Large-scale HPC deployment of Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference (SCAILFIN)

Mike Hildreth, representing the SCAILFIN team

Mike Hildreth: mhildret@nd.edu; ND Developers Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

SCAILFIN

Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free Inference

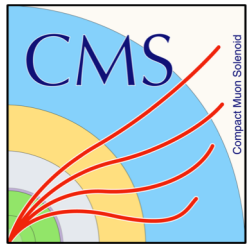
Mike Hildreth, representing the SCAILFIN team

Mike Hildreth: mhildret@nd.edu; ND Developers Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

Motivation

Major Multi-User Research Facilities involve the comparison of data collected from experiments with “synthetic” data, produced from computationally-intensive simulations.

Comparisons of experimental data and predictions from simulations are abstractions of the specific data analysis techniques developed by the respective communities over several decades. E.g.:



Motivation

Many of these data analysis tasks are often conducted manually or through *ad hoc scripts* that might not be well maintained, making reproducibility and reusability difficult. Many of these tasks do have a well-defined workflow that make automation possible, though.

REANA was created (in collaboration with DASPOS, DIANA and CERN) to address the reproducibility and reusability of the analysis pipeline.

Reproducible research data analysis platform

Motivation

In parallel:

Interest in leveraging Machine Learning (ML) and Artificial Intelligence (AI) techniques, to enhance the analysis of data from these facilities.

In particular, its application with emergent **Likelihood-Free Inference (LFI) techniques** when the predictions for the data are implicitly defined by the simulation, often leading to an intractable likelihood function. This can apply to analysis of data from LHC, LIGO, etc, but such Likelihood-Free algorithms have so far been **implemented mostly on individual machines and in ad hoc scripts because the training workflows are very complicated.**

Introduction

SCAILFIN: Scalable CyberInfrastructure for Artificial Intelligence and Likelihood Free INference

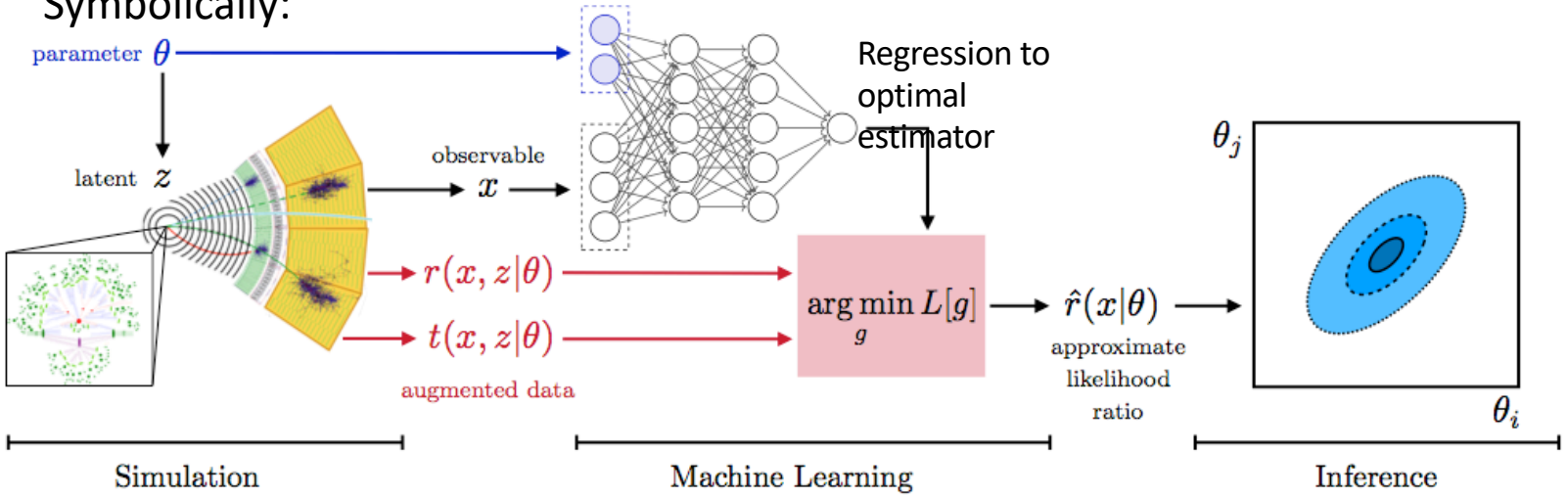
The SCAILFIN project aims to deploy artificial intelligence and likelihood-free inference techniques and software using scalable cyberinfrastructure (CI) that is developed to be integrated into existing CI elements, such as the **REANA** system, to work on HPC facilities.

PI's: Mark Neubauer, Dan Katz Kyle Cranmer, Heiko Mueller Mike Hildreth

Simulation-Based Likelihood Free Inference

J. Brehmer, K. Cranmer, G. Louppe, and J. Pavez. arXiv:1805.00020, 2018.

Symbolically:



Estimation of optimal estimator lends itself to ML methods:

- Training data derived from simulations
- Can be guided by optimal sampling based on phase space density of generator, sensitivity to physics under study

Today's Topic

SCAILFIN goals

- ...
- **extend the REANA** platform to allow remote **submission** of workflows to **HPC facilities.**

Components:

- **REANA** as the Cyber Infrastructure element to deploy AI and Likelihood-Free inference techniques.
- **VC3** (Virtual Clusters for Community Computation) in order to scale REANA to HPC resources.

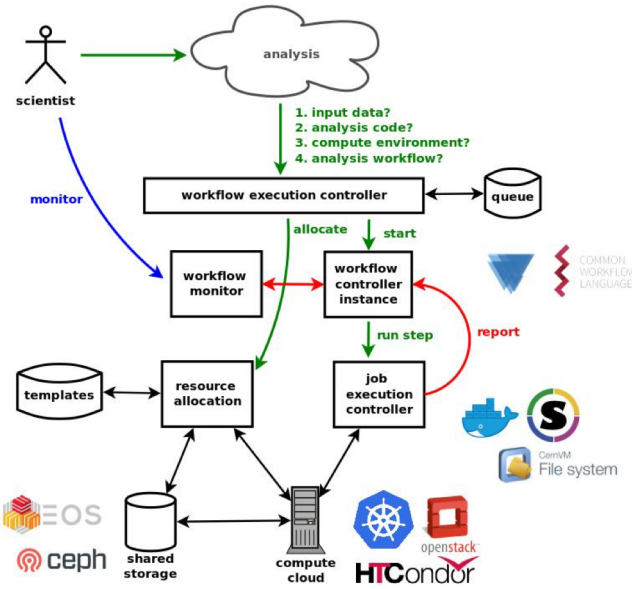
First, a brief overview of these 2 components...

reana: Reproducible Research Data Analysis Platform

Features

- Allows creation of tightly defined, container encapsulated workflows
- Built with commodity pieces
- Purpose is to allow complete reproducibility
- Sharing workflows is as easy as sharing a specification
  - (and inputs!)
- Different workflow engines supported. e.g.:
  - CWL (Common Workflow Language) : www.commonwl.org
  - Yadage (YAML based adage): yadage.readthedocs.io
  - Next: PARSL: parsl-project.org

Architecture

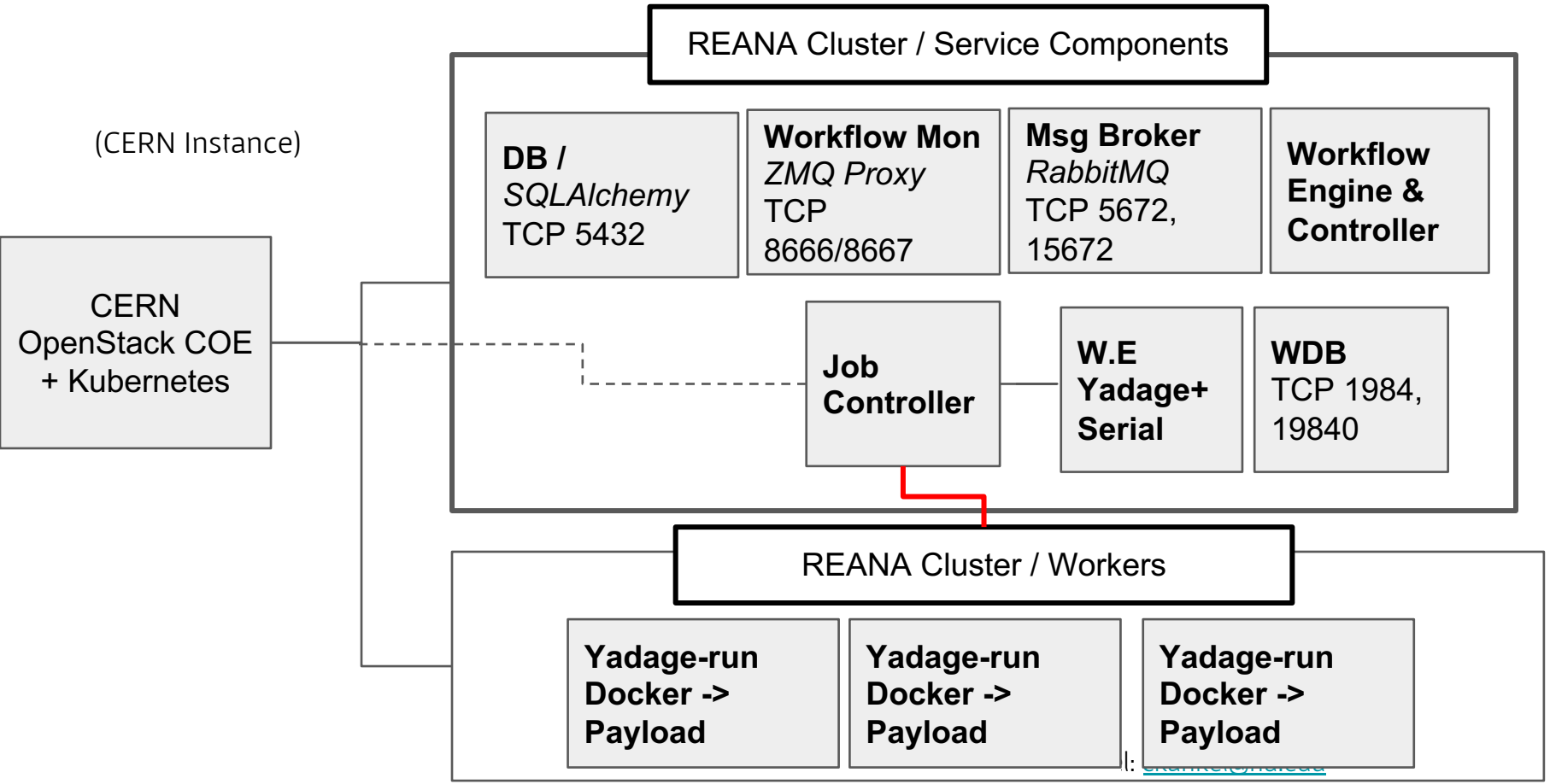


reana: Reproducible Research Data Analysis Platform

Components

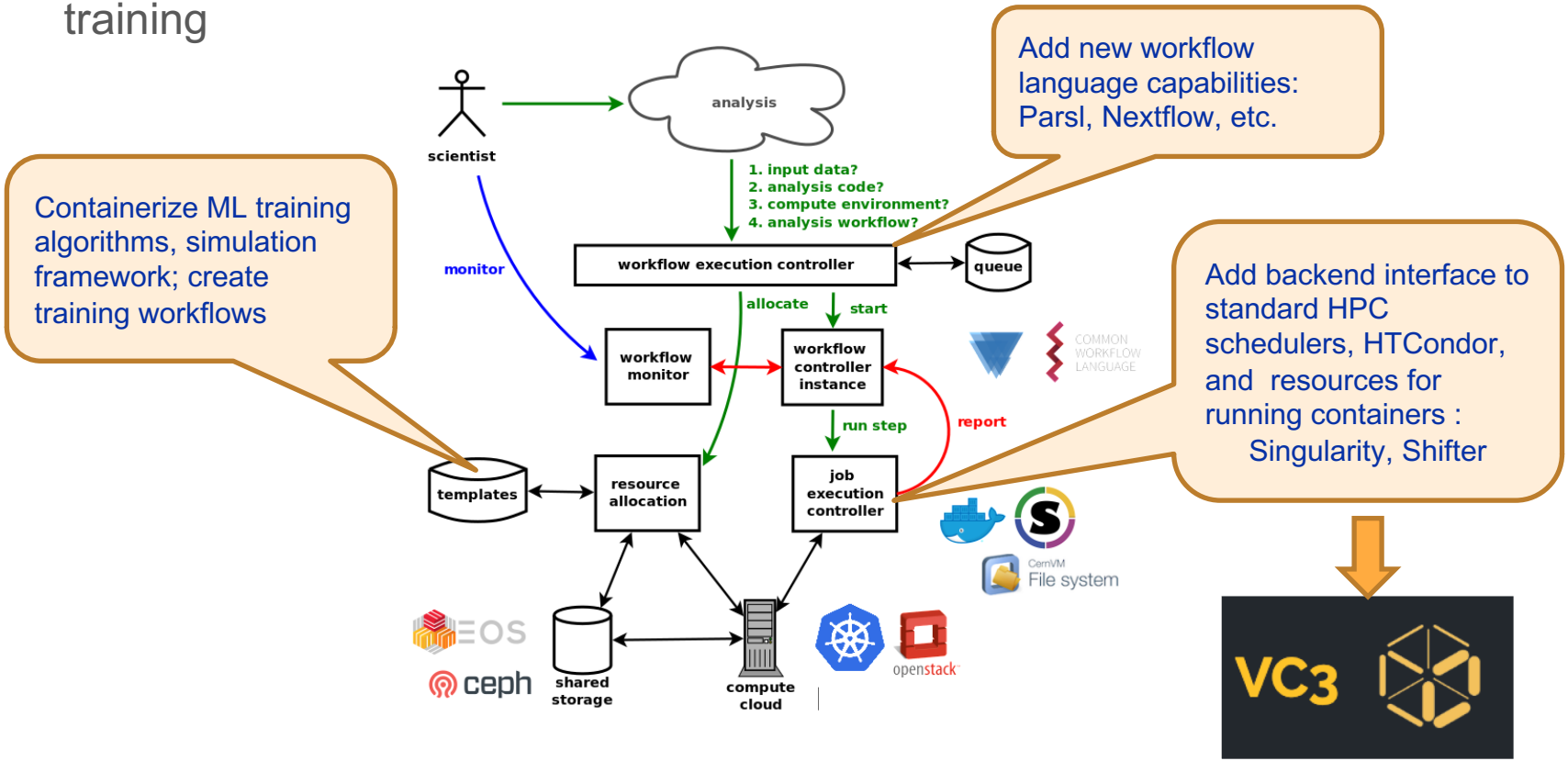
- Two major components each consisting of many sub-components
  - reana-client: User facing component.
    - Accepts workflows and and is used as interface to entire REANA system (for user).
  - reana-cluster: Workhorse.
    - Consists of many small pieces which handle workflows, dish out jobs, coordinates results, can be thought of as the job scheduler. Jobs are scheduled via Kubernetes.

reana-cluster - Simplified Diagram

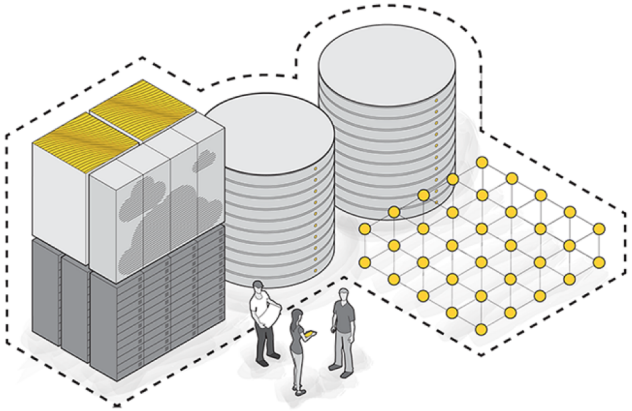


SCAILFIN: REANA Upgrade

- Extend capability of REANA framework to make it more suitable for ML training



VC3: Virtual Clusters for Community Computation



VC3: A platform for provisioning cluster frameworks over heterogeneous resources for collaborative science

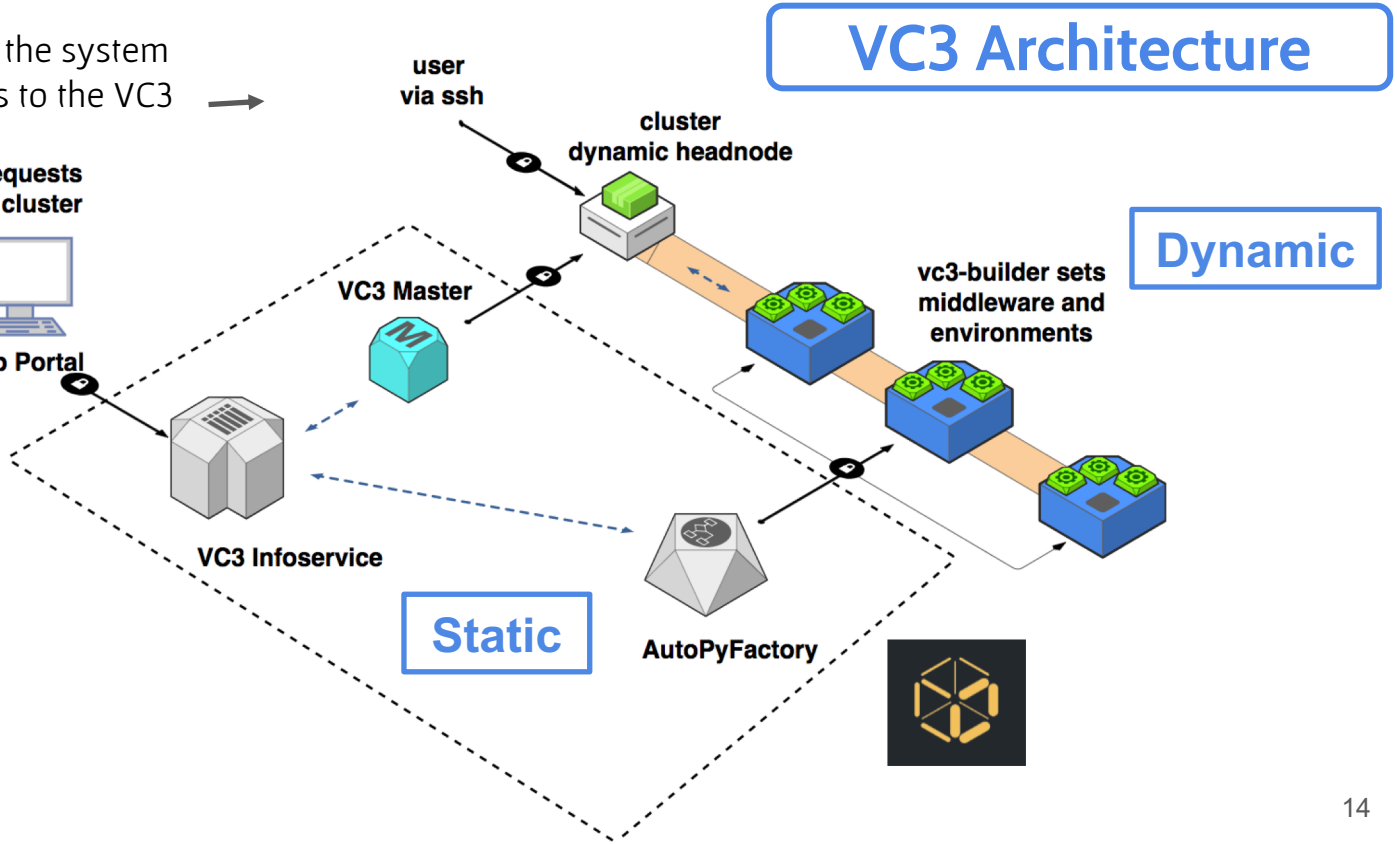
- Overlays “cluster” environment on top of diverse resource allocations
- Similar to cloud services that allow you to stand up clusters, but on “your” resources, and in “user space”: no root access needed

https://www.virtualclusters.org

VC3

VC3: Virtual Clusters for Community Computation

User adds an SSH public key to the system that will be used to grant access to the VC3 dynamic headnode



- User defines an allocation
- Selects middleware configuration
- VC3 infrastructure creates VC3 headnode and configures resources
- Workers on compute nodes communicate with VC3 headnode to receive compute workloads

VC3: Virtual Clusters for Community Computation

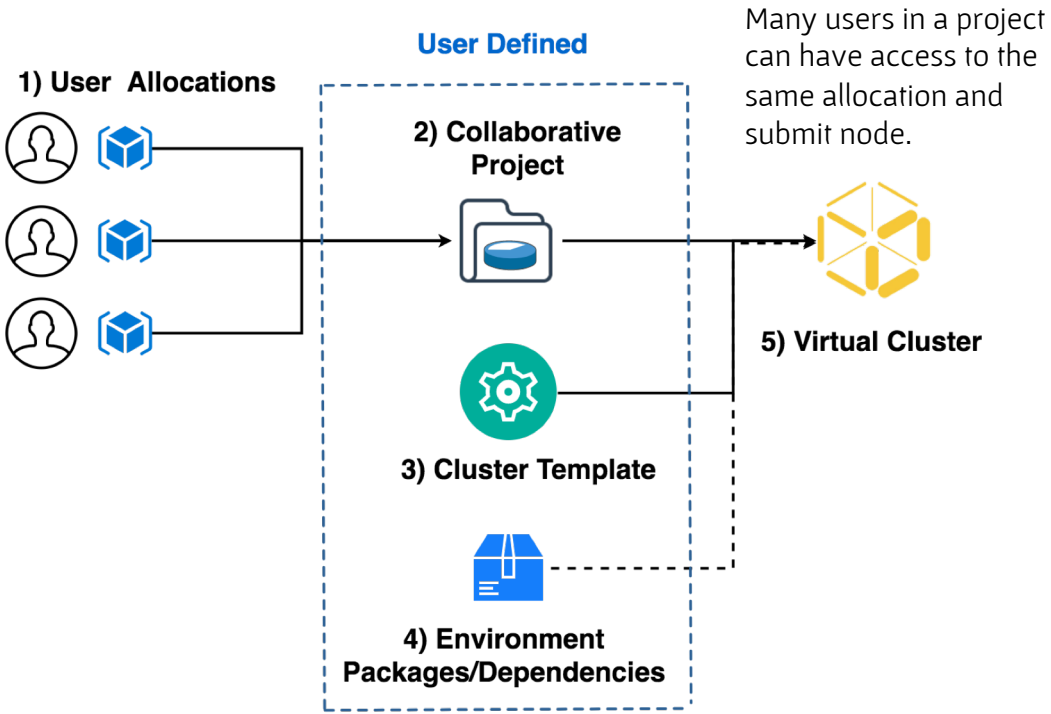
Features

- The user can select his/her own middleware for submission (E.g.: HTCondor, WorkQueue, Spark, REANA+HTCondor).
- It doesn't matter what the resource target batch system is (as long as it is supported by the translation layer for submission).

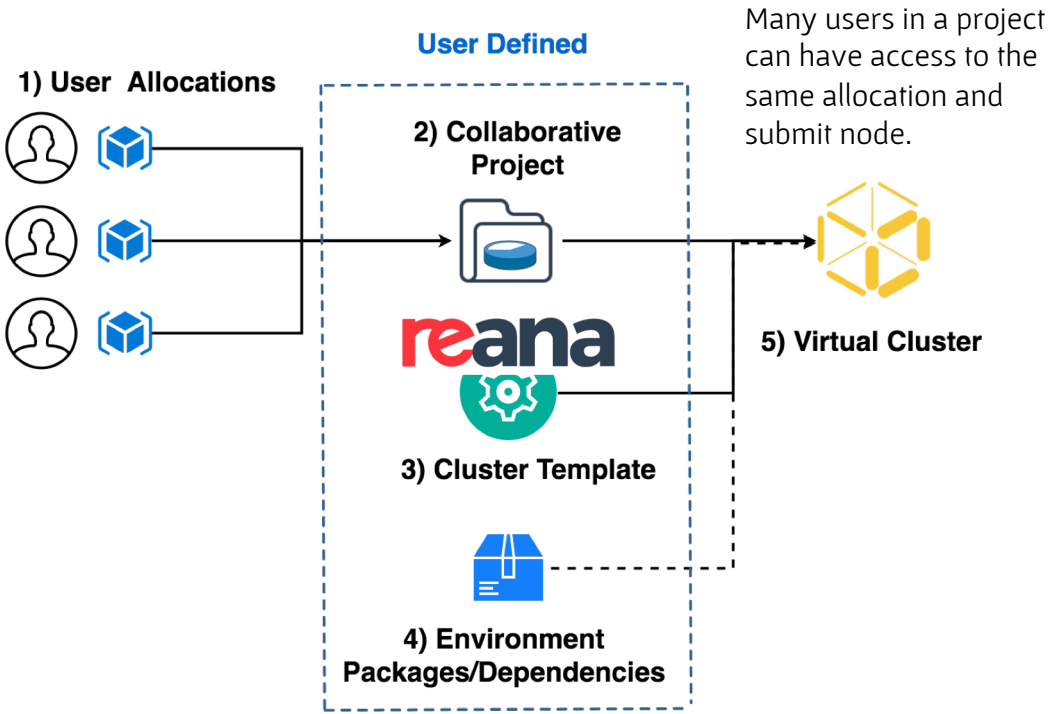
  E.g.: Torque (Blue Waters), SLURM (NERSC, PSC–Bridges, Stampede2), HTCondor, LSF, SGE, PBS.

VC3 already has most of the infrastructure needed to run REANA workflows at scale on HPC systems

Implementation: A REANA Cluster Template for VC3



Implementation: A REANA Cluster Template for VC3



Using REANA + VC3 on Blue Waters*

Blue Waters cluster:

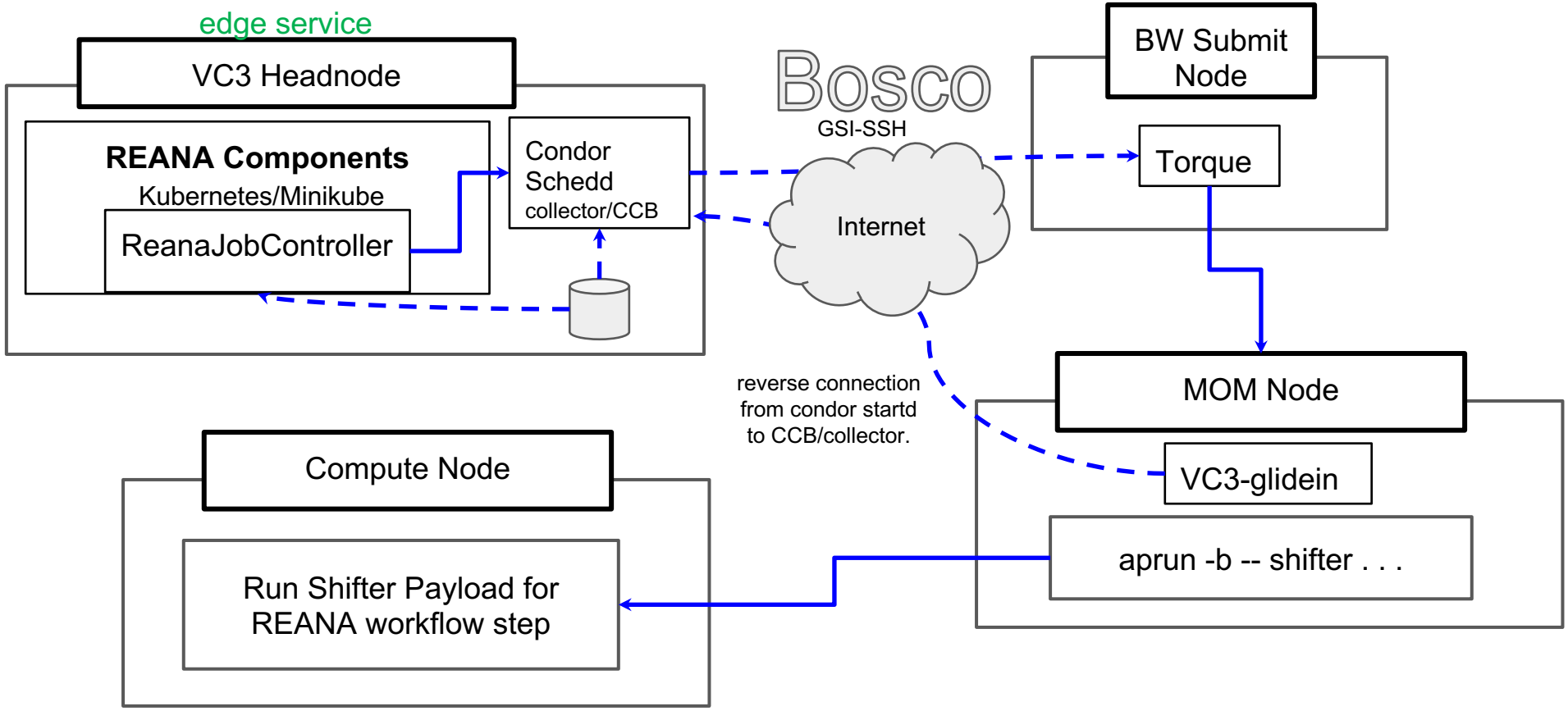
- Batch system: Torque
- Container technology: Shifter
- Authentication mechanisms:
  - Multi-factor authentication (Password + RSA token)
  - GSI-SSH tokens

Virtual cluster created on top of Blue Waters:

- VC3 Submit node with kubernetes (via minikube) and a REANA cluster deployed on the fly.
- HTCondor as the middleware
- VC3 authenticates with Blue Waters via GSI-SSH

*Note: Infrastructure worked out of the box on other resources such as the ND HPC Cluster and XSEDE/Pittsburgh

SCAILFIN on Blue Waters



Mike Hildreth: mhildret@nd.edu; ND Developers Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

SCAILFIN Developments to make this work:

REANA Modifications:

- REANA requires some form of docker supporting container technology
  - Singularity and Shifter support finished.
- REANA expects to submit to a kubernetes cluster
  - Added support for VC3 specialized HTCondor submissions through a modified reana-job-controller and a job_wrapper for every workflow step.
  - The modified reana-job-controller submits each workflow step to a local condor scheduler
- Job Wrapper Auto-detection of container technology for workflow steps. (shifter, singularity)

VC3 Modifications:

- Cluster template for REANA+HTCondor
  - Uses the standard HTCondor template as the base to create a condor pool that sends jobs to HPC resources, translating the job to the corresponding batch system submission syntax via bosco.
  - Deploys Kubernetes via minikube
  - Deploys the REANA cluster and client and set up the environment, so the user can interact with them as soon as the VC3 headnode is created.
- GSI-SSH support
  - The GSI-SSH authentication mechanism was added in the infrastructure, in order to support e.g.P XSEDE HPC centers like Blue Waters, Stampede, NERSC.
  - Proxies can be renewed through the VC3 website for the virtual cluster allocations to remain active.
- Additional patches to support Cray Linux environments (NERSC, Blue Waters)

Creating A REANA cluster on Blue Waters

Launch New Virtual Cluster

Project: bwtest1

* = INDICATES REQUIRED FIELD

VIRTUAL CLUSTER NAME (A-Z, 0-9, _ AND -)*

reanabwv1

CLUSTER TEMPLATE FRAMEWORK *

REANA+HTCondor

NUMBER OF COMPUTE WORKERS: *

2

(From VC3 Web site)

ENVIRONMENT

Select Environment

ALLOCATIONS*

khurtado-bluewaters-ncsa

EXPIRATION

If not specified, expiration defaults to 6 hours from launch, when your virtual cluster will automatically be terminated.

HOURS:

98

Cancel | Launch Virtual Cluster

End result is a REANA cluster deployed on the VC3 headnode

Components are deployed via Kubernetes (minikube)

```
(reana) [khurtado@khurtado-reanabwv1 ~]$ reana-cluster status
COMPONENT             STATUS
job-controller        Running
server                Running
db                    Running
workflow-controller   Running
message-broker        Running
REANA cluster is ready.
(reana) [khurtado@khurtado-reanabwv1 ~]$ kubectl get pods
NAME                                                     READY   STATUS      RESTARTS   AGE
batch-serial-7e79ee48-036f-4049-87ee-a3dc66d8a1da-tl7zd   0/1     Completed   0          5h54m
db-69744557df-wg4mt                                      1/1     Running     0          5h55m
job-controller-5c7f4c8b4f-sgnj6                          1/1     Running     0          5h55m
message-broker-b7d66cf55-m9p4n                           1/1     Running     0          5h55m
server-58dc985c77-n2qpn                                  2/2     Running     0          5h55m
workflow-controller-668f69d4bc-x62w7                     2/2     Running     0          5h55m
```

Blue Waters Tests...

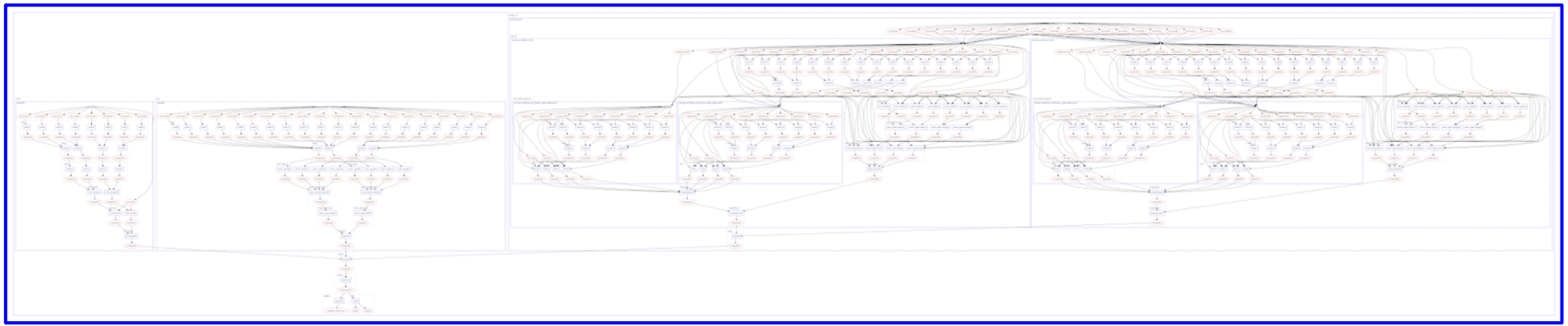
Successfully ran a complex physics test: BSM search

```
(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client status
NAME        RUN_NUMBER   CREATED              STATUS     PROGRESS
demobsmv2   1            2019-06-20T14:24:40  finished   18767/65

(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client ls | grep pdf
plot/prefit.pdf                                                          19405     2019-06-
20T15:31:49
plot/postfit.pdf                                                         19431     2019-06-
20T15:31:49

(reana) [khurtado@khurtado-reanabwvc3 condor]$ condor_history -constraint 'regexp("12610aee-f019-4888-
85e5-e9c098d28bf8", Args)' -af:h ClusterId ExitCode LastRemoteHost
ClusterId ExitCode LastRemoteHost
97        0        slot1_1@31738@nid25424
96        0        slot1_1@13090@nid27638
95        0        slot1_1@13090@nid27638
94        0        slot1_1@13090@nid27638
93        0        slot1_1@13090@nid27638
```

Blue Waters Tests...

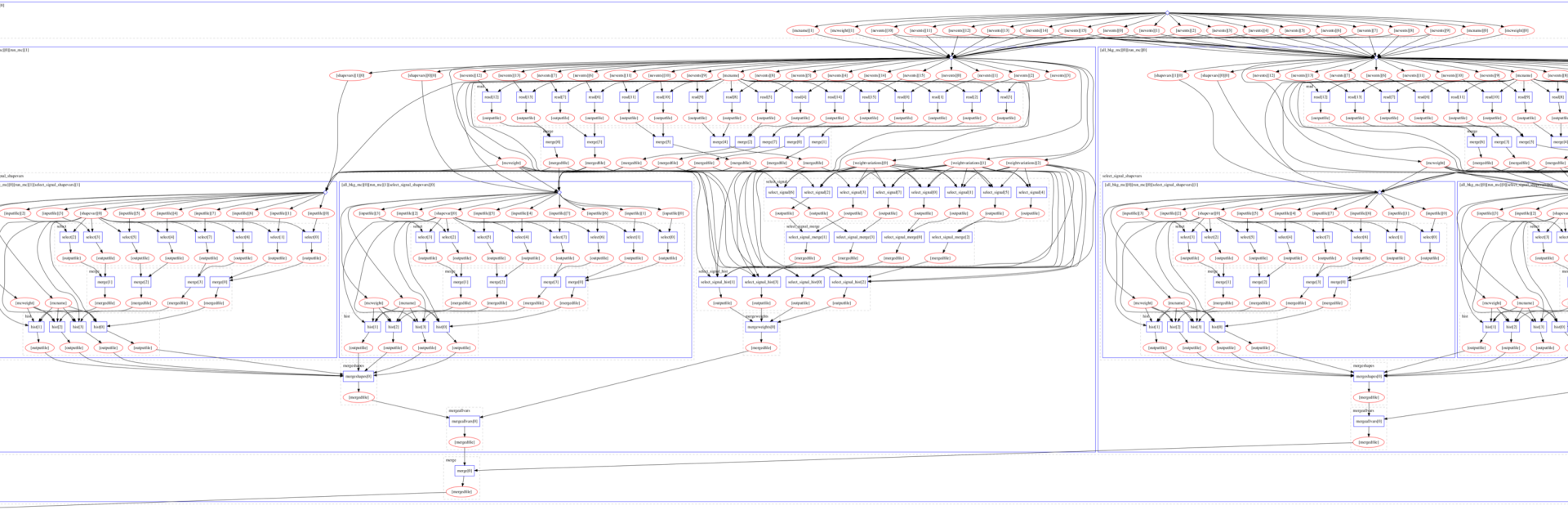
Successfully ran a complex physics test: BSM search

Here is the workflow diagram, to give you an idea of the complexity.



Blue Waters Tests...

Zoom in…



Blue Waters Tests...

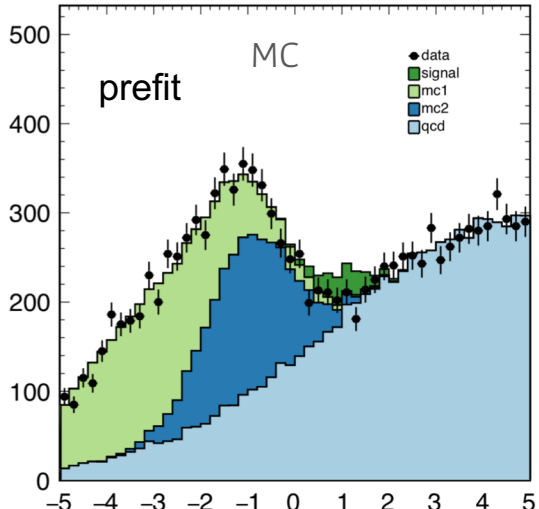
Once results are downloaded through the client, final plots look like this:

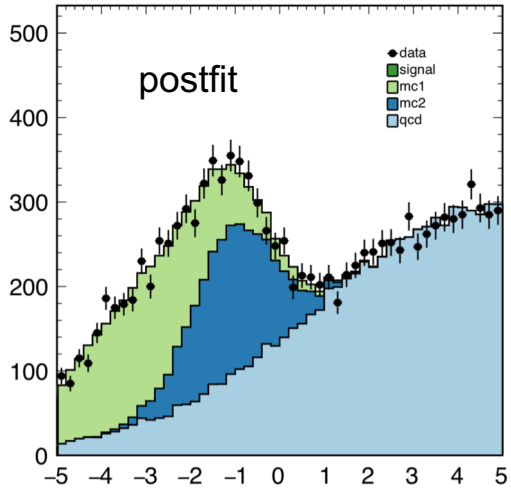
```
(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client download plot/prefit.pdf
File plot/prefit.pdf downloaded to /home/khurtado/condor.
(reana) [khurtado@khurtado-reanabwvc3 condor]$ reana-client download plot/postfit.pdf
File plot/postfit.pdf downloaded to /home/khurtado/condor.
```

Data is generated/emulated according to Standard Model expectations.

After processing, a statistical model involving both signal and control regions is built and the model is fitted against the observed data.

The signal sample is scaled down significantly to fit the data, which is expected since the data was emulated in accordance with a SM-only scenario





Conclusions, next steps

- In the process of standing up procedure on **NERSC/Cori**
  - **Mostly working through authentication issues at this point**
- Run actual Machine Learning training workflows
- Benchmark performance on variety of HPC systems/sites
- Clean up implementation and documentation
- Integration into central REANA repository at CERN
- End result will be flexible infrastructure capable of large-scale ML training

Supported by NSF Award OAC-1841448

Notre Dame Contacts = Main developers

- Kenyi Hurtado
  - khurtado@nd.edu
- Cody Kankel
  - ckankel@nd.edu

Thanks!

Mike Hildreth: mhildret@nd.edu; ND Developers Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

Backup slides

Links

- SCAILFIN Source code:
  - SCAILFIN's modified RJC
    https://github.com/scailfin/reana-job-controller/tree/job_manager
  - REANA
    - https://github.com/reanahub
  - VC3
    - https://github.com/vc3-project
- Websites
  - https://www.virtualclusters.org
  - http://www.reanahub.io/

VC3: Virtual Clusters for Community Computation

Constraints

- At present, workers need outgoing network connection for a virtual cluster to work.
    - So, resources like ALCF/Theta are out of the scope with this approach.
    - But e.g.: XSEDE resources like NERSC, Blue Waters, Stampede or PSC–Bridges do meet the outgoing network requirement for example.

SCAILFIN's REANA modifications

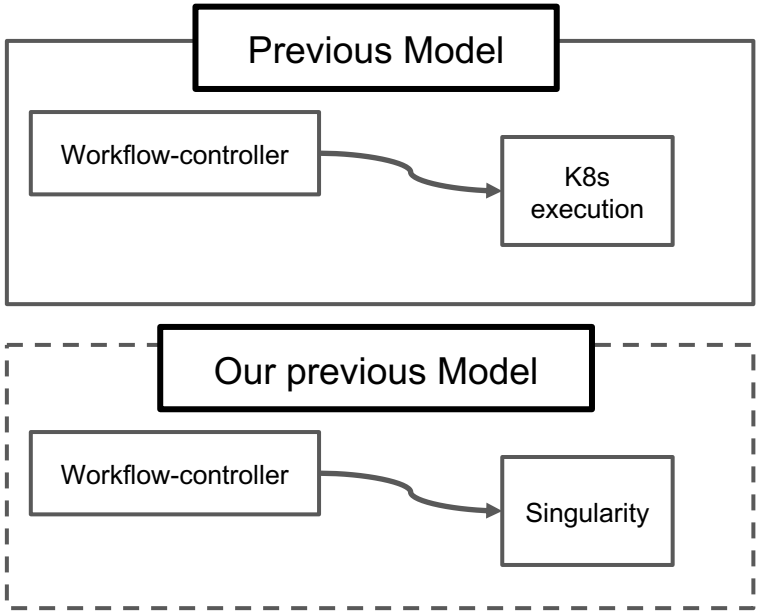
- REANA requires some form of docker supporting container technology
  - Singularity and Shifter support in the works. Possibly CharlieCloud

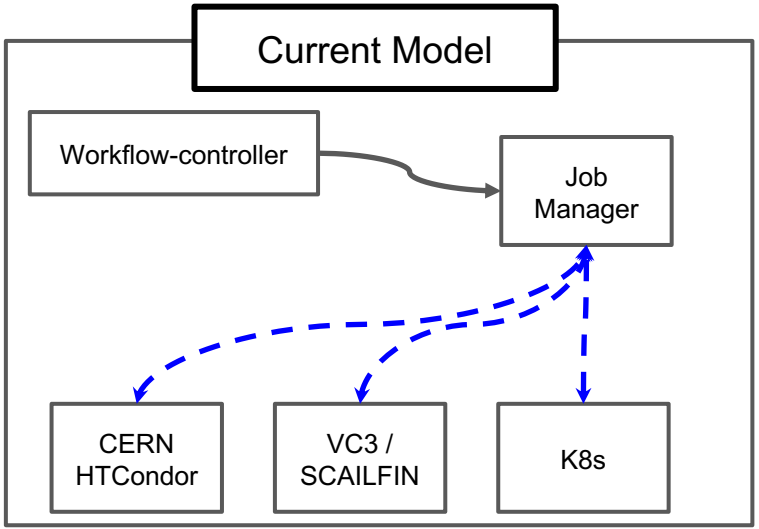
- REANA expects to submit to a kubernetes cluster
  - Added support for VC3 specialized HTCondor submissions through a modified reana-job-controller and a job_wrapper for every workflow step.
  - The modified reana-job-controller submits each workflow step to a local condor scheduler

- Job wrapper
  - Each workflow step is wrapped by a script which searches for container technology and launches each workflow step into the available container (shifter, singularity)

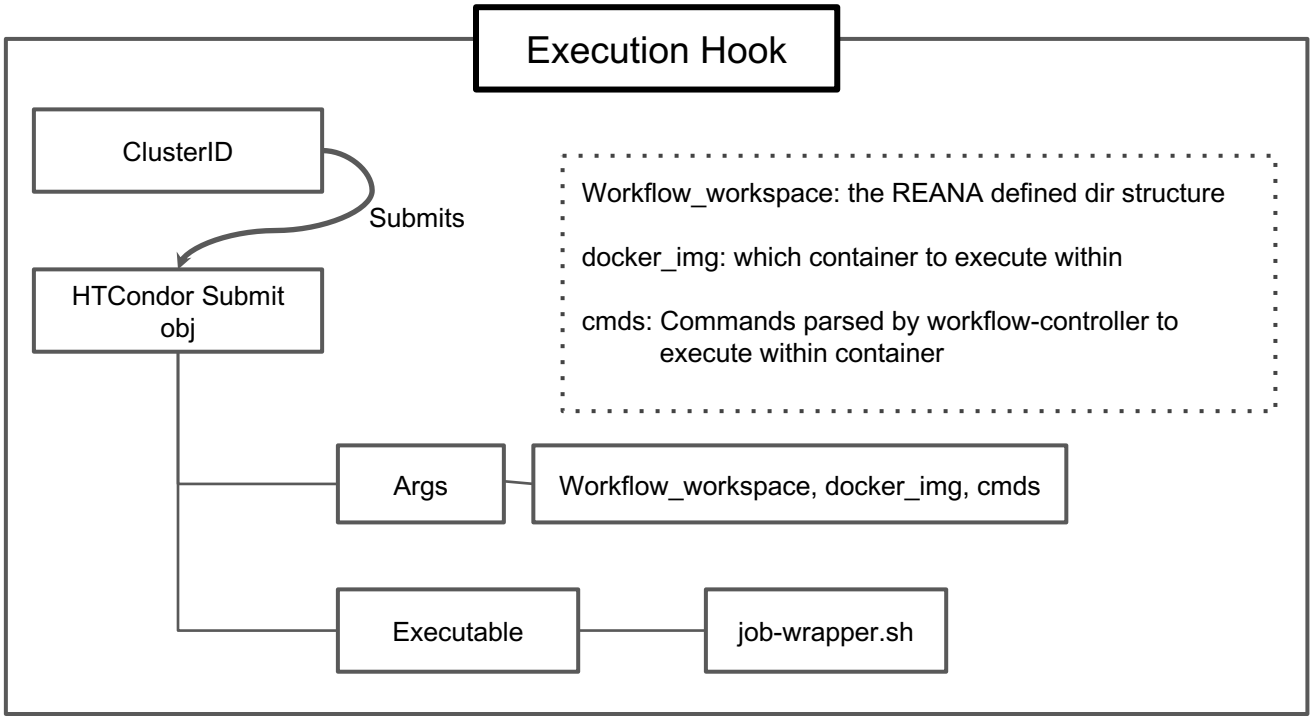
SCAILFIN's REANA modifications

- REANA developers allowed abstraction of reana-job-controller's backend
  - Allows for "plug and play" of backends / job execution component



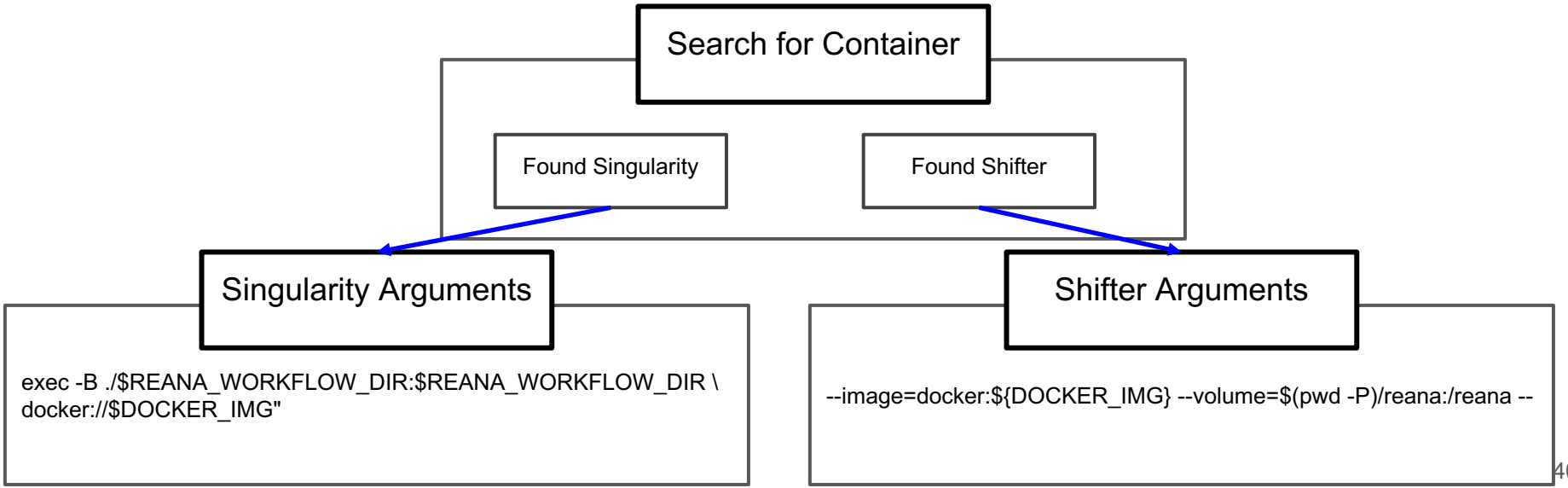


SCAILFIN's REANA modifications



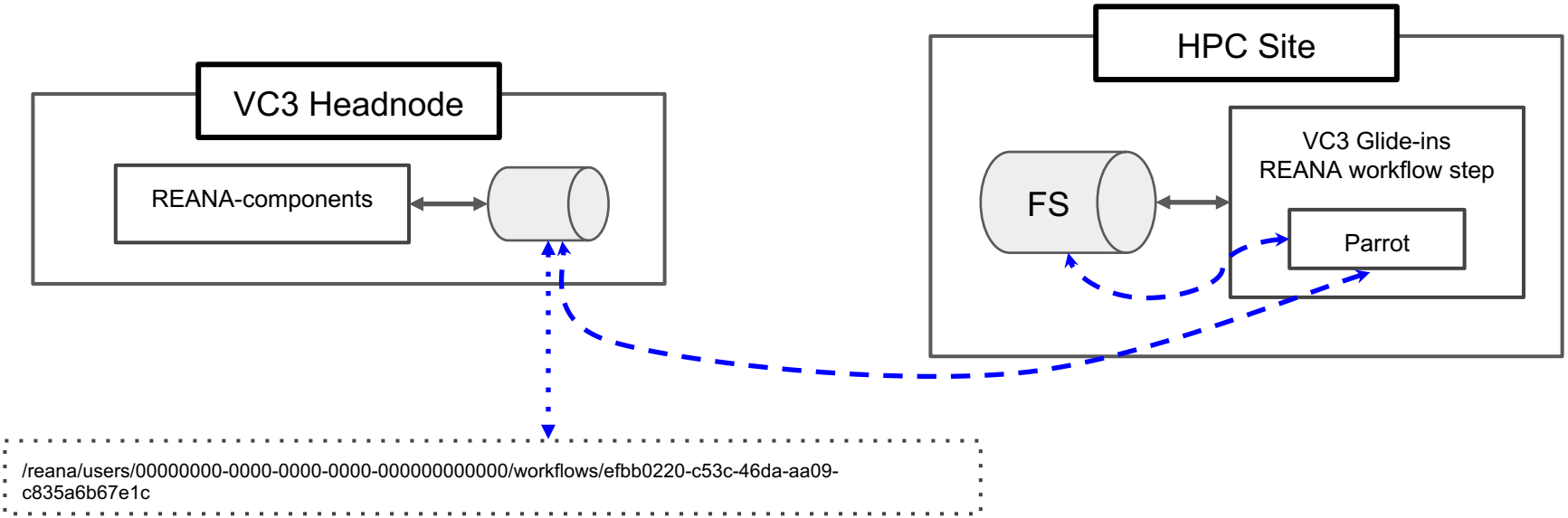
Job-wrapper Container management

- Performs check for available container technologies
  - Checks Binaries in $PATH (VC3 may auto-load these)
  - Attempts to load modules for Singularity and Shifter
  - Executes workflow-step within discovered container
    - Will choose depending on the currently set $default



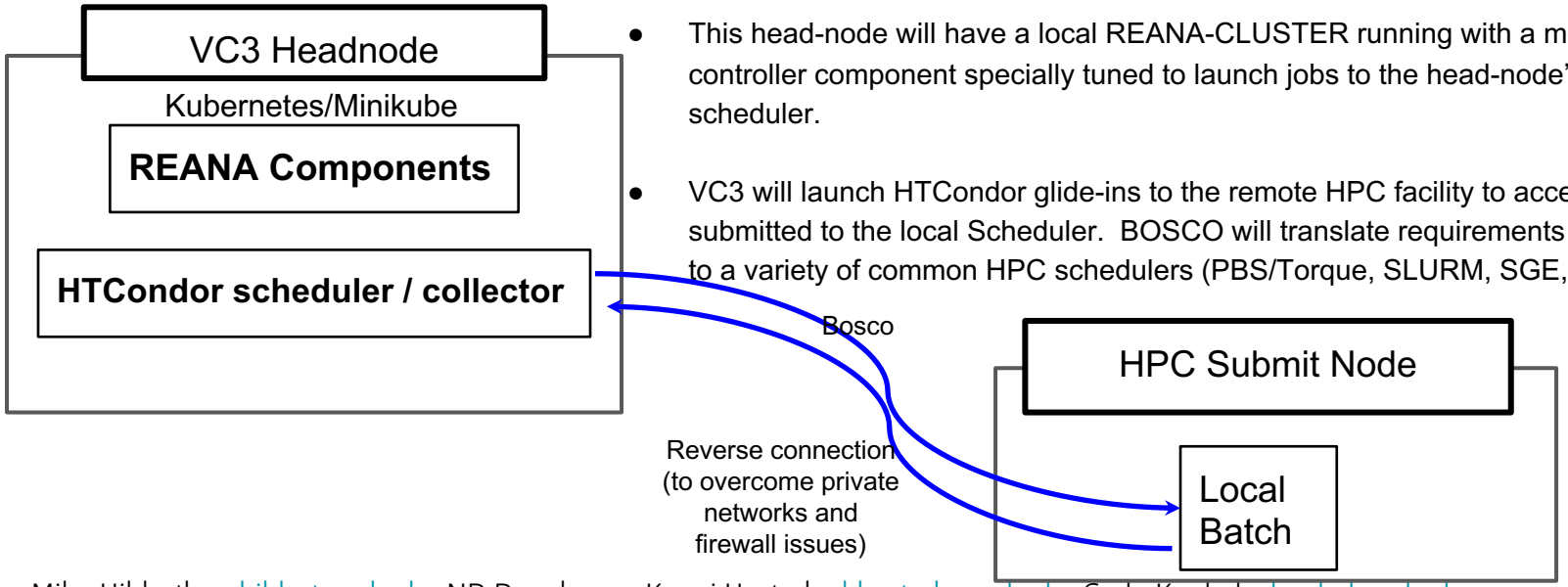
Job-wrapper File management

- REANA expects files to be where the workflow controller placed them
  - We must simulate this directory structure on the HPC infrastructure
- We can easily transfer entire directory structures with HTCondor's chirp protocol and Parrot



SCAILFIN and VC3

VC3 is utilized for remote connections to clusters.



- This head-node will have a local REANA-CLUSTER running with a modified job-controller component specially tuned to launch jobs to the head-node's HTCondor scheduler.
- VC3 will launch HTCondor glide-ins to the remote HPC facility to accept jobs submitted to the local Scheduler. BOSCO will translate requirements from HTCondor to a variety of common HPC schedulers (PBS/Torque, SLURM, SGE, etc.)

Mike Hildreth: mhildret@nd.edu; ND Developers Kenyi Hurtado: khurtado@nd.edu, Cody Kankel: ckankel@nd.edu

SCAILFIN's VC3 modifications

- Cluster template for REANA+HTCondor
  - Uses the standard HTCondor template as the base to create a condor pool that sends jobs to HPC resources, translating the job to the corresponding batch system submission syntax via bosco.
  - Deploys Kubernetes via minikube
  - Deploys the REANA cluster and client and set up the environment, so the user can interact with them as soon as the VC3 headnode is created.
- GSI-SSH support
  - The GSI-SSH authentication mechanism was added in the infrastructure, in order to support e.g.P XSEDE HPC centers like Blue Waters, Stampede, NERSC.
  - Proxies can be renewed through the VC3 website for the virtual cluster allocations to remain active.
- Additional patches to support Cray Linux environments (NERSC, Blue Waters)