Cloudy Thoughts: Opportunities and Challenges in Scientific Distributed Computing

Davide Salomoni, davide@infn.it

6th IDPASC/LIP PhD Students Workshop

25-27/6/2020

Agenda → from the Talk Abstract

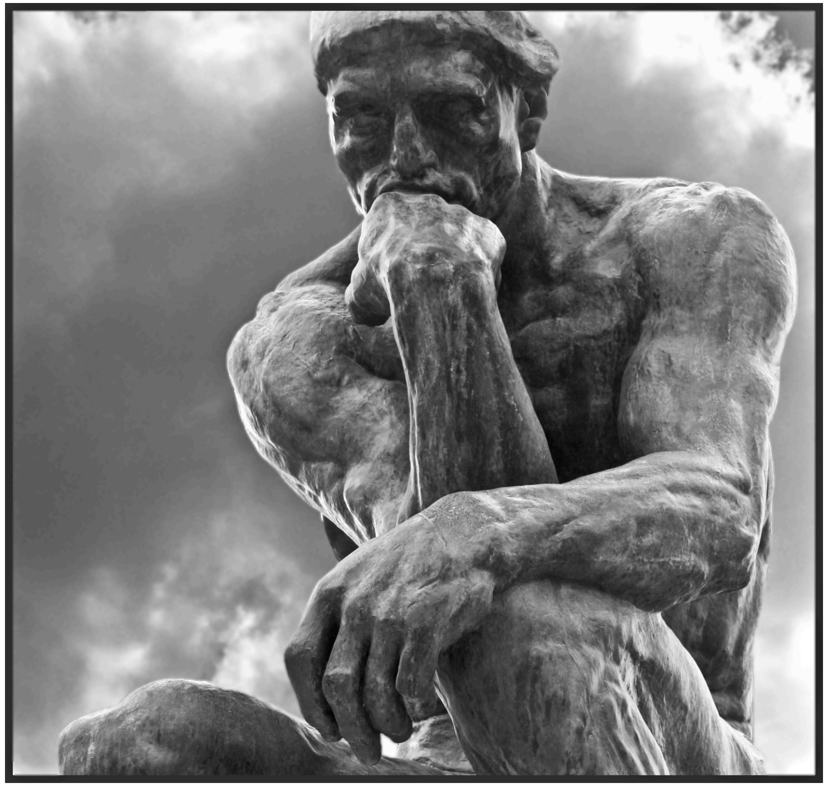
"Why should PhD students in particle physics, astrophysics and cosmology worry about the current and future challenges of distributed computing?

Should not we all focus on core scientific problems, and then let *others* deal with how to find, provision and exploit the technologies that are required?

This talk will show how that was not the case in the past and will likely not be the case also for upcoming scientific experiments and needs, pointing to some of the opportunities and challenges that lie ahead of us."

More analytically

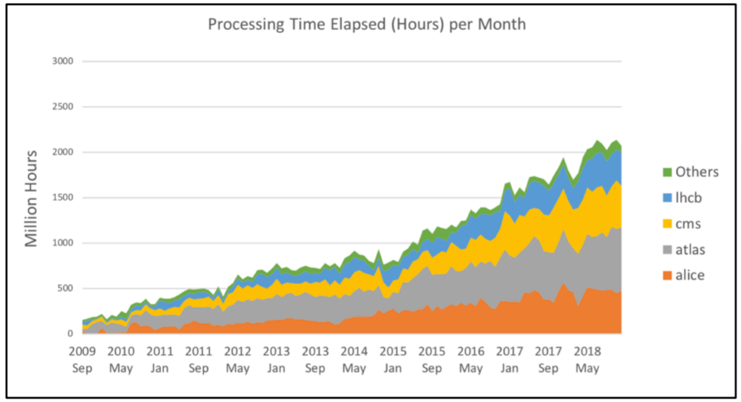
- Background
- Opportunities *and* challenges
  - Integration
  - Software
  - Technology
  - Processes
- Conclusions



Executive summary:
You will find lots of questions here, with pointers to some possible answers.

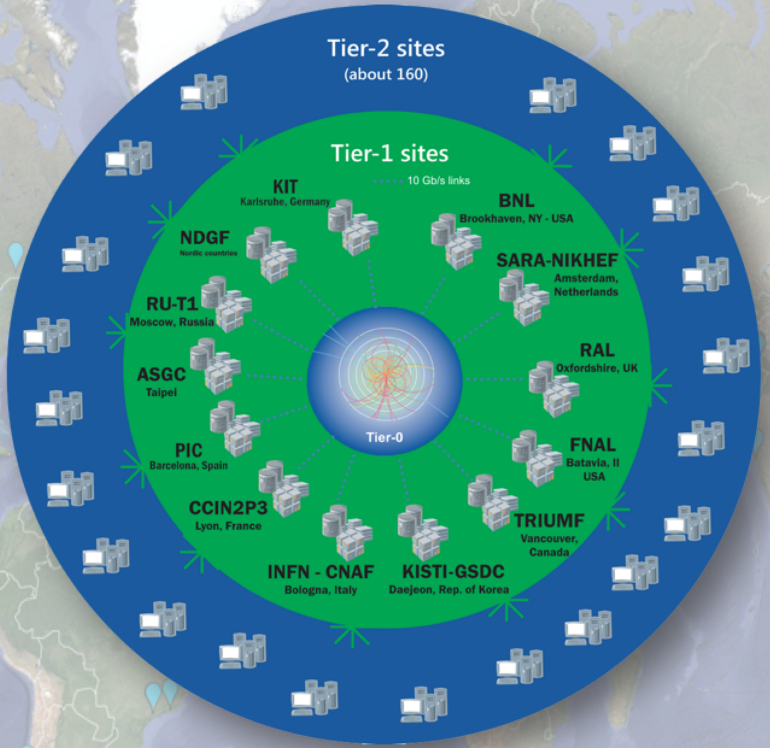
Background

Where are we coming from?



HEP computing embraced a large scale distributed model since early 2000s
Based on grid technologies, federating national and international grid initiatives

WLCG
Worldwide LHC Computing Grid



1 TB ~ 10-100 CHF

1 core ~ 100 CHF

HW lifetime: 3-5 years

167 sites, 42 countries

~1M CPU cores

~1 EB of storage

> 2 million jobs/day

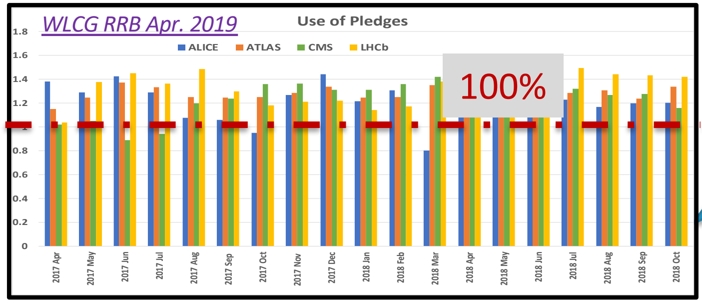
10-100 Gb links

WLCG: an International collaboration to distribute and analyse LHC data

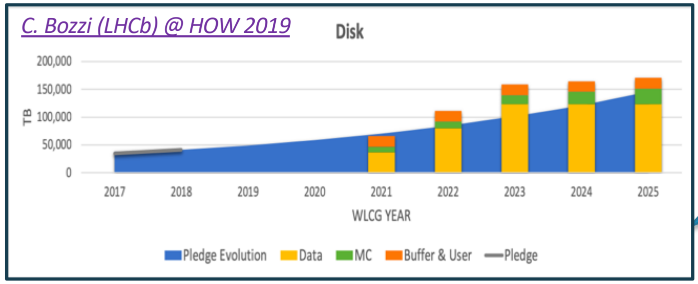
Integrates computer centres worldwide that provide computing and storage resource into a single infrastructure accessible by all LHC physicists.

The data explosion

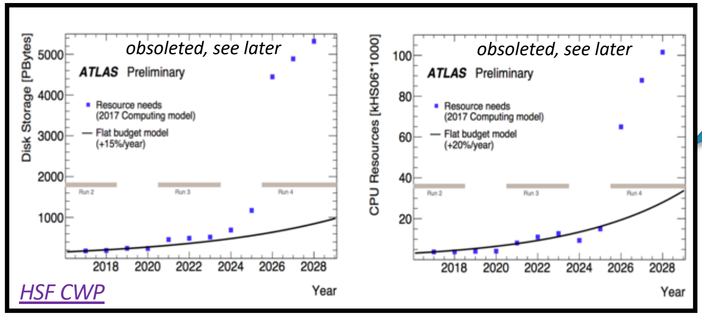
2018



Run-3



HL-LHC



Today LHC generally gets the **requested** computing resources. Extra **opportunistic** compute capacity is available

The LHC data volume challenge

ALICE and LHCb will increase considerably the data rates in 2022 (LHC Run-3)

ATLAS and CMS will increase the event rates by a factor 10 and the event complexity in Run-4 (HL-LHC)

No expected increase of funding for computing: “flat budget”

And Moore’s et al. laws?

- A few empirical laws are common when trying to predict the costs of resources with time:
  - **Moore’s law**: The number of transistors on integrated circuits doubles approximately every two years”. This can be translated into “every two years, for the same money, you get a computer twice as fast”;
  - **Kryder’s law**: “the capacity of Hard Drives doubles approximately every two years”;
  - **Butter’s law of photonics**: “The amount of data coming out of an optical fiber doubles every nine months”;
  - **Nielsen’s law**: “Bandwidth available to users increases by 50% every year.

  - … All not realistic any more …

Background: so what?

- Future (today +10y) HEP experiments do not have an easy path to computing.
  - A simple extrapolation of today's models diverges financially by a factor 10x in the next 10 years.
- If this is to remain true, computing would cost more than the accelerator and the experiments.
  - A no-go from funding agencies…
  - … which basically require a "**flat-budget**" model for computing.

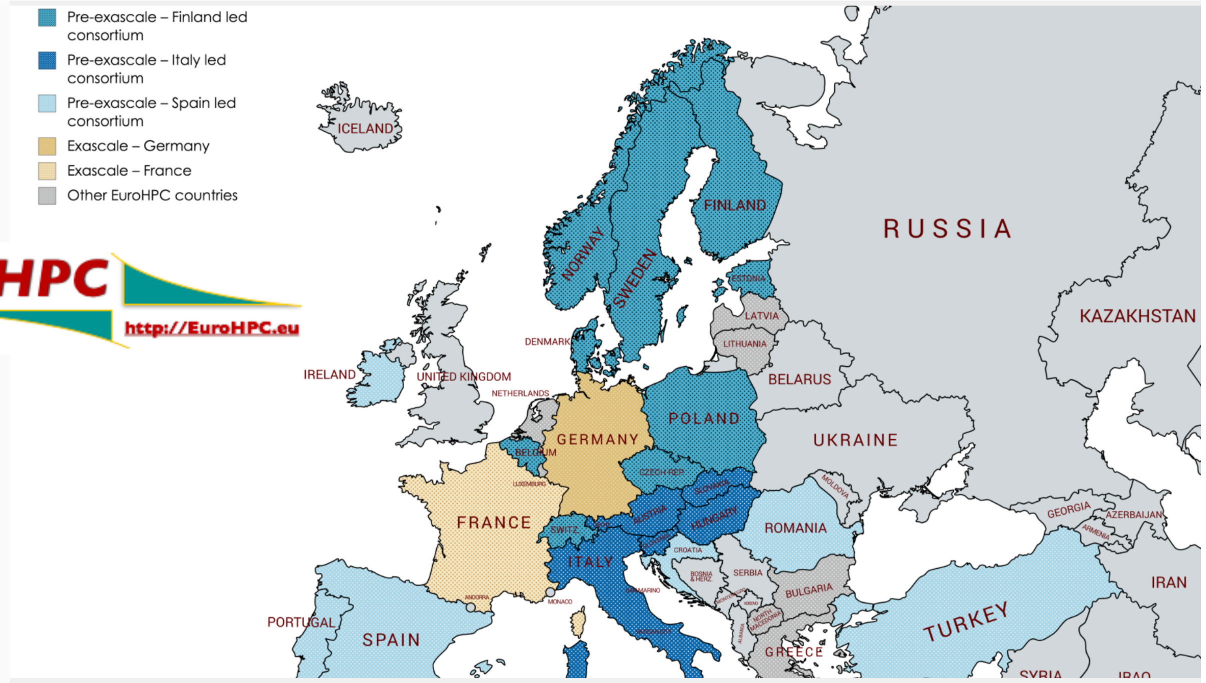
- **What are the solutions / paths we can try to follow towards a mitigation of this problem?**

HPC and Heterogeneous Resources

- Up to 25% of resources currently used by LHC experiments comes from non-Grid facilities
  - Cloud computing
  - HPC
  - HLT farms
- Potentially, *there is a lot of improvements for apps using HPC vs. standard CPUs.* However:
  - Data access? (bandwidth?)
  - Accelerator technology? (KNL, NVIDIA GPU, FPGA, TPU, AWS Nitro, …)
  - Workload scheduling? (MPI vs. batch systems vs. proprietary systems)
  - Node configurations? (e.g. low RAM / Disk)
  - Not-too-open environment? (e.g. O/S)

Using HPC "cloud-friendly" offerings is not a fluid shared IaaS transaction.
Vertical Integration is eating the datacenter

EuroHPC
http://EuroHPC.eu

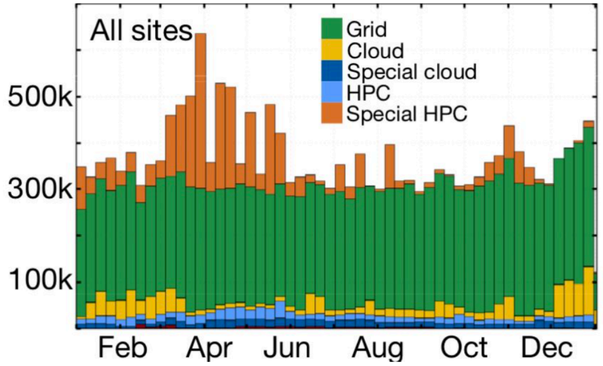


Participants in petascale & pre-exascale hosting entitites

ATLAS EXPERIMENT

D. Costanzo @ HOW 2019

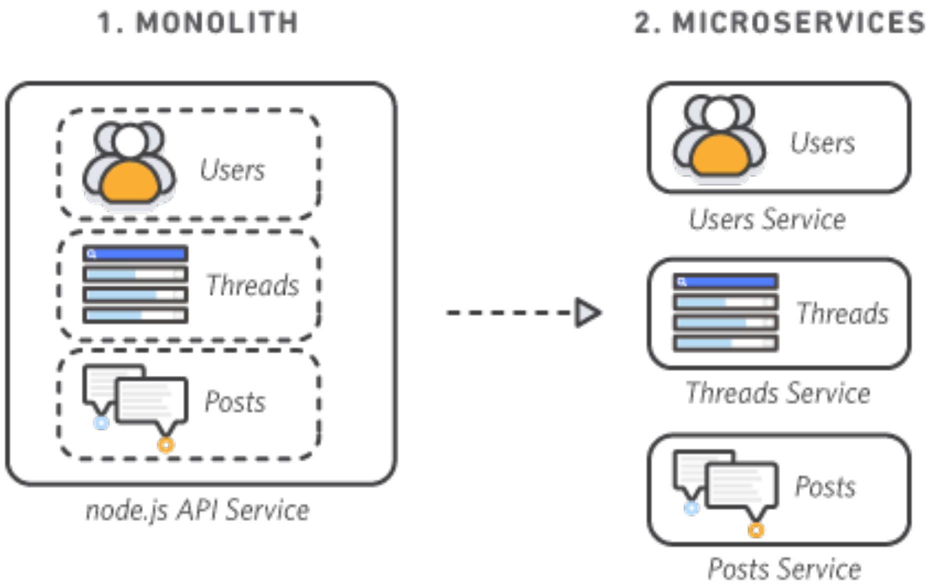
CPU usage 2018

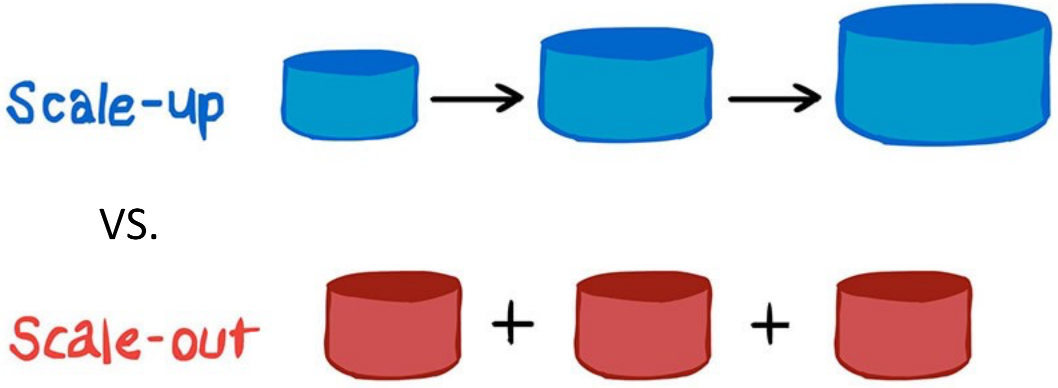


Legacy vs. Cloud

- "Porting apps to the Cloud" does **not** mean "creating a Docker container so that my monolith can run in a VM".
- Go instead the "Cloud-native" way, i.e. microservice-based: learn to *break the monolith*.

- But you also often need to *make legacy app coexist (e.g. talk) to cloud-native apps.*

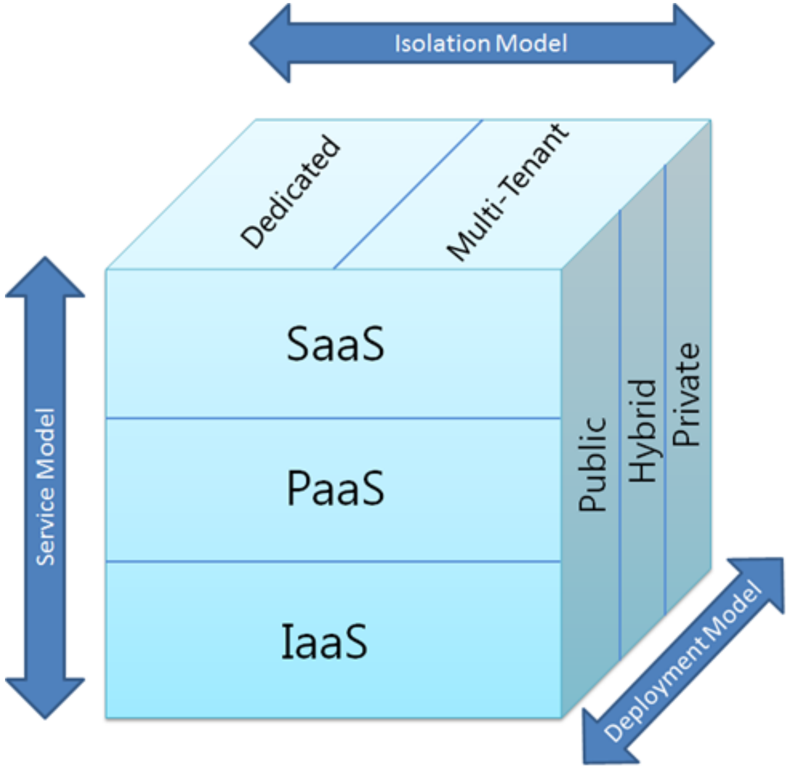




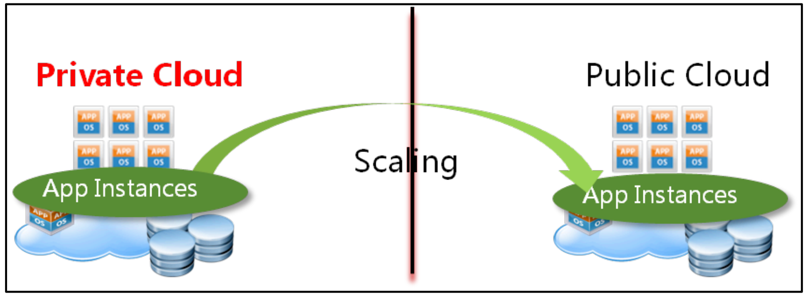
Source: https://blog.turbonomic.com/blog/on-turbonomic/to-scale-up-or-scale-out-that-is-the-question

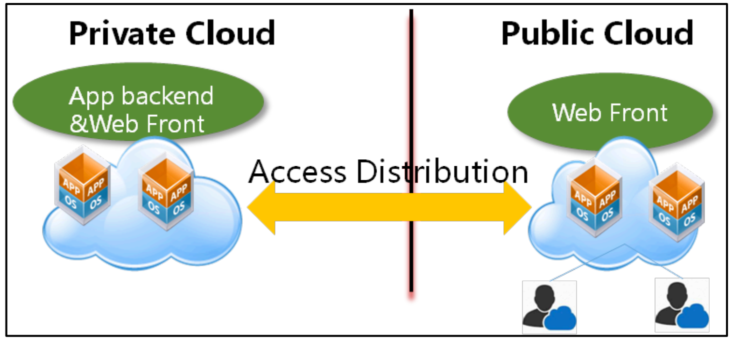
Public *or* Private? Public *and* Private?

- Hybrid Clouds is the *new normal*.
- But how can we select resources in private Clouds? Or in public Clouds? And when should we switch from one to the other?
- We need *federated Cloud resource orchestration solutions.*



Source: http://goo.gl/1jmkR





Source: https://wiki.openstack.org/wiki/Jacket

Software

Quality Assurance

- Which software metrics should we track?
- How do we *reliably measure success?*

Published: 07 March 2020

Software Quality Assurance in INDIGO-DataCloud Project: a Converging Evolution of Software Engineering Practices to Support European Research e-Infrastructures

Pablo Orviz Fernández, Mário David, Doina Cristina Duma, Elisabetta Ronchieri, Jorge Gomes & Davide Salomoni

Journal of Grid Computing **18**, 81–98(2020) | Cite this article

E. Ronchieri, M. Canaparo, M. Belgiovine and D. Salomoni, "Software Defect Prediction on Unlabelled Dataset with Machine Learning Techniques," *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, Manchester, United Kingdom, 2019, pp. 1-2, doi: 10.1109/NSS/MIC42101.2019.9059737.

Software Optimization

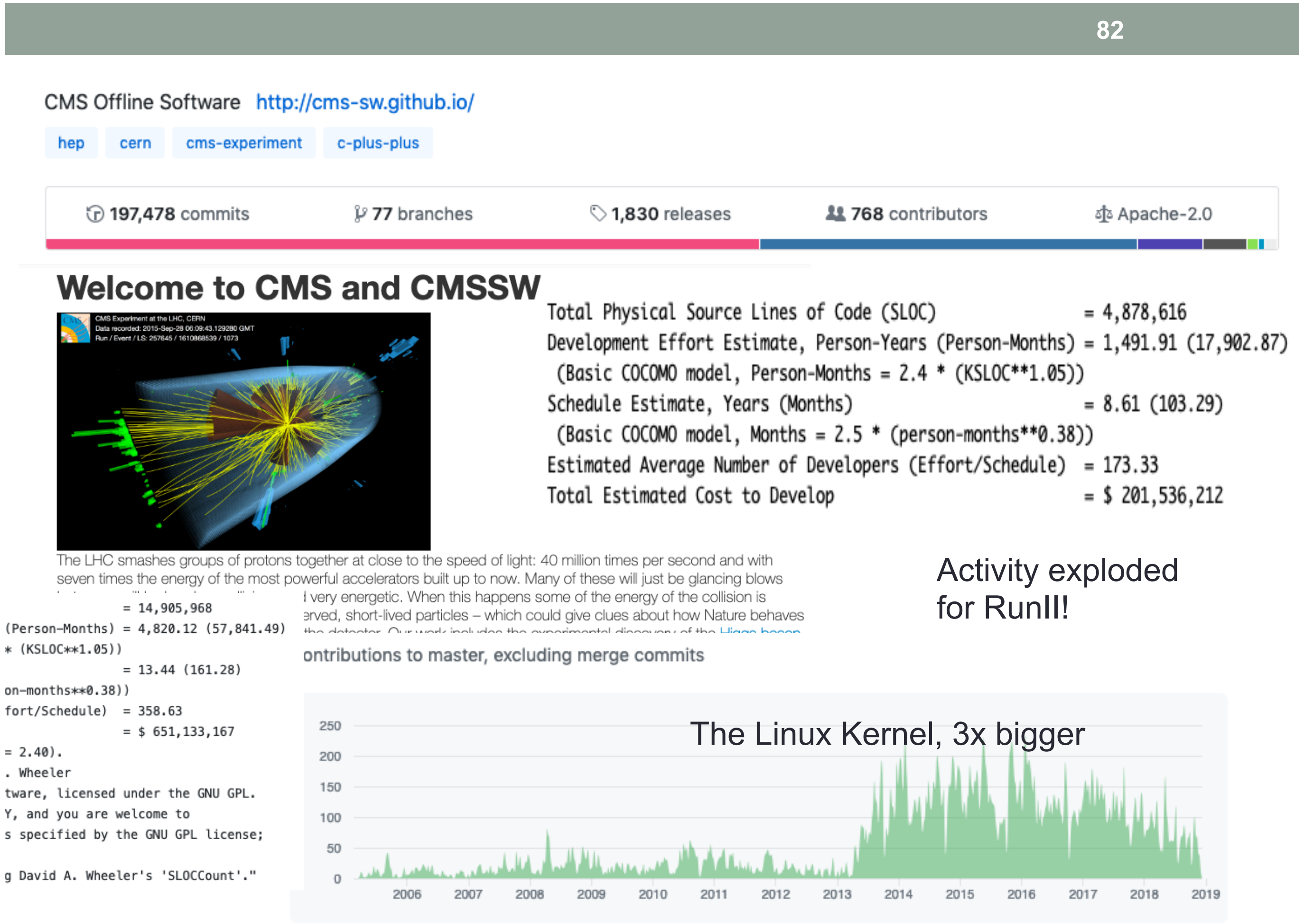
Case study: CMSSW

- Dev started early 2005
- Current size is 1120 packages, divided into 120 Subsystems

The Importance of Software and Computing to Particle Physics

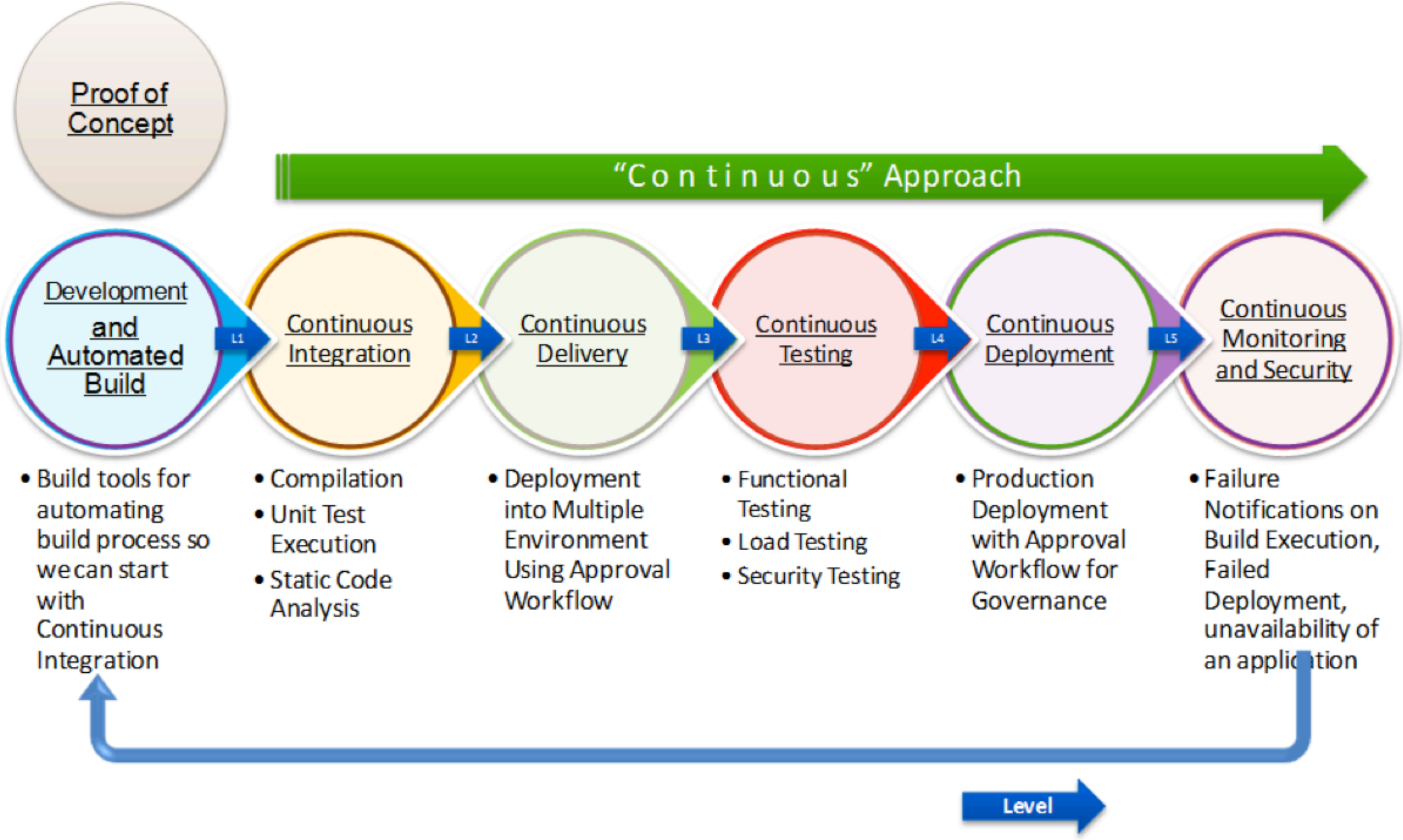
A contribution from the High-Energy Physics Software Foundation to the European Particle Physics Strategy Update 2018-2020

Current HEP software is the result of more than 20 years of development, and now must evolve to meet the challenges posed by new experimental programmes. In addition, the computing landscape is evolving rapidly and we need to exploit all the expertise available in our community, and in other scientific disciplines, in order to meet the technical challenges we are facing.
(HEP Software Foundation)

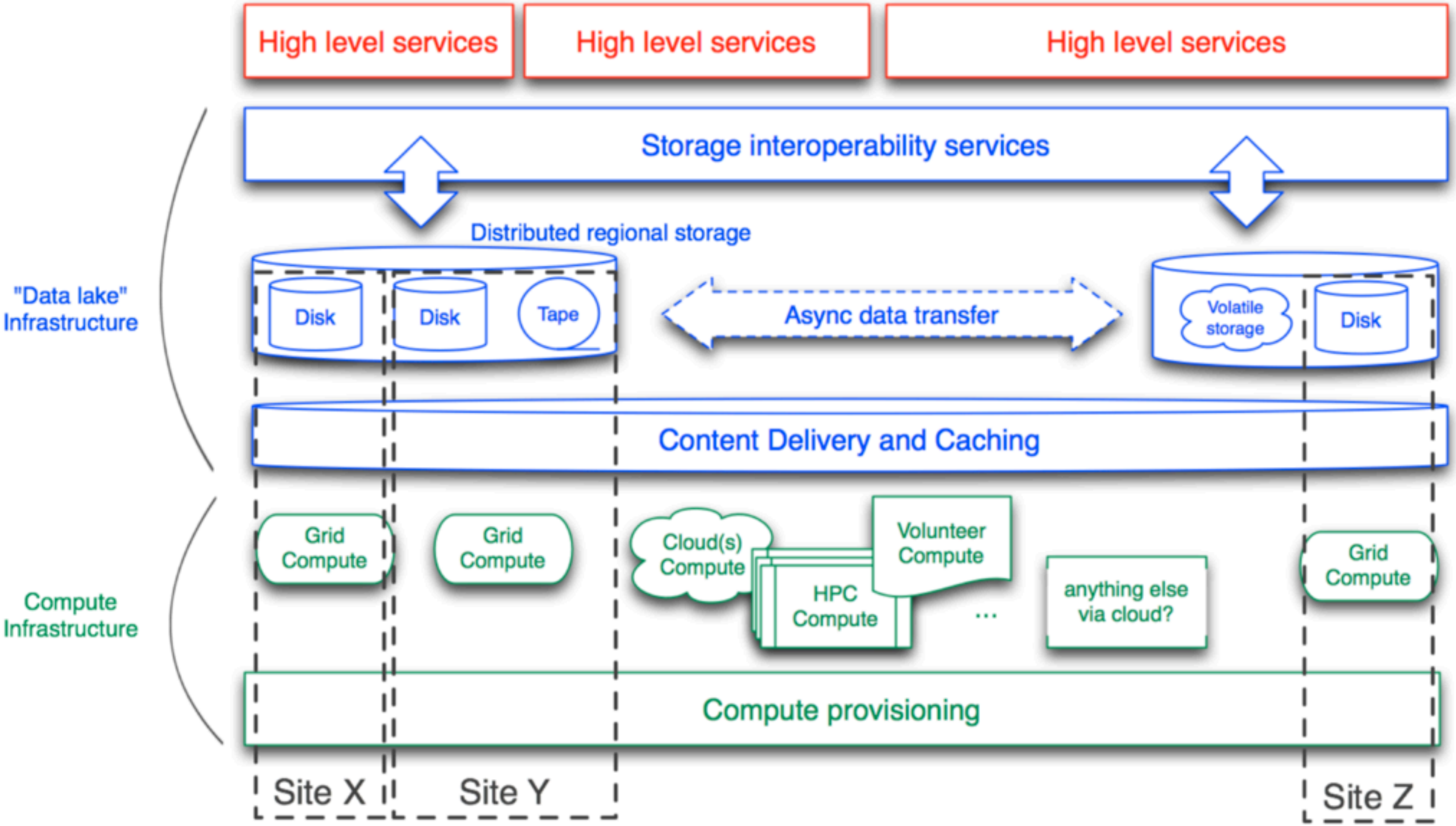


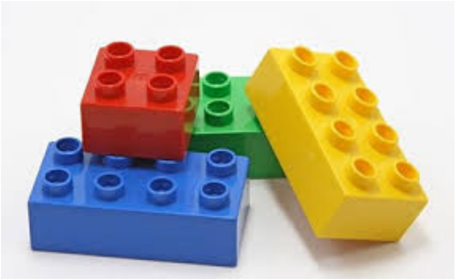
Software

Continuous *Everything*



Common, Composable Access Layers



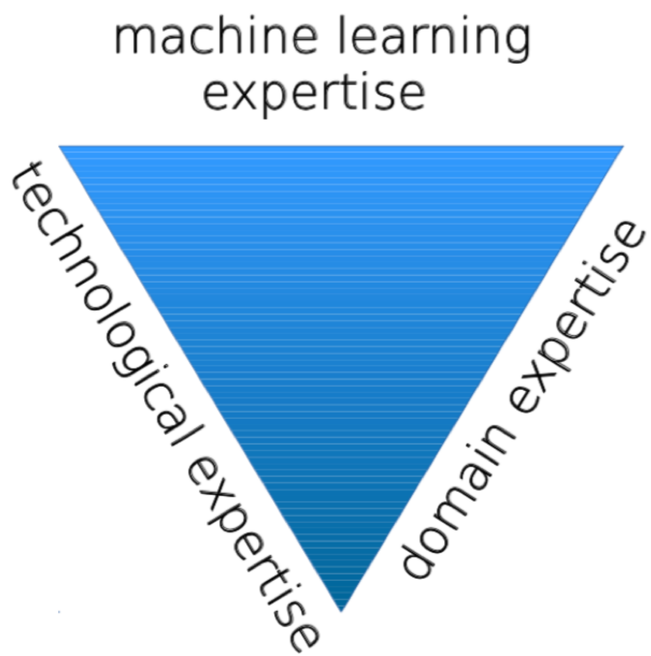


Data Management

- In a **naïve set of assumptions**, 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.

- Some key topics:
  - Data distribution policies
  - Replica management
  - Data management based on access patterns
  - Smart caches
  - Automated data pre-processing at ingestion time

On AI / Machine Learning, or “Categorizing the Know-How”



Thanks to Álvaro Lopez/CSIC

- **Category 1**: deploy an already trained ML model for somebody else to use on her own trained data set.
  - Domain knowledge
- **Category 2**: retrain (parts of) an already trained ML model to make use of its inherent knowledge and solve a new learning task.
  - Domain + ML knowledge
- **Category 3**: completely work through the ML / Deep Learning cycle with data selection, model architecture, training and testing.
  - Domain + ML + technological knowledge

Service Composition and *Infrastructure as Code*

- Create high-level **templates** of applications, services and infrastructures you want to run on *some Cloud*, using vendor-neutral languages such as TOSCA.
- Have some middleware service *automatically provision* the required resources and their relationships from the templates.
- Extend existing templates to create further services, applications or infrastructures and make the Cloud lower, infrastructural layers (IaaS) disappear from your view.
- Combine with the *Continuous Everything* paradigm.

Serverless Computing

- With **serverless**, a Cloud provider is responsible for executing a piece of code, written by yourself, by **dynamically allocating the resources needed by the code**.
- You are only charged for the resources used to run your code and only when the code runs. Your code is typically structured around functions. Thus, serverless computing is also called **Functions as a Service, or FaaS**.
  - There are semantic differences between "Serverless" and "FaaS", but we need not bother about them here. The key point is, with FaaS we focus more on the application level, and less on the infrastructural one.
- Read Hidden Technical Debt in Machine Learning Systems (2015).

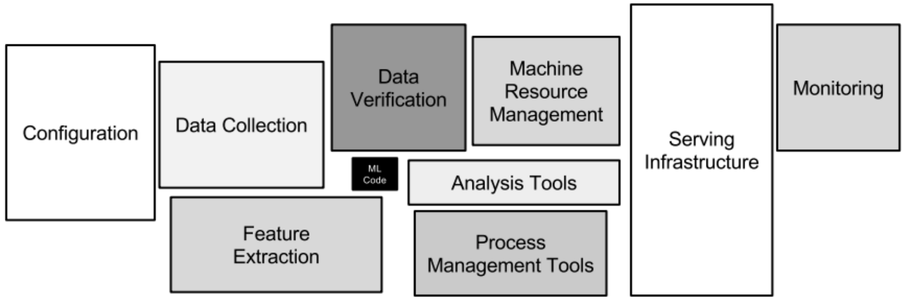


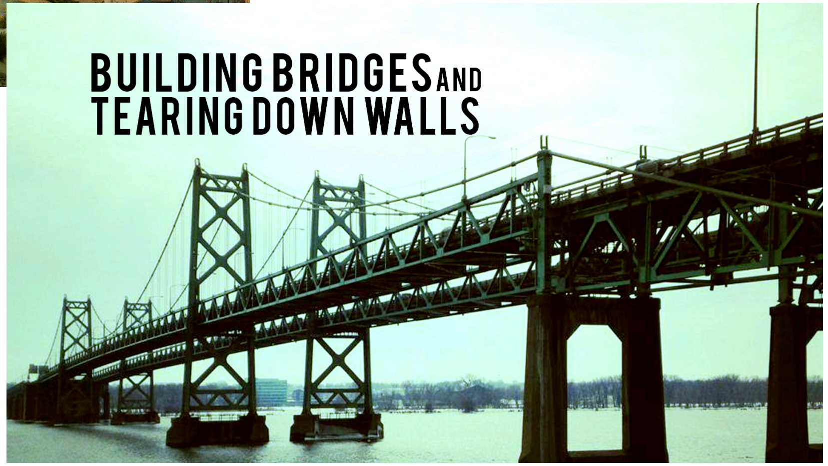
Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

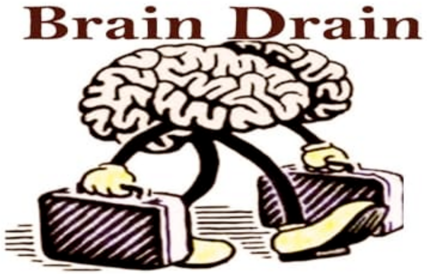
Cost Control

- How do we manage Cloud spending?
  Remember the mantra of our management: *flat budget*.
- Accounting for cost management & fine-grained billing dashboards spanning public *and* private infrastructures.
- Smart algorithms for automation in resource scheduling policies.
- Spot markets.

Know-how

- Many facets here:
  - Horizontal development vs. Vertical development.
  - How often are we told about IT constraints and technologies in our education paths?
  - Even then, disconnection between formal education and actual needs (often in both the public and private sectors).
  - Need to properly communicate in multiple languages with multiple communities: not easy!
  - Lack of recognition for technology-oriented paths in academic careers.





No lock-in, Yes look-in

- Vendor lock-ins are easy to understand and are typically a **no-go** for us (international experiments, multiple funding agencies, heterogeneous resources, long time spans, etc.)
- But so are community-specific lock-ins: use de jure *and* de facto standards, use open modular architectures!

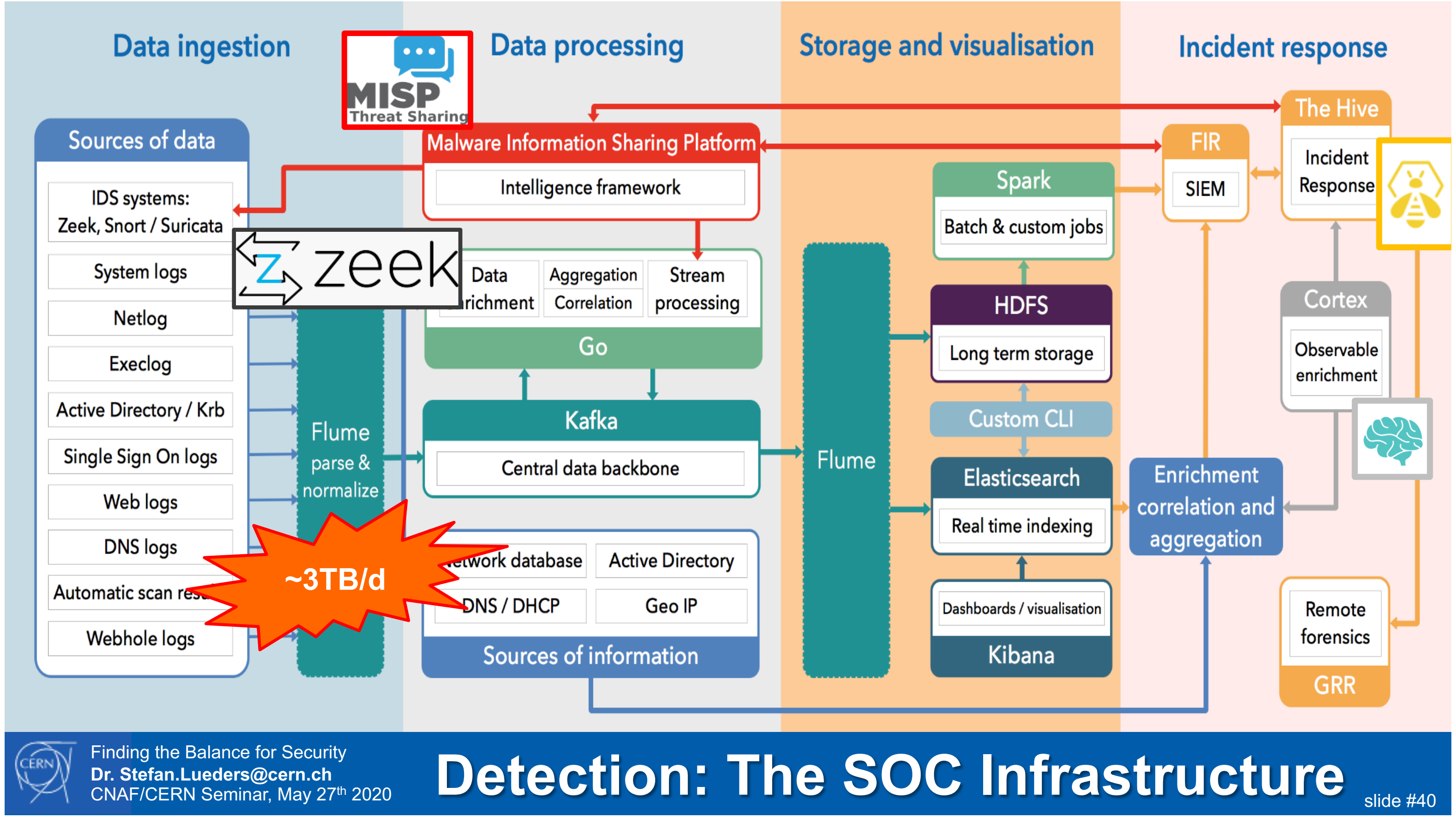
open source initiative®

Technology Transfer

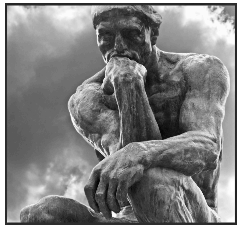
- That is, develop basic research into new technologies, products and companies, for public or private services.
- This is often a significant challenge for us, but it is also often connected to the real impact of what we do.
- Let's remember that even when we are tempted to think that *we* run the Technology Transfer show, it is normally a **two-way process**: we learn a lot from TT, and this will return back to us as future ideas, concepts and solutions.
- Last but not least, society-wise TT is typically a strong catalyst for economic development.

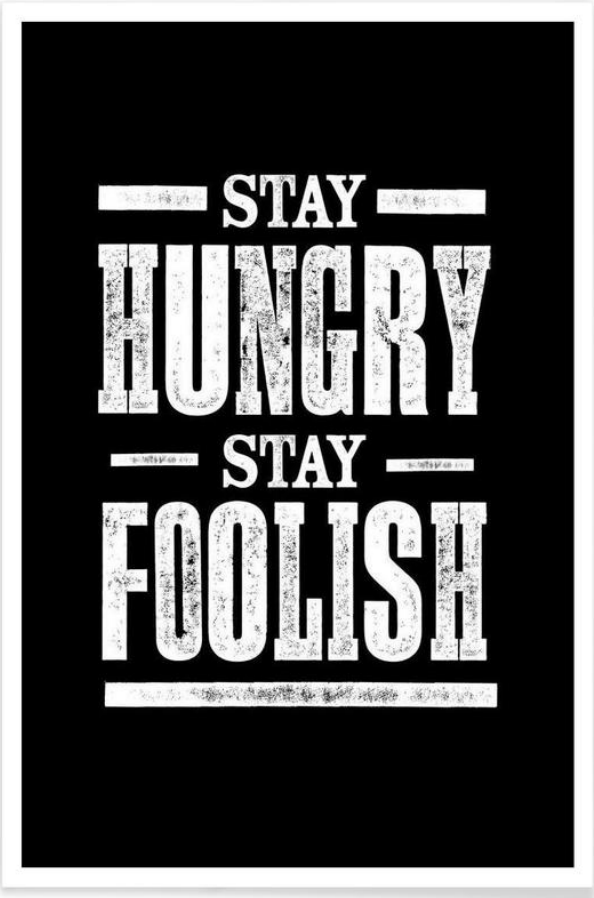
Privacy & Security

- We are open science, so why worry about security and privacy?
- It's all in "*Finding the Balance between Academic Freedom, Operations & Security*" (Stefan Lüders/CERN)



Conclusions





- We have a sizeable number of computing-related questions for the next decades of physics experiments, touching multiple fields, which still need full answers.
- Our needs are increasing much more than what is economically viable.
- The **take-away message** is that we should see these as opportunities for ourselves, science and society, rather than problems.
- A few things that I have not even mentioned, but which are *potentially* going to be very important:
  - Common e-infrastructures (such as the European Open Science Cloud, or EOSC), FAIR data, Quantum Computing, Opportunistic resources, Edge computing.

Thanks!

- Q&A now, or you can also contact me after the lecture at davide@infn.it
- Acknowledgments: Tommaso Boccali/INFN.

Stat rosa pristina nomine,
nomina nuda tenemus.
(Umberto Eco)