

Distributed Cloud Environment for PL-Grid Applications

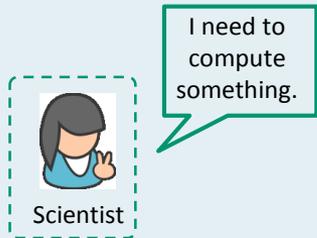
Piotr Nowakowski, Tomasz Bartyński, Tomasz Gubała, Daniel Haręźlak, Marek Kasztelnik, J. Meizner, P. Suder, M. Bubak
ACC CYFRONET AGH

KUKDM 2015

Zakopane, 12 March 2015

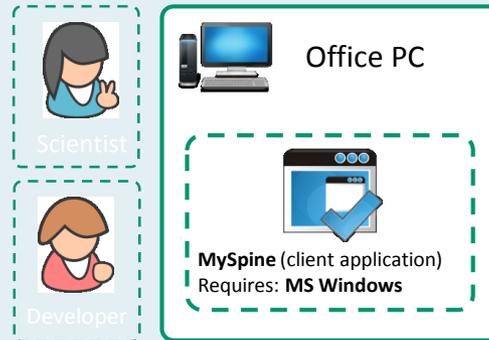
A (very) brief introduction

The challenge



Computations are an inherent part of modern e-science, particularly within the life sciences domain. As the available IT tools grow ever more sophisticated, domain scientists require help from scientific programmers and other IT specialists to be able to perform their research.

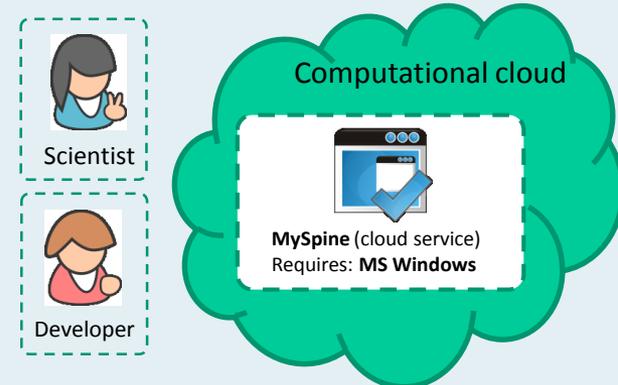
Where we are



When faced with a computational task, the first reaction is to either install the necessary software by oneself or call in help from the IT department. Either way, such traditional setups carry serious drawbacks:

- **You need to provision your own hardware (typically an office PC)**
- **Your application is only accessible from one place (typically your office)**
- **Applications and data cannot be easily deployed on other computers**

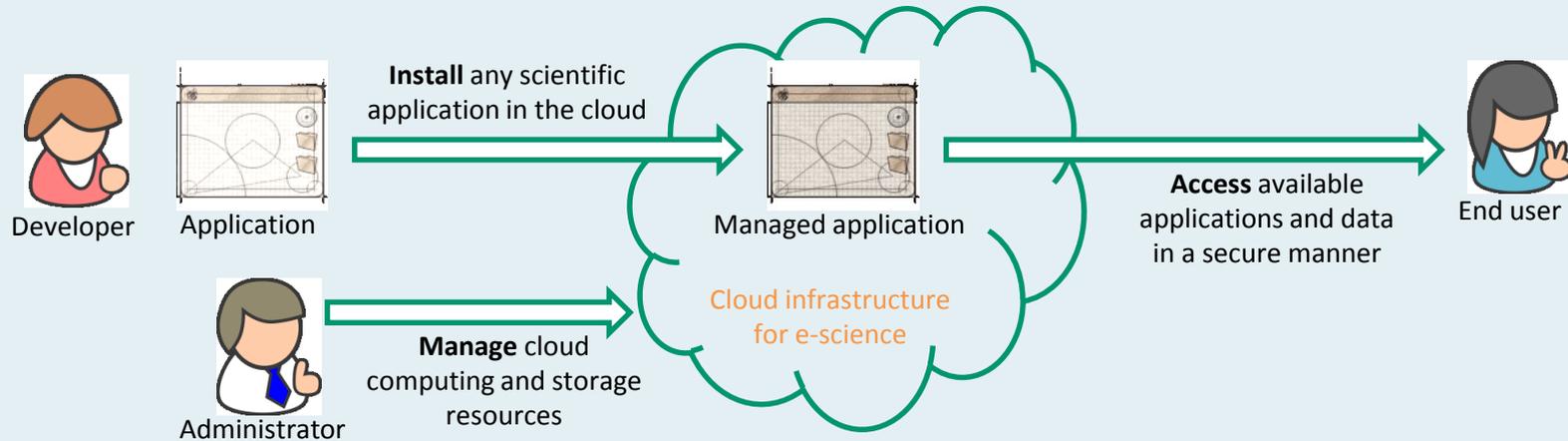
There's a better way!



Computational clouds enable us to avoid these problems entirely. A cloud-based service can perform all the functions of a locally running application, with the following benefits:

- **The hardware is provided by the cloud operator (and can be vastly more powerful than any local resources!)**
- **A cloud application is available from anywhere**
- **Once deployed, the application can be accessed by many users**

Basic functionality of the cloud platform



- Install/configure each application service (which we call a **Cloud Service** or an **Atomic Service**) once – then use them multiple times in different workflows;
- Direct access to raw virtual machines is provided for developers, with multitudes of operating systems to choose from (IaaS solution);
- Install whatever you want (root access to cloud Virtual Machines);
- The cloud platform takes over management and instantiation of Cloud Services;
- Many instances of Cloud Services can be spawned simultaneously

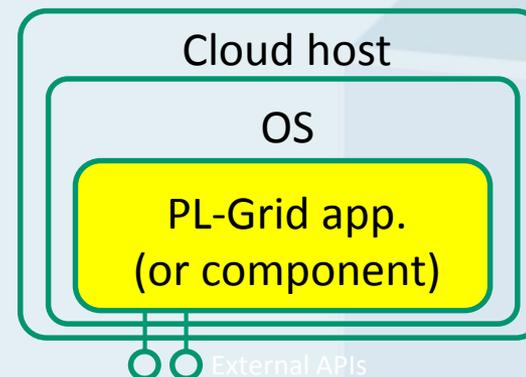
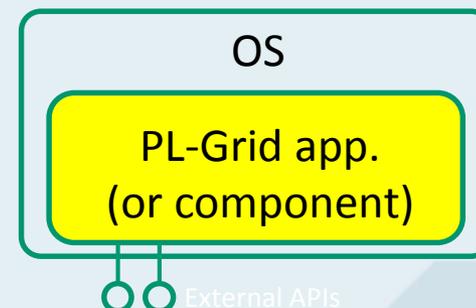
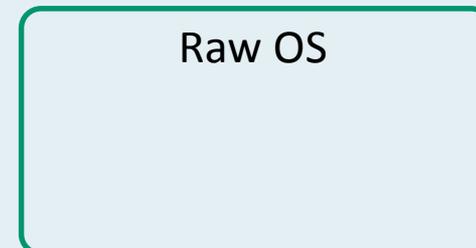


A (very) short glossary

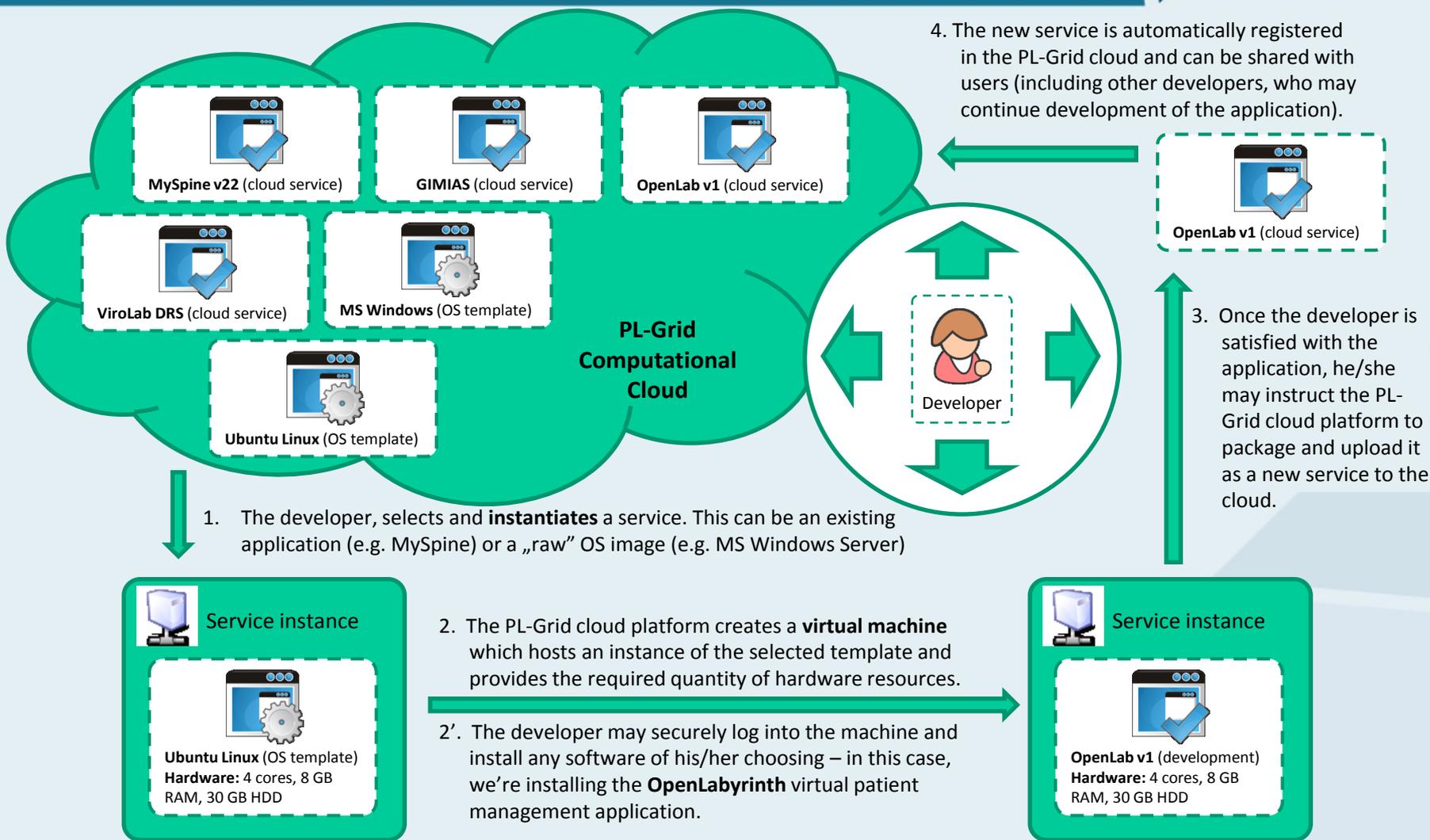
Virtual Machine: A self-contained operating system image, registered in the Cloud framework and capable of being managed by PL-Grid mechanisms.

Cloud service: A PL-Grid application (or a component thereof) installed on a Virtual Machine and registered with the cloud management tools for deployment.

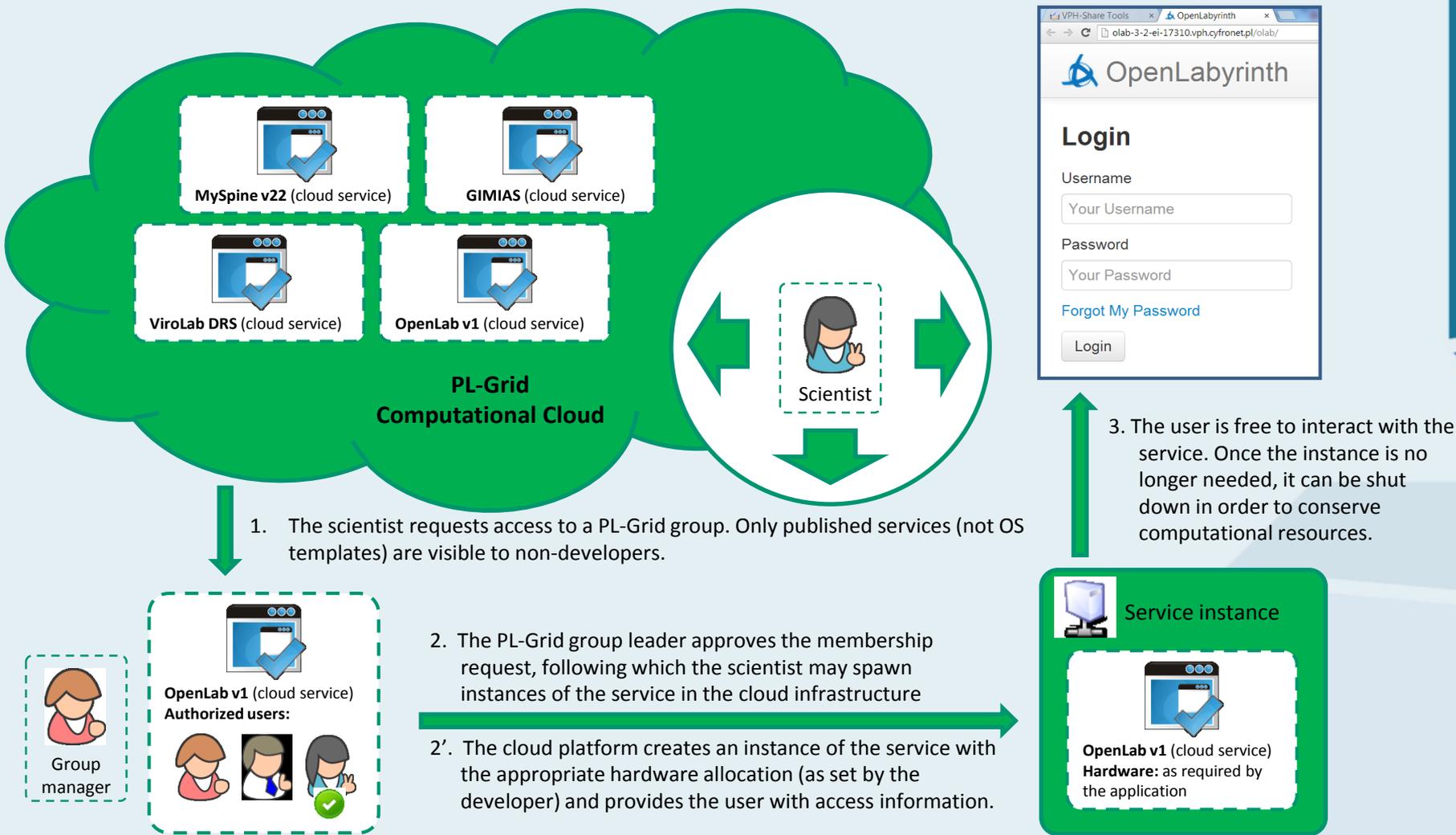
Cloud service instance: A *running* instance of a cloud service, hosted in the cloud and capable of being directly interfaced.



PL-Grid Computational Cloud: developers' view



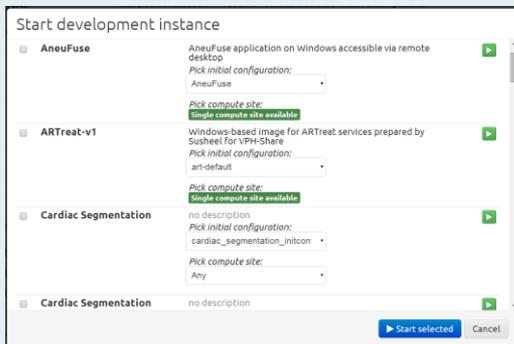
PL-Grid Computational Cloud: end users' view



Cloud platform interfaces



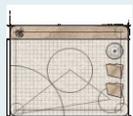
End user



Running Development Instances					
Name	IP	Location	Status	Cost	Actions
WebDRS_development_instance	10.100.8.93	Cyfronet	active	\$0.07	[stop] [refresh] [delete]
Final public release of WebDRS. Comes bundled with ranking mailer and Sierra testing service.					
Web Applications					
/	http ok	https ok			
WS/REST Services					
No services					
Other services					
ssh	149.156.10.132:55484				

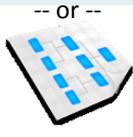
A full range of user-friendly GUIs is provided to enable service creation, instantiation and access. A comprehensive online user guide is also available.

The GUIs work by invoking a secure RESTful API which is exposed by the Atmosphere host. We refer to this API as the **Cloud Facade**.



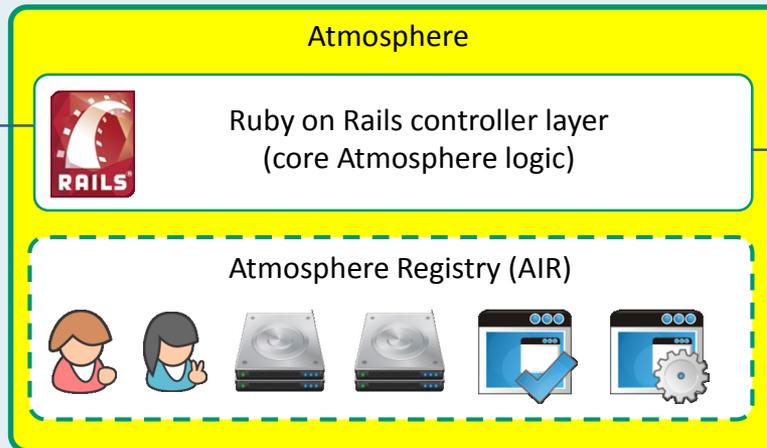
Application

-- or --



Workflow environment

Any operation which can be performed using the GUI may also be invoked programmatically by tools acting on behalf of the platform user – this includes standalone applications and workflow management environments.



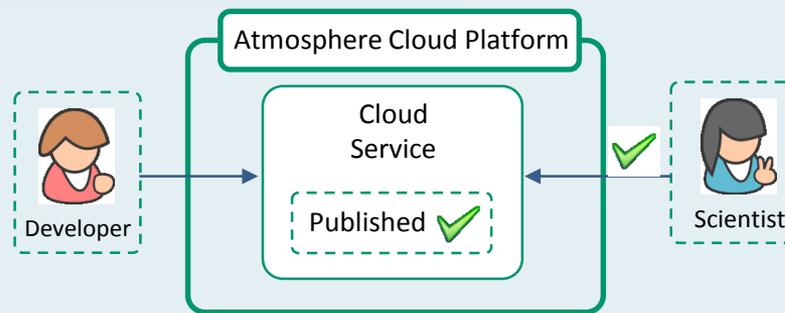
Cloud sites

All operations on cloud hardware are abstracted by the Atmosphere platform which exposes a RESTful API. For end users, a set of GUIs provides a user-friendly work environment. The API can also be directly invoked by external services (Atmosphere relies on the well-known OpenID authentication standard with PL-Grid acting as its identity provider).

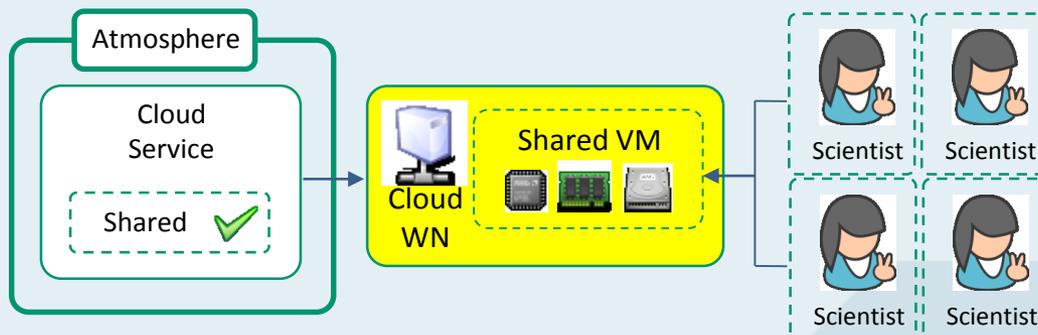


Advanced features: smart utilization of hardware resources

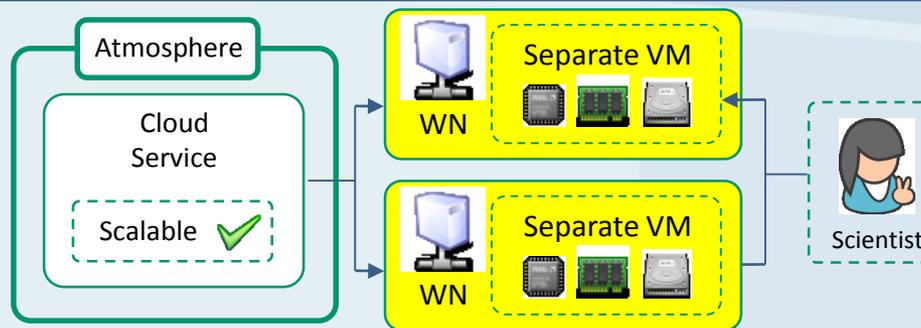
- **Published** services become visible to non-developers and can be instantiated using Atmosphere.
- Developers are free to spawn „snapshot” images of their cloud services (e.g. for backup purposes) without exposing them to external users.



- A **Shared** service is backed by a single virtual machine which „mimics” multiple instances from the users’ PoV.
- Shared services greatly conserve hardware resources and can be instantiated quickly.



- When a **Scalable** service is overloaded with requests Atmosphere can spawn additional instances in the cloud to handle the additional load.
- The process is transparent from the user’s perspective.

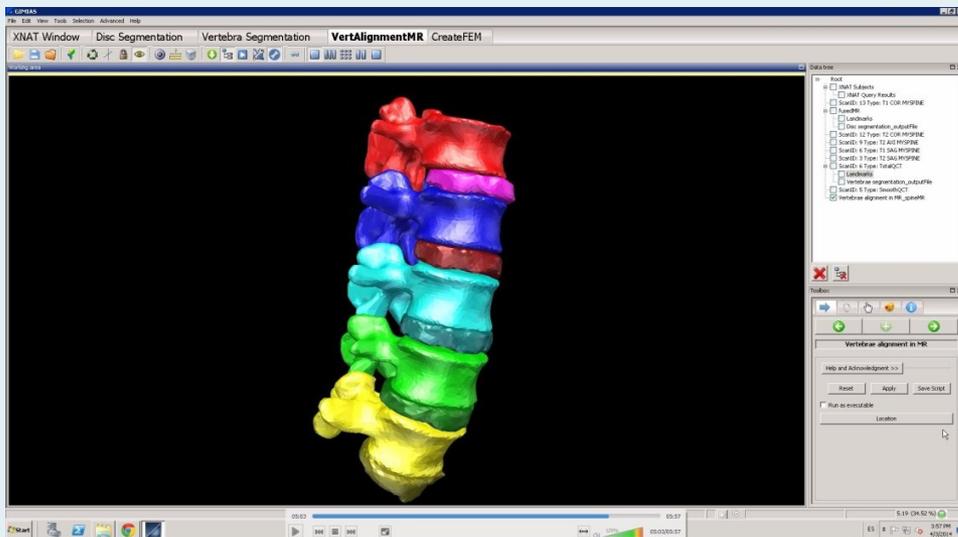


Some sample applications



A ticketing system is in place and technical support is available on a regular basis both to service developers and end users.

Online manuals and API documentation is available.



Not just a proof-of-concept deployment: a real production infrastructure with real-world applications and services. Over 150 service templates currently registered, with approximately 50 instances launched on a daily basis across three computational cloud sites.



Summary: challenges and solutions



- The Atmosphere framework provides a way to port scientific applications to the cloud
- A layer of abstraction is created over cloud-based virtual machines, enabling the platform to automatically select the best hardware resources upon which to deploy application services
- Automatic load balancing allows applications to scale up as needed
- PL-Grid Core also provides access to cloud hardware upon which scientific applications can be deployed
- A range of applications – from Linux-based SOAP/REST services all the way to rich graphical clients running under MS Windows have been successfully ported, proving the usefulness and versatility of our solution
- The platform is fully integrated with the wider PL-Grid ecosystem, including its authentication/authorization, sharing and data management mechanisms



For further information...



- A more detailed introduction to the Atmosphere cloud platform (including user manuals) can be found at <https://docs.cyfronet.pl/x/24D0>
- The PL-Grid team responsible for development and maintenance of the cloud platform is **plgg-cloud**
- You're also welcome to visit the Distributed Computing Environments (DICE) team homepage at <http://dice.cyfronet.pl> and our **brand new** GitHub site at <http://dice-cyfronet.github.io>

