example: Local Data Caches store Text





Google

Challenges for the ML Modeler



Google AI

Challenges for the ML Modeler : Debugging

- "I'm observing metrics outside the expected range, why?", or ...
- "My trained model is behaving pathologically, why?"
    - Inspect image data set, realize there's a pixel range mismatch b/w examples and expected

$$\boxed{x \in [0, 255]} \quad \textbf{vs.} \quad x \in [-1.0, 1.0]$$

    - Inspect image data set, realize bug in preprocessing (some images have intensity inverted)

vs.

    - Inspect text data set, realize bug in tokenizing (some tokens incorrectly concatenated)

```
['Ohthere', 'is', 'also', 'Gears', 'of', 'war', ',', 'other', 'character', 'in', 'the', 'halo', 'universe',
```

Google AI

Challenges for the ML Modeler : Data Set Augmentation

- “I need to gather input samples (features), to pass to humans to apply labels”

$$\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}$$

- “I have a biased dataset, I need to gather samples of underrepresented classes”

$$\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\;\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}$$

Google AI

Challenges for the ML Modeler : Data Set Augmentation

- “I need to gather input samples (features), to pass to humans to apply labels”

$$\begin{bmatrix}\bar{x}\\ \color{red}{\bar{y}}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \color{red}{\bar{y}}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \color{red}{\bar{y}}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \color{red}{\bar{y}}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \color{red}{\bar{y}}\end{bmatrix}$$

- “I have a biased dataset, I need to gather samples of underrepresented classes”

$$\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\quad \color{red}{\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}}$$

How do you do these types of things when you can’t directly inspect the data?

Differentially Private, Federated Generate Models

Federated Generative Models

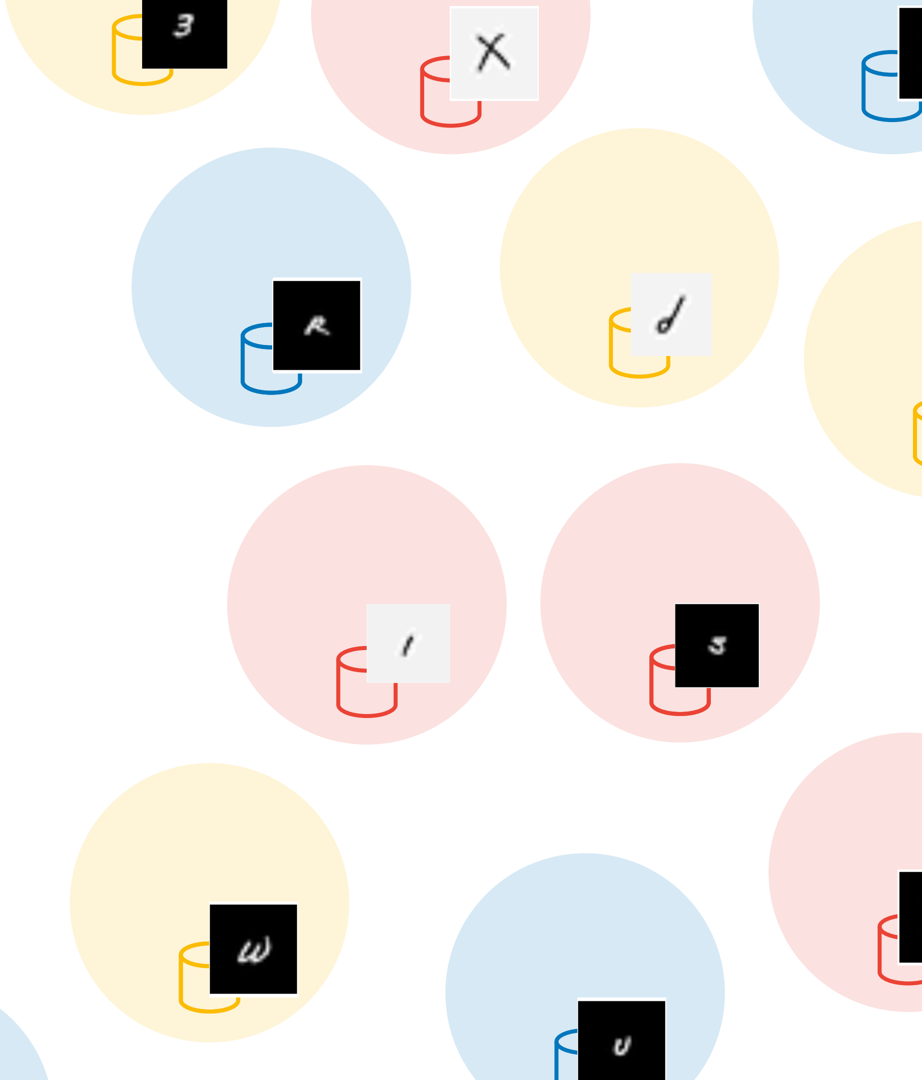
- Can we train via federation a model capable of synthesizing privatized, novel examples that match the distribution of the private, decentralized dataset?

- Privacy is paramount
    - A Federated Generative Model should not be able to memorize data unique to an individual

- Many options at our disposal:
    - Differentially Private, Federated GANs (for Image Applications)
    - Differentially Private, Federated Recurrent NNs (for Text Applications)
    - ...

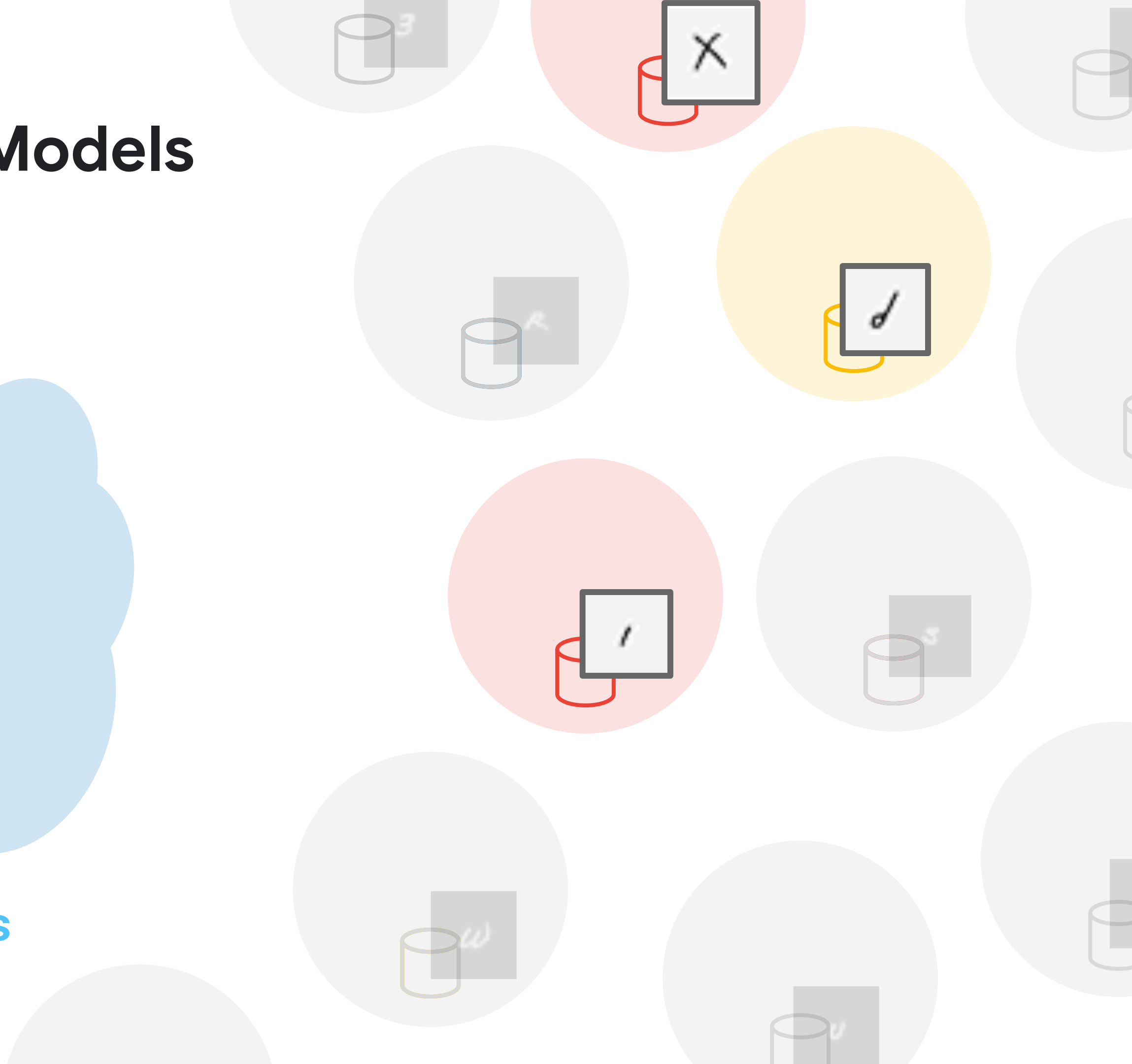
Federated Generative Models

Take an image model debugging example...

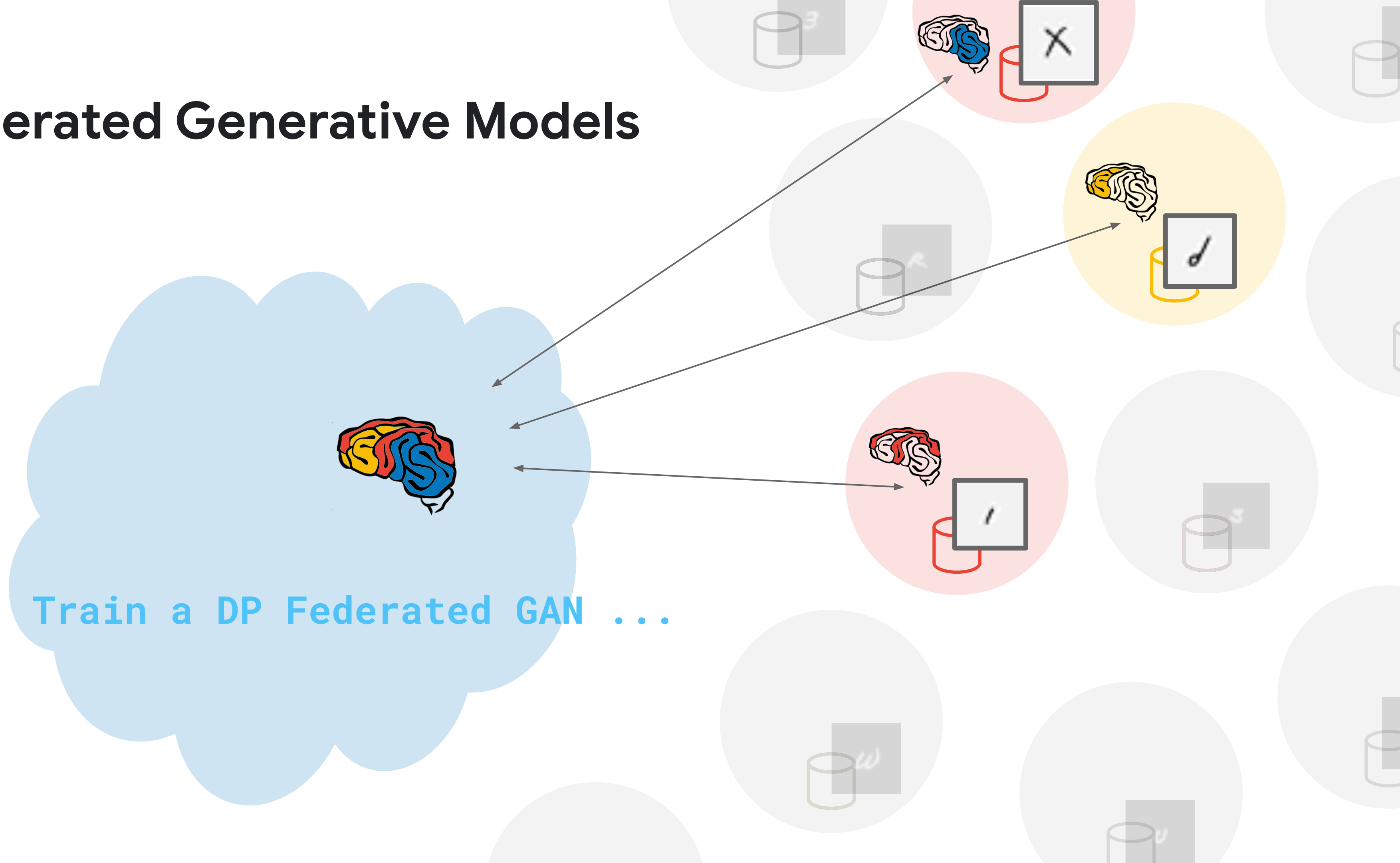


Federated Generative Models

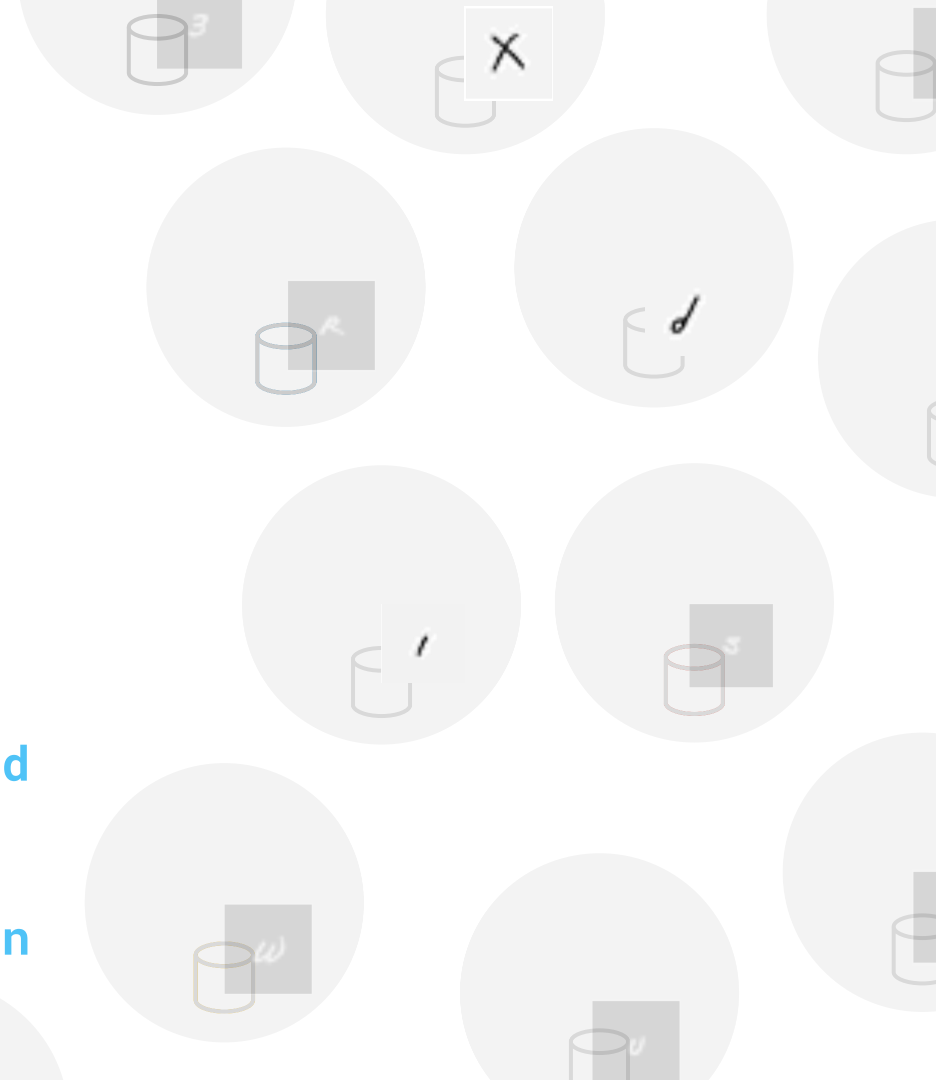
Add logic to gather samples in cases where metrics fall outside expectations



Federated Generative Models



Federated Generative Models

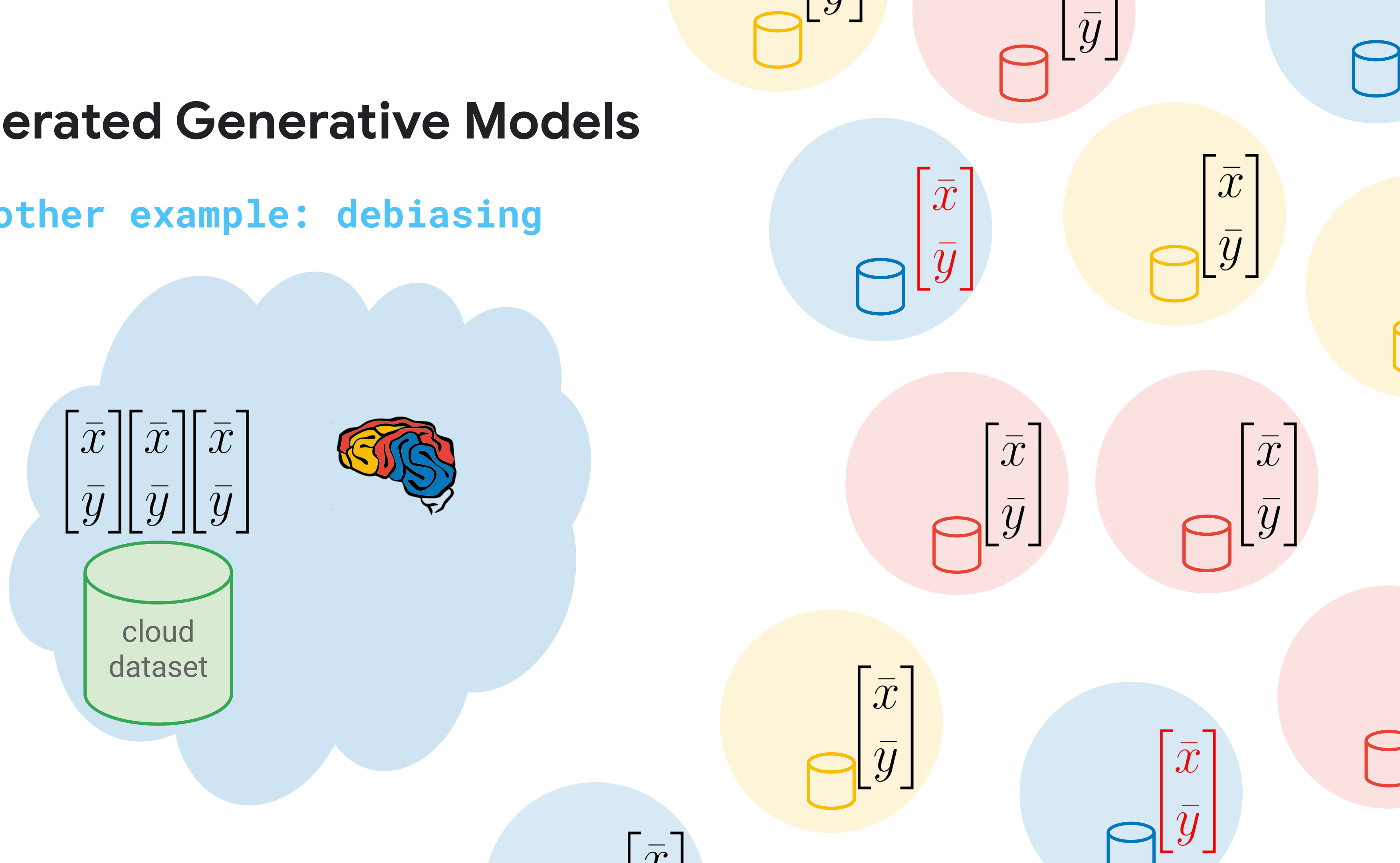


Train a DP Federated GAN and synthesize novel images (at the cloud) that match the characteristics of images in private dataset.

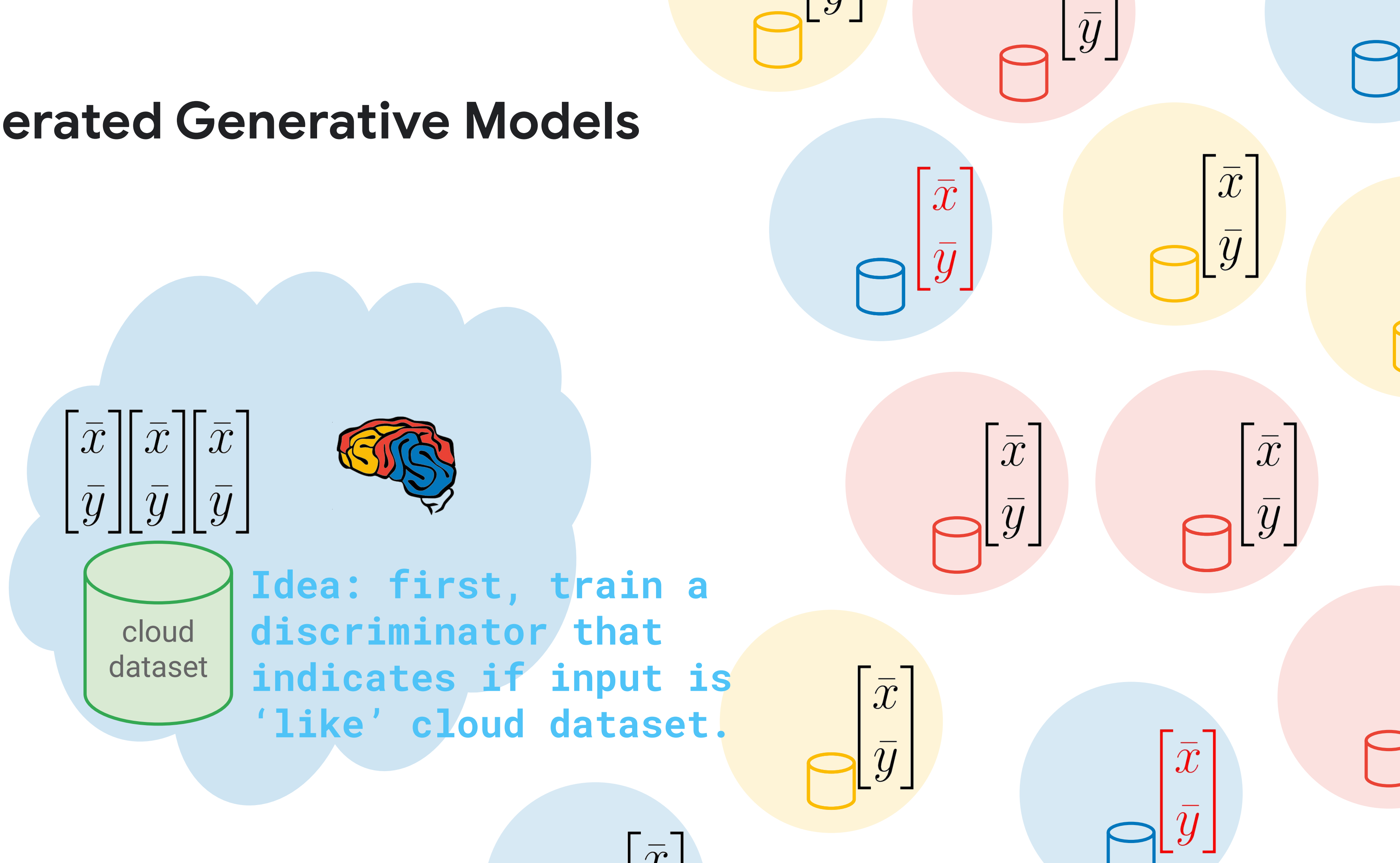
Observe intensity inversion.

Federated Generative Models

Another example: debiasing

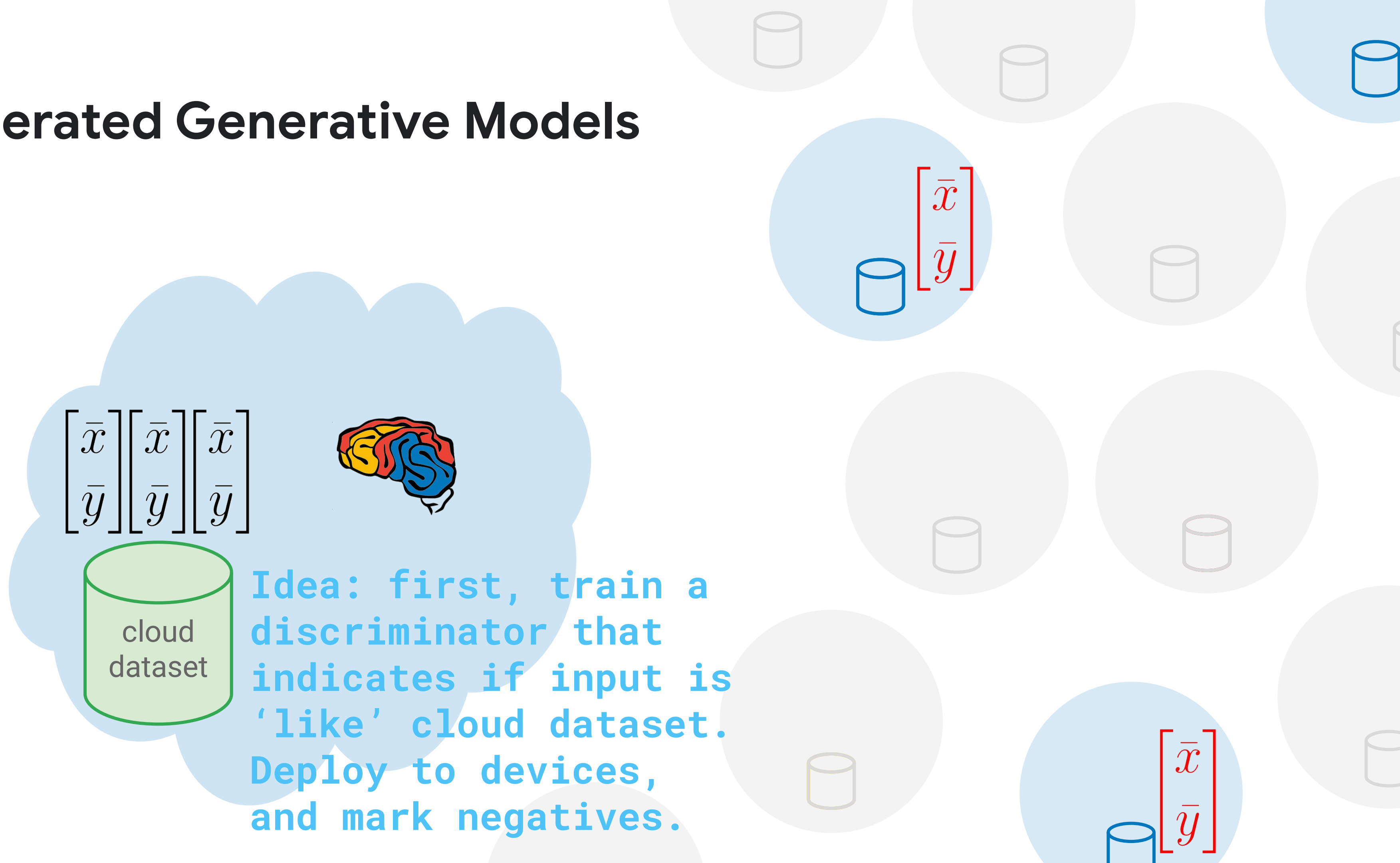


Federated Generative Models



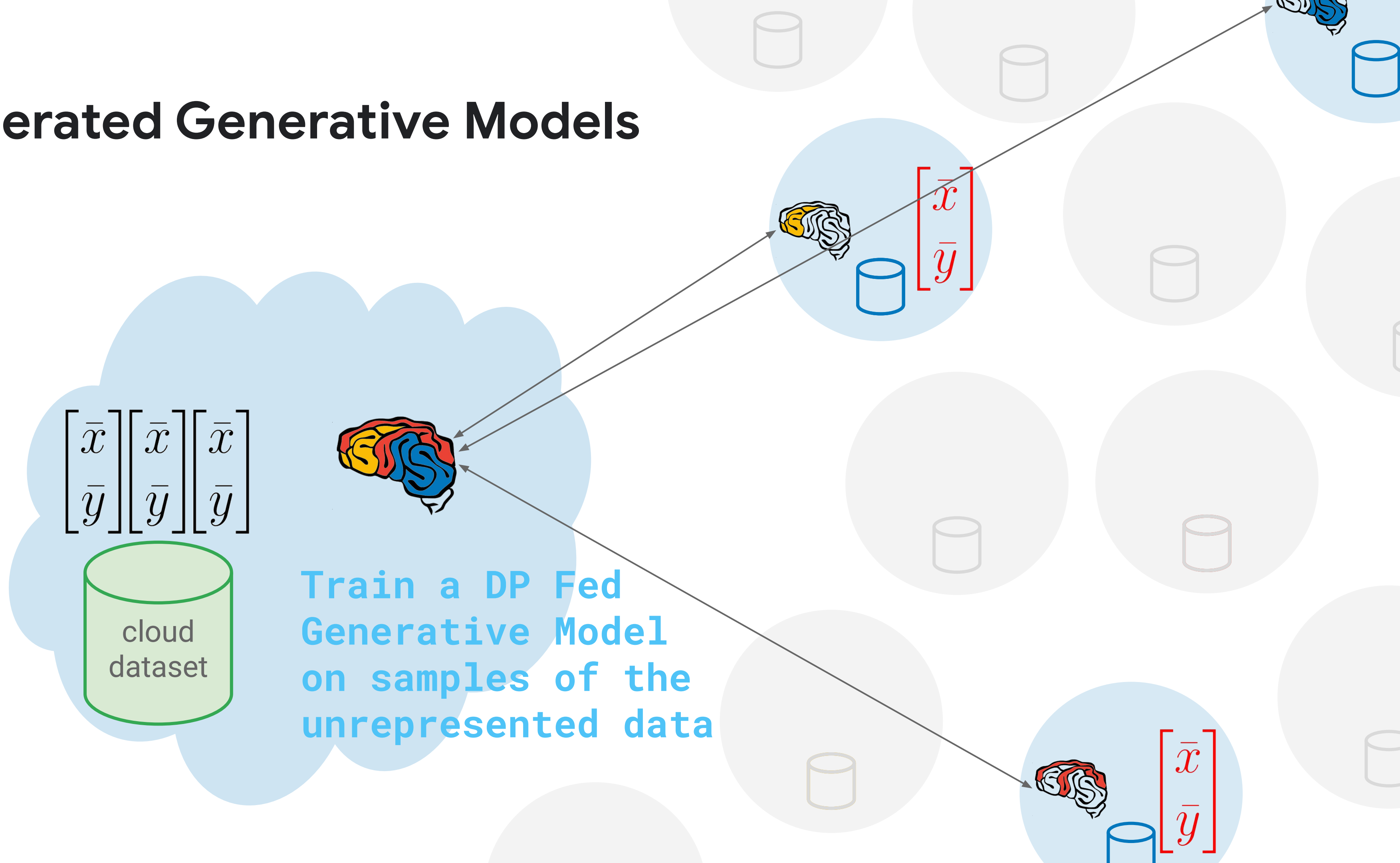
Idea: first, train a discriminator that indicates if input is ‘like’ cloud dataset.

Federated Generative Models



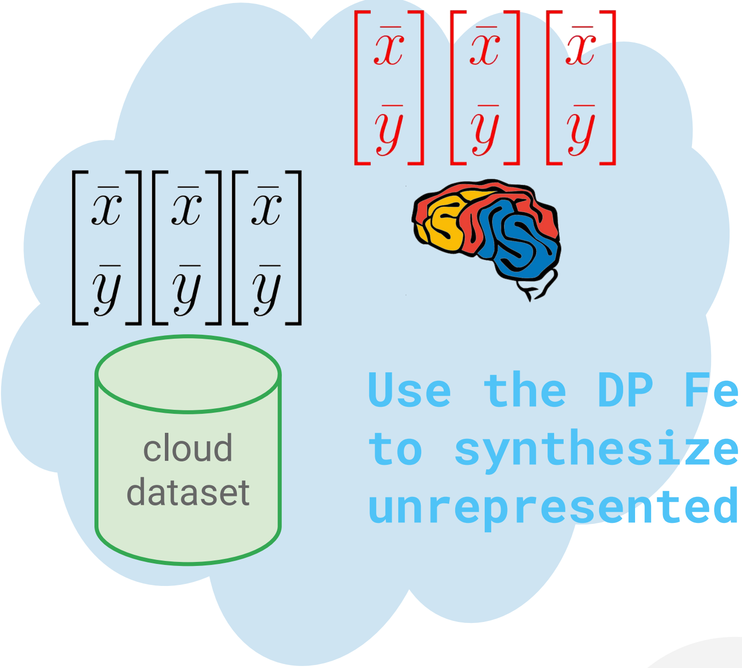
Idea: first, train a discriminator that indicates if input is 'like' cloud dataset. Deploy to devices, and mark negatives.

Federated Generative Models



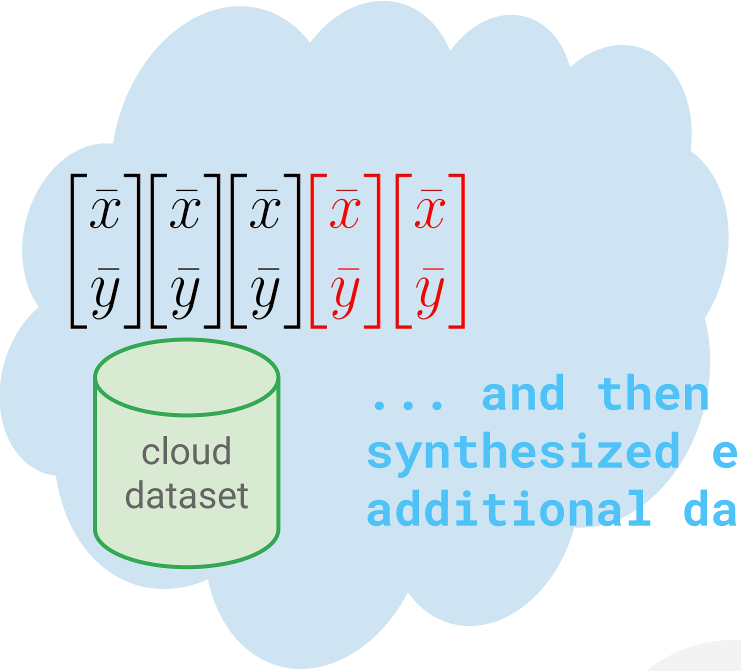
Train a DP Fed Generative Model on samples of the unrepresented data

Federated Generative Models



Use the DP Fed Generative Model to synthesize novel examples of unrepresented data...

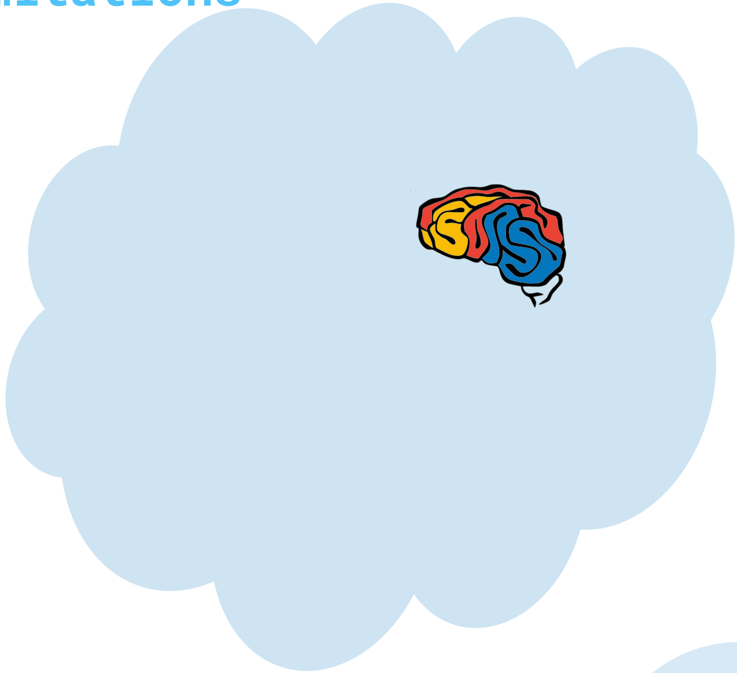
Federated Generative Models



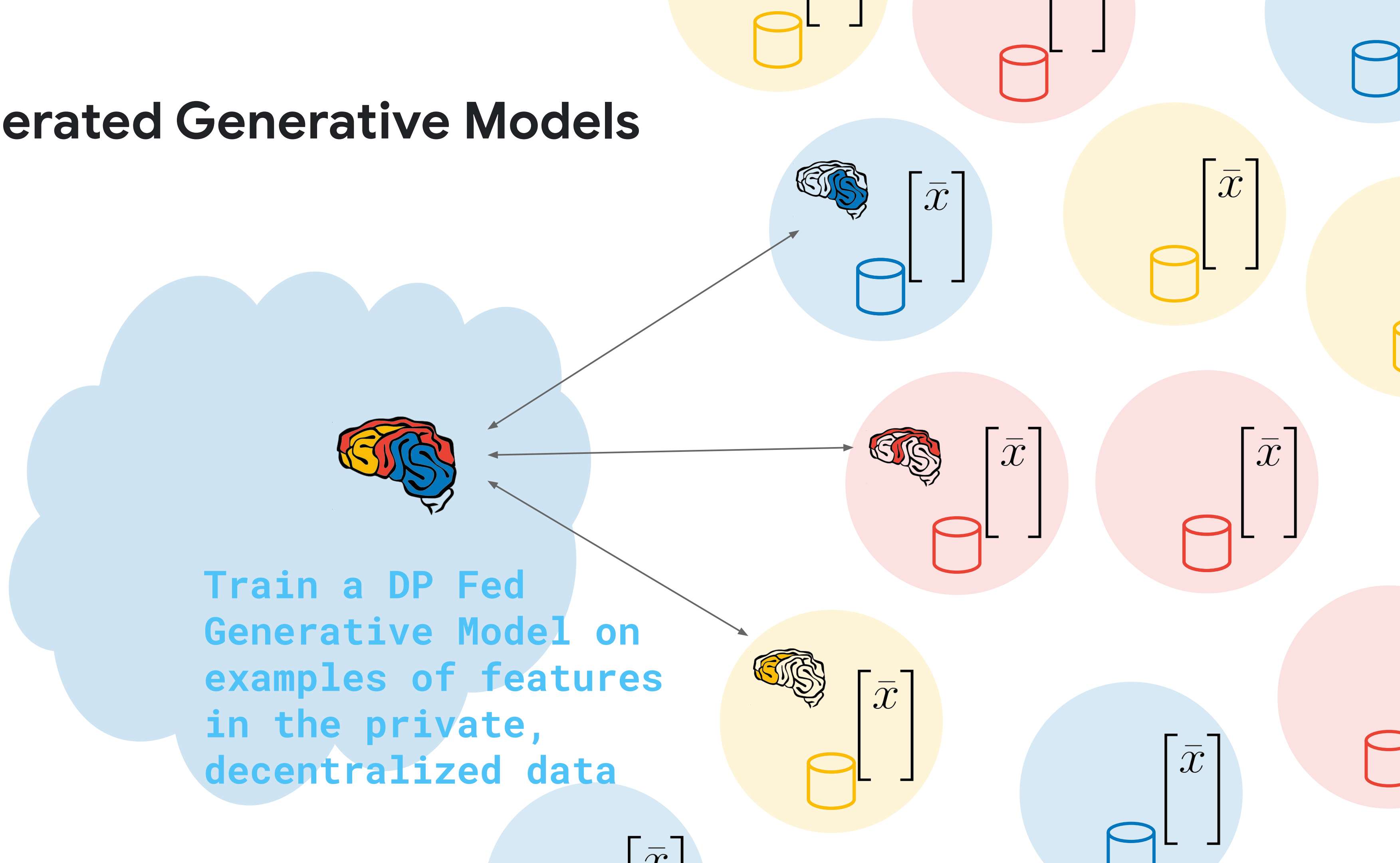
... and then developer uses synthesized examples to inform additional data collection, etc.

Federated Generative Models

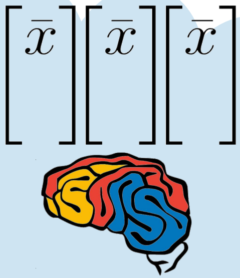
Final example: beyond self-labeling limitations



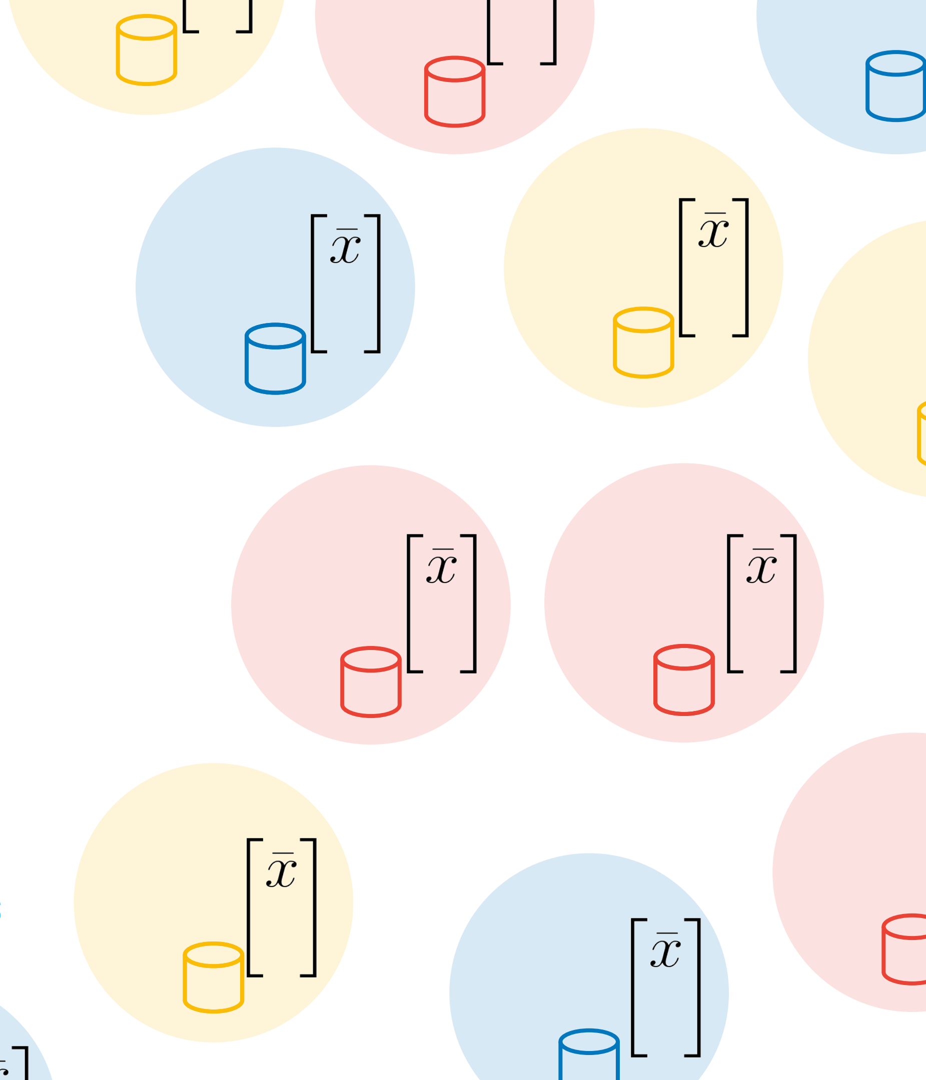
Federated Generative Models



Federated Generative Models



Use the DP Fed Generative Model to synthesis novel examples of features the developer can use ...

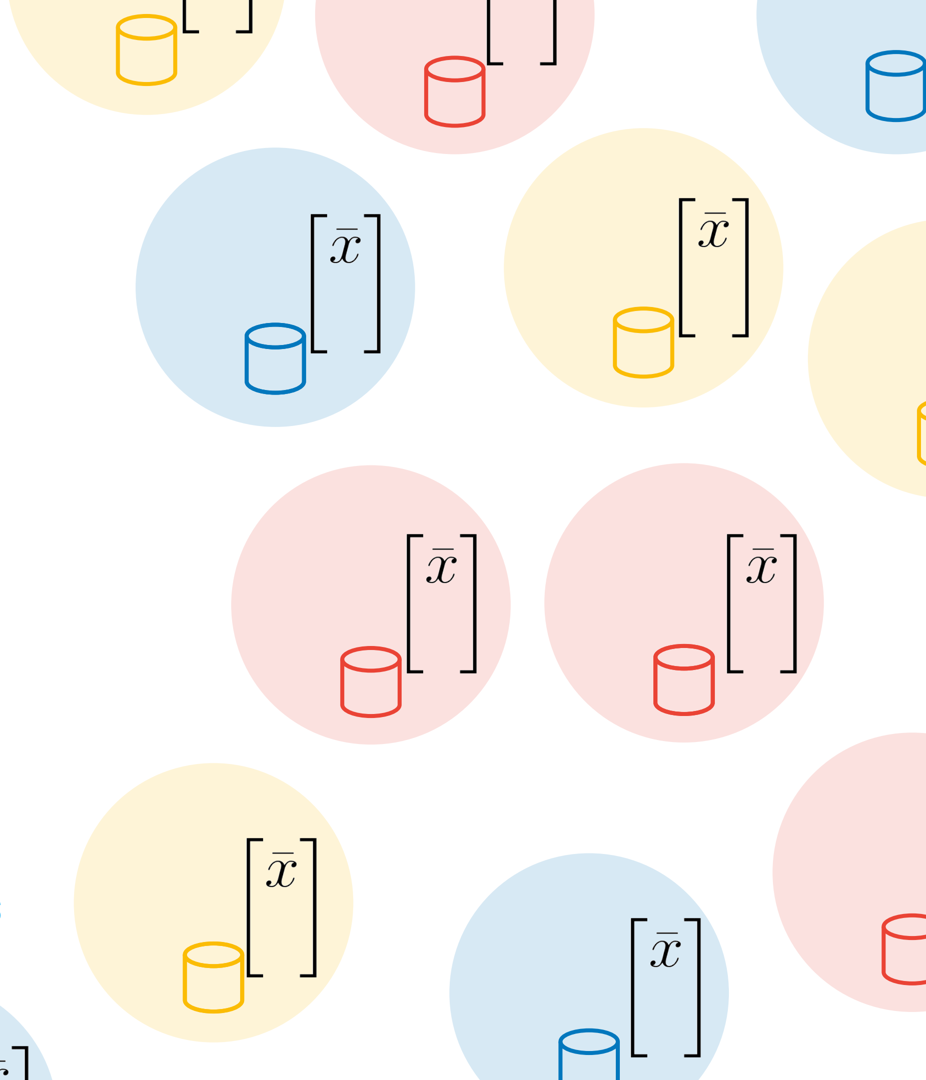


Federated Generative Models

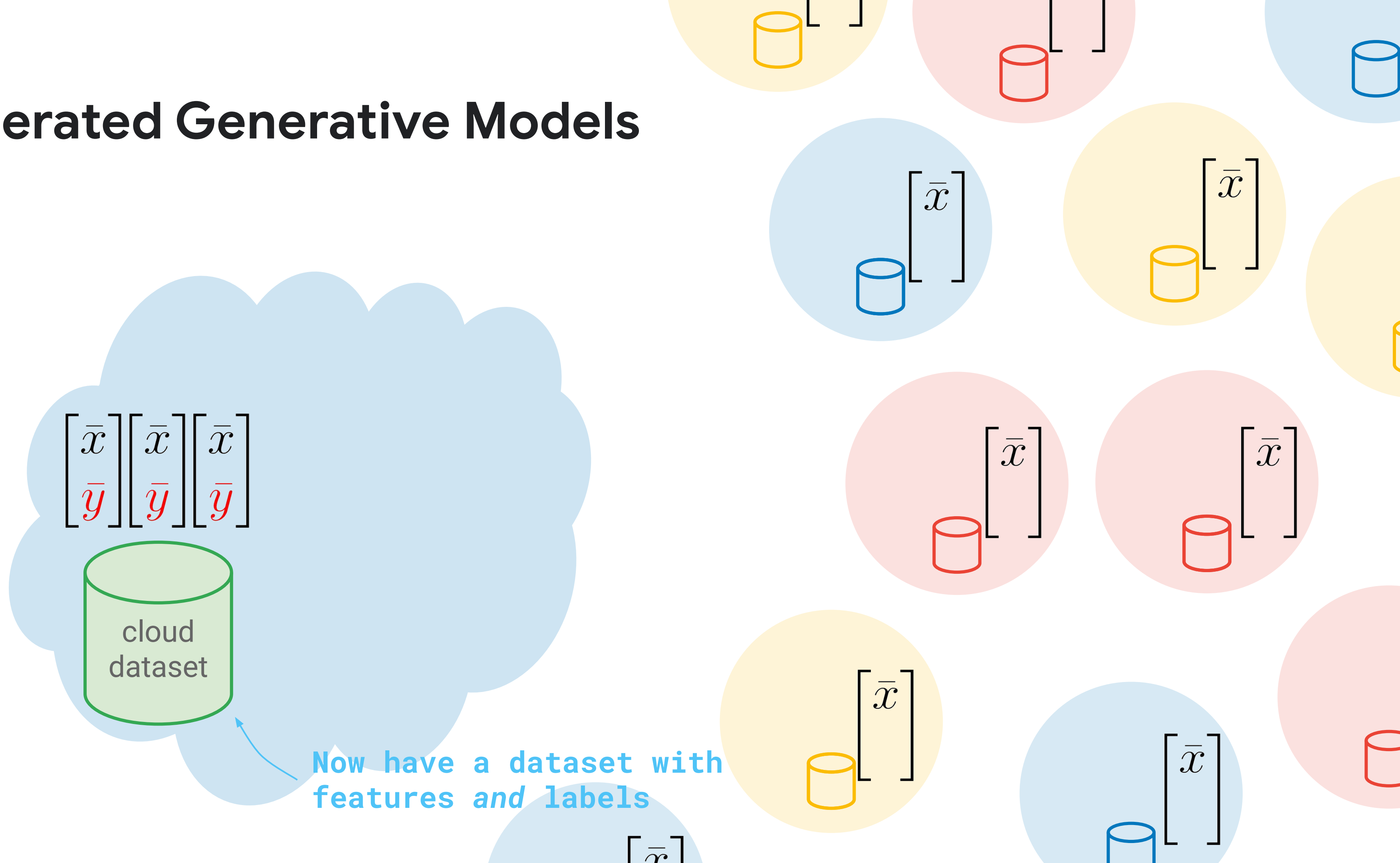
$$\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}\begin{bmatrix}\bar{x}\\ \bar{y}\end{bmatrix}$$

$$\begin{bmatrix}\bar{x}\\ \\ \end{bmatrix}\begin{bmatrix}\bar{x}\\ \\ \end{bmatrix}\begin{bmatrix}\bar{x}\\ \\ \end{bmatrix}$$

Use the DP Fed Generative Model to synthesis novel examples of features the developer can use (i.e., **label**)



Federated Generative Models



Now have a dataset with features *and* labels

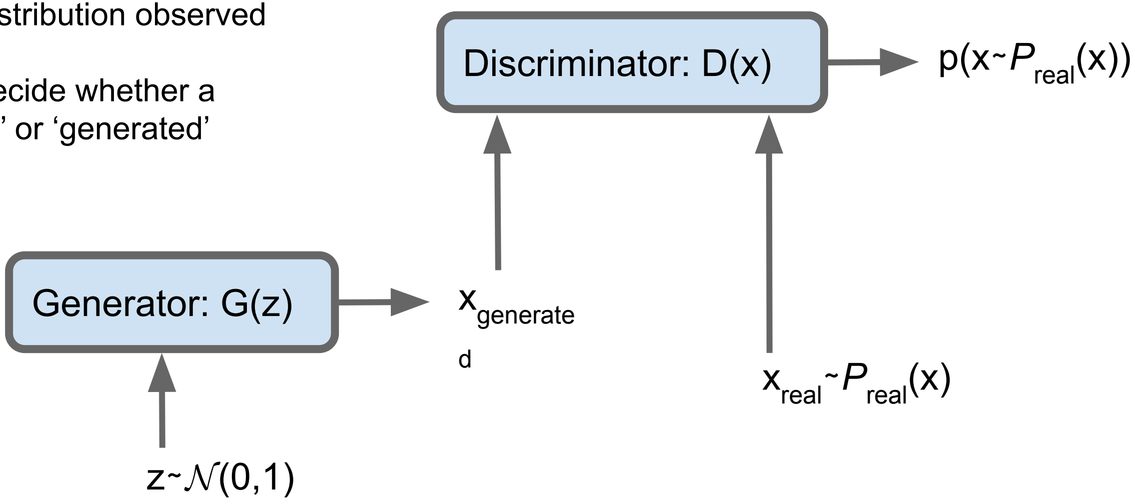
(Differentially Private) Federated GAN Algorithm

Quick Review of GANs

GANs: Generative Adversarial Networks

Two distinct NNs...

- ‘**G**’ tries to emit values that emulate a distribution observed in ‘real’ data
- ‘**D**’ tries to decide whether a value is ‘real’ or ‘generated’

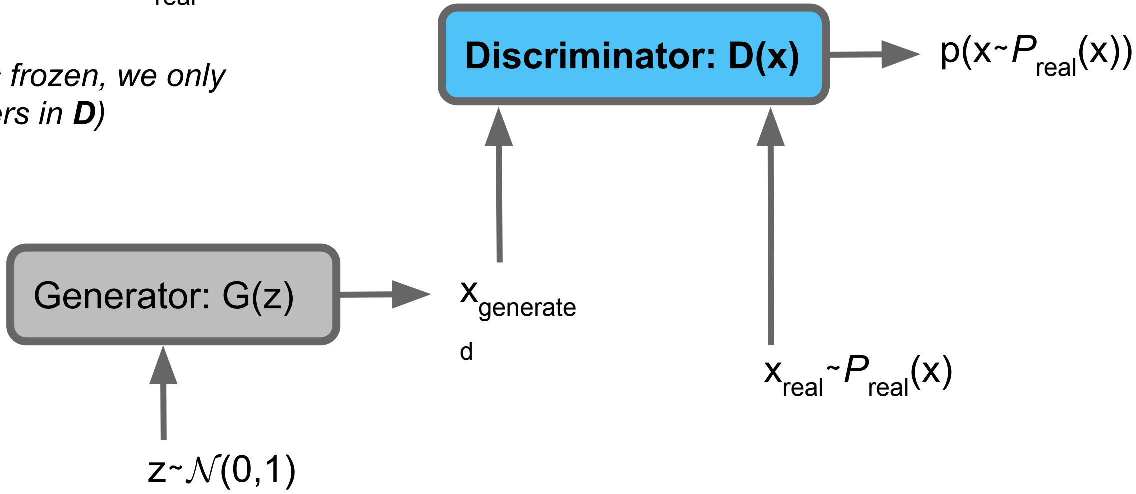


GANs: Generative Adversarial Networks

Step 1: Train the Discriminator

$$\textit{min } D(G(z)) + 1 - D(x_{real})$$

*(In this step, **G** is frozen, we only update parameters in **D**)*

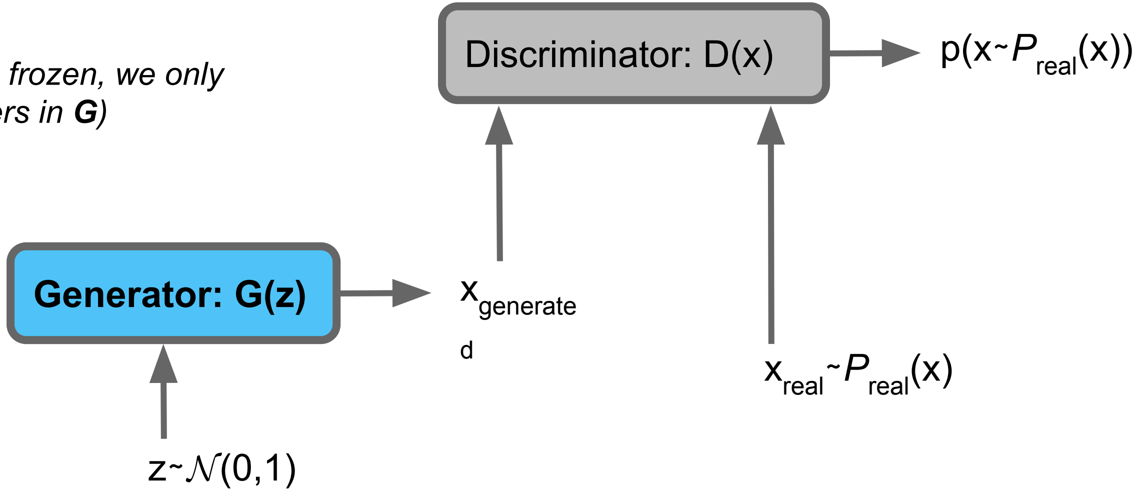


GANs: Generative Adversarial Networks

Step 2: Train the Generator

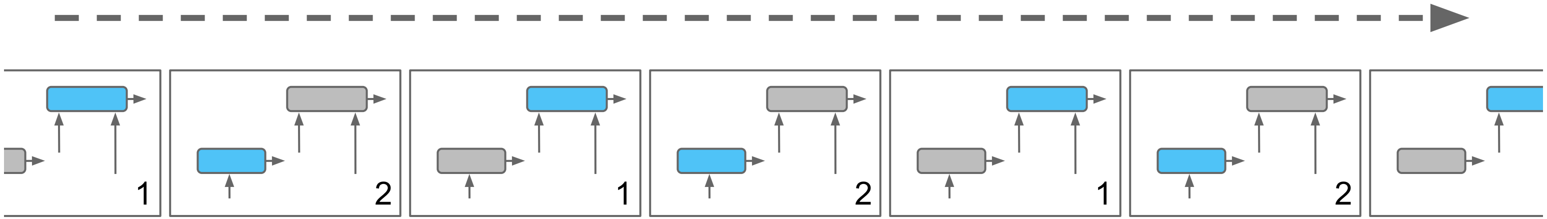
$$\max D(G(z))$$

*(In this step, **D** is frozen, we only update parameters in **G**)*



GANs: Generative Adversarial Networks

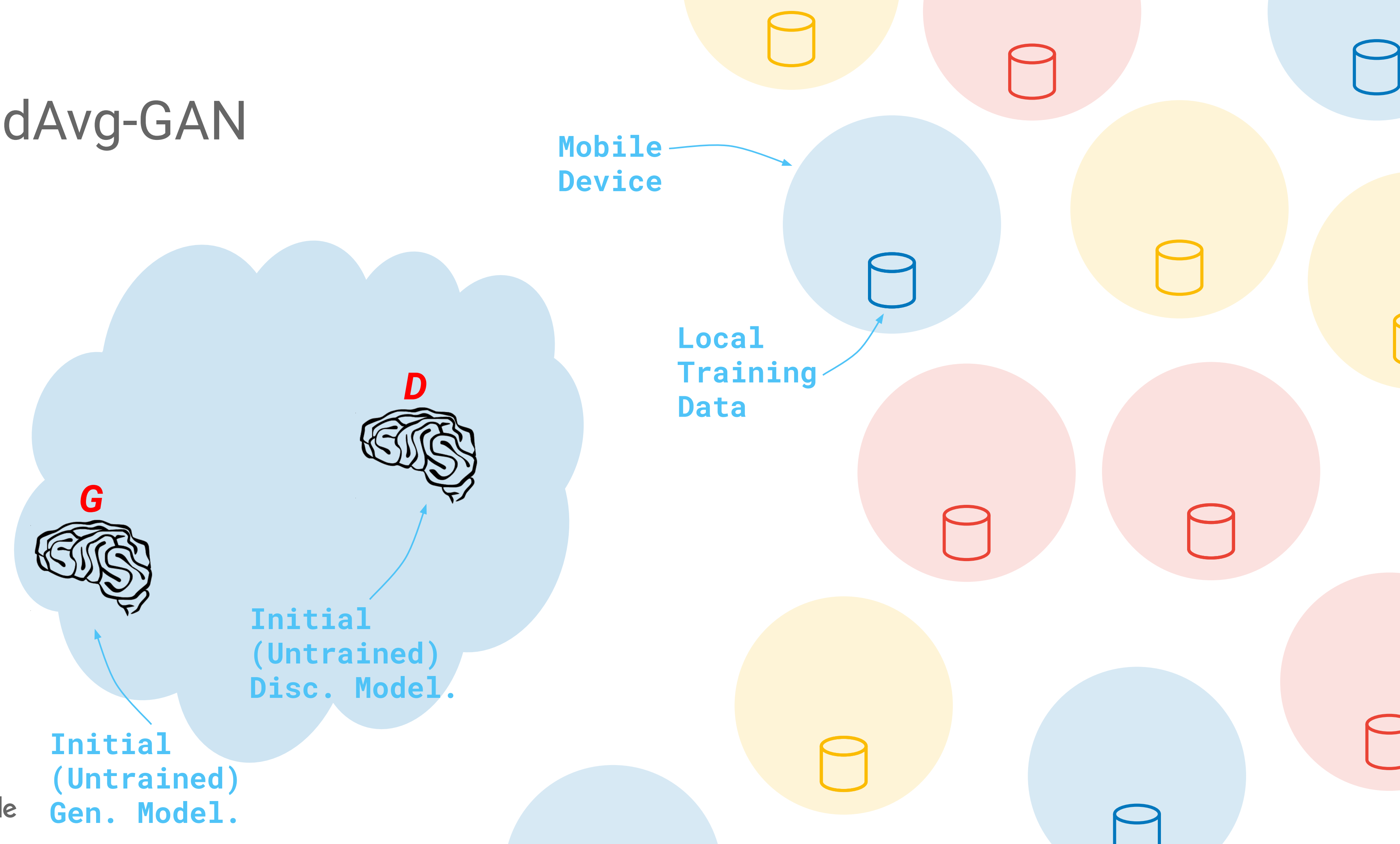
- Iteratively train the two NNs



- At convergence, you've got a NN ('**G**') which can generate novel instances that emulate the real world
    - E.g., generate novel images of human faces

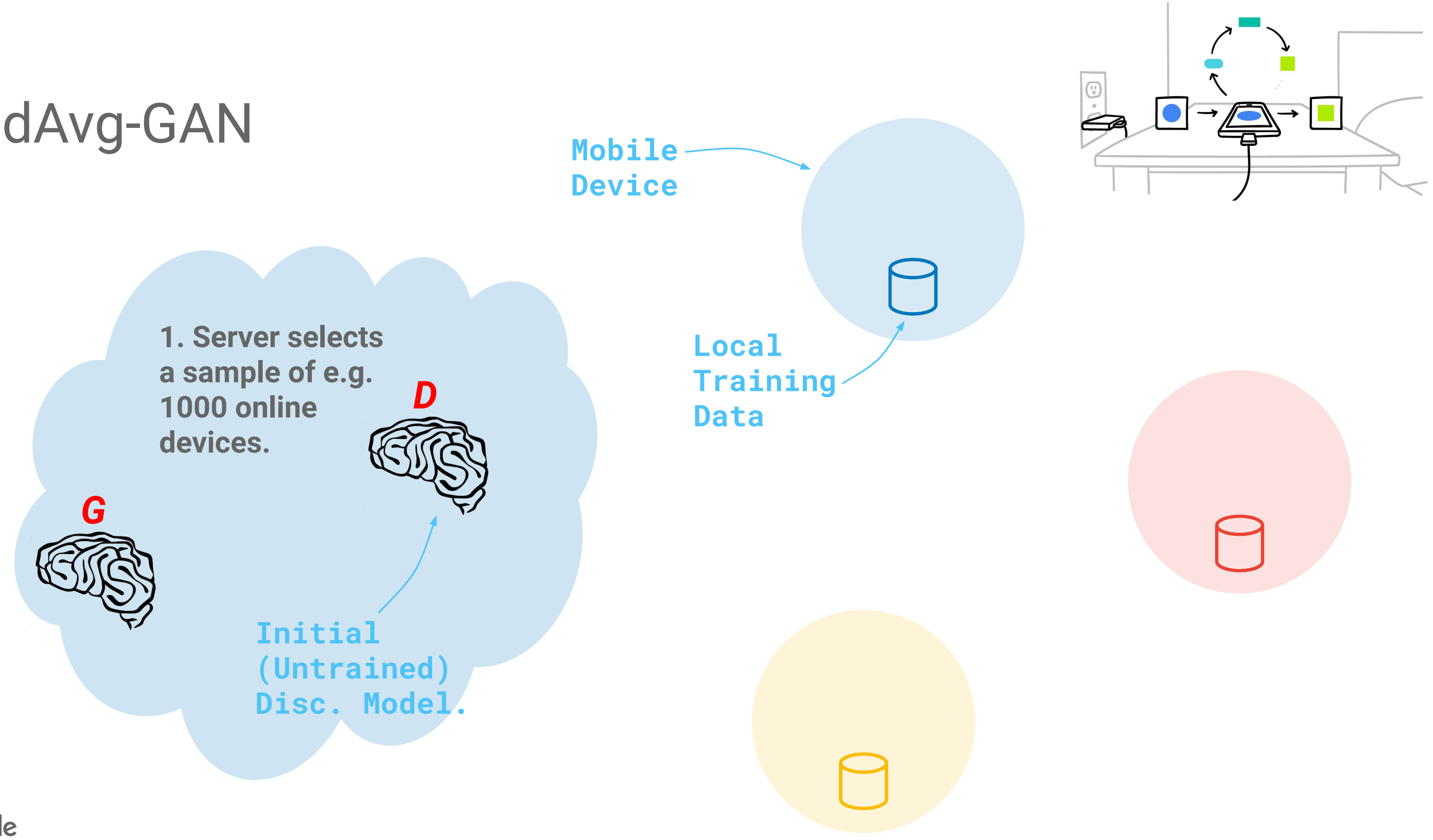
Google

FedAvg-GAN



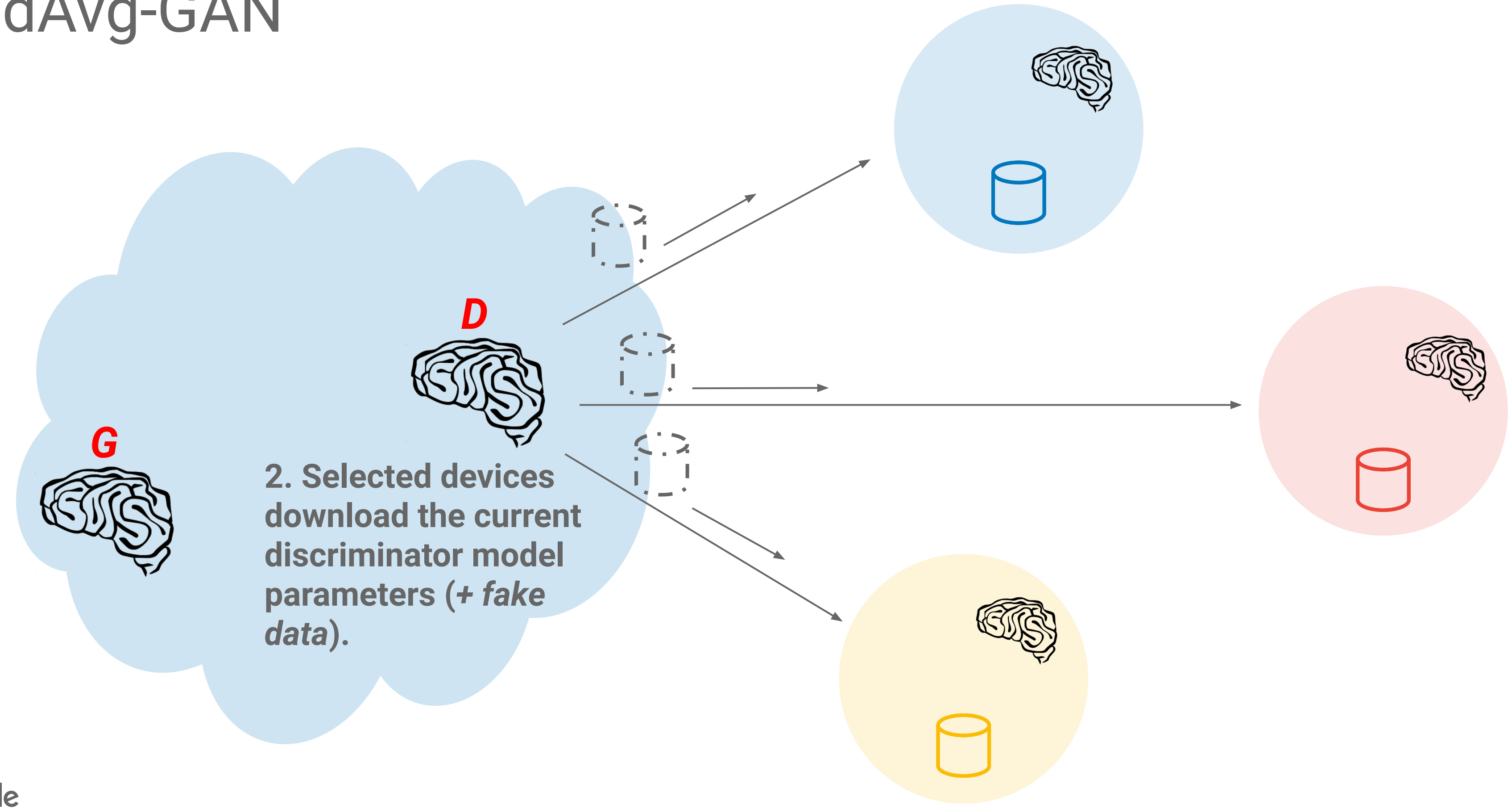
Google

FedAvg-GAN

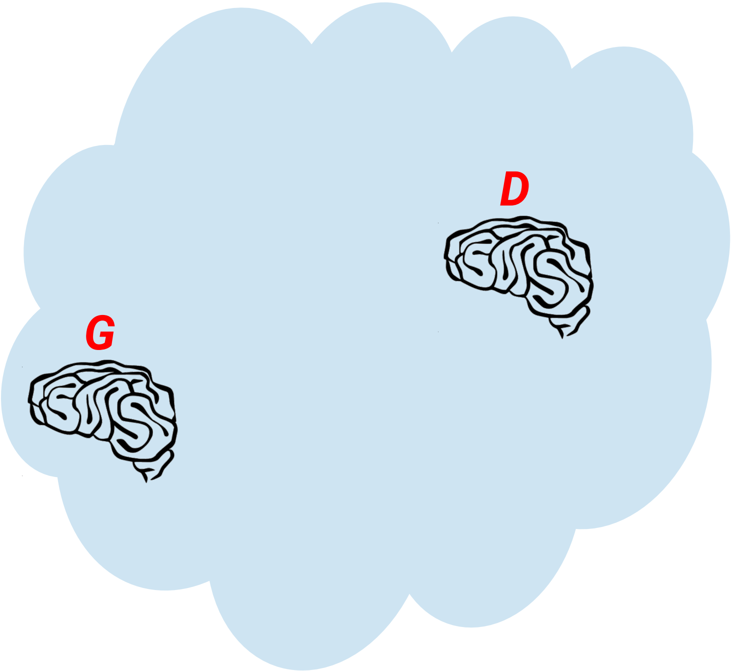


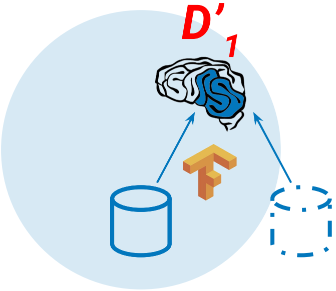
1. Server selects a sample of e.g. 1000 online devices.
D
G
Initial (Untrained) Disc. Model.
Mobile Device
Local Training Data

FedAvg-GAN

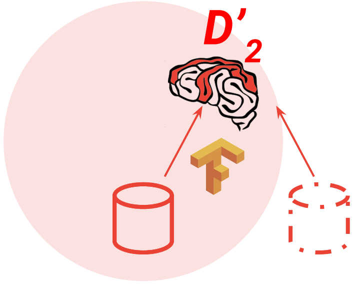

D
G
2. Selected devices download the current discriminator model parameters (+ *fake data*).

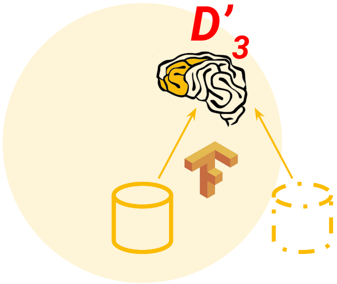
FedAvg-GAN



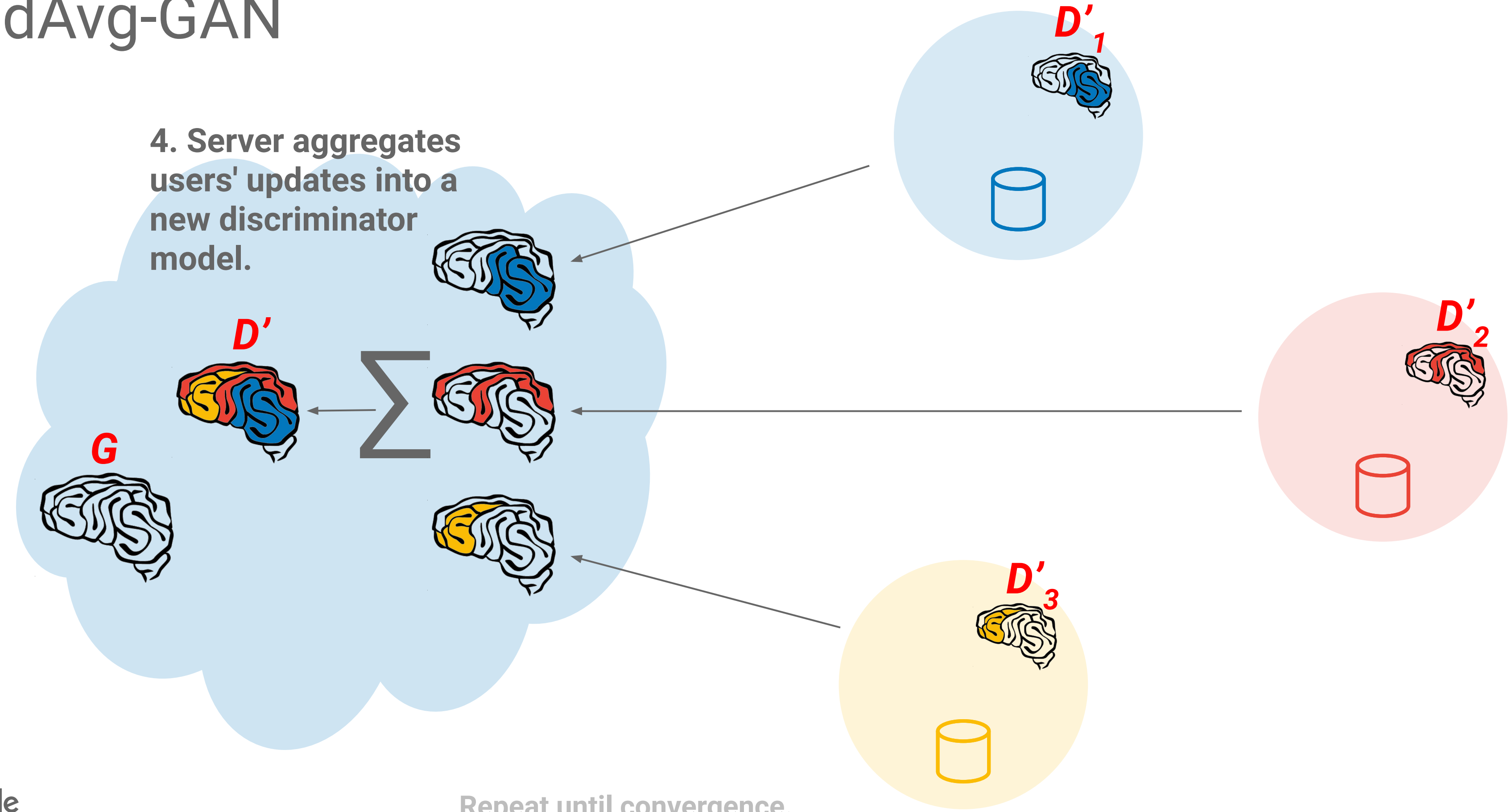


3. Users compute a discriminator update using local real training data (+ fake data)





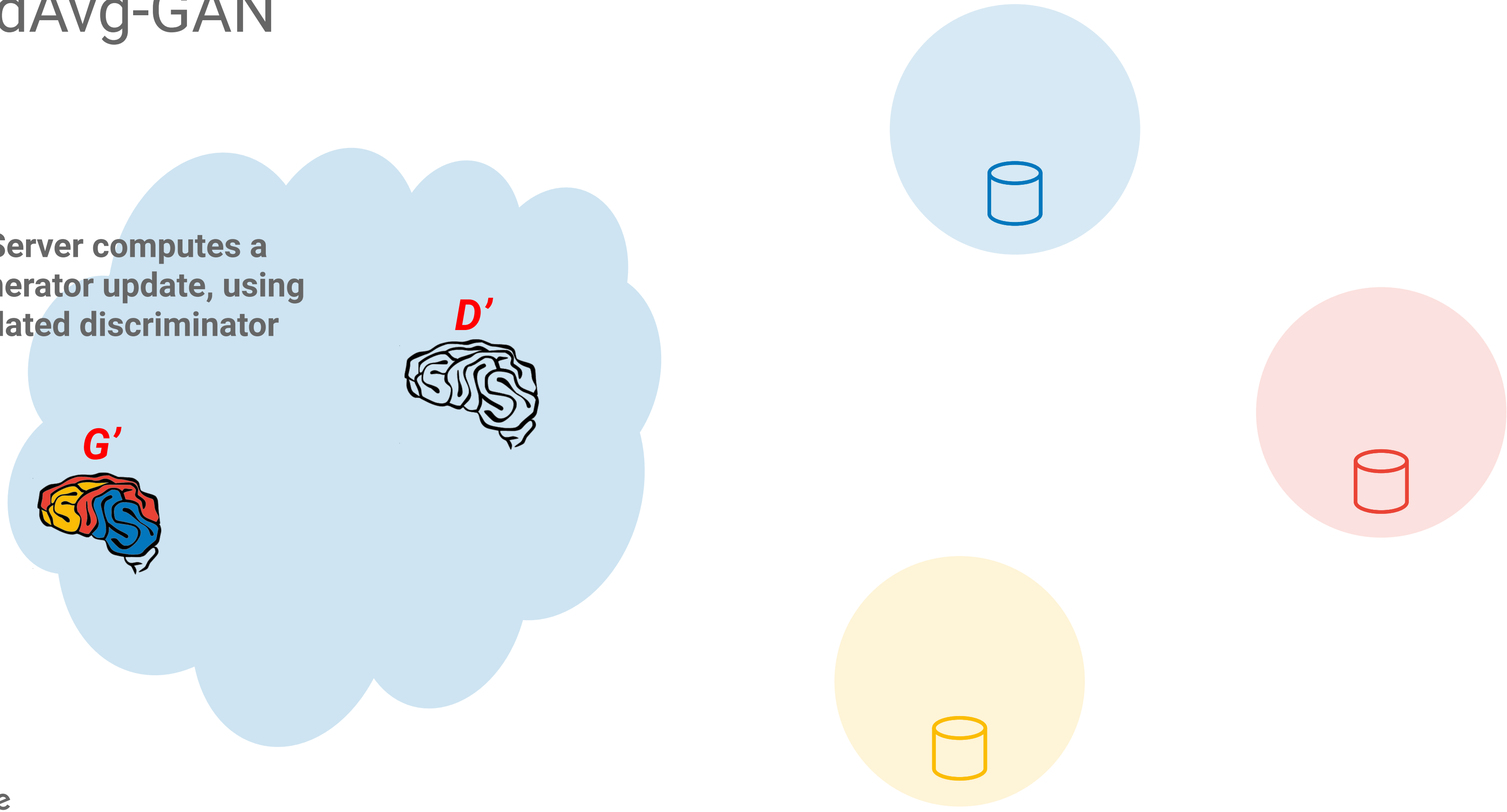
FedAvg-GAN



Repeat until convergence.

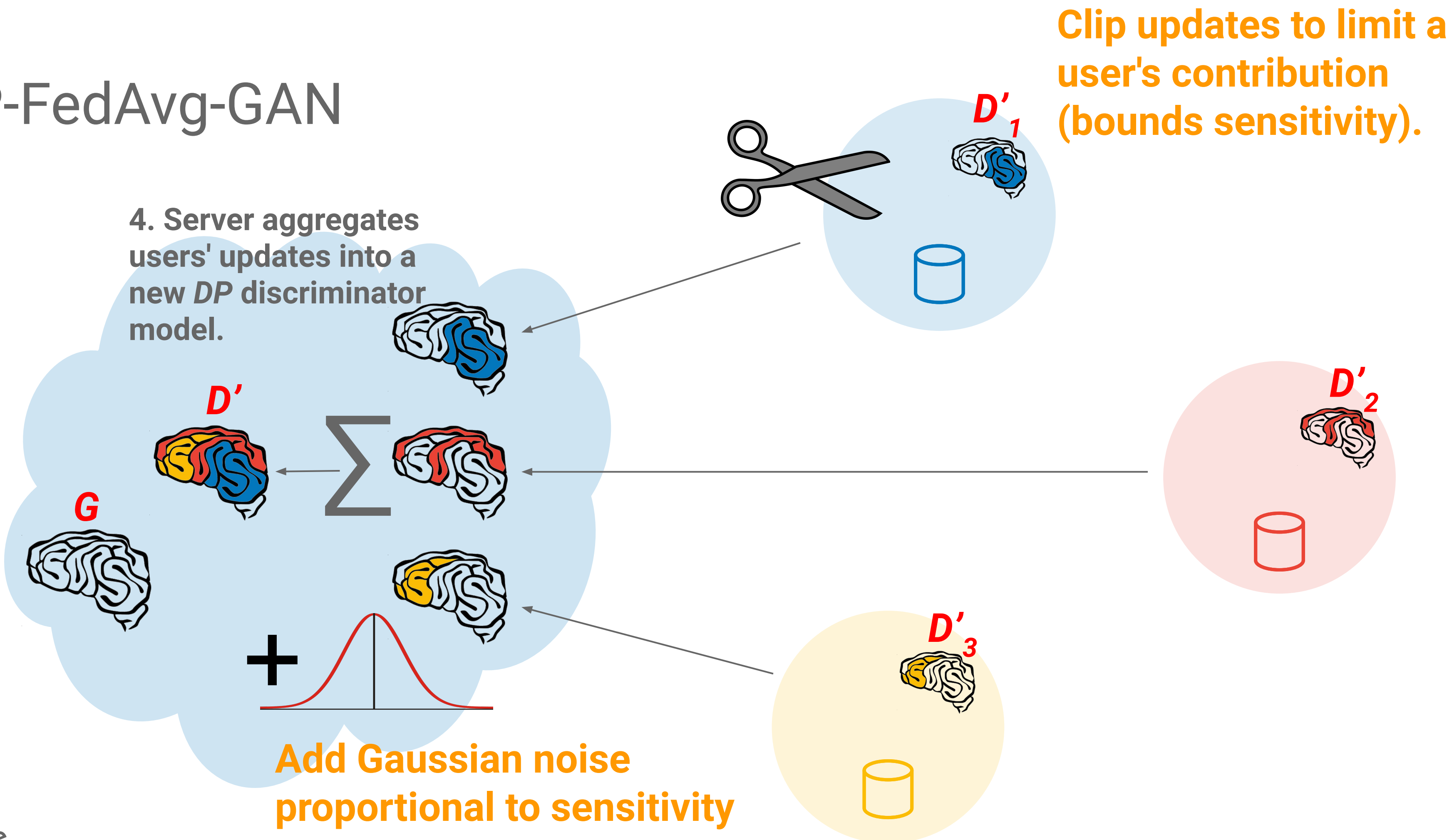
FedAvg-GAN

5. Server computes a generator update, using updated discriminator



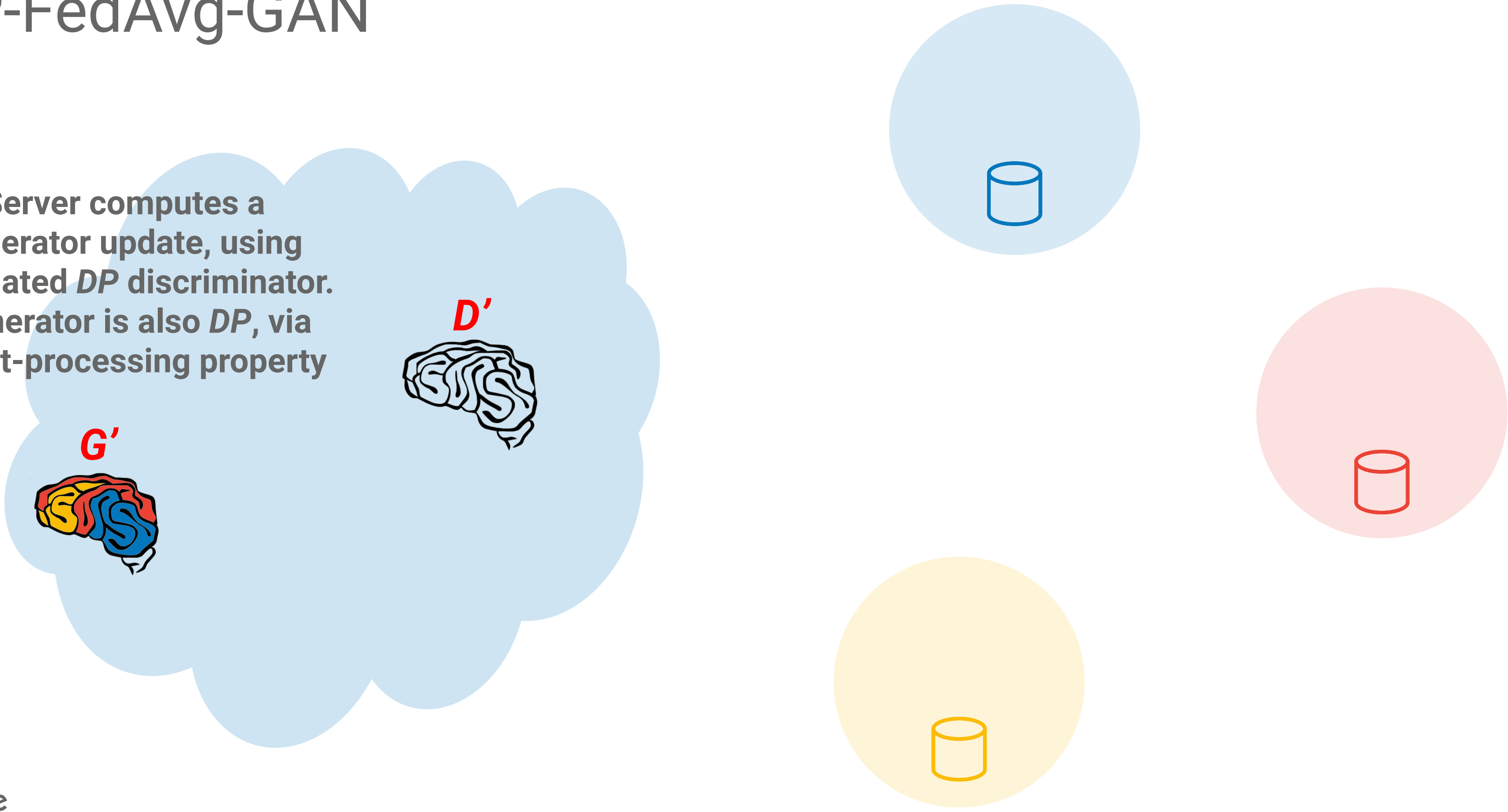
Google

DP-FedAvg-GAN



DP-FedAvg-GAN

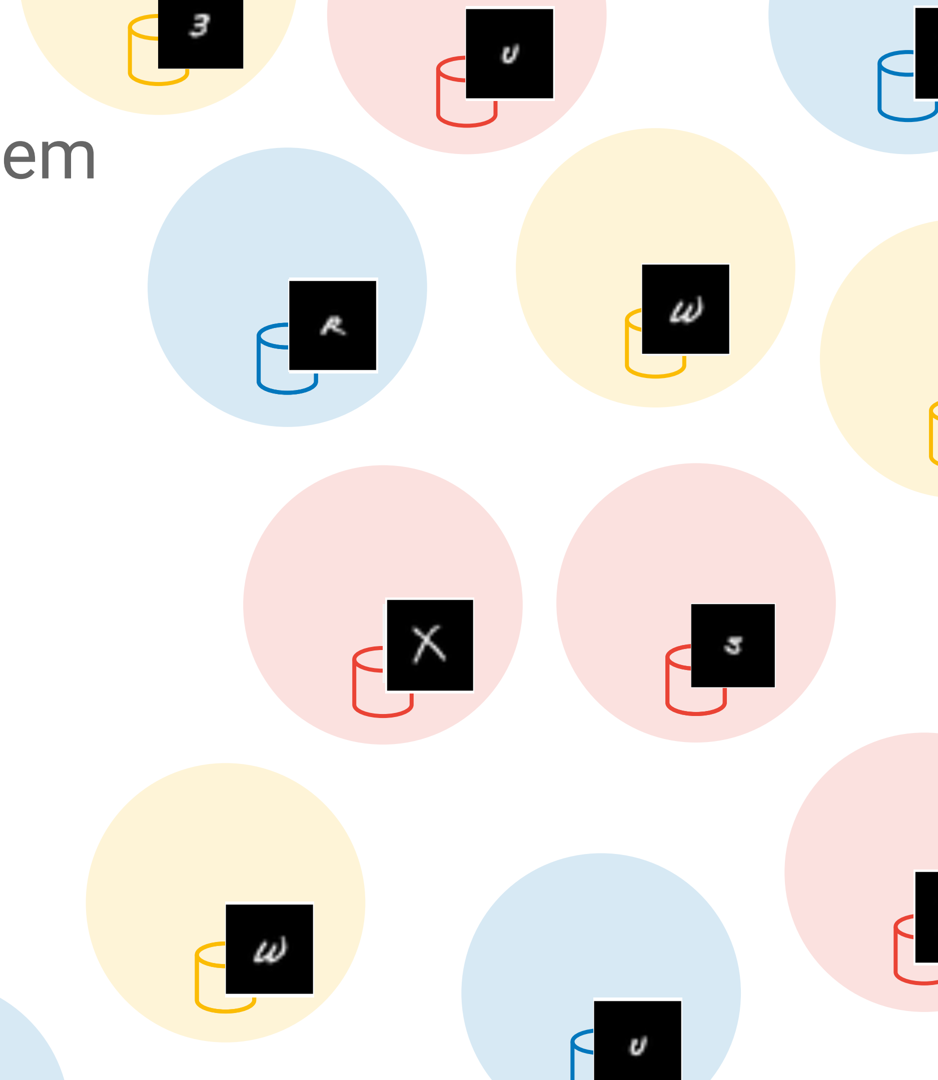
5. Server computes a generator update, using updated *DP* discriminator. Generator is also *DP*, via post-processing property



Federated GAN Example Problem: Debugging Image Classification

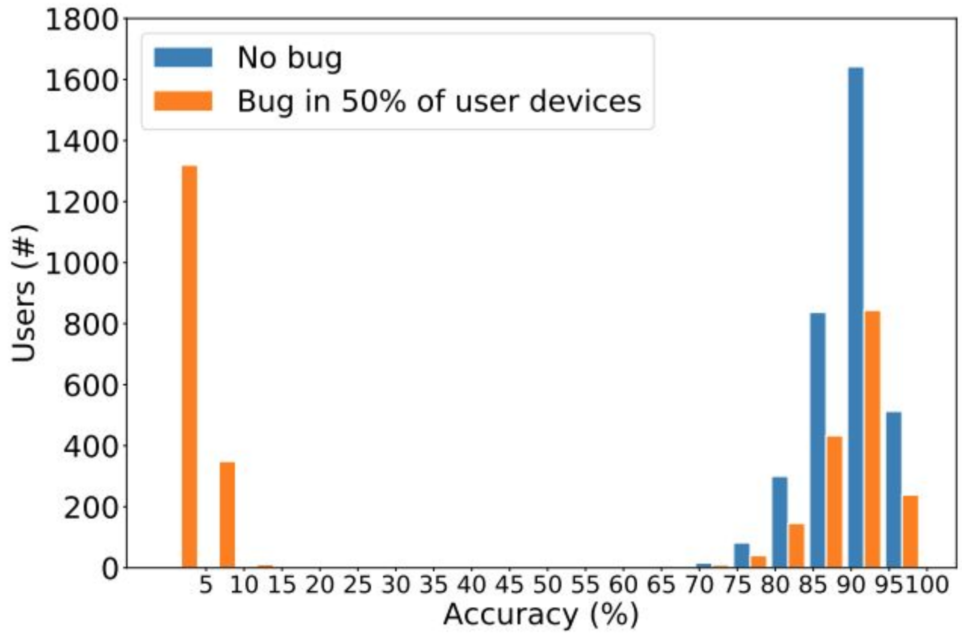
Example Federated GAN Problem

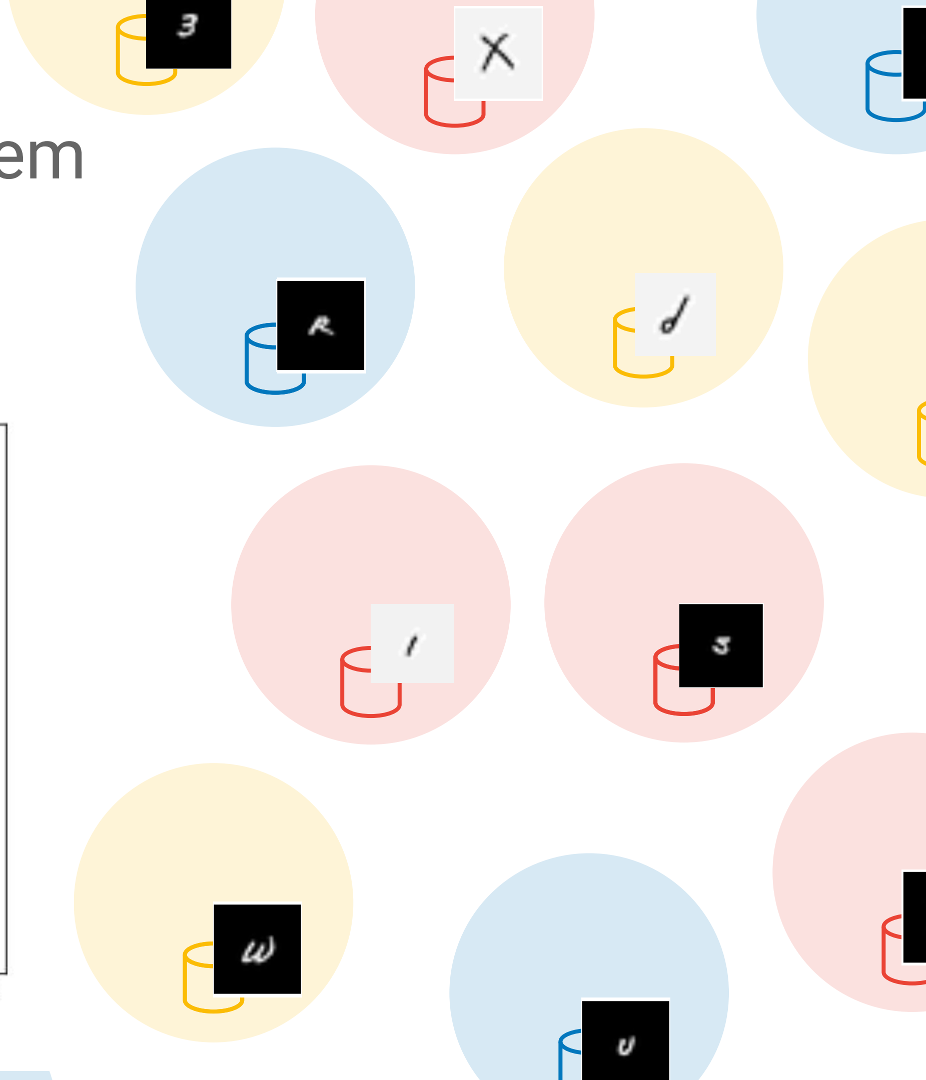
On-device inference network classifies handwritten numbers and letters. It expects raw images (from the upstream data pipeline) where background is black and character is white.



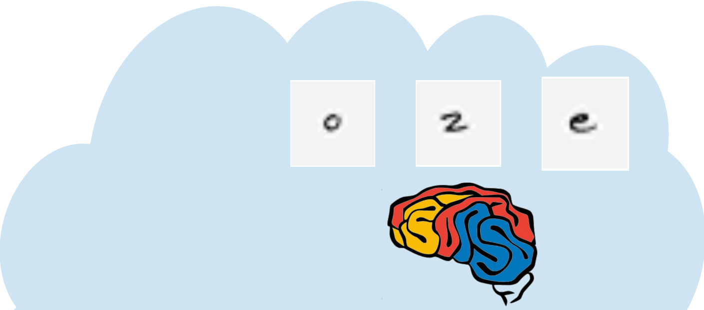
Example Federated GAN Problem

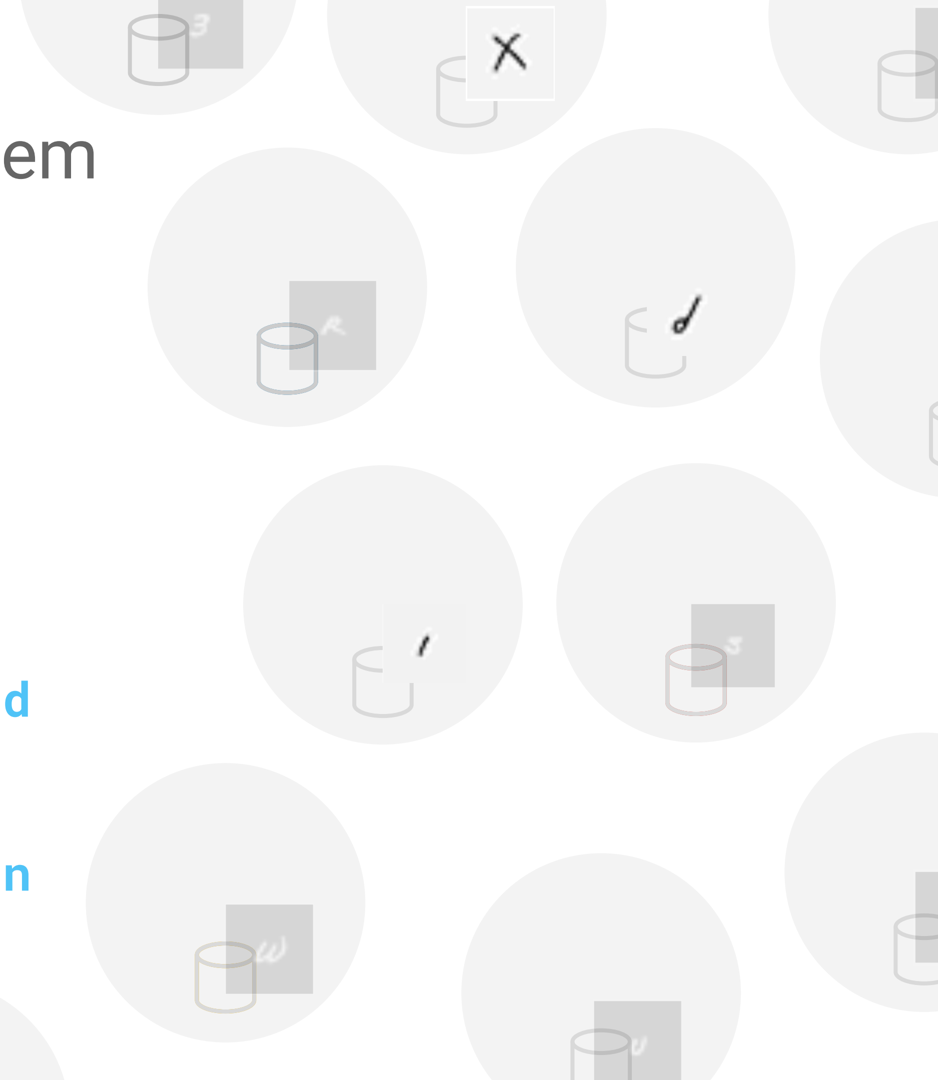
After application update, the classification accuracy drops





Example Federated GAN Problem





Train a DP Federated GAN and synthesize novel images (at the cloud) that match the characteristics of images in private dataset. Do this both for subsets with high and low class. accuracy.

Example Federated GAN Results

Population Description

Sub-Population Description

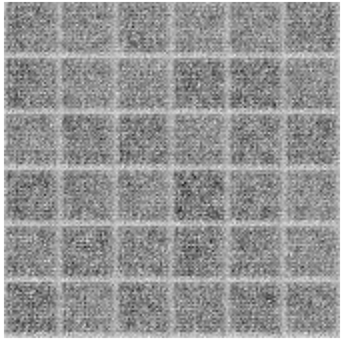
Example of Real Data on Devices in Sub-Population

GAN after 0 rds

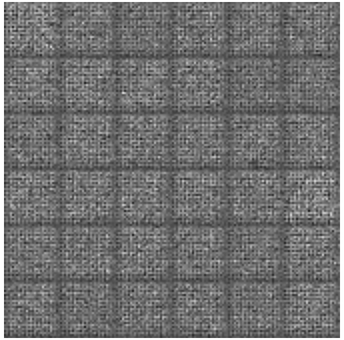
GAN after 1000 rds

EMNIST Dataset, 50% of Devices have their images ‘flipped’ (black<-> white)

Devices where data classifies with ‘low’ accuracy



Devices where data classifies with ‘high’ accuracy



Google

Example Federated GAN Results

Example of Real Data on Devices in Sub-Population

Now the modeler can discern ***this*** *difference ...*

... indicating that ***this*** *is the problem*

GAN after 1000 rds

Google

Conclusion

FL Research

FL Workshop in Seattle 6/17-18

2016 8 academic papers
2017 135
2018 256
2019 265 so far …

Multiple workshops and tutorials this year (CVPR, Google, IJCAI, NeurIPS, ...)

to pursue research in ML/data science for decentralized data with privacy by default.

to pursue research in ML/data science for decentralized data with privacy by default.

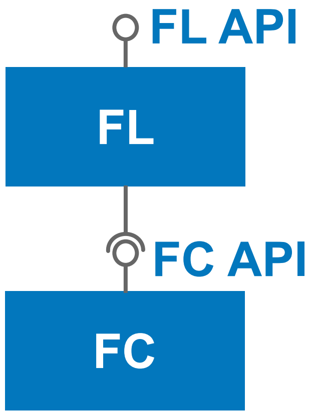
TensorFlow Federated (TFF)

An OSS framework for federated computation on decentralized data

tensorflow.org/federated
github.com/tensorflow/federated

TFF - What’s in the box

- Federated Learning (FL)
    - Implementations of federated training/evaluation
    - Can be applied to existing TF models/data
- Federated Core (FC)
    - Allows for expressing new federated algorithms
    - Local runtime for simulations



Google

```
train_data = … # uses tff.simulation.datasets.emnist.load_data()
model_fn = lambda: tff.learning.from_keras_model( … )

train = tff.learning.build_federated_averaging_process(model_fn)

state = train.initialize()
for _ in range(5):
  state, metrics = train.next(state, train_data)
  print (metrics.loss)

eval = tff.learning.build_federated_evaluation(model_fn)
metrics = eval(state.model, test_data)
```

TensorFlow Federated (TFF)

An OSS framework for federated computation on decentralized data

tensorflow.org/federated
github.com/tensorflow/federated