Brief Overview of the INDIGO-DataCloud project

Alfonso Pérez, German Moltó, Giacinto Donvito

Universitat Politècnica de Valènica, INFN-Bari

INDIGO-DataCloud is co-founded by the Horizon 2020Framework Programme

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.

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 realised 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 Addresses Cloud Gaps

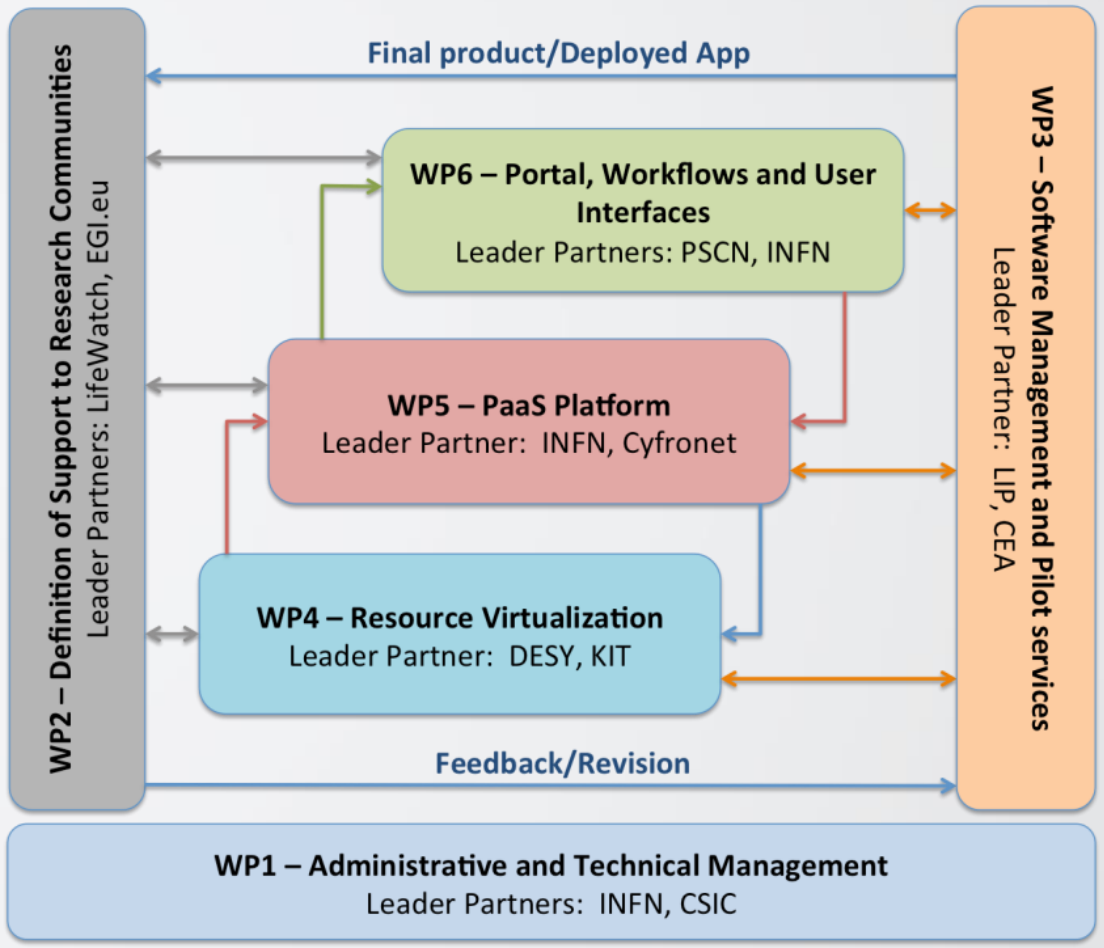
- **INDIGO focuses on use cases presented by its scientific communities** to address the gaps identified by the previously mentioned EC Report, with regard to:
  - Redundancy / reliability
  - Scalability (elasticity)
  - Resource utilization
  - Multi-tenancy issues
  - Lock-in
  - Moving to the Cloud
  - Data challenges: streaming, multimedia, big data
  - Performance
- **Reuses existing open source components** wherever possible and **contributing to upstream projects** (such as OpenStack, OpenNebula, Galaxy, etc.) for sustainability.

INDIGO and other European Projects

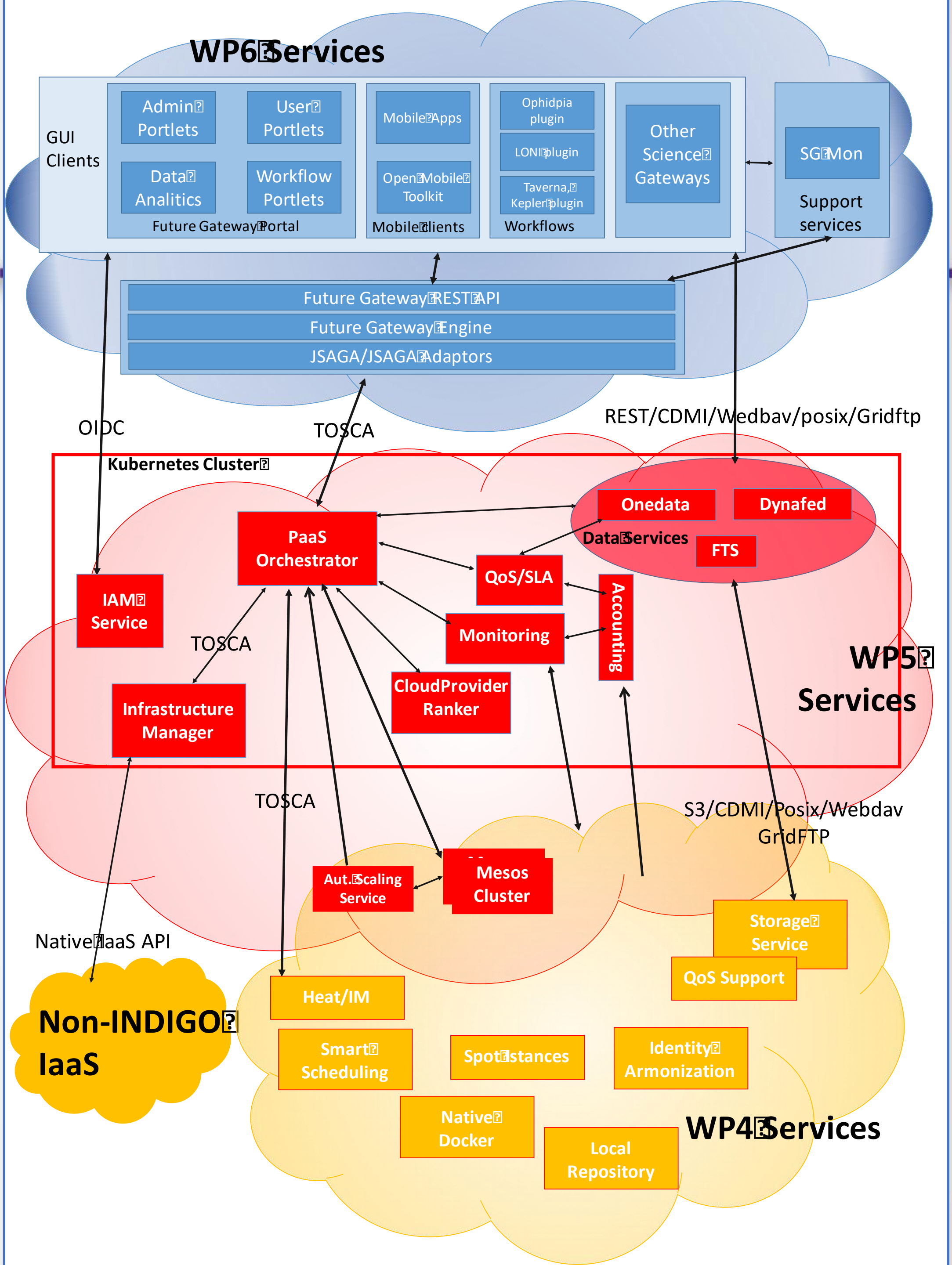
- The INDIGO services are being developed according to the requirements collected within many multidisciplinary scientific communities:
  - **ELIXIR, WeNMR, INSTRUCT, EGI-FedCloud, DARIAH, INAF-LBT, CMCC-ENES, INAF-CTA, LifeWatch-Algae-Bloom, EMSO-MOIST, EuroBioImaging**.
  - Requirements are implemented to be easily reused by other user communities.
- INDIGO has strong relationships with complementary initiatives:
  - **EGI-Engage** on the operational side
  - **AARC** with respect to AuthN/AuthZ policies.
  - **PRACE** and **EUDAT** are also expected to benefit from the deployment of INDIGO components in such infrastructures.
- Several **National/Regional infrastructures** are covered by the 26 INDIGO partners, located in 11 European countries.

Work Packages

INDIGO - DataCloud


Final product/Deployed App
WP2 – Definition of Support to Research Communities
Leader Partners: LifeWatch, EGI.eu
WP6 – Portal, Workflows and User Interfaces
Leader Partners: PSCN, INFN
WP5 – PaaS Platform
Leader Partner: INFN, Cyfronet
WP4 – Resource Virtualization
Leader Partner: DESY, KIT
WP3 – Software Management and Pilot services
Leader Partner: LIP, CEA
Feedback/Revision
WP1 – Administrative and Technical Management
Leader Partners: INFN, CSIC

INDIGO-DataCloud General Architecture



INDIGO - DataCloud

IaaS Features

- **Improved scheduling for allocation of resources** for OpenStack and OpenNebula:
  - Better scheduling algorithms.
  - Support for spot-instances.
  - Support dynamic partitioning of resources among “traditional batch systems” and Cloud infrastructures (for some LRMS).
- **Support for standards in IaaS resource orchestration engines:**
  - Use of the TOSCA standard.
- **Improved IaaS orchestration capabilities** for OpenStack and OpenNebula:
  - Custom TOSCA templates facilitate resource orchestration for end users
  - Increase scalability of deployed resources
  - Support of orchestration capabilities for OpenNebula.

IaaS Features

- **Improved QoS capabilities of storage resources**:
  - Better support of high-level storage requirements such as flexible allocation of disk or tape storage space and support for data life cycle.
- **Improved capabilities for networking support**:
  - Flexible networking support in OpenNebula.
  - Handling of network configurations through developments of the OCCI standard for both OpenNebula and OpenStack.
- **Improved and transparent support for Docker containers**:
  - Native container support in OpenNebula (ONEDock).
  - Development of standard interfaces using the OCCI protocol to drive container support in both OpenNebula and OpenStack.

PaaS Features

- **Improved capabilities in the geographical exploitation of Cloud resources**:
  - The INDIGO PaaS layer will hide the complexity of both scheduling and brokering.
- **Standard interface to access PaaS services**:
  - Use of the TOSCA standard to hide these differences.
- **Support for data requirements in Cloud resource allocations:**
  - Resources could be allocated where data is stored.
- **Integrated use of resources coming from both public and private Cloud infrastructures**

PaaS Features

- **Distributed data federations:**
  - Supporting legacy applications as well as high level capabilities for distributed QoS and Data Lifecycle Management (e.g. remote Posix access to data).
- **Integrated IaaS and PaaS support in resource allocations**:
  - E.g. storage provided at the IaaS layer is automatically made available to higher-level allocation resources performed at the PaaS layer.
- **Transparent client-side import/export of distributed Cloud data**.
  - Dropbox-like mechanisms for importing and exporting data from/to the Cloud (OneData).
- **Support for distributed data caching mechanisms and integration with existing storage infrastructures**.

PaaS Features

- **Deployment, monitoring and automatic scalability of existing applications**.
  - E.g., existing applications such as web front-ends or R-Studio servers can be automatically and dynamically deployed in highly-available and scalable configurations.
- **Integrated support for high-performance Big Data analytics**.
  - This includes custom frameworks such as Ophidia as well as general purpose engines for large-scale data processing such as Spark.
- **Support for dynamic and elastic clusters of resources**.
  - Resources and applications can be clustered through the INDIGO APIs:
    - Batch systems on-demand (such as HTCondor or Torque).
    - Extensible application platforms (such as Apache Mesos).

AAI Features

- Provide an advanced set of features that includes:
  - User authentication (supporting SAML, OIDC, X.509)
  - Identity harmonization (link heterogeneous AuthN mechanisms to a single VO identity)
  - Management of VO membership (i.e., groups and other attributes)
  - Management of registration and enrolment flows
  - Provisioning of VO structure and membership information to services
  - Management, distribution and enforcement of authorization policies

Use cases examples

INDIGO - DataCloud

INDIGO - DataCloud

UC: A web portal that exploits a batch system to run applications

- A user community maintains a “vanilla” version of portal and computing image plus some specific recipes to customize software tools and data
  - Portal and computing are part of the same image that can take different roles.
  - Customization may include:
    - Creating special users
    - Copying (and registering in the portal) reference data
    - Installing (and again registering) processing tools.
  - Typically web portal image also has a batch queue server installed.
- All the running instances share a common directory.
- Different credentials: end-user and application deployment.

UC Inspiration: Galaxy on the cloud

- Galaxy can be installed on a dedicated machine or as a front/end to a batch queue.
- Galaxy exposes a web interface and executes all the interactions (including data uploading) as jobs in a batch queue.
- Requires a shared directory among the working nodes and the front/end.
- It supports a separate storage area for different users, managing them through the portal.

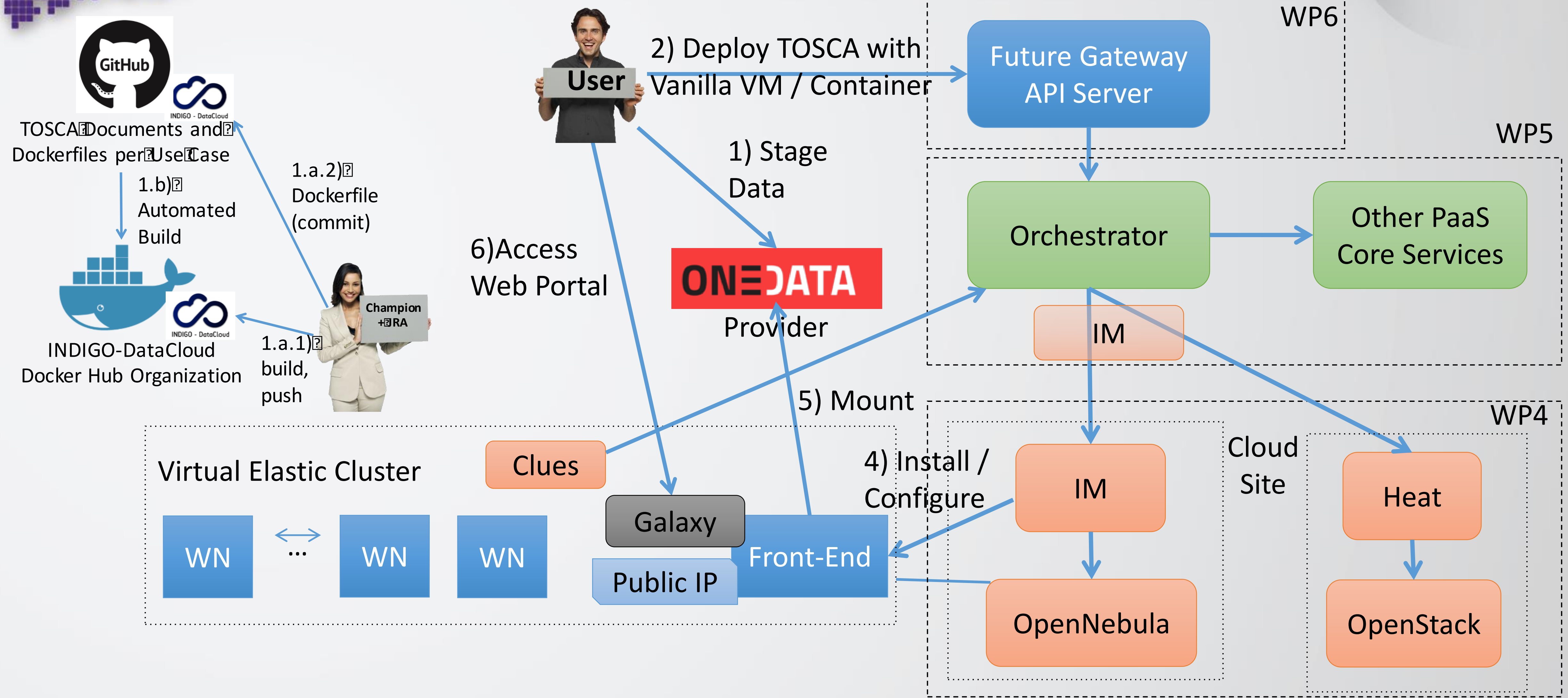
UC: A web portal that exploits a batch system to run applications

1) The web portal is instantiated, installed and configured automatically exploiting Ansible recipes and TOSCA Templates.
2) A remote posix share is automatically mounted on the web portal using Onedata
3) The same posix share is automatically mounted also on worker nodes using Onedata
4) End-users can see and access the same files via simple web browsers or similar.
5) A batch system is dynamically and automatically configured via TOSCA Templates
6) The portal is automatically configured in order to execute job on the batch cluster
7) The batch cluster is automatically scaled up & down looking at the job load on the batch system.

UC: Use Case Lifecycle

- Preliminary
  - The use case administrator creates the “vanilla” images of the portal+computing image.
  - The use case administrator, with the support of INDIGO experts, writes the TOSCA specification of the portal, queue, computing configuration.
- Group-specific
  - The use case administrator, with the support of INDIGO experts, writes specific modules for portal-specific configurations.
  - The use case administrator deploys the virtual appliance.
- Daily work
  - Users Access the portal as if it was locally deployed and submit Jobs to the system as they would have been provisioned statically.

UC: A Graphic Overview



Thank you

INDIGO - DataCloud

https://www.indigo-datacloud.eu

Better Software for Better Science.

INDIGO FAQ

- How do INDIGO achieve resource redundancy and high availability?
  - This is achieved at multiple levels:
    - **Data level**, redundancy can be implemented exploiting the capability of INDIGO's Onedata of replicating data across different data centers.
    - **Site level**, it is possible to ask for copies of data to be for example on both disk and tape using the INDIGO QoS storage features..
    - **Services**, the INDIGO architecture uses Mesos and Marathon to provide automatic service high-availability and load balancing. This automation is easily obtainable for stateless services; for stateful services this is application-dependent but it can normally be integrated into Mesos through, for example, a custom framework (examples of which are provided by INDIGO).
- How do INDIGO achieve resource scalability?
  - First of all, we can distinguish between vertical (scale up) and horizontal (scale out) scalability. INDIGO provides both:
    - Mesos and Marathon handle vertical scalability by deploying Docker containers with an increasing amount of resources.
    - The INDIGO PaaS Orchestrator handles horizontal scalability through requests made at the IaaS level to add resources when needed.

INDIGO FAQ

- How do INDIGO achieve resource scalability?
  - The INDIGO software does this in a smart way, i.e. for example it does not look at CPU load only:
    - In the case of a dynamically instantiated LRMS, it checks the status of jobs and queues and accordingly adds or remove computing nodes.
    - In the case of a Mesos cluster, in case there are applications to start and there no free resources, INDIGO starts up more nodes. This happens within the limits of the submitted TOSCA templates. In other words, any given user stays within the limits of the TOSCA template he has submitted; this is true also for what regards accounting purposes.
- How do you know when and where resources are available?
  - We are extending the Information System available in the European Grid Infrastructure (EGI) to inform the INDIGO PaaS orchestrator about the available IaaS infrastructures and about the services they provide. It is therefore possible for the INDIGO orchestrator to optimally choose a certain IaaS infrastructure given, for example, the location of a certain dataset.