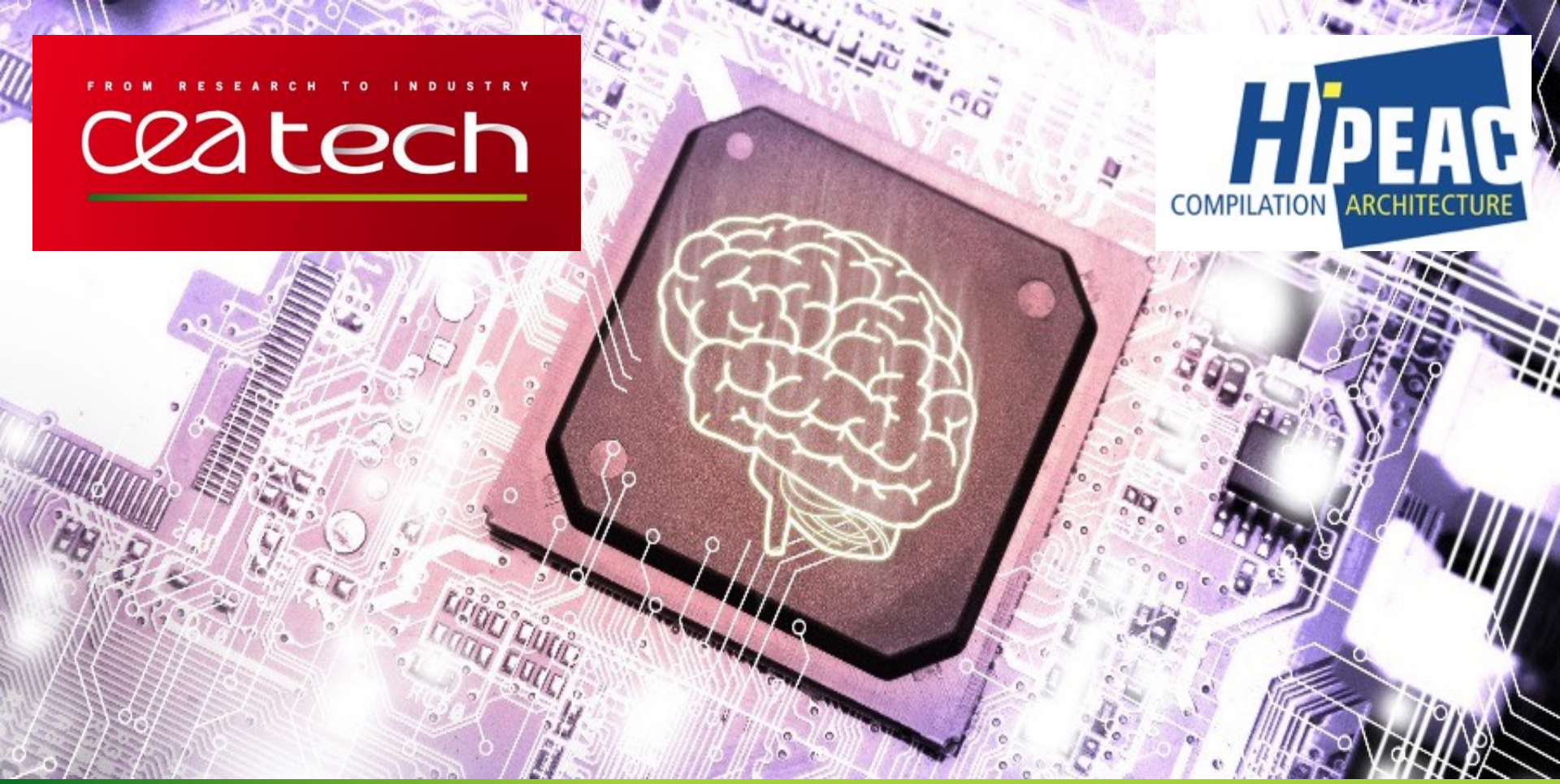



# COGNITIVE CYBER PHYSICAL SYSTEMS: NEW ERA FOR EMBEDDED SYSTEMS

**Marc Duranton**
CEA Fellow
Commissariat à l'énergie atomique et aux énergies alternatives

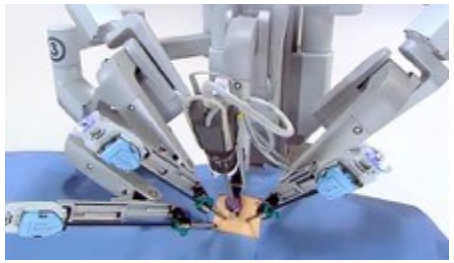
Friday September 14th, 2018

*“The best way to predict the future is to invent it.”*

Alan Kay

# Entering in
# Human and machine collaboration era








**ENABLED BY ARTIFICIAL INTELLIGENCE**
**(AND DEEP LEARNING)**

# CYBER PHYSICAL ENTANGLEMENT

- Computer are not anymore a “PC”
- They get input from the real world with sensors, not anymore with keyboards
- They interact with the world without screen
  - Thanks to progress in Deep Learning for example
- They are everywhere, morph in our environment

Smart sensors

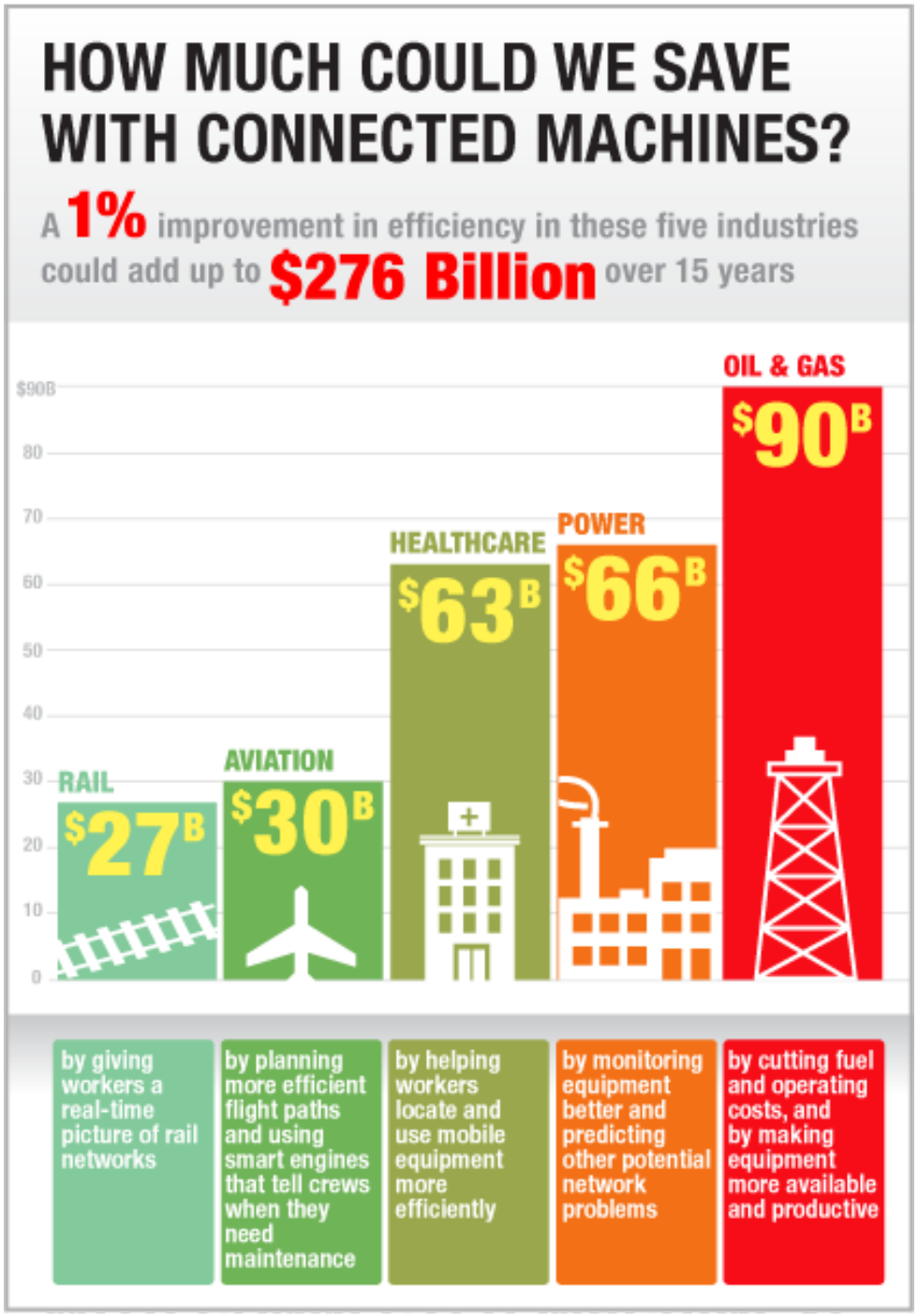
Internet of Things

Big Data

Data Analytics / Cognitive computing

Cloud / HPC

New services



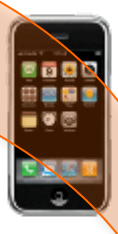

## ECONOMICAL DRIVE OF CONNECTED THINGS: BETTER EFFICIENCY IN RESOURCES AND ENERGY

# ENABLING EDGE INTELLIGENCE
## C²PS: COGNITIVE ( CYBERNETIC* AND PHYSICAL ) SYSTEMS








*Enabling Intelligent data processing at the edge:*
*Fog computing*
*Edge computing*
*Stream analytics*
*Fast data…*

Smart sensors

Internet of Things

Big Data

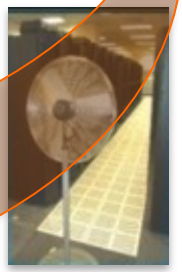
Data Analytics / Cognitive computing

Cloud / HPC

New services

*Cyber Physical Entanglement*

**Processing, Abstracting Understanding *as soon as possible***

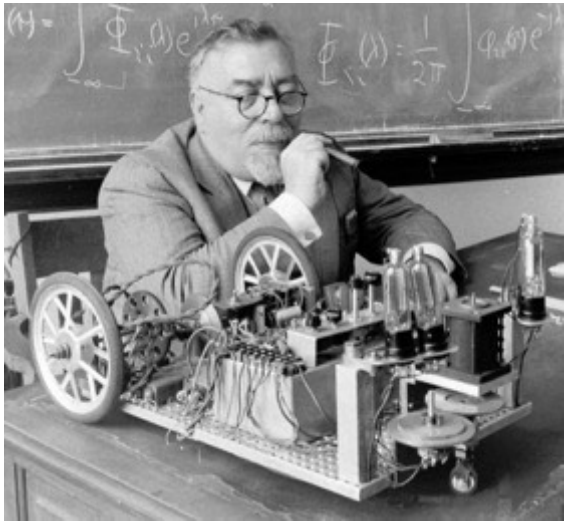



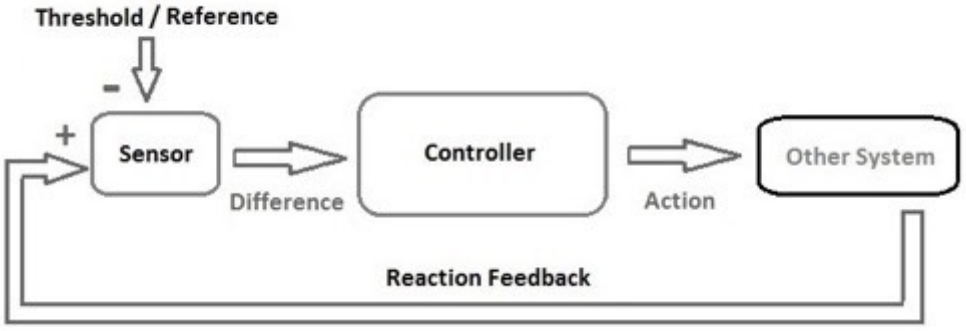
**True** collaboration between edge devices and the HPC/cloud

Transforming **data** into ***information as early as possible***

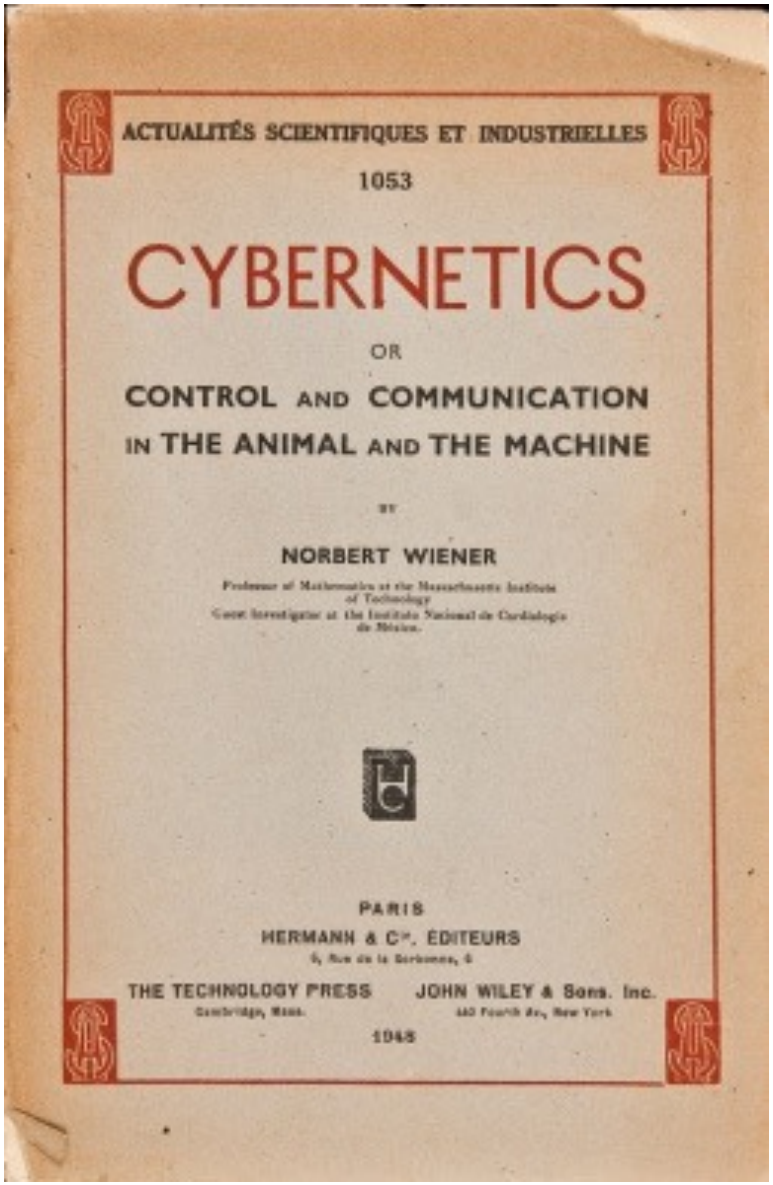
** As defined by Norbert Wiener: how humans, animals and machines control and communicate with each other.*

# 1948: NORBERT WIENER

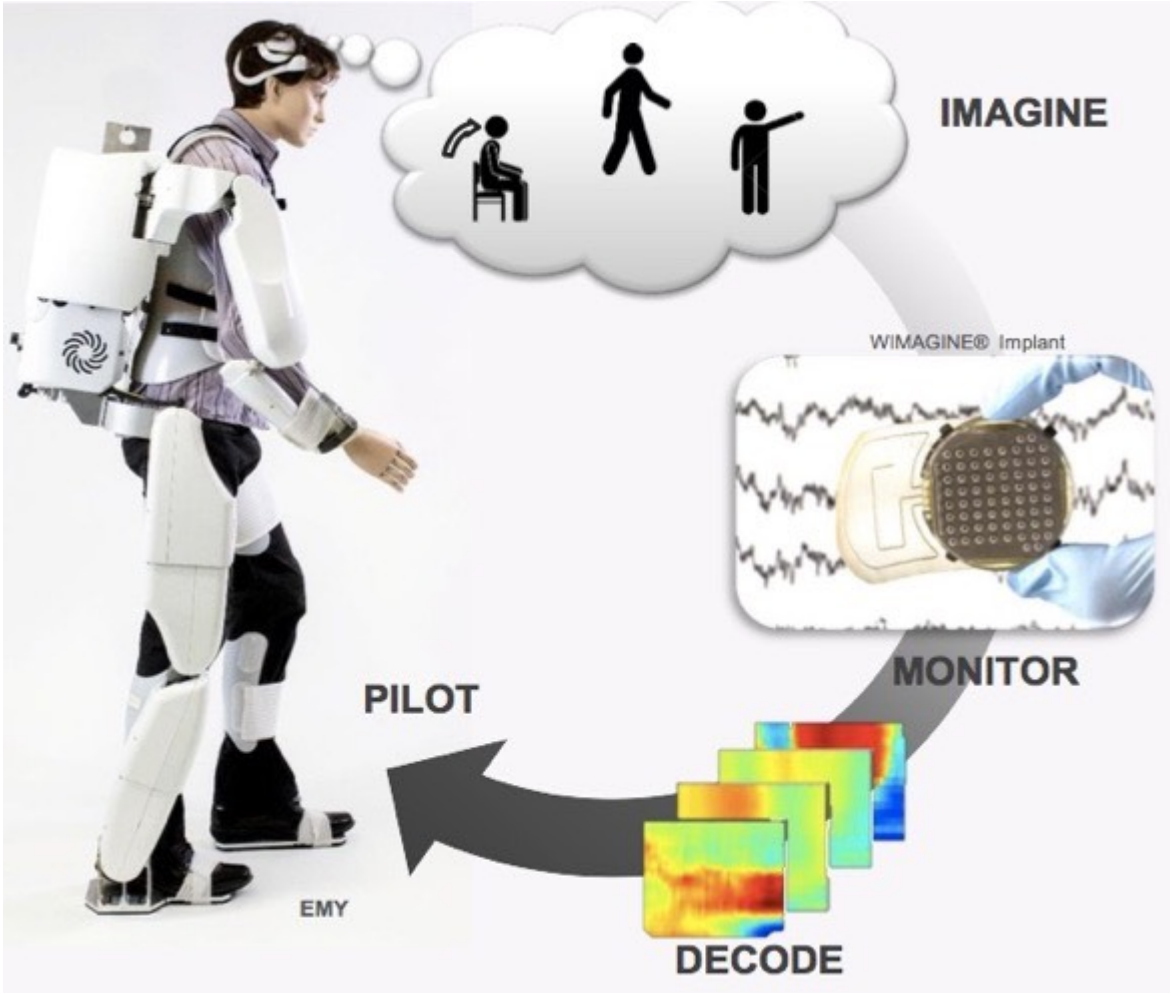





A Cybernetic Loop



# LOOKING FORWARD… EXAMPLE OF A CPS SYSTEM




Direct Brain Computer Interface (BCI)

*Here allowing a paraplegic to walk again…*

*One current limitation:* ***Required processing power – need supercomputer in a box***

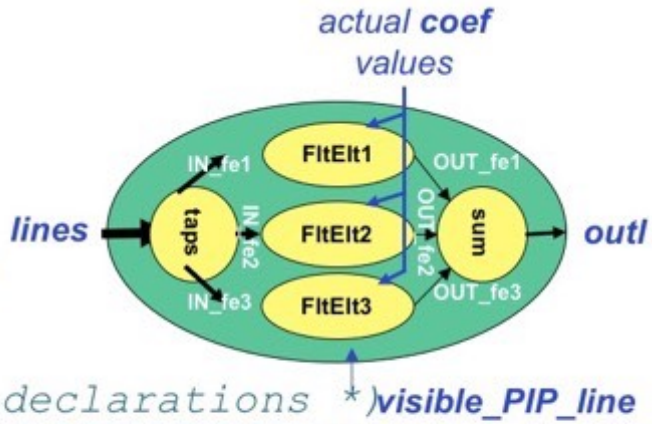
From CEA-Clinatec

## BUT COMPUTING SYSTEMS WERE NOT DESIGNED FOR CPS SYSTEMS

In nearly all hardware and software of computing systems:

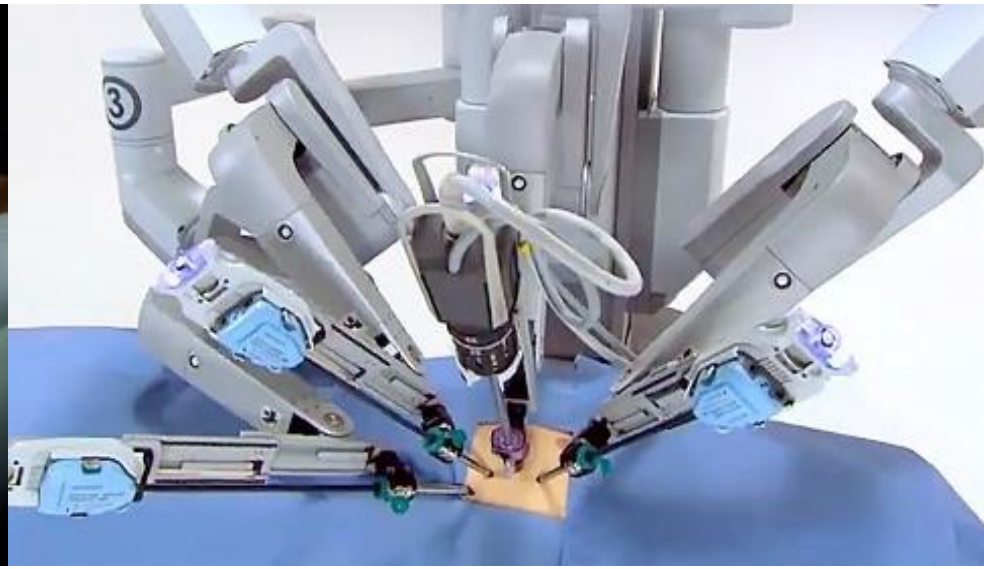
- Time is abstracted or even not present at all
  - Very few programming languages can express time or timing constraints
- All is done to have the best average performance, not predictable performances
  - Caches, out of order execution, branch prediction, speculative execution,…
  - (Hidden) compiler optimization, call to (time) unspecified libraries
- Energy is also left out of scope
  - This can have impact on data movement, optimizations
- Interaction with external world are second priorities vs. computation
  - Done with interrupts (introduced as an ***optimization***, eliminating unproductive waiting time in polling loops) which were design to be ***exceptional events***…
- Etc.

# EXAMPLE OF “TIME” AWARE PROGRAMMING MODEL




```
node vert_scaler (param float coefs[N][64],
                  in pixel lines[N][240],
                  out pixel outl[240])
{ pixel IN_fe1[3][240] ... (* other IN_feN declarations *)
  pixel OUT_fe1[240] ... (* other OUT_feN declarations *)
  index i [240]
   lines -> taps -> IN_fe1, IN_fe2, IN_fe3
   IN_fe1 -> FltElt1(coefs) -> OUT_fe1
   IN_fe2 -> FltElt2(coefs) -> OUT_fe2
   IN_fe3 -> FltElt3(coefs) -> OUT_fe3
   outl[i] = (OUT_fe1[i] + OUT_fe2[i] + OUT_fe3[i])/3
}
. . .
pixel N_port_buffer[N][240]
pixel IN_YUV2RGB[240]
extern clock frame_clock 30 Hz
clock visible_output_line 1080@frame_clock
clock visible_PIP_line visible_line_clock[500..619]
. . .
N_port_buffer -> vert_scaler(some_coefs) -> IN_YUV2RGB every
  visible_PIP_line
```

# Trust is key for critical applications



- **Beyond predictability by design** and beyond worst-case execution time (WCET)
- Capability to **build trustable systems from untrusted components**
- **Mastering trustability for complex distributed systems**, composed of black or grey boxes

# Embedded intelligence needs local high-end computing


System should be autonomous to make good decisions in all conditions

And **should not consume** most **power** of an electric car!

# Embedded intelligence needs local high-end computing



***Privacy*** will impose that some **processing should be done locally** and not be sent to the cloud.

With minimum power and wiring!

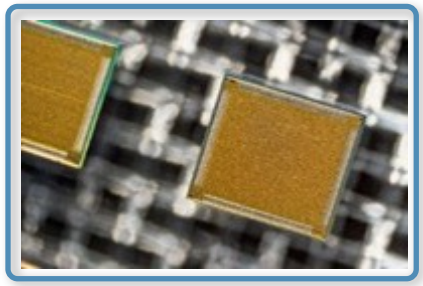
# Detecting elderly people falling in their home

## Exemple from Global Sensing Technologies


**CEA's P-Neuro: Ultra low power local processing** detecting lying people in a room

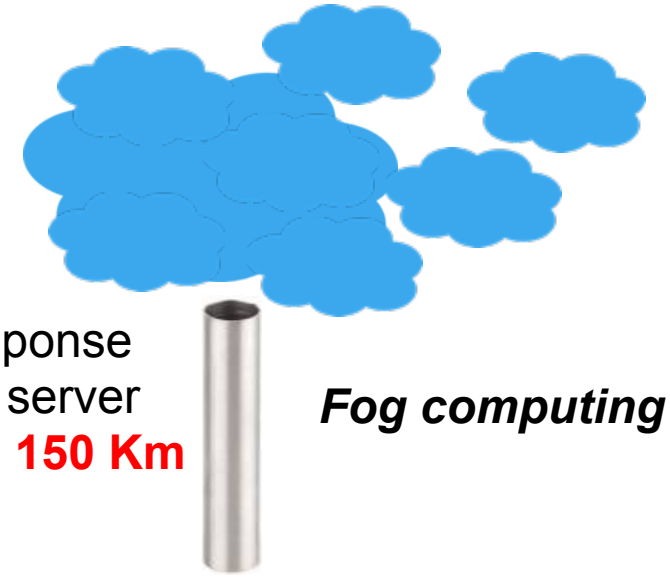
Raw data (before post-processing):

- **Standing**
- **Crouching**
- **Lying**



# Embedded intelligence needs local high-end computing

And if you need a response in less than **1ms**, the server has to be in less than **150 Km** ( the speed of light is 299 792 458 m/s )



***Fog computing***

Dumb sensors

Smart sensors: Streaming and distributed data analytics

***Bandwidth (and cost)*** will require more **local processing**

# *ENERGY OF SMART LIGHT BULBS*





**Server in Singapore**

- **0 W** power off
- **100%** energy for the light bulb

# *ENERGY OF SMART LIGHT BULBS*


- **0 W** power off
- **100%** energy for the light bulb

- Energy for the smartphone
- Wifi energy
- Home router energy
- Energy for routing to Singapore
- Energy of the server for processing
- Energy for routing from Singapore
- Home router energy
- Wifi Energy
- Energy for the light bulb electronics

All this multiplied by the number of smart light bulbs…
(And there are **2.5B light bulbs** - not yet smart - sold each year…)

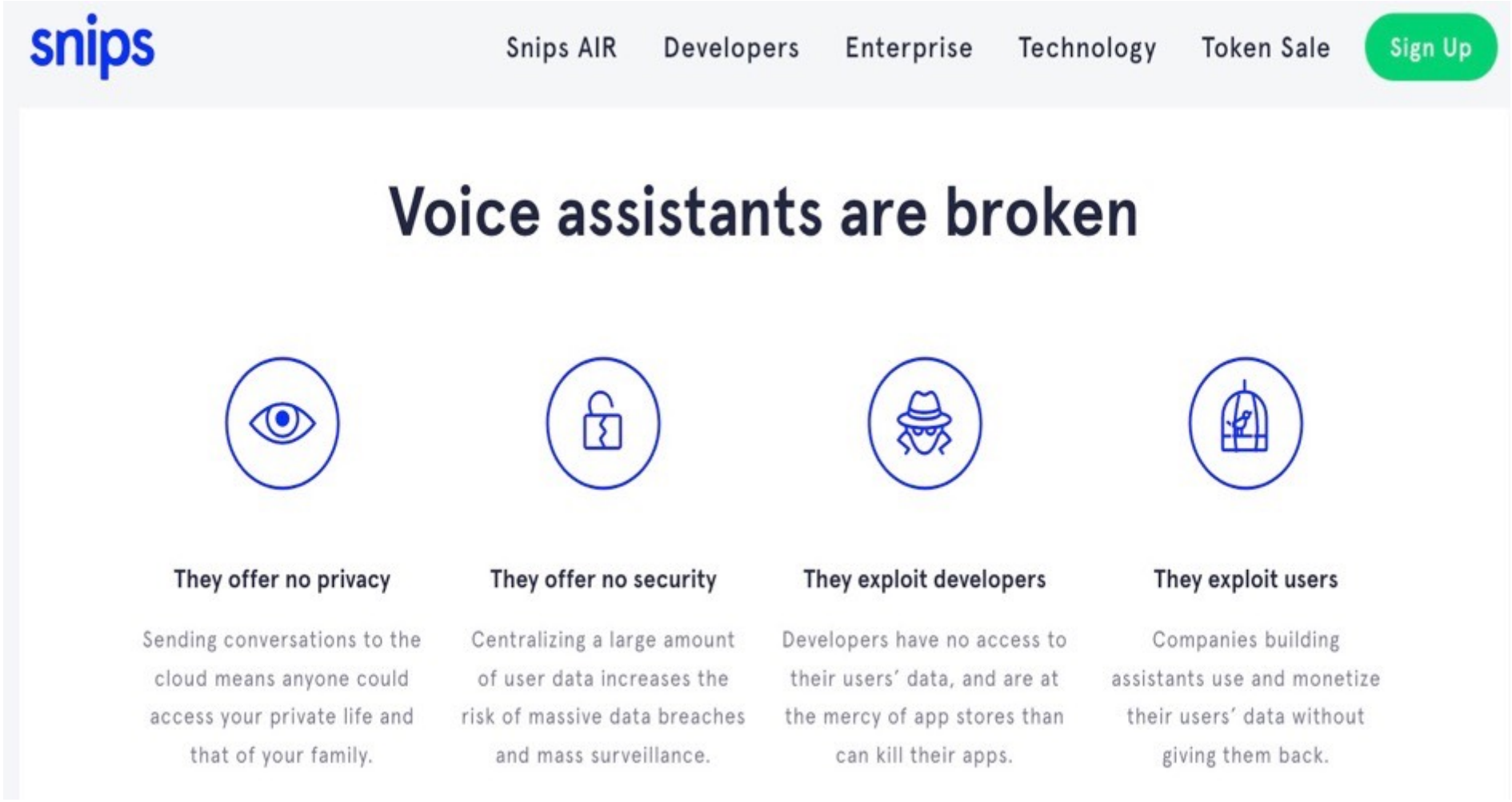
# *ENERGY OF SMART LIGHT BULBS AND WITH THE PERSONAL ASSISTANTS....*

Google Assistant

Apple Siri

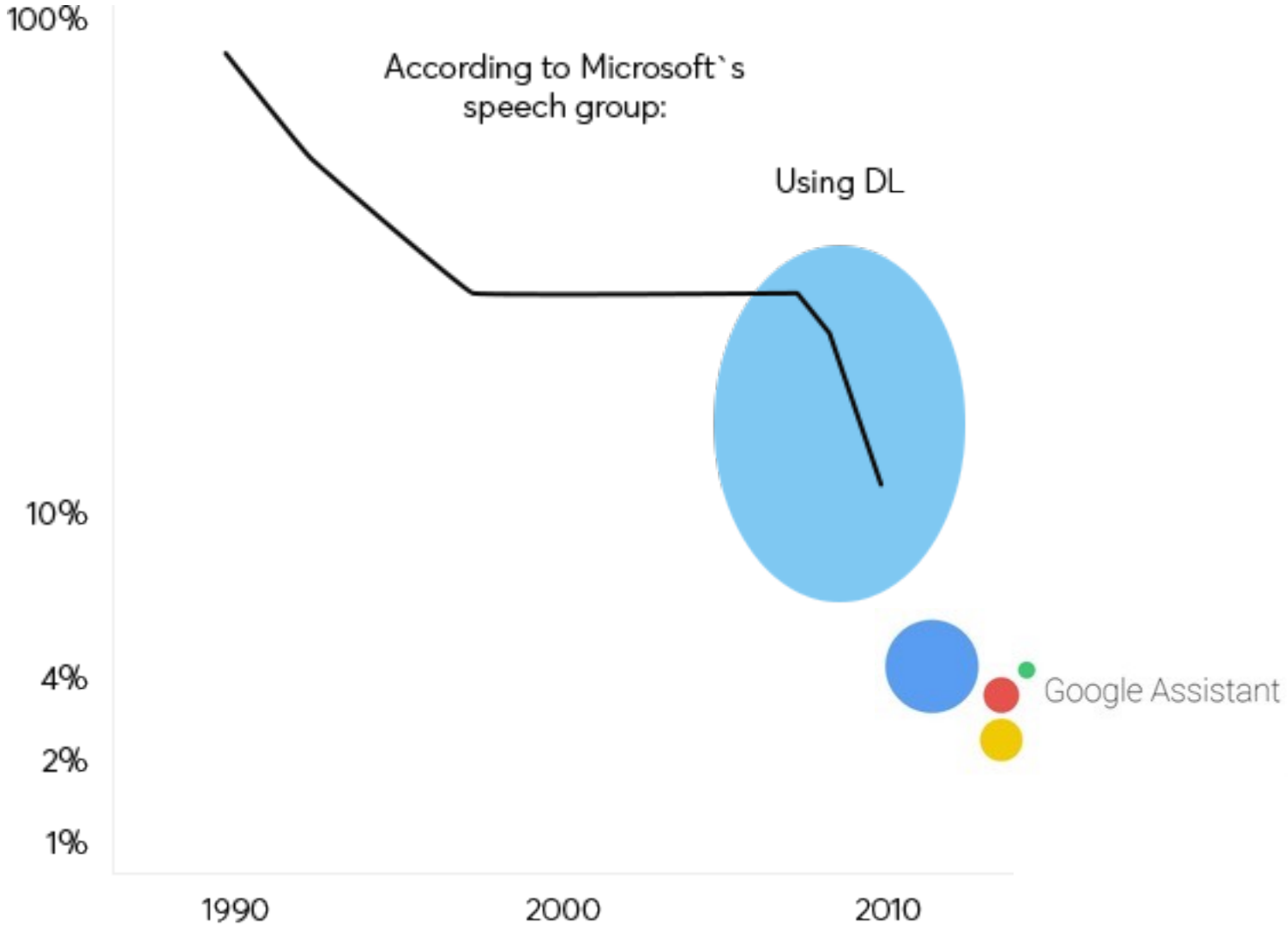

Amazon Alexa
with Zigbee

# *ENERGY OF SMART LIGHT BULBS AND WITH THE PERSONAL ASSISTANTS....*




From https://snips.ai/

# DEEP LEARNING AND VOICE RECOGNITION

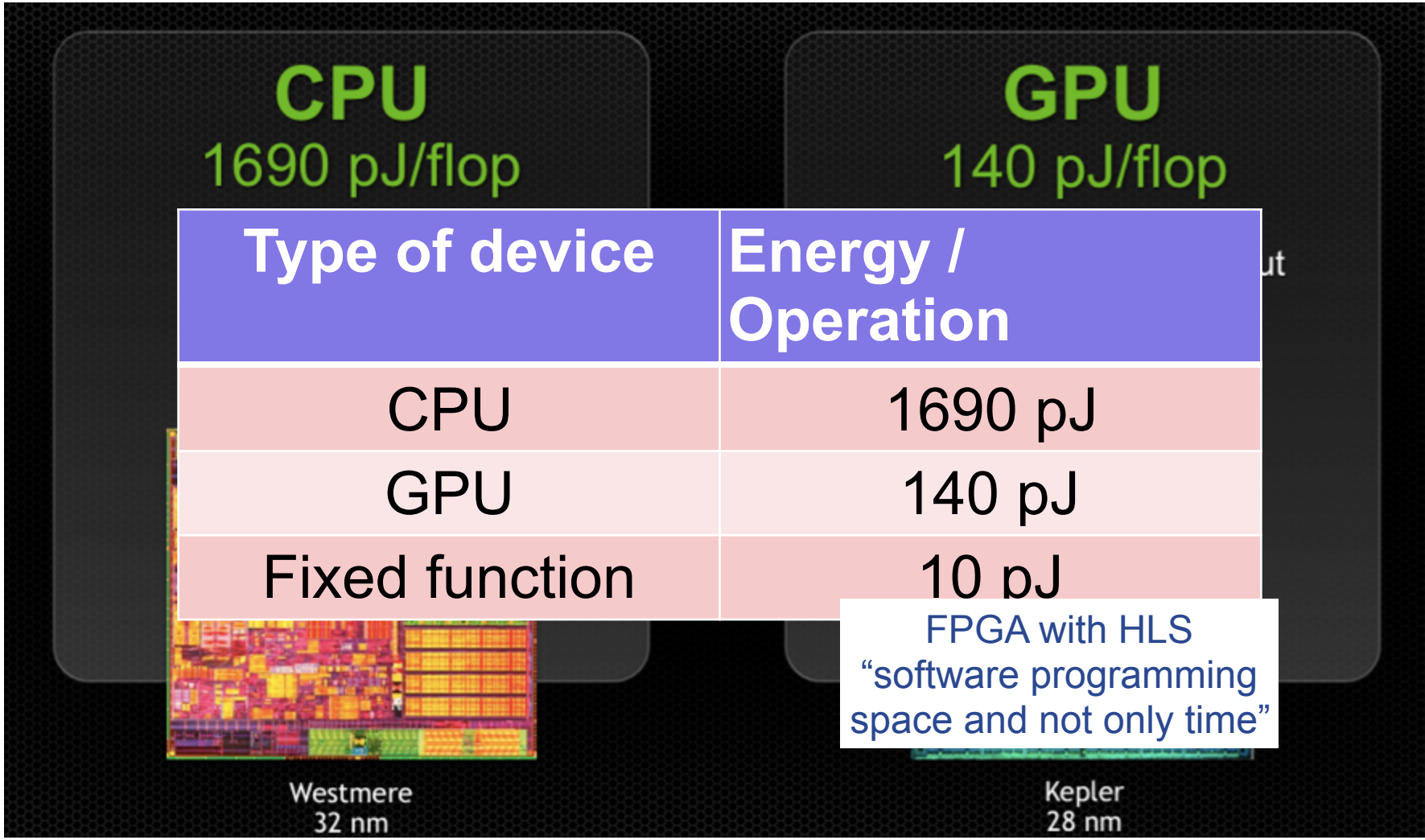



Deep Learning in **Speech Recognition**

# DEEP LEARNING AND VOICE RECOGNITION

" The need for TPUs really emerged about six years ago, when we started using computationally expensive deep learning models in more and more places throughout our products. The computational expense of using these models had us worried. If we considered a scenario where people **use Google voice** search for just **three minutes a day** and we ran deep neural nets for our speech recognition system on the processing units we were using, **we would have had to double the number of Google data centers**!"

**[https://cloudplatform.googleblog.com/2017/04/quantifying-the-performance-of-the-TPU-our-first-machine-learning-chip.html]**




| Type of device | Energy / Operation |
|---|---|
| CPU | 1690 pJ |
| GPU | 140 pJ |
| Fixed function | 10 pJ |

FPGA with HLS
“software programming space and not only time”

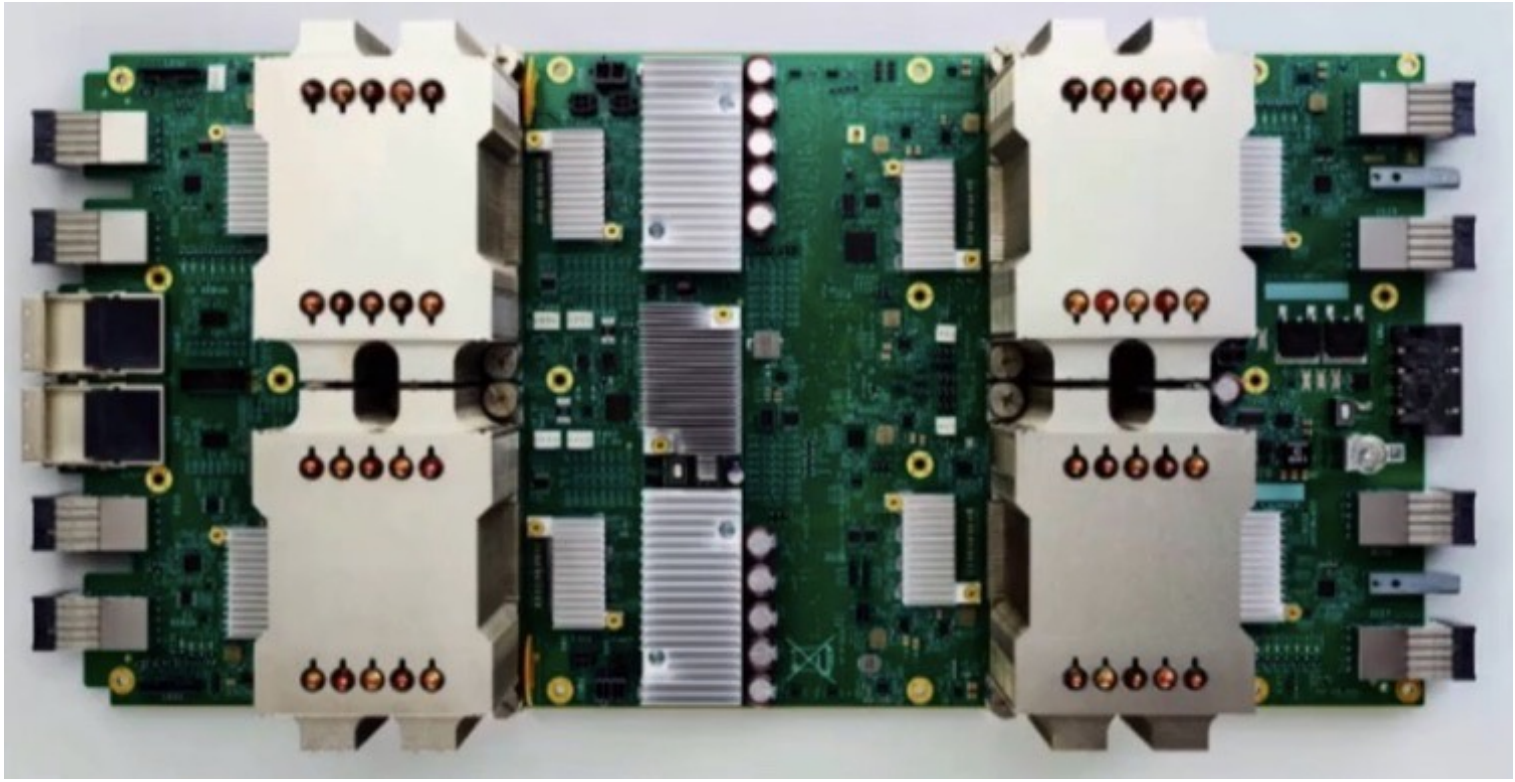
Source from Bill Dally (nVidia) « Challenges for Future Computing Systems » HiPEAC conference 2015



23

## 2017: GOOGLE'S CUSTOMIZED HARDWARE…

… required to increase energy efficiency
with **accuracy adapted to the use (e.g. float 16)**



Google's TPU2 : training and inference in a **180 teraflops$_{16}$** board
(over 200W per TPU2 chip according to the size of the heat sink)

## 2017: GOOGLE'S CUSTOMIZED TPU HARDWARE…

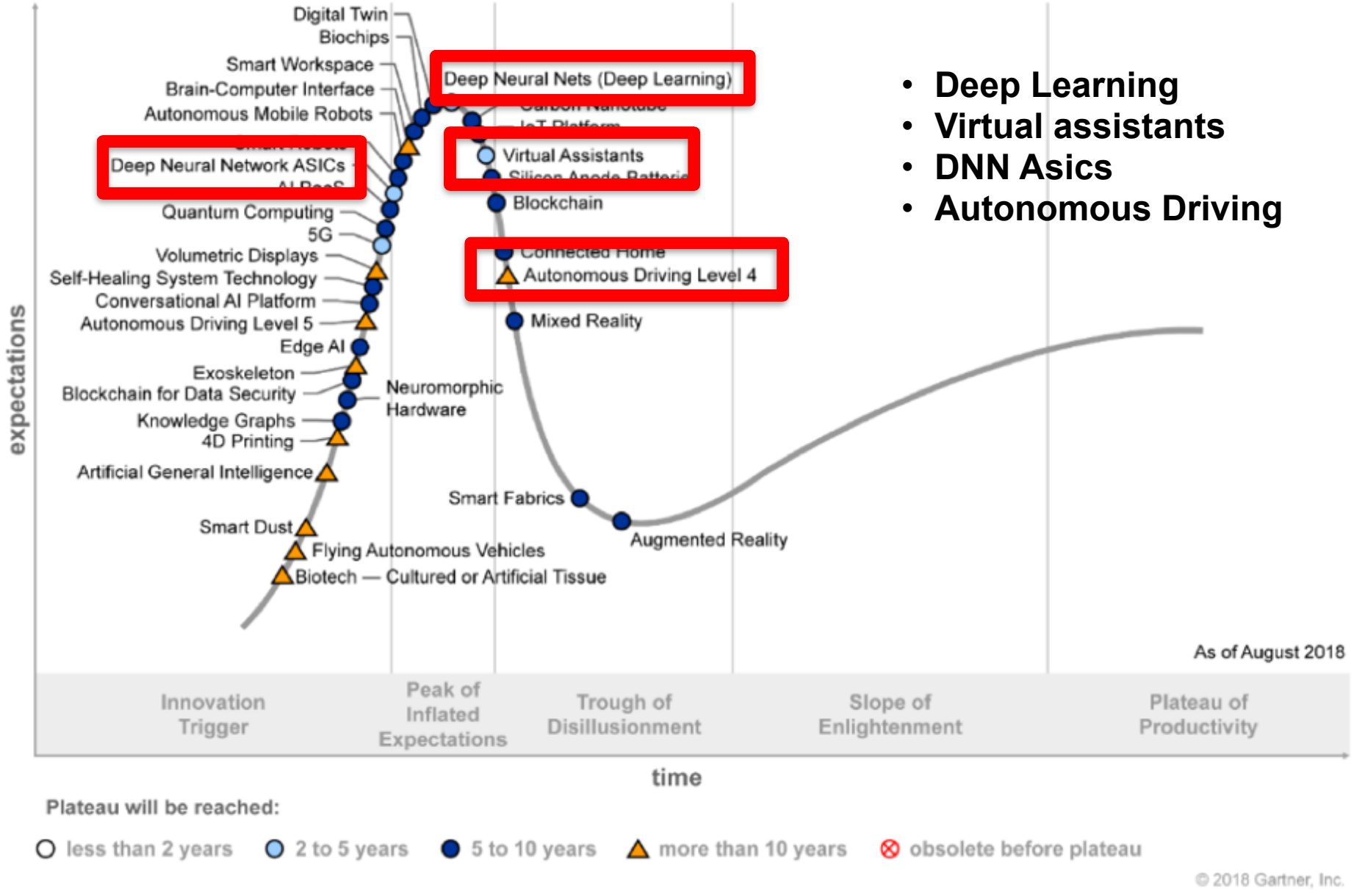
… required to increase energy efficiency
with accuracy adapted to the use (e.g. float 16)


Google's TPU2 : 11.5 $petaflops_{16}$ of machine learning number crunching
(and guessing about 400+ KW…, 100+ $GFlops_{16}$/W)

From Google

Peta = $10^{15}$ = million of milliard

# The Hype cycle - 2018

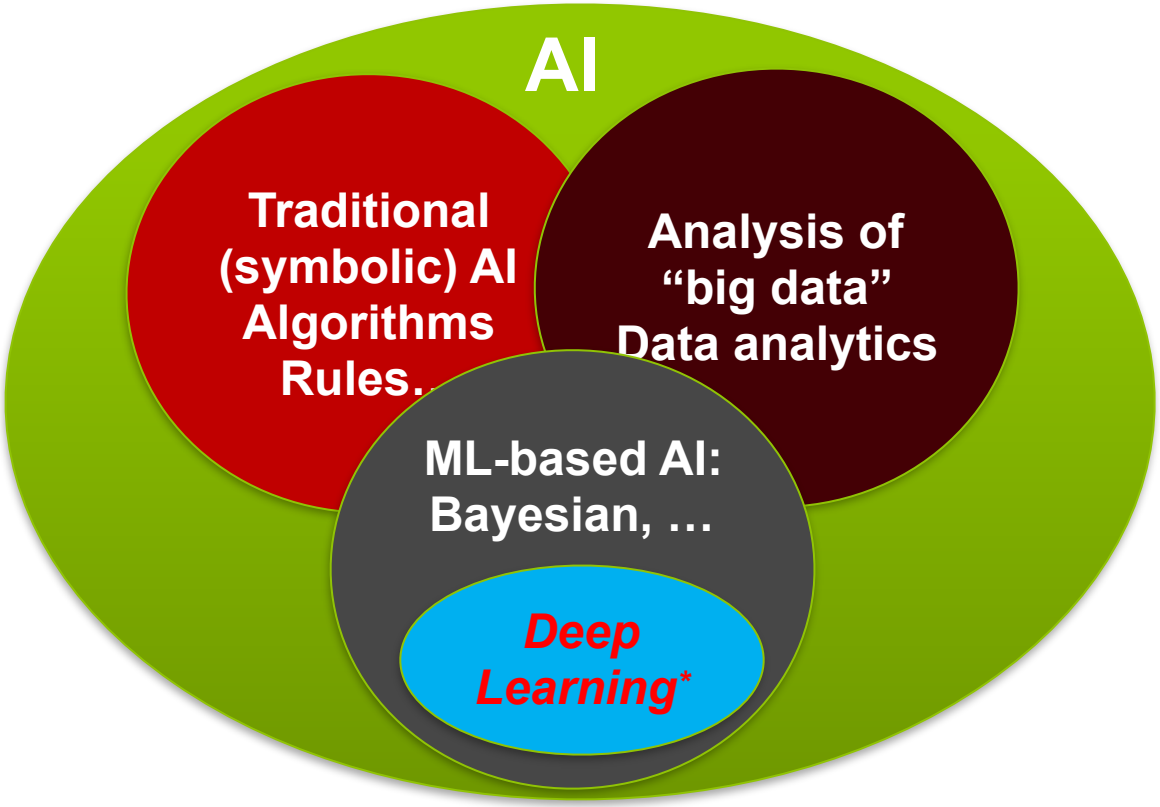



- **Deep Learning**
- **Virtual assistants**
- **DNN Asics**
- **Autonomous Driving**

"***As soon as it works, no one calls it AI anymore***"

John McCarthy

## KEY ELEMENTS OF *ARTIFICIAL INTELLIGENCE*

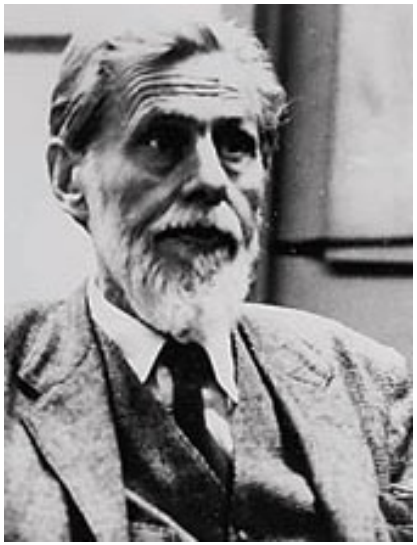



*** Reinforcement Learning, One-shot Learning, Generative Adversarial Networks, etc…**

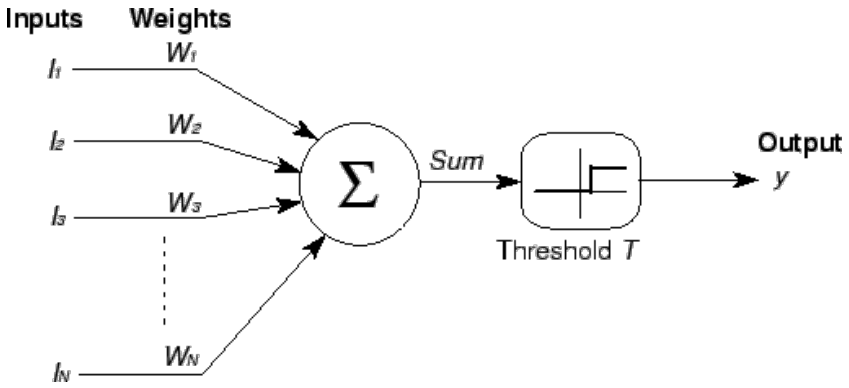
From Greg. S. Corrado, Google brain team co-founder:

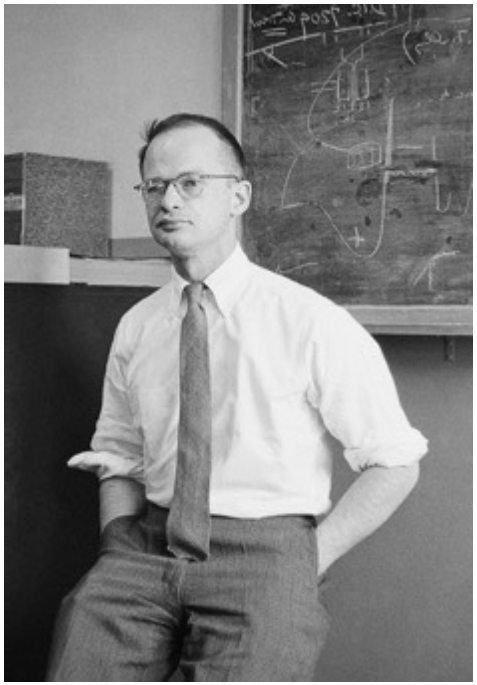
- *"Traditional AI systems are **programmed** to be clever*
- *Modern ML-based AI systems **learn** to be clever.*

# 1943: MCCULLOCH AND PITTS



Neurophysiologist and cybernetician



Logician workingin the field of computational neuroscience

They laid the foundations of formal Neural Networks

# 1943: MCCULLOCH AND PITTS

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

## A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

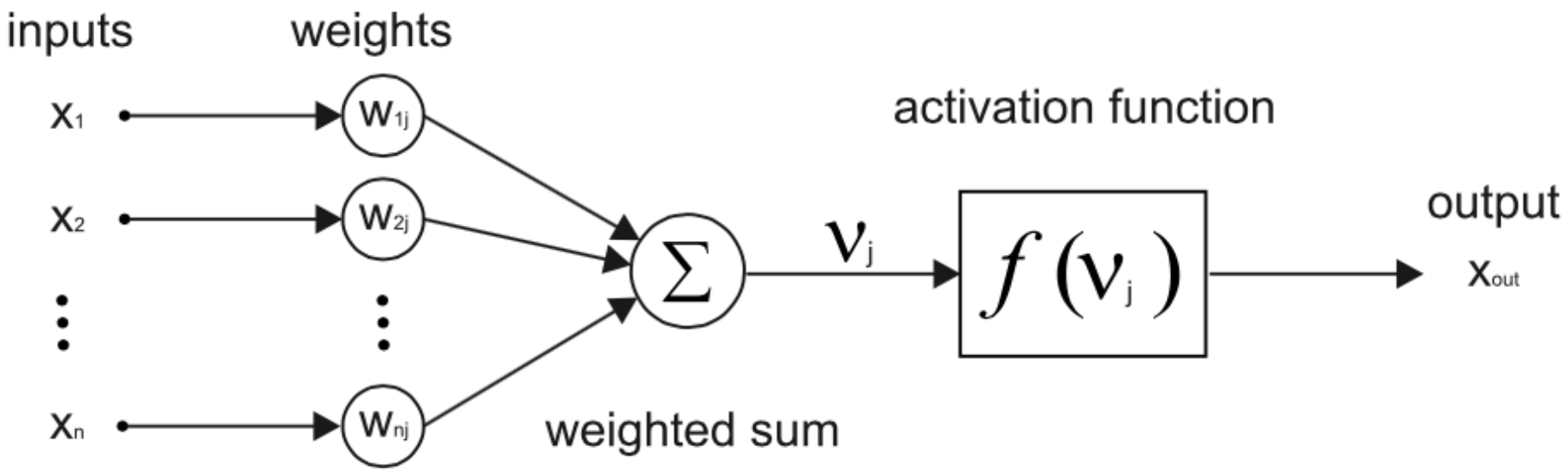
WARREN S. MCCULLOCH AND WALTER PITTS

FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

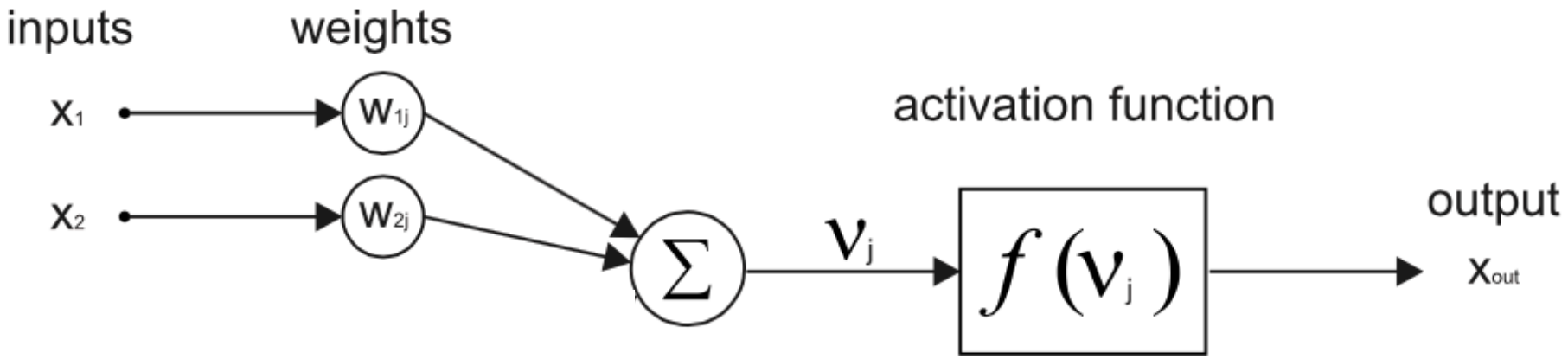
# WHAT IS A NEURAL NETWORK?

**A « formal » neuron:**



# WHAT IS A NEURAL NETWORK?

**The « formal » neuron:**



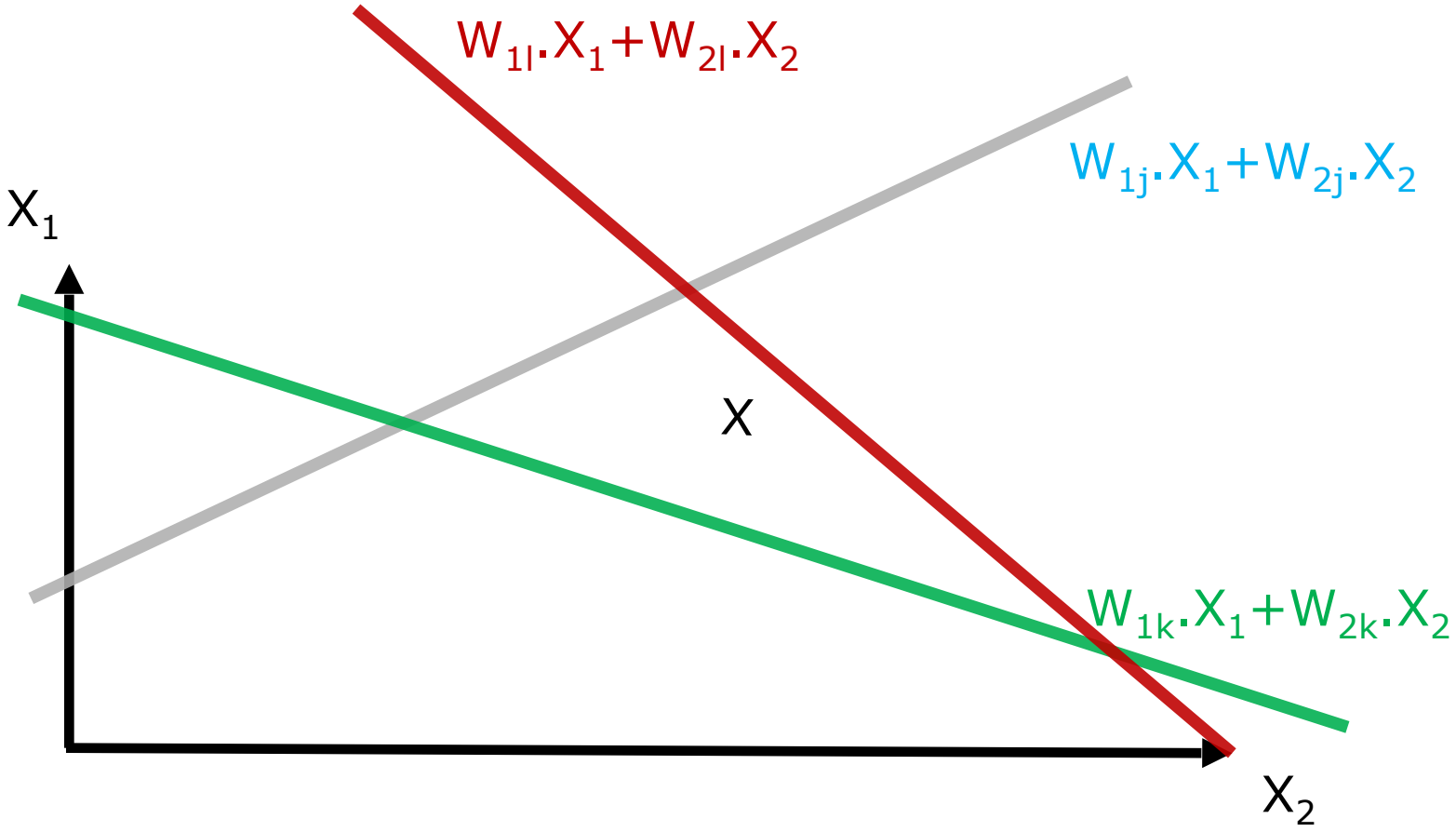

$$V_j = W_{1j}.X_1 + W_{2j}.X_2$$

It is the definition of an hyperplane

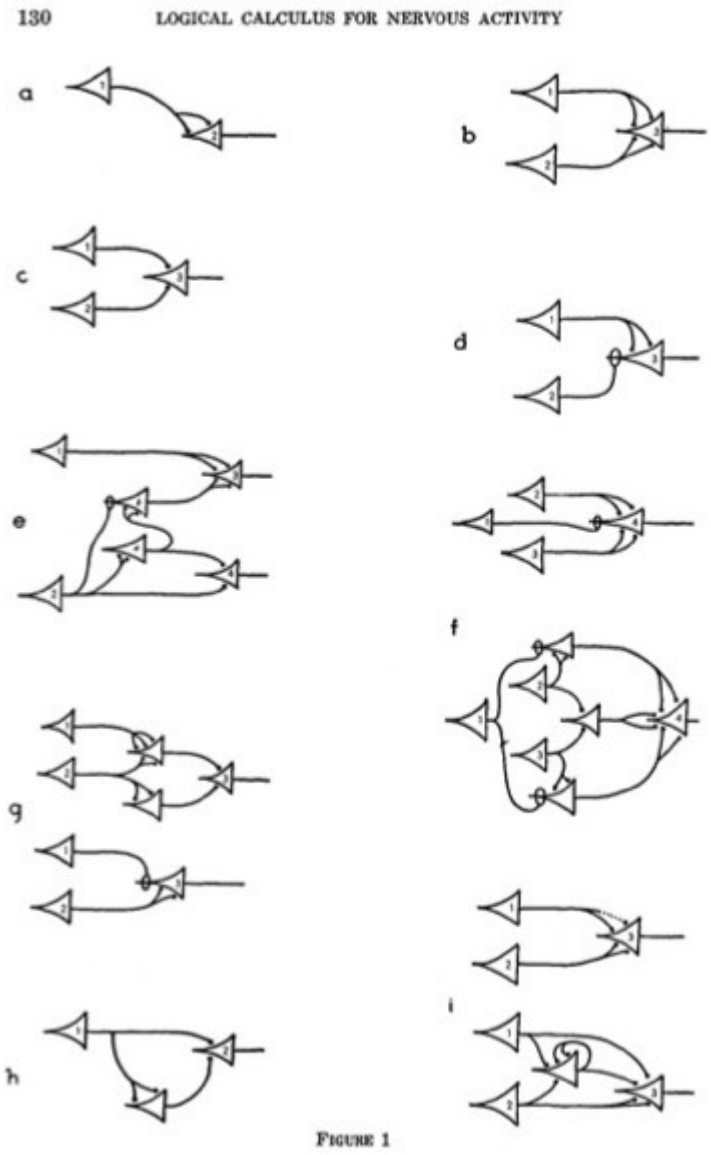
$F(V_j)$ non linear $\in\{-1,1\}$ e.g. sign() function

$X(X_1,X_2)$ is “above” or “below” the hyperplane

# WHAT IS A NEURAL NETWORK?

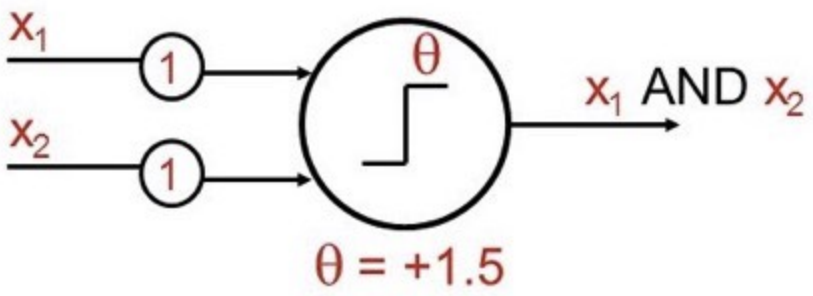


# WHAT IS A NEURAL NETWORK?

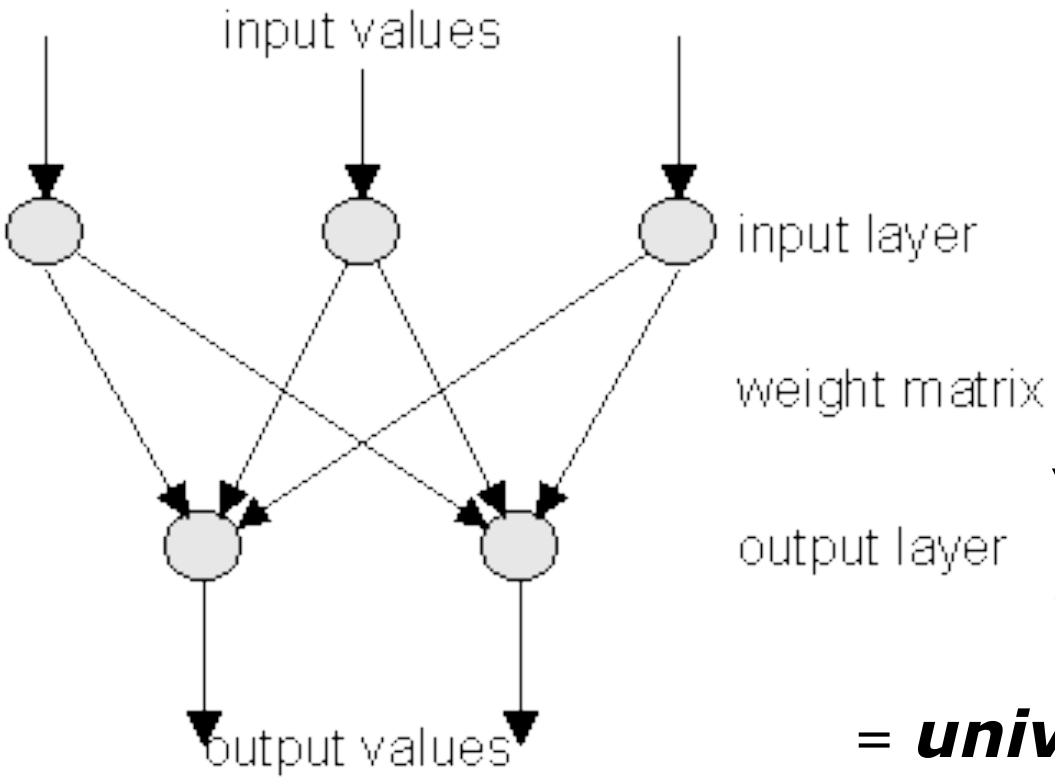



**Association of neurons to make logical functions.**
**Example: AND gate**

| IN 1 | IN 2 | OUT |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



# MULTILAYER NETWORK

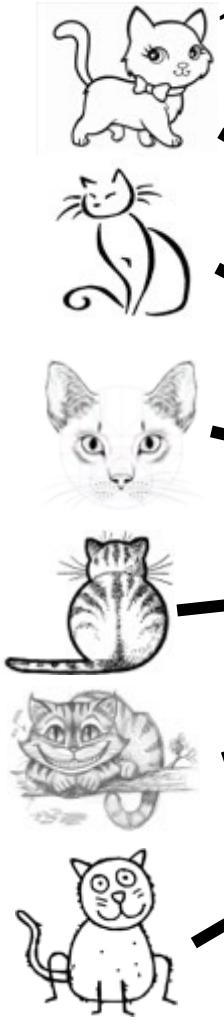



Hyperplane separation

"logic" composition
Warren McCulloch and Walter Pitts, 1943

= ***universal approximator***

# WHY DOES DEEP LEARNING WORK SO WELL?*



1 megapixel 256 grey level image

$256^{1000000}$ possible images

*It can be done by Neural Networks:* ***Universal approximator*** *made with neural networks of* ***finite*** *size*

Function ?

It is a cat

It is NOT a cat

For each possible image, we wish to compute the probability that it depicts a cat. Then, the function is defined by a list of $256^{1000,000}$ probabilities

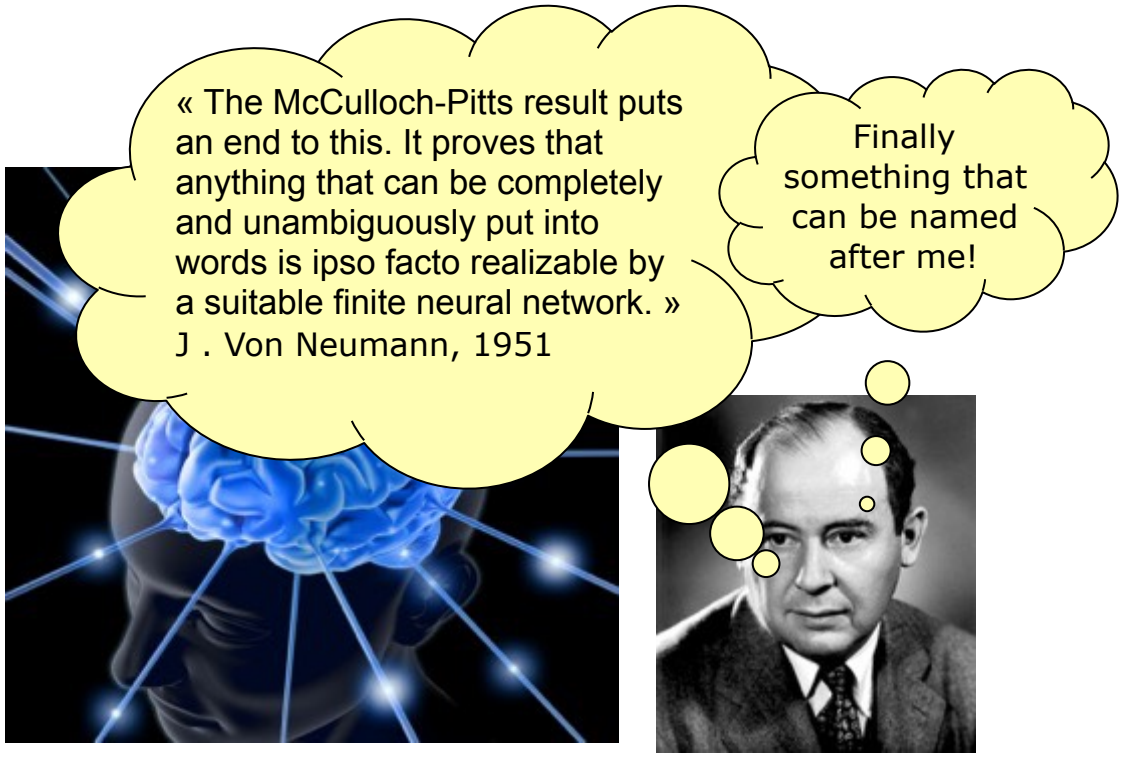
i.e., way more numbers than there are atoms in our universe (about $10^{78}$ to $10^{82}$ <<< $10^{2,408,240}$).

- Work of Henry W. Lin (Harward) , Max Tegmark (MIT), and David Rolnick (MIT) https://arxiv.org/abs/1608.08225

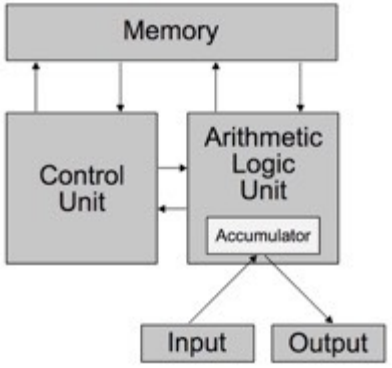
## BUT WHAT IS THE TRUE VON NEUMANN ARCHITECTURE?

In “First Draft of a Report on the EDVAC,” the first published description of a stored- program binary computing machine - the modern computer, John von Neumann suggested modelling the computer after Pitts and McCulloch’s neural networks.

# BUT WHAT IS THE TRUE VON NEUMANN ARCHITECTURE?

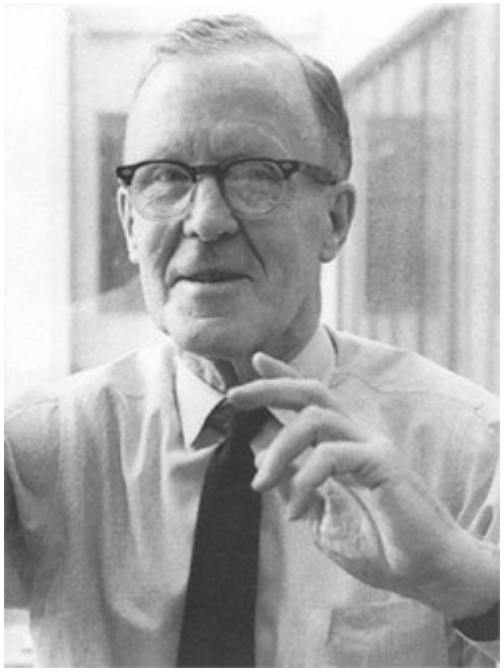



**But technology was not ready in the 50's, leading to realization with sequential processing**



Hebb's rule or Hebbian theory: an explanation for the adaptation of neurons in the brain during the learning process

**Basic mechanism for synaptic plasticity**: an increase in synaptic efficacy arises from the presynaptic cell's repeated and persistent stimulation of the postsynaptic cell.



Psychologist, working in the area of neuropsychology

Introduced by Donald Hebb in his 1949 book « *The Organization of Behavior* »

## 1980: KUNIHIKO FUKUSHIMA

# The first Deep Neural Network, inspired by the visual cortex.

**Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position**

Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

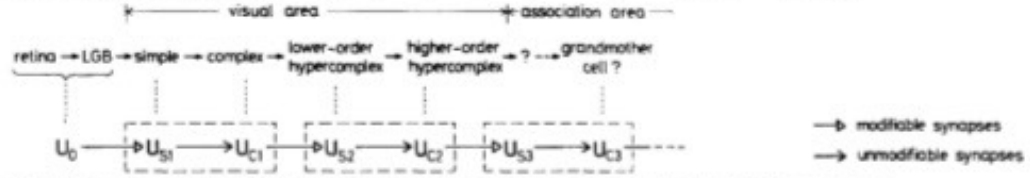



Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

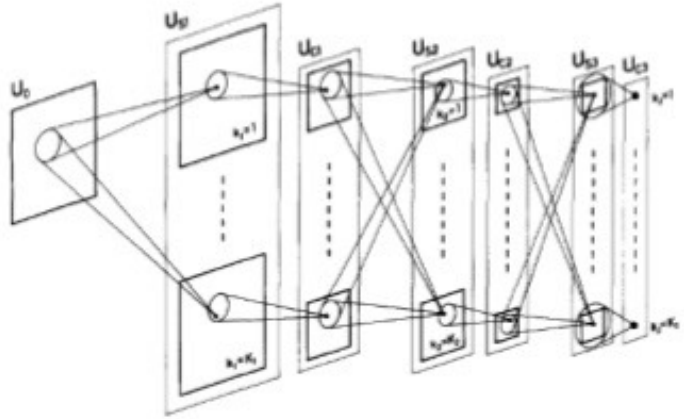


Fig. 2. Schematic diagram illustrating the interconnections between layers in the neocognitron

Biol. Cybernetics 36, 193–202 (1980)

## AROUND 1986: GEOFFREY HINTON

He was one of the first researchers who demonstrated the use of **generalized back-propagation algorithm** for training multi-layer neural networks.

He co-invented **Boltzmann machines** with David Ackley and Terry Sejnowski.

His other contributions to neural network research include distributed representations, time delay neural network, mixtures of experts, Helmholtz machines and Product of Experts

He is now working for Google.

Cognitive psychologist and computer scientist

## AROUND 1985: YANN LE CUN

In 1985, he proposed and published (in French), an early version of the learning algorithm known as **error backpropagation**
Near 1989, he developed a number of new machine learning methods, such as a biologically inspired model of image recognition called **Convolutional Neural Networks**, the "Optimal Brain Damage" regularization methods, and the Graph Transformer Networks method which he applied to handwriting recognition and OCR.

The **bank check recognition system** that he helped develop was widely deployed by NCR and other companies, reading over 10% of all the checks in the US in the late 1990s and early 2000s.

In 2013, LeCun became the first director of Facebook AI Research in New York City.

# 1990'S NEUROCOMPUTERS...

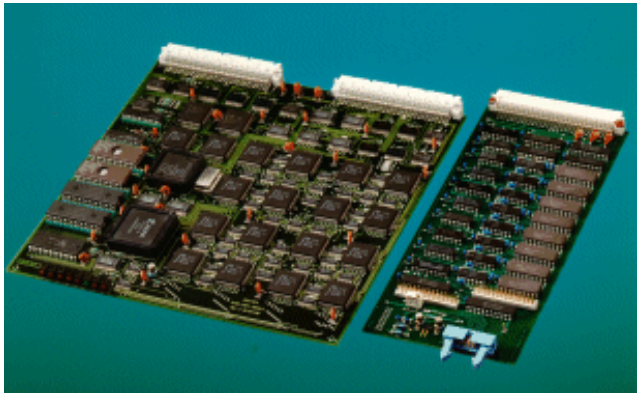
## Philips : L-Neuro

- 1st Gen 16 PEs 26 MCps (1990)
- 2nd Gen 12 PEs 720 MCps (1994)

➢ Used in satellite, fruit sorting, PCB inspection, sleep analysis, …

## CEA's MIND machine

- Hybrid analog/digital: MIND-128 (1986)
- Fully digital: MIND-1024 (1991)

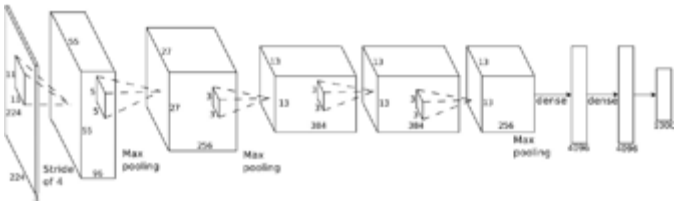


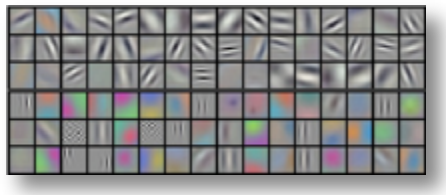
- ***Orange video-grading***
- ***Chip alignment***
- ***Sleep phase analysis***
- ***Image compression***
- ***Satellite image analysis***
- ***LHC 1st level trigger***

# 2012: DEEP NEURAL NETWORKS RISE AGAIN

They give the ***state-of-the-art performance*** e.g. in image classification

- **ImageNet classification (Hinton's team, hired by Google)**
  - 14,197,122 images, 1,000 different classes
  - Top-5 17% error rate (huge improvement) in 2012 (now ~ 3.5%)



"**Supervision**" network
Year: 2012
650,000 neurons
60,000,000 parameters
630,000,000 synapses

- **Facebook's 'DeepFace' Program (labs headed by Y. LeCun)**
  - 4.4 million images, 4,030 identities
  - 97.35% accuracy, vs. 97.53% human performance

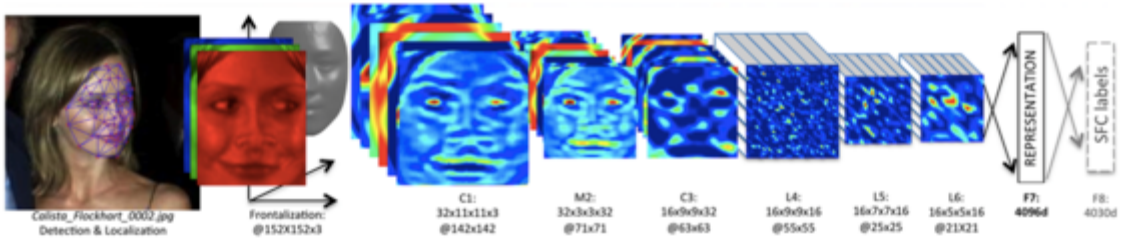



Figure 2. **Outline of the *DeepFace* architecture.** A front-end of a single convolution-pooling-convolution filtering on the rectified input, followed by three locally-connected layers and two fully-connected layers. Colors illustrate feature maps produced at each layer. The net includes more than 120 million parameters, where more than 95% come from the local and fully connected layers.

From:Y. Taigman, M. Yang, M.A. Ranzato, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification"

# ImageNet: Classification



- Give the name of the dominant object in the image
- Top-5 error rates: if correct class is not in top 5, count as error
  - Black:ConvNet, Purple: no ConvNet

| 2012 Teams | %error |
|---|---|
| Supervision (Toronto) | 15.3 |
| ISI (Tokyo) | 26.1 |
| VGG (Oxford) | 26.9 |
| XRCE/INRIA | 27.0 |
| UvA (Amsterdam) | 29.6 |
| INRIA/LEAR | 33.4 |

| 2013 Teams | %error |
|---|---|
| Clarifai (NYU spinoff) | 11.7 |
| NUS (singapore) | 12.9 |
| Zeiler-Fergus (NYU) | 13.5 |
| A. Howard | 13.5 |
| OverFeat (NYU) | 14.1 |
| UvA (Amsterdam) | 14.2 |
| Adobe | 15.2 |
| VGG (Oxford) | 15.2 |
| VGG (Oxford) | 23.0 |

| 2014 Teams | %error |
|---|---|
| GoogLeNet | 6.6 |
| VGG (Oxford) | 7.3 |
| MSRA | 8.0 |
| A. Howard | 8.1 |
| DeeperVision | 9.5 |
| NUS-BST | 9.7 |
| TTIC-ECP | 10.2 |
| XYZ | 11.2 |
| UvA | 12.1 |

## COMPETITION ON IMAGENET !

| Name of the algorithm | Date | Error on test set |
|---|---|---|
| Supervision | 2012 | 15.3% |
| Clarifai | 2013 | 11.7% |
| GoogLeNet | 2014 | 6.66% |
| **Humain level** (Adrej Karpathy) | | **5%** |
| Microsoft | 05/02/2015 | 4.94% |
| Google | 02/03/2015 | 4.82% |
| Baidu/ Deep Image | 10/05/2015 | 4.58% |
| Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences | 10/12/2015 (le CNN a 152 couches!) | 3.57% |
| Google Inception-v3 (Arxiv) | 2015 | 3.5% |
| WMW (Momenta) | 2017 | 2.2% |
| | Now | ? |

# Deep Learning is Everywhere (ConvNets are Everywhere)



- **Lots of applications at Facebook, Google, Microsoft, Baidu, Twitter, IBM…**
  - Image recognition for photo collection search
  - Image/Video Content filtering: spam, nudity, violence.
  - Search, Newsfeed ranking
- **People upload 800 million photos on Facebook every day**
  - (2 billion photos per day if we count Instagram, Messenger and Whatsapp)
- **Each photo on Facebook goes through two ConvNets within 2 seconds**
  - One for image recognition/tagging
  - One for face recognition (not activated in Europe).
- **Soon ConvNets will really be everywhere:**
  - self-driving cars, medical imaging, augemnted reality, mobile devices, smart cameras, robots, toys…..

# PIXEL WISE IMAGE SEGMENTATION

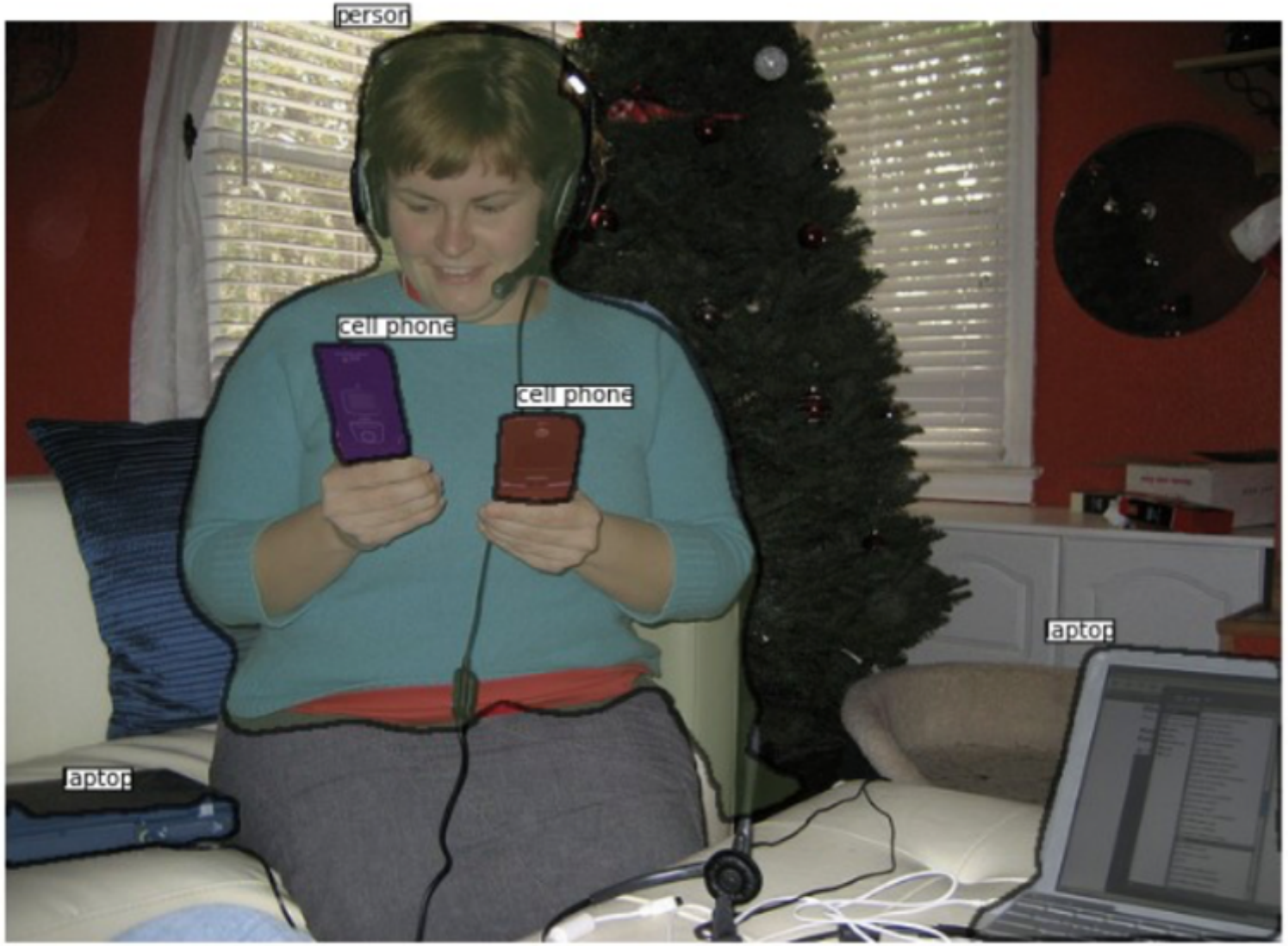
- DNN technic: Fully-CNN + Unpooling (for high resolution segmentation)


# IMAGE ROI EXTRACTION AND CLASSIFICATION

- DNN technic: Faster-RCNN (or similar: YOLO, SSD…)

# Results



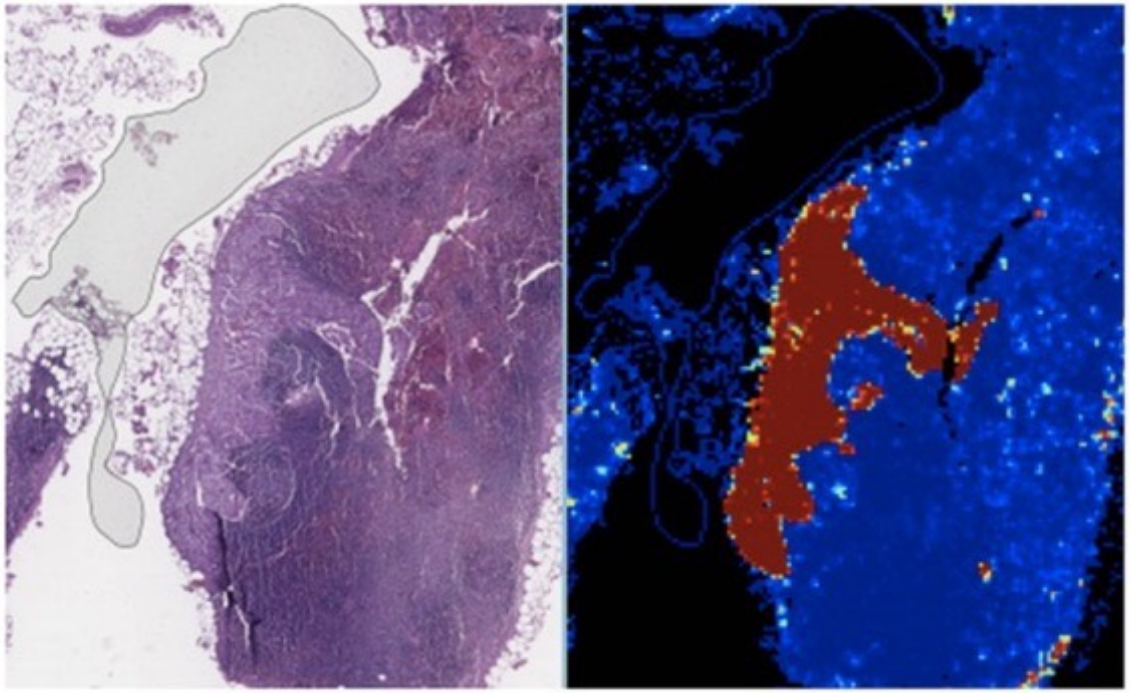


# Detecting Cancer Metastases

Tumor localization score (FROC):

Pathologist: 0.73
AI model: 0.89
(better)

*Detecting Cancer Metastases on Gigapixel Pathology Images (2017)*



From Olivier Temam

# DEEP MANTA

## MANY-TASK DEEP NEURAL NETWORK FOR VISUAL OBJECT RECOGNITION

## Applications

Driving assistance, autonomous driving
Smart city
Video-protection
Advanced Manufacturing


## Technology

1. Object detection
2. Fine-grained recognition
3. Accurate pose estimation
4. 2D/3D localisation
5. Part localisation
6. Part visibility characterization

## Performance

KITTI Benchmark:
- 1st rank in vehicle orientation estimation
- Top-10 in object detection

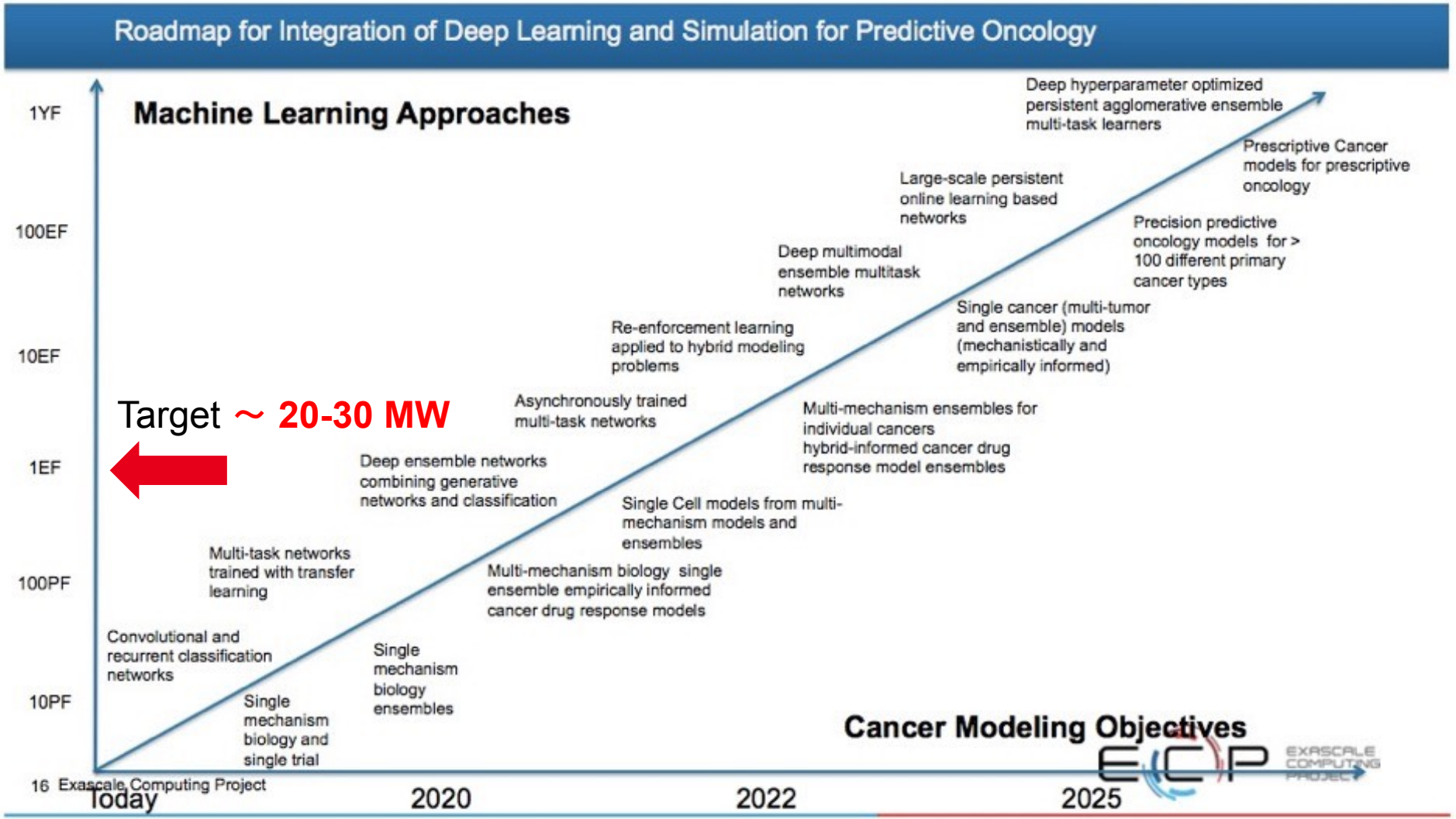
Runs at 10 Hz on Nvidia Gtx 1080

**CVPR 2017 :** F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière and T. Château
Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image.

# ALPHAGO ZERO: SELF-PLAYING TO LEARN


From doi:10.1038/nature24270 (Received 07 April 2017)

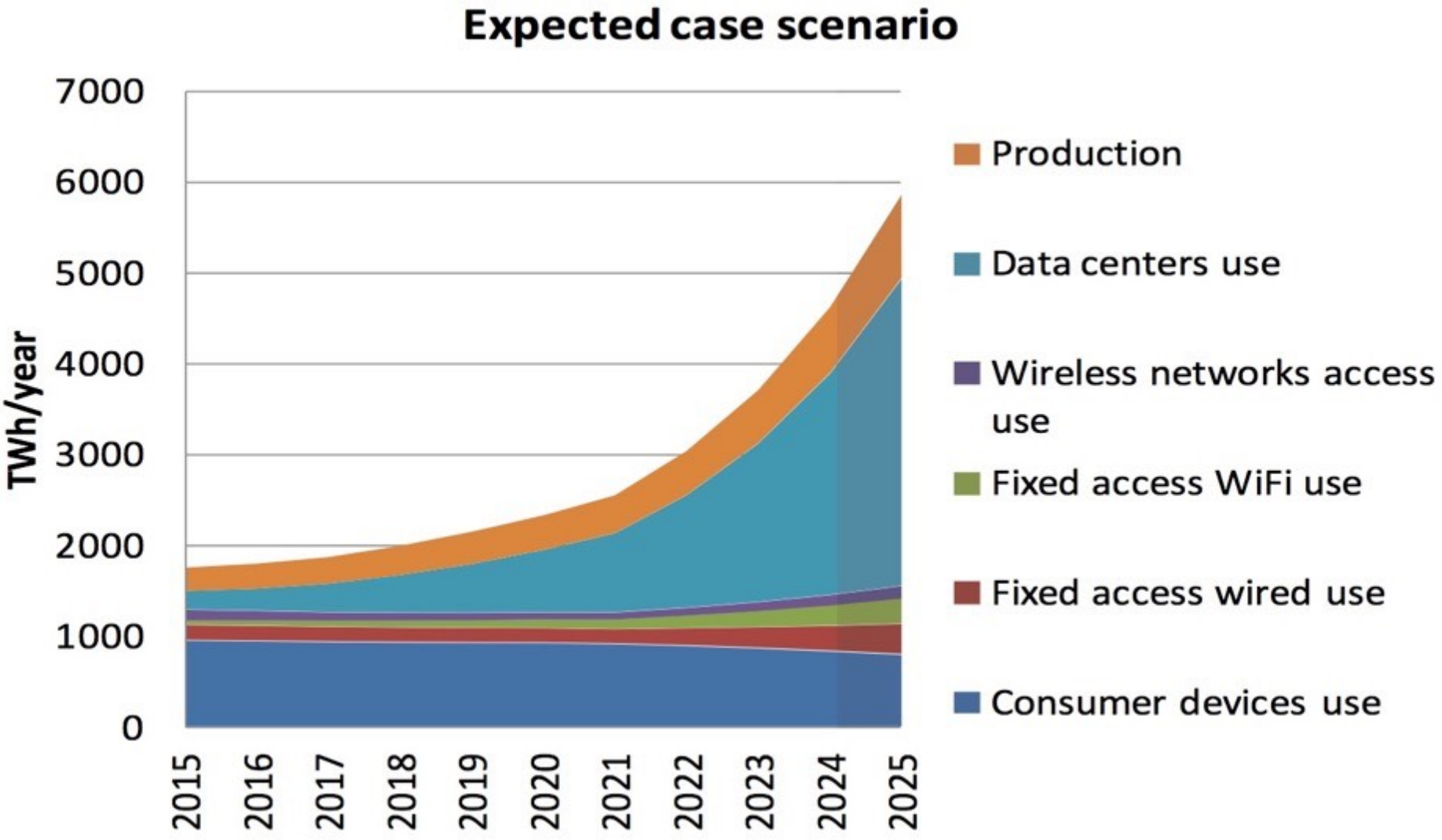
# ALWAYS MORE COMPUTING RESSOURCES




From Paul Messina, Argonne National Laboratory


HOUSTON
WE HAVE A PROBLEM...

# The problem:




From “Total Consumer Power Consumption Forecast”, Anders S.G. Andrae, October 2017

# THE END OF MOORE'S LAW

| Parameter (scale factor = a) | Classic Scaling |
|---|---|
| Dimensions | 1/a |
| Voltage | 1/a |
| Current | 1/a |
| Capacitance | 1/a |
| Power/Circuit | $1/a^2$ |
| Power Density | 1 |
| Delay/Circuit | 1/a |

Everything was easy:

- Wait for the next technology node
- Increase frequency
- Decrease Vdd

⇒ Similar increase of sequential performance

⇒ No need to recompile (except if architectural improvements)

Source: Krisztián Flautner "From niche to mainstream: can critical systems make the transition?"

# THE END OF ~~MOORE'S LAW~~ DENNARD SCALING

| Parameter (scale factor = a) | Classic Scaling | Current Scaling |
|---|---|---|
| Dimensions | 1/a | 1/a |
| Voltage | 1/a | 1 |
| Current | 1/a | 1/a |
| Capacitance | 1/a | >1/a |
| Power/Circuit | $1/a^2$ | 1/a |
| Power Density | 1 | ***a*** |
| Delay/Circuit | 1/a | ~1 |

Source: Krisztián Flautner "From niche to mainstream: can critical systems make the transition?"

# Exponential increase of performances in 33 years

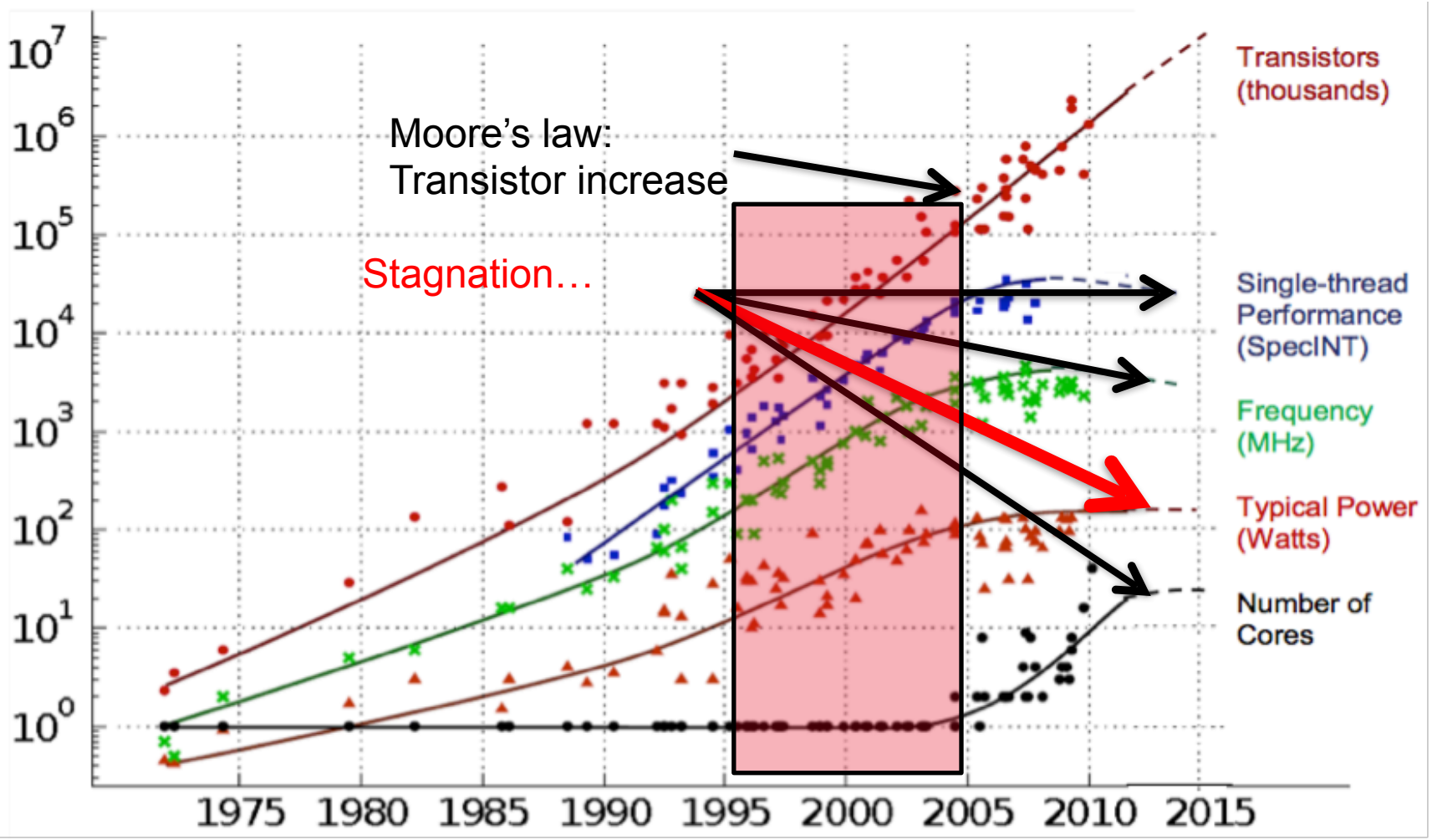
To infinity and beyond…


Production car of 1985
Lamborghini Countach 5000QV
Max speed 300 Km/h

X 100 000 000
in 33 years

Star Trek Enterprise
Year: about 2290
27 times the speed of light?

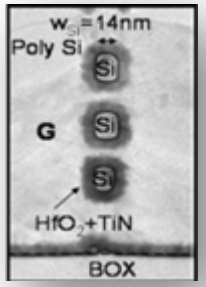
# MOORE 'S LAW AND DENNARD SCALING

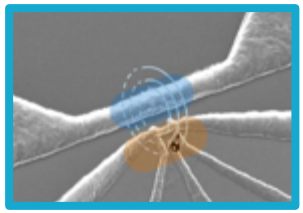



Source from C Moore, « Data Processing in ExaScale-Class Computer Systems », Salishan, April 2011

# Technology evolution

Silicon Quantum bits



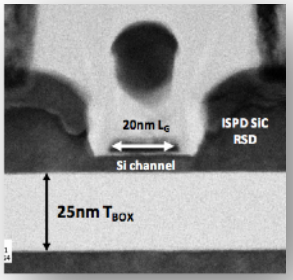



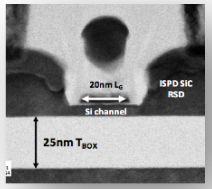
FDSOI

22FD

12FD

Next Gen





Non planar / trigate / stacked Nanowires

FinFET

28nm

14nm

10nm

2017

2018

5nm

7nm

Disruptive scaling

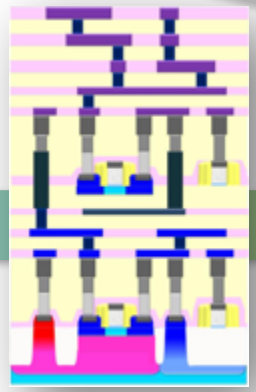
Hybrid logic

Steep slope devices

Mechanical switches

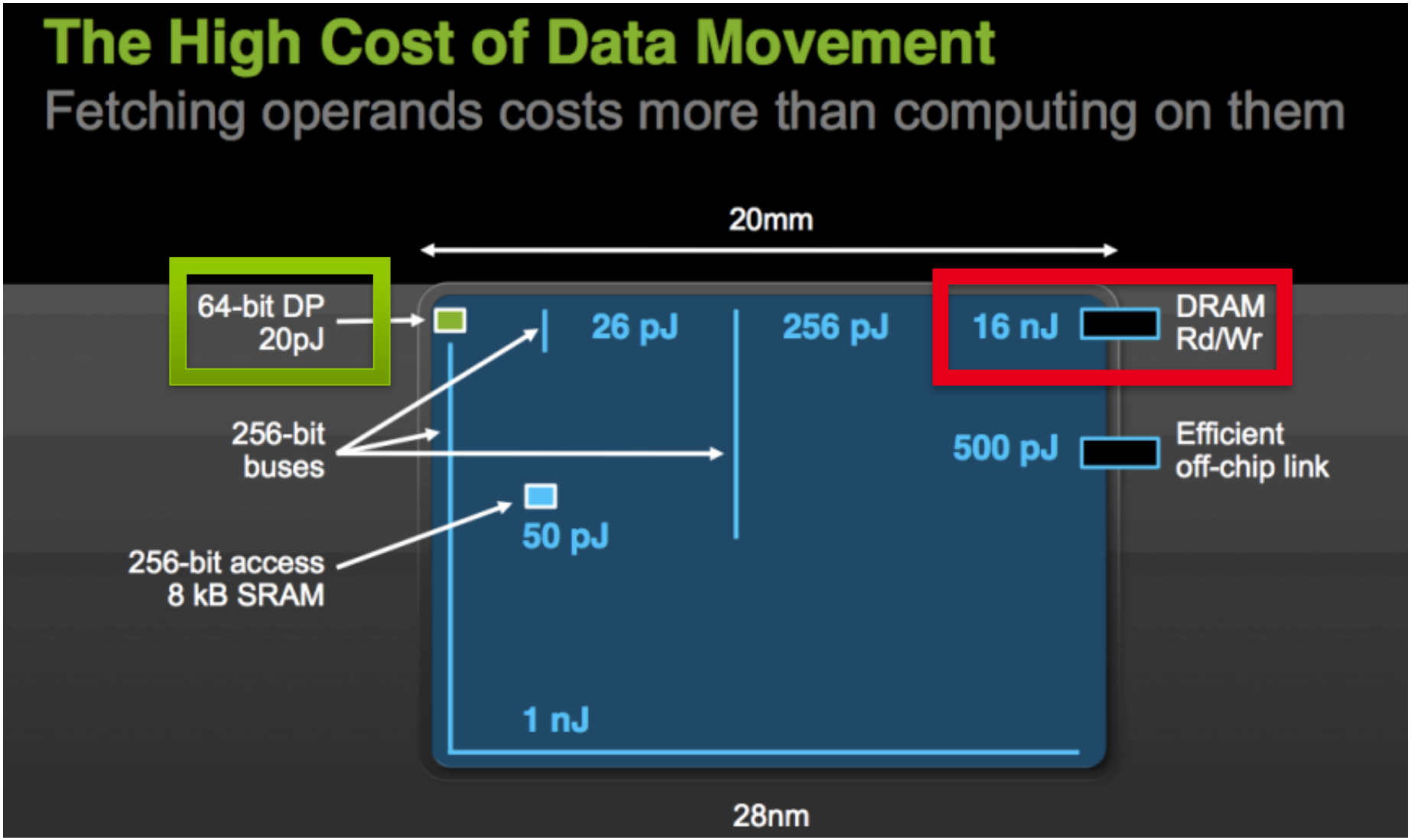
Si Quantum bits

Alternative to scaling and diversification



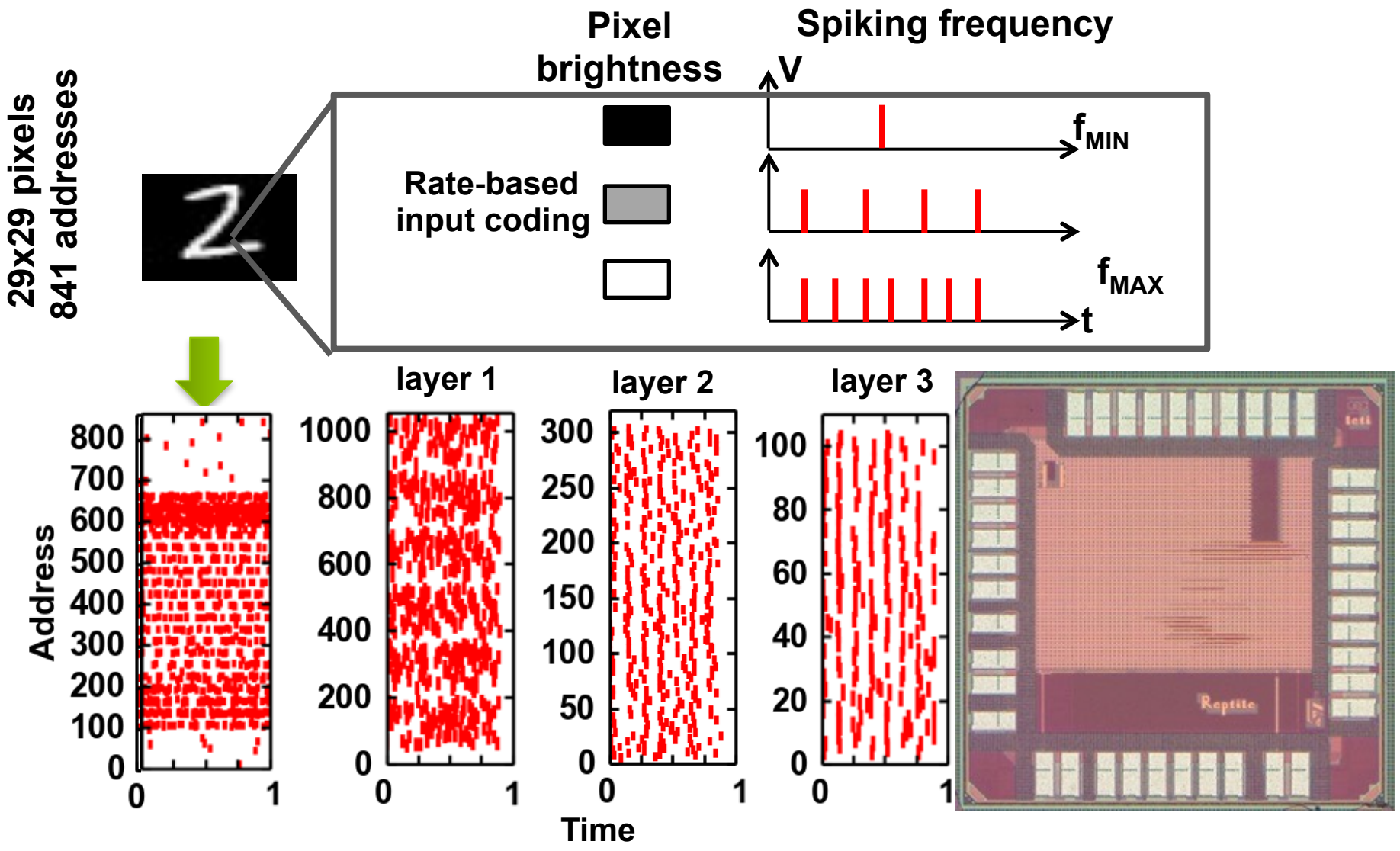
Monolithic 3D for 3D VLSI

# COST OF MOVING DATA -> COMPUTING IN MEMORY



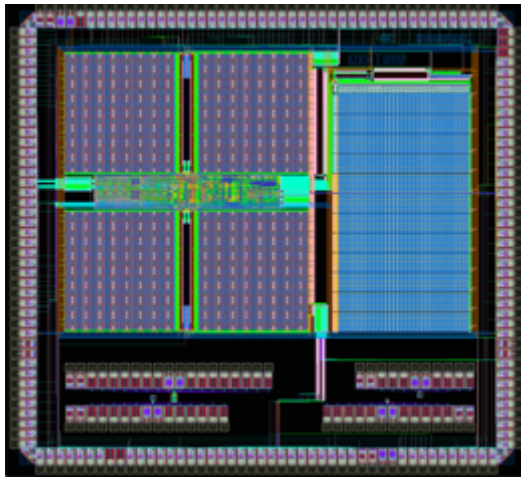

**Source: Bill Dally, « To ExaScale and Beyond »**
**www.nvidia.com/content/PDF/sc_2010/theater/Dally_SC10.pdf**

# SPIKE-BASED CODING



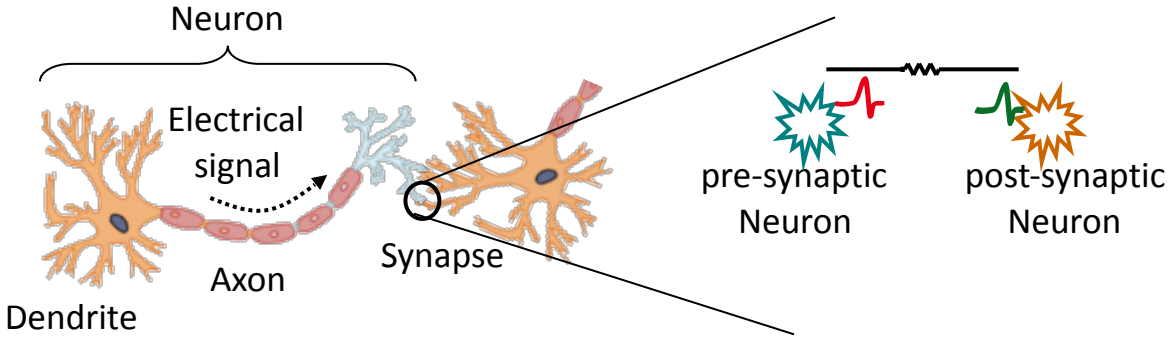
# NEUROMORPHIC ACCELERATOR: COMPUTE AND MEMORY TOGETHER IN DYNAPS-SL (INI-ZURICH)

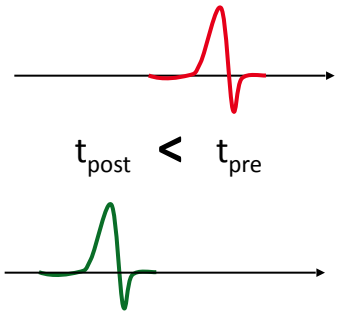
| | Neuram3 1st chip | IBM True North |
|---|---|---|
| Technology | 28 nm FDSOI | 28nm CMOS |
| Supply Voltage | 1 V | 0.7V |
| Neuron Type | Analog | Digital |
| Neurons per core | 256 | 256 |
| Core Area | 0.36 $mm^2$ | 0.094 $mm^2$ |
| Computation | Parallel processing | Time multiplexing |
| Fan In/Out | 2k/8k | 256/256 |
| Synaptic Operation per Second per Watt | **300 GSOPS/ W*1** | 46 GSOPS/W |
| Energy per synaptic event | **<2 pJ*2** | 10 pJ |
| Energy per spike | <0.375 nJ*3 | 3.9 nJ |

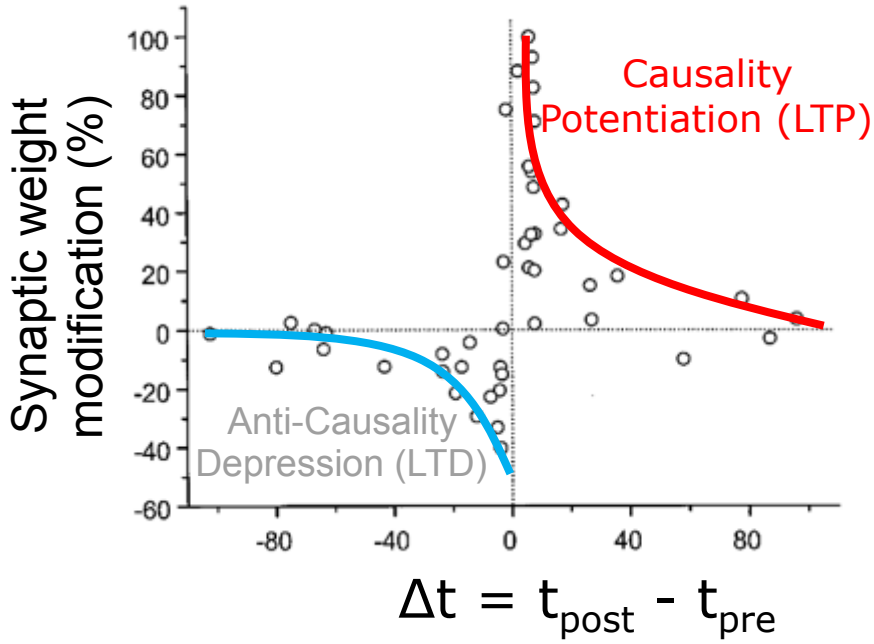


* 1 At 100Hz mean firing rate, by appending 4 local-core destinations per spike, 400 k events will be broadcast to 4 cores with 25% connectivity per event. 400 k x 1 k x 25% / 300 µ W = 300 GSOPS/W
* 2 In case of 25% match in each core, energy per synaptic event = energy per broadcast / (256*25%) =120pJ/64 = 2 pJ
* 3 Energy per spike = total power consumption / spikes numbers = 300 uW/800 k = 0.375 nJ

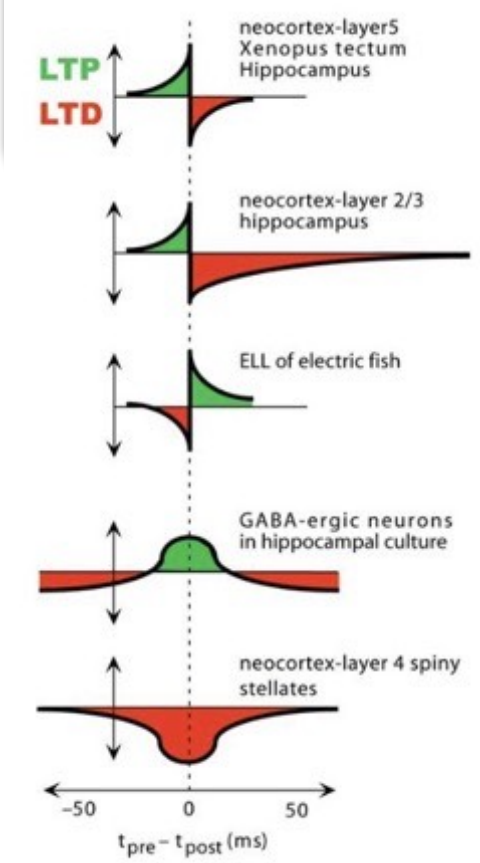
# Learning from neuroscience: STDP (Spike Timing Dependent Plasticity)



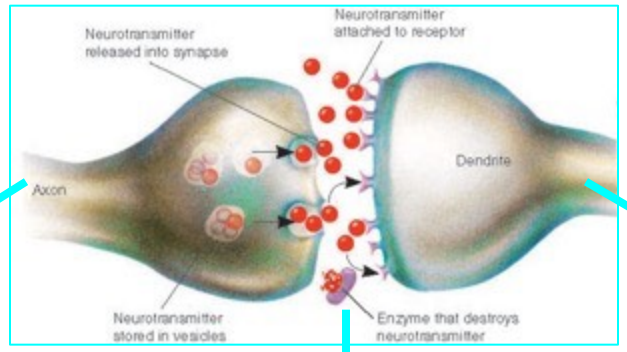




STDP = correlation detector



# Investigating RRAM as synapses
# Unsupervised learning (information coded by Spikes)



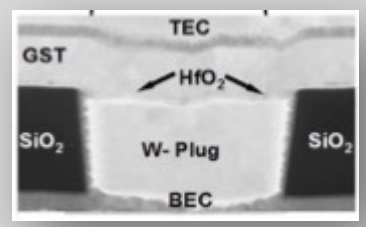

*Thermal effect*

*Electrochemical effect*

*Electronic effect*
*oxygen vacancies*

## PCM

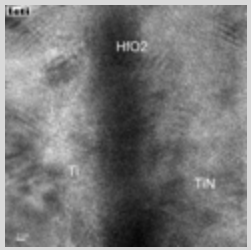
**GST**
**GeTe**
**GST + $HfO_2$**




*M.Suri, et. al, IEDM 2011*
*M.Suri, et. al, IMW 2012 , JAP 2012*
*O.Bichler et al. IEEE TED 2012*
*M.Suri et al., EPCOS 2013*
*D.Garbin et al., IEEE Nano 2013*

## OXRAM

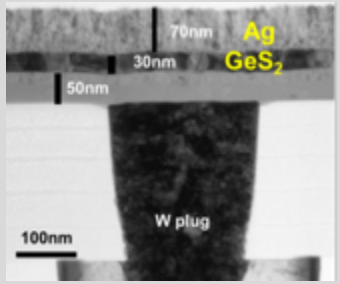
**$TiN/HfO_2/Ti/TiN$**




*D.Garbin et al. IEDM 2014*
*D.Garbin et al., IEEE TED 2015*

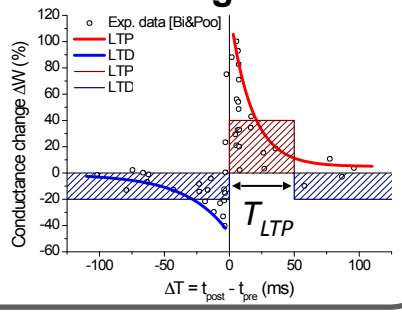
## CBRAM

**Ag / $GeS_2$**

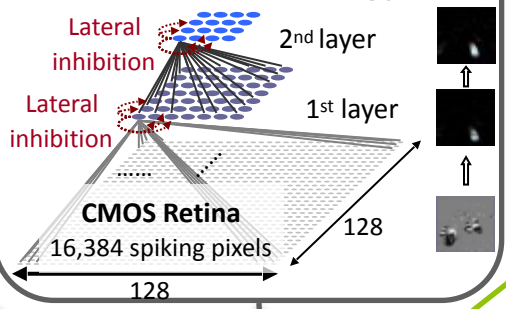


# Bio-inspired models exploration

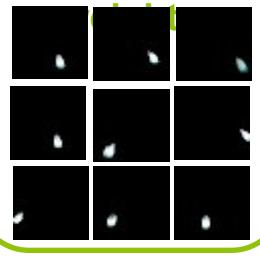
## Learning rule



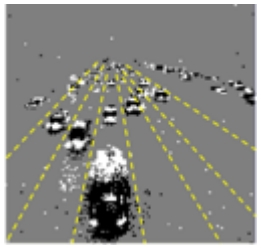

## Network topology




## Synaptic



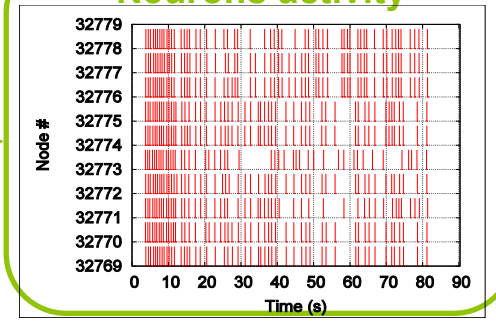
## Input stimuli



## N2-D2

**Neuromorphic simulator**

## Neurons activity

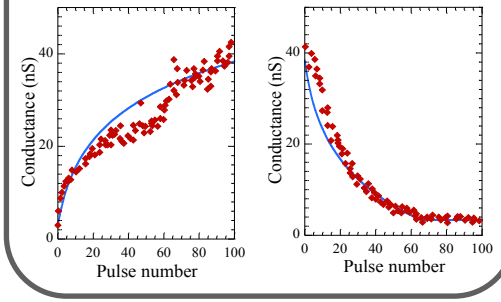



## Neuron model

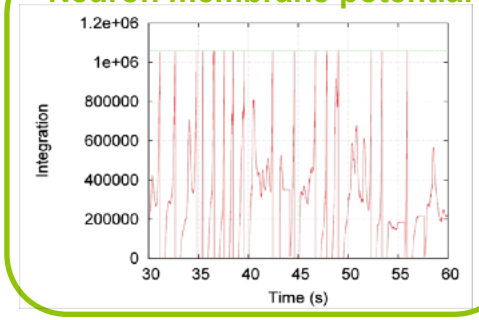
*Example: Leaky Integrate & Fire (LIF) neuron*

$$u = u \,.\, e^{-\frac{t_{spike} - t_{last\_spike}}{\tau_{leak}}} + w$$

## Synaptic model




## Neuron membrane potential




**Complete tool flow for bio-inspired synapses, neurons and learning rules network simulations**

*[O. Bichler et al., NanoArch'2014]*

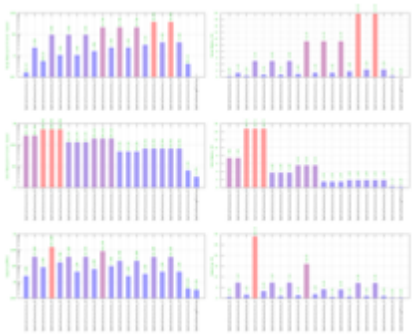
# Fast and accurate Deep Neural Networks exploration

## N2D2 INI network description file

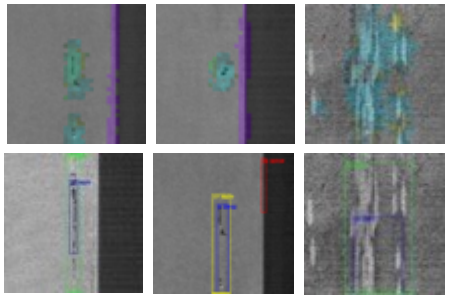
```
; Database
[database]
Type=MNIST_IDX_Database
Validation=0.2

; Environment
[env]
SizeX=24
SizeY=24
BatchSize=128

[env.Transformation]
Type=PadCropTransformation
Width=[env]SizeX
Height=[env]SizeY

[env.OnTheFlyTransformation]
Type=DistortionTransformation
ApplyTo=LearnOnly
ElasticGaussianSize=21
ElasticSigma=6.0
ElasticScaling=36.0
Scaling=10.0
Rotation=10.0

; First layer (convolutionnal)
[conv1]
Input=env
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=6
Stride=2
ConfigSection=common.config

; Second layer (convolutionnal)
[conv2]
Input=conv1
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=12
Stride=2
ConfigSection=common.config

; Third layer (fully connected)
[fc1]
Input=conv2
Type=Fc
NbOutputs=100
ConfigSection=common.config

; Output layer (fully
[fc2]
Input=fc1
Type=Fc
NbOutputs=10
ConfigSection=com

; Softmax layer
[soft]
Input=fc2
Type=Softmax
NbOutputs=10
WithLoss=1
ConfigSection=common.config

; Common solvers config
```

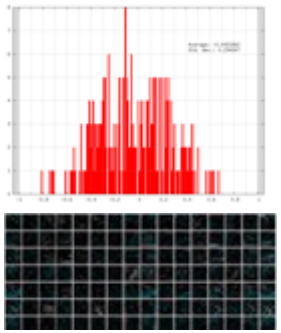
**Dataflow visualization**

**Layer-wise detailed memory and computing requirements**



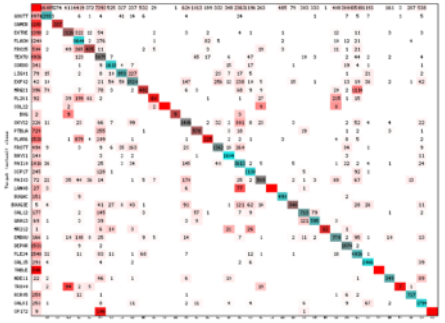
**Results visualization:**
- **Pixel-wise segmentation**
- **ROI bounding box extraction and classification**





**Layer-wise weights and kernels visualization, distribution and data-range analysis**

**Layer-wise output visualization and data-range analysis**



**Pixel-wise and object wise confusion matrix reporting**

# Example of use of N2D2

| AppObjectRecognition/ | AppFaceDetection/ | AppRoadDetection/ |
|---|---|---|
| **Live object recognition application** *based on ILSVRC2012 (ImageNet) dataset* | **Live face detection application, with gender recognition** *based on the IMDB-WIKI dataset* | **Simple road segmentation application** *based on the KITTI Road dataset* |
|  |  |  |

**N2D2 is available at https://github.com/CEA-LIST/N2D2/**

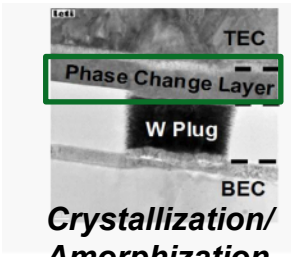
- Smallest dependencies and requirements among major frameworks:
  GCC 4.4 or Visual Studio 12 (2013) / OpenCV 2.0.0
- Easily extendable with a “plug-and-play” modular system for user-made modules

**Development of efficient solutions for Deep Learning Inference**

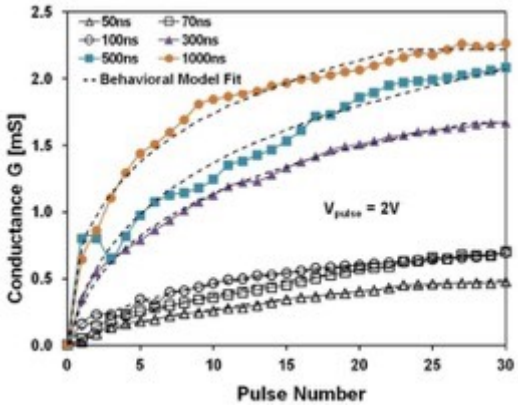
# NVM synapses implementations

## 2-PCM synapses for unsupervised cars trajectories extraction

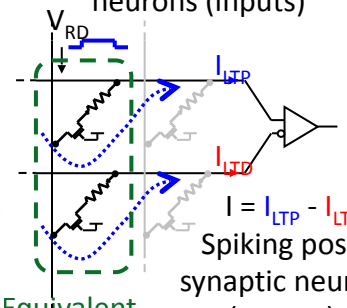
**PCM**




***Crystallization/ Amorphization***



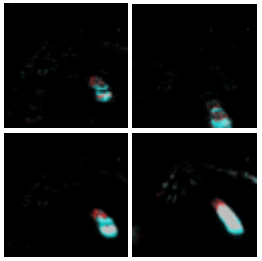

From spiking pre-synaptic neurons (inputs)




$I = I_{LTP} - I_{LTD}$

Spiking post-synaptic neuron (output)

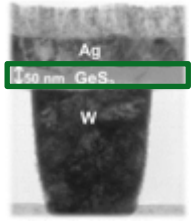
Equivalent 2-PCM synapse



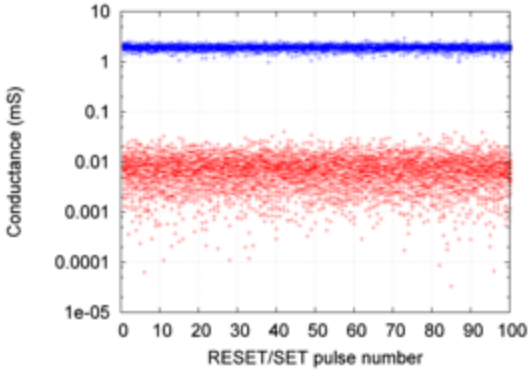
*[O. Bichler et al., Electron Devices, IEEE Transactions on, 2012]*

## CBRAM binary synapses for unsupervised MNIST handwritten digits classification with stochastic learning

**CBRAM**




***Forming/Dissolution of conductive filament***



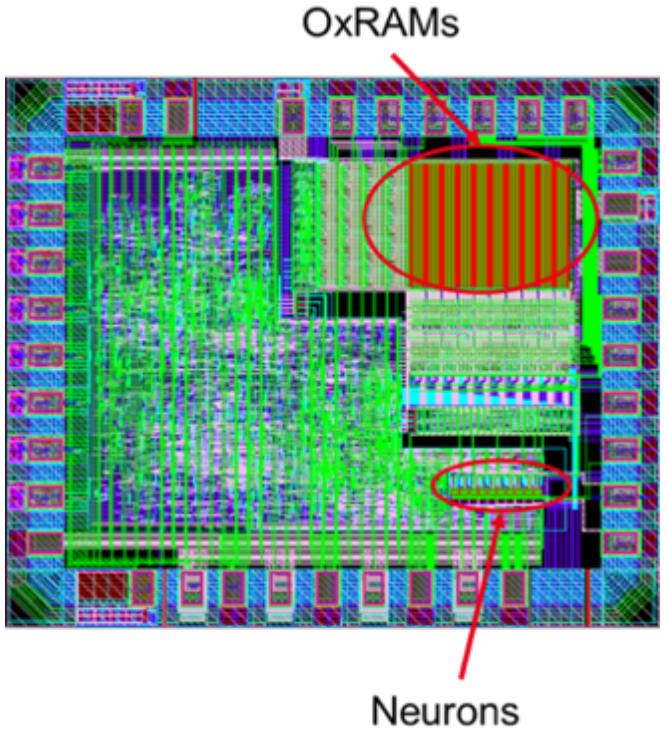

*[M. Suri et al., IEDM, 2012]*

# NVM synapses implementations

Test vehicle for spiking neural networks in 130nm CMOS with OxRAM elements between Metal 4 and Metal 5 of the back-end is done at CEA LETI.

Area is 1,8mm². It contains 10 neurons and 1440 synapses, (11,5k OxRAMs)

It can run MNIST (Characters recognition)

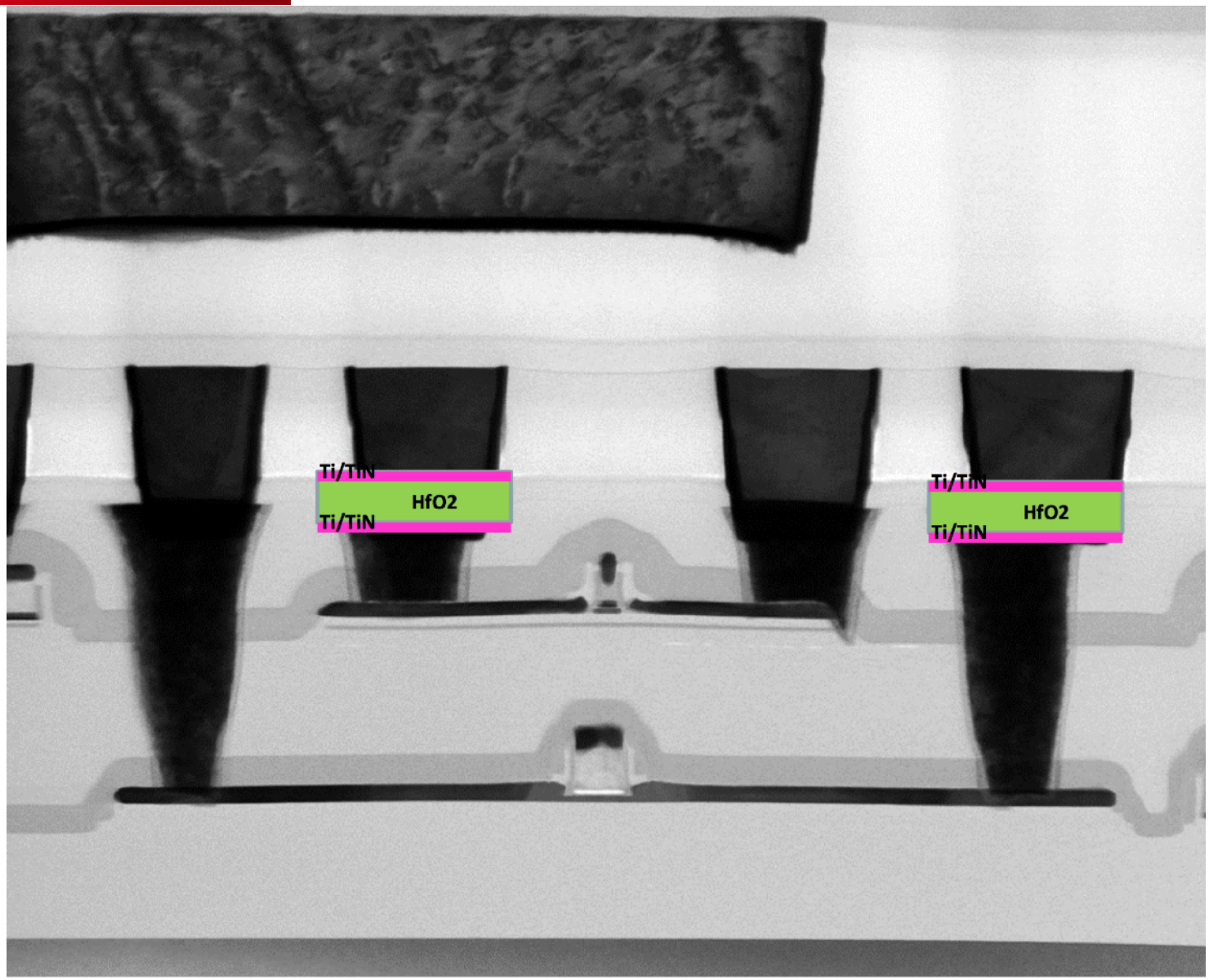



SPIRIT test chip

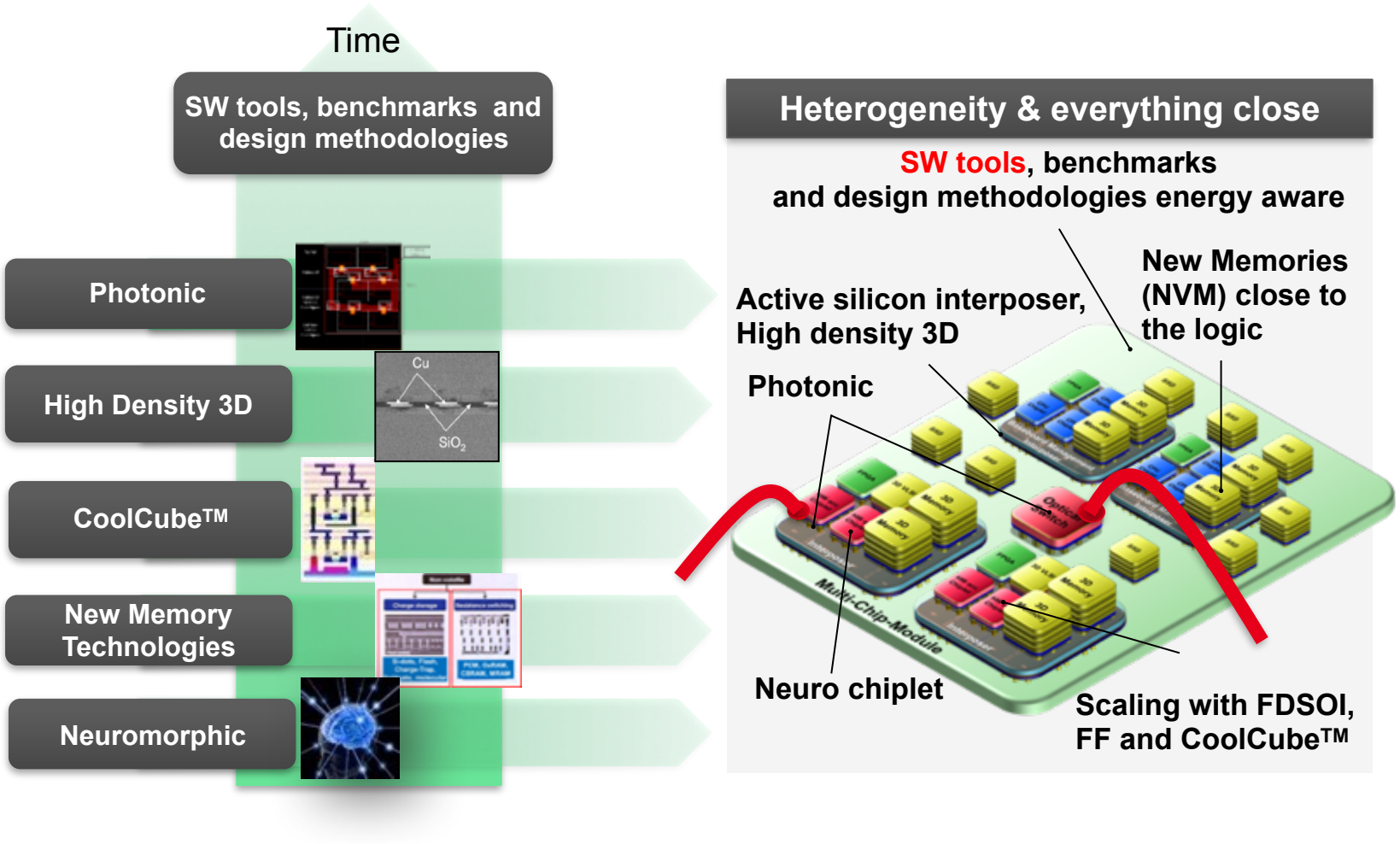
**European project: NeuRAM3**

***NEUral computing aRchitectures in Advanced Monolithic 3D-VLSI nano-technologies***

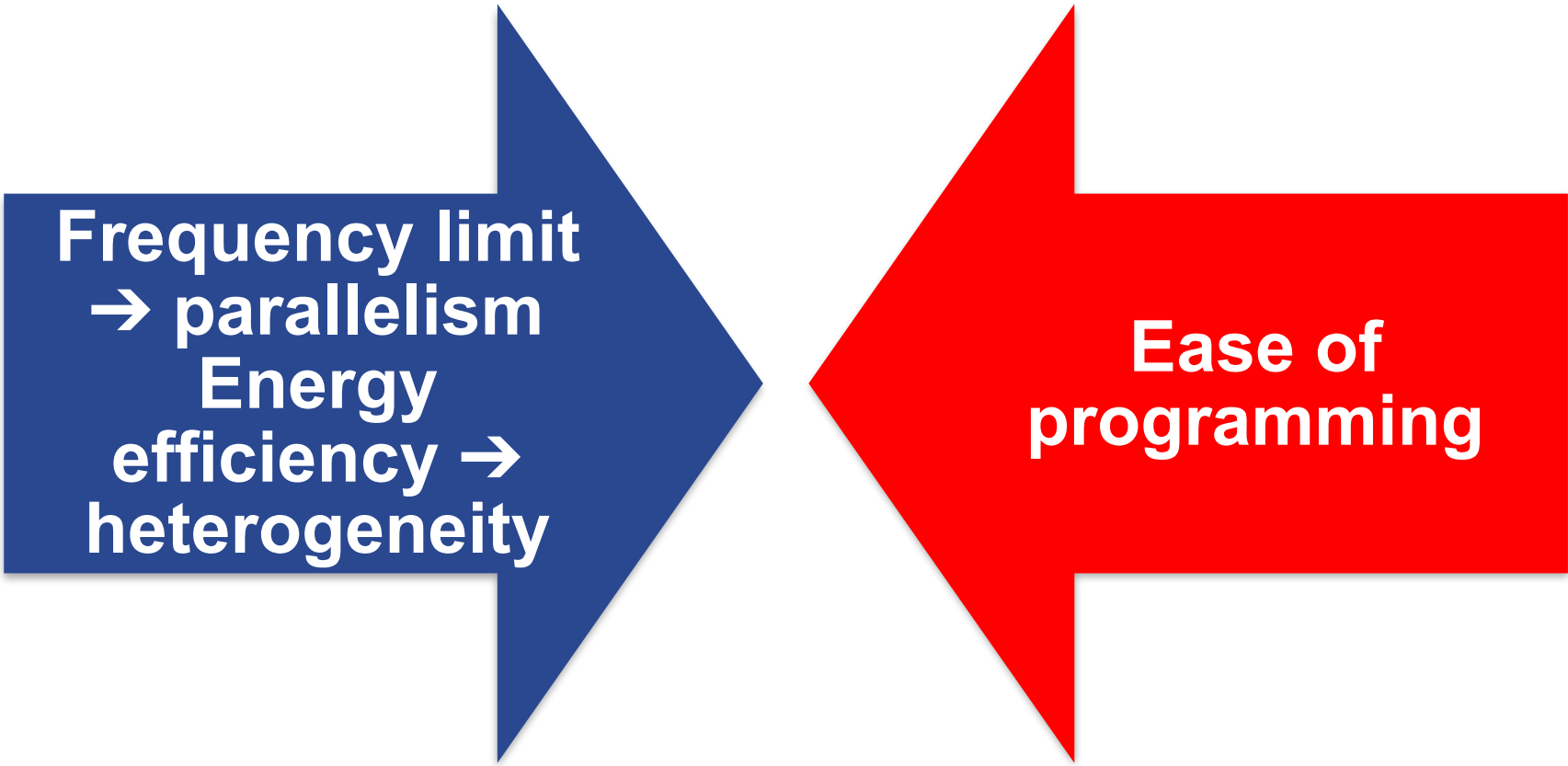
# REDUCING COMMUNICATIONS: 3D INTEGRATION COUPLED WITH RRAM



# POTENTIAL SOLUTION FOR COGNITIVE CYBER PHYSICAL SYSTEMS

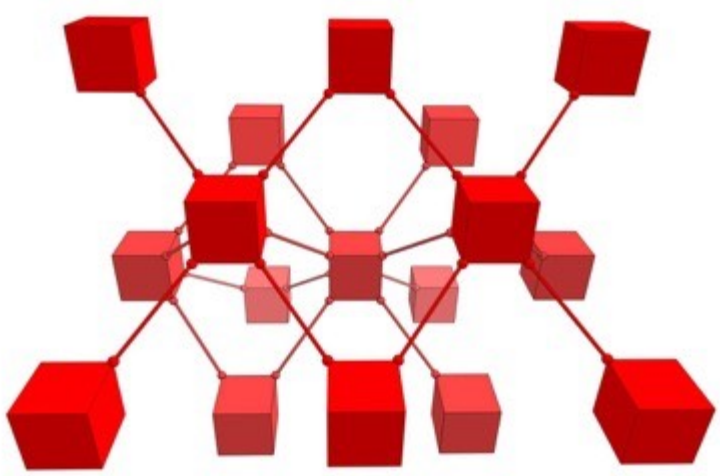


# PARALLELISM AND SPECIALIZATION ARE NOT FOR FREE…



## MANAGING COMPLEXITY….

*“Nontrivial software written with threads, semaphore, and mutexes is incomprehensible by humans”*



Edward A. Lee

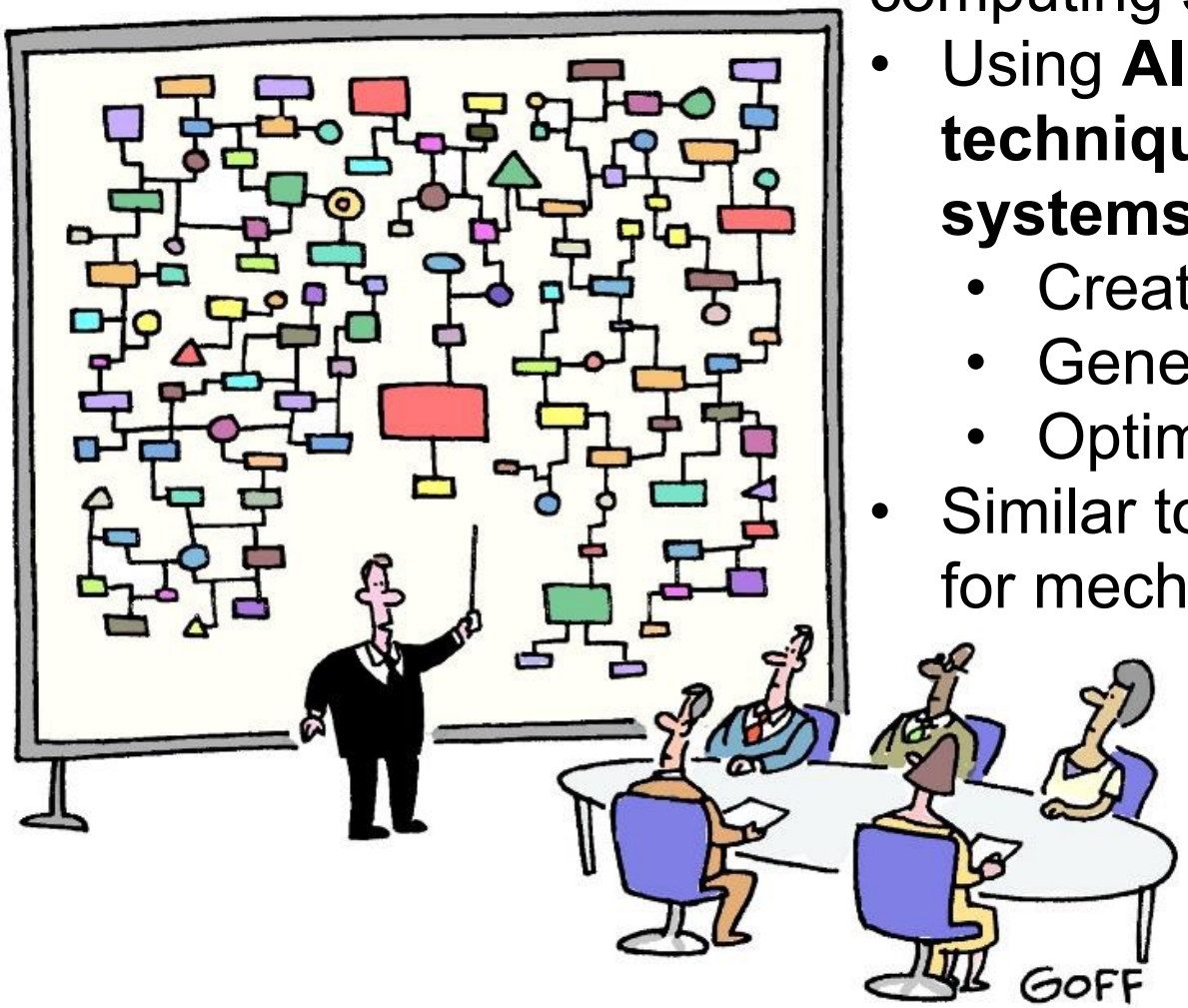
The future of embedded software
ARTEMIS 2006

Parallelism, multi-cores, heterogeneity, distributed computing, seems to be too complex for humans ?

## Managing complexity

Cognitive solutions for complex computing systems:

- Using **AI and optimization techniques for computing systems**
  - Creating new hardware
  - Generating code
  - Optimizing systems
- Similar to ***Generative design*** for mechanical engineering




"And that's why we need a computer."

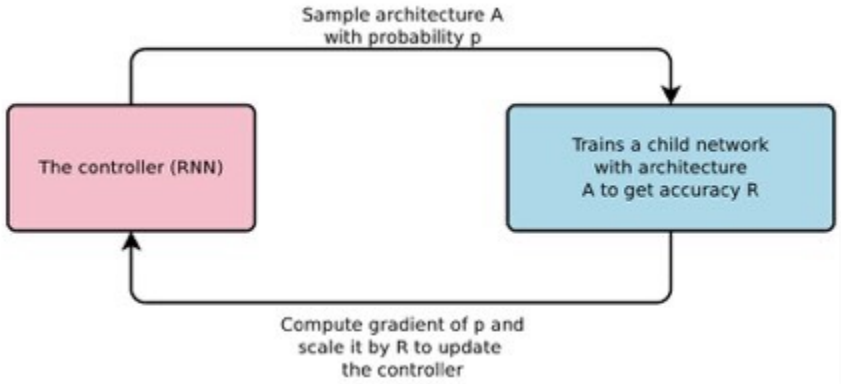
# USING AI FOR MAKING CPS SYSTEMS: “GENERATIVE DESIGN” APPROACH

The user ***only states desired goals and constraints*** ***->*** The ***complexity wall*** might ***prevent explaining*** the solution


Motorcycle swingarm: the piece that hinges the rear wheel to the bike’s frame

# 2017: GOOGLE; USING DEEP LEARNING TO DESIGN DEEP LEARNING

**"*Neural Architecture Search*", using a recurrent neural network to compose neural network architectures using reinforcement learning on CIFAR-10 (character recognition)**

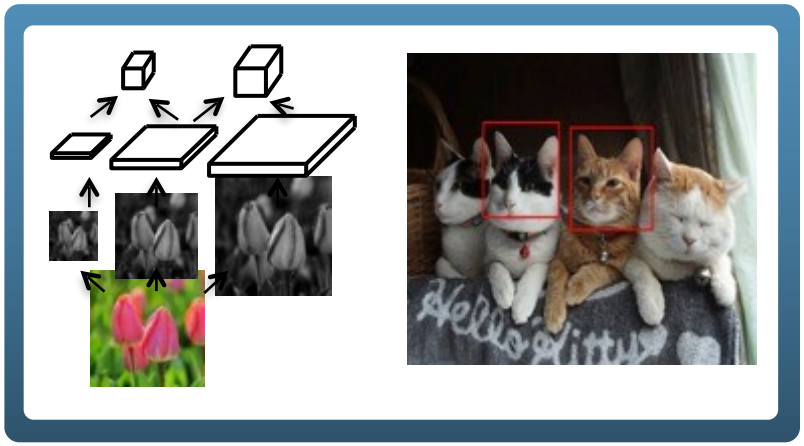



| Model | Depth | Parameters | Error rate (%) |
|---|---|---|---|
| Network in Network (Lin et al., 2013) | - | - | 8.81 |
| All-CNN (Springenberg et al., 2014) | - | - | 7.25 |
| Deeply Supervised Net (Lee et al., 2015) | - | - | 7.97 |
| Highway Network (Srivastava et al., 2015) | - | - | 7.72 |
| Scalable Bayesian Optimization (Snoek et al., 2015) | - | - | 6.37 |
| FractalNet (Larsson et al., 2016) | 21 | 38.6M | 5.22 |
| with Dropout/Drop-path | 21 | 38.6M | 4.60 |
| ResNet (He et al., 2016a) | 110 | 1.7M | 6.61 |
| ResNet (reported by Huang et al. (2016c)) | 110 | 1.7M | 6.41 |
| ResNet with Stochastic Depth (Huang et al., 2016c) | 110 | 1.7M | 5.23 |
| | 1202 | 10.2M | 4.91 |
| Wide ResNet (Zagoruyko & Komodakis, 2016) | 16 | 11.0M | 4.81 |
| | 28 | 36.5M | 4.17 |
| ResNet (pre-activation) (He et al., 2016b) | 164 | 1.7M | 5.46 |
| | 1001 | 10.2M | 4.62 |
| DenseNet ($L = 40, k = 12$) Huang et al. (2016a) | 40 | 1.0M | 5.24 |
| DenseNet($L = 100, k = 12$) Huang et al. (2016a) | 100 | 7.0M | 4.10 |
| DenseNet ($L = 100, k = 24$) Huang et al. (2016a) | 100 | 27.2M | 3.74 |
| Neural Architecture Search v1 no stride or pooling | 15 | 4.2M | 5.50 |
| Neural Architecture Search v2 predicting strides | 20 | 2.5M | 6.01 |
| Neural Architecture Search v3 max pooling | 39 | 7.1M | 4.47 |
| Neural Architecture Search v3 max pooling + more filters | 39 | 37.4M | 3.65 |

Several other interesting "**Auto-ML**" research projects

From arXiv:1611.01578v2, Barret Zoph, Quoc V. Le
Google Brain

# Q-learning based SoC energy management



- **Dynamic software applications with performance constraints, e.g., throughput**
- **Standard Linux-based operating system**
- **Multi/many core SoCs**

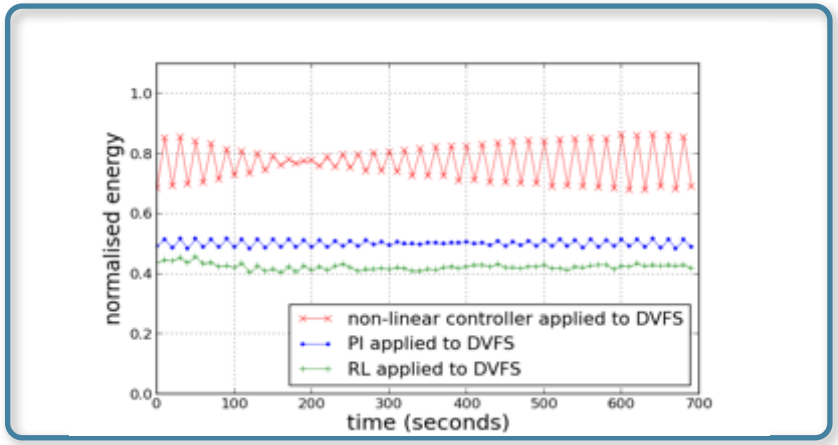
android

eLinux

Source: NXP i.MX6

Source: ST/CEA

- **Q-learning energy manager**
  - On-line, gradually learn the SoC operating points such that performance constraints are respected and **energy consumption is reduced**
  - No need to model the dynamics of the system

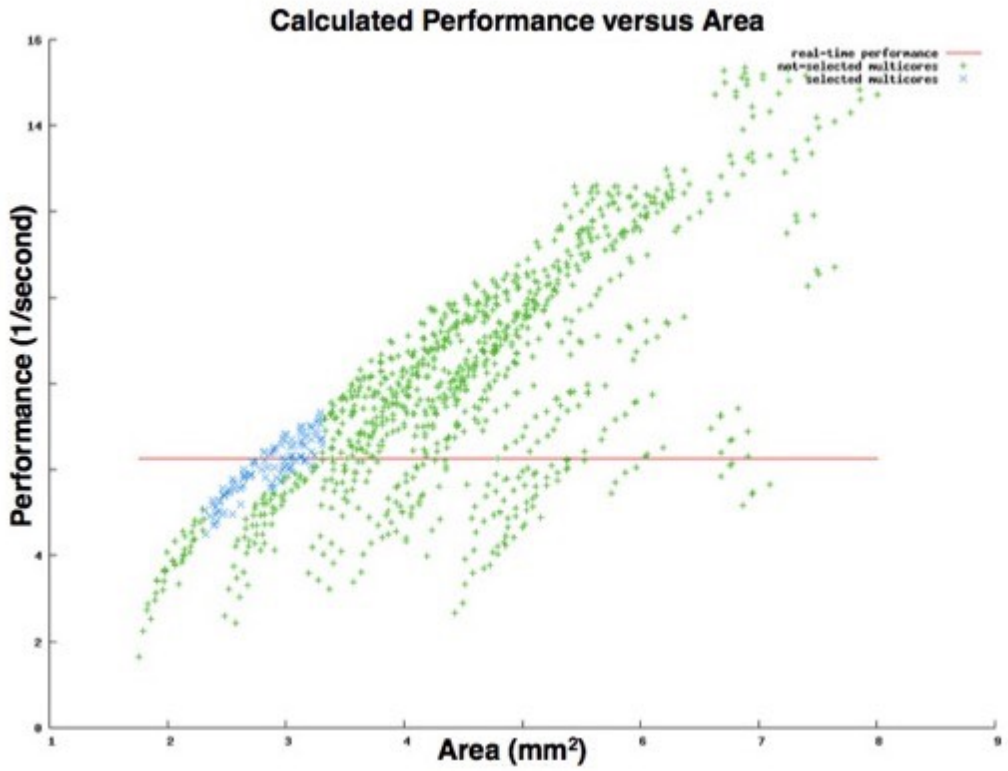



**Up to 44% energy reduction, wrt. state-of-the-art (proportional-integral and non-linear controllers)**

# EXAMPLE: DESIGN SPACE EXPLORATION FOR DESIGN MULTI-CORE PROCESSORS[1] (2010)

- Ne-XVP project – Follow-up of the TriMedia VLIW (https://en.wikipedia.org/wiki/Ne-XVP )
- 1,105,747,200 heterogeneous multicores in the design space
- 2 millions years to evaluate all design points
- AI inspired techniques allowed to reduce the induction time to only few days

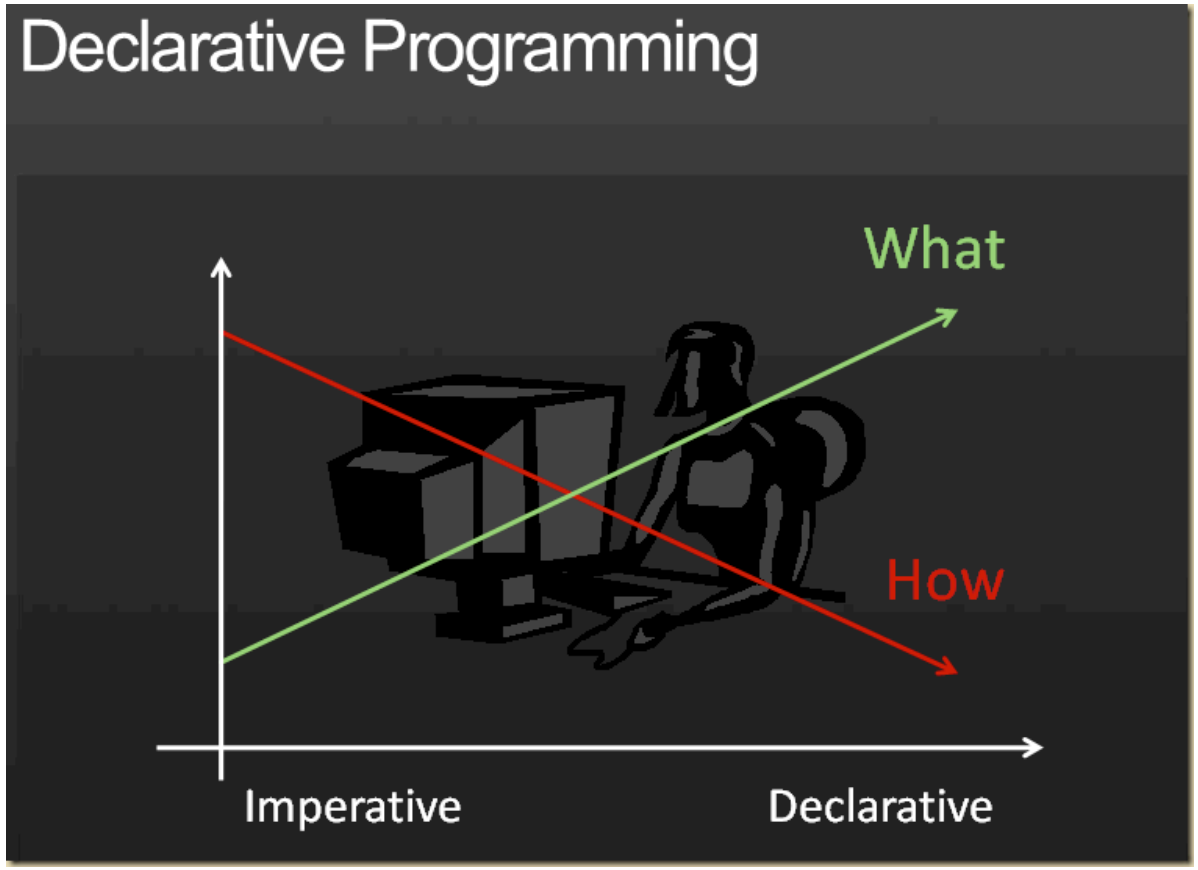
**=> *x16 performance increase***




[1] M. Duranton et all., "Rapid Technology-Aware Design Space Exploration for Embedded HeterogeneousMultiprocessors" in Processor and System-on-Chip Simulation, Ed. R. Leupers, 2010

## *PROGRAMMING 2.0*: LET THE COMPUTER DO THE JOB:

- Describing ***what*** the program should accomplish, rather than describing ***how*** to accomplish it as a sequence of the programming language primitives.
- For example, describe the ***concurrency*** of an application, not how to parallelize the code for it.
- (Good) compilers know better about architecture than humans, they are better at optimizing code…

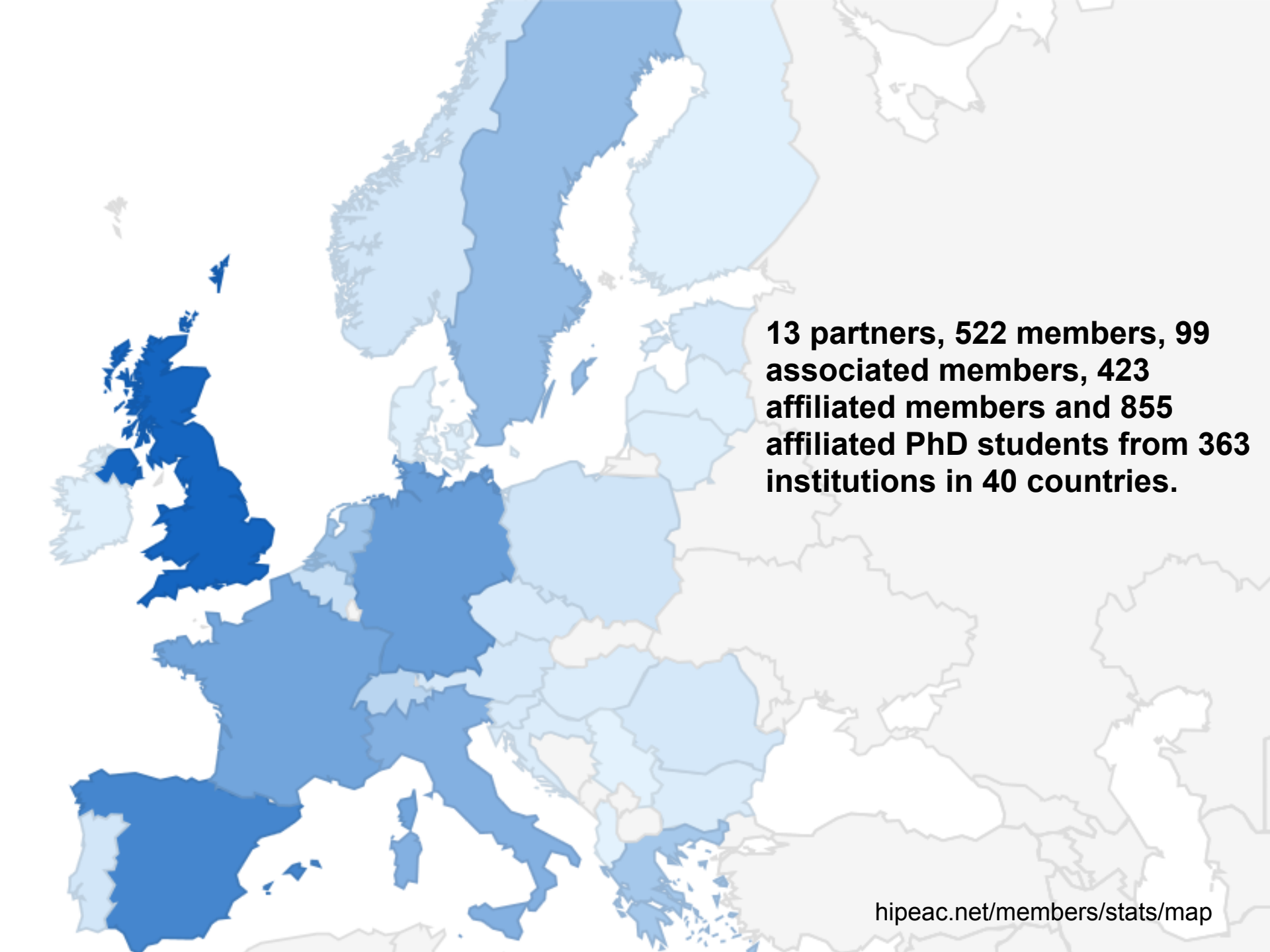


# Where it come from?

# HiPEAC
# =
# **Hi**gh-**P**erformance and **E**mbedded **A**rchitecture and **C**ompilation

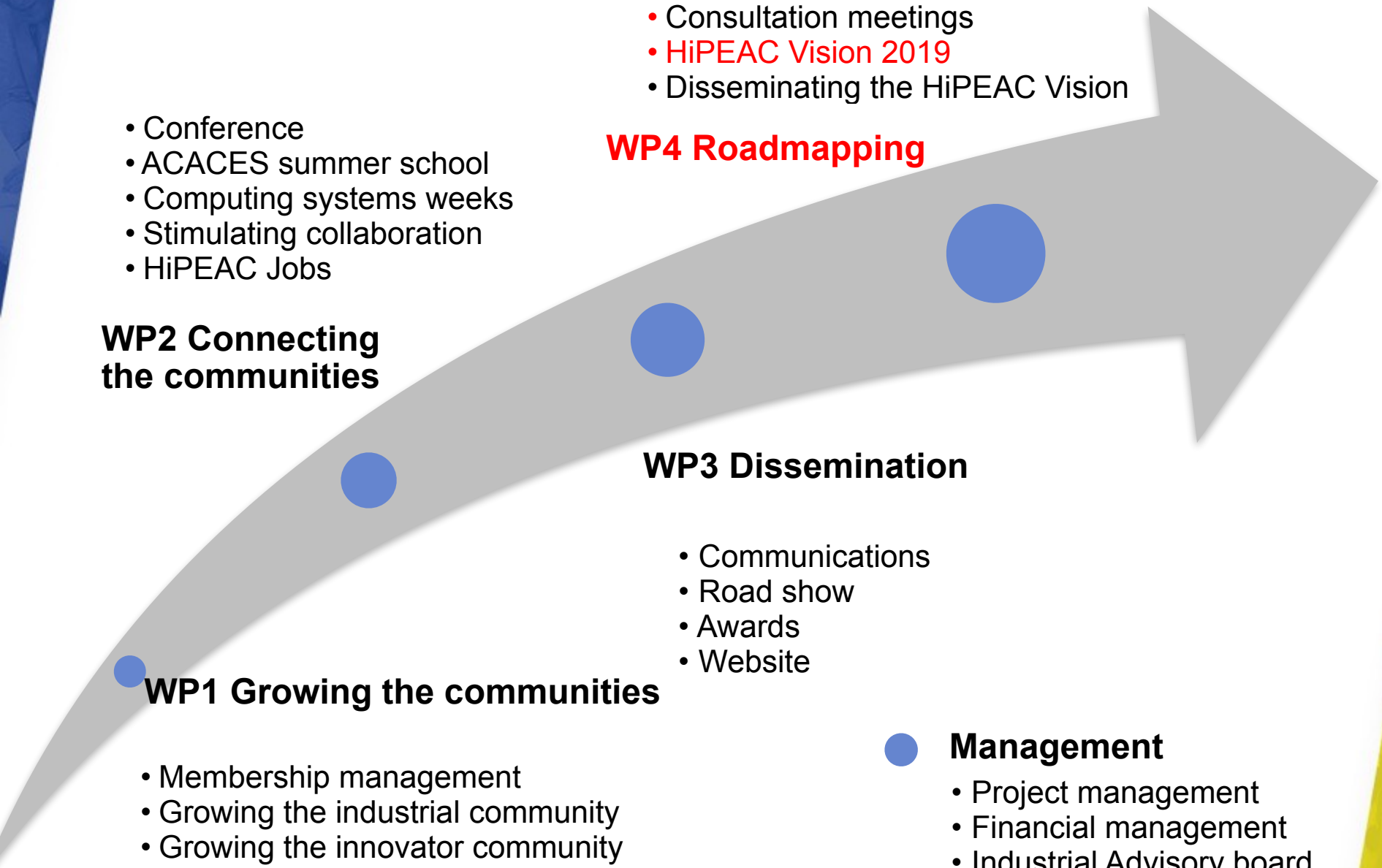
HiPEAC's mission is to steer and increase the European research in the area of high-performance and embedded computing systems,

and stimulate cooperation between
a) academia and industry and
b) computer architects and tool builders.



**13 partners, 522 members, 99 associated members, 423 affiliated members and 855 affiliated PhD students from 363 institutions in 40 countries.**

hipeac.net/members/stats/map



- Consultation meetings
- HiPEAC Vision 2019
- Disseminating the HiPEAC Vision

**WP4 Roadmapping**

- Conference
- ACACES summer school
- Computing systems weeks
- Stimulating collaboration
- HiPEAC Jobs

**WP2 Connecting the communities**

**WP3 Dissemination**

- Communications
- Road show
- Awards
- Website

**WP1 Growing the communities**

- Membership management
- Growing the industrial community
- Growing the innovator community
- Growing the stakeholder community
- Growing the new member states membership

**Management**

- Project management
- Financial management
- Industrial Advisory board

# THE HIPEAC VISION

The ***HiPEAC Vision*** Document is a deliverable of the **c**oordination and **s**upport **a**ction on **Hi**gh **P**erformance and **E**mbedded **A**rchitecture and **C**ompilation

The last HiPEAC Vision Document was published in January 2017.

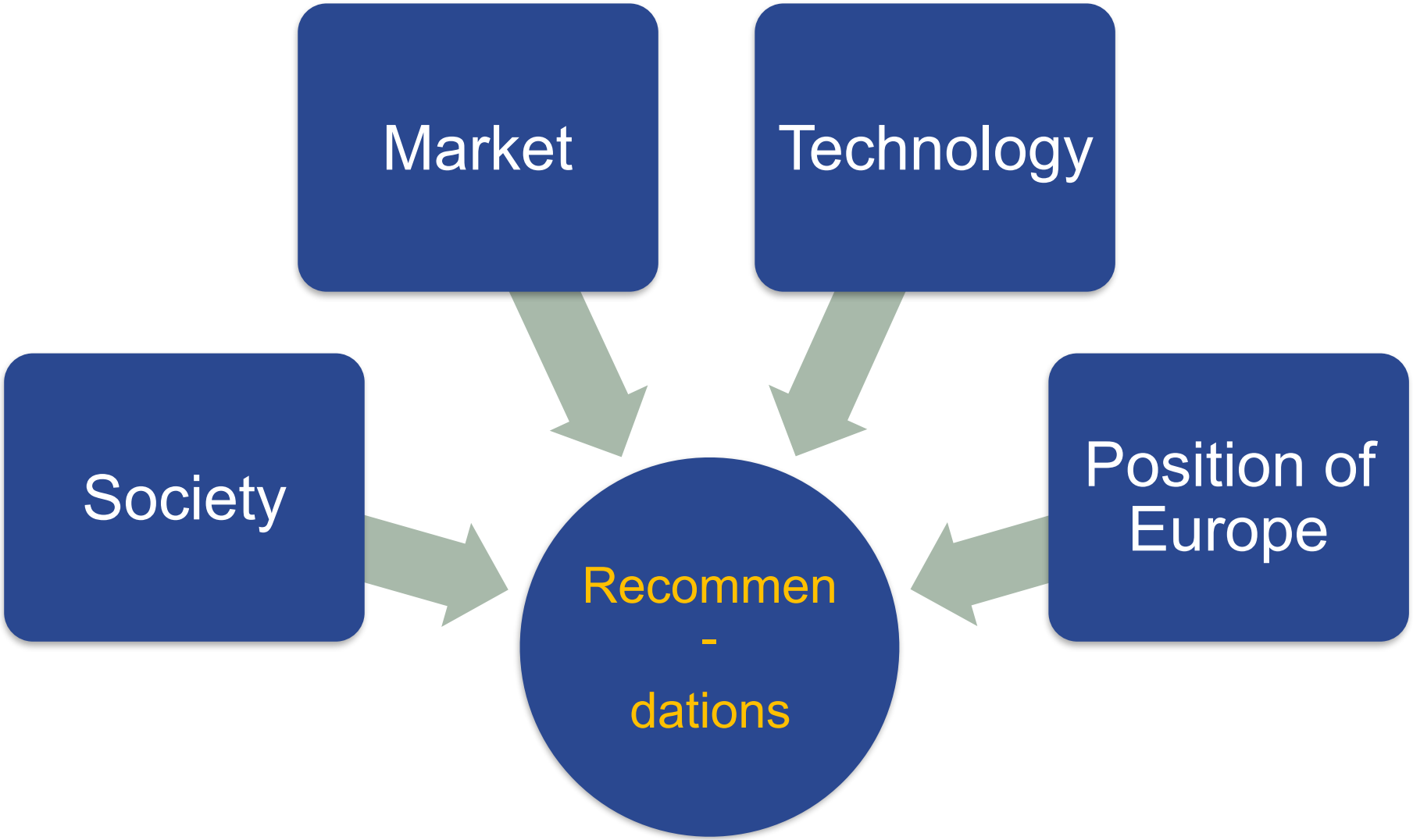
The next version is on-going (printed version for end 2018)


2008 2009 2011 2013 2015 2017

January 2017 version is available at:

**http://hipeac.net/vision**

# STRUCTURE HIPEAC VISION 2017



Guaranteeing trust

Improving performance and energy efficiency

Mastering complexity

Security, safety, privacy

Mastering parallelism and heterogeneity

Beyond predictability by design

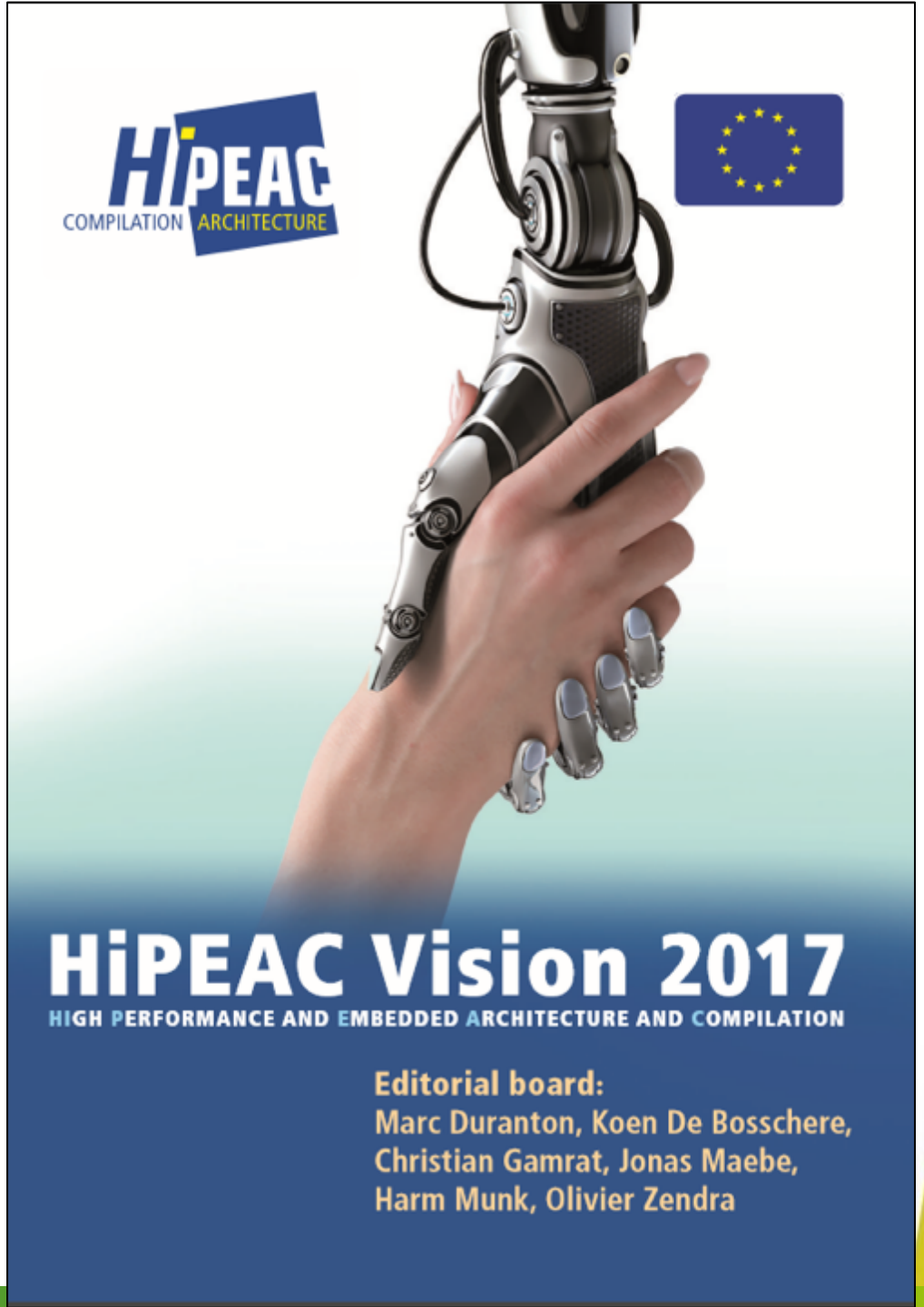

Increasing ICT workforce

*Holistic view*

HiPEAC COMPILATION ARCHITECTURE

## FOR FURTHER READING

**http://hipeac.net/vision**



# CONCLUSION: WE LIVE AN EXCITING TIME!

# Thank you for your attention

Special thank you to Olivier Bichler, Denis Dutoit, Christian Gamrat, Carlo Reita and Yann LeCun for their slides I borrowed.

marc.duranton@cea.fr

leti
Centre de Grenoble
17 rue des Martyrs
38054 Grenoble Cedex

list
Centre de Saclay
Nano-Innov PC 172
91191 Gif sur Yvette Cedex