EGI Federated Cloud
Check-in/AppDB

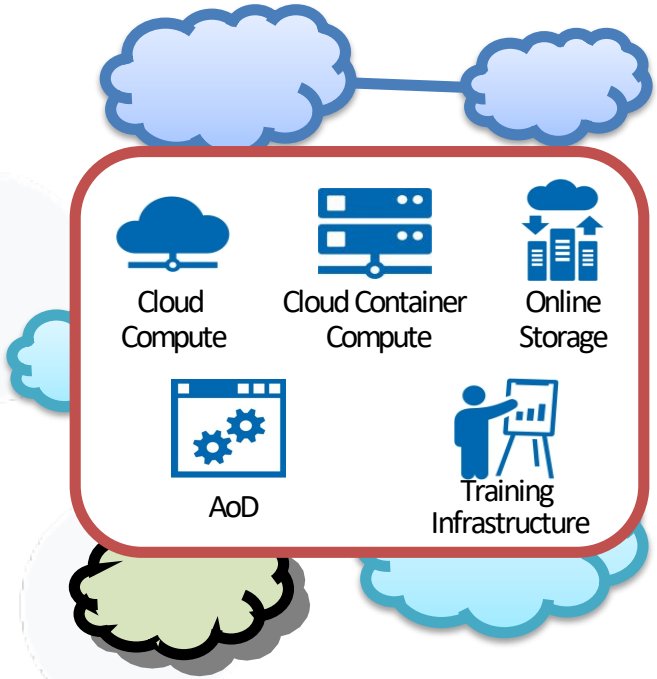
Baptiste Grenier / Enol Fernández

with material from Nicolas Liampotis and Marios Chatziangelou

EGI-Engage is co-funded by the Horizon 2020 Framework Programme
of the European Union under grant number 654142

EGI Cloud Federation

- Multi-cloud IaaS with Single Sign-On via Check-In
- Technology agnostic:
  - OpenStack,
  - OpenNebula
  - Synnefo



The infrastructure

20 resource centres
- 15 OpenStack
- 4 OpenNebula
- 1 Synnefo

5 centres under integration

2 centres expressed interest on joining

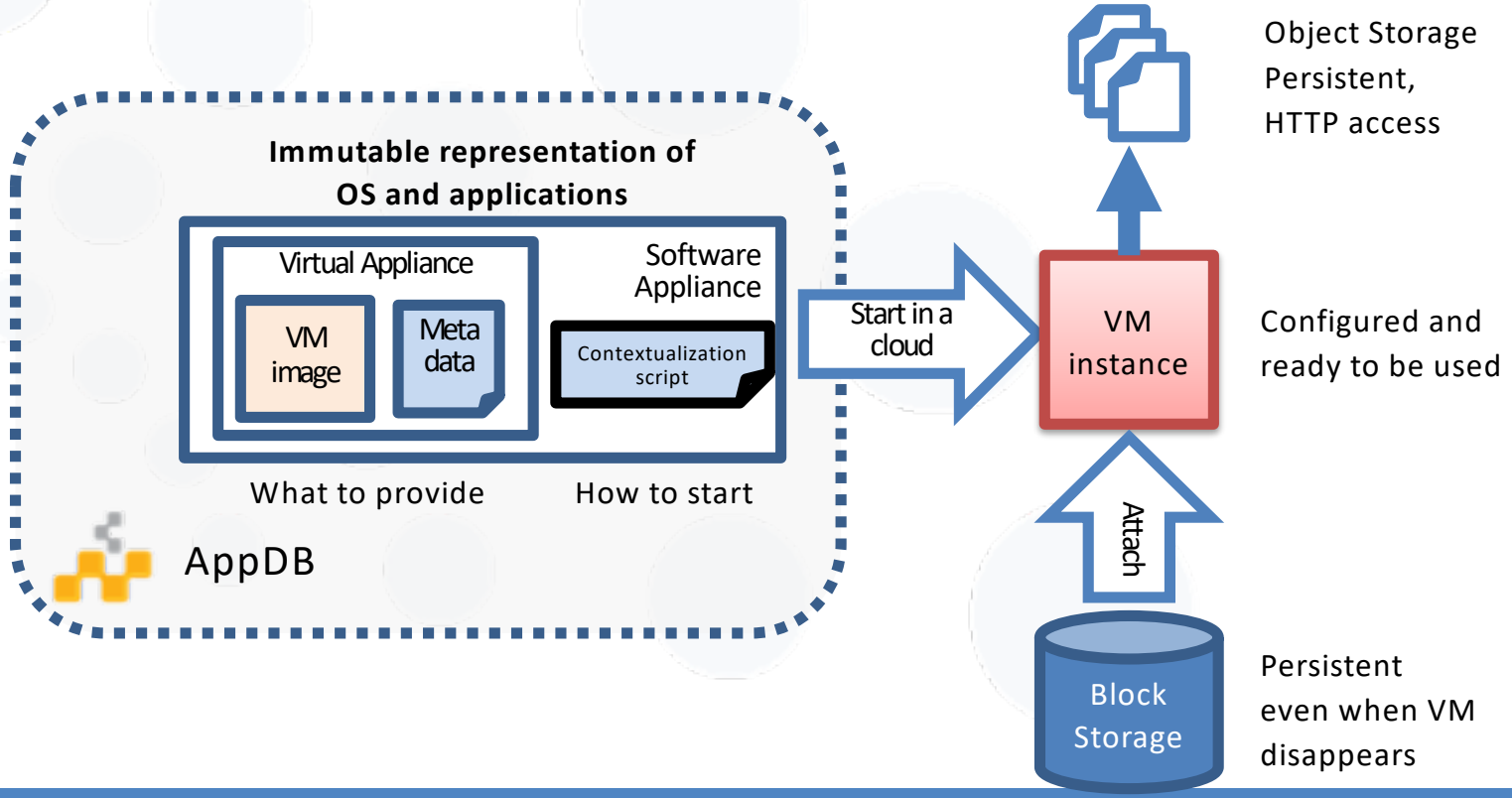
EGI Compute services

	EGI Cloud Compute	EGI Cloud Container Compute	EGI High Throughput Compute
What is it?	Multi-cloud IaaS	Kubernetes on top of EGI Cloud Compute	The *grid,* a scalable batch system
What you run?	VMs	(Docker) Containers	Jobs
Typical workloads	Lift and shift existing applications Specific OS (kernel) requirements	Cloud-native containerised applications.	Execution of parallel computing tasks to analyse large datasets.
Pros / Cons	[+] Complete control on resources, run (almost) anything you'd like [-] Complex operation	[+] Industry standard [+] Hides complexity of Kubernetes setup [-] Some Kubernetes features not available	[+] No management of resources, just submit jobs [-] Legacy interfaces [-] Porting of applications

EGI Cloud Compute

- Run Virtual Machines on demand on EGI's Cloud Federation
  - Similar to AWS EC2/EBS or GCP Compute Engine
- Diverse providers with common:
  - AuthN and AuthZ
  - VM Image catalogue
  - Information discovery
  - Accounting
  - Monitoring
  - GUI dashboard

EGI Cloud Compute – the basics



A multi-cloud IaaS for research

Users

- Single-Sign On via Check-in
- Appliance Library
- Resource discovery
- Single GUI dashboard
- Computation near data

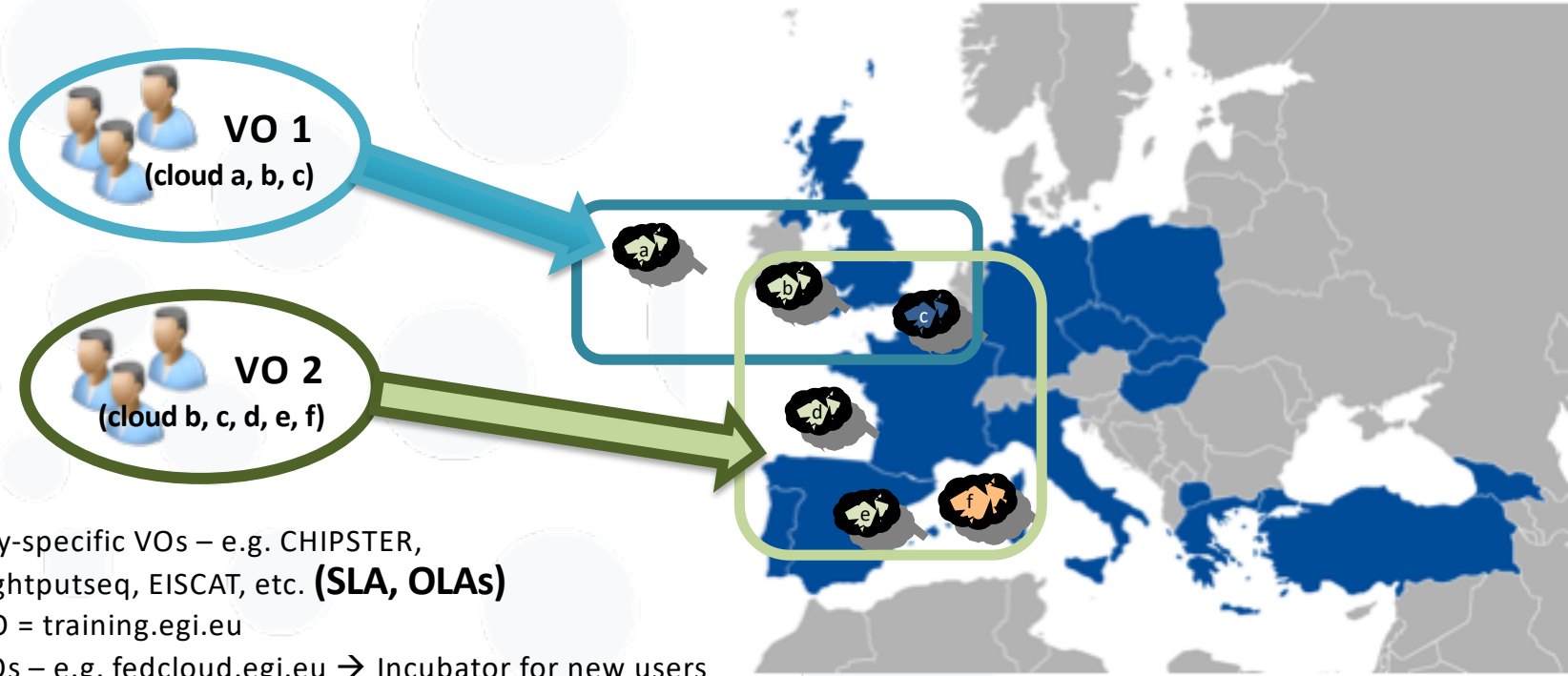
Providers

- Support international communities
- Operational and security processes
- A/R Monitoring
- Accounting
- Technology agnostic

Integration with resource providers

- **Providers keep complete control on their resources**
  - Enable access to a Virtual Organisation into local projects with EGI credentials, SLA based
  - Set of extra components using OpenStack APIs to retrieve provider information and make it available to EGI services
- **No (major) changes in operational activities**
  - Provide support through EGI channels
  - Comply with the operational and security procedures. These are based on best practices and common requirements
  - Most well-operated providers already have these activities in place, so the additional task for a site manager is to acknowledge to EGI that the task has been performed.

VO based access



1. Community-specific VOs – e.g. CHIPSTER, Highthroughtputseq, EISCAT, etc. **(SLA, OLAs)**
2. Training VO = training.egi.eu
3. Generic VOs – e.g. fedcloud.egi.eu → Incubator for new users

Browse VOs at http://operations-portal.egi.eu/vo/search (both HTC and cloud)

Check-in in a nutshell

Identity and Access Management solution that makes it easy to secure access to services and resources

Components

- IdP/SP Proxy
- User enrolment & group management
- IdP Discovery
- Token Translation

Documentation

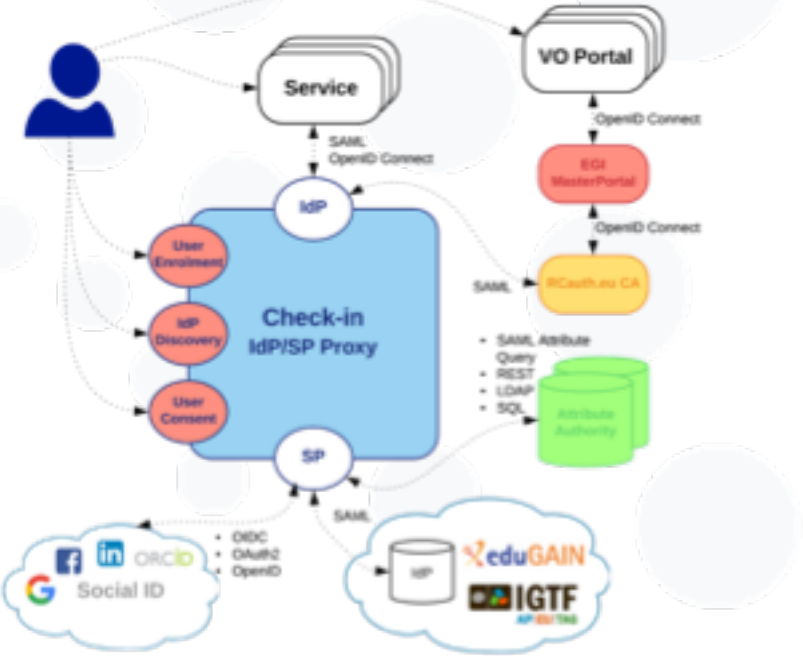
- Usage guide
- Integration guides

https://wiki.egi.eu/wiki/AAI

What benefits does Check-in bring?

- Single sign-on to services through eduGAIN, social media and other institutional or community-managed identity providers
- Only one account needed for federated access to multiple heterogeneous (web and non-web) service providers using different technologies (SAML, OpenID Connect, OAuth 2.0, X509)
- Identity linking enables access to resources using different login credentials (institutional/social)
- Assurance information associated to each authenticated identity
- Aggregation and harmonisation of authorisation information (VOs/groups, roles, assurance) from multiple sources

IdP/SP proxy

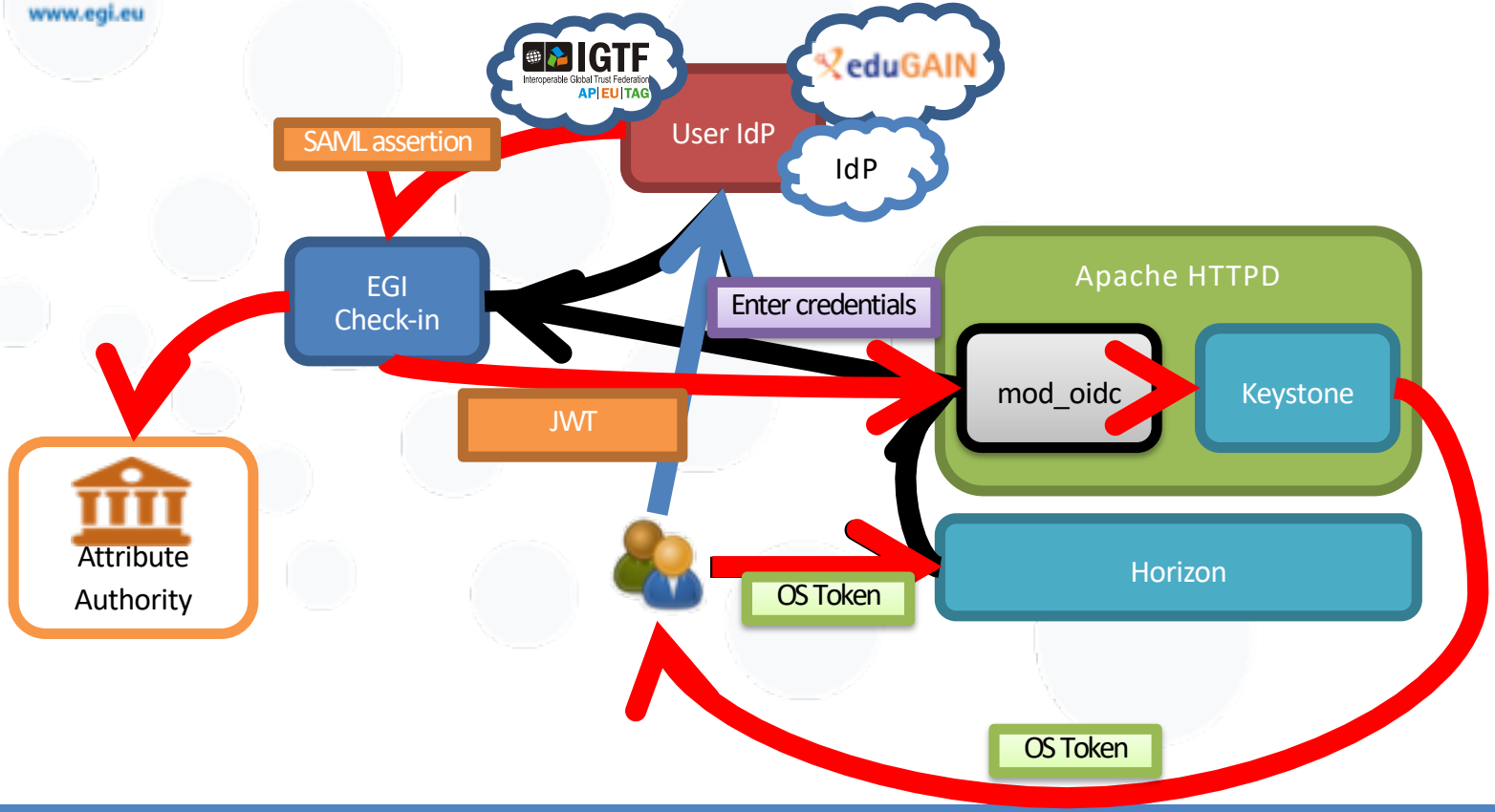


- Implementation of the AARC blueprint architecture
- Registered in eduGAIN as an SP complying with REFEDS Research & Scholarship and Sirtfi
- All community SPs can have one statically configured IdP
- No need to run an IdP Discovery Service on each community SP
- Connected SPs get consistent/harmonised user identifiers and accompanying attribute sets from different IdPs/AAs that can be interpreted in a uniform way for authorisation purposes

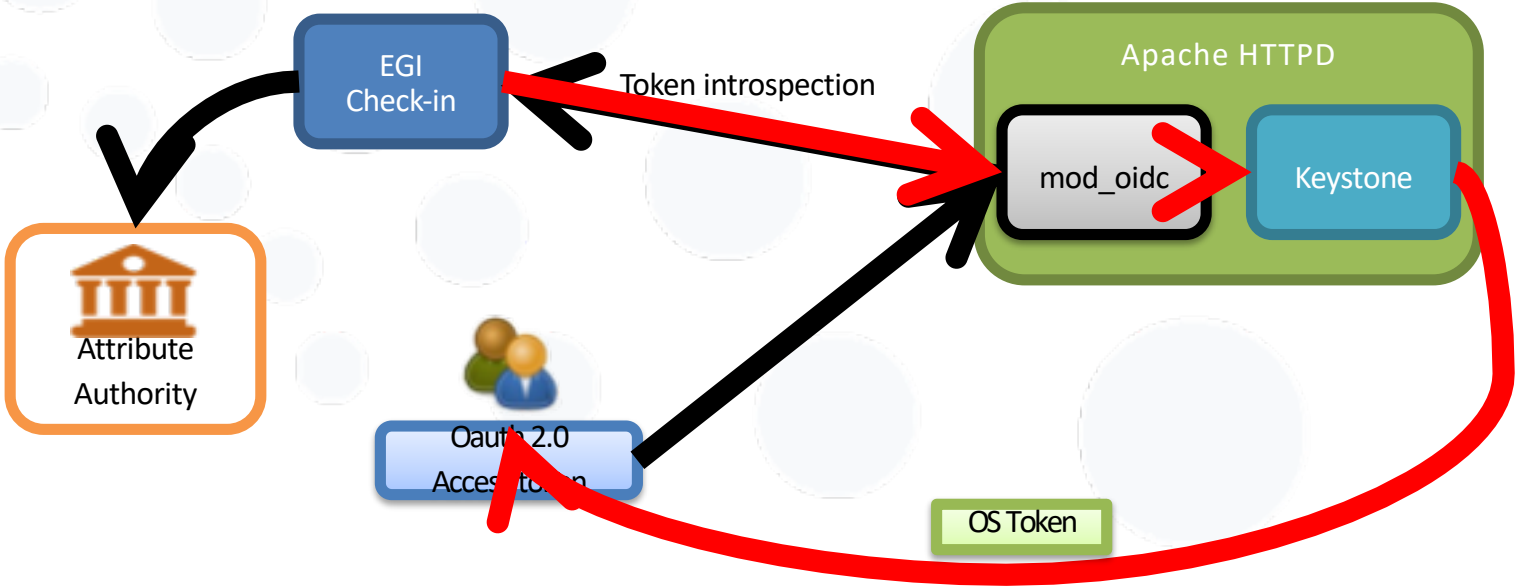
Integration of OpenStack with Check-in

- OpenID Connect (OIDC)
  - Industry standard
  - Web and non-web friendly (OAuth2.0)
  - Supported by OpenStack since Icehouse
- OpenStack Providers need to:
  1. Configure OpenStack support for OIDC (supported upstream)
  2. Add EGI Check-in as IdP for OpenStack
  3. Configure mapping of Check-in users to local projects as agreed with VOs

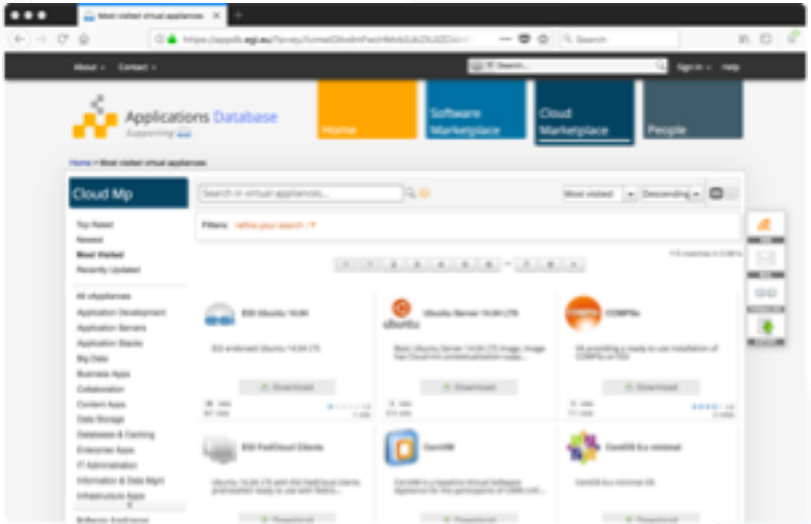
EGI Check-In + Horizon



EGI Check-In + CLI



AppDB – Cloud Marketplace (1)



Registry for virtual appliances (VA)

✓ a logical container of versioned image file & metadata bundles

Registry for software appliances (SA)

✓ a logical container of VA versions & contextualization scripts bundles

VA distribution medium

✓ distributing endorsed VAs to the resource providers/sites

AppDB – Cloud Marketplace (2)

Resource providers catalogue

- ✓ list of the VAs which are available by each site/resource provider

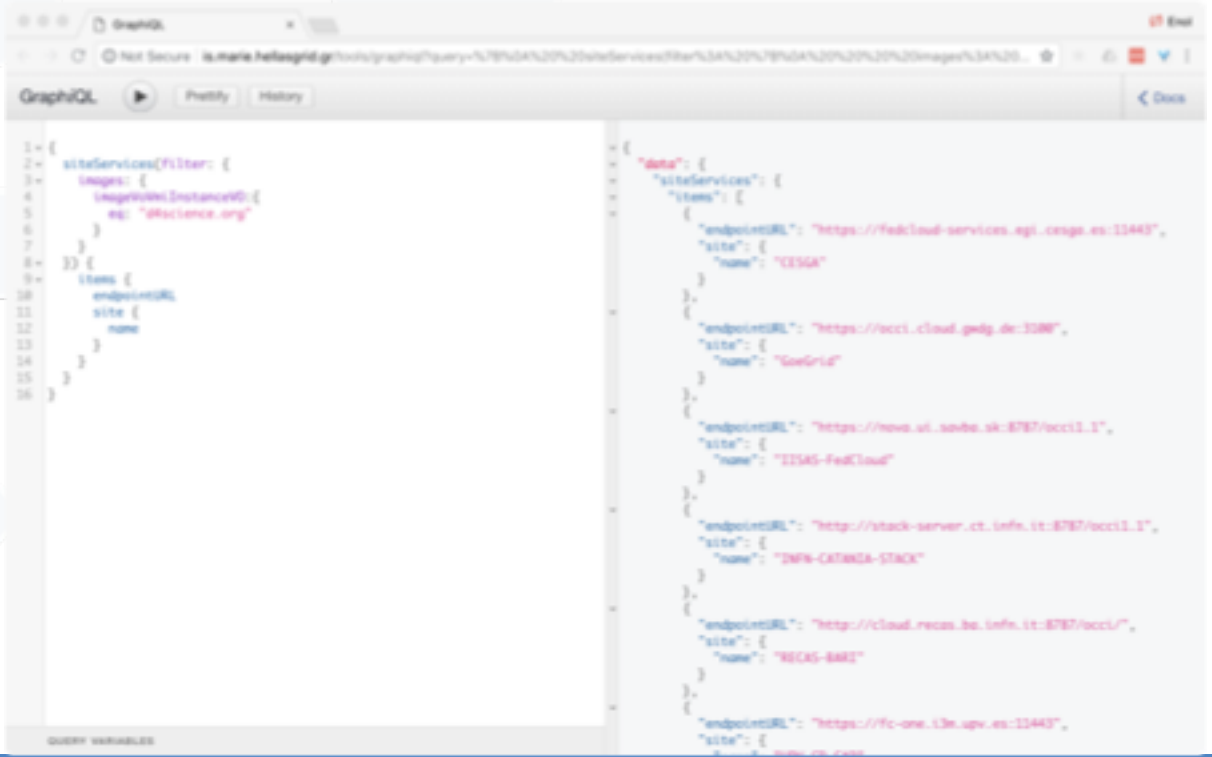
Virtual Organizations (VO) catalogue

- ✓ list of the VAs which are available for each VO member

AppDB – a browsable information system

AppDB – a GraphQL powered information system

More information at https://docs.google.com/presentation/d/19Yh3kNxl01DfcrDgQf12w-KQW5Zrd_OnYP2iGp9Kg2Y/edit?ts=5a2ab515#slide=id.p



CloudKeeper: AppDB integration

cloudkeeper

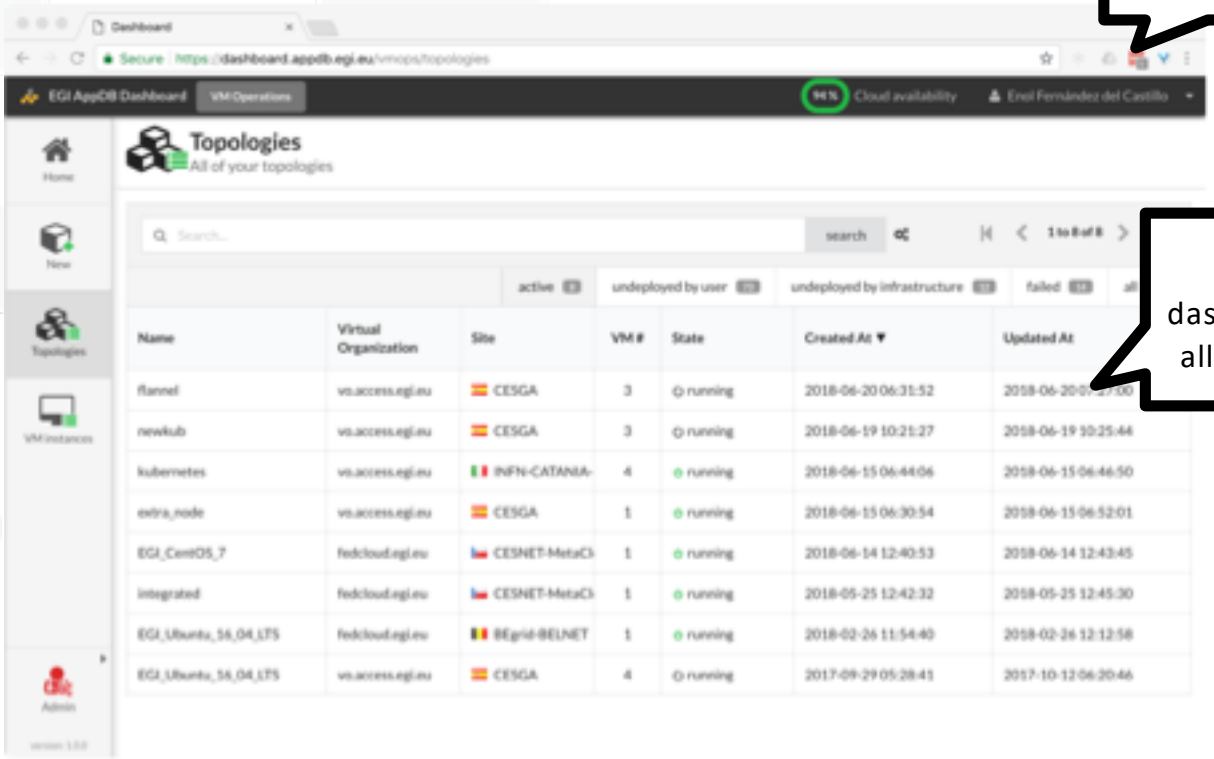
cloudkeeper ensures:

- Integrity of images
- Conversion to appropriate formats
- Correct metadata info at glance

Manage VMs via AppD Ops

Complete Check-in integration

Wizard-like creation of VMs



Single dashboard for all providers

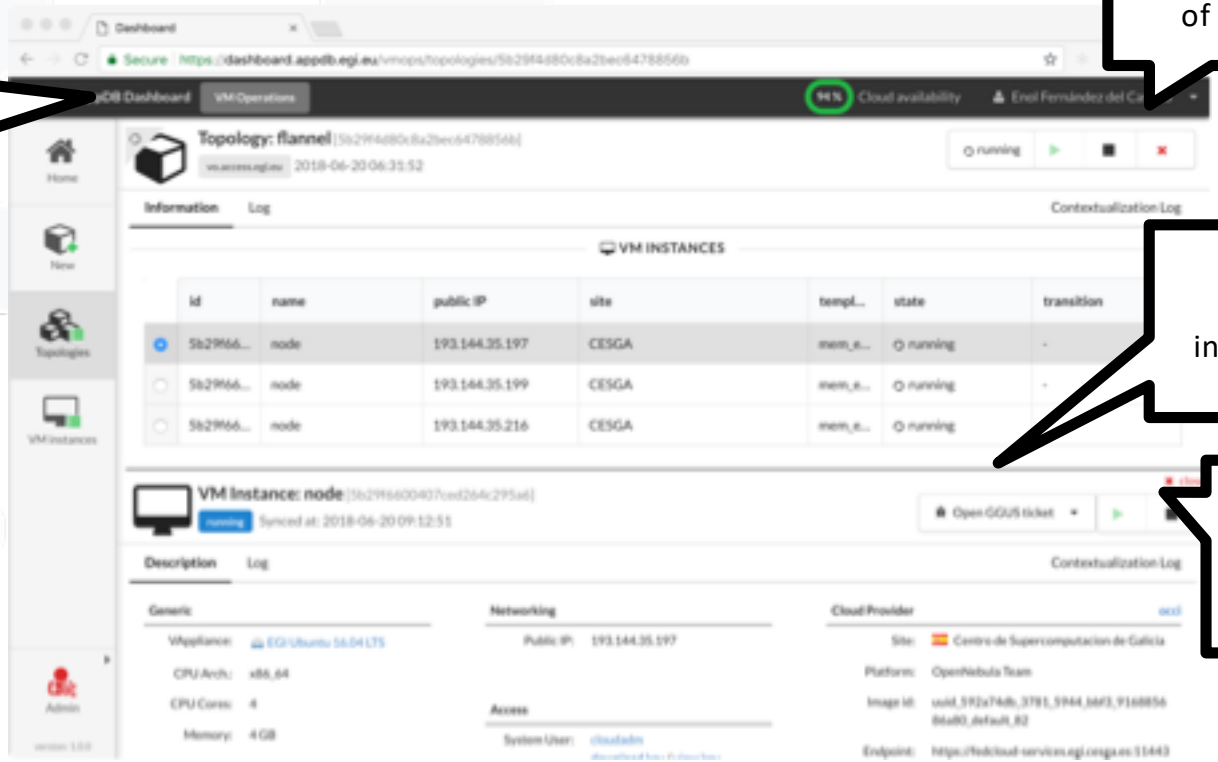
Manage VMs via AppDB VMOps

Topologies are a set of related VMs

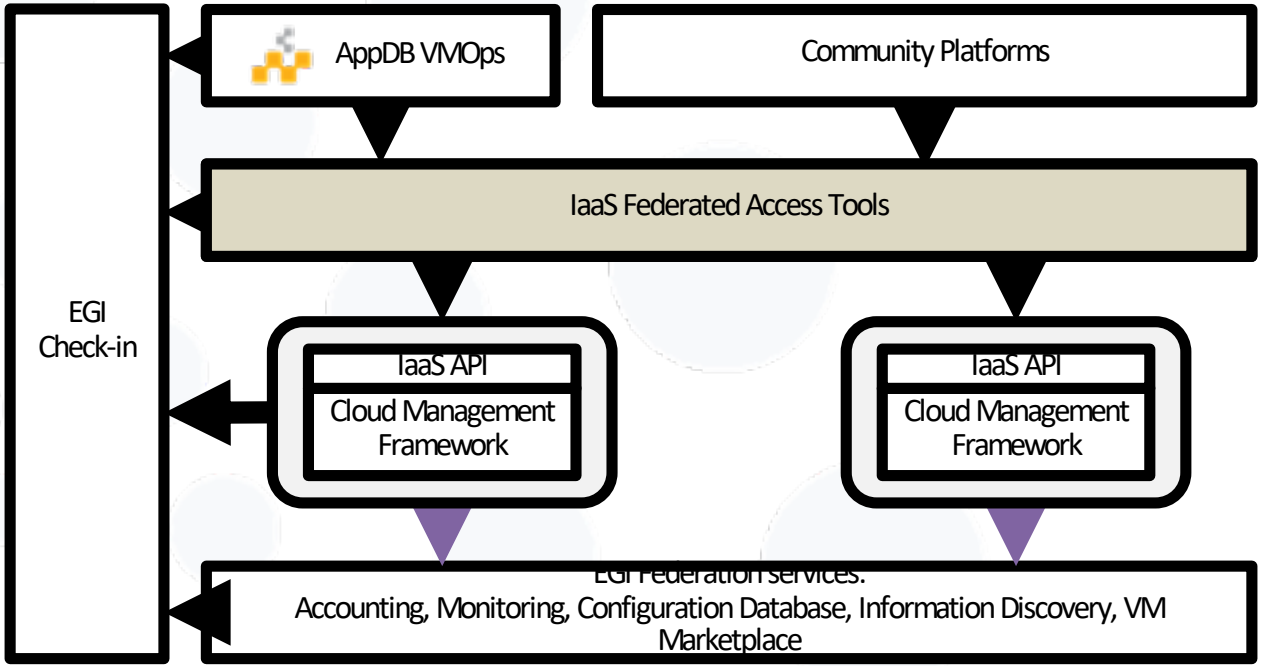
Global management of VMs

GGUS integration

Individual management of VMs



Architecture

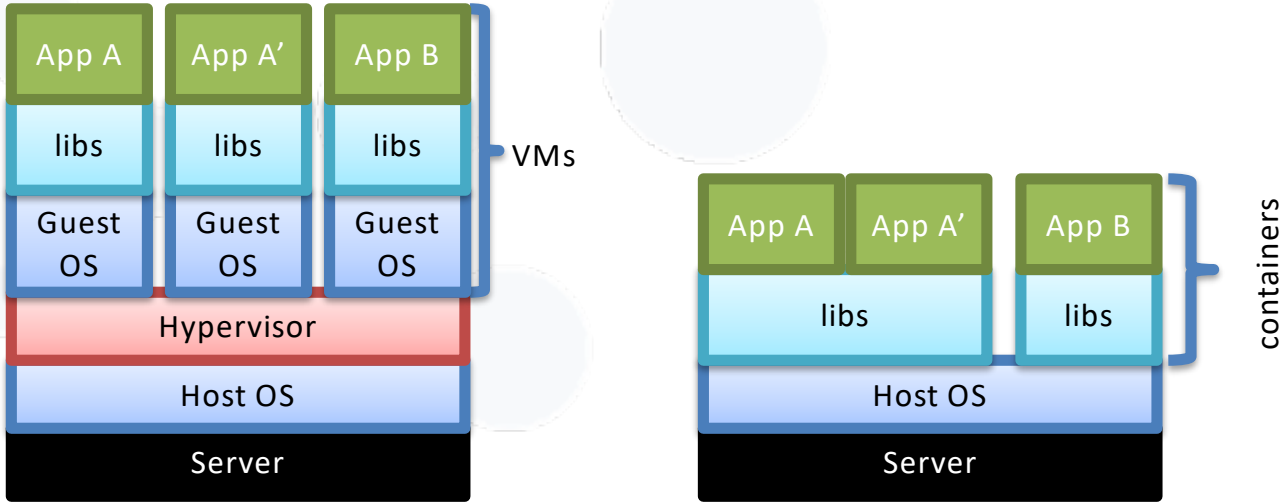


API access: dealing with heterogeneity

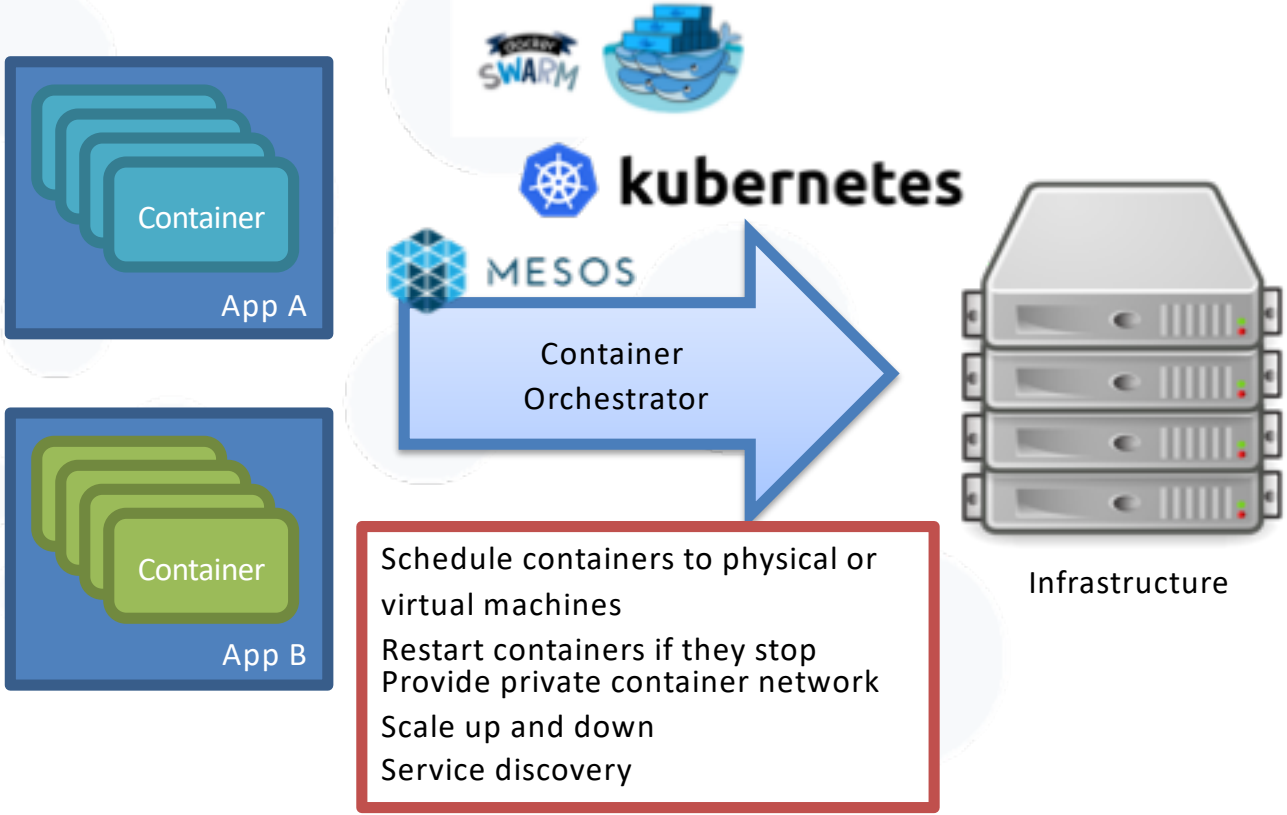
- EGI Federated Cloud no longer mandates a single API for every provider
  - OCCI still widely supported but sites are moving native APIs (mainly OpenStack!)
- Tools to deal with heterogeneity:
  - IaaS orchestration tools with support for multiple APIs:
    - Infrastructure Manager, Terraform, OCCOPUS, …
    - https://wiki.egi.eu/wiki/Federated_Cloud_IaaS_Orchestration
  - IaaS libraries with support for multiple APIs:
    - libcloud, jclouds,…
  - See guide on migrating from OCCI to IM on EGI's wiki: https://wiki.egi.eu/wiki/Federated_Cloud_OCCI_to_IM_Migration

Containers

- Containers provide virtualisation at the OS level
  - Same kernel, isolated user-space
  - Faster deployment, less overhead, easier migration…



Container orchestration



EGI Cloud Container Compute

- Run containers on top of EGI Cloud Compute VMs
- 2 (+ 1) options:
  - Single node: start the EGI Docker VM and run containers directly (or with docker compose)
  - Kubernetes: start a cluster of VMs and create a Kubernetes cluster to run your containers
    - Start the cluster using IM + Ansible
    - Working on: auto-scaling with EC3, Check-in integration at Kubernetes level
  - udocker: run containers as jobs in the EGI HTC service
- https://wiki.egi.eu/wiki/Federated_Cloud_Containers

New models for the EGI Cloud

Provide feedback on the proposal at http://go.egi.eu/egi-cloud-expansion

- Lighter federation to attract providers and users
  - Introduce new types of services and resources
  - Marketplace oriented
- 3 different models under discussion
  - Application Services
  - IaaS Alliance
  - Applications Platforms
- Objectives
  - Facilitate innovation
  - Make it easier for providers to enter the EOSC landscape
  - Offer a broader set of services for users

Under development: EGI Notebooks

- Offer Jupyter notebooks ‘as Service’
  - One-click solution, just login and start using
- EGI Features:
  - Login with Check-in
  - Persistent storage
  - Bring your own environments/kernels
  - Use EGI computing and storage resources from your notebooks

Some references

- Federated Cloud at EGI wiki: https://wiki.egi.eu/wiki/EGI_Federated_Cloud
- Installation manual: http://egi-federated-cloud-integration.readthedocs.io/
- EGI Federated Cloud list: fedcloud-tf@mailman.egi.eu

Thank you for your attention.

Questions?

Token translation – Integration with RCauth.eu Online CA

- Check-in has been integrated with the production RCAuth.eu Online CA
  - Users can retrieve X.509 proxies by authenticating through Check-in

- Check-in Master Portal retrieves end-entity certificate from RCauth.eu
- Long-lived proxy certificate stored in backend MyProxy server
- Short-lived proxies provided via:
  - Science Gateways via OIDC (so-called VO-portals)
  - users e.g. via SSH key authentication

User enrolment & group management

- Ability to create enrolment flows specific to a community's requirements
- Support for oganising users in hierarchical groups
- Ability to associate certificate and ssh key information to researcher's federated identity
- Ability to enrich researcher’s identity with community-specific attributes
- Direct (de)provisioning of information into an LDAP directory or VOMS

IdP discovery

Group membership and role information

Use of URN-formatted entitlement values based on AARC guidelines:

urn:mace:egi.eu:group:<group>[:<subgroup>*][:role=<role>]#<group-authority>

- **<group>** is the name of a VO, research collaboration or a top level arbitrary group; unique within a given <namespace>
- optional list of **<subgroup>** components represents the hierarchy of subgroups in the **<group>**
- optional **<role>** component indicates particular position of the user; scoped to the rightmost (sub)group
- **<group-authority>** indicates the authoritative source for the group membership and role information

Check-in Community AAI service options

Multi-tenant service

- All the standard Check-in authentication options
- Community management using COmanage or Perun
- Basic customisation of user-facing interfaces (e.g. community-specific themes for enrolment flows, group management)
- Basic customisation of AAI proxy behavior

Dedicated service (individual components or AAI service as a whole)

- Customisation of user-facing interfaces: WAYF, enrolment, group membership UI
- Customisation of AAI proxy behaviour (e.g. attribute aggregation rules, service entitlements/capabilities)
- Integration with the EOSC-hub AAI e-Infrastructure SP Proxies for accessing EOSC services and resources

Kubernetes

- Kubernetes is an *open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure.*
- Some concepts:
  - *Pod*: group of one or more containers, shared storage and options to run the containers
  - *Deployment* maintains the desired count of Pods all the time
  - *Service:* logical set of Pods and a policy by which to access them.
    - Exposed to the exterior of the Kubernetes cluster via mapping of ports and or Load Balancing
  - *Job*: A *job* creates one or more pods and ensures that a specified number of them successfully terminate.

Example

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend

spec:
  selector:
    matchLabels:

      app: guestbook
      tier: frontend
  replicas: 3
  template:

    metadata:
      labels:
        app: guestbook

        tier: frontend
    spec:
      containers:

      - name: php-redis
        image: gcr.io/google-samples/gb-frontend:v4
        resources:
          requests:

            cpu: 100m
            memory: 100Mi
        env:

        - name: GET_HOSTS_FROM
          value: dns
        ports:
        - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: frontend

  labels:
    app: guestbook
    tier: frontend

spec:
  # comment or delete the following line if you want to use a LoadBalancer
  type: NodePort
ports:

  - port: 80
  selector:
    app: guestbook

    tier: frontend
---
apiVersion: extensions/v1beta1

kind: Ingress
metadata:
  name: frontend
spec:
  rules:
  - host: frontend.test.fedcloud.eu
    http:

      paths:
      - backend:
          serviceName: frontend
          servicePort: 80
```

EGI Cloud Container Kubernetes

- Provides Kubernetes v1.10
- Major differences with other offerings:
  - LoadBalancer ServiceType:
    - A NGINX ingress configured by default ready to be used offering similar functionality
    - Expandable with auto-configuration of Let's Encrypt certificates
  - Dynamic provision of volumes for PersistentVolumeClaims
    - No block-storage directly available
    - NFS-based volumes available instead

A note on AAI

- EGI Cloud Compute currently relies on legacy X.509 + VOMS proxies for access to resources
  - For users without certificates:
    - PUSP with user-personalised proxies from robot certificate
    - RCAuth Online CA to obtain personal proxies from EGI Check-in identities
- Now rolling-out production providers with native OpenID Connect support
  - 2 sites now available, more coming
  - No need for certificates at all!