


KubeCon | CloudNativeCon

Europe 2020

Virtual

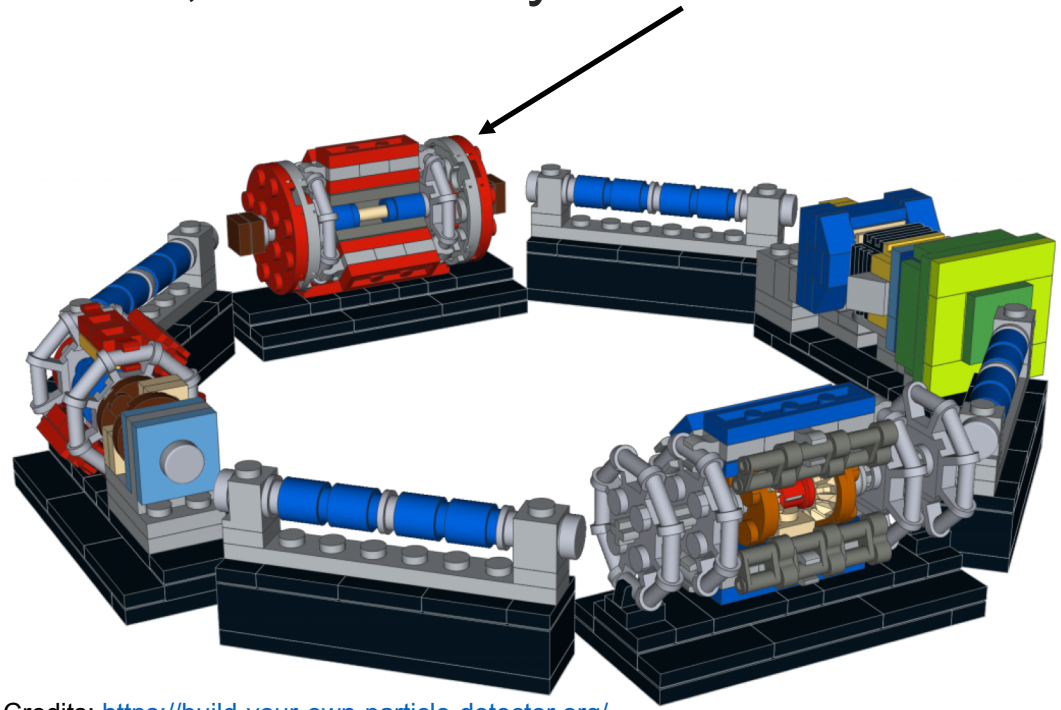
Hunting for New Particles Leveraging Legacy Infrastructure with Kubernetes

Clemens Lange (CERN)

Hello :-)

I’m a particle physicist working on the CMS experiment at the Large Hadron Collider (LHC) at CERN, Switzerland

Analysing the particle collisions provided by the LHC, recorded by the CMS detector



Credits: https://build-your-own-particle-detector.org/

High-Energy Physics (HEP)

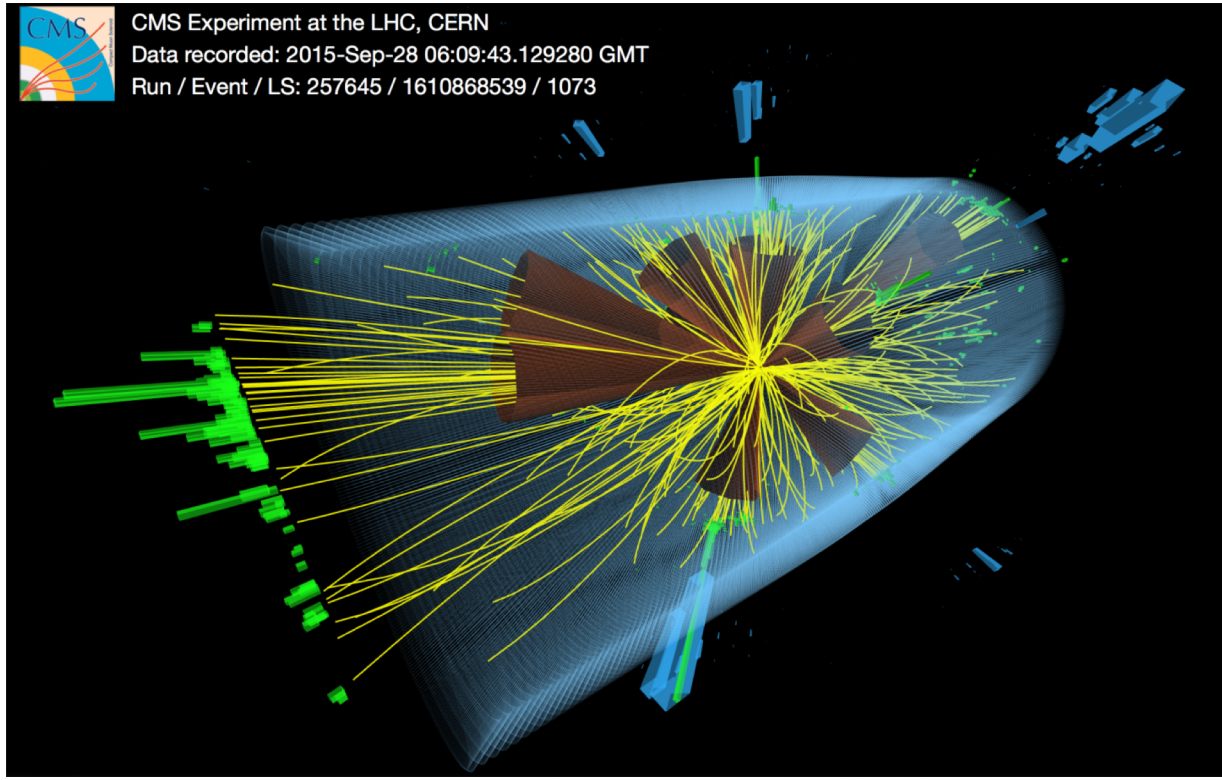
Trying to understand the smallest building blocks of matter

Particle detectors such as the CMS detector take up to 40,000,000 “photos” of the LHC collisions per second, 24/7 (almost) all year long

Up to 1000 of such photos (→events) per second stored for later analysis

Analysing tera- to petabytes of data using C++, Python, shell scripts
→ “big data” analysis

CMS facts
Weight: 14,000 tons
Height × Length: 15m × 21m
Around 100 million channels (megapixels)



Credits: https://cms.cern/

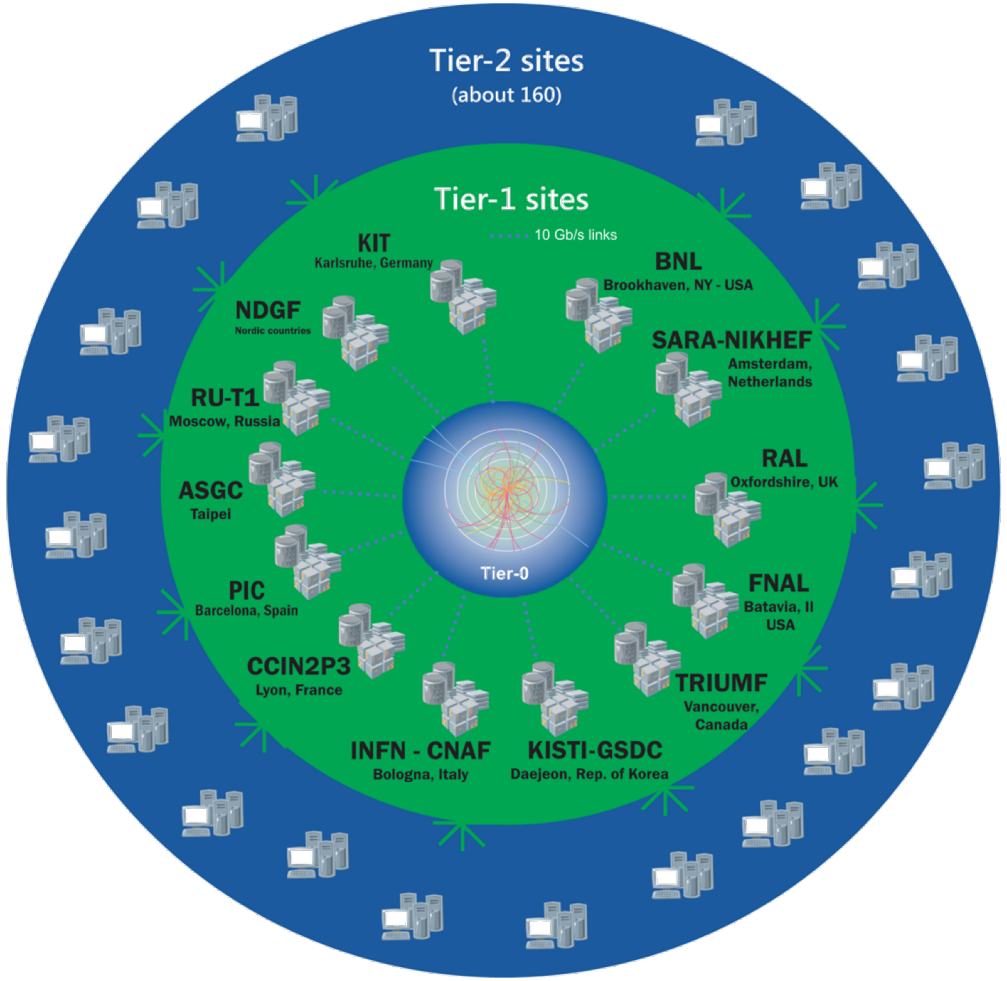
Infrastructure

The Worldwide LHC Computing Grid (around 170 computer centres)

Lots of smaller local batch farms (Tier-3)

Sites often already managed using Kubernetes

At CERN: 230k cores in HTCondor batch farm:
150k jobs running simultaneously (peak)
1.4 million jobs completed/day (peak)



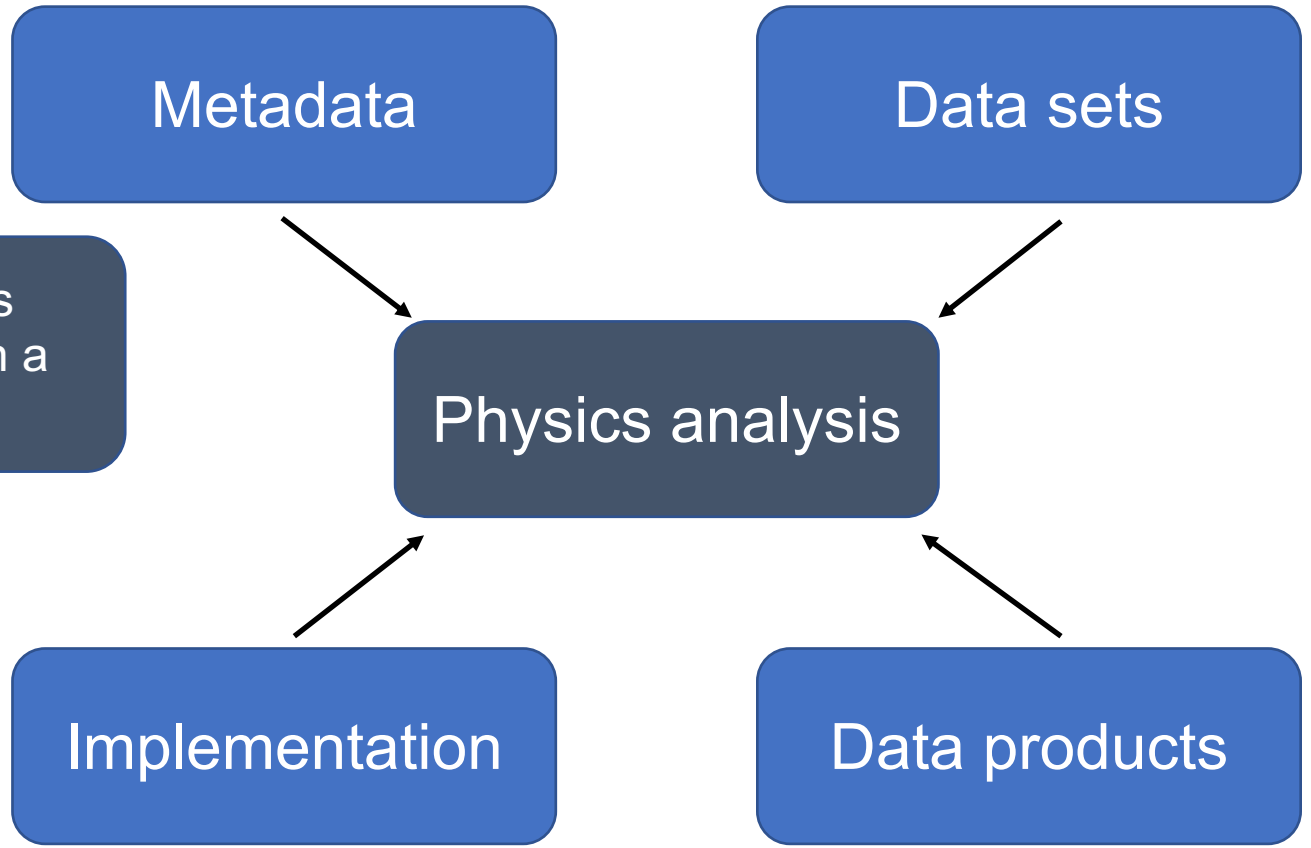
Credits: https://wlcg-public.web.cern.ch/

HEP workflows

Largest part of typical high-energy physics workflows is automated already

Challenge: implement/use automated workflows for high-level physics analyses

These address a physics question, often resulting in a paper publication



Software version control system and images are essential tools in this context

git + docker

Lots of inputs to be handled/keep track of!

Workflow “languages”

1. Capture software

Individual analysis stages in an executable way (including all dependencies)

2. Capture commands

How to run the captured software

3. Capture workflow

How to connect the individual analysis steps

Several tools under investigation and used by smaller groups

reana COMMON WORKFLOW LANGUAGE

REANA CWL implementation (HEP-focussed)

Luigi

Luigi

Yadage (HEP-specific)

Can we do it cloud-native?

Argo Workflows

My Kubernetes cluster

openstack

Provisioned with OpenStack with
CERN plugins/customisations

CernVM File system

EOS

4 nodes à
4 cores w/ 8 GB RAM

300 GB S3 object storage
300 GB CephFS block storage

kubernetes v1.18.2

Managed via GitOps
using Argo CD

argo

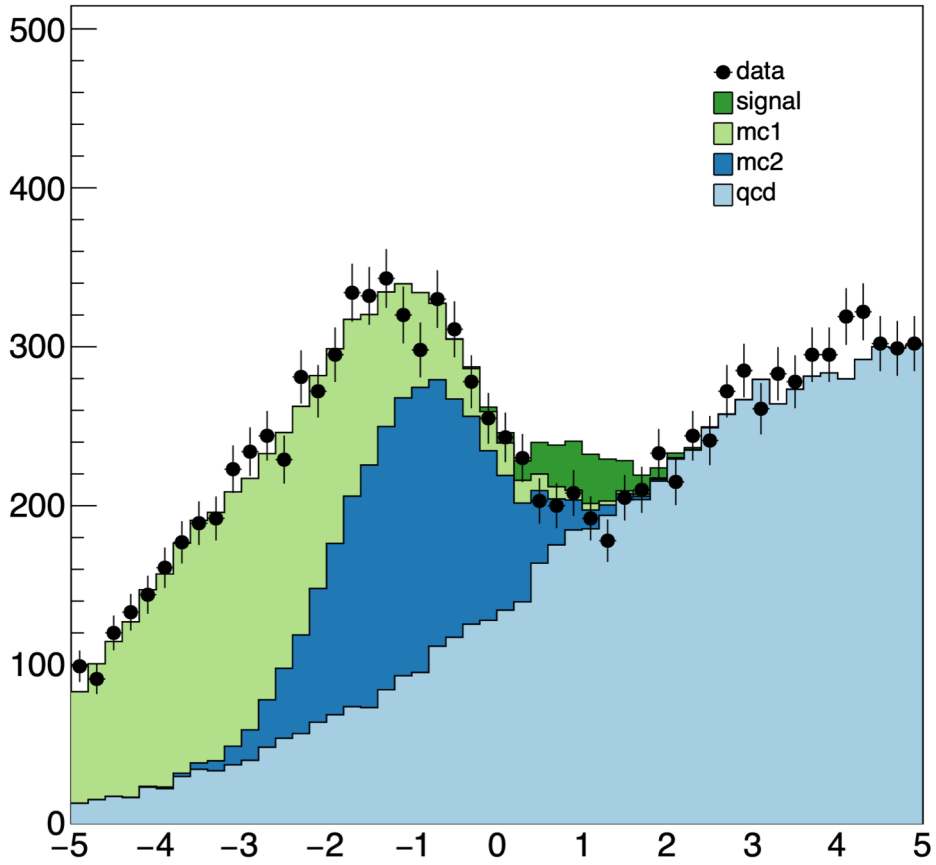
Secrets encrypted using SOPS
w/ Barbican modification
Deployed using KSOPS plugin
with Argo CD

New Physics search with Argo

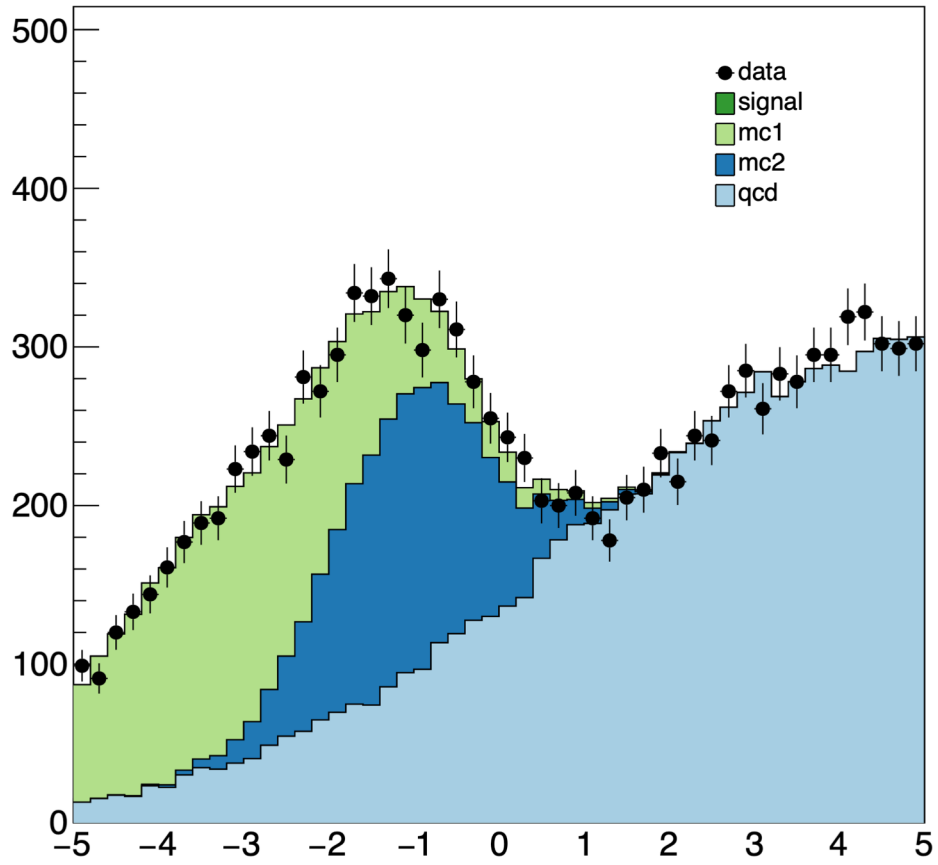
Let’s run an example workflow, searching for a new signal in the data!

Workflow results

Prefit: mind what signal would look like

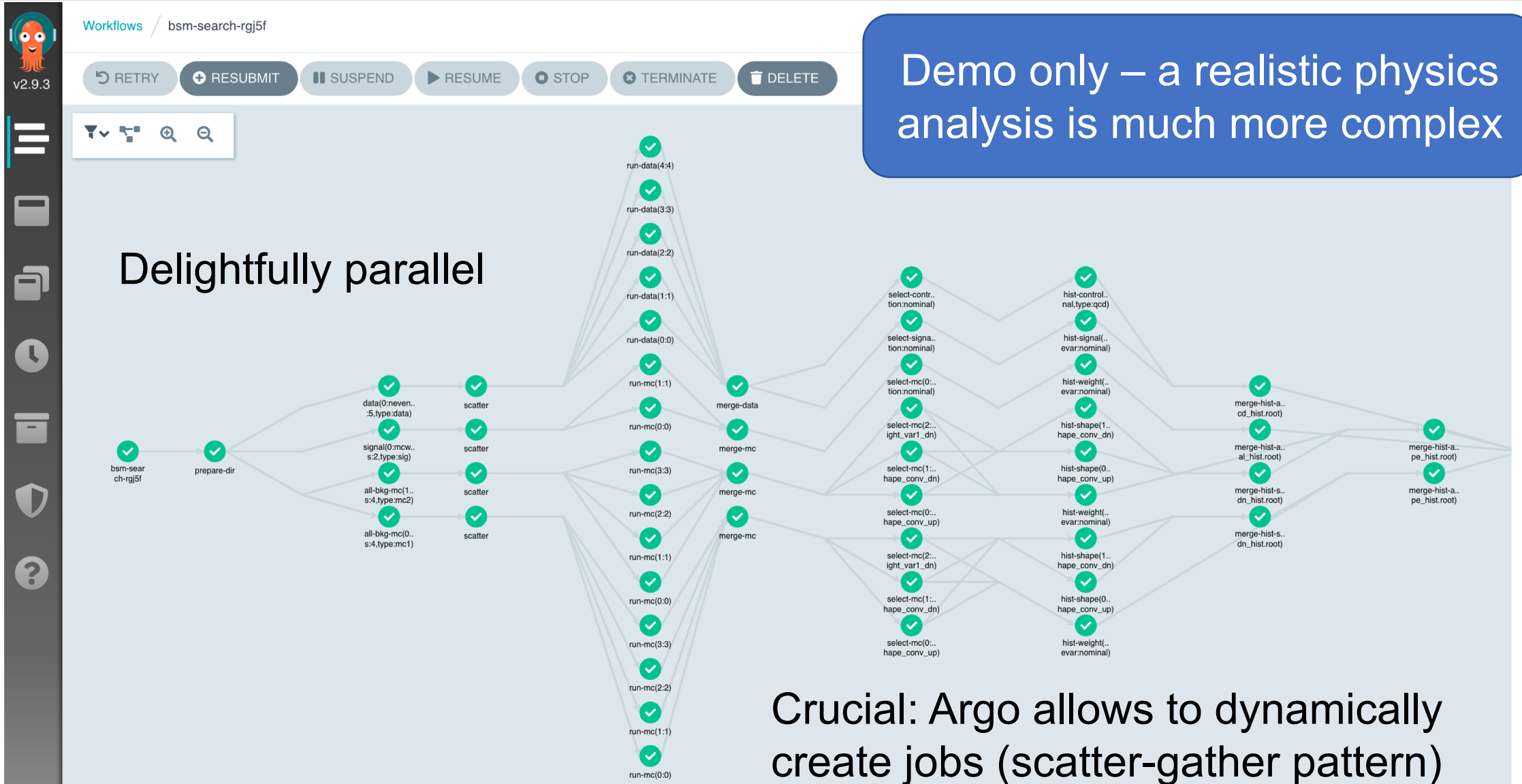


Postfit: the data do not support the presence of this signal



No new physics discovered this time!

Example workflow



Crucial: Argo allows to dynamically create jobs (scatter-gather pattern)

Based on https://github.com/reanahub/reana-demo-bsm-search

Life's unfair

A realistic physics analysis workflow cannot be run on my cluster

Relative size not even to scale…

Can I use your cores?

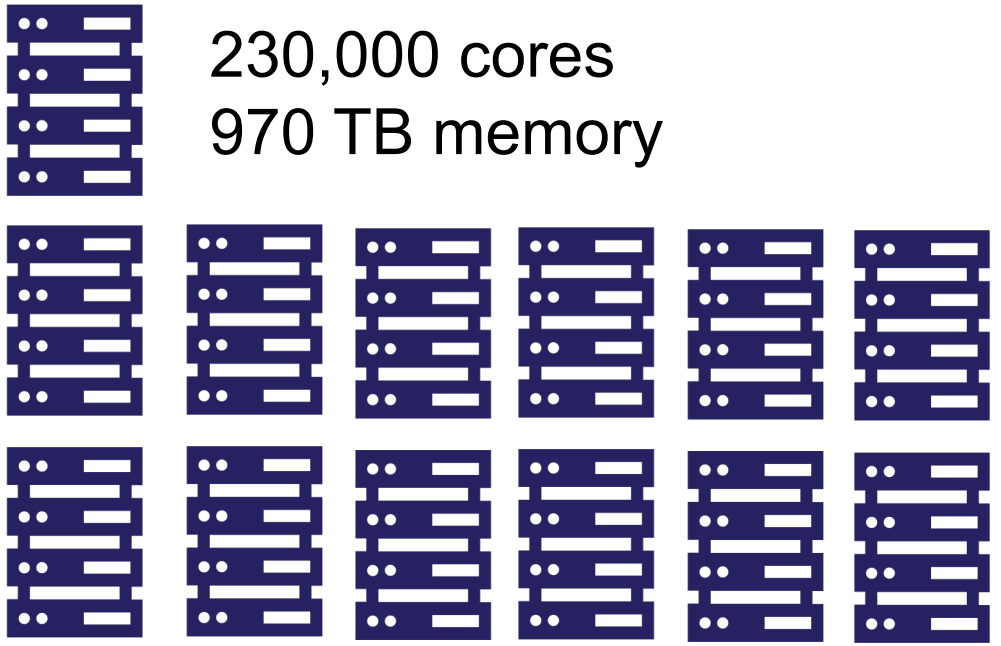
My cluster

kubernetes

16 cores

The CERN batch cluster

HTCondor
High Throughput Computing



230,000 cores
970 TB memory

HTCondor operator

Idea: introduce HTCJob Custom Resource Definition

```yaml
apiVersion:
apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: htcjobs.htc.cern.ch
spec:
  group: htc.cern.ch
  names:
    kind: HTCJob
    listKind: HTCJobList
    plural: htcjobs
    singular: htcjob
  scope: Namespaced
```

Mimic Kubernetes Jobs (reflect status Running/Failed/Succeeded)

```yaml
status:
  properties:
    active:
      type: integer
    failed:
      type: integer
    succeeded:
      type: integer
    clusterID:
      type: string
    jobIDs:
      items:
        type: string
      type: array
    uniqID:
      type: integer
```

Work by **Tadas Bareikis** (Bioinformatics student at Vilnius University) and me

Implementation

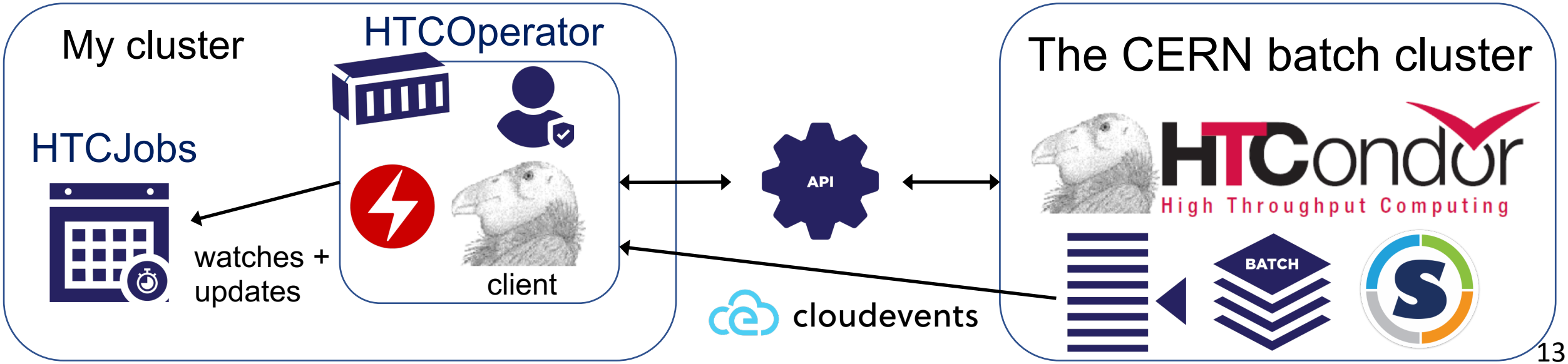
Operator SDK makes it easy to get a Kubernetes operator implemented und running

OPERATOR SDK

Built Docker container that contains HTCondor client – also knows about authentication via Kerberos (using secrets)

HTCondor Operator installed into this container – also translates image/job spec into singularity exec script

Individual jobs can additionally notify operator via CloudEvents



Argo + HTCJobs

Argo can manage any kind of Kubernetes resources:


```
- name: generate-batch
  inputs:
    parameters:
    - name: type
    - name: nevents
    - name: njobs
  resource:
    action: create
    successCondition: status.succeeded == {{inputs.parameters.njobs}}
    failureCondition: status.failed > 0
    manifest: |
      apiVersion: htc.cern.ch/v1alpha1
      kind: HTCJob
```

→ Can move the long-running steps to HTCondor!

Let’s run the same workflow as before, moving the first step to HTCondor

Closing remarks

Managed to leverage “legacy” infrastructure by means of a Kubernetes CustomResourceDefinition combined with an Operator

OPERATOR SDK

“Operator” concept extremely powerful for this purpose

Cloud-native high-energy physics workflows possible using Argo

argo

Next steps:

- Make HTCOperator more flexible
- Apply same concept to also use the grid (WLCG) ➜ see also presentation by Alessandra Forti and Lukas Heinrich

Thanks to Tadas Bareikis (Vilnius University), the Cloud Containers team at CERN (Thomas Hartland, Ricardo Rocha, Spyros Trigazis et al.), and Lukas Heinrich

CERN


KubeCon
CloudNativeCon
Europe 2020
Virtual
HELM
KEEP CLOUD NATIVE
CONNECTED