# tinyML® Summit

*Miniature dreams can come true…*

**March 28-30, 2022 | San Francisco Bay Area**

TINY ML

www.tinyML.org

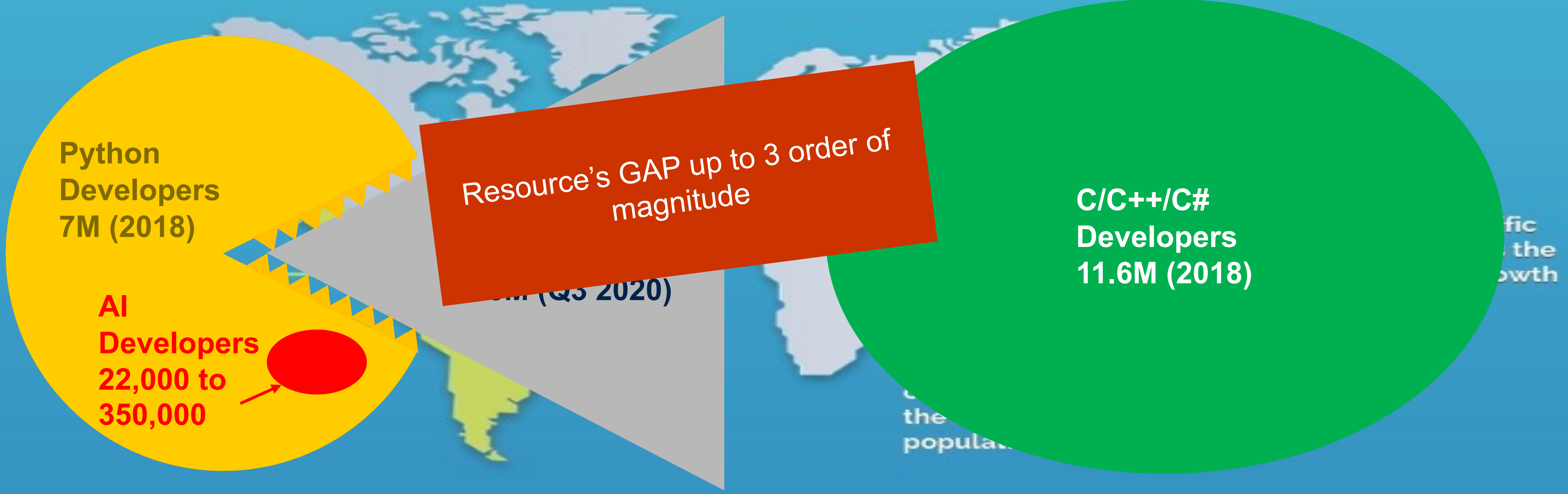
life.augmented

# Ecosystem of tools for better productivity

Danilo PAU, Technical Director, IEEE and ST Fellow, STMicroelectronics

# Global Developer Population and Demographic Study 2019, Vol 1

Source https://www.daxx.com/blog/development-trends/number-software-developers-world
*https://www.stateofai2019.com/chapter-6-the-war-for-talent/#:~:text=Estimates%20of%20the%20number%20of,AI%20originated%20in%20academia.



Python Developers 7M (2018)

AI Developers 22,000 to 350,000

Resource's GAP up to 3 order of magnitude

C/C++/C# Developers 11.6M (2018)

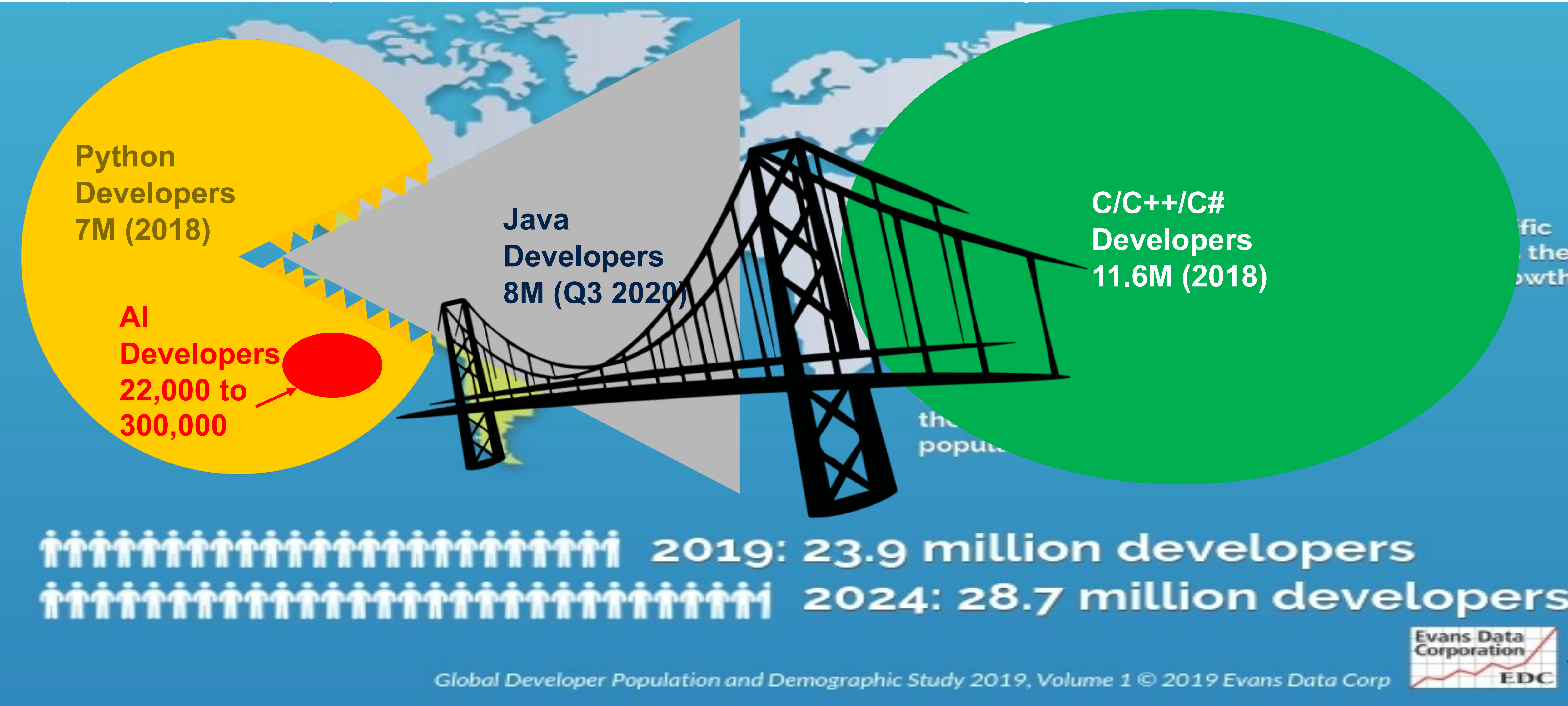
2019: 23.9 million developers

2024: 28.7 million developers

*Global Developer Population and Demographic Study 2019, Volume 1 © 2019 Evans Data Corp*

Evans Data Corporation EDC

# How to bridge the AI and embedded communities?



Python Developers 7M (2018)

AI Developers 22,000 to 300,000

Java Developers 8M (Q3 2020)

C/C++/C# Developers 11.6M (2018)

2019: 23.9 million developers

2024: 28.7 million developers

Global Developer Population and Demographic Study 2019, Volume 1 © 2019 Evans Data Corp

# More FAEs ? How many ? WRONG!

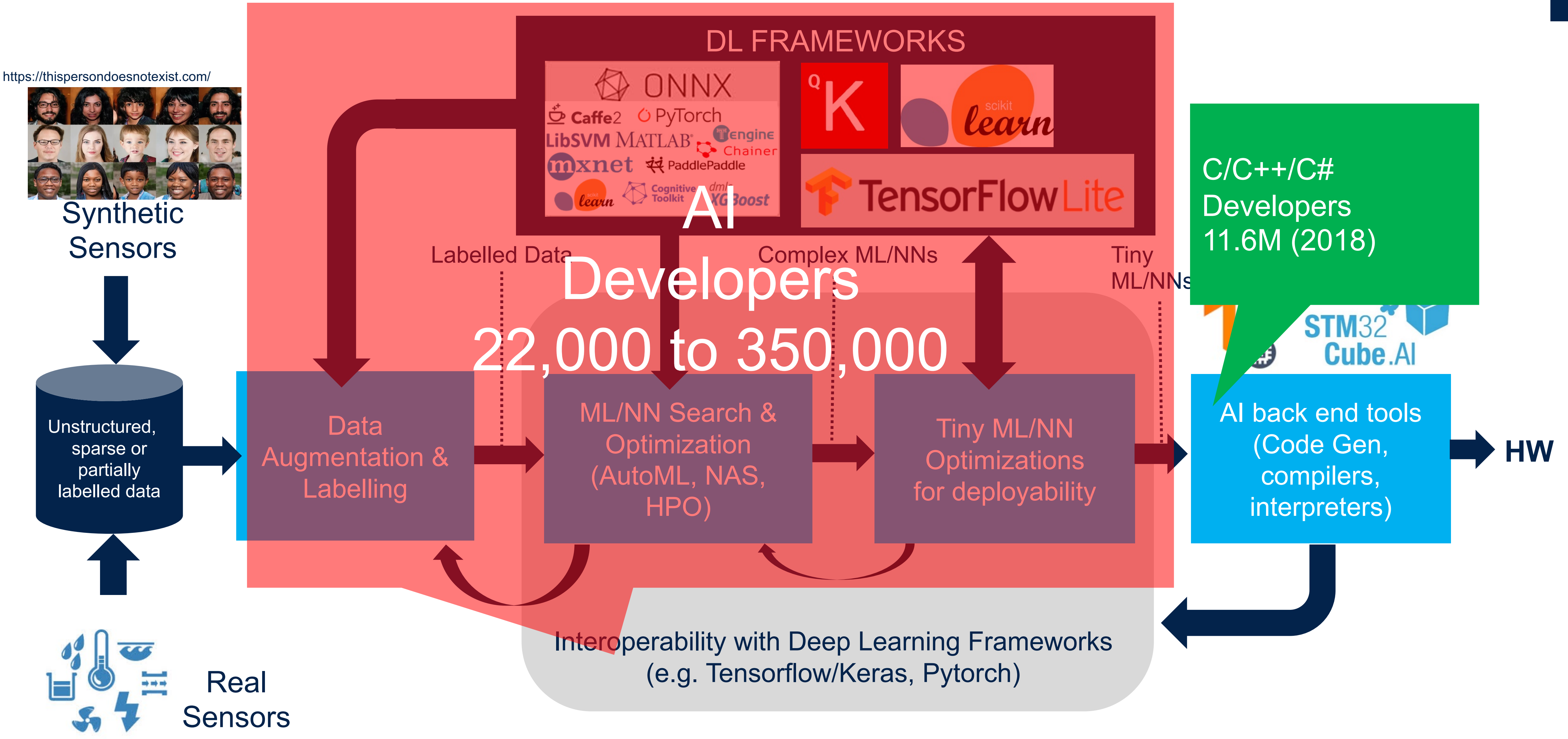
**24.1 billion**
IoT connected devices in 2030 (7.6bn 2019)

**$1.5 trillion**
IoT revenue in 2030 ($465bn 2019)

*Image source: Transforma Insights*

# Listen to their needs

# End-to-end Tools Ecosystem



https://thispersondoesnotexist.com/

Synthetic Sensors

Unstructured, sparse or partially labelled data

Real Sensors

DL FRAMEWORKS

AI Developers 22,000 to 350,000

Labelled Data

Complex ML/NNs

Tiny ML/NNs

Data Augmentation & Labelling

ML/NN Search & Optimization (AutoML, NAS, HPO)

Tiny ML/NN Optimizations for deployability

C/C++/C# Developers 11.6M (2018)

AI back end tools (Code Gen, compilers, interpreters)

HW

Interoperability with Deep Learning Frameworks (e.g. Tensorflow/Keras, Pytorch)

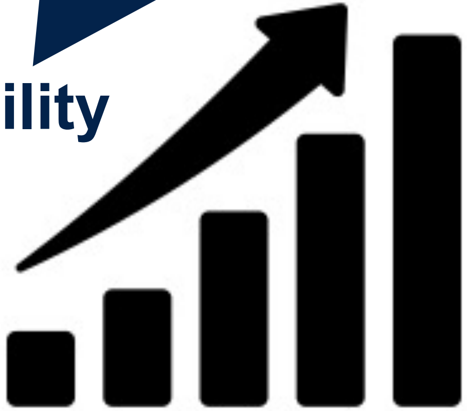
# The needs

## Interoperability

"The documented agreement reached by a group of individuals who recognize the advantage of all doing certain things in an agreed way".
Leonardo Chiariglione

Serving trillions of sensors

## Scalability



## Automation

"everything that can be automated will be automated".
First law
Shoshana ZUBOFF

## Productivity

Keep calm and
hand-craft ML

# Auto tinyML

## March 30 , 2022 – 15:10 to 17:30

**EON Tuner: AutoML for constrained devices**
Jan JONGBOOM, CTO, Edge Impulse

**Optimizing AutoML for the tinyML Future**
Elias FALLON, VP for Machine Learning, Qeexo Co.

**1 kB and not a bit more! The ideal weight for a tinyML model**
Blair NEWMAN, CTO, Neuton

**Model Optimization with QKeras' Quantization-Aware Training and Vizier's Automatic Neural Architecture Search**
Daniele MORO, Software Engineer, Google

**Automated Machine Learning under model's deployability on tiny devices**
Antonio CANDELIERI, Assistant Professor, University of Milano-Bicocca, Italy

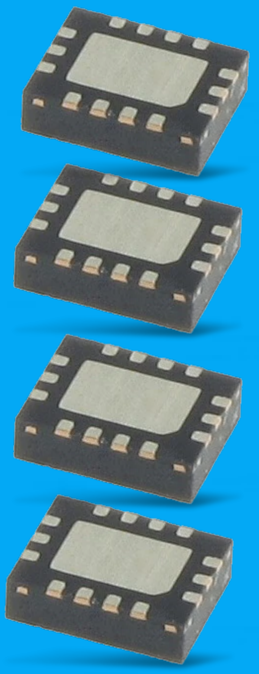
**Automating Model Optimization for Efficient Edge AI: from automated solutions to open-source toolkit**
Dave Cheng, Qualcomm, USA

Session moderator: Danilo PAU
Technical Director, IEEE and ST Fellow, STMicroelectronics

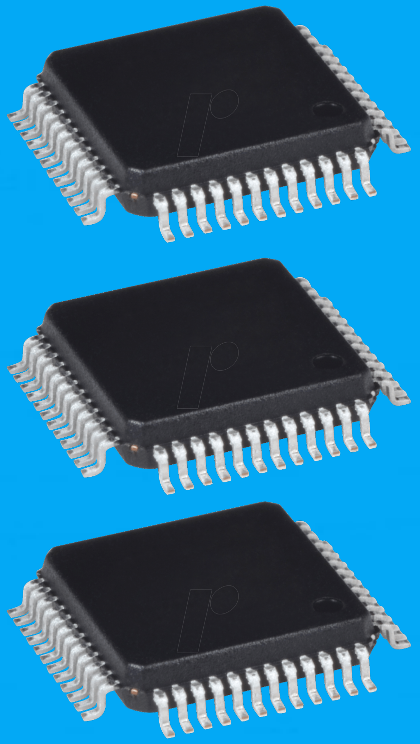
life.augmented

# Further challenges

- A sensor with **a C compiler** !
- **QKeras importer**
- Processor, 5 - 10 MHz
- Binary instructions
- Memory, 40 KiB
- µW energy envelope

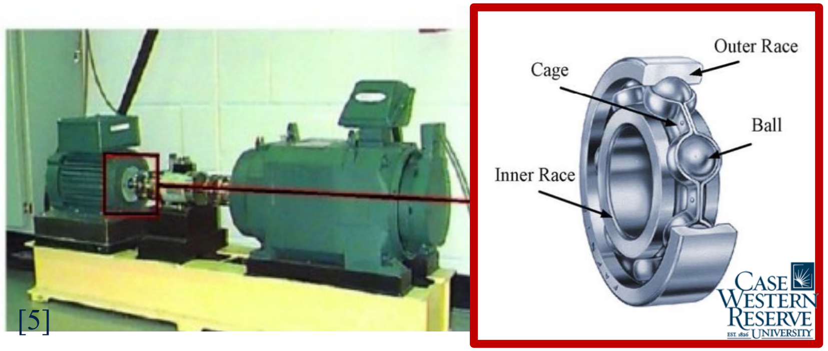


- MCU standard vs custom ISA
- **Code gen, interpreters**
- Different **compilers**
- Processor, 10s - 100s MHz
- Embedded RAM, 10s-100s KiB
- Embedded FLASH, ≤ 2MB
- mW energy envelope

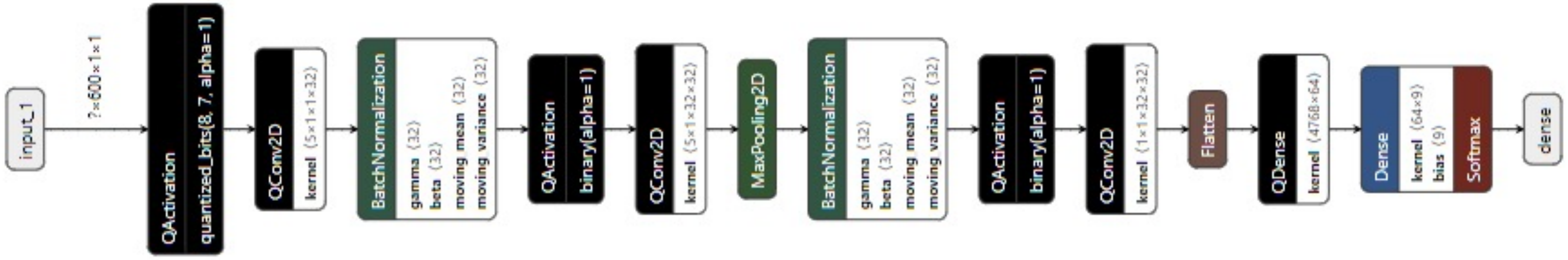


```
!pip install qkeras==0.9.0

# feature extractor
x = x_in = Input(shape)
x = QActivation("quantized_bits(8, 7, alpha=1)", name="act_0")(x)
x = QConv2D(channel_CNN, (kernel_size_CNN, 1),
    kernel_quantizer="quantized_bits(8, 7, alpha=1)",
    use_bias = False,
    name="conv2d_1")(x)
x = BatchNormalization()(x)
x = QActivation("binary(alpha=1)", name="act_1")(x)
x = QConv2D(channel_CNN, (kernel_size_CNN, 1),
    kernel_quantizer="binary(alpha=1)",
    padding="same",
    use_bias = False,
    name="conv2d_2")(x)
x = MaxPooling2D(pool_size=(pool_size_CNN,1))(x)
x = BatchNormalization()(x)
x = QActivation("binary(alpha=1)")(x)
x = QConv2D(32, (1, 1),
    kernel_quantizer="binary(alpha=1)",
    padding="same",
    use_bias = False)(x)
# CNN_Head - classifier
x = Flatten()(x)
x = QDense(64,kernel_quantizer="binary(alpha=1)",use_bias=False)(x)
x = Dense(9,activation="softmax")(x)
```

```
Number of operation types in model:
sfmult_1_32:    305152    (8.5%)
smult_8_8:      95360     (2.6%)
sxor_1_1:       3204096   (89%)

Weight profiling:
conv2d_1_weights :    160 (8-bit unit)
conv2d_2_weights :   5120 (1-bit unit)
q_conv2d_weights :   1024 (1-bit unit)
q_dense_weights : 305152 (1-bit unit)
```



# Anomaly (bearings) classification



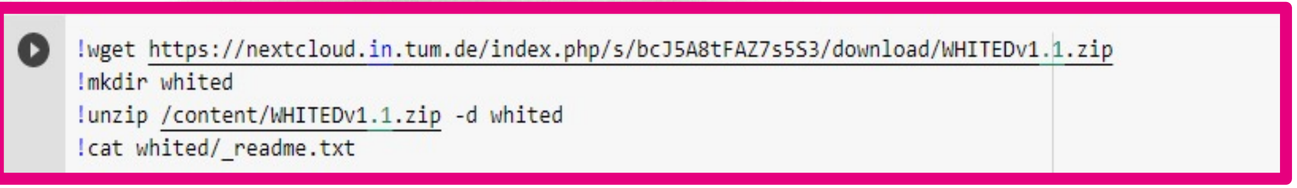
| Model | | Accuracy (average of 10 trials) % | MACC | WEIGHTS KiB | RAM KiB | STM32 inference/s |
|---|---|---|---|---|---|---|
| Keras | FP32 | 98.89 | 3,624,496 | 1218.56 | 75.5 | 19 |
| TFLite | INT8 | 25.96 | 3,634,132 | 305.44 | 24.94 | 34 |
| **QKeras** | **INT1/FP32** | **98.39** | **3,624,432** | **41.32** | **18.88** | **72** |

Note: based on X-CUBE-AI (alpha version) code generation

# Home Appliance classification

- Only current measurements
- 16 KHz sampling rate

```
!pip install qkeras==0.9.0
```



```
!wget https://nextcloud.in.tum.de/index.php/s/bcJ5A8tFAZ7s5S3/download/WHITEDv1.1.zip
!mkdir whited
!unzip /content/WHITEDv1.1.zip -d whited
!cat whited/_readme.txt
```

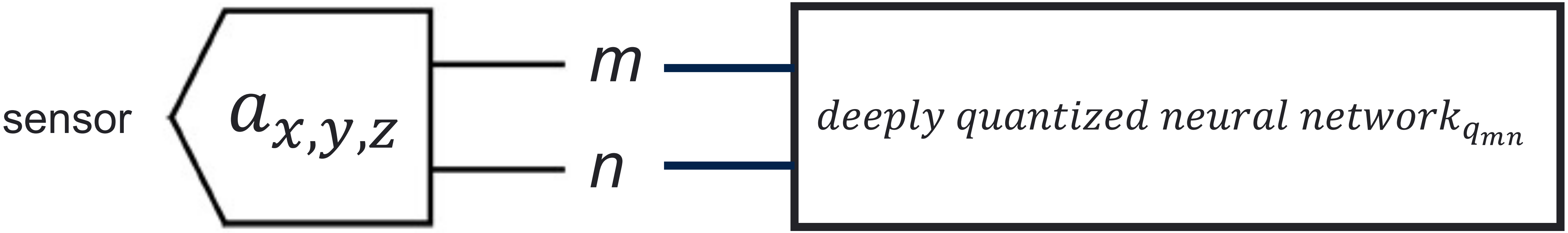
```
Number of operation types in model:
smult_8_8:   276768   (3.4%)
sxor_1_1:   7831008 (96.6%)
```

| Model | | MACC | FLASH KiB | RAM KiB | Accuracy (%) K-fold=5 | Inference/s |
|---|---|---|---|---|---|---|
| Keras | FP32 | 5,864,688 | 108.5 | 87.12 | 99.43 | 12 |
| TFLITE | INT8 | 5,888,972 | 27.68 | 22.81 | 99.39 | 30 |
| **QKeras** | **INT1/FP32** | **8,139,824** | **11.84** | **16** | **98.73** | **105** |

Note: based on X-CUBE-AI (alpha version) code generation

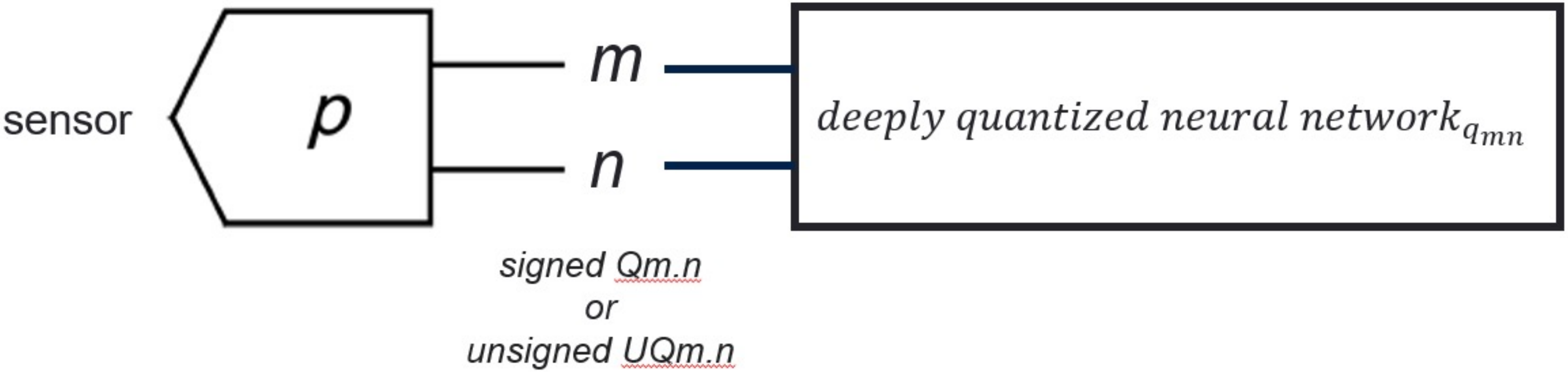
# Generalizing

```
Average Accuracy of AutoEncoder model in 10 (CWRU) trials is: 98.5167

Number of operation types per inference:
    smux_1_16                         : 7200
    sxor_1_1                          : 6080

Weight profiling:
    q_conv2d_2_weights                 : 40     (1-bit unit)
    q_conv2d_3_weights                 : 320    (1-bit unit)
    q_conv2d_4_weights                 : 320    (1-bit unit)
    q_conv2d_5_weights                 : 40     (1-bit unit)
    Total                              : 720     bits

Activations profiling:
    Total allocation                   : 512     bytes
```



*signed Qm.n*
*or*
*unsigned UQm.n*

# How to automate the dqnn design?



# Our technology starts with You

danilo.pau@st.com

life.augmented

TINY ML SUMMIT

# tinyML Summit 2022 Sponsors

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

AONdevices

arm

ASPINITY

brainchip
The Neuromorphic Computing Company

CEVA®

Deeplite

EDGE IMPULSE

emza
visual sense

FotaHub

GREENWAVES
TECHNOLOGIES

Grovety Inc.

Himax

HOTG

imagimob

infineon

itemis

KLIKA·TECH
GLOBAL IOT SOLUTIONS

LatentAI

LATTICE
SEMICONDUCTOR

Micro.ai

OmniML

NXP

PXI

Plumerai

PROPHESEE

Qeexo

Qualcomm

Rackner

RealityAI®
Engineering Solutions for the Edge

REEXEN
technology

RENESAS

SAP

seeed studio
The IoT Hardware Enabler

SensiML™

Sony
Semiconductor
Solutions
Corporation

ST
life.augmented

SA STREAM ANALYZE

synaptics®

SynSense

SYNTIANT

Tensil.ai

TensorFlow

XMOS

TINY ML SUMMIT

# Copyright Notice

**www.tinyml.org**