Open Deep Learning and Data Management of Large Datasets in Hybrid Clouds

Davide Salomoni (davide@infn.it)
INFN-CNAF
CS3, Rome - 28/1/2019

About myself

- Once upon a time a physicist, but since 1991 playing with computers, networks, data distribution and IT in general.
- Coordinator or contributor to several Cloud- or Big Data-related projects.
- I manage the Software Development & Distributed Systems group at the INFN National Computing Center (Bologna) and I am active in several IT-related groups, universities, etc.

My background: INFN (National Institute for Nuclear Physics) – www.infn.it

4 National Labs

20 Site

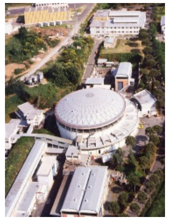
6 Linked groups

3 Nazional Centers

1 Consortium

LNL
Laboratori Nazionali di Legnaro

LNGS
Laboratori Nazionali del Gran Sasso



LNF
Laboratori Nazionali di Frascati

LNS
Laboratori Nazionali del Sud

INFN & Computing

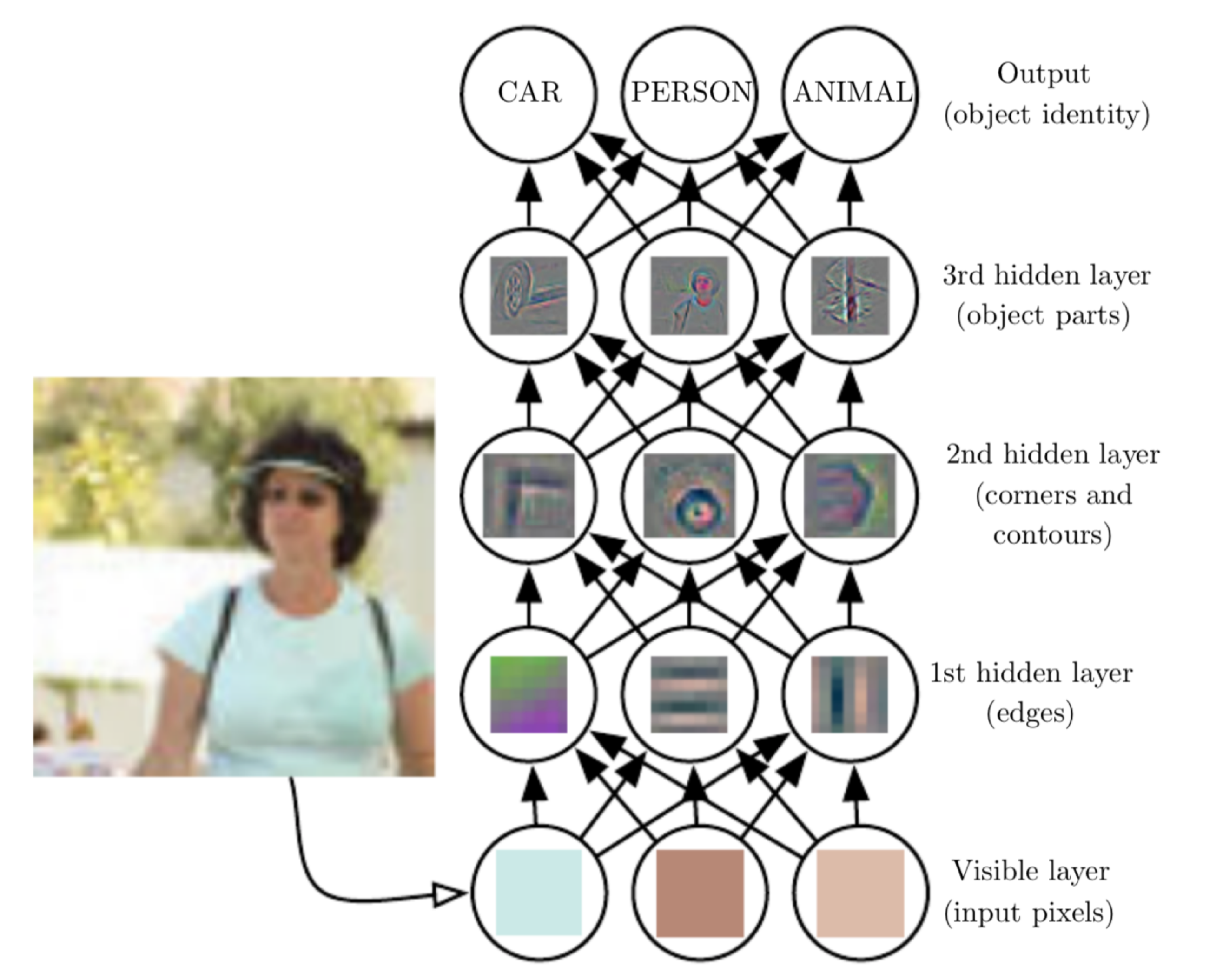
- A long tradition, from the first small clusters to GRID and Cloud based large scale computing
- INFN is not interested in computing per-se, but as an essential way to support its research
- In the past 10 years, this has meant mainly to support the experiments @ CERN (LHC)
- Currently, INFN operates:
    - 9 medium size centers (Tier-2s in the Worldwide LHC Computing GRID hierarchy)
    - 1 large Tier-1 center, at CNAF (Bologna) – certified ISO-27001
- All the INFN centers are connected with at least 10 Gbit/s dedicated connections via the GARR network, currently being upgraded to 100 Gbit/s
- Collectively, our main centers have about 65,000 CPU cores, 50PB of enterprise-level disk space, 60PB of tape storage.

This lecture

My view of a practical journey describing some of the hurdles (as well as solutions) encountered when dealing with deep learning, data management, open science and assorted infrastructures.

What is Deep Learning?

- Break a complicated mapping into a set of nested simple mappings through a set of hidden layers
  - E.g., from pixels to edges; from edges to collections of edges (corners and contours); from collections of edges to specific parts of these collections (object parts); then to complete objects.
- From Goodfellow, Bengio and Courville, https://www.deeplearningbook.org



DL as a black box?

- Do we really need have «models» to explain the behavior of something?
  - «The end of theory: the data deluge makes the scientific method obsolete», https://www.wired.com/2008/06/pb-theory/
    - → "Forget taxonomy, ontology, and psychology. Who knows why people do what they do? The point is they do it, and we can track and measure it with unprecedented fidelity. With enough data, the numbers speak for themselves."
- On the other hand, what about "interpretable AI"?
  - Can society tolerate not understanding?
  - GDPR, Recital 71: "[the data subject should have] the right ... to obtain an explanation of the decision reached".

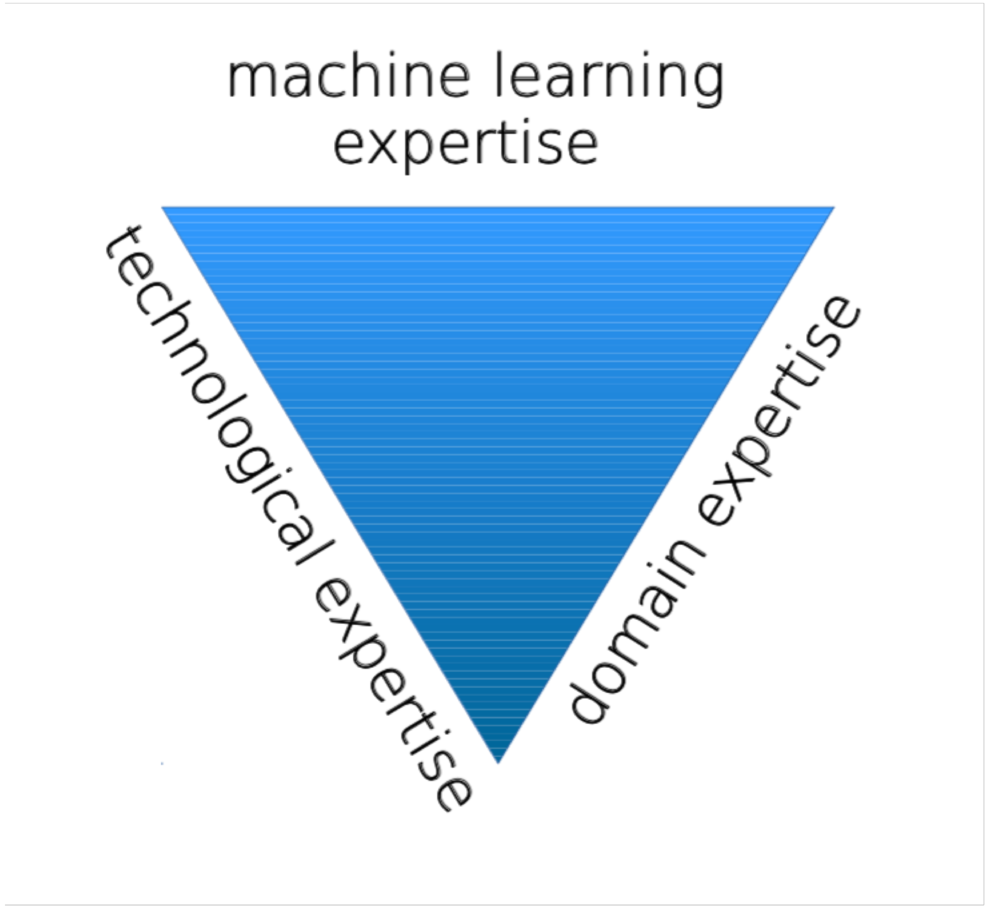
But, in practice, how does it work?

- The naïve assumption:
- I *have*:
  - A data set I want to analyze
  - Some algorithms I want to apply to this data
  - Some software that can use these algorithms
  - Some computing resources that can run this software
  - Some space where I can store my output
- I *assemble everything together* and off I am.

In fact, there are several challenges

- Is the data open? For all? Always?
- Is the data distributed? Where? How do I find and integrate it?
- How open are the programs that will process my data?
- Where can I find the resources I need?
- Does all my data require the same QoS? The same algorithms? How do I decide that?
- What if I get new data? How do I re-train my model?
- What happens if some services are not available? If there is a failure somewhere?
- How can I reproduce, tweak, publish my work?
- How much effort is needed to have all this in place?

Know-how?



Plenty of tools…

TensorFlow: speech and image recognition (Google Brain Team)

Keras: Python NN library (Francois Challet, Google)

PyTorch: DL library (Facebook KI)

Caffe: DL library (UC Berkeley)

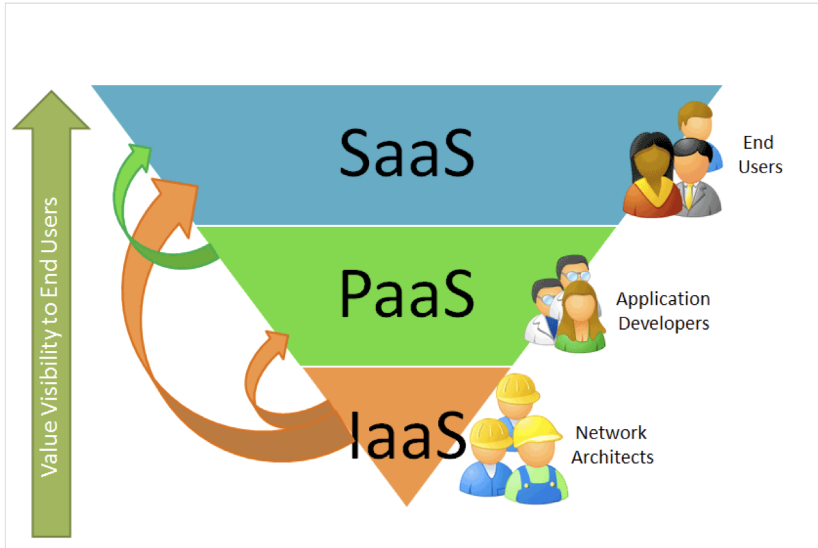
mxnet: scalable DL framework (Apache)

OpenCV computer vision

NumPy num. lin. alg.

SciPy.org sci. comp.

matplotlib plotting



Categorizing the know-how

- Category 1: Deploy a readily trained network for somebody else to use on his/her own data set

  → Domain knowledge

- Category 2: Retrain (parts of) a trained network to make use of its inherent knowledge and to solve a new learning task

  → Domain + machine learning knowledge

- Category 3: Completely work through the deep learning cycle with data selection, model architecture, training and testing

  → Domain + machine + technological knowledge

Courtesy A. Lopez

How do we go about this?

- **Goal: build added value and advanced services on top of IaaS and PaaS infrastructures**
  - Due to the nature of many scientific endeavors (but also public services and commercial industries), these infrastructures may often be *hybrid* (public + private)
- Lower the entry barrier for **non-skilled** scientists
  - Transparent execution on e-Infrastructures
  - Build ready-to-use modules, components or services, and offer them through a catalog or a configurable marketplace
  - Enable flexible service composition
  - Implement common software development techniques also for scientists' applications (DevOps)

A foundation project: INDIGO-DataCloud

INDIGO - DataCloud

- **An H2020 project** funded with 11.1M€, running **from April 2015 to September 2017**.
- **Who**: **26 European partners** in 11 European countries
  - Coordinated by the Italian National Institute for Nuclear Physics (INFN)
- **What**: **develop an open source Cloud platform** for computing and data ("DataCloud") tailored to science but applicable to other domains as well.
- **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.

The INDIGO Software

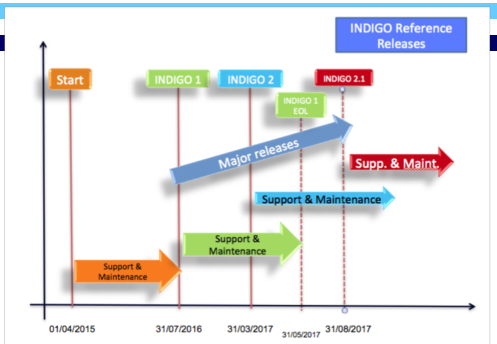


Figure 8: INDIGO-DataCloud Release Timeline

- INDIGO released two major software versions (MidnightBlue and ElectricIndigo)
- **ElectricIndigo (https://www.indigo-datacloud.eu/service-component):**
  - 47 open source modular components, distributed via 170 software packages, 50 ready-to-use Docker containers
  - Supported operating systems: CentOS 7, Ubuntu 16.04
  - Supported cloud frameworks: OpenStack Newton, OpenNebula 5.x (plus connection to Amazon, Azure, Google)
  - Download it from the INDIGO-DataCloud Software Repository: http://repo.indigo-datacloud.eu/index.html

INDIGO and the EOSC†

- **From vision to action: the EOSC-hub project**
- "EOSC-hub, funded with about 30M€, mobilizes providers from 20 major digital infrastructures, EGI, EUDAT CDI and INDIGO-DataCloud, jointly offering services, software and data for advanced data-driven research and innovation through a unified service catalogue."
- Factoids:
  - 100 Partners, 75 funded beneficiaries
  - 3874 PMs, 108 FTEs, more than 150 technical and scientific staff involved
  - 36 months: Jan 2018 – Dec 2020

† **E**uropean **O**pen **S**cience **C**loud: refer to the talk by Dr. Isabel Campos on 29/1

INDIGO and the EOSC

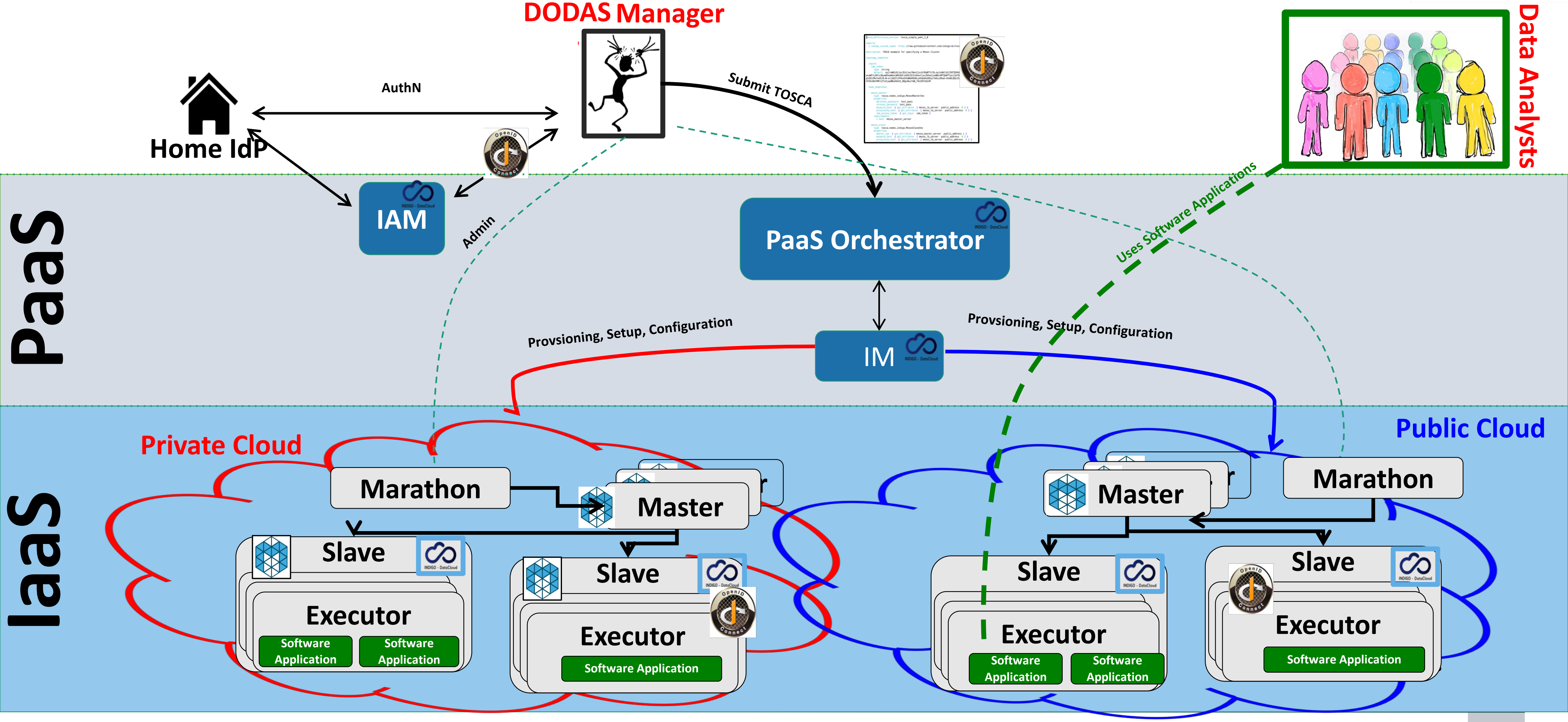
- With INDIGO, we started introducing tools and services for:
  - A common, standards-based AAI model
  - Independence from IaaS infrastructures
  - Service composition with a standard language (TOSCA)
  - Compute orchestration
  - Web and mobile based interfaces

A concrete example: DODAS

- DODAS is a service obtained by the composition of several INDIGO components.
- It provides deployment of complex and intricate set-ups "on any cloud provider" † with almost zero effort.
  - As easy as creating a virtual machine on any IaaS: a simple one-click solution.
  - The details of DODAS configurations are stored in high-level templates written in TOSCA.
  - It allows to instantiate on-demand microservices and container-based clusters to execute software applications.
- DODAS currently provides support to generate:
  - HTCondor-based Batch System as a Service
  - Big Data platform for ML as a Service
  - Plus extensions of these two, integrating community-specific services

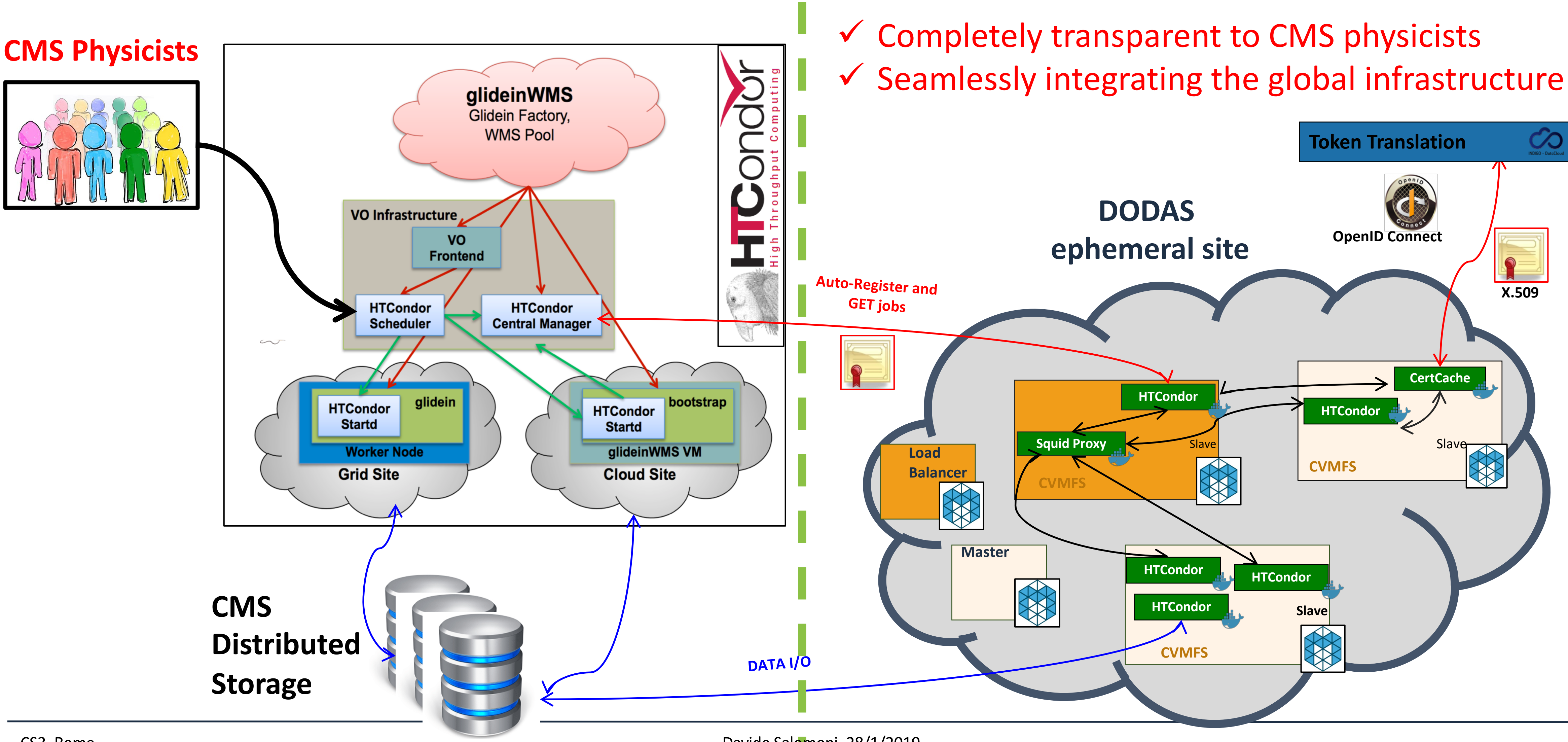
† Deployed on OpenStack-based clouds (public and private), AWS, Google Compute Cloud, Microsoft Azure

The DODAS architecture

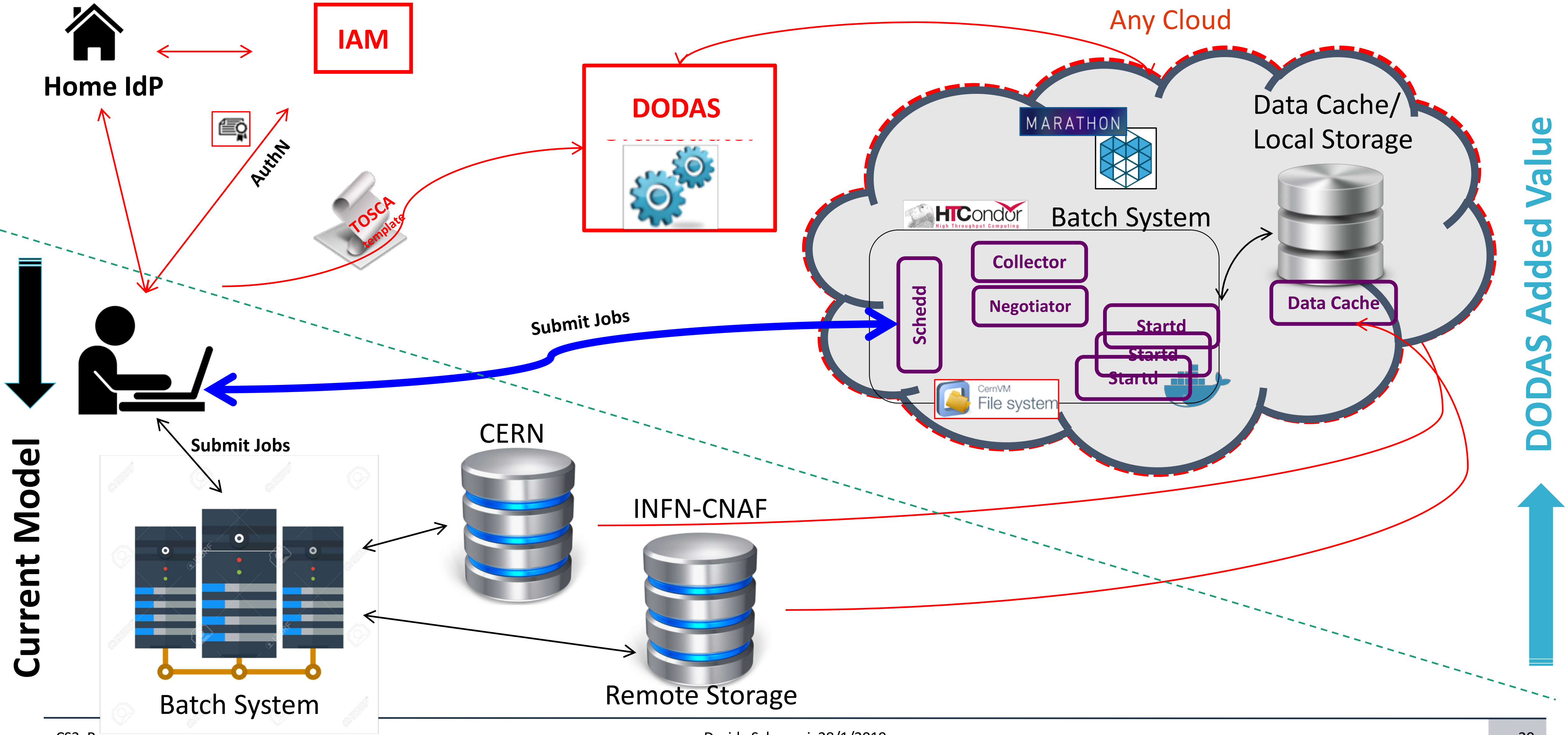


HTCondor Pool Extension: The CMS Use Case

- ✓ Completely transparent to CMS physicists
- ✓ Seamlessly integrating the global infrastructure



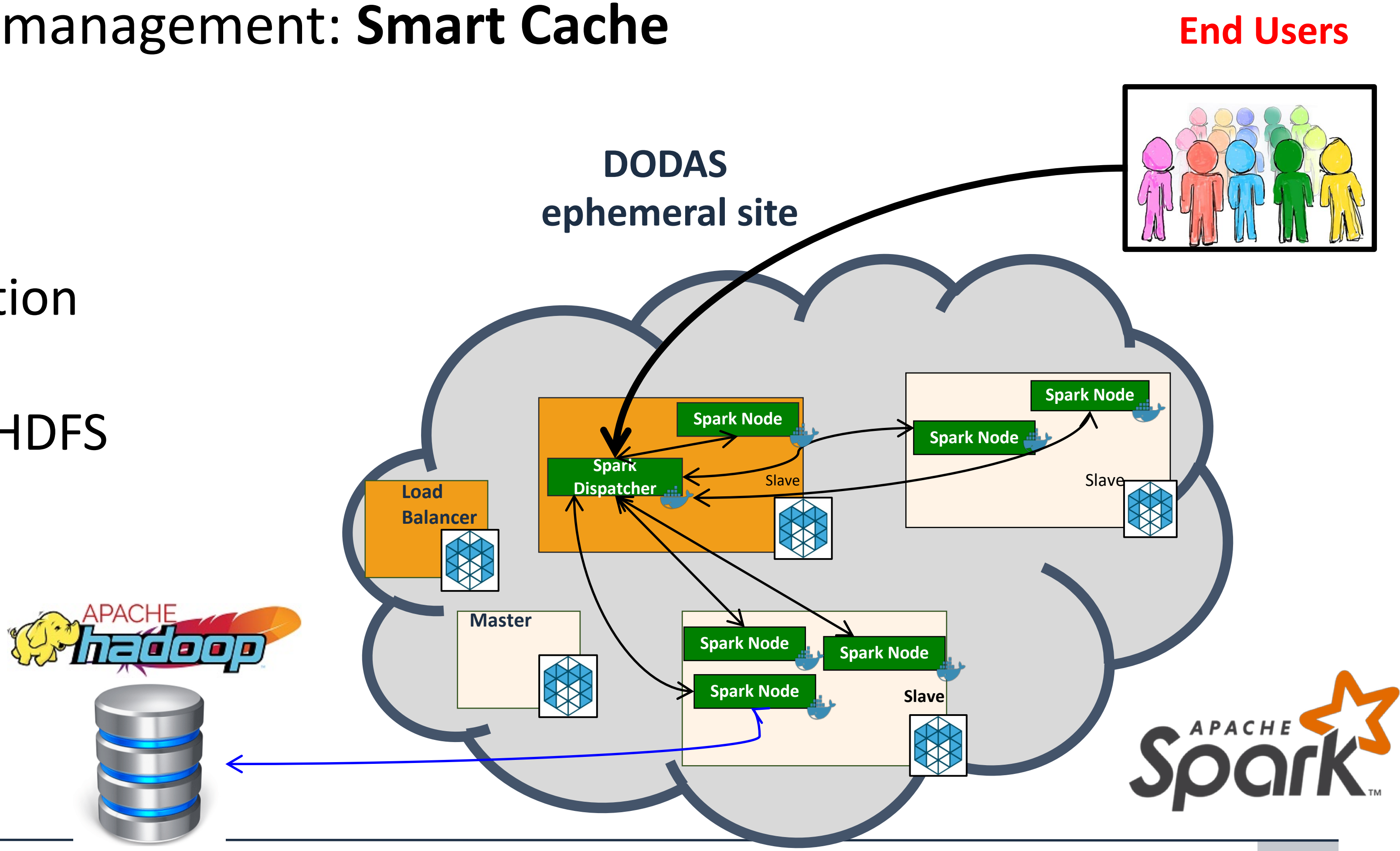
Standalone Batch System as a Service: The AMS use case



DODAS for ML as a Service

- Analysis of "Data Cache" related metadata flow
  - To improve caching layer management: **Smart Cache**

1. Reading HDFS@CERN data
2. Data enrichment and reduction with Spark jobs
   - Storing of output data in HDFS
3. Analysis of structured data



What next?

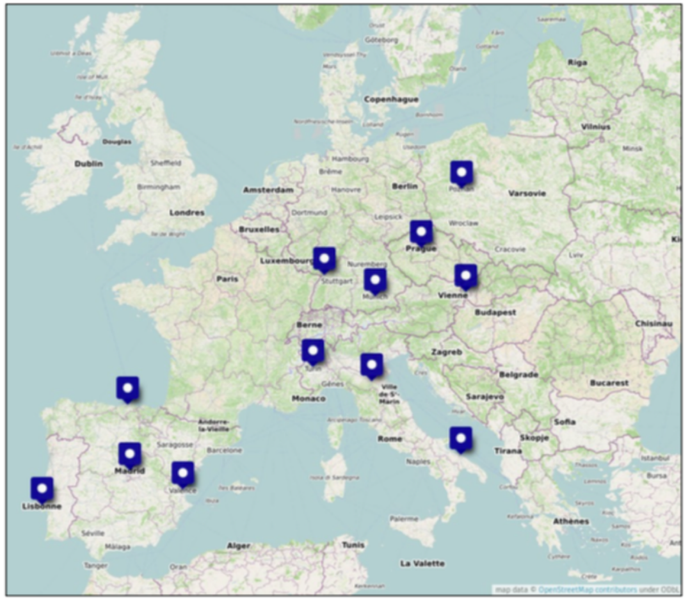
- So, we know how to dynamically instantiate clusters over many IaaS, how to compose services, how to integrate various types of AAI systems, how to integrate various components in existing frameworks, monitor and auto-scale them.
- **But what about Deep Learning?**

Right now…

- Scientists typically create a deep learning application on their personal computers.
- The deep learning model is trained in a GPU-based node (maybe also locally).
  - What happens if they do not have access to one?
- The work is published (or not)
  - Model architecture, configuration, scientific publication, etc.
- But:
  - How can a scientist easily offer it to a broader audience?
  - What about dependencies?

DEEP-Hybrid DataCloud

- Balanced set of partners
  - Strong technological background on development, implementation, deployment and operation of federated e-Infrastructures
- 9 academic partners
  - CSIC, LIP, INFN, PSNC, KIT, UPV, CESNET, IISAS, HMGU
- 1 industrial partner
  - Atos
- 6 countries
  - Spain, Italy, Poland, Germany, Czech Republic, Slovakia

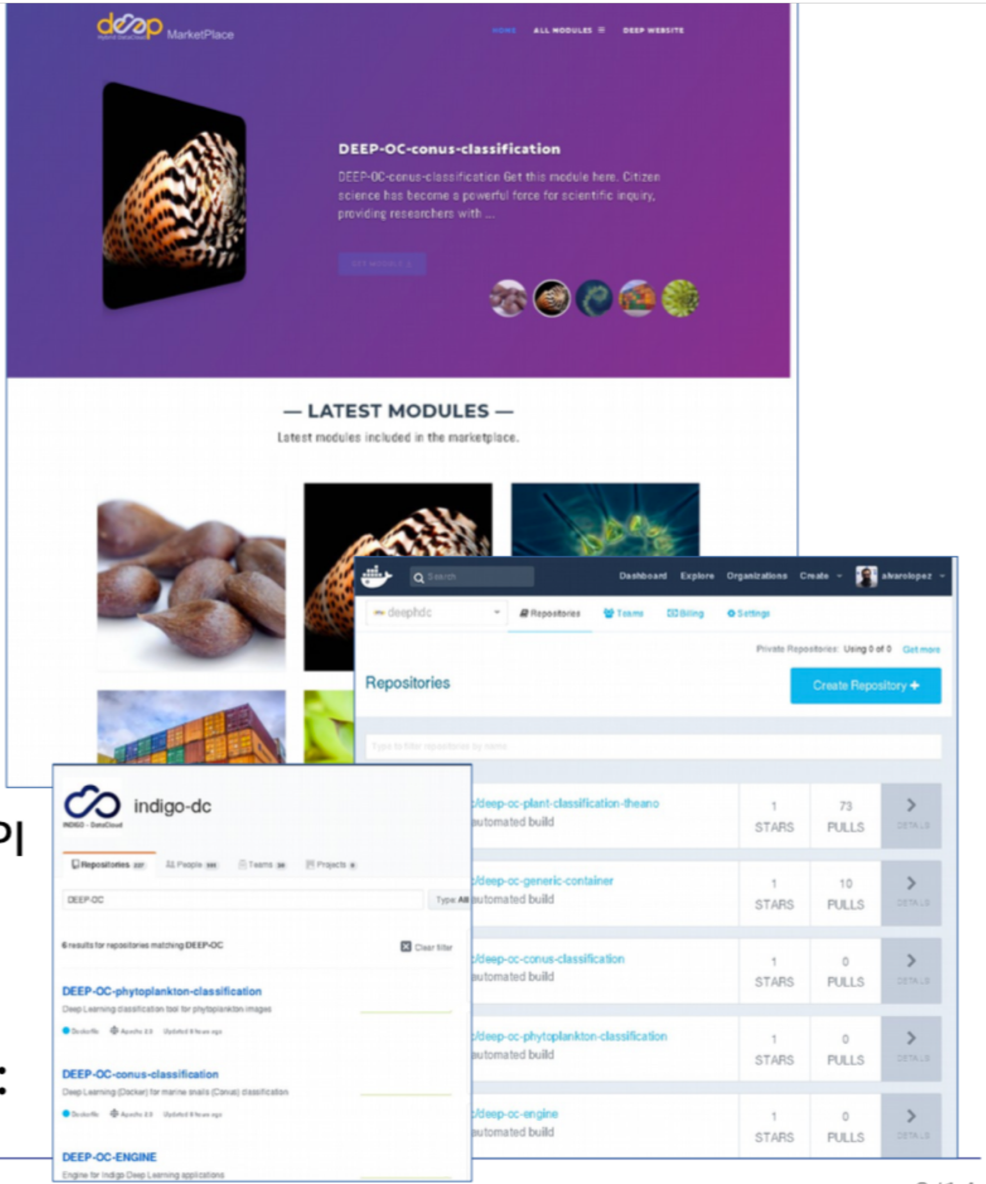


From service composition to reusable components

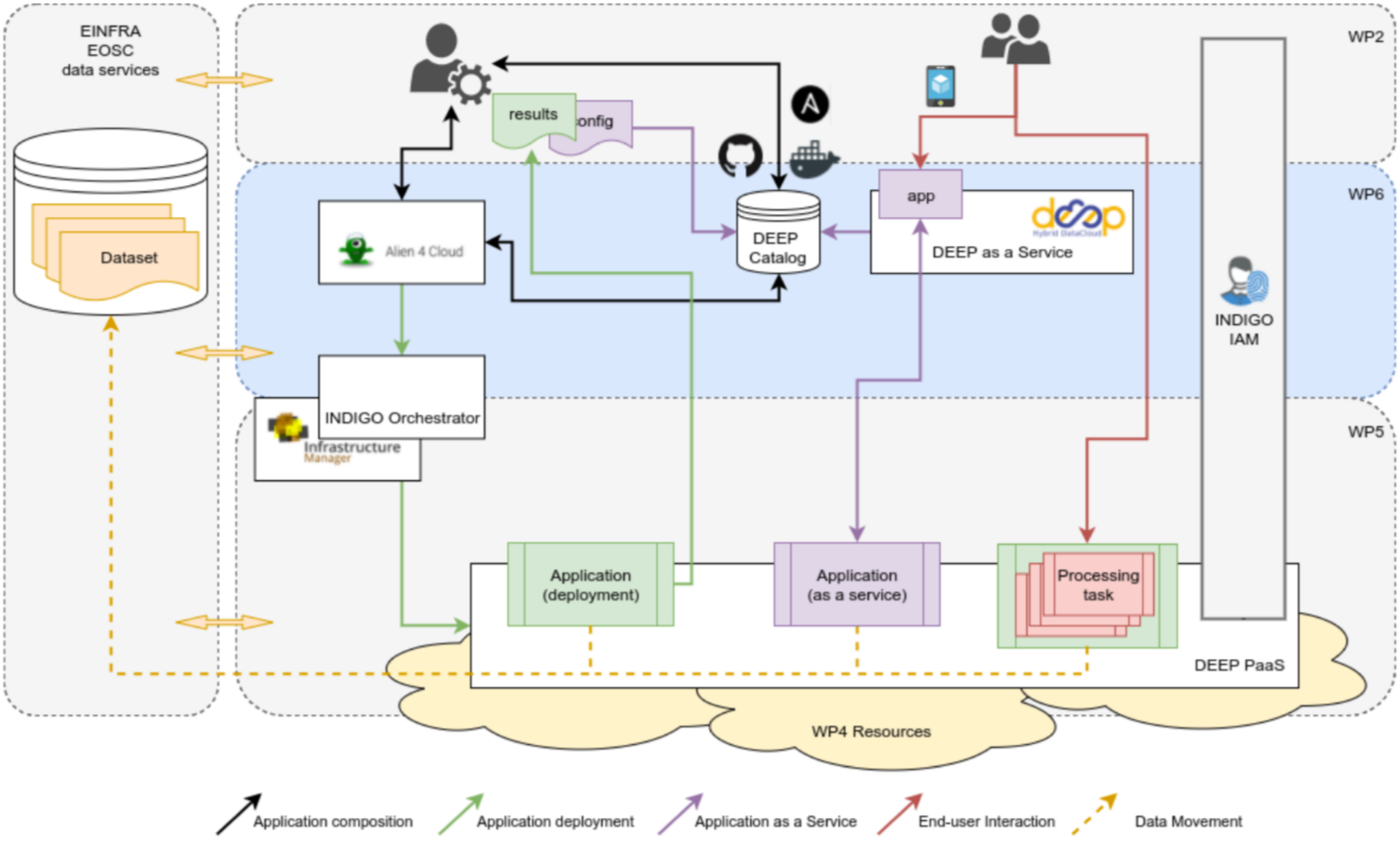
- Service composition, if done properly, provides a way to re-deploy the same topology over different infrastructures.
- Scientists should not need to deal with technologies and infrastructures.
- We have the technology to create service composition with TOSCA-based templates → **catalog of components.**

DEEP Open Catalog

- Collection of ready-to-use modules
  - Comprising machine learning, deep learning, big data analytics tools
  - ML Marketplace
    https://marketplace.deep-hybrid-datacloud.eu
  - GitHub
    https://github.com/indigo-dc?utf8=✓&q=DEEP-OC
  - DockerHub
    https://hub.docker.com/u/deephdc/
- Based on DEEPaaS API component
  - Expose underlying model functionality with a common API
  - Based on OpenAPI specifications
  - Minimal modifications to user applications.
- Goal: execute the same module on **any** infrastructure:



DEEP high level Architecture



What next?

- So, we know how to dynamically instantiate clusters over many IaaS, how to compose services, how to integrate various types of AAI systems, how to integrate various components in existing frameworks, monitor and auto-scale them.
- We also know how to re-use deep learning-based building blocks, customize, and publish them in a high-level catalog of services.
- **But what about data management?**

Right now…

- Scientists typically are oblivious to data distribution policies, in particular for:
  - QoS-based (e.g. disks vs tape vs SSD) data distribution policies, esp. cross-sites
  - Data lifecycle management
- They would like sometimes to perform some data pre-processing during data ingestion: how?
- They would like to control how replica management is done
- They would like to perform some *smart* data caching, or data management based e.g. on access patterns: how?
  - For example, automatically move to some “glacier-like” storage *unused* data, and conversely move “hot” data to some fast storage.

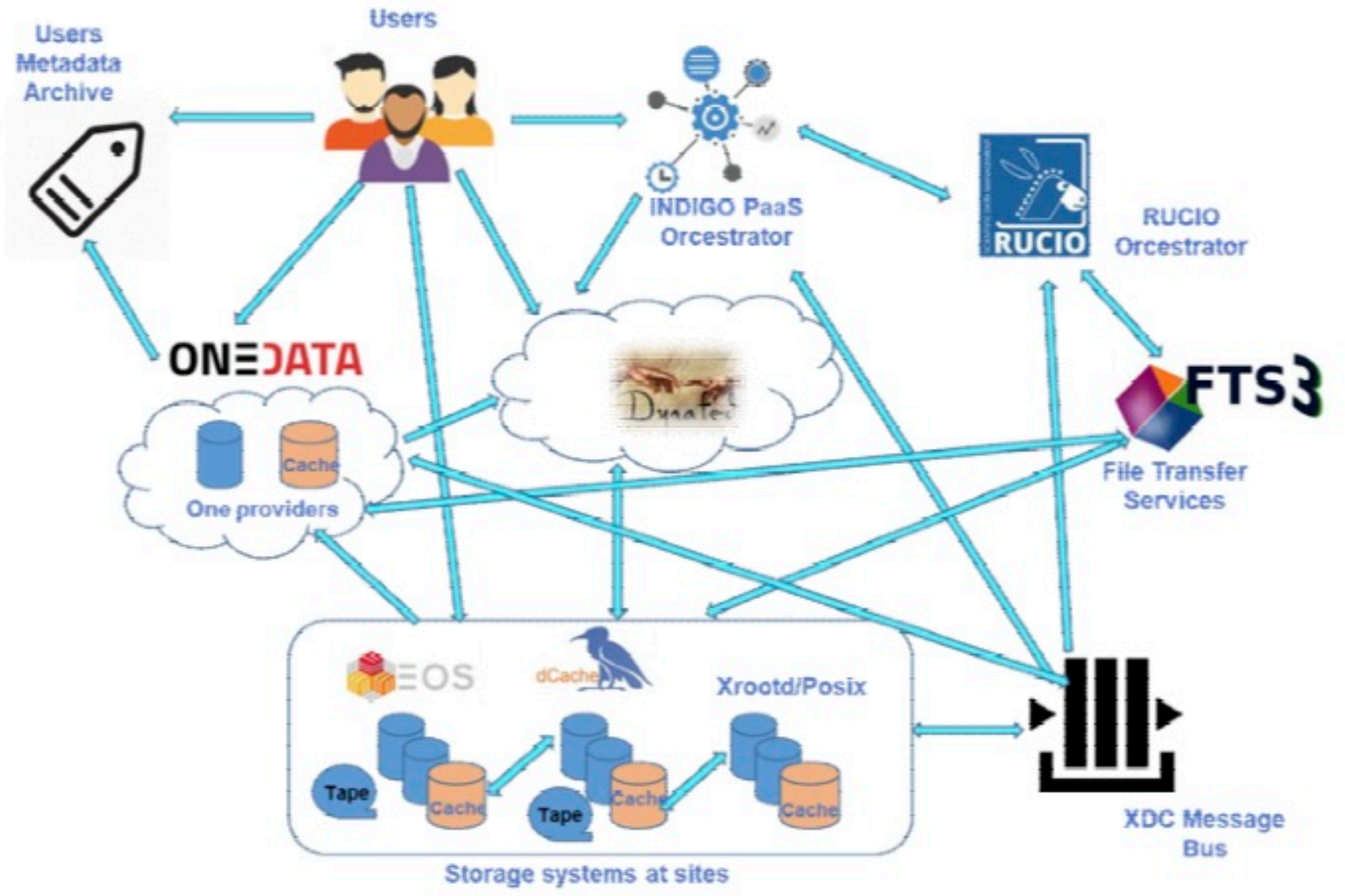
eXtreme-DataCloud (XDC)

XDC Overview

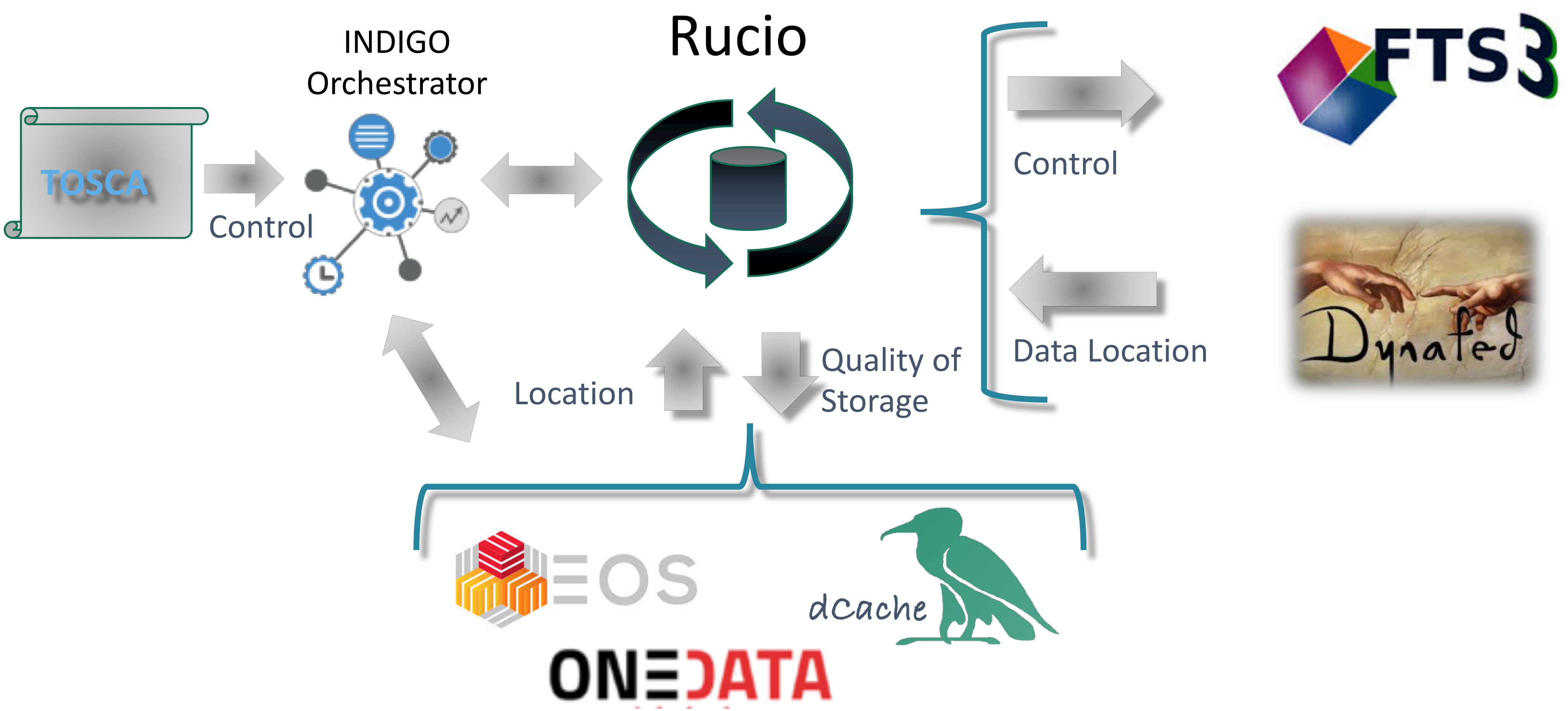
- The eXtreme DataCloud's aim: Develop scalable technologies for federating storage resources and managing data in highly distributed scientific computing environments
- XDC is a 2 year, 3M€, EU-funded software development and integration project
  - Started active work 1st Feb 2018
- The targeted platforms are the current and next generation e-Infrastructures deployed in Europe
  - European Open Science Cloud (EOSC)
  - The e-infrastructures used by the represented communities

The XDC Architecture

- **Main point**: improve already existing, production quality Data Management services by adding missing functionalities requested by research communities. Based mainly on technologies provided by the partners and by the INDIGO-Datacloud project.



XDC Control Flow for Orchestration



Storage QoS Support

- Definition currently based on a RDA working group
- ”Cloud Data Management Interface” chosen as control protocol.
  - Defined by SNIA.
  - INDIGO acknowledged by SNIA as contributor to the reference implementation.
- Implementing the defined API into GFAL, dCache, EOS and StoRM

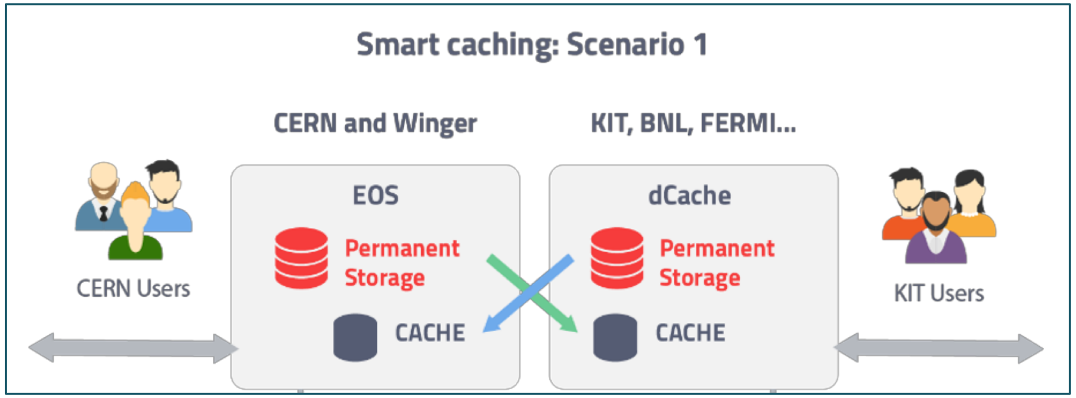
From INDIGO-DataCloud

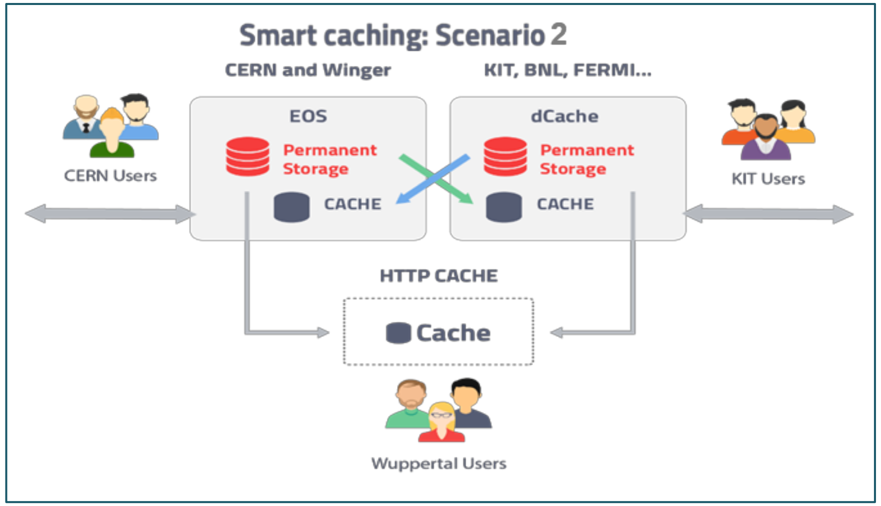
Name	Access Latency [ms]	Number of Copies	Storage Lifetime	Location	Storage type	Available Transitions
disk	100	1		DE	Processing	tape, disk+tape
disk+tape	100	2		DE	Processing	tape
DiskAndTape	50	3	20 years	DE	Processing	
DiskAndTape	50	2		IT	Processing	
DiskOnly	50	3	20 years	DE	Processing	
DiskOnly	50	1		IT	Processing	
profile1	10	3	20 years	DE	Processing	

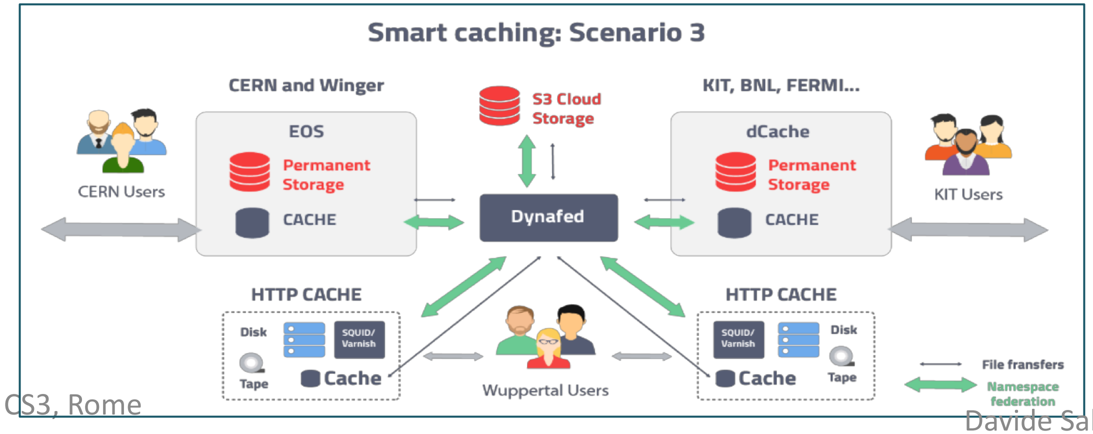
- Access Latency [ms]
- Number of Copies
- Storage Lifetime
- Location
- Available Transitions

XDC Smart caching

- Develop a global caching infrastructure supporting the following building blocks:
  - dynamic integration of satellite sites by existing data centres
  - creation of standalone caches modelled on existing http and xrootd solutions
  - federation of the above to create a large scale, regional caching infrastructure







What next?

- So, we know how to dynamically instantiate clusters over many IaaS, how to compose services, how to integrate various types of AAI systems, how to integrate various components in existing frameworks, monitor and auto-scale them.
- We also know how to re-use deep learning-based building blocks, customize, and publish them in a high-level catalog of services.
- We also know how to perform some data management automation and optimization, as well as how to call some QoS functions on storage.
- **But is this it? Did we address all the challenges mentioned above?**

In fact, there are several challenges (reloaded)

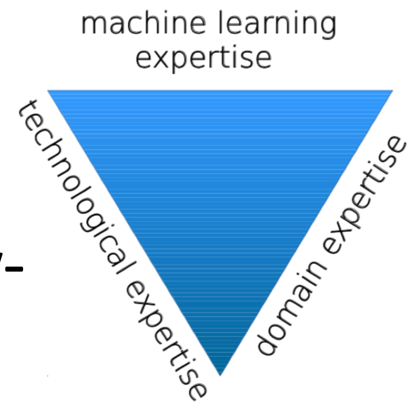
- Is the data open? For all? Always?
- Is the data distributed? Where? How do I find and integrate it?
- How open are the programs that will process my data?
- Where can I find the resources I need?
- Does all my data require the same QoS? The same algorithms? How do I decide that?
- What if I get new data? How do I re-train my model?
- What happens if some services are not available? If there is a failure somewhere?
- How can I reproduce, tweak, publish my work?
- How much effort is needed to have all this in place?

(Some of) The still missing pieces

- EOSC or not, we still need to invest effort in developing and integrating **open source modular solutions** capable of:
  - Securely connecting to multiple data sources: e-infrastructures, HPC centers, opportunistic resources, devices, storage systems, data sets, sync & share services.
  - Integrating and automating a full data + compute orchestration solution.
  - Being able to react to events, such as the insertion of new files in catalogues, DB, file systems.
  - Exploiting metadata-driven lambda-based processing (function-as-a-service) models.
  - Enabling user-centric service composition capable of triggering automated, extensible processing, deployment and monitoring for the above.
  - Supporting effective co-design of solutions by interdisciplinary scientists, as well as many other potential stakeholders, through a configurable selection of components and reproducible workflows.

In summary

- It is naïve to think that silos-based, proprietary, monolithic solutions will address the explosion of data production and related analysis, esp. with complex requirements such as those found with DL and open science.
- Transparency, support of *de jure* and *de facto* standards, provider-agnostic modular solutions are the way to go.
- We (research *and* industry) have the possibility and the know-how in Europe to create an inclusive ecosystem – the EOSC – supporting this.
- Several steps have already been taken (and some were mentioned here), but there are exciting challenges and times ahead of us, and room for many to play a significant role!



Thanks!

- Acknowledgments: Alvaro Lopez Garcia, Giacinto Donvito, Daniele Spiga, Wolfgang zu Castell, Daniele Cesini.
- For more info on some of the mentioned projects:
  - INDIGO-DataCloud, https://www.indigo-datacloud.eu
  - DEEP-Hybrid DataCloud, https://deep-hybrid-datacloud.eu
  - eXtreme-DataCloud, http://www.extreme-datacloud.eu
  - EOSC-hub, https://www.eosc-hub.eu