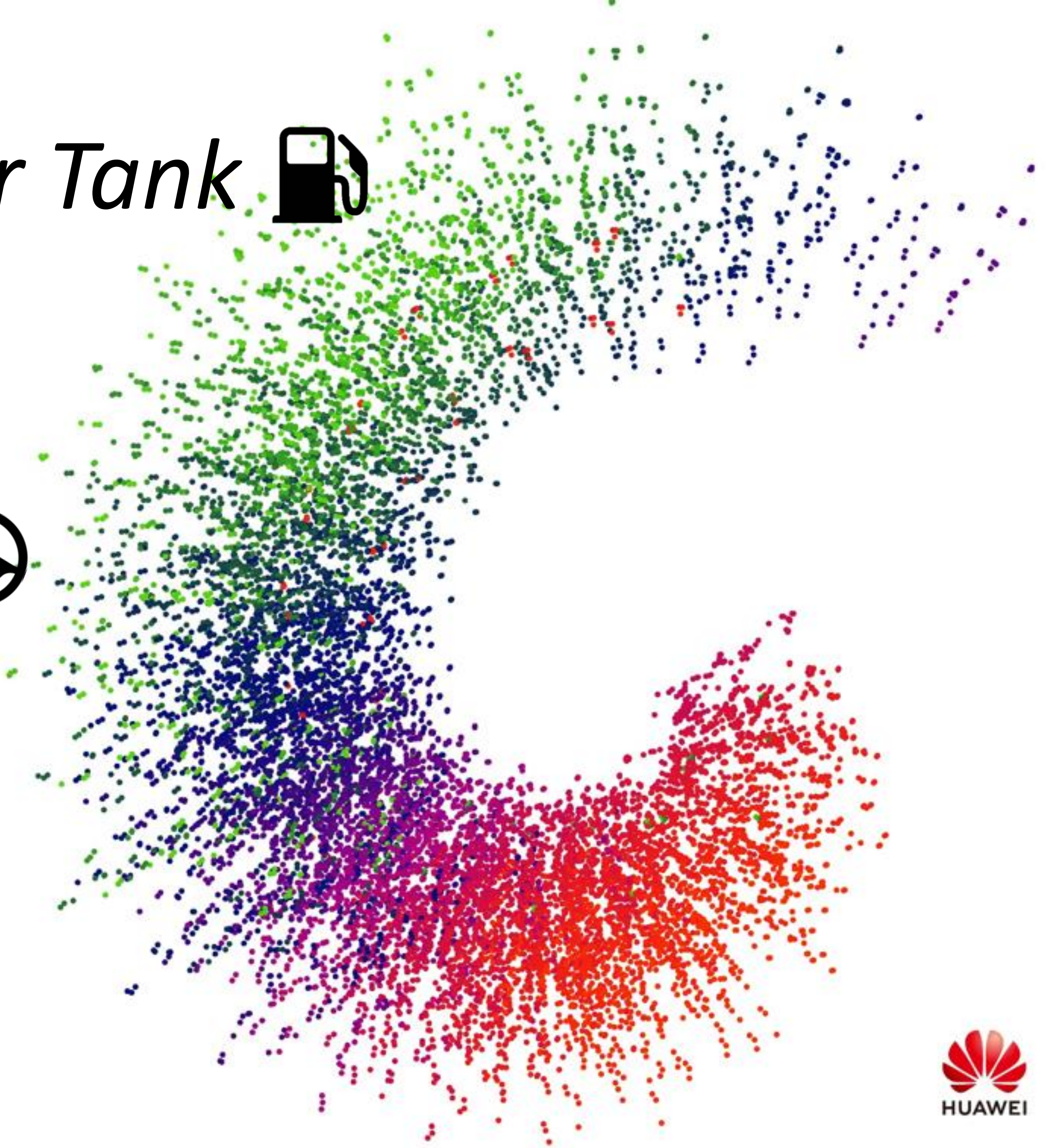
Put a Tiger in Your Tank

What benefits can hardware bring to intelligent, self-driving network operation ?



TMA, Paris 21th June 2019

Dario Rossi
dario.rossi@huawei.com

Put a Tiger in Your Tank

(I assume part of the audience
has not direct memories of the slogan,
as I myself was only 7yr old back then)

"Put a tiger in your tank" a successful 1959 slogan by Emery Smith

https://www.campaignlive.co.uk/article/history-advertising-quite-few-objects-43-esso-tiger-tails/1151980

Put a 🐯 Tiger in Your Tank ⛽

Before even considering how an
"AI network tiger" could look like,
let consider a 10,000 feet high view of
the current network problems, to see
why we would even need such a tiger ?

"Put a tiger in your tank" a successful 1959 slogan by Emery Smith
https://www.campaignlive.co.uk/article/history-advertising-quite-few-objects-43-esso-tiger-tails/1151980


Absence
of information
Encryption
operational obscurity

Excess
of information

Data deluge
operational overload

Opportunity

for AI & ML

Ascend

Unified AI chip architecture

Tackle

operational obscurity &
operational overload

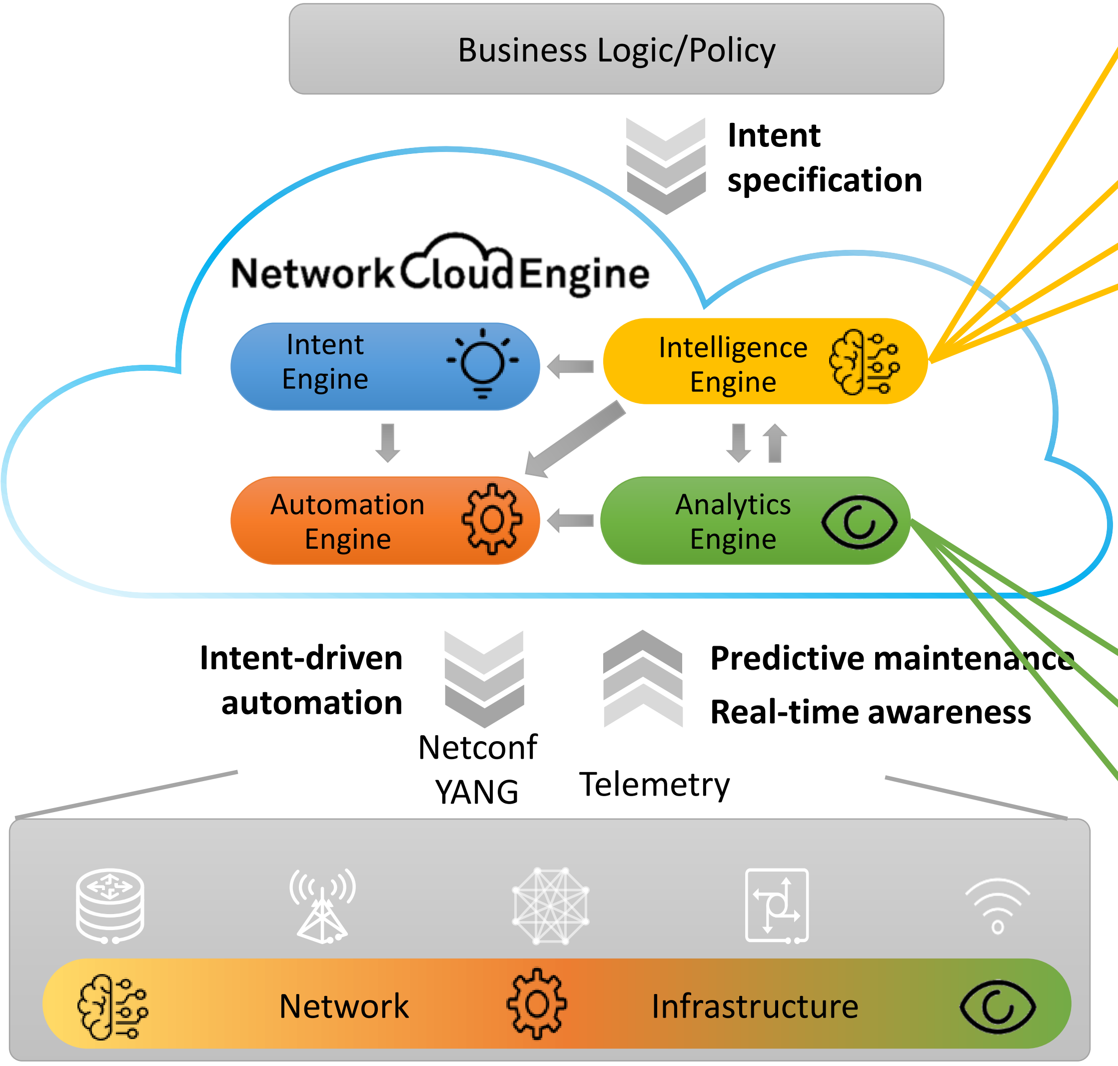
Huawei's IDN in a nutshell

(We are hiring)

Topics of Network AI team in Paris

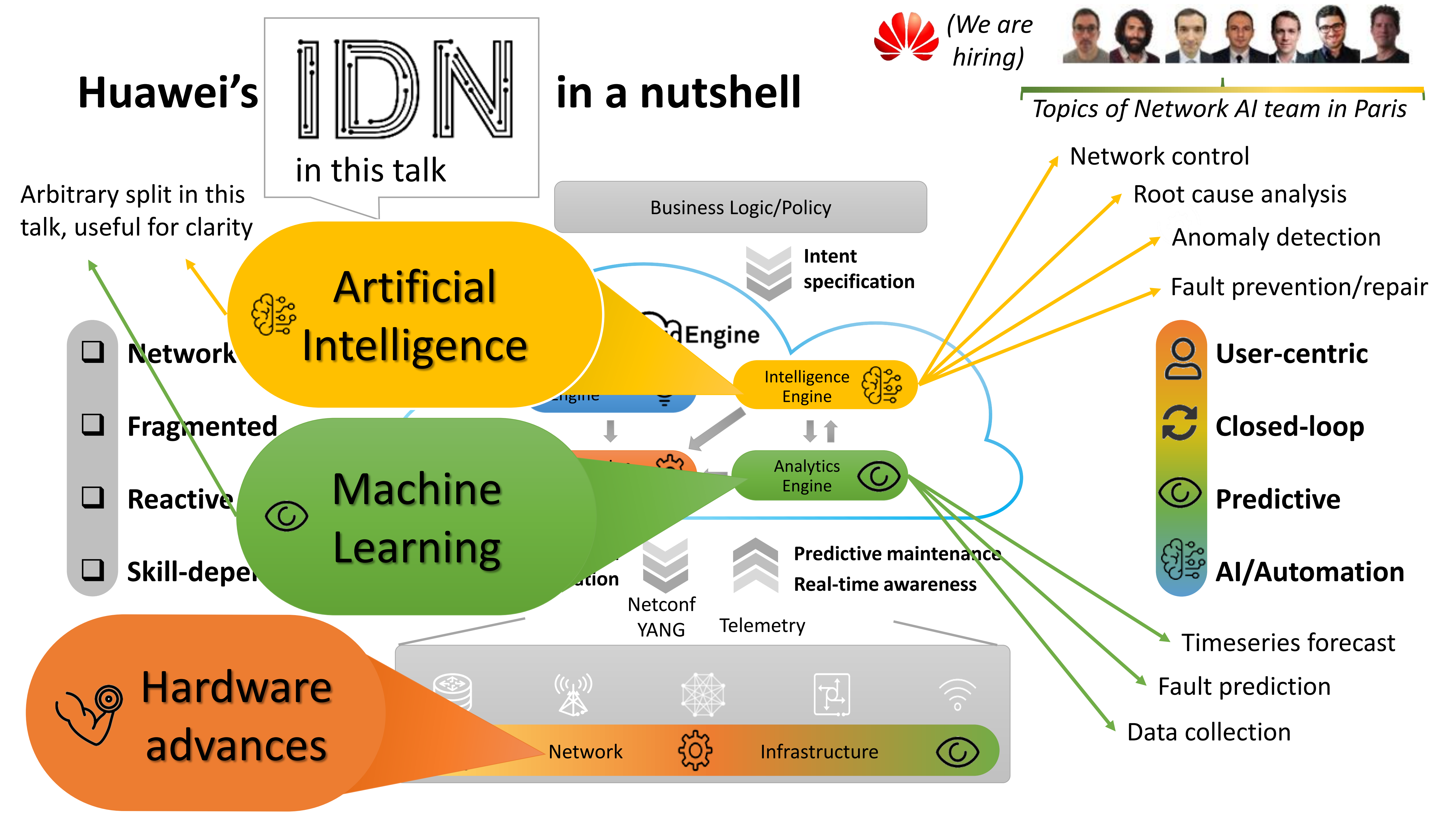
- Network control
- Root cause analysis
- Anomaly detection
- Fault prevention/repair

- ❑ **Network-centric**
- ❑ **Fragmented**
- ❑ **Reactive**
- ❑ **Skill-dependent**



- **User-centric**
- **Closed-loop**
- **Predictive**
- **AI/Automation**

- Timeseries forecast
- Fault prediction
- Data collection



Huawei's IDN in a nutshell
in this talk
(We are hiring)
Topics of Network AI team in Paris
Network control
Root cause analysis
Anomaly detection
Fault prevention/repair
Arbitrary split in this talk, useful for clarity
Business Logic/Policy
Intent specification
Artificial Intelligence
Engine
Intelligence Engine
Network
Fragmented
Reactive
Skill-depe
Machine Learning
Analytics Engine
User-centric
Closed-loop
Predictive
AI/Automation
Predictive maintenance
Real-time awareness
Netconf YANG
Telemetry
Timeseries forecast
Fault prediction
Data collection
Hardware advances
Network
Infrastructure

Agenda

Hardware advances

Network data

Understand the network

Control the network

- ❑ History
- ❑ Trends
- ❑ AI chips

So much data,

so few labels

- ❑ Explicability
- ❑ Evolution
- ❑ Security

- ❑ Closing the loop
- ❑ Humans & the loop
- ❑ System aspects

Aim of this talk

Ingredients & recipes for good AI/ML use in networks

+ Flash few examples out of our activities

Agenda

Hardware advances | Network data | Understand the network | Control the network

- ❑ History
- ❑ Trends
- ❑ AI chips

So much data,

so few labels

- ❑ Explicability
- ❑ Evolution
- ❑ Security

- ❑ Closing the loop
- ❑ Humans & the loop
- ❑ System aspects

Aim of this talk → **Ingredients & recipes for good AI/ML use in networks** → **+ Flash few examples out of our activities**

Hardware advances

Hardware advances → Fire → Steam → Electricity → Logic → Network → AI

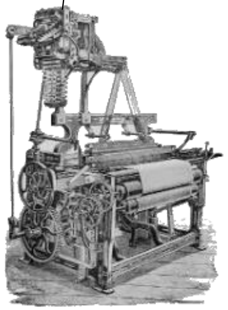
···1001010···

- ❑ History
- ❑ Trends
- ❑ AI chips

$3.5 \cdot 10^6$
years ago

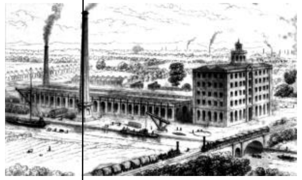
$10^3 \div 10^4$
years ago

1784
Mechanical
loom



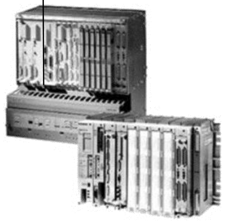
18^{th} century

1870
Assembly
line

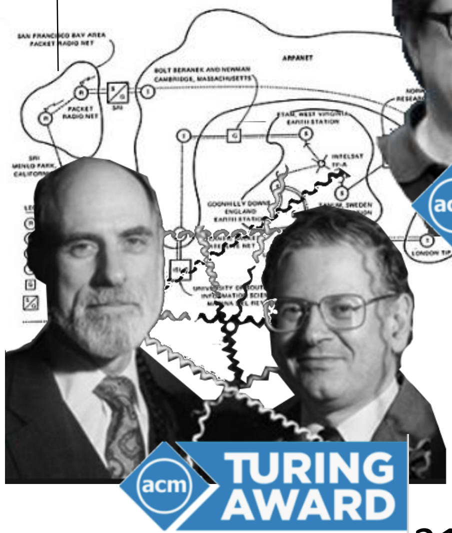


19^{th} century

1969
Programmable
Logic
Controller



1977
Internet
protocol
v0 demoed



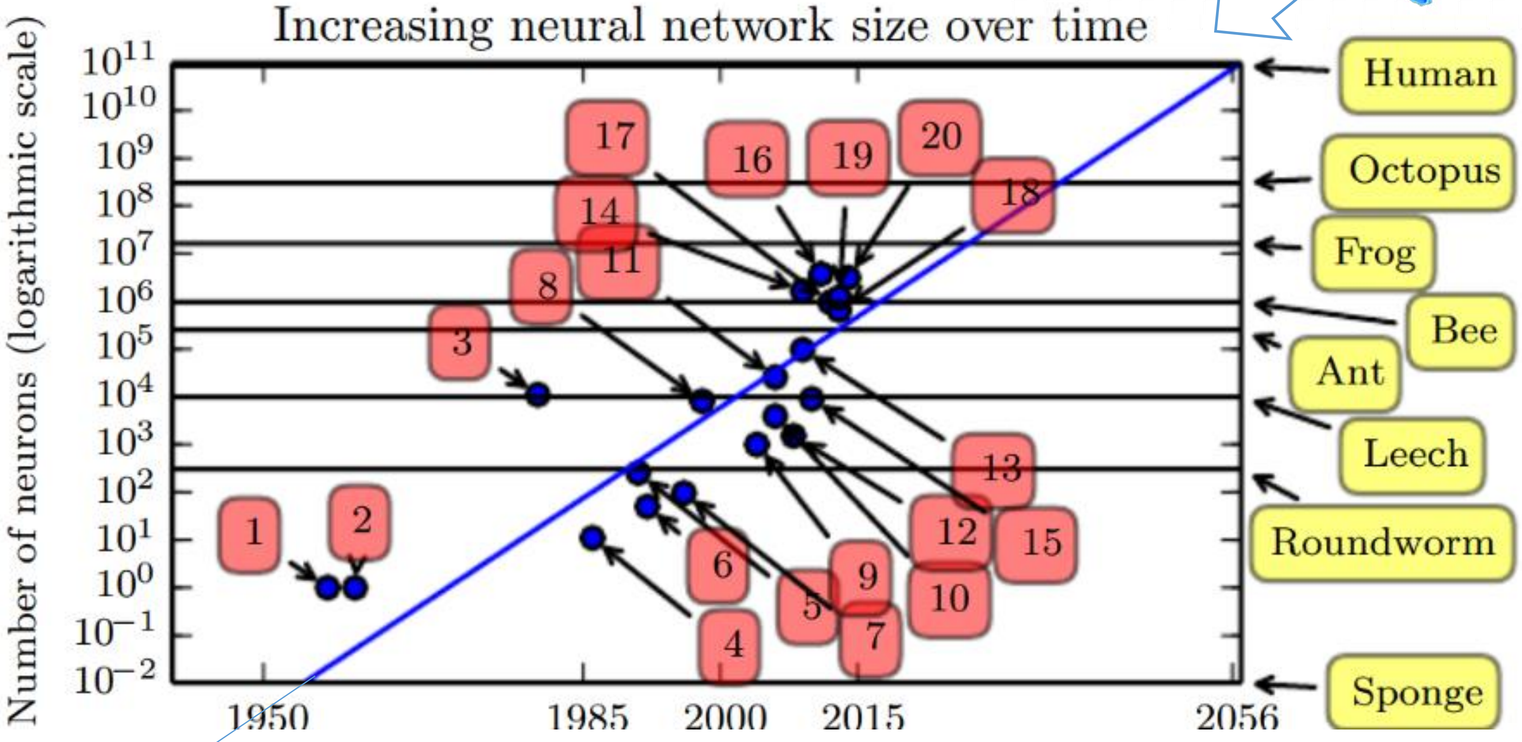
20^{th} century

21^{th} century

Deep neural networks trend

"Natural" neural network
~20 W

- ❑ History
- ❑ Trends
- ❑ AI chips



1.
Numbers of neurons increases faster than the number of transistors

Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep learning, MIT Press https://deeplearningbook.org

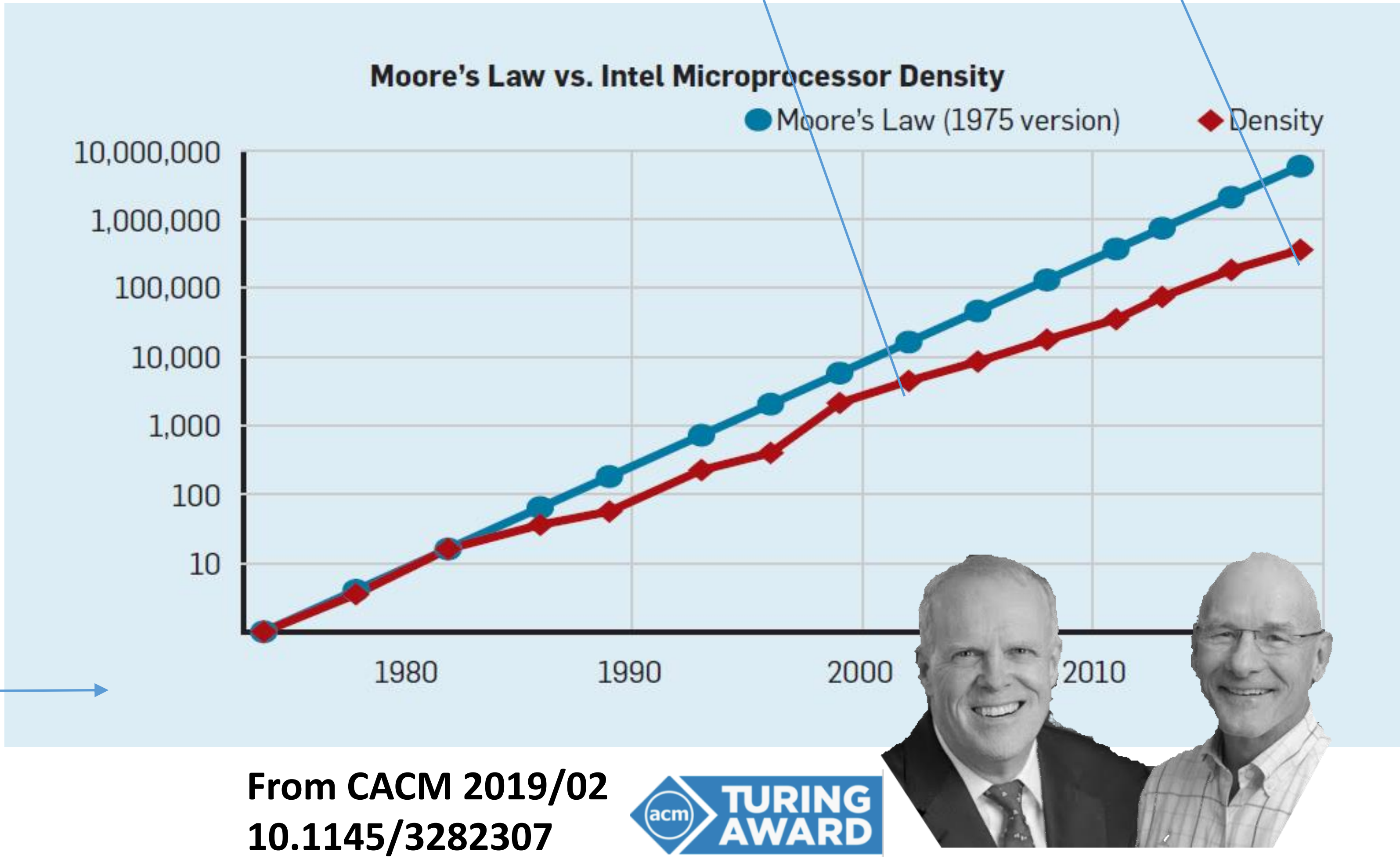
Hardware advances for general purpose computing

Hardware advances

- History
- Trends
- AI chips

2003, Moore law starts fading

2018, 15x gap



2.

Moore law will come to a stop eventually (the gap is already big)

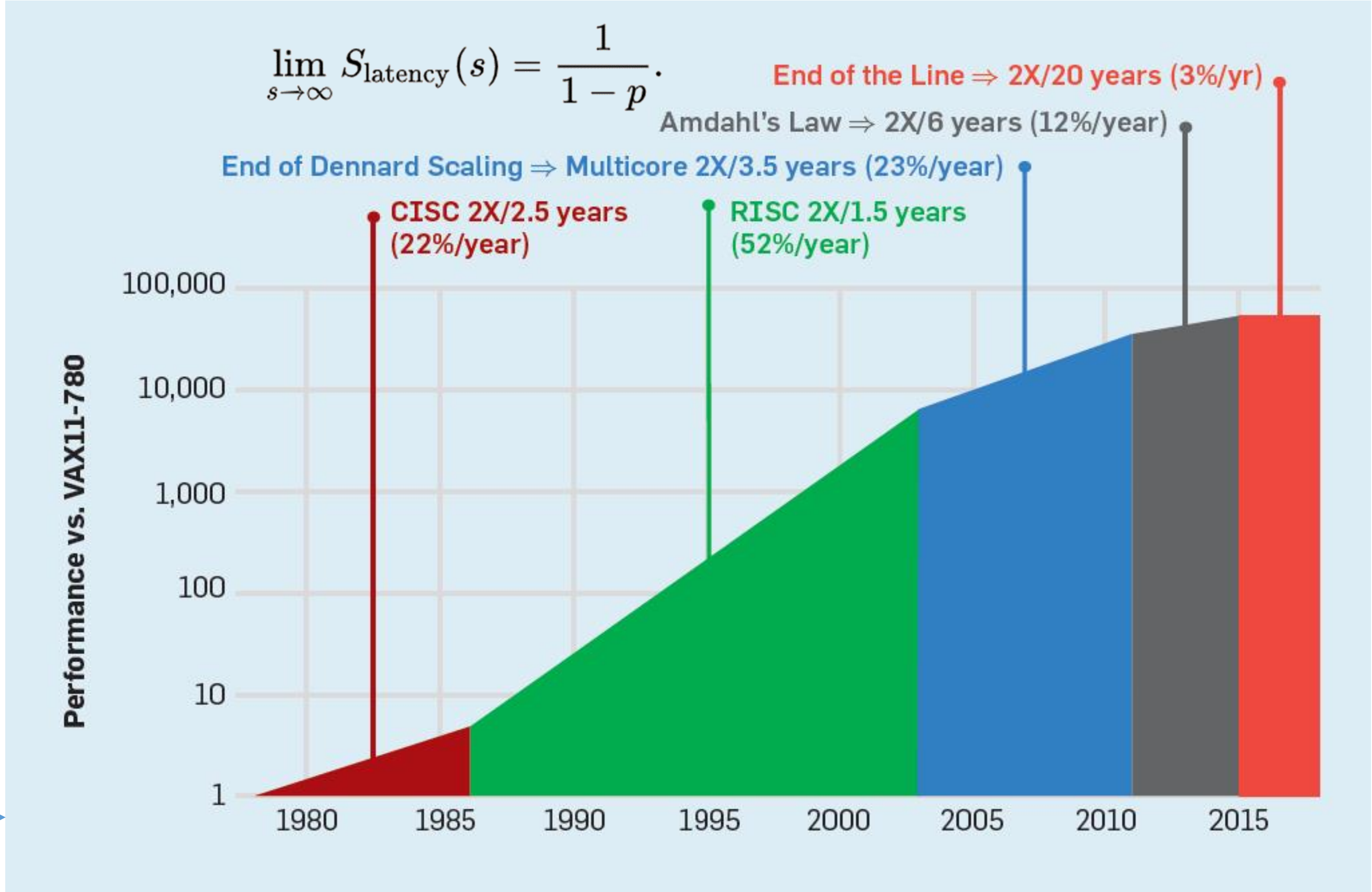
From CACM 2019/02
10.1145/3282307

TURING AWARD

Hardware advances for general purpose computing

Hardware advances

- History
- Trends
- AI chips



2b.
Computing performance increase is slowing down (it's not just Moore law…)

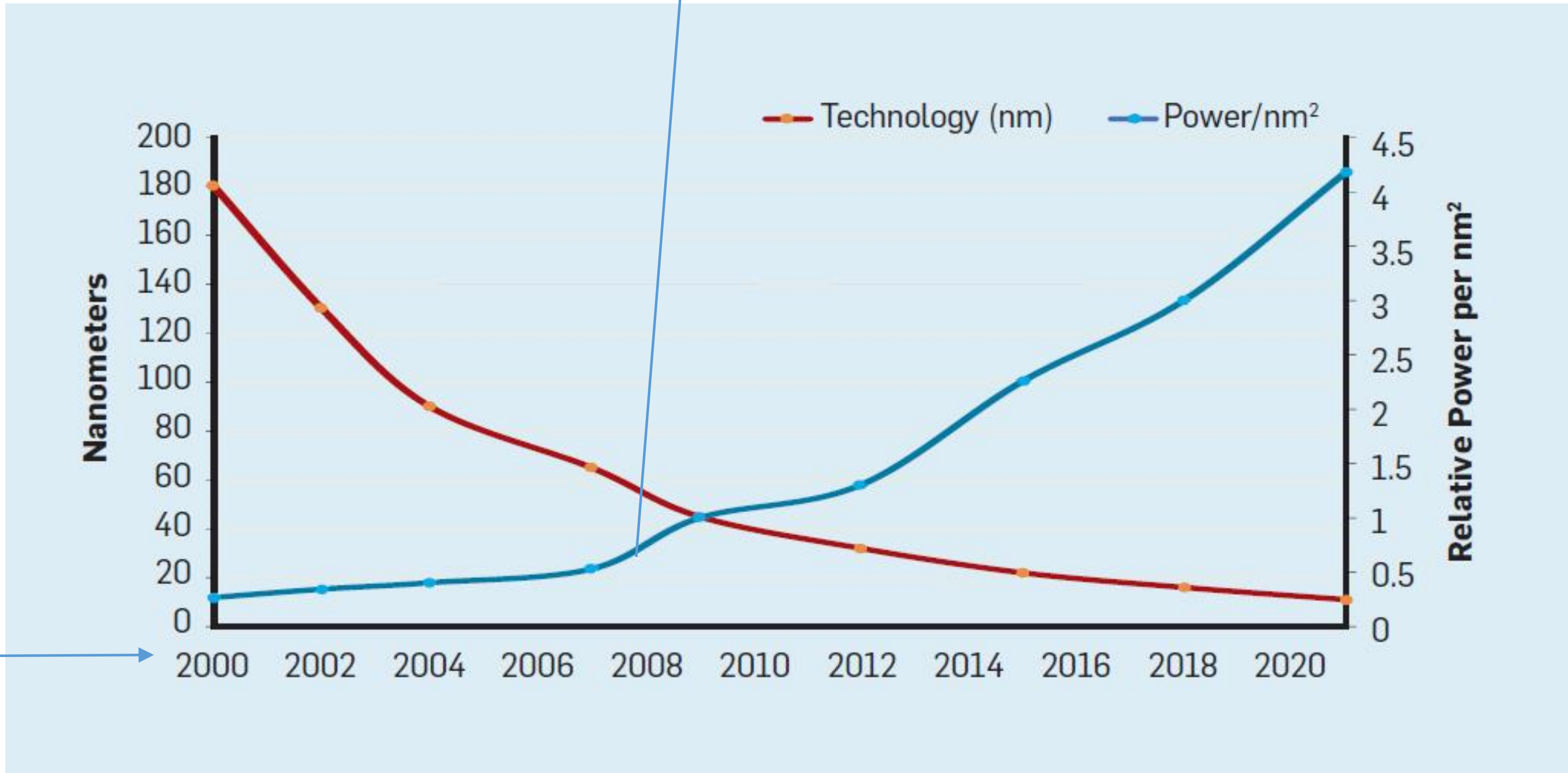
From CACM 2019/02
10.1145/3282307

Hardware advances for general purpose computing

Hardware advances

- History
- Trends
- AI chips

2008, end of Dennard scaling



2c.
Dennard scaling also practically stopped,
(⇨ multicore, but limit gain due to Amdahl law)

From CACM 2019/02
10.1145/3282307

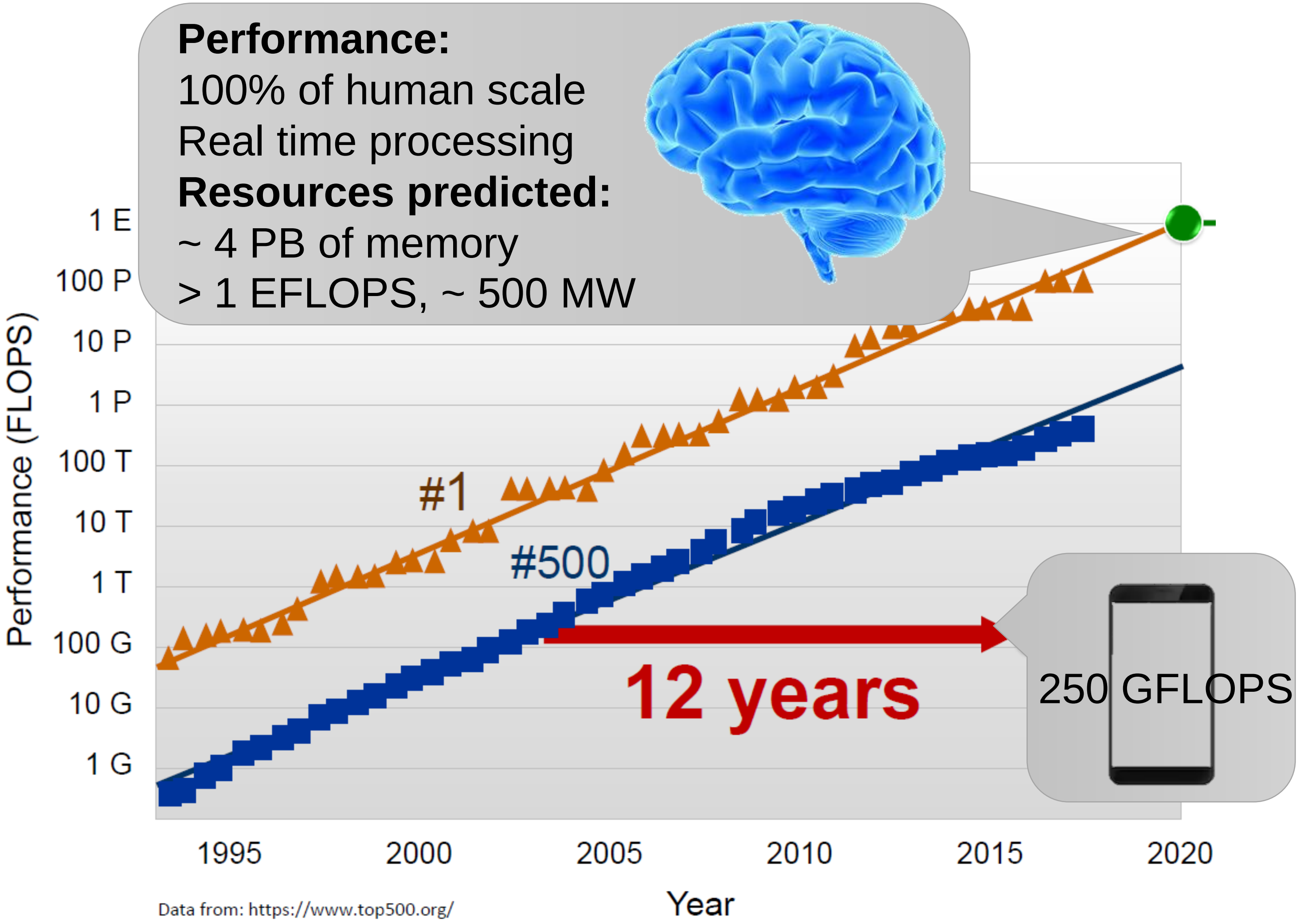
Hardware consumptions for artificial neural networks ?

Hardware advances

- ❑ History
- ❑ Trends
- ❑ AI chips

3.

General purpose designs hitting a power wall



Courtesy H.S. Philip Wong (黃漢森), Stanford & TSMC

Hardware bottleneck for packet processing ?

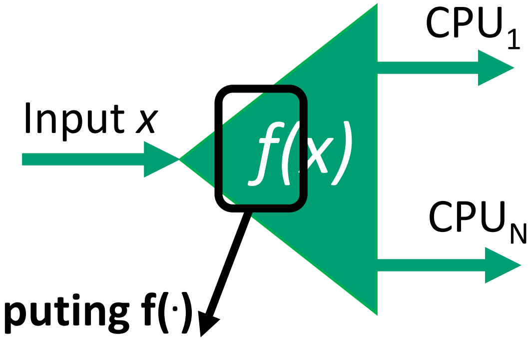
Hardware advances

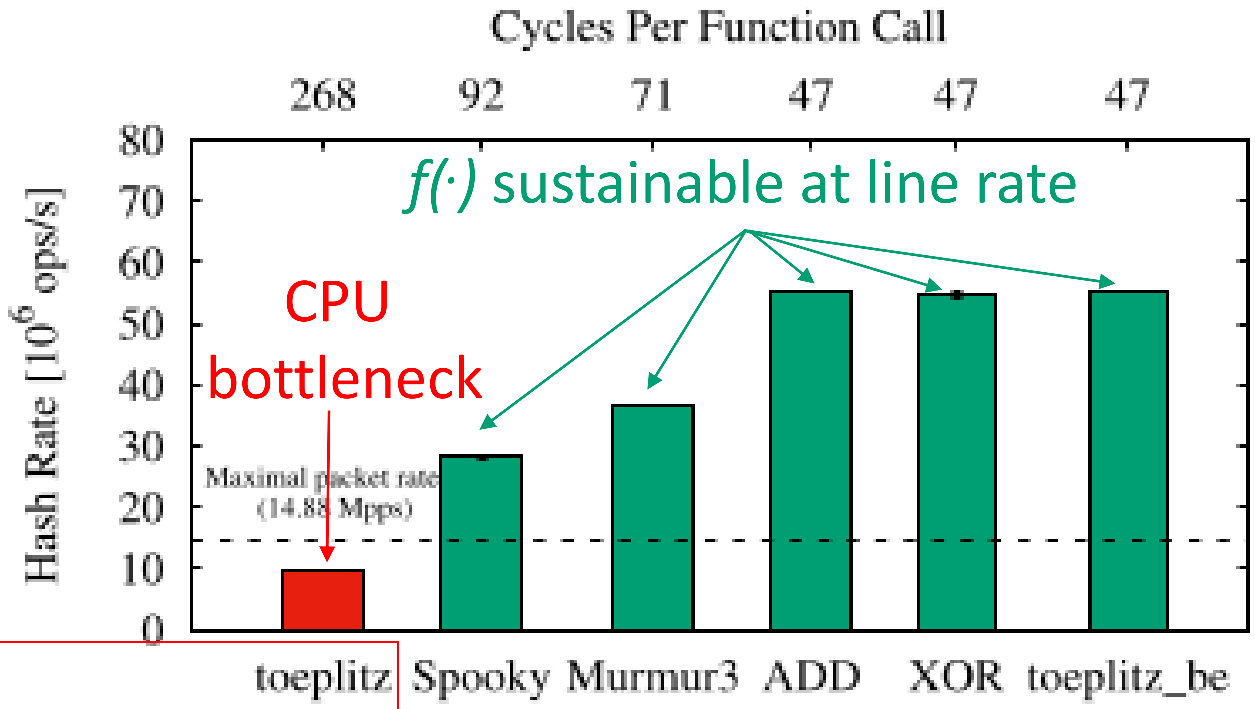
- History
- Trends
- AI chips

4.

Normal packet processing can hit a memory bottleneck

Example:
All-software flow-preserving load balancing





Same RSS function used by hardware NICs

"FloWatcher-DPDK: lightweight line-rate flow-level monitoring in software" TNSM'19

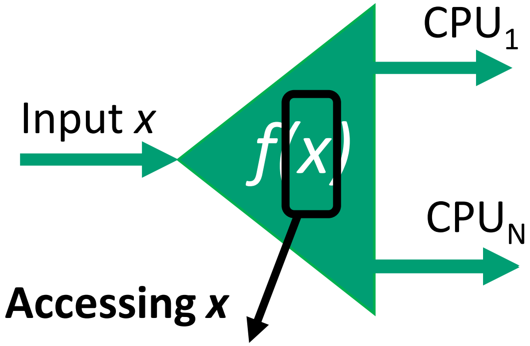
Hardware bottleneck for packet processing ?

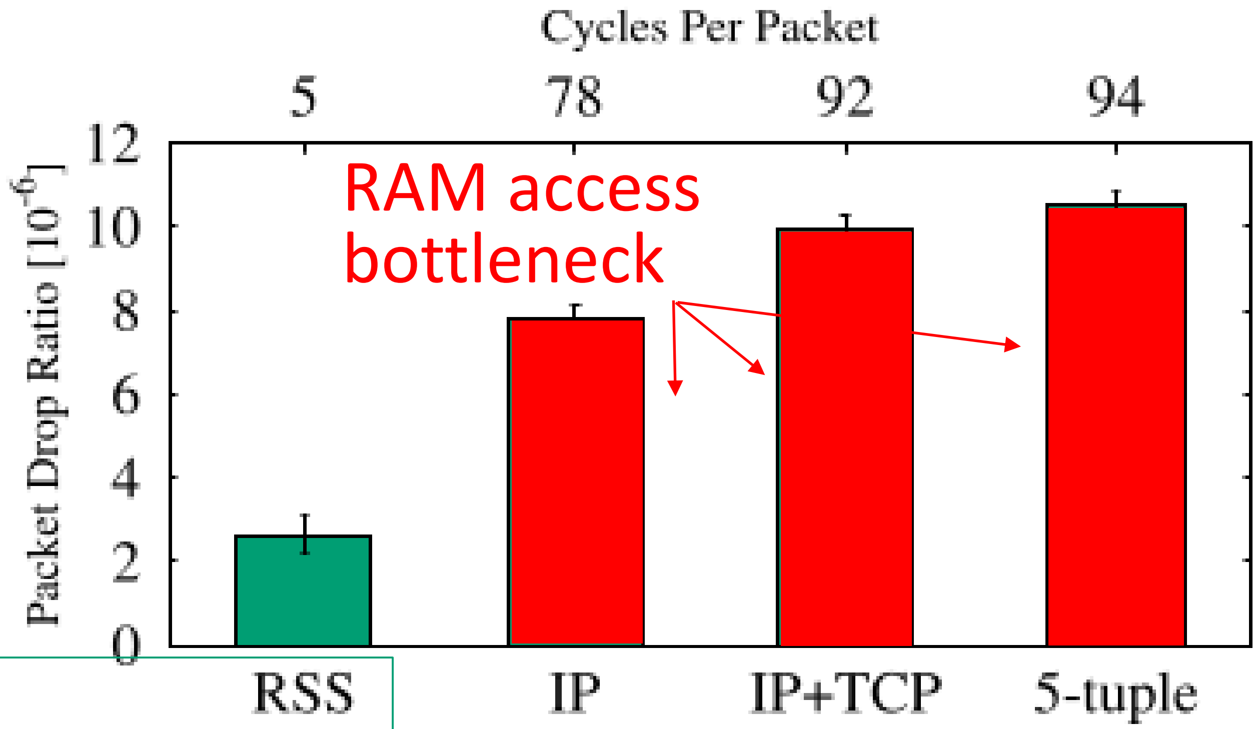
Hardware advances

- ❑ History
- ❑ **Trends**
- ❑ AI chips

4.
Normal packet processing can hit a memory bottleneck

Example:
All-software flow-preserving load balancing





=accessing *f(x)* results directly from NIC *mbuf*

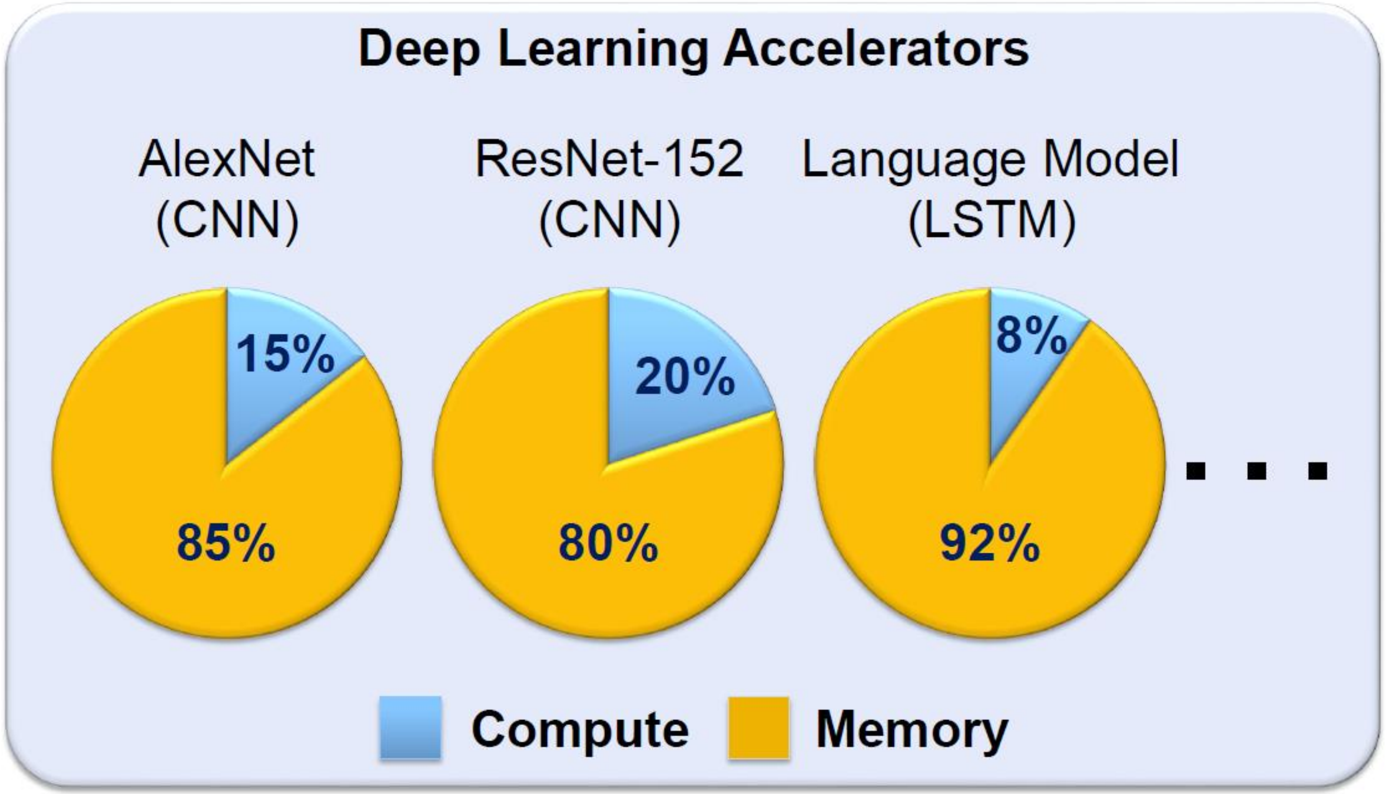
"FloWatcher-DPDK: lightweight line-rate flow-level monitoring in software" TNSM'19

Hardware bottleneck for AI processing ?

Hardware advances

- ❑ History
- ❑ Trends
- ❑ AI chips

4b.
General purpose designs
hit a memory wall for AI too!



Intel performance counter monitors 2 CPUs, 8-cores/CPU + 128GB DRAM

Source: S. Mitra (Stanford)

Hardware design trends

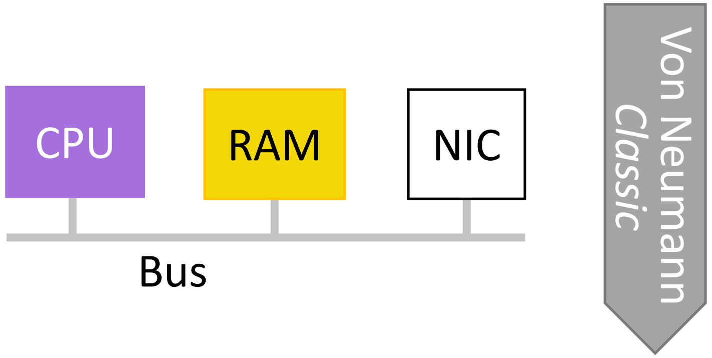
H.S. Philip Wong (黃漢森),
Stanford & TSMC

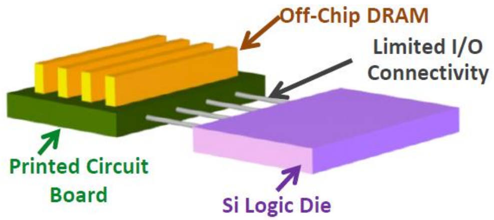
Hardware advances

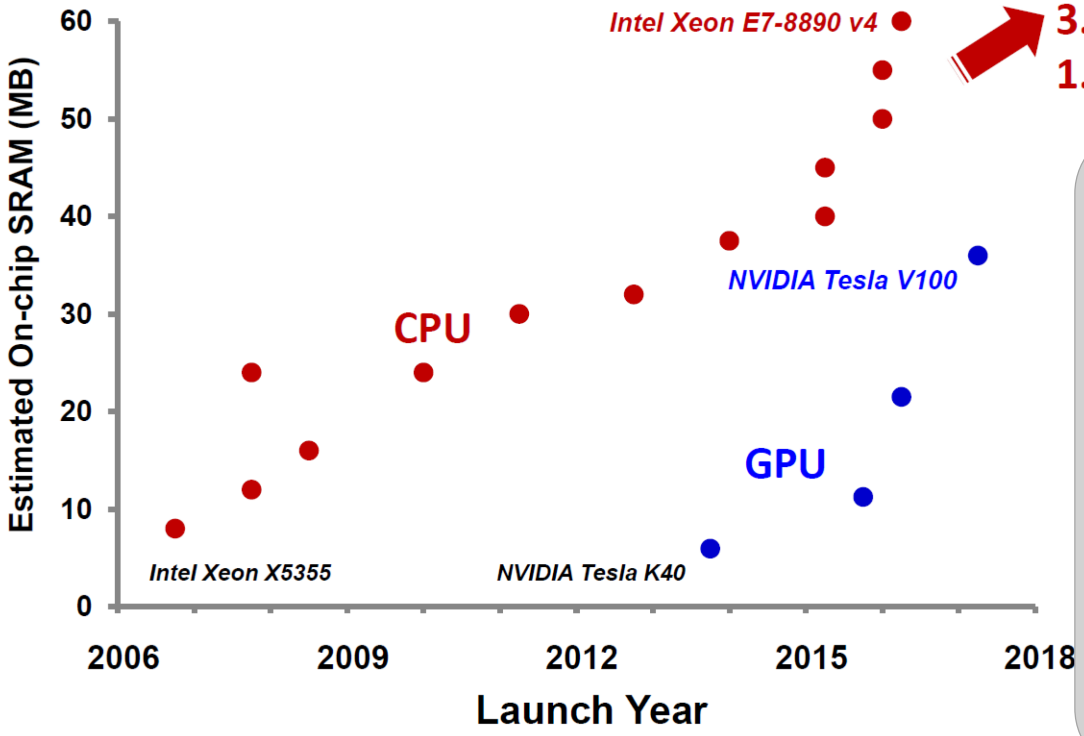
- ❑ History
- ❑ Trends
- ❑ AI chips

⇨

Go beyond classic
Von Neumann architectures







Source: W. Hwang, Prof. S. Mitra (Stanford)

3.8 GBytes @ 1.4 nm node

Recall
~ 4 PB memory

10^{11} neurons each connected to 10^4 synapses

H.S. Philip Wong (黃漢森),
Stanford & TSMC

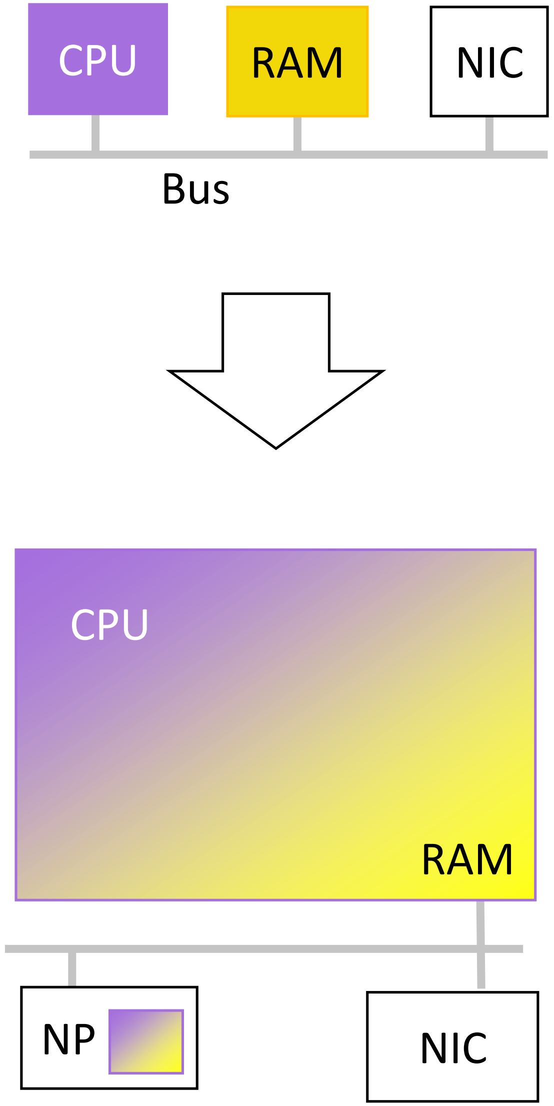
Hardware design trends

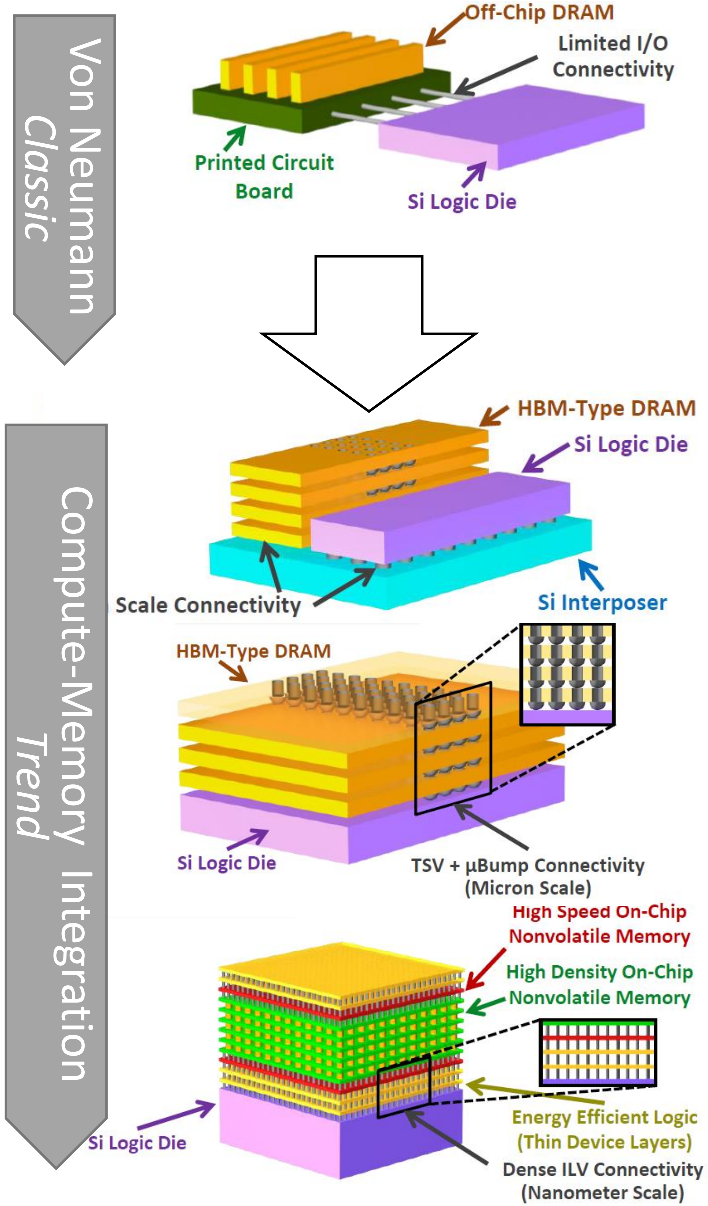
Hardware
advances

- ❑ History
- ❑ Trends
- ❑ AI chips

⇨

Go beyond classic
Von Neumann architectures
(⇨ memory-compute integration)





Hardware design trends

Hardware advances

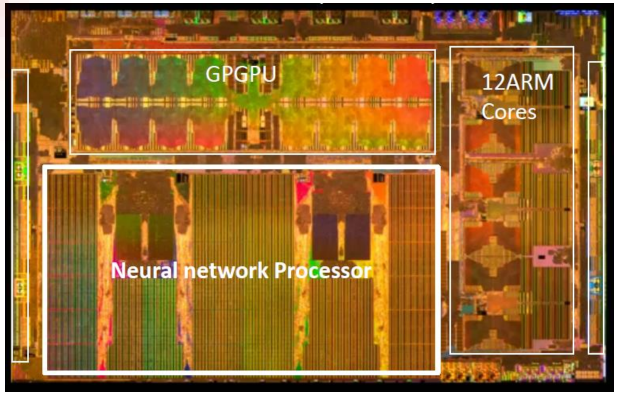
- ❑ History
- ❑ Trends
- ❑ AI chips

⇨

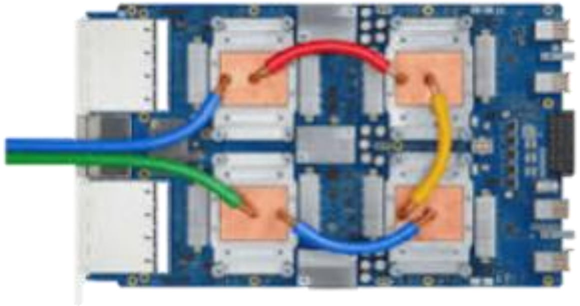
Go beyond classic
Von Neumann architectures
(⇨ design tailored for CNNs)

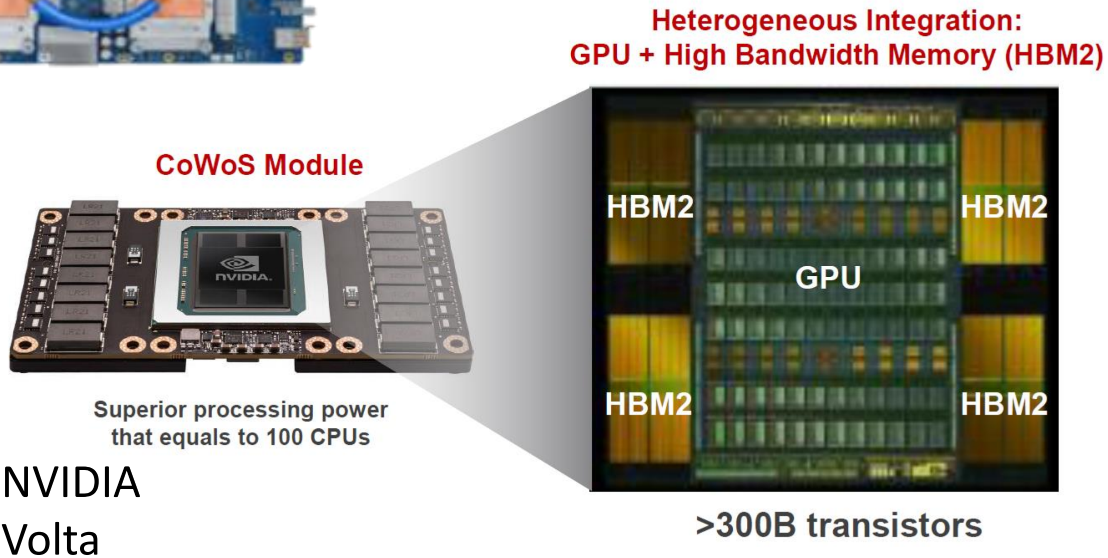
Huawei Ascend

Tesla FSD



Google TPU v3.0





NVIDIA Volta

Hardware design trends

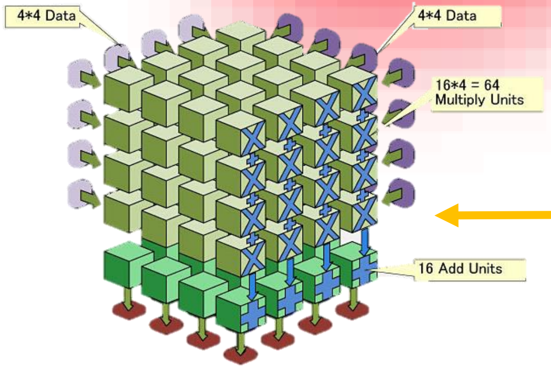
Hardware advances

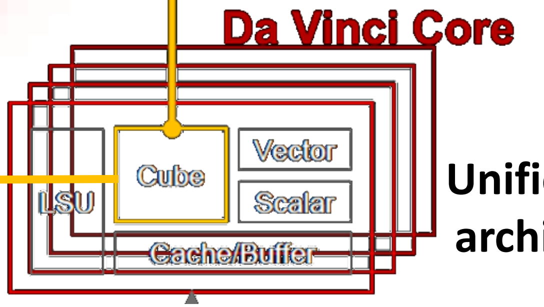
- ❑ History
- ❑ Trends
- ❑ AI chips

⇨

Go beyond classic
Von Neumann architectures
(⇨ flexible design, edge intelligence)

Huawei
Ascend





DaVinci
Unified chip
architecture

Ascend310 (Mini)
FP16： 8 TFLOPS
INT8： 16 TOPS

Power: 8W
Process: 12nm

Ascend910 (Max)
FP16: 256 TFLOPS
INT8: 512 TOPS

Power: 350W
Process: 7+ nm

DaVinci chips → DaVinci Server → DaVinci Cluster

Hardware design trends

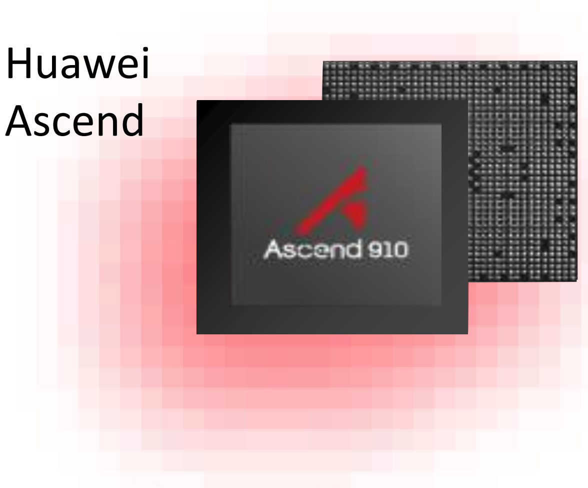
Hardware advances

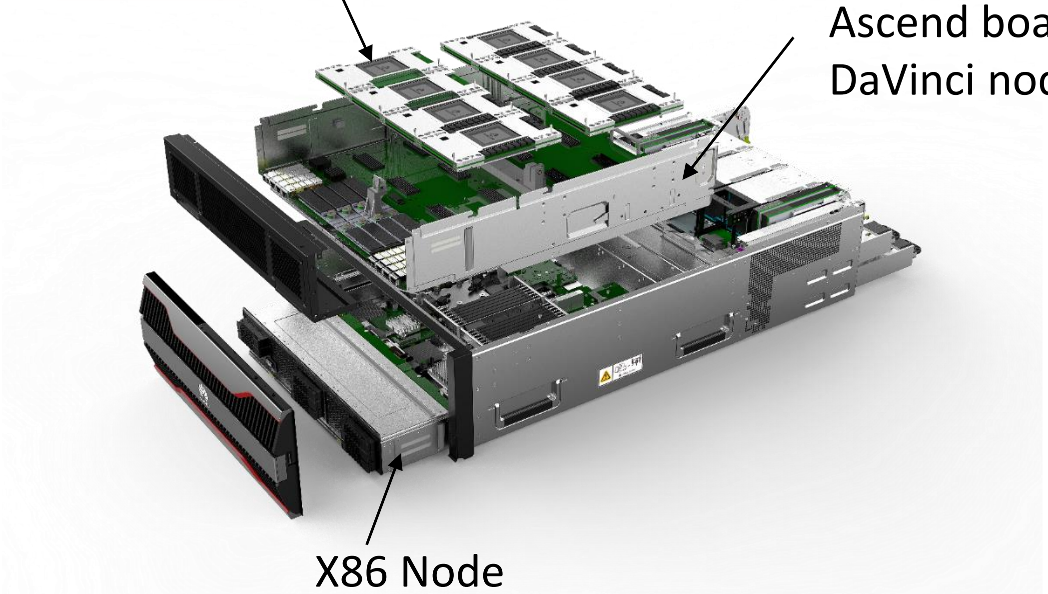
- ❑ History
- ❑ Trends
- ❑ AI chips

⇨

Go beyond classic
Von Neumann architectures
(⇨ flexible design, cloud)

Huawei Ascend





DaVinci chips → DaVinci Server → DaVinci Cluster

Hardware design trends

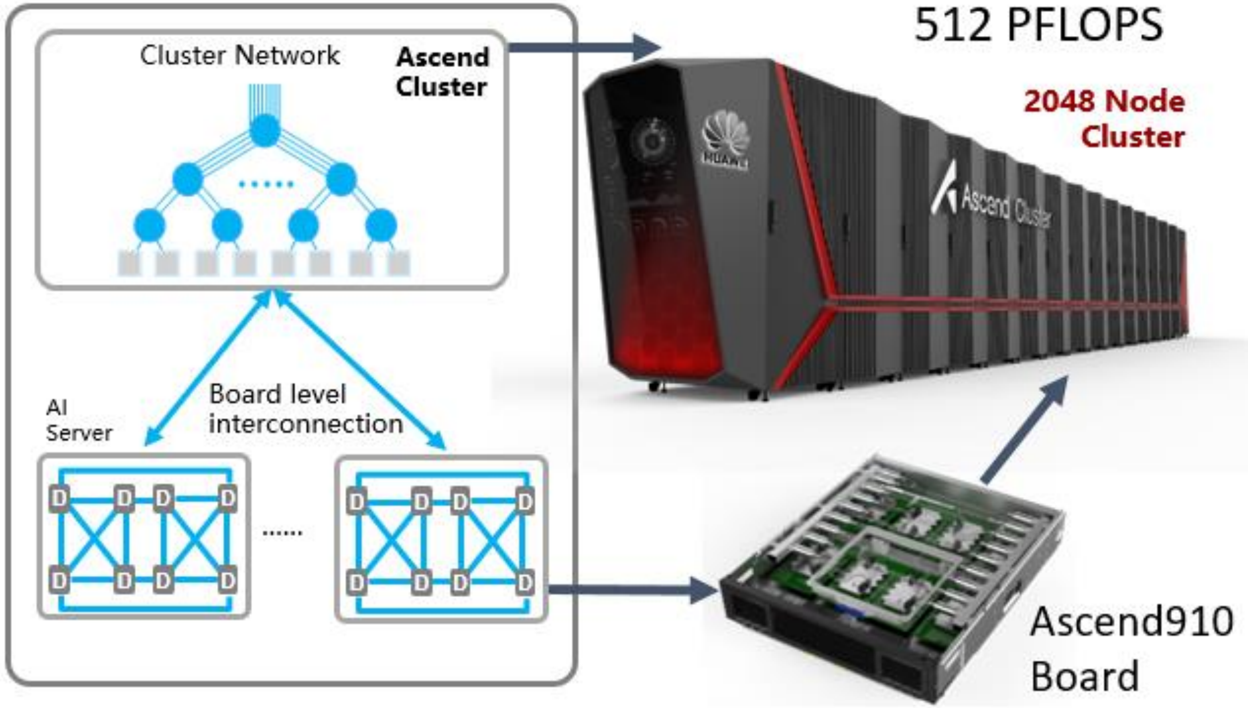
Hardware advances

- ❑ History
- ❑ Trends
- ❑ AI chips

➩

Go beyond classic
Von Neumann architectures
(➩ flexible design, hyperscale)

Huawei Ascend



DaVinci chips → DaVinci Server → DaVinci Cluster

Hardware is key, but software needed to exploit it!

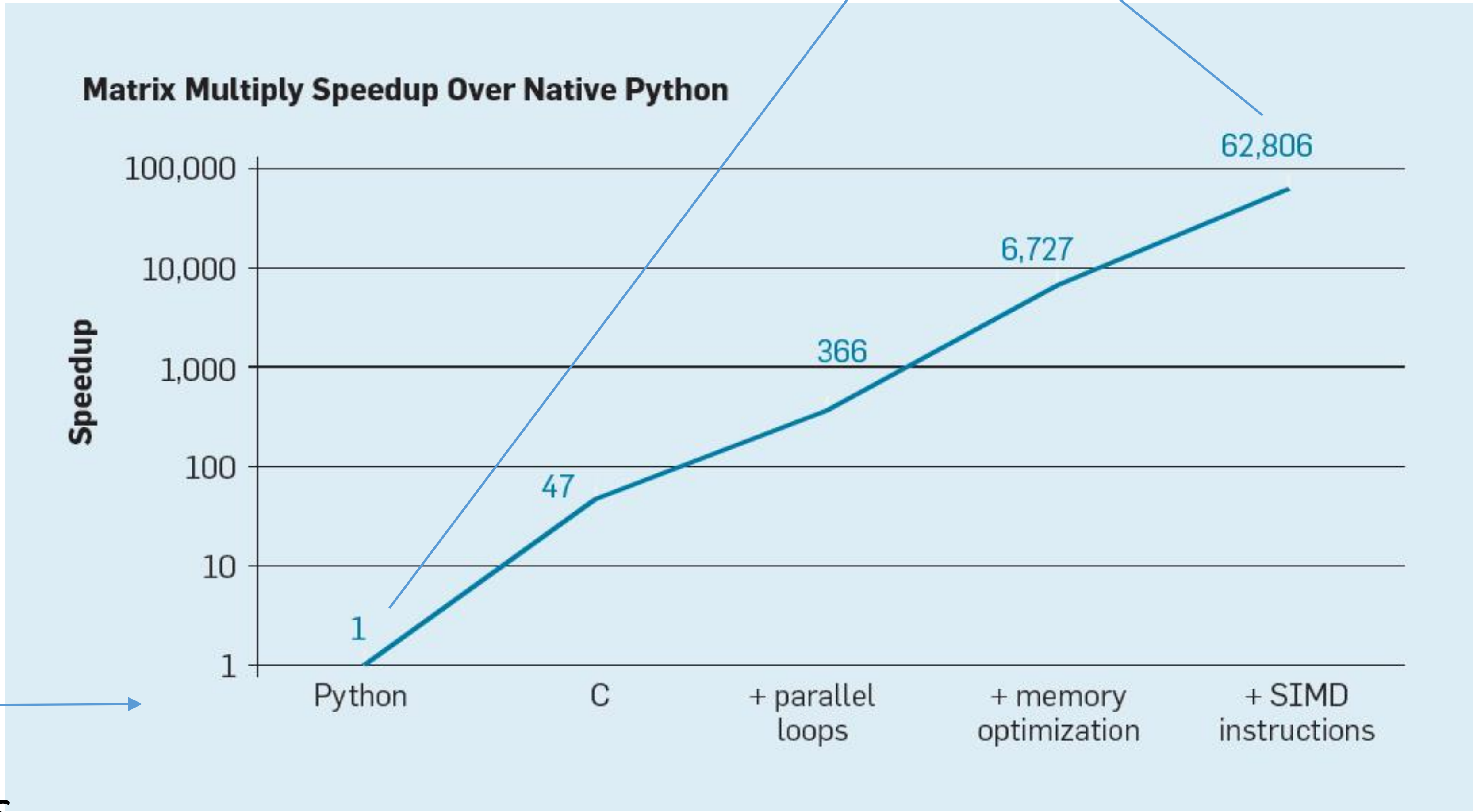
Hardware advances

- History
- Trends
- AI chips

➩

Go beyond classic Von Neumann architectures (➩ software still matters)

A bit extreme example, but valid point!



Ex. from Leiserson. C, “There plenty of room at the top”
Illustration from CACM 2019/02 10.1145/3282307

Hardware is key, but software needed to exploit it!

Hardware advances

- History
- Trends
- AI chips

Don't expect the L3 cross-compiler to just do *all* the magic

The more you know, the better *your* program

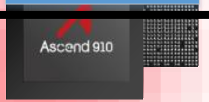
No free lunch…

Ascend software stack

Level 3 Library (written by novice programmer)		TBE LIB			
Level 3 Compiler (mathematical programming model)	TVM/XLA	TBE	Halide	Halide	
Level 2 Library (written by skilled programmer)	CudaNN/ CuBLAS	TIK LIB			
Level 2 Compiler (parallel/kernel programming model)	Cuda/OpenCL	TIK			
Level 1 Library (written by expert)		CCE Lib			Eigen/numpy
Low Level 1 Compiler (Intrinsic C) (Architecture defined programming)		CCE C	Intrinsic C	Intrinsic C	C/C++ LLVM
Instruction Set Architecture	GPU	NPU	Vision Processor	DSP	CPU + SVE

Software

Hardware



Agenda

Hardware advances

Network data

Understand the network

Control the network

- ❑ History
- ❑ Trends
- ❑ AI chips

So much data,

so few labels

- ❑ Explicability
- ❑ Evolution
- ❑ Security

- ❑ Closing the loop
- ❑ Humans & the loop
- ❑ System aspects

Aim of this talk

Ingredients & recipes for good AI/ML use in networks

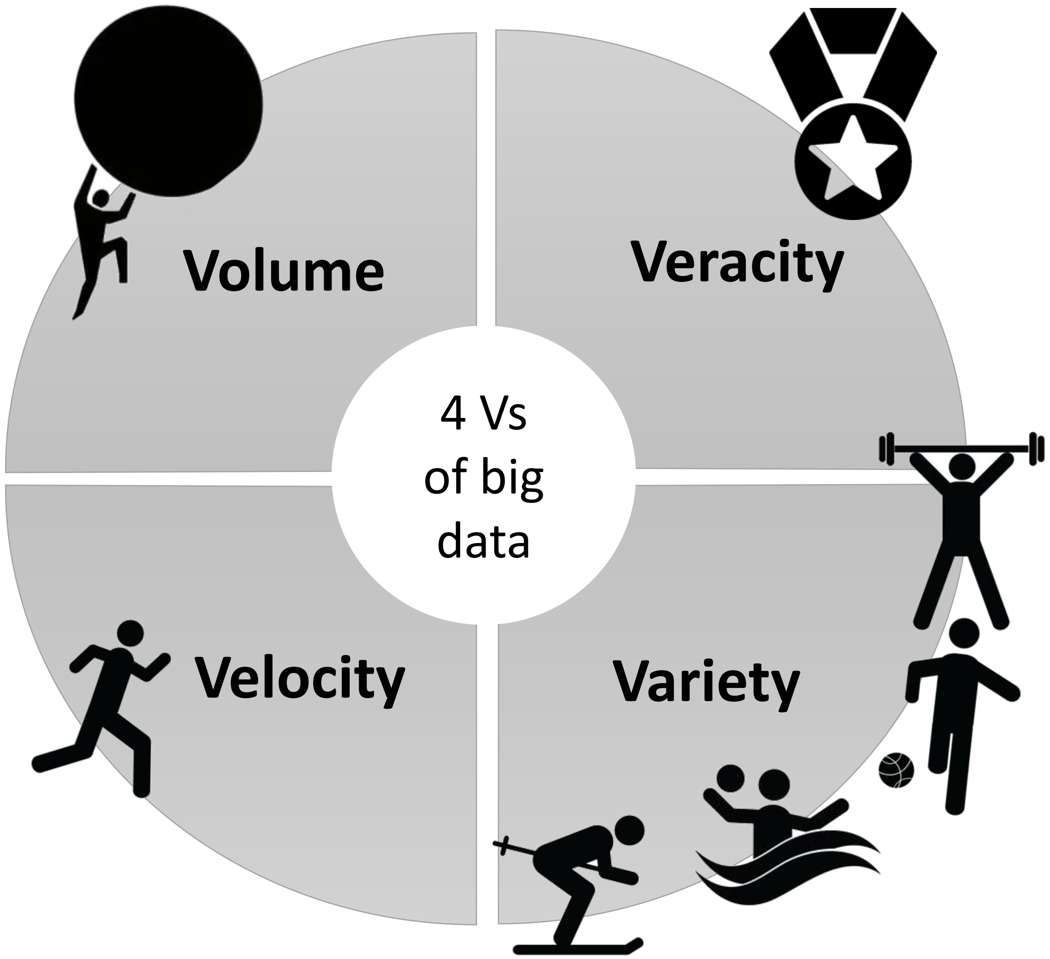
+ Flash few examples out of our activities

Networking data for ML / AI

Network data

So much data,

so few labels

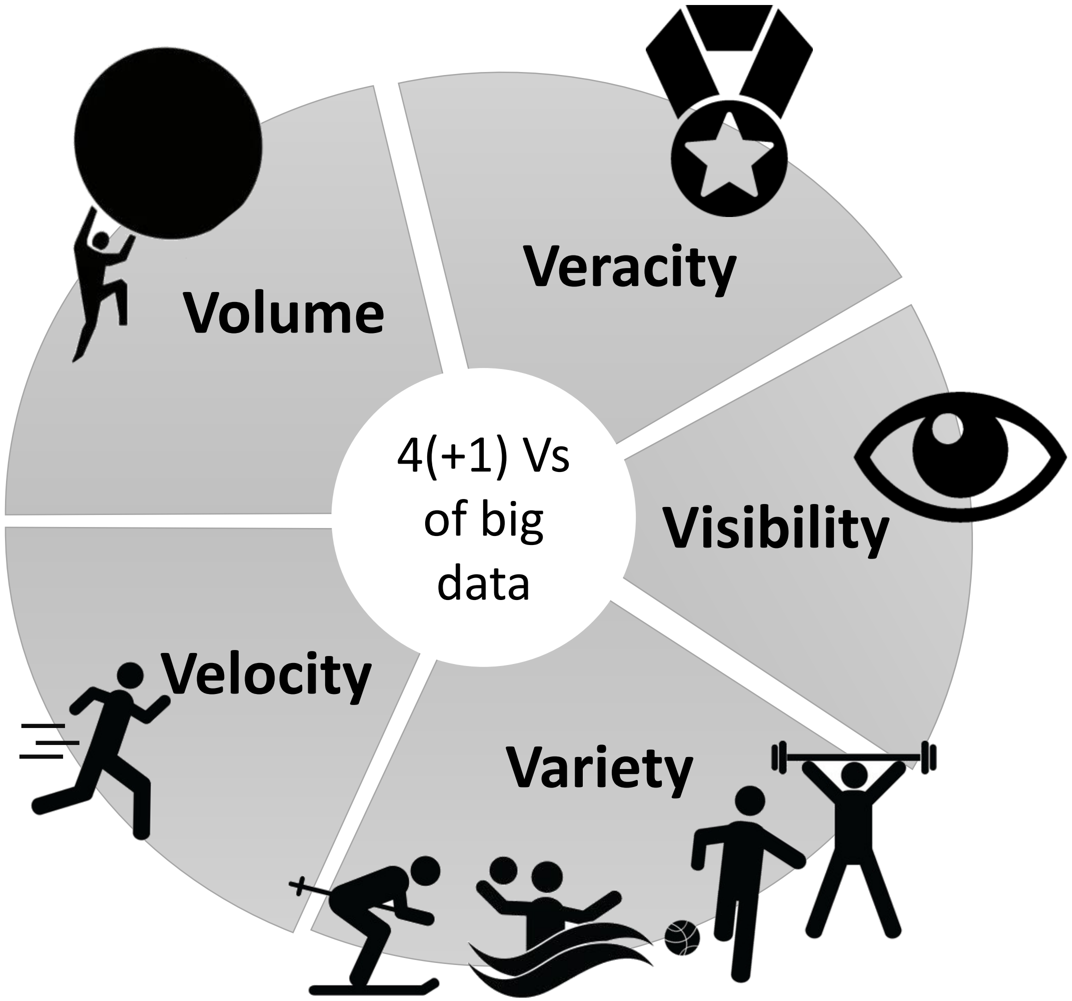


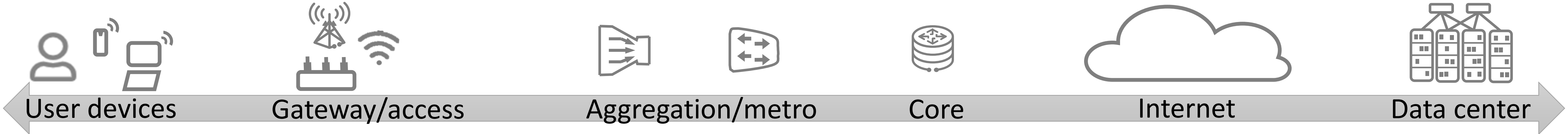
Networking data for ML / AI

Network data

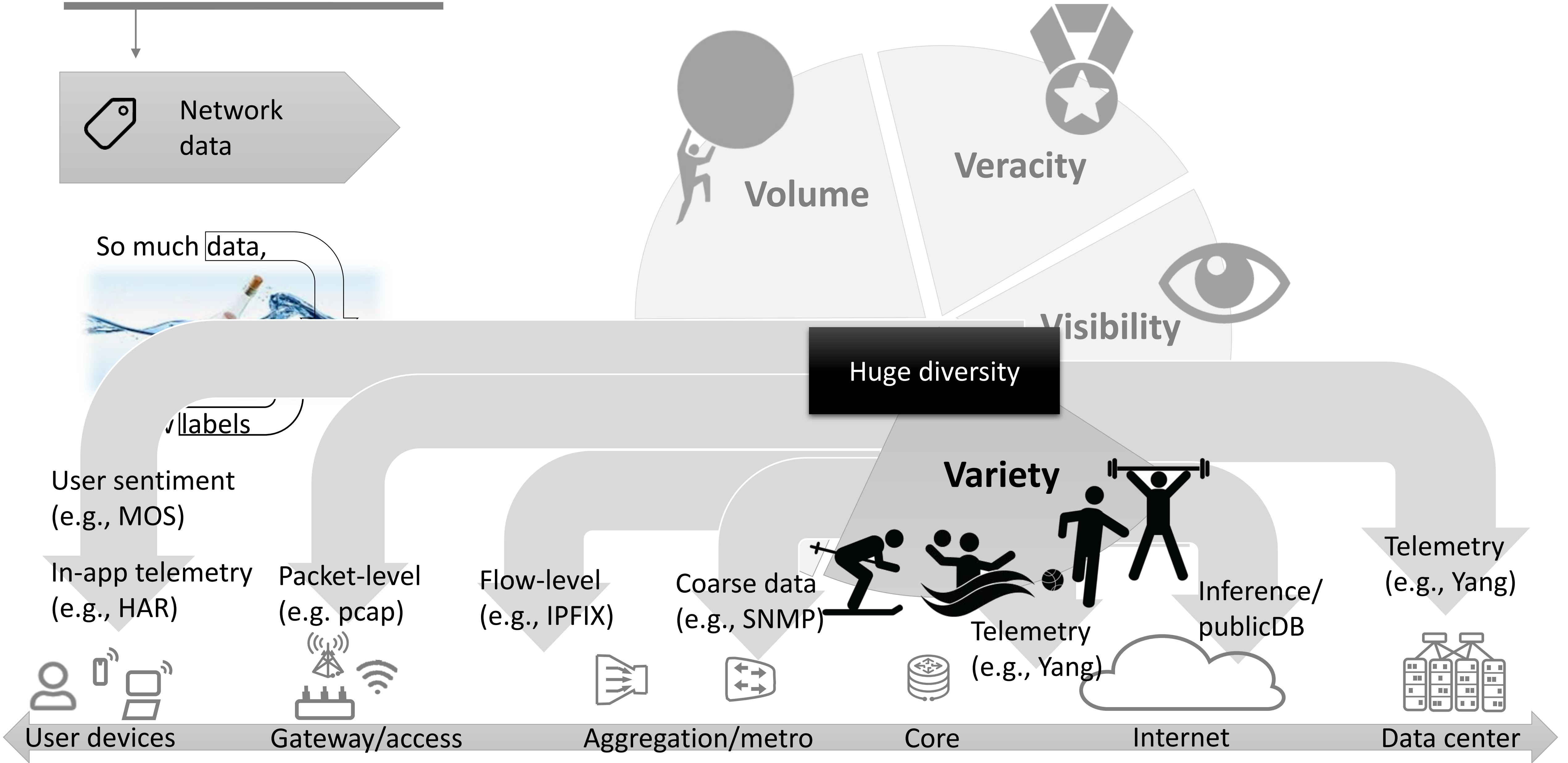
So much data,

so few labels





Networking data for ML / AI



https://www6.slac.stanford.edu/research/scientific-computing
https://home.cern/resources/faqs/facts-and-figures-about-lhc

Networking data for ML / AI

Network data

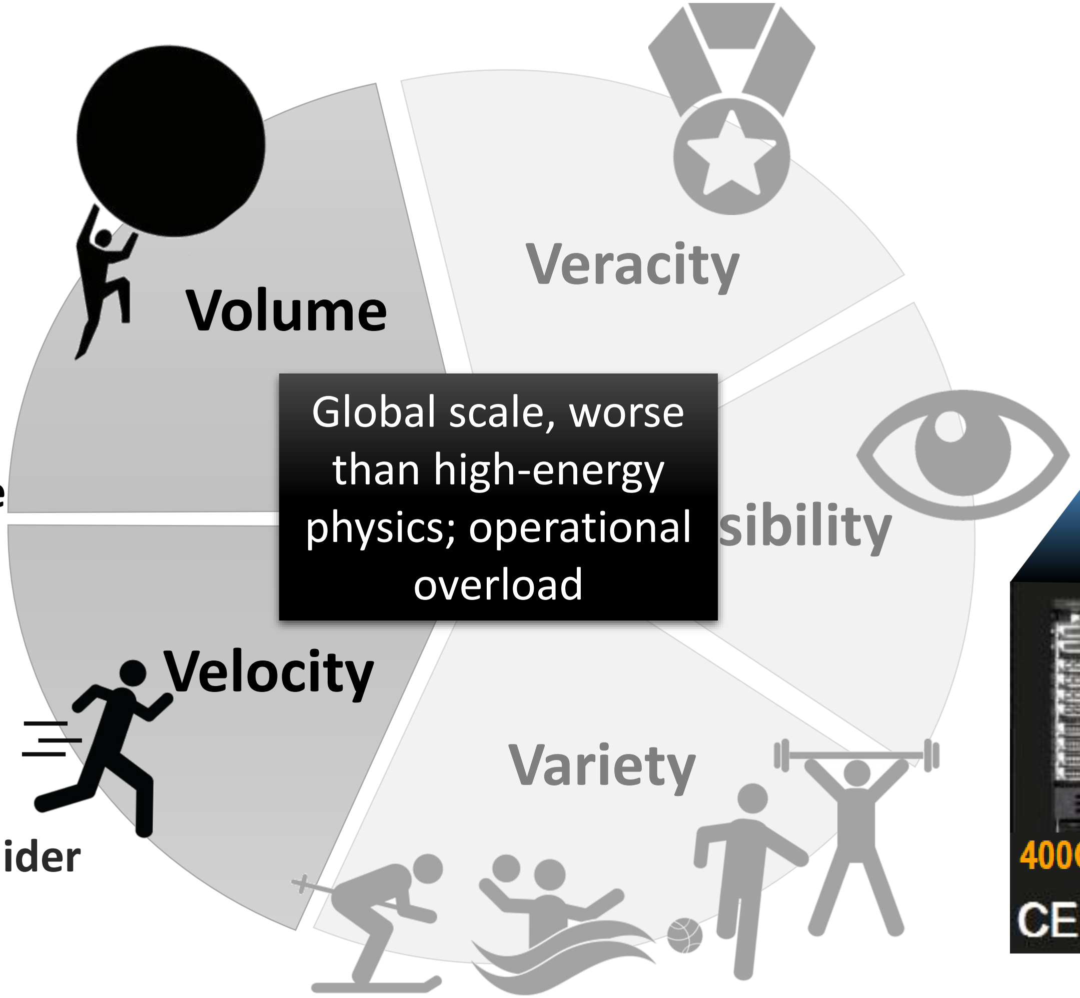
So much data,

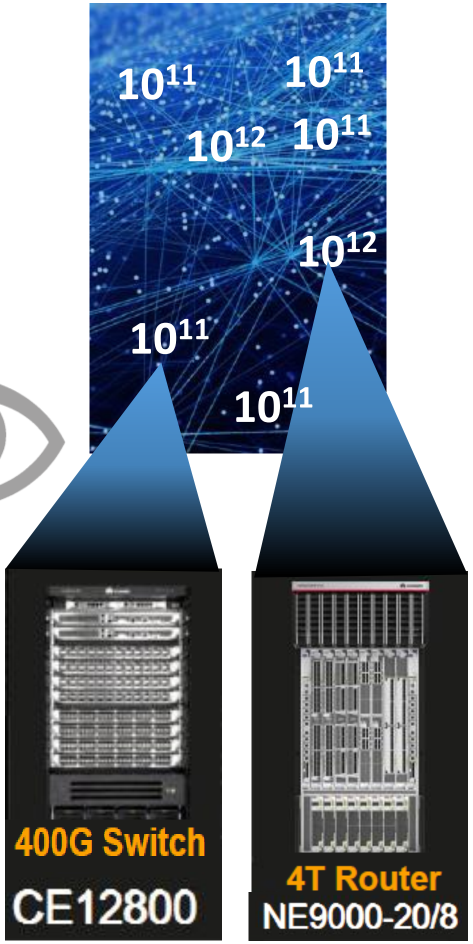
so few labels

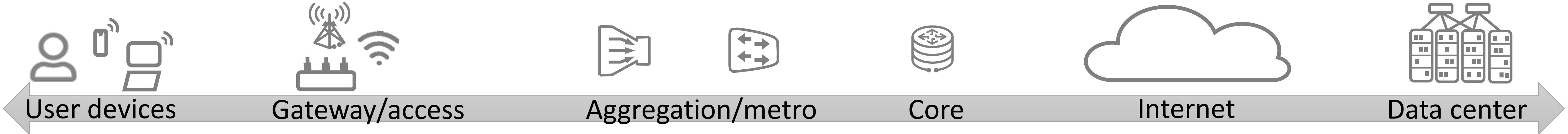
Linac Coherent Light Source
10^{10} bps, 4 PB

Large Synoptic Survey Telescope
10^{8} pixel/sec

Large Hadron Collider (LHC) 10^{12} bps





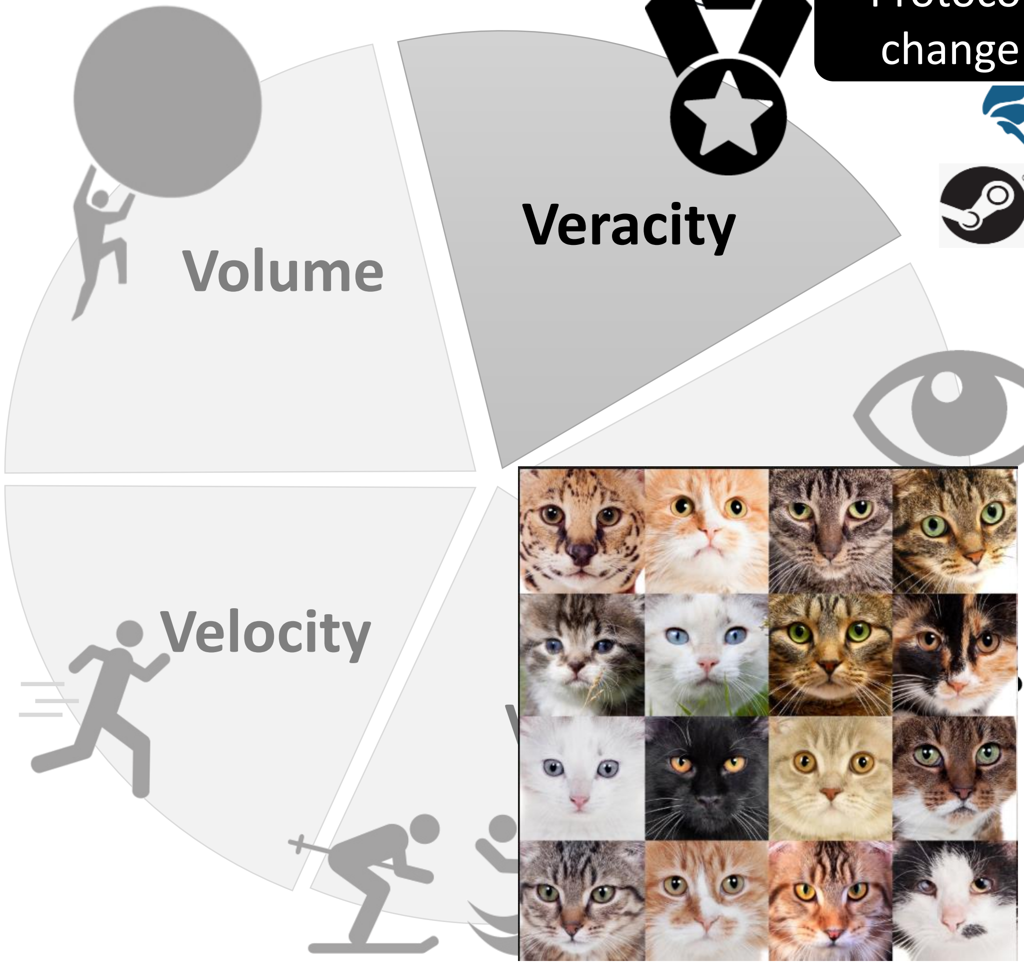


Networking data for ML / AI

Network data

So much data,

so few labels

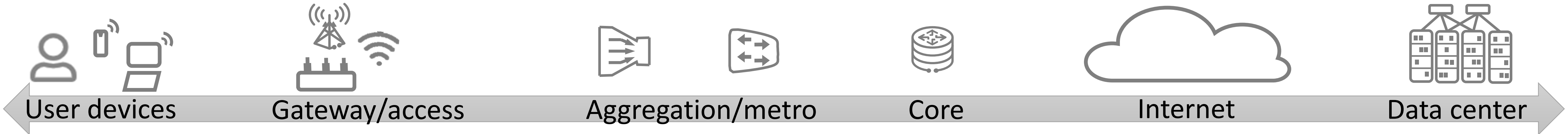


Protocol continuosly evolve, change and die. So do *labels*

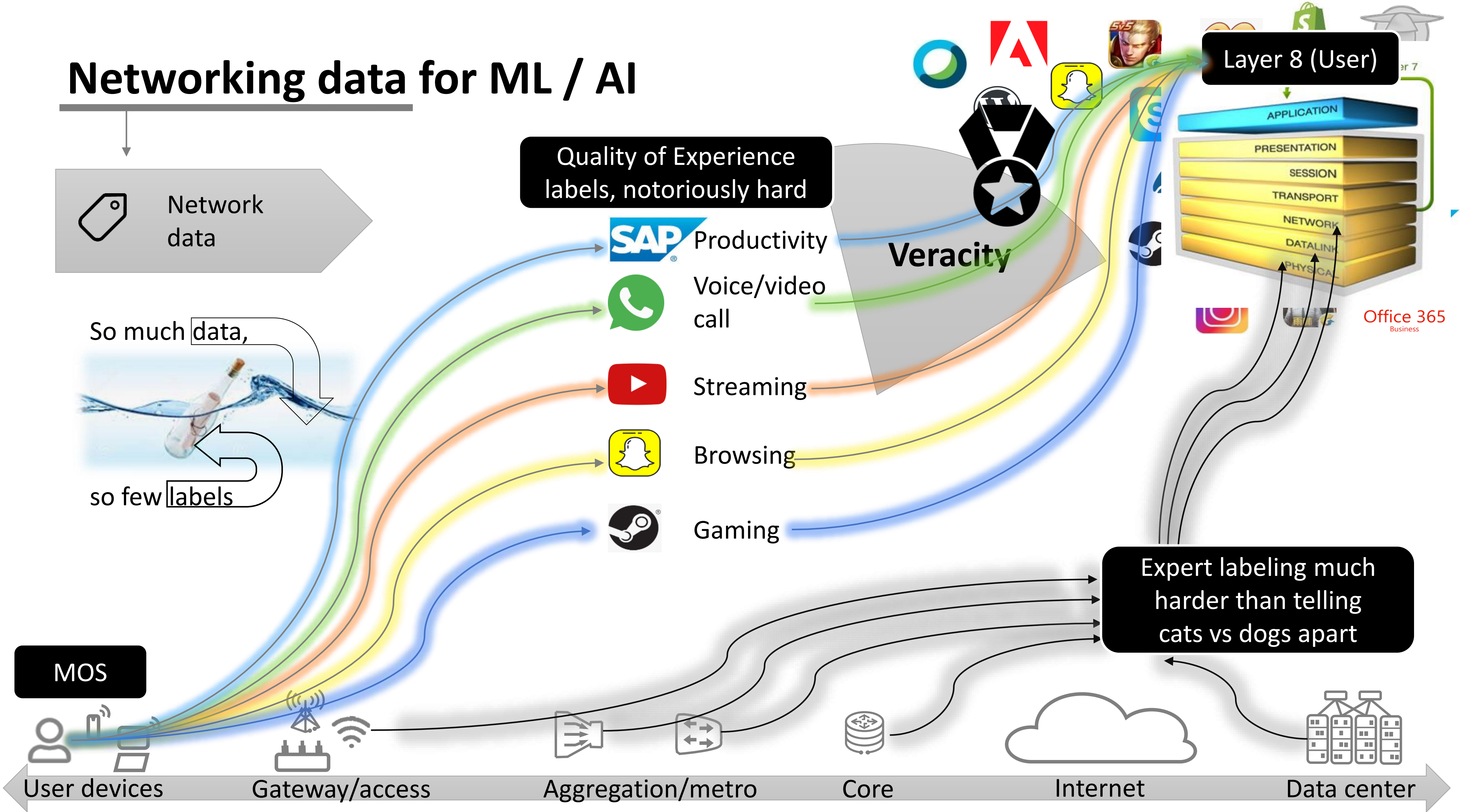
Cats are cats since 10^6 years

IMAGENET

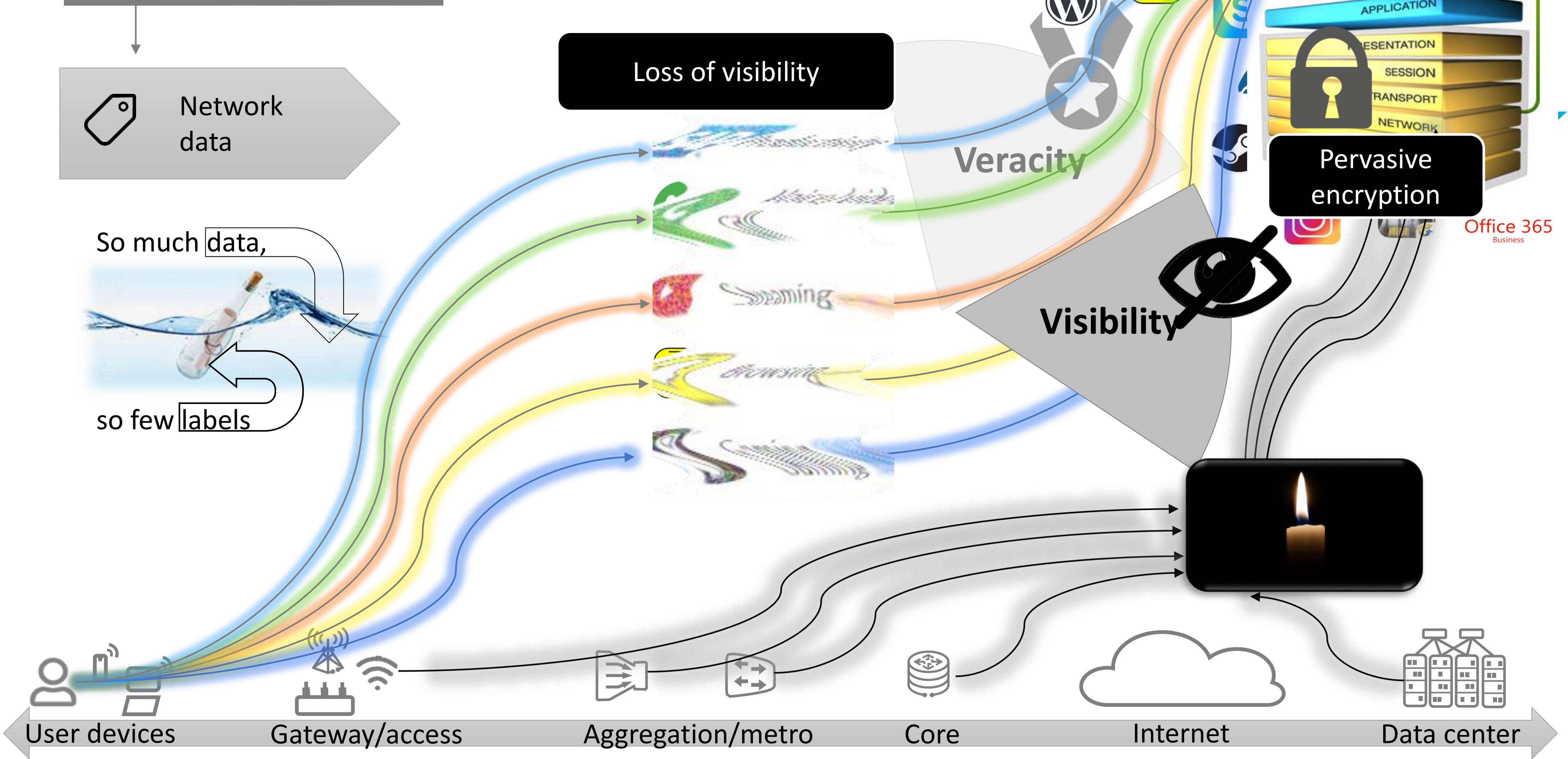
$1.5 \cdot 10^7$ labeled images



Networking data for ML / AI



Networking data for ML / AI



Networking data : added ML / AI value

It's optimal! (increase efficiency, same budget)
It's automated! (decrease human effort, save money)

So much data, Application packets 1 2 3 4 5 6 ...

New traffic flows

so few labels

Inputs

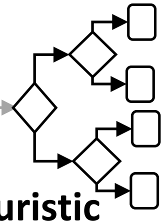
"Ground truth"

Labeled instances of applications of interest ... used for training

Example: Automated Application Recognition

Algorithm / system

Expert models

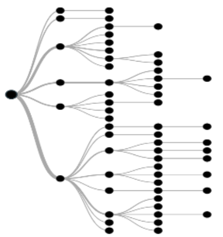


Reverse engineering & heuristic

Machine learning

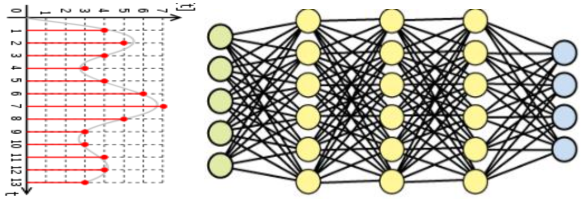
Mean packet size, flow rate, timing

Feature extraction



Classification

Deep Neural Networks



Feature extraction + Classification

Output

Prediction for each flow

- **Expert model**: manual effort, difficult to maintain
- **Machine learning**: algorithms to automatically learn optimal separation boundaries from *engineered* data
- **Deep Neural Nets**: algorithms to automatically learn non-linear functions from *raw data*

User devices — Gateway/access — Aggregation/metro — Core — Internet — Data center

Agenda

Hardware advances

Network data

Understand the network

Control the network

- ❑ History
- ❑ Trends
- ❑ AI chips

So much data,

so few labels

- ❑ Explicability
- ❑ Evolution
- ❑ Security

- ❑ Closing the loop
- ❑ Humans & the loop
- ❑ System aspects

Aim of this talk

Ingredients & recipes for good AI/ML use in networks

+ Flash few examples out of our activities

ML-powered networks

Care about interpretability, not just performance as a black-box

Favor understandable models (eg trees) when good enough, use soft-state output (eg confidence), maintain ability to switch from scientific data to the original "domain expert", etc.

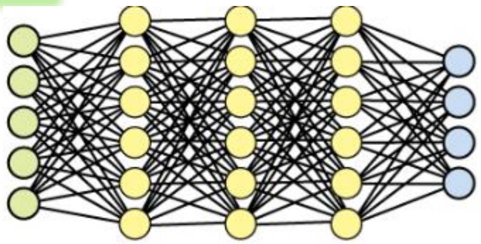
Understand the network

Some jobs will be lost, but humans operators will remain even with self-driving networks

- ❑ Explicability
- ❑ Evolution
- ❑ Security

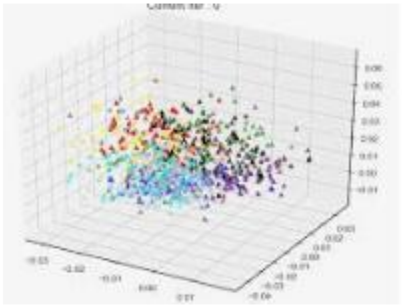
Several techniques inherently *as efficient as obscure*

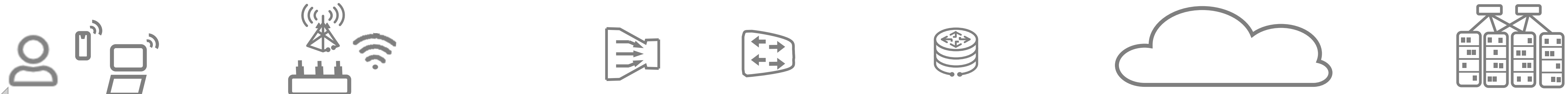
- ❑ Convolutional Neural Networks
  - ➔ weights of densely connected neurons?
- ❑ Support Vector Machines
  - ➔ representative examples of each class?



Often difficult to explain results to a *domain expert*

- ❑ Dimensionality reduction (PCA / tSNE)
  - ➔ very compact, but how to interpret?
- ❑ Outlier detection
  - ➔ along which of the many dimension?

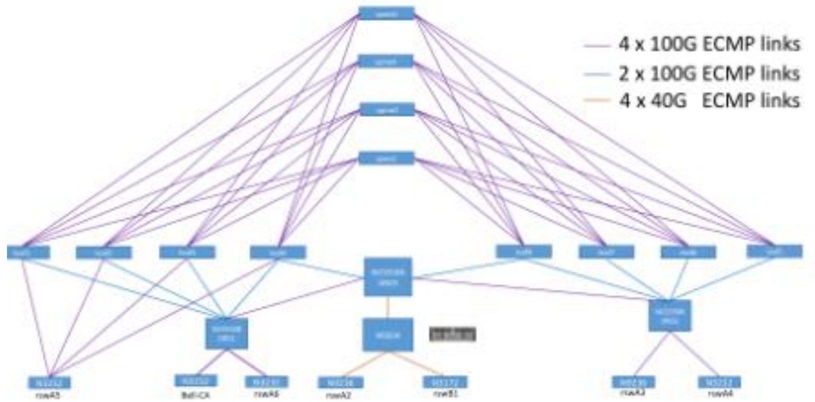




User devices | Gateway/access | Aggregation/metro | Core | Internet | Data center

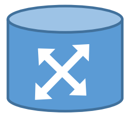
Example #1 Human-readable anomaly detection

WAN Routers
- Routers expose nearly 70,000 YANG features
- Scarcity of labeled data
  - Anomalies are very rare
  - Root cause analysis complex and time-consuming



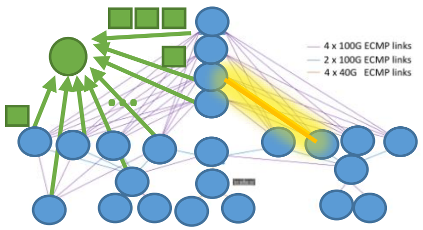
DCN routers & switches
- All BGP DCN (RFC 7938)
- 30 nodes, 1 collector
- YANG telemetry
  - 700+ interfaces,
  - data and control planes features

Local (node-level)



Goal: Combined anomaly detection and root cause analysis; automatically identify the KPIs (=features) selected by experts

Global (network-level)



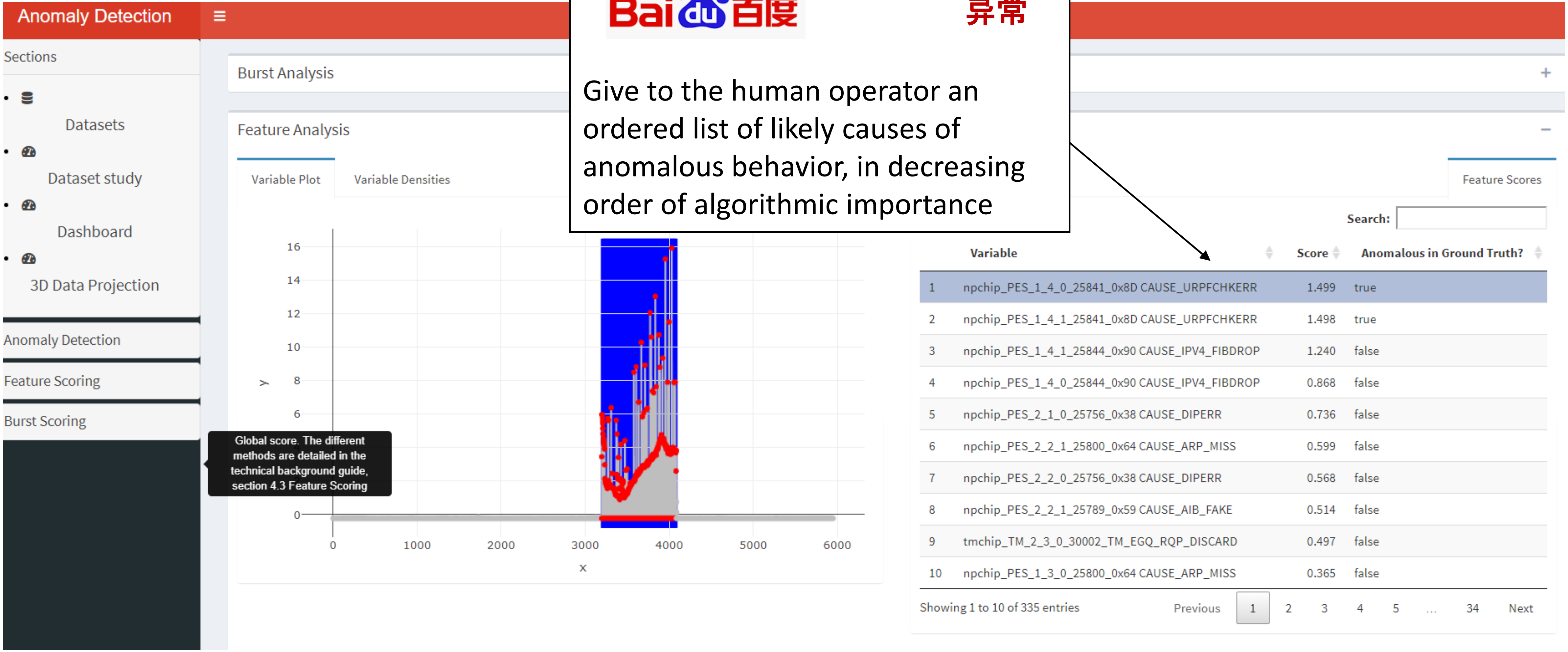
Goal: Distribute intelligence to reduce streamed data volume for anomaly detection

Example #1 Human-readable anomaly detection

Like Baidu for network anomalies

异常

Give to the human operator an ordered list of likely causes of anomalous behavior, in decreasing order of algorithmic importance

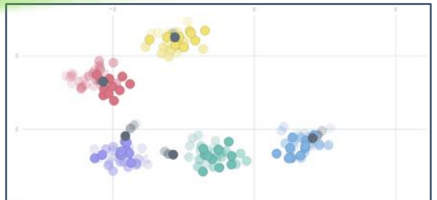


ML-powered networks

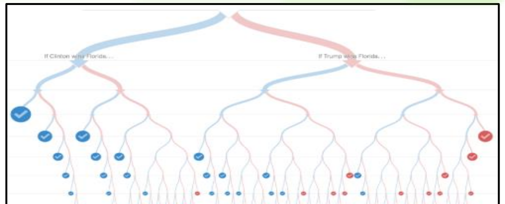
Understand the network

- ❑ Explicability
- ❑ **Evolution**
- ❑ Security

Online/streaming ML algorithms



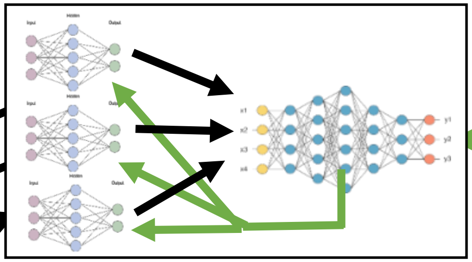
Unsupervised



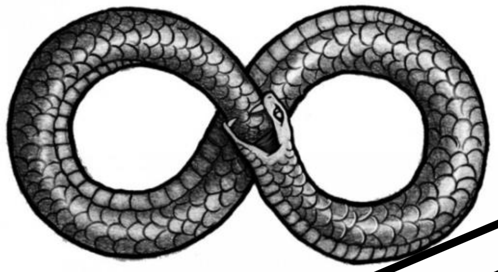
Supervised

- ❑ *Network evolves, so should your models*
- ▪ Clustering (e.g, Dgrid, DenStream, CluStream)
- ▪ Trees (e.g., Hoeffding tree, Adaptive Random Forest)

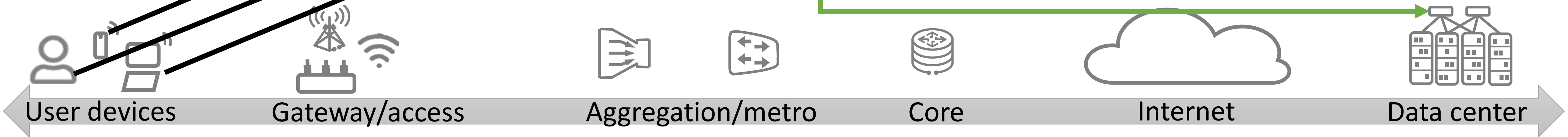
Model fusion



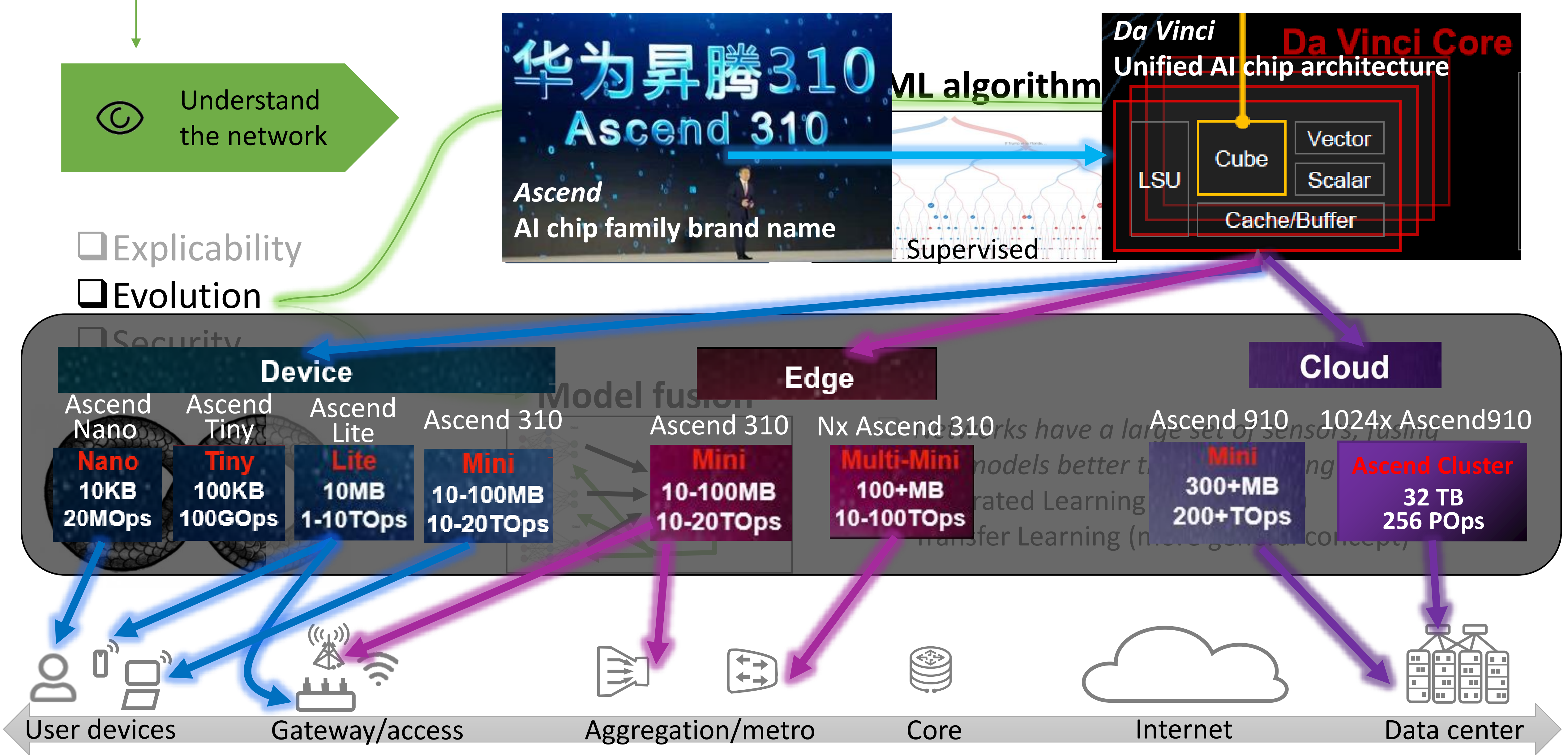
Federated learning



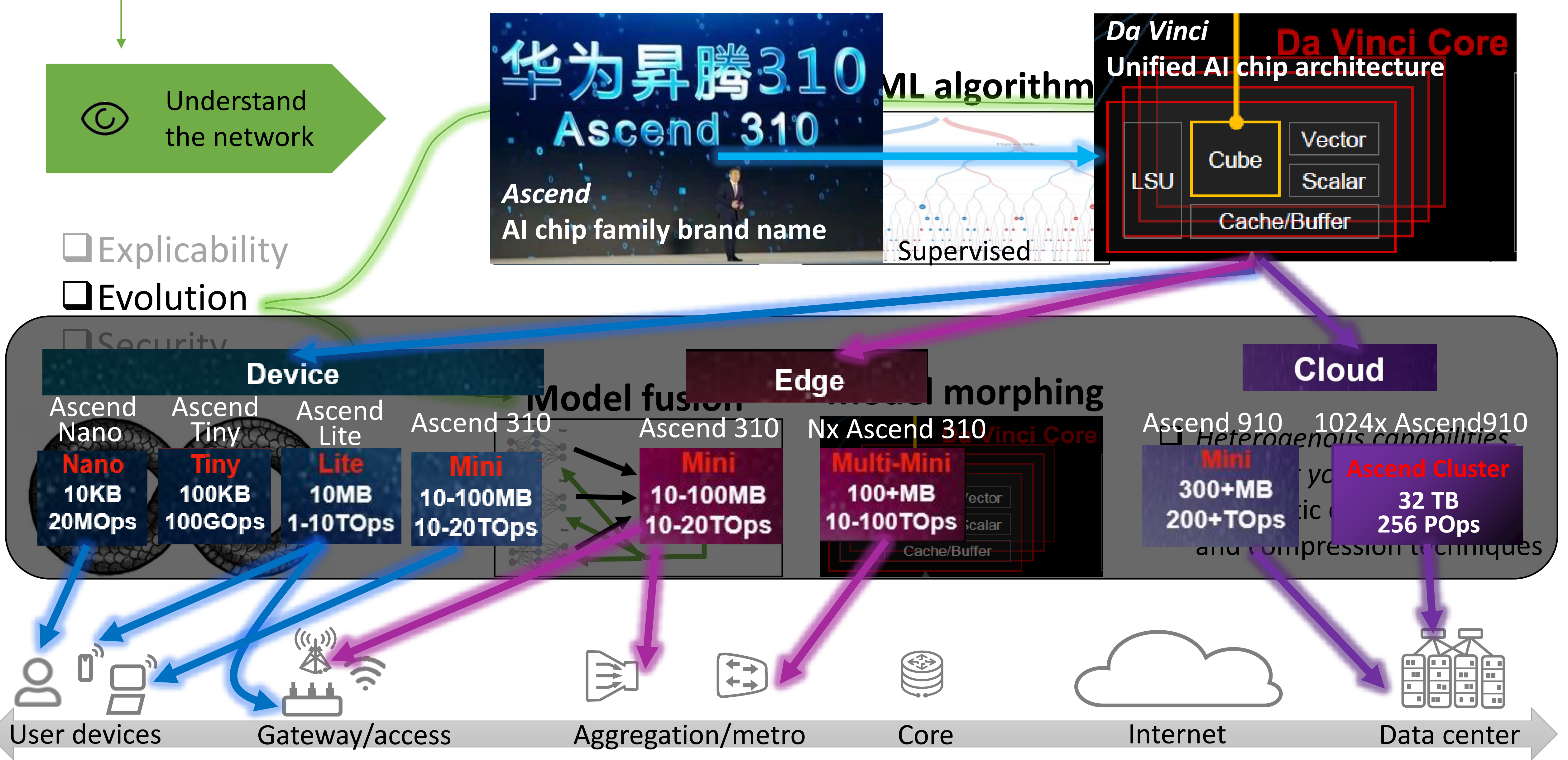
- ❑ *Networks have a large set of sensors, fusing this models better than exchanging data*
- ▪ Federated Learning (at the edge)
- ▪ Transfer Learning (more general concept)



ML-powered networks



ML-powered networks



ML-powered networks

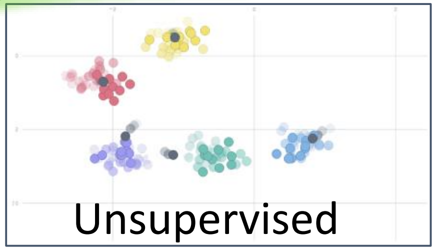
In ML, the journey matters more than the destination

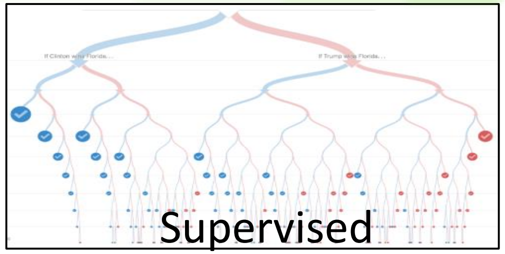
Use (supervised/unsupervised) stream learning techniques,
Enfuse models automatically (eg, federated/transfer learning,)
Transform models automatically (eg, specialize/quantize models)
AIOps (automate model catalog management and deployment)

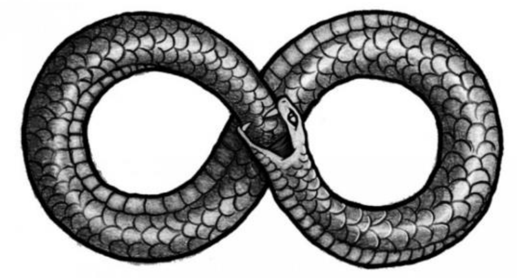
Understand the network

- Explicability
- Evolution
- Security

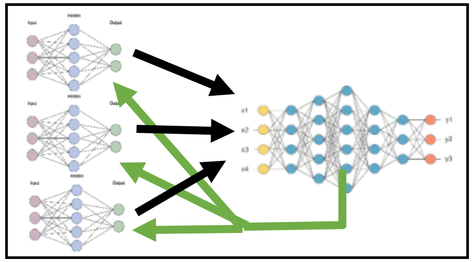
Online/streaming ML algorithms





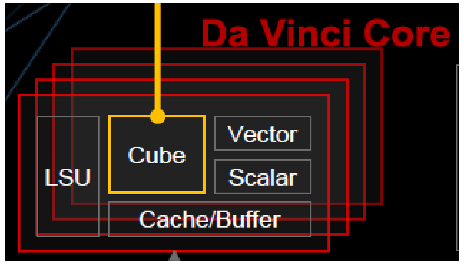


Model fusion

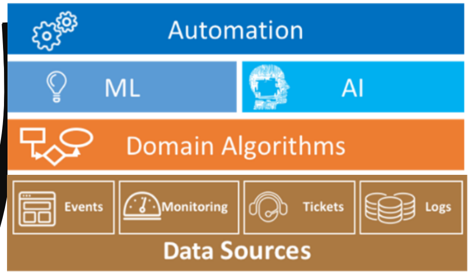


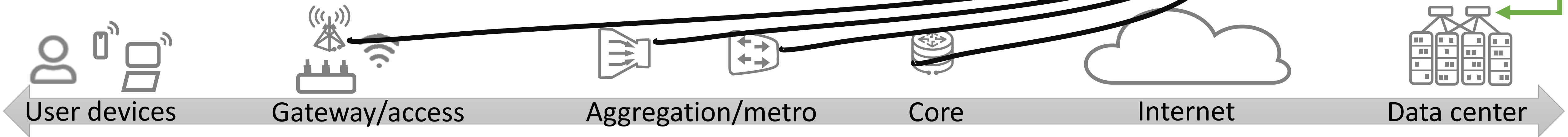
Federated learning

+Model morphing

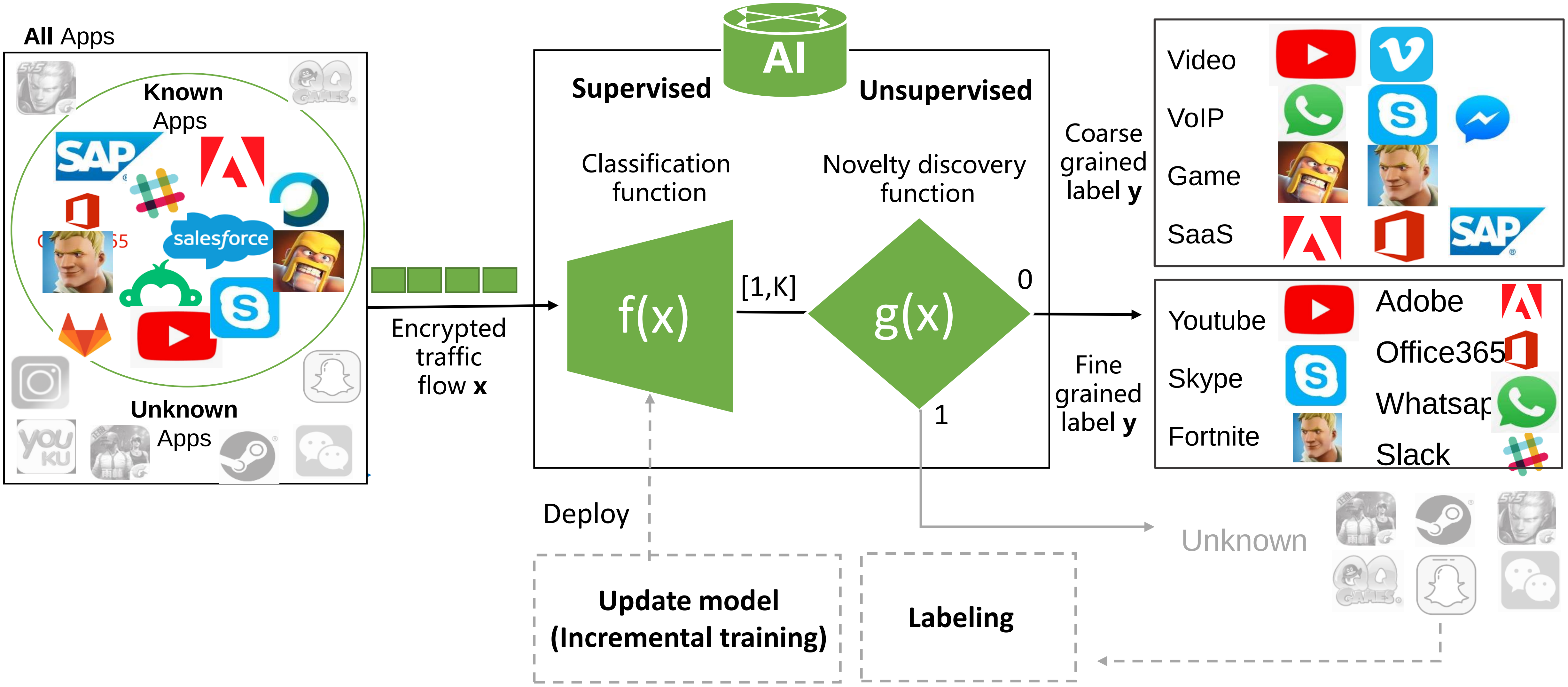


+ Embrace AIOps

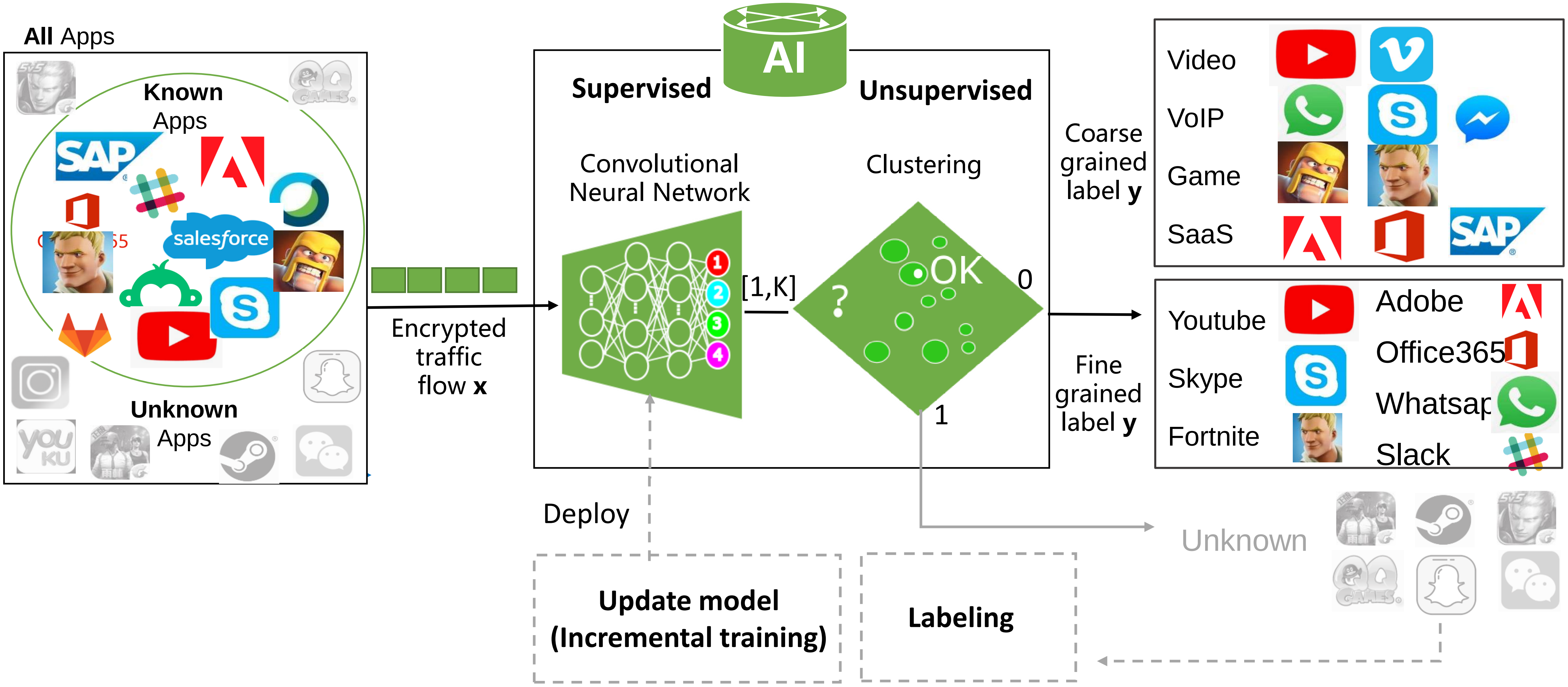




Example #2 Encrypted & unknown traffic classification

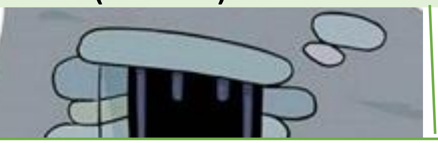


Example #2 Encrypted & unknown traffic classification



ML-powered networks

Just as network protocols, ML can (& will) be hacked



Understand the network

- ❑ Explicability
- ❑ Evolution
- ❑ Security

ML Evasion

- ❑ Can happen locally, when a model is deployed
- ❑ *E.g.*, Adversary circumvents/alters traffic classification results by purposely altering its own features

Adversarial ML

- ❑ Can happen for streaming techniques, during the learning phase
- ❑ Adversary alters the ML training process by purposedly mislabeling data, affects all systems

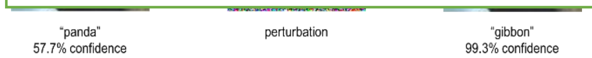
Leak of sensitive information

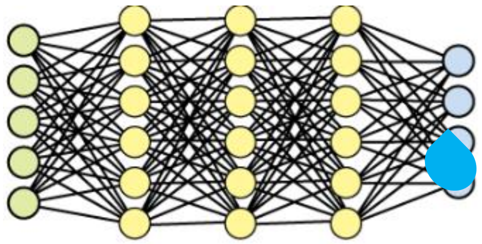
- ❑ E.g, adversary extracts information from shared/accessible ML models

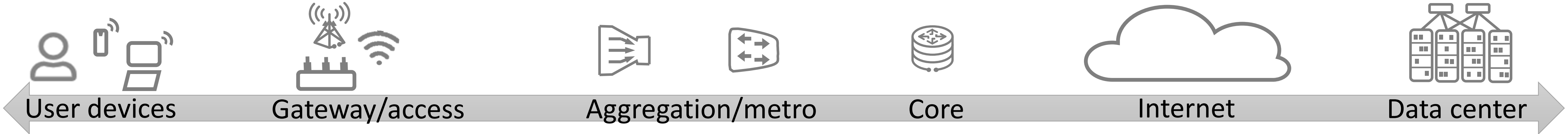
Foreworded is forearmed !

Robust training, differential privacy, etc.

No silver bullet exist though: security is the art of making the right tradeoffs







Agenda

- Hardware advances
- Network data
- Understand the network
- **Control the network**

- ❑ History
- ❑ Trends
- ❑ AI chips

So much data,

so few labels

- ❑ Explicability
- ❑ Evolution
- ❑ Security

- ❑ Closing the loop
- ❑ Humans & the loop
- ❑ System aspects

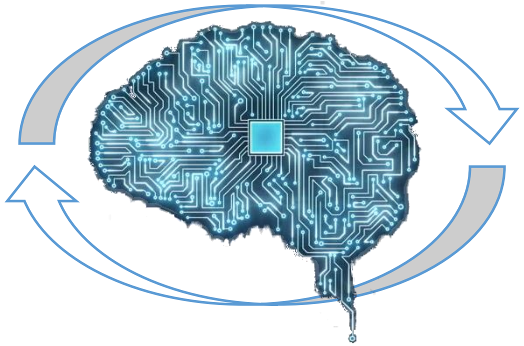
Aim of this talk → **Ingredients & recipes for good AI/ML use in networks** → **+ Flash few examples out of our activities**

AI-powered networks

When closing the loop,
mind the gap!

Control
the network

- ❑ Closing the loop
- ❑ Humans & the loop
- ❑ System aspects



Games (Go state space ~ 10^{100})

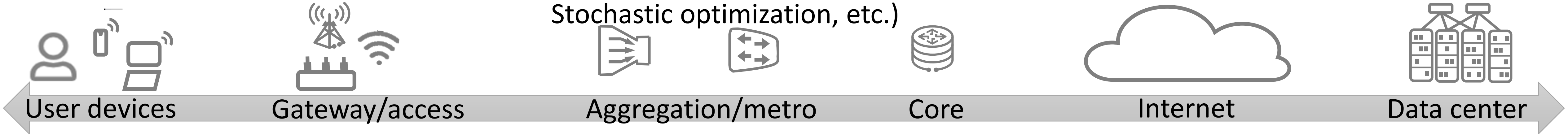
- ❑ AlphaGo (10,000s of human amateur and professional
  3 days training, 1920 CPUs, 280 GPUs, elo rating 3.16)
- ❑ AlphaGo Zero (simply plays against itself)
  4 TPUs, 40 days to beat AlphaGo Master, achieving elo
- ❑ Alpha Zero (just out, not peer reviewed)
- ❑ Portability? Add one row ▣ to the board !! Add a ● pl

Networks (state space $\mathbb{R}^N$, with N>>100)

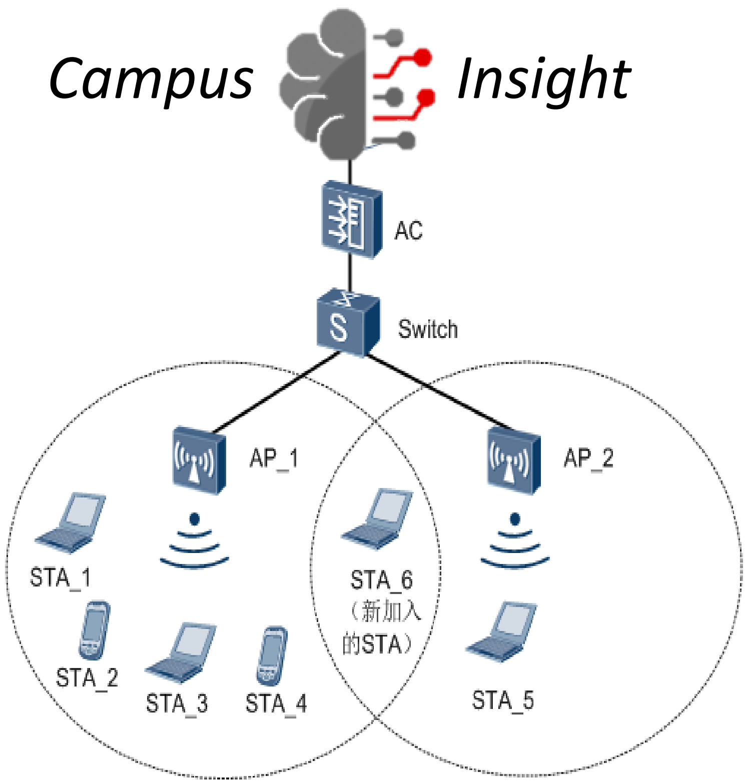
- ❑ Portability is essential: you cannot sell an AI product that will make performance *worse* for over a month !
- ❑ Results coupled with delay of telemetry, and delay to actuate actions in the controller
- ❑ Convergence speed matters ! for any techniques (Reinforcement learning, Deep reinforcement learning, Stochastic optimization, etc.)

Need to leverage simulation/emulation (eg digital twin) to speedup training & anticipate actions reward

Internal state of AI algorithms cannot be debugged by humans, telemetry of uttermost importance

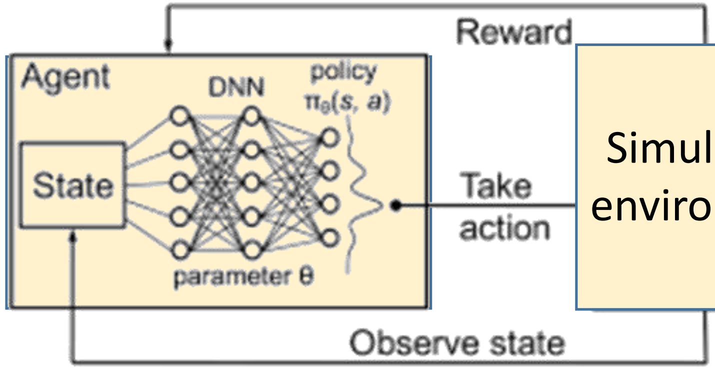


Example #3 WLAN traffic optimization



(Deep) reinforcement learning

Reward= f(T, Δ, QoE, I, RSSI, …)



Simulation environment

Emulation environment

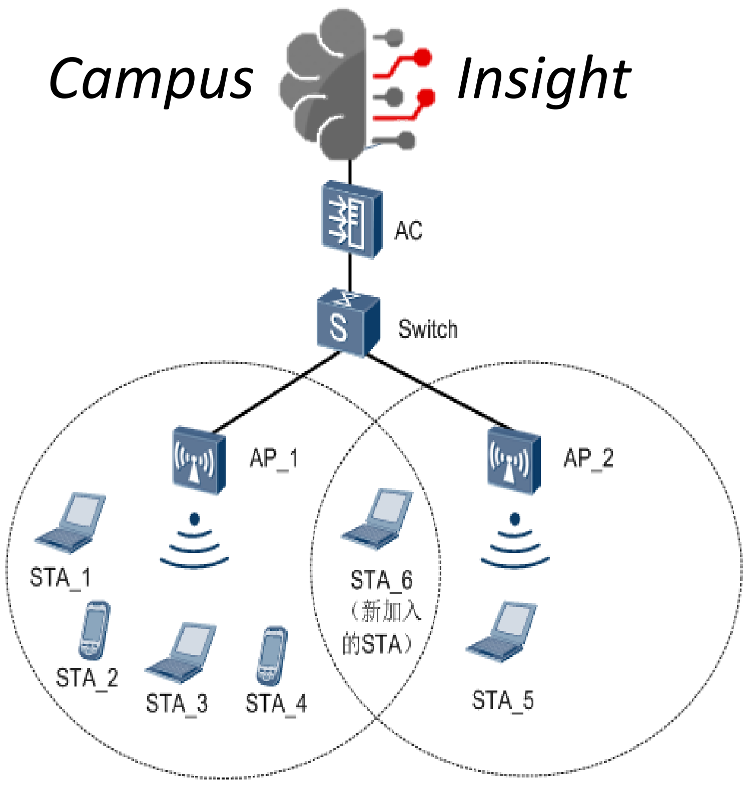
Real world

Speedup state exploration

Combine multiple environments

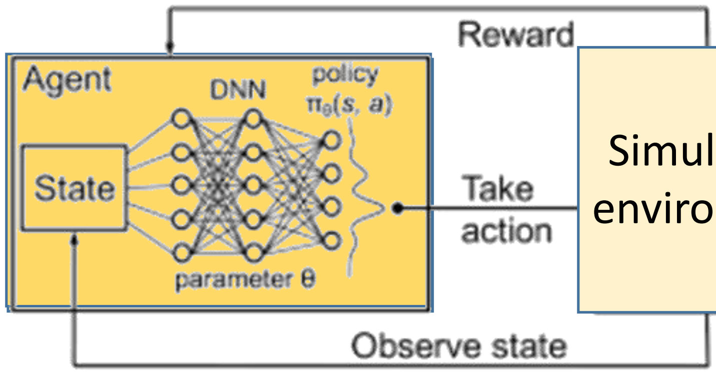
Simulation | Emulation | Real world

Example #3 WLAN traffic optimization



(Deep) reinforcement learning

Reward= f(T, ∆, QoE, I, RSSI, …)



Simulation environment

Emulation environment

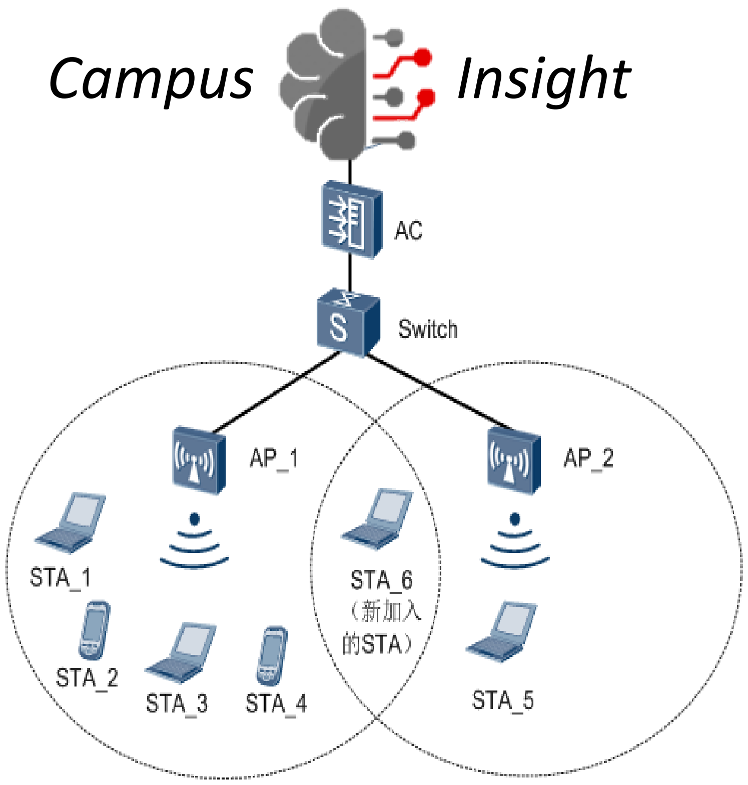
Real world

Speedup state exploration

Combine multiple environments

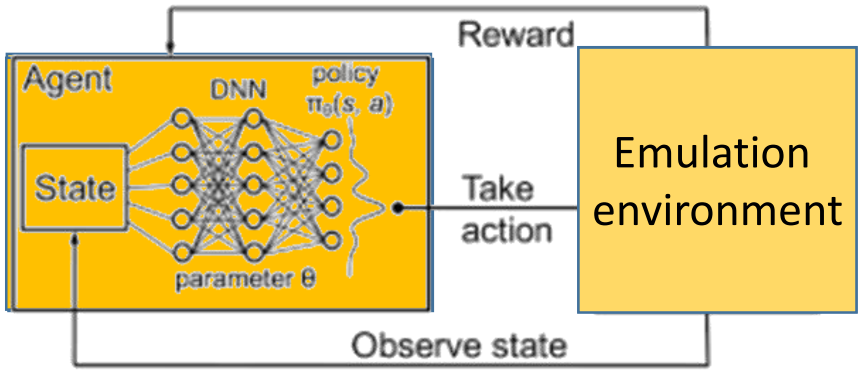
Simulation | Emulation | Real world

Example #3 WLAN traffic optimization



(Deep) reinforcement learning

Reward= f(T, Δ, QoE, I, RSSI, …)



Real
world

Speedup state exploration

Combine multiple environments

Simulation | Emulation | Real world

AI-powered networks

Keep humans in the (slow) loop, facilitate their interaction with AI

Control
the network

- ❑ Closing the loop
- ❑ **Humans & the loop**
- ❑ System aspects

QoE driven network management

In most cases, *users* in the end-to-end loop

- ❑ Must avoid humans in the *fast* loop (else it breaks the autonomic paradigm)
- ❑ Useful to keep humans in the *slow* loop (e.g. involve end-users to ensure AI controlled networks works better than before!)

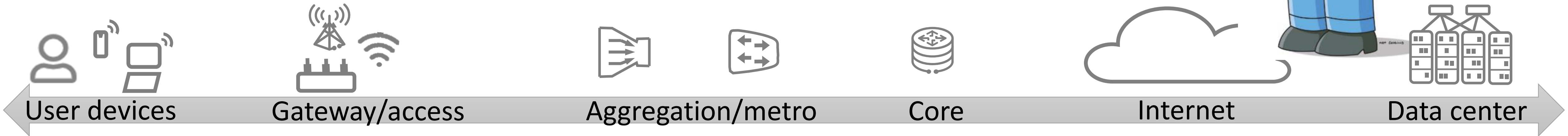
Human-resilient AI

In most cases, *human operators* will not have a clue (or anyway will not be experts) of AI technologies

- ❑ AI should be resilient in spite of poor/adversarial training, bad calibration, overfitting, unfairness, …
- ❑ Artificial intelligence must use techniques to be robust and survive in spite of human stupidity….

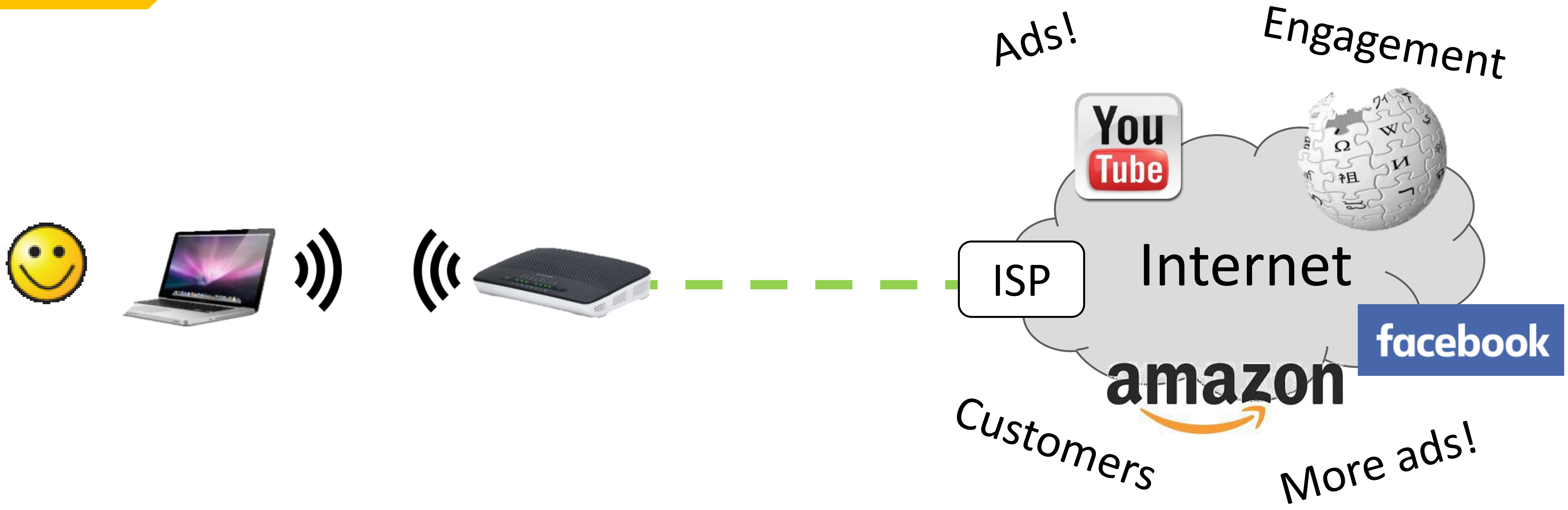
UI to empower AIOps & online models with streams of labels

Automated techniques to make interaction with non- AI expert Homer-proof (eg. Huawei's ModelArts)



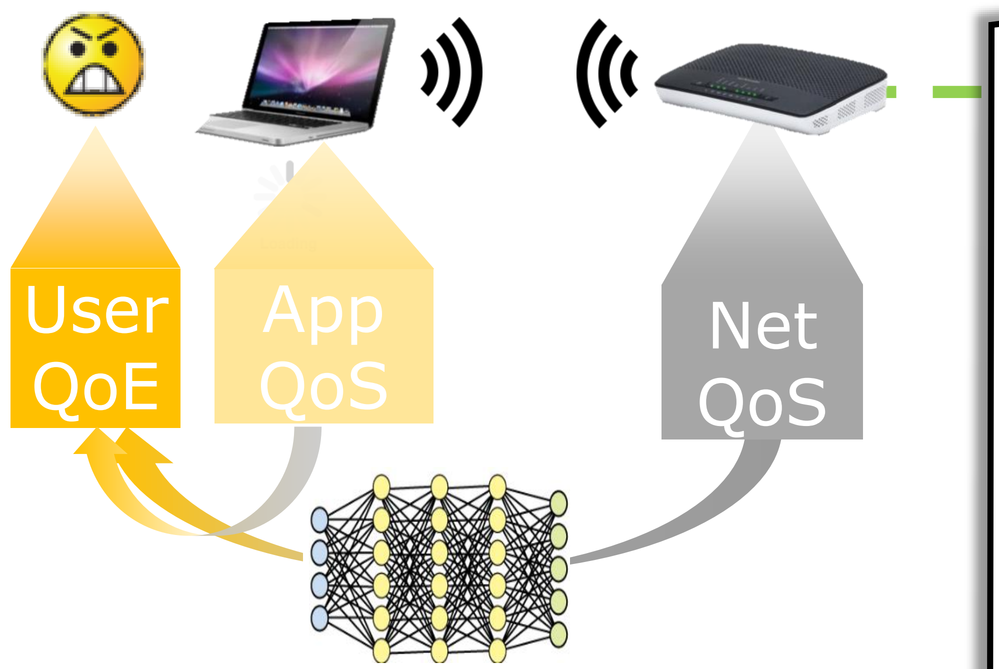
Example #4

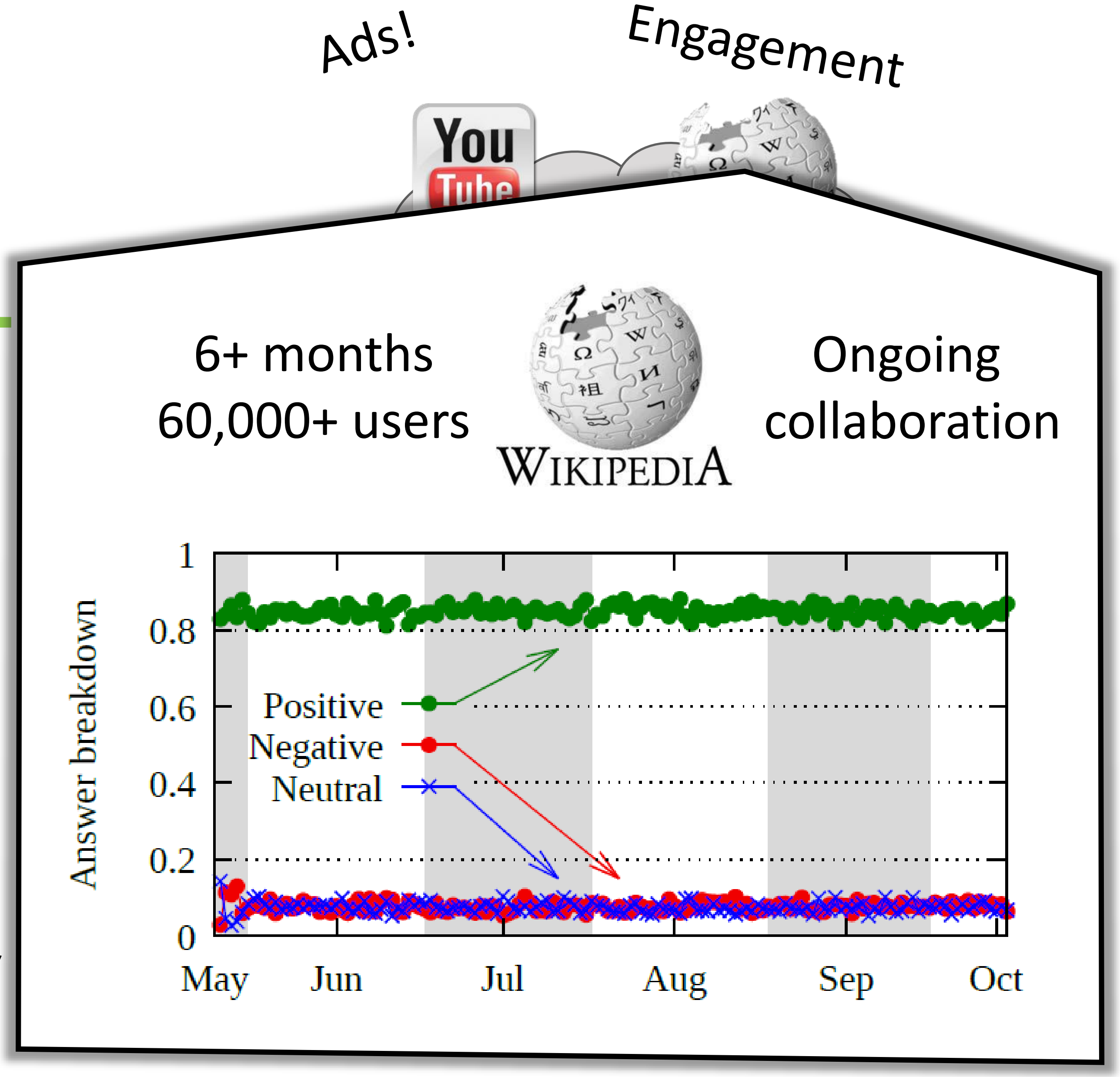
Web Quality of Experience



Offering Good user QoE is a common goal

Example #4 Web Quality of Experience





Detecting/preventing user

AI-powered networks

Statistical approach not a silver bullet. AI resource allocation !

Control the network

- ❑ Closing the loop
- ❑ Humans & the loop
- ❑ System aspects

Need for deterministic algorithms

- ❑ Machine learning is not a *silver bullet:*
  - ML accuracy 99.9% (dream model) 100,000 configuration lines = 100 errors
  - Ops, the problem just got a worse nightmare
- ❑ Autonomus configuration must use formal model for rigorous and deterministic guarantees

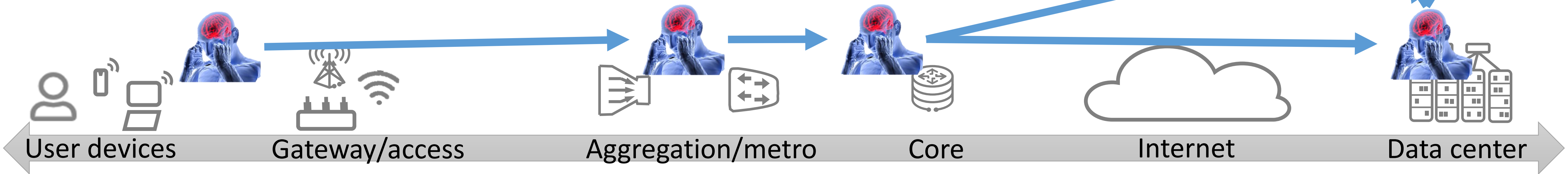
Deep knowledge of tools needed. ML/AI otherwise is just a buzzword

Commoditization of tensor processing unit adds several interesting dimensions to the classic resource allocation proble

AI-resource allocation

AI powered chips extend the NFV resource allocation problem to a new dimension: the chip memory/processing resources!

- ❑ New tradeoff: chip memory/operations vs bandwidth
- ❑ New problems: how to split the in-network processing?



Takeway messages

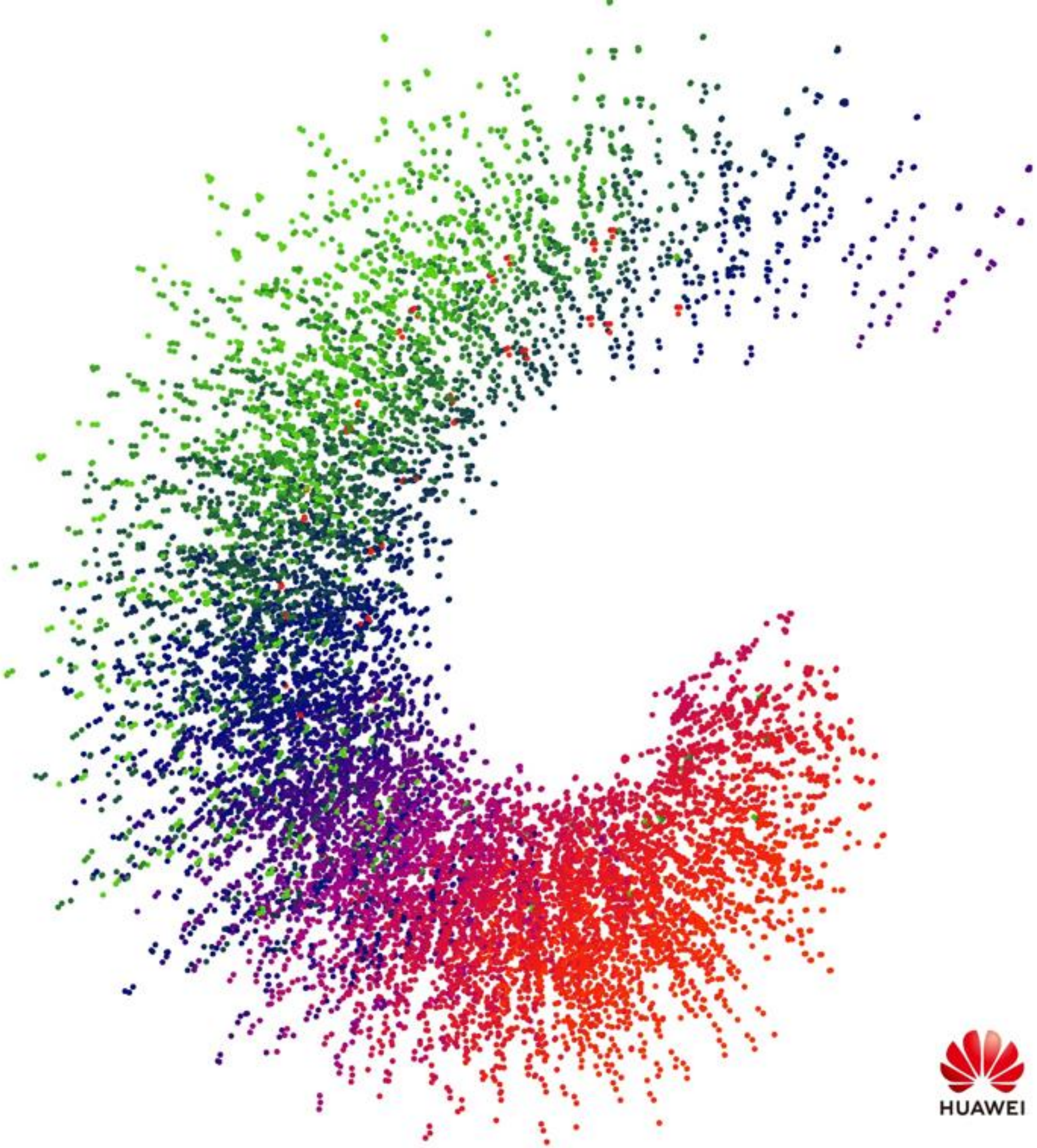
Hardware advances
Recent hardware advances true enablers of “edge intelligence”

Network data
Heterogeneous, asynchronous, evolving unlabeled massive data

Understand the network
- In ML, the journey matters more than the destination
- Care about interpretability, not just performance as a black-box
- Just as network protocols, ML can (& will) be hacked

Control the network
- Keep humans in the (slow) loop, facilitate their interaction with AI
- When closing the loop, mind the gap!
- Statistical approach not a silver bullet. AI resource allocation !

Thanks



Dario Rossi,
Chief Expert Network AI
dario.rossi@huawei.com
https://nonsns.github.io

HUAWEI