# Reliable and Trustworthy Artificial Intelligence

## Lecture 13: Federated Learning

Dimitar I. Dimitrov

ETH Zurich

Fall 2021

# Federated Learning Motivation

# Federated Learning - Motivation

Deep Learning requires a lot of task-dependent data to learn:

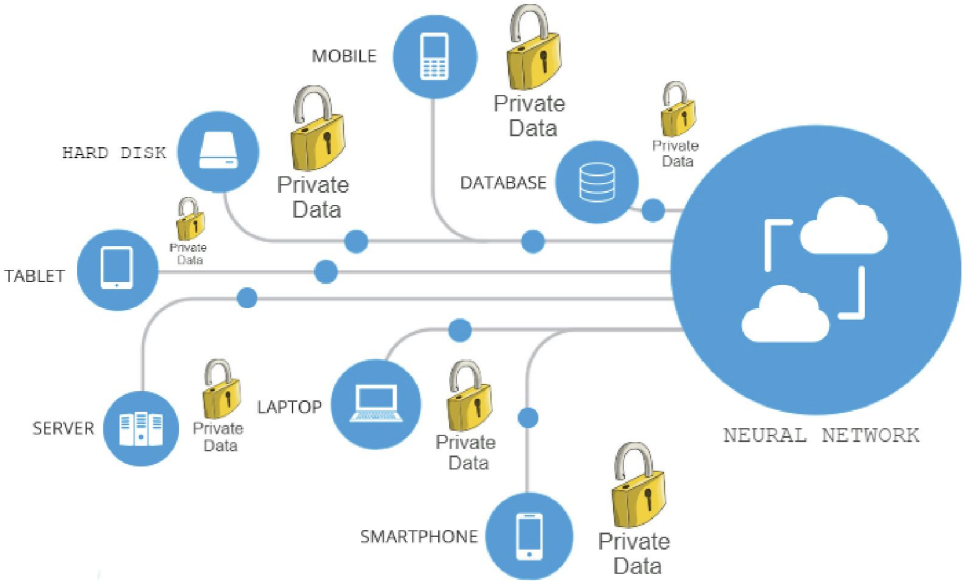
- Requires **access to a lot of data**:
  - **Only accessible to companies** with many data sources/users
  - **Privacy concerns** of the individual users
  - **Legal issues** surrounding the use of user data



- Or **scraping data** from the internet
  - **Copyright issues**
  - **Data might not be available** for the task being solved

# Federated Learning - Basic Idea

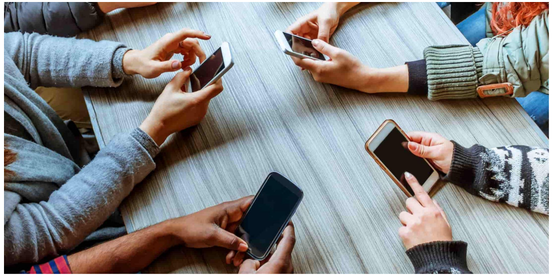
Can we learn without forcing individual data sources to share their data?
Yes, using federated learning



# Types of Federated Learning

## **Cross-device** setting:

- **Millions of sources** of data
- Each source is **contacted rarely** to participate in training
- Sources might **dynamically drop in and out** of the learning process
- There is **small amount of data** per-source
- **Example:** Google training spell checker on phone users



## **Cross-sillo** setting:

- **Small number of sources** of data
- Data sources participate in the training **constantly**
- **More data** per source
- Often at different sources **the data is heterogenous**
- **Example:** Hospitals jointly training a model to predict cancer from X-ray images

# Overview of Federated Learning

# Federated Learning - Overview

Elements:

- **Federated Server** - Stores the global neural network model and manages clients

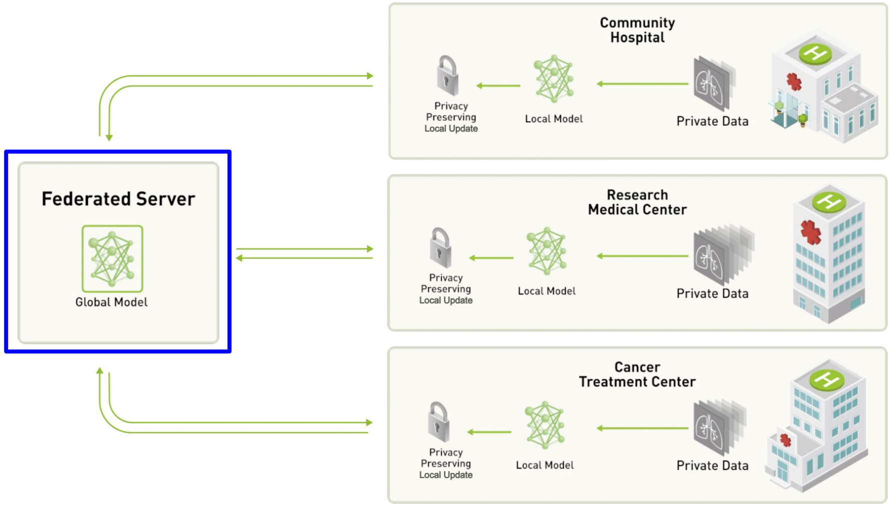


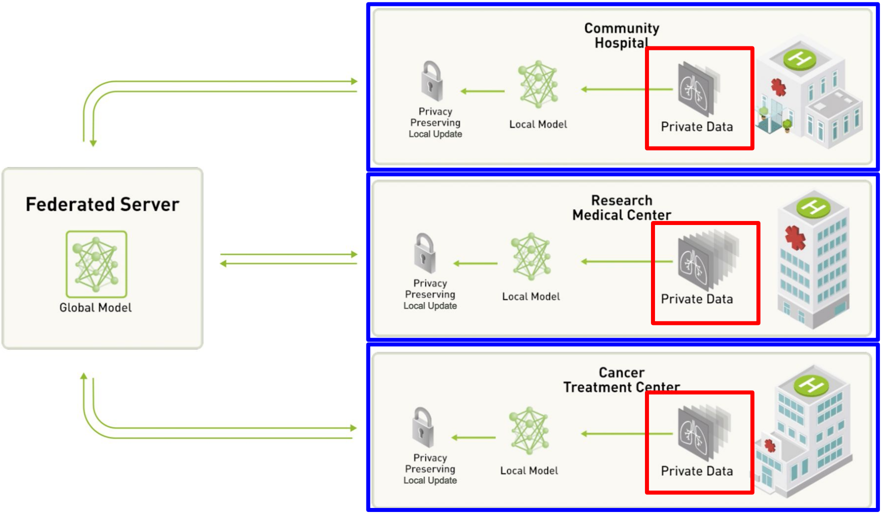



Image thaken from https://blogs.nvidia.com/blog/2019/10/13/what-is-federated-learning/

# Federated Learning - Overview

Elements:

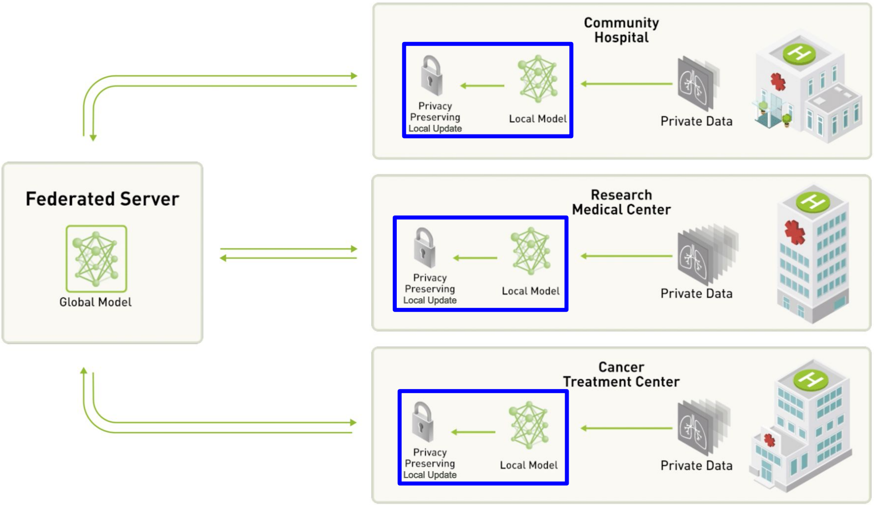
- Federated Server - Stores the global neural network model and manages clients
- **Individual Clients** - Use **private data** to train local models



# Federated Learning - Overview

Elements:

- Federated Server - Stores the global neural network model and manages clients
- Individual Clients - Use private data to train local models
- **Local Updates** - Computed from the local model to **preserve data privacy** and shared with server



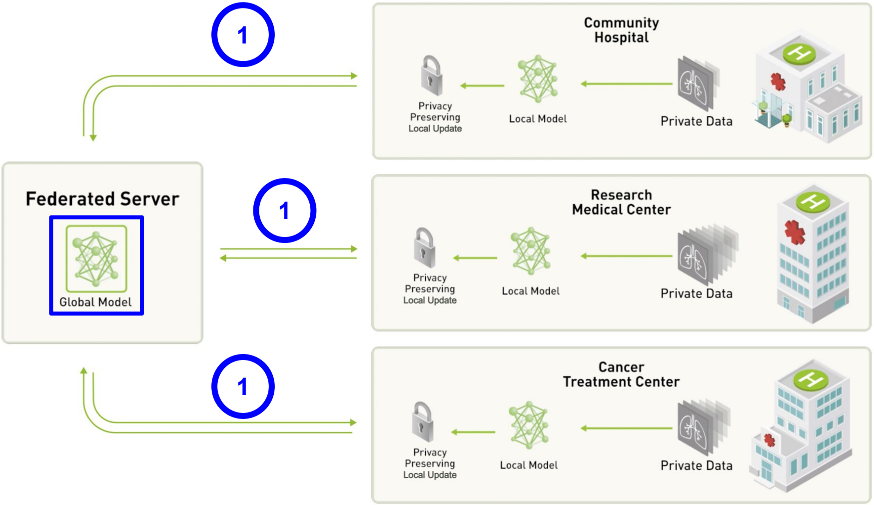
# A Round of Training in Federated Learning

## Step 1: Global Model sent to clients

Server needs to **decide which clients are selected** to participate in the round

Choice is can be based on:

- Cross-device vs Cross-silo?
- How often we have selected the clients so far?
- How much data is available at different client?
- How much the client has improved global model ?

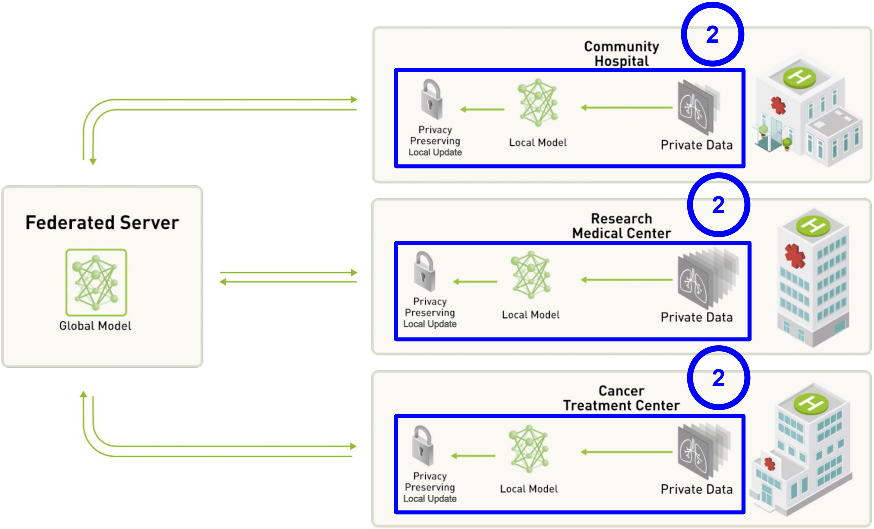


# A Round of Training in Federated Learning

## Step 2: Local computation on private data

- Update the global model using the **private data** to produce **local models**
- Use the **local models** to produce **local update**

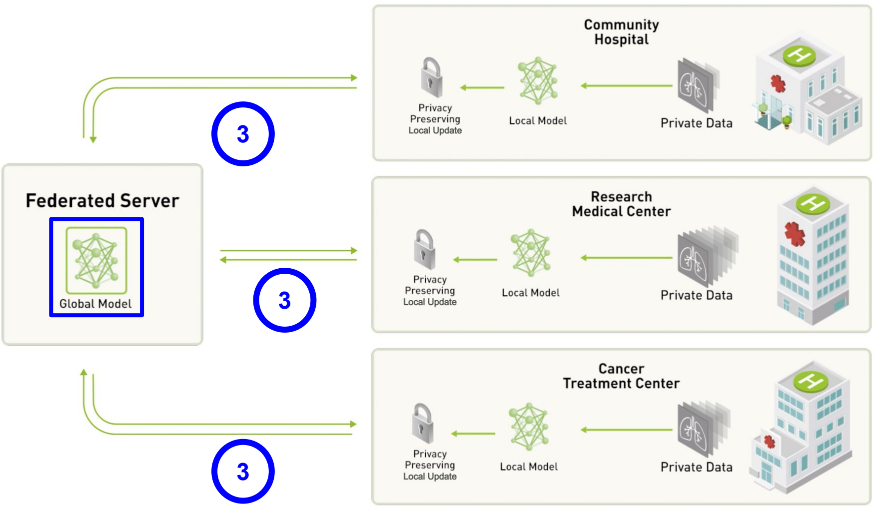
**Considerations** for constructing the local update:
- **Private data must not be exposed**
- Update needs to **improve performance on private data**



# A Round of Training in Federated Learning

## Step 3: Aggregation of Local Updates

- Clients send **information** to the Server:
  - **Local Updates**
  - **Local Training Statistics** - e.g number of datapoints used, number of local SGD iterations, batch norm statistics
- Server **aggregates the local updates** to produce a **new global model**:
  - Usually a weighted average of local updates
  - Weights selected based on the Local Training Statistics



# Common Federated Learning Algorithms - FedSGD

## **Client updates**:

- **Local private data** $\{x_i^k, y_i^k\}$ is chosen randomly
- **Local updates** are given by the **network gradient** $g_s$ with respect to the **global model weights** $\Theta_t$

$$\{x_i^k, y_i^k\} \sim \mathcal{D}_k$$
$$g_k \leftarrow \nabla_{\Theta_t} \mathcal{L}(f_{\Theta_t}(x_i^k), y_i^k)$$

## **Server aggregation:**

- The server applies the **average gradient update** $g_c$ to the global model using standard single step **SGD**

$$g_c \leftarrow \frac{1}{K} \sum_{k=1}^{K} g_k$$
$$\Theta_{t+1} \leftarrow \Theta_t - \gamma g_c$$

## **Pros:**

- Guarantees of **convergence** to a local minima

## **Cons:**

- Only single step of SGD before sending an update.**Communication is very expensive**



* McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

# Common Federated Learning Algorithms - FedAvg

**Client updates**:

- **Several iterations** of **SGD** on private data
- **Local updates** consist of **local model parameters** after SGD

**for** each local epoch $i$ from 1 to $E$ **do**

$\{x_i^k, y_i^k\} \sim \mathcal{D}_k$

$\Theta_{t+1}^k \leftarrow \Theta_{t+1}^k - \gamma \nabla_{\Theta_t} \mathcal{L}(f_{\Theta_t}(x_i^k), y_i^k)$

return $\Theta_{t+1}^k$

**Server aggregation:**

- Aggregation just **averages the weights** of the local models from different clients based on the number of examples in each client $n_k$

$$\Theta_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \Theta_{t+1}^k$$

**Pros:**

- Allows several SGD steps to be executed before communication which leads to **less communication** overhead

**Cons:**

- **No guarantees of convergence**



* McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.

# Attacks in Federated Learning

# Server-side Attacks: Gradient Inversion

Honest-but-Curious Server:

- **Aim:** Expose private data of clients
- **Allowed:** Look at clients' local updates and statistics, old global model parameters
- **Not Allowed:** Malicious updates of the global model

Client

$$f_\Theta(x_i) \xrightarrow{\text{Local FedSGD Update}} \nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i)$$

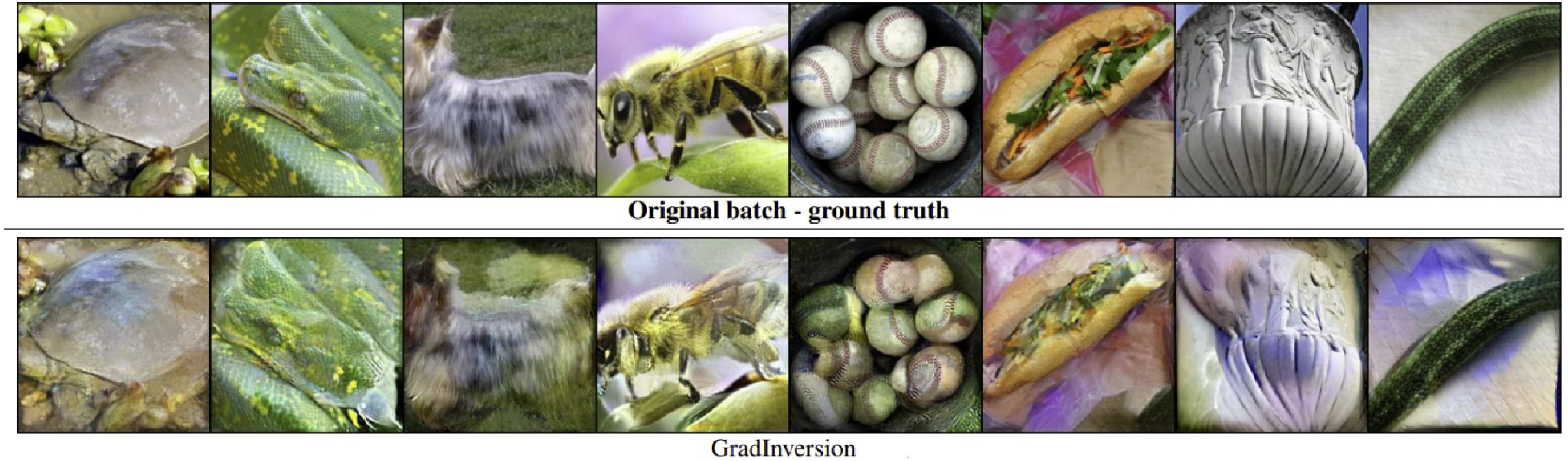
Server

$$\tilde{x}_i, \tilde{y}_i = \underbrace{g(\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i), f_\Theta)}_{\text{Gradient inversion function}}$$

$$g(\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i), f_\Theta) = \underset{(x_i^*, y_i^*)}{\operatorname{argmin}} \|\overbrace{\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i) - \nabla_\Theta \mathcal{L}(f_\Theta(x_i^*), y_i^*)}^{\text{Match gradients of reconstructed and client data}}\|_p + \overbrace{\mathcal{R}(x_i^*)}^{\text{Regularizer to enforce realistic reconstructed images}}$$

# Server-side Attacks: Gradient Inversion SOTA

Gradients leak a lot of information about the client data




https://www.sri.inf.ethz.ch/

*Yin, Hongxu, et al. "See through Gradients: Image Batch Recovery via GradInversion." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

# Gradient Inversion - Our work

Probabilistic view of the gradient inversion function:

- We assume that the client sent gradient $g_k$ is a corrupted version of the true gradient $\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i) \longrightarrow$ **Many defenses can be interpreted that way**
- We show that under these circumstances existing attacks can be viewed as the **Bayesian optimal adversary**:

$$g(\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i), f_\Theta) = \underset{(x_i^*, y_i^*)}{\text{argmax}} \underbrace{p(\nabla_\Theta \mathcal{L}(f_\Theta(x_i^*), y_i^*) = g_k | x_i^*, y_i^*)}_{\text{Posterior probability of observing the gradient the client supplied with current data estimate}} \bullet \underbrace{p(x_i^*)}_{\text{Image prior probability}}$$

$$g(\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i), f_\Theta) = \underset{(x_i^*, y_i^*)}{\text{argmin}} \|\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i) - \nabla_\Theta \mathcal{L}(f_\Theta(x_i^*), y_i^*)\|_p + \mathcal{R}(x_i^*)$$

**Key point: Many existing attacks including SOTA can be seen as instantiations of our framework**

# Gradient Inversion - Our work

## How to select prior and posterior?

- **Strong image priors** like PixelCNN are preferable. Affects results drastically
- Model **posterior** probability based on **defense mechanism** applied

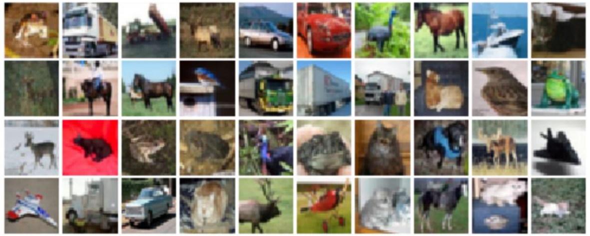
$$g(\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i), f_\Theta) = \underset{(x_i^*, y_i^*)}{\operatorname{argmax}} \; \overbrace{p(\nabla_\Theta \mathcal{L}(f_\Theta(x_i^*), y_i^*) = g_k | x_i^*, y_i^*)}^{\text{Posterior probability of observing the gradient the client supplied with current data estimate}} \cdot \overbrace{p(x_i^*)}^{\text{Image prior probability}}$$

**Key point: Optimal attack depends on the defense**

# Gradient Inversion - Our work

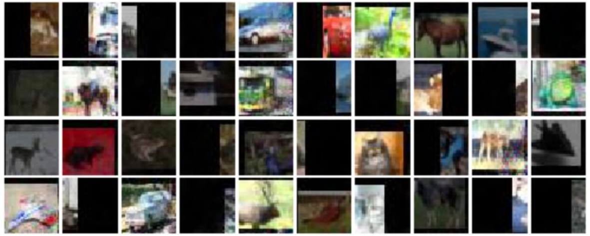
## Experiments

- We use Bayes Optimal attack to break several heuristic defenses



(a) Original

(b) Soteria



(c) ATS

(d) PRECODE

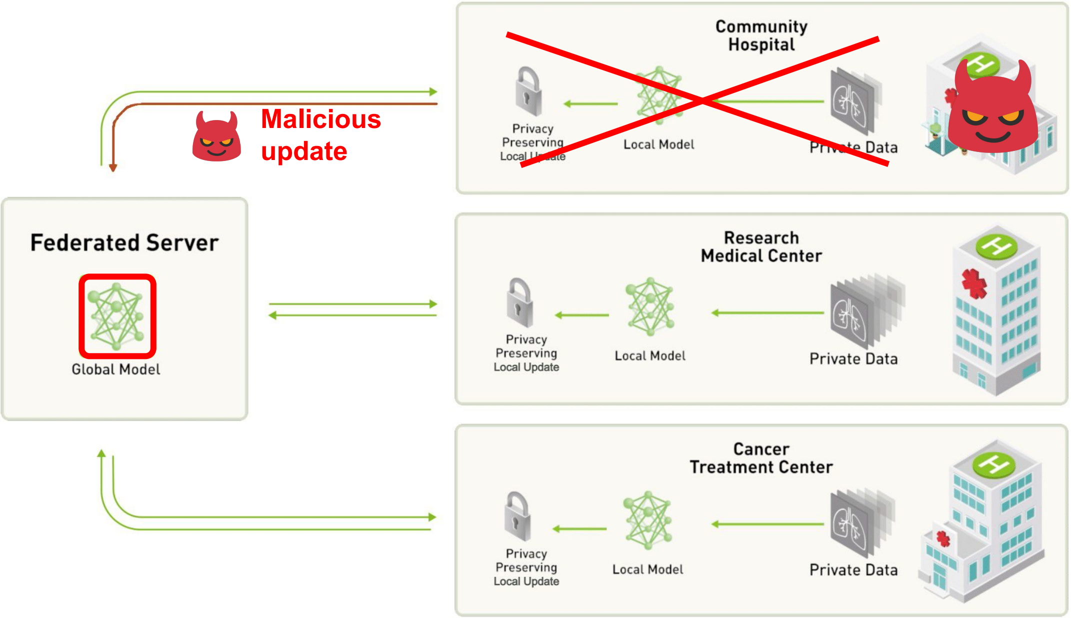


Balunović, Mislav, et al. "Bayesian Framework for Gradient Leakage." *arXiv preprint arXiv:2111.04706* (2021)

# Client-side Attacks: Poisoning

## Client Poisoning Attack:

- **Aim:** Force to model to diverge or to have bad behaviour on certain data
- **Allowed:** Send malicious local updates to the server
- **Allowed:** Coordination between multiple malicious clients
- **Not Allowed:** Changes to the model aggregation scheme



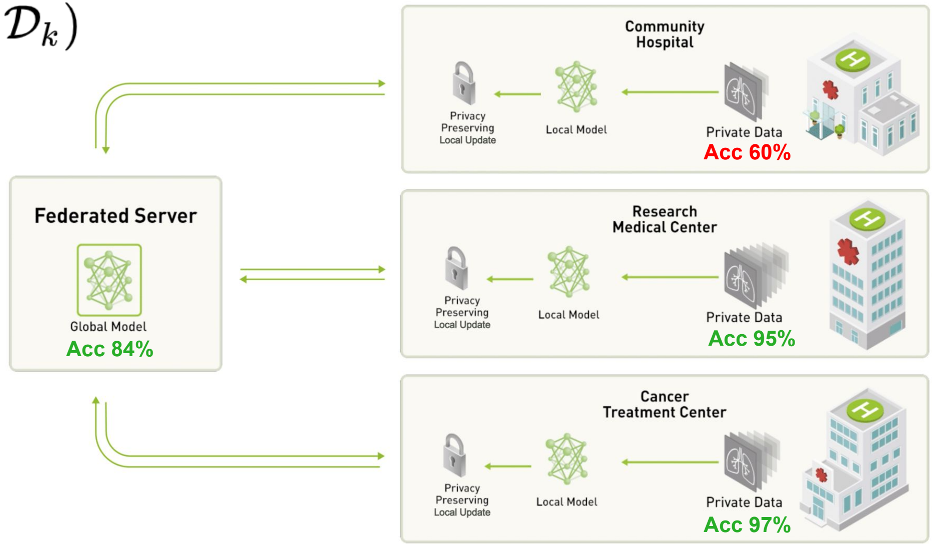
# Other Concerns: Client Fairness

FedSGD optimizes the average of individual client losses:

$$\underset{\Theta}{\operatorname{argmin}}\, \mathcal{L}(\Theta, \mathcal{D}) = \frac{1}{k} \sum_{k=1}^{K} \mathcal{L}(\Theta, \mathcal{D}_k)$$

**Issue:** Possibly big **variance of accuracy** of the global model for **individual clients** despite **good performance on average**

**Cause: Data heterogeneity**

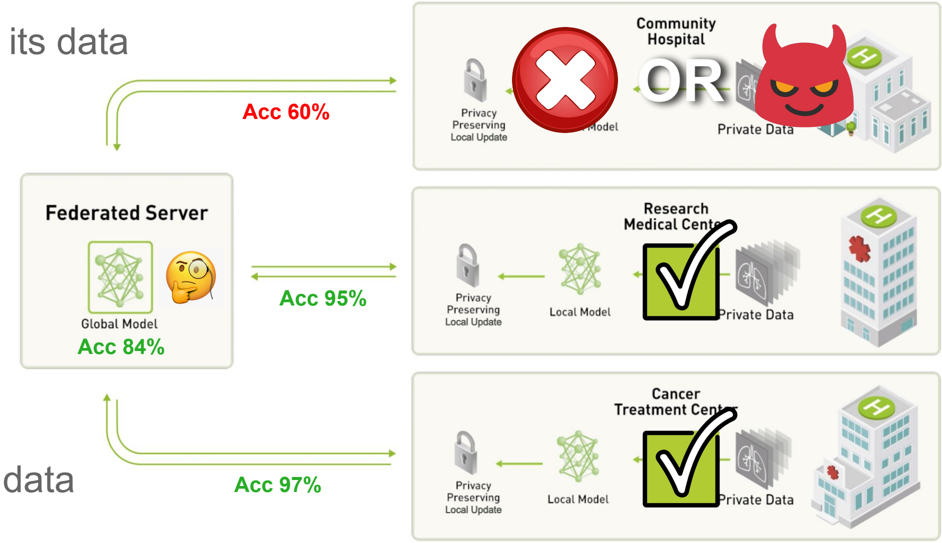


# Trade-off: Client Fairness vs Robustness to Poisoning

**Issue:** Server **cannot** distinguish between **highly heterogeneous** client and **adversarial** client

**Case 1:**
Server **allows the client to adapt** the model more to its data
⟹ **Adversary** will affect global model **more easily**

**Case 2:**
Server **prevents the client to adapt** the model to its data
⟹ **No Client Fairness**



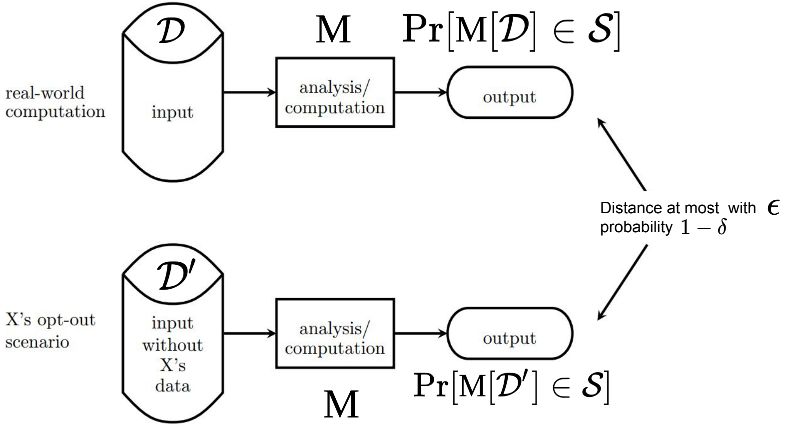
# Defenses in Federated Learning

# Defenses against Gradient Inversion - Differential Privacy

## Formal Definition of Differential Privacy:

The stochastic algorithm $\mathrm{M}$ is differentiably private with parameters $\epsilon$ and $\delta$ if for all similar input datasets $\mathcal{D}$ and $\mathcal{D}'$ and all possible outputs $\mathcal{S}$ of the algorithm, it is true that:

$$\Pr[\mathrm{M}[\mathcal{D}] \in \mathcal{S}] \leq \exp[\epsilon] \Pr[\mathrm{M}[\mathcal{D}'] \in \mathcal{S}] + \delta$$




## Intuitive Definition of Differential Privacy:

**Small changes of the input** are **indistinguishable** to observer that only see the output of the algorithm **with high probability**

# Defenses against Gradient Inversion - DP-SGD

**Key Idea:** DP-SGD **clips the gradients** of individual clients and **adds noise** to them to achieve differential privacy. **Only the modified gradients are sent to server**

$$g_k = \underbrace{\min\left(\frac{C}{\|\nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i)\|_2}, 1\right) \cdot \nabla_\Theta \mathcal{L}(f_\Theta(x_i), y_i)}_{\text{Clip gradient norm to } C} + \underbrace{\xi_i \quad \text{with} \quad \xi_i \sim \mathcal{N}(0, \sigma^2 C^2 I)}_{\text{Add Gaussian noise}}$$

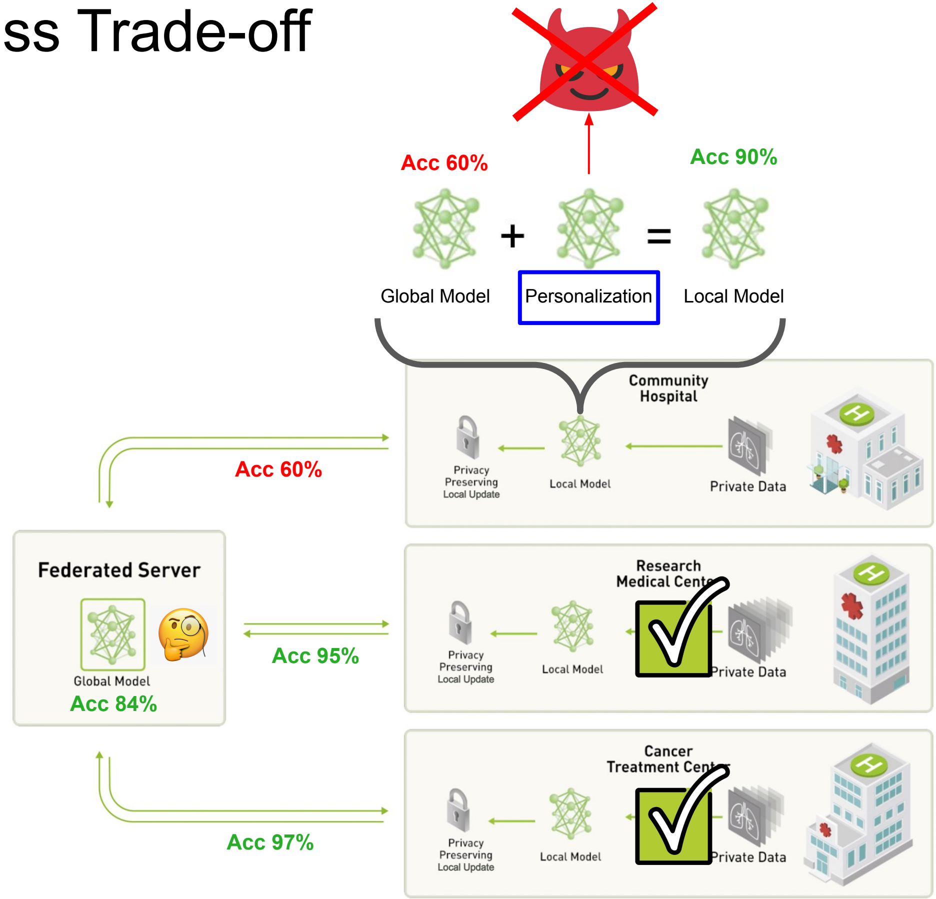
$\sigma$ is a complicated function of desired $\epsilon$ and $\delta$

# Solving the Robustness vs Fairness Trade-off

**Key Idea:** Model personalization

## Advantages:

- Model is allowed to personalize the global model locally
  $\Longrightarrow$ **Client Fairness**
- Personalizations are **only** local
  $\Longrightarrow$ **Adversary cannot use them to corrupt the model**



# Future work

## GradInversion:

- Beyond images (e.g **Text, Audio**) - Deng, Jieren, et al. "TAG: Gradient Attack on Transformer-based Language Models." Findings of the Association for Computational Linguistics: EMNLP 2021. 2021.
- Adapting to **FedAvg** - Geng, Jiahui, et al. "Towards General Deep Leakage in Federated Learning." arXiv preprint arXiv:2110.09074 (2021).
- Defenses **beyond DP-SGD** - Can we exploit more **problem-specific information**?

## Poisoning Attacks:

- Current attacks are successful only under **unrealistic assumptions** - Shejwalkar, Virat, et al. "Back to the drawing board: A critical evaluation of poisoning attacks on federated learning." arXiv preprint arXiv:2108.10241 (2021).
- Exploring **different model personalization schemes** - active area of research


https://www.sri.inf.ethz.ch/