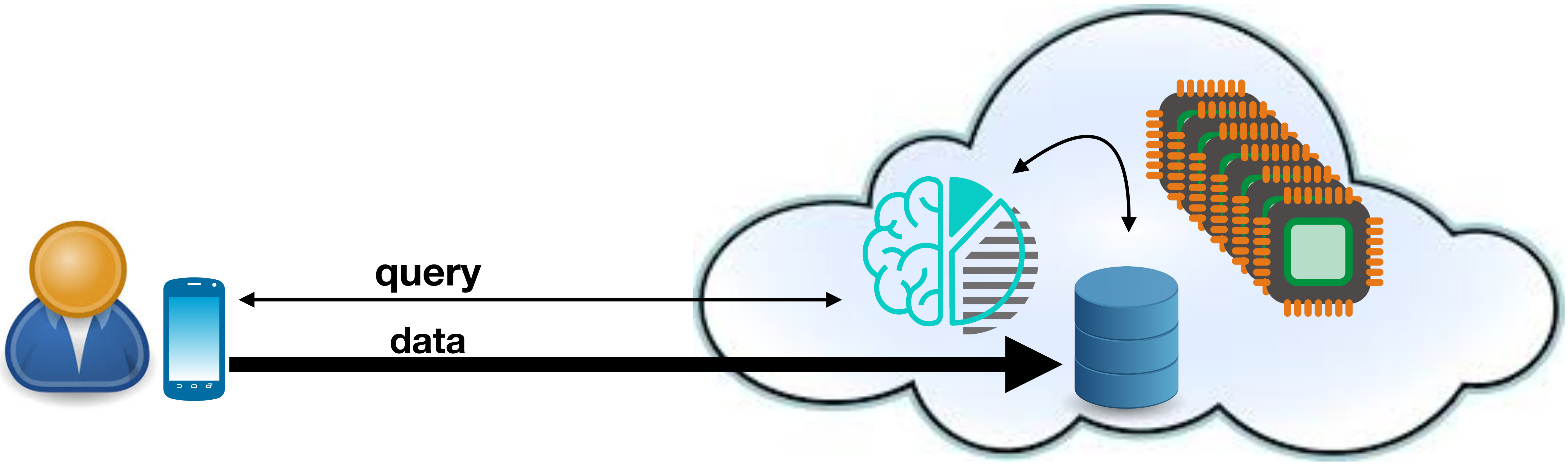
Privacy Preserving AI
Federated Learning
and
Homomorphic Encryption

romeo kienzler - IBM CODAIT
center for open source data and ai technologies

state of the art



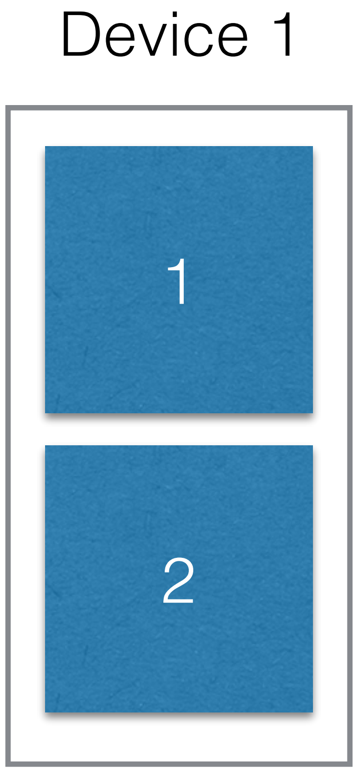
Problem #1

data privacy

Problem #2

competitive advantage / information cartels

homomorphic encryption



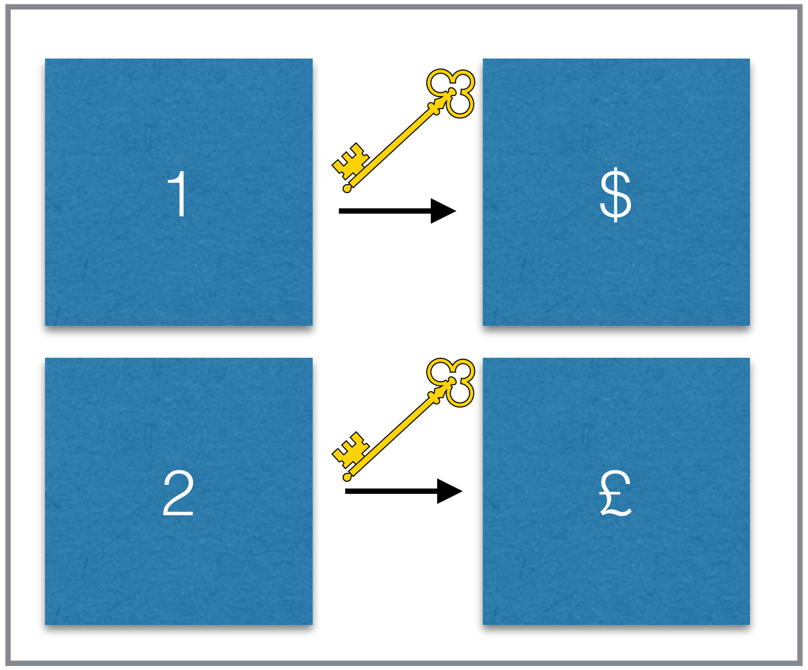
Device 1

1

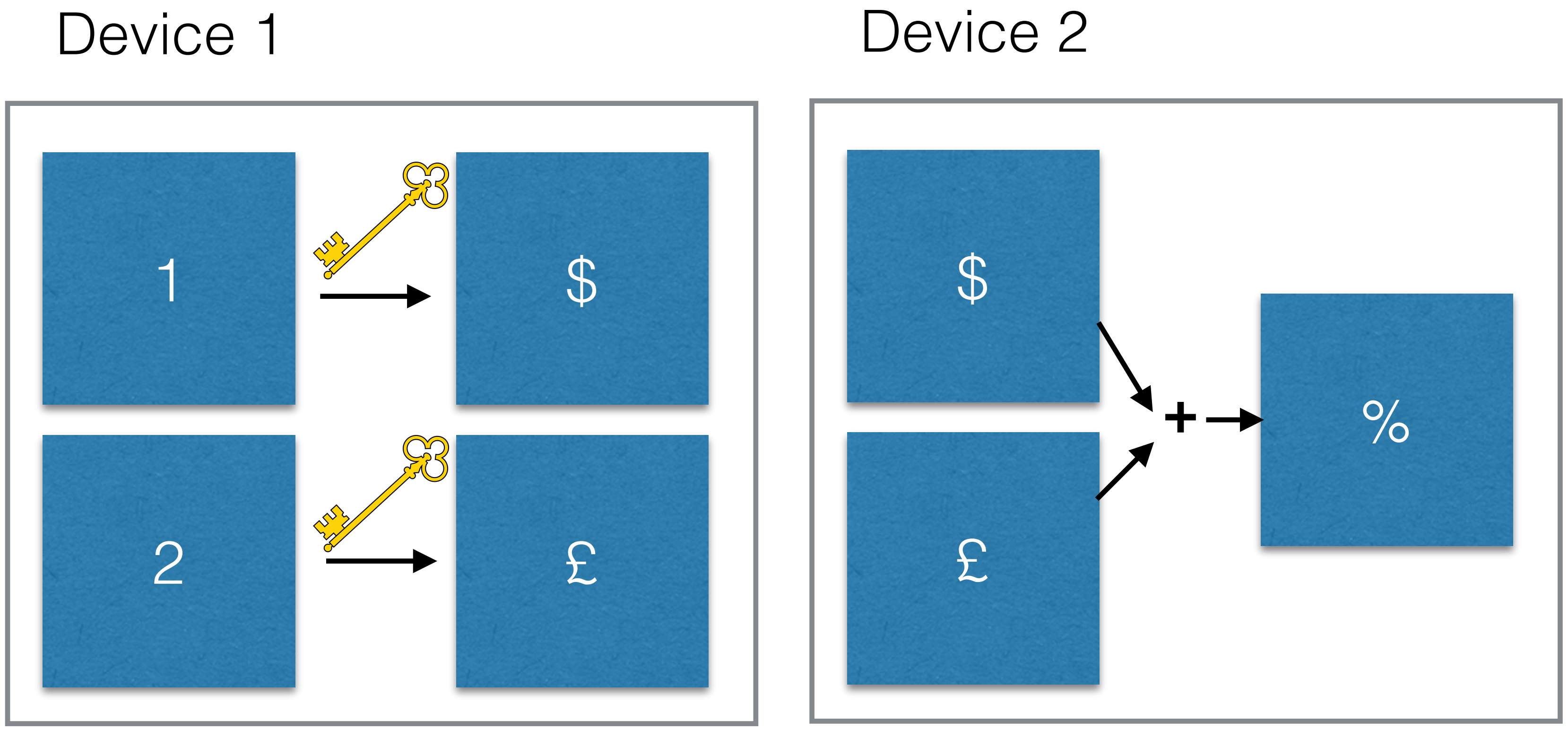
2

homomorphic encryption

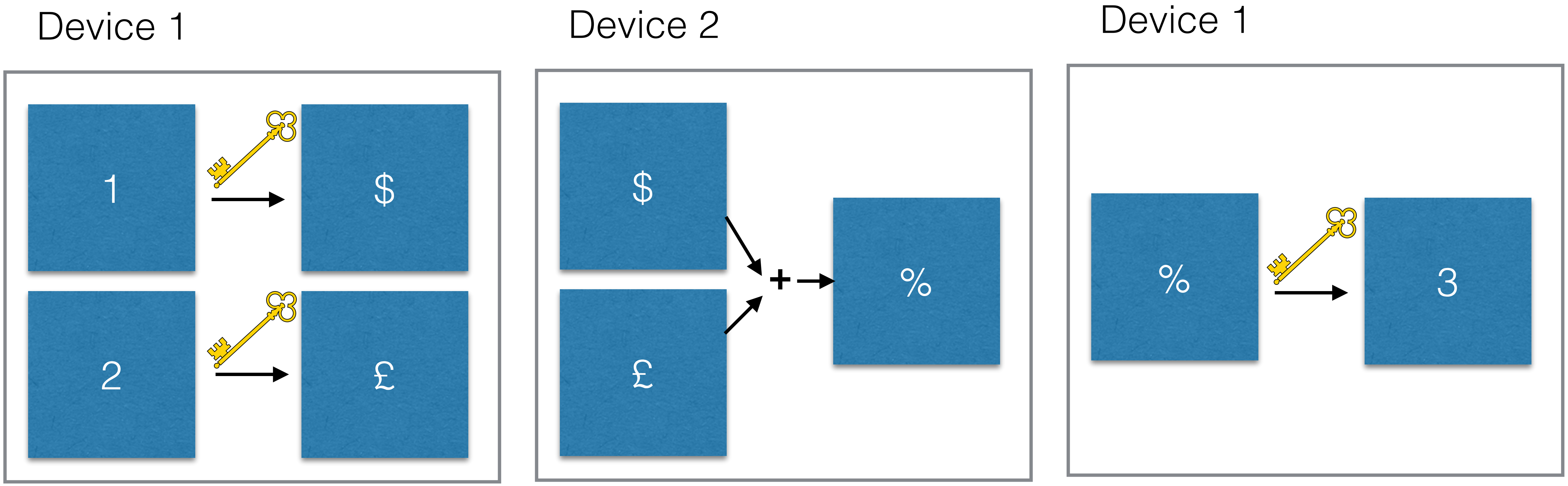
Device 1



homomorphic encryption



homomorphic encryption

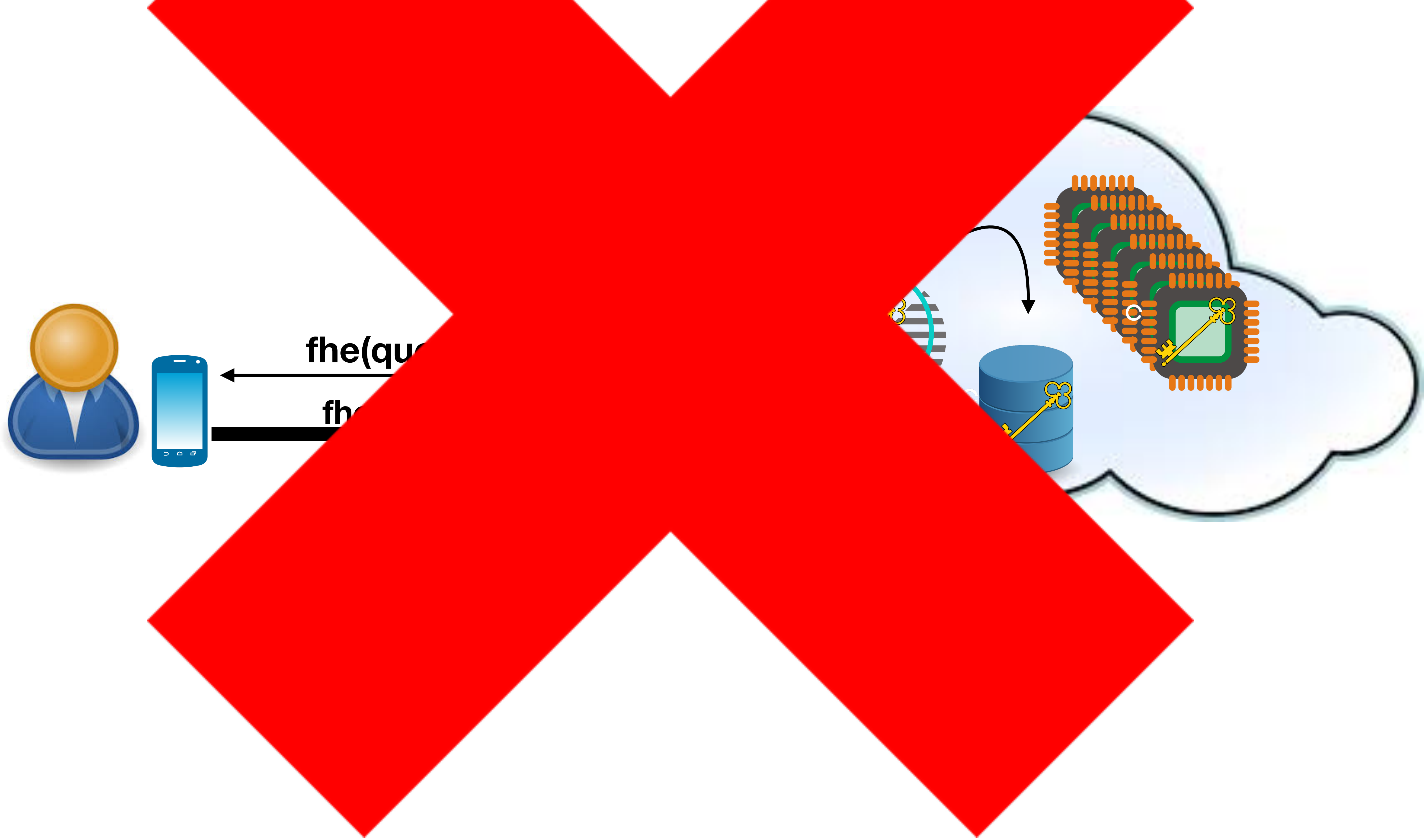


homo rphic encryption

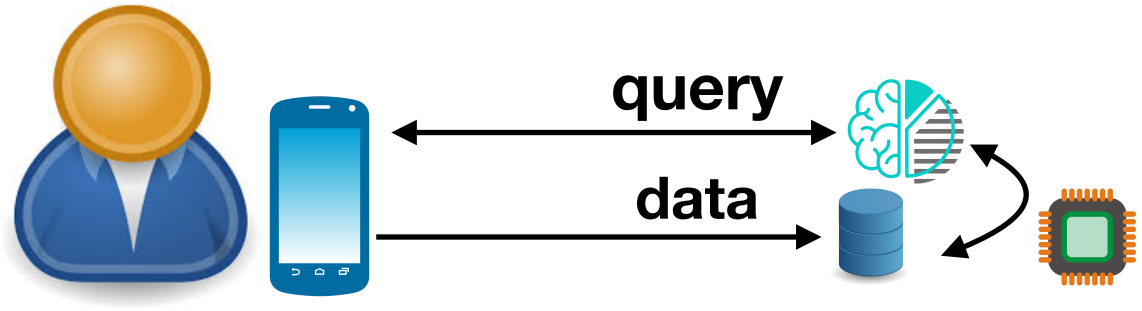
At its present state, this library is mostly meant for researchers working on HE and its uses.

https://gith omenc/HElib

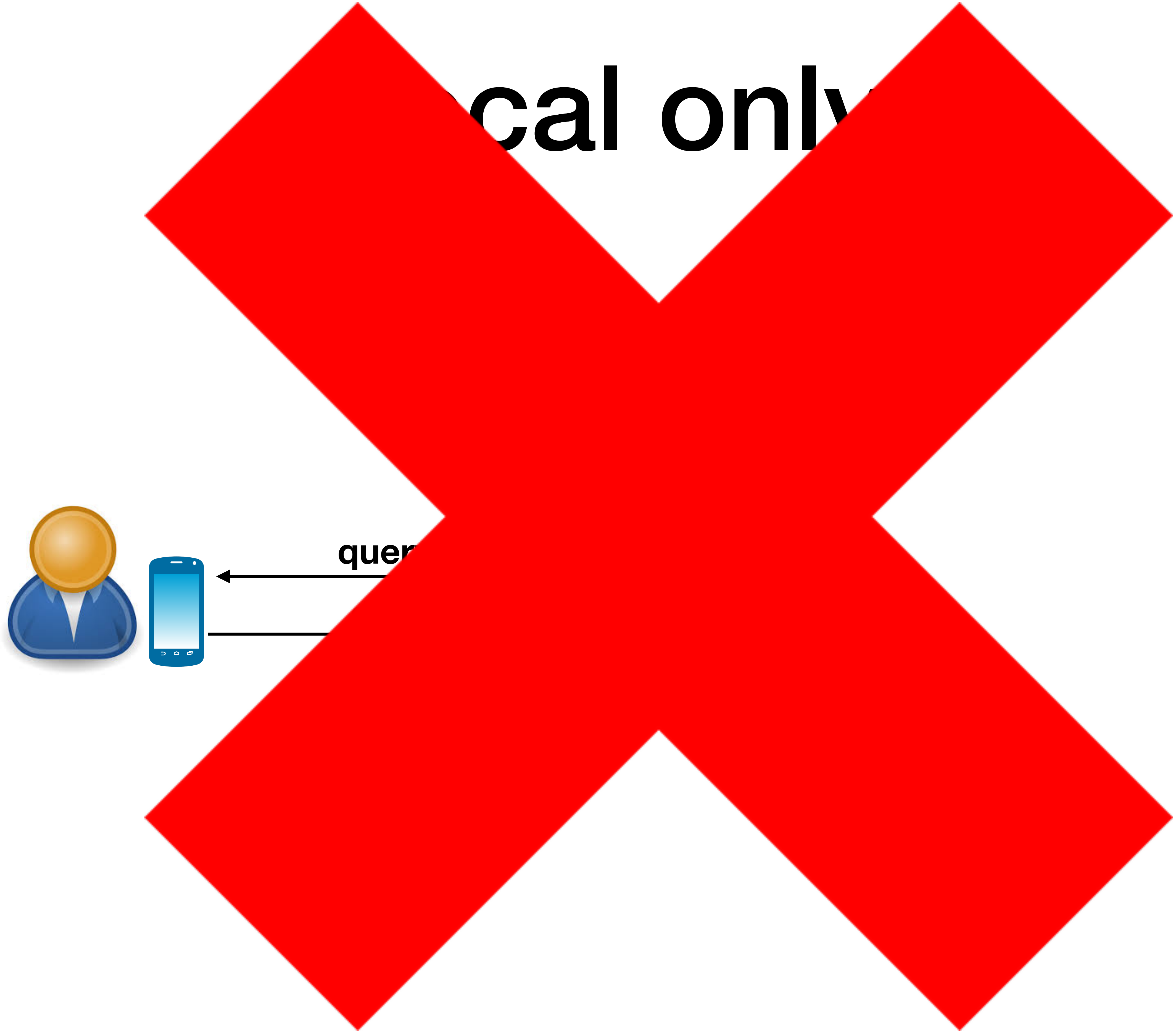
hom rphic en ption



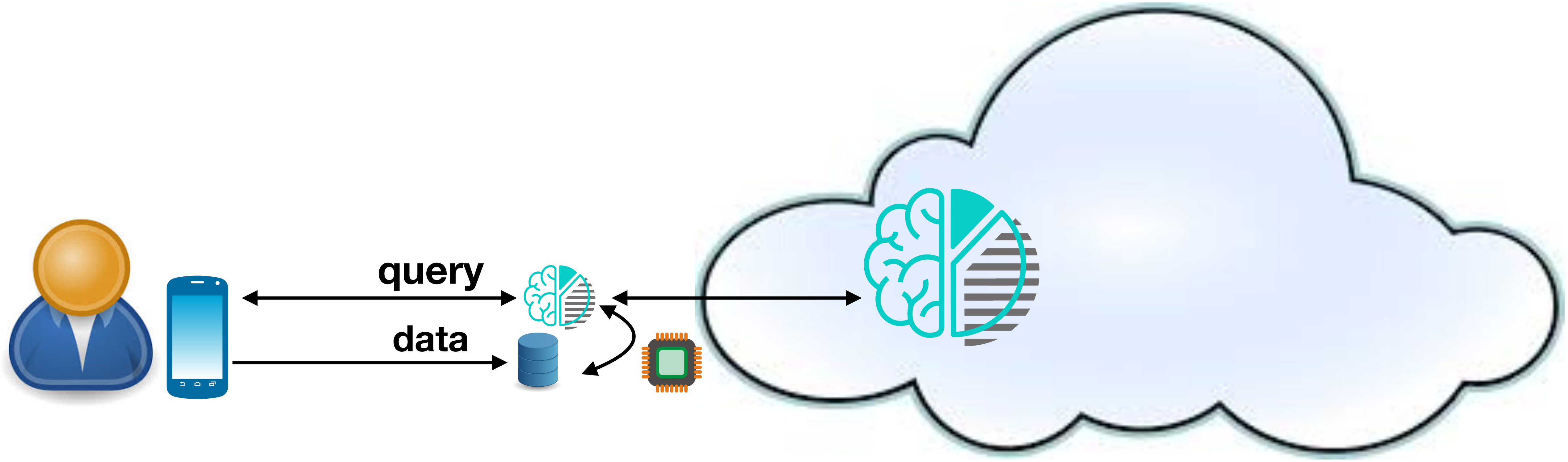
local only?

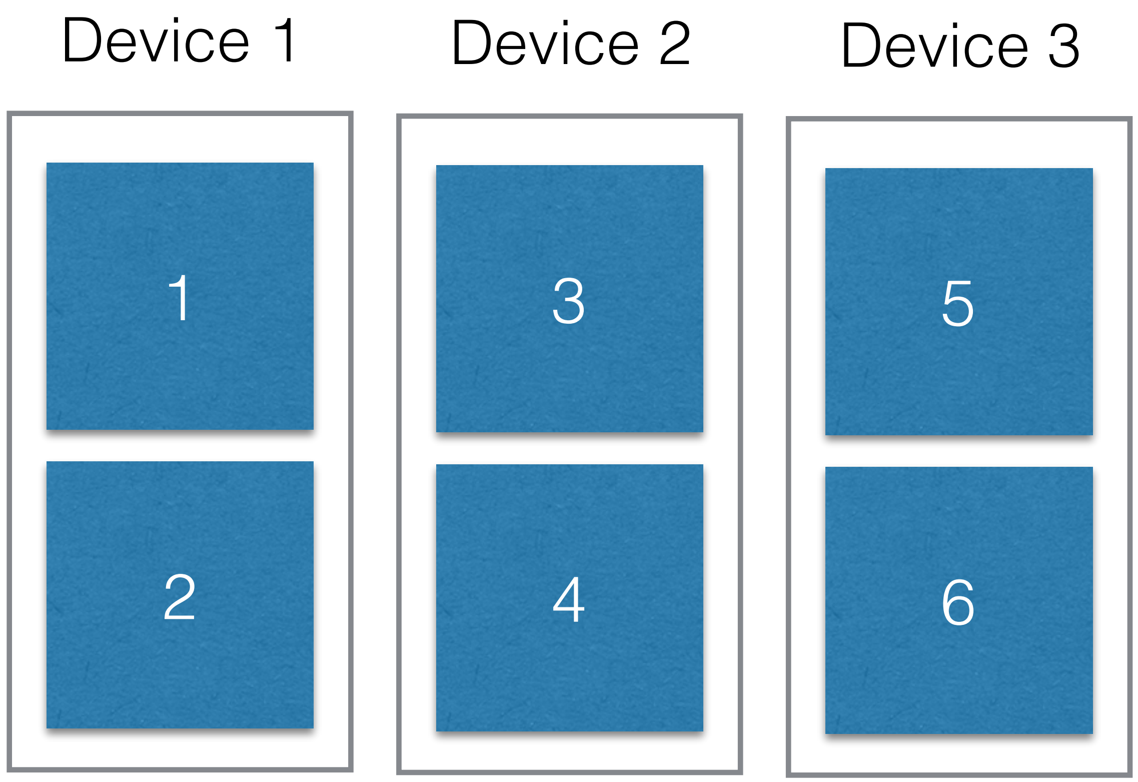


cal only



federated learning





Device 1

Device 2

Device 3

1

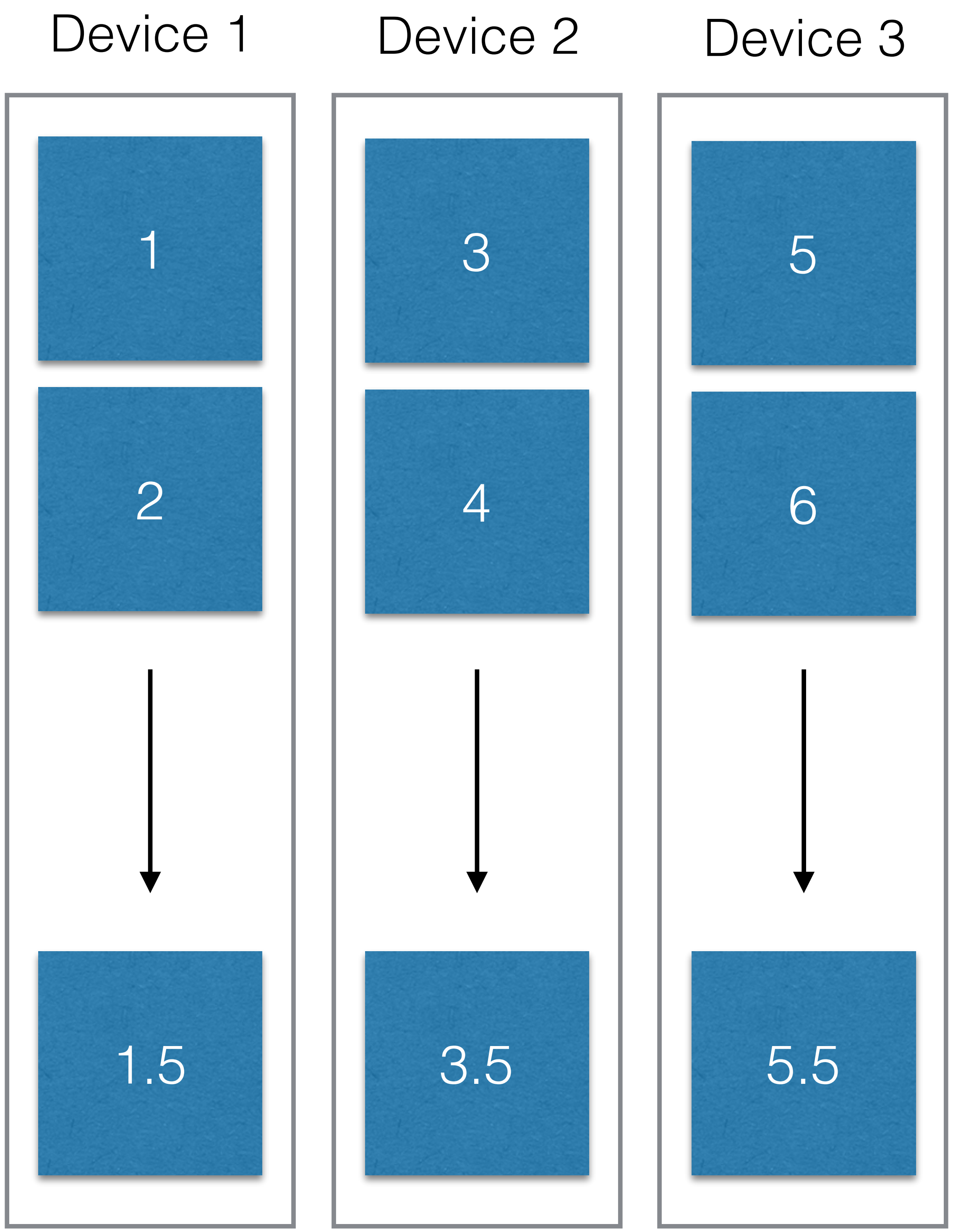
2

3

4

5

6



Device 1

1

2

1.5

Device 2

3

4

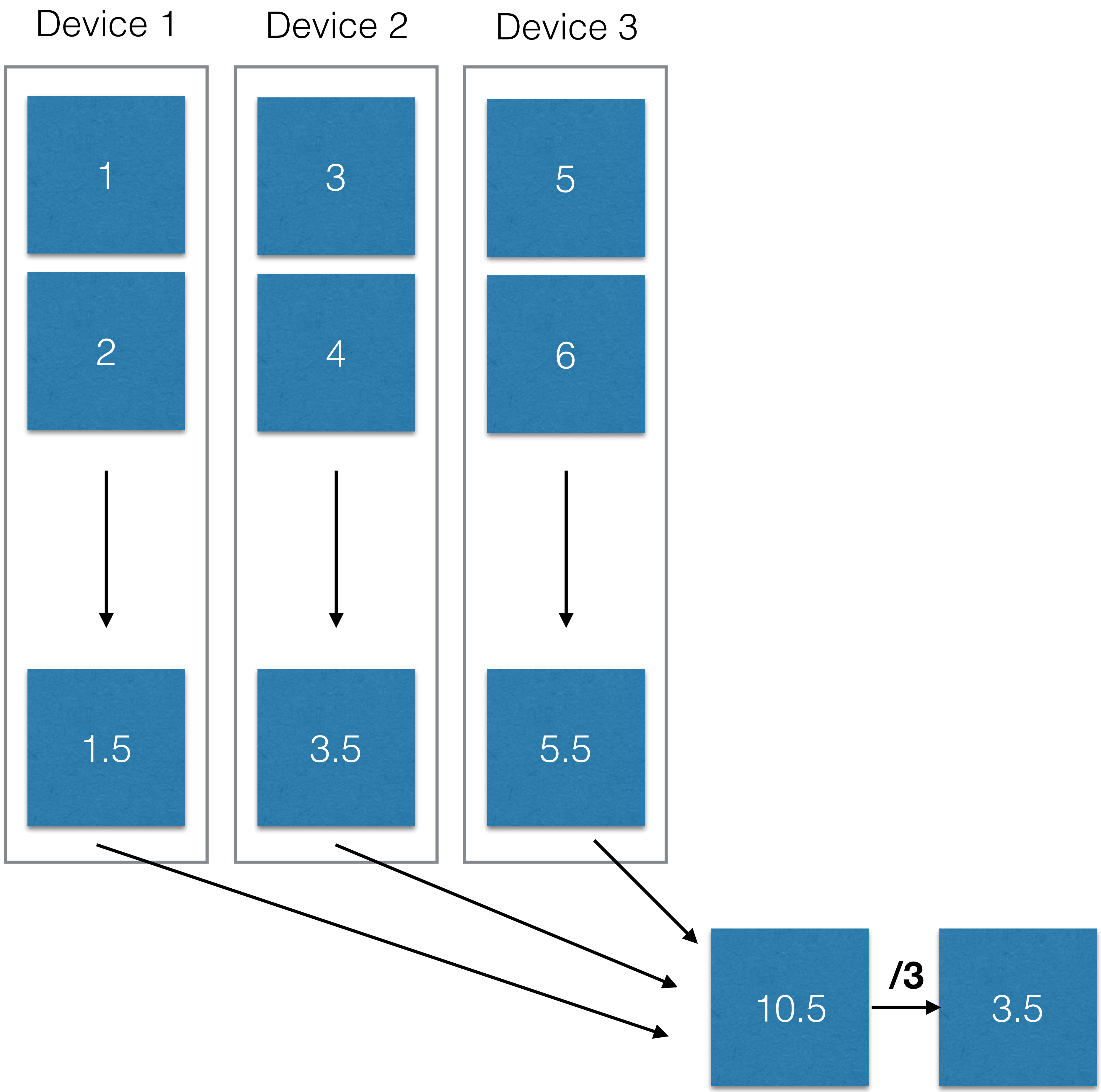
3.5

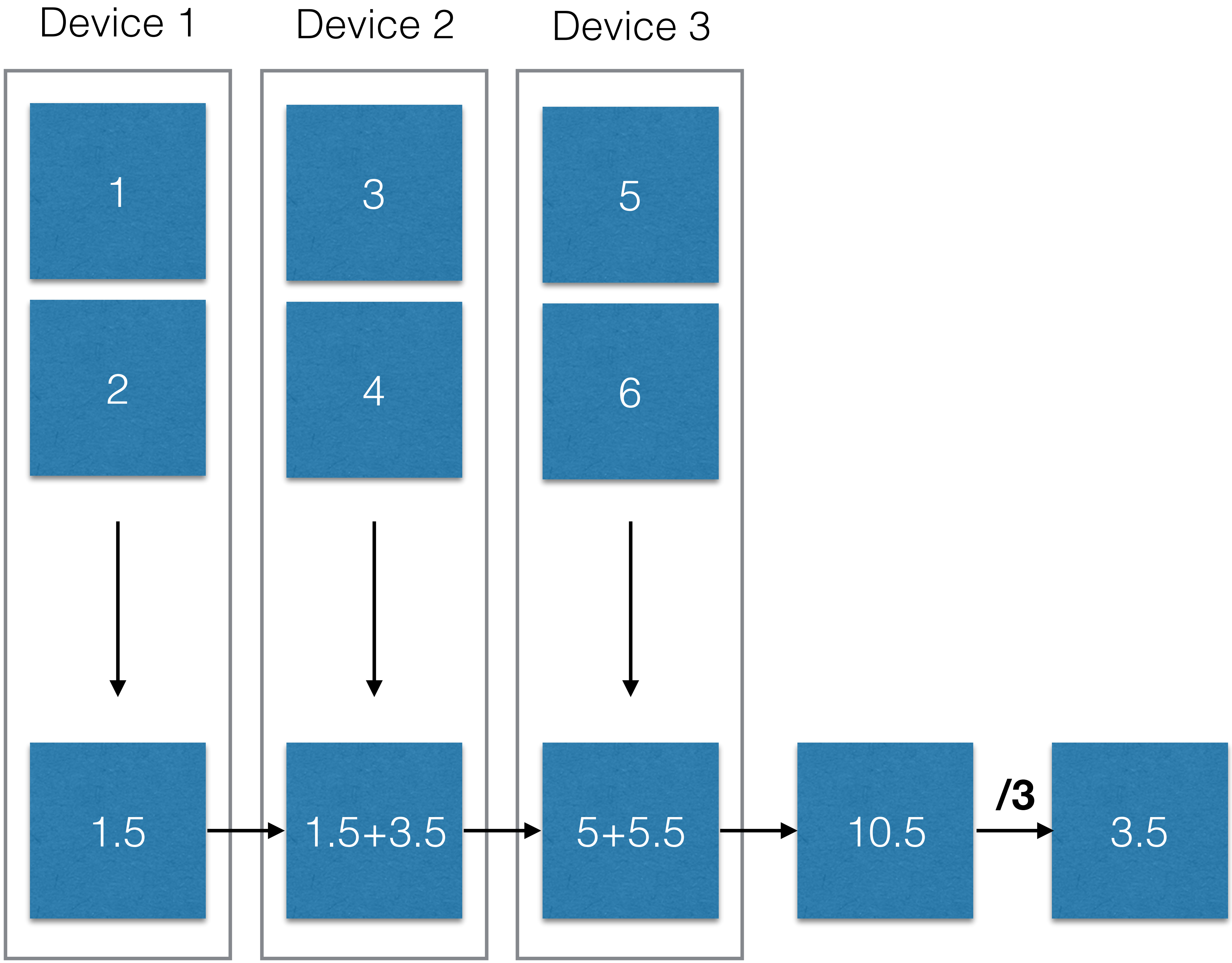
Device 3

5

6

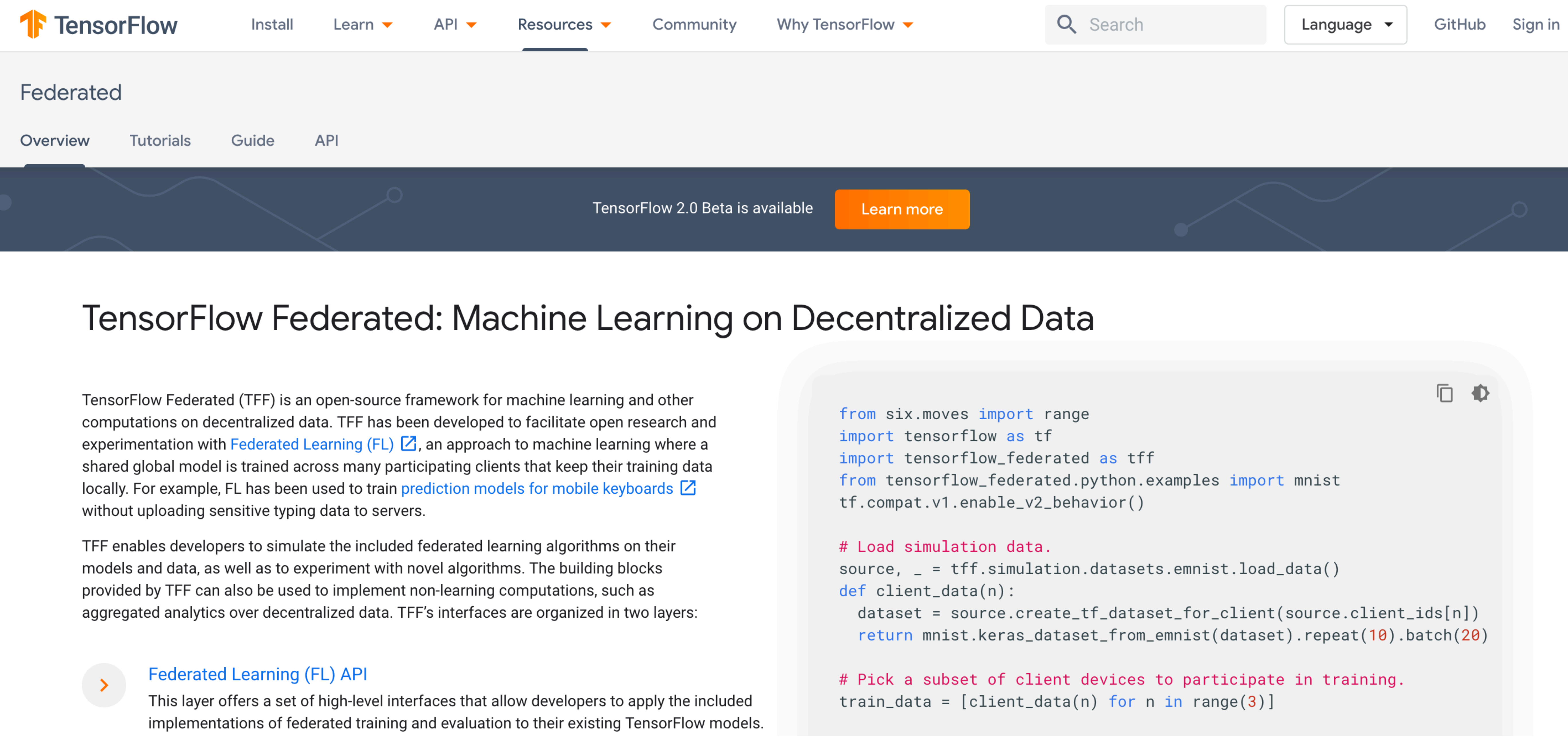
5.5


Device 1
Device 2
Device 3
1
3
5
2
4
6
1.5
3.5
5.5
10.5
/3
3.5


Device 1
Device 2
Device 3
1
3
5
2
4
6
1.5
1.5+3.5
5+5.5
10.5
/3
3.5

Romeo, please show them the Neural Network Parallel Training stuff….

nnparallelization2019 - slide 21



TensorFlow Federated: Machine Learning on Decentralized Data

TensorFlow Federated (TFF) is an open-source framework for machine learning and other computations on decentralized data. TFF has been developed to facilitate open research and experimentation with Federated Learning (FL), an approach to machine learning where a shared global model is trained across many participating clients that keep their training data locally. For example, FL has been used to train prediction models for mobile keyboards without uploading sensitive typing data to servers.

TFF enables developers to simulate the included federated learning algorithms on their models and data, as well as to experiment with novel algorithms. The building blocks provided by TFF can also be used to implement non-learning computations, such as aggregated analytics over decentralized data. TFF's interfaces are organized in two layers:

Federated Learning (FL) API

This layer offers a set of high-level interfaces that allow developers to apply the included implementations of federated training and evaluation to their existing TensorFlow models.

```
from six.moves import range
import tensorflow as tf
import tensorflow_federated as tff
from tensorflow_federated.python.examples import mnist
tf.compat.v1.enable_v2_behavior()

# Load simulation data.
source, _ = tff.simulation.datasets.emnist.load_data()
def client_data(n):
  dataset = source.create_tf_dataset_for_client(source.client_ids[n])
  return mnist.keras_dataset_from_emnist(dataset).repeat(10).batch(20)

# Pick a subset of client devices to participate in training.
train_data = [client_data(n) for n in range(3)]
```

Problem #1

data privacy

Problem #1

da[illegible]acy

Problem #2

competitive advantage / information cartels

F [illegible] 2

competitiv [illegible] tion cartels

Romeo, please show them the Common Voice stuff….

ai_strategy - slide 101

Romeo, please show them the Facebook stuff….

…ask me later…

(Lightning Talk)