Co-sponsored by:

Our team is hosted by:

Principles of Distributed Systems, Cloud Computing, and Big Data Processing

@Alosup

Prof. dr. ir. Alexandru Iosup

Massivizing Computer Systems

Note: Much of the graphical material courtesy Google Images. Many thanks to all authors!

ÉCOLE SUPÉRIEURE D'ÉLECTRICITÉ

UNIVERSITATEA POLITEHNICA BUCURESTI

PARALELLIZATION AND GRID INTEGRATION OF A MOBILE ROBOTICS APPLICATION

Karel Capek,
Rossumovi Univerzální Roboti

1921

Author:

Alexandru IOSUP

2003

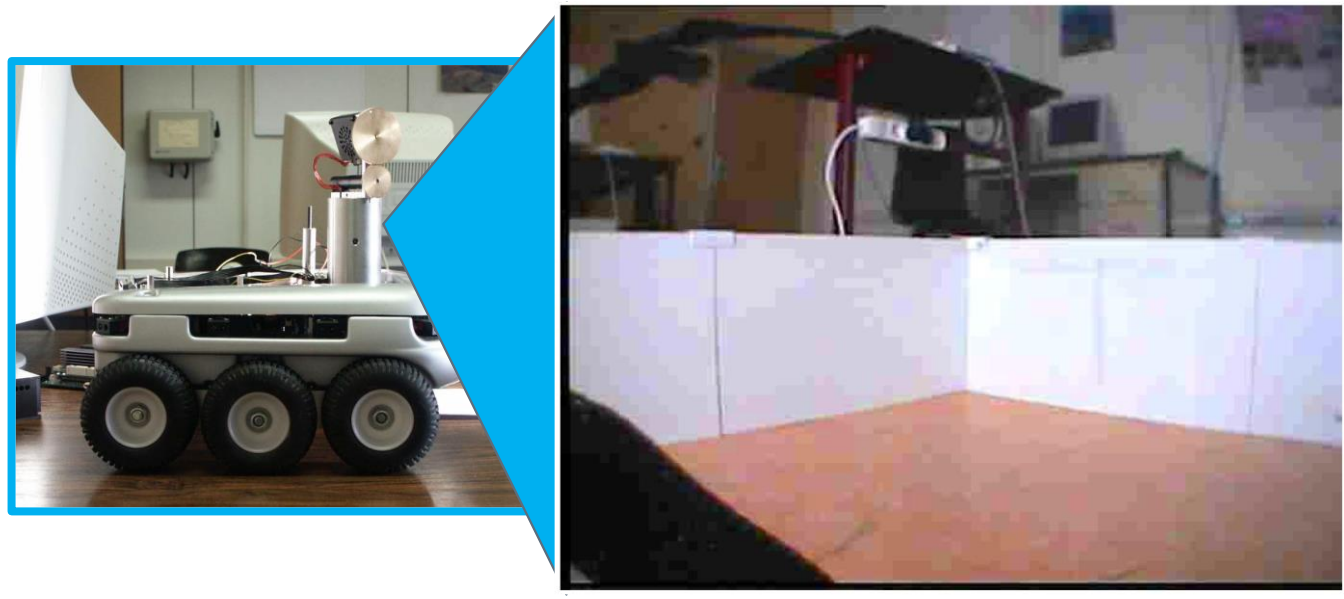
Disney,
Big Hero 6

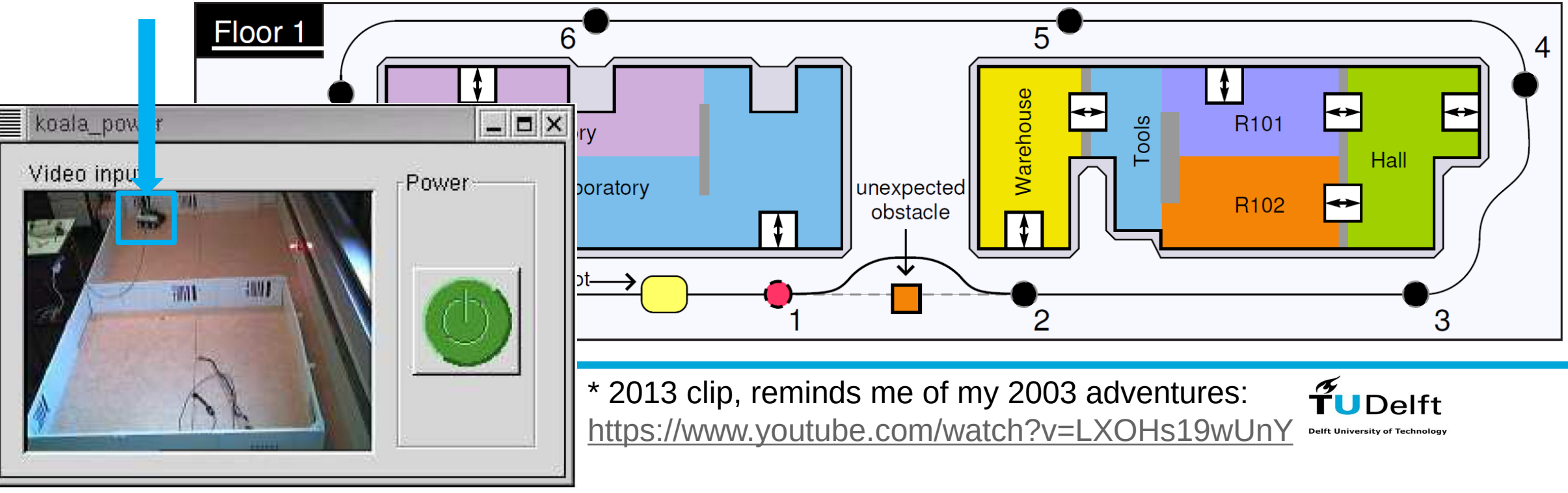
2016

VU VRIJE UNIVERSITEIT AMSTERDAM

TU Delft
Delft University of Technology

A Robot Rolls* Into a Bar…





* 2013 clip, reminds me of my 2003 adventures:
https://www.youtube.com/watch?v=LXOHs19wUnY

TU Delft
Delft University of Technology

What Is a Robot?

1. “By the late 1920s, just about any machine that replaced a human job with automation or remote control was referred to as a robot.” [1]
   → An *industrial robot* is any device that replaces human labor. (JP)
   - A computer running a program, even an abacus, is a robot

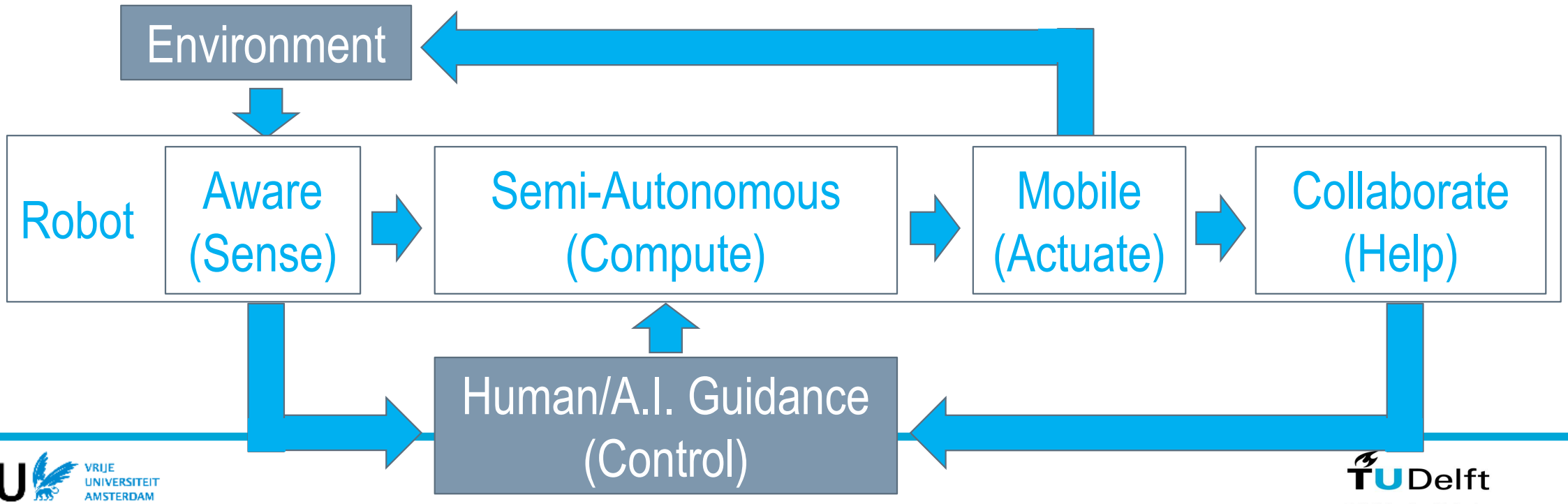
2. “An *industrial robot* is a multifunctional, reprogrammable manipulator that can move materials, pieces, tools and special devices, following variable programmed trajectories, to realize different tasks.”
   Robotics Industry Association, USA
   - Must have a body, traverse the real world
   - A conveyor belt is a robot

[1] https://www.theatlantic.com/technology/archive/2016/03/what-is-a-human/473166/

What Is a Robot? Sense-Compute-Actuate-Help [1]

(My definition) A *robot* is an object manipulator that, in a robot-swarm:

- Is self-aware, using sensorial and historical data to reason about its state and goals
- Is semi-autonomous, taking in minimal hints to guide itself and hint its swarm
- Is mobile relative to its environment, possibly also relative to its swarm-mates



VU VRIJE UNIVERSITEIT AMSTERDAM

TU Delft Delft University of Technology

[1] Extends http://www.ifr.org/service-robots/ (https://hbr.org/2015/04/what-is-a-robot-anyway) and includes SCA model of Andy Rubin, https://www.wired.com/2016/02/editors-letter-march-2016/

Today's State of the Art


self-driving car

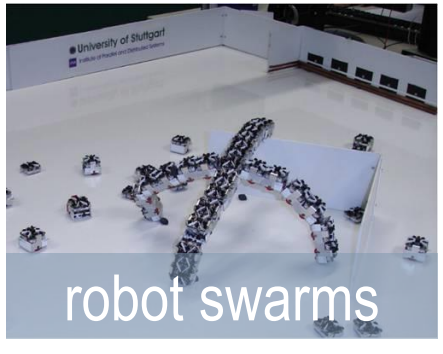

exoskeleton


auto-parking

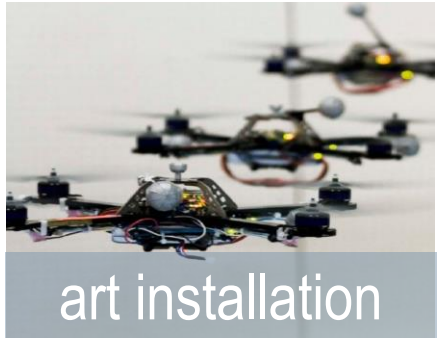

robo pets

serve and protect


disaster area navi



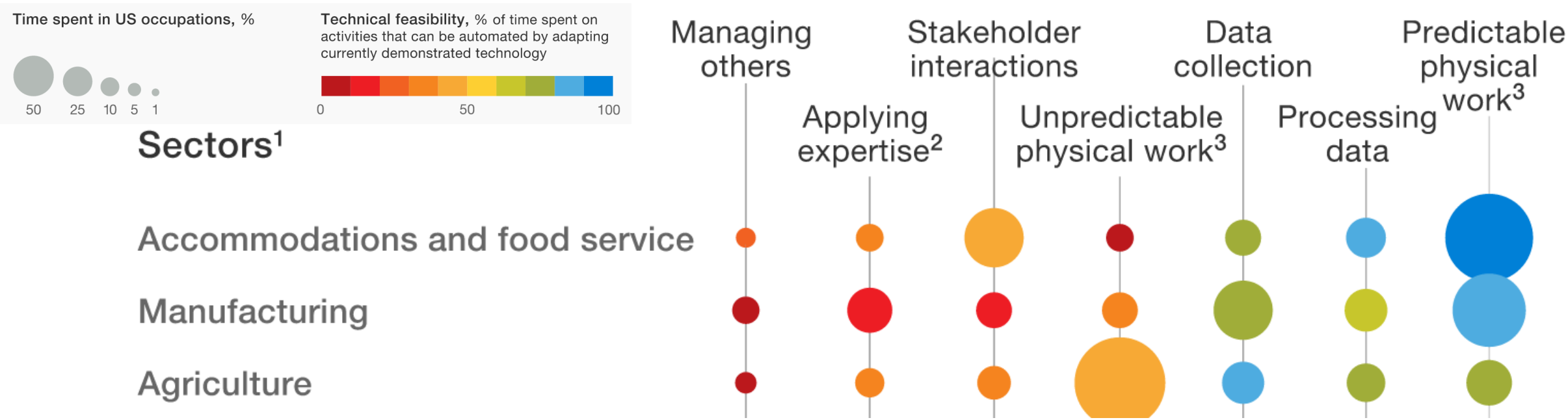
robot swarms


art installation


be my doctor

Much to Do in Robotics

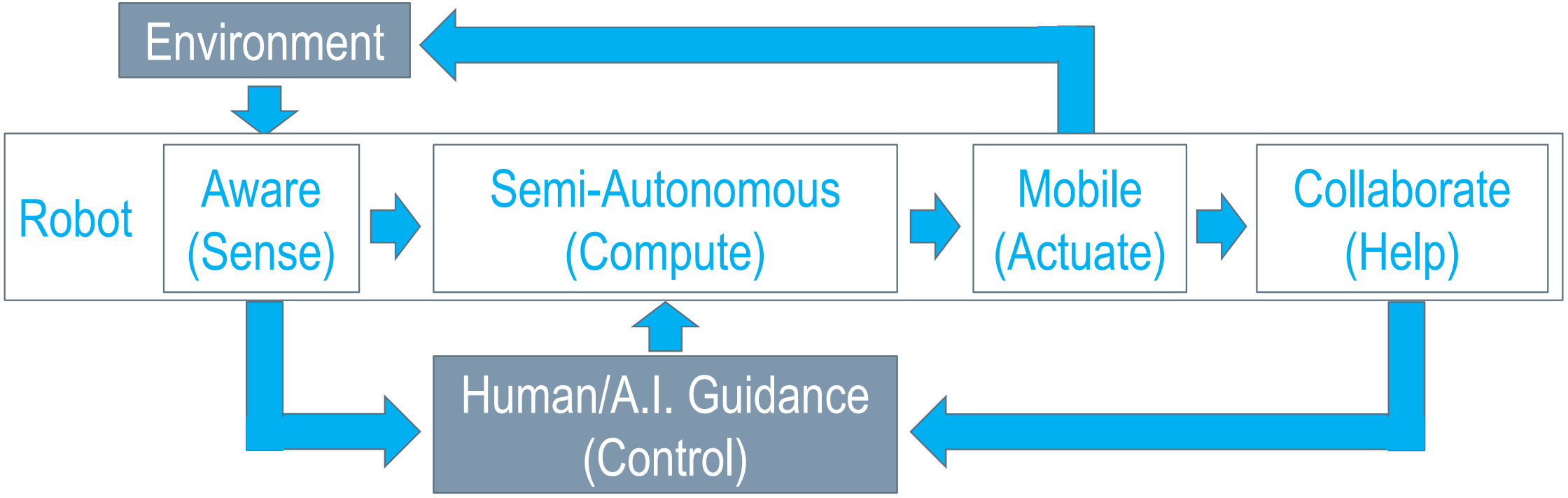
- Building modern robotics-based systems (this lecture)
- Ethics, from Asimov's Three Laws to The Future of Jobs
- …



World Economic Forum https://www.weforum.org/reports/the-future-of-jobs and McKinsey http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/where-machines-could-replace-humans-and-where-they-cant-yet

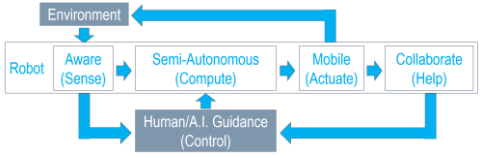
A Reference Architecture for Robotics Systems

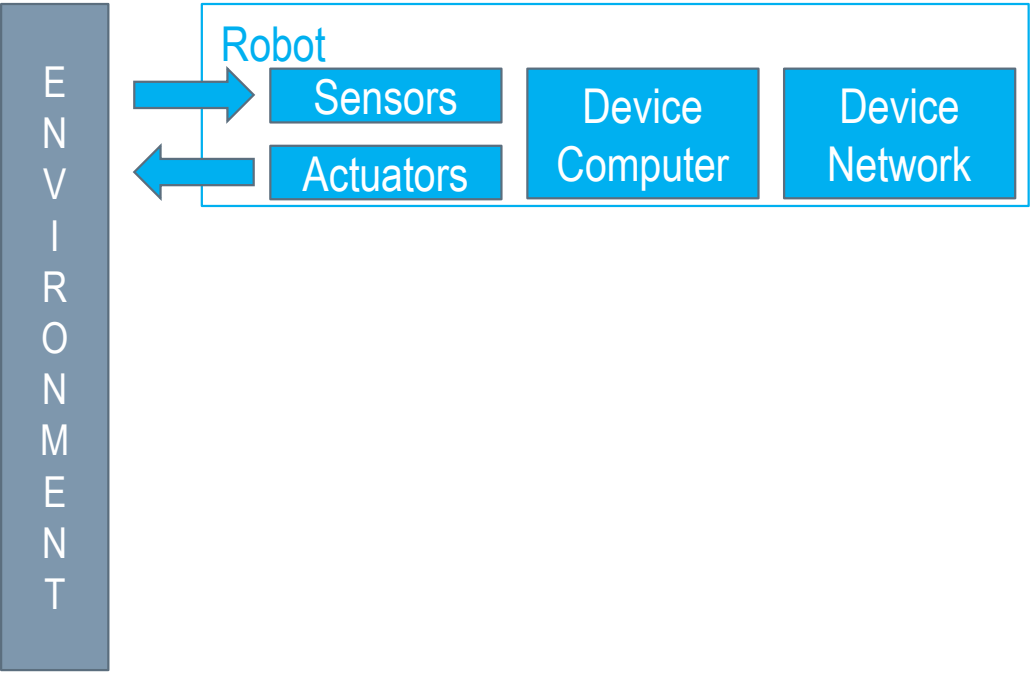
One Robot



A Reference Architecture for Robotics Systems

One Robot

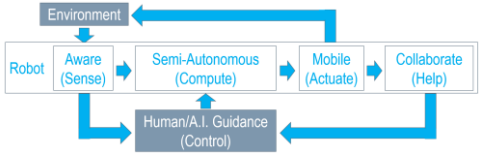


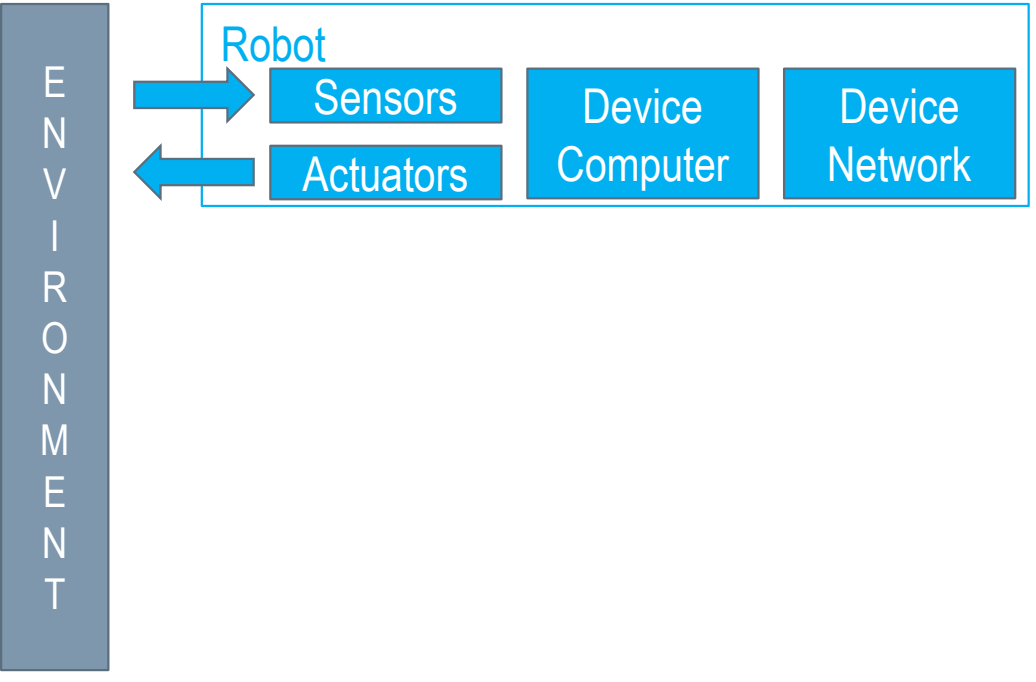


How to get hints from the human controller?

A Reference Architecture for Robotics Systems

One Robot





Human/A.I. Guidance (Control)

Internet Network

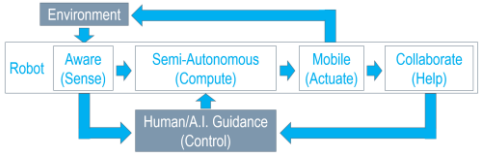
(WiFi, LiFi, Cellular, etc.)

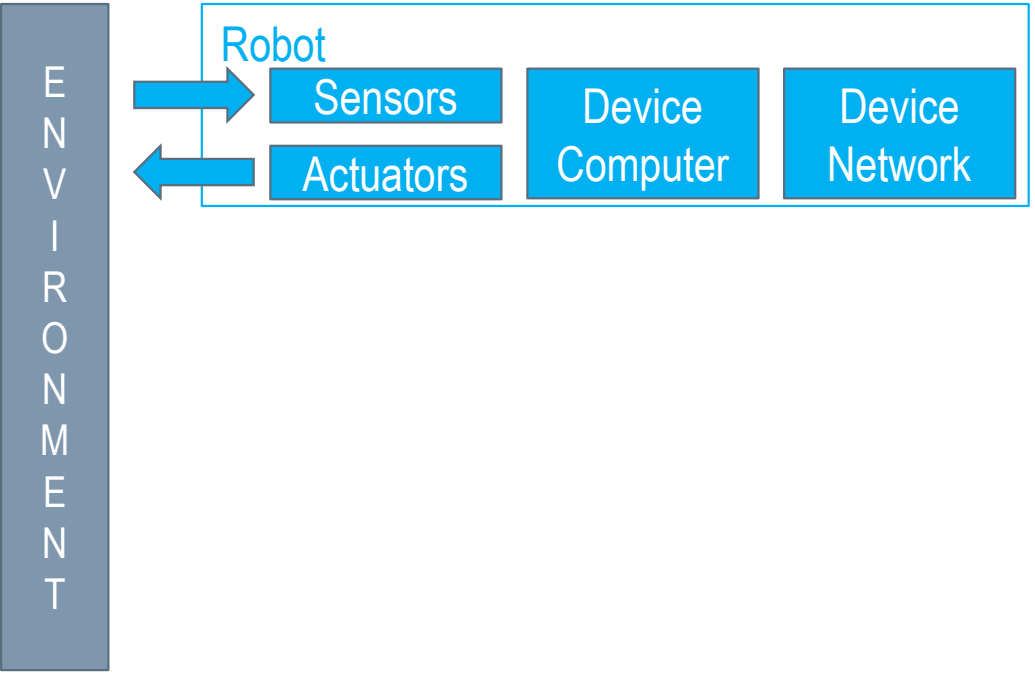
What if hints need to be timely, cannot traverse the Internet in time?

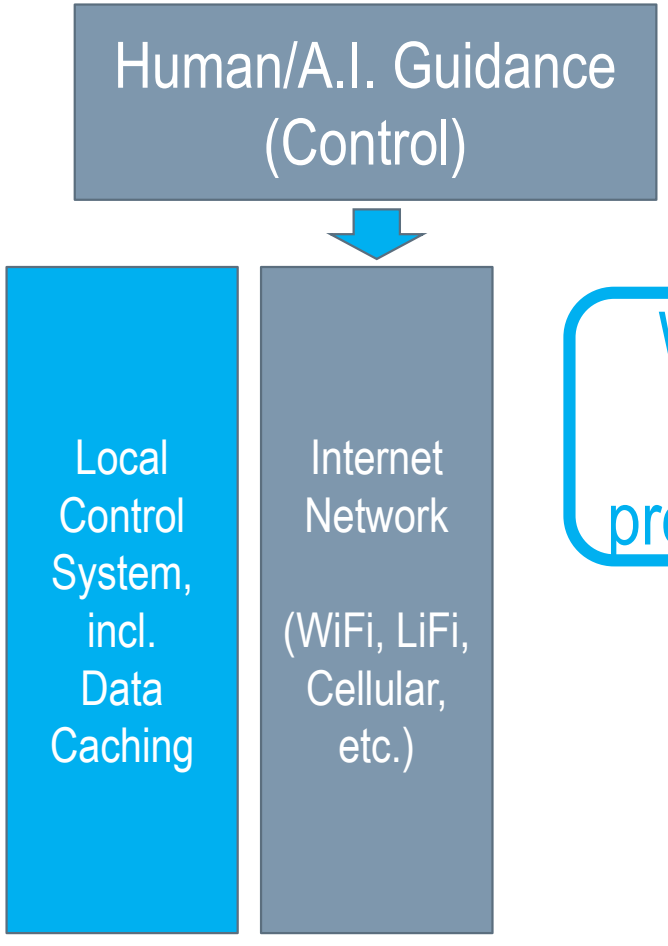
What if guidance should integrate hints from multiple controllers?

A Reference Architecture for Robotics Systems

One Robot



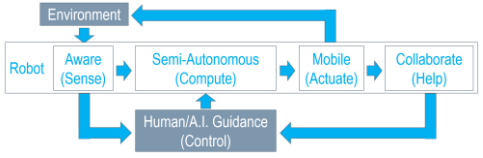




What if hints require complex data processing, algorithms?

A Reference Architecture for Robotics Systems

One Robot



ENVIRONMENT

Robot

Sensors

Actuators

Device Computer

Device Network

Human/A.I. Guidance (Control)

Local Control System, incl. Data Caching

Internet Network

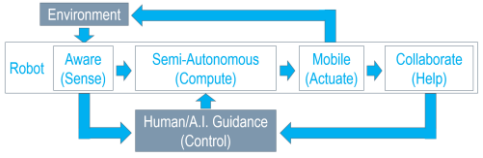
(WiFi, LiFi, Cellular, etc.)

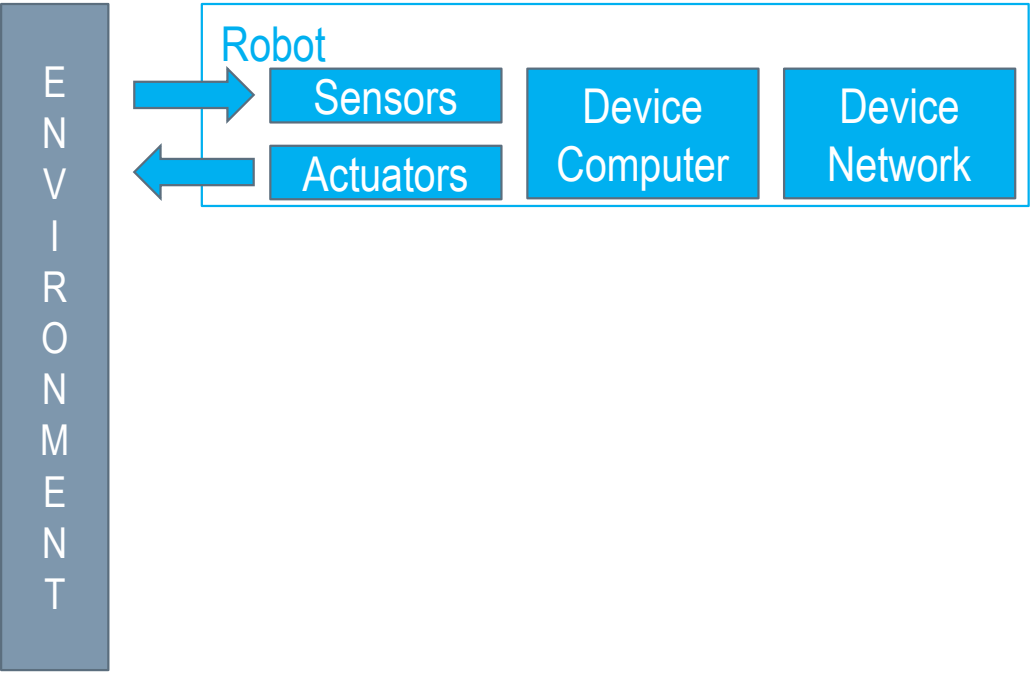
Timely hints, iff. based on simple data processing, algorithms?

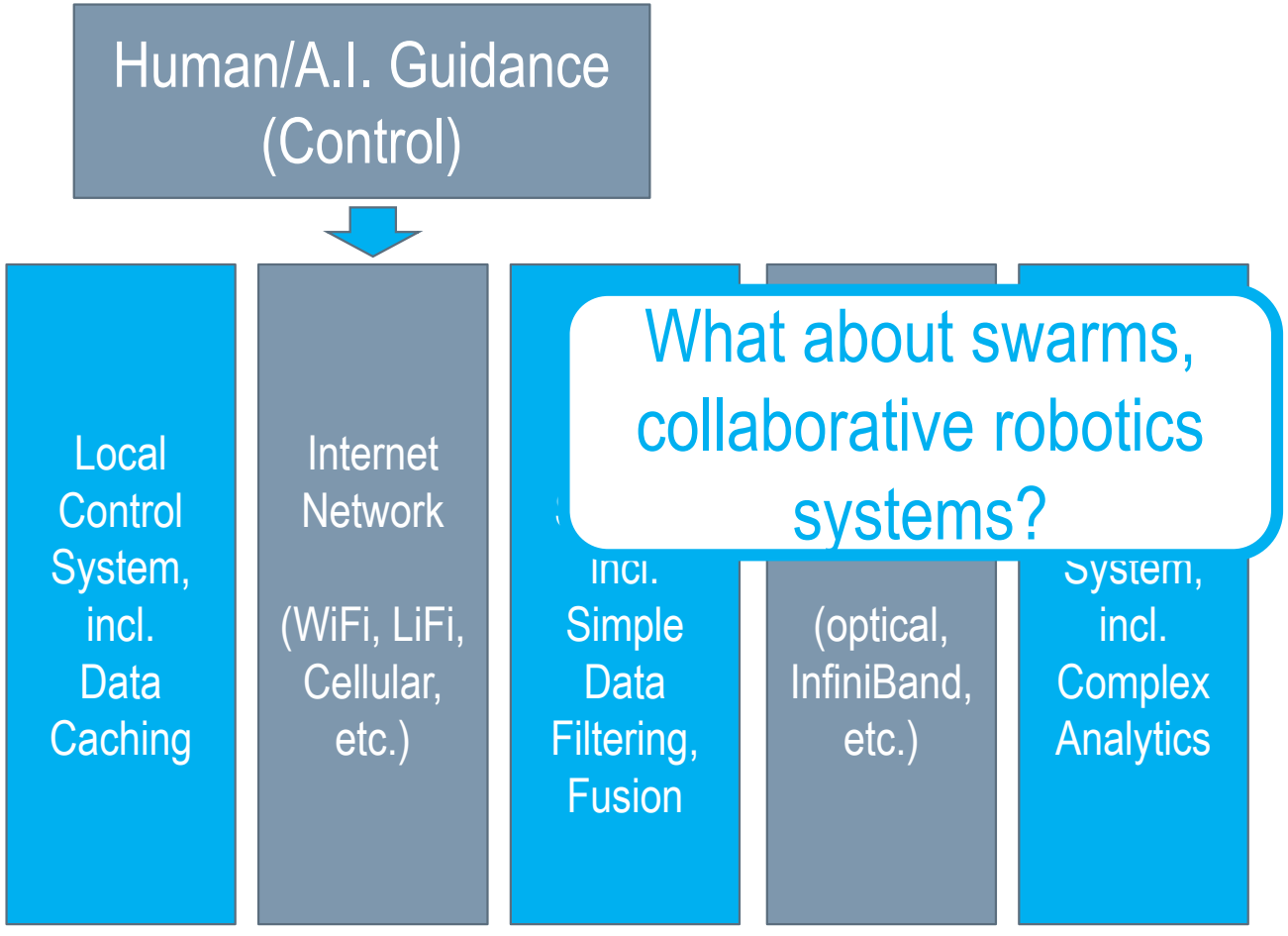
Avoid large data transfers when unnecessary?

A Reference Architecture for Robotics Systems

One Robot

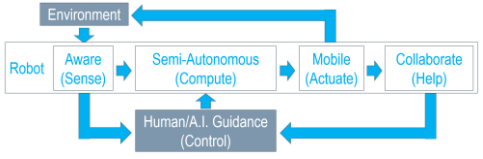


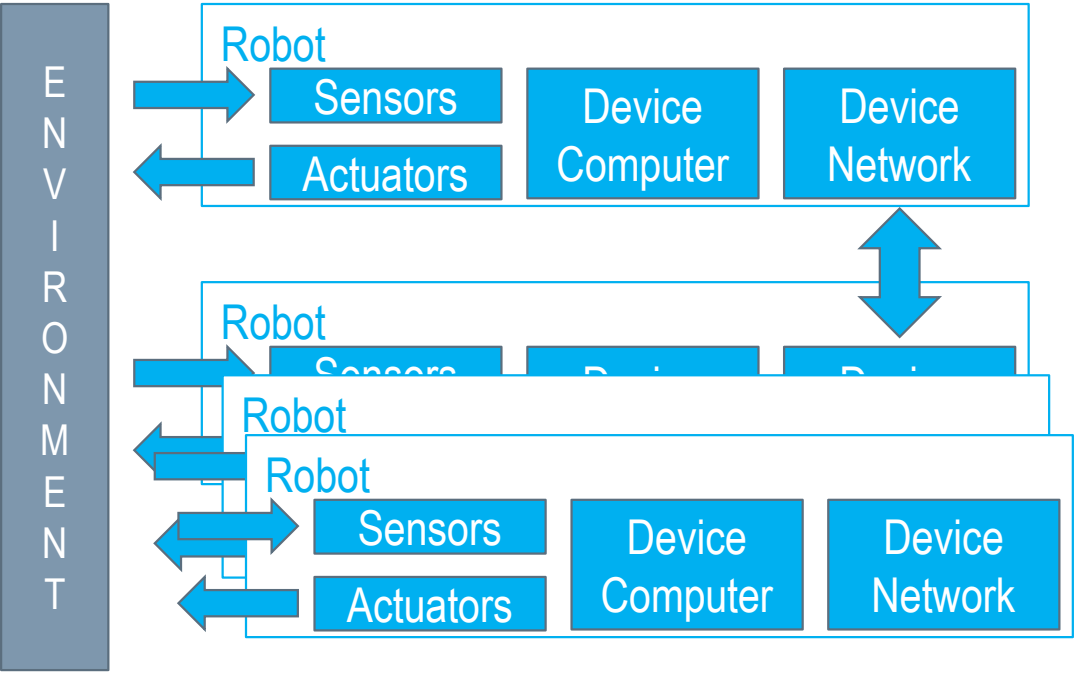




A Reference Architecture for Robotics Systems

Many Robots

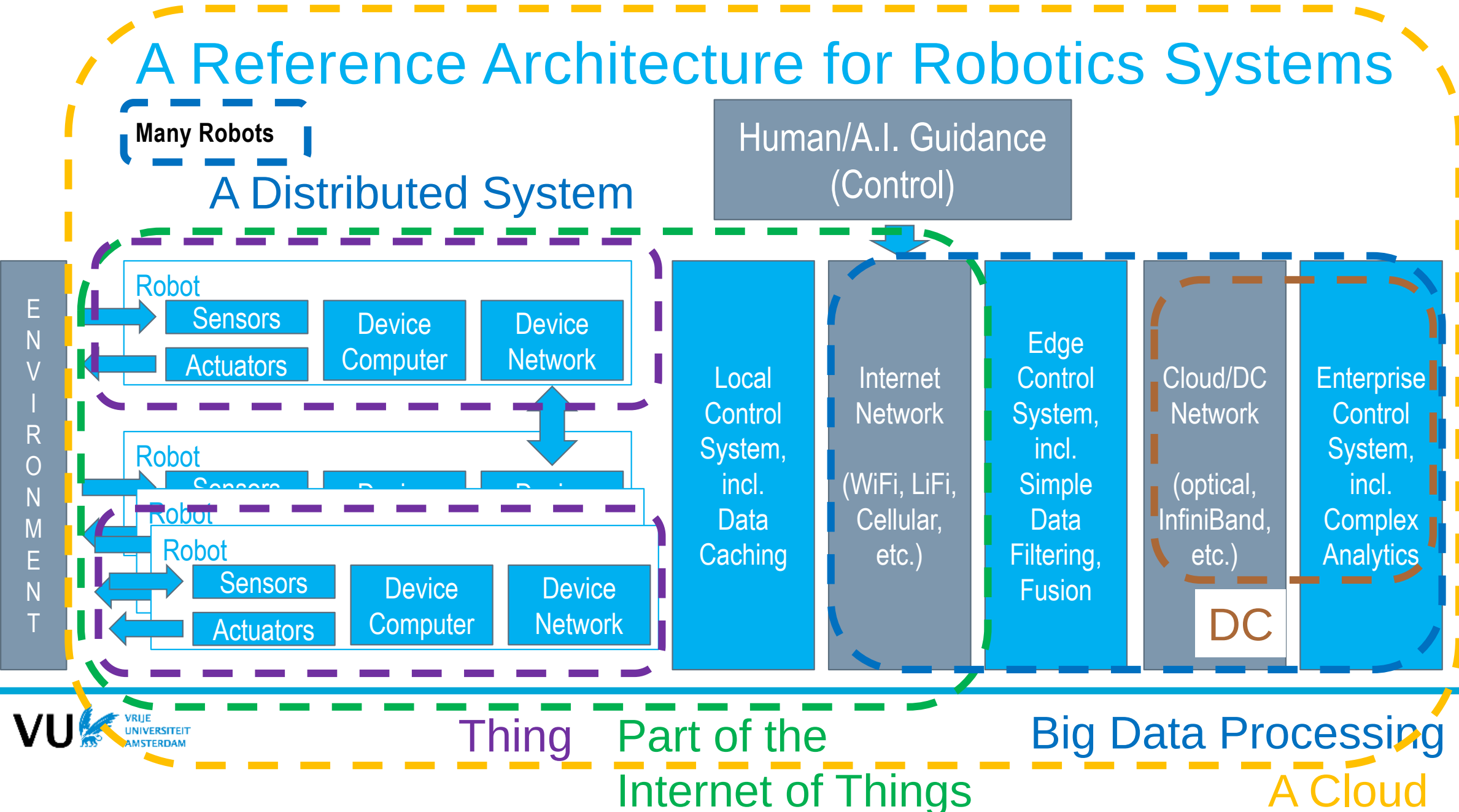




Human/A.I. Guidance (Control)

Local Control System, incl. Data Caching	Internet Network (WiFi, LiFi, Cellular, etc.)	Edge Control System, incl. Simple Data Filtering, Fusion	Cloud/DC Network (optical, InfiniBand, etc.)	Enterprise Control System, incl. Complex Analytics

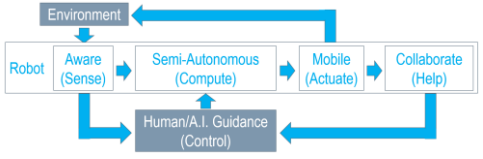
A Reference Architecture for Robotics Systems

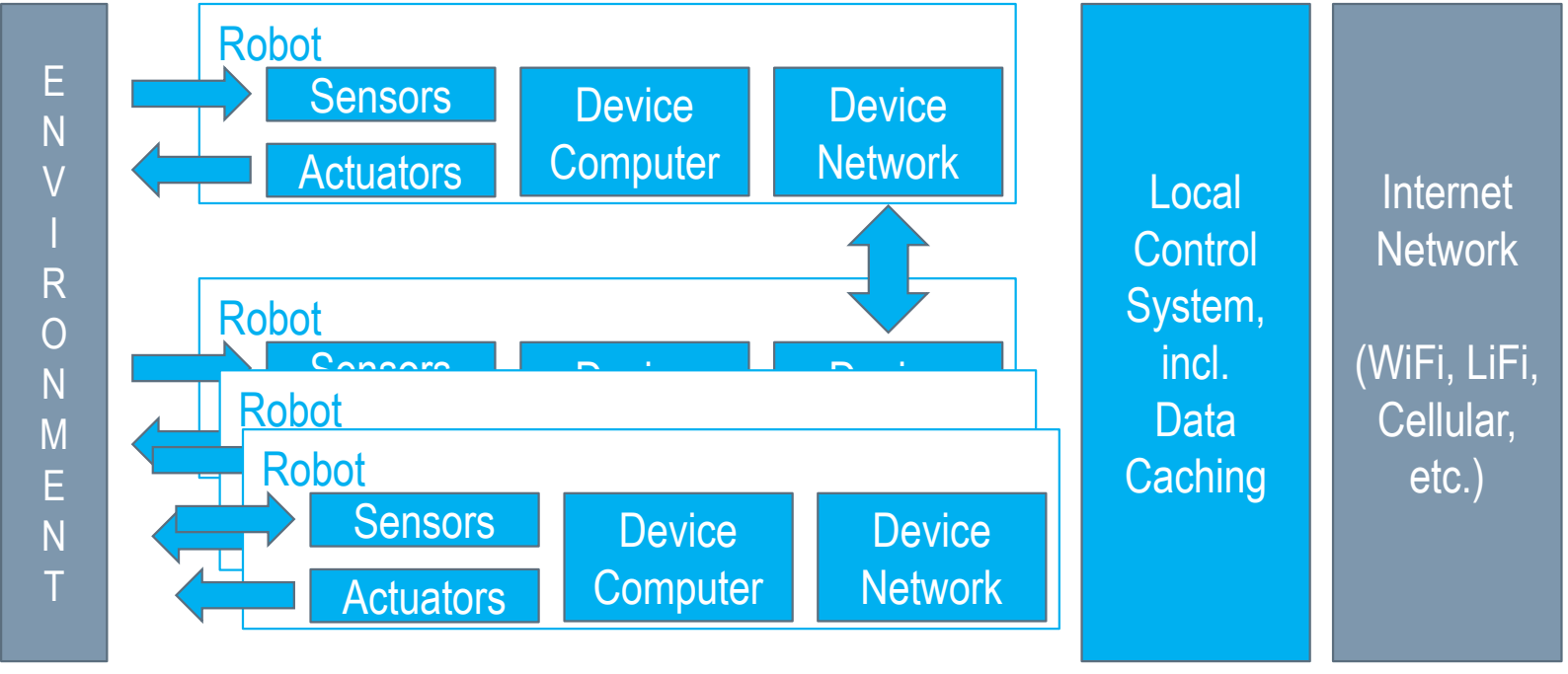


VU VRIJE UNIVERSITEIT AMSTERDAM

A Short Introduction to Distributed Systems

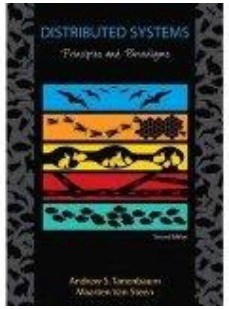
Many Robots





What is a Distributed System?

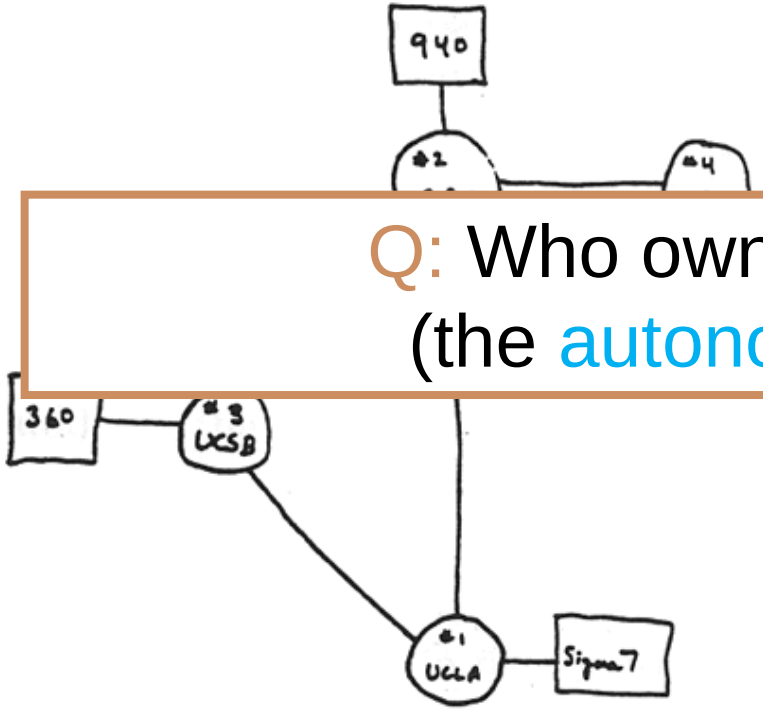
"A collection of independent computers that appears to its users as a single coherent system
- **Steen and Tanenbaum** in
Distributed Systems: Principles and Paradigms

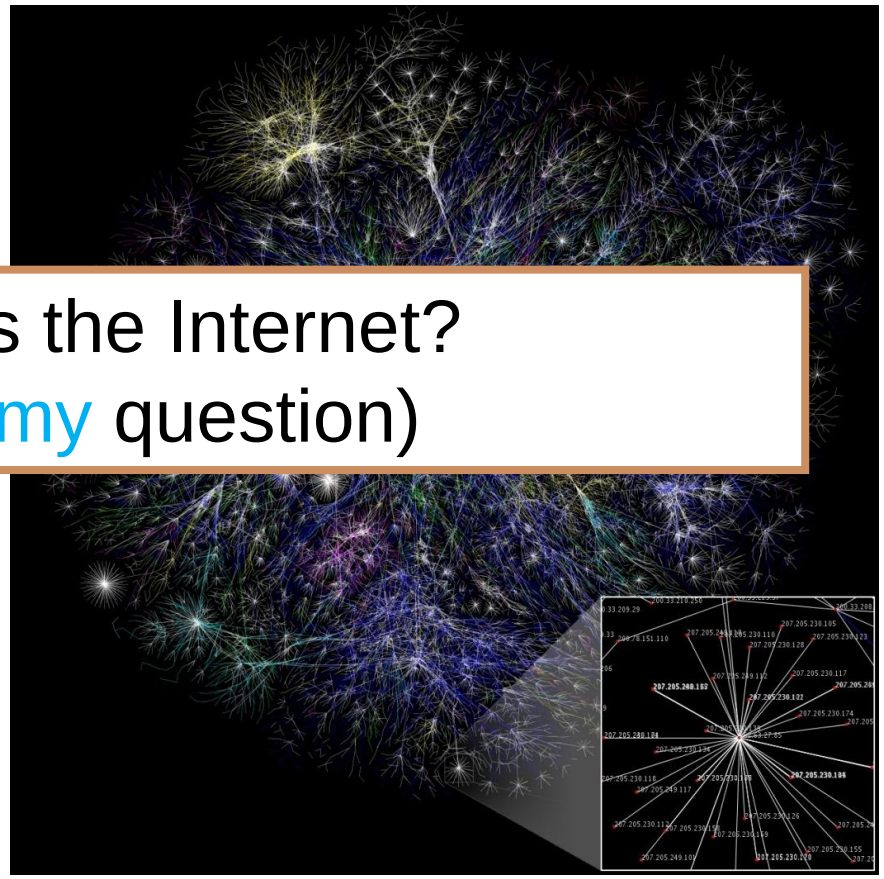


"You know you have a distributed system when the crash of a computer you've never heard of stops you from getting any work done."
- **Leslie Lamport** in Security Engineering, Ch.6

"an application that executes a collection of protocols to coordinate the actions of multiple processes on a network, such that all components cooperate together to perform a single or small set of related tasks."
- Google University, Introduction to DS Design
http://www.hpcs.cs.tsukuba.ac.jp/~tatebe/lecture/h23/dsys/dsd-tutorial.html

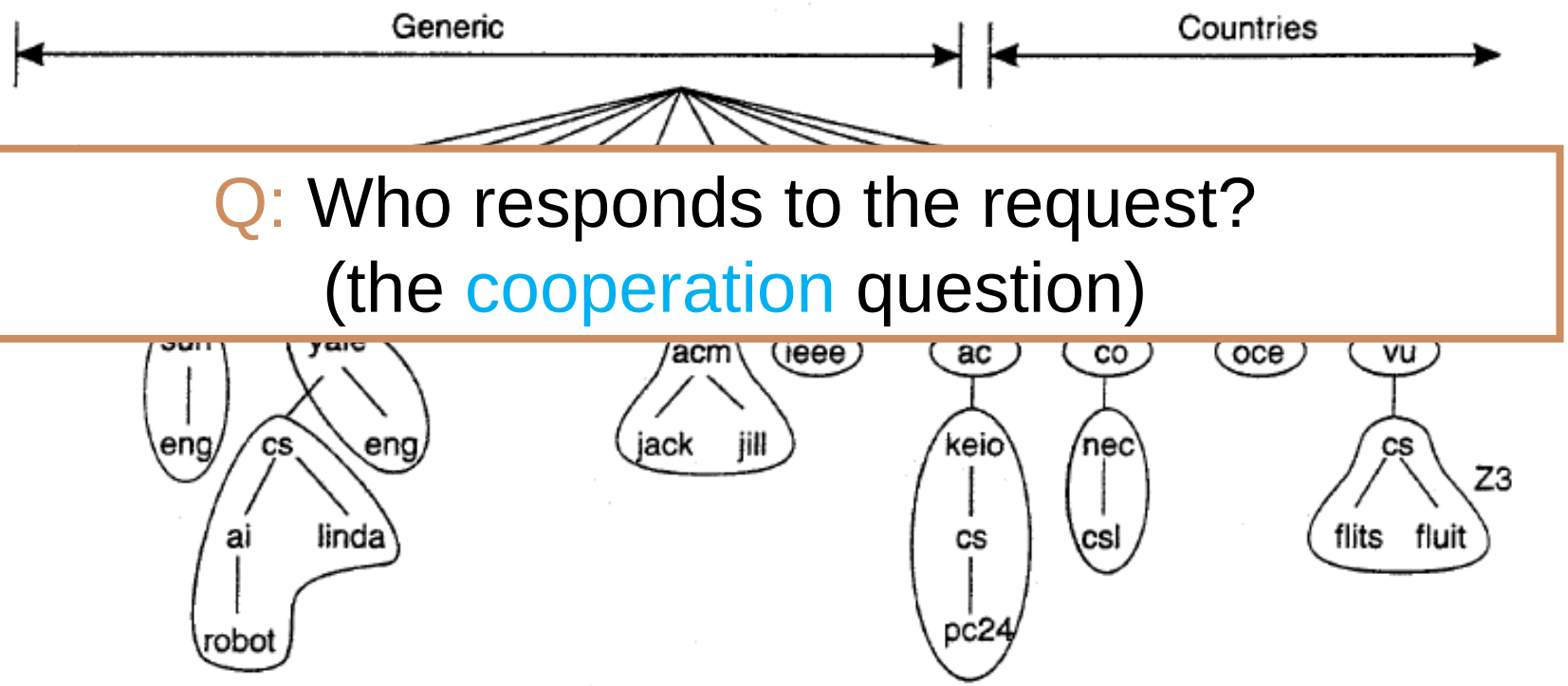
The Internet Is a Distributed System





Q: Who owns the Internet?
(the autonomy question)

The Domain Name System (DNS)



Q: Who responds to the request?
(the cooperation question)

The Google Data Centers

Q: How to compile the query answer?
(the communication question)

View our data centers in a larger map

Americas

- Berkeley County, South Carolina
- Council Bluffs, Iowa
- Douglas County, Georgia
- Quilicura, Chile
- Mayes County, Oklahoma
- Lenoir, North Carolina
- The Dalles, Oregon

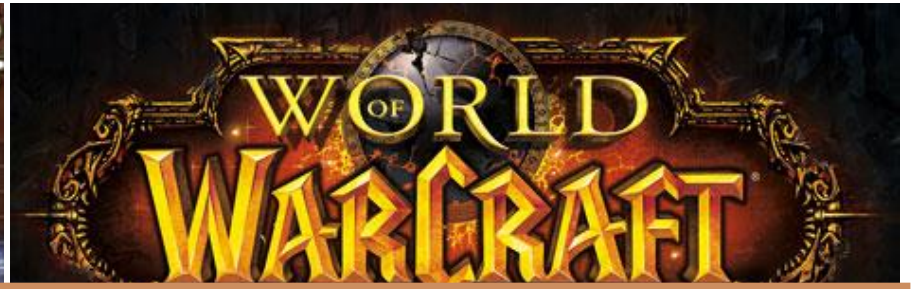
Asia

- Hong Kong
- Singapore
- Taiwan

Europe

- Hamina, Finland
- St Ghislain, Belgium
- Dublin, Ireland

The Online Gaming World



Q: What happens when the performance drops?
(the non-functional question)

- 10 data centers
- 13,250 server blades,
  75,000+ cores
- 1.3PB storage
- 68 sysadmins (1/1,000 cores)

http://www.datacenterknowledge.com/archives/2009/11/25/wows-back-end-10-data-centers-75000-cores/

The Core Idea through An Example
BitTorrent: A Distributed System

Q: Autonomy? Cooperation? Communication?

Q: Does this system scale? Why? How?

Q: What is the structure of this system?
What is the state of each node? How do they synchronize?

Q: How does the performance of this system change with the increase in the number of users?

Q: When is this system available?
What does it do to increase its reliability?

Q: Is this system efficient?

Q: Which parts of this system need consistency?
Achieved?

Main Characteristics of DS

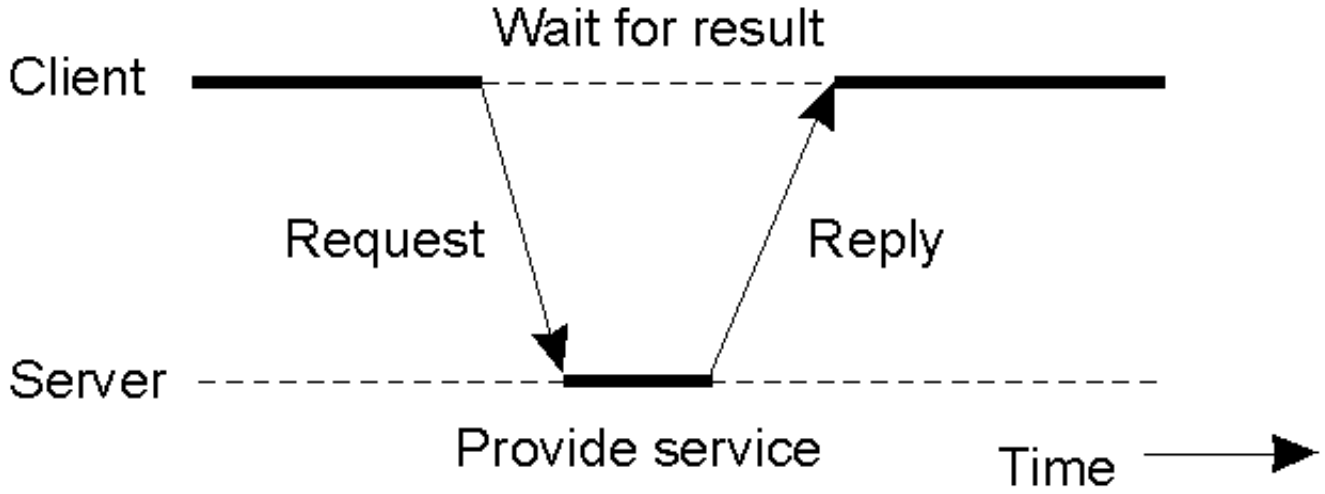
1. Scalability
2. Predictable high performance
3. Reliability and availability
4. Efficiency (resource sharing)
5. Consistent (distributed) state
6. Close-to-users
7. So many other concerns:
   Security, Inter-operability, …

How to Build a Distributed System?

Models and Tools

- General Frameworks
  - Client/Server: roll your own, WebSockets
  - Decentralized, peers/actors: Akka
  - Hybrid/Hierarchical/Decentralized, message passing: MPI
- Specialized Frameworks
  - Data-intensive: Hadoop
  - Compute-/Data-intensive: Pegasus, Triana, Kepler

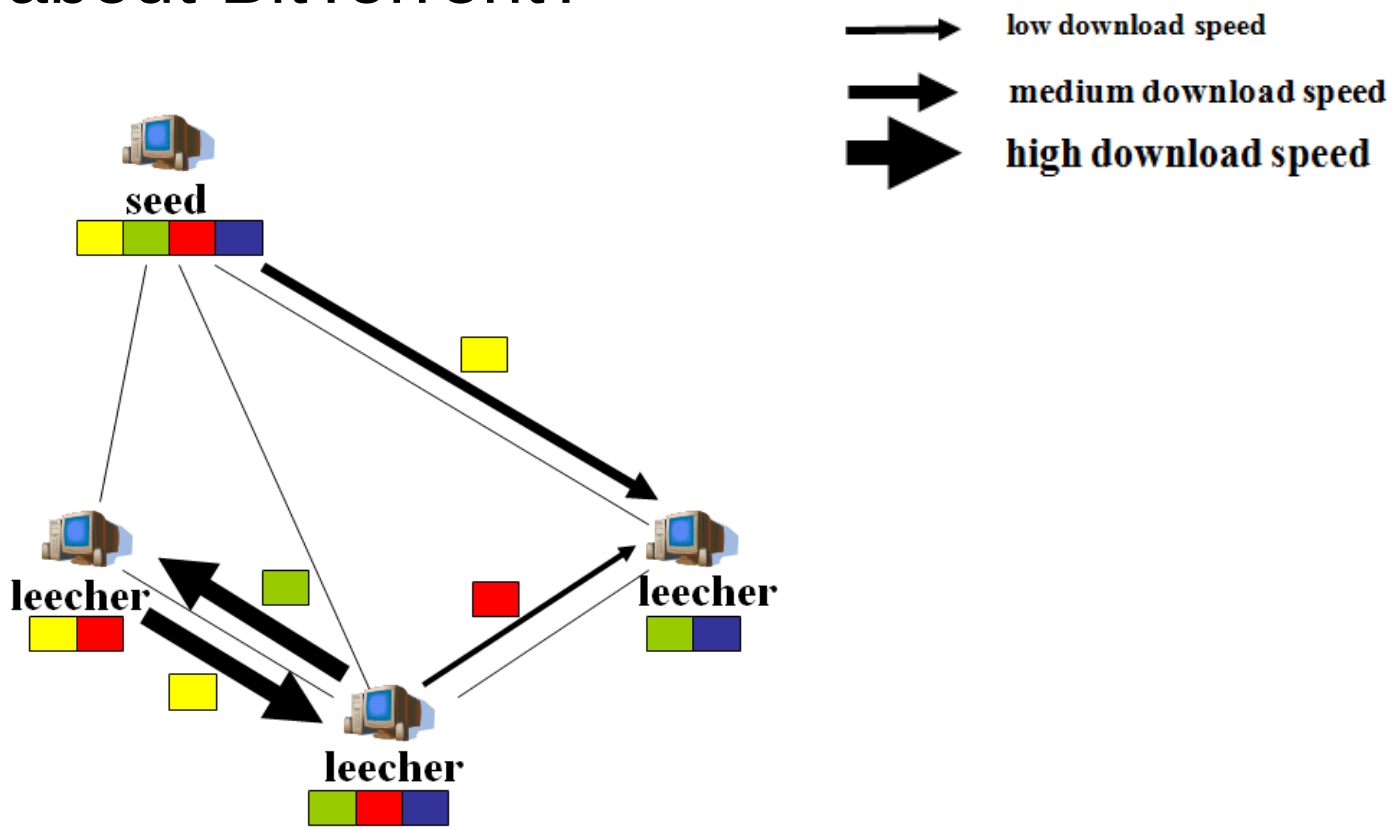
The Centralized Client/Server Model



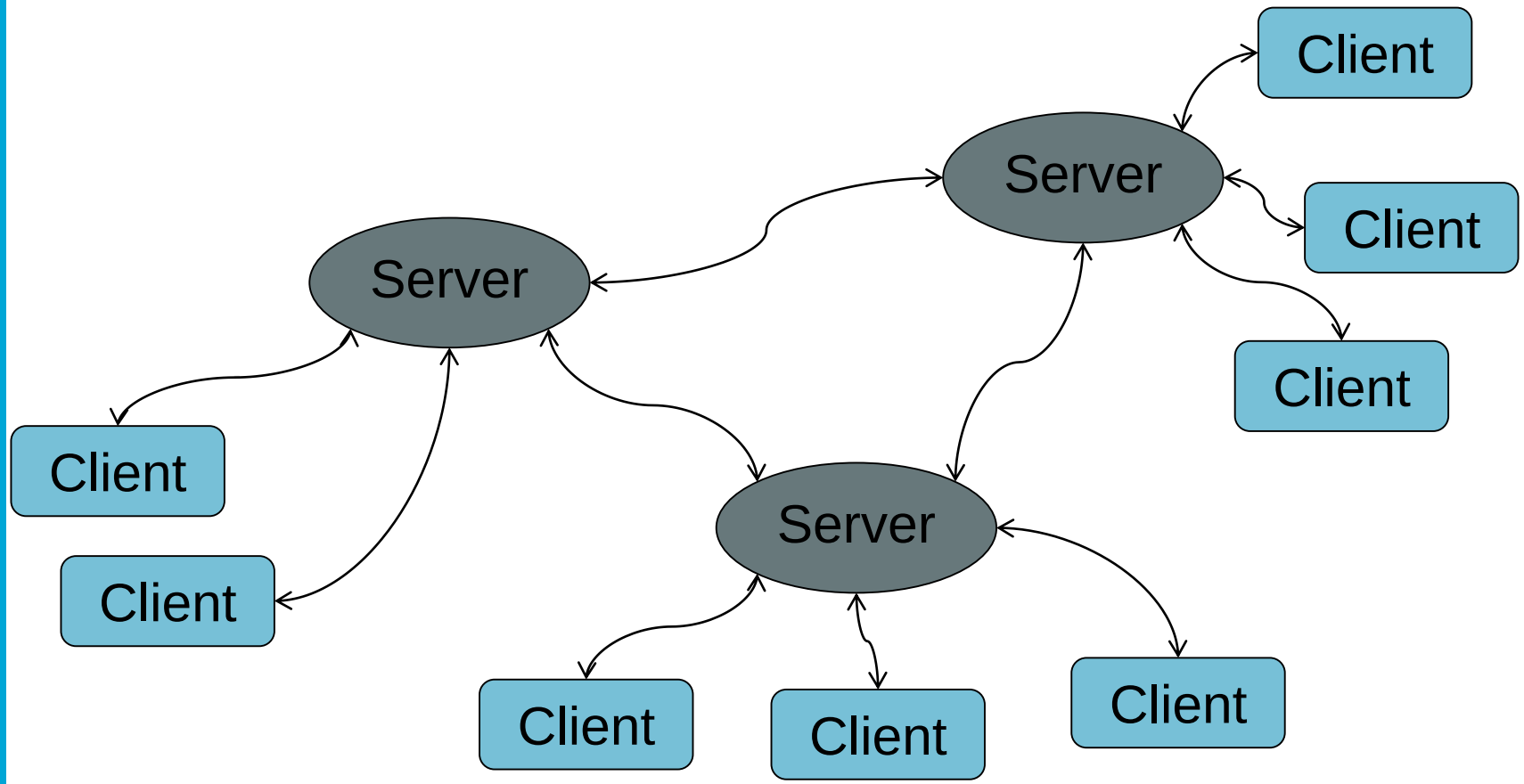
- Client-server model = message exchange with specific pattern: the client sends requests, then the server replies
- Special forms of client-server model:
  - in imperative languages**, remote procedure calls (RPC)**
  - in OO languages, **remote method invocations (RMI)**
  - on the web, **http**

The Decentralized (fully: Peer-to-Peer) Model

- Remember what we have discussed about BitTorrent?



Hybrids: Multiple Servers or Hierarchical P2P Systems or …



The CAP Theorem

- Properties of distributed systems
  - Consistency (C)—a single up-to-date copy of the data
  - Availability (A)—possible to perform updates on the data
  - Partitions (P)—tolerance to interruption of network

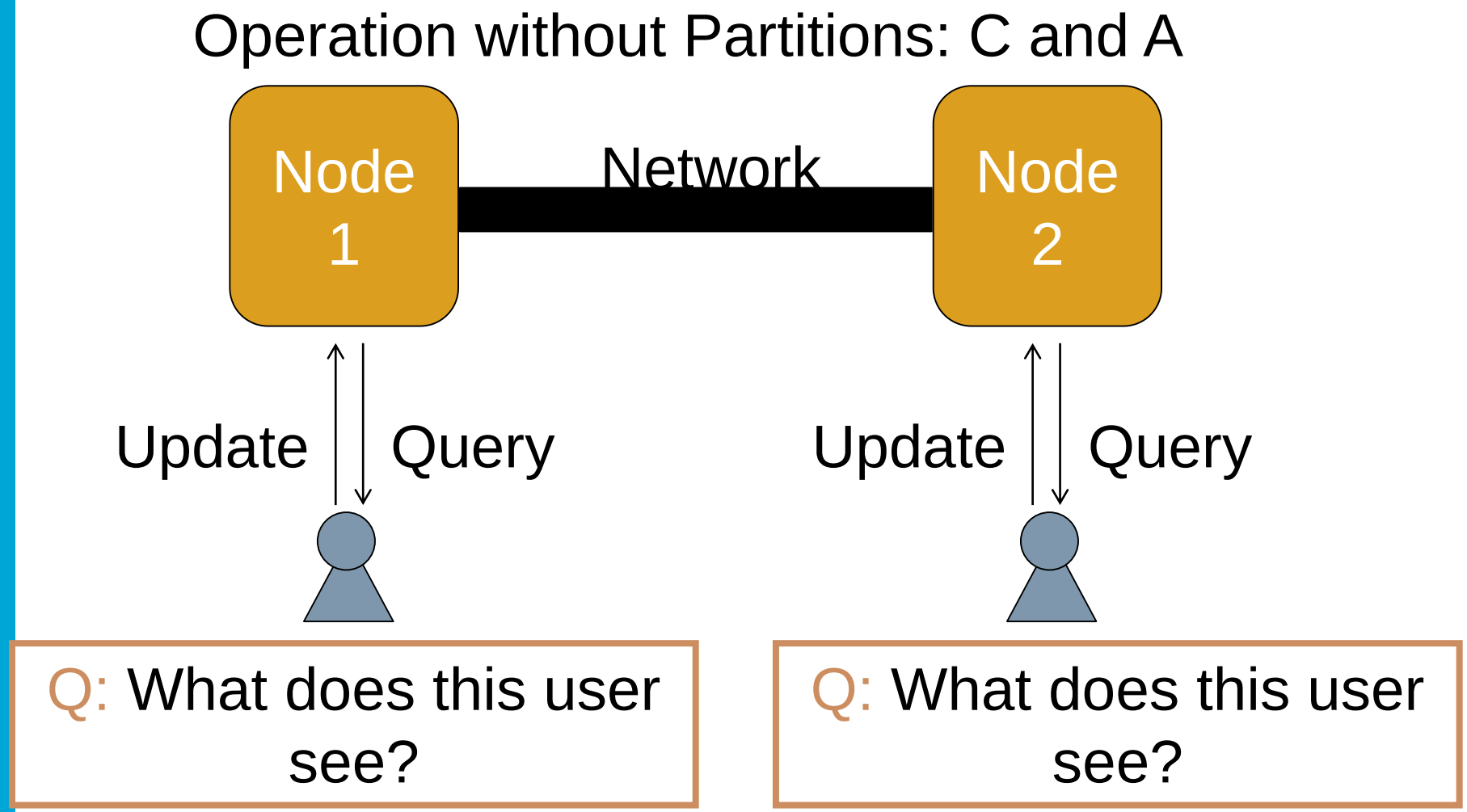
- The CAP Theorem, Eric Brewer (Cal):

> *"Any networked shared-data system can have at most two out three of the CAP properties"*
> - **Eric Brewer** at
> Principles Of Distributed Computing (PODC), 2000

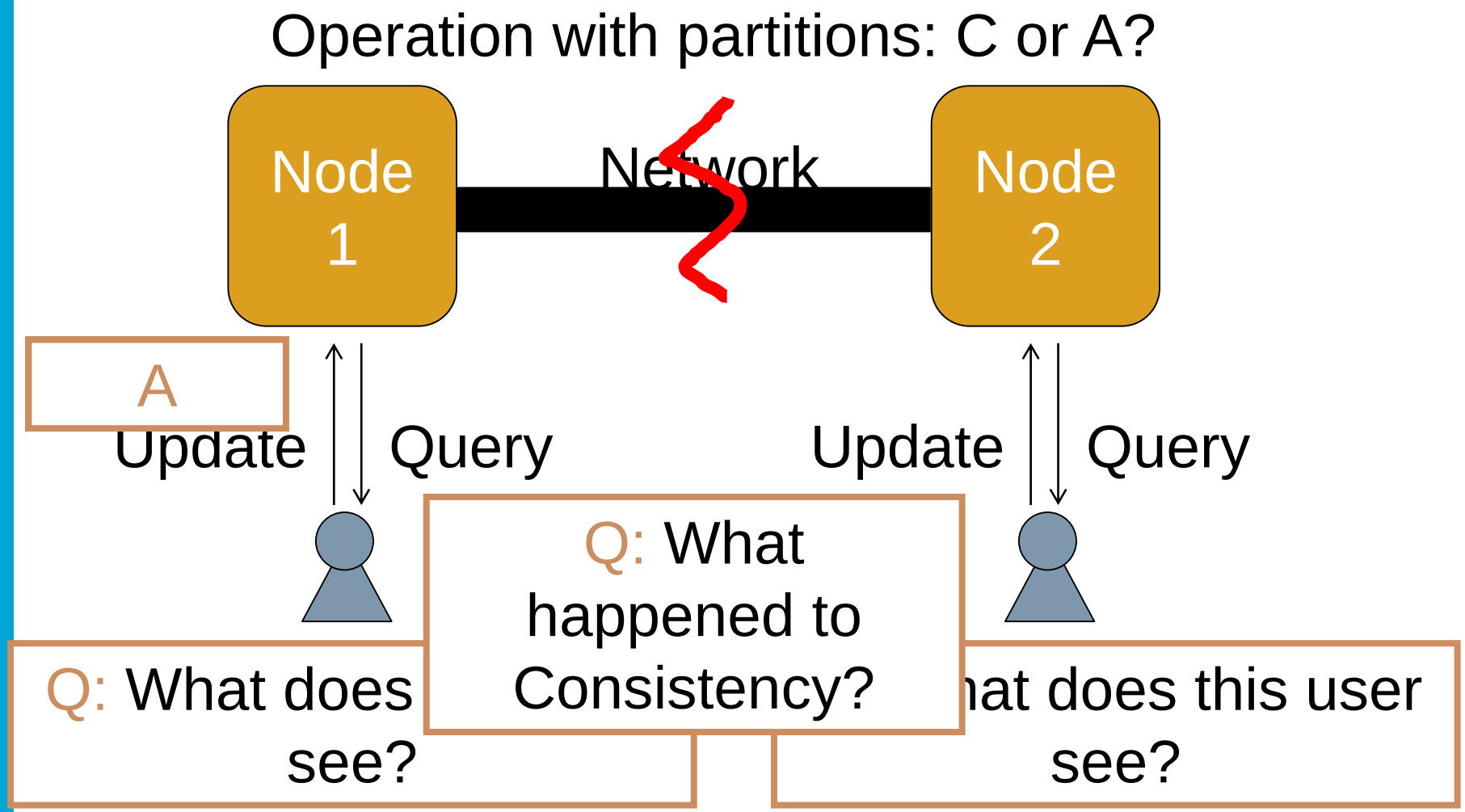
proof

Seth Gilbert, Nancy A. Lynch: Perspectives on the CAP Theorem. IEEE Computer 45(2): 30-36 (2012)

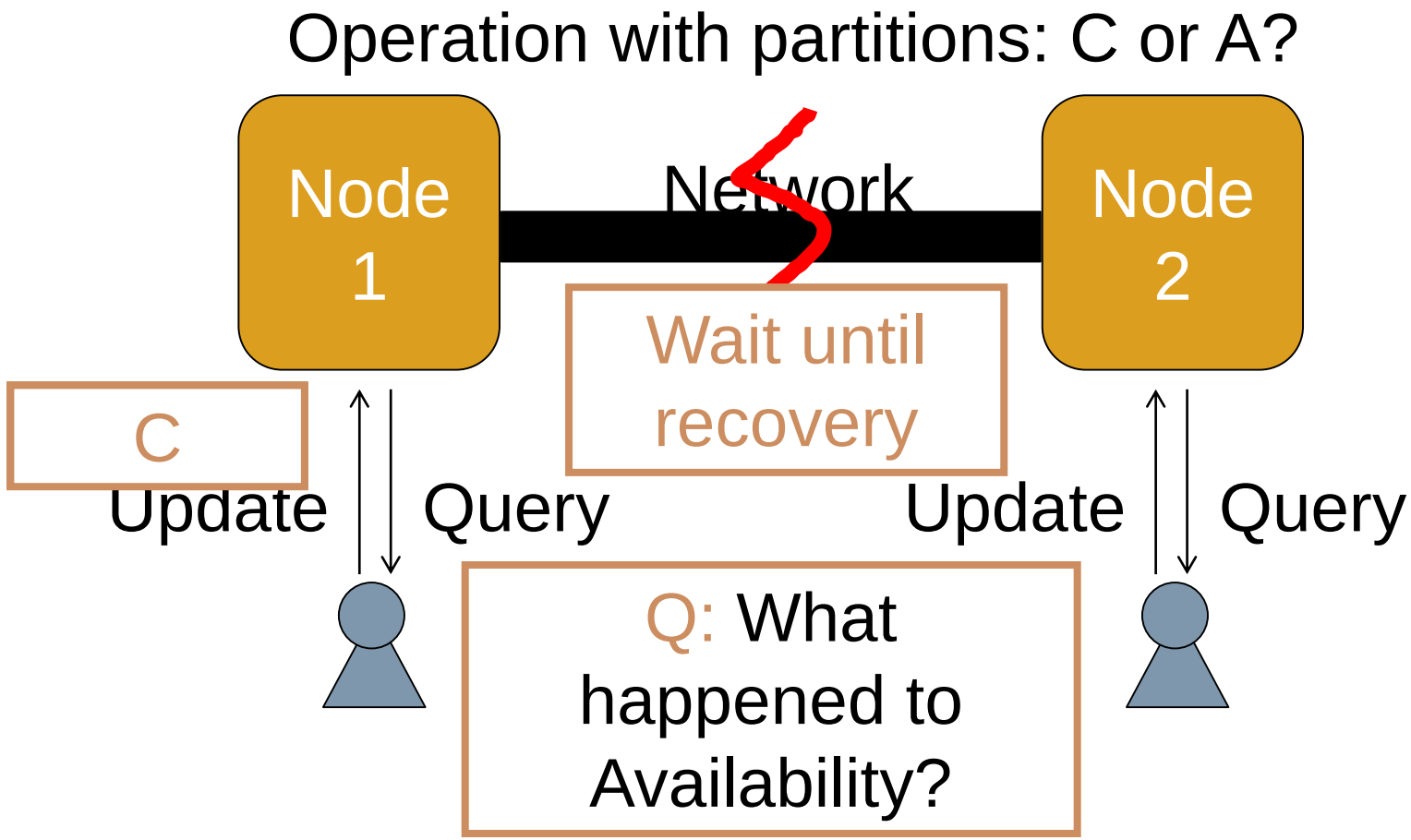
The CAP Theorem: a Visual Proof



The CAP Theorem: a Visual Proof



The CAP Theorem: a Visual Proof



The CAP Theorem: 12 Years After

What does "2 of 3" mean in practice?

Q: Can you forfeit P?

- Partitions may not occur in small clusters
- Partitions do occur in data centers

Q: Can you use C or A in client-server?

- No, if client cannot reach service
- Yes, if client can operate offline (choose A over C)

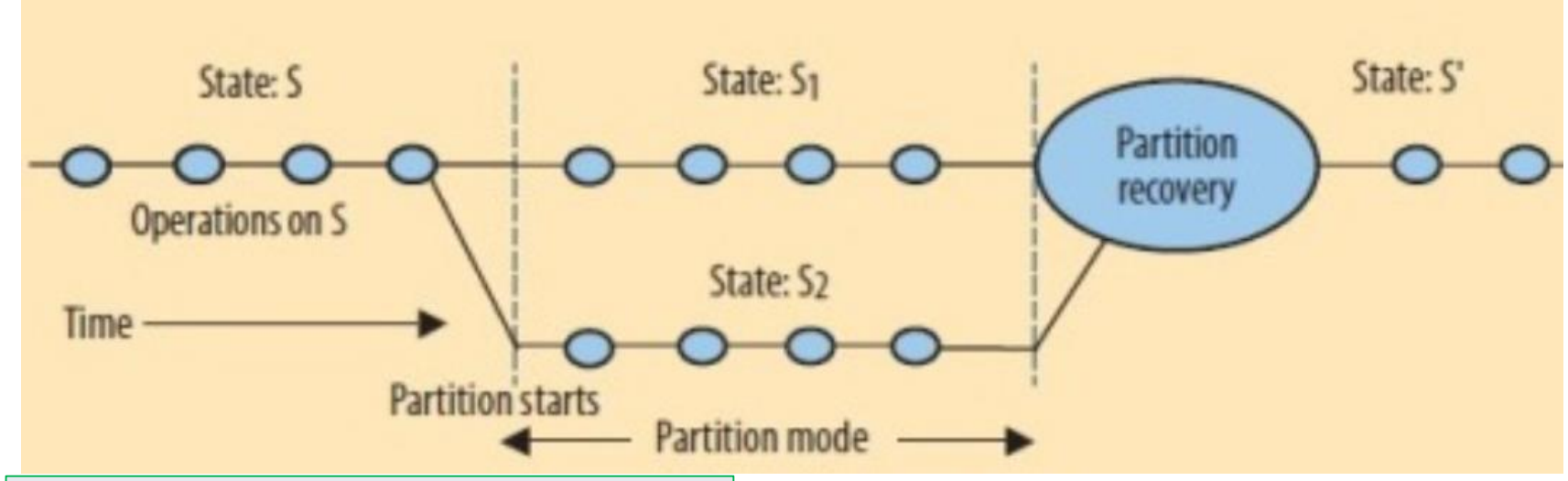
Q: Can you forfeit C? (homework)

Eric A. Brewer: Pushing the CAP: Strategies for Consistency and Availability. IEEE Computer 45(2): 23-29 (2012)

The CAP Theorem: 12 Years After

Few Ps, so No more 2 of 3, rather ***probabilistic C, A, and P***

"The modern CAP goal should be to maximize combinations of consistency and availability that make sense for the specific application. […] operation during a partition and recovery afterwards"



Update to 2000 statement

Eric A. Brewer: Pushing the CAP: Strategies for Consistency and Availability. IEEE Computer 45(2): 23-29 (2012)

Distributed Systems, a Design Example
2Fast: Collaborative Downloading

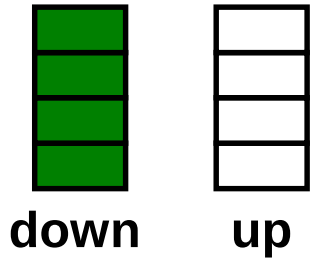
“In two years time we will all have petabytes on our key chains and will not need BitTorrent at all”
(anonymous, for the sake of this course, 2005)

But what if this would not happen?

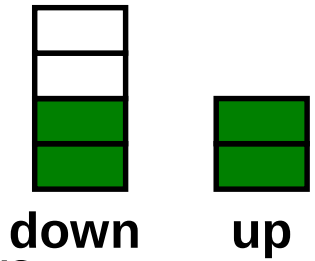
P. Garbacki, A. Iosup, D.H.J. Epema, and M. van Steen, "2Fast: Collaborative Downloads in P2P Networks," *6-th IEEE International Conference on Peer-to-Peer Computing*, 2006 (**best-paper award**).

Peer-to-peer data transfer protocols

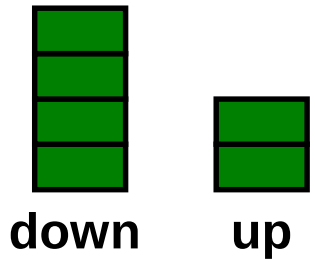
- Gnutella, Kazaa
  - no incentives for bandwidth sharing
  - free-riders sensitive
  - poor utilization of upload bandwidth



- BitTorrent (BT), Slurpie
  - tit-for-tat enforces fairness
  - temporal fairness cannot handle asymmetric links
  - poor utilization of download bandwidth

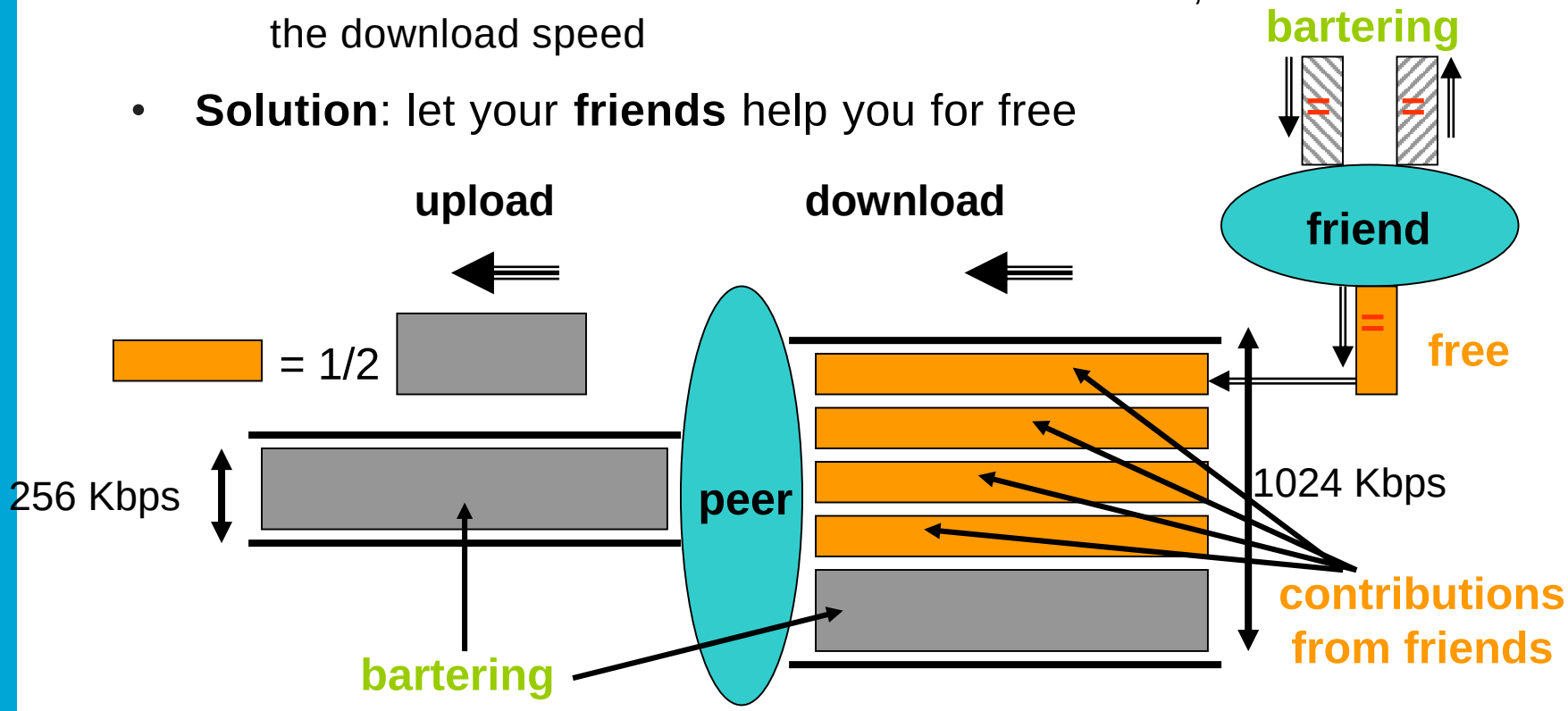


- **2Fast: BT+collaborative downloads**
  - no tit-for-tat within a single session
  - cross-session bandwidth sharing
  - full utilization of upload AND download links



Cooperative downloads: basic idea

- **Problem**:
  - most users have **asymmetric** upload/download links
  - because of the **tit-for-tat** mechanism of Bittorrent, this restricts the download speed
- **Solution**: let your **friends** help you for free



Two protocol extensions

- **Redundant chunks download**
  - **problem**: discrimination of helpers; more restrictive chunk selection + fewer chunks to offer, so limited bartering possibilities
  - **solution**: the same chunk may be downloaded by different helpers
- **Sharing of swarm information**
  - **problem**: slow start; finding suitable bartering partners takes time
  - **solution**: collaborating peers exchange information on other peers in the swarm

DOWNLOAD SPEED-UP: ANALYTICAL MODEL

- Every helper **equally splits its upload capacity** between bartering and helping the collector
- So **every additional helper** increases the download speedup of the collector by 0.5, up to a point
- The **maximum number of useful helpers** (and so the maximum speedup) can easily be computed
- N, S: the numbers of **leechers** and **seeders** in the system
- c, μ: the download/upload capacity of all peers
- **Download bandwidth** of the collector with **h helpers**:

free from seeders

$$\frac{S}{N}\mu + \mu + \frac{1}{2}\sum_{i=1}^{h}\left(\frac{S}{N}+1\right)\mu$$

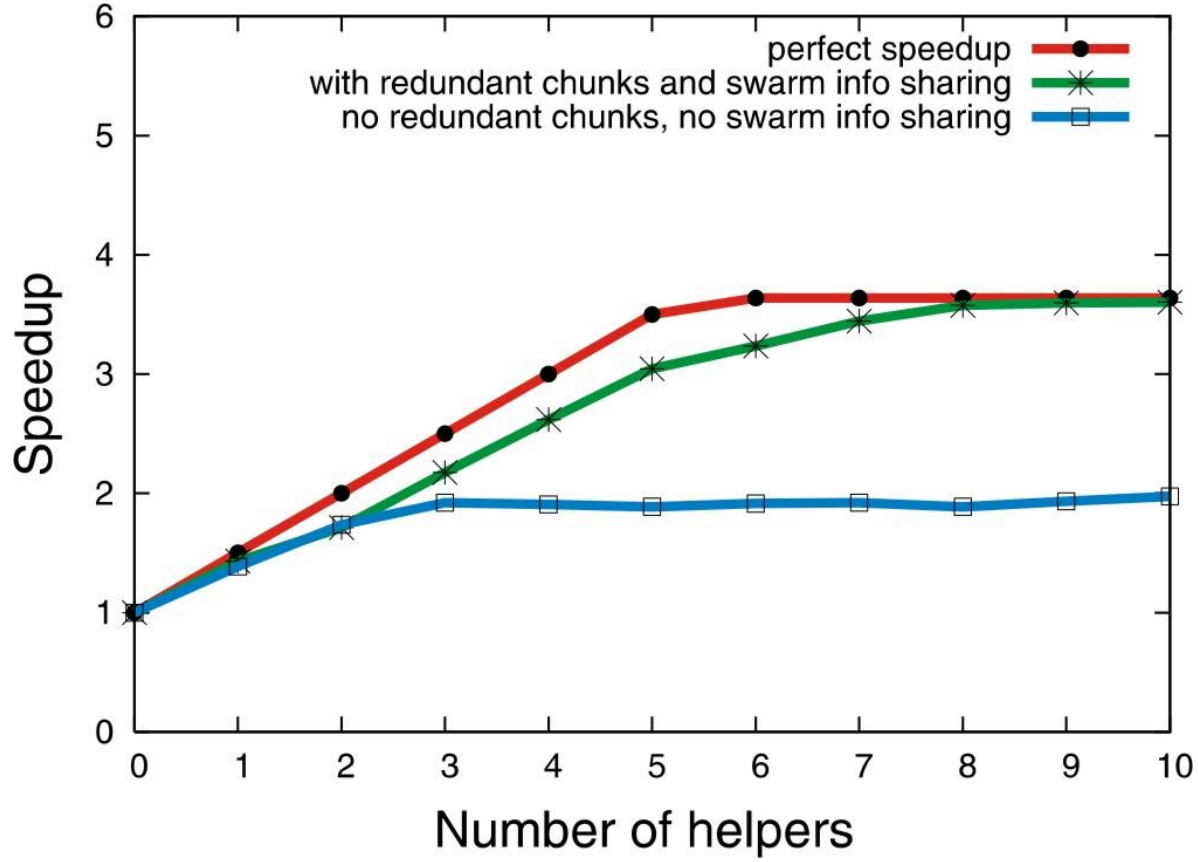
bartering

from helpers

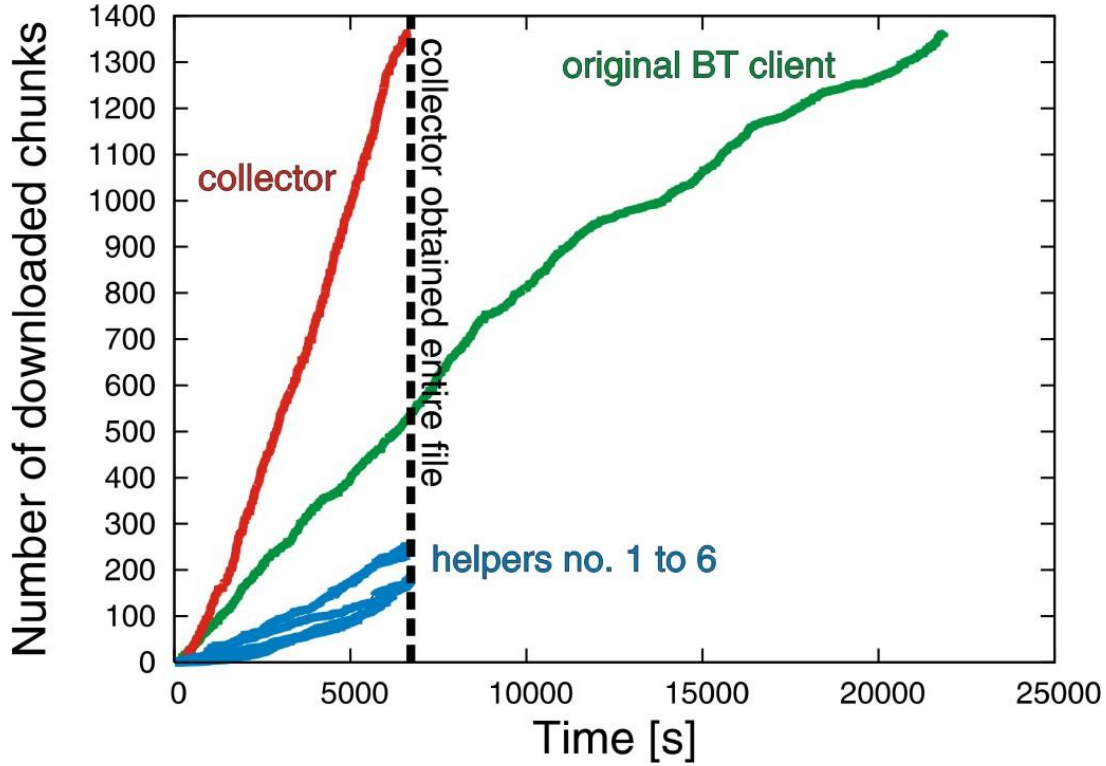
Experimental setup

- Experiments performed in a real environment – collaborating peers connect to existing BitTorrent swarms
- Collaborating peers connected through ADSL links: 256kbps up / 1024kbps down
- Downloaded file size: 700 MB
- Swarm size: 100 leechers, 10 seeders

Speedup vs number of helpers

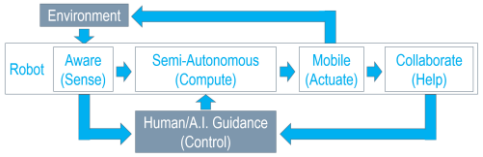


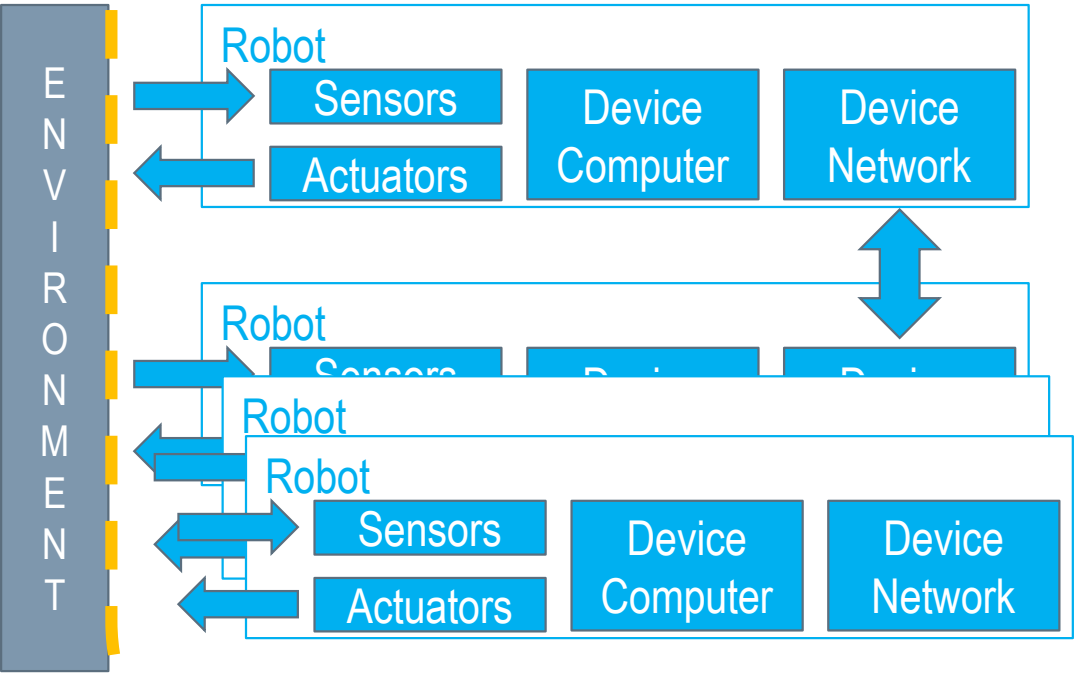
Download progress



A Short Introduction to Cloud Computing

Many Robots





Human/A.I. Guidance (Control)

Local Control System, incl. Data Caching

Internet Network (WiFi, LiFi, Cellular, etc.)

Edge Control System, incl. Simple Data Filtering, Fusion

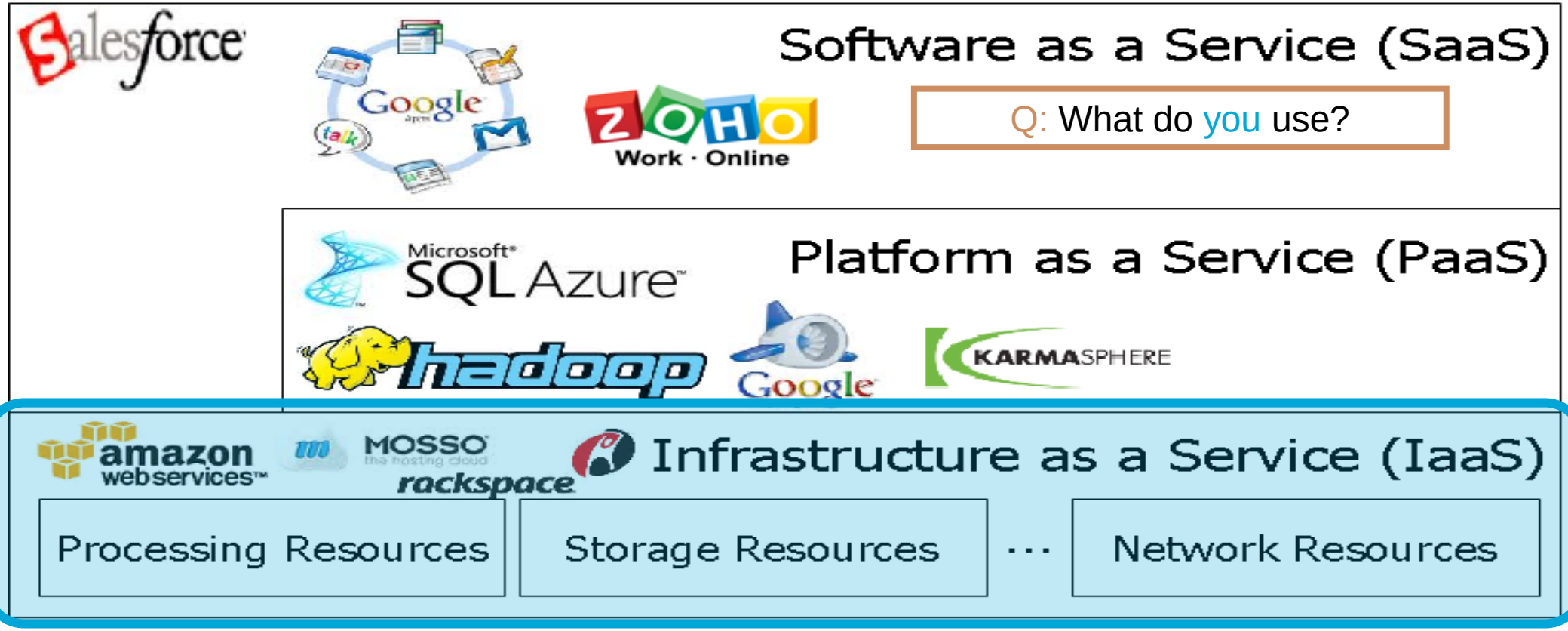
Cloud/DC Network (optical, InfiniBand, etc.)

DC

Enterprise Control System, incl. Complex Analytics

What is Cloud Computing?
A Useful IT Service

"Use only when you want! Pay only for what you use!"

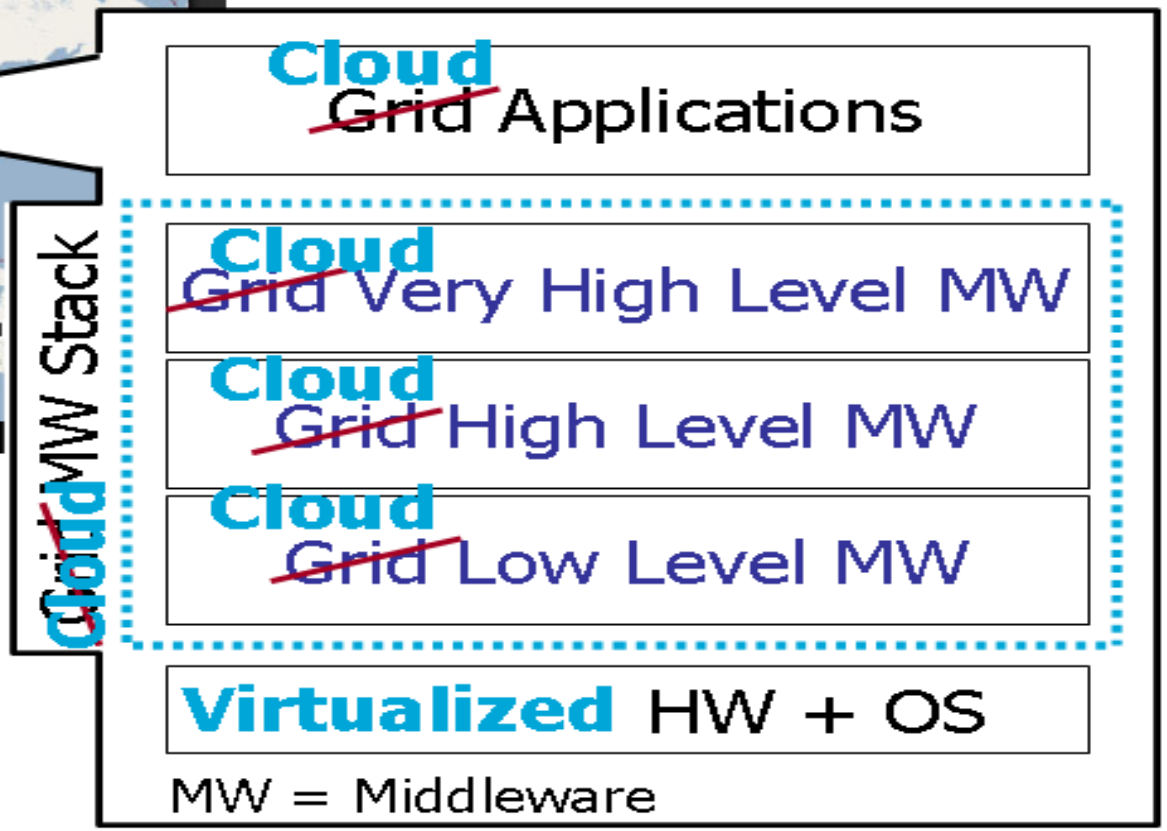


What is Cloud Computing?
2. A Descendant* of the Grid Idea

Source: http://royal.pingdom.com/2008/04/11/map-of-all-google-data-center-locations/

"A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [+ for] nontrivial QoS." I. Foster, 1998 + 1999

* Subset.



Have you noticed "QoS"? What is that?

Main Characteristics of IaaS Clouds

1. On-Demand Pay-per-Use
2. Elasticity (cloud concept of Scalability)
3. Resource Pooling
4. Fully automated IT services
5. Quality of Service

Q: Sounds great, but …
How can we make
all this stuff happen?

Introducing datacenters & datacenter-based clouds = ICT service creation for everyone

Factories Powering the Goods Economy and Better Living Standards

Creator

Factory (1980)

Consumer

Datacenters = Digital Factories Powering the Digital Economy and Better Living Standards

Creator

Datacenter (2010)

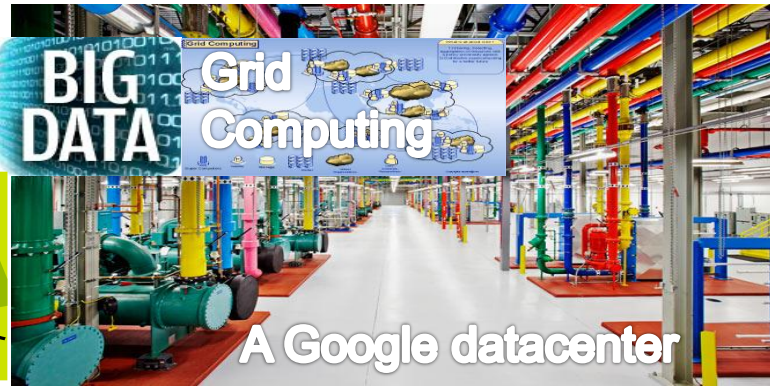
Consumer

We want more of this

"15% ICT market is simple DC services"
"Already 60+ bn.€/year"
"Consume more energy than the transport industry"

We want less of this (and other issues raise by DCs)

Scientific Challenges in Cloud Computing



How to massivize?

- Super-scalable, super-flexible, yet efficient ICT infrastructure
- End-to-end automation of large-scale, simple and complex processes
- Dynamic, compute- and data-intensive workloads
- Evolving, heterogeneous hardware and software
- Strict performance, cost, energy, reliability, and fairness requirements
- … all these, without needing much expertise from customers

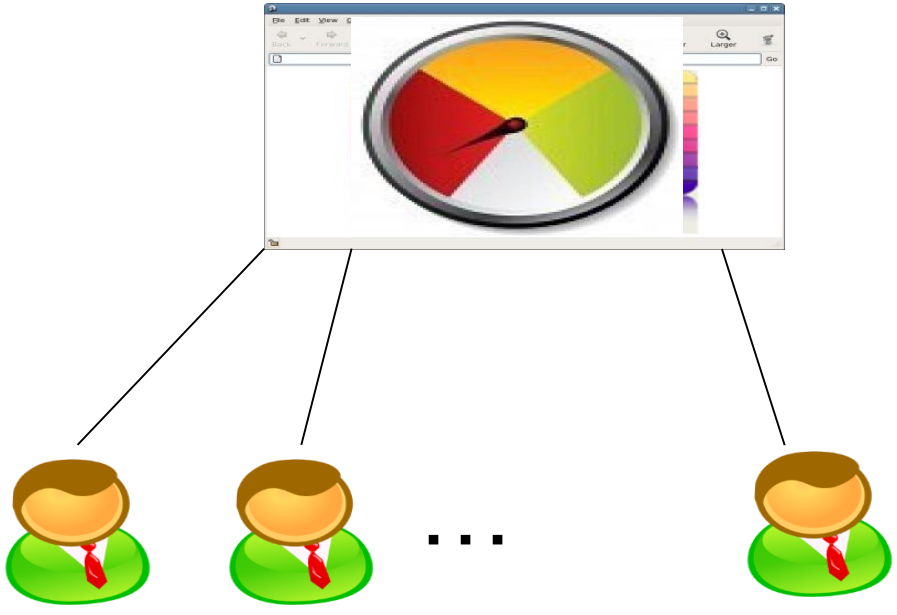
↖ There's a lot we don't know how to do yet… you can help!

Joe Has an Idea ($$$)

Solution #1

Buy then Maintain

- Big up-front commitment
- Load variability: NOT supported



Inside a Cloud Datacenter: Infrastructure as a Service

Q: So are we just **shifting the ownership problem**, that is, to the cloud owner?

The Pizza-Box Stack

- The 1U server

The Pizza-Box Stack

- The 1U server

The Pizza-Box Stack

- The 1U server

The Pizza-Box Stack

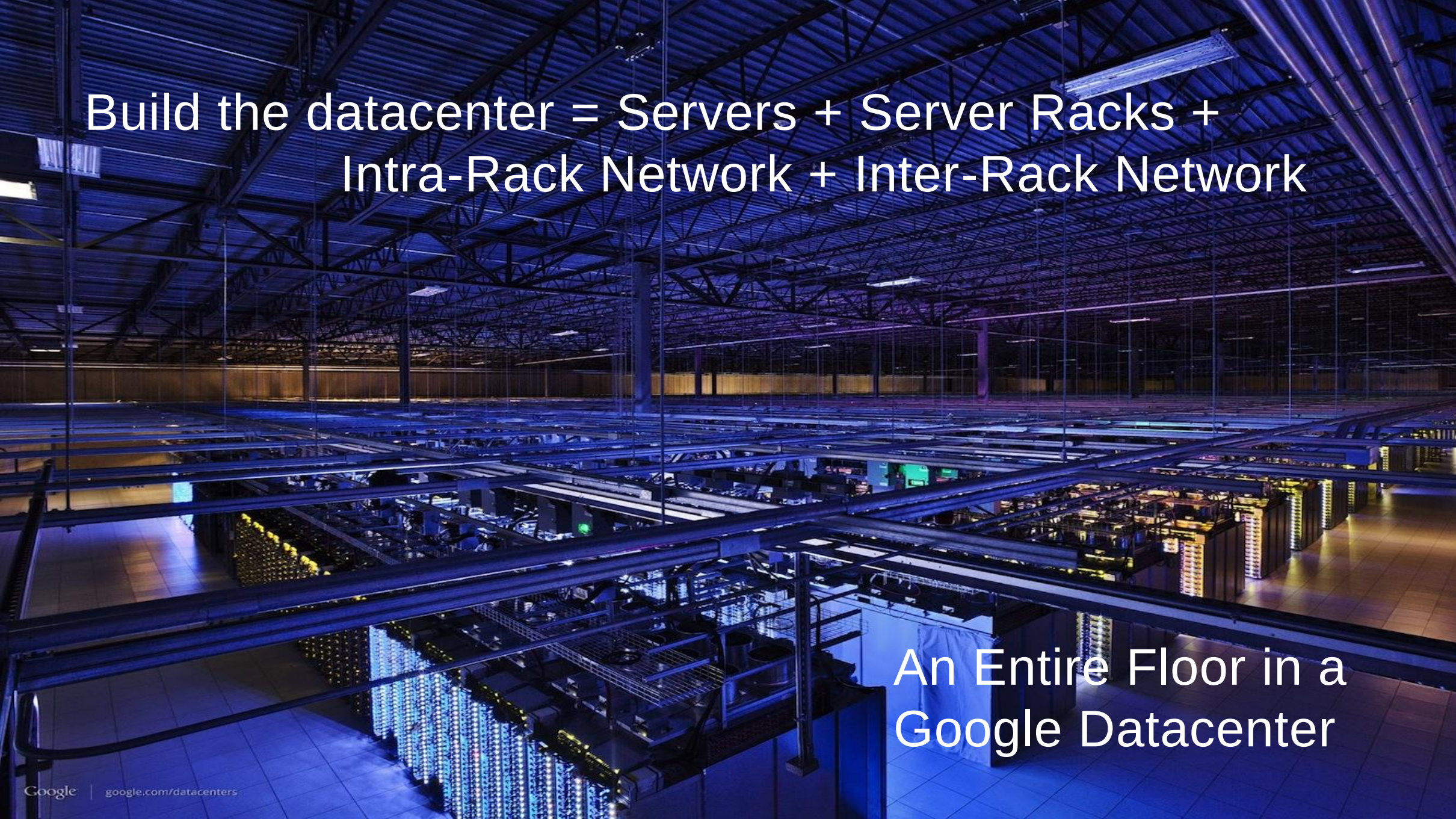
- The 1U server
- The 19” server rack (42U is now standard)

The Data Center Network

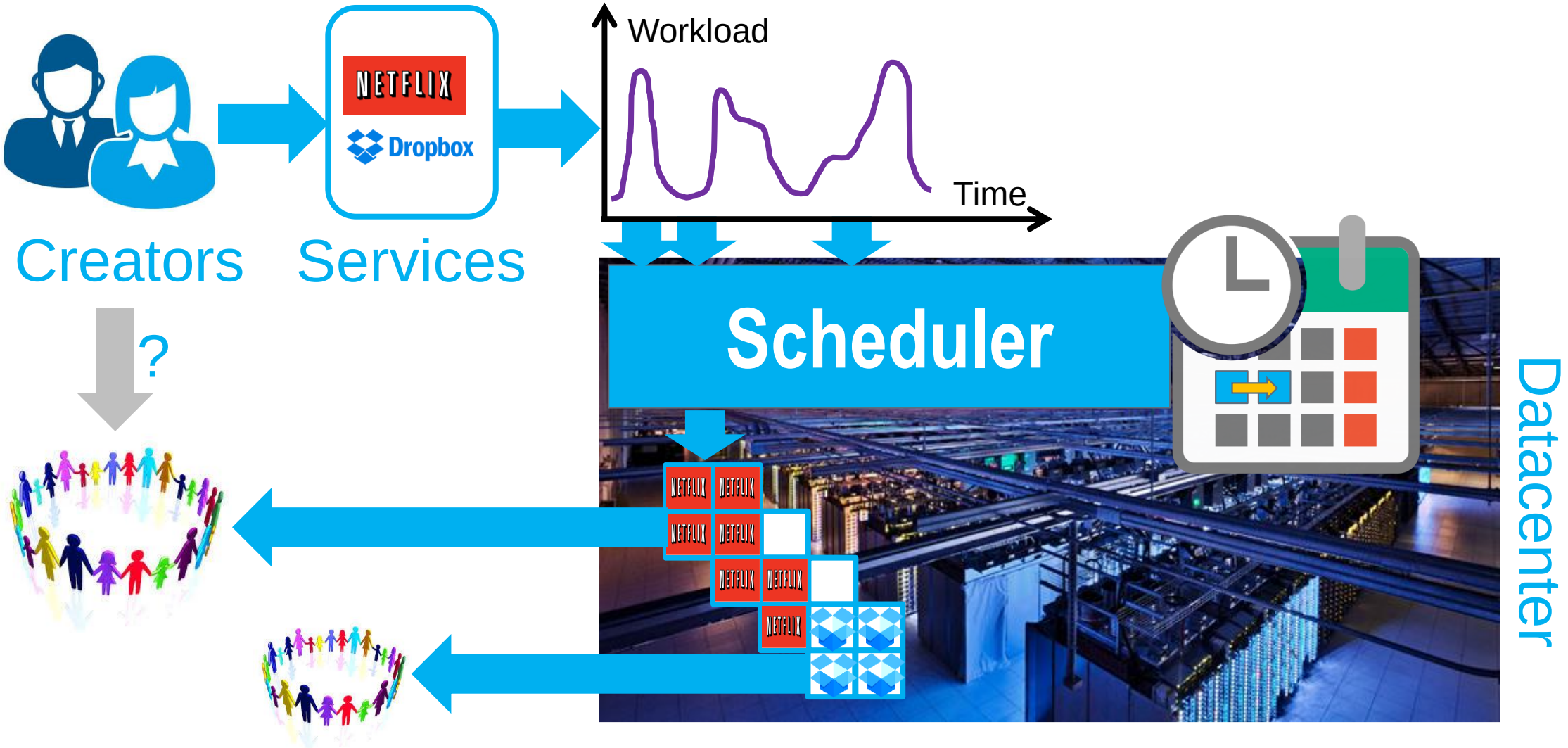
- Network bandwidth per rack
  - 1 x 48-port GigE switch = 40 UP-, 8 DOWN-links

- Network bandwidth per socket
  - (fast) 1 Gbps for 10 GigE rack switch
  - (slow) 100 Mbps for 1 GigE rack switch
  - (exorbitant) 10 GBps for ncHT3 (supercomputing class)
  - (research) 1 Tbps!~100 GBps optical (not yet production-ready)

Source: Dennis Abts (Google, Inc.) and John Kim (KAIST), High Performance Data Center Networks, 2011.


Build the datacenter = Servers + Server Racks +
Intra-Rack Network + Inter-Rack Network
An Entire Floor in a
Google Datacenter
Google | google.com/datacenters

The Scheduler

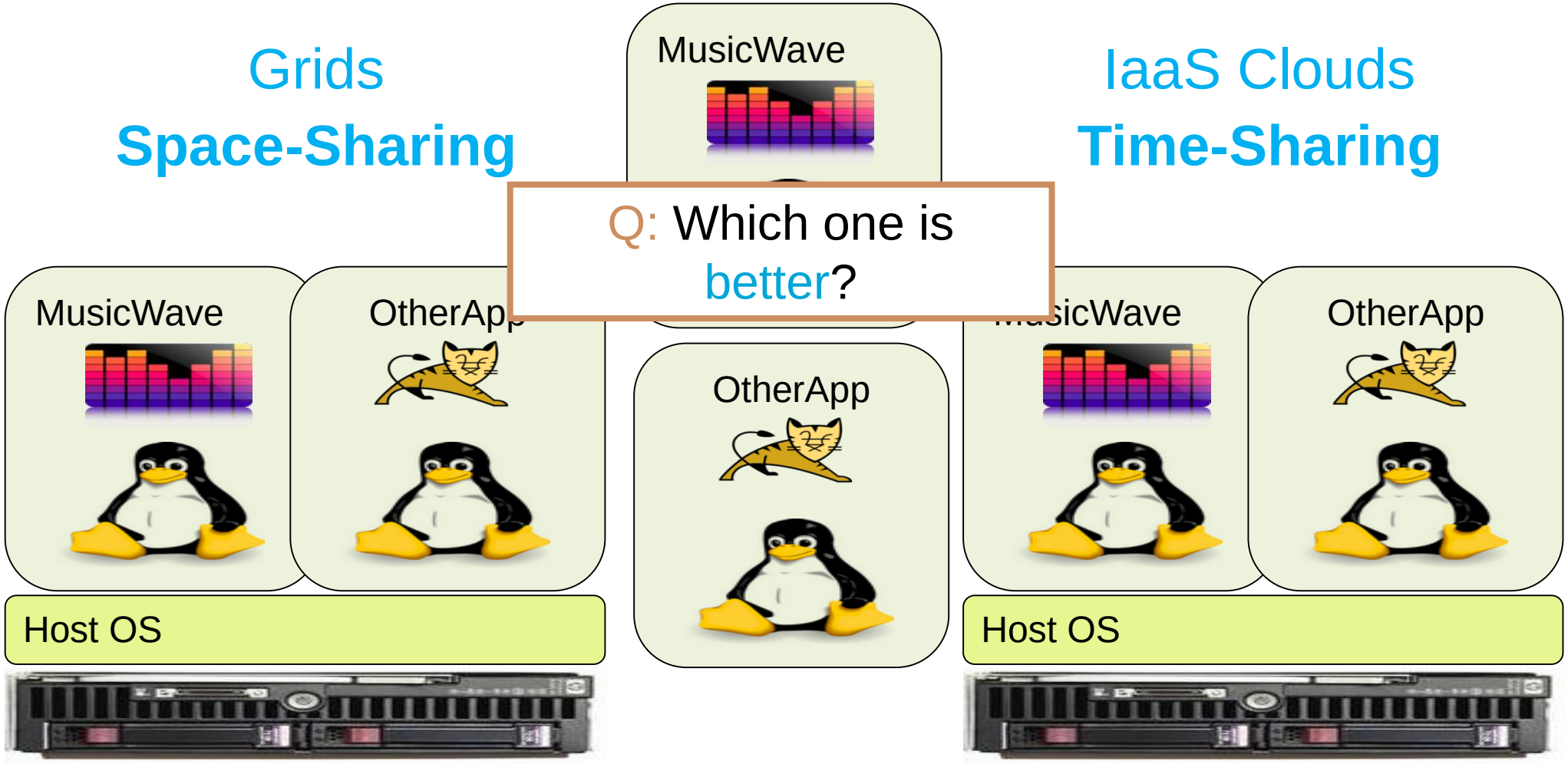


Resource Sharing Models

Grids
Space-Sharing

IaaS Clouds
Time-Sharing

Q: Which one is better?

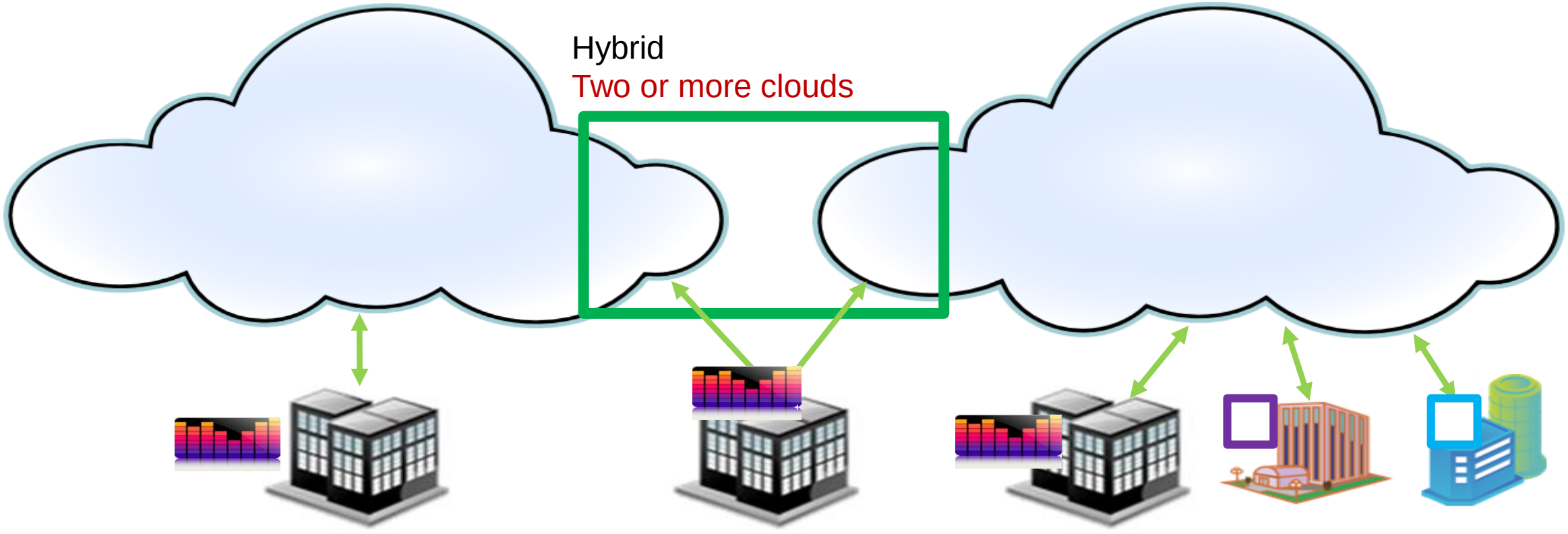


Where to Lease Resources From?
IaaS Cloud Deployment Models

Private
On-premises

Public
Off-premises

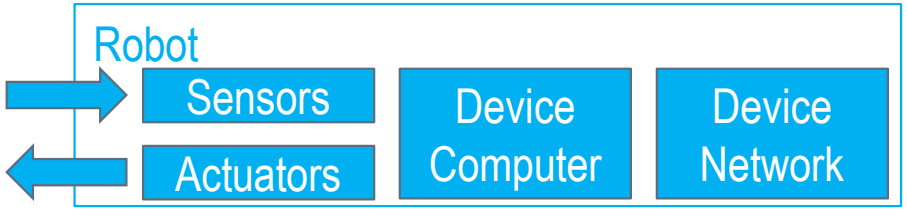
Hybrid
Two or more clouds



(Source: A. Antoniou, MSc Defense, TU Delft, 2012. Original idea: Mell and Grance, NIST Spec.Pub. 800-145, Sep 2011.)

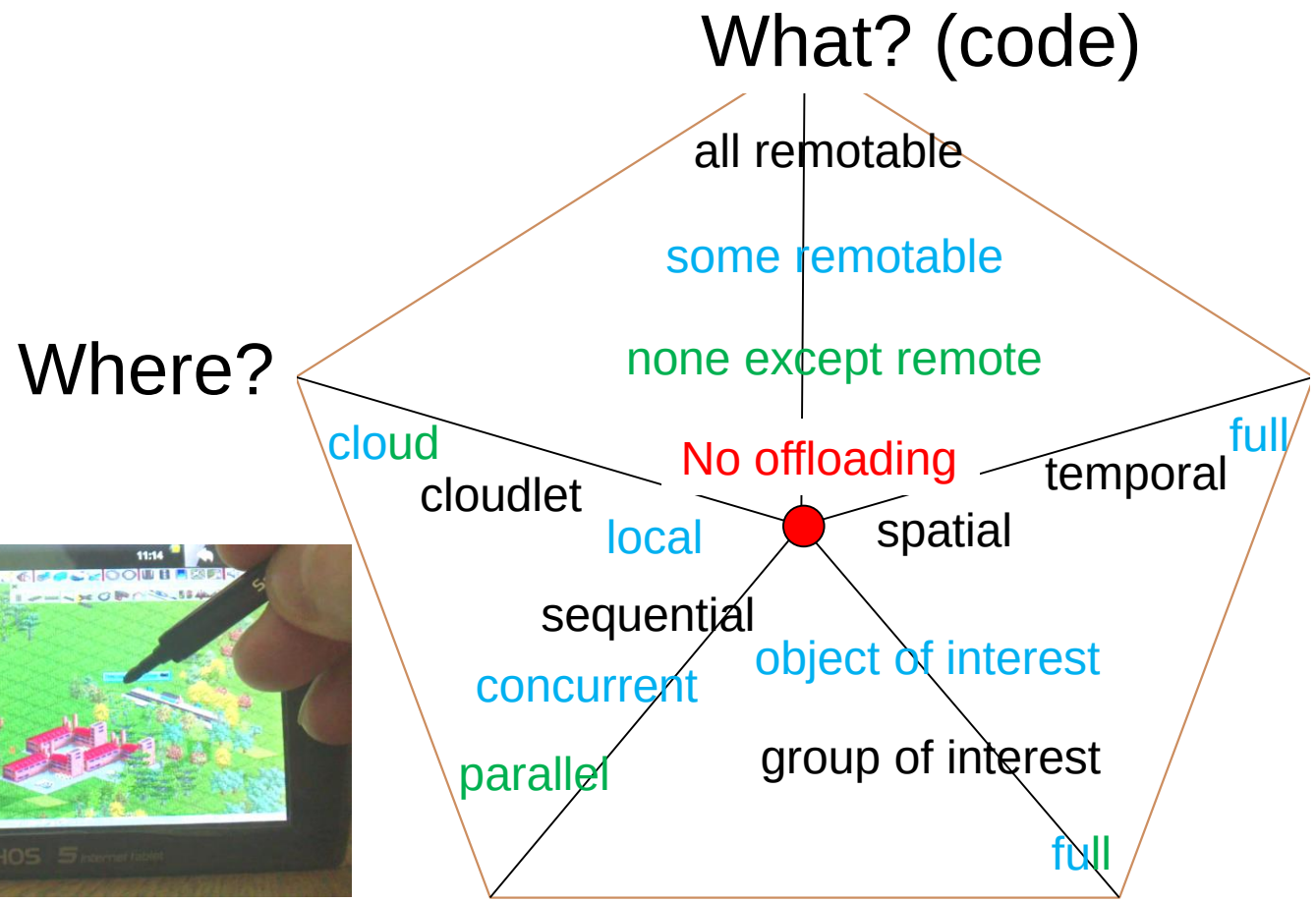
How to Offload to a Cloud? A Simple Recipe

1. Identify the main operational blocks



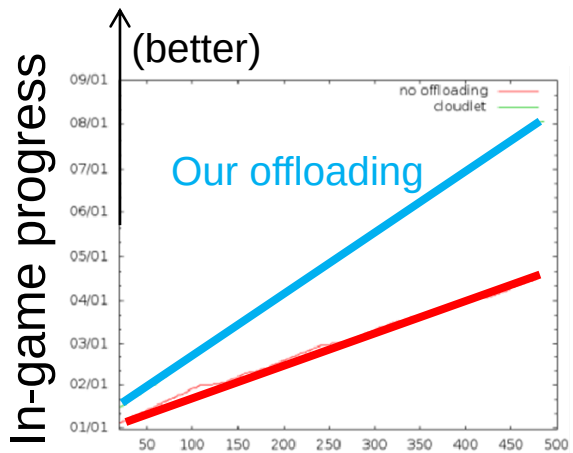
2. Decide which functional or non-functional aspect to optimize
3. Decide on strategy for offloading (What? When? How?)
4. Decide on architecture for distributed system (Where?)
5. Implement, test, and tune.

Exploratory Space for Offloading Strategy



Cloud gaming / SuperServer: OnLive, Gaikai, Sony Online, Microsoft Live, etc.

source: youtube.com



How?

What? (data)

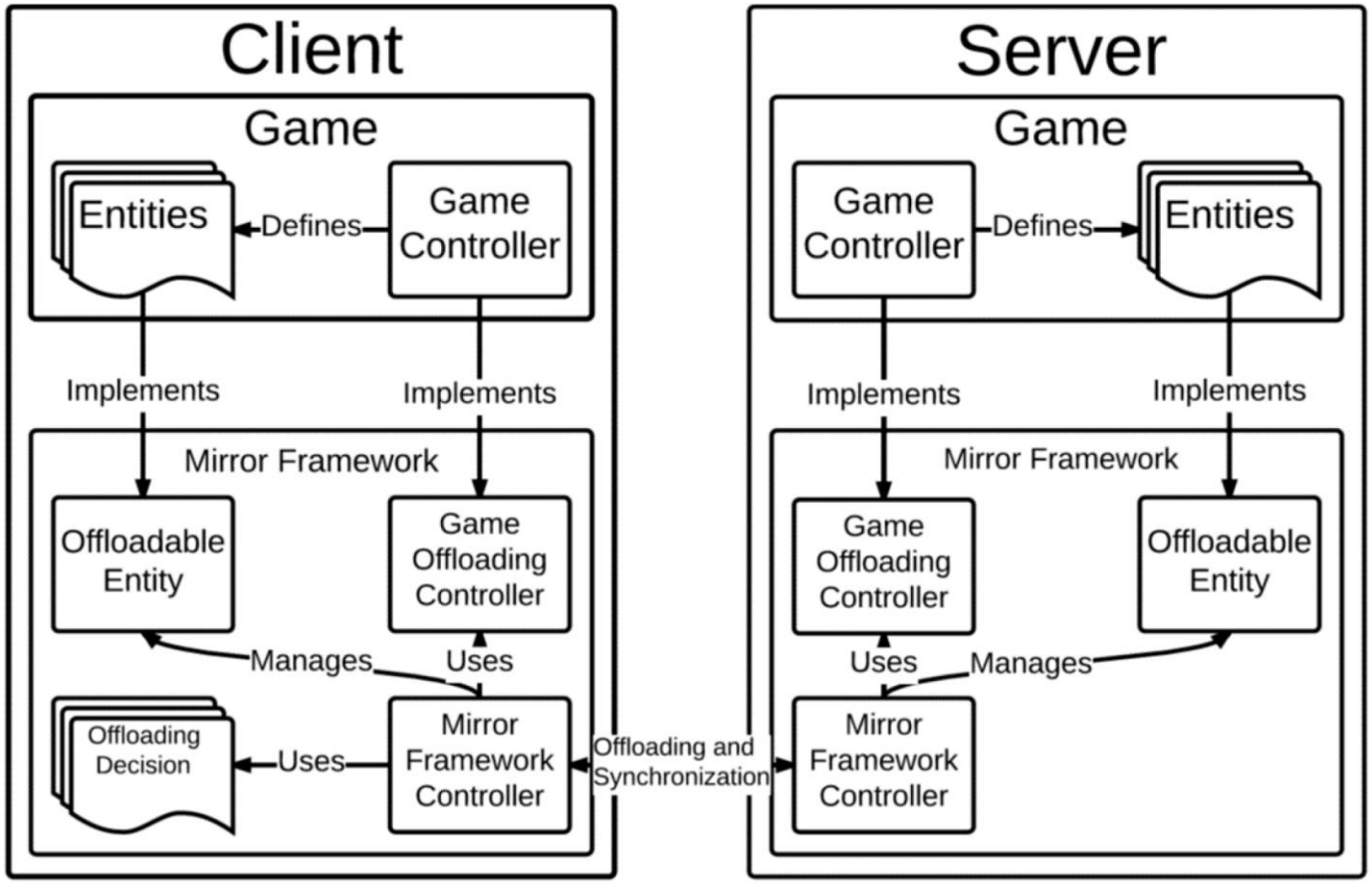
A.-C. Olteanu, N. Tapus, A. Iosup: Extending the Capabilities of Mobile Devices for Online Social Applications through Cloud Offloading. CCGRID 2013: 160-163.

VU VRIJE UNIVERSITEIT AMSTERDAM

TU Delft Delft University of Technology

The Mirror Framework (tried w online gaming “bots”)

- Key ideas:
  - Fine-grained replication
  - What? Objects
  - How? Concurrently
  - When? Temporal imbalance
  - Where? Cloud + Local



Minghai H. Jiang, Otto W. Visser, I.S. Wishnu B. Prasetya, Alexandru Iosup, Mirror: A Computation-offloading Framework for Sophisticated Mobile Games, WoWMom 2017. (**request digital copy**)

Further Reading

Elastic Big Data and Computing

- B. Ghit, N. Yigitbasi (Intel Research Labs, Portland), A. Iosup, and D. Epema. Balanced Resource Allocations Across Multiple Dynamic MapReduce Clusters. SIGMETRICS 2014
- L. Fei, B. Ghit, A. Iosup, D. H. J. Epema: KOALA-C: A task allocator for integrated multicluster and multicloud environments. CLUSTER 2014: 57-65
- K. Deng, J. Song, K. Ren, A. Iosup: Exploring portfolio scheduling for long-term execution of scientific workloads in IaaS clouds. SC 2013: 55

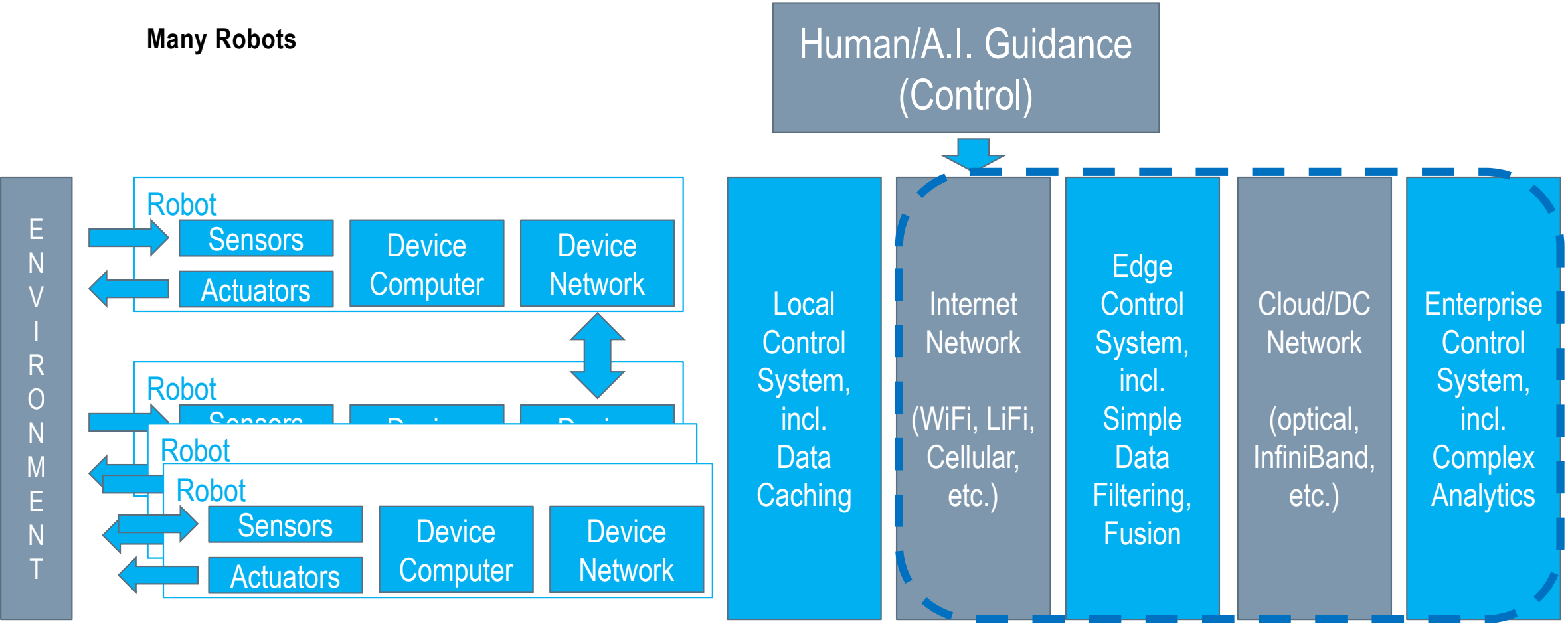
Time-Based Analytics

- B. Ghit, M. Capota, T. Hegeman, J. Hidders, D. Epema, and A. Iosup. V for Vicissitude: The Challenge of Scaling Complex Big Data Workflows. Winners IEEE Scale Challenge 2014

- http://www.pds.ewi.tudelft.nl/epema
- http://www.st.ewi.tudelft.nl/~iosup/research.html
- http://www.st.ewi.tudelft.nl/~iosup/research_cloud.html
- http://www.pds.ewi.tudelft.nl/

A Brief Introduction to Big Data Processing

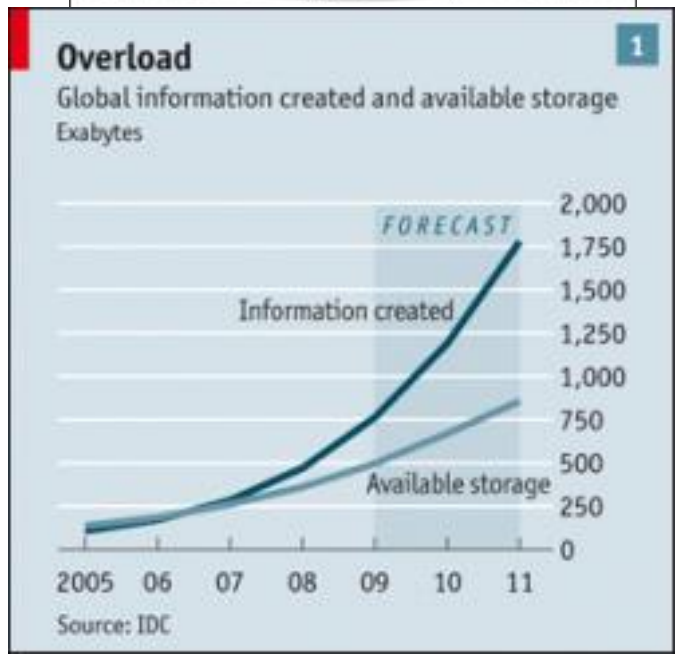
Many Robots



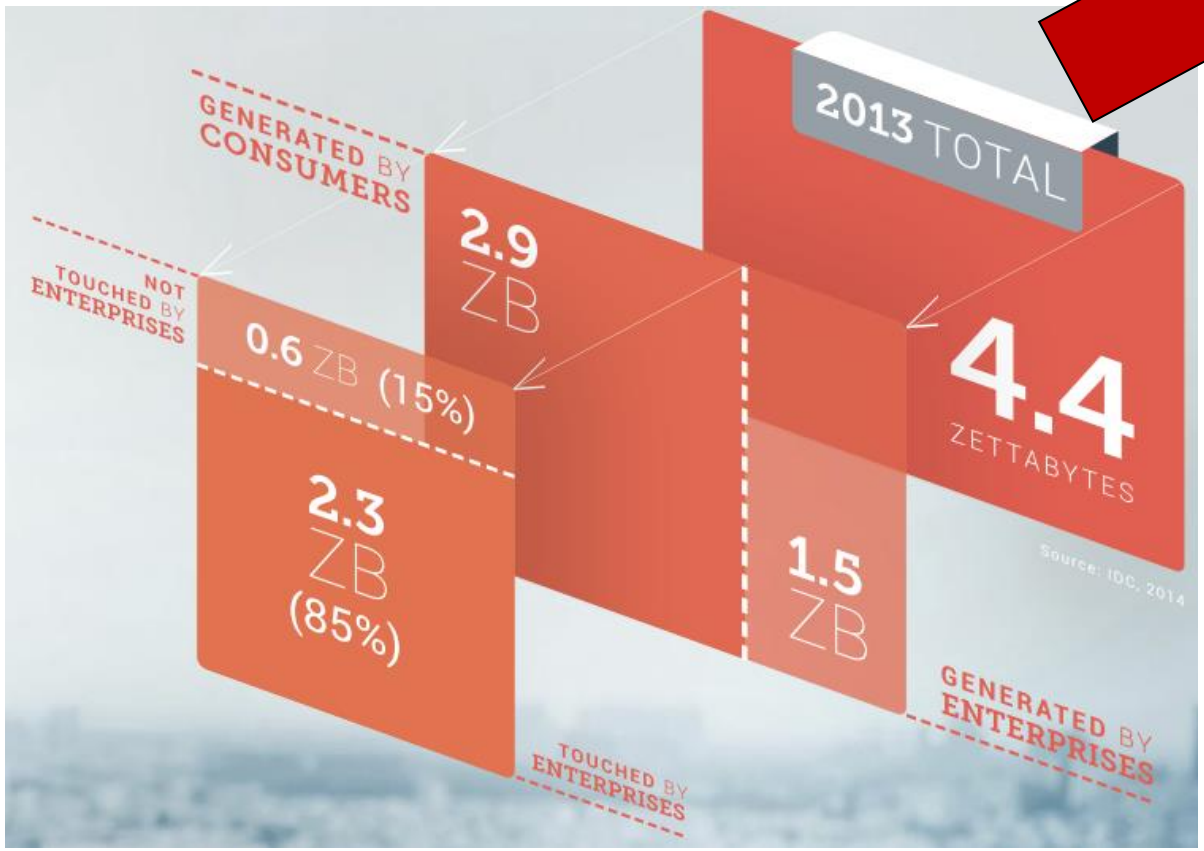
The “Data Deluge”: The Why

"Everywhere you look, the quantity of information in the world is soaring. According to one estimate, mankind created 150 exabytes (billion gigabytes) of data in 2005. This year, it will create 1,200 exabytes. Merely keeping up with this flood, and storing the bits that might be useful, is difficult enough. Analysing it, to spot patterns and extract useful information, is harder still.“

The Data Deluge, The Economist, 25 February 2010



“Data Deluge” In Summary



Data Deluge =
data generated by humans and devices (IoT)

- Interacting
- Understanding
- Deciding
- Creating

Sources: IDC, EMC 2014.

The Overwhelming Growth of Knowledge

"When 12 men founded the Royal Society in 1660, it was possible for an educated person to encompass all of scientific knowledge. […]

In the last 50 years, such has been the pace of scientific advance that even the best scientists cannot keep up with discoveries at frontiers outside their own field."

Tony Blair,
PM Speech, May 2002

Number of Publications	1993 1997	1997 2001
United States	1,248,733	1,265,808
EU15 (net total)	1,180,730	1,347,985
United Kingdom	309,683	342,535
Germany	268,393	318,286
Japan	289,751	336,858
France	203,814	232,058
Canada	168,331	166,216
Italy	122,398	147,023
Switzerland	57,664	66,761
Netherlands	83,600	92,526

Data: King,The scientific impact of nations,Nature'04.

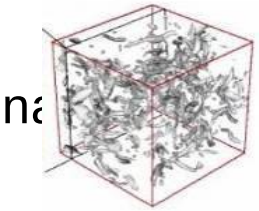
The Fourth Paradigm: The What

From Hypothesis to Data

1. Thousand years ago:
   science was **empirical** describing natural phenomena

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{4\pi G\rho}{3} - \mathrm{K}\frac{c^2}{a^2}$$

2. Last few hundred years:
   theoretical branch using models, generalizations



3. Last few decades:
   a **computational** branch simulating complex phenomena

4. Today (**the Fourth Paradigm**):
   data exploration

unify theory, experiment, and simulation

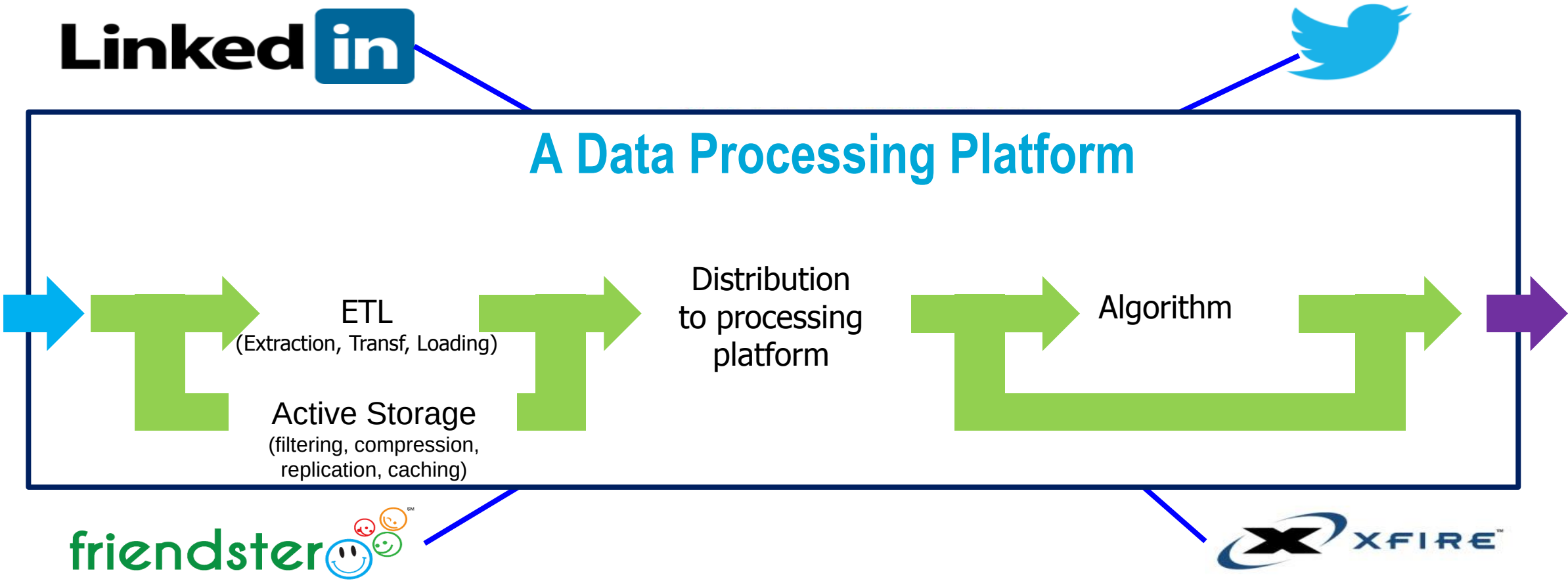
- Data captured by instruments or generated by simulator
- Processed by software
- Information/Knowledge stored in computer
- Scientist analyzes results using data management and statistics

Source: Jim Gray and "The Fourth Paradigm", http://research.microsoft.com/en-us/collaboration/fourthparadigm/

Data at the Core of Our Society: The LinkedIn Example

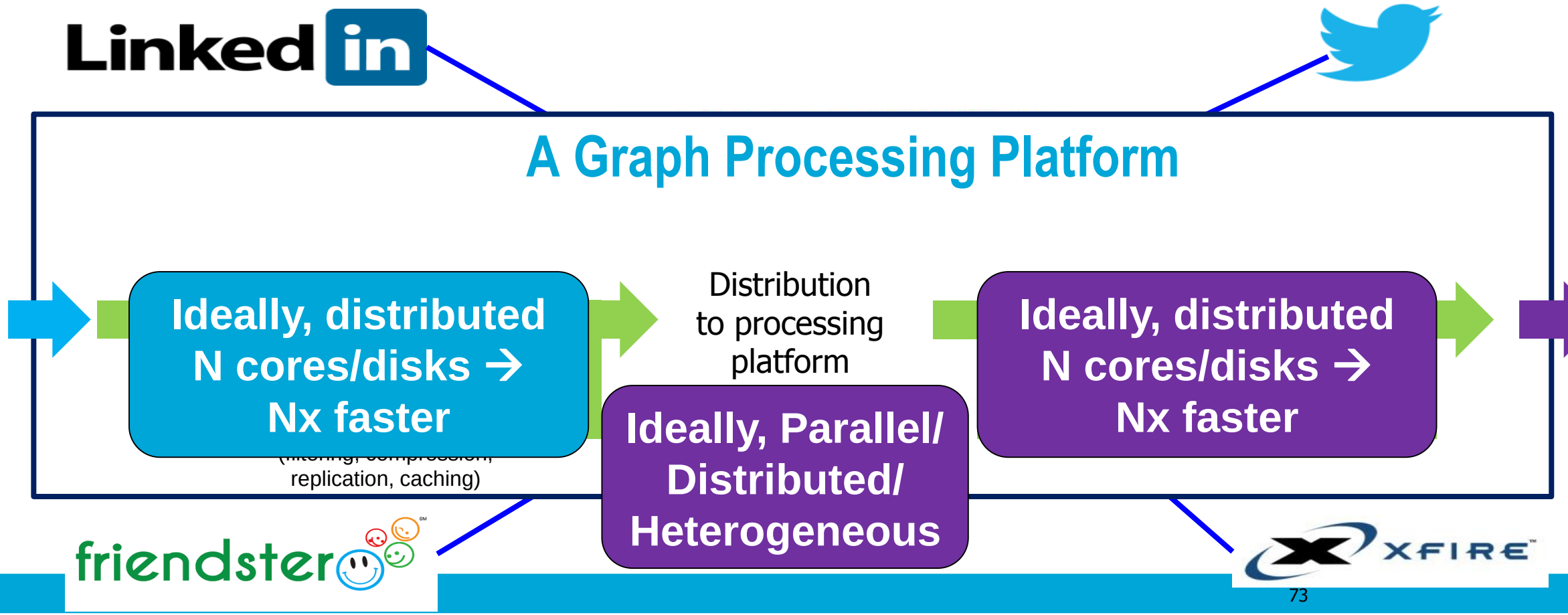
Sources: Vincenzo Cosenza, The State of LinkedIn, http://vincos.it/the-state-of-linkedin/
via Christopher Penn, http://www.shiftcomm.com/2014/02/state-linkedin-social-media-dark-horse/

How to do Data Analysis?



Interactive processing not considered in this presentation.
Streaming not considered in this presentation.

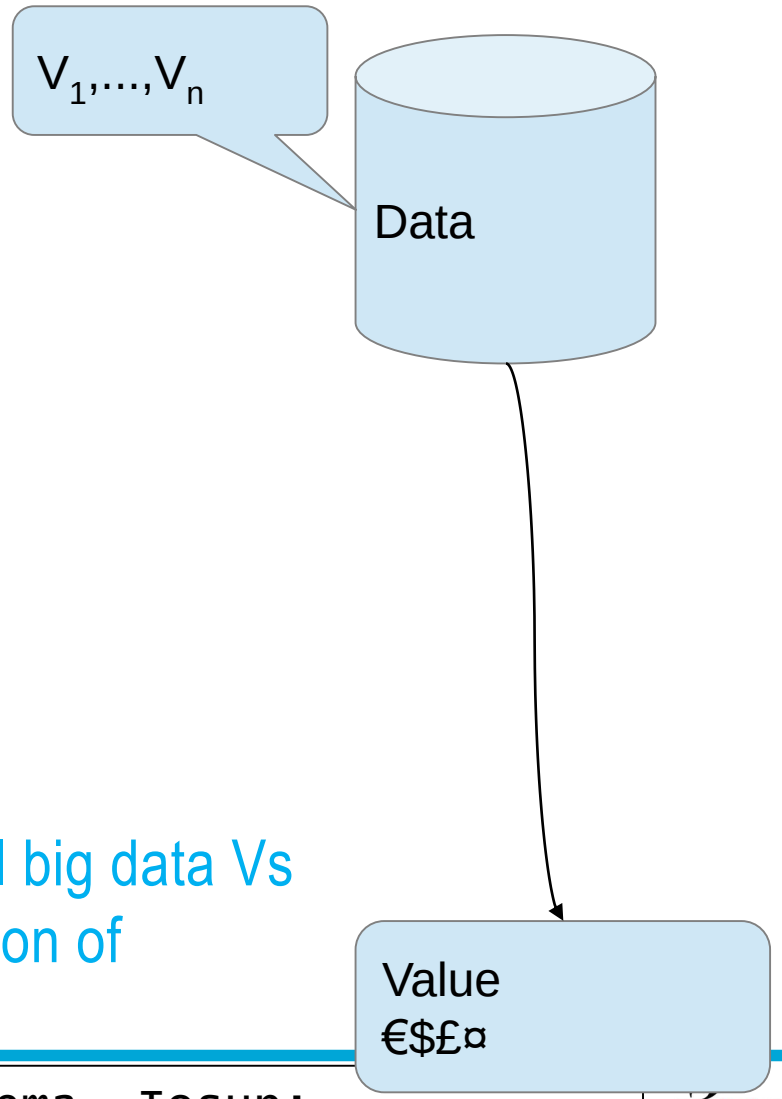
Distributed Big Data: The Promise of Performance



Interactive processing not considered in this presentation.
Streaming not considered in this presentation.

In Practice: big data Vs

- Volume – large scale of data
- Velocity – timeliness of data
- Variety – different forms of data
- Veracity – uncertainty of data
- Vicissitude – dynamic combination of several big data Vs in processing systems that support the addition of new queries at run-time

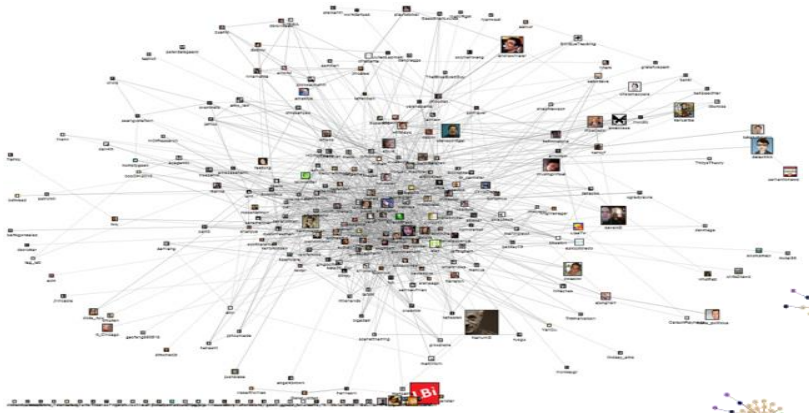


Ghit, Capota, Hegeman, Hidders, Epema, Iosup: V for Vicissitude: The Challenge of Scaling Complex Big Data Workflows. CCGRID 2014: 927-932

Data and Big Data Processing Systems Everywhere

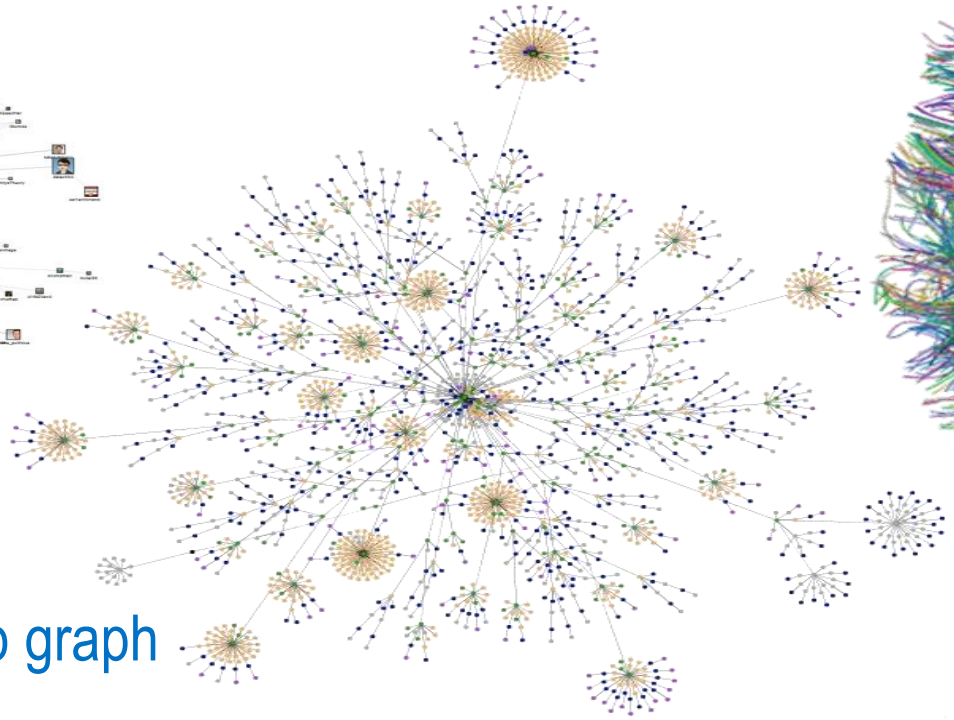
… Yet We Are in a Crisis Caused by Ecosystems

Example: processing large-scale graphs



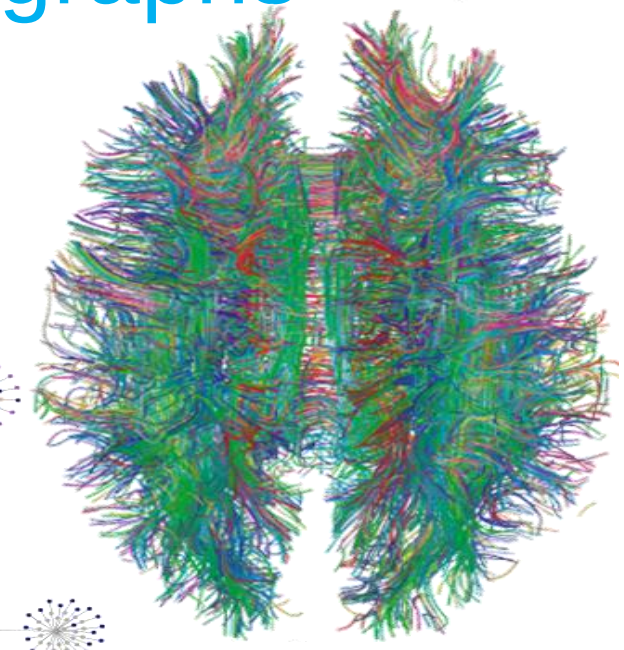
Social network

~1 billion vertices
~100 billion connections



Web graph

~50 billion pages
~1 trillion hyperlinks

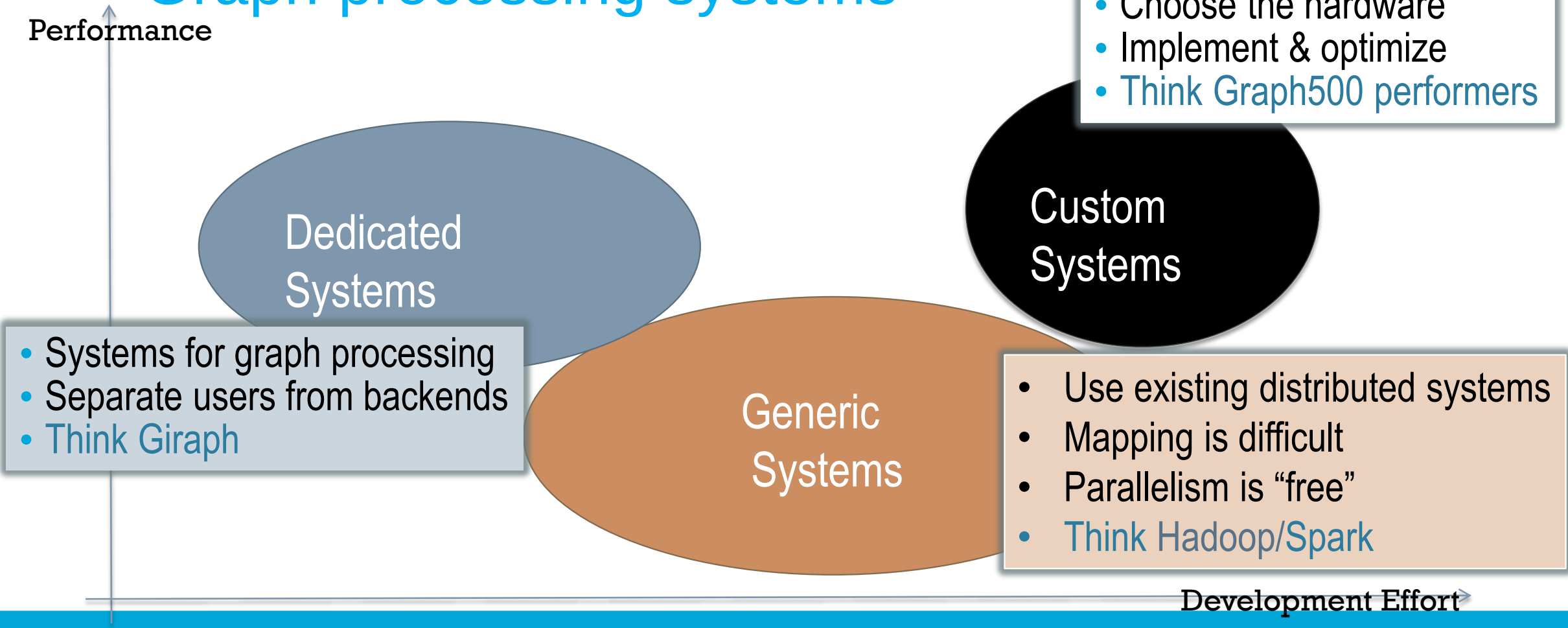


Brain network

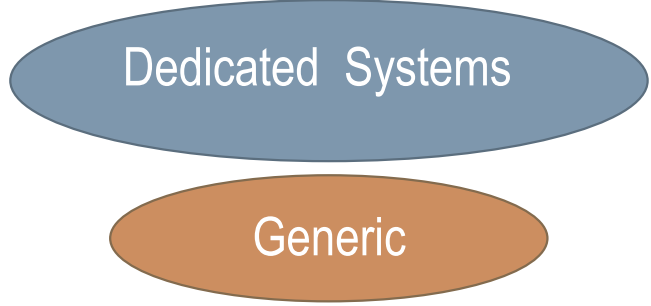
~100 billion neurons
~100 trillion connections

Source: Smith, CHI'10; Blog webpage; Gigandet et al., PLoS ONE 3(12)]

Graph processing systems



System diversity (CPU-based)



ORACLE PGX

Intel Graphmat

Neo4j
the graph database

Trinity

hadoop

YARN

StratoSphere
Above the Clouds

APACHE GIRAPH

GraphLab

GraphX

System diversity (GPU-enabled)

Dedicated Systems

Generic

medusa-gpu

Medusa: Simplified Graph Processing on GPUs

NetSysLab

mapgraph Beta

Massively Parallel Graph processing on GPUs

TOTEM

Gunrock

High-performance Graph Primitives on GPU

VertexAPI2

What is the performance of graph-processing platforms?

Metrics Diversity

Graph Diversity

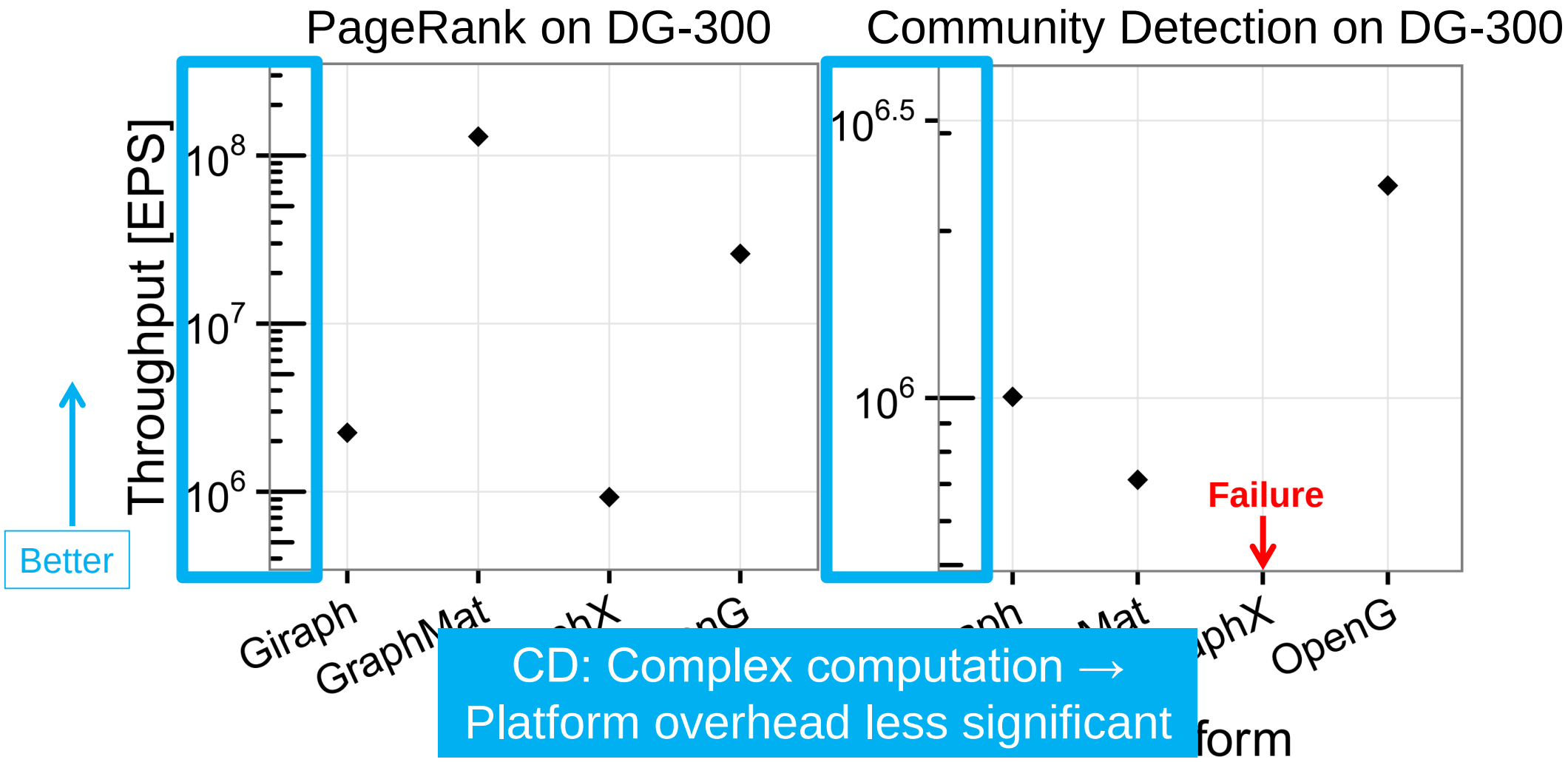
Algorithm Diversity

Graphalytics = comprehensive benchmarking suite for graph processing across many platforms

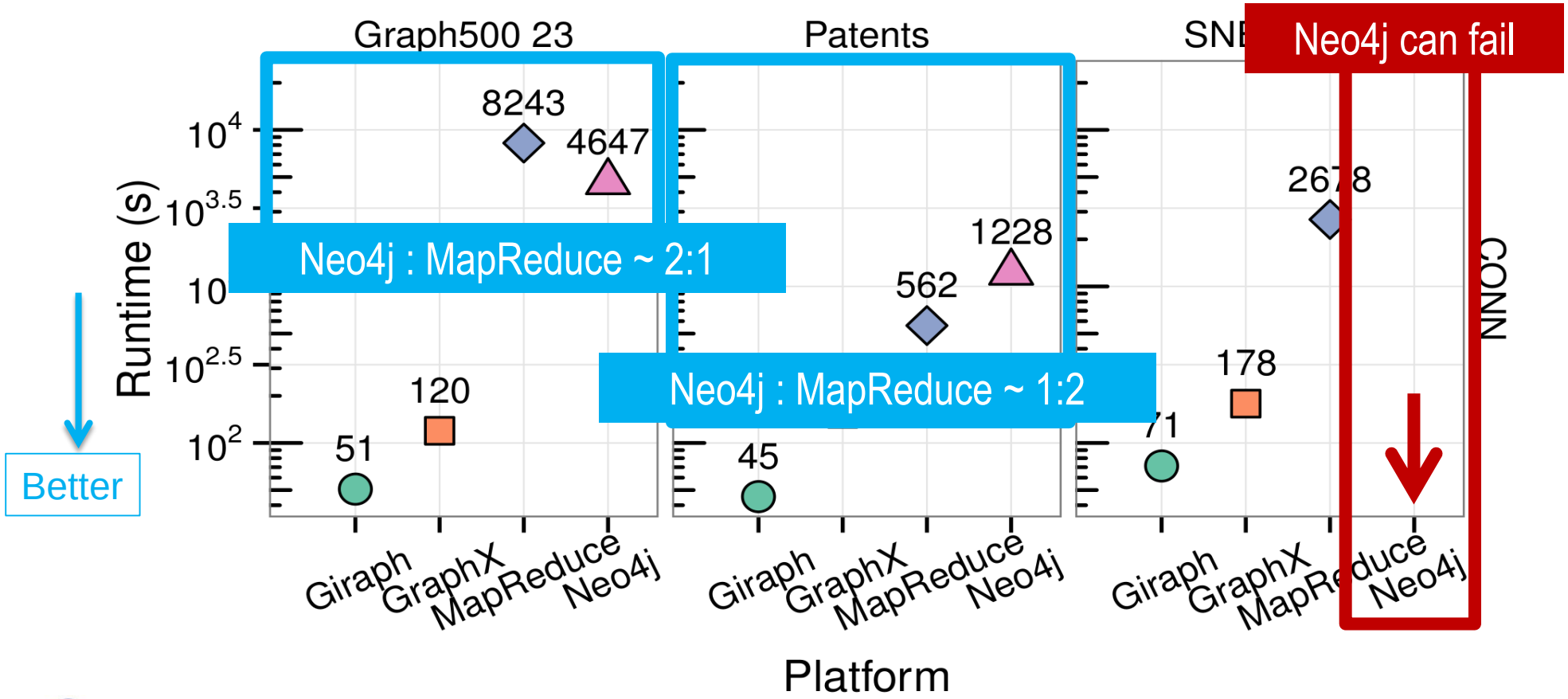
LDBC The graph & RDF benchmark reference

TU Delft Delft University of Technology

The Algorithm Has Large Impact

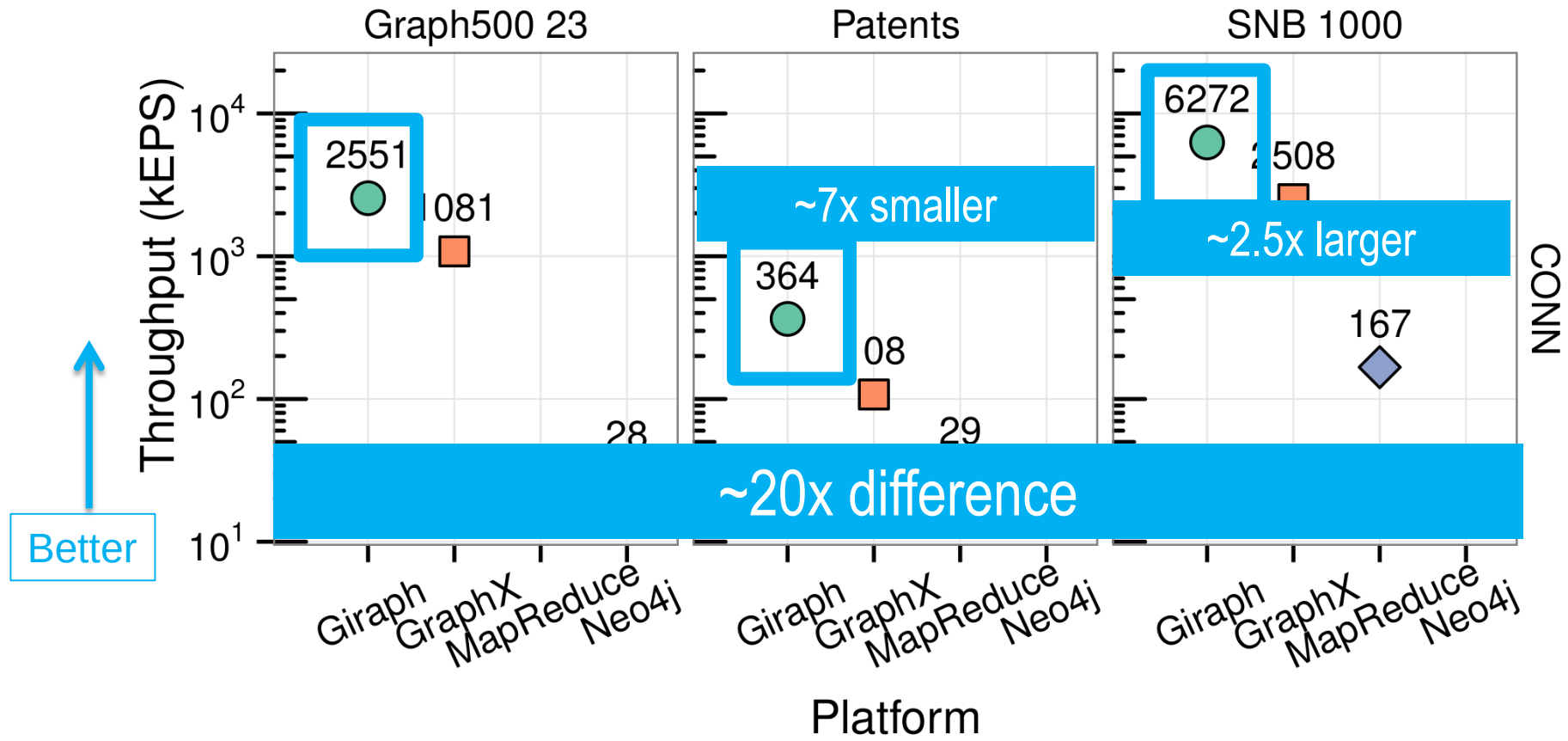


Runtime: The Dataset Has Large Impact



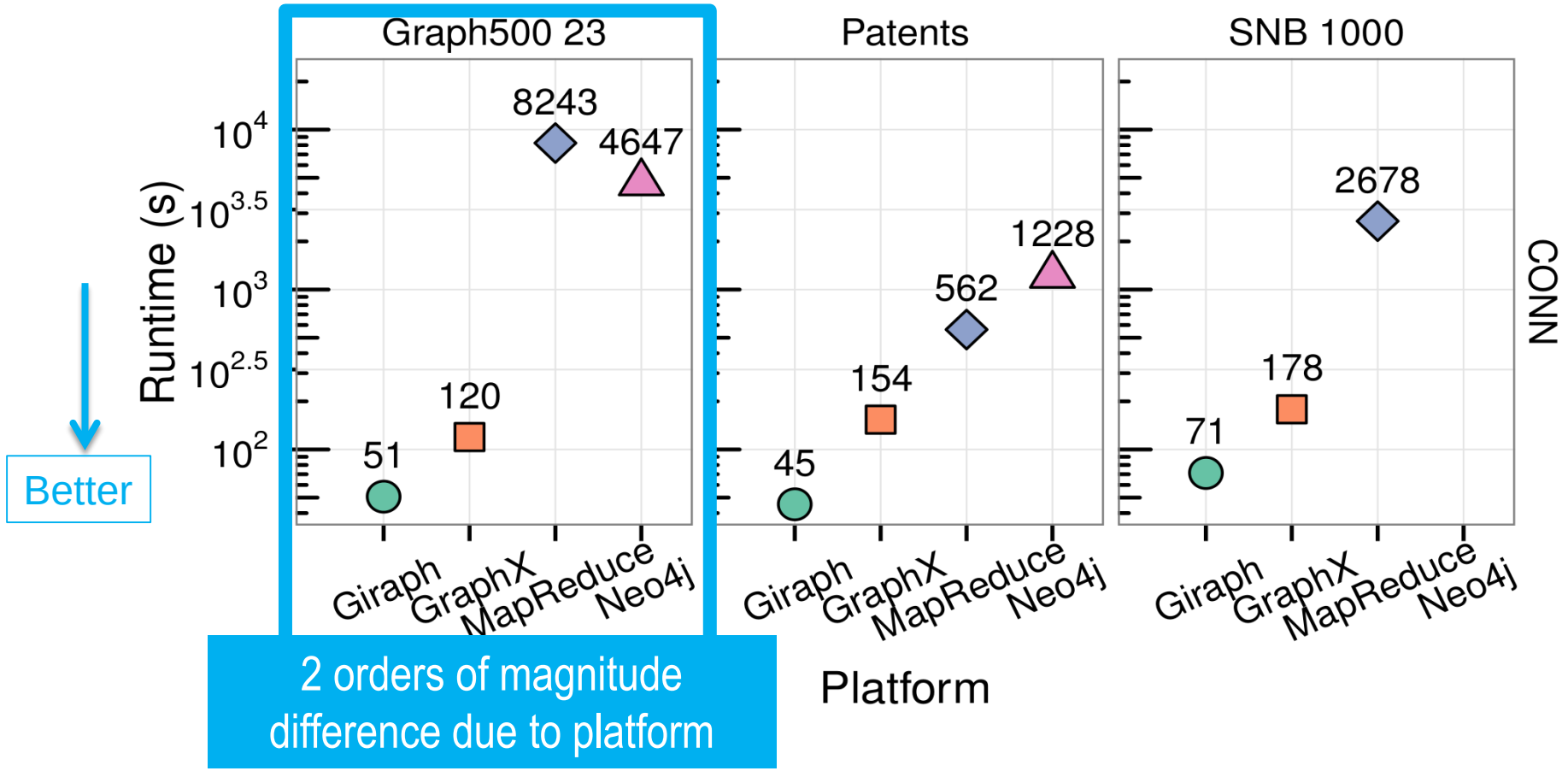
M. Capota et al., Graphalytics: A Big Data Benchmark for Graph-Processing Platforms. SIGMOD GRADES 2015

Throughput: The Dataset Has Large Impact



M. Capota et al., Graphalytics: A Big Data Benchmark for Graph-Processing Platforms. SIGMOD GRADES 2015

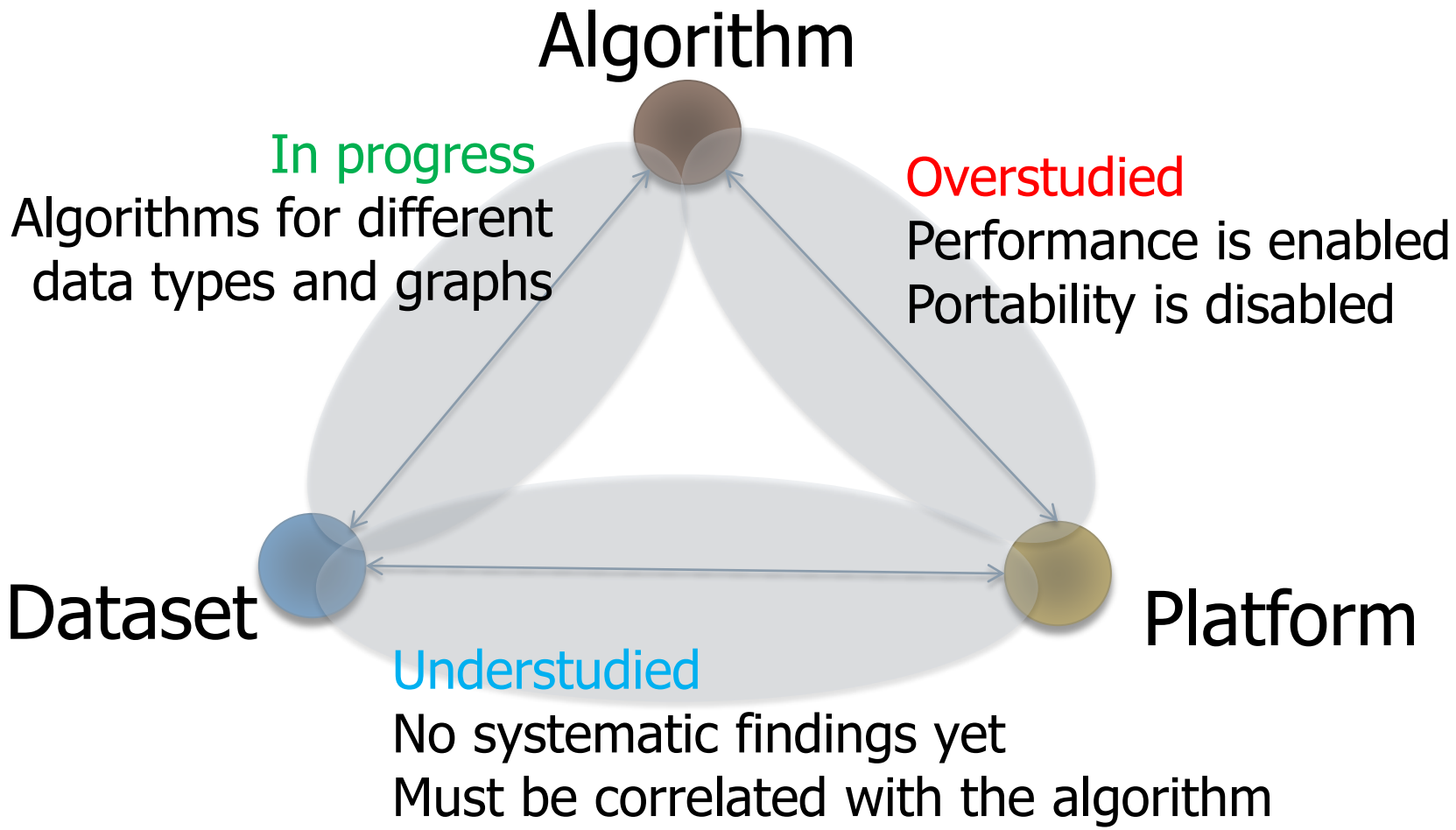
Runtime: The Platform Has Large Impact



M. Capota et al., Graphalytics: A Big Data Benchmark for Graph-Processing Platforms. SIGMOD GRADES 2015

Introduced by Ana Lucia Varbanescu.

The Platform-Algorithm-Dataset (PAD) Triangle for Performance Engineering of Graph-Processing Systems



Reading List

- Alexandru Iosup, Tim Hegeman, Wing Lung Ngai, Stijn Heldens, Arnau Prat Perez, Thomas Manhardt, Hassan Chaffi, Mihai Capotă, Narayanan Sundaram, Michael Anderson, Ilie Gabriel Tanase, Yinglong Xia, Lifeng Nai, Peter Boncz, *LDBC Graphalytics: A Benchmark forLarge-Scale Graph Analysis on Parallel andDistributed Platforms*, VLDB'16.
- Mihai Capotă, Tim Hegeman, Alexandru Iosup, Arnau Prat-Pérez, Orri Erling, and Peter Boncz, *Graphalytics: A Big Data Benchmark for Graph-Processing Platforms*, International Workshop on Graph Data Management Experiences and Systems (GRADES), 2015.
- Guo et al., *An Empirical Performance Evaluation of GPU-Enabled Graph-Processing Systems*. CCGRID'15.
- A. Iosup, A. L. Varbanescu, M. Capotă, T. Hegeman, Y. Guo, W. L. Ngai, and M. Verstraaten, *Towards Benchmarking IaaS and PaaS Clouds for Graph Analytics*, Workshop on Big Data Benchmarking (WBDB), 2014.
- Y. Guo, M. Biczak, A. L. Varbanescu, A. Iosup, C. Martella, and T. L. Willke, *How Well Do Graph-Processing Platforms Perform? An Empirical Performance Evaluation and Analysis*, IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2014, pp. 395–404.
- Y. Guo, A. L. Varbanescu, A. Iosup, C. Martella, and T. L. Willke, *Benchmarking graph-processing platforms*, ACM/SPEC International Conference on Performance Engineering (ICPE), 2014, pp. 289–292.

Take-Home Message + What to Do Next?

1. Robotics = distr.sys., clouds, big data
   - Consider the reference architecture
2. Distributed systems
   - CAP theorem: what does “2 of 3” mean?
   - Collaborative operations
3. Cloud computing
   - Datacenters and elasticity as core enablers
   - Cloud offloading is challenging
4. Big Data systems
   - Ecosystem navigation is challenging

Check the reading list(s)

How to make your robotics application cloud-aware?

How to make your robotics application big-data-aware?

Disclaimer: images used in this presentation obtained via Google Images.

- Images used in this lecture courtesy to many anonymous contributors to Google Images, and to Google Image Search.
- Many thanks!