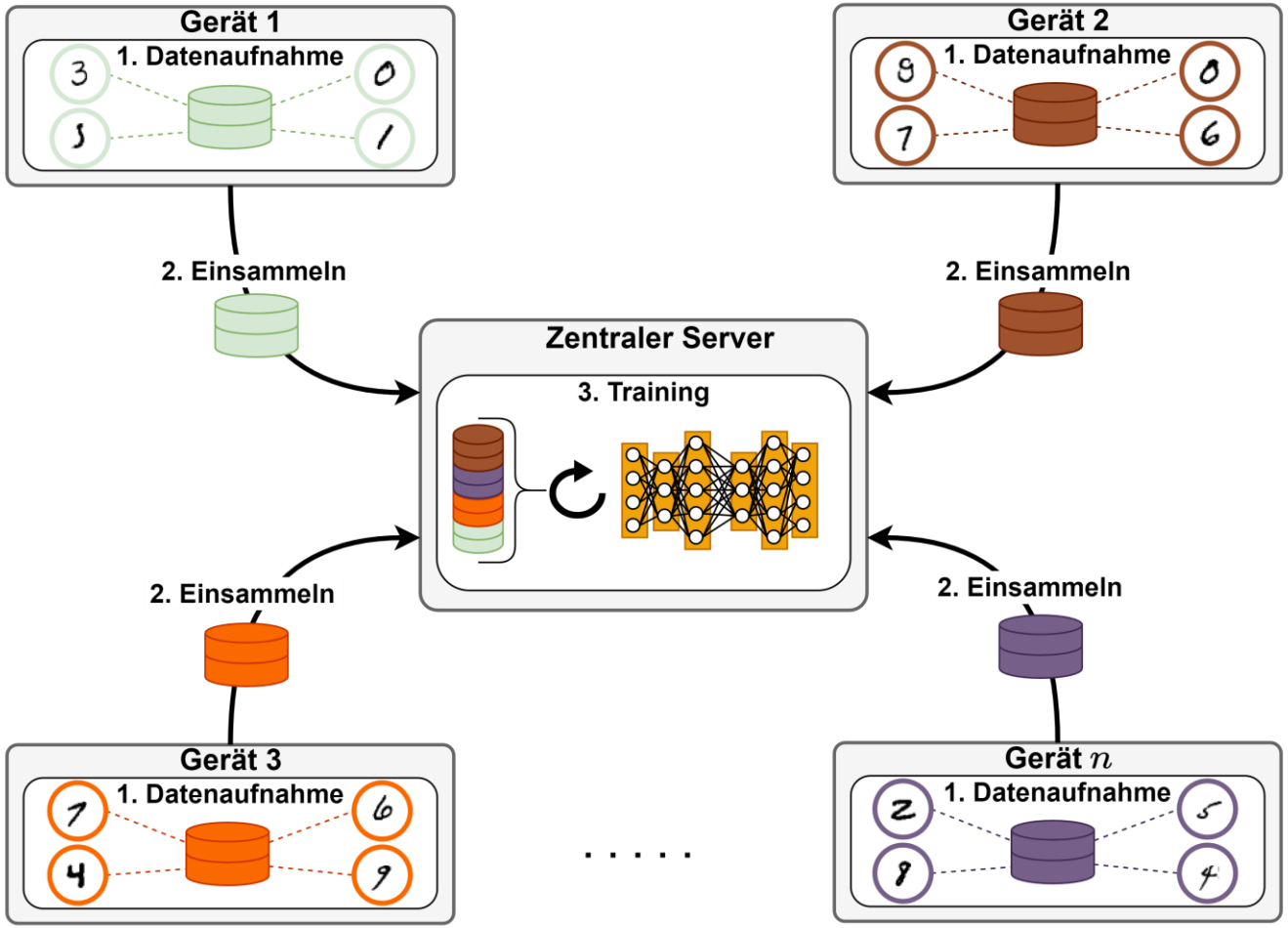
Von Zentral zu Edge: Dezentrales Federated Learning auf Mikrocontrollern

Dr.-Ing. Lars Wulfert

Smart Systems Conference, 19.11.2025, Dortmund

Machine Learning

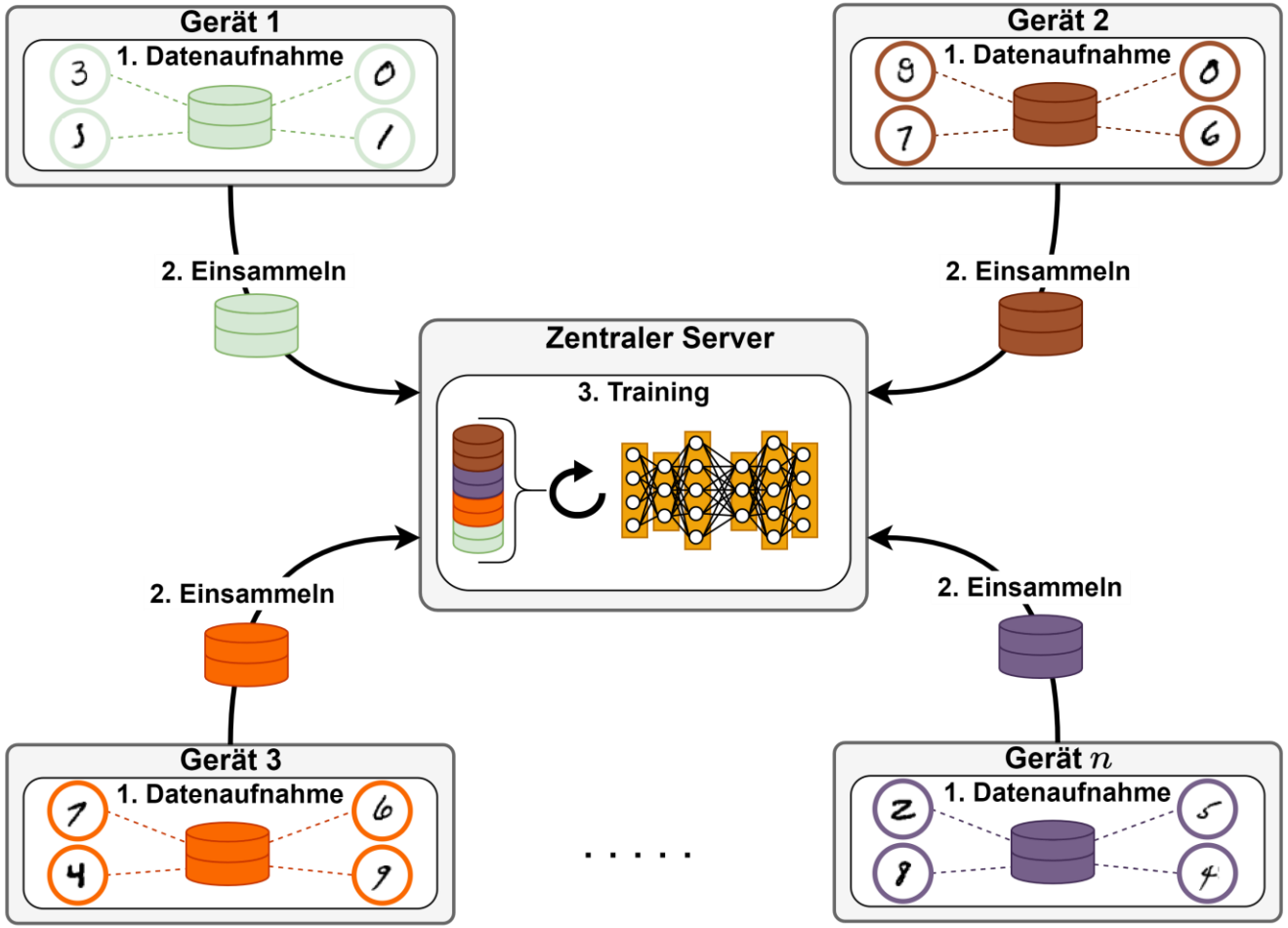


Maschinelles Lernen (ML)

1. Datenaufnahme auf den Geräten
2. Daten vom Server einsammeln
3. Training des Modells im Server
4. Ausführung des Modells auf dem Gerät

ML-Modell | Lokale Daten → Kommunikation | Training

Machine Learning



Maschinelles Lernen (ML)

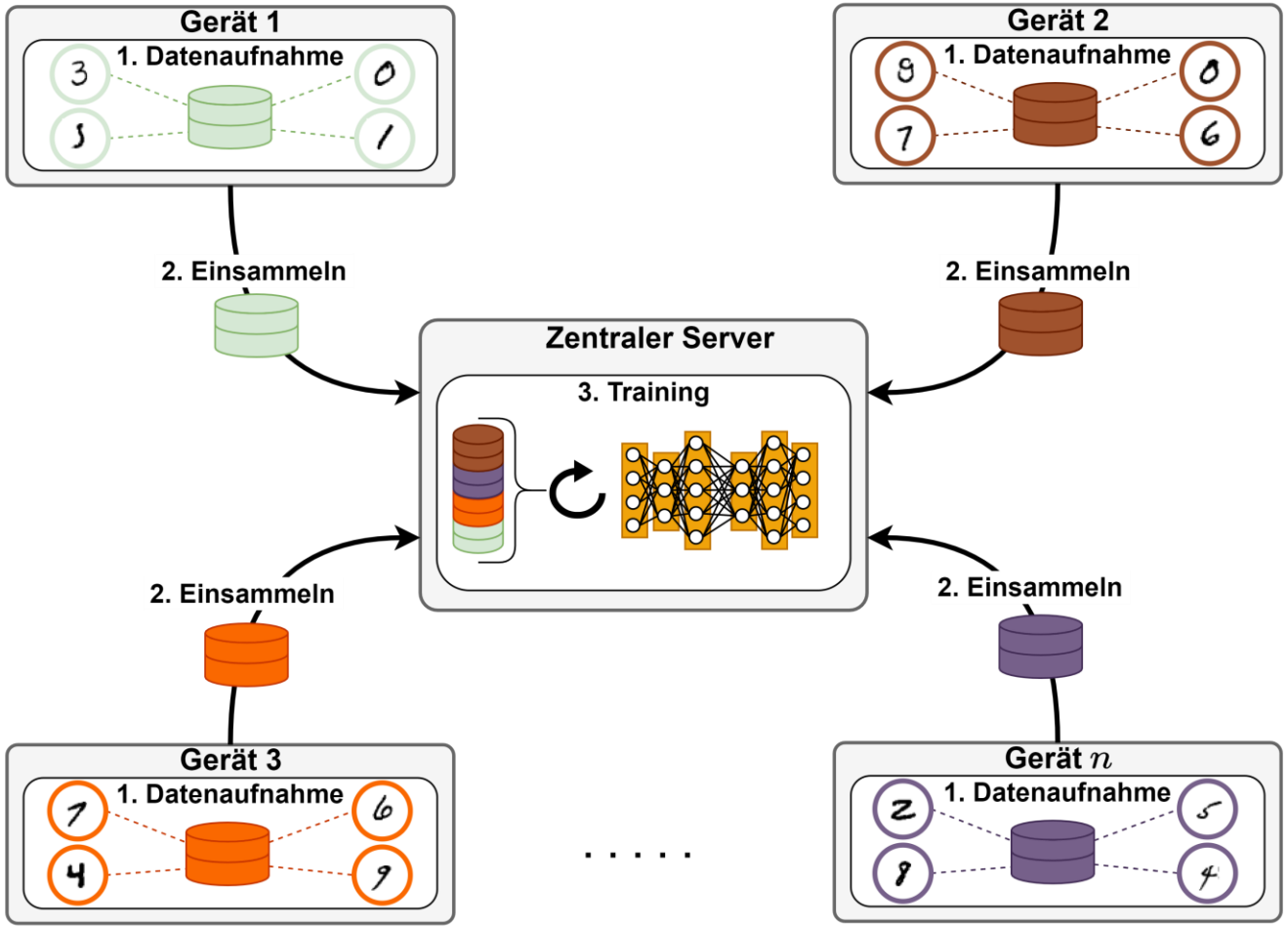
1. Datenaufnahme auf den Geräten
2. Daten vom Server einsammeln
3. Training des Modells im Server
4. Ausführung des Modells auf dem Gerät

Vorteile

- ✓ Verwendung performanter Server
- ✓ Skalierbarkeit
- ✓ Nutzung zentraler Datengrundlage

Fraunhofer IMS

Machine Learning



Maschinelles Lernen (ML)

1. Datenaufnahme auf den Geräten
2. Daten vom Server einsammeln
3. Training des Modells im Server
4. Ausführung des Modells auf dem Gerät

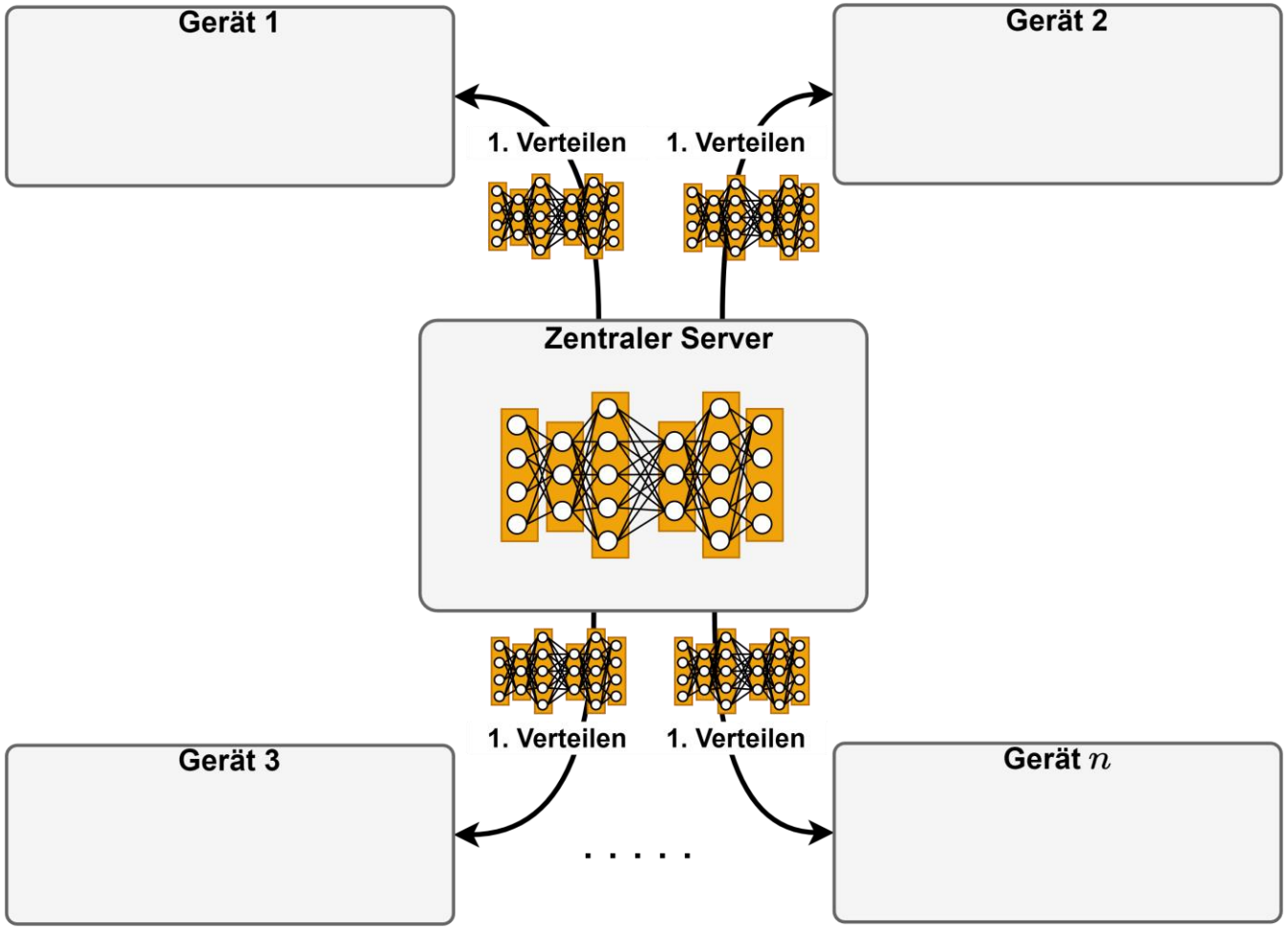
Vorteile

- ✓ Verwendung performanter Server
- ✓ Skalierbarkeit
- ✓ Nutzung zentraler Datengrundlage

Nachteile

- × Netzwerkbelastung
- × DSGVO kann die Verwendung von personenbezogenen Daten einschränken
- × **Datensicherheit & Privatsphäre von sensiblen Daten nicht immer gewährleistet**

Zentrales Federated Learning

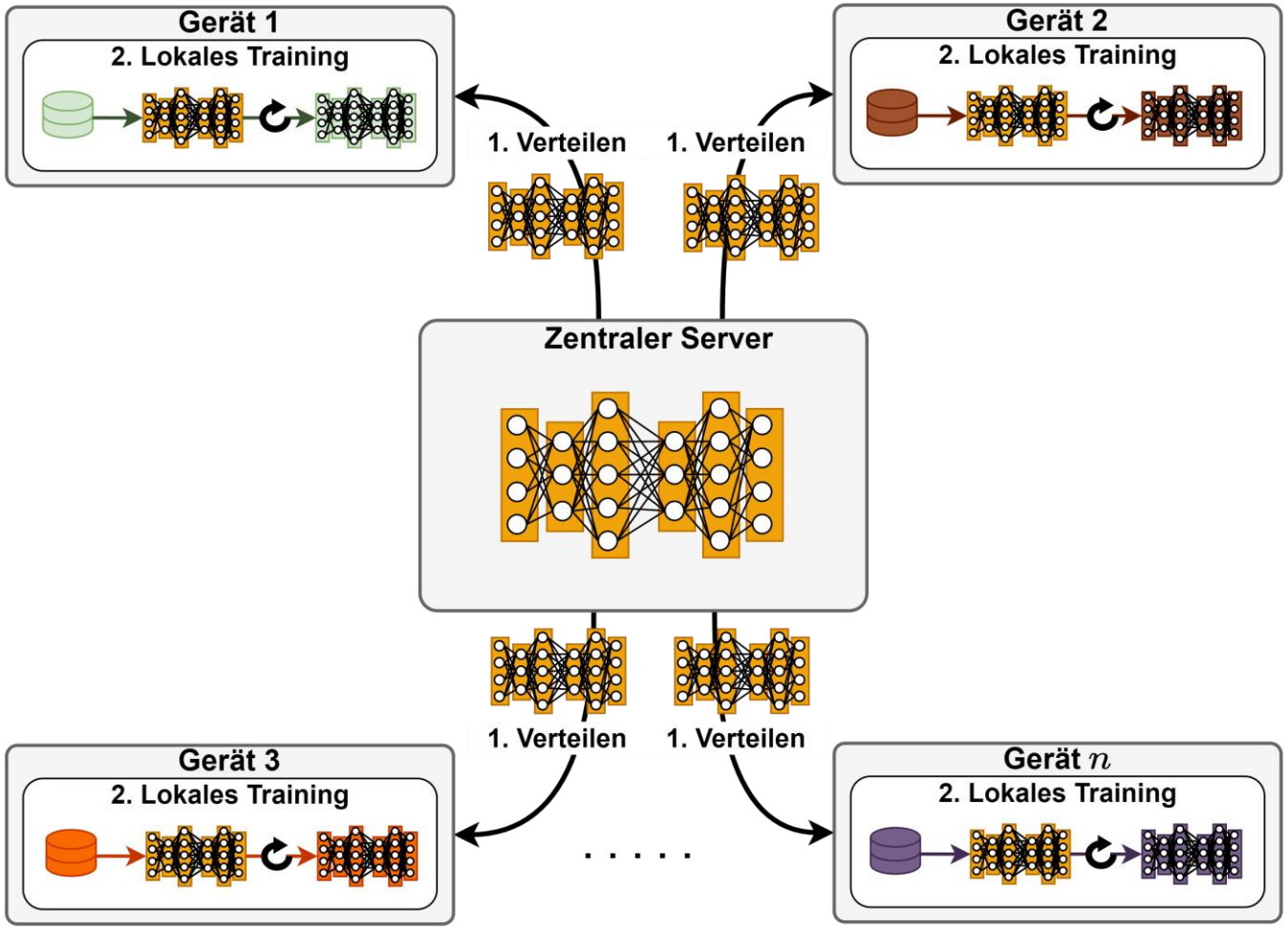


Zentrales föderales Lernen (ZFL)

1. Globales Modell verteilen

Globales Modell | Lokales Modell | Lokale Daten → Kommunikation | Training

Zentrales Federated Learning

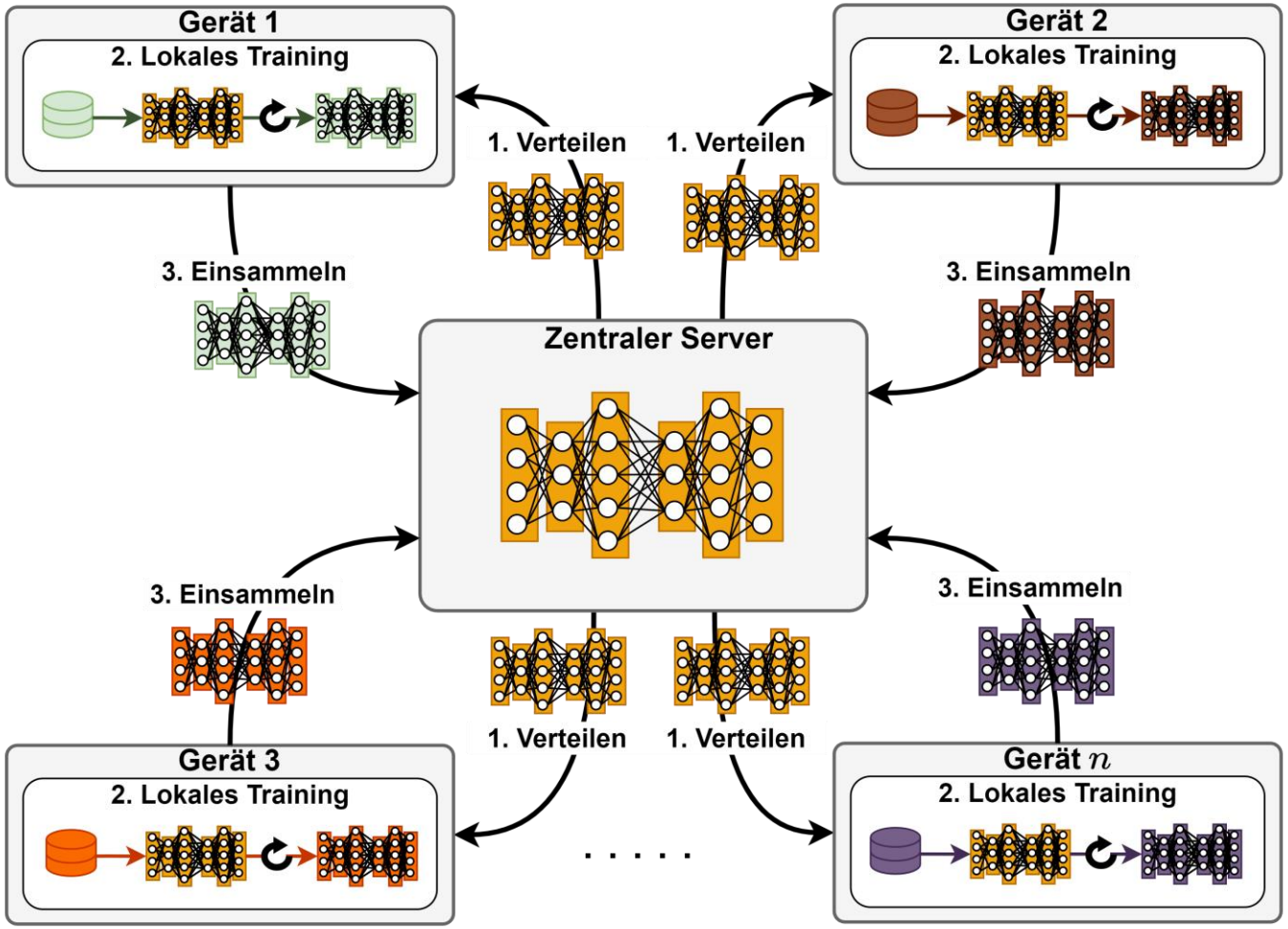


Zentrales föderales Lernen (ZFL) [12]

1. Globales Modell verteilen
2. Lokales Training

Globales Modell | Lokales Modell | Lokale Daten → Kommunikation | Training

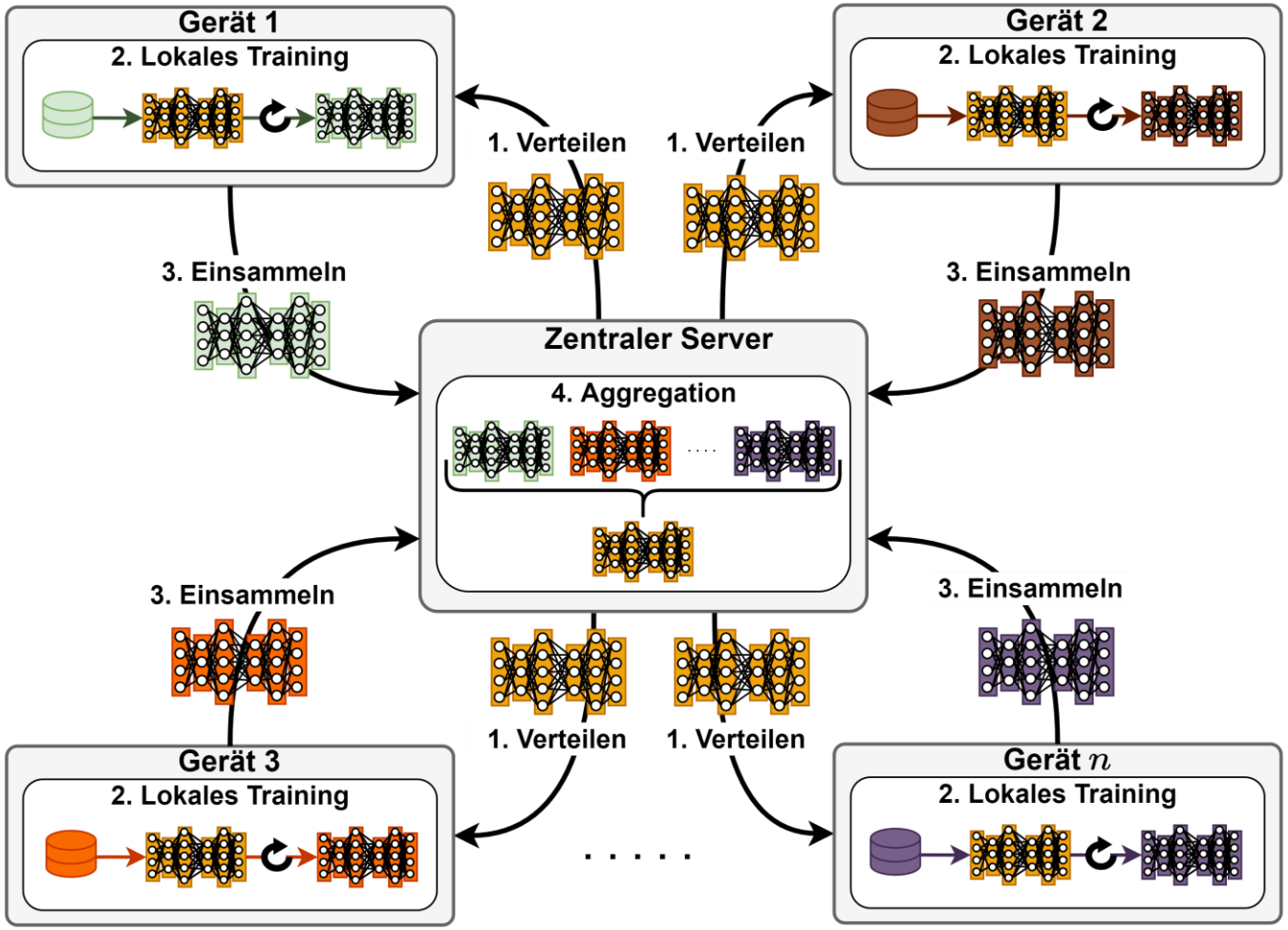
Zentrales Federated Learning



Zentrales föderales Lernen (ZFL) [12]

1. Globales Modell verteilen
2. Lokales Training
3. Einsammeln der lokalen Modelle

Zentrales Federated Learning

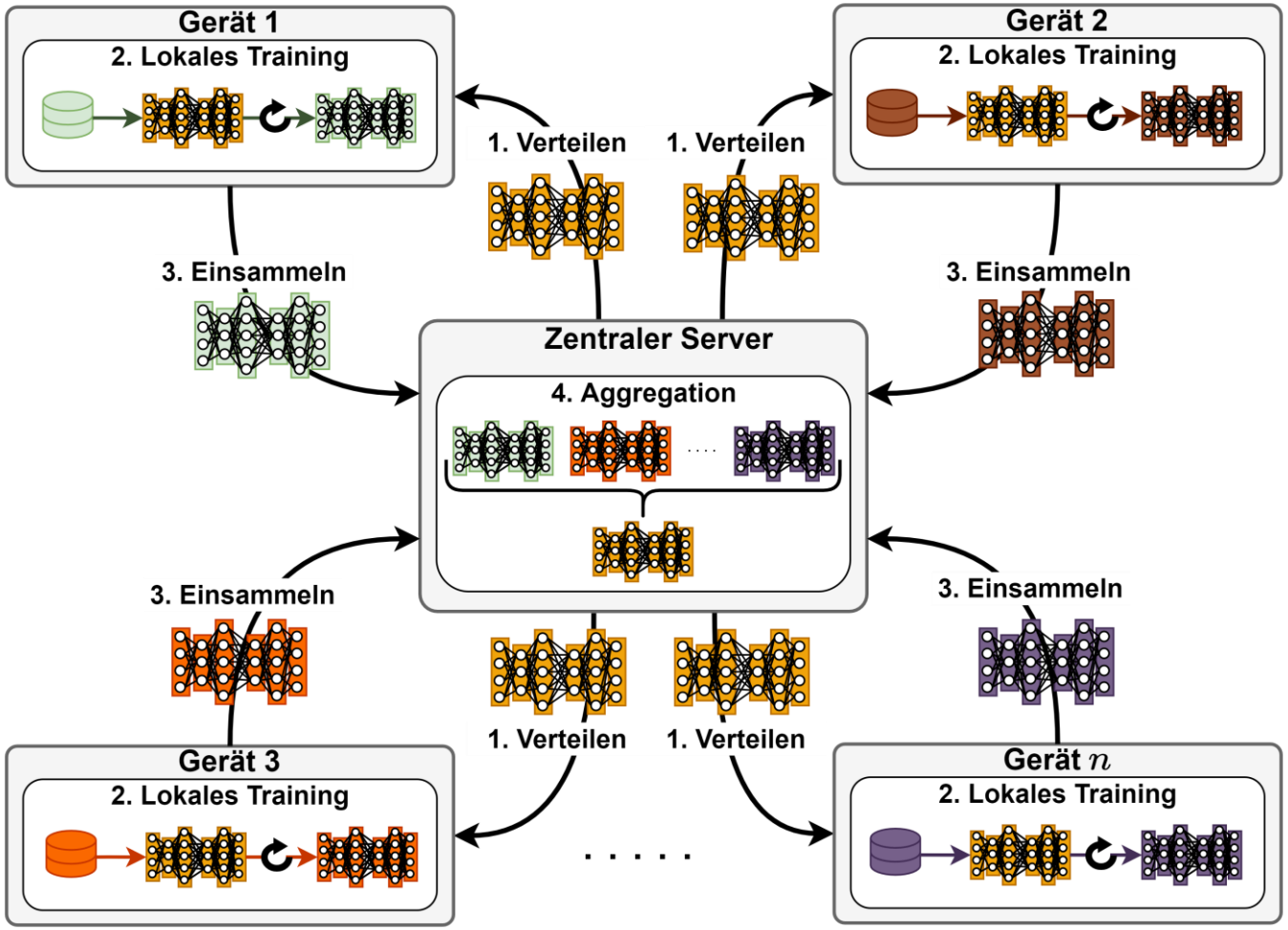


Zentrales föderales Lernen (ZFL)

1. Globales Modell verteilen
2. Lokales Training
3. Einsammeln der lokalen Modelle
4. Aggregation

Globales Modell | Lokales Modell | Lokale Daten → Kommunikation | Training

Zentrales Federated Learning



Zentrales föderales Lernen (ZFL)

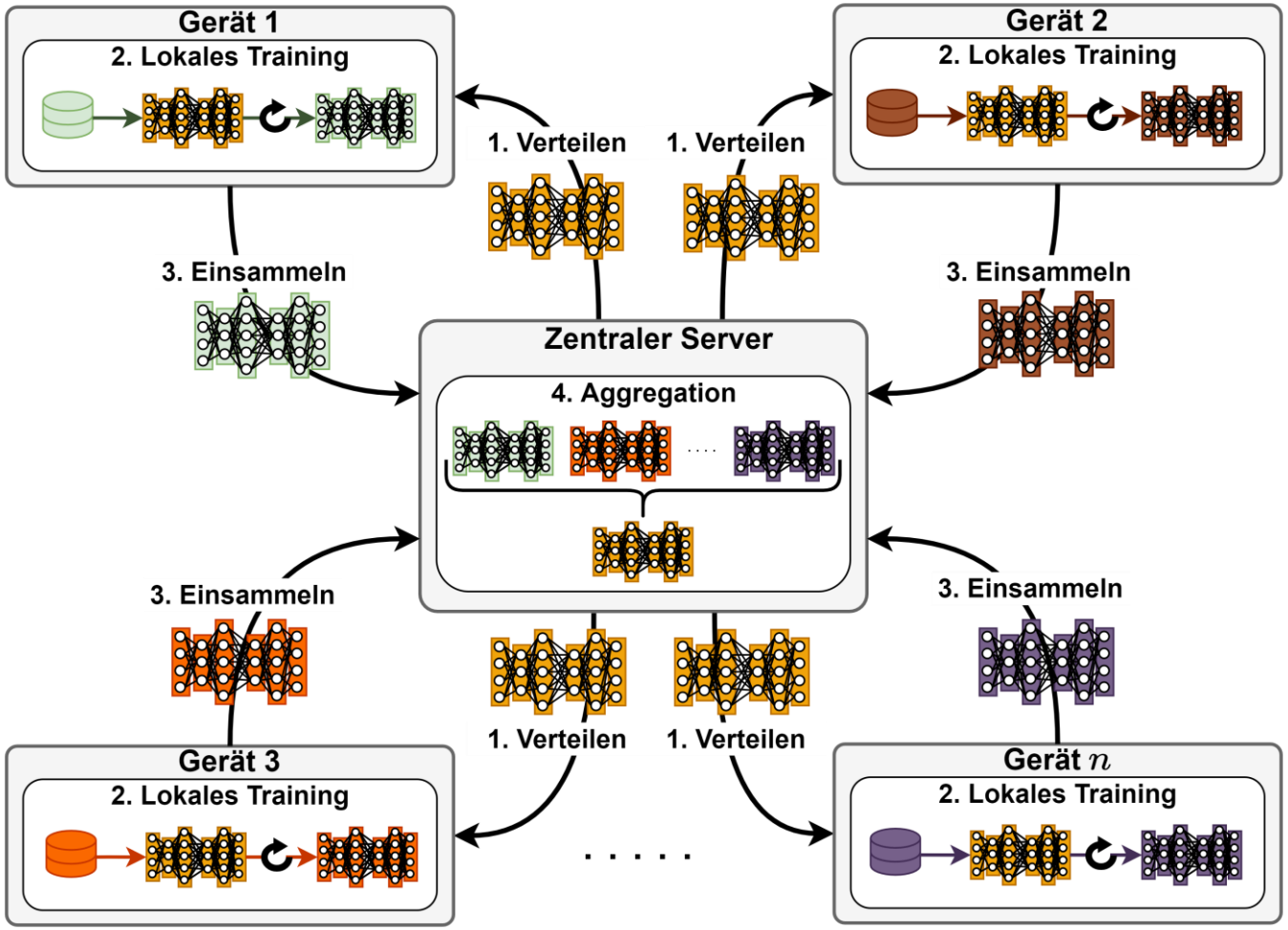
1. Globales Modell verteilen
2. Lokales Training
3. Einsammeln der lokalen Modelle
4. Aggregation

Vorteile

- ✓ Neue Daten werden schnell integriert
  Reduzierung der Datenübertragung
- ✓ Verbesserung der Datensicherheit

Globales Modell | Lokales Modell | Lokale Daten → Kommunikation | Training

Zentrales Federated Learning



Zentrales föderales Lernen (ZFL)

1. Globales Modell verteilen
2. Lokales Training
3. Einsammeln der lokalen Modelle
4. Aggregation

Vorteile

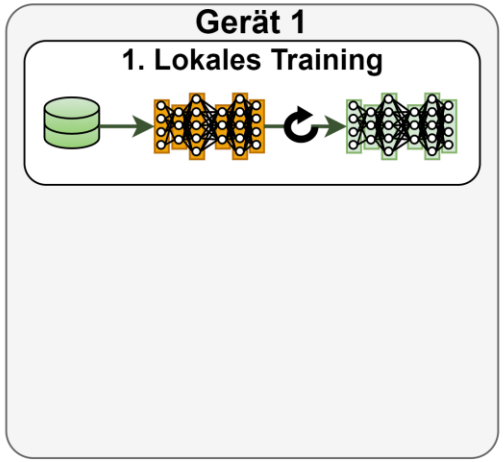
- ✓ Neue Daten werden schnell integriert
  Reduzierung der Datenübertragung
- ✓ Verbesserung der Datensicherheit

Nachteile

- × Ausfall des zentralen Servers kann ZFL verhindern
- × Schlechte Skalierbarkeit
- × Server kann korrumpiert sein

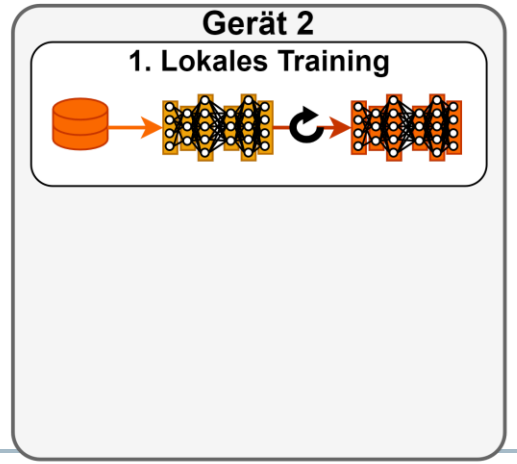
Globales Modell | Lokales Modell | Lokale Daten → Kommunikation | Training

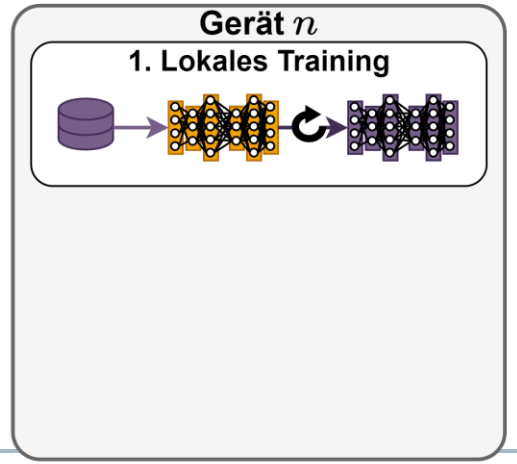
Dezentrales Federated Learning



Dezentrales föderales Lernen (DFL)

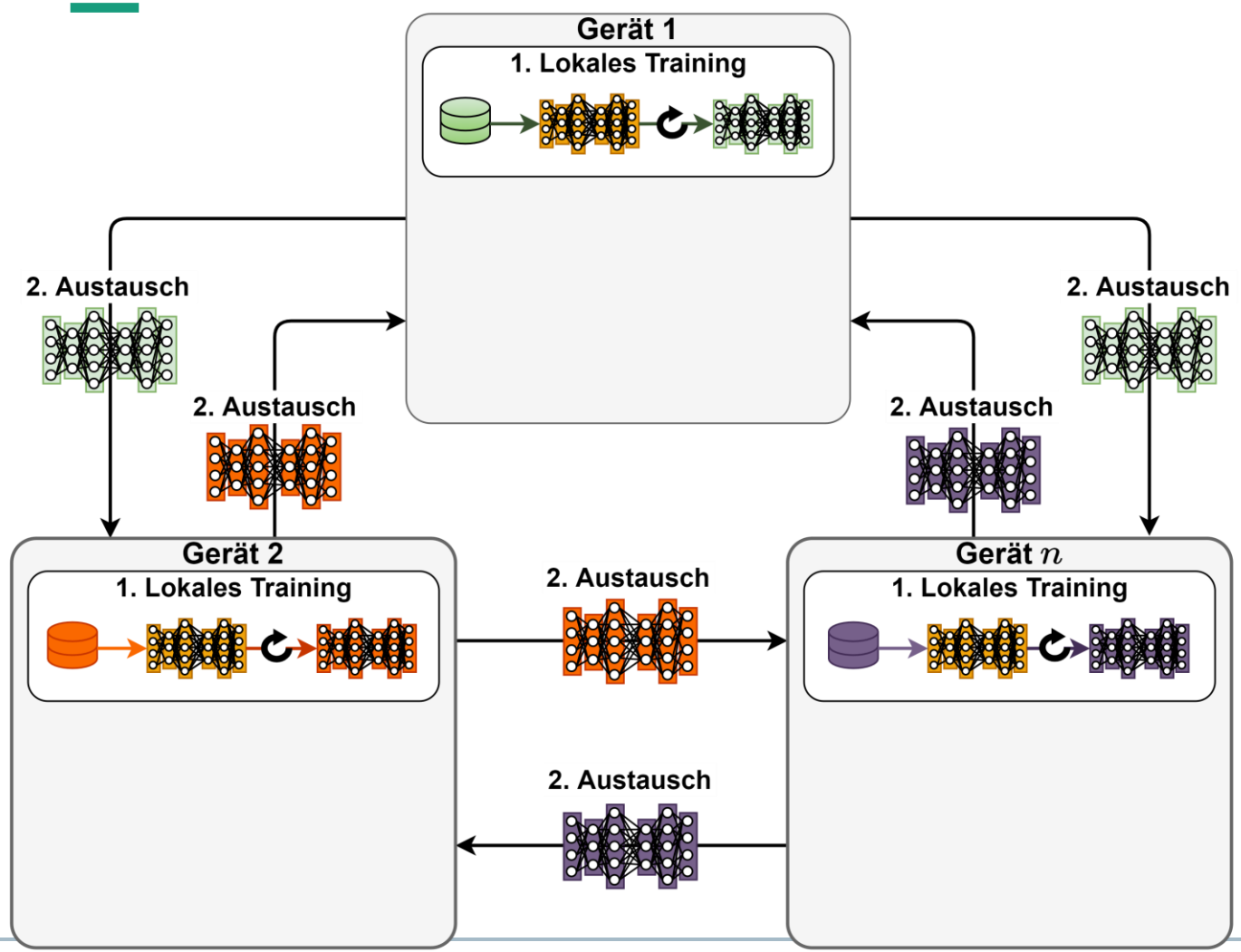
1. Lokales Modell Training





Fraunhofer IMS

Dezentrales Federated Learning

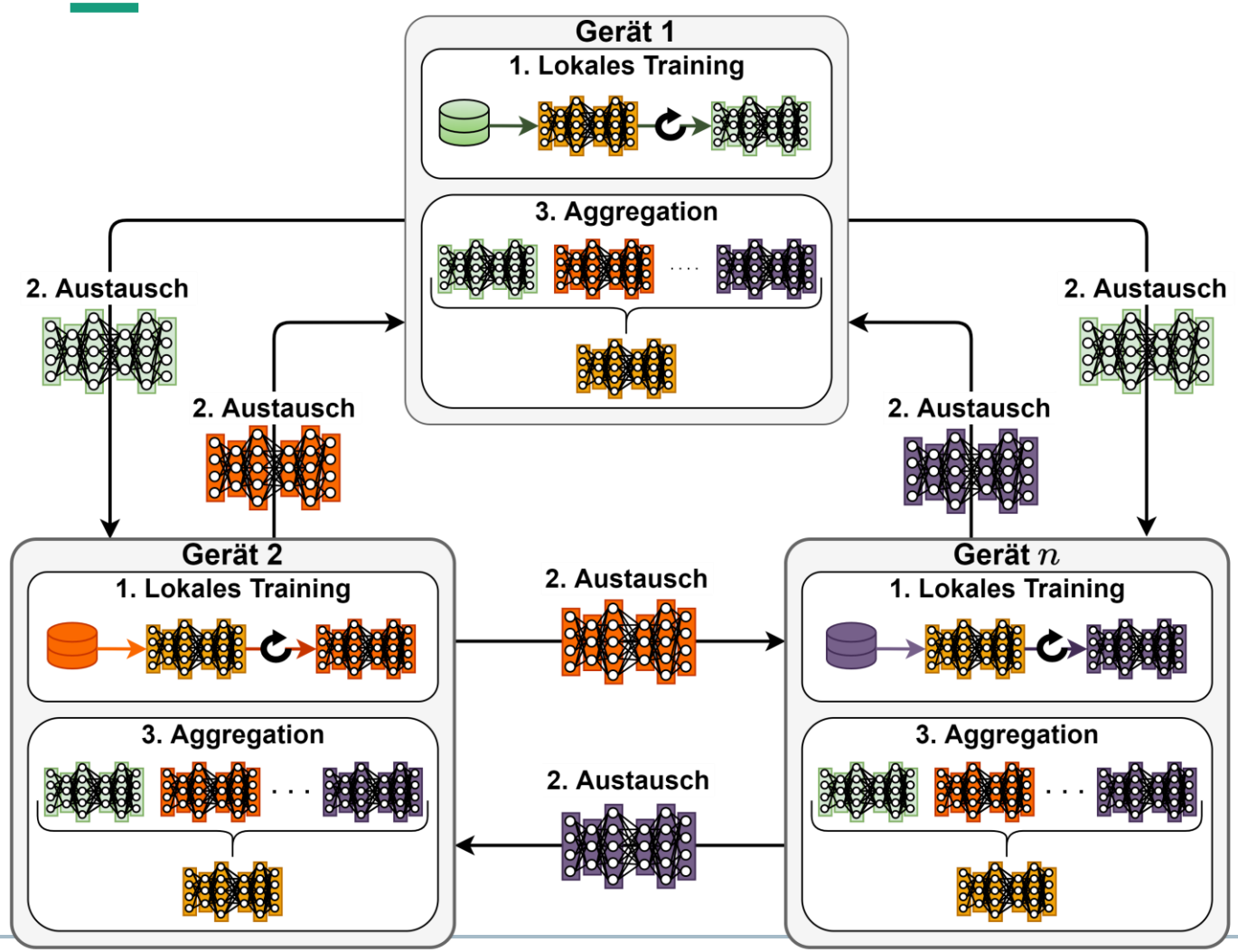


Dezentrales föderales Lernen (DFL)

1. Lokales Modell Training
2. Modelle austauschen

Globales Modell | Lokales Modell | Lokale Daten → Kommunikation | Training

Dezentrales Federated Learning



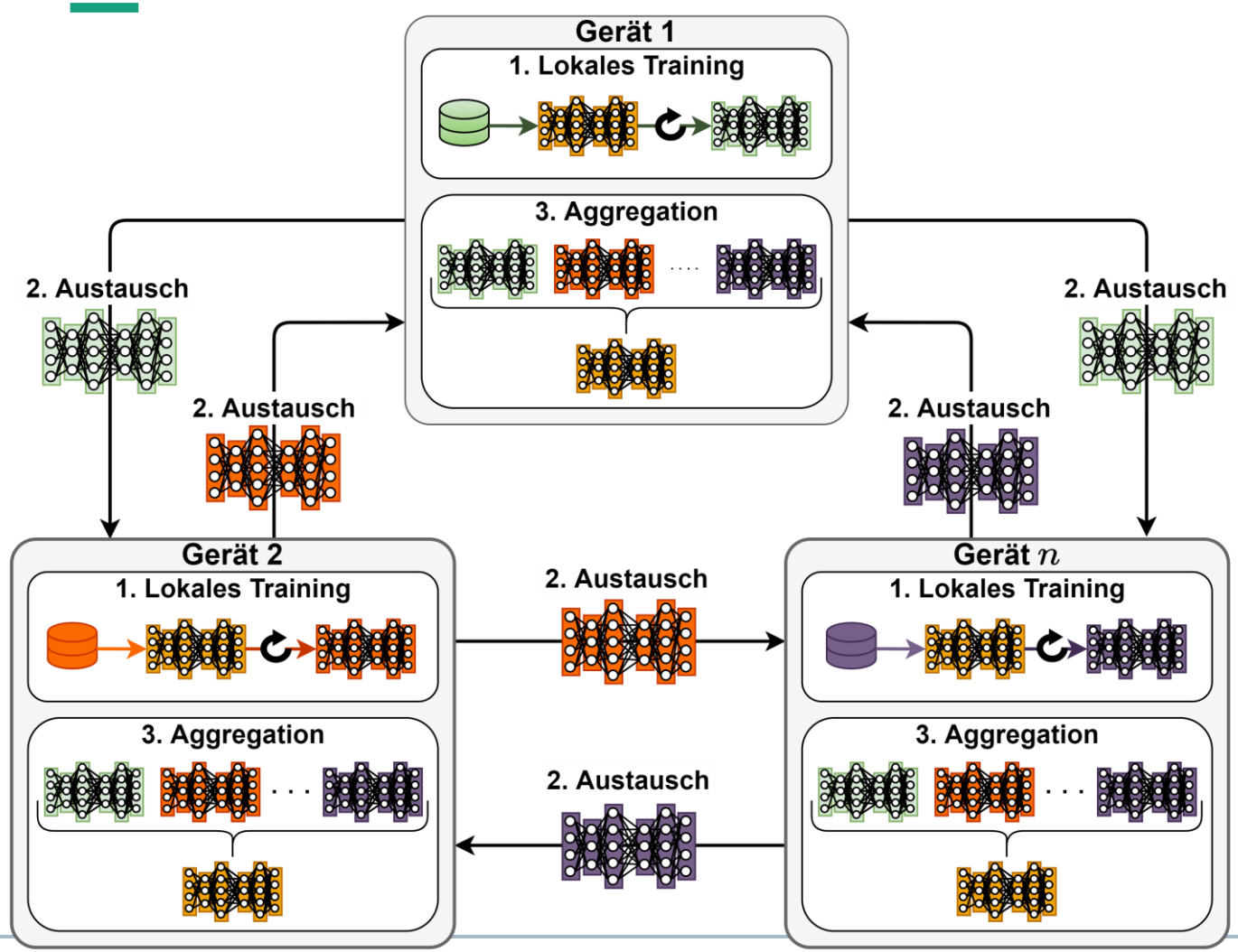
Dezentrales föderales Lernen (DFL)

1. Lokales Modell Training
2. Modelle austauschen
3. Aggregation

Globales Modell | Lokales Modell | Lokale Daten → Kommunikation | Training

Fraunhofer IMS

Dezentrales Federated Learning



Dezentrales föderales Lernen (DFL)

1. Lokales Modell Training
2. Modelle austauschen
3. Aggregation

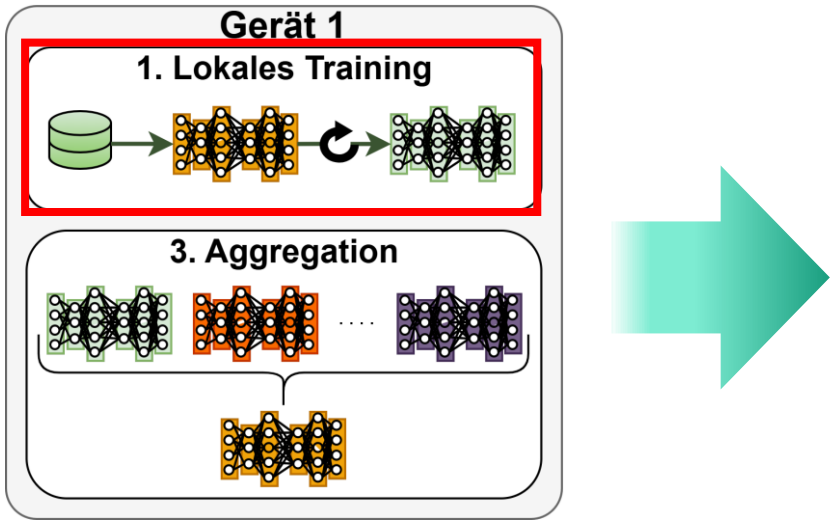
Nachteile & Herausforderungen

- × Eine heterogene Datenverteilungen stellt Herausforderungen an die Genauigkeit
- × Hoher Kommunikationsaufwand
- × On-Device Training auf Mikrocontrollern stellt Herausforderung an die Speicherkapazität

Fraunhofer IMS

Herausforderung 1

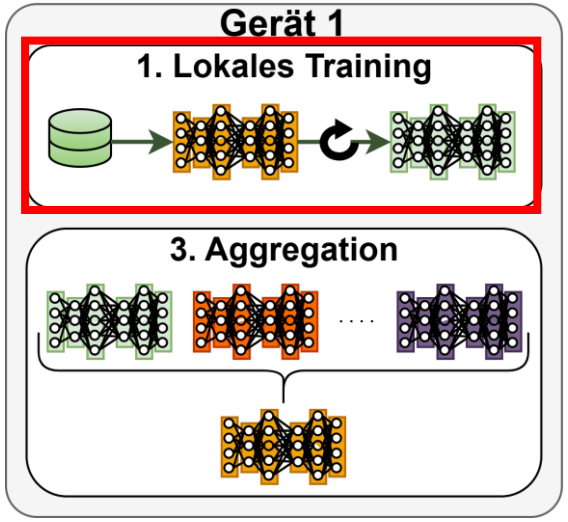
On-Device Training



Fraunhofer IMS

Herausforderung 1

On-Device Training



PyTorch

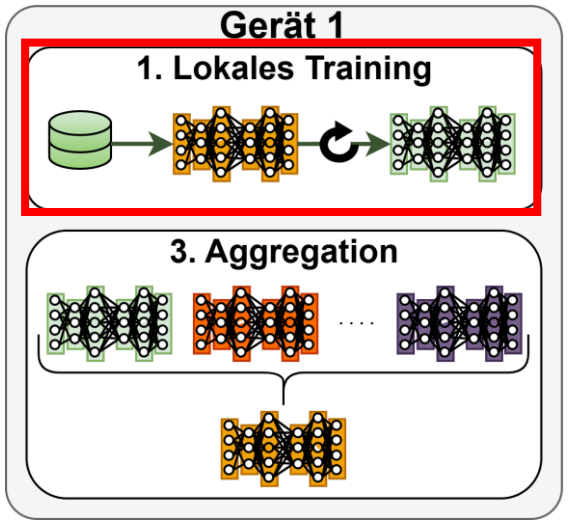
TensorFlow

Keras

Fraunhofer IMS

Herausforderung 1

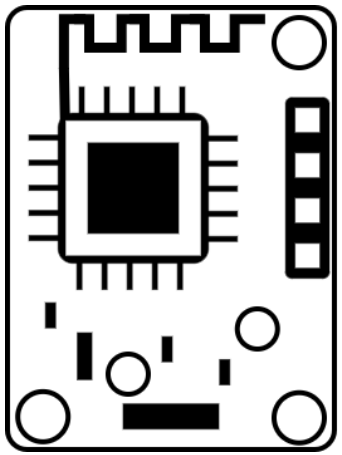
On-Device Training



PyTorch

TensorFlow

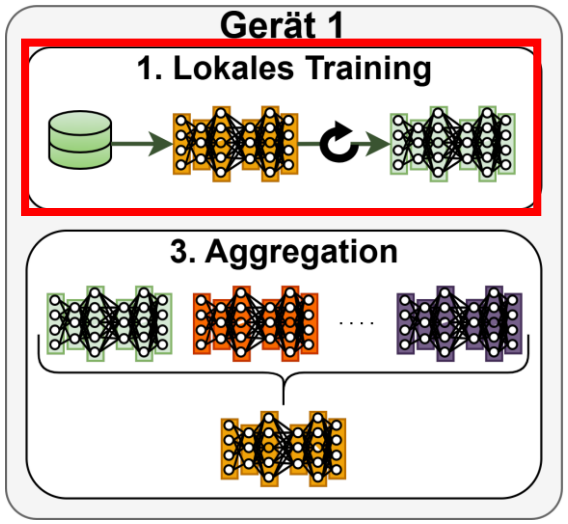
Keras



Speicher	Cloud ML (NVIDIA V100 [34])	Red.	Mobile ML (iPhone 15 [35])	Red.	TinyML (ESP32 [36])
RAM	> 16 GB	2,67 ×	> 6 GB	11.538 ×	520 kB

Herausforderung 1

On-Device Training

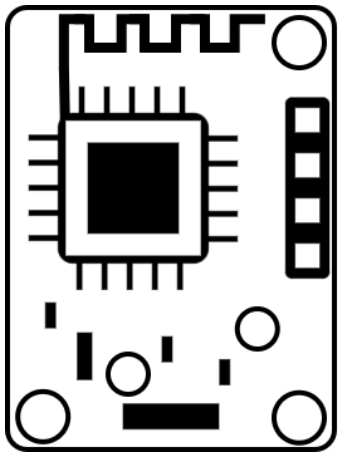


- Frameworks benötigen Python
- Frameworks benötigen zu viel Speicherplatz
- MCU besitzen keine performanten CPUs & GPUs

PyTorch

TensorFlow

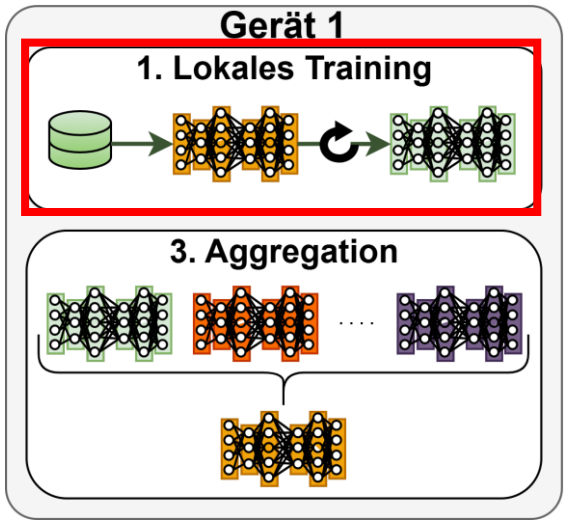
Keras



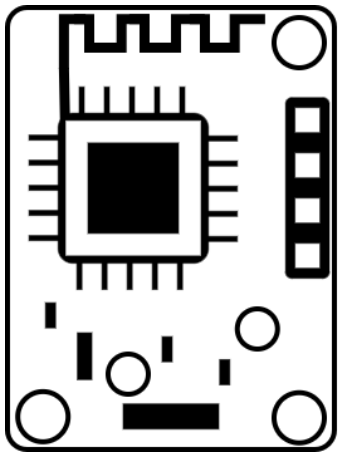
Speicher	Cloud ML (NVIDIA V100 [34])	Red.	Mobile ML (iPhone 15 [35])	Red.	TinyML (ESP32 [36])
RAM	> 16 GB	2,67 ×	> 6 GB	11.538 ×	520 kB

Herausforderung 1

On-Device Training



- Frameworks benötigen Python
- Frameworks benötigen zu viel Speicherplatz
- MCU besitzen keine performanten CPUs & GPUs

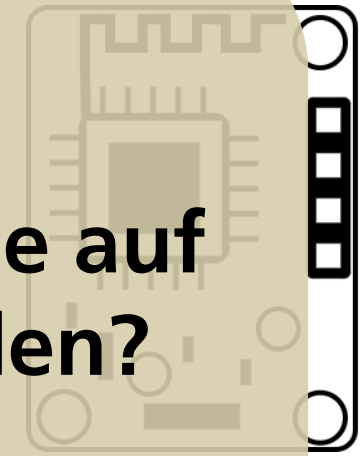


Speicher	Cloud ML (NVIDIA V100 [34])	Red. →	Mobile ML (iPhone 15 [35])	Red. →	TinyML (ESP32 [36])
RAM	> 16 GB	2,67 ×	> 6 GB	11.538 ×	520 kB

Herausforderung 1

On-Device Training

- Frameworks benötigen Python
- Frameworks benötigen zu viel Speicherplatz
- MCU besitzen keine performanten CPUs & GPUs



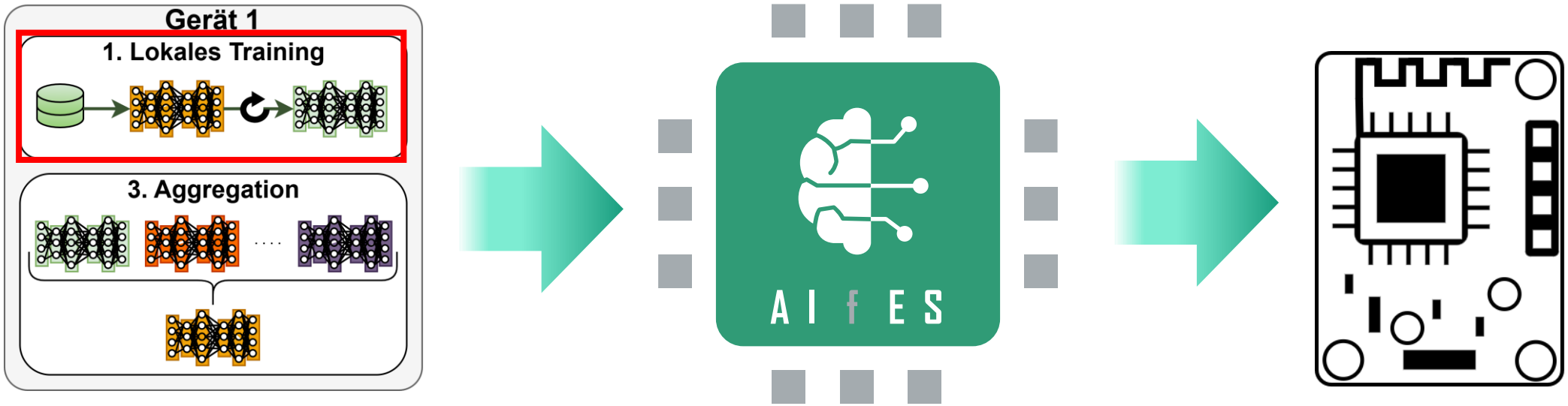
Wie können Machine Learning Modelle auf einem Mikrocontroller trainiert werden?

🤔

Speicher	Cloud ML (NVIDIA V100 [34])	Red.	Mobile ML (iPhone 15 [35])	Red.	TinyML (ESP32 [36])
RAM	> 16 GB	2,67 ×	> 6 GB	11.538 ×	520 kB

Herausforderung 1

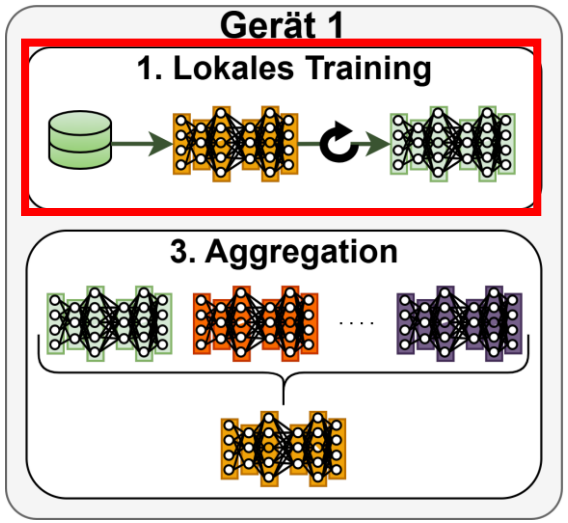
On-Device Training

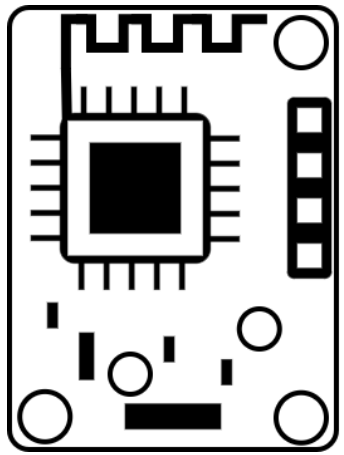


Fraunhofer IMS

Herausforderung 1

On-Device Training





- **AIfES** (AI for Embedded Systems)
- ML Framework für Mikrocontroller und Open Source
- Unterstützt Inference und On-Device Training von
  - Fully Connected Neural Networks
  - Convolutional Neural Networks

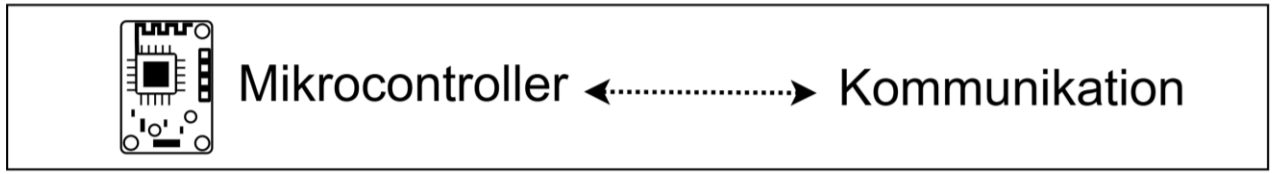
Wulfert et al., IEEE - TPAMI

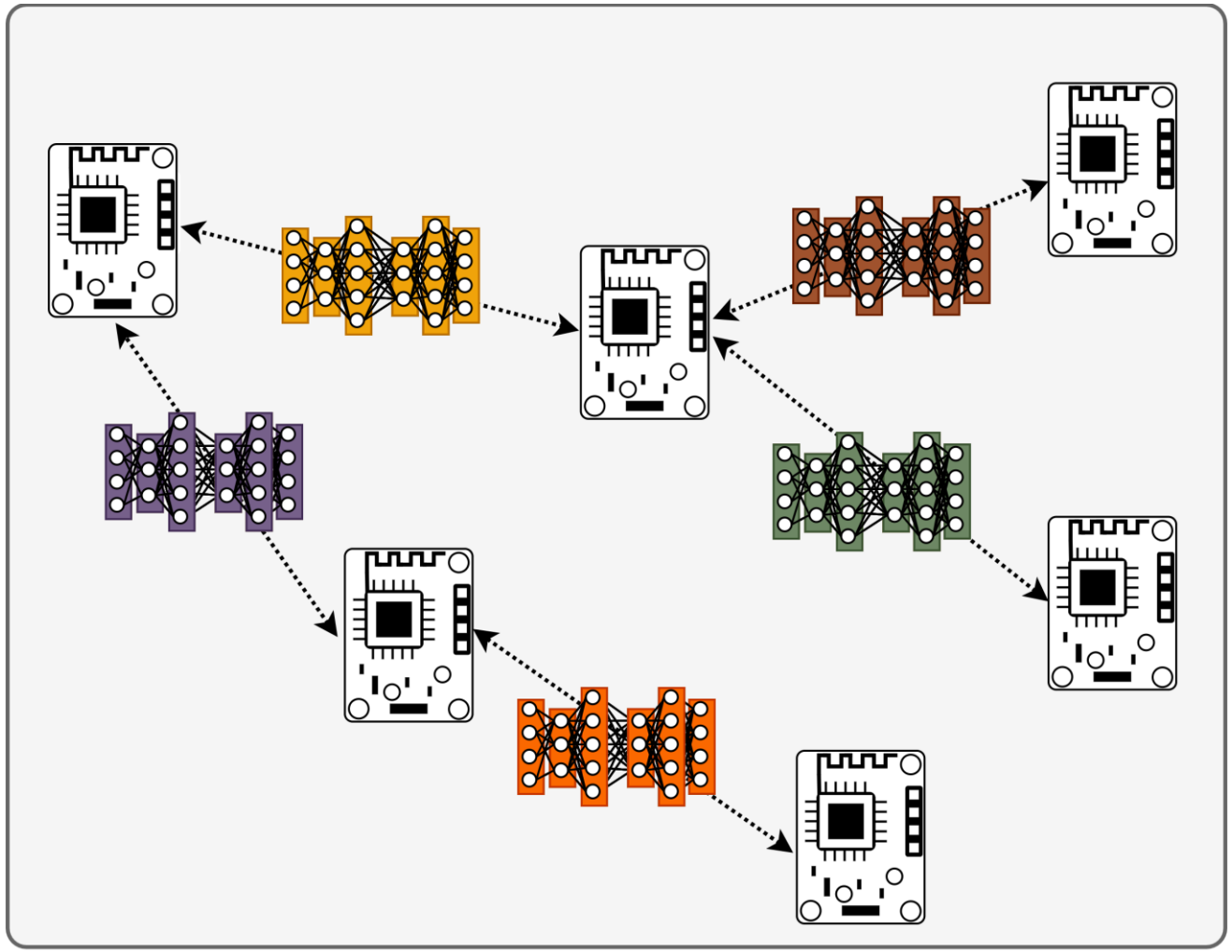
GitHub

Fraunhofer IMS

Herausforderung 2

Federated Learning auf Mikrocontrollern

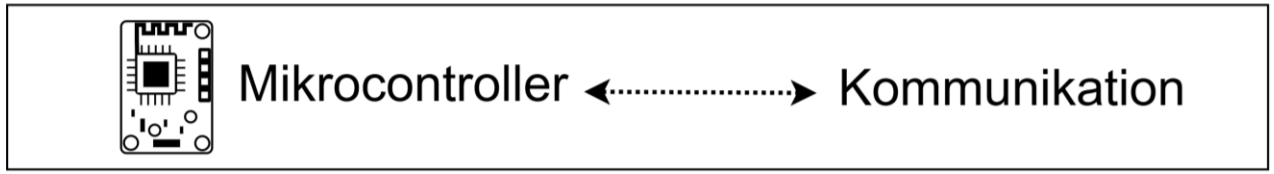


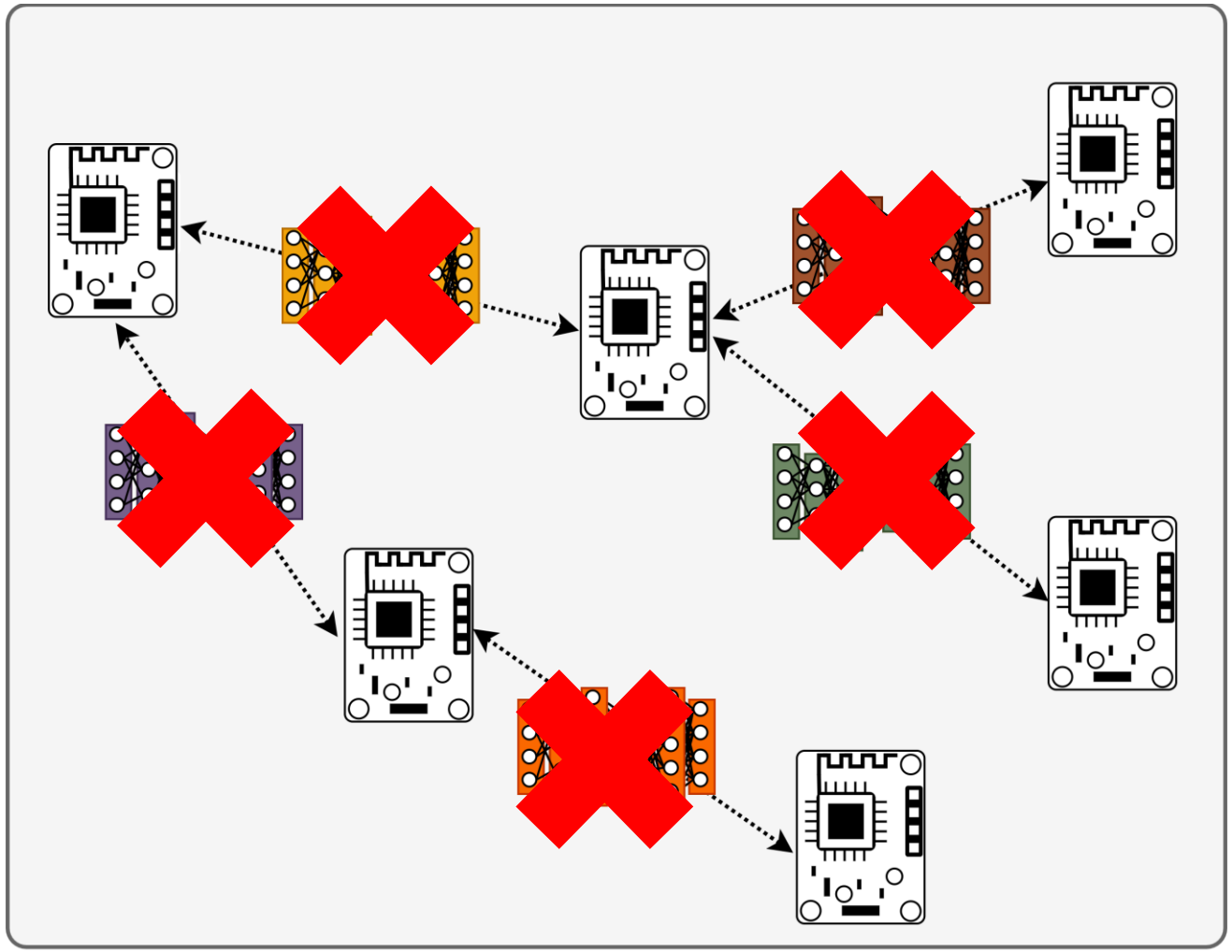


Fraunhofer IMS

Herausforderung 2

Federated Learning auf Mikrocontrollern

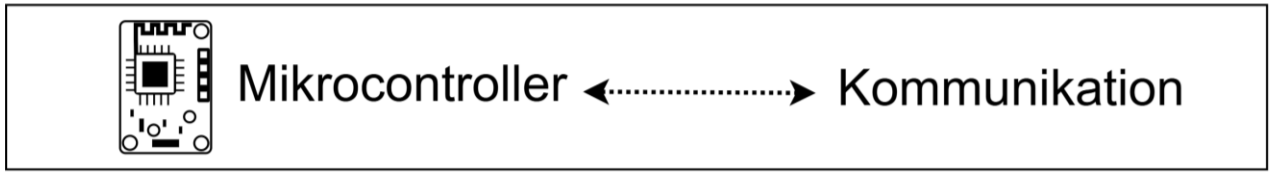




Fraunhofer IMS

Herausforderung 2

Federated Learning auf Mikrocontrollern

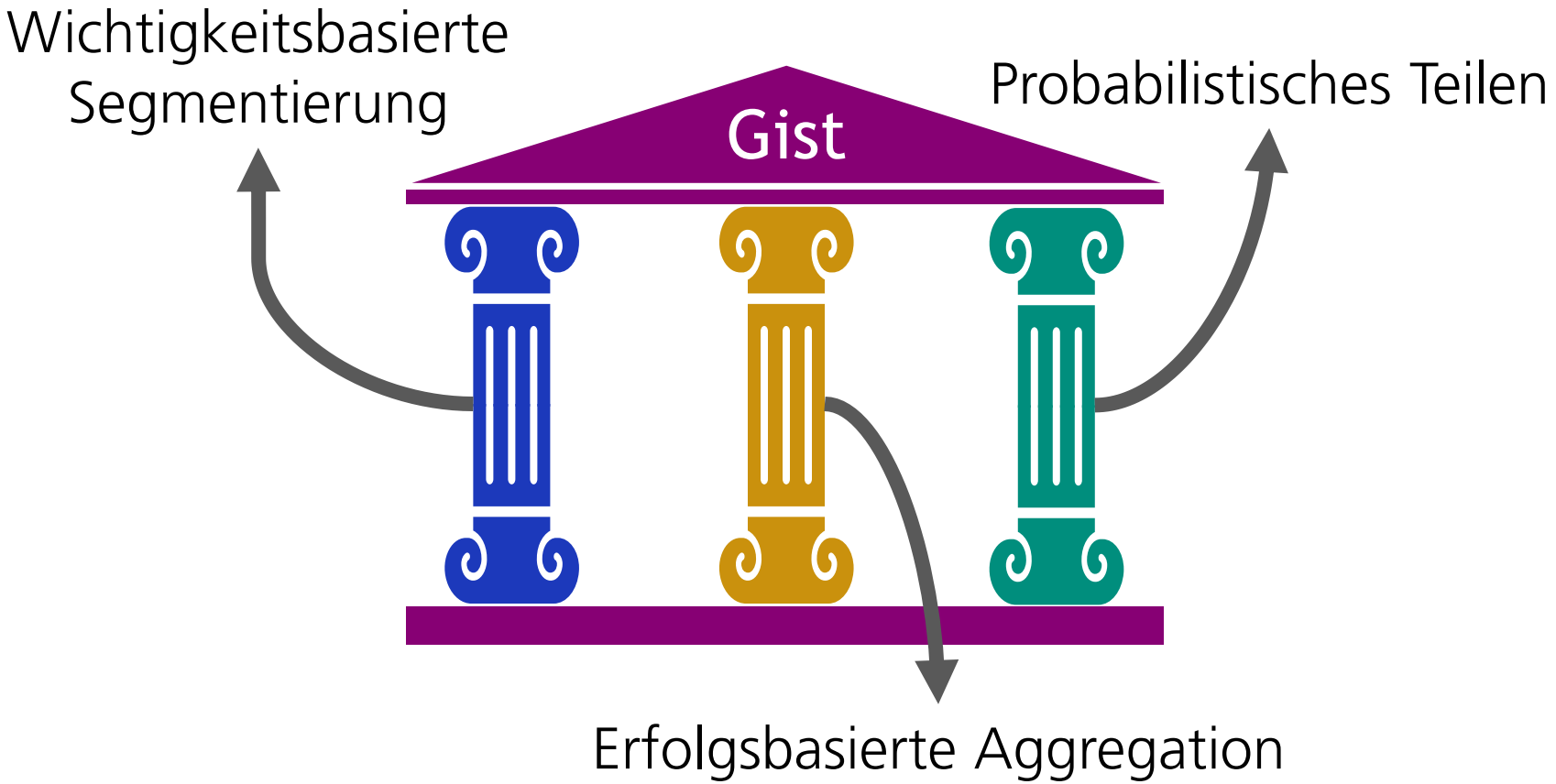


Ist es möglich, die Kommunikation/Rechenleistung zu reduzieren und gleichzeitig die Konvergenz zu erhalten?

🤔

Herausforderung 2

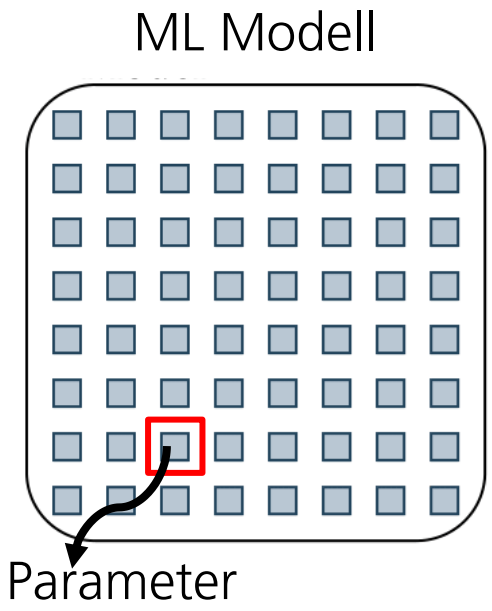
Gist



Asadi, Wulfert et al., „Gist - Optimizing …", ACM MobiCom FLEdge-AI '25 – **Best Paper Award**
Asadi, Wulfert et al., „Road to Tiny Reality …", MobiCom Posters 2025

Gist

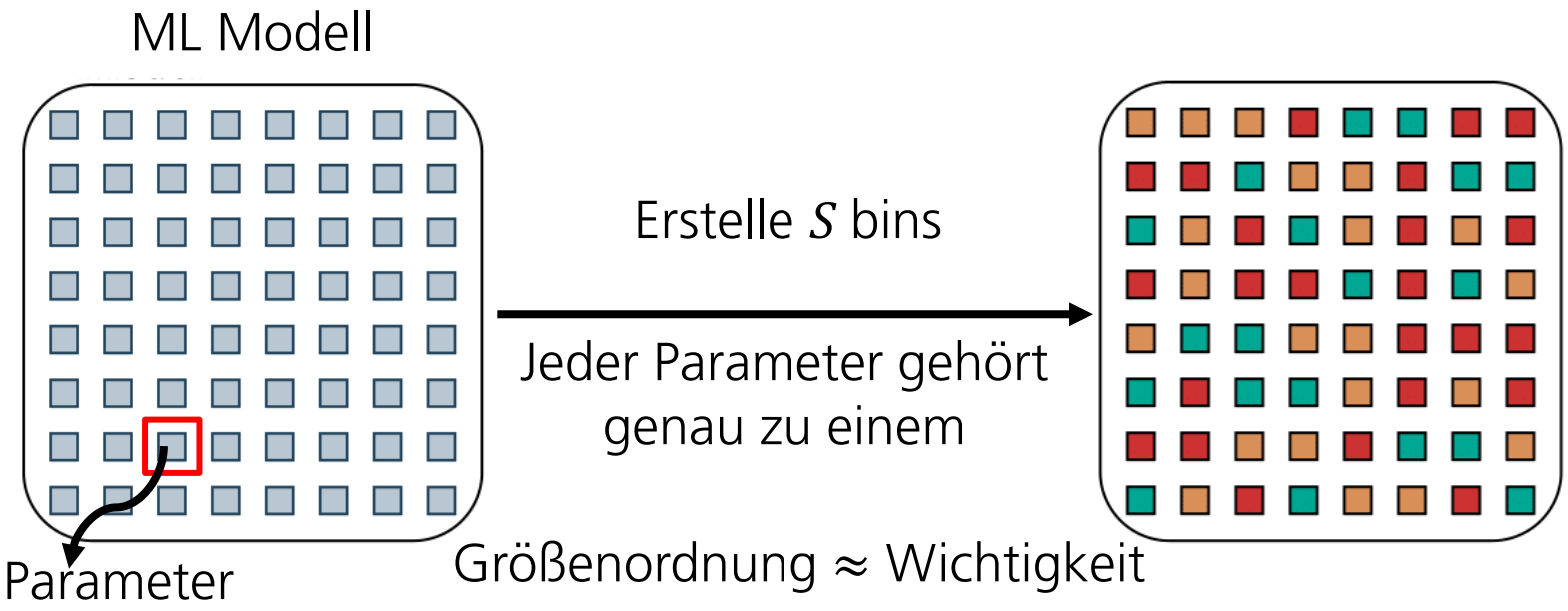
Segmentierung



Fraunhofer IMS

Gist

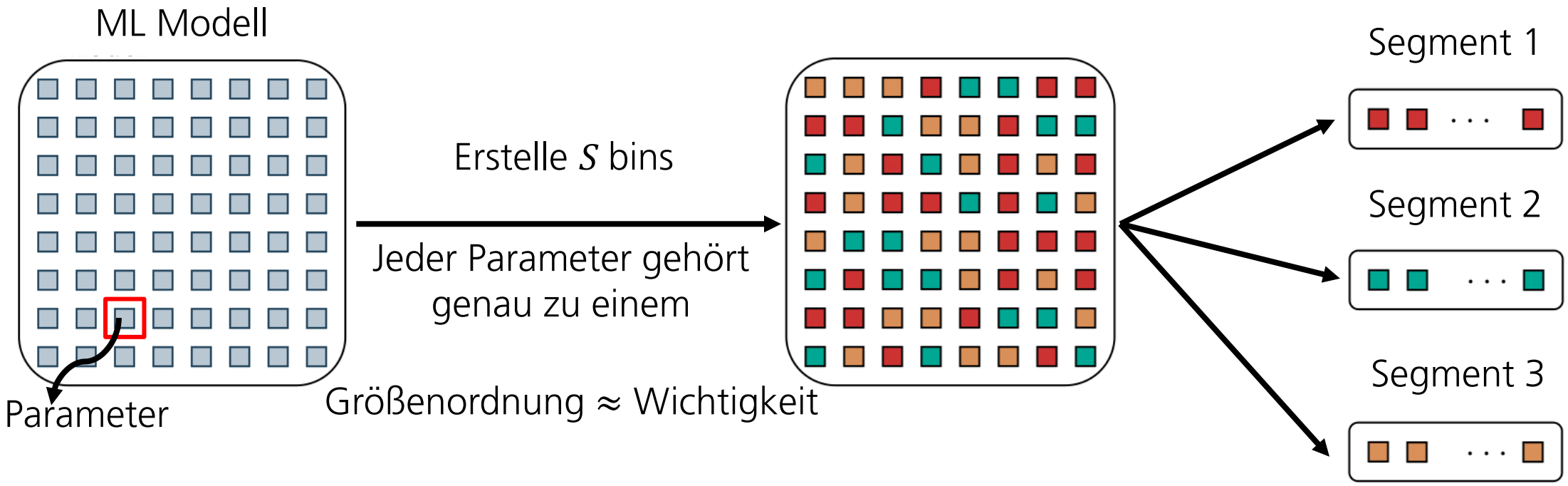
Segmentierung



Fraunhofer IMS

Gist

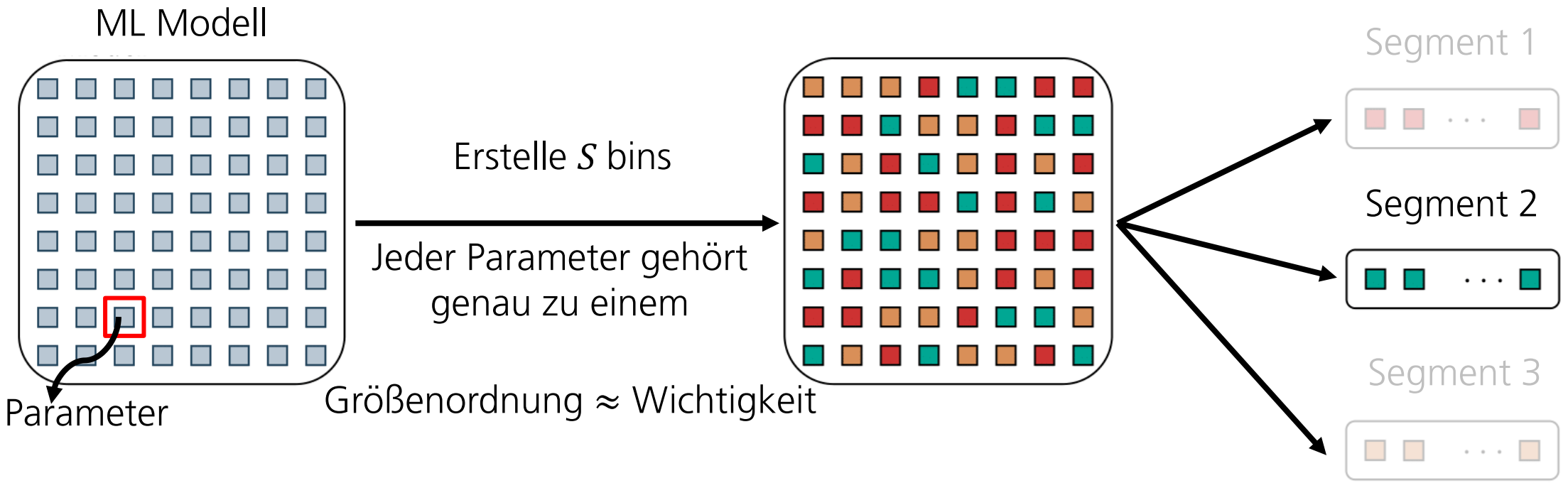
Segmentierung



Fraunhofer IMS

Gist

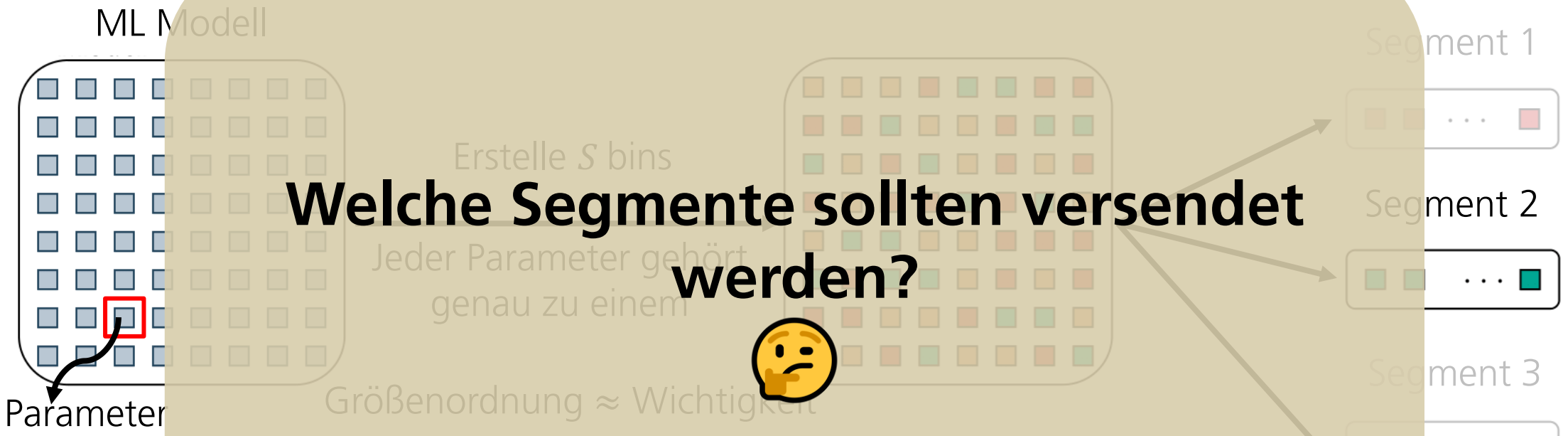
Segmentierung



Konzentration des Kommunikationsbudgets auf wirkungsvolle Parameter!

Gist

Segmentierung



Welche Segmente sollten versendet werden?

🤔

Erstelle S bins

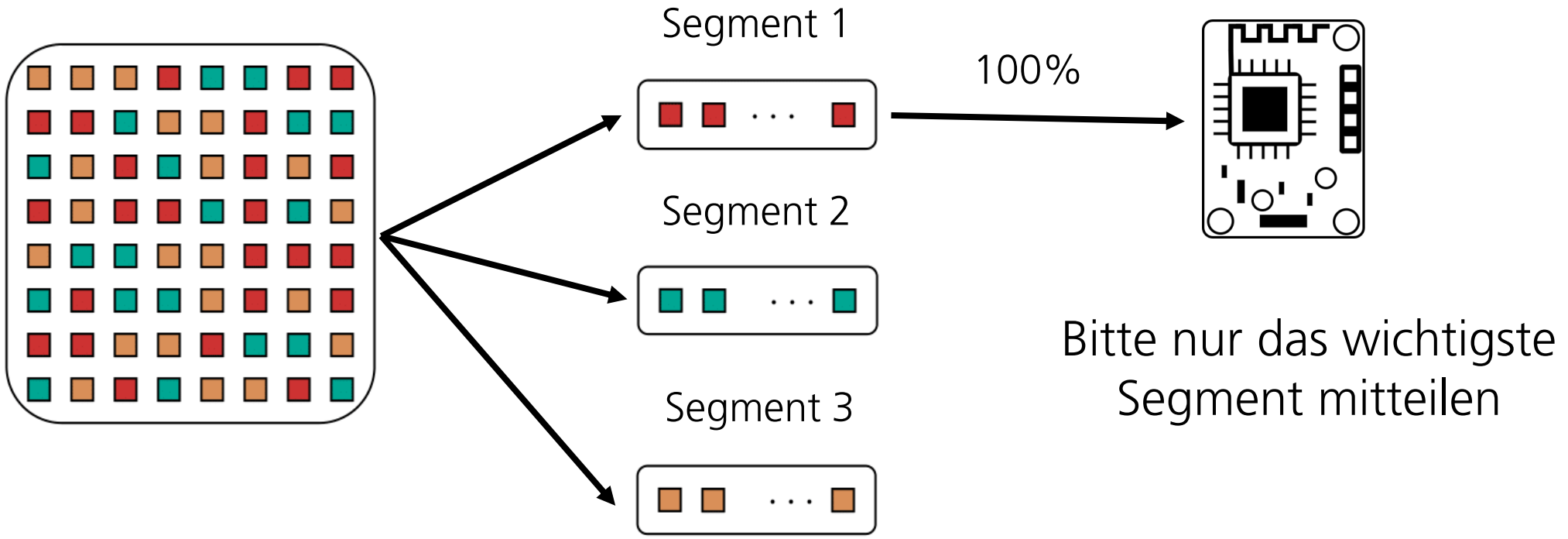
Jeder Parameter gehört genau zu einem

Größenordnung ≈ Wichtigkeit

Konzentration des Kommunikationsbudgets auf wirkungsvolle Parameter!

Gist

Probabilistisches Teilen



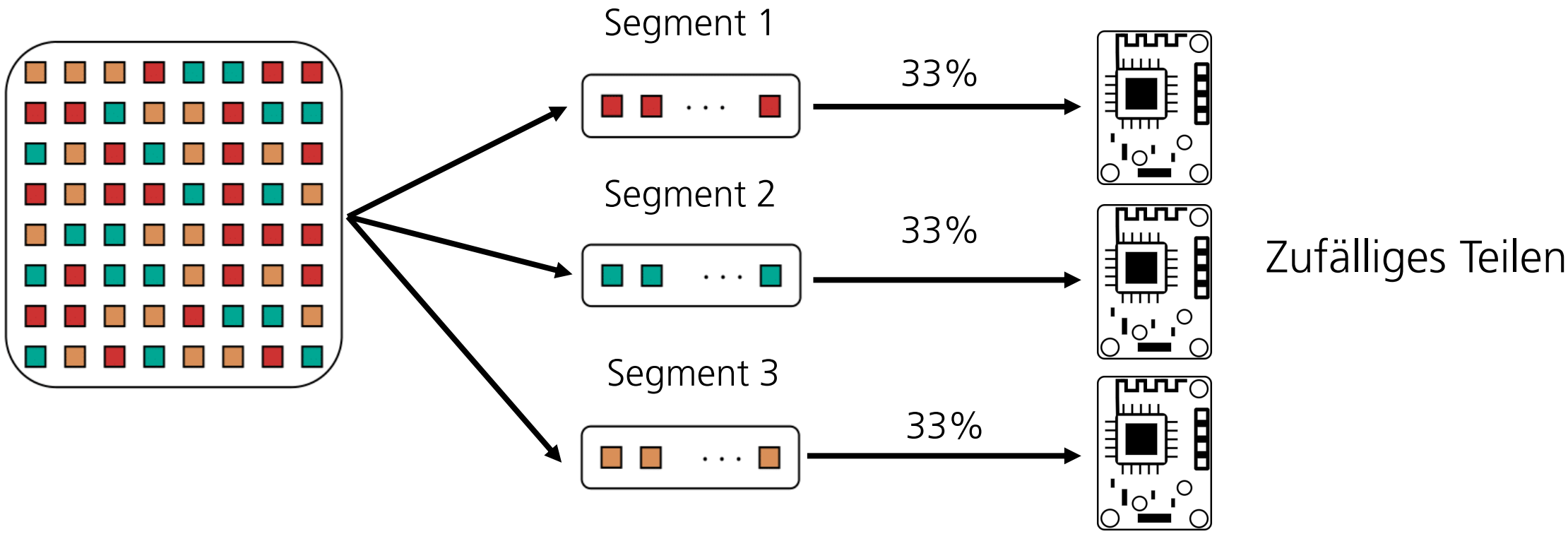
Unzureichende Parameterabdeckung/-vielfalt

➔ Modellabweichung

➔ Beeinträchtigung der Konvergenz!

Gist

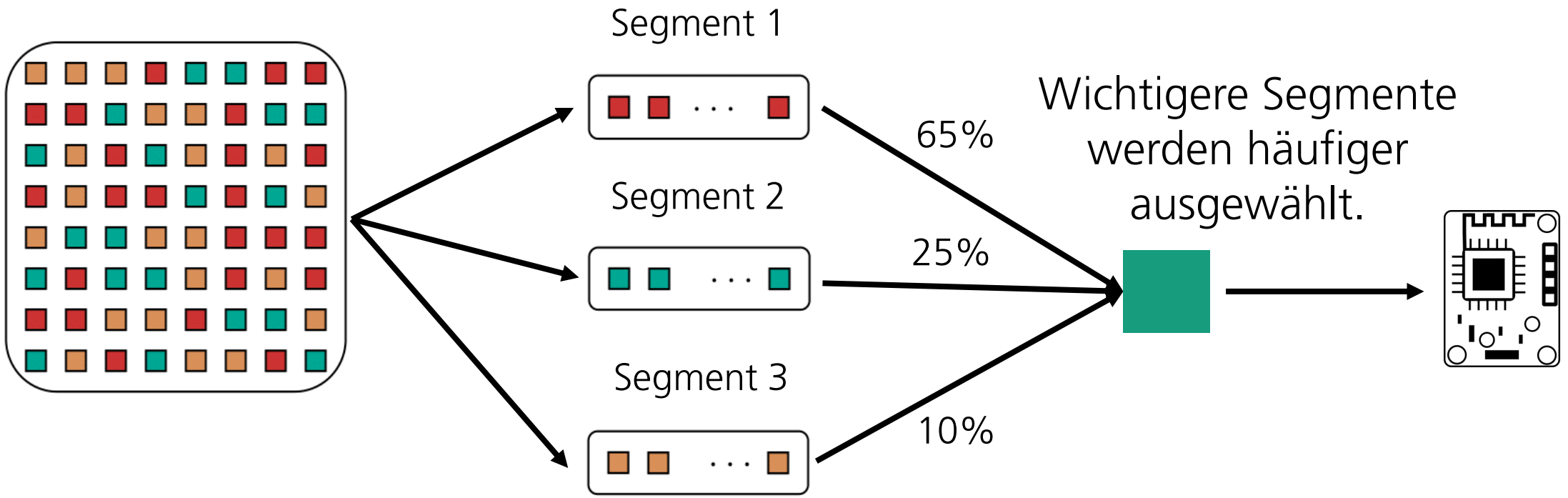
Probabilistisches Teilen



Bessere Vielfalt, jedoch wirkt es wie zufällige Segmentierung oder sogar noch unvorteilhafter!

Gist

Probabilistisches Teilen



Wichtige Segmente verwenden, weniger wichtige trotzdem erkunden.

Fraunhofer IMS

Gist

Probabilistisches Teilen

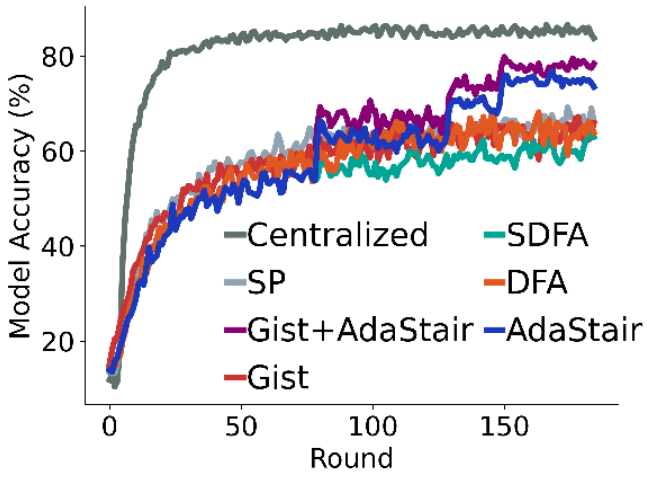
Schnellere Verbreitung nützlicher Informationen ohne Verlust der Vielfalt!

Fraunhofer IMS

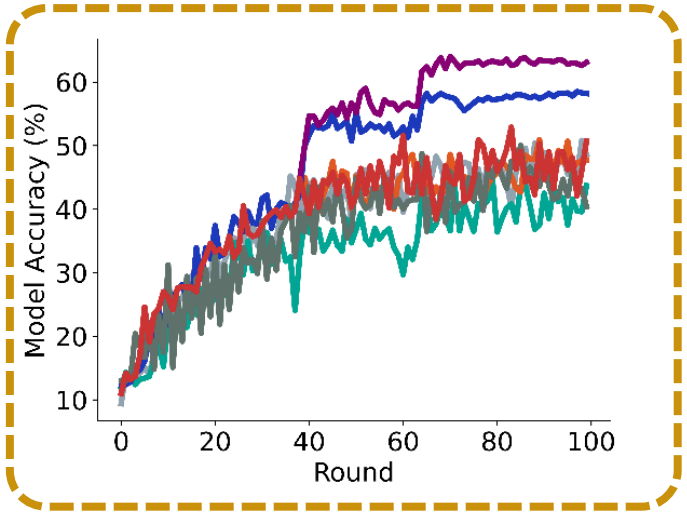
Gist

Wie schneidet **Gist** bei verschiedenen Modellen und Datensätzen ab?

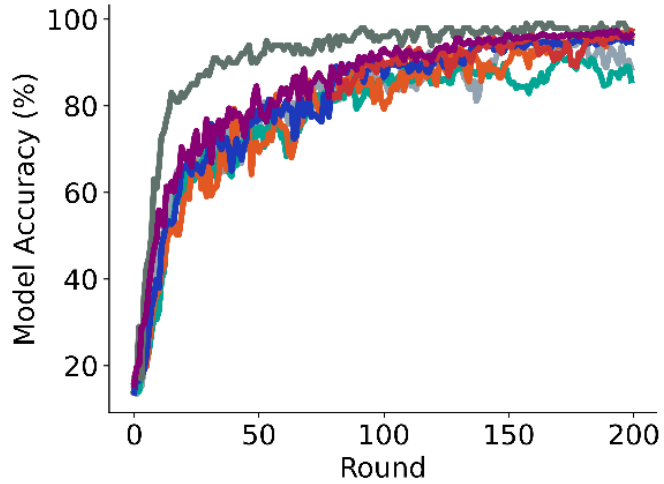
FCN -MNIST



MobileNet – CIFAR10



LeNet5 - MNIST



Gist > Basislinien um ~4–5 % im Laufe der Zeit und bei Konvergenz.

Gist erzielt eine Steigerung von bis zu *20 %*, wenn es mit Adastair kombiniert wird.

Gist hat einen größeren Einfluss auf komplexere Modelle.

AdaStair: Wulfert et al., CoNEXT: DistributedML'23

Zusammenfassung

AIfES

Open Source ML Framework für eingebettete Systeme

GitHub

Paper

Zusammenfassung

AIfES

Open Source ML Framework für eingebettete Systeme

GitHub

Paper

Gist

Ein dezentralisierter Segmentierungs- und Aggregationsansatz für Tiny Federated Learning auf Mikrocontrollern

- Höhere Genauigkeit und schnellere Konvergenz, insbesondere bei komplexeren Modellen.
- Weitere Details und Ergebnisse finden Sie im Paper.

Gist

Zwe³

Fraunhofer IMS

Vielen Dank!

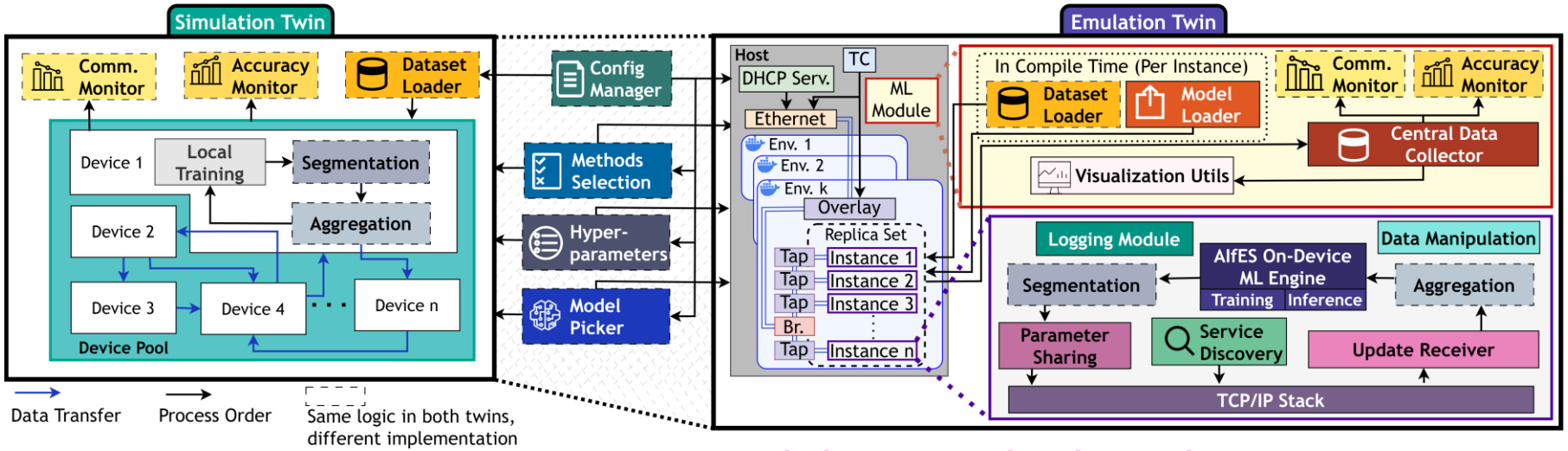
Dr. Lars Wulfert

Smart Embedded Systems
lars.wulfert@ims.fraunhofer.de

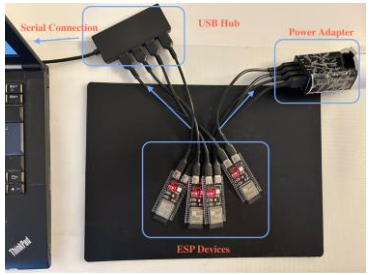
www.aifes.ai

Follow us on LinkedIn

SCAN ME



Same modules are deployed on real devices for measurement



Simulation helps faster iterations while emulation provides higher fidelity!

Asadi, Wulfert et al. “Road to Tiny Reality…” MobiCom Posters 2025

Fraunhofer IMS