

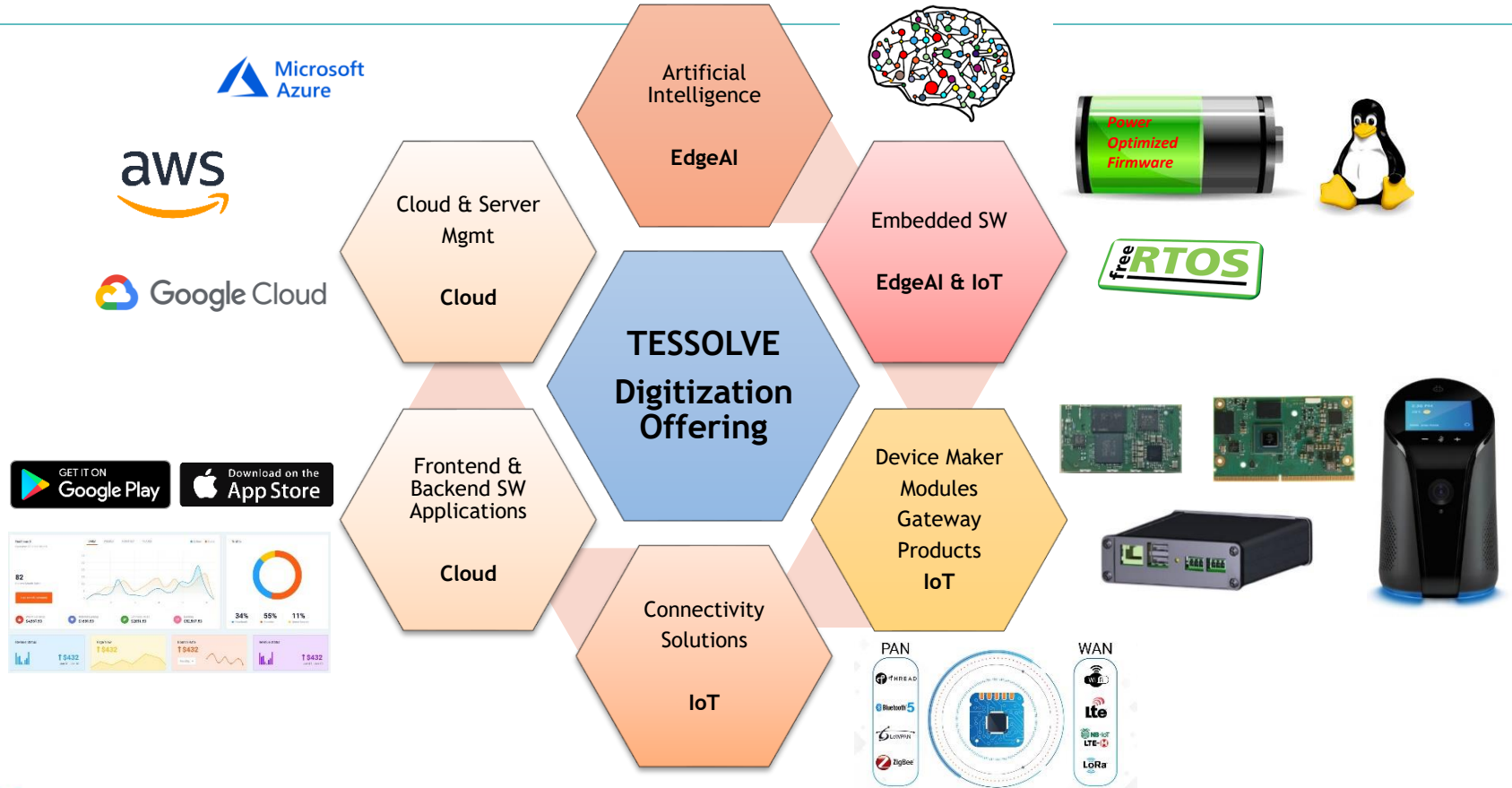
# Building a Trusted AI-based System – Building Blocks

Techworks NMI Oct/2021

AI and sensors, how to build a smart system, opening  
new possibilities

Speaker: Aurélien Desombre  
[aurelien.desombre@tessolve.com](mailto:aurelien.desombre@tessolve.com)

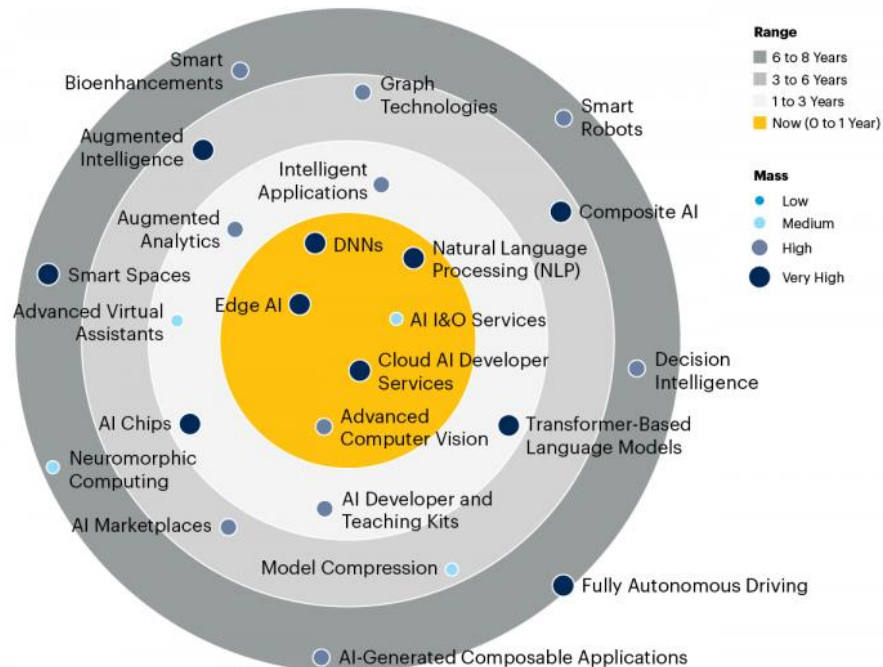
# Digitization TESSOLVE Ecosystem



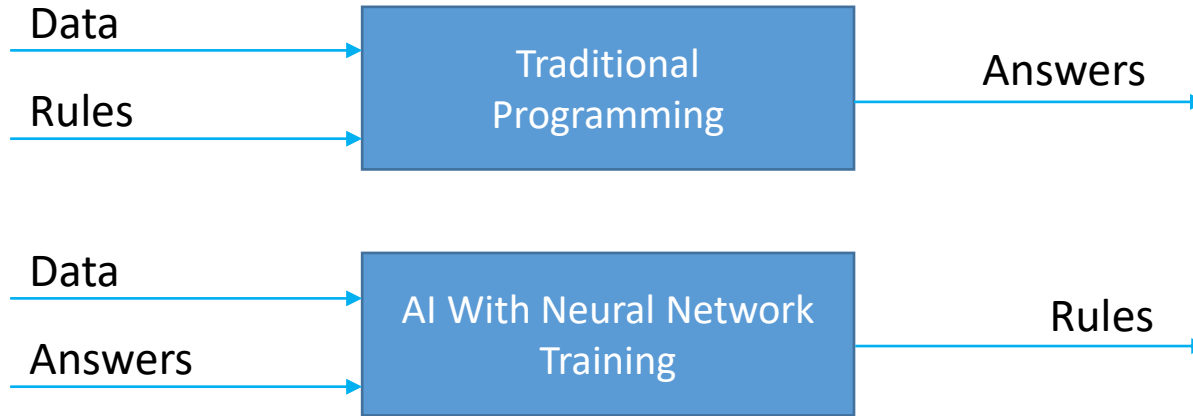
- The Global Artificial Intelligence (AI) Chip Market was valued at USD 9.29 billion in 2019, and is estimated to garner USD 253.30 billion by 2030, at a CAGR of 35.0% during the forecast period between 2020 and 2030.

- Source: [Research and Markets: Artificial Intelligence Chip Market by Chip Type, by Application, by Architecture, by Processing Type, by End User - Global Opportunity Analysis and Industry Forecast, 2020-2030](#)

Impact Radar for Emerging Technologies and Trends: Artificial Intelligence

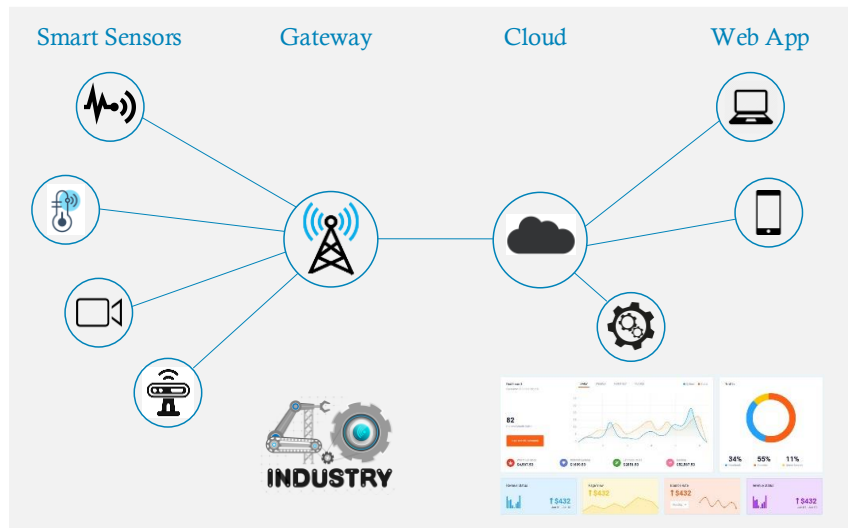


Source: Gartner (October 2020)  
DNNs = deep neural networks, I&O = implementation and operations  
732340\_C



- It enables new scenarios – new use cases which was not able to be programmed with traditional algorithm
- Data Scientists using big data are the key players to feed data of neural network, this input is called a dataset (data classified with a label). This is used for NN training (Training dataset) and NN accuracy check (Test dataset)
- **Note:** it provides a key advantage to major internet players (e.g. GAFA), using your collected data to train their own NN. They are the best positioned to feed NN (with audio, pictures, ...) enable voice recognition, face authentication, object recognition and much more

- **AI algorithms which are processed locally on a hardware device**
  - Input: based on sensor data or local signals: Current/Voltage, T°C, vibration, audio, camera ...
  - Smart: take decision and send events instead of sensor/signal data
- **Advantages**
  - Decentralized AI and Real Time (Low latency)
  - Privacy: data are kept local and not pushed to cloud
  - Reduce communication and data requirements (only events are sent, not data)
  - Power Optimized
  - Cost Effective
- **Physical targets**
  - Usually running on MCU, typically ARM Cortex M4 and M7 families
  - Specific MCU vendors are providing library and ecosystem to transform your NN in C library (from Tensorflow lite for example)
- **EdgeAI use case examples**
  - Predictive Maintenance using vibration and mic sensors for Industry 4.0
  - Digitization of analog metering using OCR
  - Object Detection using Camera for automatic sorting, consumer goods (oven selecting automatically a heating program)
  - Audio Scene classification for intrusion detection or a break of a product for asset management
  - Face recognition for login and authentication



## ● Smart Sensors

- Audio Sensor: Audio Scene Classification (ASC) – e.g.: Normal / Abnormal noise detection
- Camera Sensor : Optical Character Recognition (OCR), Object Recognition (OR), Automated Optical Inspection (AOI)
- Vibration, Temperature, Current & Voltage, Pressure sensor: Predictive Maintenance (PM) for Early failure mode detection

## ● PAN Gateway connectivity:

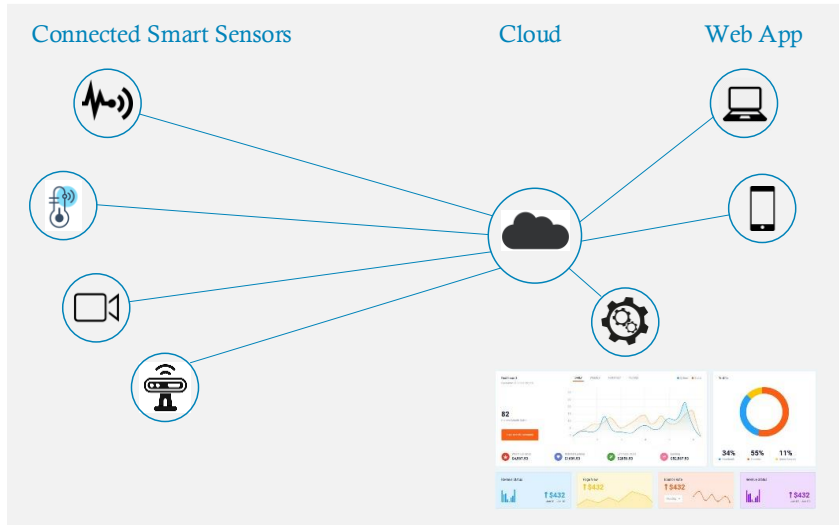
- Wireless Mesh: Open Thread, Zigbee
- Wireless Long Range connection: LoRaWAN

## ● WAN Gateway connectivity:

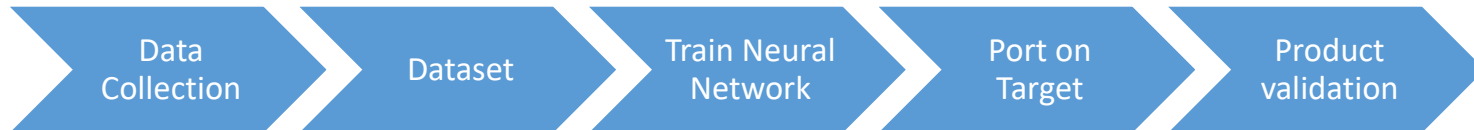
- Multiple gateways
- WAN 4G / Ethernet Gateway

## ● Cloud

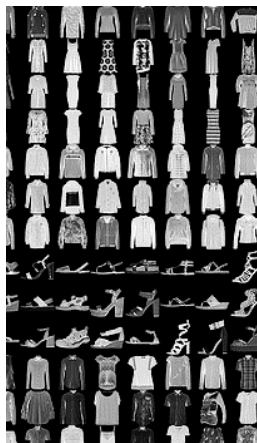
- Private Cloud
- Public Cloud: AWS, Azure, Google cloud



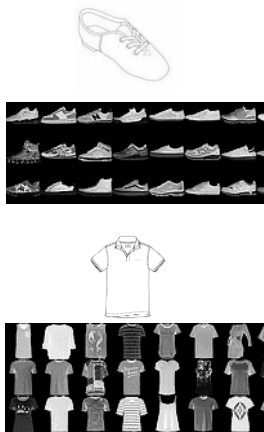
- **Without a gateway**
  - For stand-alone sensors
  - Enabling roaming
- **WAN Gateway connectivity:**
  - LTE-M
  - Nb-IOT
  - LoRaWAN with public infrastructure
  - Sigfox



- Sensor output,
- Image & Video,
- Audio,
- Data,
- Statistics...

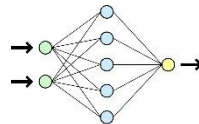


- Labelling
- Dataset reuse or creation



## Train Neural Network

- Deep Learning
- Best NN model selection
- Simulation
- Accuracy
- Resources req.



TensorFlow  
with Keras  
PyTorch

## Port on Target

- Proof Of Concept on final target
- Best Processing Platform
- System Integration
- Customer demo



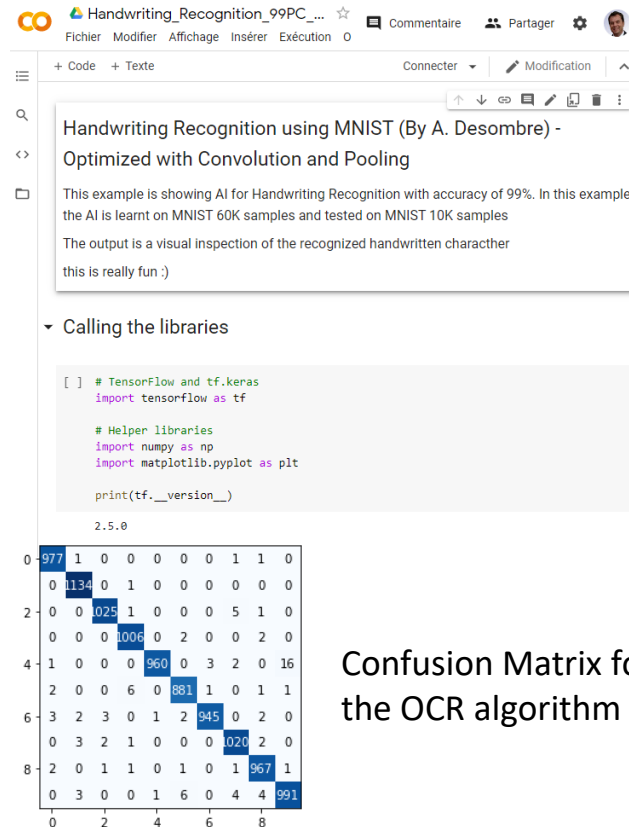
## Product validation

- End user field trials
- Accuracy
  - Reliability
  - Ergonomics
  - Functionality targets





- Dataset is a set of digit images (between 0 and 9) with 60k images for training and 10k for test and accuracy check (each image is having a label between 0 and 9)
- In this example, NN training is done using Tensorflow with Keras on Google Colab
- The scripting language is Python
- The NN training steps are:
  - Importing the training dataset,
  - Defining the NN layers,
  - Compiling the NN,
  - Training the NN
  - Testing the accuracy with a test dataset
- Quality of the dataset and NN model are key for the accuracy of detection – Accuracy output can be a simple graph easy to interpret like the confusion matrix



Handwriting Recognition using MNIST (By A. Desombre) - Optimized with Convolution and Pooling

This example is showing AI for Handwriting Recognition with accuracy of 99%. In this example the AI is learnt on MNIST 60K samples and tested on MNIST 10K samples  
The output is a visual inspection of the recognized handwritten character  
this is really fun :)

Calling the libraries

```
[ ] # TensorFlow and tf.keras
import tensorflow as tf

# Helper libraries
import numpy as np
import matplotlib.pyplot as plt

print(tf.__version__)

2.5.0
```

0	977	1	0	0	0	0	0	1	1	0
0	1134	0	1	0	0	0	0	0	0	0
2	0	0	1025	1	0	0	0	5	1	0
0	0	0	1006	0	2	0	0	2	0	0
4	1	0	0	0	960	0	3	2	0	16
2	0	0	6	0	881	1	0	1	1	1
6	3	2	3	0	1	2	945	0	2	0
0	3	2	1	0	0	0	0	1020	2	0
8	2	0	1	1	0	1	0	1	967	1
0	3	0	0	1	6	0	4	4	991	1
	0	2	4	6	8					

Confusion Matrix for the OCR algorithm

- **AI Pre-trained model or Open AI?**

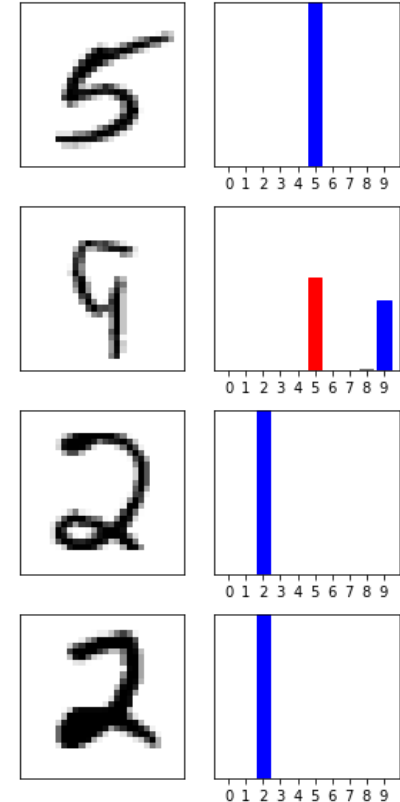
- Pre-Trained AI = AI NN trained before deployment
  - Advantage: All products deployed will have same behaviour, known and qualified accuracy
  - Inconvenient: Non evolutive solution unless system update
- Open AI = AI which will continue to learn on site
  - Advantage: System in permanent evolution
  - Inconvenient: System can learn mistakes / No common behaviour between similar devices

- **Probability of detection and accuracy?**

- AI output is not deterministic, it's a probability of detection between 0 and 1
- The SW algorithm can decide on which probability indicates an accurate detection
- This requires very detailed verification plans and field test trials to validate the correctness of detection in real environment

- **Sensor accuracy and real situation**

- Sensor accuracy and real situation can significantly affect AI accuracy
- For example:
  - A camera used for object detection will be affected by sun illumination
  - A Microphone used with a background noise will not be able to get a glass break event if its input is saturated
- This is a key point for sensor selection, AI model to make it accurate in real field conditions



- Digitization System Architect
- Neural Network and Machine Learning
  - Data Scientists to manage big data and create dataset
  - Neural Network engineers (Training, NN Optimization, test result analysis)
- Power & Memory Optimized Firmware development
  - MCU and ARM Cortex M FW Engineers
- Cloud Architect and SW developer
  - Cloud Architect Engineer
  - Cloud SW Engineer
  - Web UI Developer
- Product Design
  - HW Architect
  - Schematics and PCB Designer
  - Mechanical engineer
  - Product Management: Qualification, Industrialization & Production

Q&A  
&  
THANK YOU



Aurélien Desombre  
Program Director  
Embedded Systems and Digitization  
[aurelien.desombre@tessolve.com](mailto:aurelien.desombre@tessolve.com)