CAGE (Cloud for Astrophysics GatEway)
Beyond Science Gateway

Autore/i: A. Calanducci, A. Costa, E. Sciacca,
F. Vitello, U. Becciani

INAF
ISTITUTO NAZIONALE
DI ASTROFISICA
NATIONAL INSTITUTE
FOR ASTROPHYSICS

ASTRI

AC DC
Astri/CTA Data Challenge

Science Gateway Limitations

- Science Gateways provide simplified access to cloud computing resources for scientific applications (workflows, pipelines, etc)
- They are generally implemented with a Web portal interface
  - not suitable for interactive and high demanding User Interfaces (complex visualization systems, AR, VR)
- Access to SG require additional credentials (certificates, proxies) that is authorized to the target cloud infrastructures (EGI FedCloud)

Beyond Science Gateway

- Enabling client other than Web to access Clouds:
  - Interactive apps generally implemented as Desktop apps
    - capable of access native features, such as the GPU for 3d graphics, fluid maps navigation, etc
  - legacy code run as a script from a command line prompt
  - the new generation of smartphone/tablet can enable scientists to new form of scientific simulation
    - exploiting hardware sensors (GPS, AR, VR, local machine learning, barometer, accelerometers, and so on)

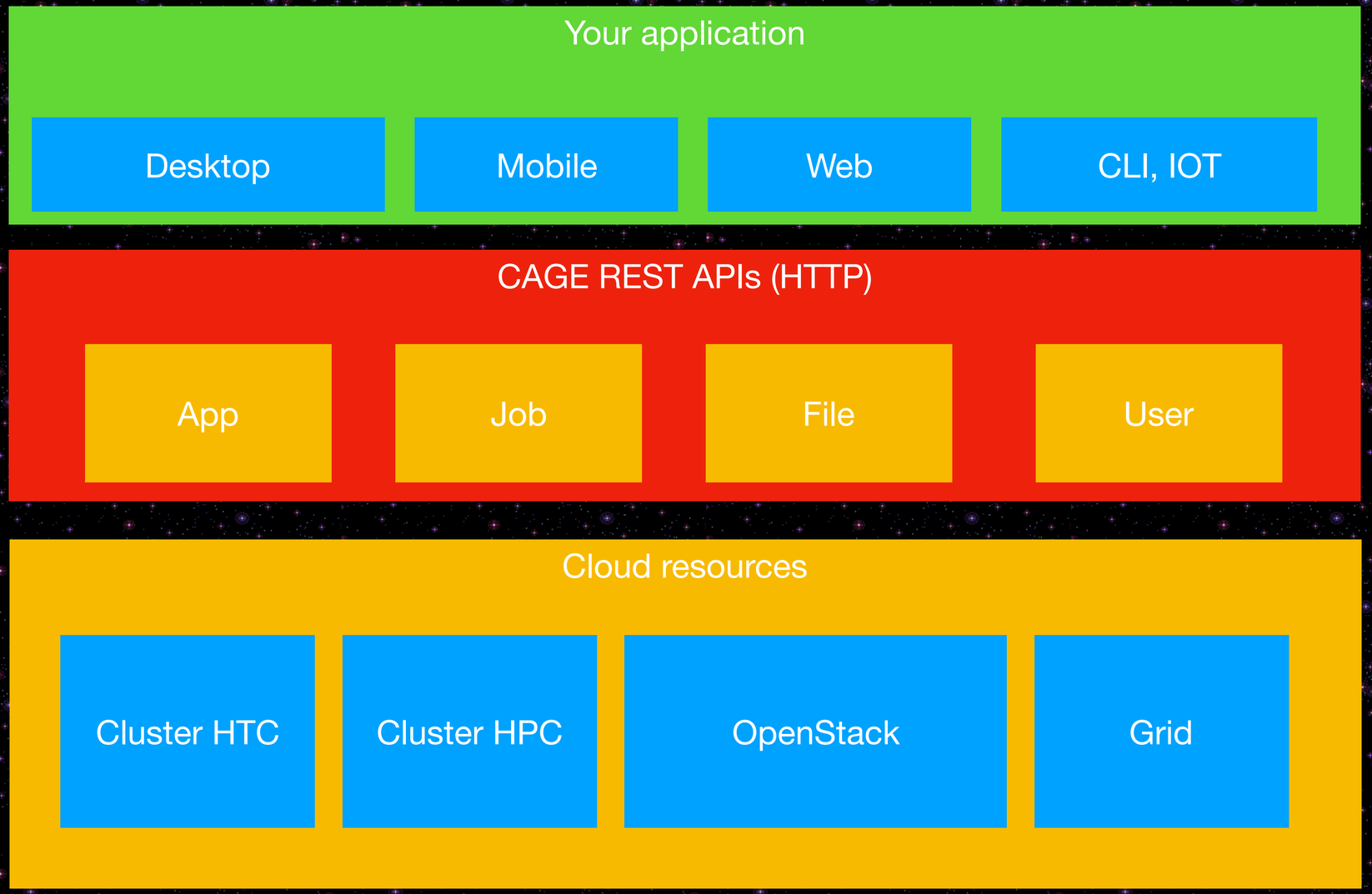
Entering CAGE: Cloud for Astrophysics GatEway

- a **minimalistic** environment to cloud resources from **desktop**, **mobile**, **web**, **command line** scripts, **embedded devices** or any client capable of using the **HTTP protocol**
- a simple set of **RESTful APIs** to manage 4 entities to define and run scientific workflows on any execution environment

CAGE Entities

- **App**: any form of workflow that can be run on Cloud resources
  - **executables** and any environment **configuration** setting that specifies where and how the app can be run
- **Job**: instance of an App, with its own input parameters and data
  - monitor job's execution, retrieving final output
- **File**: a input or output data file
  - associated with Jobs
- **User:** user associated with each app
  - create, login/logout user, ACLs support

CAGE layered architecture

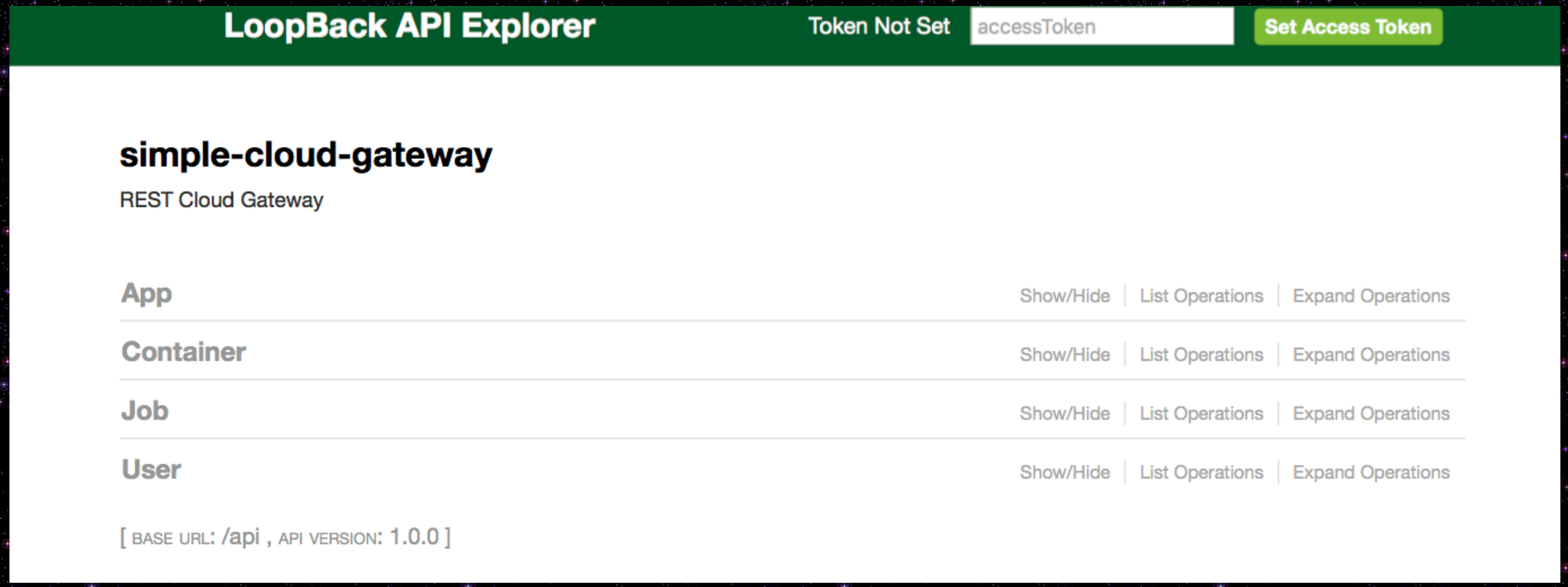


CAGE User Management

- CAGE has its **own users**
- Users belong to the institution that deploy a CAGE installation
- A CAGE server instance is configured with credentials (Robot certificate) enabled to interact with the target execution infrastructures
- CAGE keeps track of every API request of its users
- A powerful ACL system can be employed to limit apps/data to a given set of users

CAGE current prototype

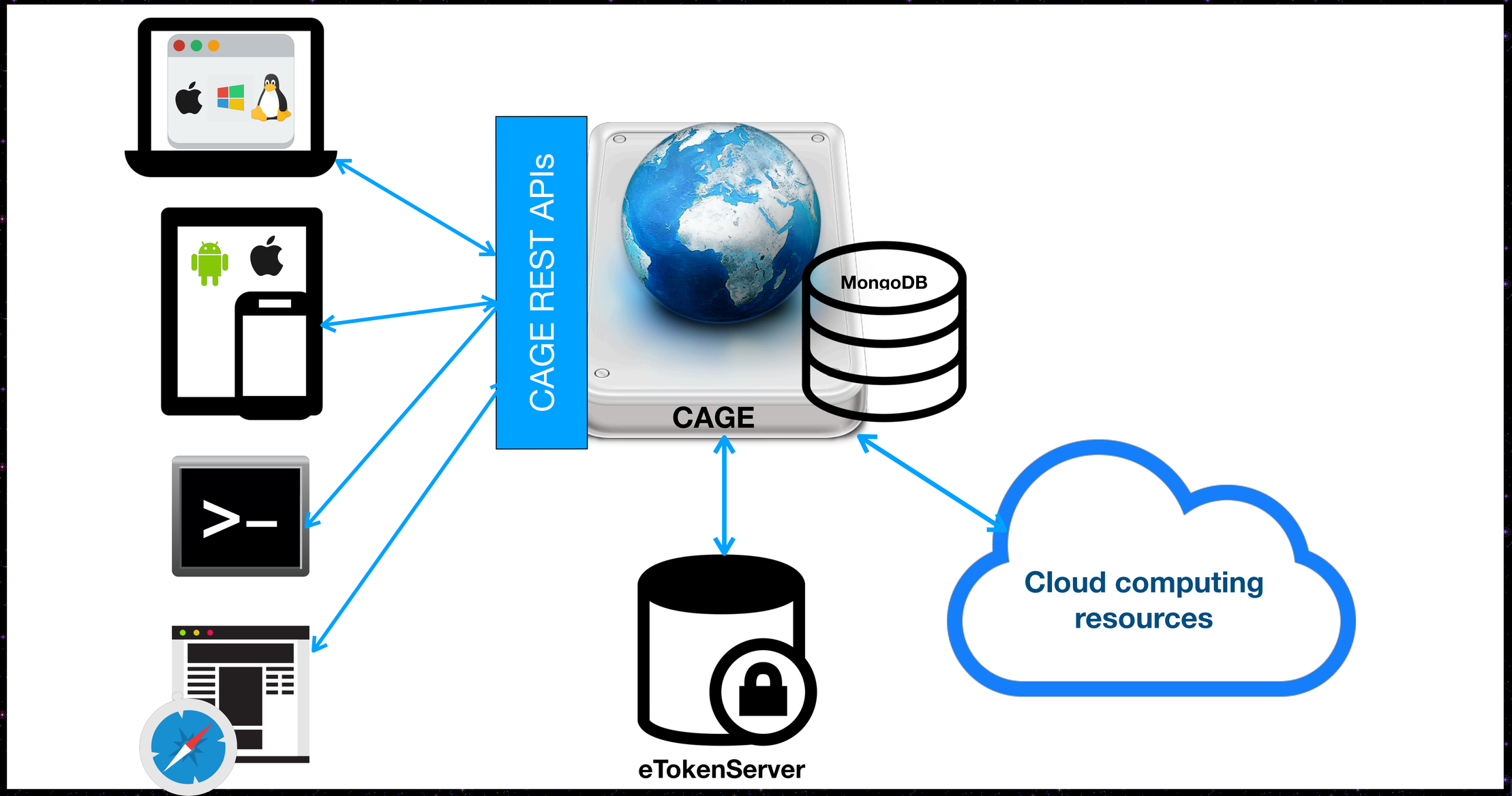
- Each Entity exposed as a set of RESTful API
  - provider canonical CRUD (Create, Read, Update, Delete) operations



CAGE current prototype

- Currently able to run workflows on the EGI FedCloud infrastructure
  - a robot certificate is deployed on a Token Server provided by EGI
- An access token is used to implement sessions among different request

Current prototype architecture



CAGE implementation details

- API Server built using Node.js and IBM Loopback framework
- MongoDB used to store app data
- (Currently) input and output data stored locally on the CAGE server
  - (planned) integrate with Cloud Storage (Openstack Swift)

Thank you for your attention

Do you have any questions?

Autore/i:
Contatti: antonio.calanducci@inaf.it

INAF
ISTITUTO NAZIONALE DI ASTROFISICA
NATIONAL INSTITUTE FOR ASTROPHYSICS

ASTRI

AC DC
Astri/CTA Data Challenge