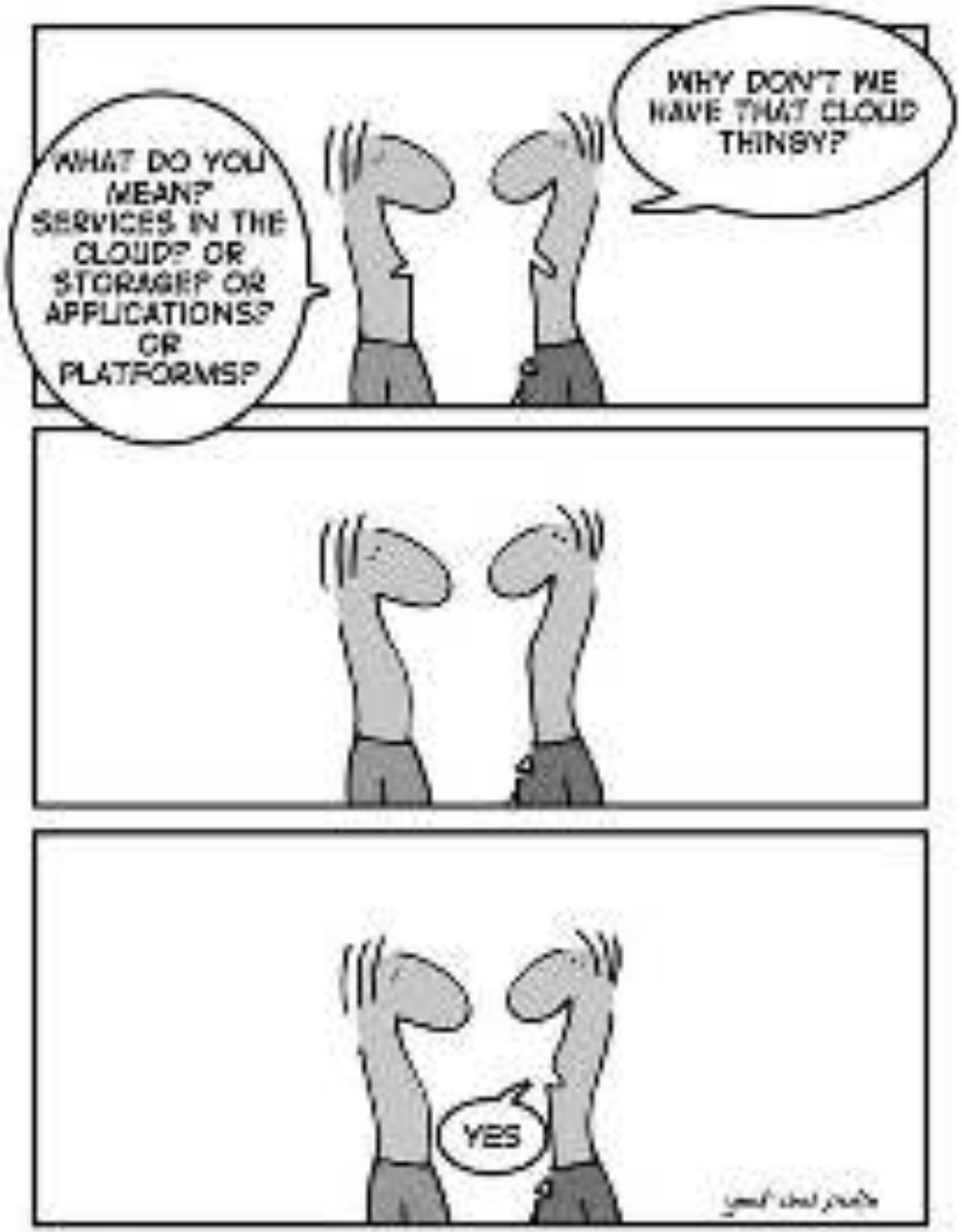
Introducing Cloud Computing and OpenStack

Davide Salomoni, INFN-CNAF

School on Open Science Cloud

Perugia, 5-9/6/2017

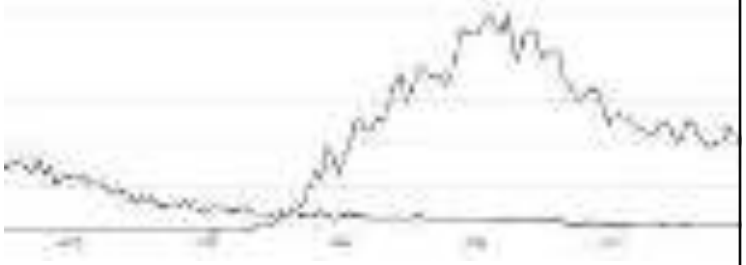


Source: https://goo.gl/KZfn8z

Cloud Computing: definition

- 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:

"Grid Computing" vs.
"Cloud Computing"Google
lookup stats
(from Google Trends)



1 Supplying
2 information and communication technologies
3 as a service

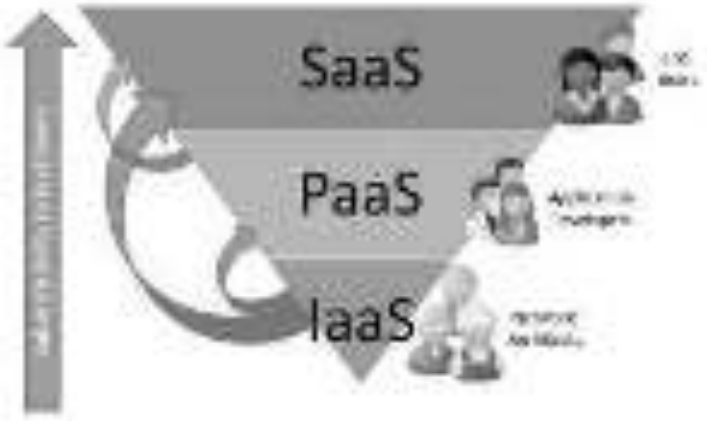
Why the Cloud?
(or, what was missing?)

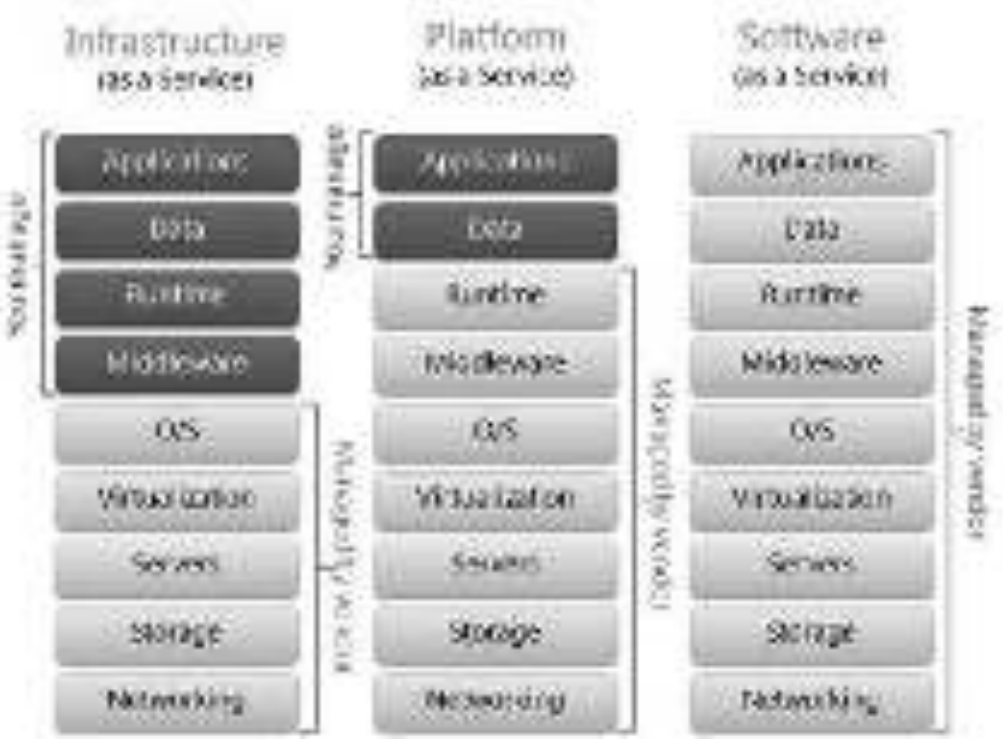
1. Ease of access to IT resources for small as well as big companies and [scientific] communities.
2. Software and financial sustainability.
3. Robustness (mitigation of vulnerabilities).
4. Modular and scalable architecture (or, flexibility).
5. Open source software, vendor independence.
6. Clear business model(s).

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***.





The emphasis on “service”

- In the standard Cloud definition (“Supplying information and communication technologies as a service”), the ***service toward the Cloud users is the essential part*** – e.g. for usability, flexibility, reliability, etc.
- Cloud computing is indeed typically modeled around *services* primarily linked to:
  - Infrastructure (**IaaS** → Infrastructure as a **Service**)
  - Platform (**PaaS** → Platform as a **Service**)
  - Software (**SaaS** → Software as a **Service**)

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

- **IaaS**, the basic building blocks of a data center:
  - **Storage** → I want to store data, lots of data, at low cost
  - **Compute** → give me a machine where I can host my services or run my applications
  - **Network** → create a "Software-Defined Network" infrastructure for me
- In many cases, in a "virtual" form (see later)
- No need to know details, no need to contacts administrators to install something

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

- **PaaS**, a computing platform providing you with several building blocks or components that you can request programmatically or statically. For example:
  - A cluster of systems with operating system and an entire execution environment installed and configured.
  - A web server (or a cluster of web servers) with database(s), virtual storage, load balancers, other dependencies.

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

- With **SaaS**, you are directly given access to some application software. You don't have to worry about the installation, setup and running of the application. You typically access SaaS apps via a web browser.
- For example: gmail, social media such as Facebook, Twitter, 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

Remember what matters…

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

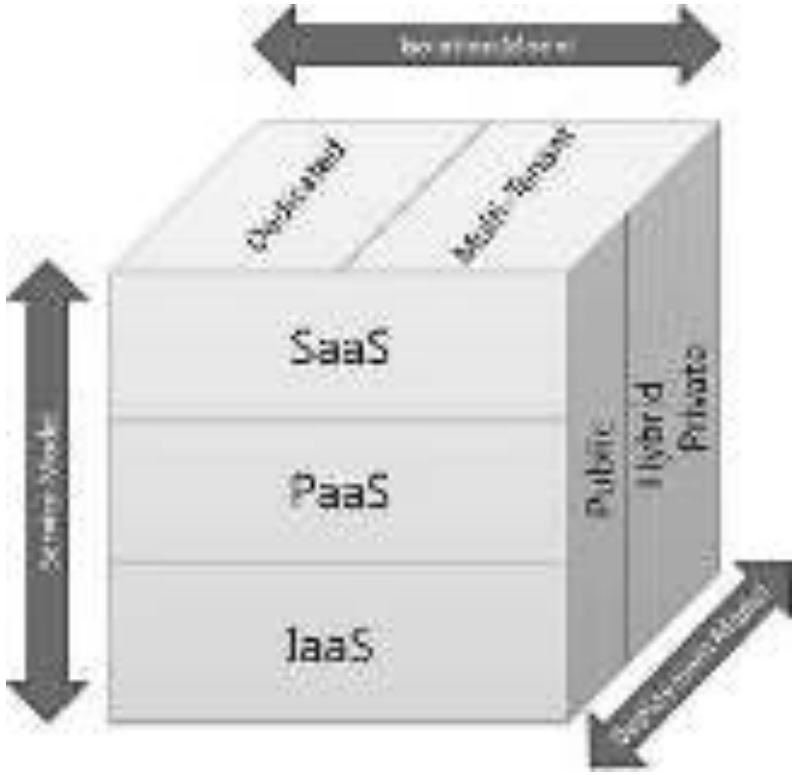
TRUE!

… however, **without Cloud providers (public or private)**, and without **efficient and effective** ways of managing distributed resources, applications cannot be deployed!
(rather obvious isn't it)

How do you provision, find and use Cloud resources, then?

Let's add dimensions

- Beyond the *service models* (IaaS, PaaS, SaaS), important parts to define and understand Cloud computing are the models linked to:
  - ***deployment*** (where I distribute services)
  - ***isolation*** (how I isolate services)



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

Deployment: “Cloud types”

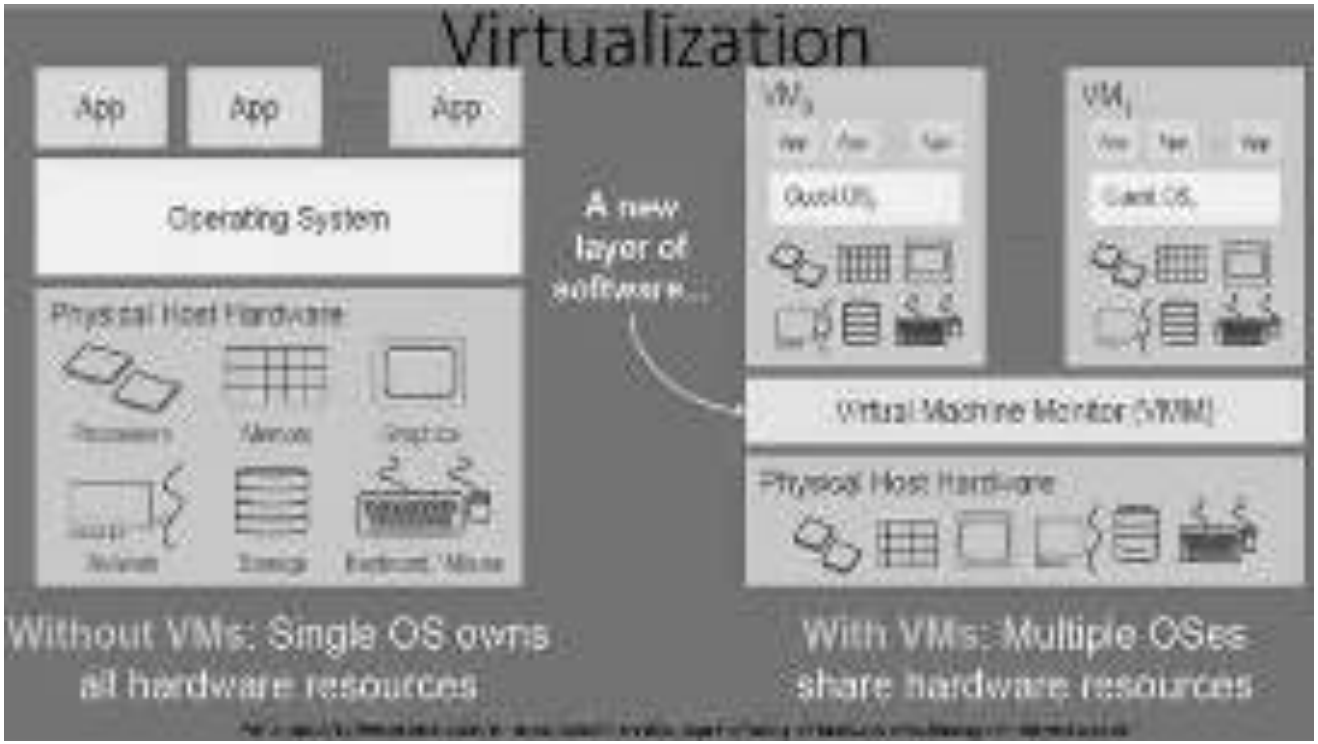
- **Private Cloud:**
  - The infrastructure is **procured for exclusive *use*** by a single organization. Management, operation, ownership, location of the private cloud, however, can be independent by the organization using it.
- **Community Cloud:**
  - The infrastructure is **available to a community** 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 of e.g. data or applications.

Isolation

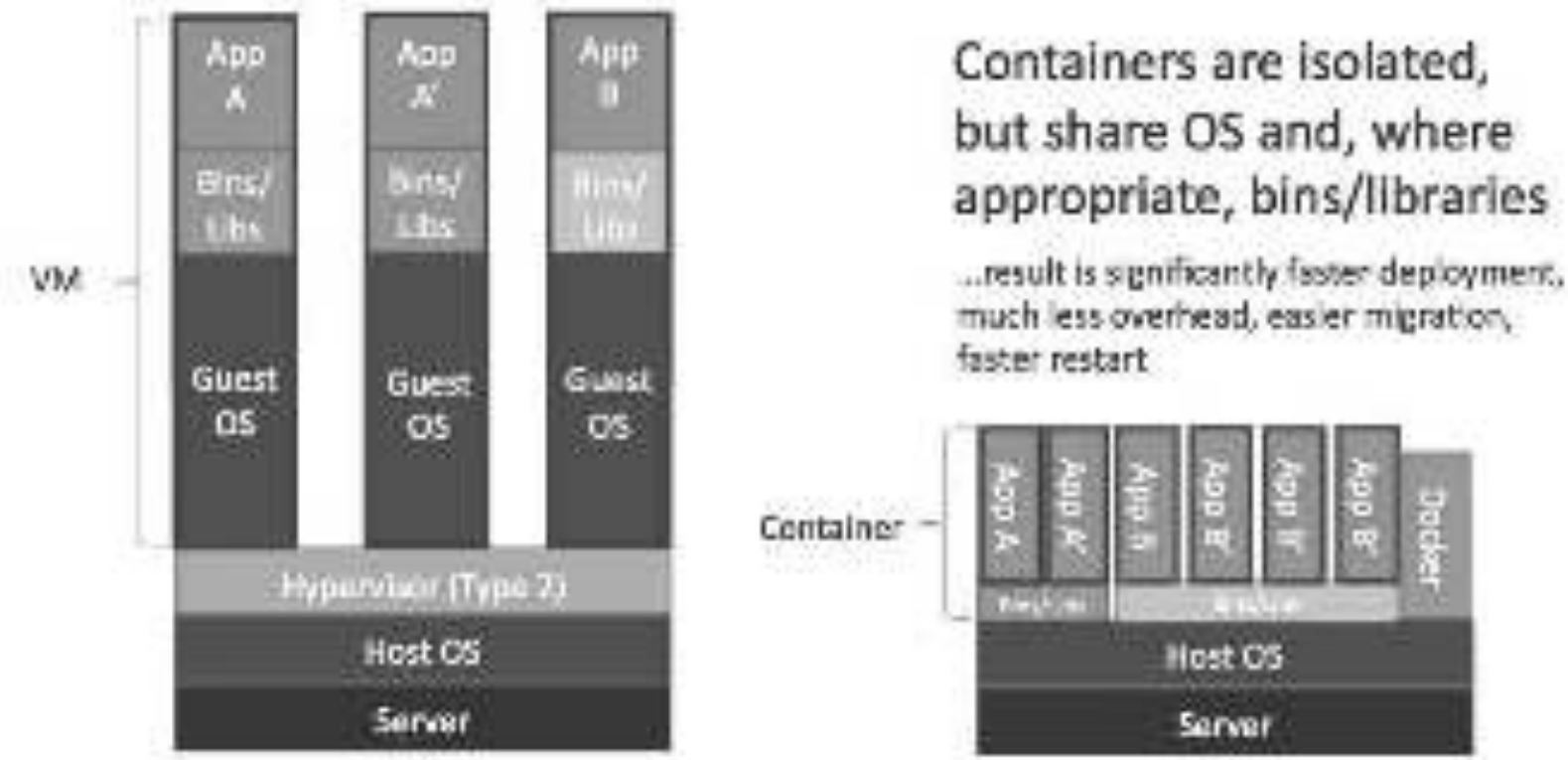
- Cloud **isolation models** are important and often ignored. We could have :
  - Dedicated infrastructures
  - Multi-tenant infrastructures (i.e., with several [types of] customers)
- The isolation type is essential in many regards, such as:
  - Resource segmentation
  - Data protection
  - Application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

Virtualization 101

- Informally, by *virtualization* we mean the creation of a virtual version of *something*.
  - For instance, an hardware platform, an operating system, a storage device, a network resource.
  - Through an **abstraction**: an intermediate level between hardware/software and applications, simplifying and hiding underlying details.



Containers, or “lightweight VMs”

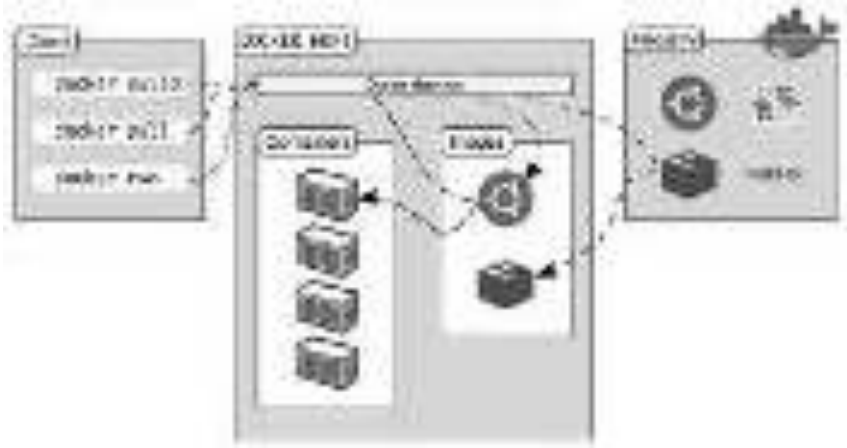


Source: Docker (http://goo.gl/4jh8cX)

Docker containers

Docker: Build-Ship-Run

- Docker is an open source engine for the easy creation of lightweight, portable, self-sufficient containers from any application.
- The same container that a developer builds and tests on a laptop can run at scale, in production, on VMs, private, public clouds and more.
- Docker features:
  - versioning (git-like)
  - component re-use
  - sharing (public repository)



Dockerfiles

- A **Dockerfile** is a script, composed of various commands (instructions) and arguments listed successively to automatically perform actions on a base image in order to create (or form) a new one.

```
FROM ubuntu
MAINTAINER Romin Irani (email@domain.com)
RUN apt-get update
RUN apt-get install -y nginx
ENTRYPOINT ["/usr/sbin/nginx","-g","daemon off;"]
EXPOSE 80
```

Docker: ship and run

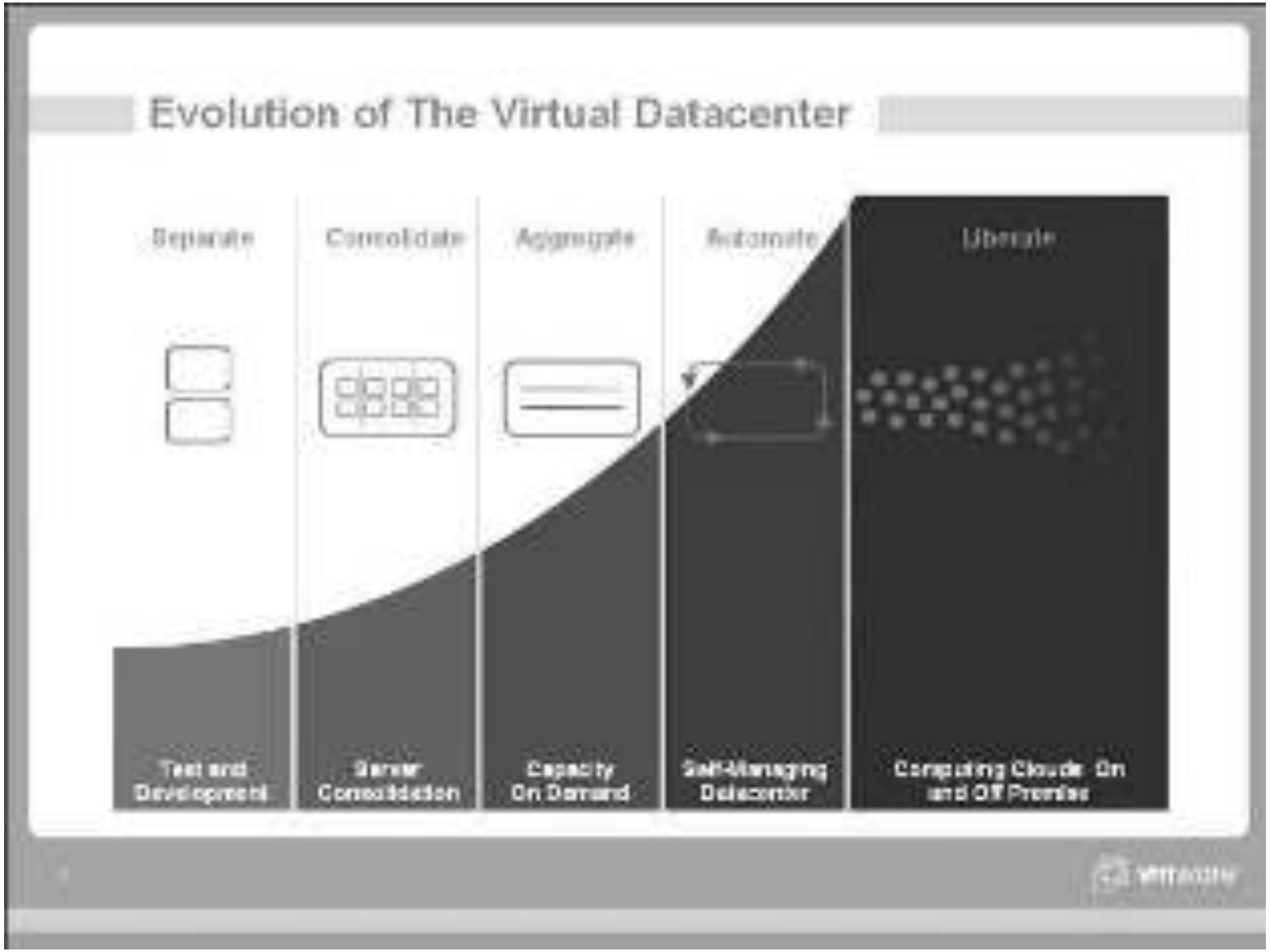
- Ship a Docker Image
  - *docker push indigodatacloudapp/ambertools*
- Fetch a Docker Image
  - *docker pull indigodatacloudapp/ambertools*
    - download docker image from Docker Hub to a local Docker repository
- Run a docker container
  - *docker run […] indigodatacloudapp/ambertools*
    - creates a docker container out of the docker image

Virtualization or Cloud Computing?

- Cloud computing can also be provisioned *without* virtualization technologies.
  - However, virtualization technologies often allow to reduce operational and capital account expenses.
  - On the other hand, being able to very rapidly provision VMs is not very efficient, if e.g. it takes several months to purchase and install the physical hosts where VMs will run.
    - In addition: is the *time and effort* spent for provisioning and managing a virtualization layer recovered e.g. by savings, associated to not having to dedicate many physical servers?
    - Hence the importance of using *installation, monitoring and accounting tools that are as automated as possible* (→ not a trivial task)

In summary: virtualization vs. Cloud computing

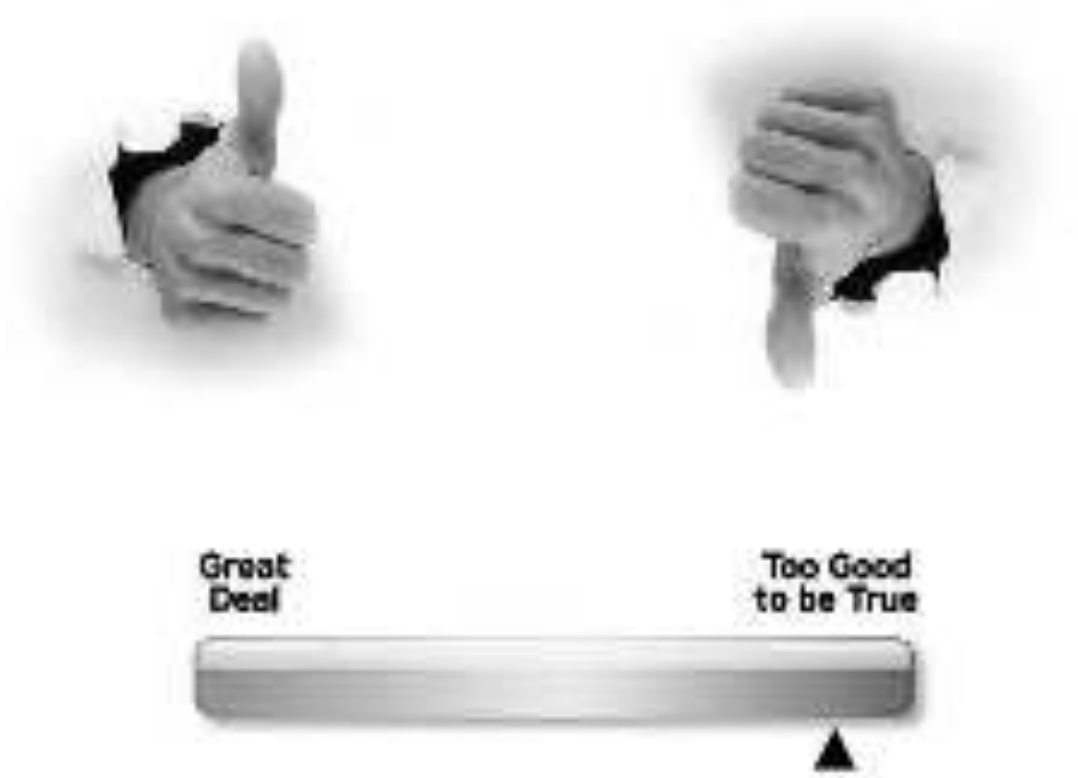
- **Installing/reinstalling** servers or applications using VMs per se ***is not Cloud computing***.
- Let's verify this with the **5 Cloud postulates** shown previously:
  - Self-service, on-demand → **NO** (typically, VMs are provisioned by an IT department)
  - Access through the network → **NO** (deployment limited to "internal customers")
  - Resource pooling → **YES**
  - Elasticity → **NO** (typically, an IT department takes care of installing O/S and software, and not necessarily in a scalable way)
  - Pay-per-use → **NO** (billing is often not according to a pay-as-you-go model, but rather based on traditional flat bills)



Fonte: http://goo.gl/7yHnXB

Cool! I'll buy Cloud computing!

- All good with Clouds?



Some Cloud-related risks

- **Security and privacy**
- **Lock-in**
- **Isolation failure**
- **Management interface compromise**
- **Insecure or incomplete data deletion**

- Some examples taken from a very popular public Cloud follow…

Non-exclusive rights

- **You lose ownership**
  - Amazon (for instance) could develop products directly competing with what you yourself develop on AWS, adopt technologies that you are using…
  - … or assist somebody else in developing products competing with yours.

Example of a typical ToC (Amazon)

13.3 Independent Contractors; Non-Exclusive Rights. We and you are independent contractors, and neither party, nor any of their respective affiliates, is an agent of the other for any purpose or has the authority to bind the other. Both parties reserve the right: (a) to develop or have developed for it products, services, concepts, systems, or techniques that are similar to or compete with the products, services, concepts, systems, or techniques developed or contemplated by the other party and (b) to assist third party developers or systems integrators who may offer products or services which compete with the other party's products or services.

Unavailability

- **Limitations of liability** in case of unavailability of data or services.
  - Due e.g. to power outages, system failures, or to any other service interruption.
  - Or due to unauthorized access, alteration, loss, or anything else of data or any other content stored in AWS.

11. Limitations of Liability.

WE AND OUR AFFILIATES OR LICENSORS WILL NOT BE LIABLE TO YOU FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES, INCLUDING DAMAGES FOR LOSS OF PROFITS, GOODWILL, USE, DATA, OR OTHER INTANGIBLE LOSSES, ARISING OUT OF THE USE OF OR INABILITY TO USE THE SERVICES, ANY SERVICE OR THIRD PARTY CONTENT, OR ANY OTHER CLAIM OF ANY KIND, WHETHER OR NOT ANY OF OUR AFFILIATES OR LICENSORS WILL BE RESPONSIBLE FOR ANY COMPENSATION, REIMBURSEMENT, OR DAMAGES ARISING IN CONNECTION WITH: (A) YOUR INABILITY TO USE THE SERVICES, INCLUDING AS A RESULT OF ANY (I) TERMINATION OR SUSPENSION OF THIS AGREEMENT OR YOUR USE OF OR ACCESS TO THE SERVICE OFFERINGS, (II) OUR DISCONTINUATION OF ANY OR ALL OF THE SERVICE OFFERINGS, OR (III) WITHOUT LIMITING ANY OBLIGATIONS UNDER THE SLAS, ANY OTHER INTERRUPTION OR ANOMALY IN ALL OR ANY PORTION OF THE SERVICES FOR ANY REASON, INCLUDING AS A RESULT OF POWER OUTAGES, SYSTEM FAILURES OR OTHER INTERRUPTIONS; (B) THE COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; (C) ANY INVESTMENTS, EXPENDITURES, OR COMMITMENTS BY YOU IN CONNECTION WITH THIS AGREEMENT OR YOUR USE OF OR ACCESS TO THE SERVICE OR ANY SERVICE OFFERING; (D) ANY UNAUTHORIZED ACCESS TO, ALTERATION OF, OR THE DELETION, DESTRUCTION, DAMAGE, LOSS OR FAILURE TO STORE ANY DATA OR OTHER CONTENT; AND (E) THE DURATION OR FREQUENCY OF ANY USE OF THE SERVICES BY YOU. IN ANY CASE, OUR AND OUR AFFILIATES AND LICENSORS' AGGREGATE LIABILITY UNDER THIS AGREEMENT WILL BE LIMITED TO THE AMOUNT YOU ACTUALLY PAY US UNDER THIS AGREEMENT FOR THE SERVICE THAT GAVE RISE TO THE CLAIM DURING THE 12 MONTHS PRECEDING THE CLAIM.

No guarantees

- **Disclaimers = no guarantee** that the service will be "uninterrupted, error free or free of harmful components".
  - Or that what is stored in AWS is safe, is not lost, or damaged.
  - What if I decided to use AWS to store my scientific data (some tens of PB maybe…)

10. Disclaimers.

THE SERVICE OFFERINGS ARE PROVIDED "AS IS." WE AND OUR AFFILIATES AND LICENSORS MAKE NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE REGARDING THE SERVICE OFFERINGS OR THE THIRD PARTY CONTENT, INCLUDING ANY WARRANTY THAT THE SERVICE OFFERINGS OR THIRD PARTY CONTENT WILL BE UNINTERRUPTED, ERROR FREE OR FREE OF HARMFUL COMPONENTS, OR THAT ANY CONTENT, INCLUDING YOUR CONTENT OR THE THIRD PARTY CONTENT, WILL BE SECURE OR NOT OTHERWISE LOST OR DAMAGED. EXCEPT TO THE EXTENT PROHIBITED BY LAW, WE AND OUR AFFILIATES AND LICENSORS DISCLAIM ALL WARRANTIES, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR QUIET ENJOYMENT, AND ANY WARRANTIES ARISING OUT OF ANY COURSE OF DEALING OR USAGE OF TRADE.

But you are responsible

- **You are responsible to make sure your data, code, etc. is safe, protected from unauthorized access, and *you are responsible for your own backup*** (again – with if it's in the order of several PB?)

4.2 Other Security and Backup. You are responsible for properly configuring and using the Service Offerings and taking your own steps to maintain appropriate security, protection and backup of Your Content, which may include the use of encryption technology to protect Your Content from unauthorized access and routine archiving Your Content. AWS log-in credentials and private keys generated by the Services are for your internal use only and you may not sell, transfer or sublicense them to any other entity or person, except that you may disclose your private key to your agents and subcontractors performing work on your behalf.

Data property / privacy?

- When a contract with a Cloud provider gets cancelled, how can we make sure that **all our data is removed**?
- And how can I avoid ***vendor lock-in***?
- But where is my data? How about ***tapping***?

Microsoft admits Patriot Act can access EU-based cloud data

NSA infiltrates links to Yahoo, Google data centers worldwide, Snowden documents say

http://backgroundchecks.org/justdeleteme/

Some of the sites “impossible to be deleted from” (the ones in **black**):

- Backblaze (Cloud backup)
- Blogger, Picasa (and other Google services)
- eDreams
- Evernote
- Netflix
- OpenShift
- Pastebin
- PlayStation Network
- Slashdot
- Udacity
- Wikipedia
- Wordpress.com

Maturity

- Cloud computing is a set of modern technologies **sometimes not entirely stable**.
  - Tuning and experts are often needed.
  - And sometimes, in order to avoid complex configurations (even in public Clouds) shortcuts are taken…
    - For example for what regards tenant isolation, or the application of security patches.
  - … on the other hand, in order to address complex problems, sometimes “non production-ready” solutions are offered.

Last but not least, the big misunderstanding

- **Capacity is not infinite** (although this is one of the postulates of Cloud computing). *Nor are credit card limits*.
  - Hence, resources might not be available when we need them; or, if available, they might not have the characteristics we need.
  - Unless maybe we are willing to pay some hefty over-provisioning costs.

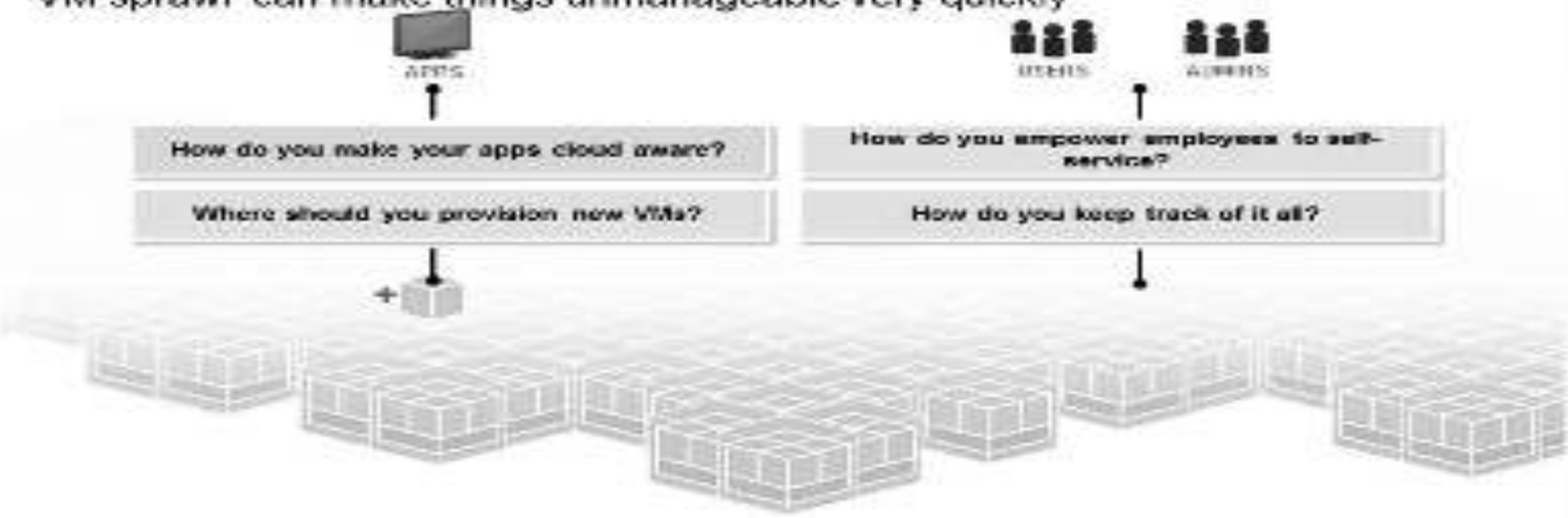
But we could build a private Cloud!

- Assuming we are able to build it ourselves, shouldn't everything be under our control then?


Datacenters are being virtualized, Servers are first
Hypervisors provide abstraction between SW and HW (Servers)
HOST 1
HOST 2
HOST 3
HOST 4, ETC.
VMs
Hypervisor
Hardware abstraction for each server
Better resource utilization for each server
Next: Storage, Network…the building blocks
Compute Pool
Virtualized Servers
Network Pool
Virtualized Networks
Storage Pool
Virtualized Storage
Flexibility, Efficiency are key drivers
Resource pools for apps starting to form…
2. Cloud Data Center
3. Cloud Federation

But questions arise as the environment grows…

"VM sprawl" can make things unmanageable very quickly

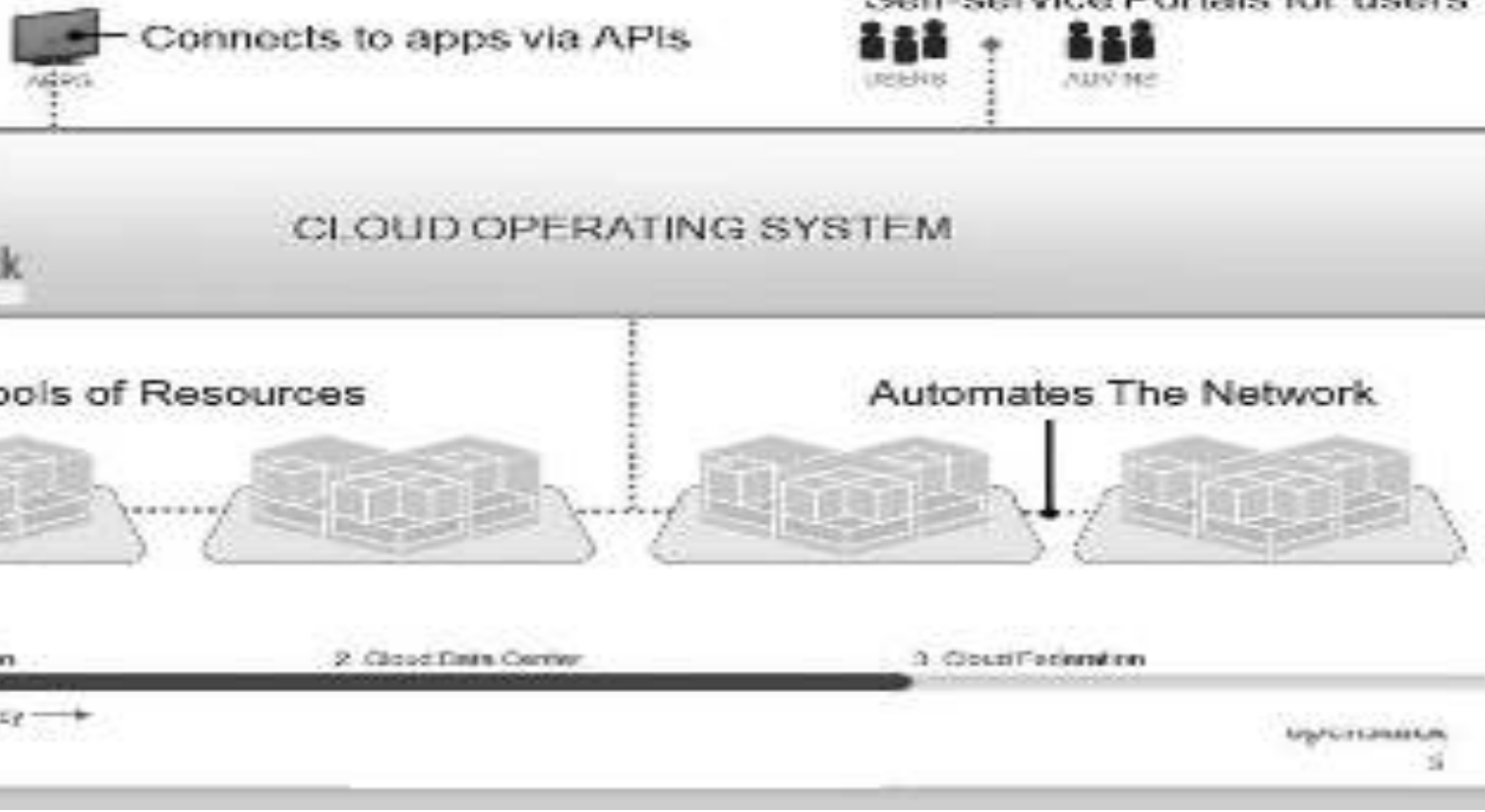


A Cloud Management Layer Is Missing

CloudEnablers

Solution: OpenStack, The Cloud Operating System

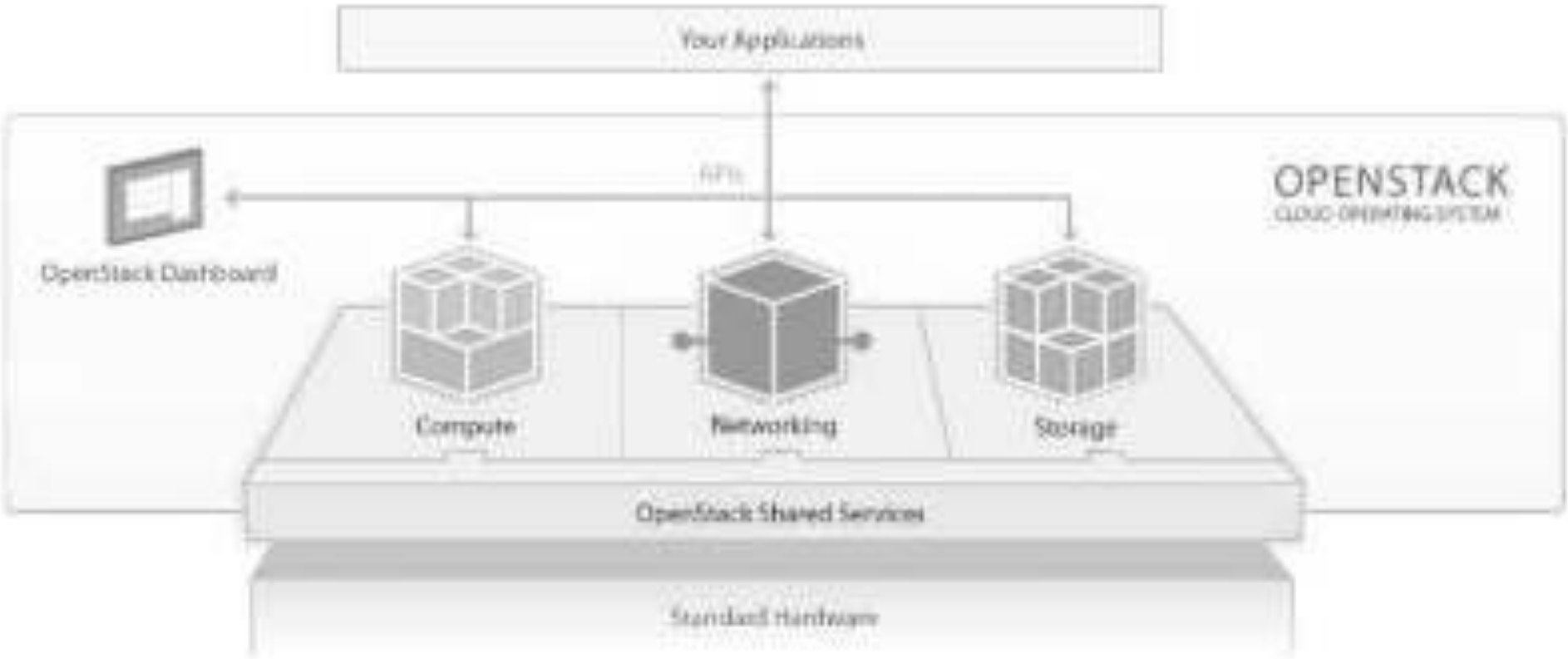
A new management layer that adds automation and control



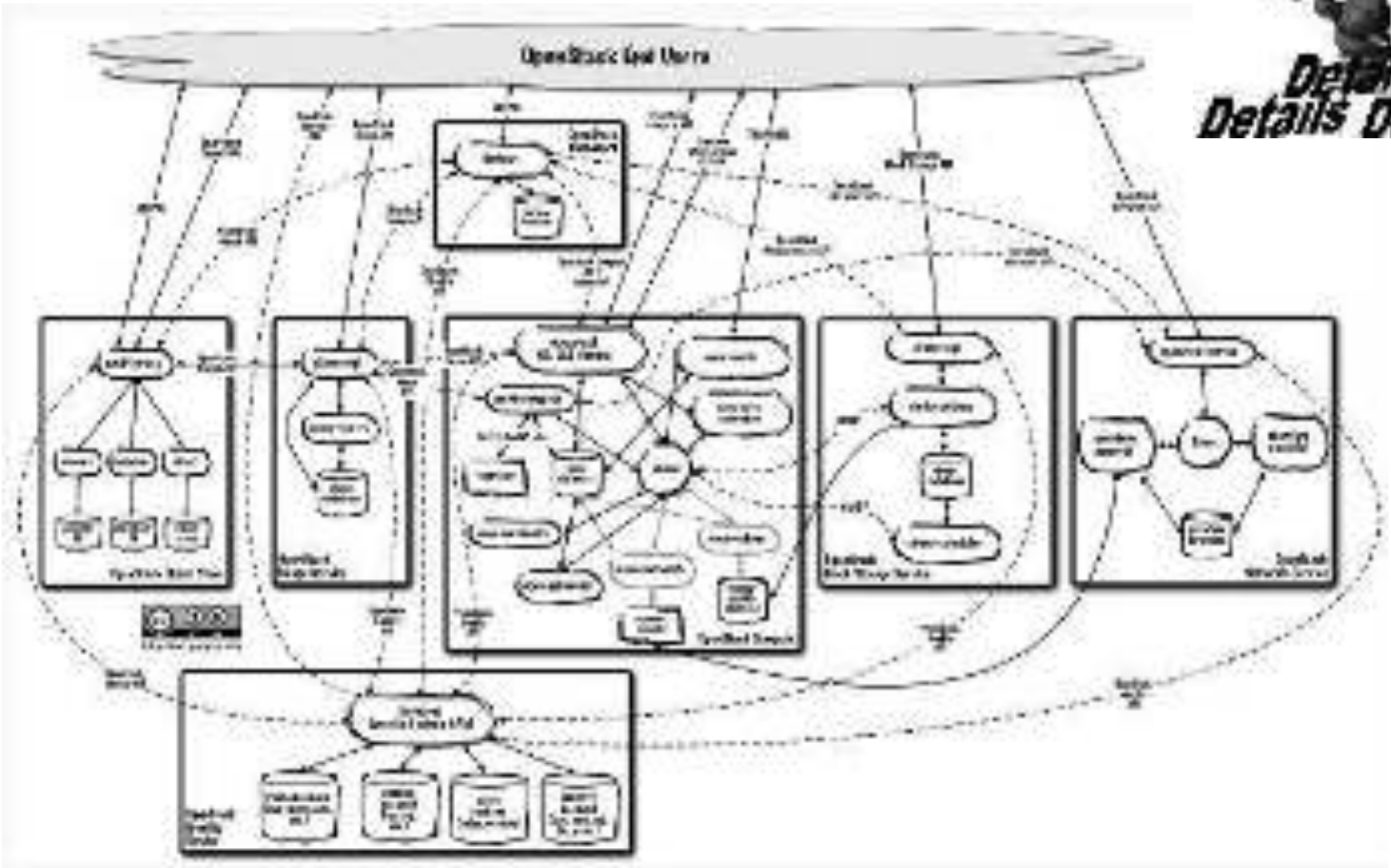
OpenStack (www.openstack.org)

- It is an **open source Cloud Management Framework** for building public and private Cloud infrastructures, **mostly at the IaaS (infrastructural) level.**
- Born in 2010 as an initiative among NASA and Rackspace, it now counts 600+ supporting companies, is deployed in 180+ countries, has 20+ million lines of code.

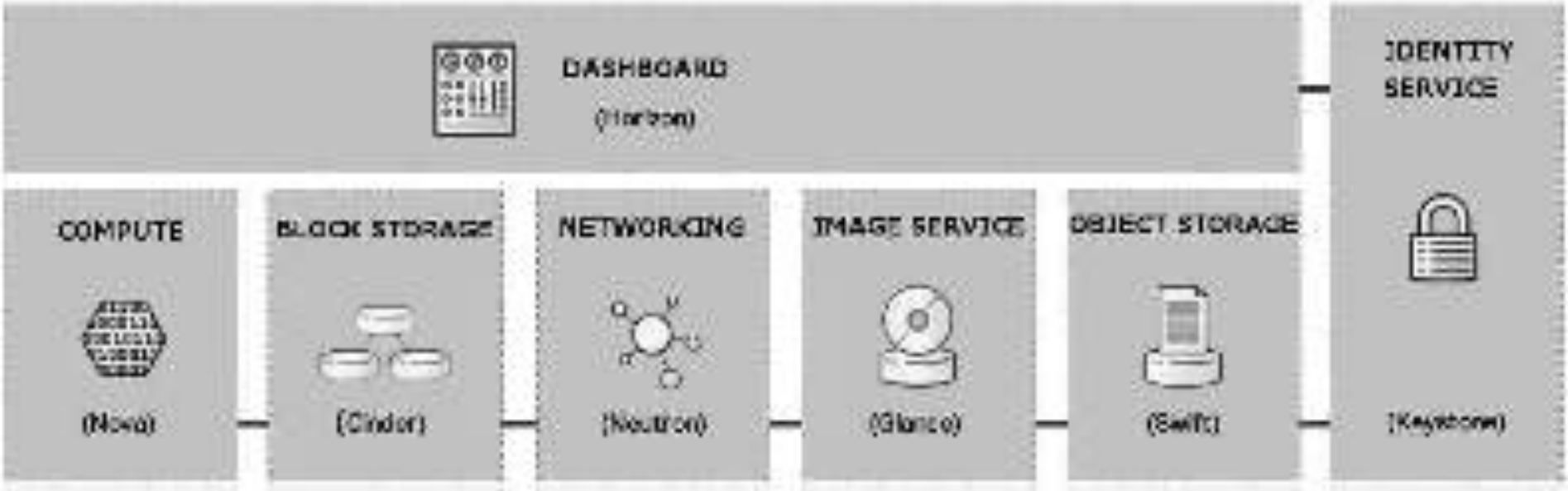
OpenStack, a high-level view



OpenStack, a not-so-high-level view (aka “the devil is in the details”)



OpenStack components (1)



OpenStack components (2)

- Most OpenStack components map to a "*-as-a-Service" capability. For example:

*-as-a-Service Capability	OpenStack component
Compute	Nova
Storage	Swift (object)
	Manila (file)
	Cinder (block)
	Glance (images)
Identity	Keystone
Network	Neutron

Storage types

On-instance / ephemeral	Block storage (cinder)	Object Storage (swift)	File Storage (manila)
Runs operating systems and provides scratch space	Used for adding additional persistent storage to a virtual machine (VM)	Used for storing virtual machine images and data	Used for providing file shares to a virtual machine
Persists until VM is terminated	Persists until deleted	Persists until deleted	Persists until deleted
Access associated with a VM	Access associated with a VM	Available from anywhere	Access can be provided to a VM
Implemented as a filesystem underlying OpenStack Compute	Mounted via OpenStack Block storage controlled protocol (for example, iSCSI)	REST API	Provides Shared File System service via nfs, cifs, glusterfs, or hdfs protocol
Encryption is available	Encryption is available	Work in progress - expected for the Mitaka release	Encryption is not available yet
Administrator configures size setting, based on flavors	Sizings based on need	Easily scalable for future growth	Sizing based on need
Example: 10 GB first disk, 30 GB/core second disk	Example: 1 TB "extra hard drive"	Example: 10s of TBs of data set storage	Example: 1 TB of file share

OpenStack Hype

- HP announced in May 2014 an investment of more than $1 billion over the next two years on cloud-related products and services around OpenStack. One year and a half later, in October 2015, HP threw in the towel on public cloud, announcing that HP Helion Cloud is ending operations from 31st January 2016. HP has recently announced partnership agreements with AWS and Microsoft Azure.

- Rackspace, one of the two founders of OpenStack, announced in August 2014 that it was discontinuing its pure Infrastructure as a Service (IaaS) offering, and in October 2015 they announced a partnership with AWS. The results? the shares of RackSpace have fallen to lowest level in 5+years. Investors don't seem very optimistic with its OpenStack strategy.

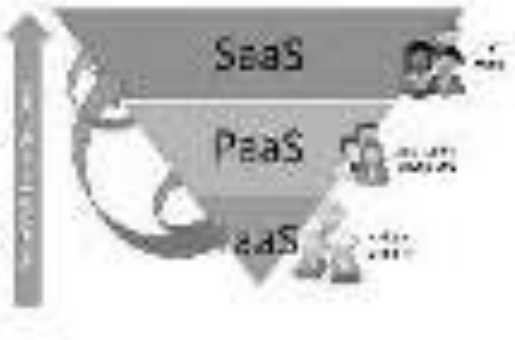
Canonical founder Mark Shuttleworth predicts total collapse of OpenStack 'big tent'

Canonical founder Mark Shuttleworth has described the vast majority of vendors touting their work with OpenStack as "bullshit as a service" – and warned that the 'big tent' to which they belong is due for collapse.

Tamlin Magee
April 27, 2016

So, what is OpenStack useful for?

- Remember, OpenStack was born and still is a Cloud management framework **targeted at the IaaS level (infrastructure).**
- So, as a user you could easily instantiate e.g. virtual machines from an OpenStack-based cloud.
- What next?



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

Moving beyond IaaS

Source: OpenShift blog

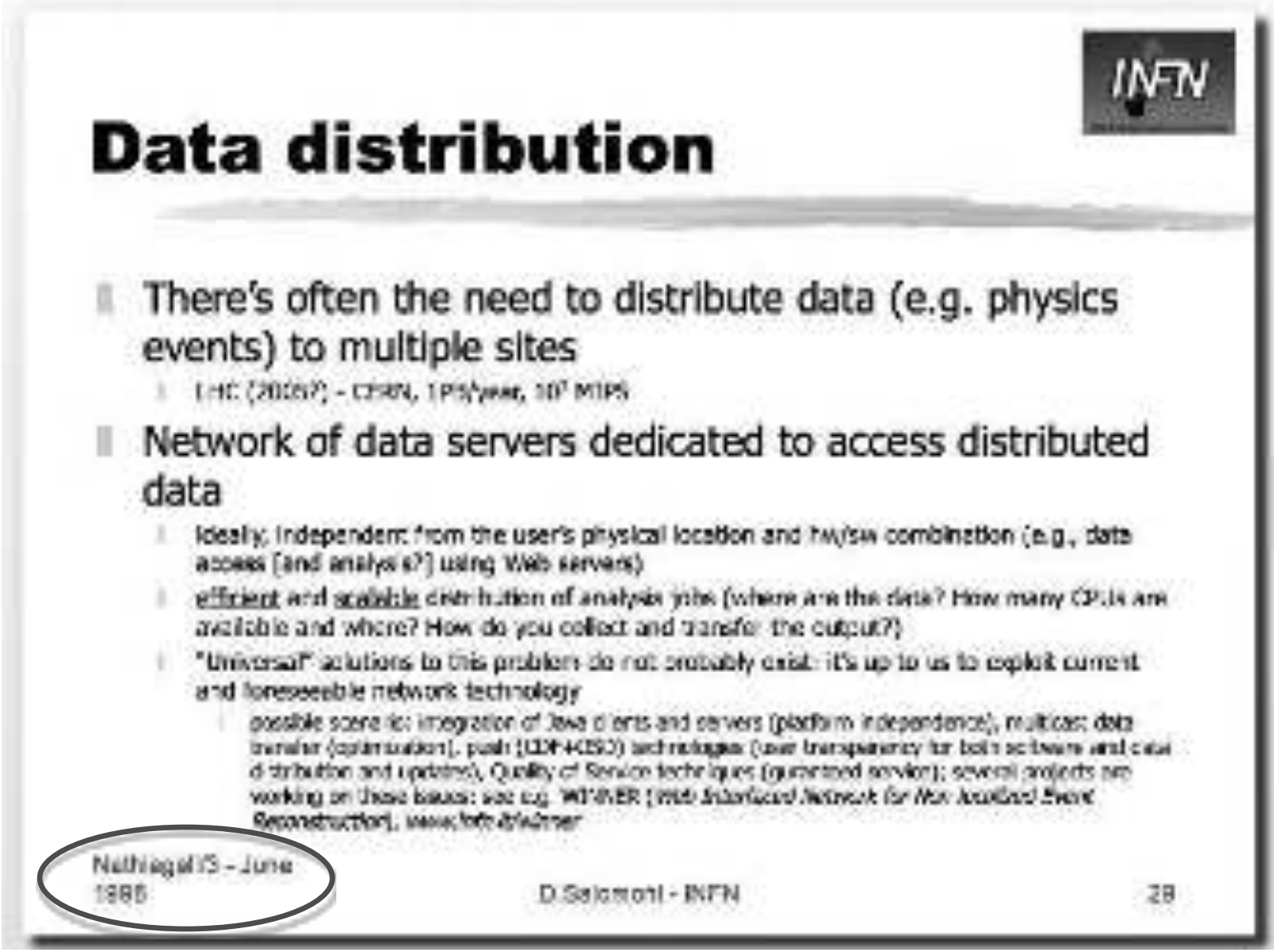
Cloud: anything missing?

- Open **interoperation** / federation across (proprietary) CLOUD solutions at
  - IaaS,
  - PaaS,
  - and SaaS levels
- Managing **multitenancy**
  - At large scale…
  - … and in heterogeneous environments
- Dynamic and seamless **elasticity**
  - For both private and public cloud…
  - … and for complex or infrequent requirements
- **Data management** in a Cloud environment
  - Due to technical…
  - … as well as to legal problems

Filling these gaps should lead to:

- **Interoperable PaaS/SaaS services addressing both public and private Cloud infrastructures.**
- **Migration of legacy applications to the Cloud.**
- **Increased focus on user-oriented, high-value solutions.**

A flashback from a while ago…



Data distribution

- There's often the need to distribute data (e.g. physics events) to multiple sites
  - LHC (2005?) - CERN, 1 PB/year, 10^7 MIPS
- Network of data servers dedicated to access distributed data
  - Ideally: independent from the user's physical location and hw/sw combination (e.g., data access [and analysis?] using Web servers)
  - efficient and scalable distribution of analysis jobs (where are the data? How many CPUs are available and where? How do you collect and transfer the output?)
  - "Universal" solutions to this problem do not probably exist: it's up to us to exploit current and foreseeable network technology
    - possible scenario: integration of Java clients and servers (platform independence), multicast data transfer (optimization), push (CORBA/IIOP) technologies (user transparency for both software and data distribution and updates), Quality of Service techniques (guaranteed service); several projects are working on these issues: see e.g. WIBNER (*Web Interfaced Network for Mix Assisted Event Reconstruction*), www.infn.it/wibner

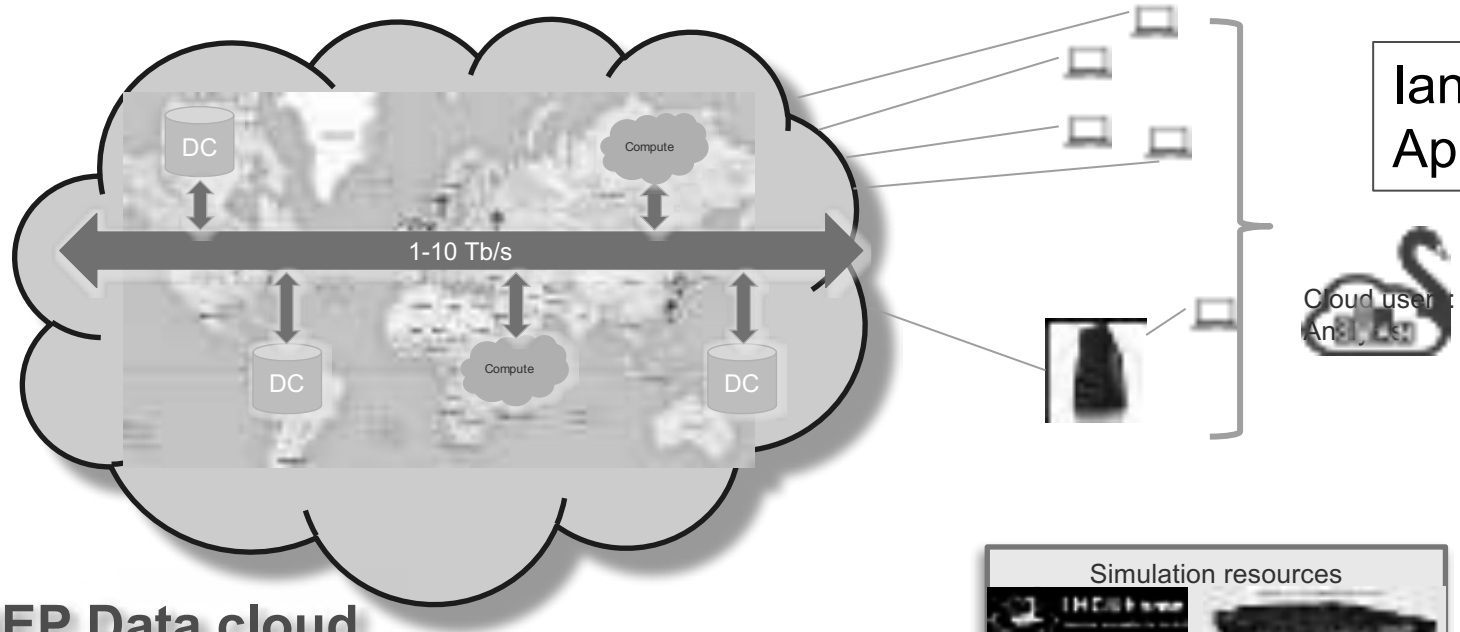
Nathiagali'95 - June 1995

D.Salomoni - INFN

28

Back to the future…

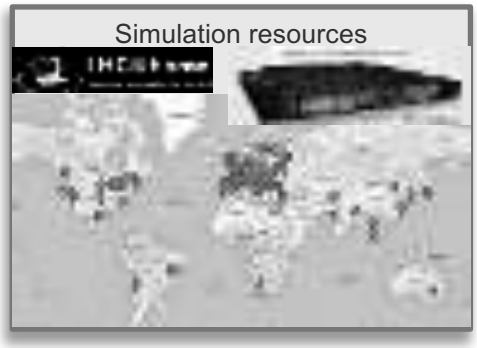
Possible Model for future HEP computing infrastructure



Ian Bird, CERN, April 2017

HEP Data cloud

Storage and compute



Nihil sub sole novum.

Eccl. 1:10

From the Paper “Advances in Cloud”

- **EC Expert Group Report on Cloud Computing**, http://cordis.europa.eu/fp7/ict/ssai/docs/future-cc-2may-finalreport-experts.pdf

To reach the full promises of CLOUD computing, major aspects have not yet been developed and realized and in some cases not even researched. Prominent among these are **open interoperation across (proprietary) CLOUD solutions at IaaS, PaaS and SaaS levels**. A second issue is **managing multitenancy** at large scale and in heterogeneous environments. A third is **dynamic and seamless elasticity** from in- house CLOUD to public CLOUDs for unusual (scale, complexity) and/or infrequent requirements. A fourth is **data management in a CLOUD environment**: bandwidth may not permit shipping data to the CLOUD environment and there are many associated legal problems concerning security and privacy. All these challenges are opportunities towards a more powerful CLOUD ecosystem.

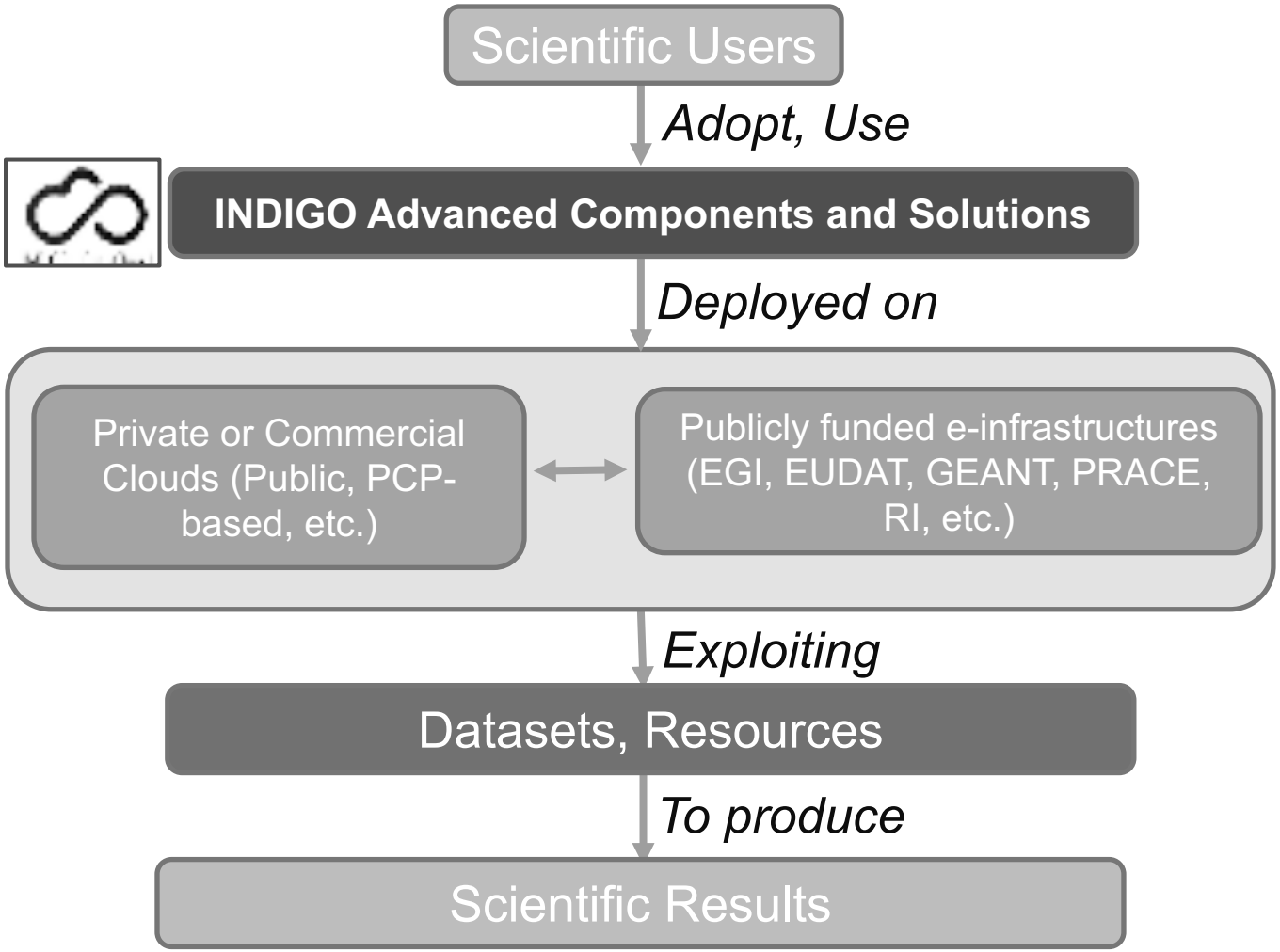
[…] **A major opportunity for Europe involves finding a SaaS interoperable solution across multiple CLOUD platforms.**

Another lies in migrating legacy applications without losing the benefits of the CLOUD, i.e. exploiting the main characteristics, such as elasticity etc.

INDIGO-DataCloud

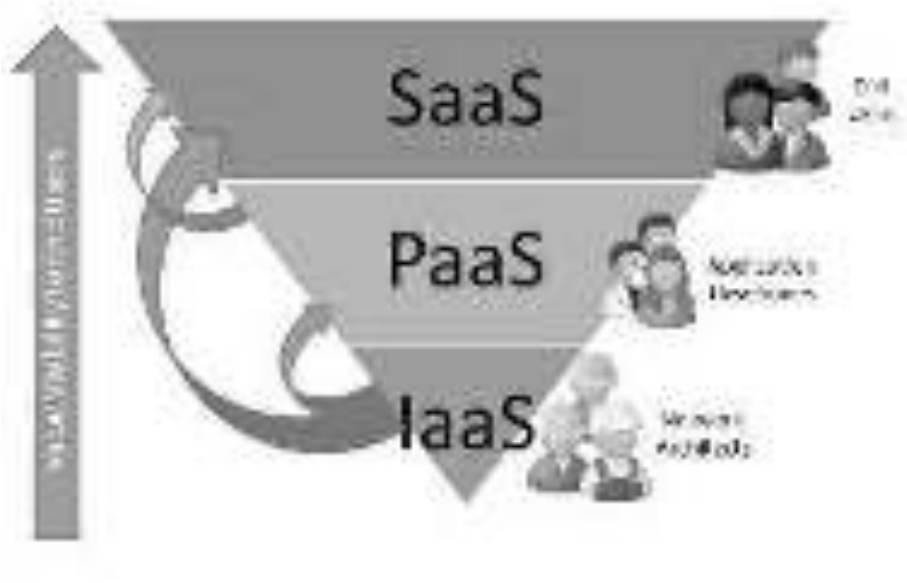
- **An H2020 project** approved in January 2015 in the EINFRA-1-2014 call
  - 11.1M€, 30 months (**from April 2015 to September 2017**)
- **Who**: **26 European partners** in 11 European countries
  - Coordination by the Italian National Institute for Nuclear Physics (INFN)
  - Including developers of distributed software, industrial partners, research institutes, universities, e-infrastructures
- **What**: **develop an open source Cloud platform** for computing and data ("DataCloud") tailored to science.
- **For**: **multi-disciplinary scientific communities**
  - E.g. structural biology, earth science, physics, bioinformatics, cultural heritage, astrophysics, life science, climatology
- **Where**: deployable on **hybrid (public or private) Cloud infrastructures**
  - INDIGO = **IN**tegrating **D**istributed data **I**nfrastructures for **G**lobal Expl**O**itation
- **Why**: answer to the technological **needs of scientists** seeking to easily exploit distributed Cloud/Grid compute and data resources.

INDIGO produces software components for exploitation of Cloud resources



Cloud enhancements…

- … needed at several levels:
  - At the authentication / authorization level (AAI)
  - At the site level (IaaS)
  - For a true interoperable, programmable Cloud (PaaS)
  - For the user front-ends (SaaS+)



AAI main requirements

- **Authentication**
  - Support for **federated AuthN** & social logins
- **Identity Harmonization**
  - Link multiple accounts to a single identity, providing **a persistent identifier**
- **Authorization**
  - Attribute-based, dynamic
  - Consistent across heterogeneous infrastructures
- **Delegation**
  - Provide the **ability for services to act on behalf of a user**
  - Support **offline access** for long-running applications
- **Provisioning**
  - Provision/de-provision identities to services or relying resources
- **Token translation**
  - Enable **integration with services relying on heterogeneous AuthN mechanisms**

At the IaaS level

- Provide **support for containers** in addition to the already existing VM support at the IaaS layer
  - Support for selected Cloud Management Frameworks.
  - Using *open standard* interfaces.
  - With automatic repository synchronization.
- Improve **on-demand compute capabilities** at the local datacenters
  - Improving cloud scheduling, a mandatory requirement for production systems.
  - Facilitating container execution in HPC and HTC systems.
  - Providing site-level orchestration.
  - Implement storage-independent storage QoS

At the PaaS level

- **Many features** are needed at this level to support full exploitation of hybrid clouds and programmatic usage of cloud resources:
  - An **Orchestrator Service**, coupled with some **ranking** of cloud providers
  - A **Cloud Information Provider** handling images / containers
  - A **Monitoring Service**
  - An **Accounting Service**
  - A tool to **deploy virtual infrastructures** over multiple back-ends (clouds)
  - A service to **manage applications** deployed on the cloud
  - A **configuration management database**
  - An **SLA/QoS Manager**
  - An **elasticity manager** for cloud infrastructures
  - Flexible **data services**
  - A way to manage all the components above ☺

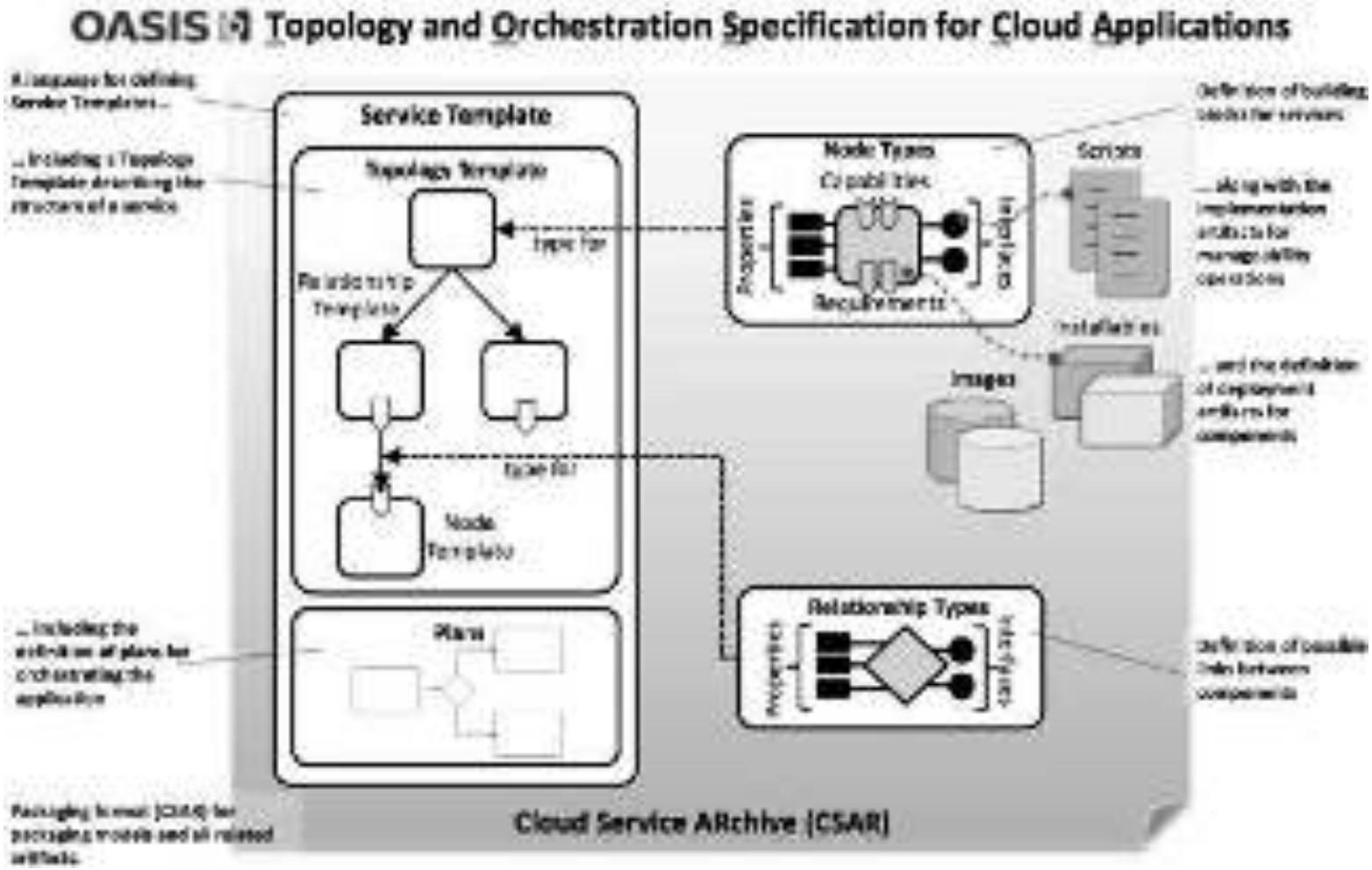
The INDIGO PaaS Orchestrator

- It **takes deployment requests**, expressed through templates **written in the TOSCA language**, and deploys them **on the best cloud site** available.
- The Orchestrator **collects all the information needed** to deploy a service consuming others PaaS APIs:
  - QoS/SLA Service: get the prioritized list of SLAs of the user
  - Monitoring Service: get the status of the underlying IaaS platforms and their resource availability;
  - CloudProviderRanker Service: sort the list sites on the basis of rules defined per user/group/use-case;
- The orchestrator then **delegates the deployment** to some other components interfacing with Cloud infrastructures.
- This is an essential components to manage multiple clouds.

TOSCA

- **T**opology and **O**rchestration **S**pecification for **C**loud **A**pplications
- Standardizes a language to describe:
  - The structure of an IT service (its **topology** model)
  - How to orchestrate operational behavior (plans such as build, deploy, patch, shutdown, etc.)
- It's a declarative model that spans applications, virtual and physical infrastructures.

TOSCA in a nutshell



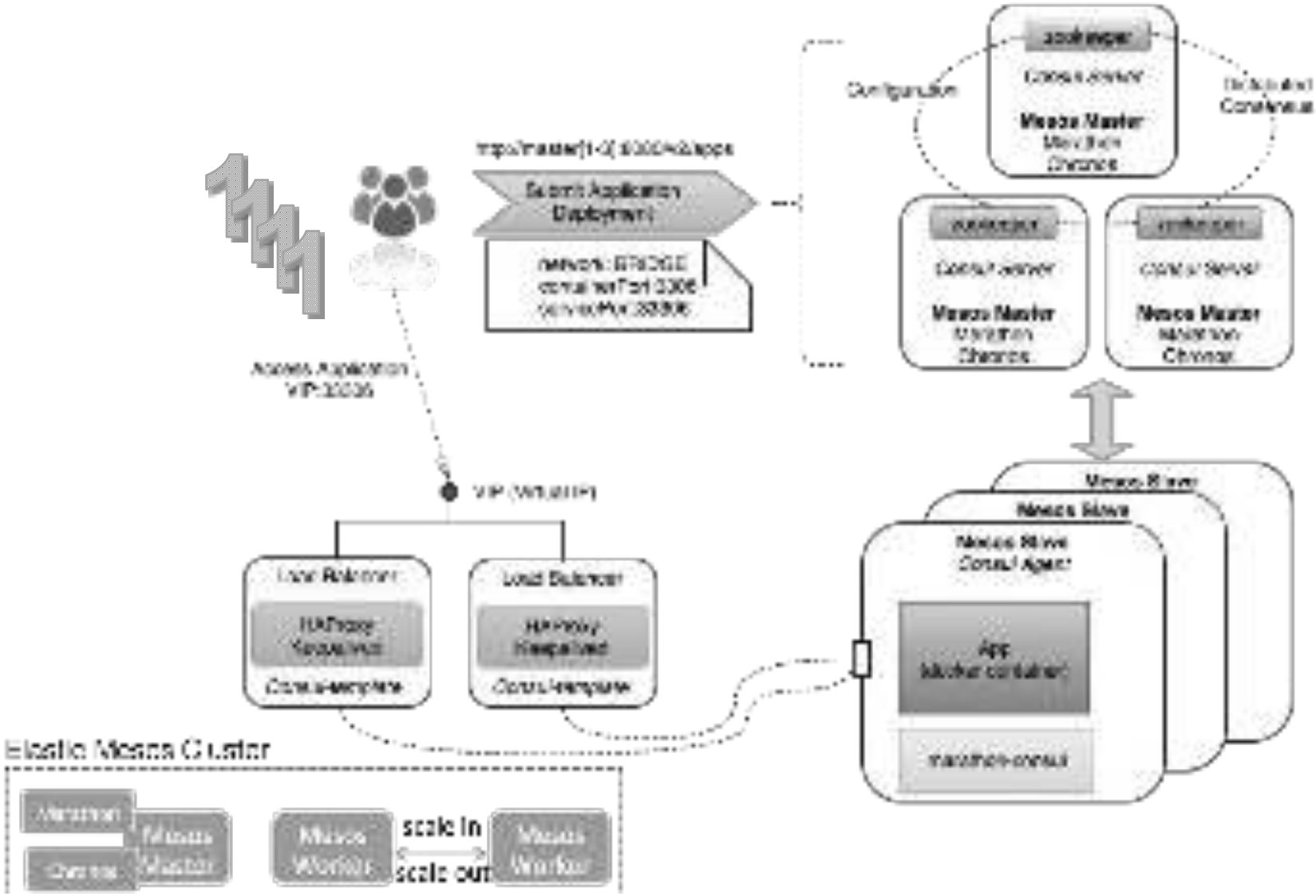
TOSCA topology template example

```
tosca_definitions_version: tosca_simple_yaml_1_0

description: Template for deploying a single server with predefined properties.

topology_template:
  inputs:
    cpus:
      type: integer
      description: Number of CPUs for the server.
      constraints:
        - valid_values: [ 1, 2, 4, 8 ]

  node_templates:
    my_server:
      type: tosca.nodes.Compute
      capabilities:
        # Host container properties
        host:
          properties:
            # Compute properties
            num_cpus: { get_input: cpus }
            mem_size: 2048 MB
            disk_size: 10 GB

  outputs:
    server_ip:
      description: The private IP address of the provisioned server.
      value: { get_attribute: [ my_server, private_address ] }
```

The INDIGO Application Deployment Service

- This is the service in charge of **scheduling, spawning, executing and monitoring** applications and services on a distributed infrastructure.
- The core of this component consists of an elastic **Apache Mesos** [http://mesos.apache.org] cluster with slave nodes dynamically provisioned and distributed on IaaS sites (almost any IaaS! OpenStack, OpenNebula, Amazon, Azure)
- Apache Mesos provides efficient resource isolation and sharing across distributed applications.
- The INDIGO PaaS then uses these Mesos frameworks:
  - **Marathon** to deploy, monitor and scale **Long-Running services**, ensuring that they are always up and running.
  - **Chronos** to run user **applications** (jobs), taking care of fetching input data, handling dependencies among jobs, rescheduling failed jobs.

An INDIGO Mesos Cluster

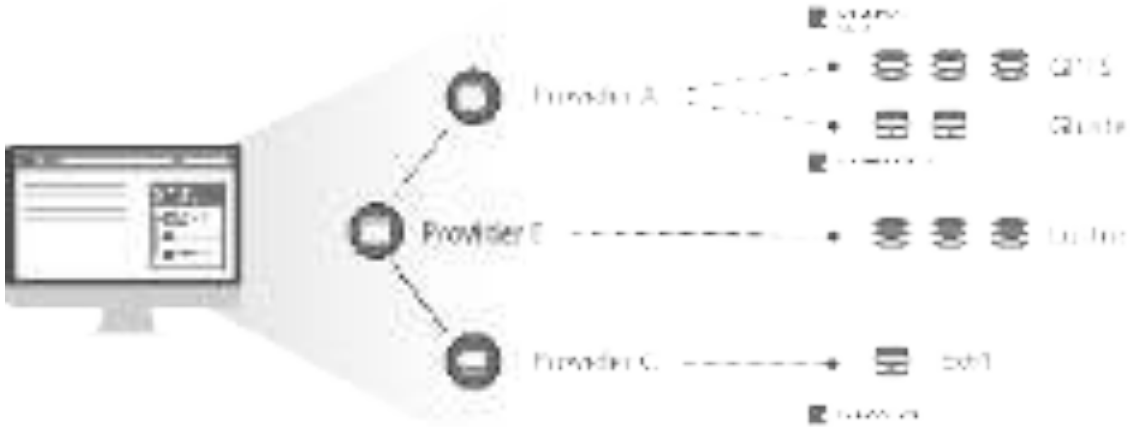


This gives us…

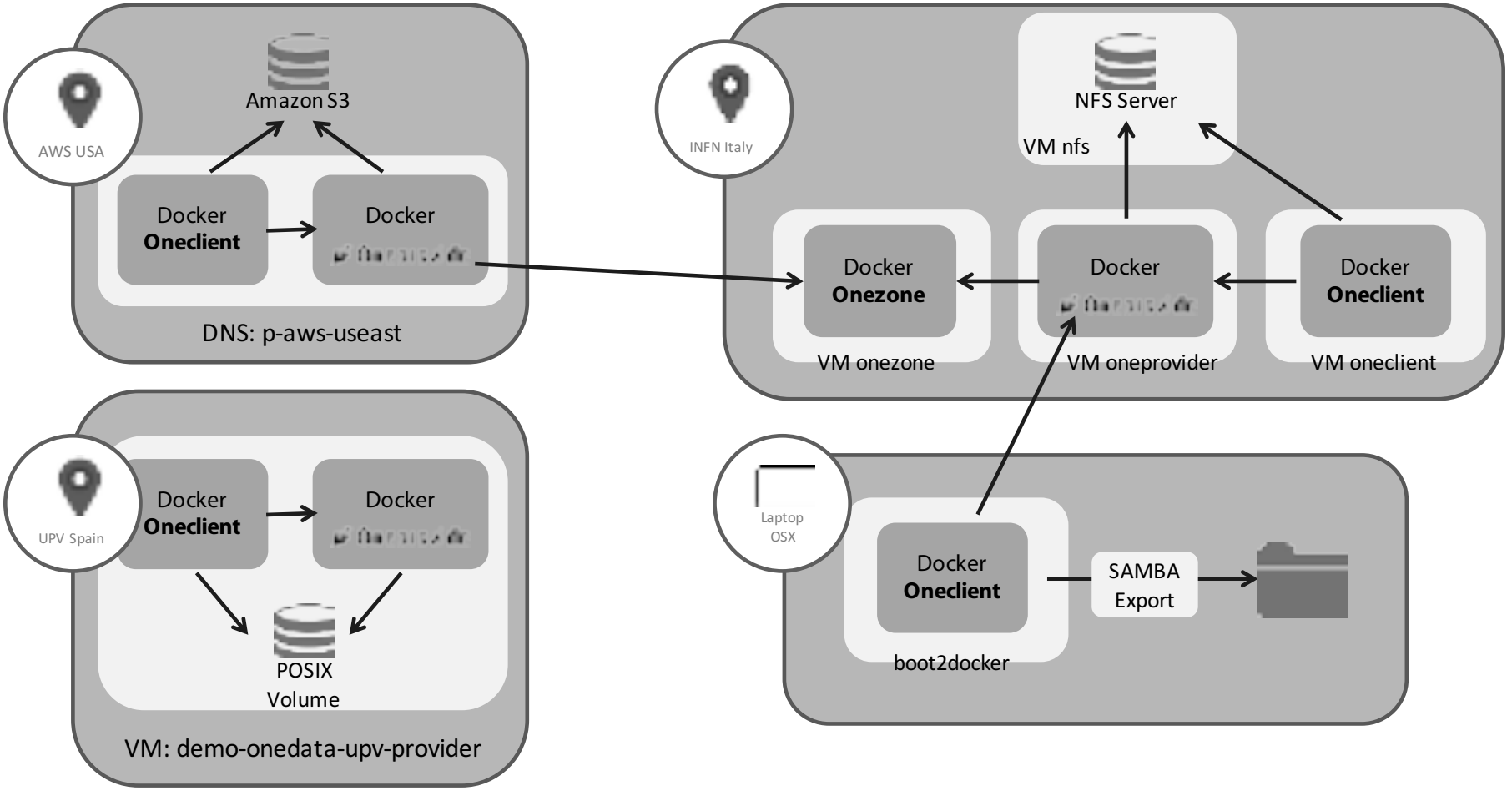
- The possibility to have **batch execution of dockerized applications** exploiting Mesos/Chronos capabilities and the INDIGO virtual storage solutions (see later)
- On-demand provisioning of full **batch system clusters** exploiting Mesos/Marathon
  - E.g. HTCondor on Mesos: all the services (worker nodes, schedd, etc.) are run as docker containers and automatically configured by INDIGO components
  - Automatic scaling of services and resources

Data Services: Onedata

- **Unified data access** over heterogeneous infrastructures
- **High-performance** data **access** and **migration**
- **POSIX interface** for accessing large data sets without pre-staging them
- Flexible security framework based on **access tokens** and **ACL's**
- **Metadata editing** and querying using **key-value** pairs, **JSON** and **RDF**
- Support for **POSIX**, **Ceph**, OpenStack **SWIFT** and Amazon **S3**
- **Graphical User Interface** for easy data management
- Comprehensive **REST API** for integrating with other services



Data Federation through INDIGO Onedata



ElectricIndigo

- **NEW**: our second and final major software release, called **ElectricIndigo**
  - For technical details, see the parallel sections **on Thursday**
- **Fact sheet** (https://www.indigo-datacloud.eu/service-component):
  - 40 modular components, distributed via 170 software packages, 50 ready-to-use Docker containers
  - Operating systems: CentOS 7, Ubuntu 16.04
  - Cloud frameworks: OpenStack Newton, OpenNebula 5.x
  - Download it from the INDIGO-DataCloud Software Repository: http://repo.indigo-datacloud.eu/index.html

ElectricIndigo:
Application-level Interfaces for Cloud Providers and Automated Service Composition

- Fasily **port applications to public and private Clouds** using open programmable interfaces, user-level cortainers, and standards-based languages to **automate definition, composition and instantiation of complex set-ups**.

- **Typical questions**: How can I run my application on Cloud provider X? What if I want to use Docker but my provider does not support it? How do I automate the creation and management over public or private Clouds of dynamic clusters running multiple services?

ElectricIndigo:
Flexible Identity and Access Management

- **Manage access and policies to distributed resources** using multiple methods such as **OpenID-Connect, SAML, X.509** digital certificates, through **programmable interfaces and web front-ends.**

- **Typical questions**: How can I manage access to distributed resources by users, identified through diverse methods? (e.g. Google ID, digital certificates) How should I modify / write my apps to benefit from that?

ElectricIndigo:
Data Management and Data Analytics Solutions

- **Distribute and access data** through multiple providers via **virtual file systems and automated replication and caching**, exploiting scalable, **high-performance data mining and analytics**.

- **Typical questions**: How can I automatically replicate datasets to multiple sites? Can I transparently access my distributed datasets from my app? Can I cache the most accessed data, so that it's close to where users need it? How do I instantiate clusters and databases for big data analysis?

ElectricIndigo:
Programmable Web Portals, Mobile Applications

- **Create and interface web portals or mobile apps**, exploiting **distributed data as well as compute resources** located in public and private Cloud infrastructures.

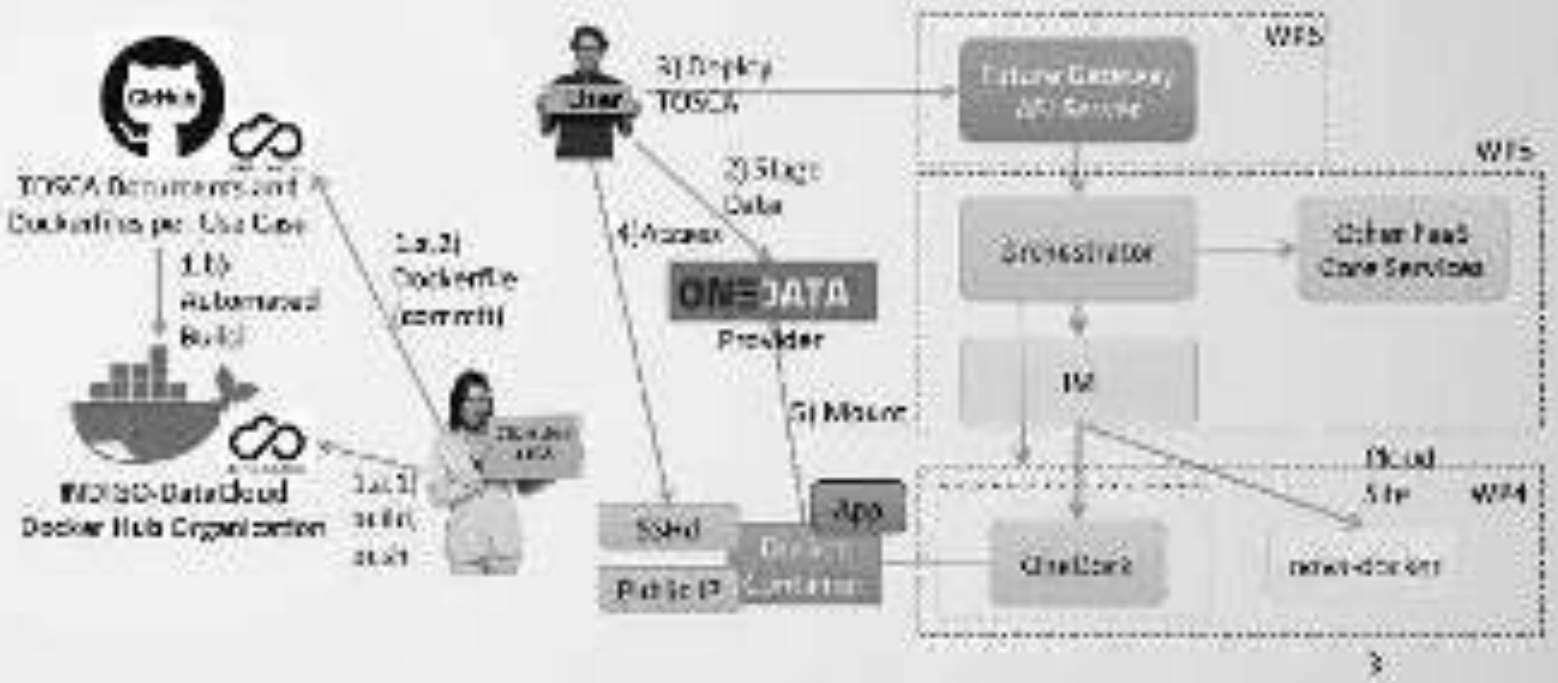
- **Typical questions**: How can I easily provide my app with a pluggable, extensible web front-end? Can this front-end interface with all the features provided by INDIGO? How can I write an INDIGO-enabled app for Android or iOS?

ElectricIndigo:
Enhanced and Scalable Services for Data Centers and Resource Providers

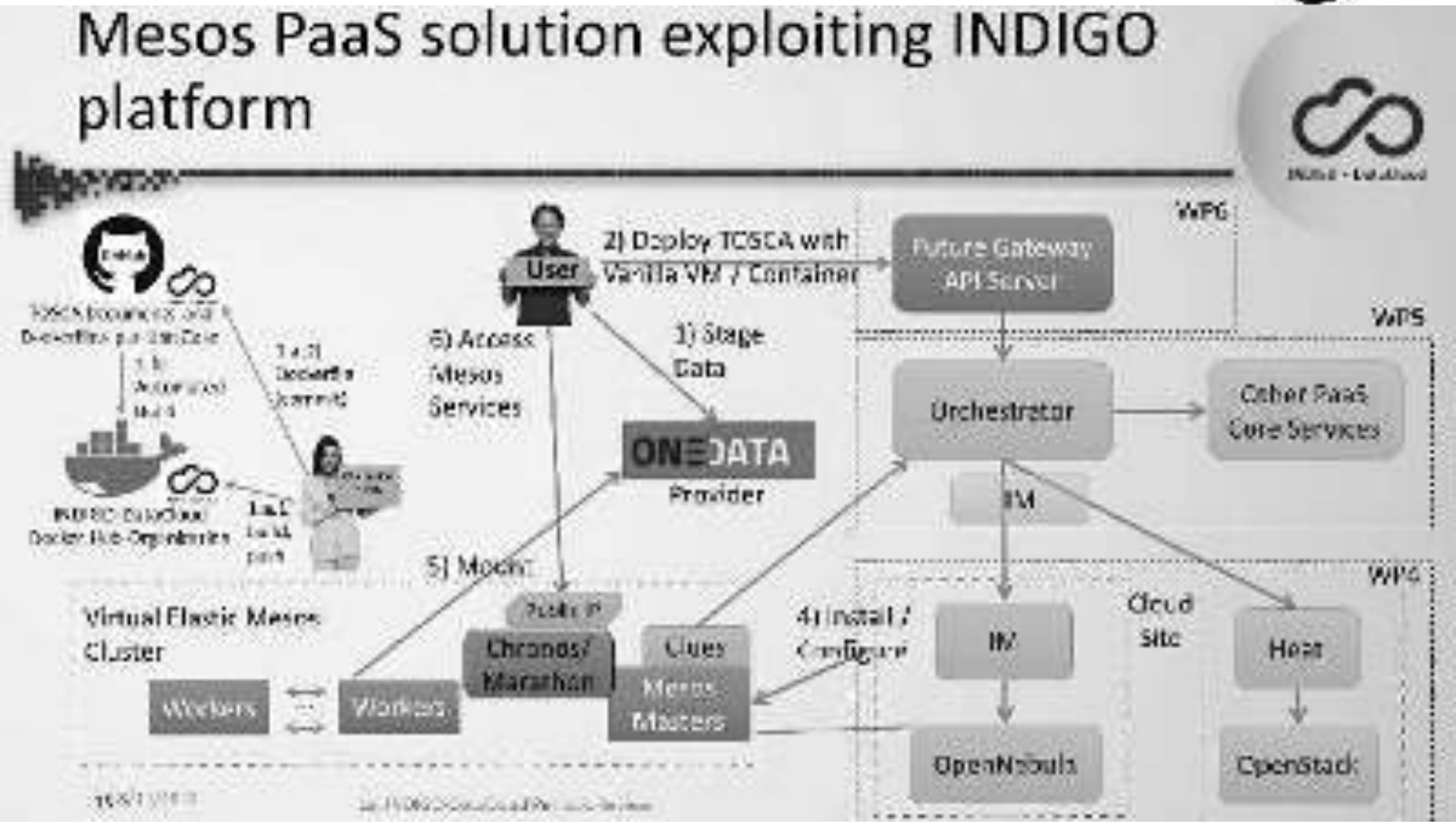
- **Increase the efficiency of existing Cloud infrastructures** based on OpenStack or OpenNebula through **advanced scheduling, flexible cloud / batch management, network orchestration** and interfacing of high-level Cloud services to existing storage systems.

- **Typical questions**: How can my cloud data centers provide flexible and fair scheduling policies for access to resources? How do I balance traditional vs. cloud resources in my data center? How do I connect novel INDIGO features to my existing systems? How can I manage storage Quality of Service?

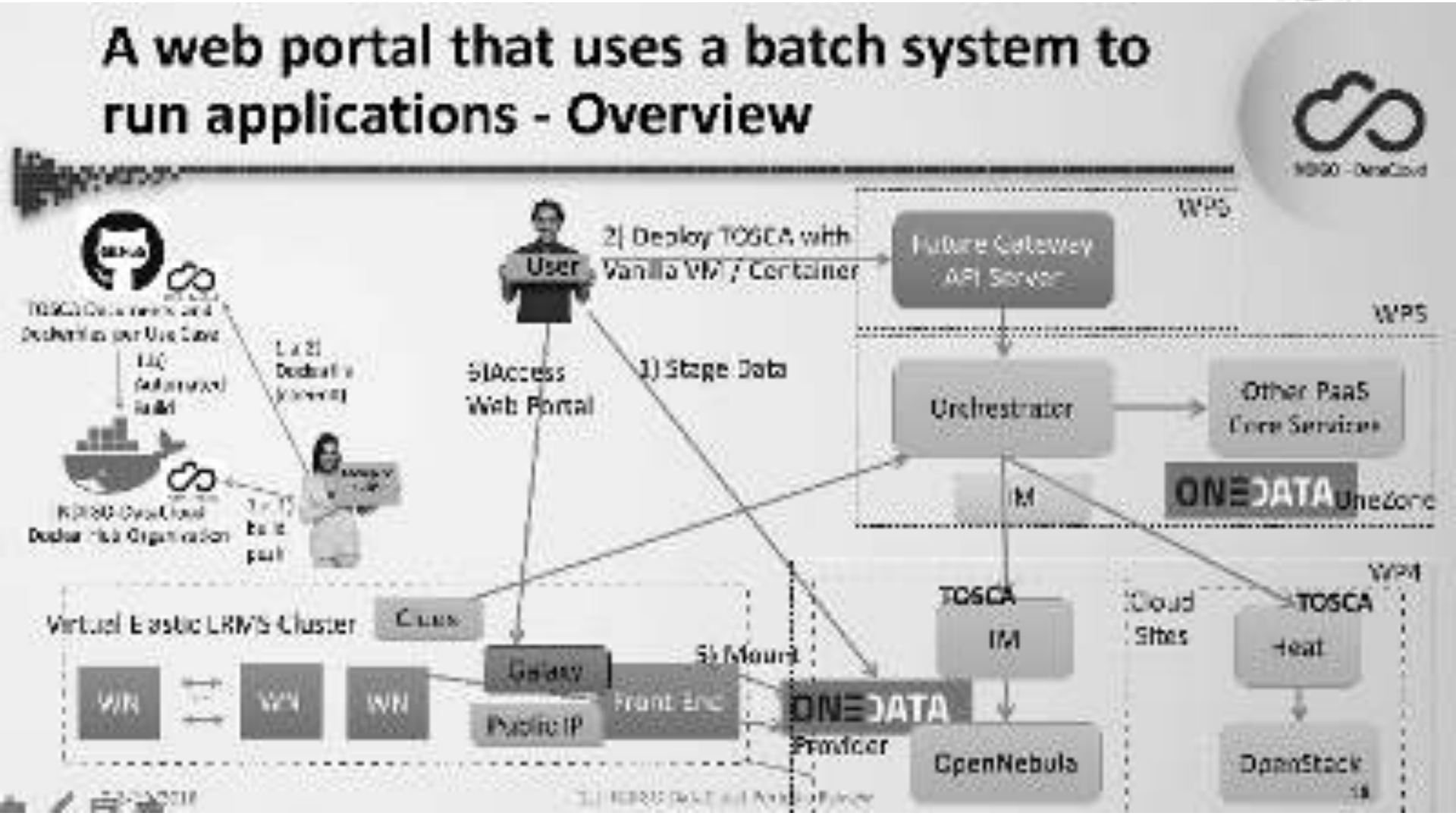
Interactive usage of a Docker container with *ssh* - Overview



Mesos PaaS solution exploiting INDIGO platform



A web portal that uses a batch system to run applications - Overview



Top 10 Strategic Technology Trends, 2014-2017 (Gartner)

Quiz: find the Cloud

2014	2015	2016	2017
• Mobile Device Diversity and Management • Mobile Apps and Applications • The Internet of Everything • Hybrid Cloud and IT as Service Broker • Cloud/Client Architecture • The Era of Personal Cloud • Software Defined Anything • Web-Scale IT • Smart Machines • 3-D Printing	• Computing Everywhere • The Internet of Things • 3D Printing • Advanced, Pervasive and Invisible Analytics • Context-Rich Systems • Smart Machines • Cloud/Client Computing • Software-Defined Applications and Infrastructure • Web-Scale IT • Risk-Based Security and Self-Protection	• The Device Mesh • Ambient User Experience • 3D Printing Materials • Information of Everything • Advanced Machine Learning • Autonomous Agents and Things • Adaptive Security Architecture • Advanced System Architecture • Mesh App and Service Architecture • Internet of Things Platforms	• AI and Advanced Machine Learning • Intelligent Apps • Intelligent Things • Virtual and Augmented Reality • Digital Twin • Blockchain and Distributed Ledgers • Conversational System • Mesh App and Service Architecture • Digital Technology Platforms • Adaptive Security Architecture
https://goo.gl/LcVnBk	https://goo.gl/W9lP91	https://goo.gl/gury81	https://goo.gl/C3nRsj

Remember the definition of “Cloud”

1. Supplying
2. information and communication technologies
3. as a service

Trends for 2017, example #1

- **AI and Advanced Machine Learning**
  - "Artificial intelligence (AI) and advanced machine learning (ML) are composed of many technologies and techniques (e.g., deep learning, neural networks, natural-language processing [NLP]). The more advanced techniques move beyond traditional rule-based algorithms to create systems that understand, learn, predict, adapt and potentially operate autonomously."
  - → this is applied to both **physical devices** (e.g. robots, autonomous vehicles) as well as to **apps and services**.
  - All of them must **1) supply 2) services via 3) distributed communication → Cloud!**

Trends for 2017, example #2

- **Mesh App and Service Architecture**
  - “In the mesh app and service architecture (MASA), mobile apps, web apps, desktop apps and IoT apps link to a broad mesh of back-end services to create what users view as an "application." The architecture **encapsulates services and exposes APIs** at multiple levels and **across organizational boundaries** balancing the demand for agility and scalability of services with **composition and reuse** of services.”
  - Again, this is **Cloud!**
- **The keywords: collaborative, distributed, complex, standards-based models**

At the European level…

- Two main projects to build a **European Open Science Cloud (EOSC)**, one already approved and one submitted a couple of months ago
  - Plus several others proposals, more development-oriented
- **EOSCpilot (running)**
  - Design and trial a stakeholder-driven governance framework for the EOSC;
  - Contribute to the development of European open science policy and best practice;
  - Develop demonstrators of integrated services and infrastructures in a number of scientific domains, showcasing interoperability and its benefits;
  - Engage with a broad range of stakeholders, crossing borders and communities, to build trust and skills.

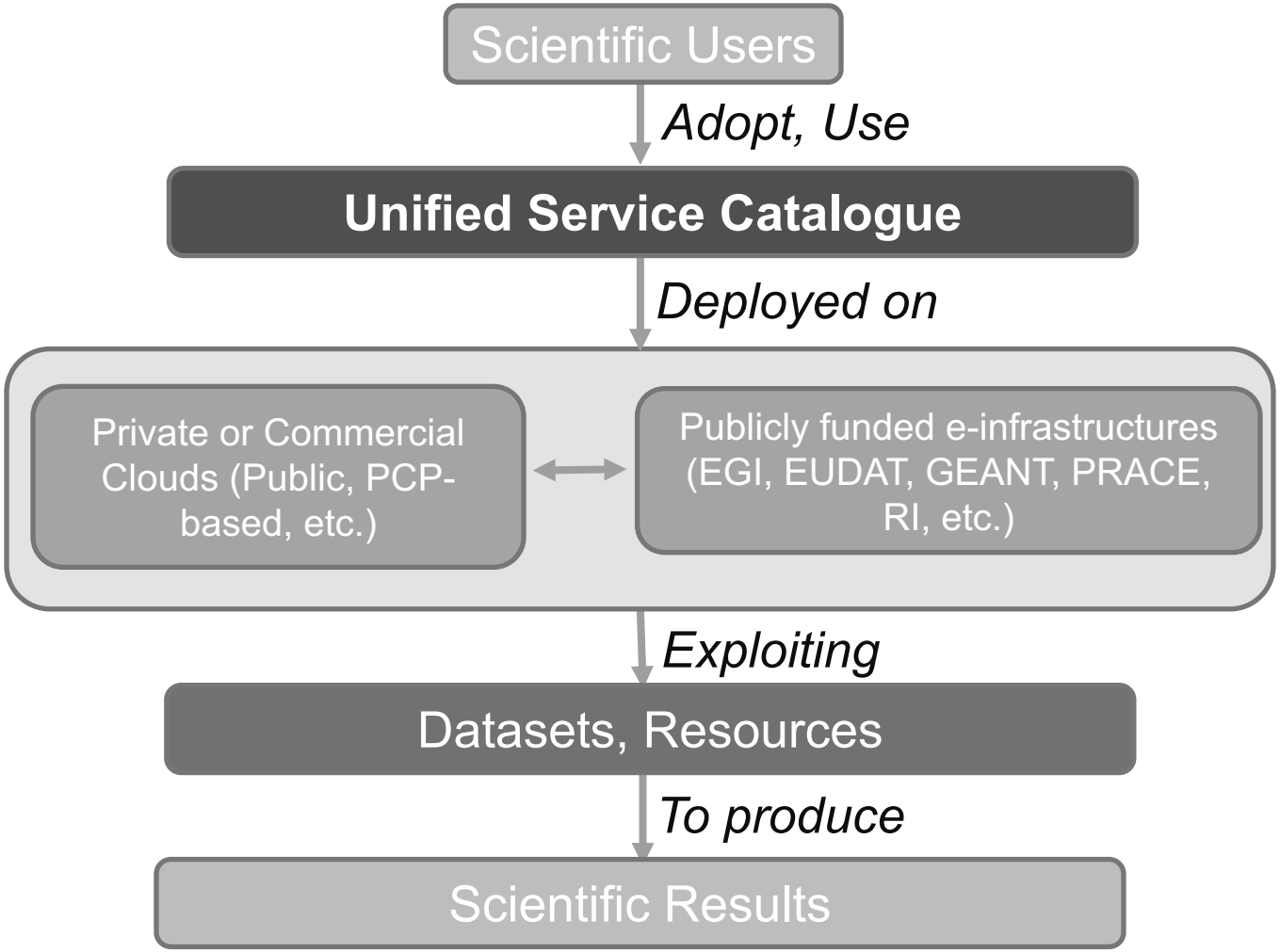
EOSC-hub (under evaluation)

- A joint proposal by EGI, EUDAT, INDIGO-DataCloud

- Objectives:
  - **Simplify access** to a broad portfolio of products, resources and services provided by different actors through an open service catalogue
  - **Remove fragmentation** of service provisioning and access to digital services
  - **Increase innovation** capacity of digital Infrastructures
  - **Consolidate** digital infrastructures by expanding capacities and capabilities, improving discoverability, access, interoperability and sharing across research communities and countries
  - **Extend access** to integrated compute, storage, data and software to new user groups including high-education and industry, increase the user base
  - **Expand human capacity** (consolidate/expand a distributed network of experts and service operators at local/national level)

The ultimate EOSC goal

- A **Unified Service Catalogue** available to multi-disciplinary scientific communities.
- **Services should be provisioned over any EOSC sub-infrastructure**: EGI, EUDAT, PRACE, community-based research infrastructures, commercial clouds, etc.



Where are we going, then?

- We (scientists) developed ”the Grid”
  - It has been working very well, but there was no uptake by the market, and it has serious sustainability / technological issues.
- The market has given us “the Cloud”
  - We are exploring several ways to use it, using private and public infrastructures.
  - It does not always work the way we want, so we are adapting it to our needs, and contributing several of these efforts back to the (open source) market.
- There are **still a lot of technological challenges** and contributions that await our/your input
  - Also through strong collaborations with industry

Recap

- **Seven statements to ponder at the end (adapted from https://goo.gl/w48zjl):**

1. The Cloud is not the Grid and is not the same as virtualization.
2. Cloud computing requires standard APIs.
3. The value is primarily in the applications, not in the infrastructure.
4. Moving to the Cloud is not a cure for bad practices.
5. Security is what you make of it.
6. There is no easy “instant Cloud solution”.
7. Know-how, flexibility, adaptation of technologies to create bespoke solutions are key.

Thank you

https://www.indigo-datacloud.eu

Better Software for Better Science.

"Stat rosa pristina nomine, nomina nuda tenemus"