LAMERF

LABORATÓRIO DE PESQUISAS EM MICROELETRÔNICA E RADIOFREQUÊNCIAS

UFERSA

PER SCIENTIA ARIDAM TERRAM FLORET

LIEB

TinyML

Is the future of machine learning tiny?

DeepUfersa

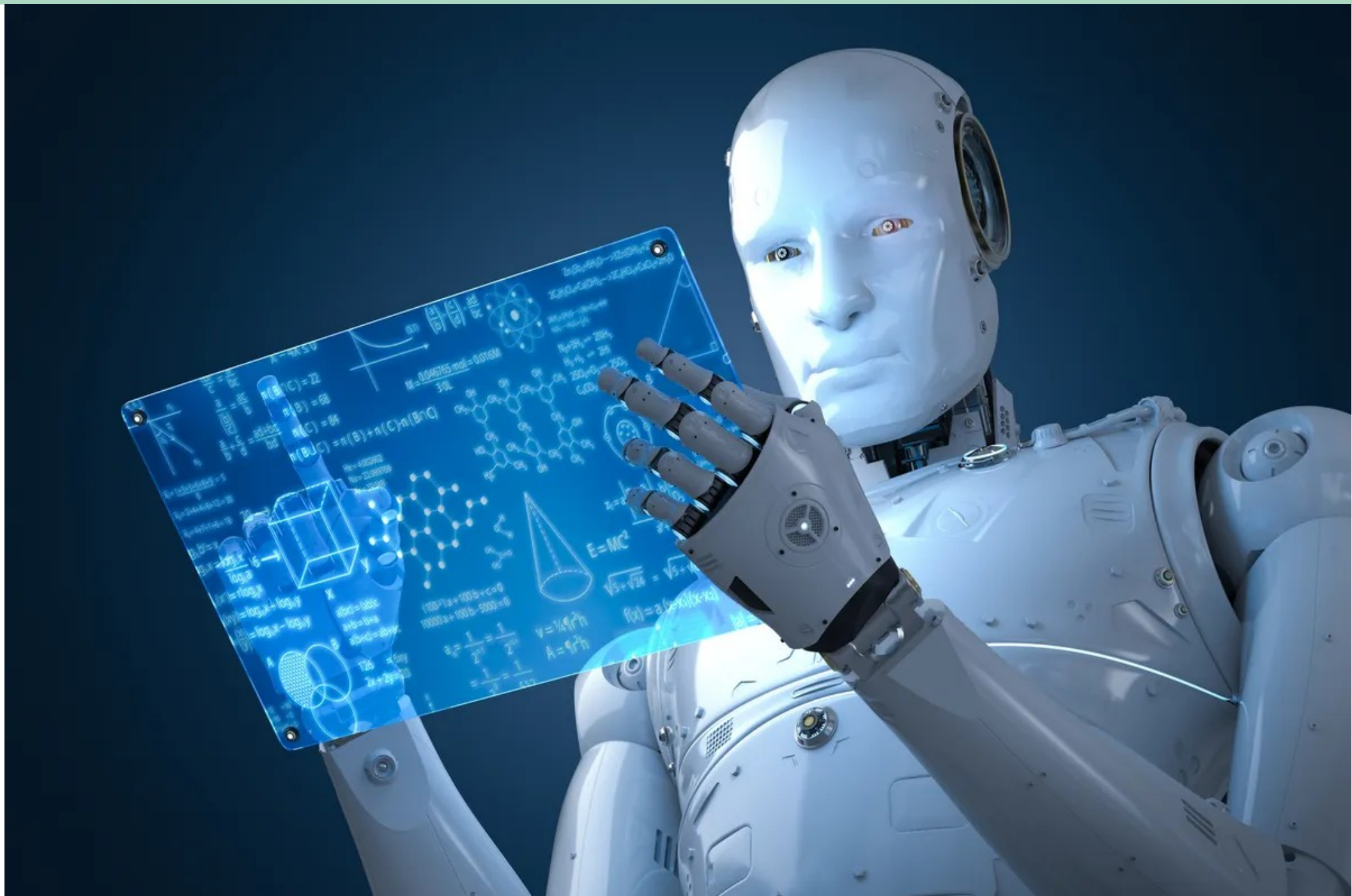
Outline

- Intro
- TinyML

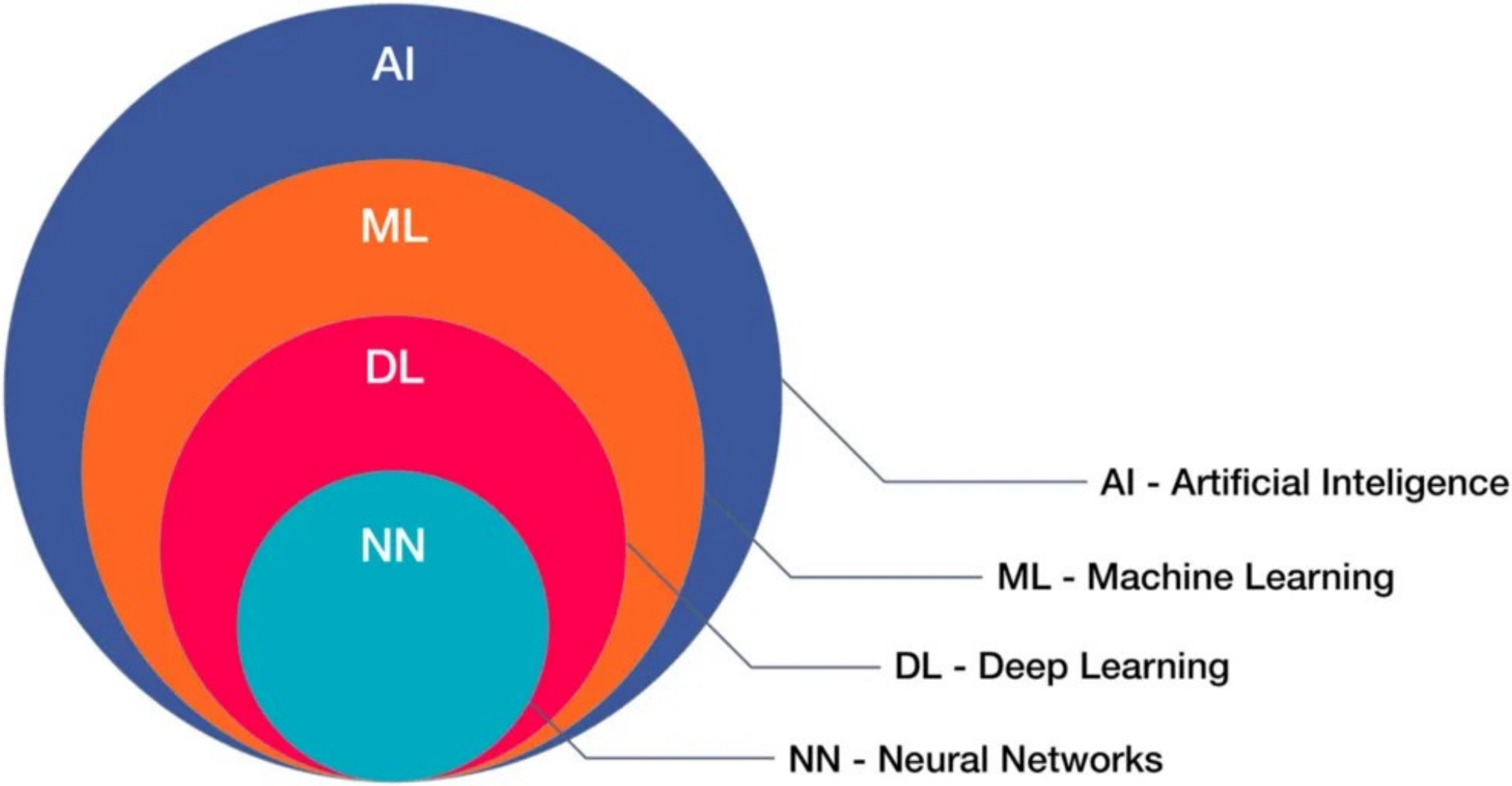
DeepUfersa

Intro

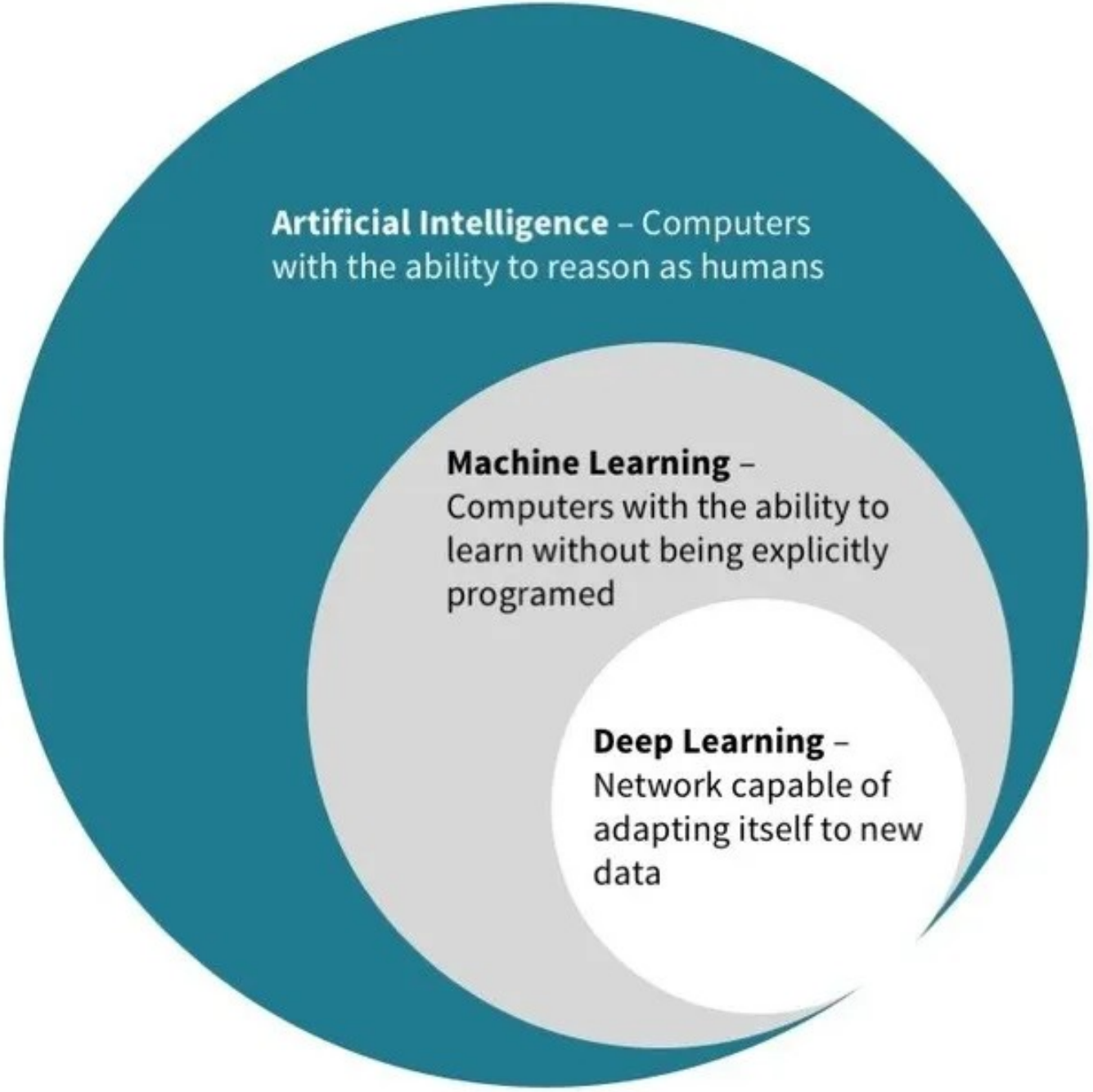
What is Machine Learning?



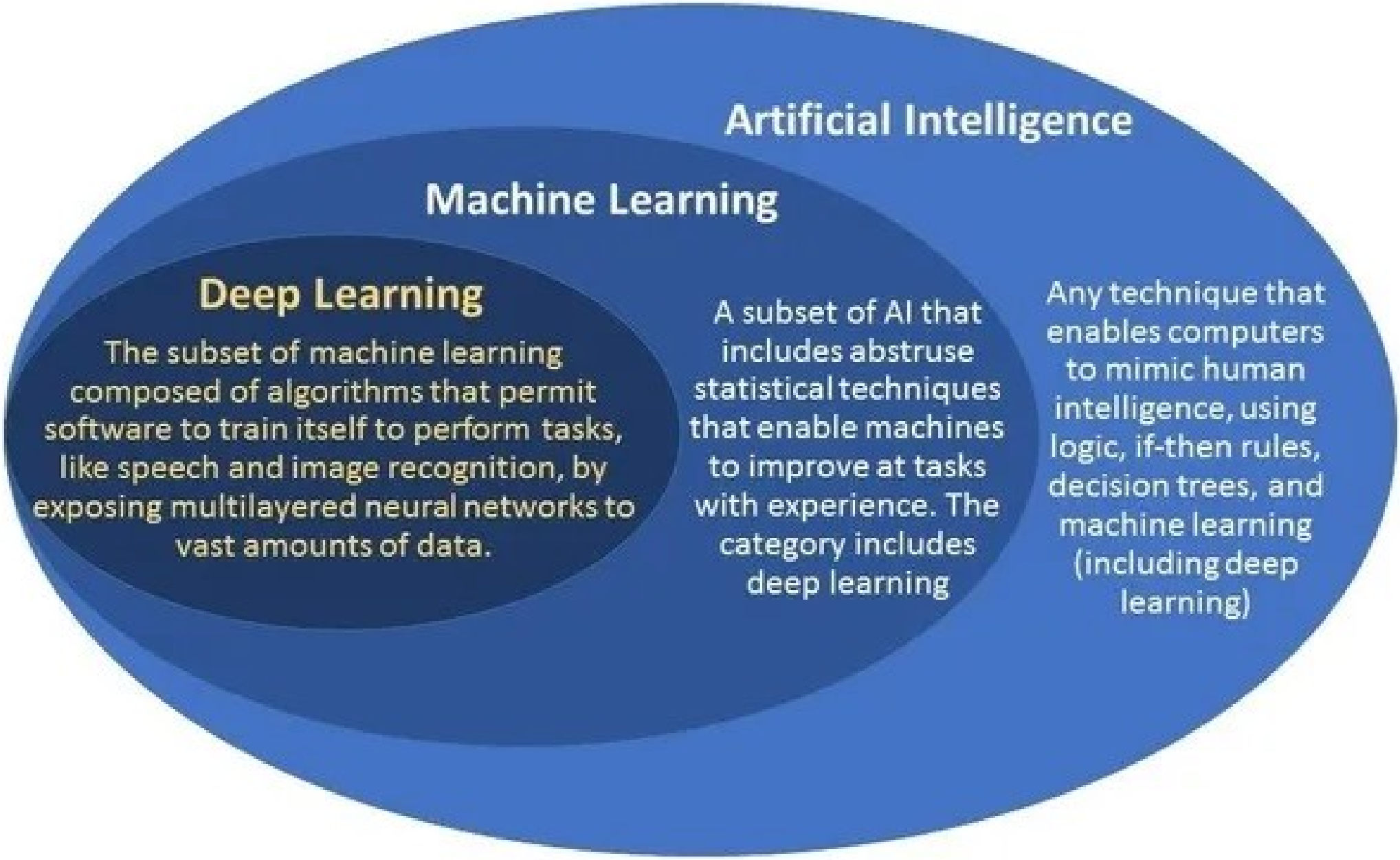
What is Machine Learning?



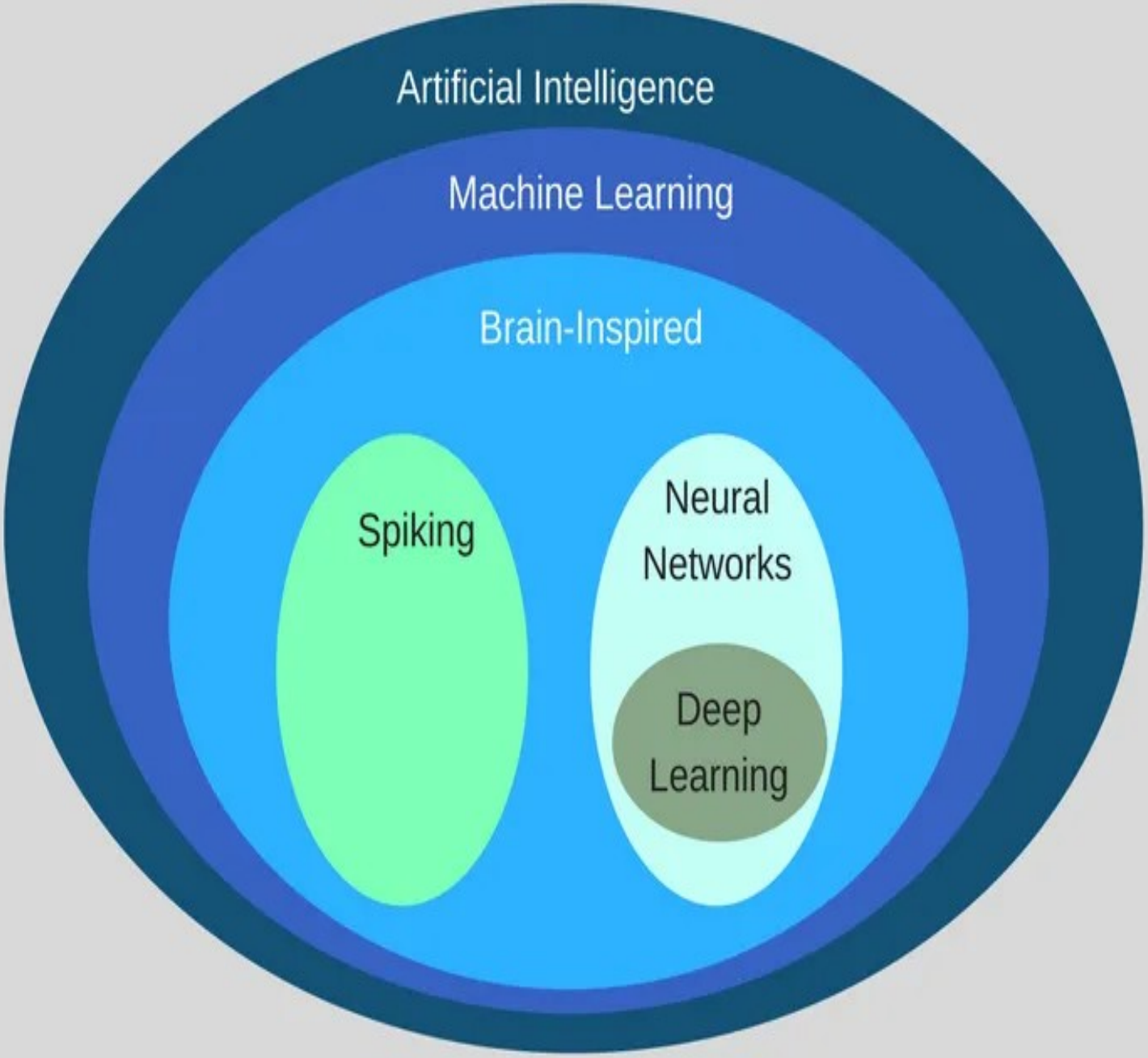
What is Machine Learning?



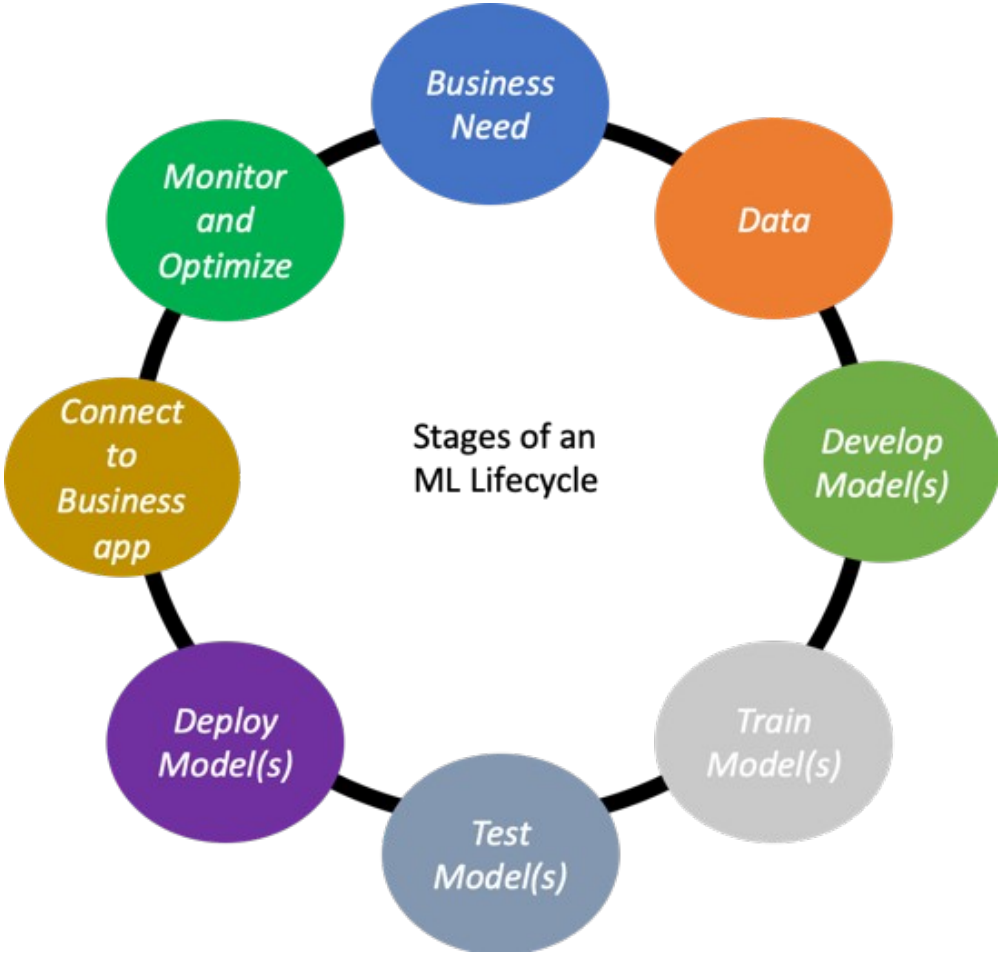
What is Machine Learning?



What is Machine Learning?



ML flow?


Business Need
Data
Develop Model(s)
Train Model(s)
Test Model(s)
Deploy Model(s)
Connect to Business app
Monitor and Optimize
Stages of an ML Lifecycle


STEP 01
Data collection
STEP 02
Data preparation
STEP 03
Choosing a model
STEP 04
Training
STEP 05
Evaluation
STEP 06
Parameter tuning
STEP 07
Prediction

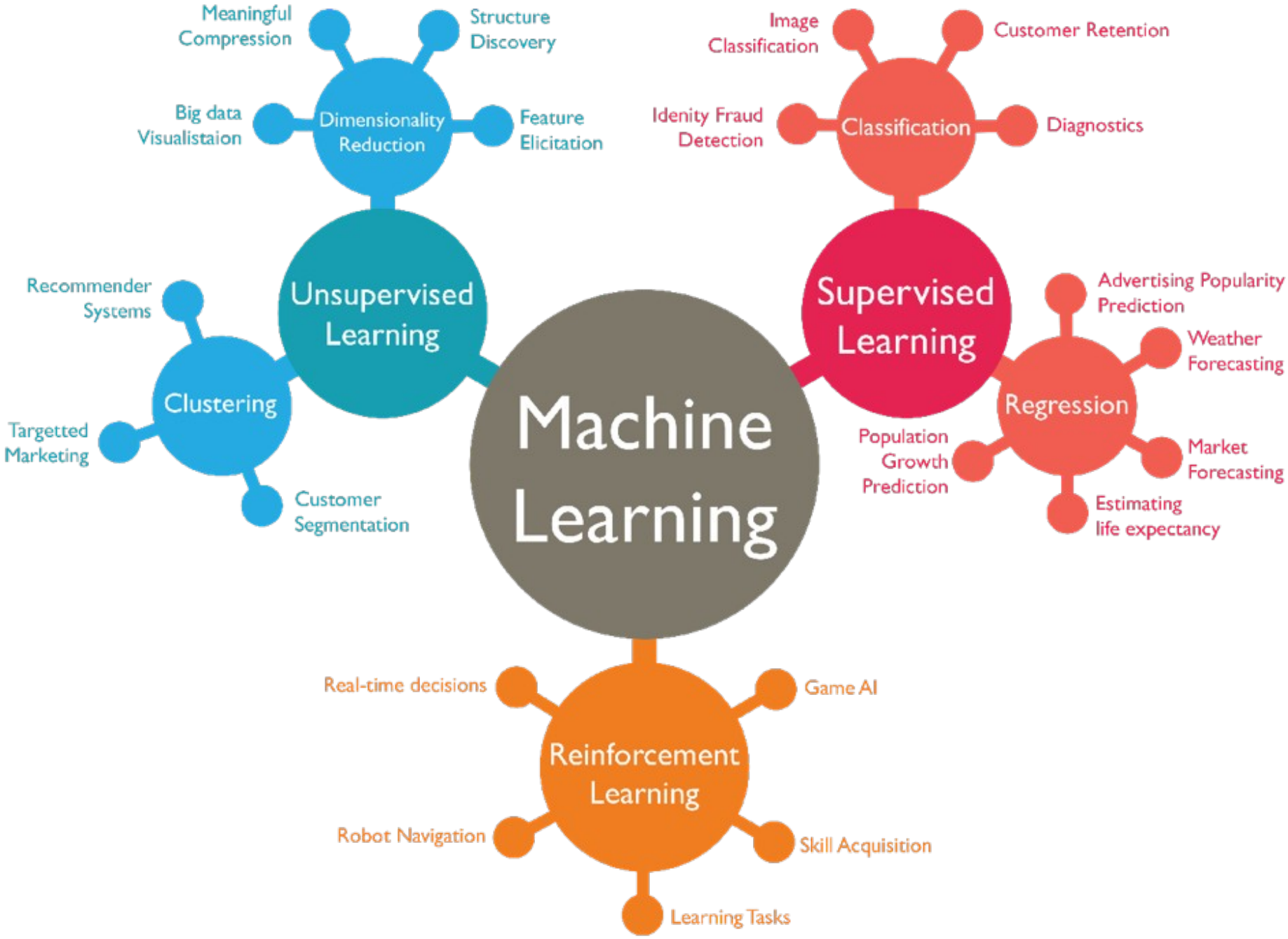
ML algorithms and models?

- An “algorithm” in machine learning is a procedure that run on data to create a machine learning “model.”

A Brief Taxonomy of ML Models

ML Model Type	Uses Cases
Linear regression/classification	Patterns in numeric data, such as financial spreadsheets
Graphic models	Fraud detection or sentiment awareness
Decision trees/Random forests	Predicting outcomes
Deep learning neural networks	Computer vision, natural language processing and more

ML algorithms and models?



ML tools?

NumPy

pandas

scikit learn

PyTorch

matplotlib

Keras
A deep learning library

Google colab

jupyter

TensorFlow

DeepUFERSA?

- Contribute to the Artificial Intelligence Field;
- Engage EE/CS Students;
- R&D (both Pure and Applied);

  - Extension?

  - More?

DeepUfersa

TinyML

TinyML

- Is TinyML the Embedded Machine Learning?

TinyML

- Is TinyML the Embedded Machine Learning?

  Yes and No!

TinyML

- Is TinyML the Embedded Machine Learning?

  Yes and No!

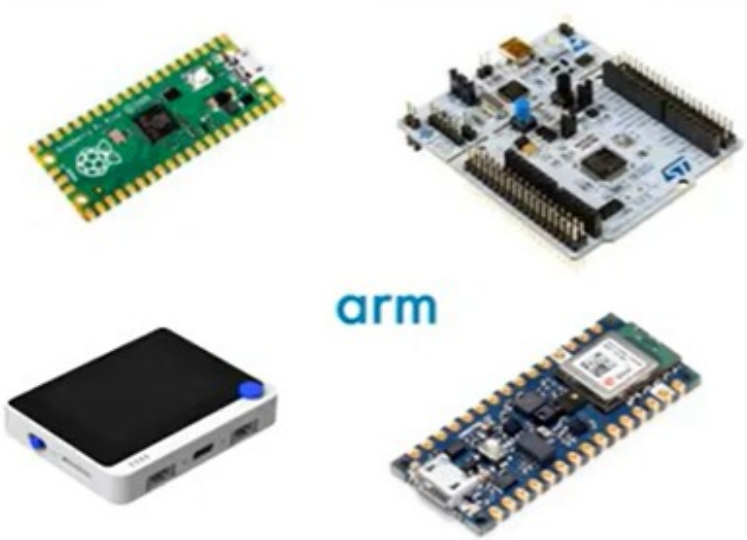
- Embedded Systems has widely concepts that covers a broader range of devices and applications.

TinyML

- Is TinyML the Embedded Machine Learning?

  Yes and No!

- Embedded Systems has widely concepts that covers a broader range of devices and applications.

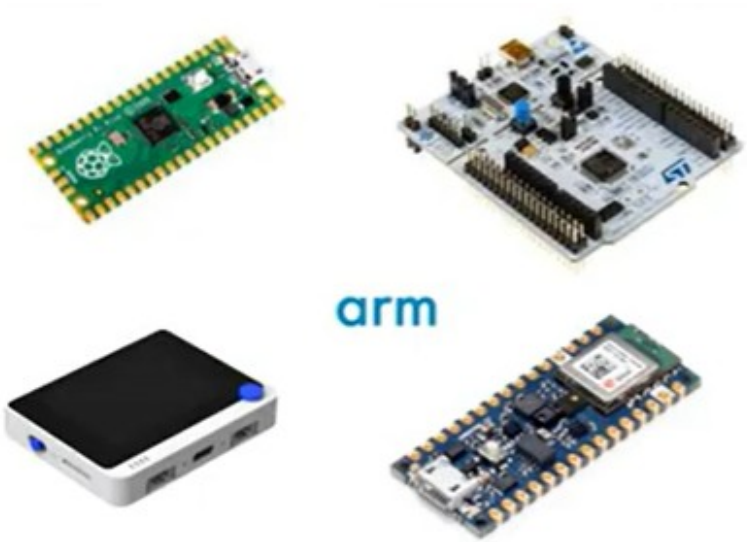


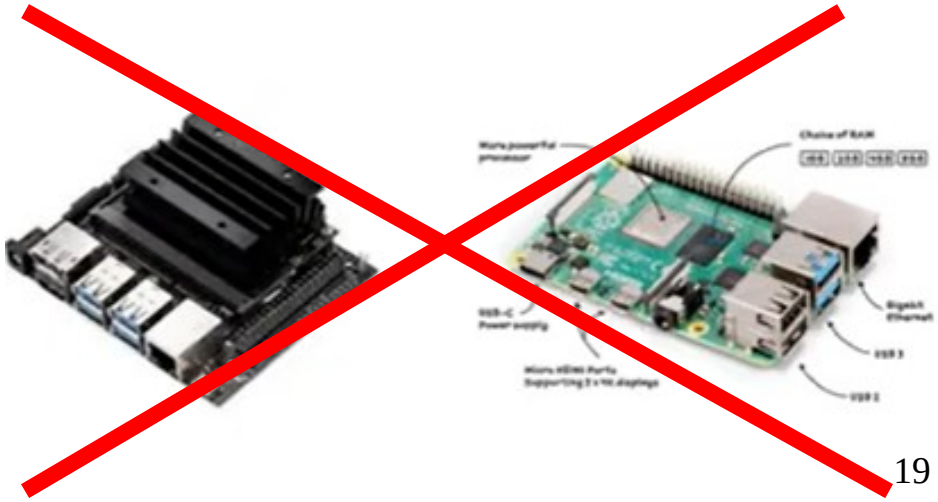
TinyML

- Is TinyML the Embedded Machine Learning?

  Yes and No!

- Embedded Systems has widely concepts that covers a broader range of devices and applications.





TinyML

- Is TinyML the Embedded Machine Learning?

  Yes and No!

- Embedded Systems has widely concepts that covers a broader range of devices and applications.

TinyML

- Is TinyML the Embedded Machine Learning?

concepts that

d applications.

TinyML

- Is TinyML the Embedded Machine Learning?

Yes and No!

concepts that

d applications.

Internet of Things (IoT)

TinyML

- Is TinyML the Embedded Machine Learning?

  Yes and No!

- Embedded Systems has widely concepts that covers a broader range of devices and applications.

TinyML covers specific areas in Embedded Systems as in resource constrained edge devices.

TinyML

- Is TinyML the Embedded Machine Learning?

  Yes and No!

- Embedded Systems has widely concepts that covers a broader range of devices and applications.

TinyML covers specific areas in Embedded Systems as in resource constrained edge devices.

IoT (provides data)

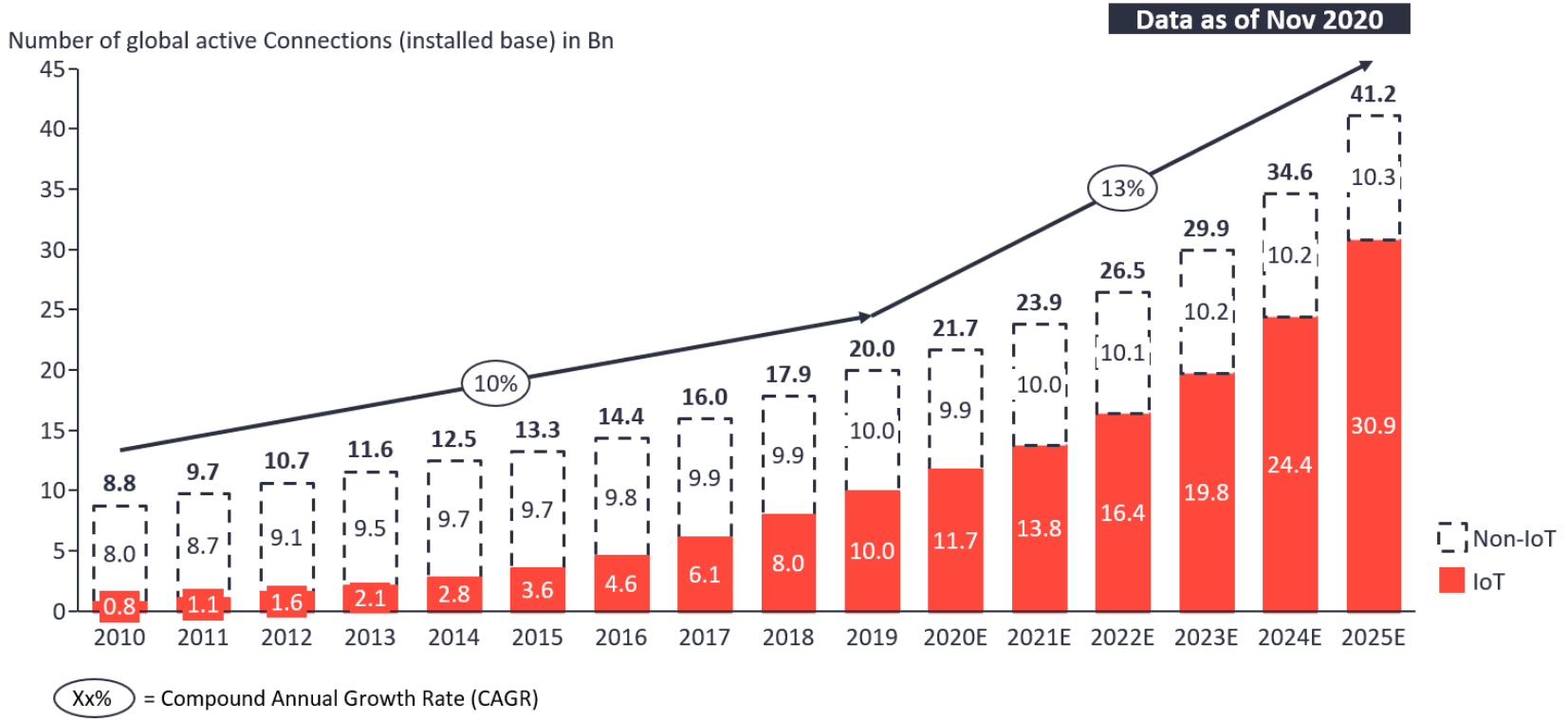
The future of Machine Learning is Tiny!

IOT ANALYTICS

Insights that empower you to understand IoT markets

Total number of device connections (incl. Non-IoT)

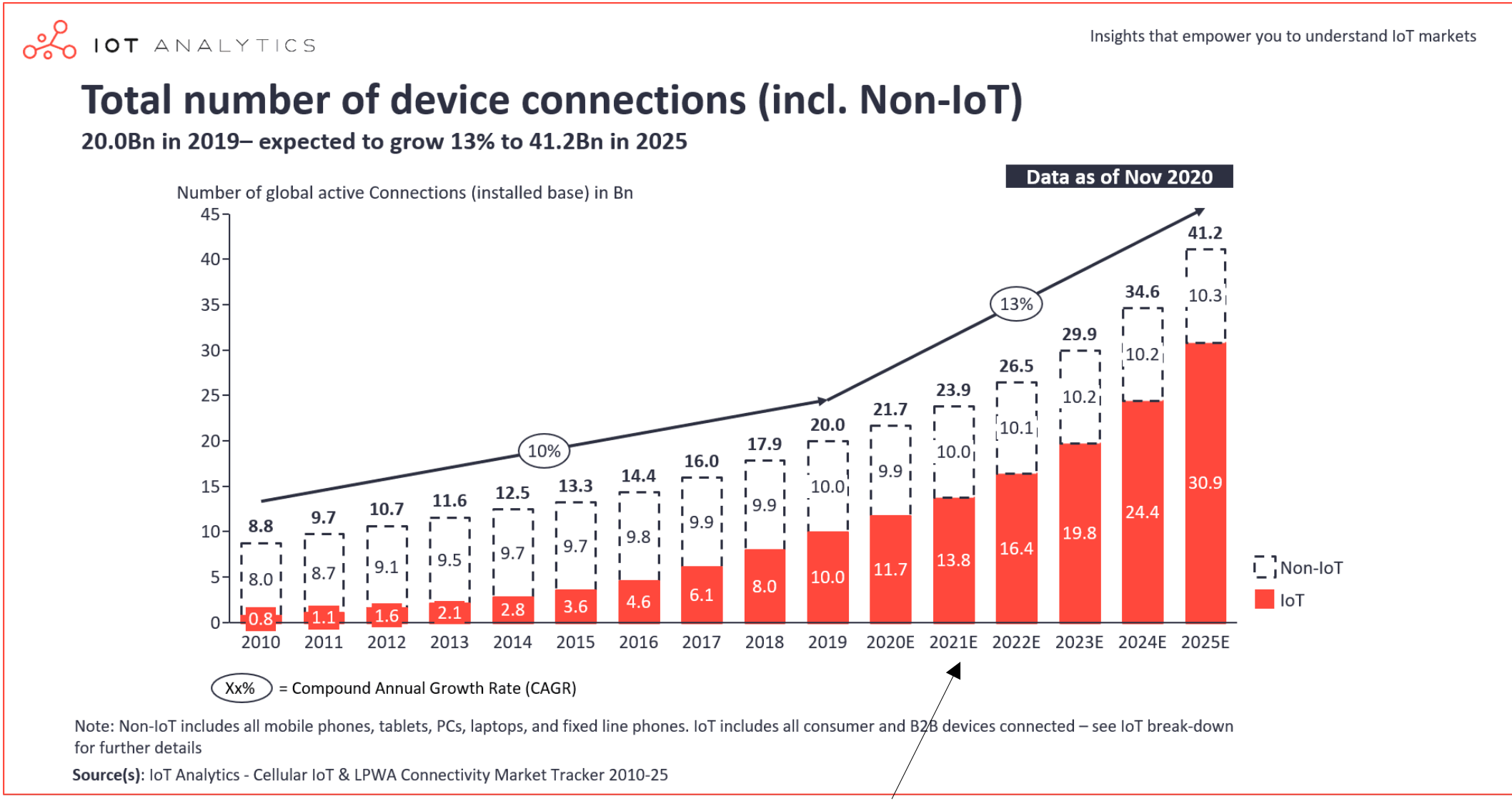
20.0Bn in 2019– expected to grow 13% to 41.2Bn in 2025



Note: Non-IoT includes all mobile phones, tablets, PCs, laptops, and fixed line phones. IoT includes all consumer and B2B devices connected – see IoT break-down for further details

Source(s): IoT Analytics - Cellular IoT & LPWA Connectivity Market Tracker 2010-25

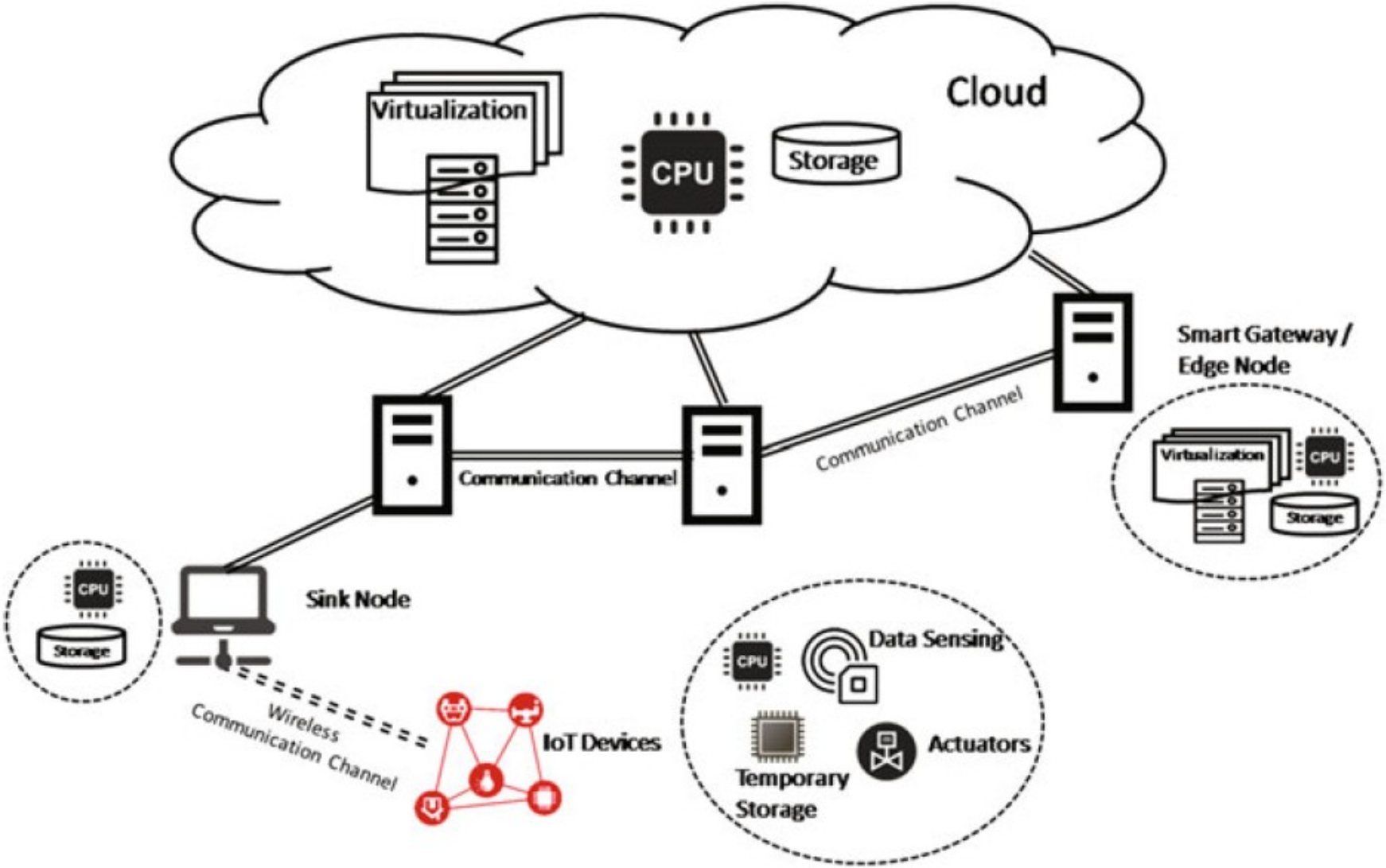
The future of Machine Learning is Tiny!



Resource constrained edge devices

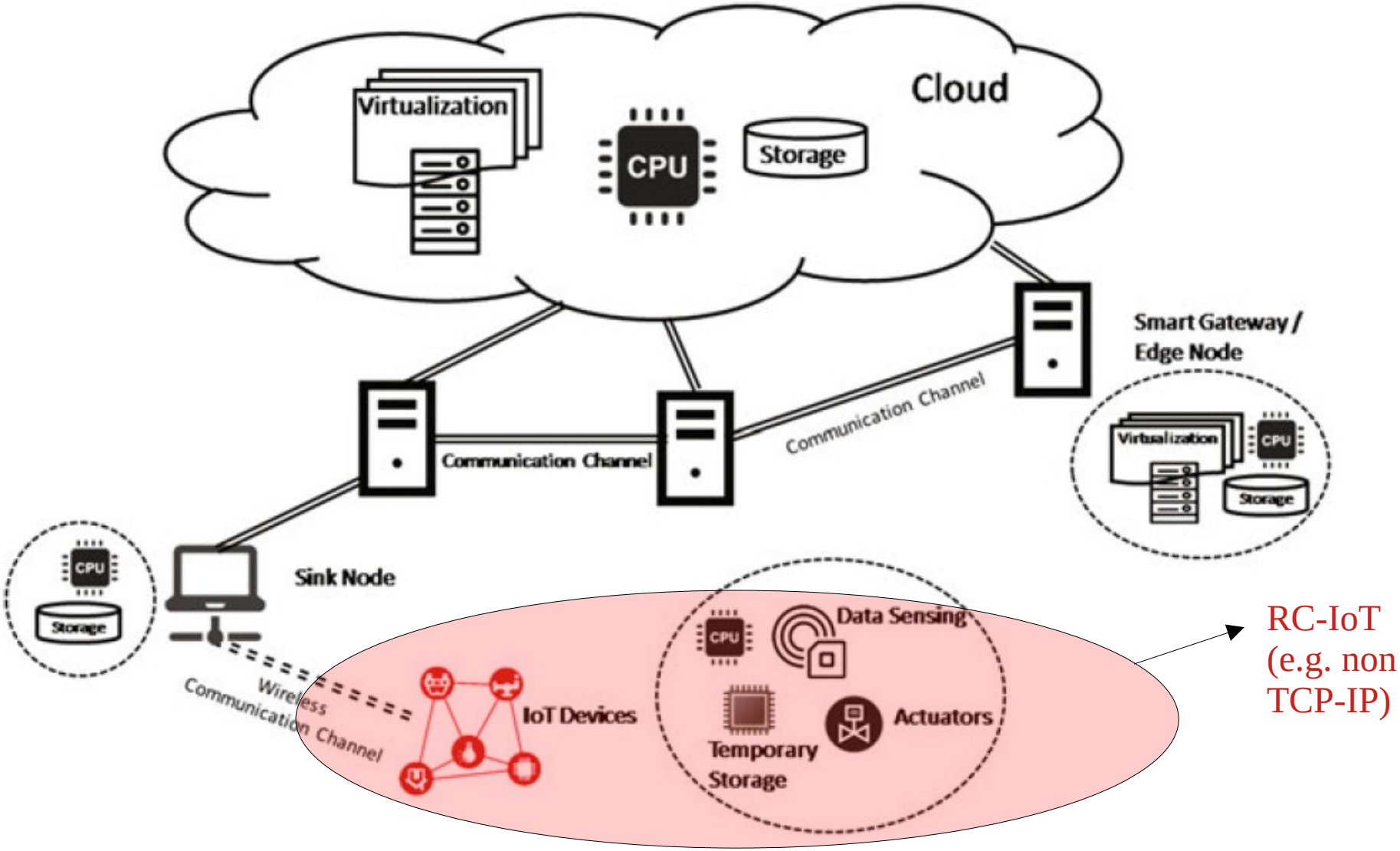
TinyML resources

Resource-rich vs resource-constrained



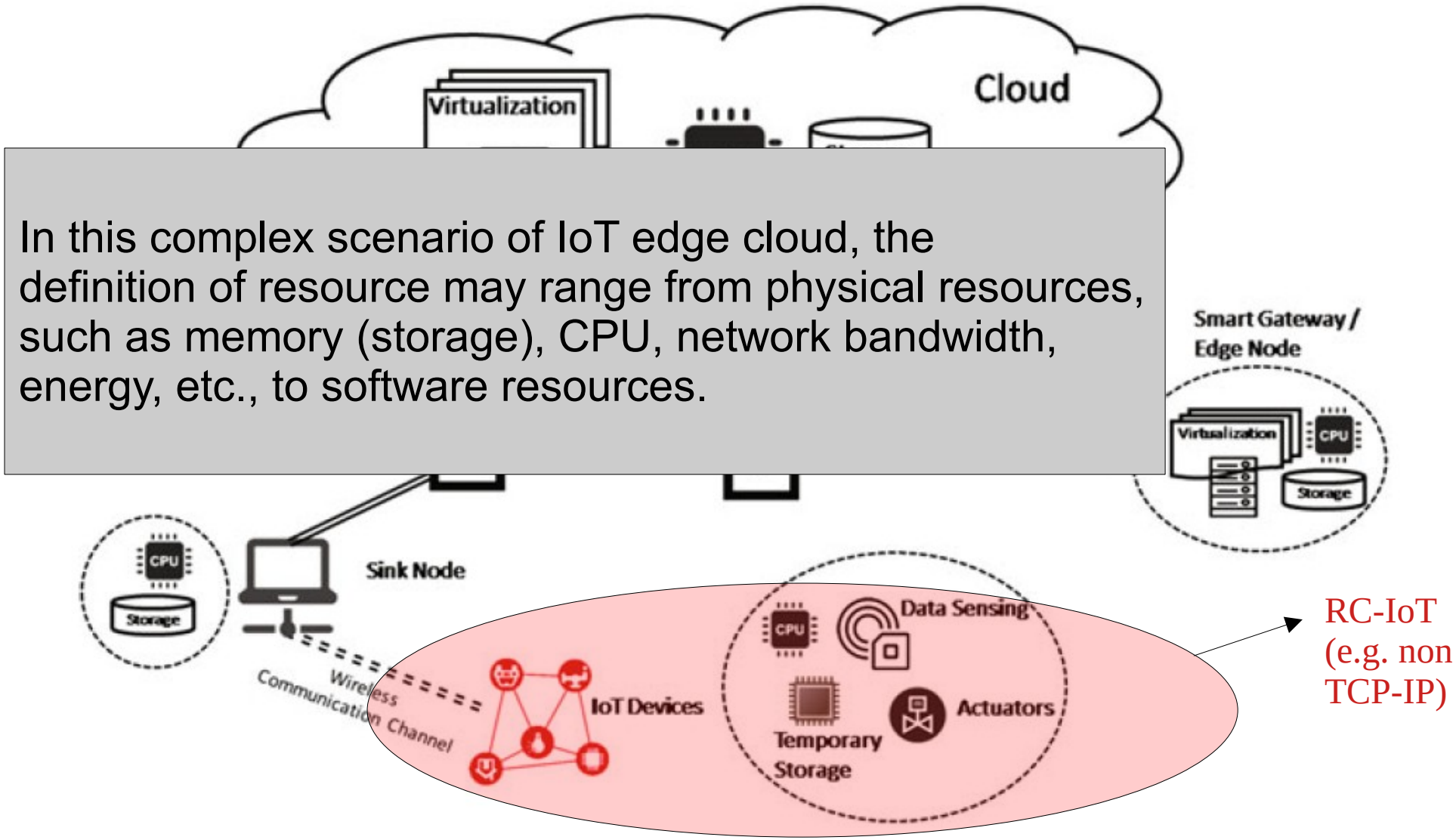
TinyML resources

Resource-rich vs resource-constrained



TinyML resources

Resource-rich vs resource-constrained

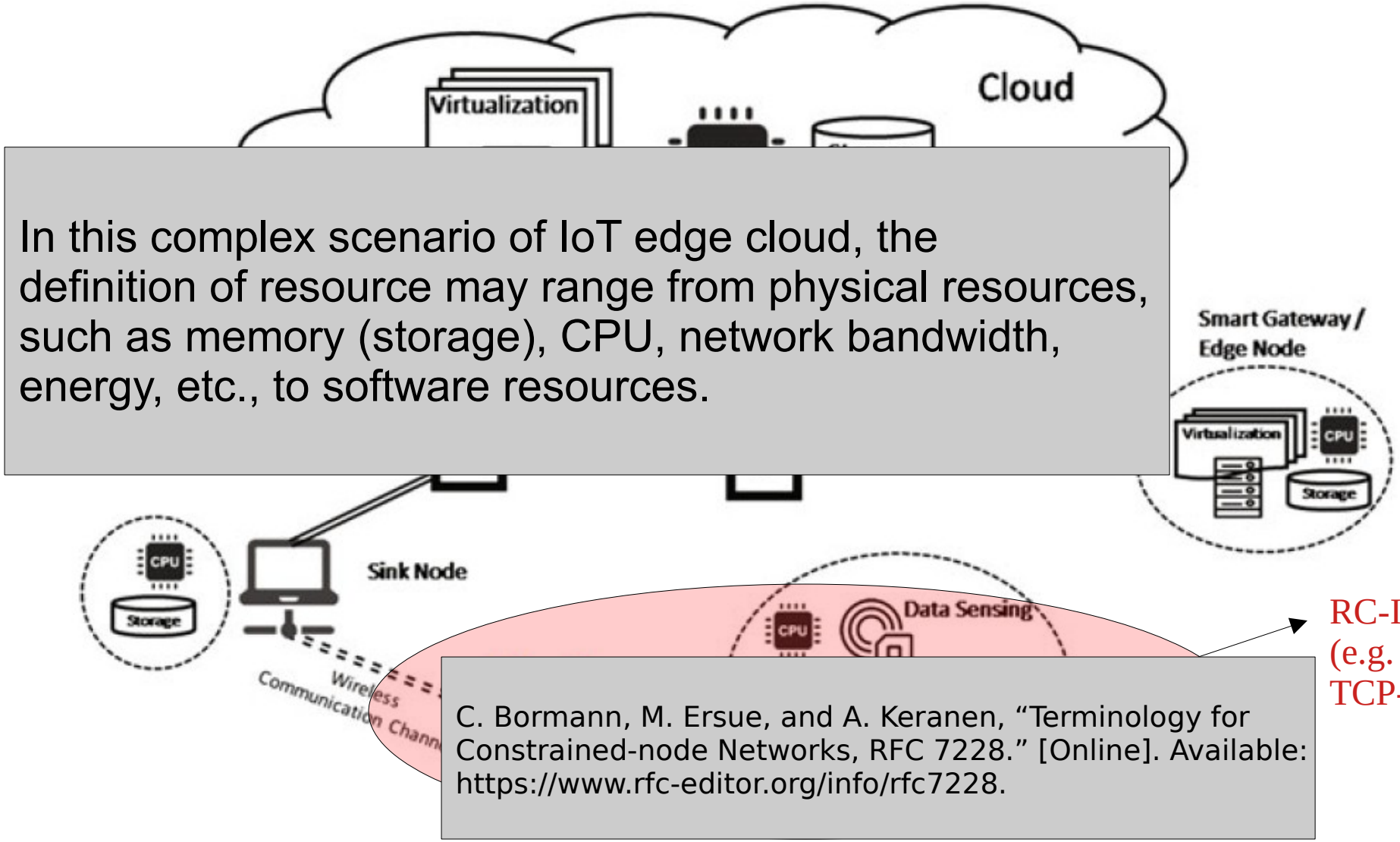


In this complex scenario of IoT edge cloud, the definition of resource may range from physical resources, such as memory (storage), CPU, network bandwidth, energy, etc., to software resources.

RC-IoT (e.g. non TCP-IP)

TinyML resources

Resource-rich vs resource-constrained



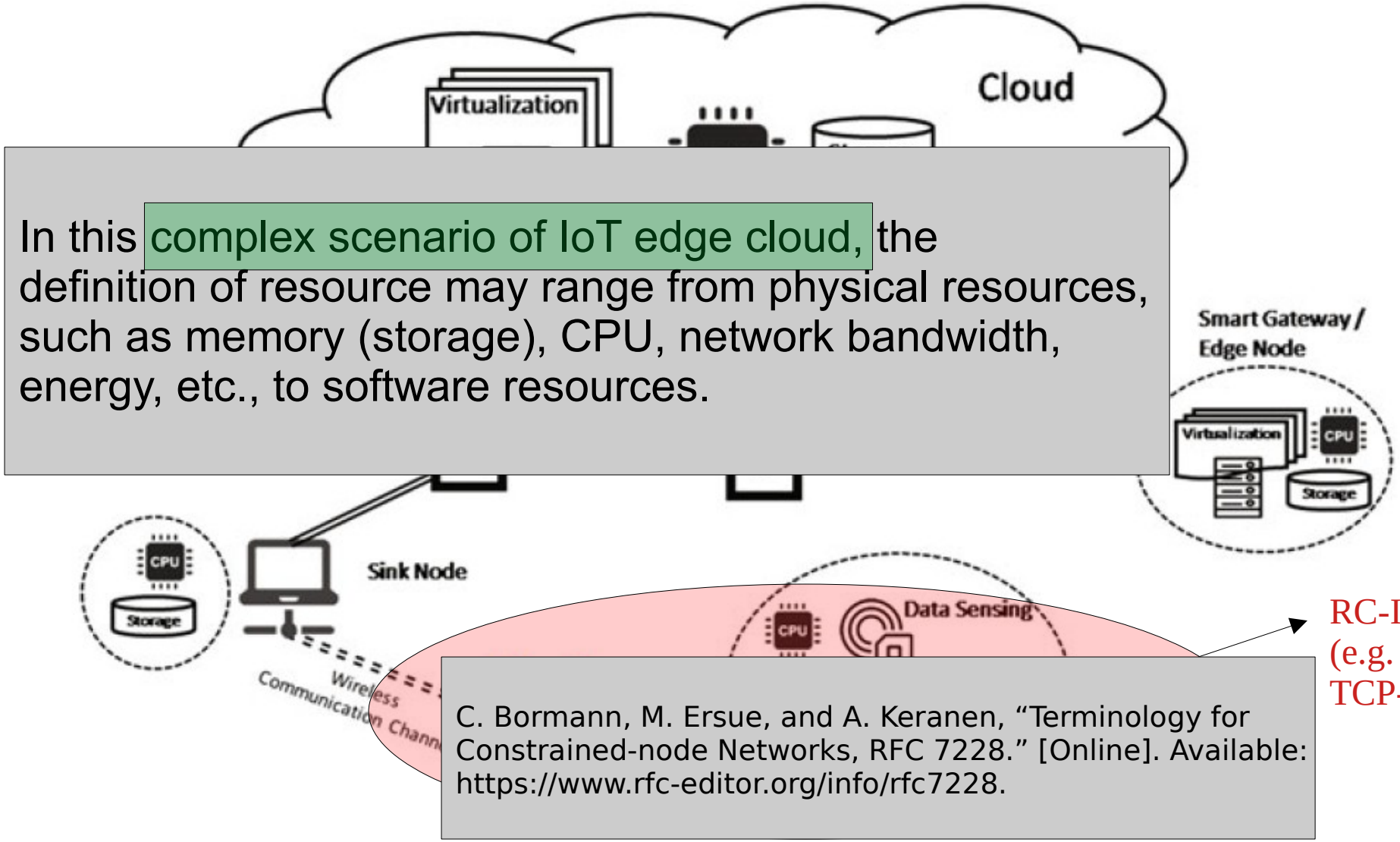
In this complex scenario of IoT edge cloud, the definition of resource may range from physical resources, such as memory (storage), CPU, network bandwidth, energy, etc., to software resources.

RC-IoT (e.g. non TCP-IP)

C. Bormann, M. Ersue, and A. Keranen, “Terminology for Constrained-node Networks, RFC 7228.” [Online]. Available: https://www.rfc-editor.org/info/rfc7228.

TinyML resources

Resource-rich vs resource-constrained



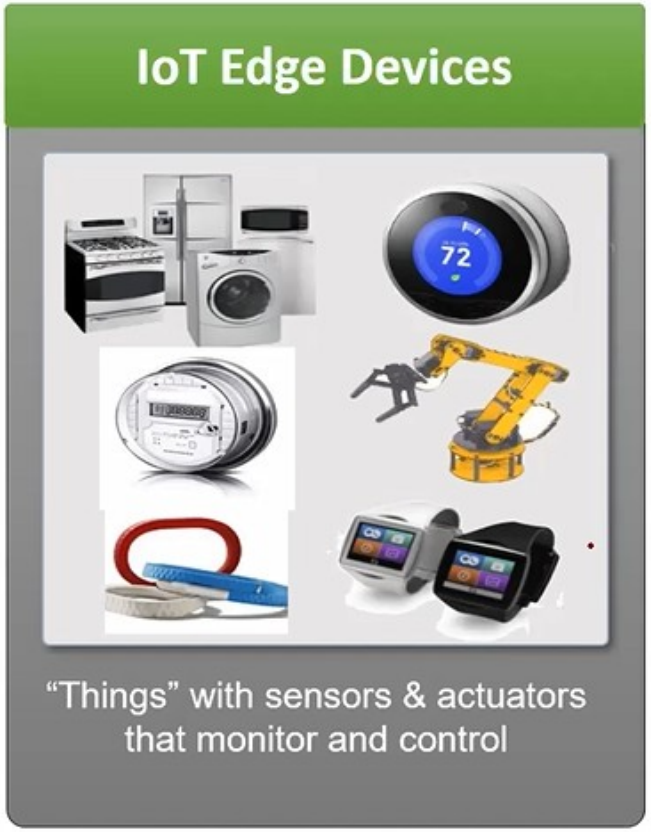
In this complex scenario of IoT edge cloud, the definition of resource may range from physical resources, such as memory (storage), CPU, network bandwidth, energy, etc., to software resources.

RC-IoT (e.g. non TCP-IP)

C. Bormann, M. Ersue, and A. Keranen, “Terminology for Constrained-node Networks, RFC 7228.” [Online]. Available: https://www.rfc-editor.org/info/rfc7228.

Edge Computing

IoT brings new computing/comm. paradigms



Edge Computing

IoT brings new computing/comm. paradigms

Where to compute?

IoT Edge Devices

"Things" with sensors & actuators that monitor and control

Aggregation Layers (Hubs/Gateways)

Connectivity & Interfaces to aggregate the edge data to send to the cloud

Remote Processing (Cloud Based)

Applications to analyze the data and offer cloud services

Computing/Communication Paradigm

Communicate or compute? That is the question!

Context-Aware IoT Intelligence

Communicate or compute? That is the question!

Context-Aware Intelligence in Resource-Constrained IoT Nodes: Opportunities and Challenges

Publisher: IEEE Cite This PDF

Baibhab Chatterjee ; Ningyuan Cao ; Arijit Raychowdhury ; Shreyas Sen **All Authors**

Abstract

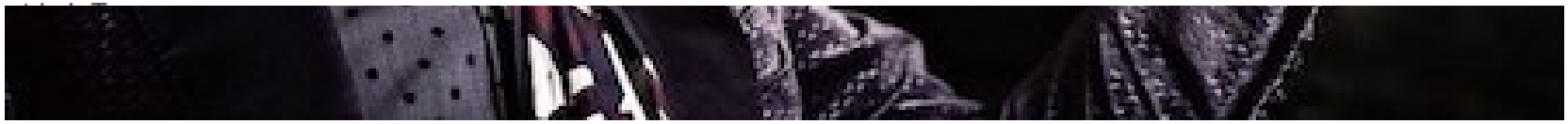
Document Sections

» Background and Motivation

» Challenges in

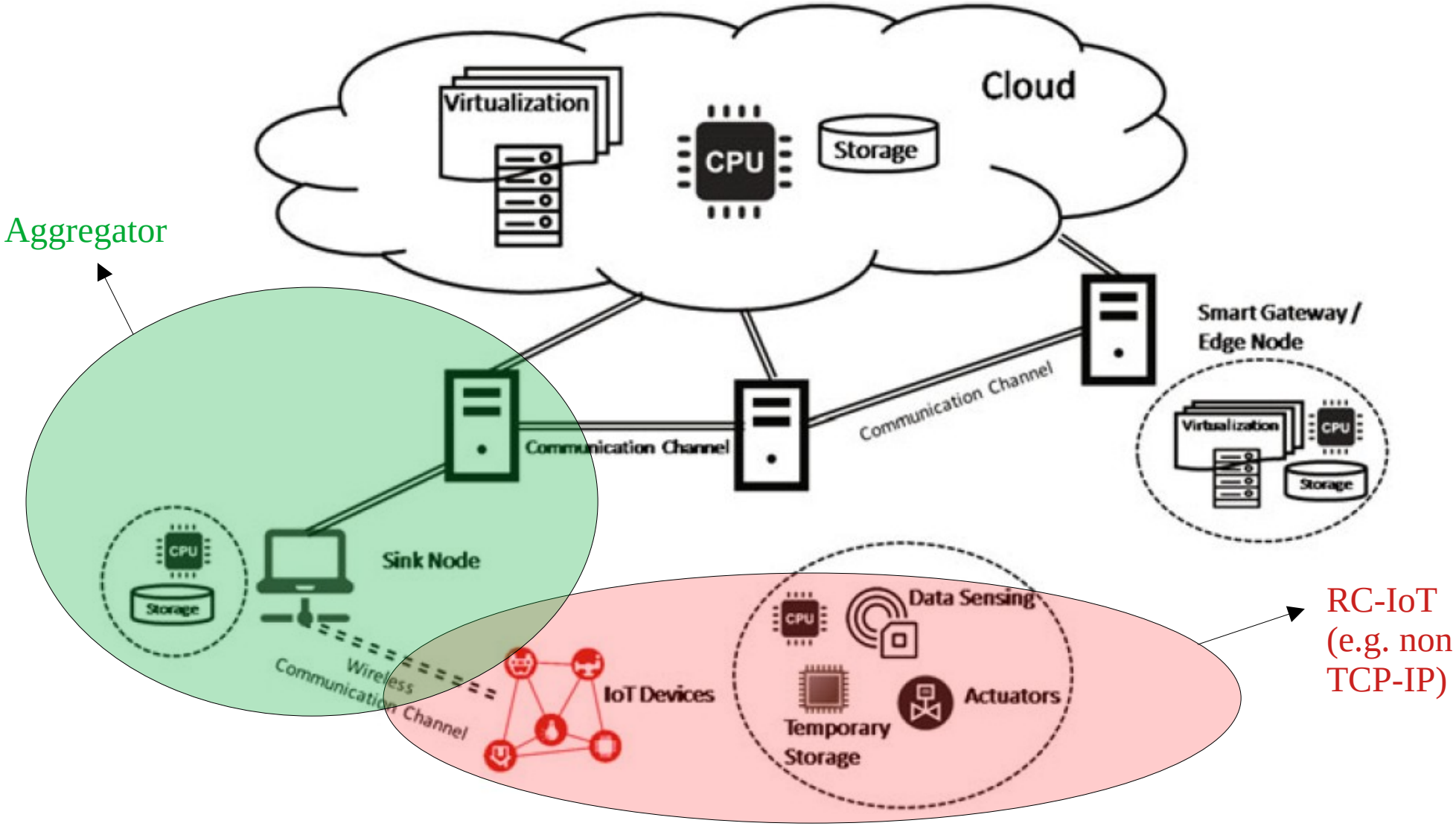
Abstract:
Editor's note: This article provides an academic perspective of the problem, starting with a survey of recent advances in intelligent sensing, computation, communication, and energy management for resource-constrained IoT sensor nodes and leading to a future outlook and needs. -Shreyas Sen, Purdue University.

Published in: IEEE Design & Test (Volume: 36 , Issue: 2, April 2019)

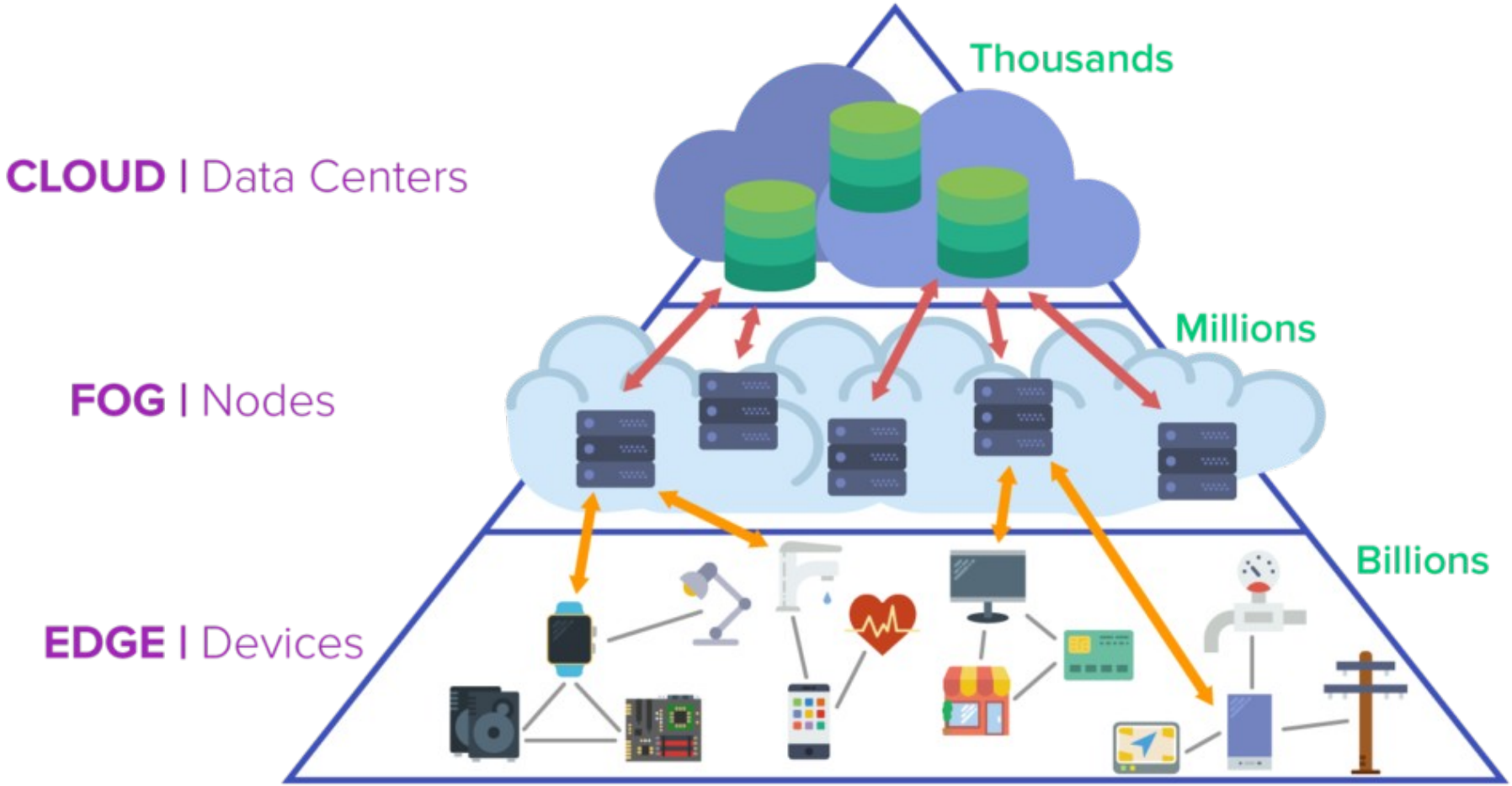


Edge x Fog x Cloud

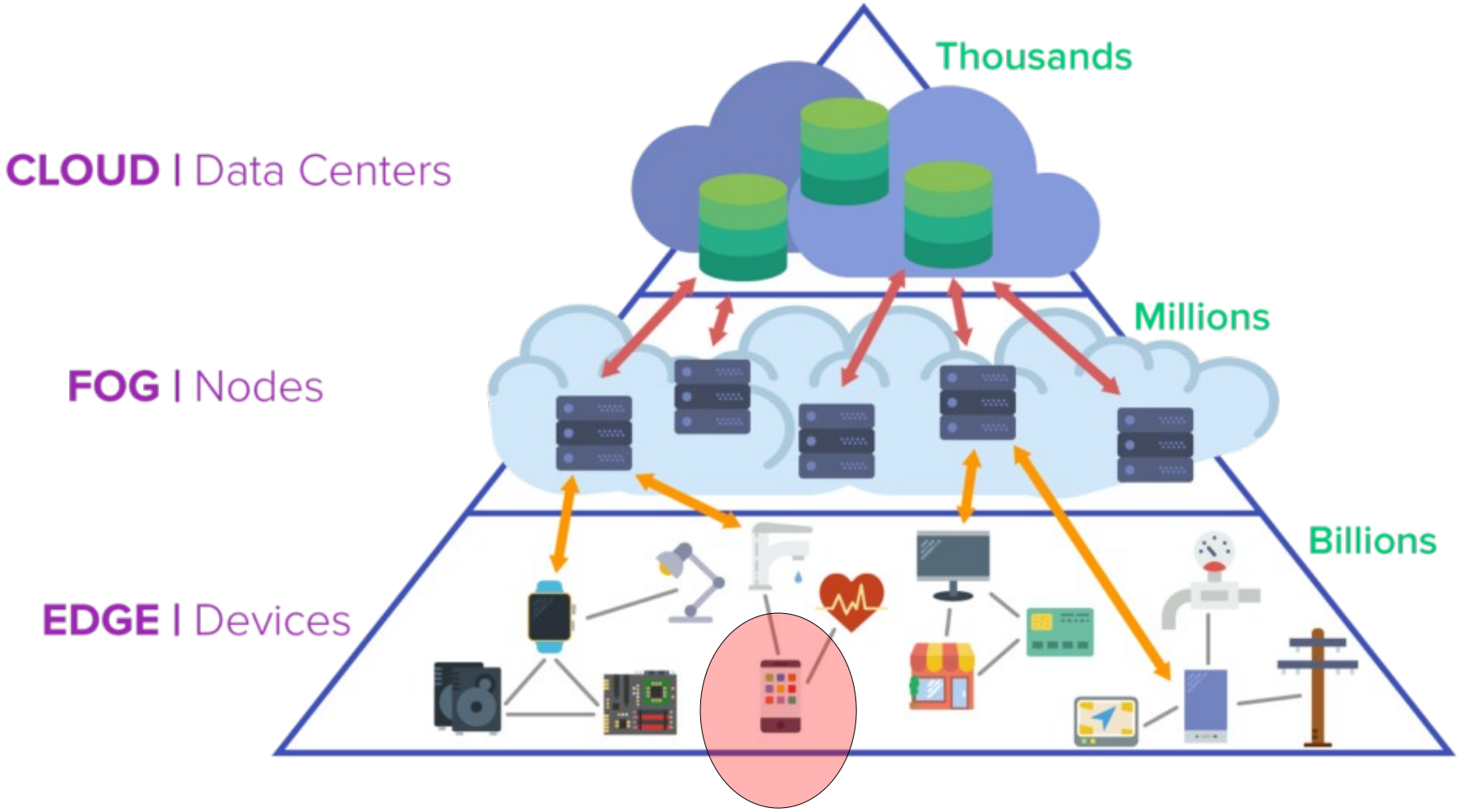
Resource-rich vs resource-constrained



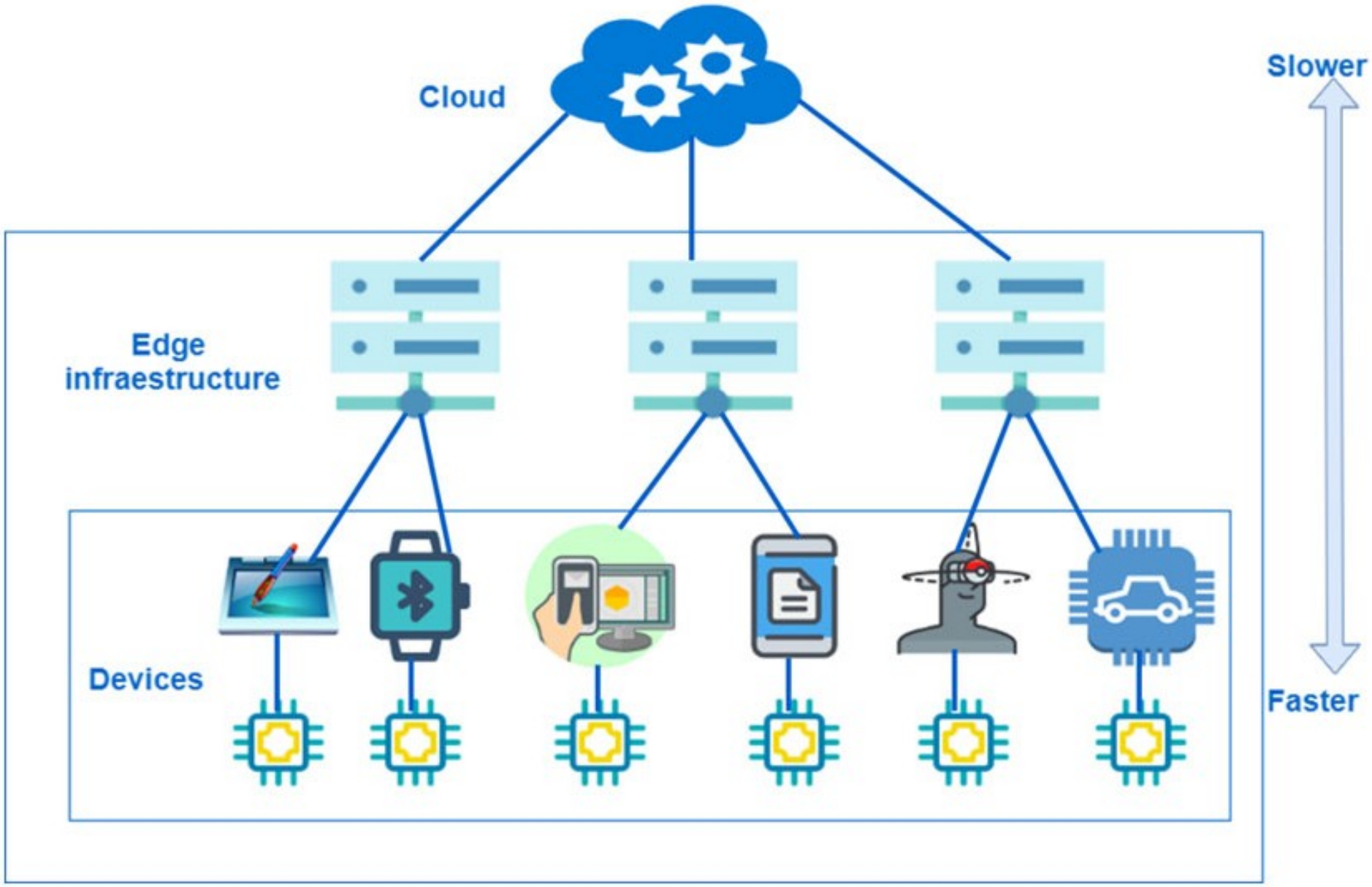
Edge x Fog x Cloud



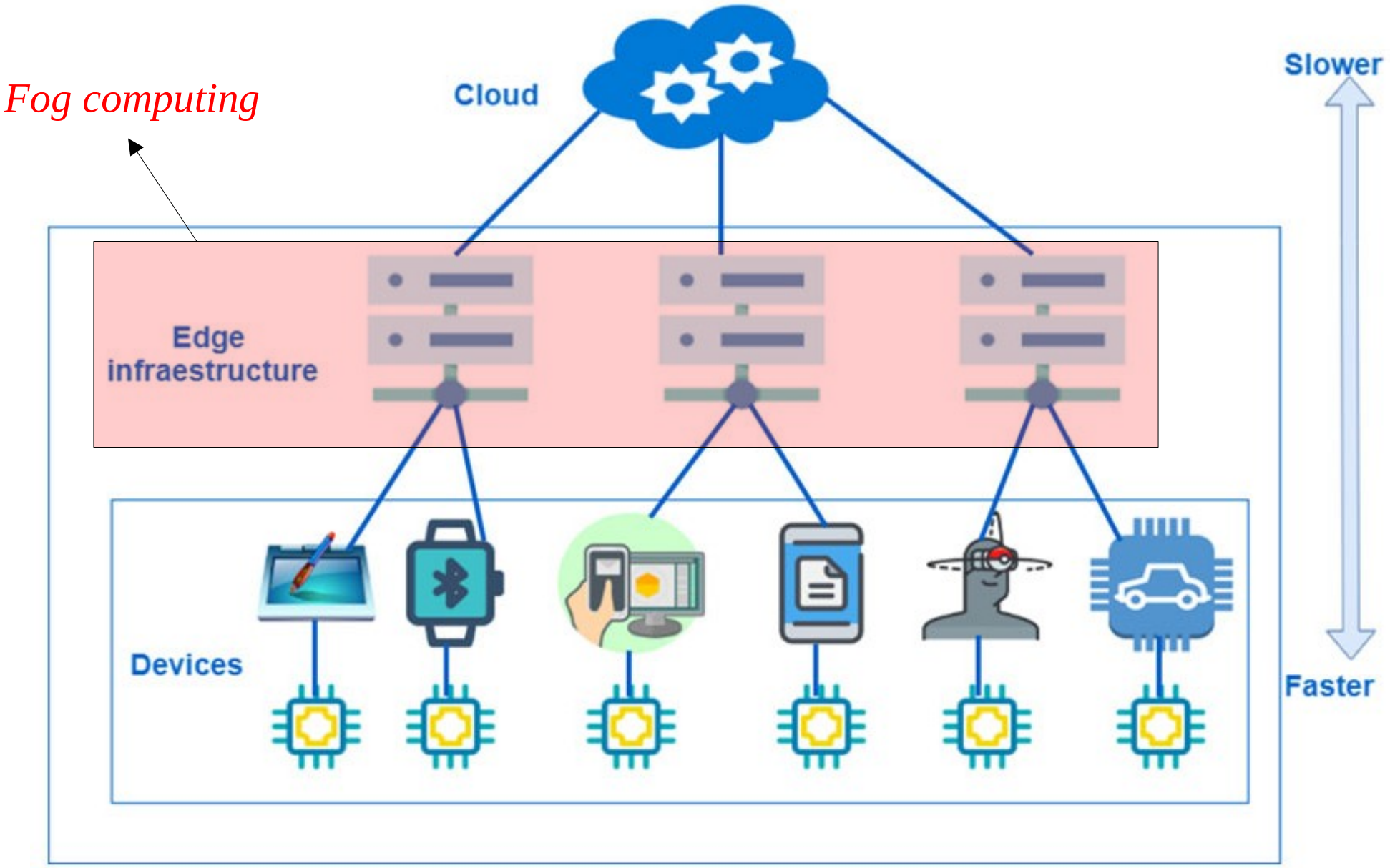
Edge x Fog x Cloud



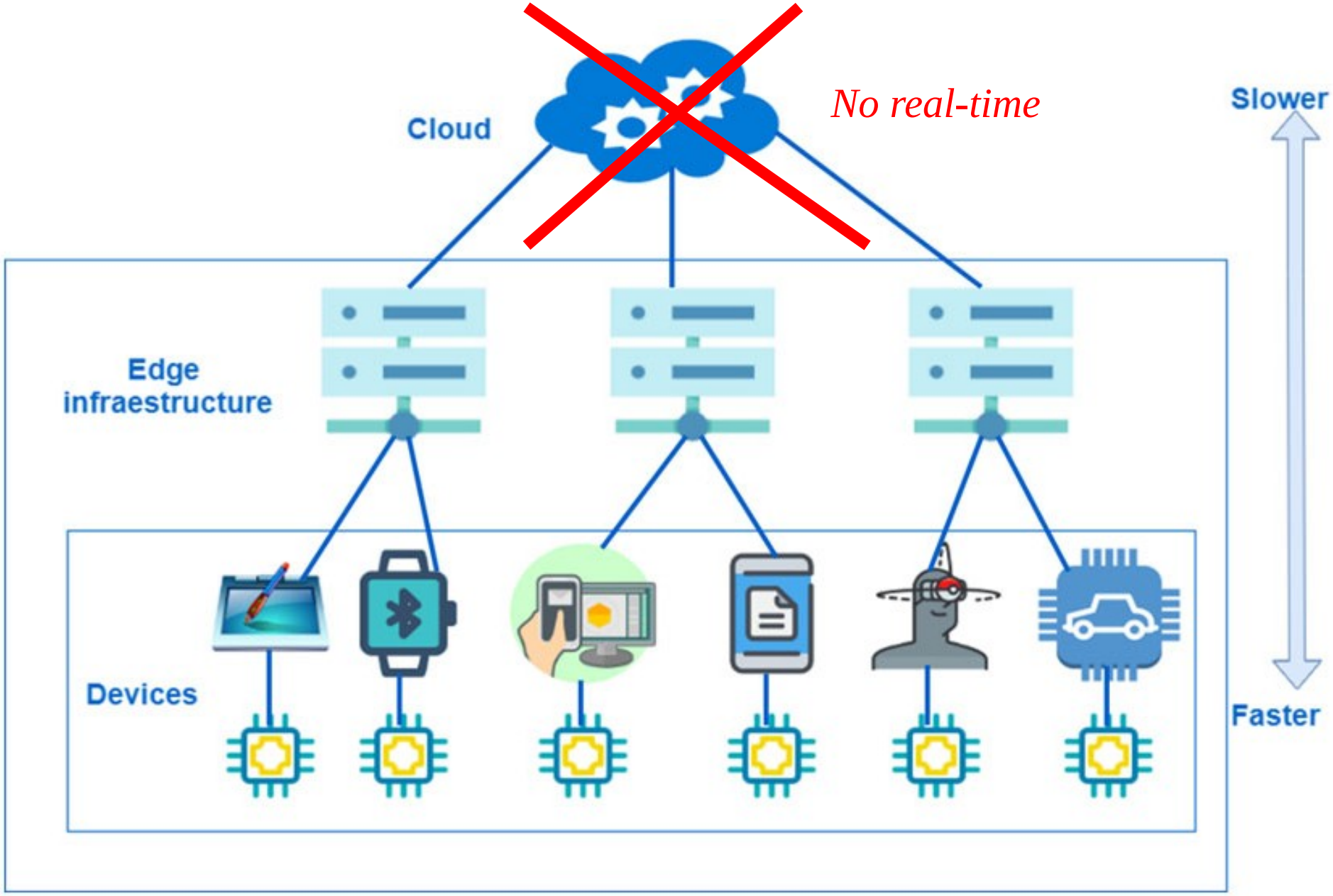
Edge x Fog x Cloud



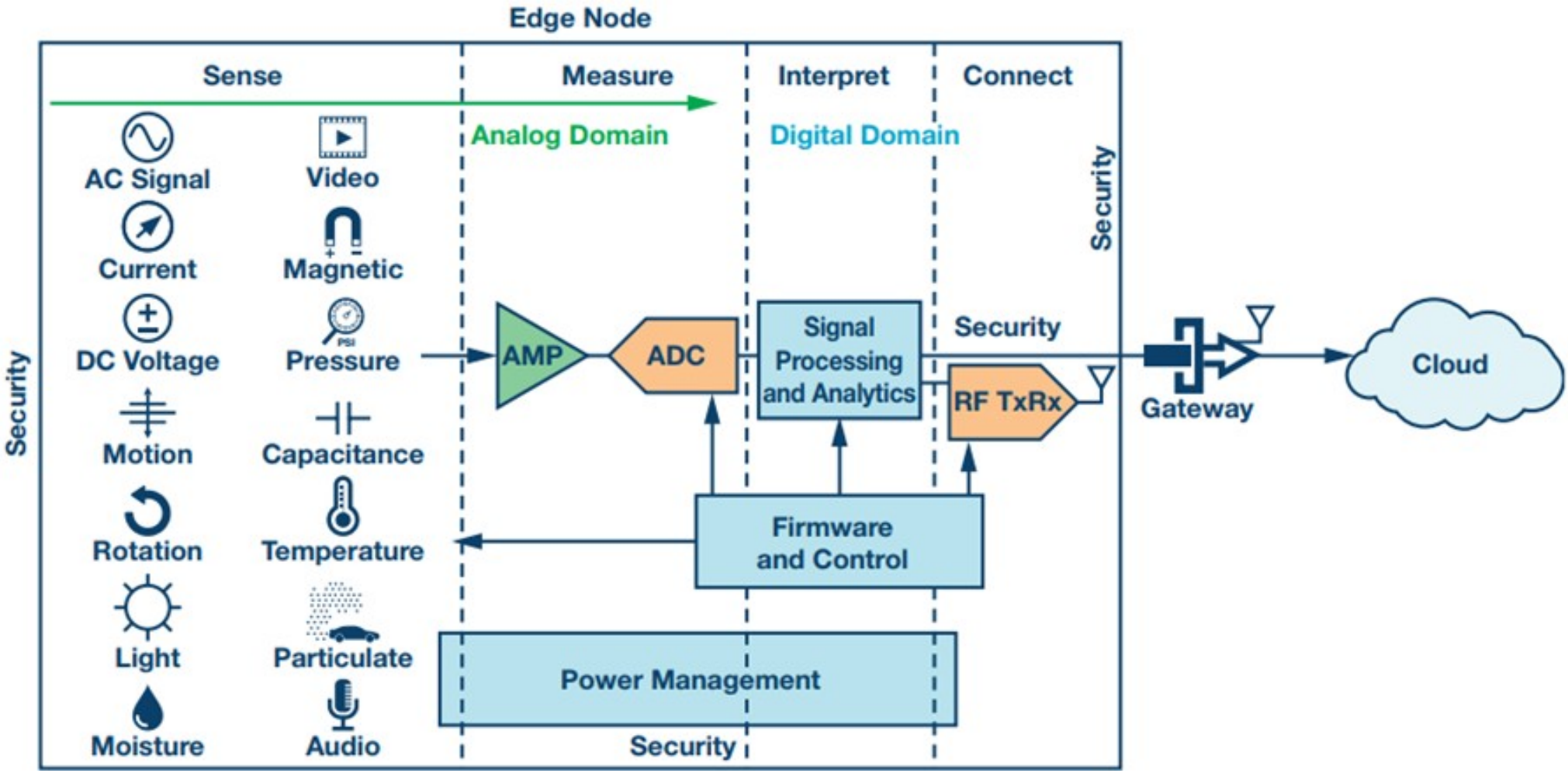
Edge x Fog x Cloud



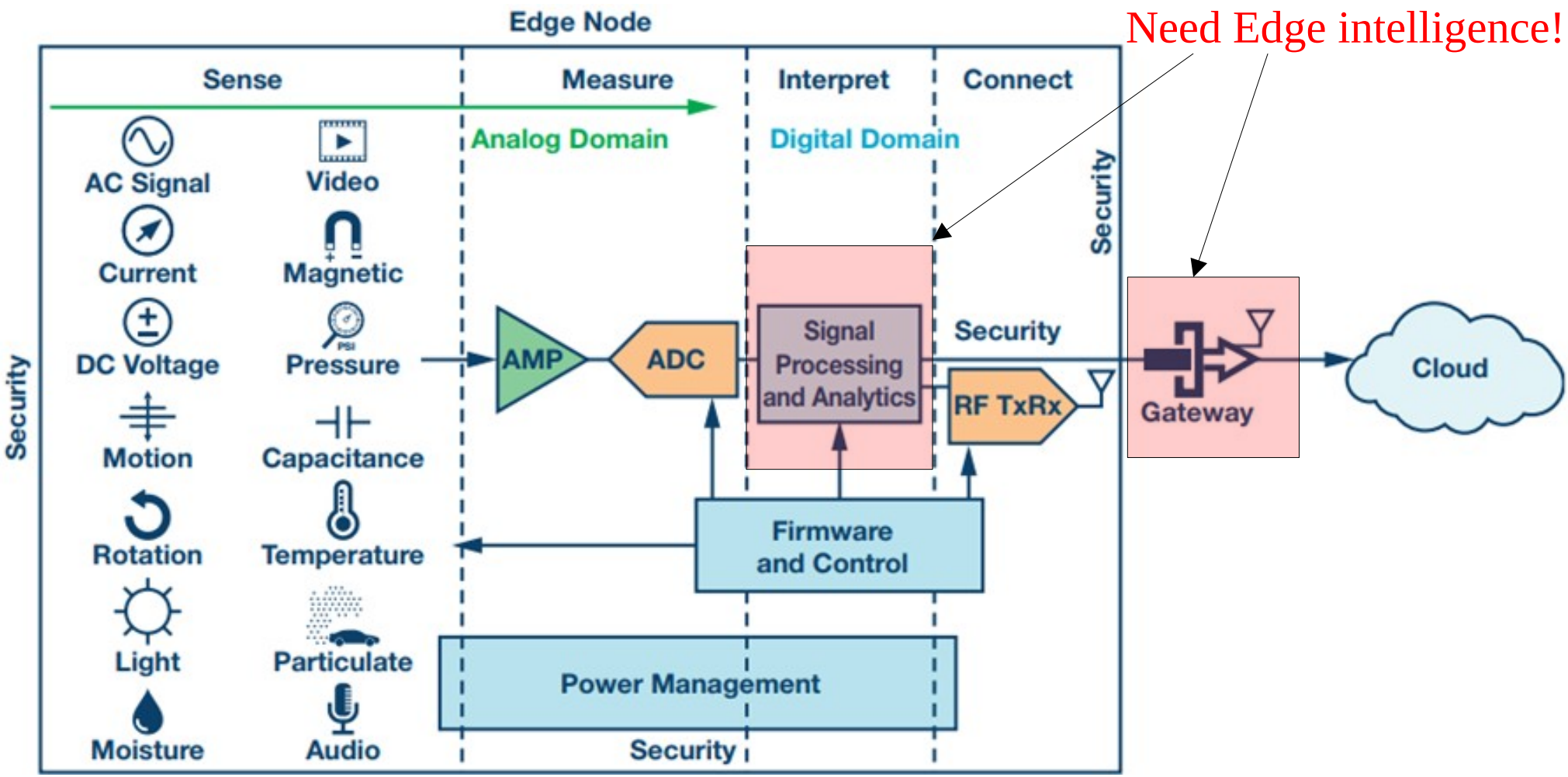
Edge x Fog x Cloud



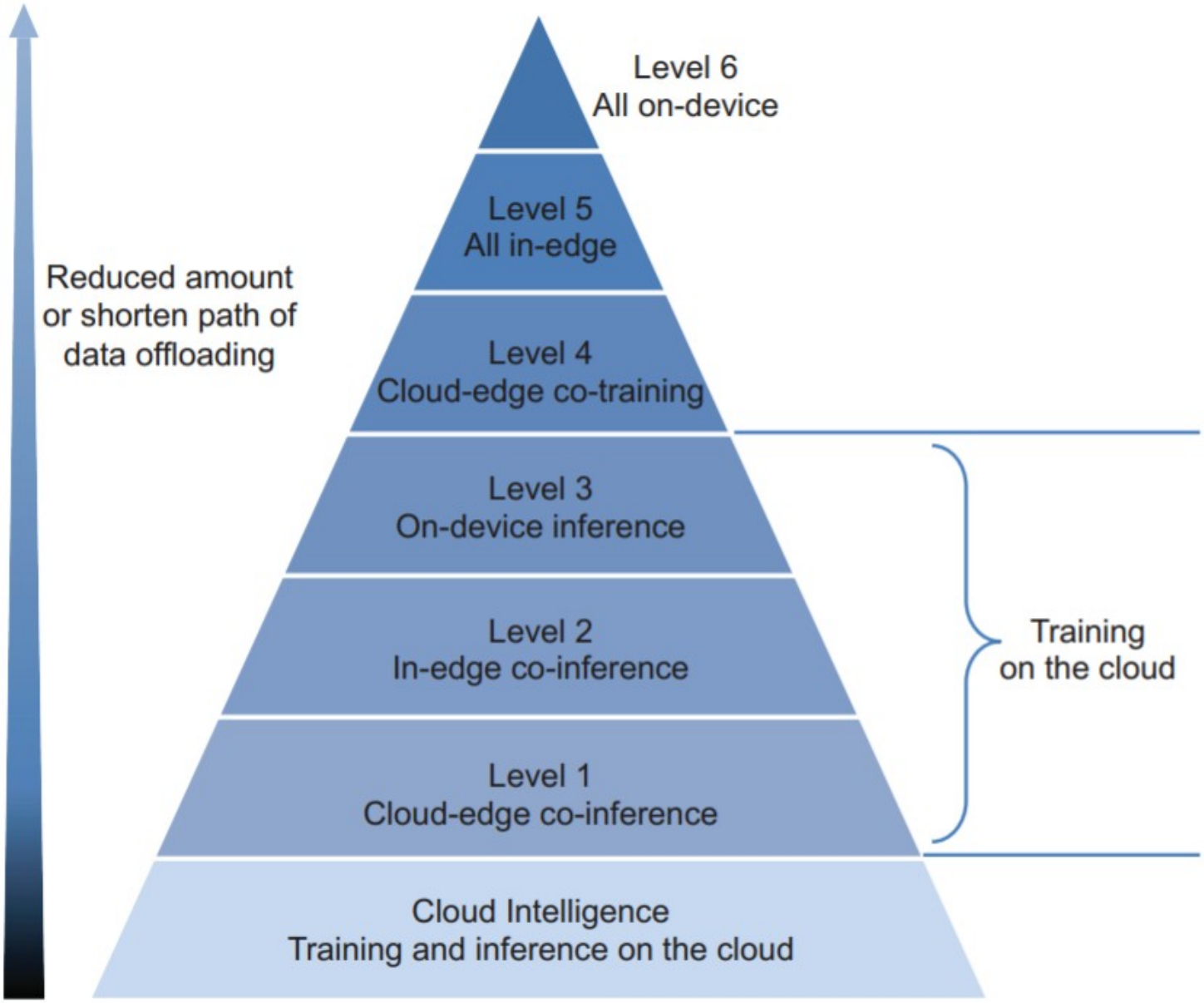
Edge device


Edge Node
Sense
Measure
Interpret
Connect
Analog Domain
Digital Domain
AC Signal
Video
Current
Magnetic
DC Voltage
Pressure
Motion
Capacitance
Rotation
Temperature
Light
Particulate
Moisture
Audio
AMP
ADC
Signal Processing and Analytics
Security
RF TxRx
Gateway
Cloud
Firmware and Control
Power Management
Security
Security
Security

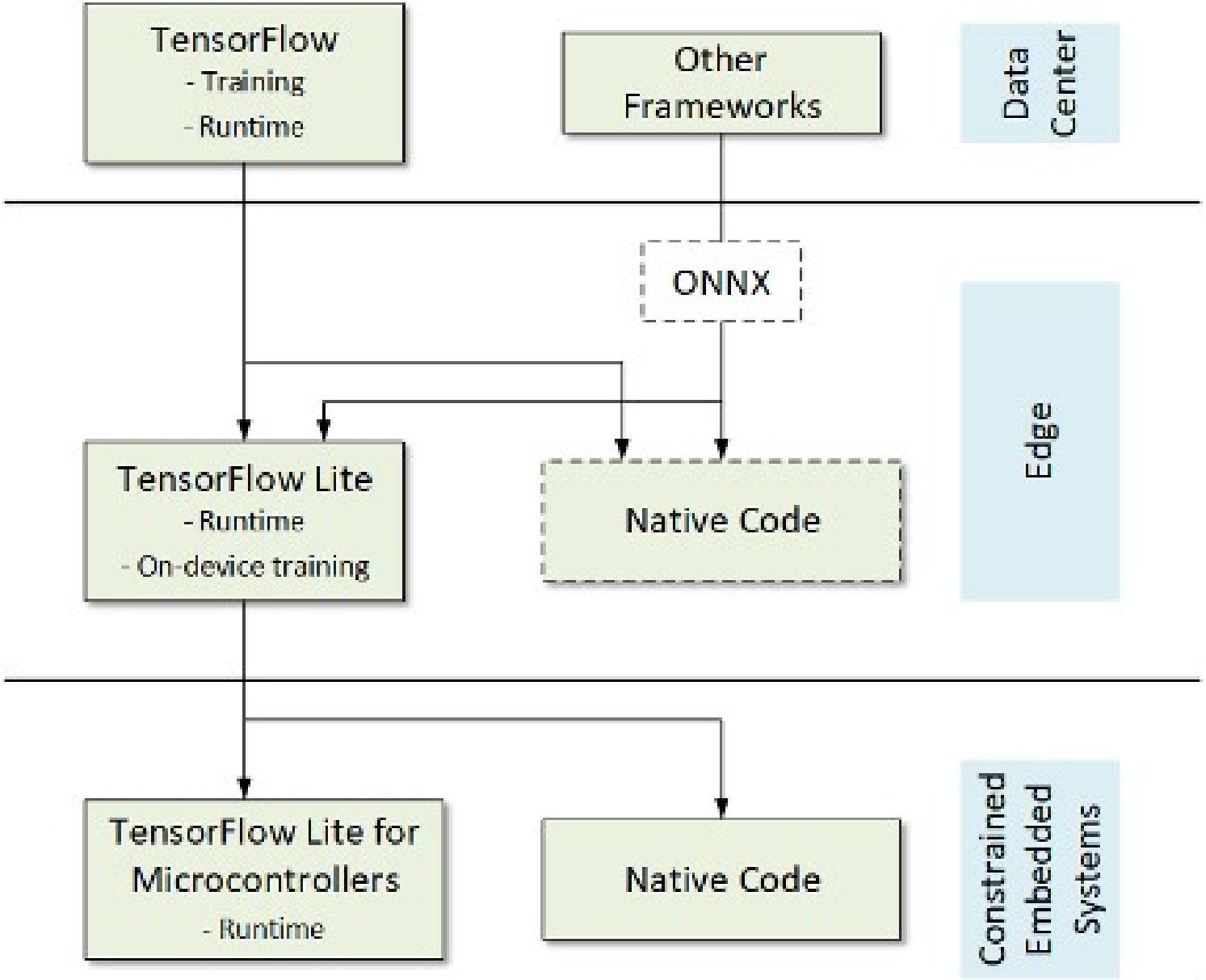
Edge device


Need Edge intelligence!
Edge Node
Sense
Measure
Interpret
Connect
Analog Domain
Digital Domain
AC Signal
Video
Current
Magnetic
DC Voltage
Pressure
Motion
Capacitance
Rotation
Temperature
Light
Particulate
Moisture
Audio
Security
AMP
ADC
Signal Processing and Analytics
Security
RF TxRx
Gateway
Cloud
Firmware and Control
Power Management
Security
Security

Edge intelligence



TinyML space



TinyML Programming

- Are there any programming language for TinyML?

TinyML Programming

- Are there any programming language for TinyML?

  Yes and No!

TinyML Programming

- Are there any programming language for TinyML?

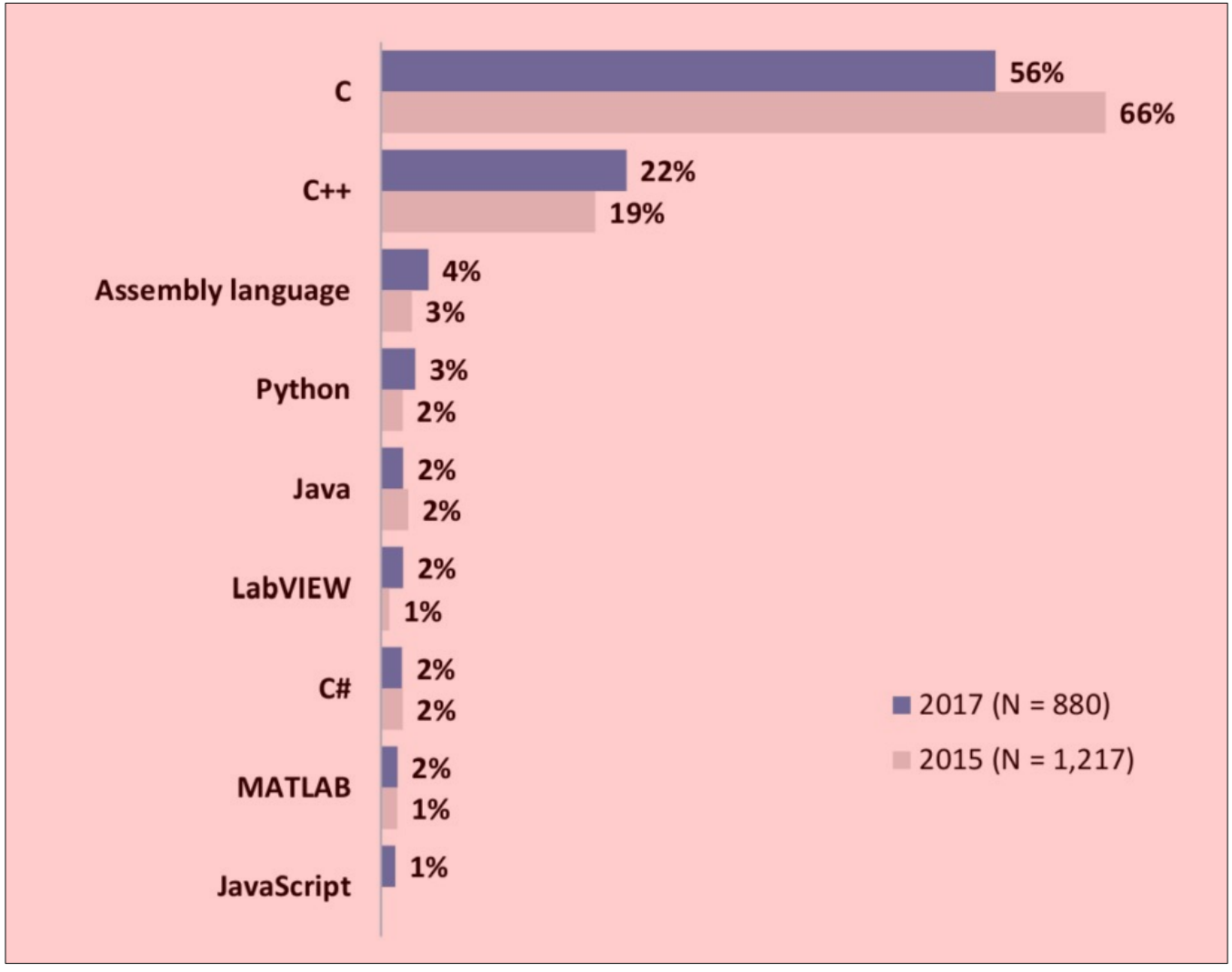
  Yes and No!

- Embedded Systems are dominated by C/C++ languages.

TinyML Programming

- Are there any programming language for TinyML?

  Yes and No!



ed by C/C++

TinyML Programming

- Are there any programming language for TinyML?

  Yes and No!

- Future computing paradigms guides to HW/SW efficient programming.

TinyML Programming

- Are there any programming language for TinyML?

  Yes and No!

- Future computing paradigms guides to HW/SW efficient programming.

HW/SW Gap!

TinyML Programming

- HW/SW Gap?

TinyML Programming

- HW/SW Gap?

SW development using high-level languages with dynamic typing are showing highly non efficient.

TinyML Programming

- HW/SW Gap?

SW development using high-level languages with dynamic typing are showing highly non efficient.

Table 1. Speedups from performance engineering a program that multiplies two 4096-by-4096 matrices. Each version represents a successive refinement of the original Python code. "Running time" is the running time of the version. "GFLOPS" is the billions of 64-bit floating-point operations per second that the version executes. "Absolute speedup" is time relative to Python, and "relative speedup," which we show with an additional digit of precision, is time relative to the preceding line. "Fraction of peak" is GFLOPS relative to the computer's peak 835 GFLOPS. See Methods for more details.

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	–	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

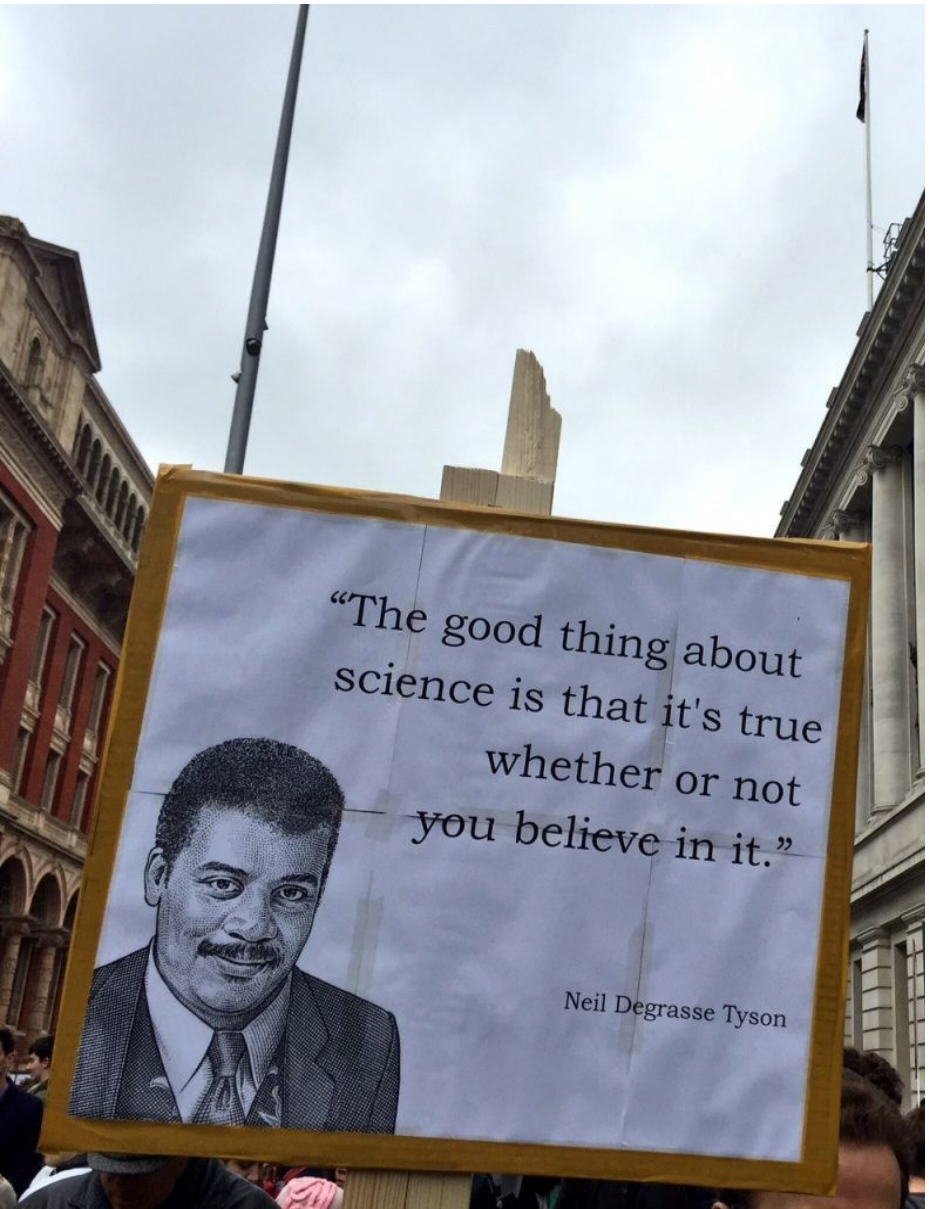
Science Junho/2020

There's plenty of room at the Top: What will drive computer performance after Moore's law?

Charles E. Leiserson, Neil C. Thompson, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez and Tao B. Schardl

TinyML Programming

- HW/SW Gap?



TinyML Programming

- HW/SW Gap?

SW development using high-level languages with dynamic typing are showing highly non efficient.

Table 1. Speedups from performance engineering a program that multiplies two 4096-by-4096 matrices. Each version represents a successive refinement of the original Python code. "Running time" is the running time of the version. "GFLOPS" is the billions of 64-bit floating-point operations per second that the version executes. "Absolute speedup" is time relative to Python, and "relative speedup," which we show with an additional digit of precision, is time relative to the preceding line. "Fraction of peak" is GFLOPS relative to the computer's peak 835 GFLOPS. See Methods for more details.

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	–	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	366	7.8	0.24
5	Parallel divide and conquer	3.80	36.180	6,727	18.4	4.33
6	plus vectorization	1.10	124.914	23,224	3.5	14.96
7	plus AVX intrinsics	0.41	337.812	62,806	2.7	40.45

Science Junho/2020

There's plenty of room at the Top: What will drive computer performance after Moore's law?

Charles E. Leiserson, Neil C. Thompson, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez and Tao B. Schardl

TinyML Programming

➢ HW/SW Gap?

SW development using high-level languages with dynamic typing are showing highly non efficient.

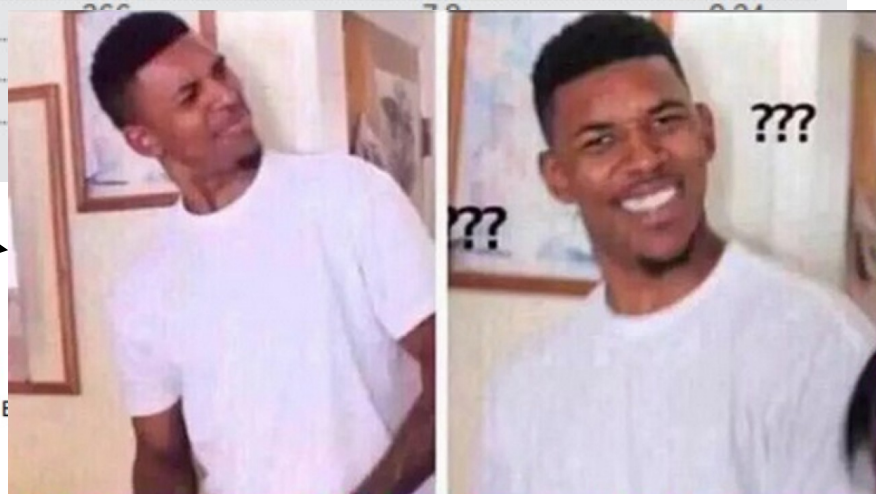
Table 1. Speedups from performance engineering a program that multiplies two 4096-by-4096 matrices. Each version represents a successive refinement of the original Python code. "Running time" is the running time of the version. "GFLOPS" is the billions of 64-bit floating-point operations per second that the version executes. "Absolute speedup" is time relative to Python, and "relative speedup," which we show with an additional digit of precision, is time relative to the preceding line. "Fraction of peak" is GFLOPS relative to the computer's peak 835 GFLOPS. See Methods for more details.

Version	Implementation	Running time (s)	GFLOPS	Absolute speedup	Relative speedup	Fraction of peak (%)
1	Python	25,552.48	0.005	1	–	0.00
2	Java	2,372.68	0.058	11	10.8	0.01
3	C	542.67	0.253	47	4.4	0.03
4	Parallel loops	69.80	1.969	[illegible]	[illegible]	[illegible]
5	Parallel divide and conquer	3.80	36.180	[illegible]	[illegible]	[illegible]
6	plus vectorization	1.10	124.914	[illegible]	[illegible]	[illegible]
7	plus AVX intrinsics	0.41	337.812	[illegible]	[illegible]	[illegible]

Science Junho/2020

There's plenty of room at the Top: What will drive computer performance after Moore's law?

Charles E. Leiserson, Neil C. Thompson, Joel S. Emer, Bradley C. Kuszmaul, Butler W. Lampson, Daniel Sanchez and Tao B Schardl



Vectorization

***Vectorization* (ISA Extensions)...**

- 1996 Intel MMX
- 1998 AMD 3DNow!
- 1999 Intel SSE on P3
- 2001 Intel SSE2 on P4
- 2003 Intel SSE3 (since Prescott P4)
- 2006 Intel SupplementalSSE3 (since Woodcrest Xeons)
- 2006 Intel SSE4 (4.1 and 4.2)
- 2007 AMD SSE5 (proposed 2007, implemented 2011)
- 2008 Intel AVX (proposed 2008, implemented 2011 in Intel Westmere and AMD Bulldozer)
  - XMM registers go from 128 bit to 256 bit, called YMM.

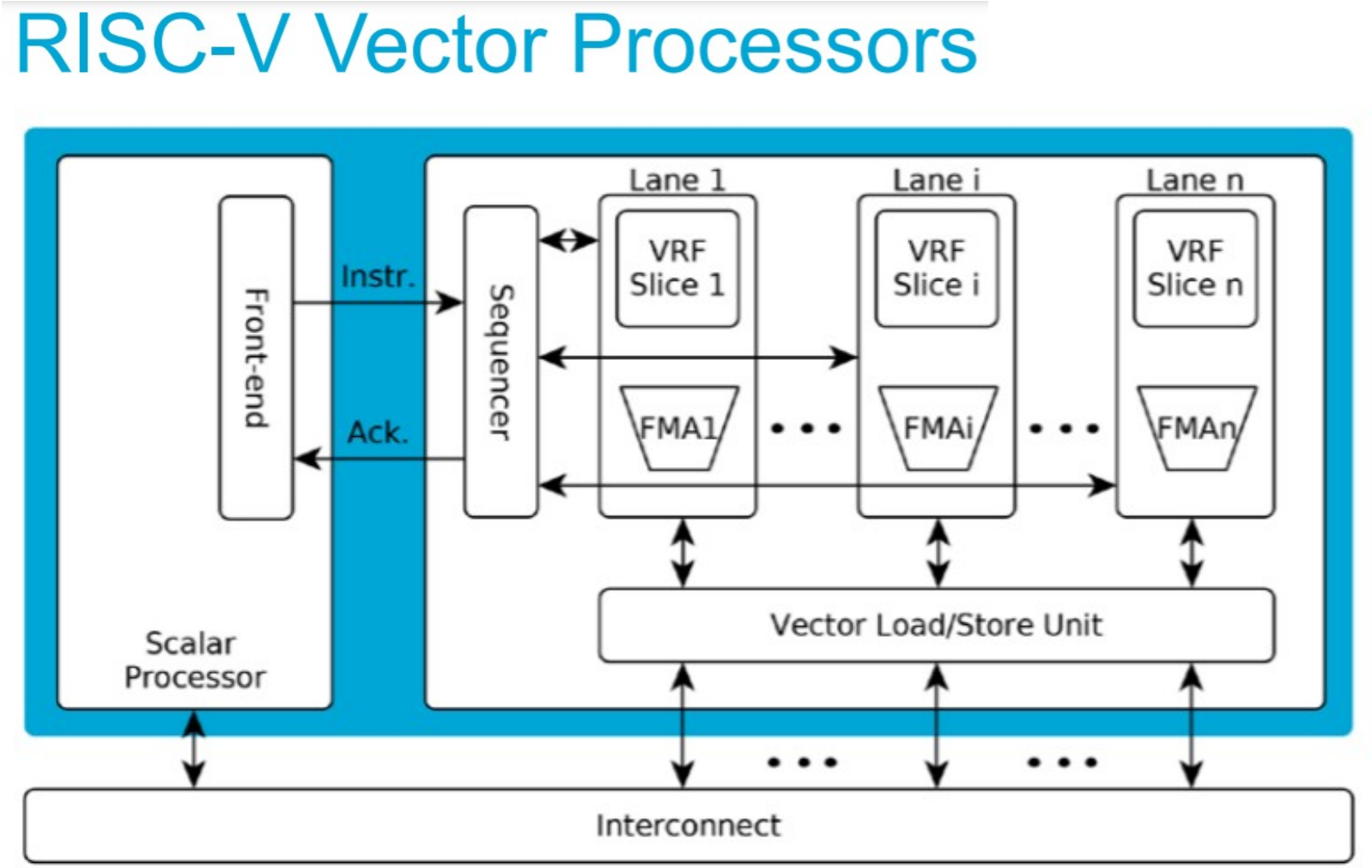
Vectorization

***Vectorization* (ISA Extensions)...**

Scalable Vector Extension (SVE) on ARMv8-A

Vectorization

Vectorization* (ISA Extensions)*...



Future Opportunities

HW/SW gap...

- Close relationship between application developers and hardware designers will be needed (or at least tools to bring then closer);

- Vectorization improvements by acceleration in the HW/SW context...

Future Opportunities

HW/SW gap...

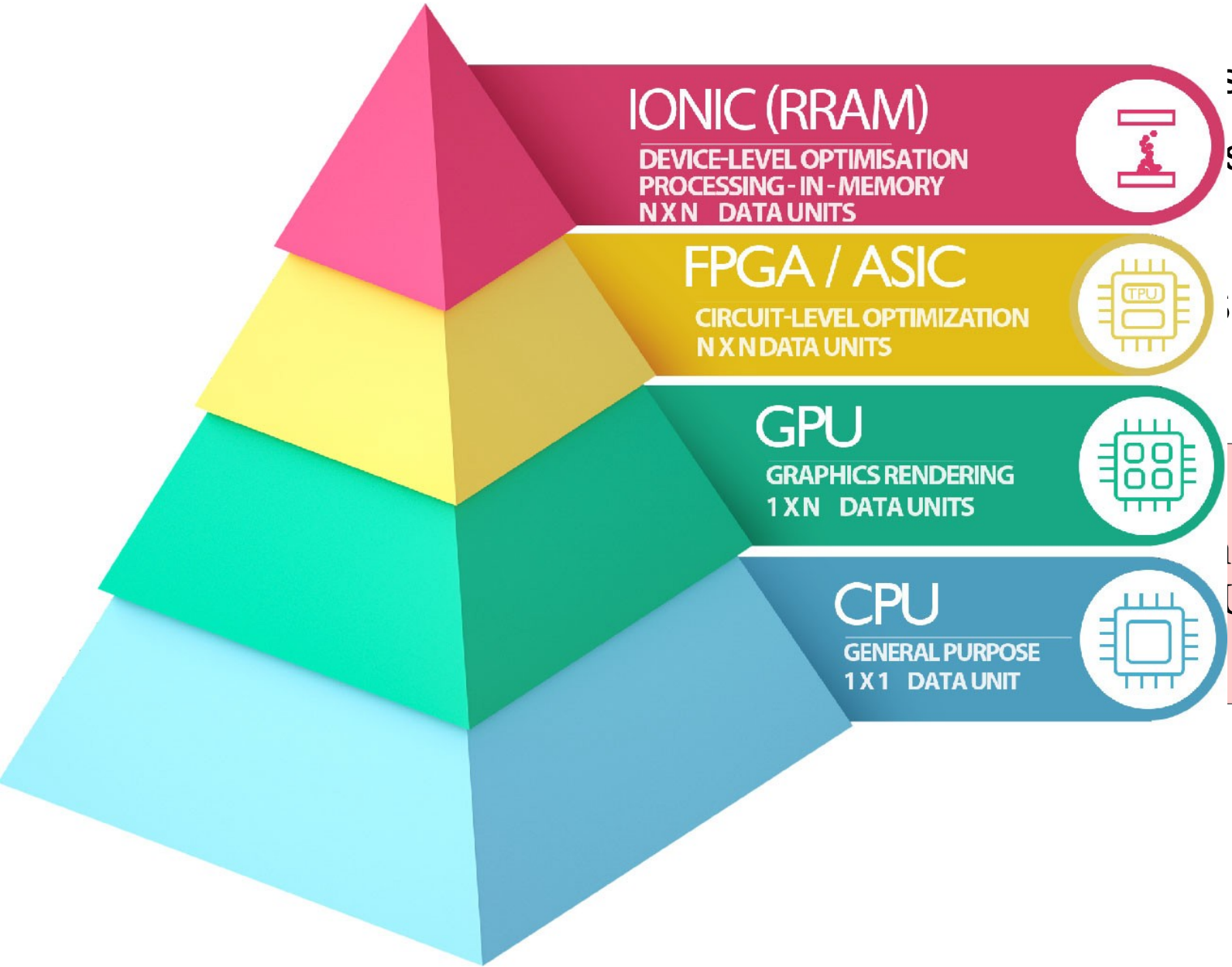
- Close relationship between application developers and hardware designers will be needed (or at least tools to bring then closer);

- Vectorization improvements by acceleration in the HW/SW context...

Hardware Acceleration and heterogeneous computing can help to close this gap using XPU technologies (CPU + GPU + TPU/NPU + FPGA)

Future Opportunities

HW/SW gap...



s and hardware

ser);

W context...

elp to close

U + FPGA)

Future Opportunities

Domain-Specific Architectures & Languages

- DSAs can be used to accelerate specific tasks optimizing the overall system performance;

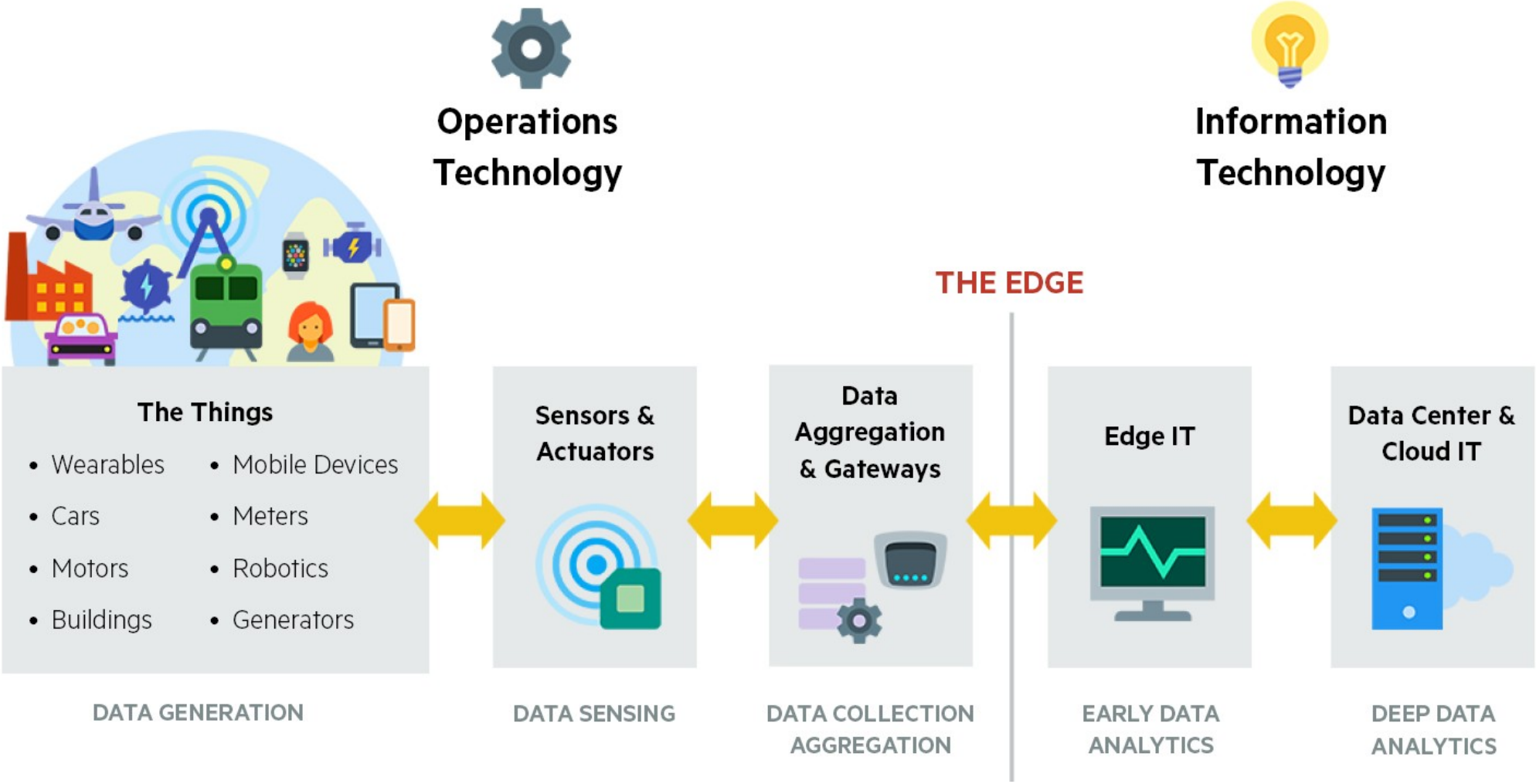
- In order for DSAs to be fully incorporated into CPU-based microprocessor systems it is necessary to use new programming languages that make the parallelism explicit: DSL;

- DSL and DSA brings a new paradigm and a needs new environments, which converge in the area of hardware acceleration using XPUs and heterogeneous computing.

Future Opportunities

Embedded Processors & Edge Computing

- Computing vs Communication paradigm...

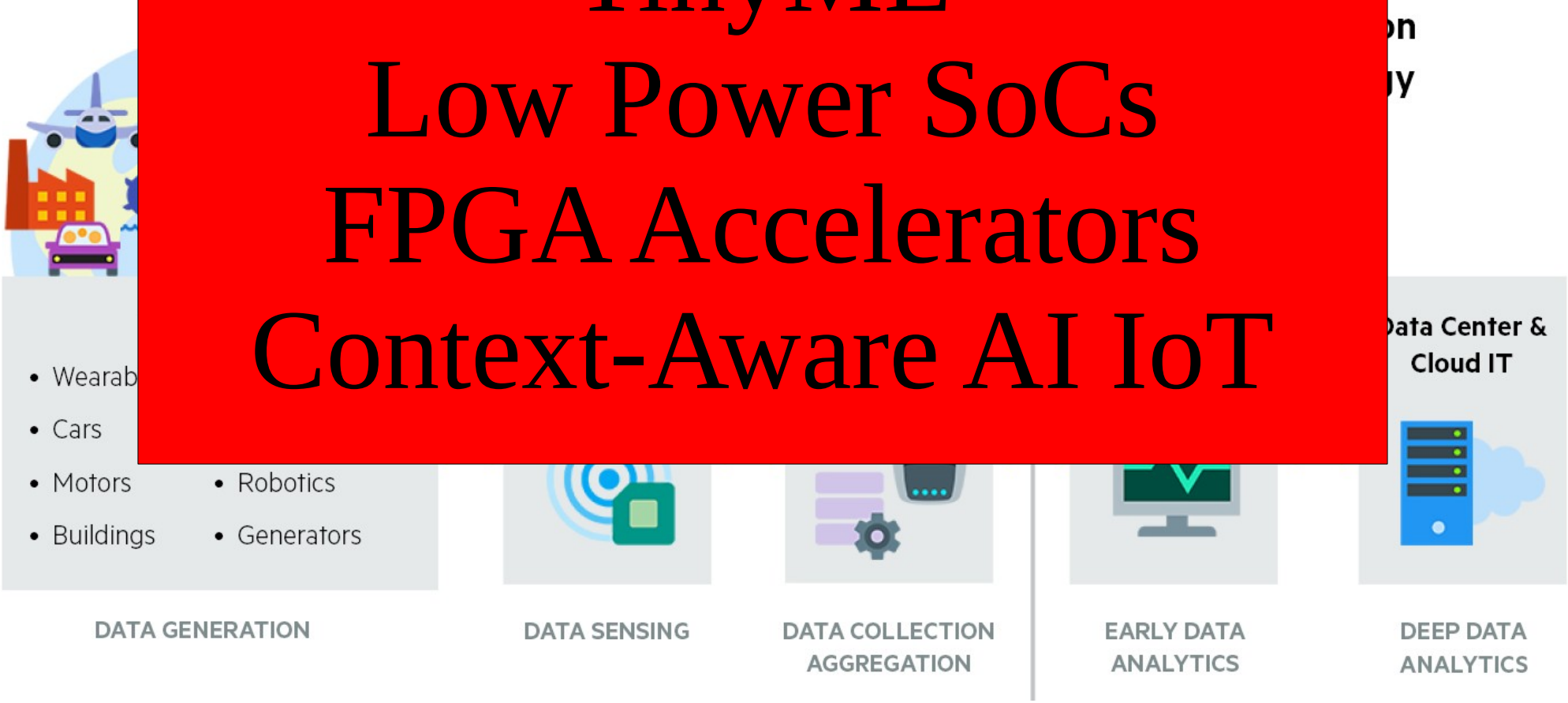


Future Opportunities

Embedded Processors & Edge Computing

- Computing vs Communication paradigm...

TinyML
Low Power SoCs
FPGA Accelerators
Context-Aware AI IoT

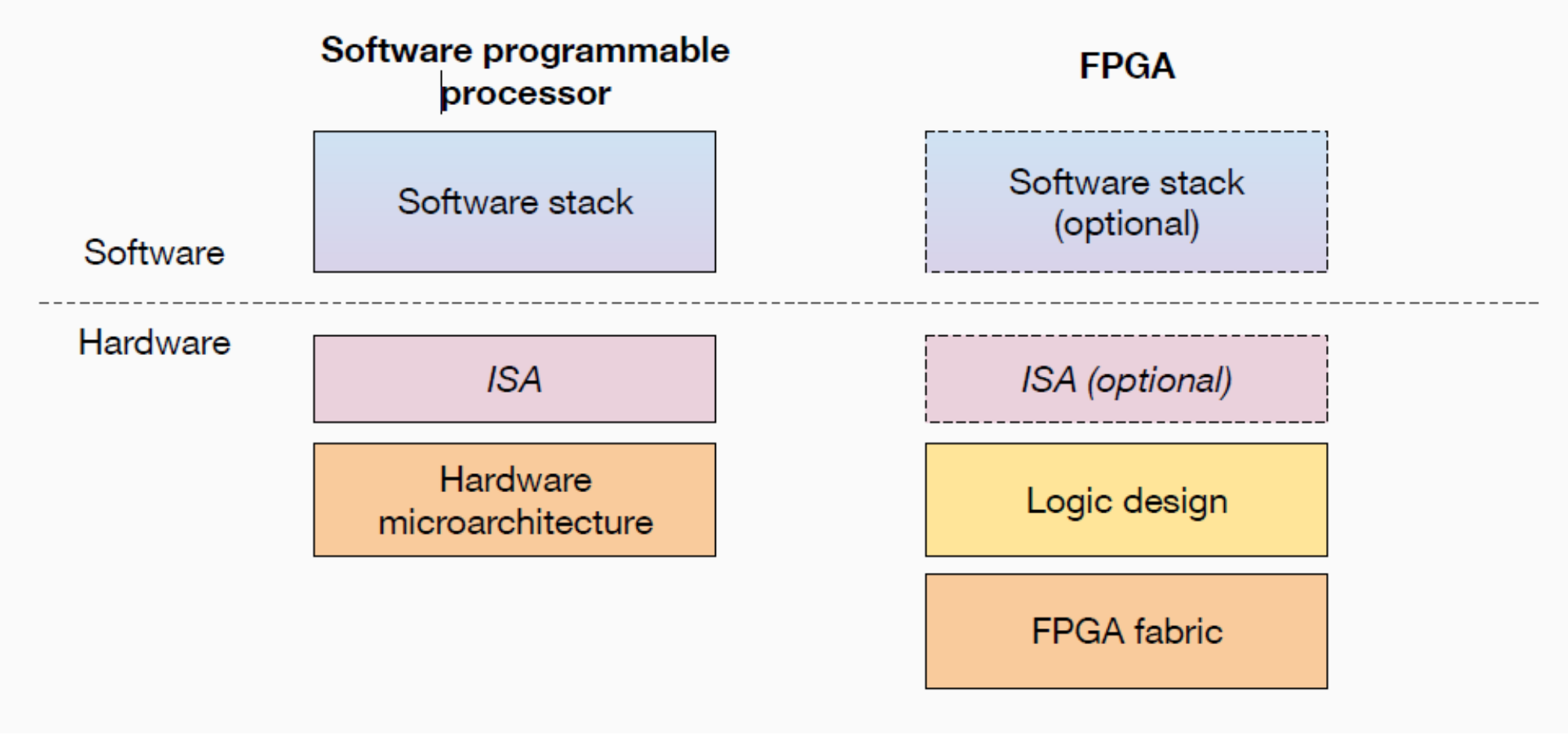


TinyML vs FPGA ML Acceleration

- FPGA ML Acceleration is more related to HW!

TinyML vs FPGA ML Acceleration

- FPGA ML Acceleration is more related to HW!

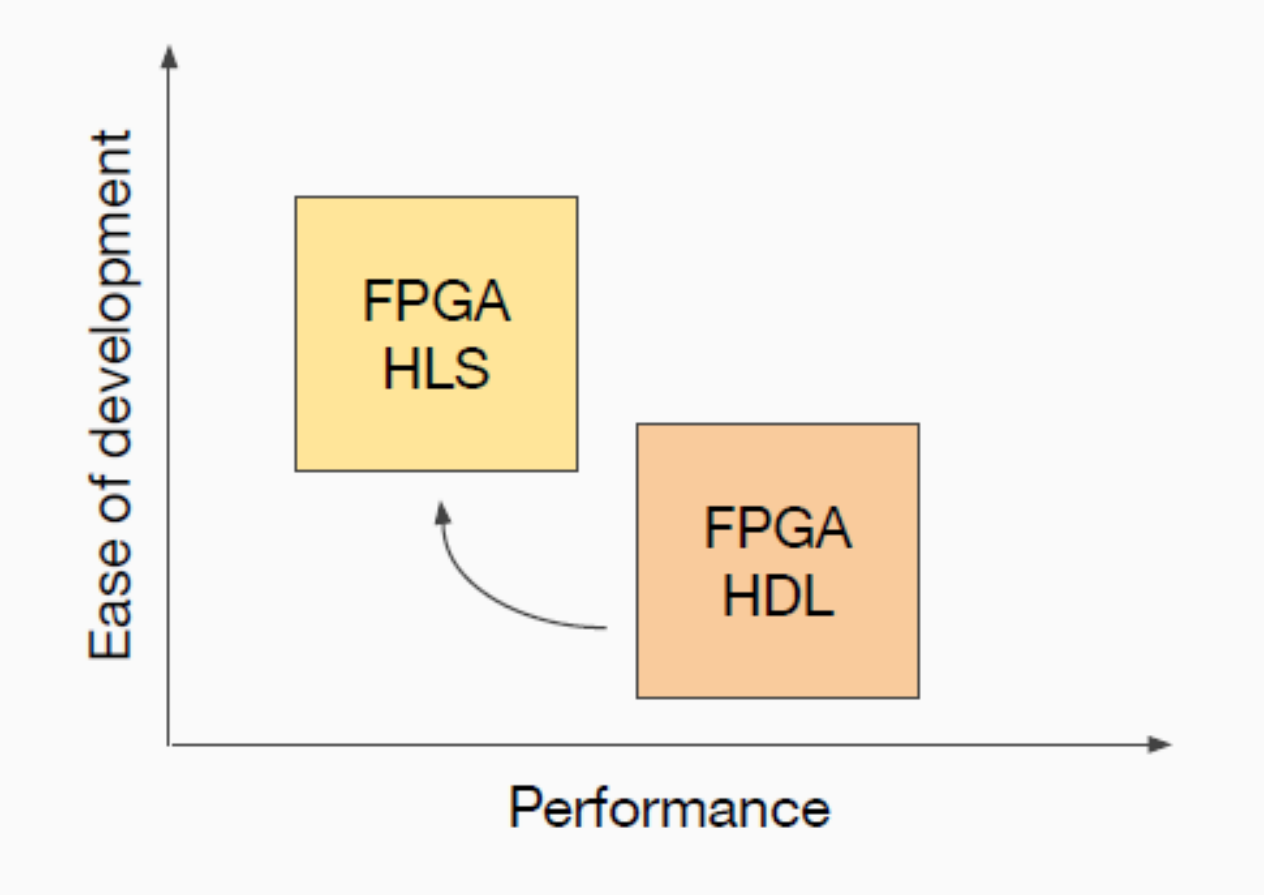


TinyML vs FPGA ML Acceleration

- FPGA ML Acceleration is more related to HW!
- Higher parallelism and full HW reconfigurability;
- Hardware description languages / RTL design;
- HLS...

TinyML vs FPGA ML Acceleration

- FPGA ML Acceleration is more related to HW!
- Higher parallelism and full HW reconfigurability;
- Hardware description languages / RTL design;
- HLS...



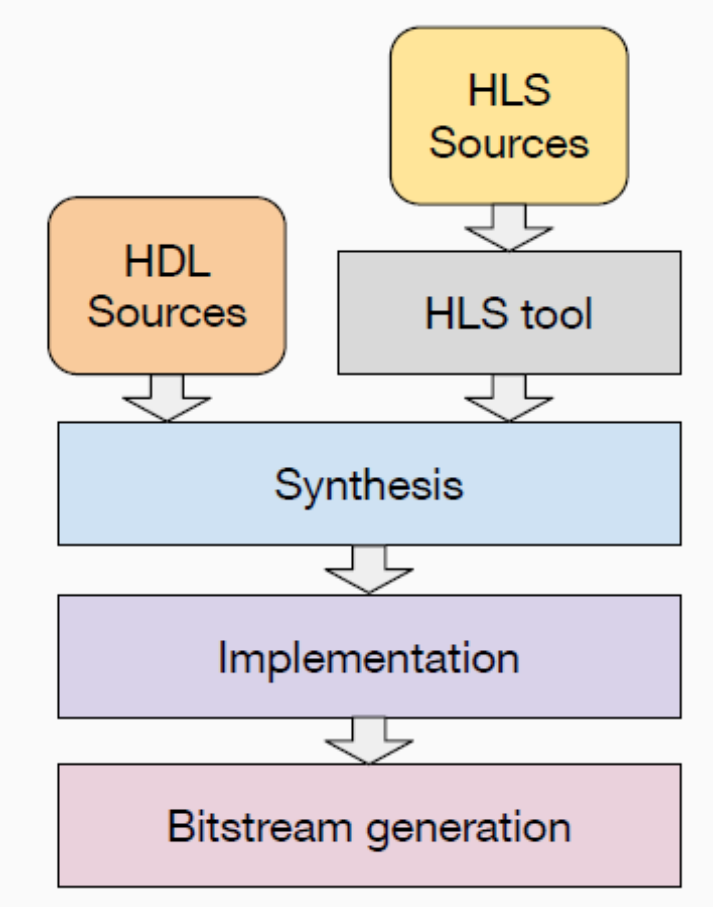
TinyML vs FPGA ML Acceleration

- FPGA ML Acceleration is more related to HW!
- Higher parallelism and full HW reconfigurability;
- Hardware description languages / RTL design;
- HLS...

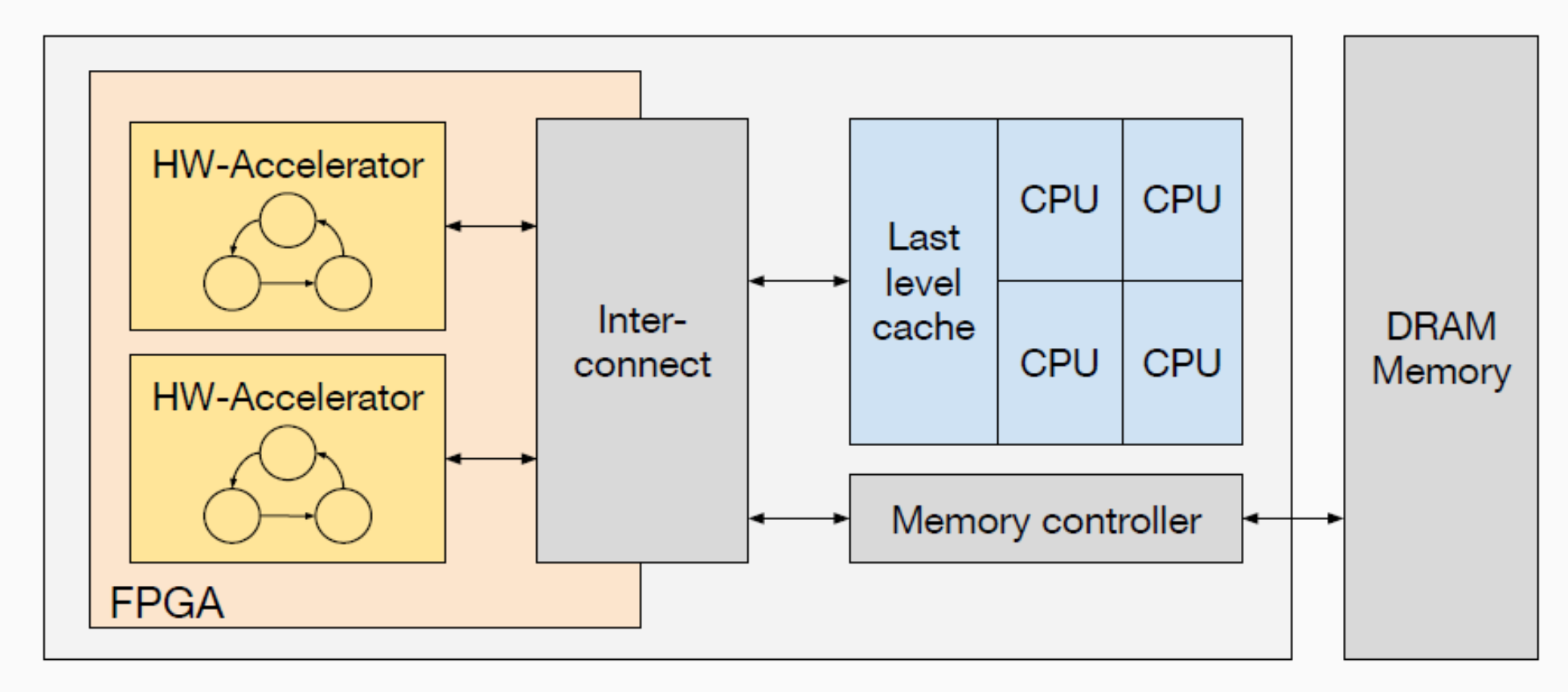
CFU-Playground

TinyML vs FPGA ML Acceleration

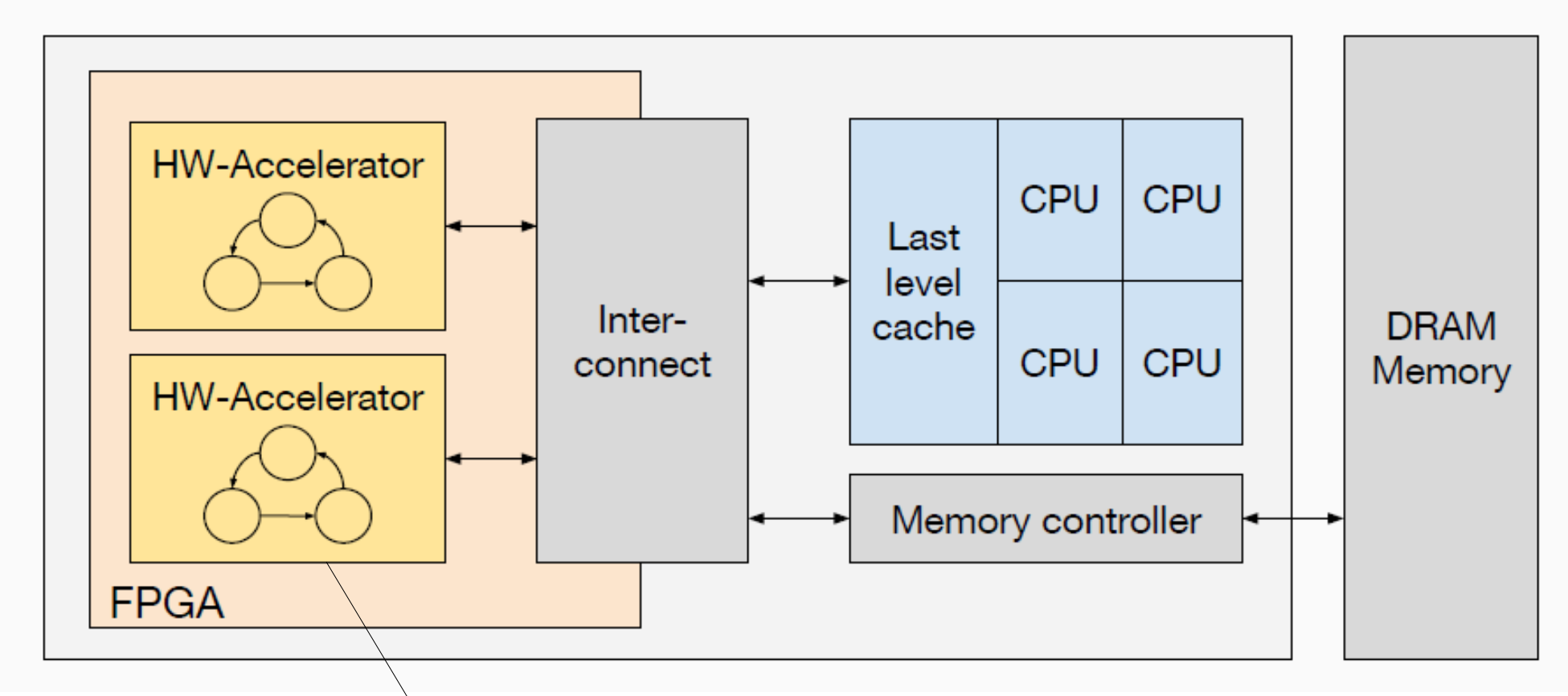
- FPGA ML Acceleration is more related to HW!
- Higher parallelism and full HW reconfigurability;
- Hardware description languages / RTL design;
- HLS…
- Another toolchain...



TinyML with heterogeneous SoC-FPGA

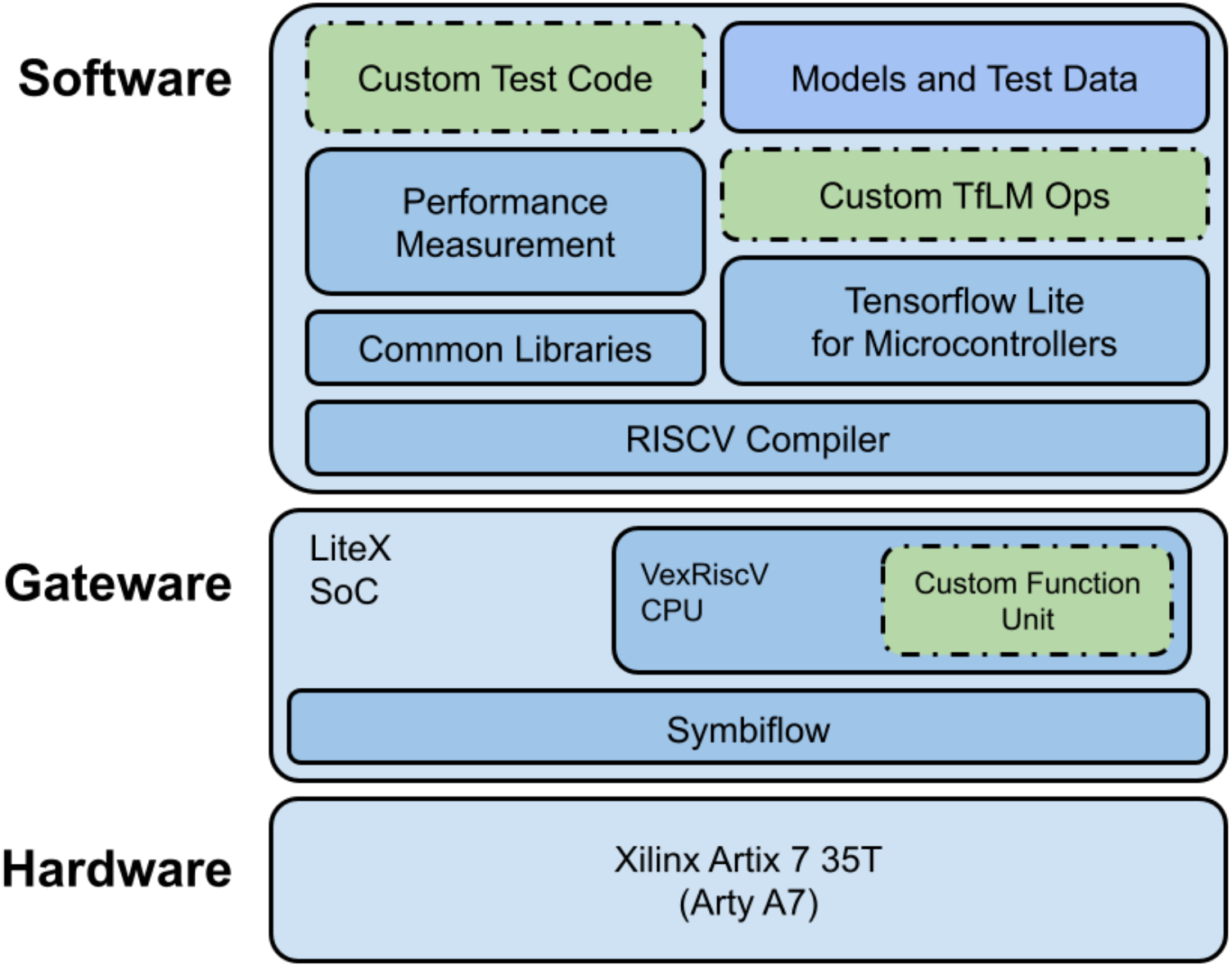


TinyML with heterogeneous SoC-FPGA



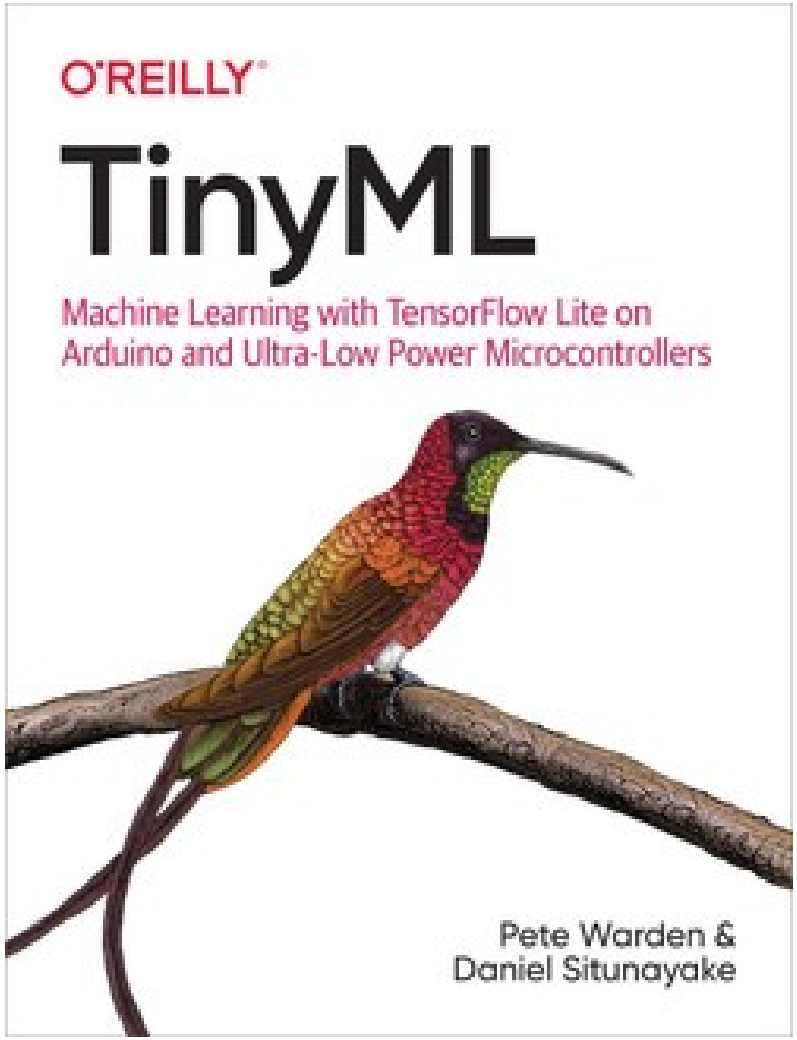
Need another talk only for
FPGA ML Acceleration

TinyML with CFU Playground

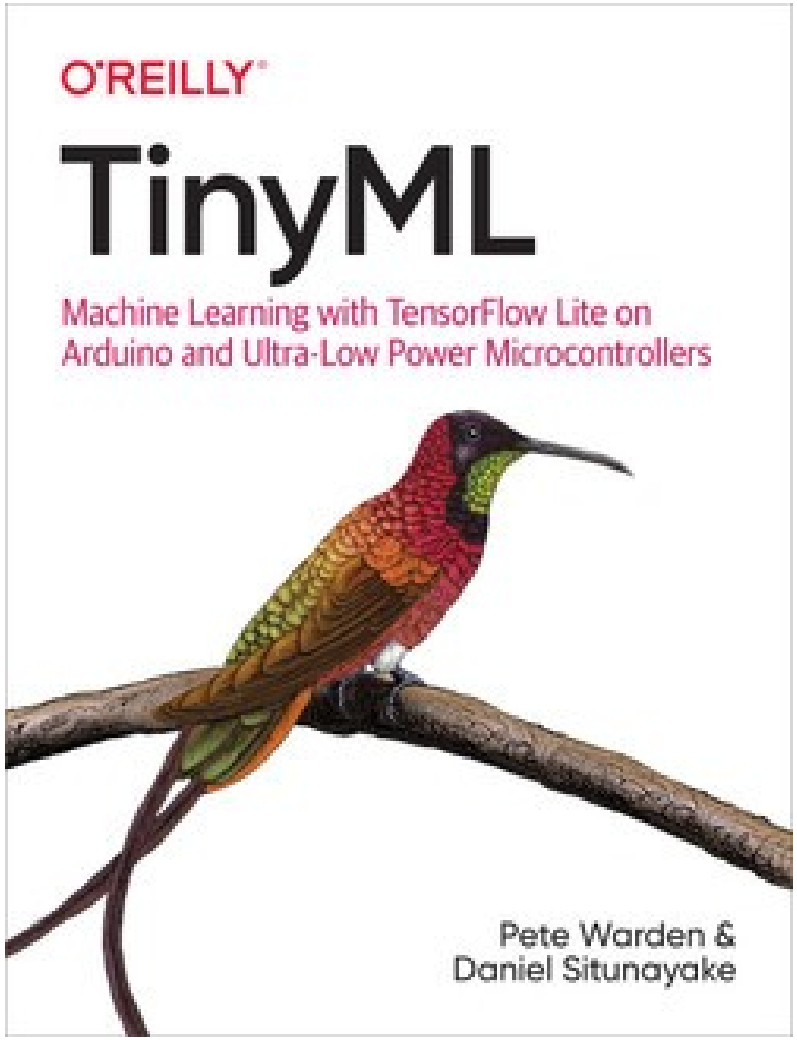


CFU Playground Overview

More on TinyML

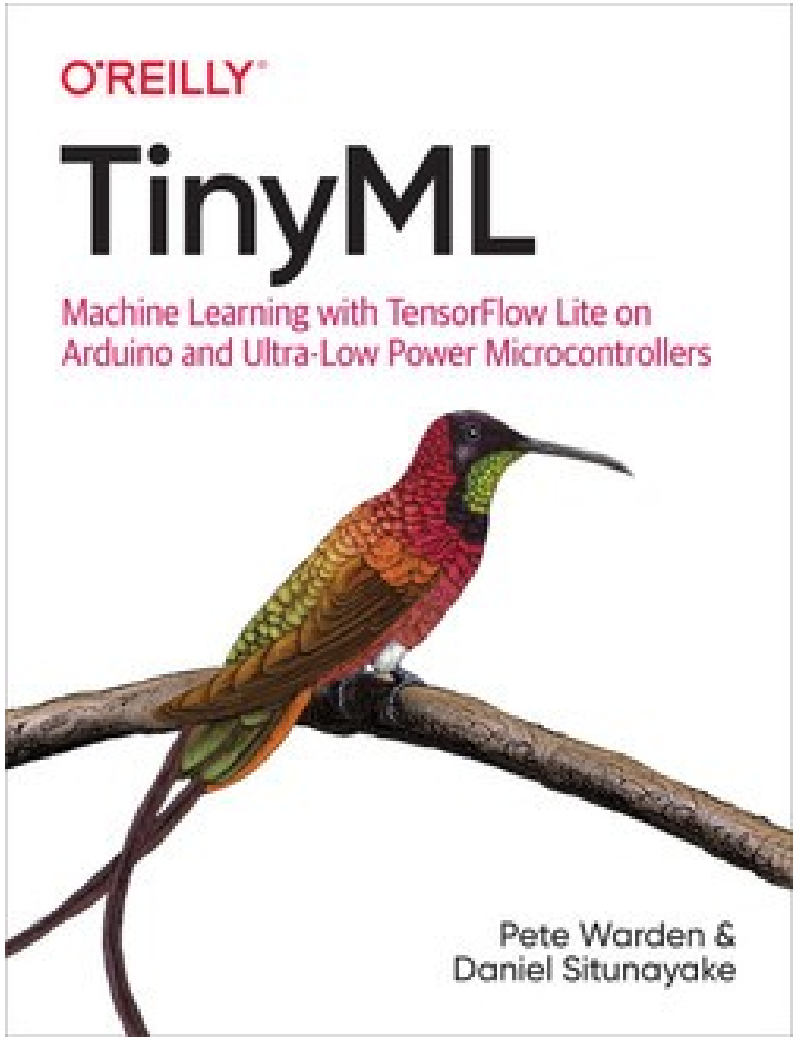


More on TinyML



The term **TinyML** was coined by Pete Warden...

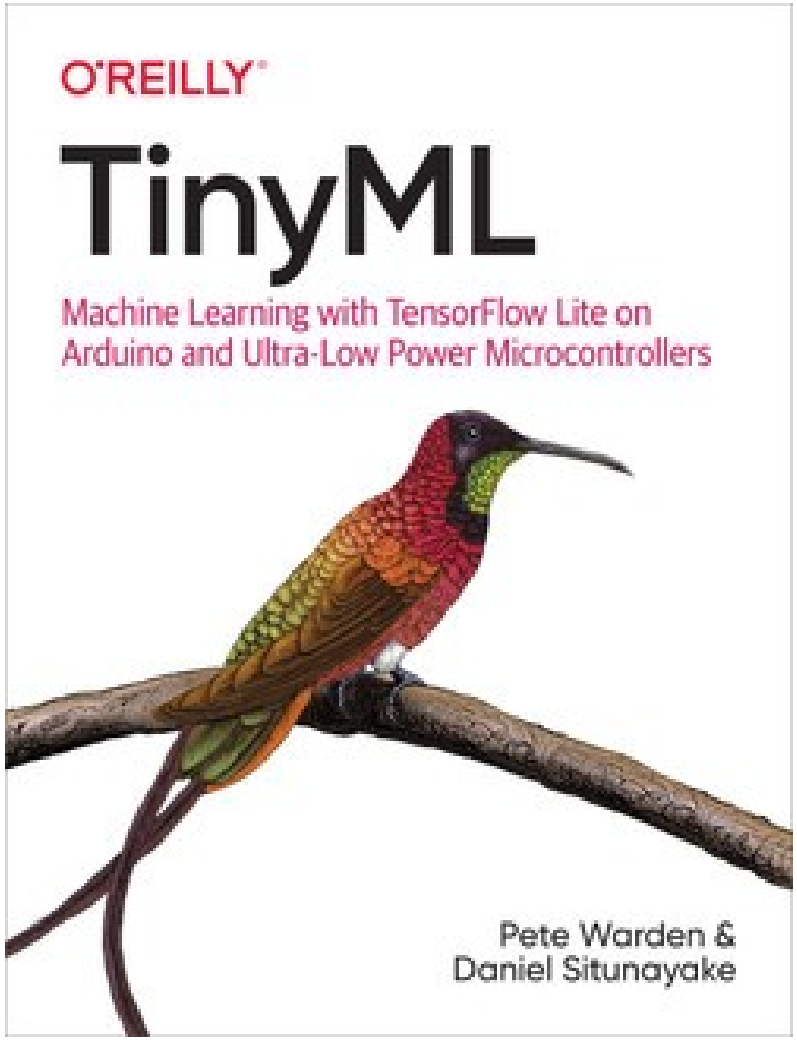
More on TinyML



The term **TinyML** was coined by Pete Warden...

Get the hands dirty with the walk-through provided by the book.

More on TinyML



The term **TinyML** was coined by Pete Warden...

Get the hands dirty with the walk-through provided by the book.

So, what is next?