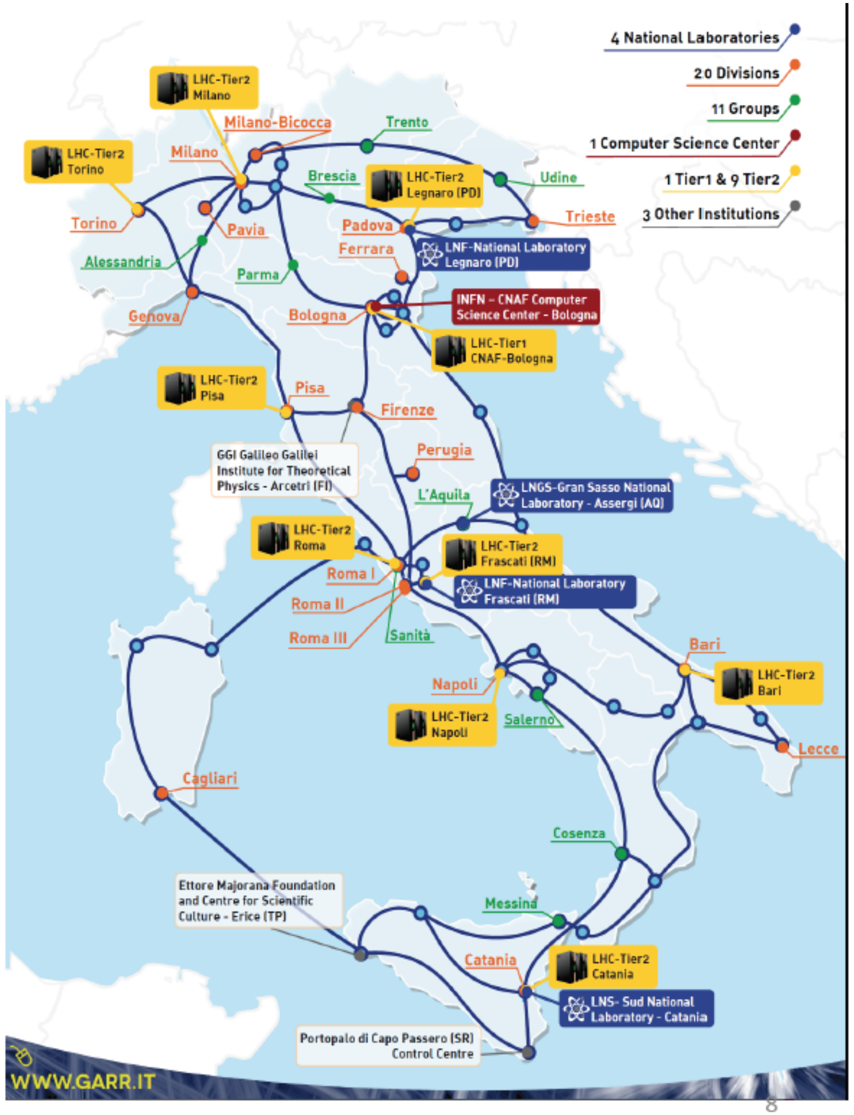
Introduction to Intelligent Infrastructures

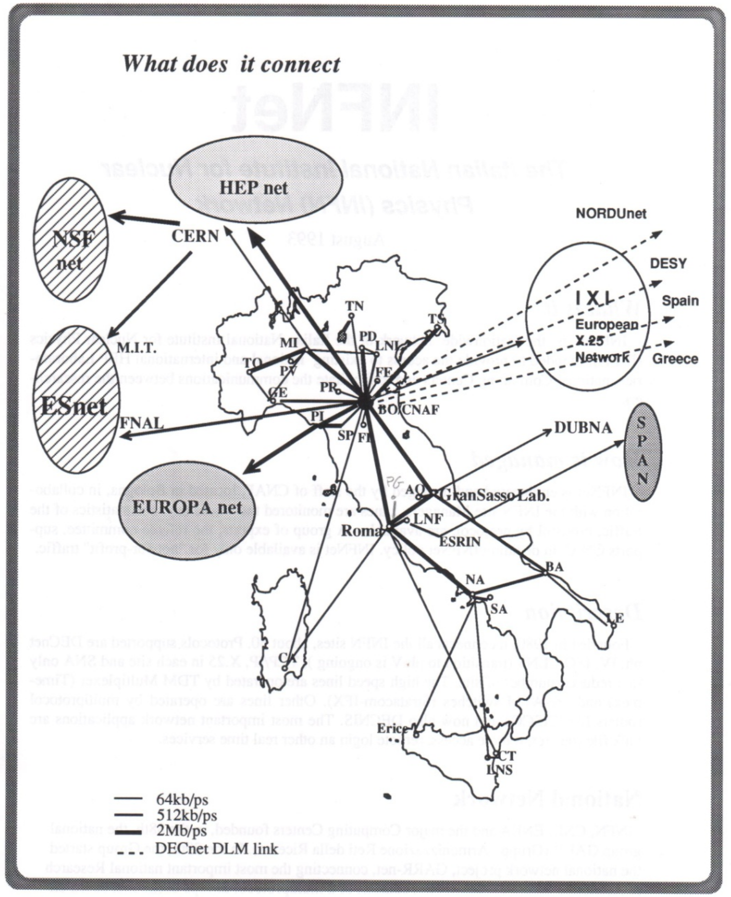
Davide Salomoni

davide@infn.it

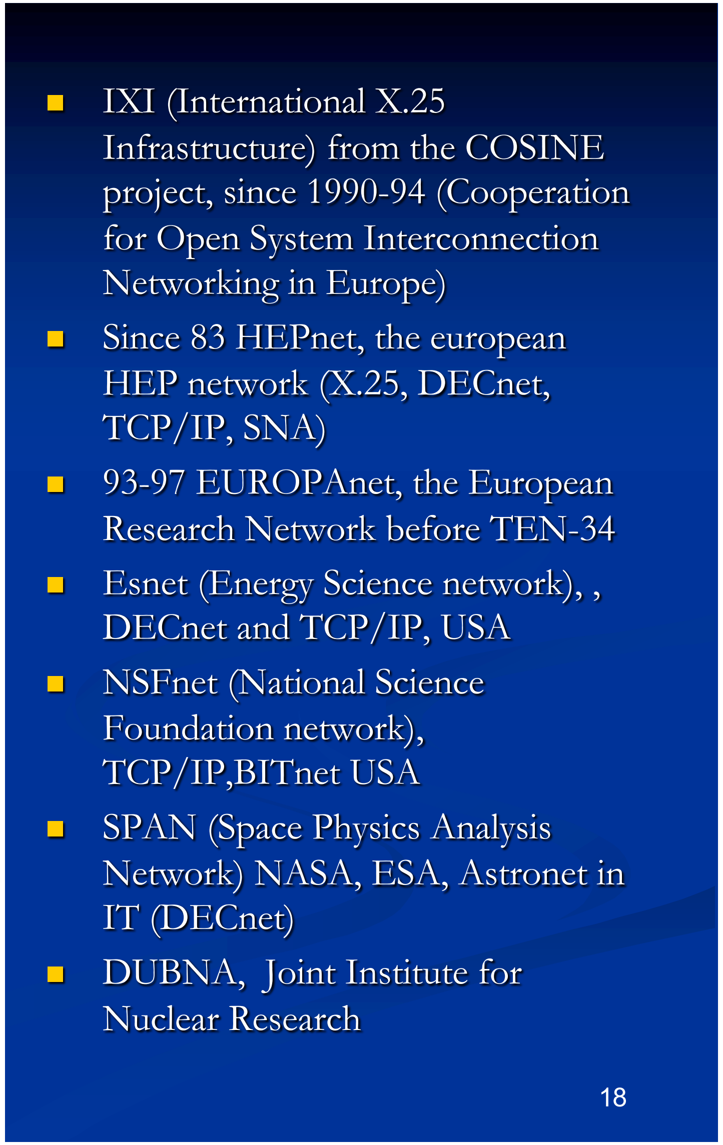
Infrastructures? You have already seen this…



… which in 1993 was this…



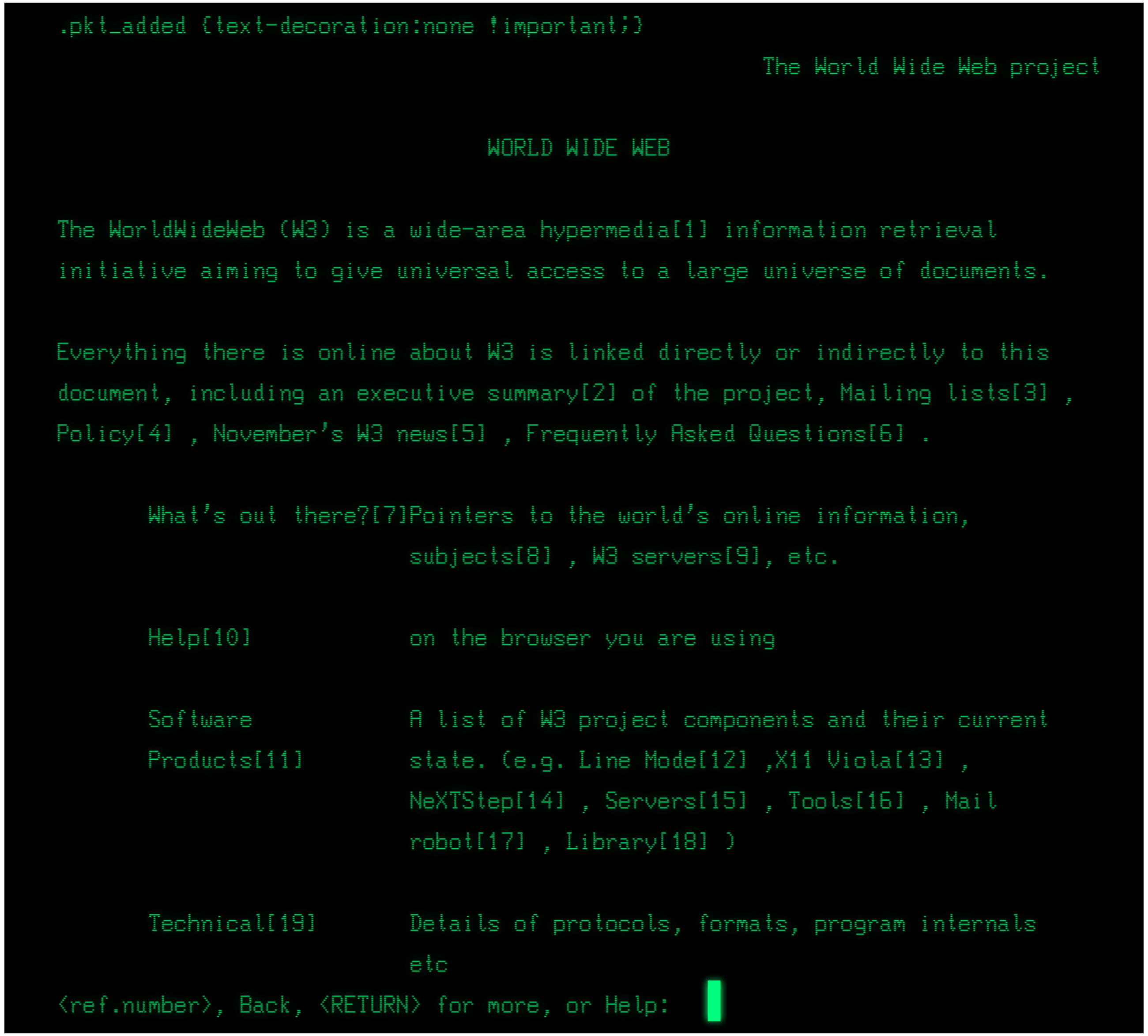
INFNet/GARR 1993



- IXI (International X.25 Infrastructure) from the COSINE project, since 1990-94 (Cooperation for Open System Interconnection Networking in Europe)
- Since 83 HEPnet, the european HEP network (X.25, DECnet, TCP/IP, SNA)
- 93-97 EUROPAnet, the European Research Network before TEN-34
- Esnet (Energy Science network), , DECnet and TCP/IP, USA
- NSFnet (National Science Foundation network), TCP/IP,BITnet USA
- SPAN (Space Physics Analysis Network) NASA, ESA, Astronet in IT (DECnet)
- DUBNA, Joint Institute for Nuclear Research

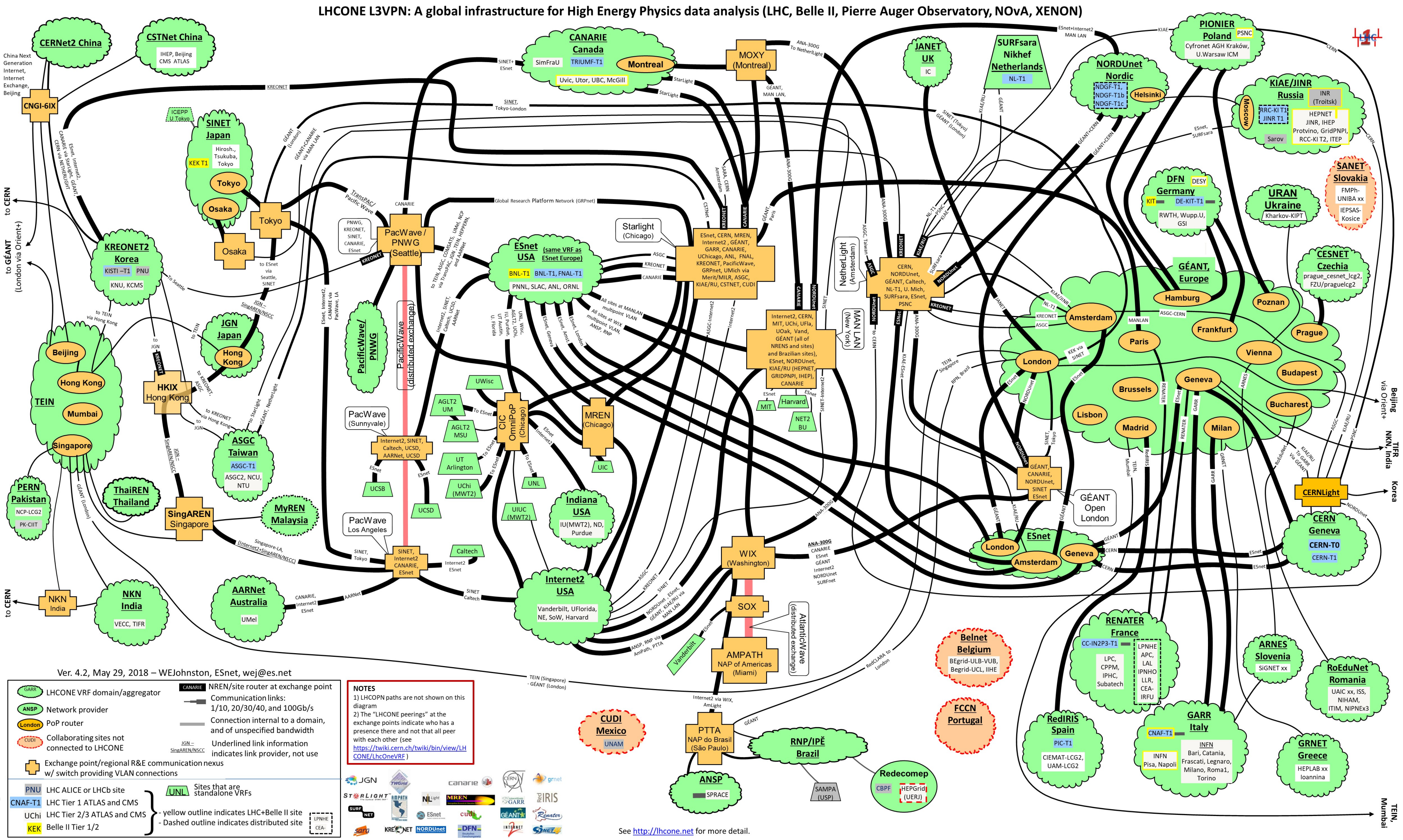
18

… at the time accessed with this:



```
.pkt_added {text-decoration:none !important;}
                                                      The World Wide Web project

                                 WORLD WIDE WEB

The WorldWideWeb (W3) is a wide-area hypermedia[1] information retrieval
initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this
document, including an executive summary[2] of the project, Mailing lists[3] ,
Policy[4] , November's W3 news[5] , Frequently Asked Questions[6] .

       What's out there?[7]Pointers to the world's online information,
                           subjects[8] , W3 servers[9], etc.

       Help[10]            on the browser you are using

       Software            A list of W3 project components and their current
       Products[11]        state. (e.g. Line Mode[12] ,X11 Viola[13] ,
                           NeXTStep[14] , Servers[15] , Tools[16] , Mail
                           robot[17] , Library[18] )

       Technical[19]       Details of protocols, formats, program internals
                           etc
<ref.number>, Back, <RETURN> for more, or Help:
```

Today things are more similar to this (this is just *a part* of a network layer):



LHCONE L3VPN: A global infrastructure for High Energy Physics data analysis (LHC, Belle II, Pierre Auger Observatory, NOvA, XENON)

Ver. 4.2, May 29, 2018 – WEJohnston, ESnet, wej@es.net

GARR	LHCONE VRF domain/aggregator
ANSP	Network provider
London	PoP router
CUDI	Collaborating sites not connected to LHCONE
✚	Exchange point/regional R&E communication nexus w/ switch providing VLAN connections
CANARIE	NREN/site router at exchange point
▬	Communication links: 1/10, 20/30/40, and 100Gb/s
▬	Connection internal to a domain, and of unspecified bandwidth
JGN – SingAREN/NSCC	Underlined link information indicates link provider, not use
PNU	LHC ALICE or LHCb site
CNAF-T1	LHC Tier 1 ATLAS and CMS
UChi	LHC Tier 2/3 ATLAS and CMS
KEK	Belle II Tier 1/2
UNL	Sites that are standalone VRFs

- yellow outline indicates LHC+Belle II site
- Dashed outline indicates distributed site (LPNHE CEA-)

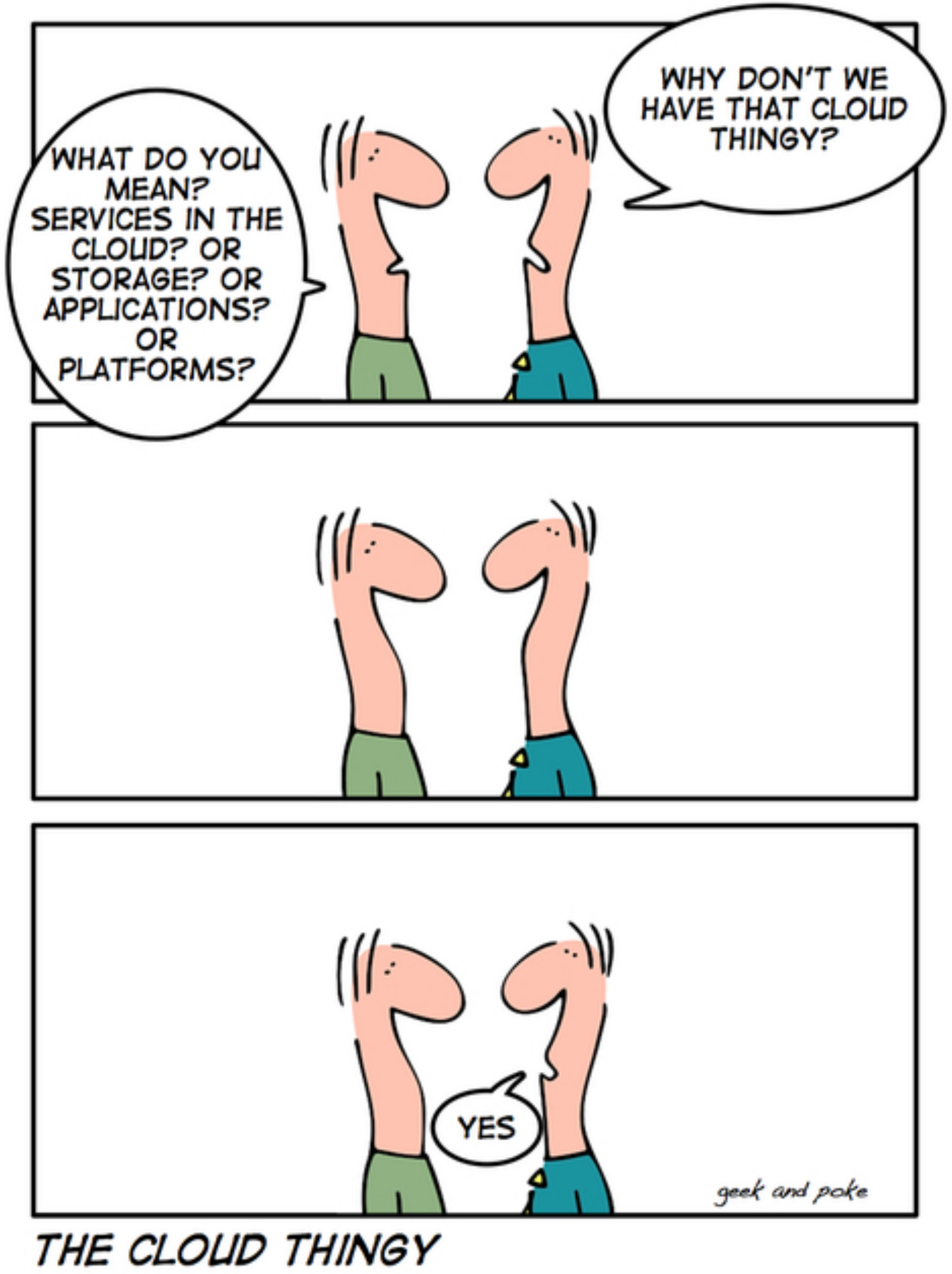
NOTES
1) LHCOPN paths are not shown on this diagram
2) The "LHCONE peerings" at the exchange points indicate who has a presence there and not that all peer with each other (see https://twiki.cern.ch/twiki/bin/view/LHCONE/LhcOneVRF)

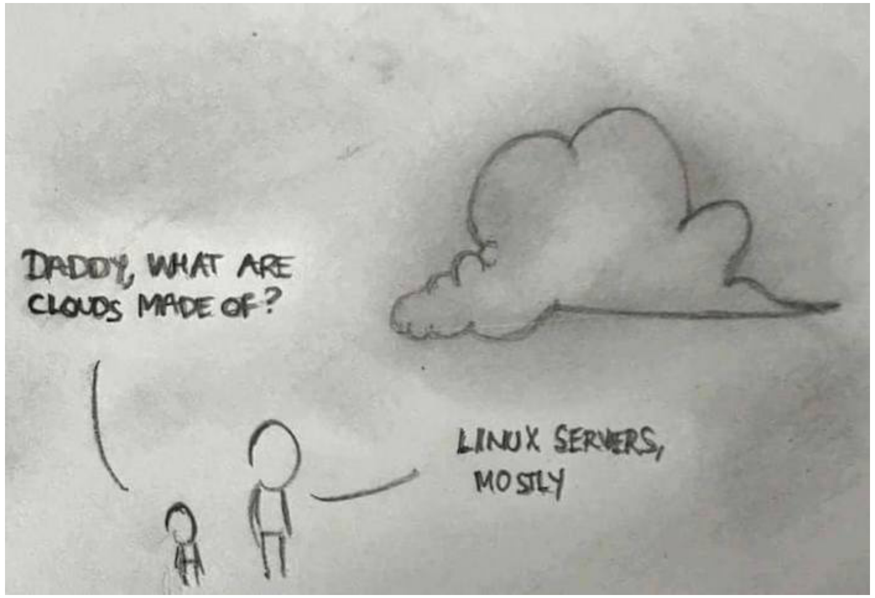
See http://lhcone.net for more detail.

What are *[e-]infrastructures* then?

- They are **virtual places** where a broad spectrum of resources for advanced data-driven research can be *found*, *accessed* and *used* by those who need them.
- We'll later give some meaning to the term *intelligent* as in "*intelligent infrastructures*".
- In modern terms, we often think of infrastructures as part of **the Cloud**, or of *Cloud Computing*.
- But *what is the Cloud*, really?

The «Cloud»


WHY DON'T WE HAVE THAT CLOUD THINGY?
WHAT DO YOU MEAN? SERVICES IN THE CLOUD? OR STORAGE? OR APPLICATIONS? OR PLATFORMS?
YES
geek and poke
THE CLOUD THINGY


DADDY, WHAT ARE CLOUDS MADE OF?
LINUX SERVERS, MOSTLY

There is no cloud
it's just someone else's computer

Cloud Computing, defined

From Wikipedia:

- “Cloud computing is an information technology (IT) paradigm that enables:
  - **ubiquitous access**
  - **to shared pools**
  - **of configurable system resources**
  - **and higher-level services** that can be
  - **rapidly provisioned with minimal management effort**, often
  - **over the Internet.**
- Cloud computing relies on the sharing of resources to achieve **coherence and economies of scale, similar to a public utility.**”

kWh
AC WATTHOUR METER
amazon web services

What is Cloud Computing?

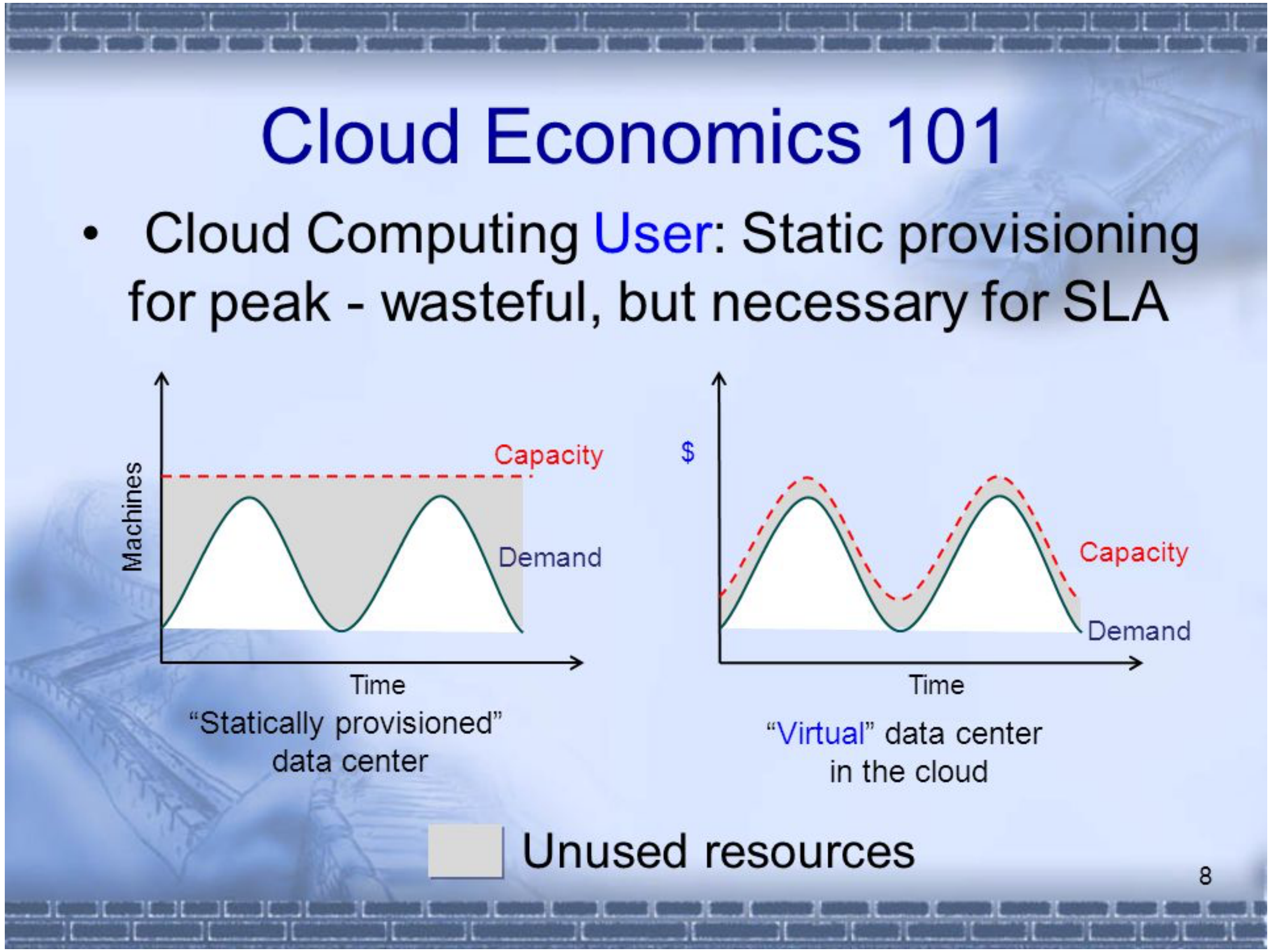
An analogy: think of electricity services…

You simply plug into a vast electrical grid managed by experts to get a low cost, reliable power supply – available to you with much greater efficiency than you could generate on your own.

Power is a utility service - available to you on-demand and you pay only for what you use.

Source: https://www.slideshare.net/AmazonWebServices/introduction-to-amazon-web-services-7708257

Why is Cloud computing useful in general?



Cloud Economics 101

- Cloud Computing User: Static provisioning for peak - wasteful, but necessary for SLA

“Statically provisioned” data center

“Virtual” data center in the cloud

▨ Unused resources

Source: http://slideplayer.com/slide/6085063/

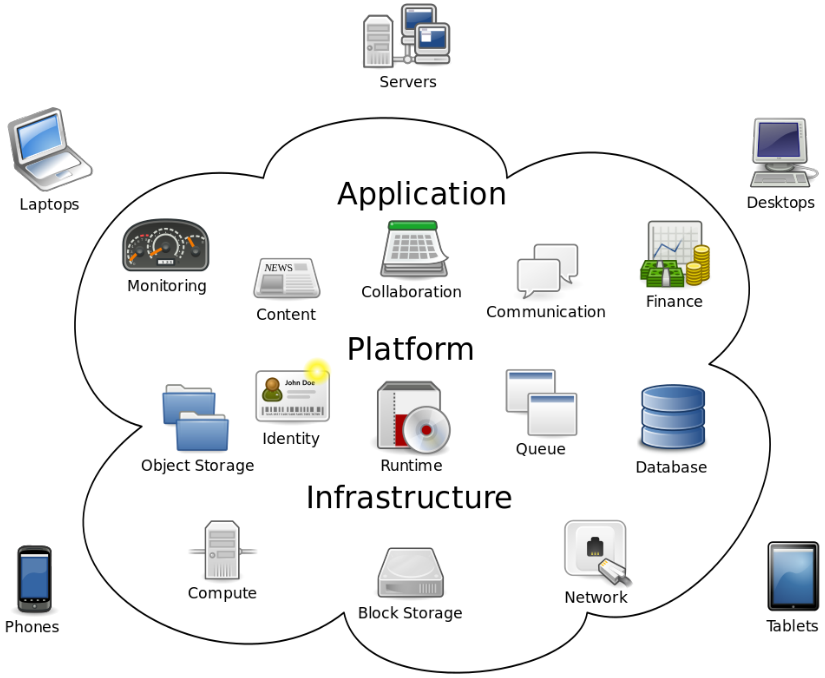
Cloud Computing, defined again

- The **canonical definition** comes from the US National Institute of Standards and Technology (NIST), http://goo.gl/eBGBk.
- In a nutshell, Cloud Computing deals with:

1 Supplying

2 information and communication technologies

3 as a service

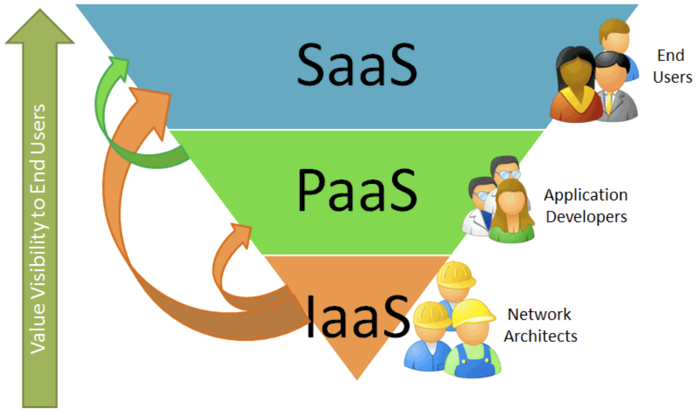


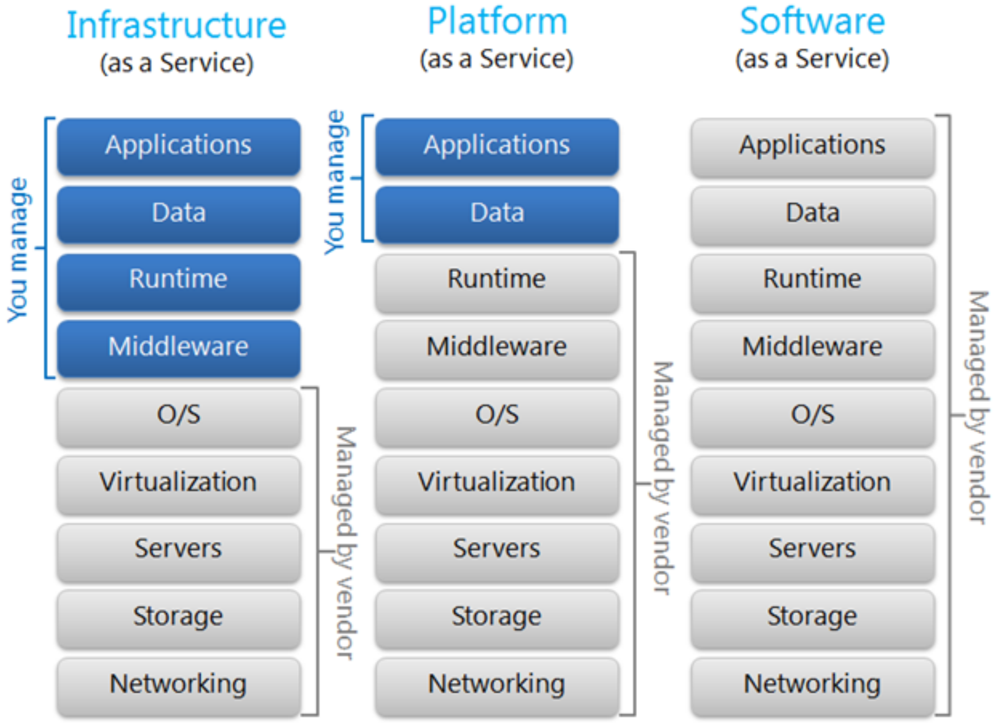
Cloud computing

The 5 Cloud postulates

1. Self-service, on-demand provisioning
2. Network-based access
3. Resource sharing
4. Elasticity (*with infinite resources*)
5. Pay-per-use

What matters at the end ***are the applications***.





Infrastructure (as a Service)	Platform (as a Service)	Software (as a Service)
Applications	Applications	Applications
Data	Data	Data
Runtime	Runtime	Runtime
Middleware	Middleware	Middleware
O/S	O/S	O/S
Virtualization	Virtualization	Virtualization
Servers	Servers	Servers
Storage	Storage	Storage
Networking	Networking	Networking

What is **IaaS**, i.e. Infrastructure as a Service

- The **IaaS** layer includes the basic building blocks of a data center:
  - **Storage** → storing data, maybe lots of data, possibly at low cost.
  - **Compute** → machines where I can host my services or run my applications.
  - **Network** → connecting resources through some "Software-Defined Network" infrastructure.
- In many cases, the provisioned Cloud infrastructure may be "virtual" (we'll explore this more in detail later).
- There is no need to know low-level details and no need to contact administrators to install anything.

What is **PaaS**, i.e. Platform as a Service

- The **PaaS** layer is a programmable platform allowing you to request a Cloud provider several ready-to-use components, that you can then use in and for your applications.
- For example, through a PaaS layer you could request:
  - An application server or a web server (or even a cluster of web servers) completed with database(s), virtual storage, load balancers and other dependencies.
  - An entire cluster of systems with a certain operating system and an entire environment already installed, configured and ready to be extended by you.

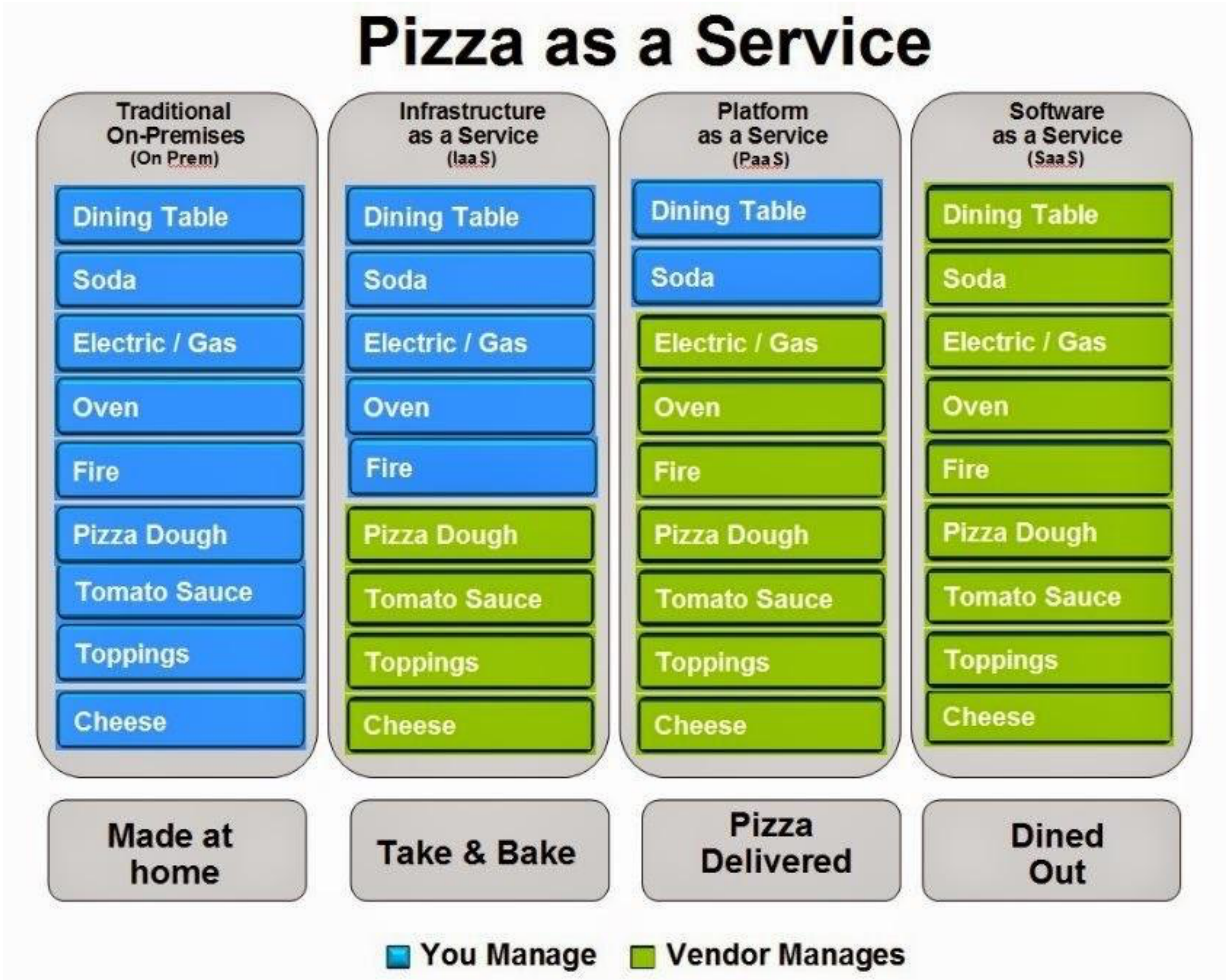
What is **SaaS**, i.e. Software as a Service

- With the **SaaS** layer, you are directly given access to some application software. You don't have to worry about the installation, setup and running of that software. You typically access SaaS apps via a web browser or an application, possibly mobile.
- Some SaaS examples: Gmail and in general Google Apps, Office365, social media such as Facebook, Twitter, Instagram, etc.

IaaS vs. PaaS vs. SaaS

	IaaS	PaaS	SaaS
What you get	You get the infrastructure. Freedom to use or install any OS or software	You get what you demand: software, hardware, OS, environment.	You don't have to worry about anything. A pre-installed, pre-configured package as per your requirement is given.
Deals with	Virtual Machines, Storage (Hard Disks), Servers, Network, Load Balancers etc.	Runtimes (like java runtimes), Databases (like mysql, Oracle), Web Servers (Tomcat etc.)	Applications like email (Gmail, Yahoo mail etc.), Social Networking sites (Facebook etc.)
Popularity	Highly skilled developers, researchers who require custom configuration as per their requirement or field of research.	Most popular among developers as they can directly focus on the development of their possibly complex apps or scripts.	Most popular among normal consumers or companies which rely on software such as email, file sharing, social networking as they don't have to worry about the technicalities.

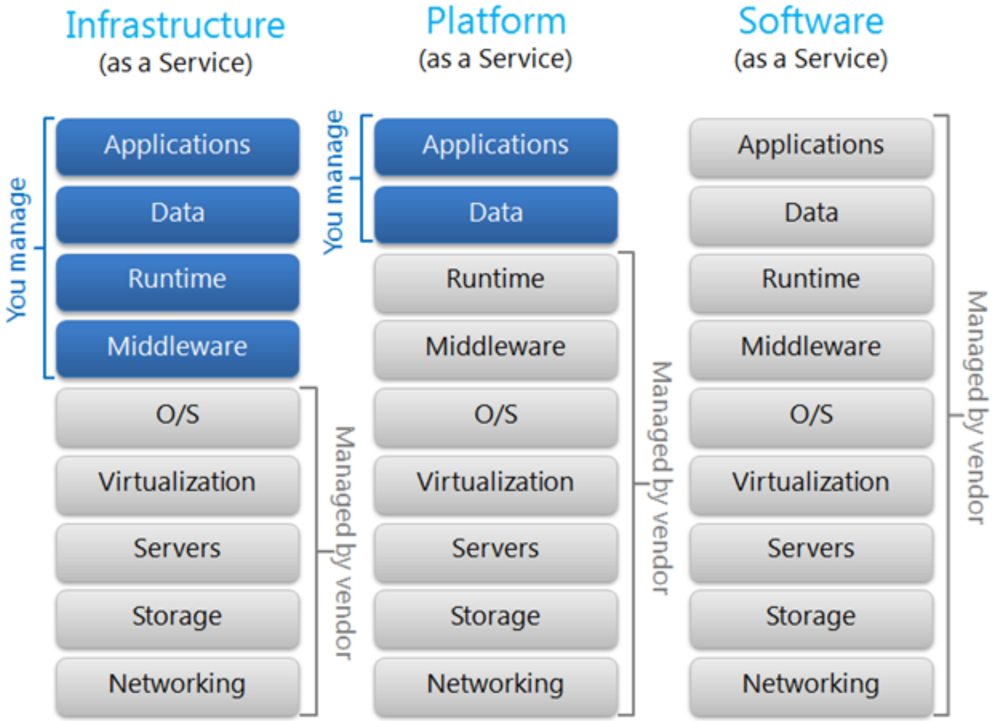
See https://goo.gl/ZwZtMQ



Pizza as a Service

Traditional On-Premises (On Prem)	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
🟦 Dining Table	🟦 Dining Table	🟦 Dining Table	🟩 Dining Table
🟦 Soda	🟦 Soda	🟦 Soda	🟩 Soda
🟦 Electric / Gas	🟦 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas
🟦 Oven	🟦 Oven	🟩 Oven	🟩 Oven
🟦 Fire	🟦 Fire	🟩 Fire	🟩 Fire
🟦 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough
🟦 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce
🟦 Toppings	🟩 Toppings	🟩 Toppings	🟩 Toppings
🟦 Cheese	🟩 Cheese	🟩 Cheese	🟩 Cheese
Made at home	Take & Bake	Pizza Delivered	Dined Out

🟦 You Manage 🟩 Vendor Manages

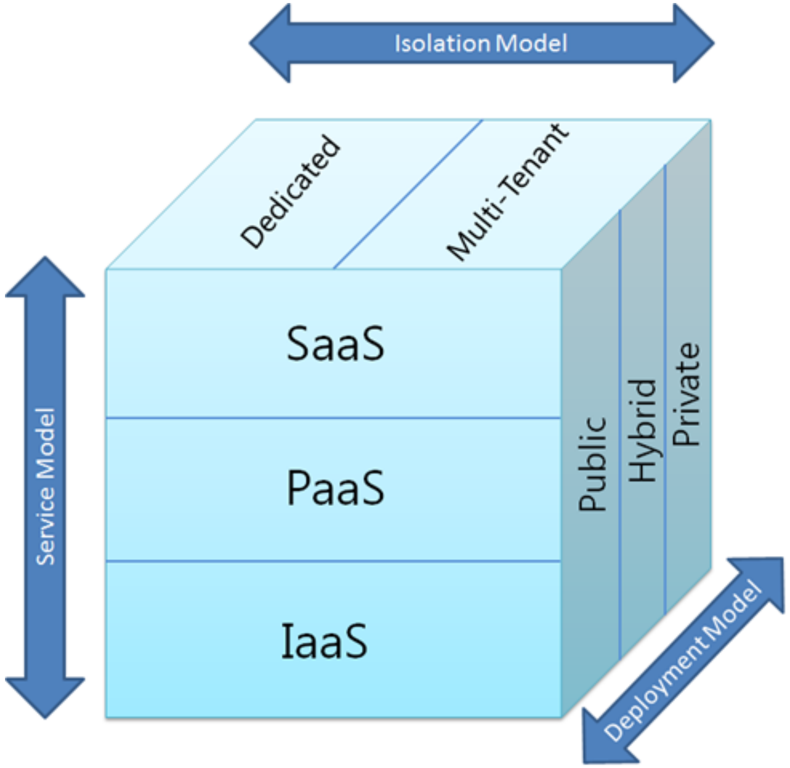


Infrastructure (as a Service)	Platform (as a Service)	Software (as a Service)
Applications	Applications	Applications
Data	Data	Data
Runtime	Runtime	Runtime
Middleware	Middleware	Middleware
O/S	O/S	O/S
Virtualization	Virtualization	Virtualization
Servers	Servers	Servers
Storage	Storage	Storage
Networking	Networking	Networking

Source: https://www.episerver.com/learn/resources/blog/fred-bals/pizza-as-a-service/

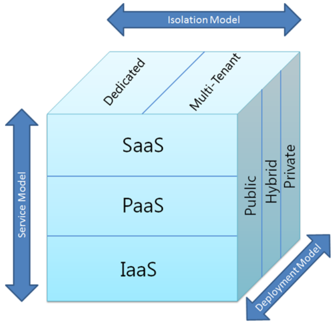
Let’s add some dimensions

- Beyond the *service models* (IaaS, PaaS, SaaS), in order to better understand the Cloud we need to consider other dimensions related to:
  - **Deployment** (where I distribute services).
  - **Isolation** (how I isolate services).



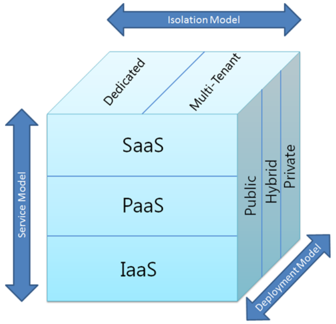
Source: http://goo.gl/1jmkR

Deployment: the «cloud types»



- **Private Cloud:**
  - The resources are **procured for exclusive *use*** by a single organization. Management, operation, ownership, location of the private cloud, however, can be independent by the organization using it.
- **Community Cloud:**
  - The infrastructure is **available to a community** of organizations sharing a common goal (for instance: mission, security requirements, adherence to common regulatory rules, etc.)
- **Public Cloud:**
  - The infrastructure is **available to the public** at large. Management can be either public or private. The location is at some service supplier premises.
- **Hybrid Cloud:**
  - The infrastructure is a **combination of two or more Cloud infrastructures** (private, public, community Cloud), connected so that there is some form of portability and interoperability of e.g. data or applications.

Isolation: how alone am I?



- Cloud **isolation models** are important and often somewhat ignored. We could have:
  - Dedicated infrastructures.
  - Multi-tenant infrastructures (i.e., with several [types of] customers).
- The isolation type is essential in many regards, such as:
  - Resource segmentation
  - Data protection
  - Application security
  - Auditing
  - Disaster recovery

PaaS: let’s make it more concrete

- IaaS = **Infrastructure** as a Service. That’s quite straightforward: computers (or *Virtual Machines* – VMs), disks, etc. provisioned over a distributed infrastructure – the Cloud.
- SaaS = **Software** as a Service. That’s also straightforward: Cloud-based solutions (applications, services) that are ready to use, such as email, collaborative office suites, etc.
- PaaS = **Platform** as a Service. But what does *this* mean, in practice?

The **PaaS** can be seen as a way to **program** a Cloud infrastructure, i.e. it provides a platform and an environment allowing developers to build applications and services.

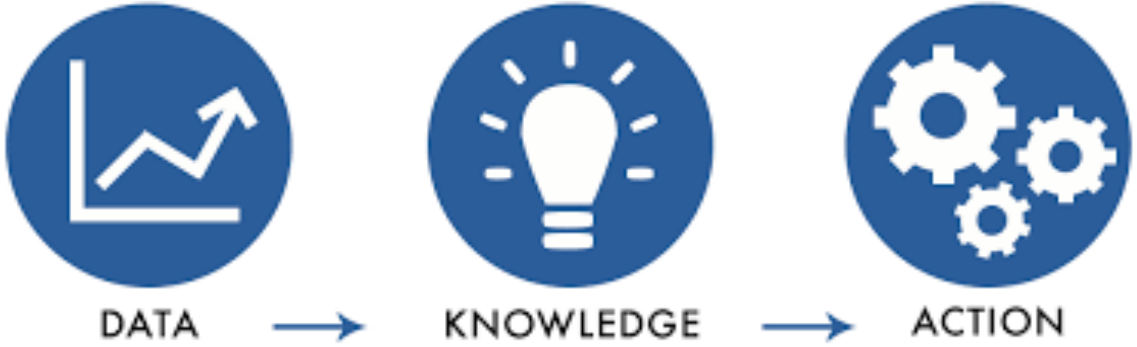
PaaS for developers

- The PaaS layer allows users to ***create and orchestrate software applications*** using tools supplied by a Cloud provider.
  - PaaS services can include ***preconfigured features*** that customers can subscribe to; they can choose to include those features that meet their requirements.
- The infrastructure and the PaaS services are ***managed*** for customers and ***support*** is normally available.
- Services are ***constantly updated***, with existing features upgraded and additional features added.
- PaaS providers can ***assist developers*** from the conception of their original ideas to the creation of applications, down to testing and deployment.

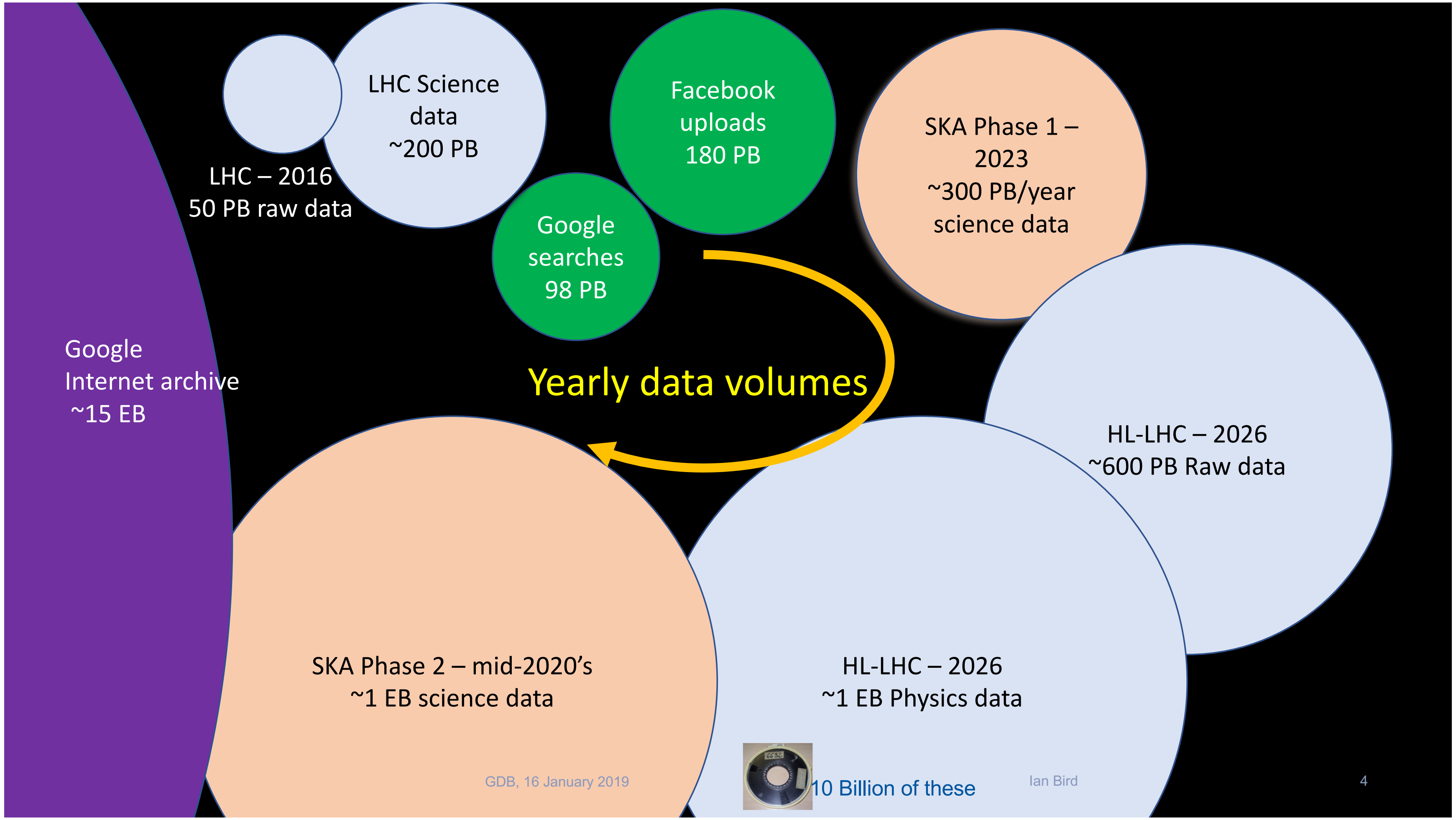
Which PaaS services are out there?

- AWS offers for example **Elastic Beanstalk** (https://aws.amazon.com/elasticbeanstalk/), which is useful for “deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.”
- There are several other PaaS frameworks that can be used, some supported by public Cloud providers, others that can be installed on private resources. For example:
  - **The INDIGO-DataCloud PaaS layer**, https://www.indigo-datacloud.eu/service-component
  - **Cloudify**, https://cloudify.co
  - **RedHat OpenShift**, https://www.openshift.com
- Later in this school we’ll learn more about some PaaS services.

Why are we talking about all this stuff?



Why are we talking about all this stuff?



LHC Science data ~200 PB

LHC – 2016
50 PB raw data

Facebook uploads 180 PB

Google searches 98 PB

SKA Phase 1 – 2023
~300 PB/year science data

Google
Internet archive
~15 EB

Yearly data volumes

HL-LHC – 2026
~600 PB Raw data

SKA Phase 2 – mid-2020's
~1 EB science data

HL-LHC – 2026
~1 EB Physics data

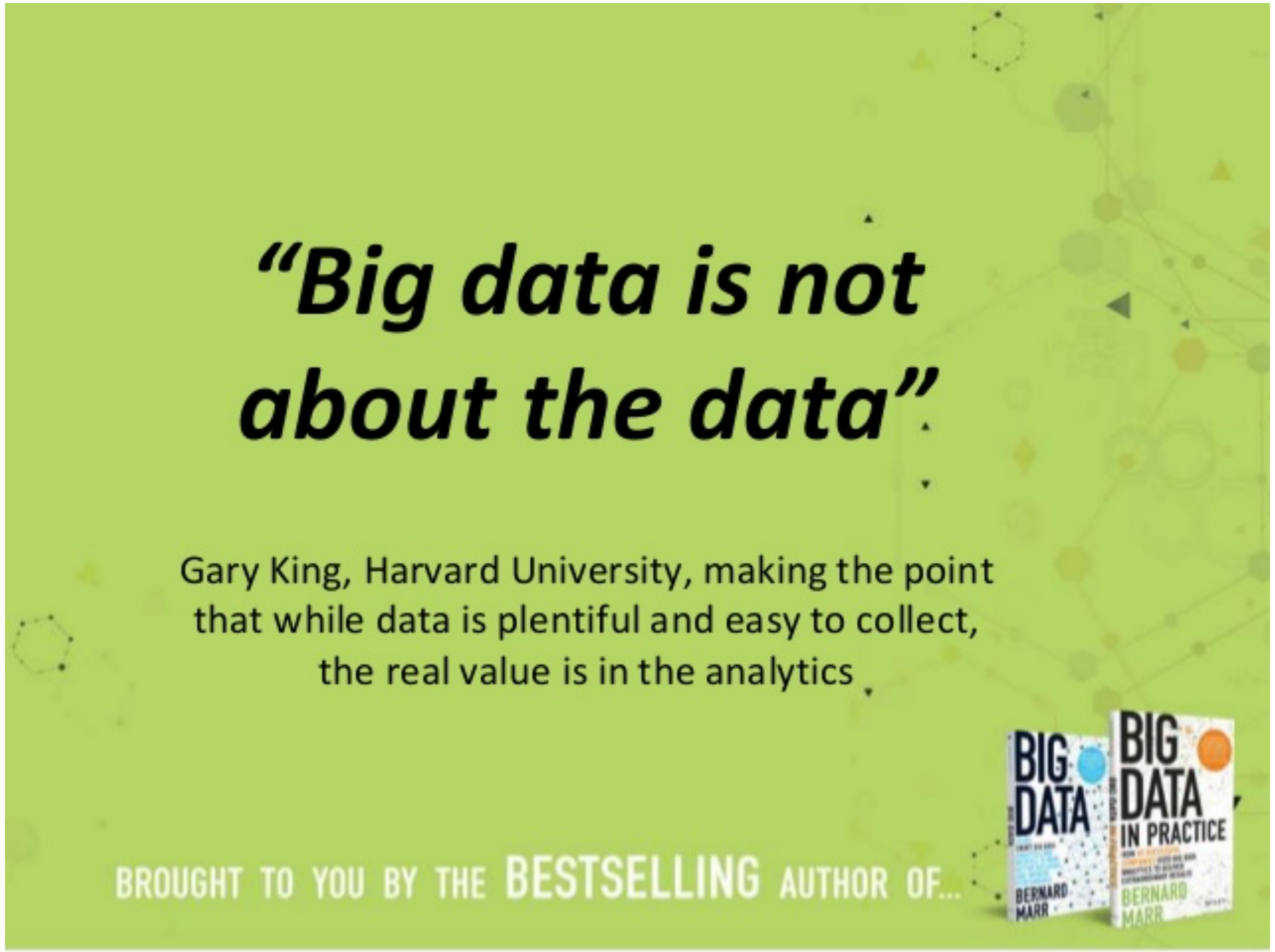
10 Billion of these

GDB, 16 January 2019

Ian Bird

4

However…



“Big data is not about the data”

Gary King, Harvard University, making the point that while data is plentiful and easy to collect, the real value is in the analytics

BROUGHT TO YOU BY THE BESTSELLING AUTHOR OF...

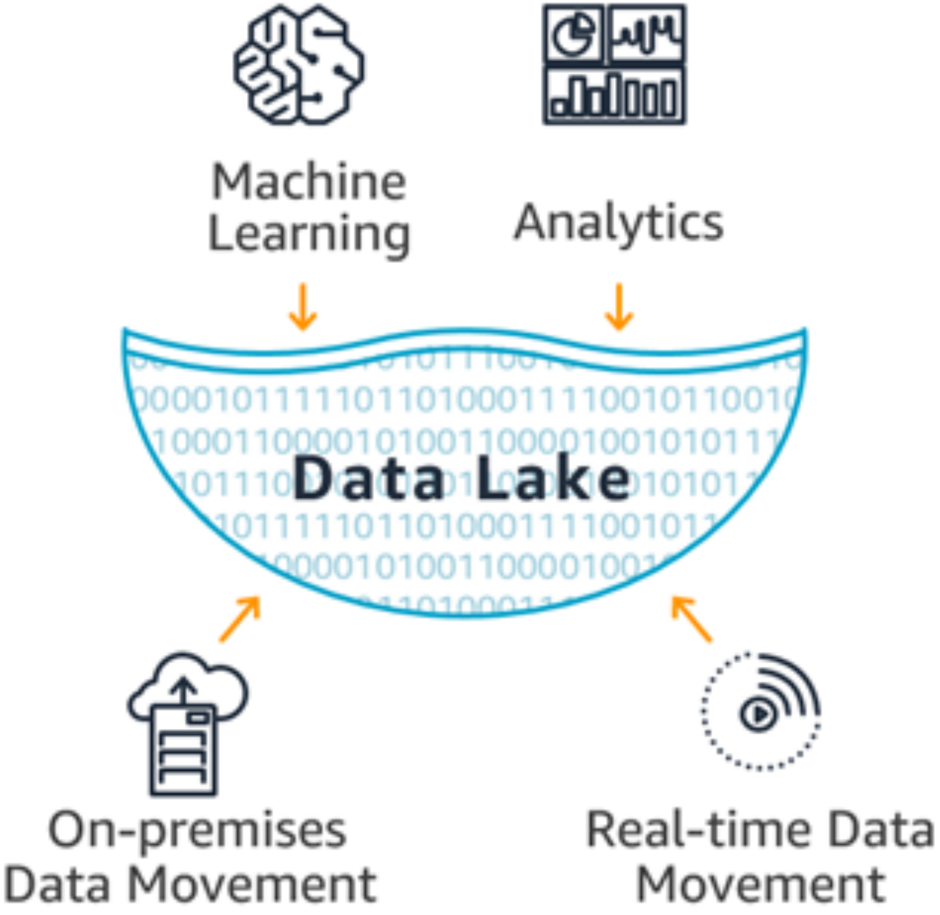
BIG DATA
BERNARD MARR
BIG DATA IN PRACTICE
BERNARD MARR

Source: https://www.slideshare.net/BernardMarr/big-data-best-quotes/3-Big_data_is_notabout_the

The “FAIR Data” slogan

- “The FAIR Guiding Principles for scientific data management and stewardship”, https://www.nature.com/articles/sdata201618 (2016)
- **FAIR** Data:
    - **F**indable
    - **A**ccessible
    - **I**nteroperable
    - **R**eusable
- These are principles that should “put specific emphasis on enhancing the ability of machines to automatically find and use the data, in addition to supporting its reuse by individuals”.
- **Whatever the formal definition or principles**, we must have ways to organize and treat data with *intelligence*, in order to extract value from it.

A “Data Lake”



Credits: Amazon

How do we build a “scientific data lake”?

- … which must be unfragmented, hybrid, agile, extensible, inclusive of existing solutions and know-how?

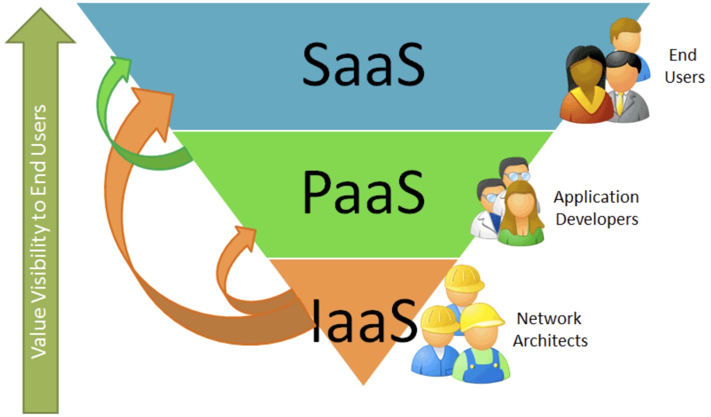
First of all, what is really a ”data lake”?

Let’s see it from two angles:

1. **Service-orientation**: it is a **cloud of data services**, where open access and open science are key words.
2. **Support-orientation**: It is realized out of a **backbone** composed of a limited set of data centers offering resources and know-how.
   - In particular, **know-how** (not only money or political pressure) is a critical component to decree the success of a solution vs. its irrelevance.

The added value for scientists

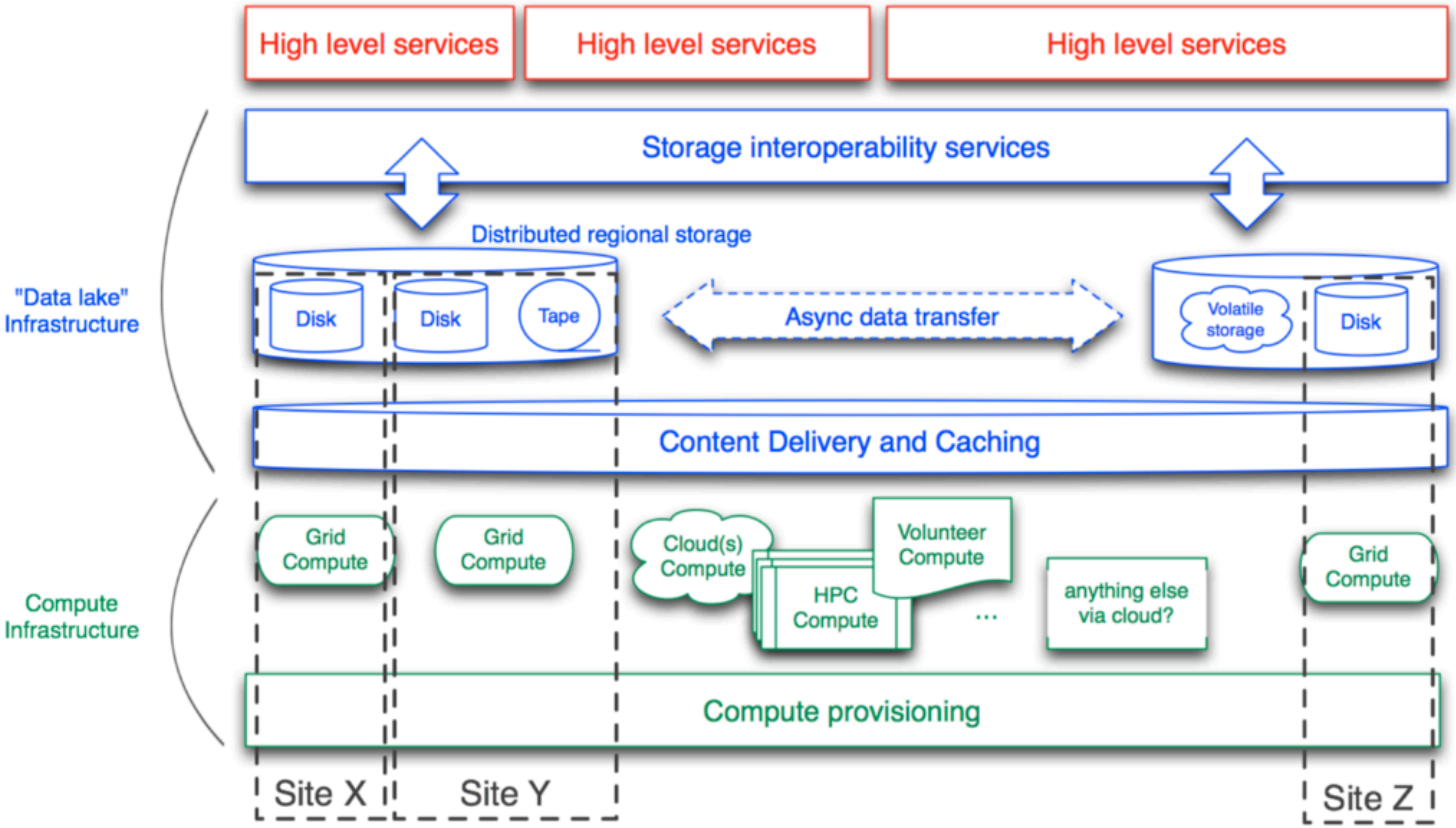
- Focus on **real added value solutions**, not on just infrastructure or hardware – which nevertheless must be obviously available in some form.
- Focus on **bridging gaps across scientific domains and technology through open standards**, not on silos: research, education & agility.
- Focus on **progressive peer-to-peer agreements with larger entities** beyond the “lake”.

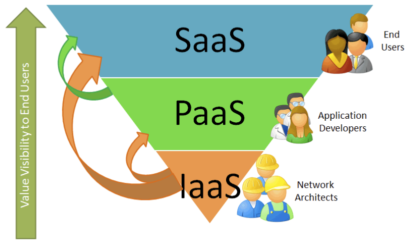


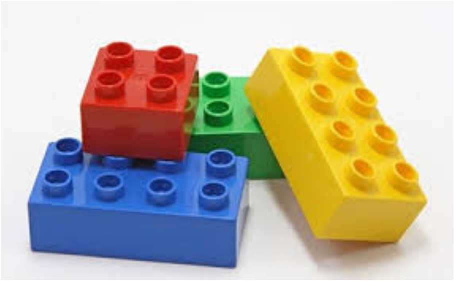
The water ecosystem as a planning model

- **The Pond**: a “single” center.
- **The Lake**: a backbone-centered federation centered on specific needs *but* using general solutions as much as possible.
- **The River**: the conduits to more general upper infrastructures.
- **The Sea**: a large, multi-purpose, many-stakeholders resource / solution set (e.g., the European Open Science Cloud)
- **The Ocean**: a worldwide collection of solutions.

Architecturally



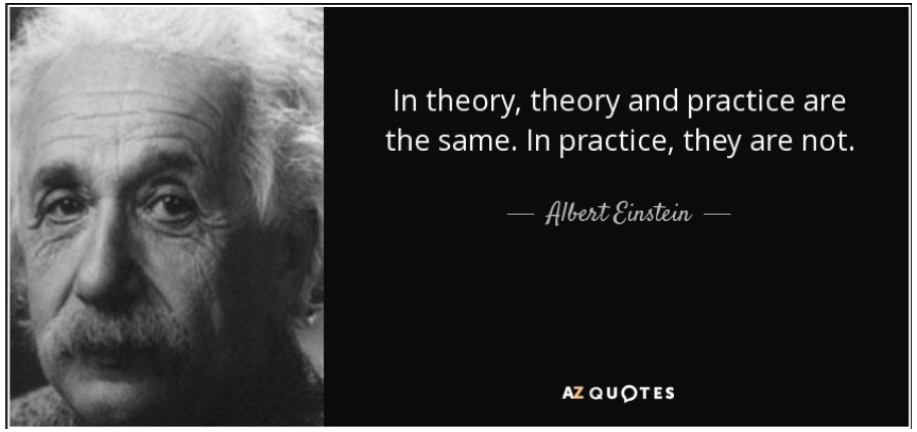




Concretely: a typical data processing workflow

- In a **naïve set of assumptions**, I *have*:
  - A data set I want to analyze.
  - Some algorithms I want to apply to this data.
  - Some software that can use these algorithms.
  - Some computing resources that can run this software.
  - Some space where I can store my output.

- I *assemble everything together* and off I am.



In theory, theory and practice are the same. In practice, they are not.

— Albert Einstein —

AZ QUOTES

In fact, there are several challenges

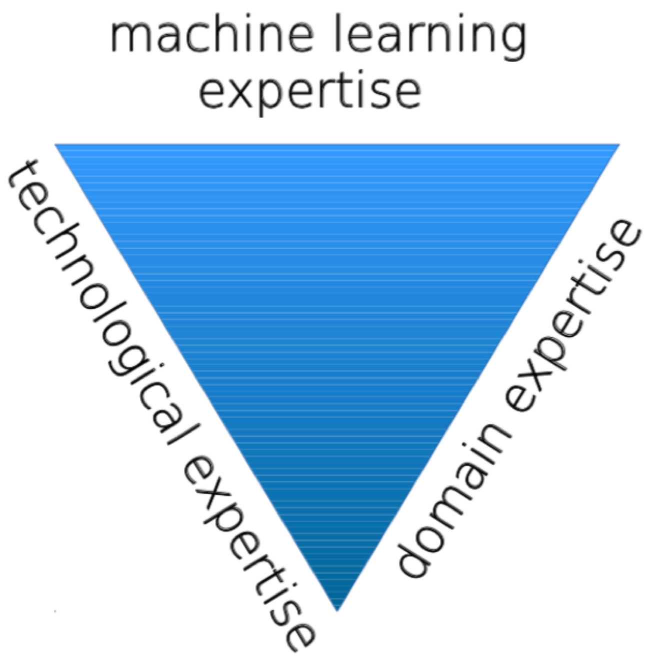
(which go well beyond the "FAIR Data" mantra)

- **Accessing Data:**
  - Is the data open? For all? Always?
  - Is the data distributed? Where? How do I find and integrate it?
- **Processing Data:**
  - Where can I find the resources I need for my workflow / data processing?
  - Does all my data require the same Quality of Service (QoS)? The same algorithms? How do I decide that?
  - How open are the tools that will process my data?
  - What happens if some services I need or use are not available? If there is a failure somewhere?
- **Post-processing Data:**
  - What if I get new data? How do I re-train my model?
  - How can I reproduce, tweak and (re-)publish my work?

In the end, **how much effort and know-how** is needed to have all this in place?

Take Machine Learning.

Which *know-how* do I need to effectively use it?



What matters, at the end, are the applications.
But how to properly get to the application level?

Choices…

TensorFlow: speech and image recognition (Google Brain Team)

Keras: Python NN library (Francois Challet, Google)

PyTorch: DL library (Facebook KI)

Caffe: DL library (UC Berkeley)

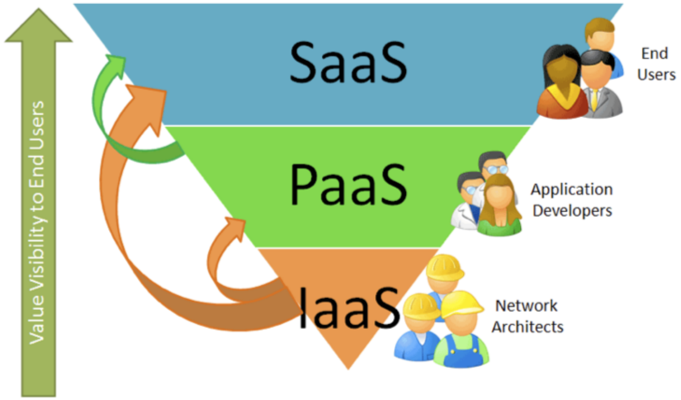
mxnet: scalable DL framework (Apache)

OpenCV
computer vision

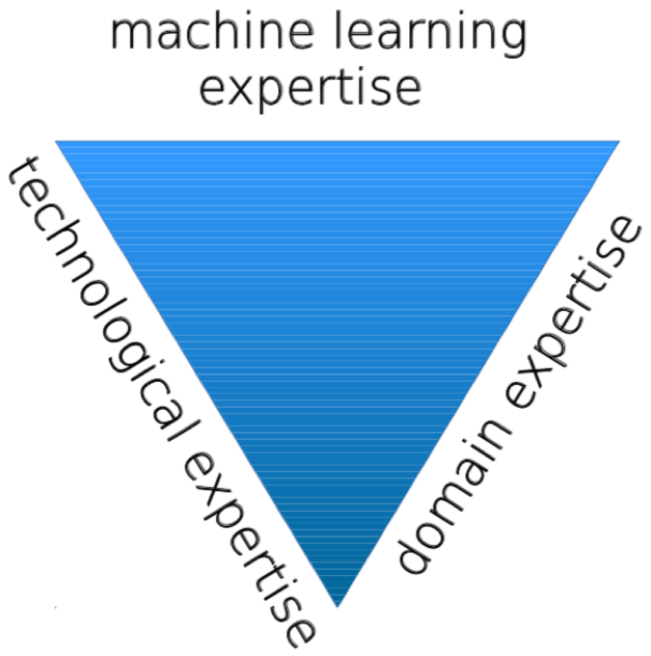
NumPy
num. lin. alg.

SciPy.org
sci. comp.

matplotlib
plotting



Categorizing the know-how

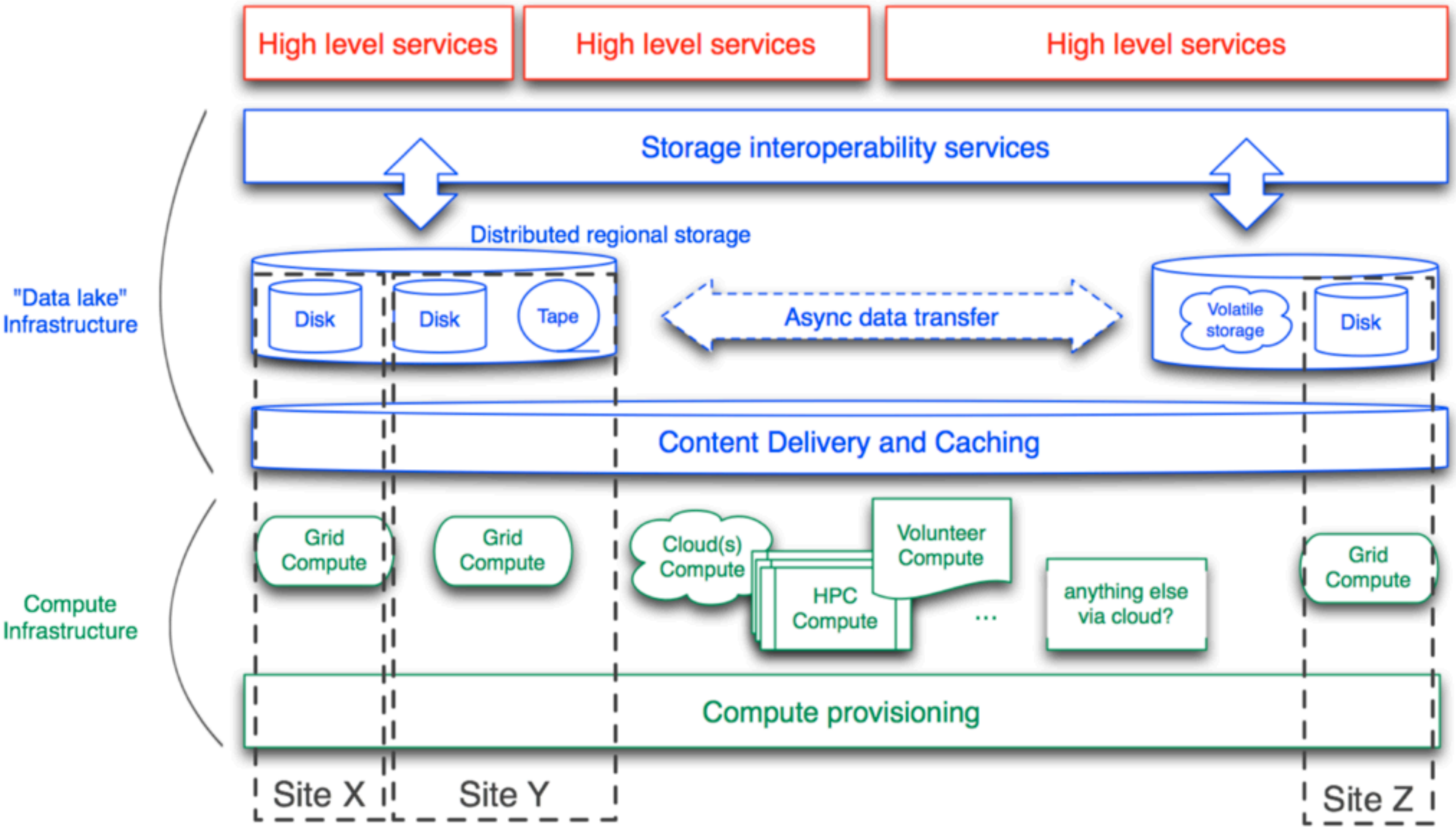


- **Category 1**: deploy an already trained ML model for somebody else to use on her own trained data set.
  - Domain knowledge
- **Category 2**: retrain (parts of) an already trained ML model to make use of its inherent knowledge and solve a new learning task.
  - Domain + ML knowledge
- **Category 3**: completely work through the ML / Deep Learning cycle with data selection, model architecture, training and testing.
  - Domain + ML + technological knowledge

How do we go about this?

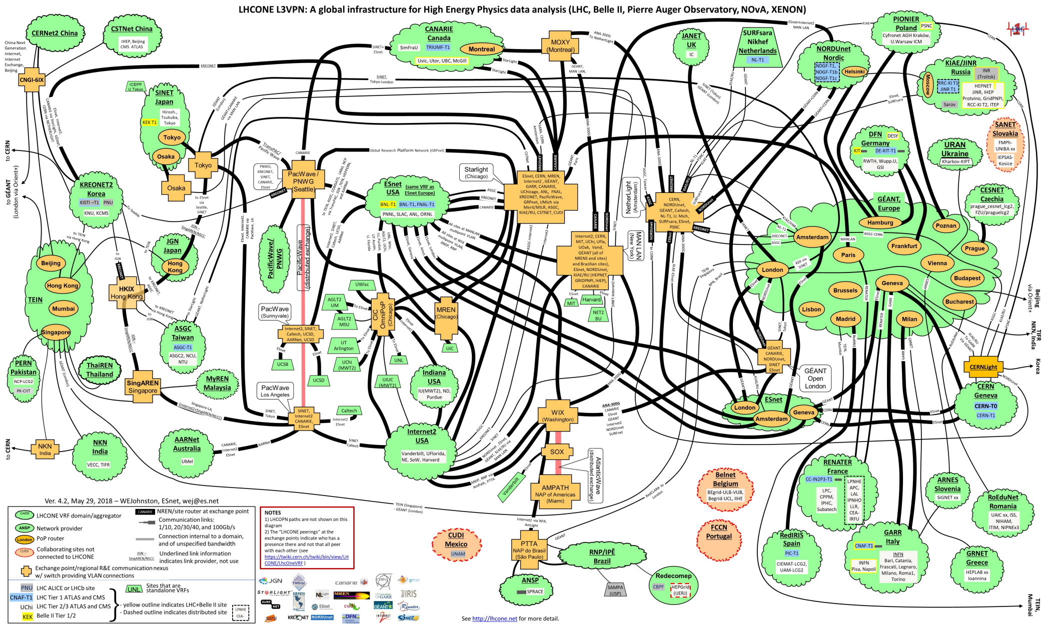
- **Objective 1: build added value services on top of IaaS & PaaS infrastructures**
  - Due to the nature of many scientific endeavors (but also public services and industry), these infrastructures may often be hybrid, i.e. public + private.
- **Objective 2: lower the entry barrier for non-skilled scientists**
  - Transparent ("ZeroOps") execution on e-Infrastructures.
  - Offer ready-to-use modules, components or services through a catalog, or rather a configurable marketplace.
  - Enable flexible service composition.
  - Implement common software development techniques also for scientists' applications ("DevOps").

What are *intelligent* infrastructures, then?



What are *intelligent* infrastructures, then?

- They are an *ensemble* of resources, tools, and know-how (machine- and human-based!) that allow us to:
  - Move or replicate data around. Do you think this is simple? Think again.
  - Automate the instantiation of ad-hoc clusters over several IaaS, *compose services*, integrate authentication and authorization, integrate these capabilities into your components, monitor and auto-scale everything. Review the picture on a part of just a network layer at the beginning of this talk to start and appreciate how complex this can be.



LHCONE L3VPN: A global infrastructure for High Energy Physics data analysis (LHC, Belle II, Pierre Auger Observatory, NOvA, XENON)

Ver. 4.2, May 29, 2018 – WEJohnston, ESnet, wej@es.net

GARR LHCONE VRF domain/aggregator
ANSP Network provider
London PoP router
CUDI Collaborating sites not connected to LHCONE
Exchange point/regional R&E communication nexus w/ switch providing VLAN connections
CANARIE NREN/site router at exchange point
Communication links: 1/10, 20/30/40, and 100Gb/s
Connection internal to a domain, and of unspecified bandwidth
JGN – SingAREN/NSCC Underlined link information indicates link provider, not use
PNU LHC ALICE or LHCb site
CNAF-T1 LHC Tier 1 ATLAS and CMS
UChi LHC Tier 2/3 ATLAS and CMS
KEK Belle II Tier 1/2
UNL Sites that are standalone VRFs
- yellow outline indicates LHC+Belle II site
- Dashed outline indicates distributed site

NOTES
1) LHCOPN paths are not shown on this diagram
2) The "LHCONE peerings" at the exchange points indicate who has a presence there and not that all peer with each other (see https://twiki.cern.ch/twiki/bin/view/LHCONE/LhcOneVRF)

See http://lhcone.net for more detail.

- You will see a simplified application of these points on Thursday in both lectures on hands-on exercises.

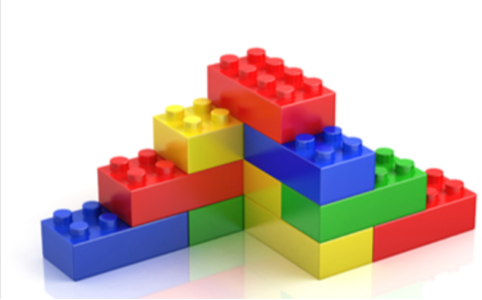
What about ML / Deep Learning?

Right now:

- Scientists typically create a deep learning application on their personal computers.
- The deep learning model is trained in a GPU-based node (maybe also locally).
    - What happens if they do not have access to one?
- The work is published (or not)
    - Model architecture, configuration, scientific publication, etc.
- **However:**
    - How can a scientist easily offer all this to a broader audience? (i.e., reuse the effort and the results)
    - What about dependencies?

From service composition to reusable components

- **Service composition**, if done properly, provides a way to re-deploy the same topology and the same set of solution (both possibly rather complex) in an automated way, over different infrastructures.
- With proper service composition, scientists should not need to deal with technologies and infrastructures.
- We currently have open technologies to create service composition with high-level descriptions that avoid details.
- During the school, we'll explore ways to reuse distributed solutions through a **catalog of components.**

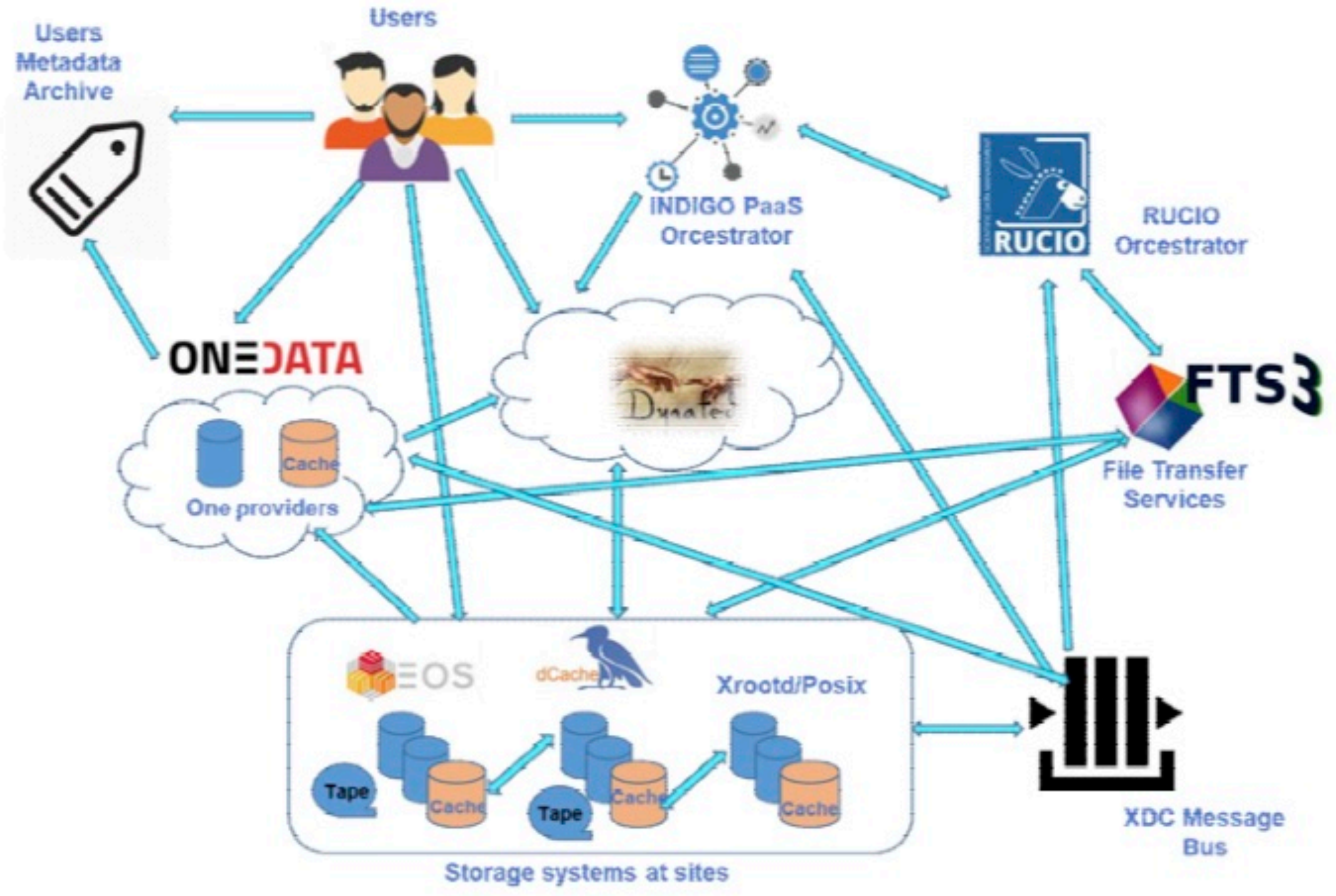


What about data management?

Right now:

- Scientists are typically oblivious to data distribution policies, in particular for:
  - QoS-based (e.g. disks vs tape vs SSD) data distribution policies, esp. cross-sites.
  - Data lifecycle management.
- Sometimes, they would like to perform some data pre-processing during data ingestion: how?
- They would like to control how replica management is done.
- They would like to perform some *smart* data caching, or data management based e.g. on access patterns: how?
  - For example, automatically move *unused* data to some “glacier-like” storage, and conversely move “hot” data to some fast storage.

Data Management and Integrated Data Orchestration



Recap

- We have said that **intelligent infrastructures**:
    - Allow us to dynamically instantiate *ad-hoc* clusters over many IaaS resources, compose services, integrate various types of AAI systems, integrate various components into existing frameworks, monitor and auto-scale them.
    - Let us re-use deep learning-based building blocks, customize, and publish them in high-level catalogs of services for deployment in hybrid (public and private) Clouds.
    - Allow us to perform data management automation, orchestration and optimization functions, as well as call QoS functions on storage.
- **We generally have solutions for several (not all) of these things**, and you will explore some of them in the SOSC hands-on projects and exercises.

Some further **key technical points** for *intelligent infrastructures* (1)

- **Infrastructure as Code**: focus on problem-oriented, code-based dynamic solutions that program the infrastructure, not the other way around. We could also call this solution co-design.
- **Event-driven support to [stream] processing**, i.e. reaction to changes in data sets or in general to resource availability.
- **Intelligence as a Service,** e.g. Machine/Deep Learning as a Service, but also Competence Centers to analyze and build bespoke solutions.
- **Caching and linked data.** Compute & data locality is not guaranteed in data lake, therefore an effective content-delivery service is needed.
- **Service and data replication and data reproducibility** across the multiple backbone data centers.

Some further **key technical points** for *intelligent infrastructures* (2)

- **Data life cycle management**, including data QoS & effective data management plans.
- **User-level (or user-friendly) workflows** to overcome technology barriers for non IT-expert scientists.
- **Connection to multiple data sources** such e-infrastructures, HPC centers, opportunistic resources, devices, storage systems, data sets, sync & share services. This should happen through an open and technology neutral view of infrastructures.
- **Integration with smaller compute centers** or to **commercial providers,** if advantageous.
- **Integration with national infrastructures**, and/or possibly also with super-national ones.

In summary

- **It is naïve to think that silos-based, in-house, proprietary, monolithic solutions will address the explosion of data production and related analysis**, esp. with complex requirements such as those found with Deep Learning and open science.
- **Transparency, support of *de jure* and *de facto* standards, provider-agnostic modular solutions** are the only way to go.
- **Many things remain to be done** to have complete, simple to use solutions for open, distributed science. You will see some of the current issues with your own eyes during this week at SOSC.
- But these are very exciting times to get engaged in these fields: **don't be afraid to take up the challenge, and have fun!**