Scientific Clouds

Davide Salomoni
INFN CNAF

Varenna, July 2014

Summer Course on “Grid and Cloud Computing – Concepts and Practical Applications”

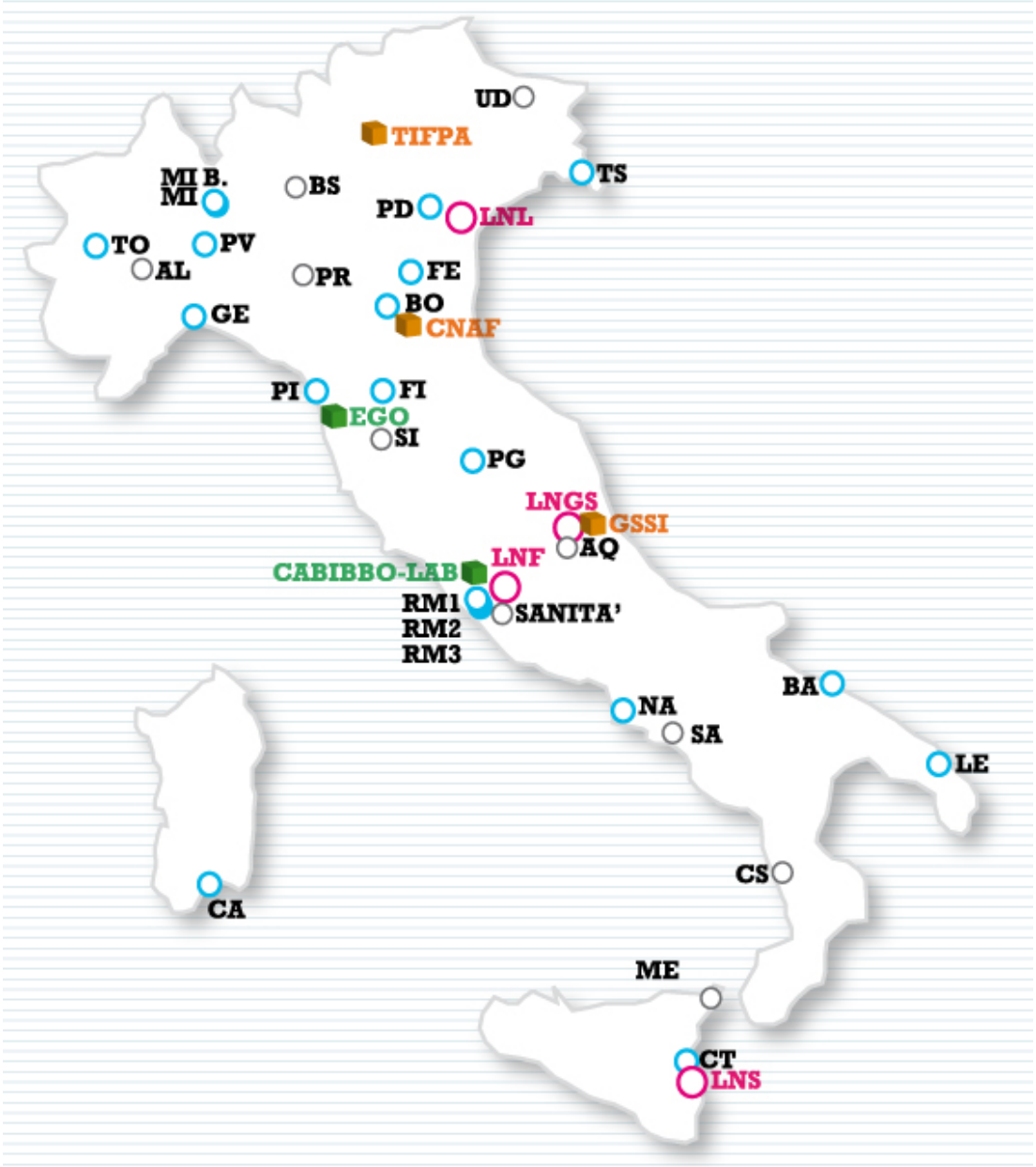
About myself

- **Davide Salomoni**
  - Physics degree in 1990. From 1991 to 1998 I worked at the INFN National Computing Center (CNAF) on network management, R&D on network protocols (IP, DECnet, SNA, etc.). I was the first manager of the then-newborn GARR (Italian National Research Network) NOC.
  - From 1999 to 2005 I worked at SLAC (USA), Colt Telecom (Netherlands) and NIKHEF (Netherlands) on local networking, data acquisition, development of commercial Internet solutions, Grid-based distributed infrastructures.
  - Since 2006 I am with INFN again. I was for some years coordinator of the computing division of the CNAF Computing Center; I am now an INFN Director of Technology, manager of the CNAF R&D department and responsible for several Cloud-related projects.
  - E-mail: Davide.Salomoni@cnaf.infn.it

    http://www.linkedin.com/in/davidesalomoni

INFN

- **INFN** = Istituto Nazionale di Fisica Nucleare (National Institute for Nuclear Physics)
  - Italian research agency dedicated to the study of the fundamental constituents of matter and the laws that govern them, under the supervision of the Ministry of Education, Universities and Research (MIUR).
  - It conducts theoretical and experimental research in the fields of subnuclear, nuclear and astroparticle physics, in close collaboration with Italian universities.
    - Strong experience and know-how also in cutting-edge technology and instruments. Among them, long-standing experience on HPC, distributed storage and computing (Grids and Clouds)
- Strong focus also on **technology transfer programs**.
  - Transfer of technologies and know-how to Italian and European companies, developed within INFN scientific programs.



- **INFN sites**
  - About 30 among full INFN branches and collaboration groups hosted in university departments
  - 4 national laboratories: Catania, Frascati, Gran Sasso, Legnaro
  - 3 national centers:
    - CNAF, National Center for Research and Development in Informatics and Telematics, Bologna
    - GSSI, Gran Sasso Science Institute, L'Aquila
    - TIFPA, Trento Institute for Fundamental Physics and Applications, a Trento

CNAF

- **INFN Center established in Bologna** since the early 60s, with the goal to analyze physics events from bubble chambers.
  - Hence the acronym CNAF = Centro Nazionale Analisi Fotogrammi.
  - Its mission evolved along the years: Italian reference center for the development of scientific networking in the 90s, focus on distributed computing since 2001.
- **CNAF hosts the INFN National Computing Center** (called “Tier-1”), currently utilized by about 20 international scientific collaborations, including the 4 LHC experiments.
  - About 20,000 computing cores, 15 PB disk storage, 18 PB tape storage, 40 Gbit/sec of WAN connectivity.
- Several initiatives linked to **research and development** and to the preparation/exploitation of national or international projects related to Cloud computing.

Agenda

- **Cloud computing: definition & technology recap**
- Some pros and cons
- Apps in the Cloud?
- Is anything missing?
- Conclusions

Cloud computing

- “The horse is here to stay, but the automobile is only a novelty - a fad.” (President of the Michigan Savings Bank, 1903)

- “Television won't last because people will soon get tired of staring at a plywood box every night.” (Darryl Zanuck, 20th Century Fox, 1946)

- “By 2005 or so, it will become clear that the Internet's impact on the economy has been no greater than the fax machine's.” (Paul Krugman, New York Times economist, Nobel prize for economics “for his analysis of trade patterns and location of economic activity”, 1998)

- “If I had asked people what they wanted, they would have said faster horses.” (Henry Ford)
- “If you think you understand quantum mechanics, you don't understand quantum mechanics.” (Richard Feynman)

Cloud computing

- The classical definition come from the US National Institute of Standards and Technology (NIST) (http://goo.gl/eBGBk)

- In summary, Cloud computing deals with:

Supplying
information and communication technologies (ICT)
as a service

The 5 Cloud Postulates

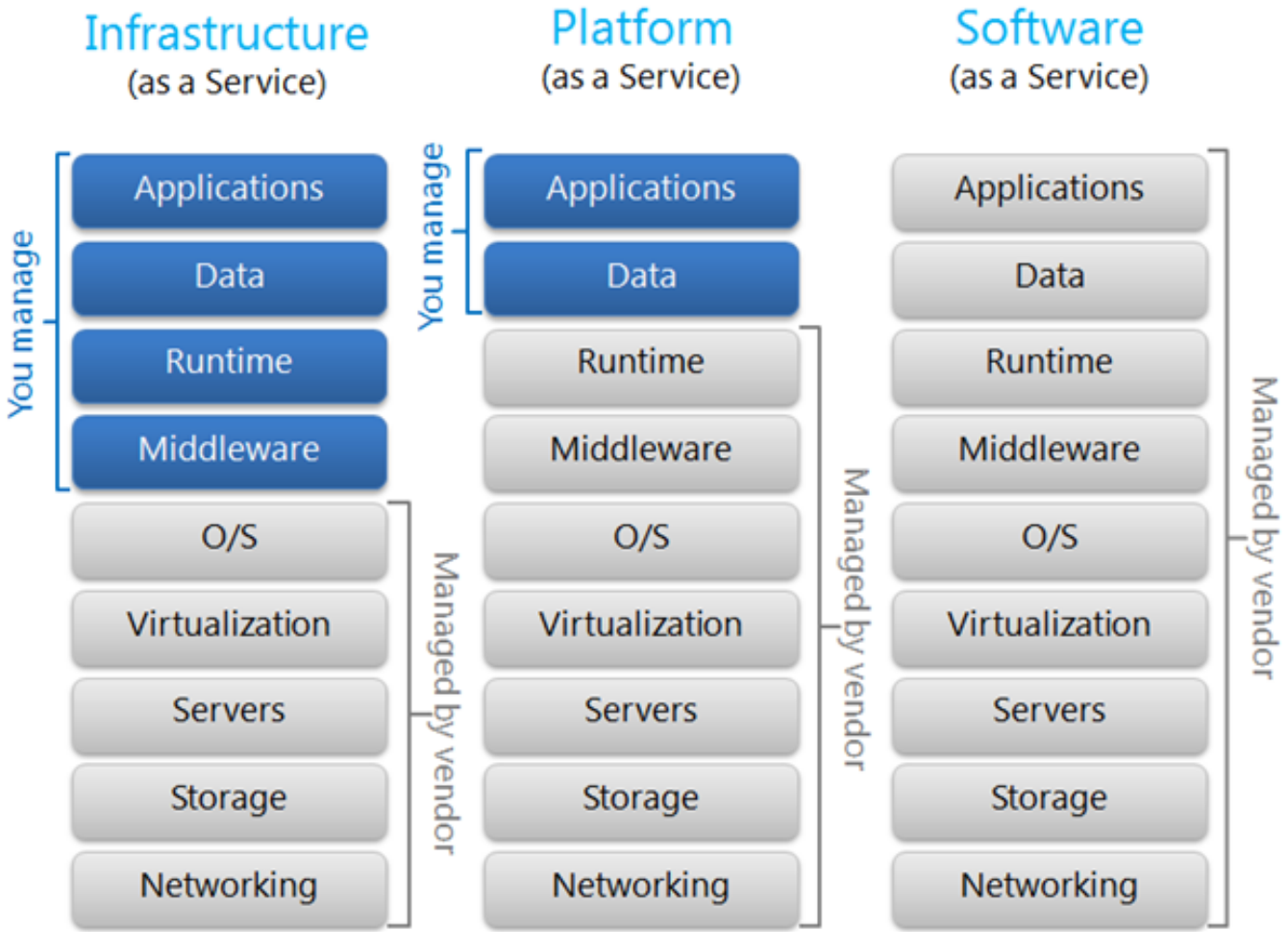
- **Self-service, on-demand**
  - The customer autonomously ask what he needs, when he needs it (and hopefully gets it).
- **Access through the network**
  - This takes for granted that some sort of (usually broadband) intranet or internet networking is available.
- **Resource pooling**
  - The customer does not care about resource details, managed by the Cloud resource providers.
- **Elasticity**
  - The Cloud service can rapidly scale, according to the customer needs.
- **Pay as you go**
  - The customer only pays for resources actually used.

An analogy: car rentals

- Self-service, on-demand
  - Online of phone-based booking.
- Network
  - Worldwide car rental network.
- Resource pooling
  - There's a whole pool of cars, and it's the car rental business to manage the number of cars that are needed.
- Elasticity
  - The number of available cars usually depends on market requests.
- Pay as you go
  - Customers only pay for the time they are using the car. (and are not bothered with things such as insurances, tire changing, etc.)

Source: http://goo.gl/cEa8M

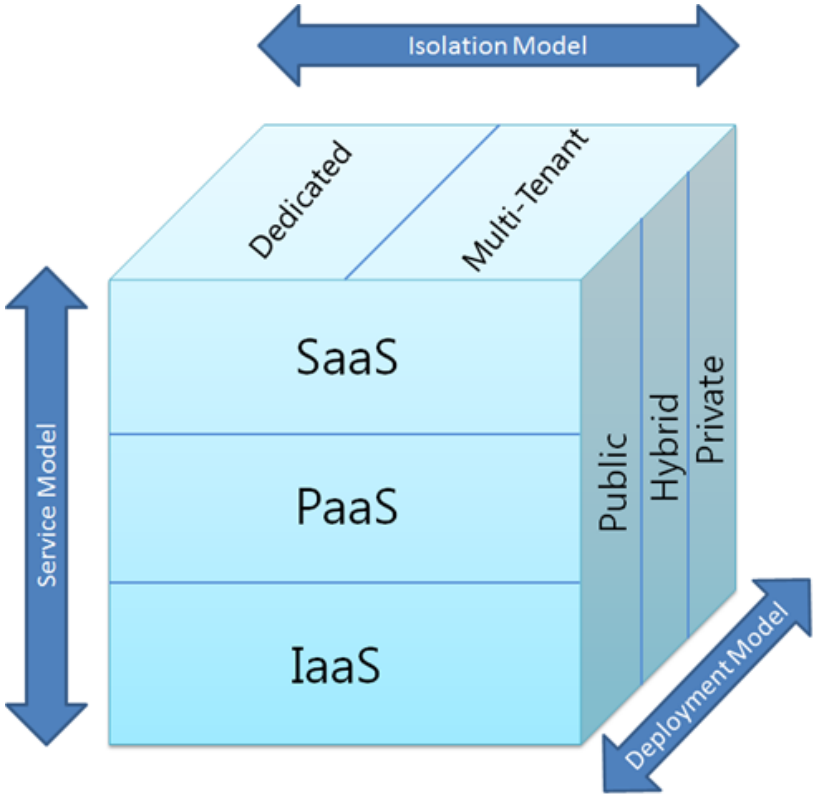
Who does what?



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

Let's add dimensions

- Beyond the *service models*, 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.

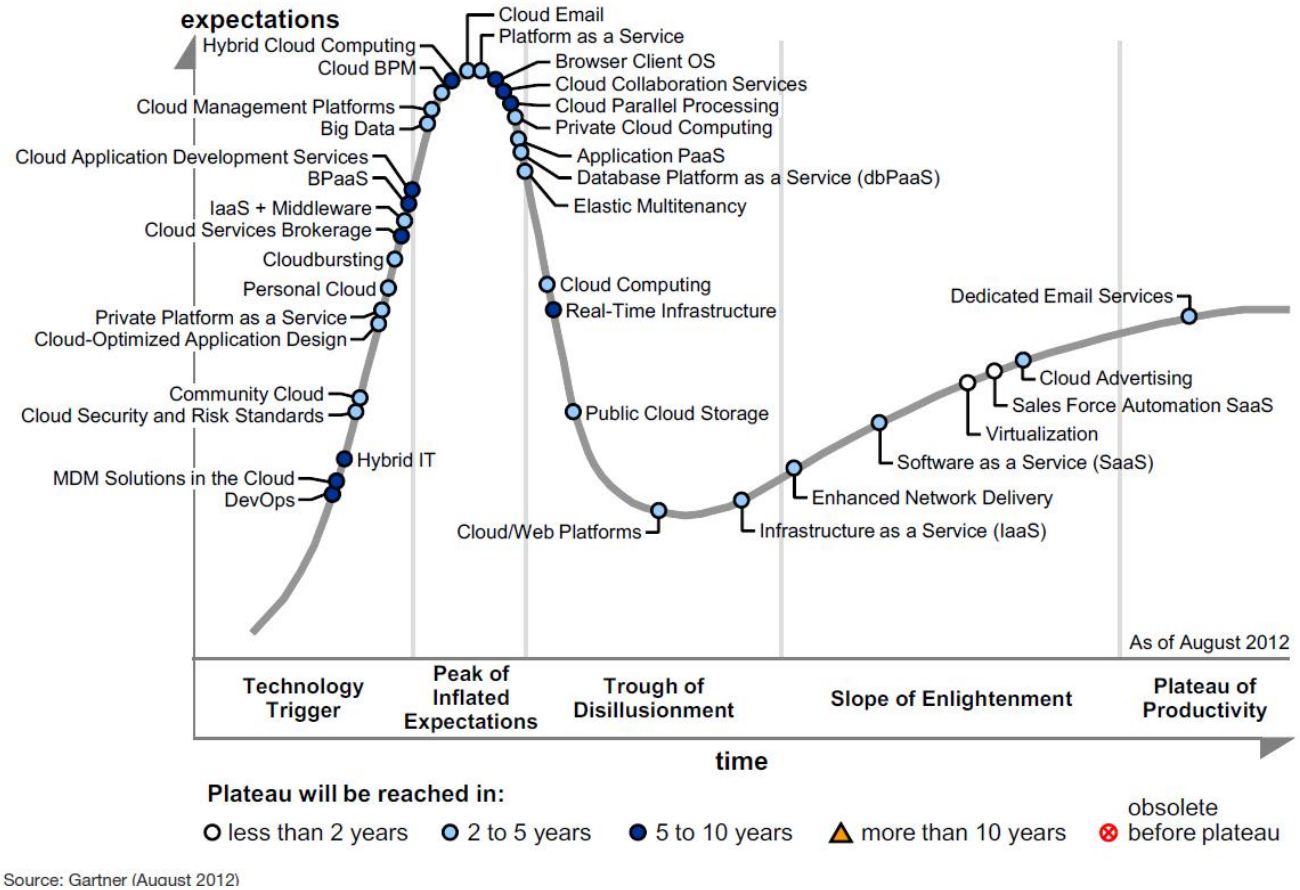
Isolation

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

Where are we in the Cloud hype?

- Forbes, quoting Gartner (http://goo.gl/4r1AM), gave an estimate of the maturity of technologies associated to Cloud computing

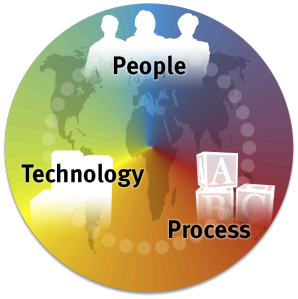
Figure 1. Hype Cycle for Cloud Computing, 2012



Source: Gartner (August 2012)

Is it Virtualization or Cloud Computing?

- Cloud computing can be provisioned also *without* virtualization technologies.
  - However, virtualization technologies often allow to reduce operational and capital account expenses.
  - At the same time, being able to very rapidly provision VMs is not very efficient, if it takes several months to provision and install the physical hosts where VMs will run (procurement processes anyone?)
    - In addition, it the *time* spent for provisioning and managing a virtualization layer recovered by savings associated to not having to dedicate physical servers?
    - Do not discard the importance of using installation, monitoring and accounting tools as automated as possible.



Virtualization: advantages (1/2)

- Server consolidation.
  - Several VMs on a physical host.
  - Reduces hardware procurement costs and may simplify activities such as management and monitoring.
- Sandboxing.
  - Application isolation.
  - Software development, testing, debugging.
  - Creation of dedicated environments for legacy applications.
- Creation of VMs *on-demand.*

Virtualization: advantages (2/2)

- Decoupling between hardware and software.
  - VM Suspend/Resume.
  - VM migration across physical servers (with various degrees of complication).
- Testing of new O/S or application versions.
  - Or of *old* versions: data preservation.
- Emulation of hardware not available on the physical hosts.
- Execution of applications not compatible with the O/S installed on the physical host.

Virtualization: disadvantages

- Security.
  - The same hardware handles several O/S, managed by a software layer: increase of the probability of having bugs and of the number of *attack vectors*:
    - VM-to-VM network attacks.
    - VM-to-HV (KVM o XEN). KVM is Linux kernel module. Xen is directly connected to hardware → everything can be compromised.
    - VM-to-QEMU. QEMU is a complex and large software. If it gets compromised, the attack can reach the O/S layer.

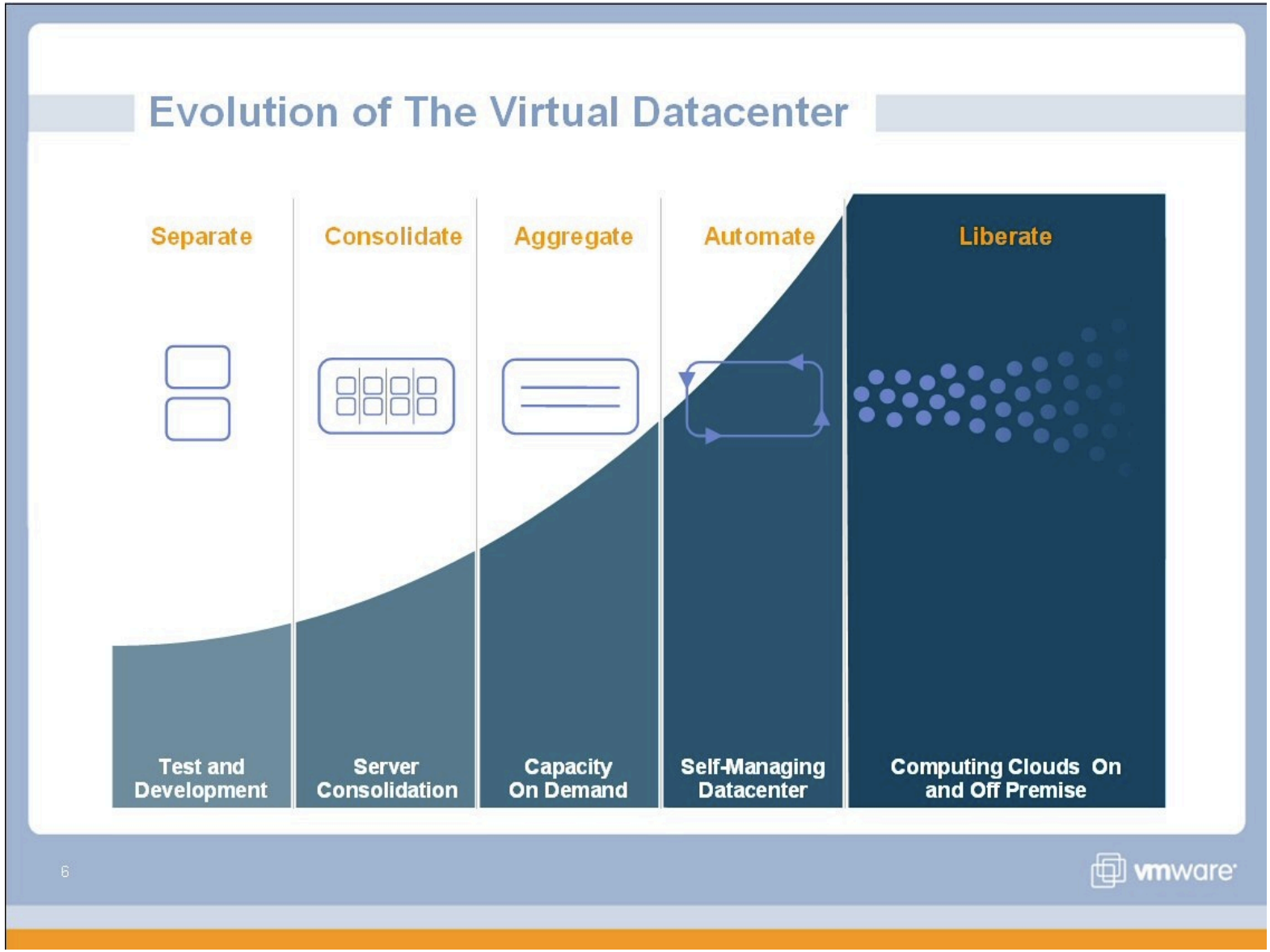
- Performance.
  - Overhead for the physical machine.
  - Less performance for the VM, especially with I/O.

On security

- **Security** is not something to be taken lightly. Besides issues linked to wrong network isolation (a potentially very complex area), here are some examples of recent security exploits:
  - CloudBurst, 2008 - VMware, result: full breakout.
  - Xen Ownage Trilogy, 2011 - Xen, result: full breakout.
  - VirtuNoid, 2011 - KVM, result: full breakout.
  - SYSRET-64, 2012 - Xen, result: full breakout.
  - VMDK Has Left The Building, 2012 - VMware, result: data leakage, loss.
  - KVM IOAPIC, SET MSR, TIME, 2013 - KVM, result: DoS, potential breakout.
- A useful text (born out of the OpenStack framework but with several generally useful and valid concepts) is the **OpenStack Security Guide**, available on http://goo.gl/ibkNsD.
- Important message: there is a definitely positive impact on security brought by the adoption of **efficient provisioning and configuration systems**.

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)

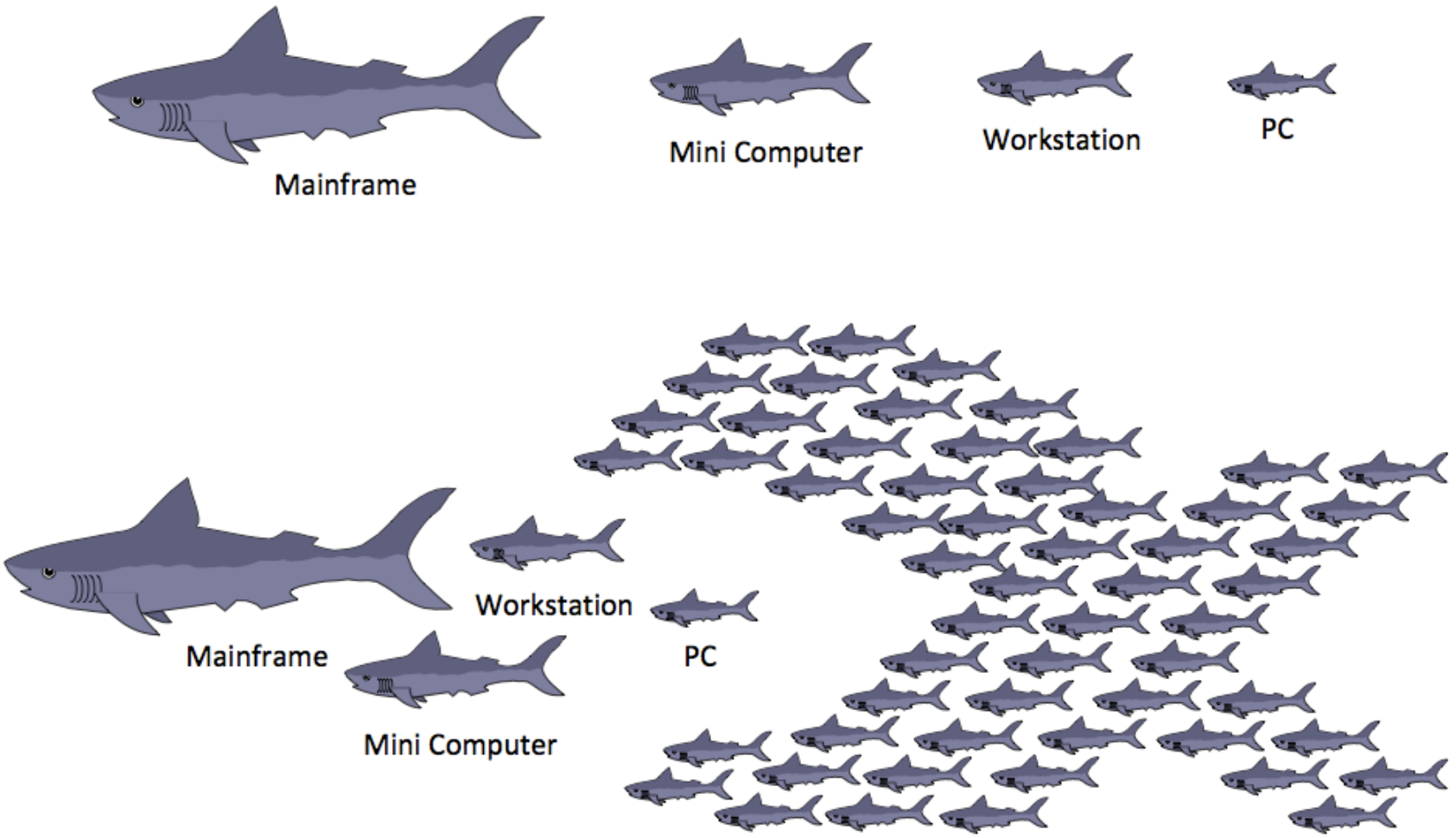


Source: http://goo.gl/7yHnXB

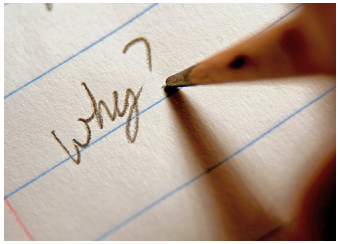
Agenda

- Cloud computing: definition & technology recap
- **Some pros and cons**
- Apps in the Cloud?
- Is anything missing?
- Conclusions

Distributed Computing



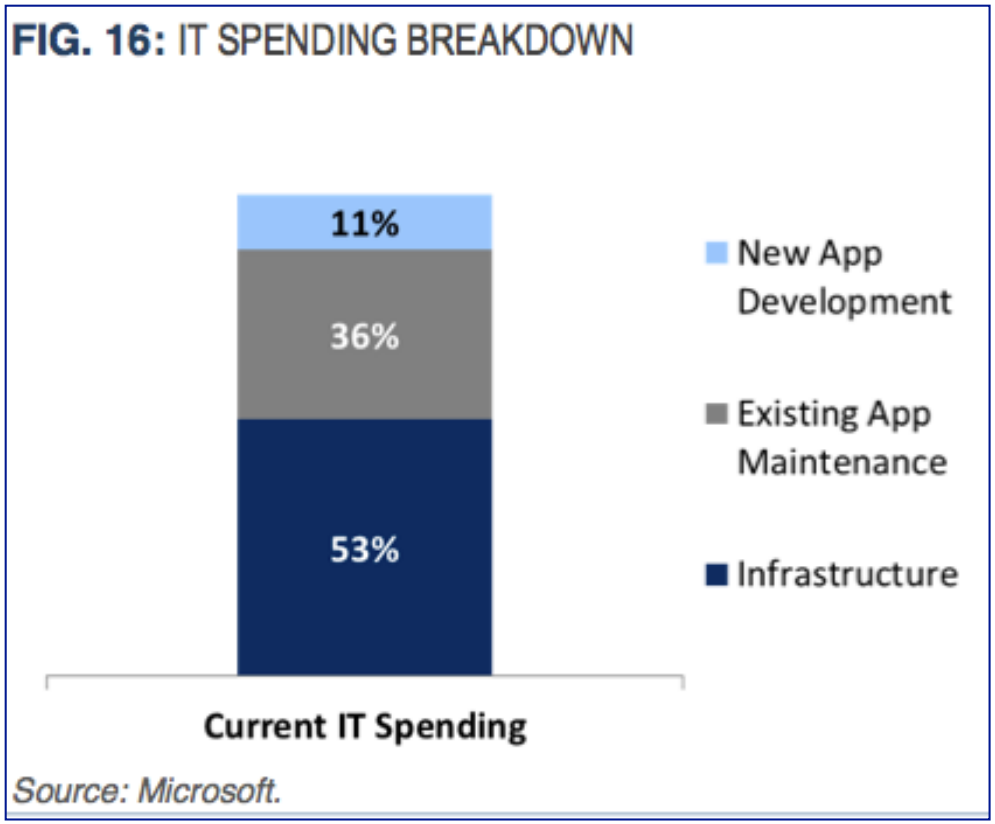
Recently



- "Cisco and Partners to Build World's Largest Global Intercloud" – cf. http://newsroom.cisco.com/release/1373639 (March 24, 2014)
- Quotes:
  - Cisco expects to invest **over $1 billion** to build its expanded cloud business over the next two years
  - The Cisco OpenStack-enabled Intercloud is designed to allow organizations and users to combine and move workloads – including data and applications – across different public or private clouds as needed
  - We (Cisco) expect to expand the addressable cloud market for Cisco and our partners **from $22Bn to $88Bn** between 2013-2017. (see http://blogs.cisco.com/news/introducing-ciscos-global-intercloud/)

Pros and Cons

- Starting from the main features of Cloud computing, one could derive a number of lists related to pros and cons.
- The following slides compare some of the main characteristics of Cloud vs. traditional infrastructures.



Cost reduction

- A large-scale resource center allows to lower **costs per- server**.
  - E.g. for energy consumption (it *might be* more likely that a big center pays energy less than a small one).
  - Manpower costs are spread over more resources.
  - Hardware procurement costs are lower thanks to economies of scale.
- The more customers there are, the less **per-customer management costs** are relevant.
- Resource aggregation leads to an increase in **efficiency utilization**.
  - And to lowering relative complexity.
    - (*if* coupled with good planning and organization)

Flexibility and scalability

- **Self-service provisioning**
  - To be compared with typical procurement time in a traditional data center.
- **Scale-out**
  - If necessary, it is possible to add resources through other Cloud infrastructures (instead of purchasing new resources).
- **Mitigation of growth uncertainties**
  - Often business grown patterns are unknown; with traditional infrastructures this can lead of excessive over-provisioning.
- **Flexible modulation of the workload type**
  - With Cloud computing it is possible to dynamically decide to change the workload type (i.e. CPU-intensive vs. I/O-intensive vs. HPC) without the need to utilize dedicated resources, or to take technological decisions too far in advance.

More resources for everybody

- In some sense, **“democratization” of access to resources.**
  - For instance, a Cloud infrastructure may facilitate access and utilization to resources by SMEs or by small scientific experiments.
  - Ability to engage in problems that could not be tackled in the past because of time- or cost-related limits.
- In general, Cloud computing – *with its promise of infinite resources* – *may* shift problem models **from Batch to Real-time** or to quasi-Real-time.
- Especially with SaaS solutions (e.g. Google Apps, Dropbox), Cloud computing might provide **access to ubiquitous resources on multiple platforms**.
  - Including mobile platforms and Internet appliances.
  - Without installing applications, O/S, licenses, etc.

Business opportunities

- Cloud computing offers **business opportunities** (that might also be exploited by government bodies in some cases!) not merely linked to reselling software frameworks developed by others.
  - E.g. technological development, integration, training, customization, support.
  - The Open Source model is in this case particularly appropriate (see e.g. Red Hat, Canonical, Mirantis, but also IBM, HP, etc.).

A couple of words on Cloud markets

- The following is potentially true not only for public (i.e. privately owned) Cloud providers, but also e.g. for gov't agencies willing to provide Cloud services.
- **Traditional services** are nowadays very efficient, so it is very difficult to differentiate (and make money) out of them. Unless perhaps you:
  - Leverage state-of-the-art technologies (this normally comes at some not-negligible cost anyway).
  - Profit from *big* economies of scale (not always easy).
- So how do you differentiate yourself? And how is this relevant for scientific Clouds?

Market differentiation

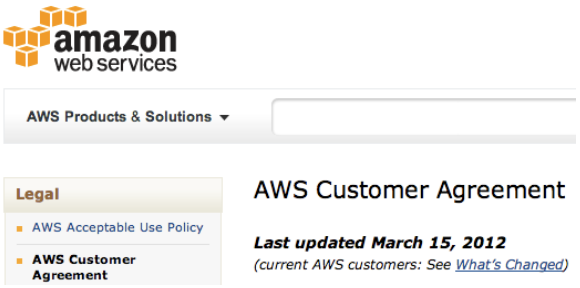
- In the market, there are several *thousands* Cloud service providers. **How do you select one?** Or how do you *become one yourself*? What are the key points to differentiate your offering?
- The core part: "**leverage new services and new ways to consume them**" (thanks to Arturo Suarez, http://goo.gl/p0z2x6). Some themes:
  - Hosted private clouds
  - Virtual data centers
  - Different storage flavors
  - Network virtualization
  - SLA
  - PaaS
  - Hybrid Clouds and federations
  - Flexible, reliable billing
  - Easy provisioning portals

- Are commercial clouds capable of **supporting scientific needs** using commercial frameworks?
  - We probably need to extend (not rewrite!) what's available.
  - But we may need (or want) to do so in a way that maintenance and sustainability is *somewhat* offloaded away from scientific communities (e.g. community support, or commercial support if our features are interesting enough for the market).

Now some disadvantages…

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, OR DATA), EVEN IF A PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. FURTHER, NEITHER WE NOR 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 UNANTICIPATED OR UNSCHEDULED DOWNTIME OF ALL OR A 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 OFFERINGS; OR (D) ANY UNAUTHORIZED ACCESS TO, ALTERATION OF, OR THE DELETION, DESTRUCTION, DAMAGE, LOSS OR FAILURE TO STORE ANY OF YOUR CONTENT OR OTHER DATA. 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 some 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***?

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

Maturity

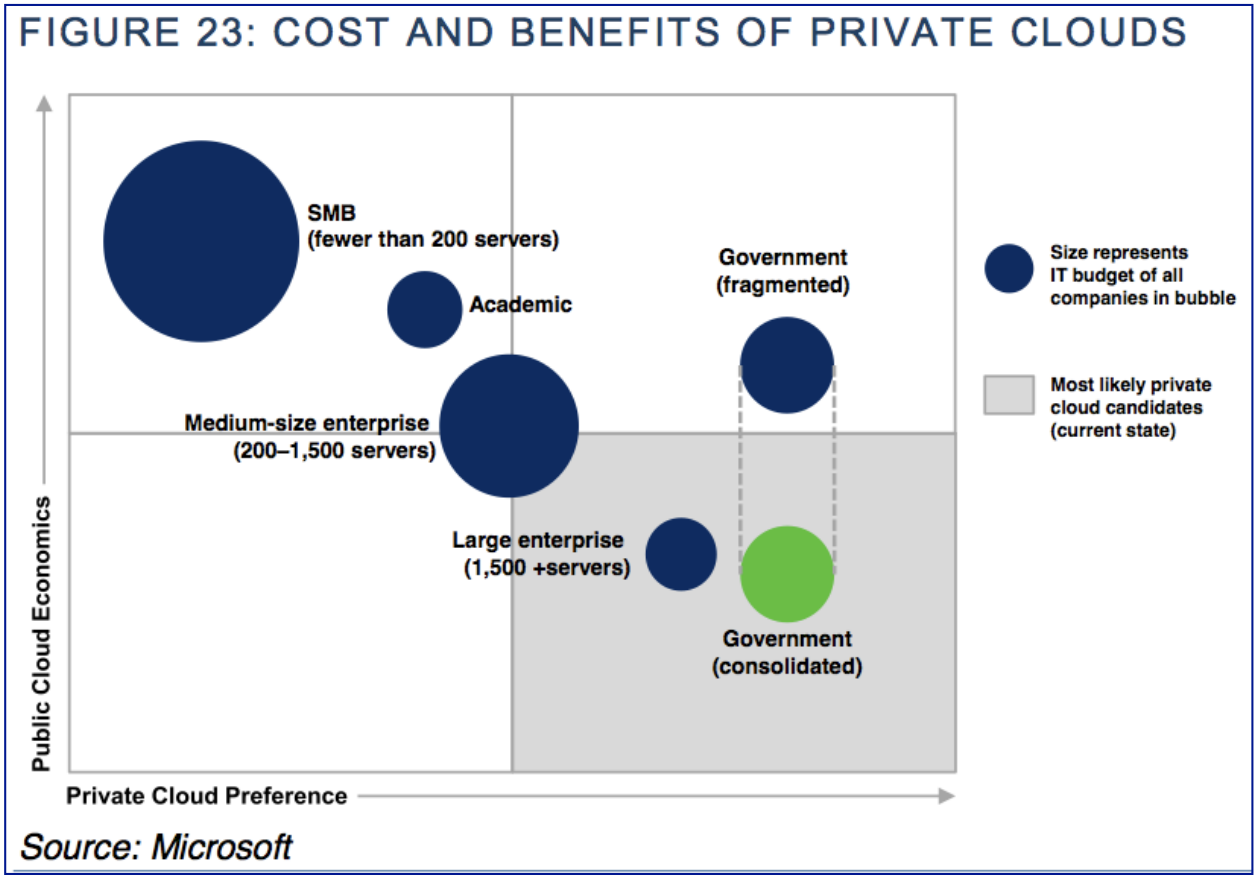
- Cloud computing is a set of modern technologies **sometimes not entirely stable**.
  - Tuning and experts are often needed.
  - And sometimes, in order to avoid complex configurations (even in public Clouds) shortcuts are taken…
    - For example linked to tenant isolation.
    - Or to the application of security patches.
  - … on the other hand; sometimes, in order to address complex problems, “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.

Private vs. Public Clouds

- Albeit a simplification, this is a very interesting picture:



Agenda

- Cloud computing: definition & technology recap
- Some pros and cons
- **Apps in the Cloud?**
- Is anything missing?
- Conclusions

But how to use Cloud computing?

- A key question, assuming that Cloud computing answers issues such as vendor independence, use of standard interfaces, cost savings, etc.:

How to migrate apps to Cloud computing?

Migrating applications to Cloud infrastructures

- **Goal**: moving an existing application from a local data center to reference Cloud infrastructure.
- First ask yourself what are the **technical and business reasons** leading to the migration and if they are really sound. Typically:
  - Cost reduction → resource pooling, pay-per-use
  - "Business agility" → simplification of deployment
  - Management savings → performance (due e.g. to high-performance platforms, auto-scaling), ease of system administration (e.g. outsourcing ops responsibilities)
- **Public or private Cloud?**
  - WAN-level traffic? (typically very expensive)
  - Security?
  - Integration with other legacy applications? (e.g. strong coupling with applications running on AS400 or similar platforms)

How to migrate? (SaaS)

- Are there readily-available **SaaS alternatives** through Cloud providers?
  - In reality, this is not a migration, but a transposition of applications. It is worth considering, though.
  - Note that even with a SaaS solution, data might need to be migrated (hence further costs).
  - Beware of long-term costs, depending e.g. on the number of users, and on the type of contract. This is particularly true when significant investments have already been made with existing data centers.
  - Security? (e.g. for sensitive data hosted by a Cloud provider.)

How to migrate? (PaaS)

- Is a **PaaS model** applicable?
  - This might be the case if e.g. the application is based on standard application servers, such as Java EE or .NET.
  - A Cloud provider might typically provide also database back-end systems (e.g. SQL- or noSQL-based).
  - Beware of the fact that the application server infrastructure (and / or the back-end databases) might be shared with other customers (security! Performance!) …
  - … and to the fact that not all PaaS features needed to the application might be available through the selected (perhaps by somebody else) Cloud provider.

How to migrate? (IaaS)

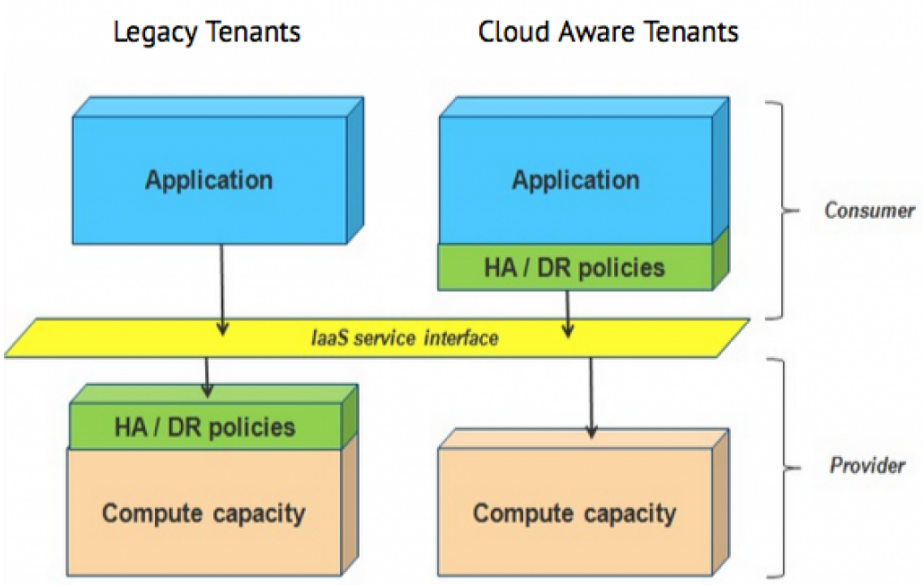
- **Is there hardware and software compatibility** of the application with the Cloud provider?
  - E.g. x86-compatible platform, a certain O/S.
  - Typically, one shares VM provisioned by a Cloud provider on the same physical hardware used by other customers (security!).
  - What scalability is there? How is load-balancing implemented e.g. when auto-scaling? (is there *any* auto-scaling feature available?)

Looking at a typical application architecture

- In **multi-tiered applications** there are usually 3 tiers:
  - *Data management* tier, dedicated to managing databases (be they relational or not).
  - *Business logic* tier, typically through application platforms such as Java EE or .NET.
  - *Presentation* tier, interfacing the application with the user or with other external components.
- *If* the application is well structured, *it might be possible* to **migrate individual tiers to the Cloud** independently.
  - For instance, moving to the Cloud only the presentation layer.
  - But it is not a given that this is effective nor that it is advantageous, for example when there is high network traffic across tiers. Thorough application profiling is needed to evaluate potential associated costs.
    - And to evaluate requirements for what regards CPU, RAM, storage (e.g. IOPS) and network.

Is my application "cloud-friendly"?

- **"Cloud-aware" applications:**
  - Distributed
  - Stateless
  - Fail-over in the app
  - Scaling in the app
- **"Legacy" applications:**
  - Client-server
  - Monolithic, no horizontal scalability
  - Fail-over in the infrastructure
  - Scaling in the infrastructure



Source: VMware

Analogy: pets vs. cattle

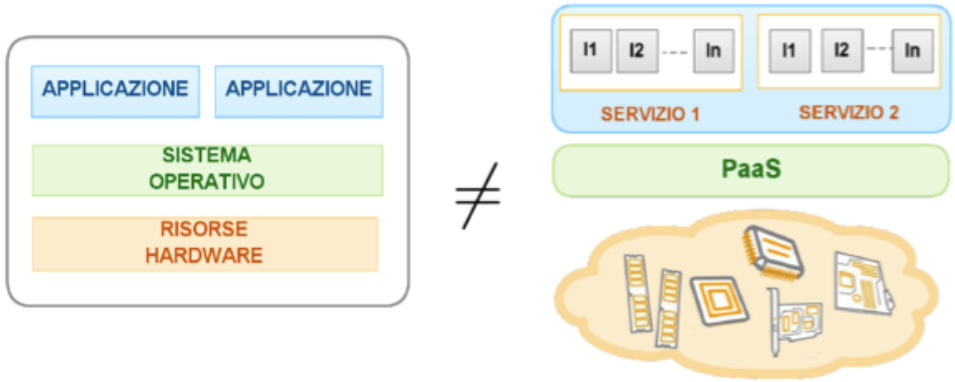
- **“Legacy”** applications are treated as **pets**. They are unique and often irreplaceable.
- **“Cloud”** application are treated as **cattle**. For instance, when a cow falls ill, we replace it with another taken from the pool we have at our disposal. All of them share the same identical technical function.

Fonte: http://goo.gl/Gx0ly

What are the differences?

Le architetture delle piattaforme Cloud sono del tutto diverse da quelle convenzionali

- Applicazioni ⇒ Fornite come servizi web
- Istanza ⇒ Entità operativa che fruisce i servizi
- Elasticità ⇒ Capacità di adattarsi (scaling) all'esigenza corrente
  - Ottenuta mediante la replicazione delle istanze



Applicazioni progettate per piattaforme convenzionali sono inadeguate agli ambienti Cloud

Quali considerazioni architetturali sono necessarie?

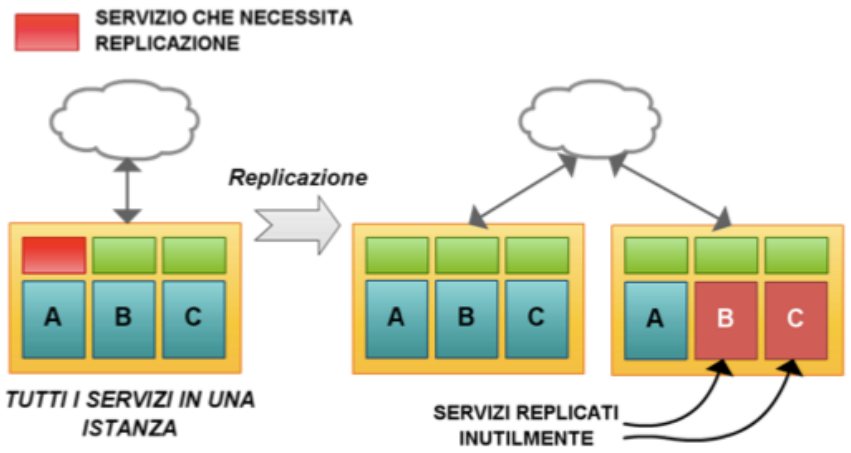
Fonte: F.Cacco, Tesi di laurea in Informatica, UniPD, Ottobre 2013

Service decomposition

Necessario replicare solamente i componenti che ne hanno reale necessità

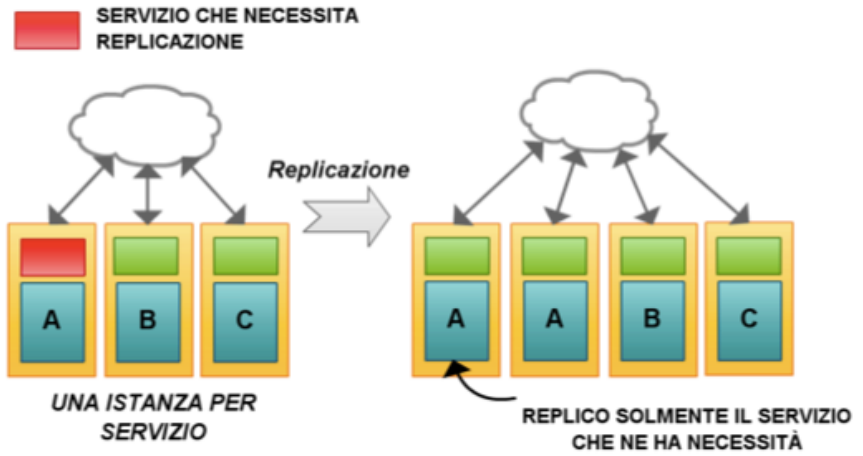
Tutti i servizi in una unica istanza

- *Per scalare un servizio devo replicare l'istanza che fornisce tutti i servizi*



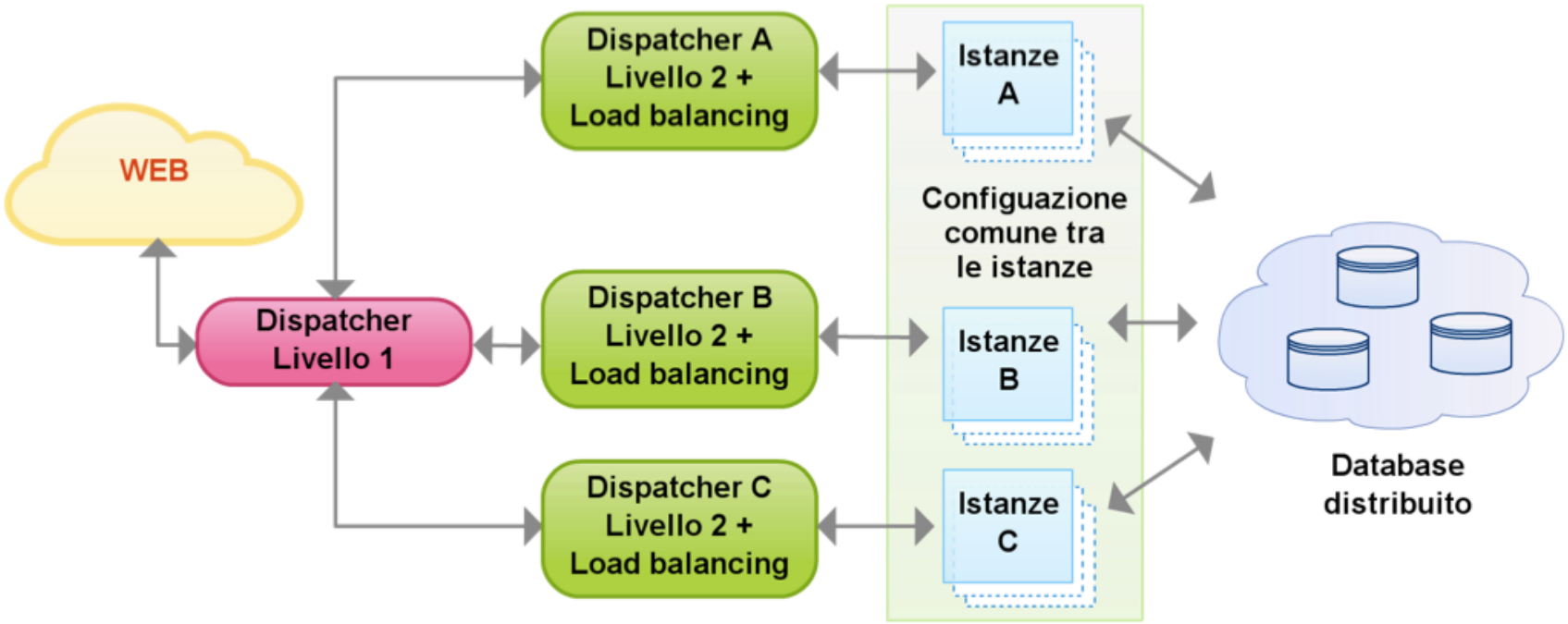
Una servizio per istanza

- *Per scalare un servizio posso replicare solamente l'istanza che lo fornisce*



Fonte: F.Cacco, Tesi di laurea in Informatica, UniPD, Ottobre 2013

Architectural Summary



Fonte: F.Cacco, Tesi di laurea in Informatica, UniPD, Ottobre 2013

Agenda

- Cloud computing: definition & technology recap
- Some pros and cons
- Apps in the Cloud?
- **Is anything missing?**
- Conclusions

Is anything missing?

- We have seen more of less *what* Cloud computing is and, very broadly, *how* we could adapt our applications to the Cloud.
- But **is there anything we are missing**, esp. for what regards scientific applications?

WLCG Specific needs – 1

- Introduction of use of federated identities (eduGAIN)
  - Requires integration with cloud software stacks
  - We may continue to use X509 under the covers
- Ability to federate clouds *for a science community*
  - Requires a distributed authorization service
  - In the way that VOMS provides this today

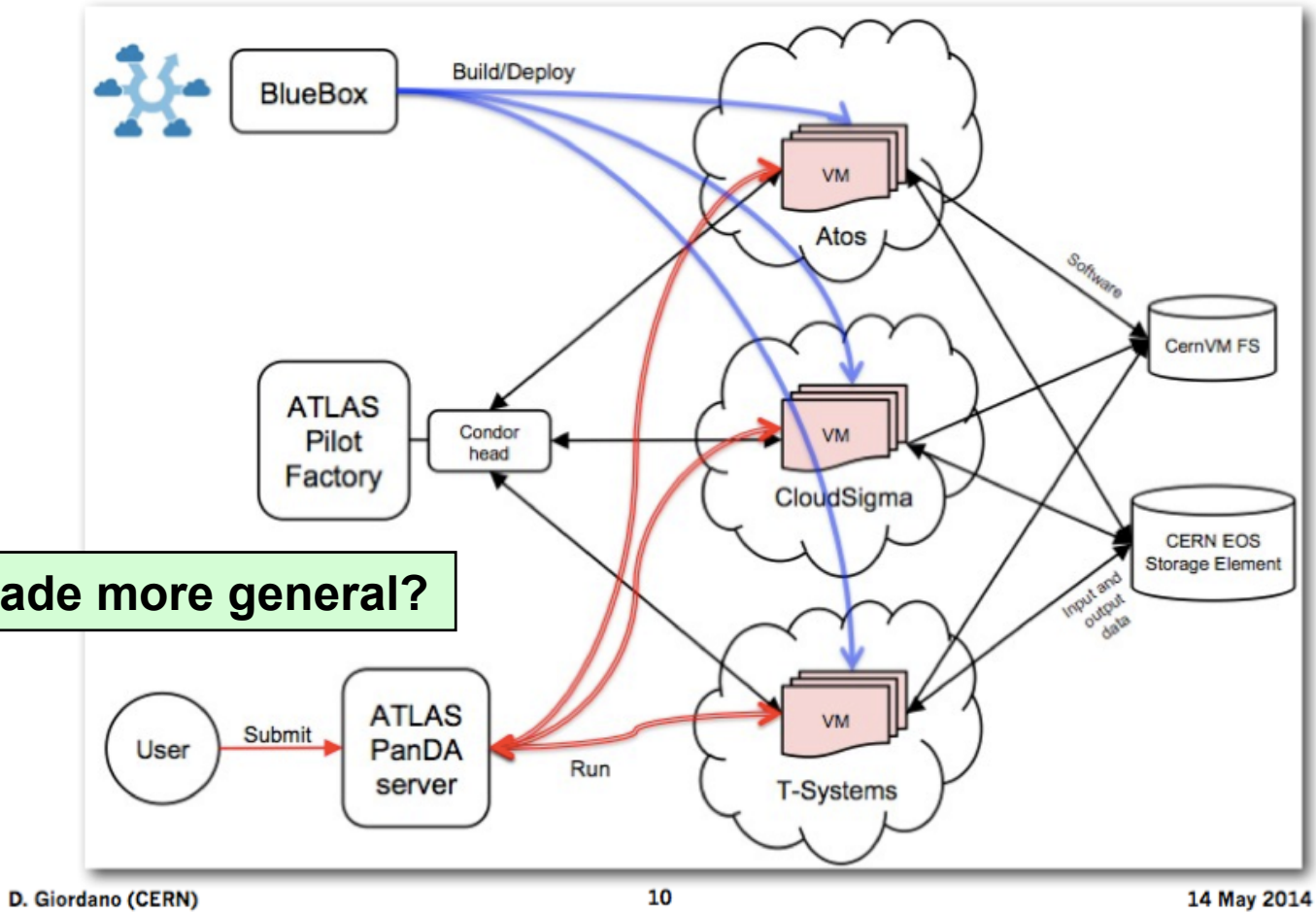
Ian Bird

WLCG Specific needs – 2

Ian Bird

- Problem of managing "scheduling" in a resource-constrained environment
  - i.e. we will not have *elastic* clouds
  - How do I provide a proof cluster for X hours to a single user while maintaining average provision for the collaboration as a whole?
  - Cannot avoid queuing at some level
  - LHC experiments will talk to Openstack directly – with pilot jobs, need for complex batch queues is reduced
- Current thinking at CERN:
  - Coarse provisioning via Openstack
  - Fine grained via batch clusters
- PaaS general use cases:
  - Batch system with manager and WN's
  - Interactive cluster (e.g. Proof, etc.)
  - Analytics clusters: Hadoop+apps – probably generally interesting

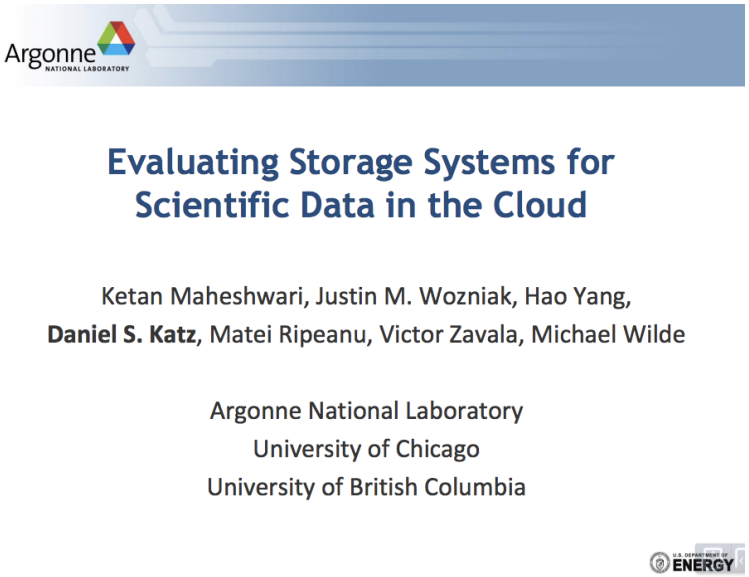
A (currently experiment-specific) Cloud job flow



Can this be made more general?

Networking

- If you transfer data across e.g. AWS regions, you should check carefully bandwidth and latencies.
  - "A cluster of 1000 nodes between Japan and Singapore might be faster than the one between US-east and US-west."
- Smart storage strategies are needed:
  - exploit locality, replication, caching. Carefully chosen storage servers can benefit cloud executions.
- See http://datasys.cs.iit.edu/events/ScienceCloud2014/s05.pdf


Argonne NATIONAL LABORATORY

Evaluating Storage Systems for Scientific Data in the Cloud

Ketan Maheshwari, Justin M. Wozniak, Hao Yang,
Daniel S. Katz, Matei Ripeanu, Victor Zavala, Michael Wilde

Argonne National Laboratory
University of Chicago
University of British Columbia

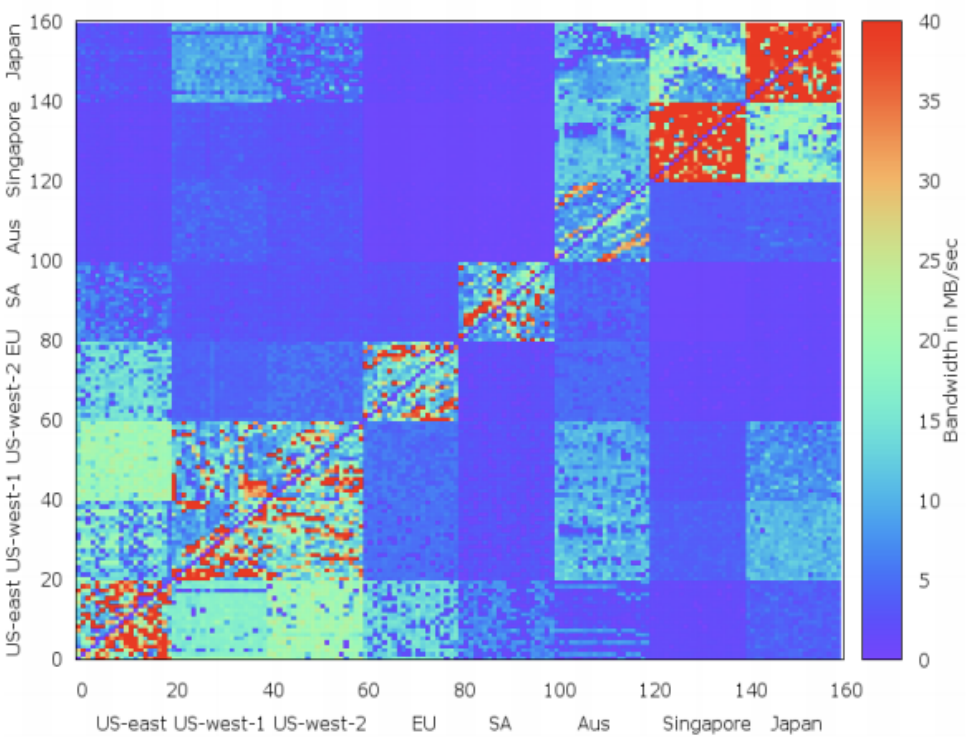
ENERGY

How do we (can we) control networking?

Cloud Regions

Cloud Regions Bandwidths: Some Observations

- North American regions well-connected
- EU well-connected to US-east
- Aus well-connected to US-west and Japan, Singapore
- Japan and Singapore well-connected among themselves but poorly connected with rest of the world

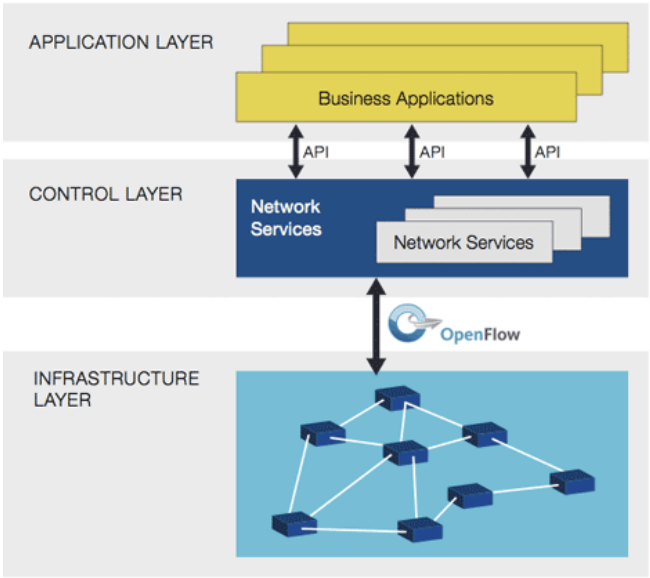


Maheshwari et. al., swift-lang.org

6

OpenFlow-based SDN

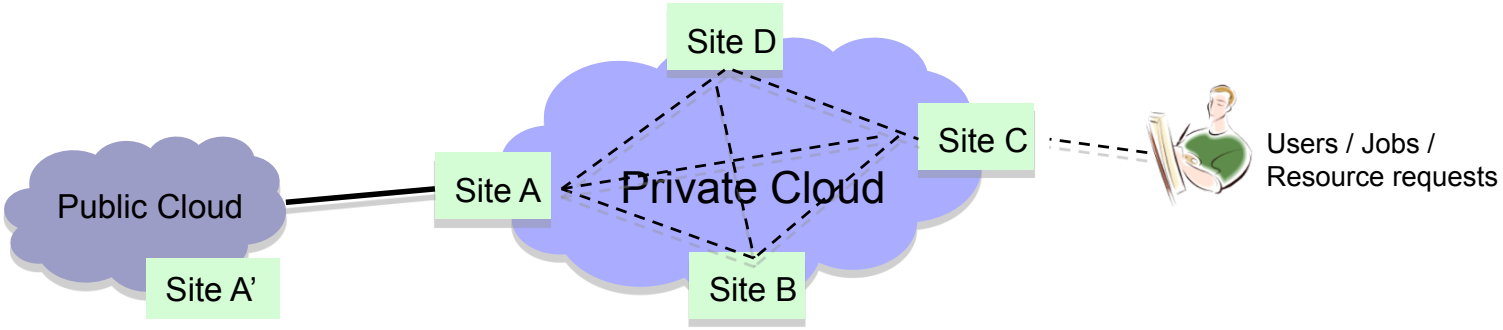
- Network behavior should be directly programmable through Software-Defined Networks (SDN)
- Network services are separated from the network forwarding plane.
- Control and forwarding planes are connected through the OpenFlow protocol (see http://goo.gl/VwMyFX)



Can we assume we have “infinite” bandwidth? How constrained are we by ISP technology support?

Network virtualization

- Dynamically Distributed Virtual Data Centers
- Given the constant increase and commoditizing of network bandwidth, it should be possible to *dynamically*:
  - Exploit hybrid Cloud deployments.
  - Transparently **extend** data centers to remote locations. IP addressing is extended to the remote location, making for a much simpler connection to e.g. storage subsystems than traditional *federations*.



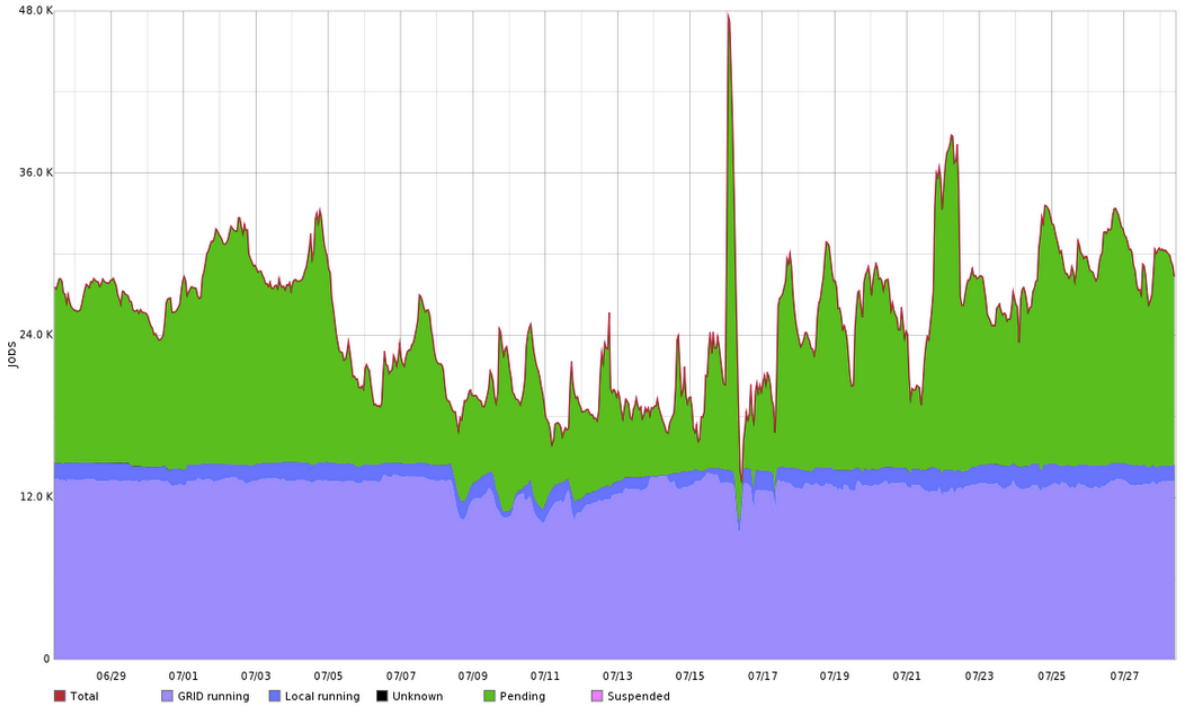
Using VMs in **existing** data centers

- Not every data center is migrating (or even willing to migrate) all its resources to Cloud computing.
- There's often the need to **maintain existing operational workflows, customers, etc.** that often rely on batch systems to prioritize scientific workloads.
- Several systems exist, allowing to manage VMs through some integration with an already deployed batch system
  - One is WNoDeS – Worker Nodes on Demand Service – developed and deployed at the INFN Tier-1 since 2008, and used to handle VMs in the Tier-1 production farm alongside (= in the same cluster of) normal, non-virtualized workloads.
  - Scalability proven to several thousand running VMs, together with several tens of thousands of non-virtualized workloads.
- See e.g. "Accessing Scientific Applications through the WNoDeS Cloud Virtualization Framework", http://goo.gl/mU0ufz

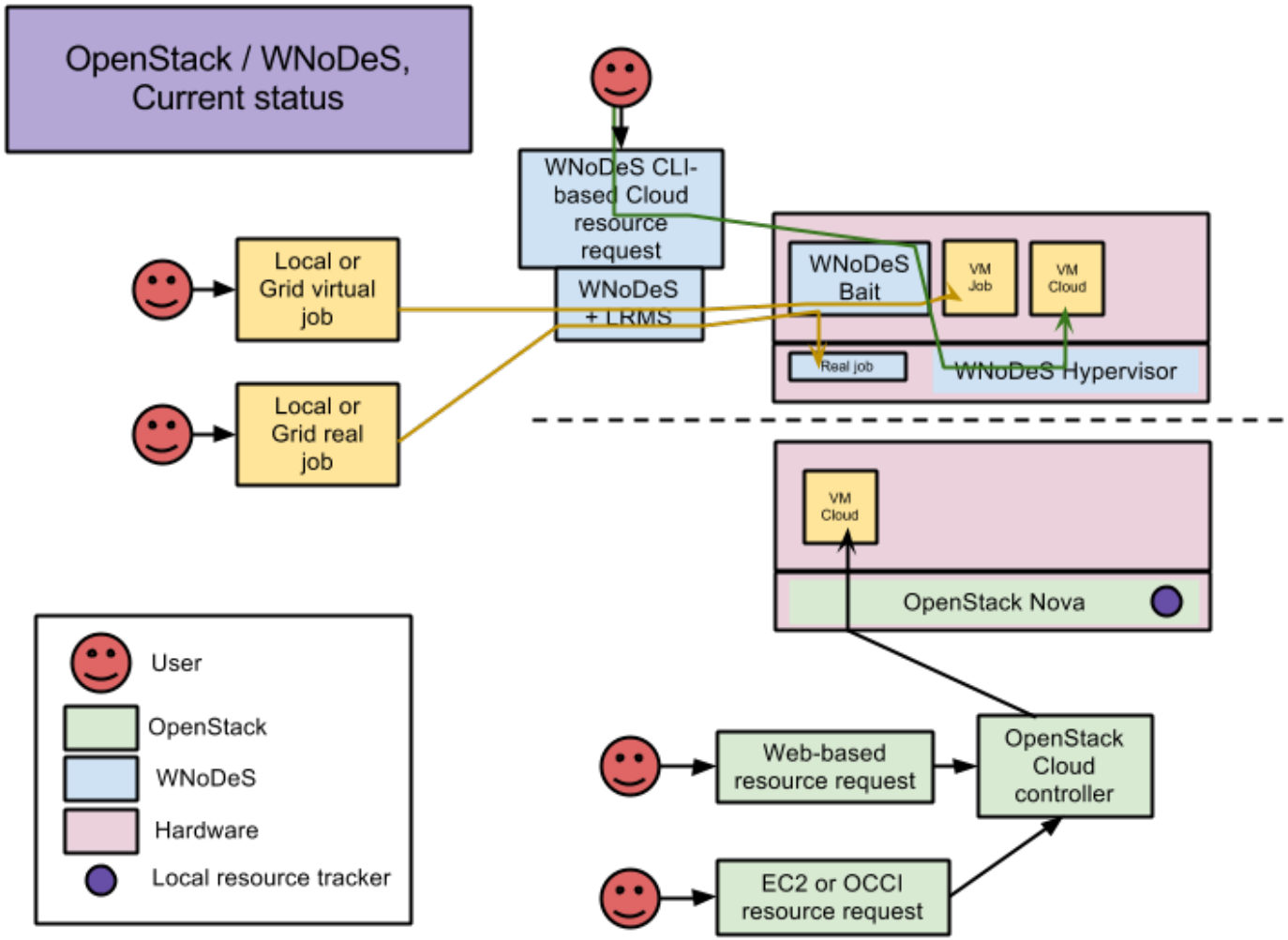
A typical scientific resource center utilization pattern

- This is a typical utilization pattern of the INFN Tier-1 Computing Center.
  - A single shared cluster, serving tens of scientific collaborations.
  - 100% utilized all the time, with **Grid, Cloud and local workloads sharing the same resources**, with or without VMs (on the same hardware).
  - In green you can see the *pending jobs*: several [tens of] thousands of them at any given time.

Problem solved?



The two worlds, separated



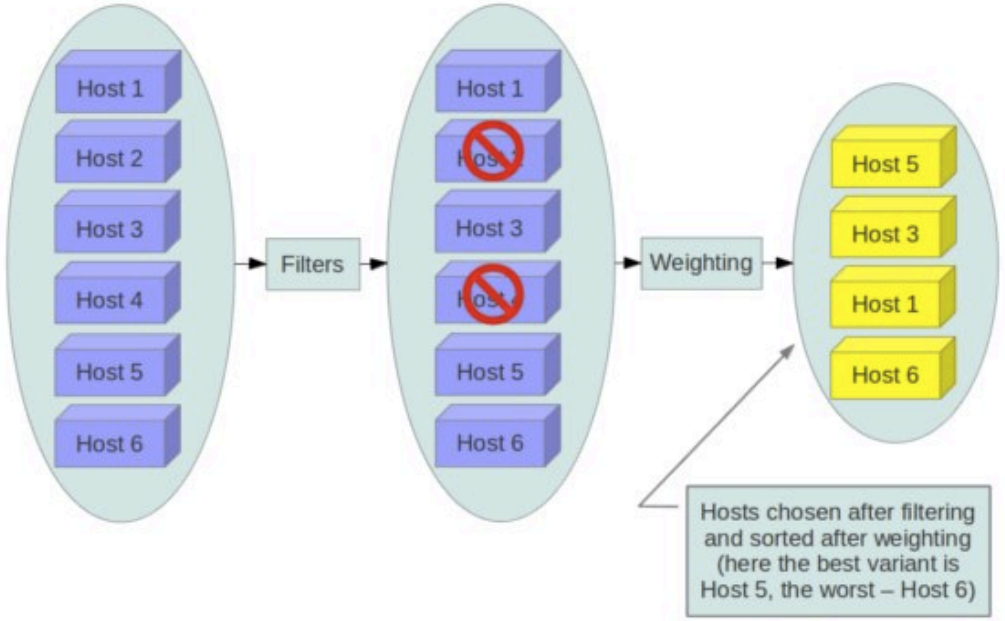
WACK: blending the two worlds

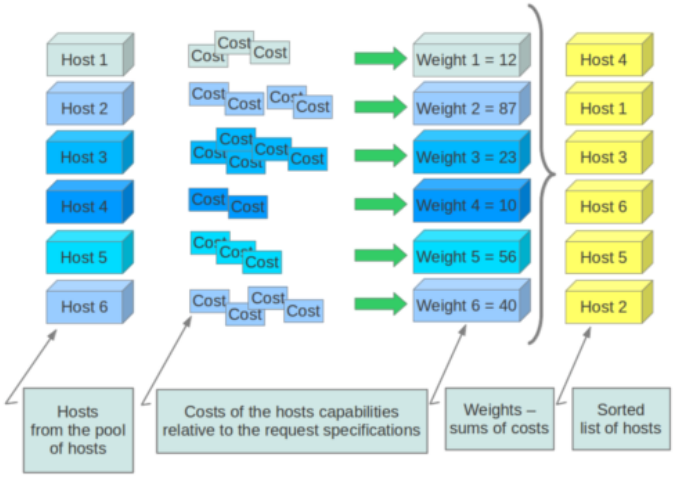
- **WACK**: **W**NoDeS in OpenSt**ack**, an INFN project reusing several OpenStack components with the goal to (guess what) integrate WNoDeS in OpenStack.
  - For example, image management through Glance; virtualization handled directly by Nova; and an EC2 simple interface allows to connect directly to EC2 Clouds.
- This makes it possible to support with minimal changes in a computing center:
  - Traditional workloads
  - VMs running jobs
  - Cloud instantiations through OpenStack APIs
- All within a common cluster (hence, optimization).
- But, this still requires a “traditional” resource center, with batch systems etc.
  - Is this going to be the future?

QUACK: adding queues

- **QUACK**: an INFN project to introduce **Qu**eues in OpenSt**ack**
- Let's have a look at the OpenStack Nova scheduler
  - See http://goo.gl/sLXCo and http://goo.gl/o4Swo
  - The important point: no free resources → no VM
  - This is the negation of the *infinite capacity* postulate

```
All compute nodes (also known as hosts in terms of OpenStack) periodically publish
their status, resources available and hardware capabilities to nova-scheduler through
the queue. nova-scheduler then collects this data and uses it to make decisions when
a request comes in.
```





QUACK (2)

- The first assumption is that **it is not required that a batch system is running at a computer center**.
- We need to alter the existing Nova scheduler to introduce scheduling policies into the scheduler itself.
  - But we don't want to write a scheduler ourselves.
  - INFN is prototyping a modified version of the OpenStack Nova scheduler using Nova plug-ins that re-uses the **SLURM fair-sharing engine**.
- This would allow one to provide flexible resource allocation queuing and fair-sharing to Cloud tenants.
  - Extremely useful for scientific applications – probably much less so for interactive (or "fast") usage.

"What is fair?" (Miron)

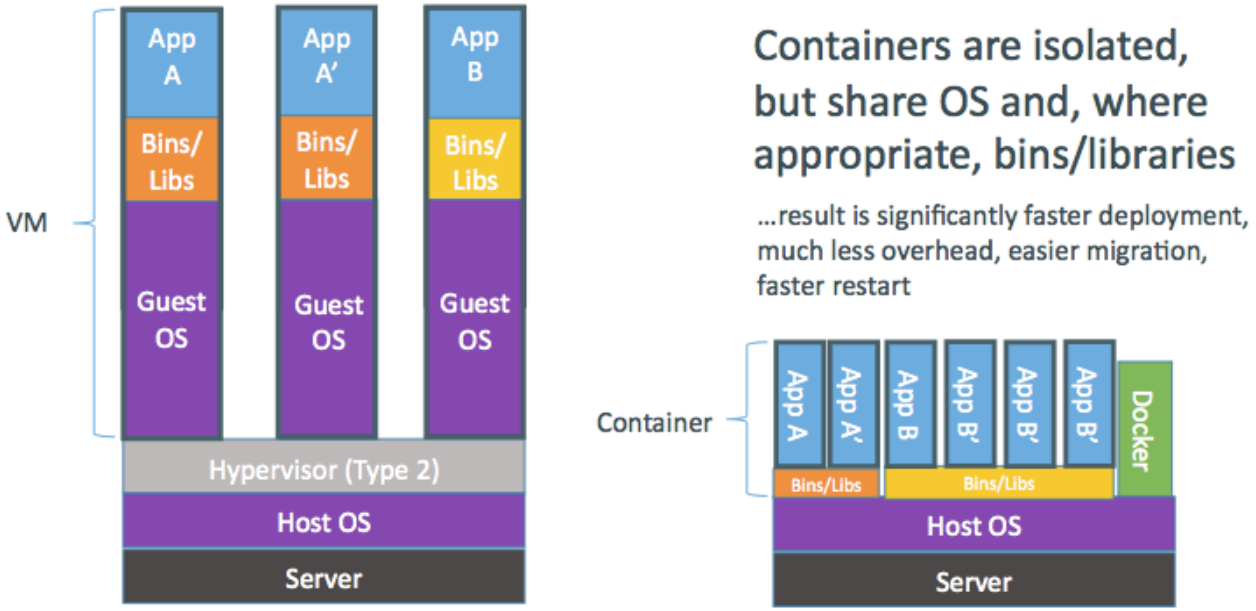
Would you say this raises interest in the Cloud industry? Why?

Virtualization penalties

- We already said that virtualization introduces some penalties.
  - With regard to CPU and esp. I/O performance.
  - Or in the inability to effectively manage some specialized devices, such as GPUs, or fast network interconnections (like InfiniBand).
- What about a more lightweight way of doing virtualization that might overcome the issues above?

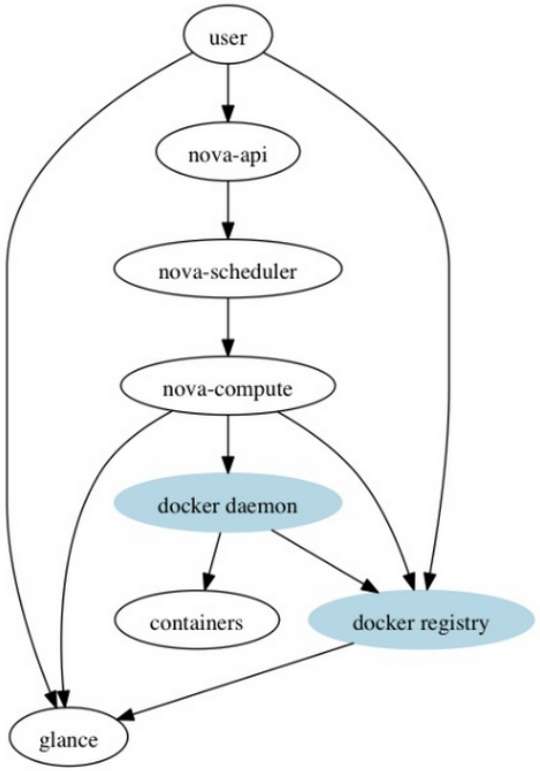
Nova drivers

- OpenStack Nova supports several hypervisors (Xen, KVM, etc.)
- But Nova is not limited to hypervisor or to VMs! Two examples: *containers* and *bare metal*.



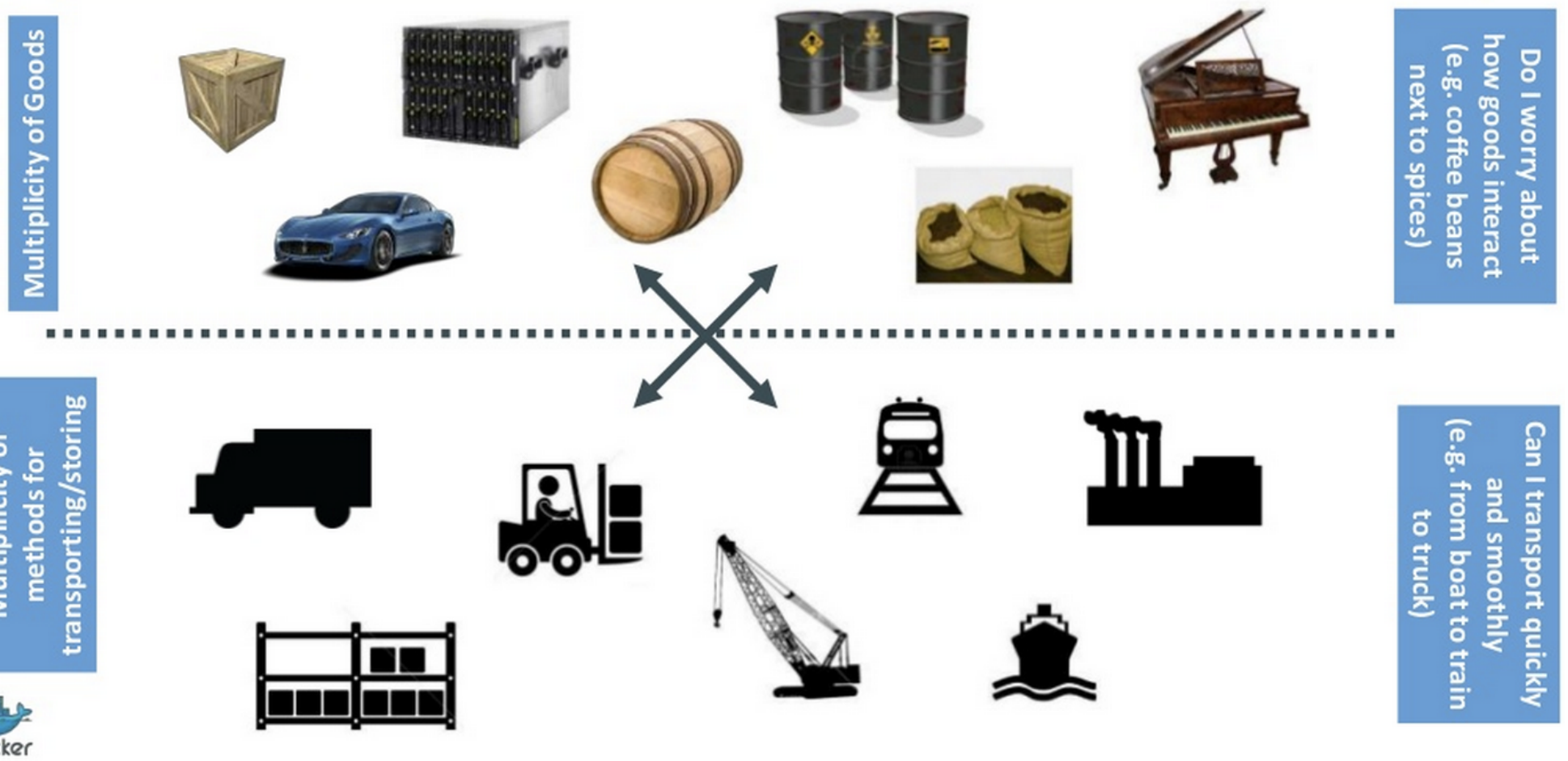
Containers: Docker

- Docker (http://goo.gl/lnm8jQ) is a project for the creation of *lightweight* containers, portable and self-consistent for any application.
- Important: the focus here is on *applications*, not on virtual machines.
- Containers: “chroot on steroids”
  - A Linux system within Linux.
  - A group of processes in an isolated environment. From the inside, it looks like a VM.
  - Controlling use of resources such as CPU, memory and I/O through well-known Linux features such as cgroups.
- Can be integrated with Nova

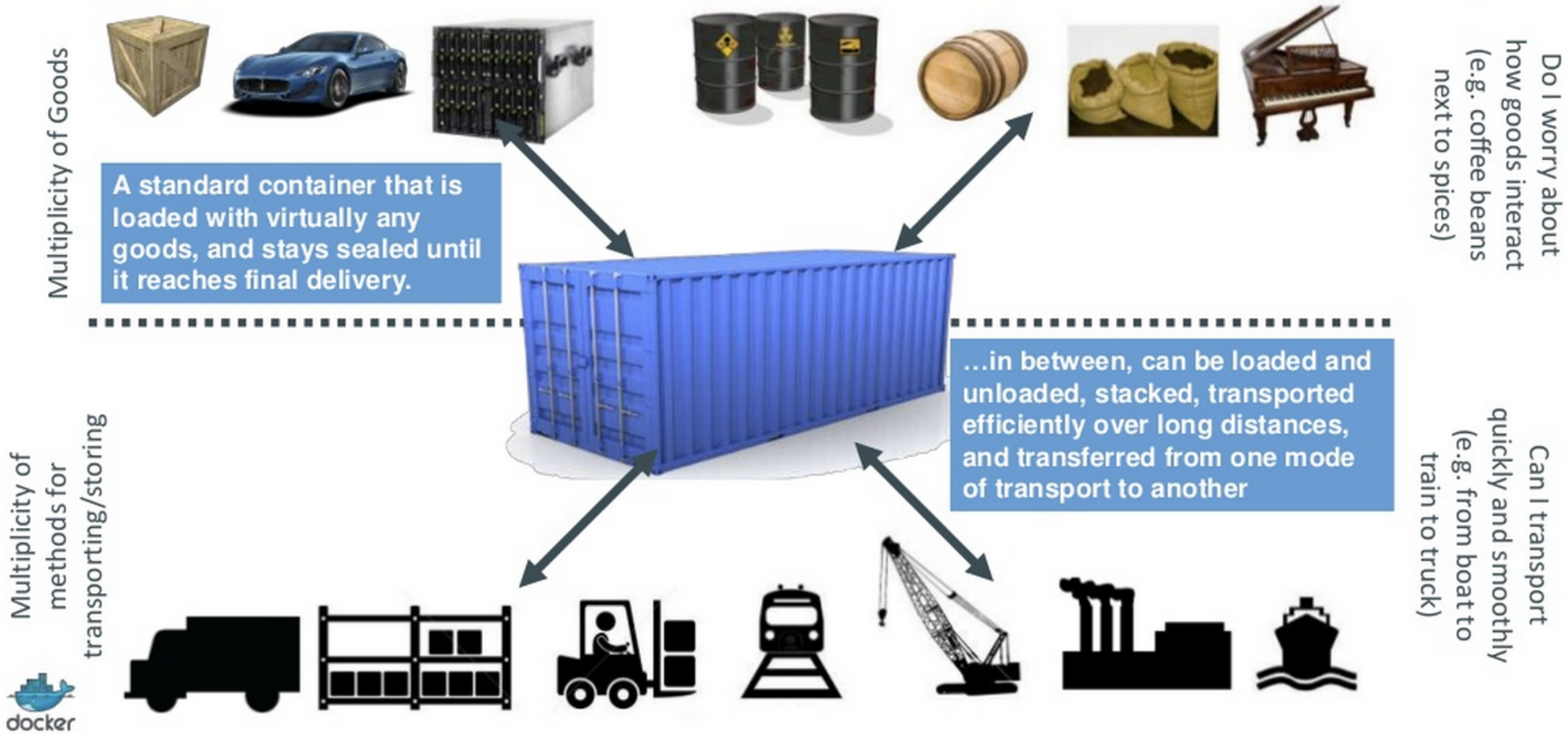


Nova+Docker Architecture Overview

Cargo Transport Pre-1960



Solution: Intermodal Shipping Container



Intermodal Shipping Container Ecosystem

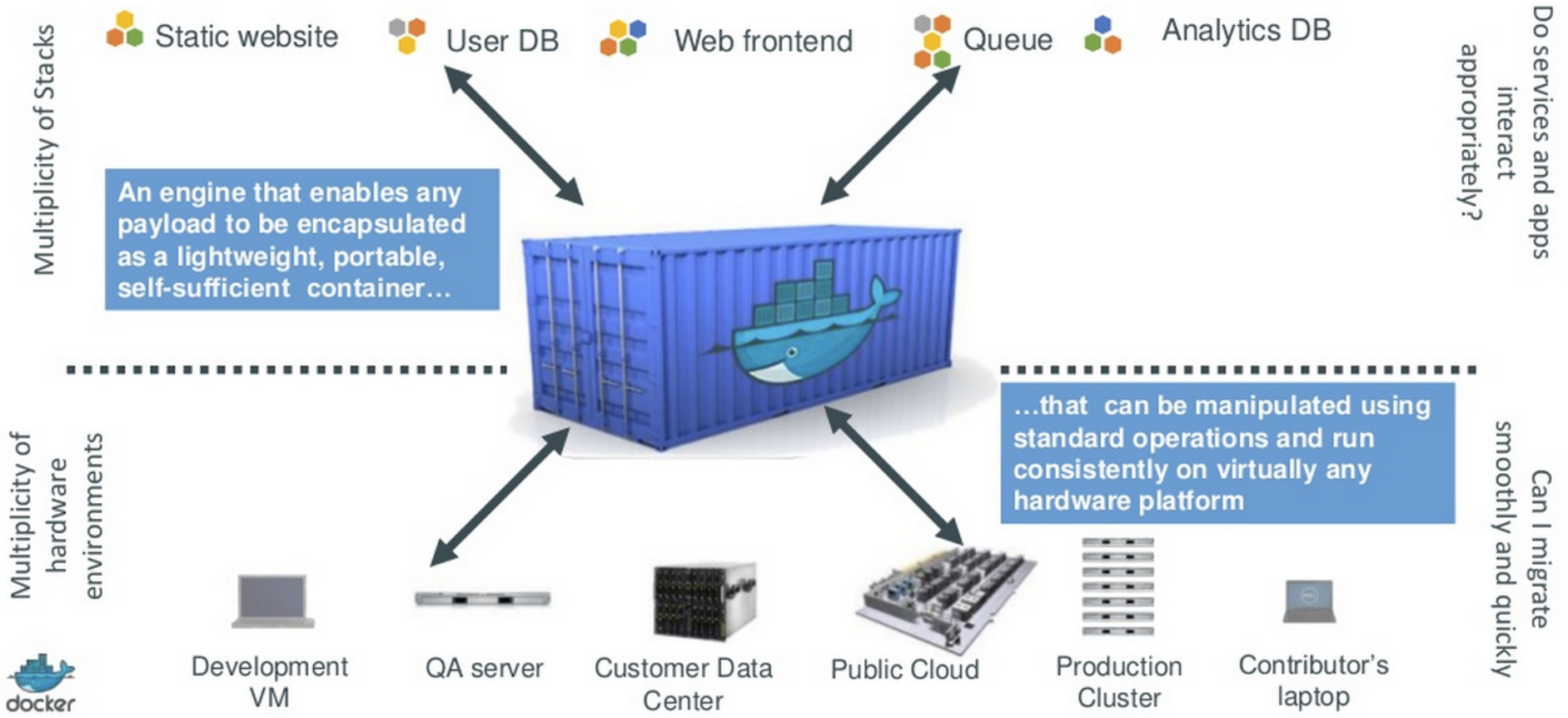
- 90% of all cargo now shipped in a standard container
- Order of magnitude reduction in cost and time to load and unload ships
- Massive reduction in losses due to theft or damage
- Huge reduction in freight cost as percent of final goods (from >25% to <3%)

→ massive globalizations

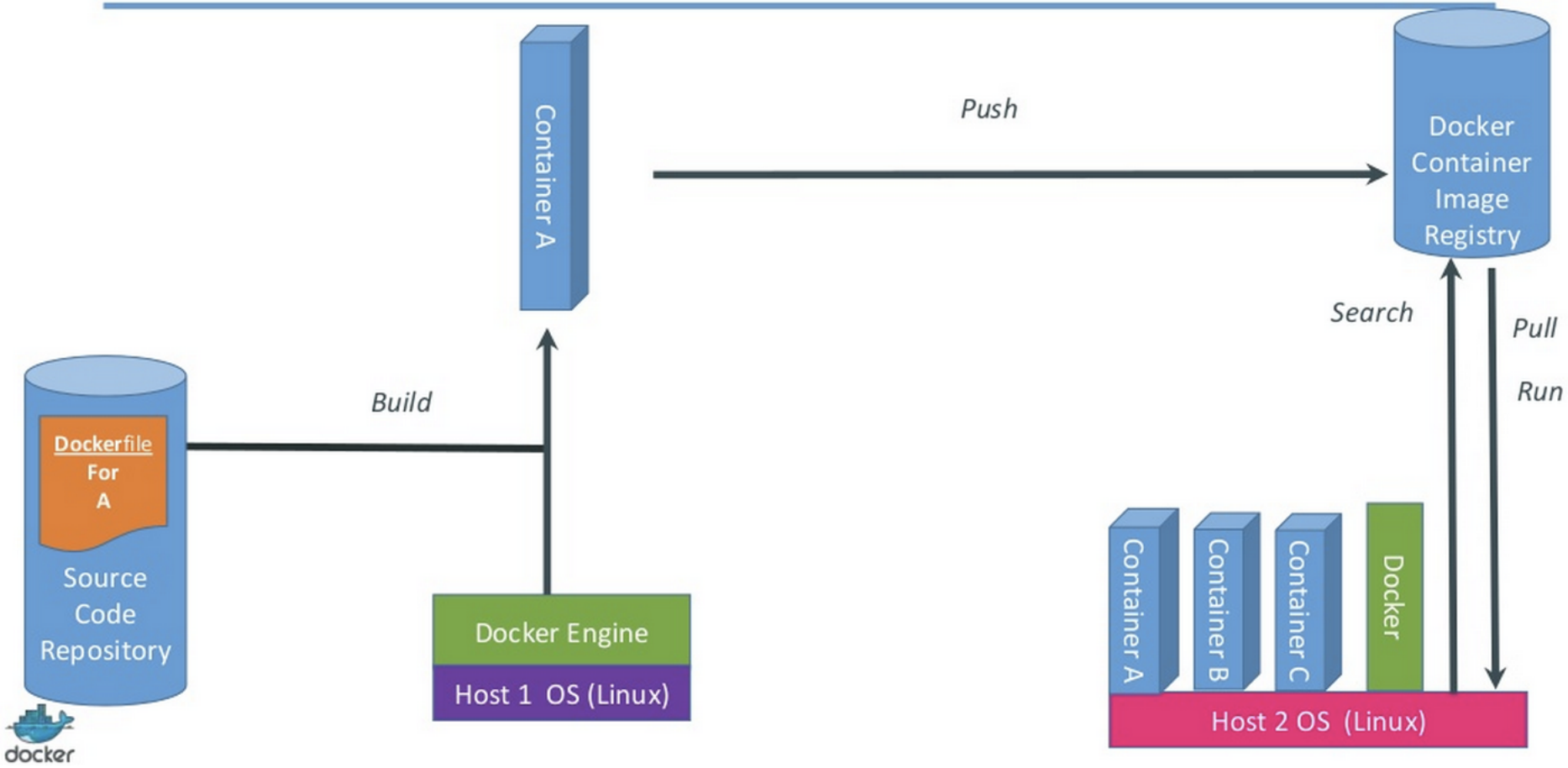
- 5000 ships deliver 200M containers per year

docker

Docker is a shipping container system for code



What are the basics of the Docker system?



Docker in summary

Is this the future for software delivery?

- **Portable deployment across machines.**
  - Docker defines a format for bundling an application and all its dependencies into a single object which can be transferred to any docker-enabled machine, and executed there, with the guarantee that the execution environment exposed to the application will be the same.
- **Application-centric.**
  - Docker is optimized for the deployment of applications, as opposed to machines.
- **Automatic build.**
  - Docker includes tools for developers to automatically assemble a container from their source code, with full control over application dependencies, build tools, packaging etc.
- **Versioning.**
  - Docker includes git-like capabilities for tracking successive versions of a container, inspecting the diff between versions, committing new versions, rolling back etc.
- **Component re-use.**
  - Any Docker container can be used as a "base image" to create more specialized components.
- **Sharing.**
  - Docker has access to a public registry (http://index.docker.io) where thousands of people have uploaded useful containers.
- **Tool ecosystem.**
  - Docker defines an API for automating and customizing the creation and deployment of containers.

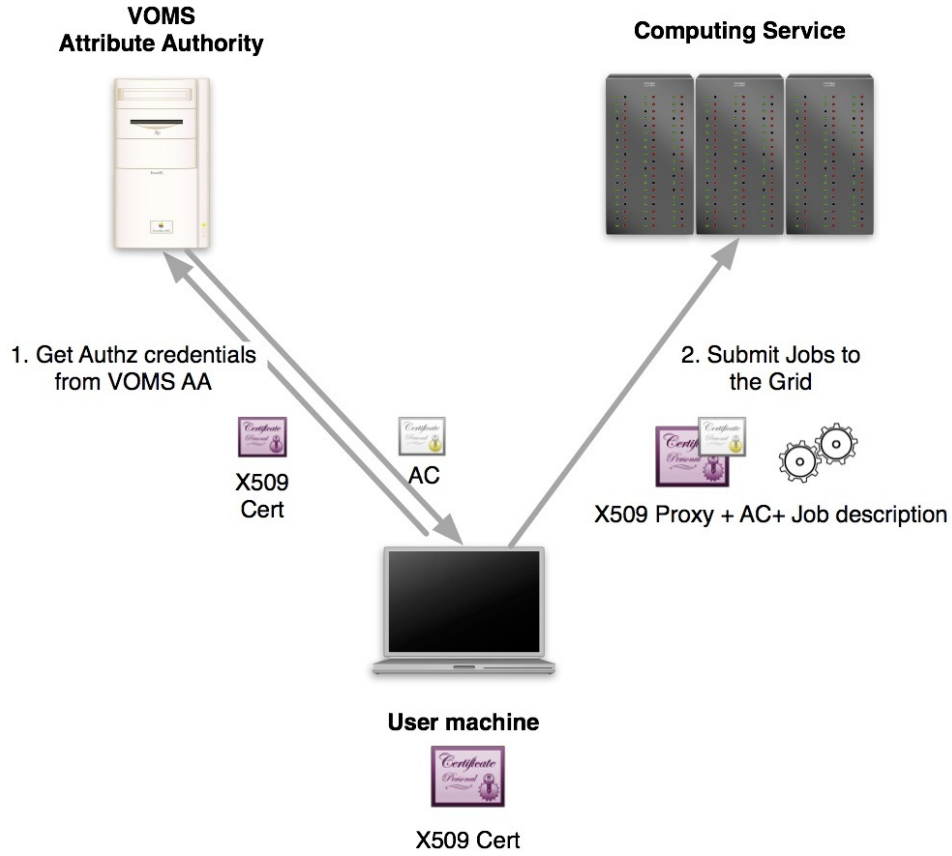
Distributed Authorization

- In the Grid world we have ways to define and manage Virtual Organizations (VO), and to create, propagate and manage access control policies.
- **We need something similar for Cloud computing**, without being limited by the technologies used in the Grid world.
- See Identity and Access Management (IAM) policies and roles on Amazon EC2 for a starting point (http://goo.gl/rRcBHa).

The Virtual Organization Membership Service (VOMS)

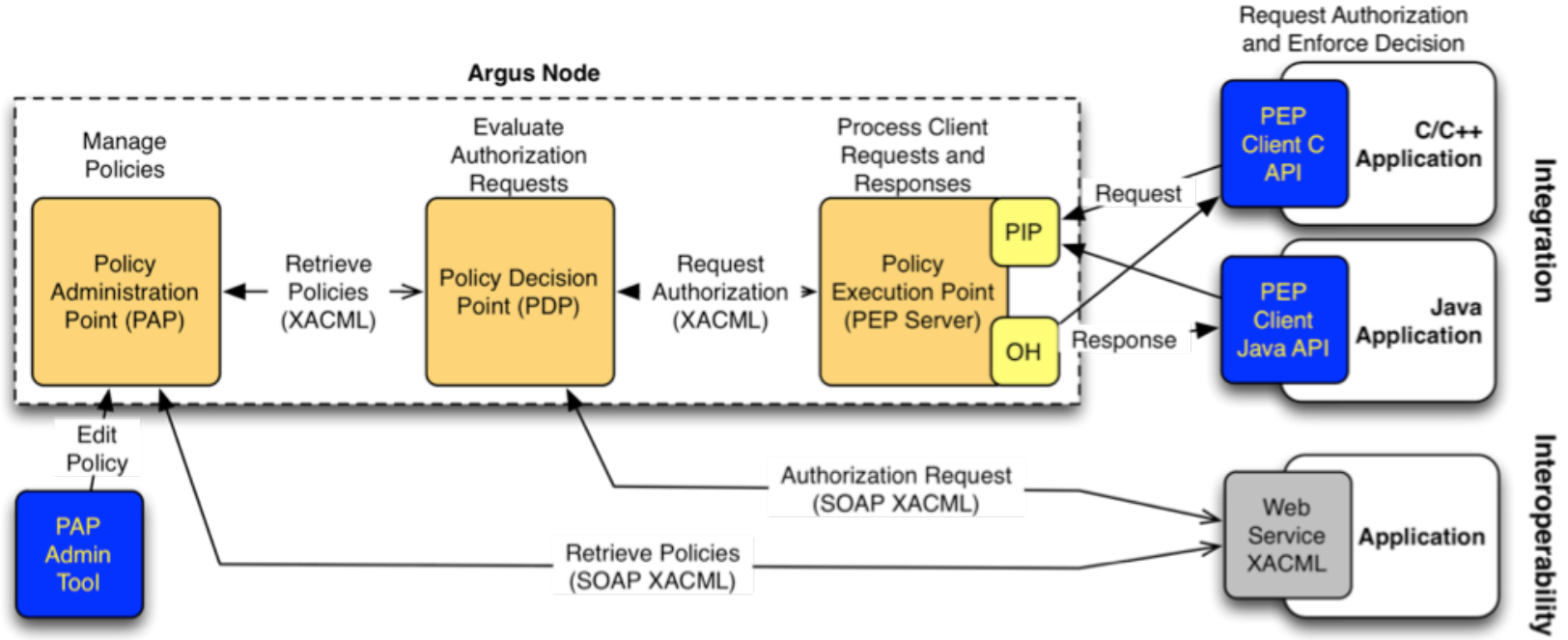
- **Attribute authority**
  - Issues attributes (in the form of X509 attribute certificates or SAML assertions) expressing membership information of a subject within a Virtual Organization (VO).
  - These attributes are used to authorize access to Grid resources.
- **VO Registration and management service**
  - Users must be registered at a VO in order to obtain credentials that can be used to access Grid resources.
  - Administrators handle user registration requests and define the structure of the VO by defining groups, roles, attributes that will be assigned to users.
- See http://italiangrid.github.io/voms

VOMS-based authorization



- Deployed on a large-scale production infrastructure for many years.

The Argus authorization service



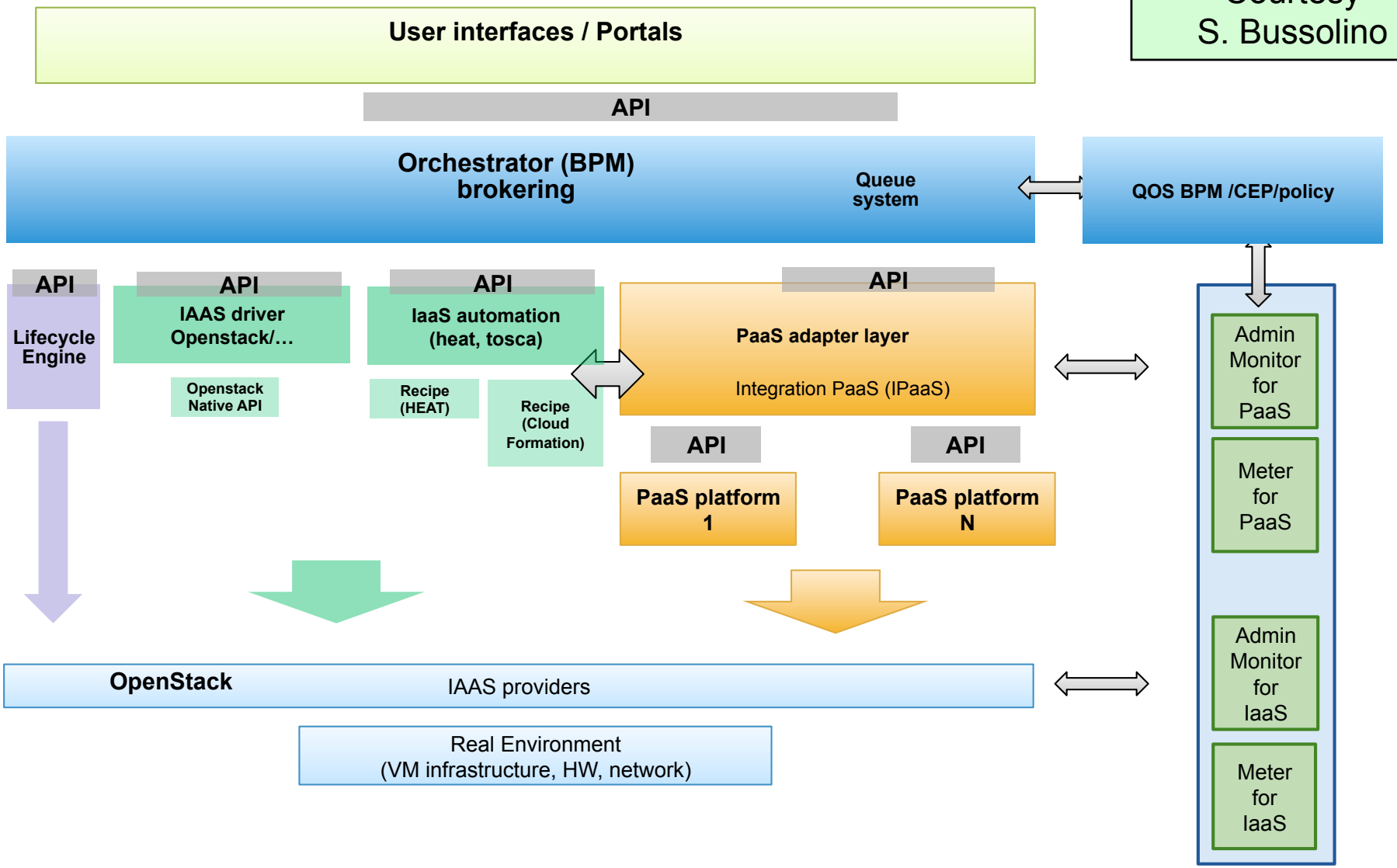
- A flexible and generic authorization system:
  - Based on the XACML 2 standard.
  - Renders consistent authorization decisions based on XACML policies.
  - Supports aggregation of policies from remote Argus endpoints (hierarchical policies composition).
  - Deployed on a large-scale production infrastructure.

PaaS architecture (1/2)

- We (kind of) know what PaaS is. But how do we abstract from what the market provides us with, so that we are able to use *all* resources that might be available to us?
- Let's assume that IaaS offerings are pervasive, simple enough, and based on well-known standards.
  - Be they *de facto* (such as EC2 or S3) or *de jure* (such as OCCI or CDMI) standards.
- Which components are still needed?

PaaS architecture (2/2)

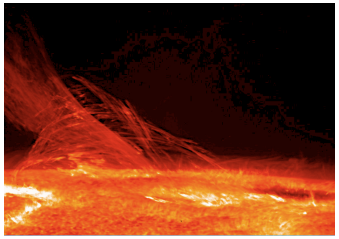
Courtesy
S. Bussolino



PaaS solutions

- The use of **computing clusters** is still very popular in both HTC and HPC communities. But in order to reduce upfront investments and maintenance costs, one would like to have *custom virtual clusters*.
  - Not really new, in both the open source (e.g. Hadoop, Nimbus) and commercial worlds.
  - However, we still need ways to:
    - *Dynamically* create virtual clusters. I.e., elastic adaptation to the workload submitted by users, adding / removing nodes.
    - Move beyond job-level solutions. Can I *get access* to my nodes?
    - Do not depend on specific LRMS (batch system) implementations.
    - Do not depend on specific Cloud infrastructures (EC2, for example).
    - If desired or necessary, take into account something more sophisticated than “number of instances”, e.g. cost-aware schedulers (whatever “cost” means).
    - Easily express requirements that define what a cluster of *resources* (not necessarily just VMs) should look like and what should it contain (or contain not).

Heat



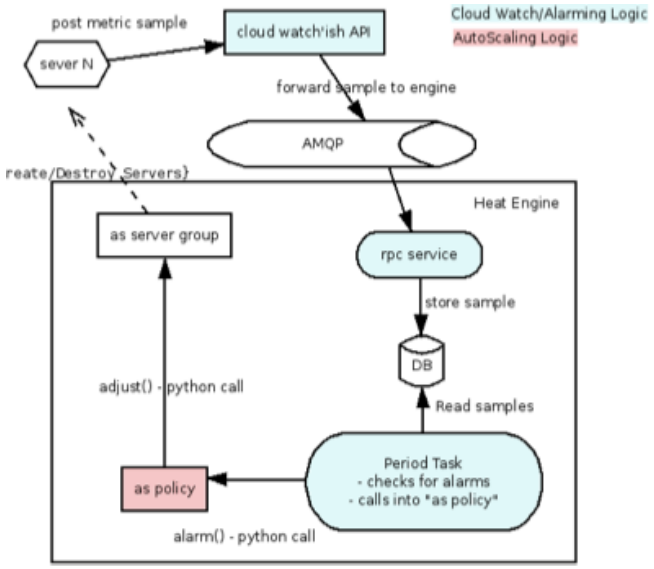
- Heat (http://goo.gl/7j8fv) is a component of the OpenStack Orchestration program.
- It allows creation of Cloud applications defined by text templates. Heat *resources* (not necessarily just VMs) communicating among them, define **stacks**. Stacks:
  - Can be integrated in automated configuration tools, such as Puppet.
  - Can automatically manage high-availability and auto-scaling configurations.

Heat templates

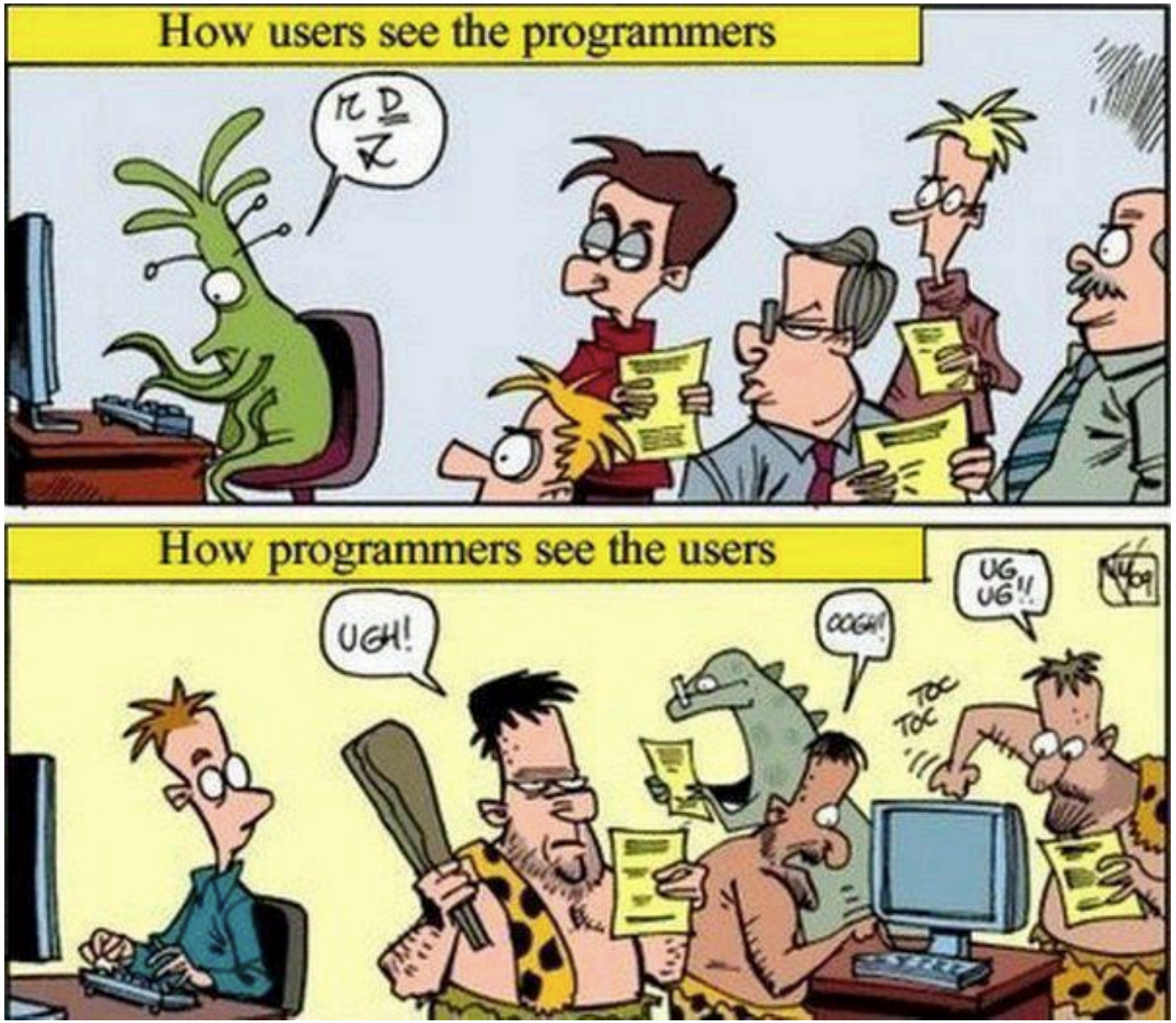
- The Heat native format for templates in undergoing some changes. The idea, however, is to make it compatible with the AWS Cloud Formation (http://goo.gl/4tvGl) templates.
- Just as an example, this is a template to instantiate a VM on a private network.

```
heat_template_version: 2014-05-14
description: Test Template

parameters:
  ImageID:
    type: string
    description: Image use to boot a server
  NetID:
    type: string
    description: Network ID for the server

resources:
  server1:
    type: OS::Nova::Server
    properties:
      name: "Test server Net"
      image: { get_param: ImageID }
      flavor: "m1.small"
      networks:
      - uuid: { get_param: NetID }

outputs:
  server1_private_ip:
    description: IP address of the server in the private network
    value: { get_attr: [ server1, first_address ] }
```

Heat auto-scaling



- In OpenStack Havana:
  - AWS::AutoScaling::AutoScalingGroup
  - AWS::AutoScaling::ScalingPolicy
- In OpenStack Icehouse, also:
  - OS::Heat::InstanceGroup
  - OS::Heat::AutoScalingGroup
  - OS::Heat::ScalingPolicy
- ScalingGroup – a group that can scale an arbitrary set of Heat resources.
- ScalingPolicy – affects the number of scaling units in a group (+1%, -10%, etc.)
- Have a look at http://goo.gl/K94wg3 for a template that creates an auto-scaling Wordpress cluster with load balancing, a scaling policy based on Ceilometer (metering) alarms and user data (could be a Puppet template) used to install packages and customize VMs.



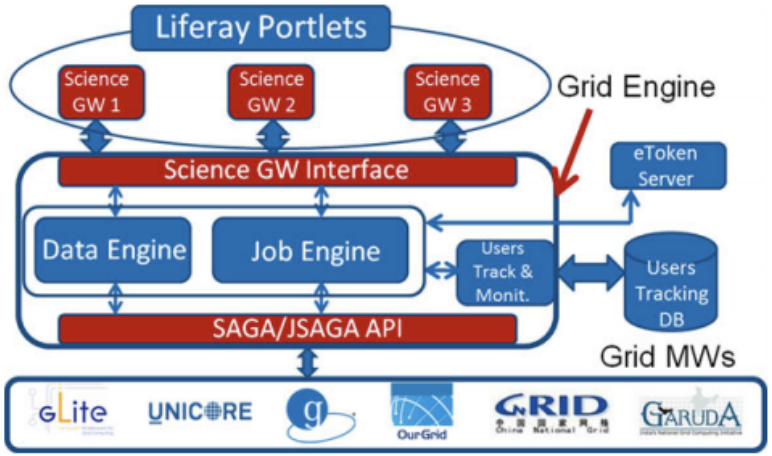
Source: http://goo.gl/wT8XEq

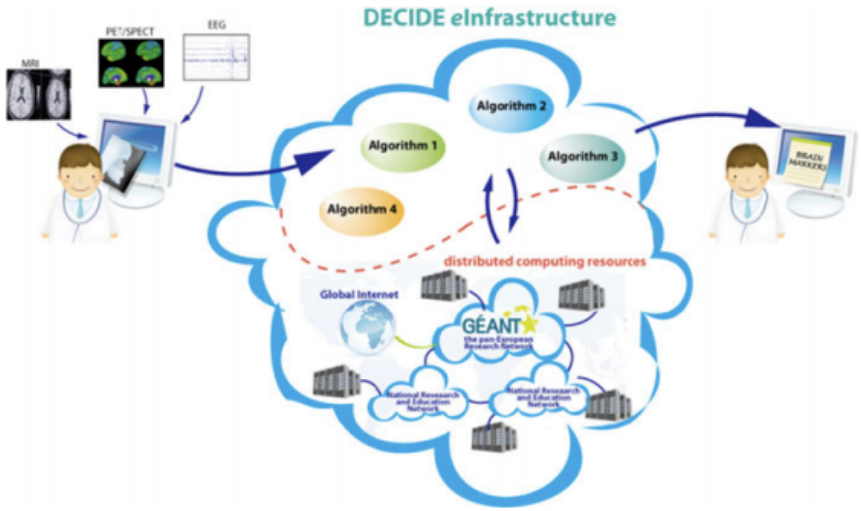
Portals

- Trying to fill the gap between technology and users, esp. for communities not really committed / fluent / interested in technological details.
- A practical example: DECIDE = Diagnostic Enhancement of Confidence by an International Distributed Environment.
  - This a **combination** of layers involving **research networks, Grid resources and domain-specific applications**.
  - A EU 7th Framework Program project, based on the GEANT network and relying on the EGI infrastructure.
  - The goal: computer-aided extraction of diagnostic disease markers for Alzheimer disease and schizophrenia.
    - Each year, 1.4 million people in Europe develop some form of dementia (one every 24s).

DECIDE

- A Science Gateway based on Liferay portlets and on worldwide standards.
  - See V.Ardizzone et al, The DECIDE Science Gateway, J Grid Computing (2012) 10:689-707





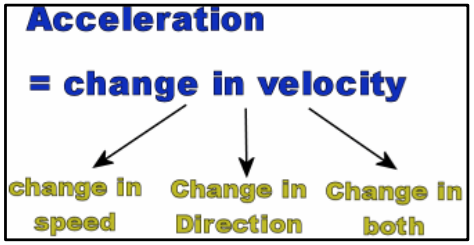
Some questions

- **Why does this deserve a publication?** Or special funding?
  - I.e.: how do you make this the norm rather than the exception?
- Would it be possible to structure this around API's that can be used by portals, desktops and mobile applications, and that can interconnect – using Cloud parlance – IaaS, PaaS and SaaS?
- And how do you support big data workflows for eScience, *easily* integrating them into these portals?

Agenda

- Cloud computing: definition & technology recap
- Some pros and cons
- Apps in the Cloud?
- Is anything missing?
- **Conclusions**

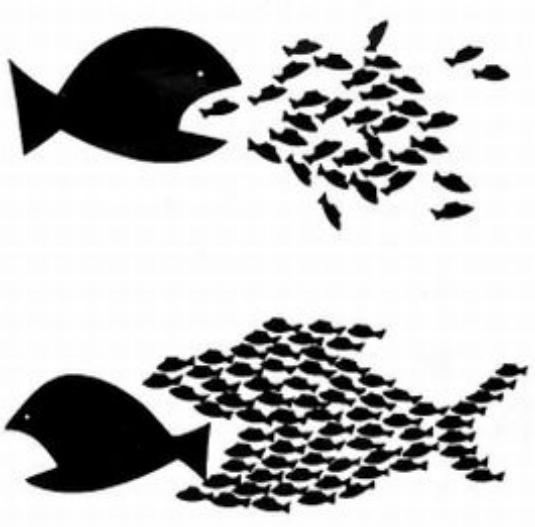
How to evolve/sustain ourselves?



- Trends:
  - Use and extend what's available on the market
  - Look for assistance to deploy and support it
  - Which important / essential features are we still missing? Will there be somebody interested in implementing / maintaining them?
    - Watch out for the "I am capable of writing everything myself, and in more efficient way than you" syndrome
- Are we ready / capable of adapting ourselves (our applications) to market offerings?
- Sustainability… Community (which community) support? Liaise with industry to entice them in supporting us? Assume external (e.g. EC) funding will sort this out for us?
- Is it fundamentally sustainable *not* to rely on externally funded e-infrastructures for Science?

(Some provisional) Conclusions

- Currently, the reference platform for public Cloud computing (technologically and market-wise) is still Amazon, offering numerous advanced services, at a very large scale.
  - There are, however, several reasons justifying the existence of private Clouds or, in general, of Cloud infrastructures that are not Amazon's.
  - In terms of open Cloud frameworks, OpenStack (which we did not discuss here) has at this time a remarkable popularity and expansion; but this is actually one of the reason why it is somewhat difficult to make sense of all its evolutions and proposed solutions, sometimes apparently running after Amazon.
  - But a crucial point is that customers (be they scientists or not) wish to deploy applications without being confined to technological limit imposed by single vendors.
  - *This*, together with the strong evolution we observe from IaaS to PaaS and SaaS models due to the added value given by these abstractions, is what will drive the evolution of Cloud frameworks (OpenStack included) in the coming months.
  - There are still missing components that are key to scientific communities "in the Cloud" to be developed, made available, exploited, and this from several viewpoints: technology, economics, vision, policy.
  - Last but not least, let us remind ourselves of the strategic and business importance, in particular for private and community Clouds linked to universities, research centers and public administrations, of sharing methods and technological solutions through an effective federation of know-how and resources.



Cloud computing, then…

- "It's this nonsense. What are you talking about? It's not water vapor! All it is, is a computer attached to a network. You just change a term, and think you've invented technology."
  - Larry Ellison, co-founder and CEO, Oracle Corporation, September 2009
- "Q: what's the Oracle strategy on Cloud computing? A: Oracle has two main goals on Cloud computing. The first is to make sure it is ready for adoption […] The second is to support both private and public Cloud computing."
  - Oracle Cloud Computing FAQ, October 2010
- "Unlike competitors with narrow views of the cloud, Oracle provides the broadest, most complete, and integrated cloud offerings in the industry."
  - Oracle Cloud Computing, May 2014 (http://goo.gl/oDVOtA)
- **"The truth is rarely pure and never simple."**
  - Oscar Wilde, The important of Being Earnest, 1895

Acknowledgments and more information

- This lecture was made possible also through important contributions of the INFN CNAF R&D Group.

- For more information:
  Davide.Salomoni@cnaf.infn.it
  http://www.linkedin.com/in/davidesalomoni

At last, the fossil evidence to prove our theory! The dinosaurs died off – not because of a meteor or climate change – but because their cloud computing platform collapsed!

Source: http://goo.gl/gnol3z