Privacy Preserving ML

Federated Learning

Differential Privacy

Homomorphic Encryption

Trusted Exection Environments

Romeo Kienzler

IBM CODAIT

IBM Center for Open Source Data and AI Technologies

state of the art

Problem #1

data privacy

Problem #2

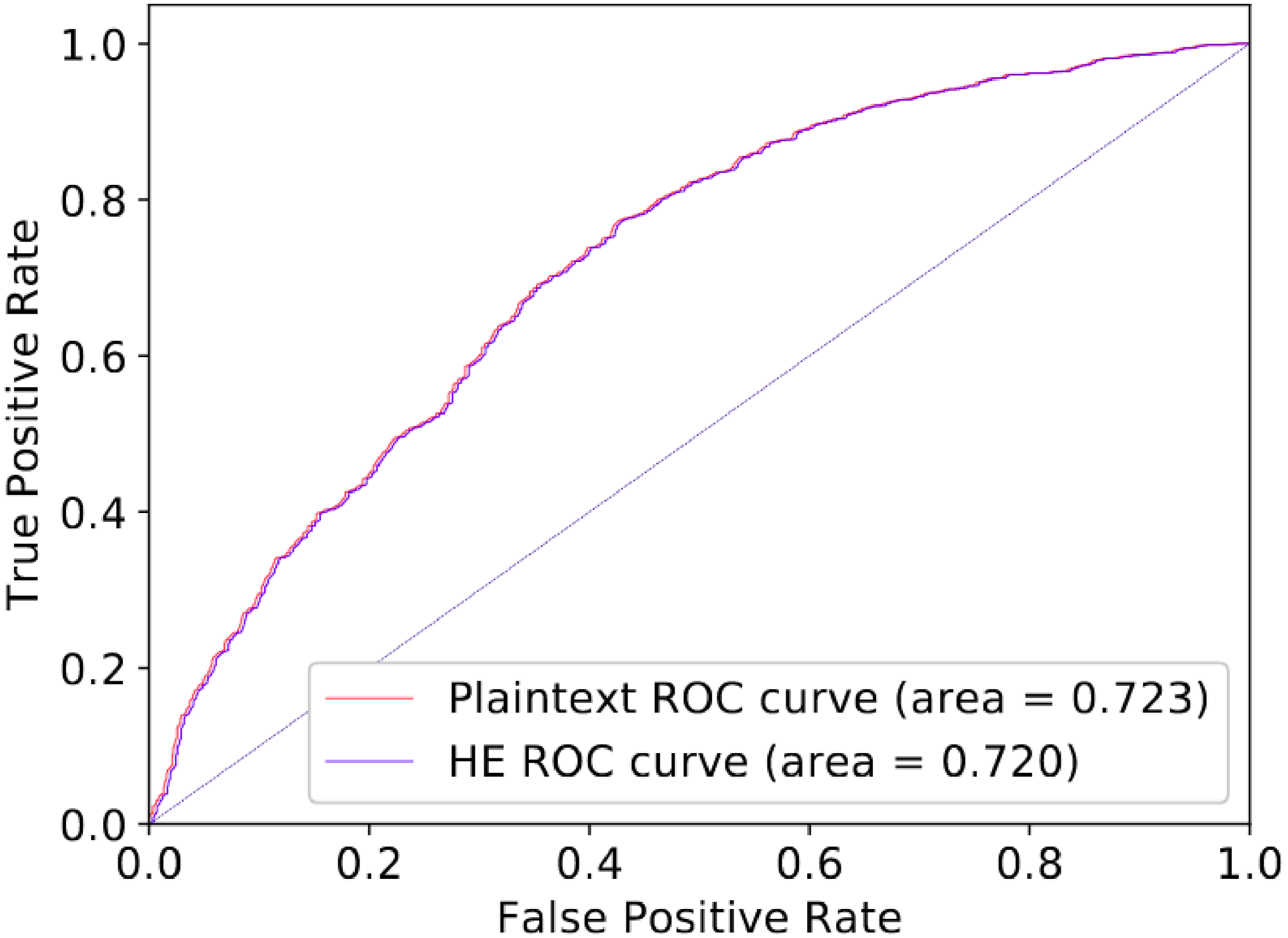
competitive advantage / information cartels

homomorphic encryption

Towards a Homomorphic Machine Learning Big Data Pipeline for the Financial Services Sector

Oliver Masters[1], Hamish Hunt[1], Enrico Steffinlongo[1], Jack Crawford[1], Flavio Bergamaschi[1], Maria Eugenia Dela Rosa[2], Caio Cesar Quini[2], Camila T. Alves[2], Fernanda de Souza[2], and Deise Goncalves Ferreira[2]

[1] IBM Research, Hursley, UK
{oliver.masters,enrico.steffinlongo,jack.crawford}@ibm.com
{hamishhun,flavio}@uk.ibm.com
[2] Banco Bradesco SA, Osasco, SP, Brasil
{maria.e.delarosa,caio.quini,camila.t.alves,fernanda.souza,deise.g.ferreira}@bradesco.com.br


True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Plaintext ROC curve (area = 0.723)
HE ROC curve (area = 0.720)

https://github.com/homenc/HElib

Lab 1: Homomorphic Encrytion

https://github.com/romeokienzler/ppml

he_1.ipynb

labs.cognitiveclass.ai

Labs

IBM Skills Network Labs are a place for you to practice the data science, machine learning, and AI skills you're learning in your online courses. You have access to JupyterLab, Zeppelin, and RStudio preinstalled with Apache Spark and the necessary packages to learn new skills in Python, R, and Scala.

Log in with social

Click button to log in.

Cognitive Class

What do you want **to do** today?

Manage Data

My Data

Build Analytics

JupyterLab

Coming soon

RStudio IDE

Coming soon

Prepare Data

Coming soon

Develop Applications

Theia — Cloud IDE

Theia — Cloud IDE (With Docker)

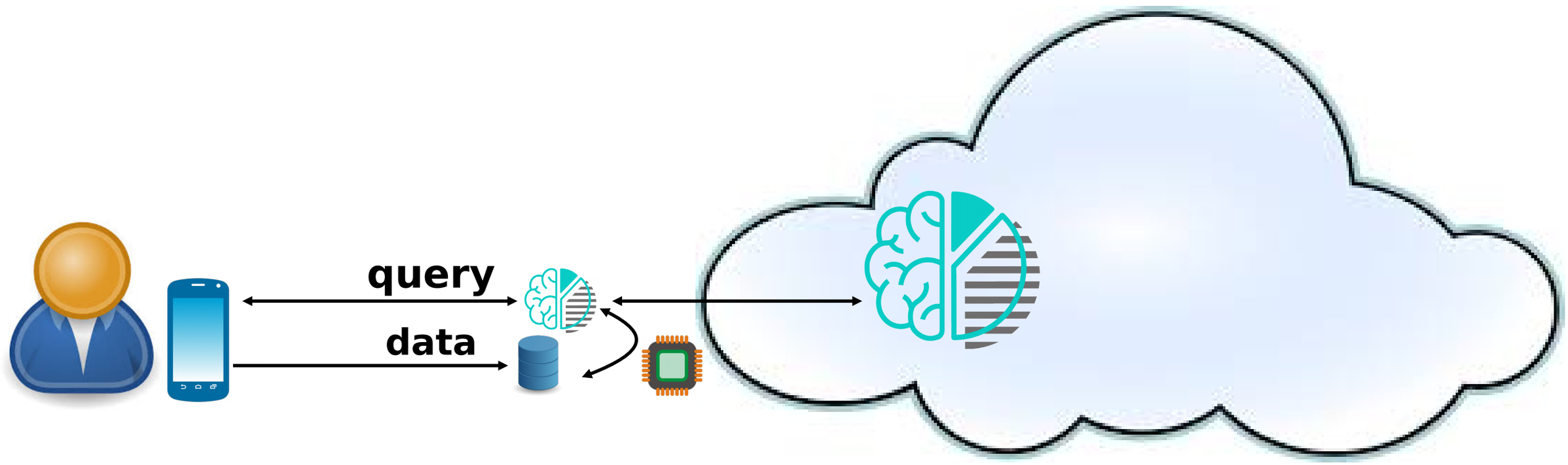
Theia — Cloud IDE (With OpenShift)

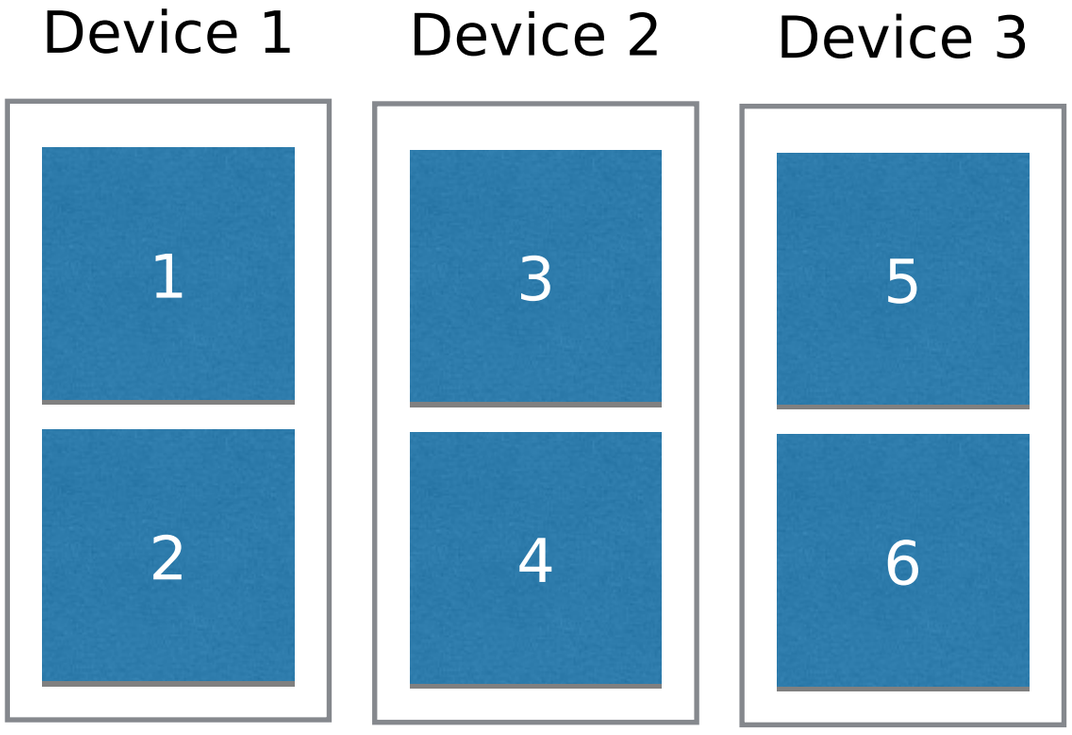
Build Analytics on GPU

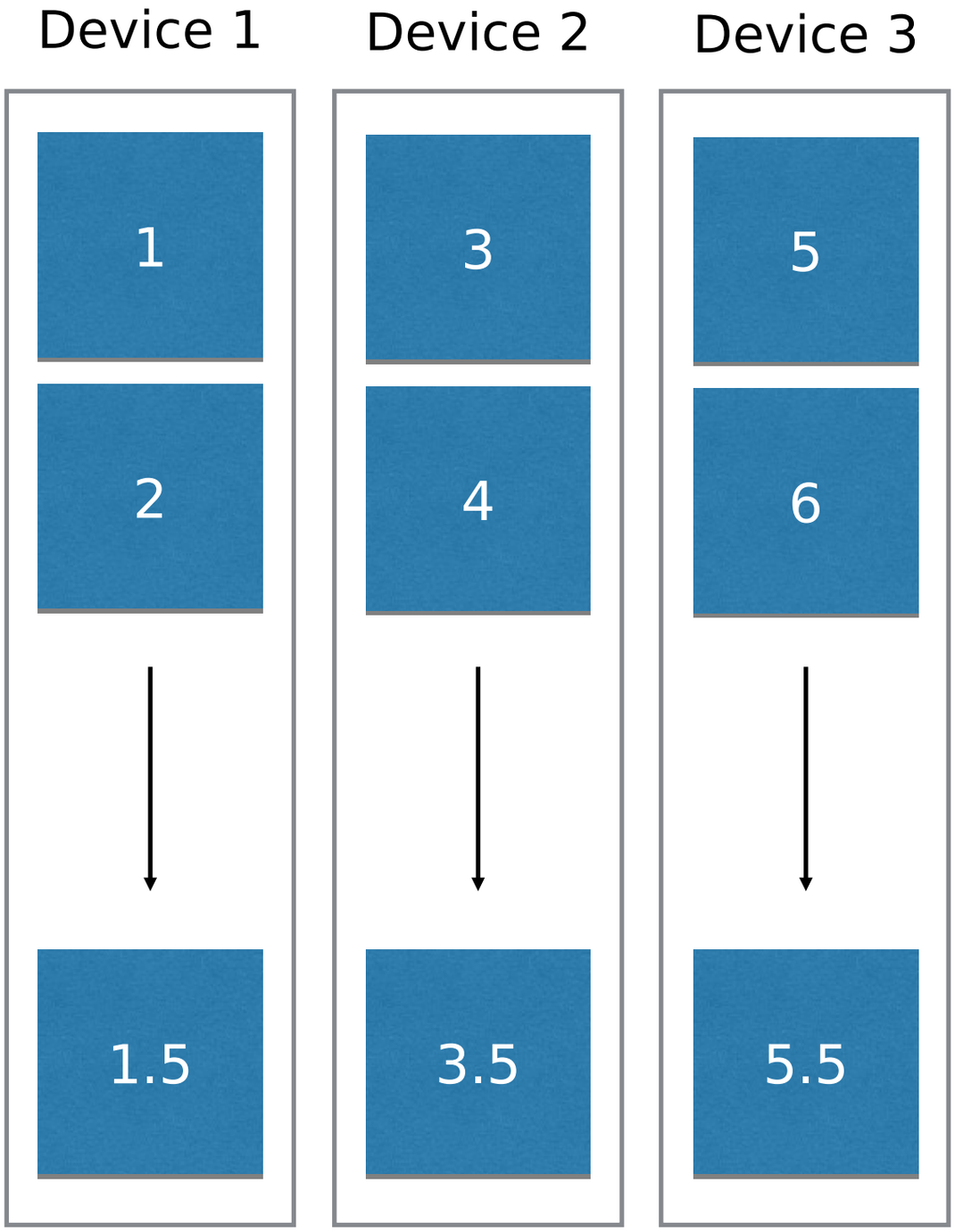
Coming soon

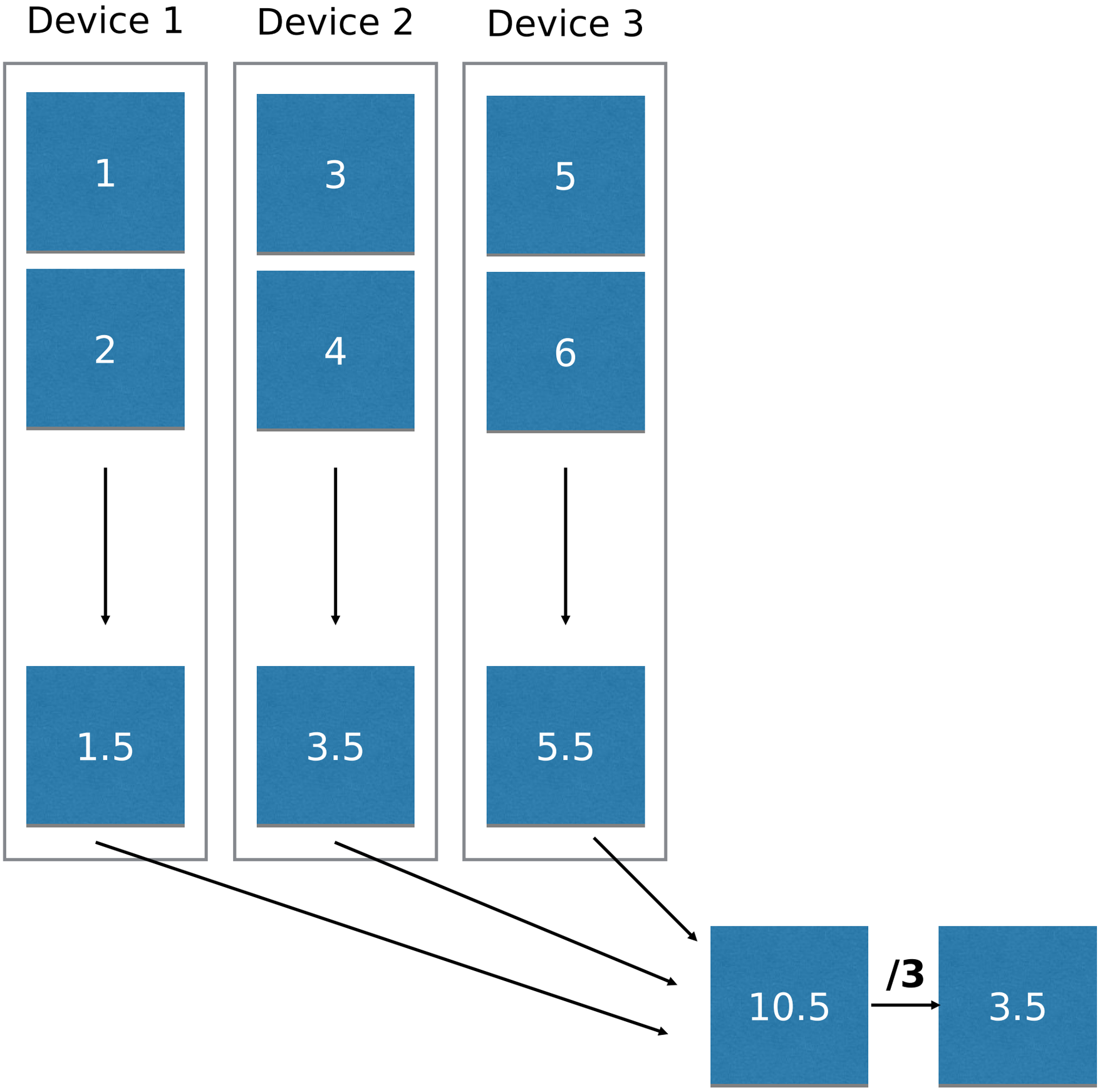
Coming soon

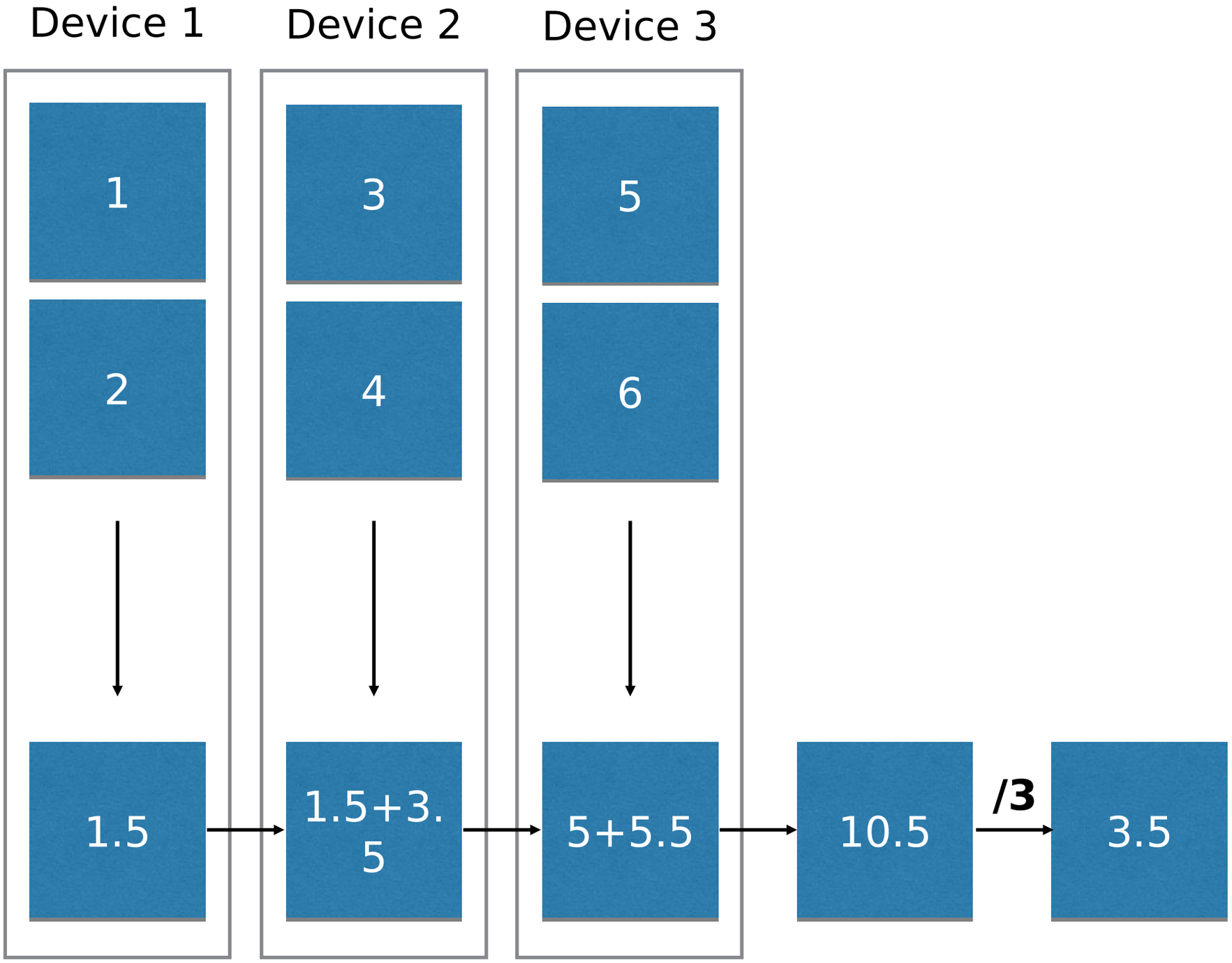
federated learning




Device 1
Device 2
Device 3
1
2
3
4
5
6

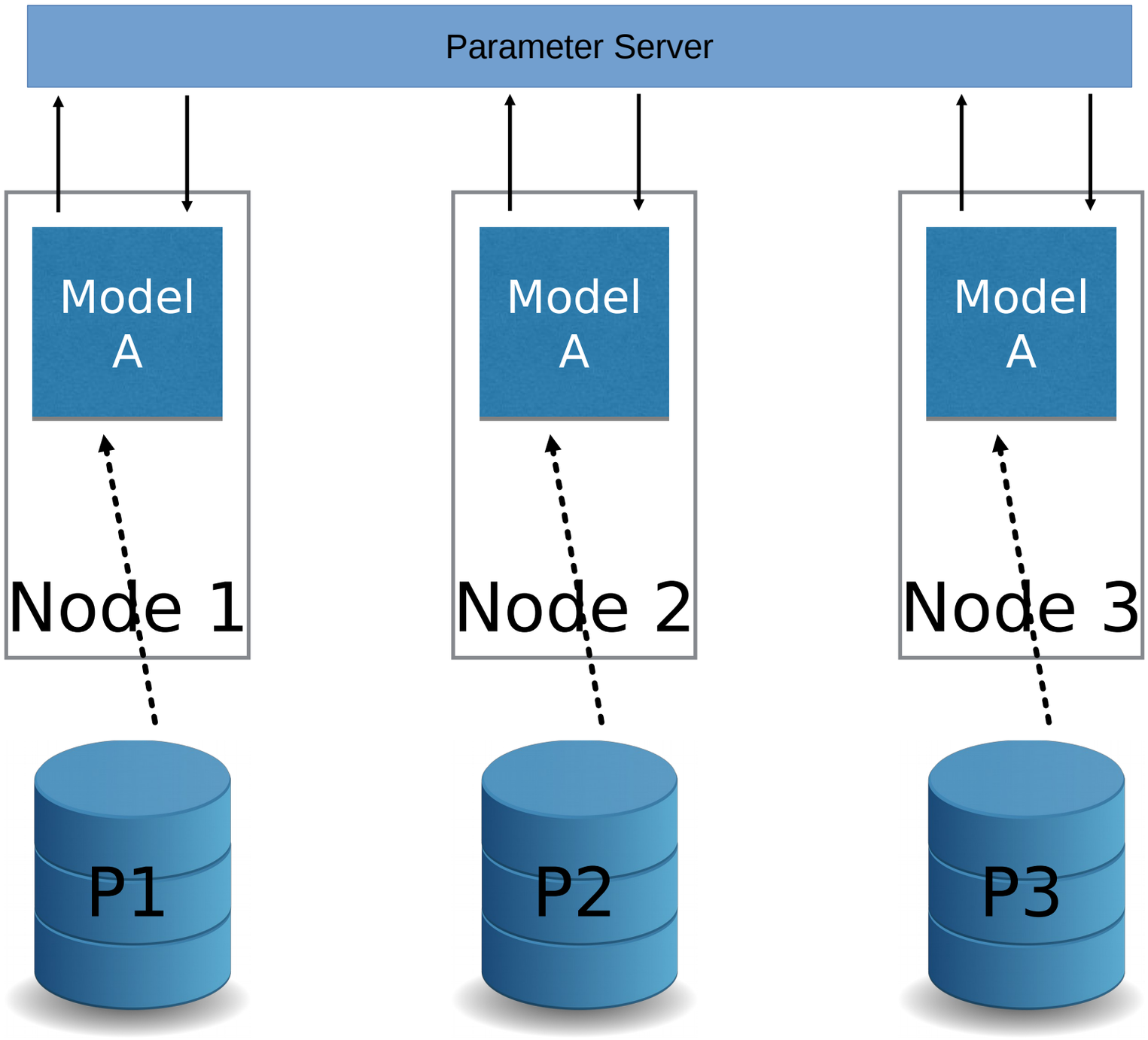

Device 1
Device 2
Device 3
1
2
3
4
5
6
1.5
3.5
5.5


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


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

data parallelism

aka. “Jeff Dean style” parameter averaging



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)]
```

Lab 1: Homomorphic Encrytion

https://github.com/romeokienzler/ppml

fl_1.ipynb

differential privacy

Federated Learning Idea:
Share aggregates only

differential privacy

Problem:
https://en.wikipedia.org/wiki/Reconstruction_attack

differential privacy

Example:

Sales by County + Sales by LOB + Total Sales

May reveal sales of entities which are alone in a LOB/county combination

differential privacy

Conclusion:

No privacy without noise

ε-differential privacy

Summary:

Calibrating noise to sensitivity in private data analysis.

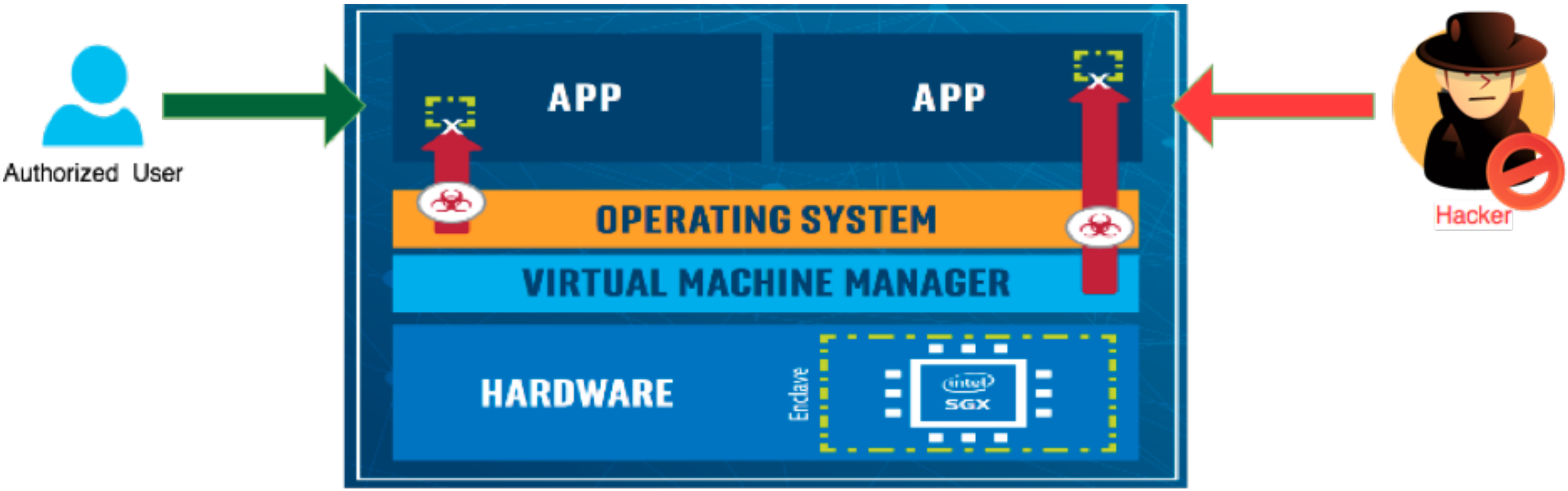
ε-differential privacy

In other words:

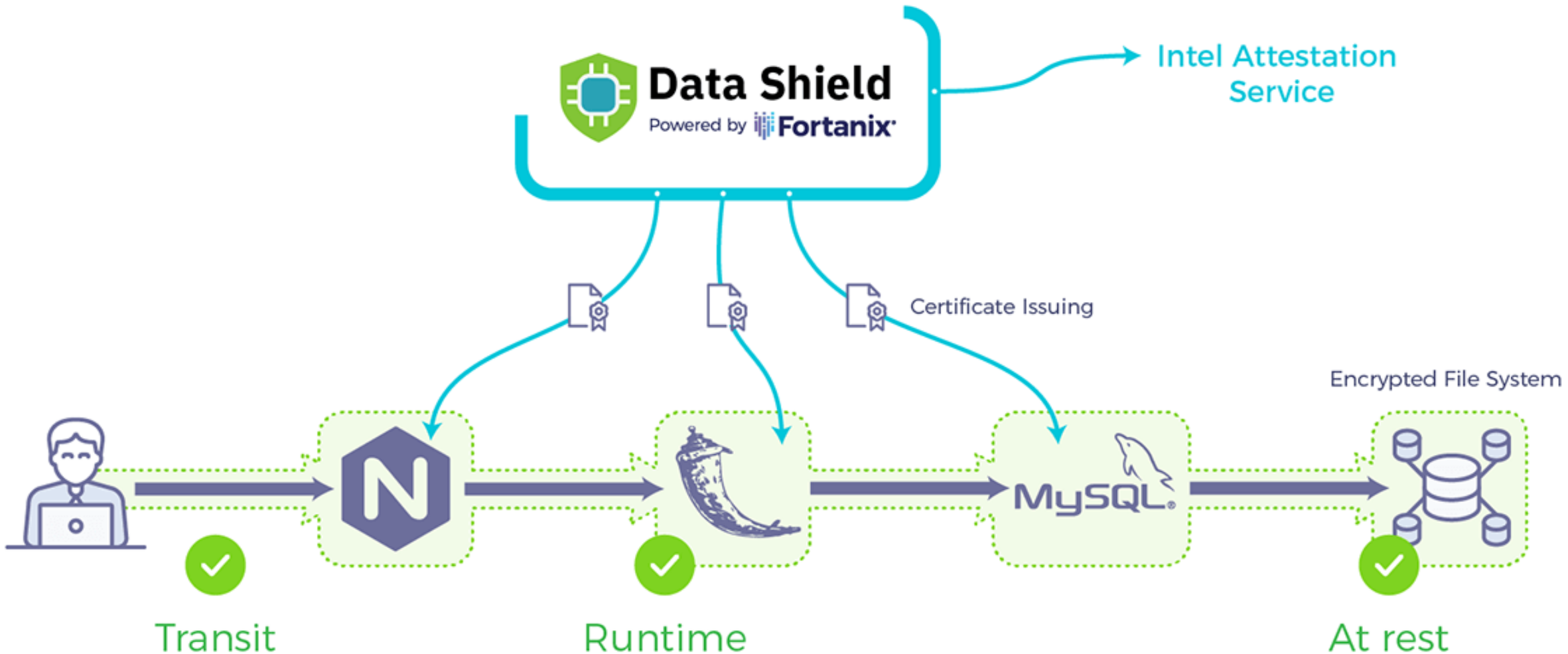
The more individuals are involved in an aggregate, the less noise needs to be added

Intel SGX

(Software Guard Extensions)



IBM Data Shield



Problem #1

data privacy

Proble #1

da acy

Problem #2

competitive advantage / information cartels

Problem #2

competitive advan information cartels

...discussion...

(questions, comments, additions, complaints, suggestions, feedback, ...)