Analysis Preservation and Systematic Reinterpretation within the ATLAS Experiment

Kyle Cranmer, L. Heinrich
on behalf of the ATLAS collaboration

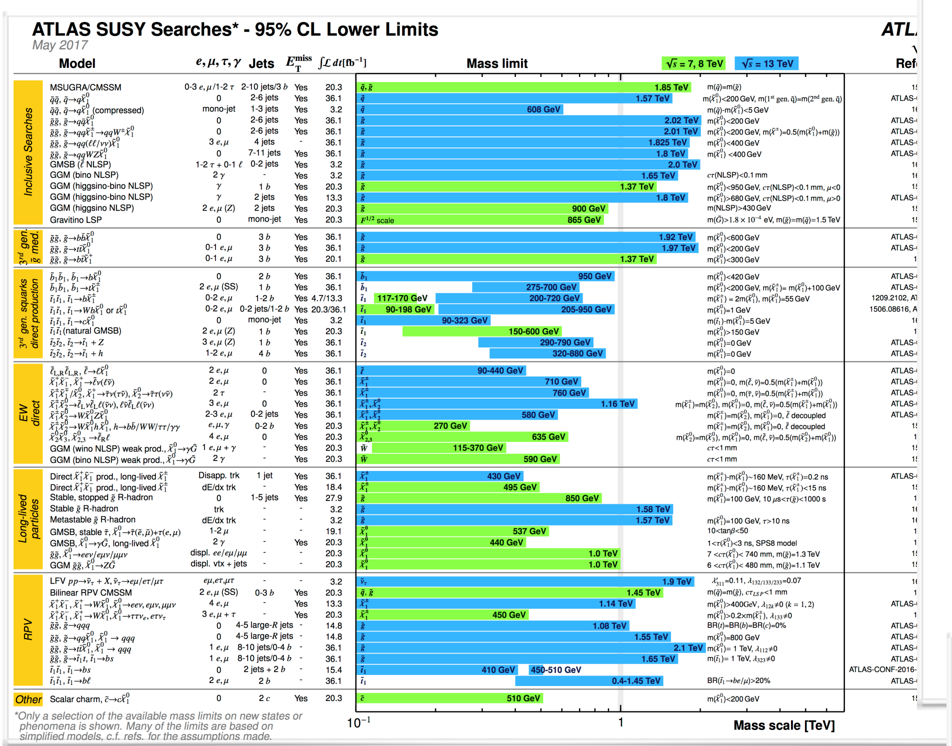
ACAT 2017

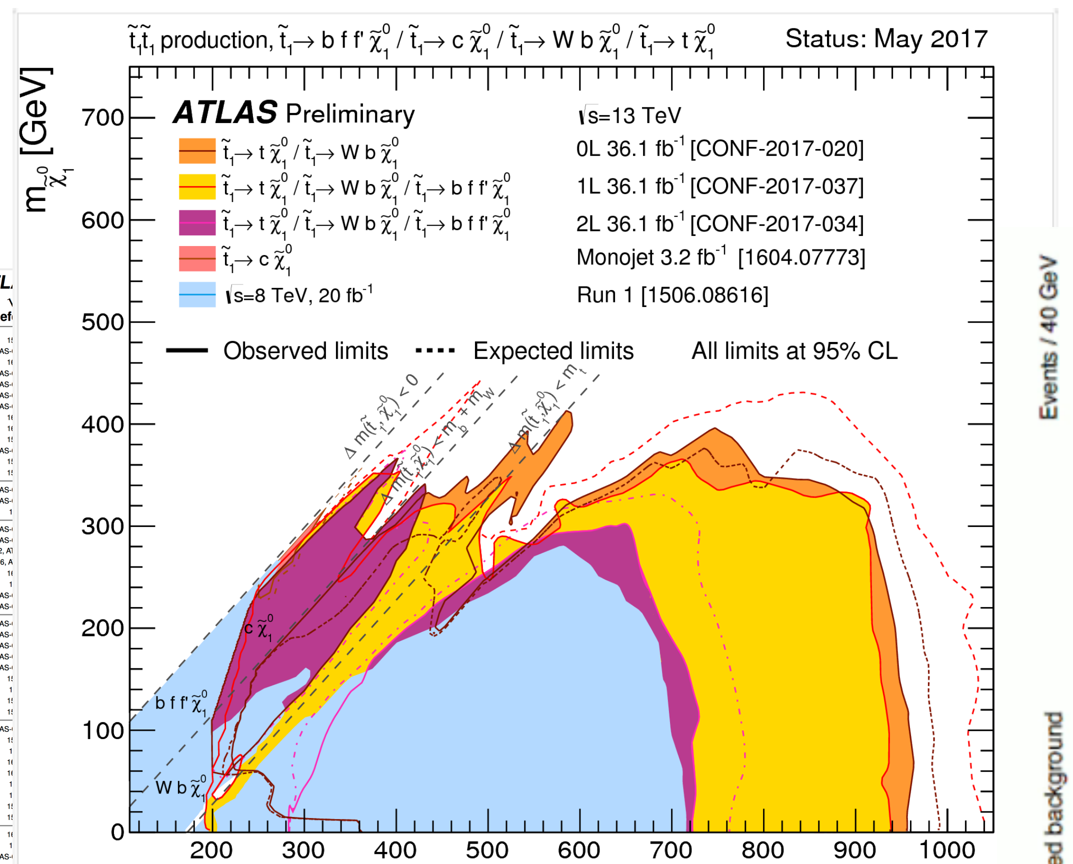
NEW YORK UNIVERSITY

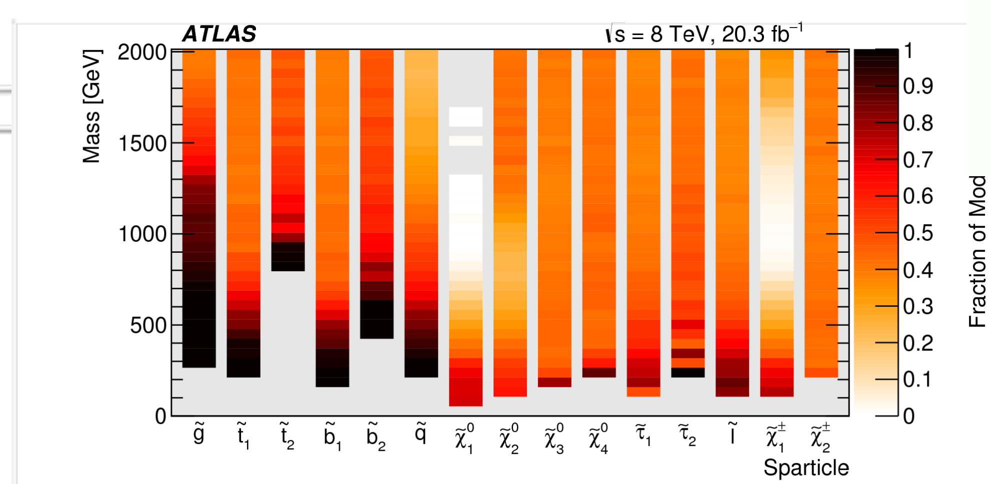
The Need for Reinterpretation

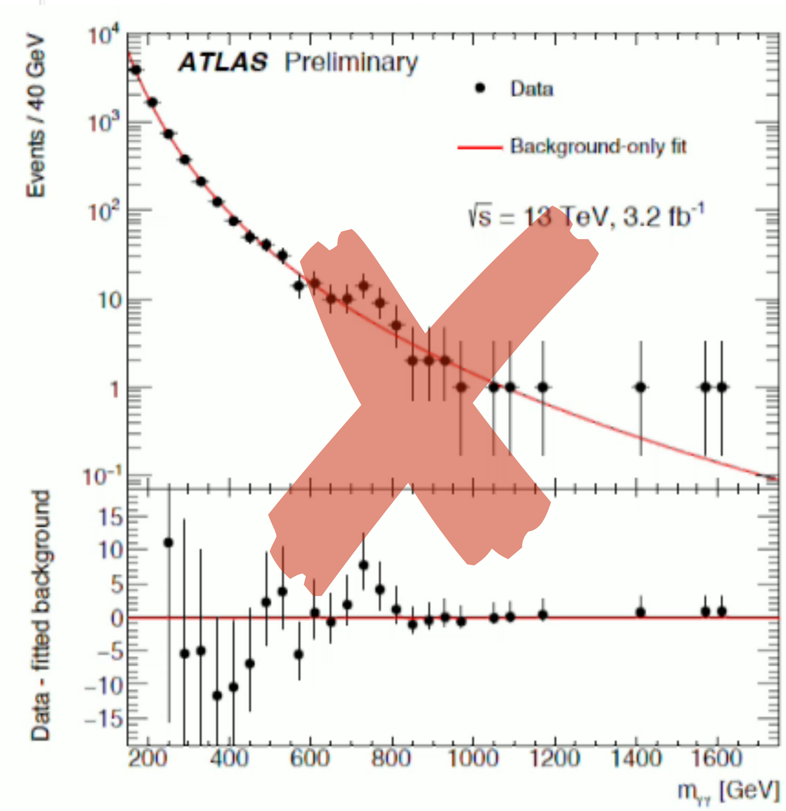
After the Higgs discovery completed the Standard Model, the search for BSM physics has become an even higher priority.

ATLAS is producing tons of results.. so far we have not found any significant excess (sorry no 750 GeV…)









The Need for Reinterpretation

Where is the New Physics?

- hide in unexpected places, complex final states, low-rate / low-acceptance scenarios (e.g. compressed models, models spreading across many topologies)
- not be reachable at all at the LHC

Q

how do we exploit the LHC data to maximize our understanding of the model landscape?

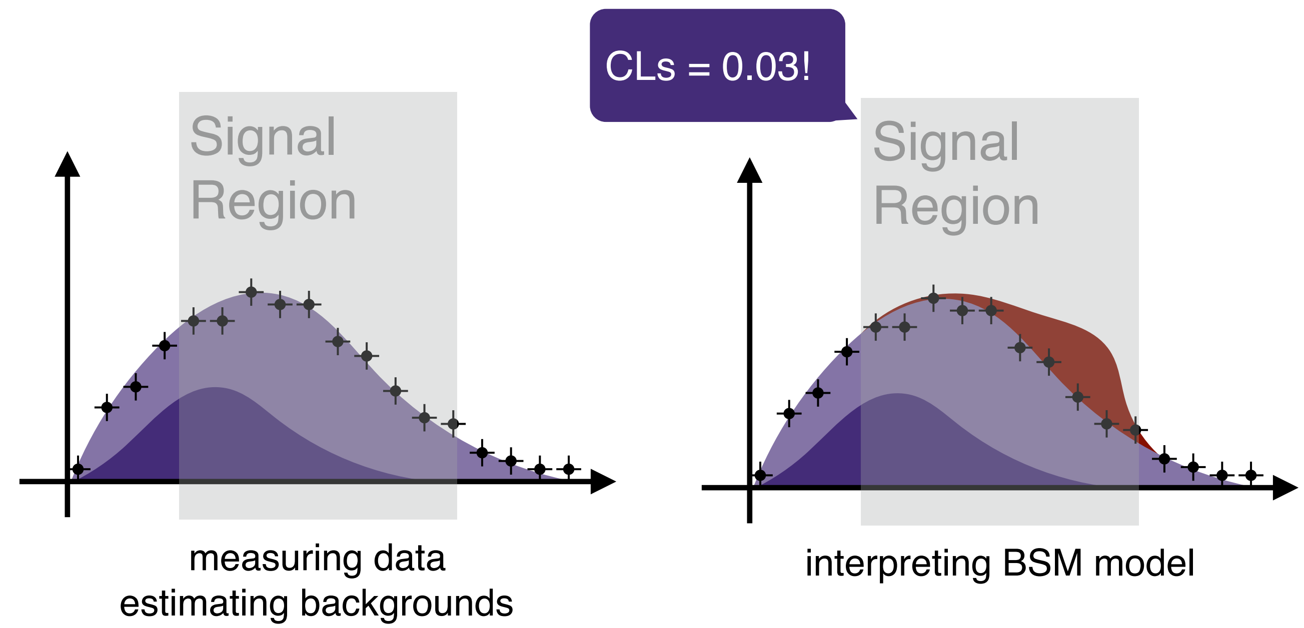
there are ***many more*** candidate models than we have graduate students to design dedicated analyses for each new model — let's make the most of the analyses that we do have. **Many of them are sensitive to a whole range of models.**

A

leverage modern analysis preservation and reusability techniques to *re-interpret existing analyses*

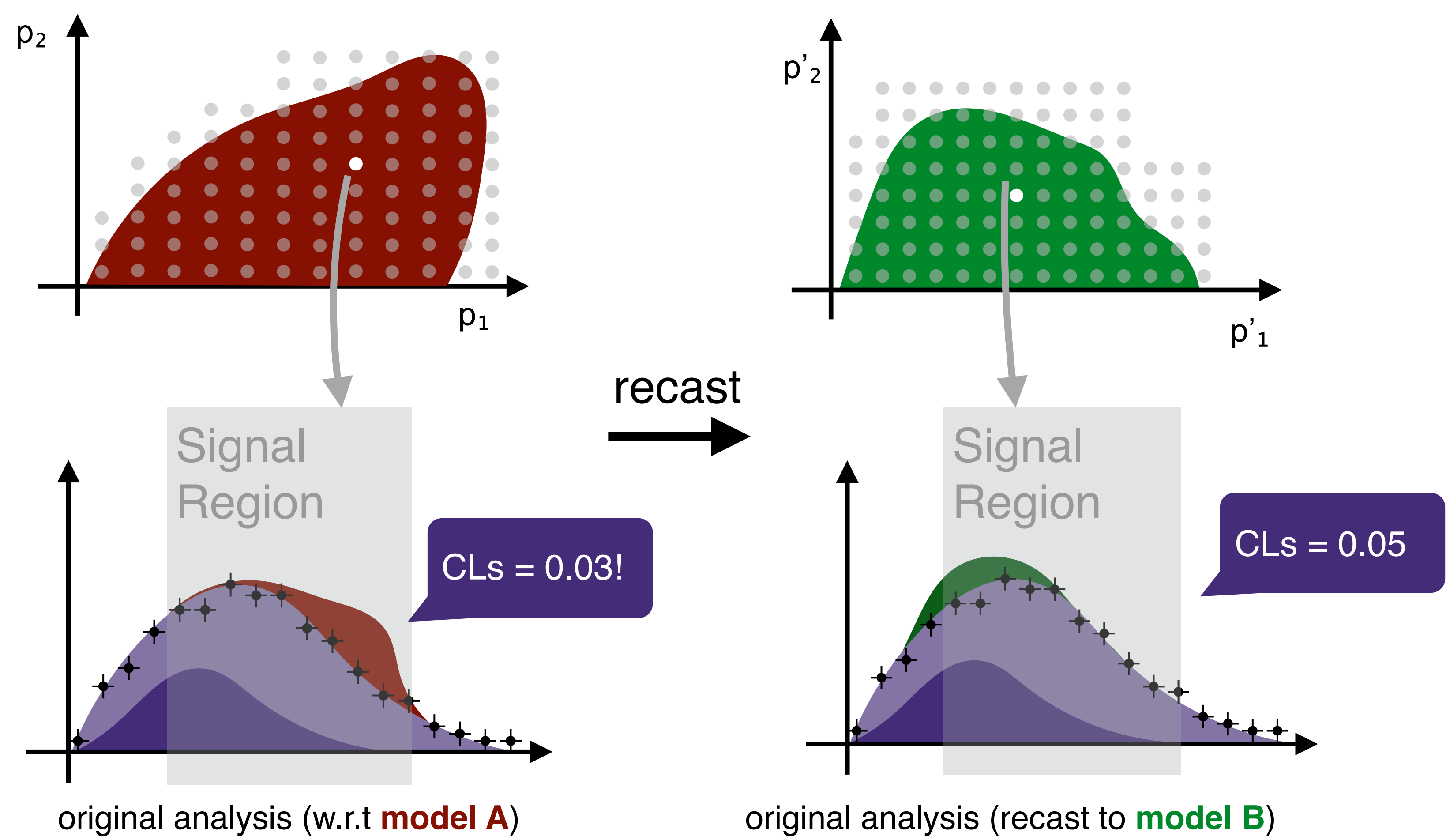
Most of the work goes into: **taking data, designing**, **validating** the analysis strategy, **understanding Standard Model backgrounds.**

Model interpretation come at the end, and are technically the **easiest part:** analysis pipeline is **fixed** after unblinding, MC dataset sizes small. Analysis teams routinely check hundreds of parameter points (of their favorite model).



Reinterpreting / Recasting:

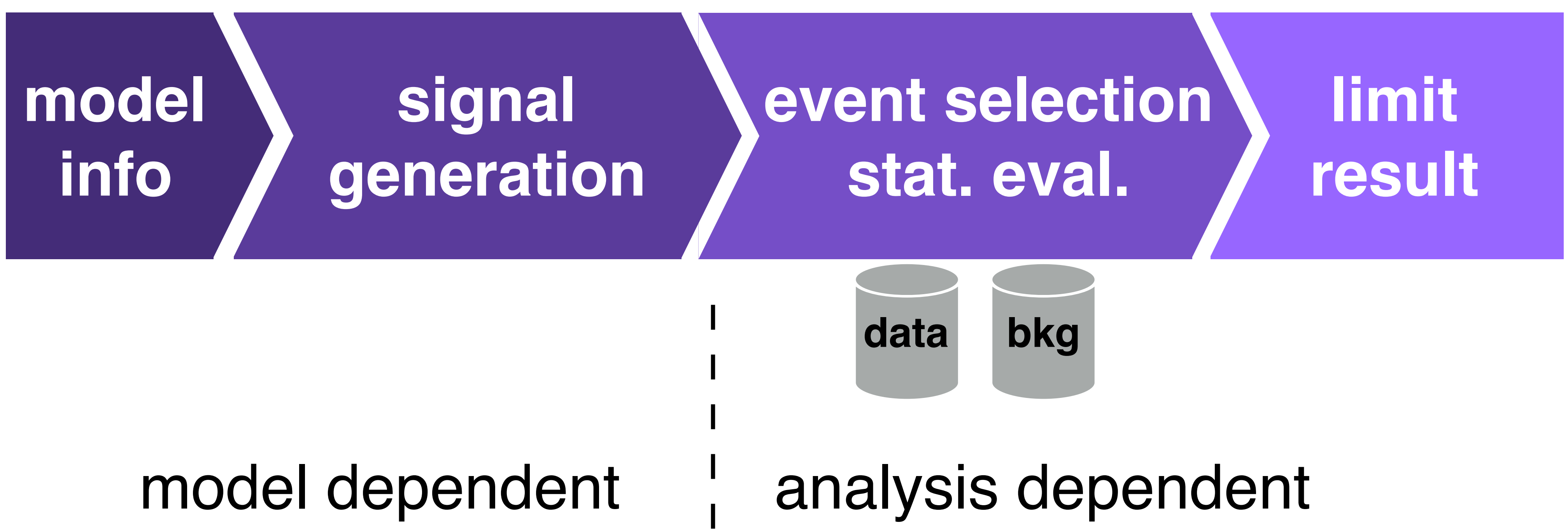
1. predicting BSM contributions of new model under a given analysis (**same** event selection**)**
2. statistical analysis of new signal with respect to ***same*** data and background estimates as original analysis to derive **new limits**



The Recipe for Reinterpretation

Reinterpretation follows a straight-forward recipe with three ingredients:

1. Ability to generate new signal model (incl. access to DetSim, Reco)[1]
2. Access to the analysis / event selection logic
3. Access to data and background distributions (incl. systematic variations) for statistical analysis



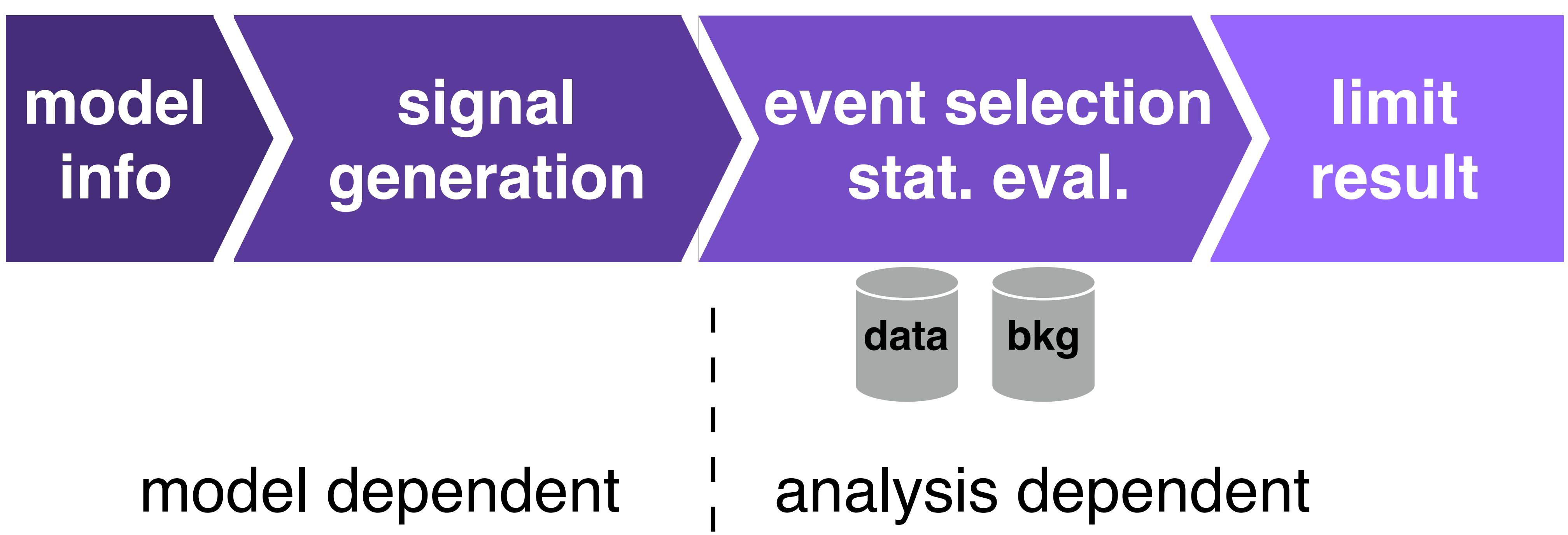
[1] unless analysis unfolded — but then other issues

The Recipe for Reinterpretation

Reinterpretation follows a straight-forward recipe with three ingredients:

1. Ability to generate new signal model (incl. access to DetSim, Reco)[1]
2. Access to the analysis / event selection logic
3. Access to data and background distributions (incl. systematic variations) for statistical analysis

- **true implementation of recipe requires access to collaboration-internal data/software**
- **there is an eco-system built by the pheno community to approximately implement the recipe CheckMate, ATOM, SModelS, SUSY-AI, etc..**

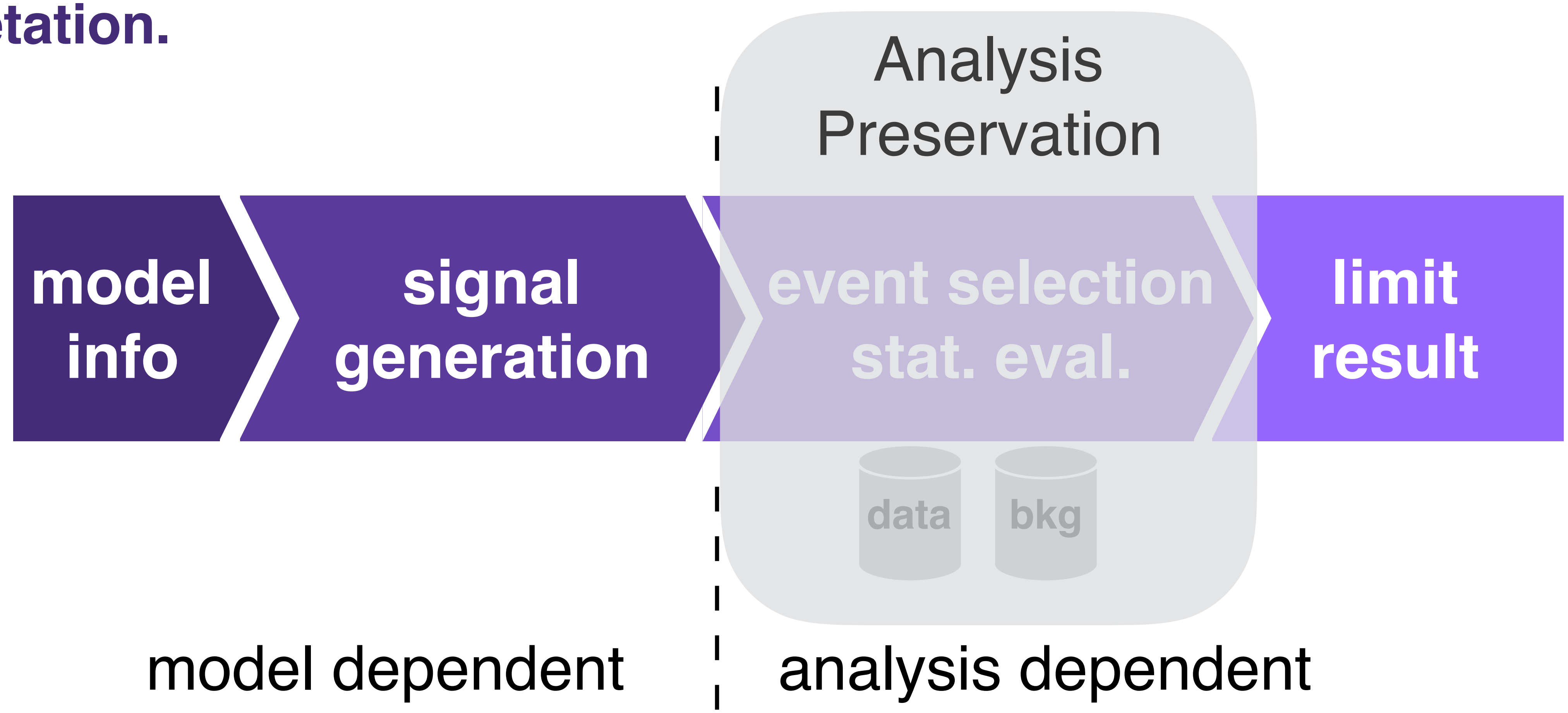


[1] unless analysis unfolded — but then other issues

ATLAS has well-managed and standardized process to generate new signals — almost anyone can do it / request new samples — not the bottleneck.

The most challenging aspect for ATLAS has been the preservation of the downstream analysis code

Solving analysis preservation / reusability enables systematic reinterpretation.



[1] unless analysis unfolded — but then other issues

Analysis Preservation in ATLAS

Challenges for analysis preservation

- need to preserve such that it can be **re-run on new input**
- real ATLAS analyses are complex. Not a single file in a common framework (like e.g. Rivet, CheckMate, LHADA). **There's a reason have our own computing model.**
  - code is very diverse. many frameworks, scripts, etc..
- distributed teams, code, data: **one person rarely is able to run the entire analysis** pipeline — some develop event selection, some background estimates, some statistical analysis

To preserve analyses, we needed to respect the tools, workflows people use. instead of forcing a re-implementation, develop toolchain to capture what they are already doing.

1. capture software (*including all dependencies)* needed to run individual parts of an analysis (e.g. event selection) in a future-proof way.
2. capture logic how the many pieces of the analysis fit into an *analysis workflow* that can be re-executed on a new signal

Analysis Preservation in ATLAS

comprehensive software capture was intractable until recently (VMs??). Now progress in IT industry has **made it feasible — Linux Containers.** Technology with wide industry support — will be here for foreseeable future.

revolutionized software distribution & archival — "app store for generic software". Many additional tools that help deploy / run Linux Containers in "the cloud" (Google, Amazon, Microsoft, etc…).

Containers are now becoming a major topic in LHC collaborations. Simplifies a lot of our computing in many ways.

technology stack enabling realistic analysis preservation has become available recently

Containerizing common ATLAS software:

centralized effort by software infrastructure team to build base images on which standard ATLAS software (e.g. offline reconstruction, analysis frameworks) run. (see poster by M .Vogel)

Choice between "lean" and "fat" containers

- **provides software preservation for common ATLAS workloads (reconstruction, simulation, etc)**
- **provides easy-to-use base images for end-users**

Enabling User-Level Analysis Software Preservation via Continuous Integration

Crucial to have streamlined system to capture user analysis code into containers — not everyone will become container expert.

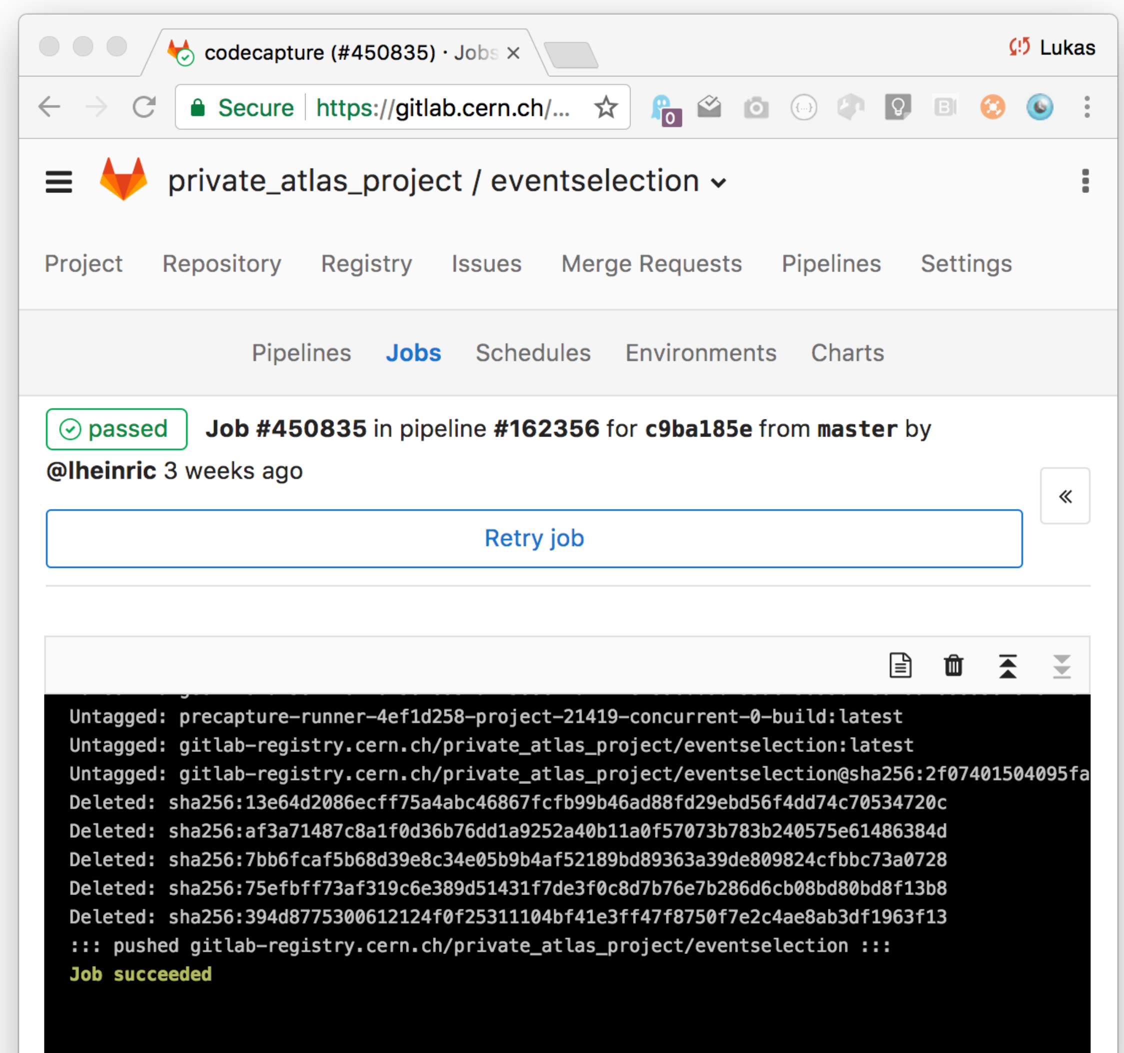
Increasing usage of CERN GitLab installation for managing user code allows us to use built-in continuous integration infrastructure to capture code **at development time. Avoids asking people for code after publication.**

docker.build.sh 98 Bytes

```
echo "building analysis code..."

rcsetup Base,2.4.14
rc find_packages
rc compile

echo "done.. "
```

.gitlab-ci.yml 219 Bytes

```
codecapture:
  tags: [code-capture]
  variables:
    BUILD_BASE_IMAGE: lukasheinrich/recast_cvmf
  image: gitlab-registry.cern.ch/codecapture/co
  script:
   - /codecapture_utils/steering.sh
```



Declarative Parametrized Workflow Description

developed declarative workflow language ***yadage*** based on ubiquitous industry standards (JSON) to describe logic between separate analysis stages.

allows arbitrary, runtime-dependent, directed acyclic graphs of containerized analysis workfloads

models original analysis workflow instead of forcing common interface.

Can be developed, validated ***during analysis developement,*** stored as data fragment in repo

parameters to bind ***new inputs*** *(samples + xsec)*

```
stages:
  - name: selection
    dependencies: ['init']
    scheduler:
      scheduler_type: singlestep-stage
      parameters:
        cxaodfile: {stages: init, output: cxaodfile, unwrap: true}
        did:       {stages: init, output: did, unwrap: true}
        xsec_pb:   {stages: init, output: xsec_pb, unwrap: true}
        k_factor:  {stages: init, output: k_factor, unwrap: true}
        filt_eff:  {stages: init, output: filt_eff, unwrap: true}
        nametag: 'recast'
        mctype: {stages: init, output: mctype, unwrap: true}
        outputdir: '{workdir}/output'
      step: {$ref: 'selscript.yml#'}
  - name: fit
    dependencies: ['selection']
    scheduler:
      scheduler_type: singlestep-stage
      parameters:
        fitinputdir: '{workdir}/fitinputs'
        signalfile: {stages: selection, output: out, unwrap: true}
        zerolepbkg: 'root://eosuser.cern.ch///eos/project/r/recast/a
        onelepbkg: 'root://eosuser.cern.ch///eos/project/r/recast/at
        twolepbkg: 'root://eosuser.cern.ch///eos/project/r/recast/at
        nametag: 'recast'
        limitfile: '{workdir}/limits.txt'
        plotdir: '{workdir}/plots'
      step: {$ref: 'fitscript.yml#'}
```

referencing ***archived assets*** (data, bkg histos)

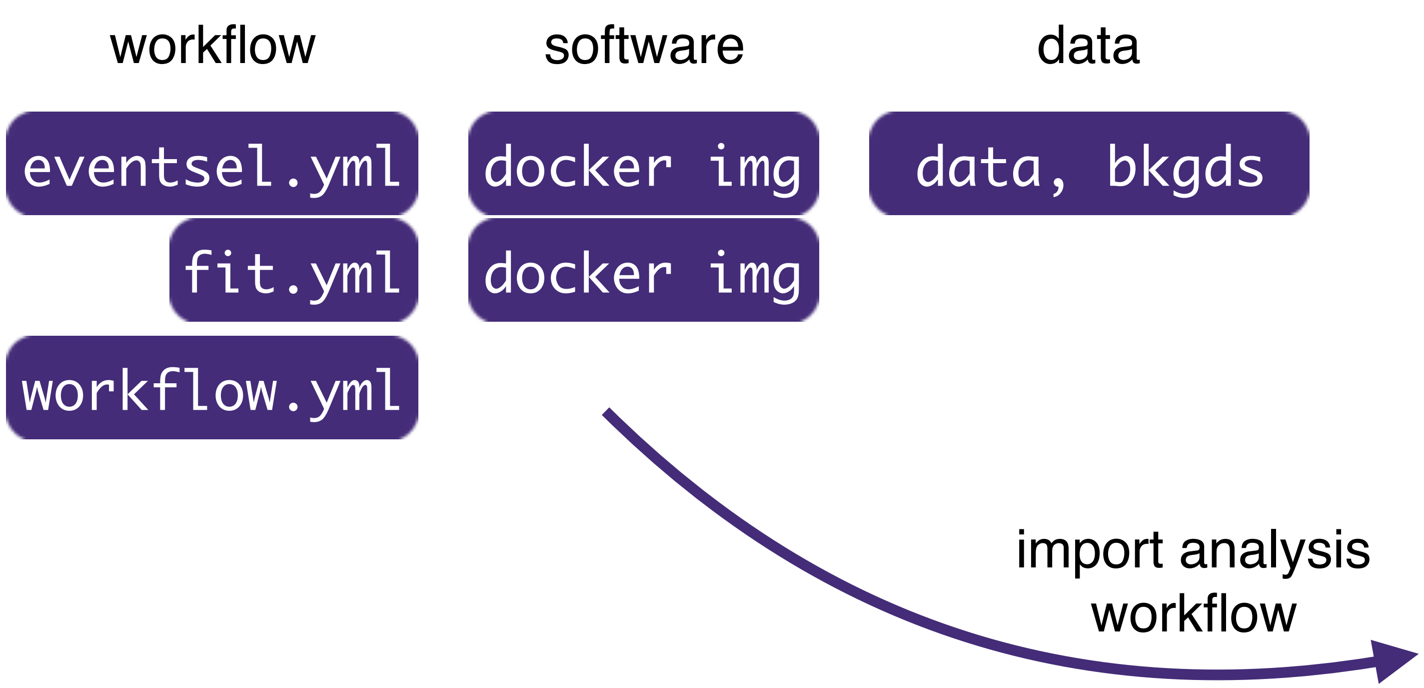
selscript.yml 974 Bytes

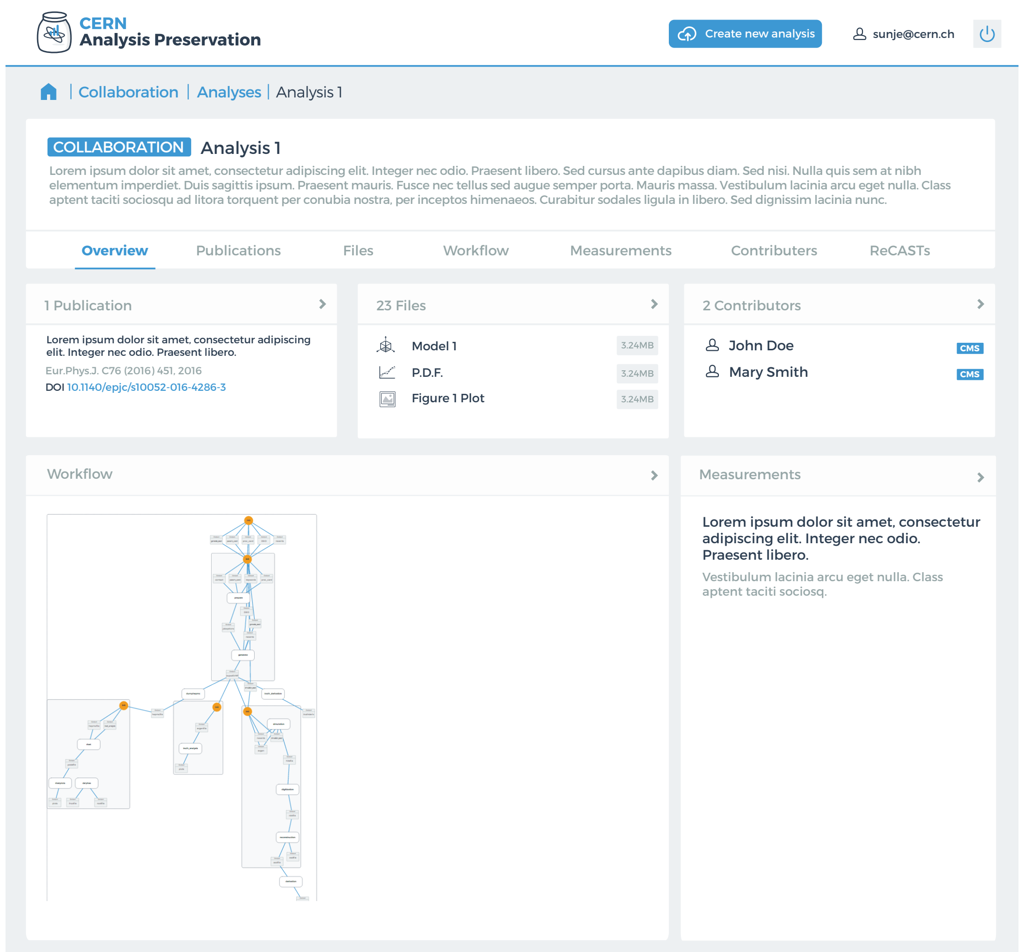
```
process:
  process_type: 'interpolated-script-cmd'
  interpreter: bash
  script: |
    source ~/.bashrc
    source ./rcSetup.sh
    /recast_auth/getkrb.sh

    echo {cxaodfile} > /tmp/filelist
    cat FrameworkSub/data/XSections_13TeV.txt|grep -v '^{did}'> tmp.txt
    mv tmp.txt FrameworkSub/data/XSections_13TeV.txt
    echo 'adding this line to XSections_13TeV.txt'
    echo '{did} {xsec_pb} {k_factor} {filt_eff} {nametag} {mctype}_RECAST'
    echo '{did} {xsec_pb} {k_factor} {filt_eff} {nametag} {mctype}_RECAST' >> F
    hsg5frameworkReadCxAOD_monoVH {outputdir} data/FrameworkExe_monoVH/framewor
publisher:
    publisher_type: interpolated-pub
    publish:
      out: '{outputdir}/hist-testrun.root'
environment:
  environment_type: 'docker-encapsulated'
  image: gitlab-registry.cern.ch/prieck/monohbb16_preservation/eventselection
  resources:
    - CVMFS
    - GRIDProxy
```

Analysis Preservation in ATLAS

Close collaboration with **CERN Analysis Preservation (CAP) Portal.** Invenio-based portal that can natively ingest:

- code and software environments (clone repos, import docker images)
- native support for workflow specification (yadage workflows adhere to JSON schema spec)
- ingestion of data assets (data + background histograms / ntuples, etc..)





CERN Analysis Preservation

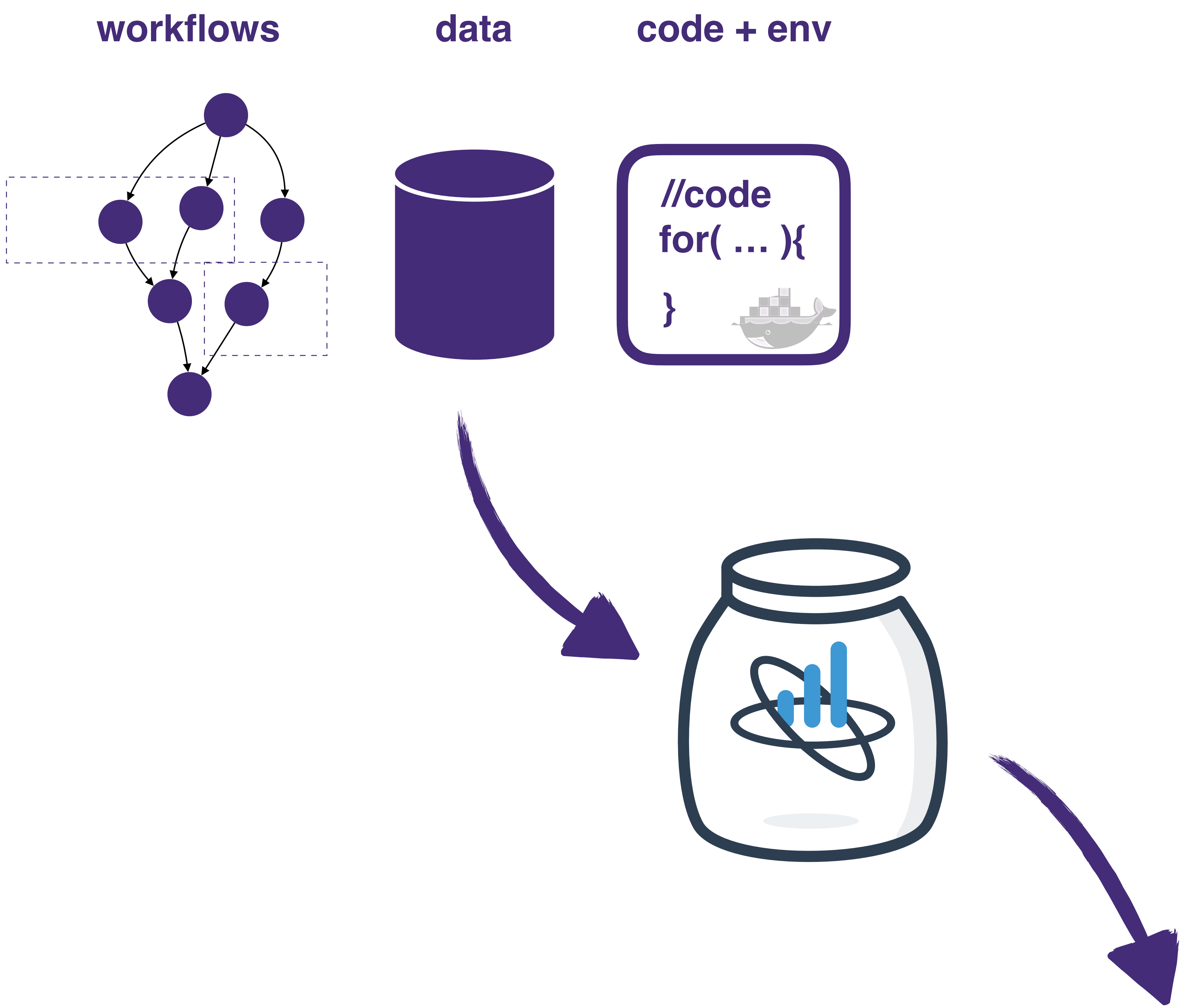
to recap..

analysis preservation during development

capture analysis software using linux containers through continuous integration

standards-based workflow definition to build workflow graph of analysis stages

deep integration with CERN Analysis Preservation

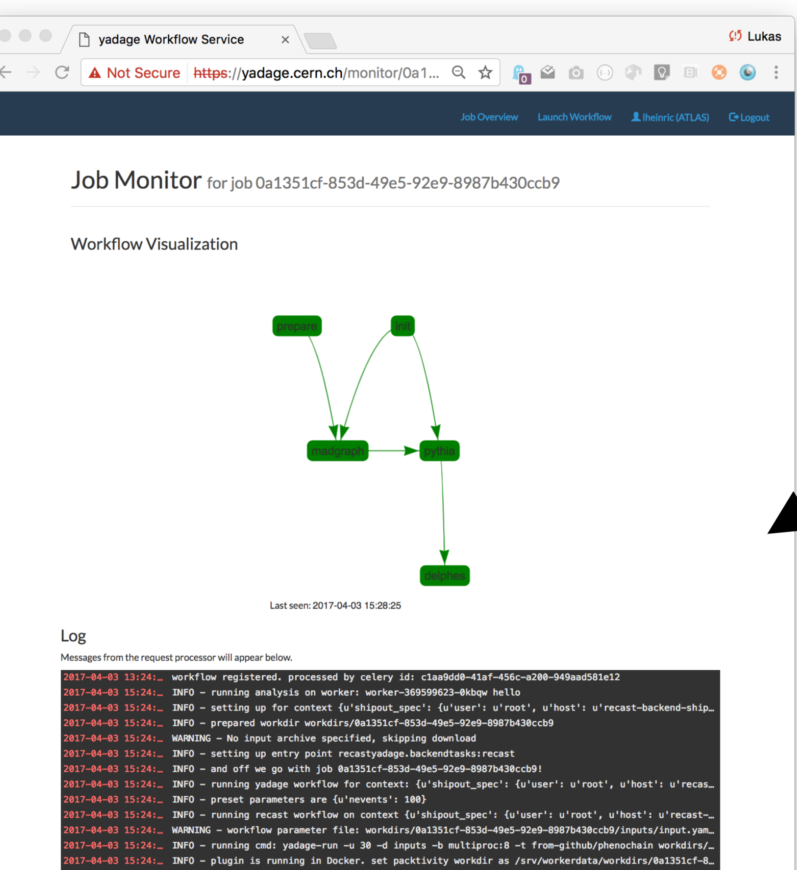


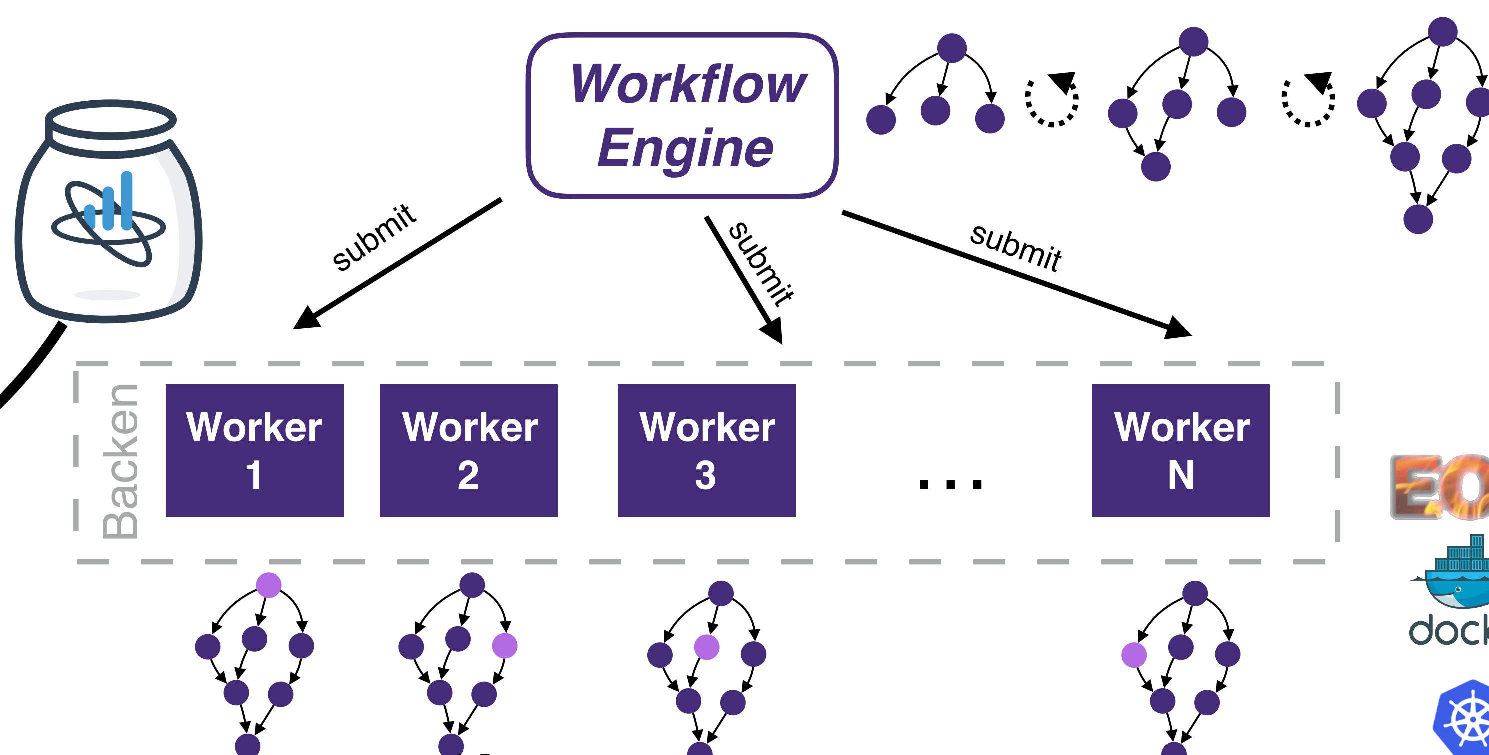
Once archived, we can build new client applications by re-using analyses

REANA

REANA — generic Workflows-as-a-Service platform

- main unit of work is a containerized workflow as stored in CAP
- “workflow engines” control what container jobs to submit
  - yadage support at launch
  - exploring Common Workflow Language
- deployed natively on Kubernetes. Deployable at AWS, GCE, CERN OpenStack.
- Multi-experiment support (tested w/ ATLAS, LHCb)
- Joint Effort by CERN IT and SIS, DASPOS, DIANA-HEP, funded by NSF, MSDSE

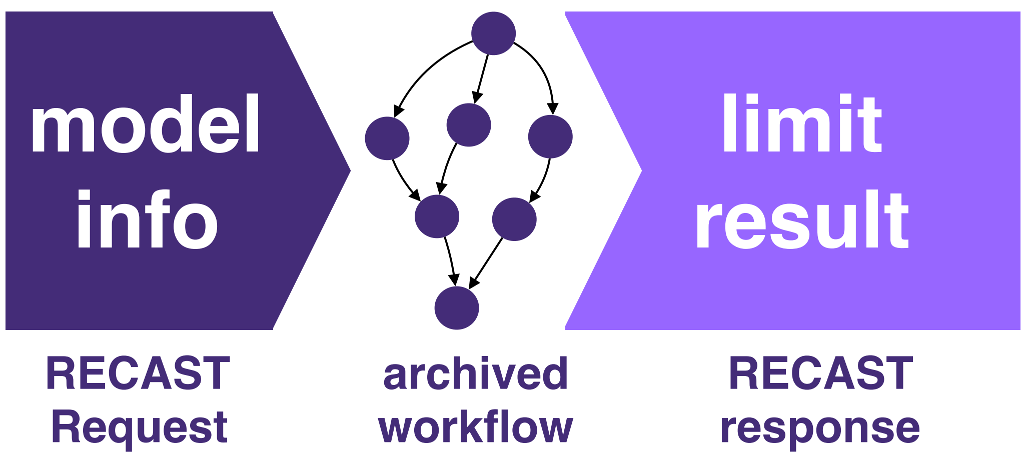




RECAST

RECAST — a semantic layer on top of REANA for reinterpretations

Reinterpretations provide a fundamental interface that can be satisfied by a (combination of) containerized workflows.



Idea:

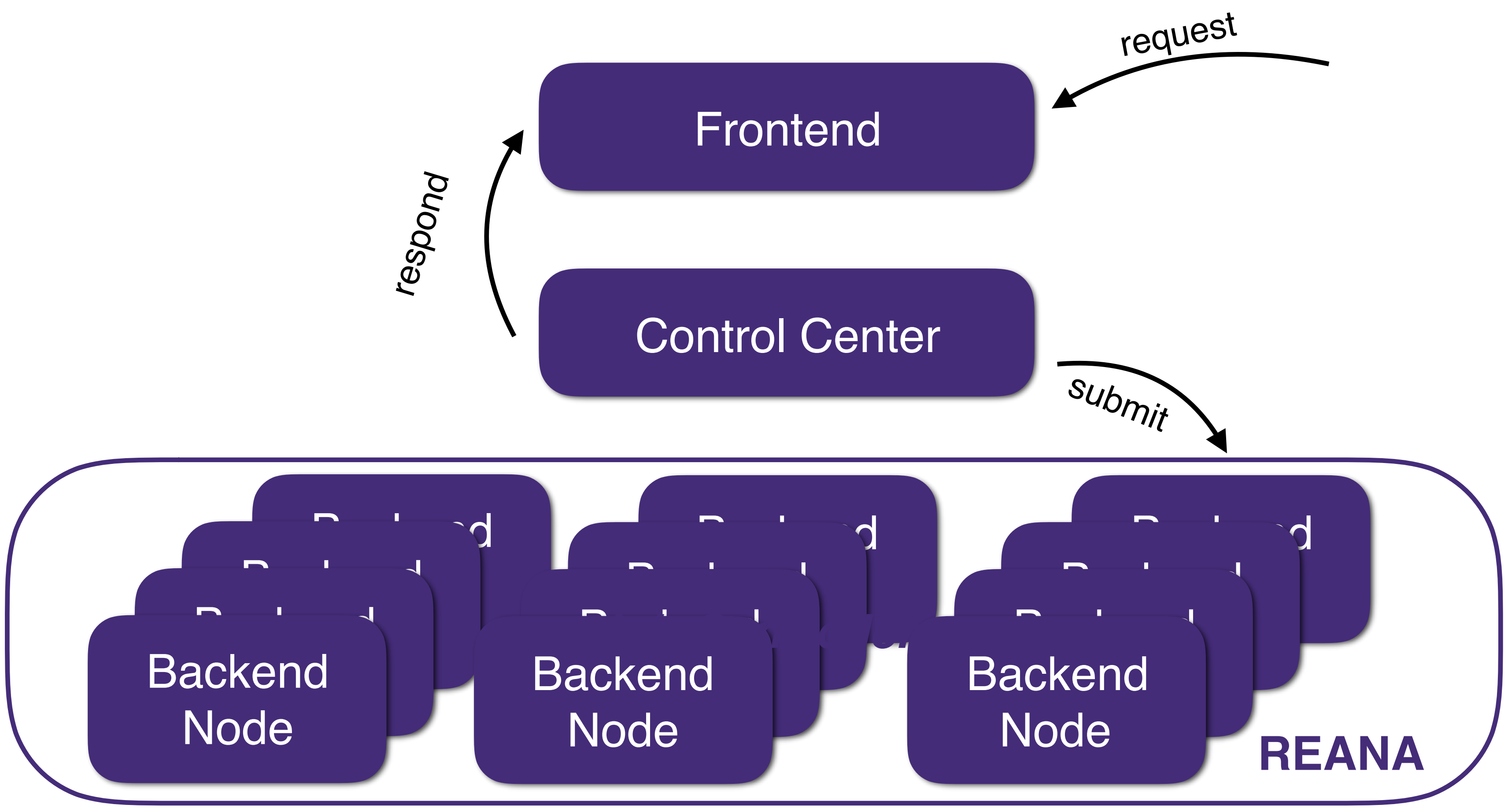
provide web-based interface to request and process reinterpretations powered by preserved analyses in CAP

- produce reinterpretations of same fidelity as original result (not just approximations)
- allow users to request new reinterpretations, provide parameter points, model data (SLHA files, etc.)
- control interface to manage and view reinterpretation, fulfillment progress, submission API
- Integration with publishing APIs (HepData, Inspire, etc.)

RECAST — Infrastucture

Frontend: register requests and analyses, manage request data, show reinterpretation results
Control Center: web interface to view and process requests, manage implementation catalogue.
Backend Cluster: REANA workflow service.

Deployed on CERN infrastructure
Eco-system of tools and libraries: REST API, python bindings, command line tools.



Importing a new analysis:

```
$> recast import_analysis from_web cds/1525880
```

import via API import from ArXiv, Inspire, CDS (for CONF notes)

Creating new Parameter Scans:

```
$> recast createscan scan.yml
```

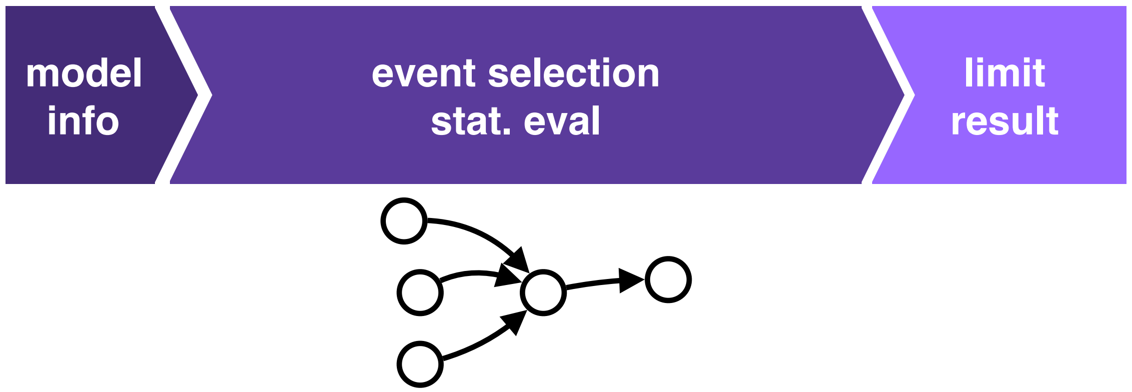
provide list of parameter points
and add concrete request data
as zip file for each point
(format specified by unique identifier)

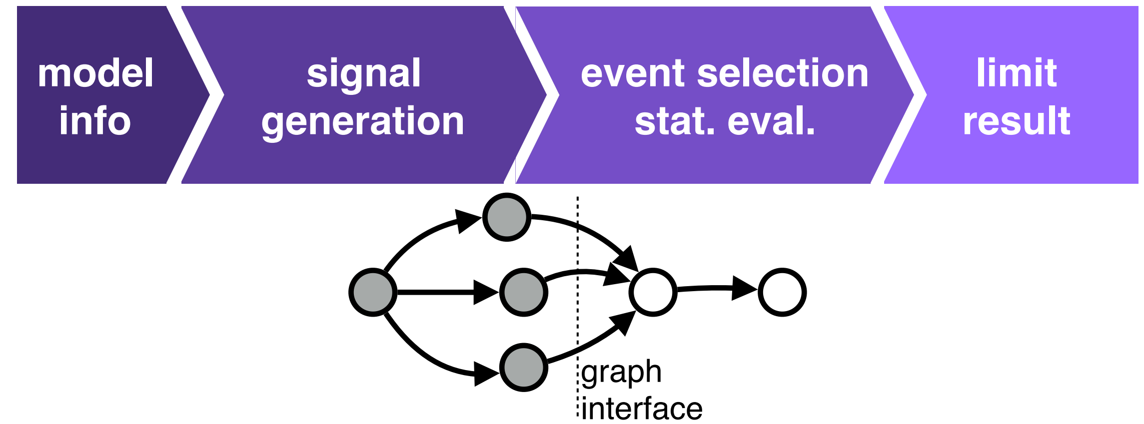

```
type: recast_scan
title: 'Validation of ATLAS-CONF-2013-024 in Neutralino/Stop Mass Plane'
pubkey: 'cds/1525880'
request_format: 'standard_format'
description: >
  this reinterpretation re-validates the original grid in of ATLAS-CONF-2013-024
reason: >
  The grid is useful to validate / check third-party
  (such as CheckMate) implementations of the ATLAS-CONF-2013-024 as it
additional_information: >
  the grid consists of 36 points in the stop / neutralino mass plane.
  The input parameters are given in the standard format consisting of
  model parameters and number of events.parameters: [mStop, mNeutralino]
parameters: [mStop, mNeutralino]
points:
- coordinates: [200.0, 0.0]
  data: data/200.0_0.0.zip
- coordinates: [300.0, 0.0]
  data: data/300.0_0.0.zip
- coordinates: [300.0, 100.0]
  data: data/300.0_100.0.zip
  ...
```

RECAST

Processing Requests using archived workflows:

if model info (request) data already in form that analysis workflow expects, can just process in a single-pass



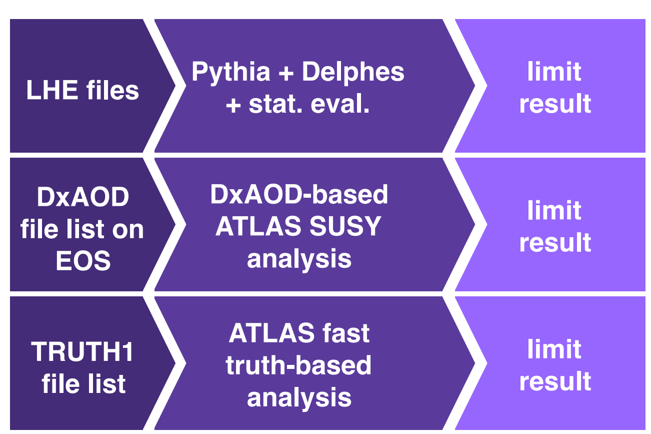


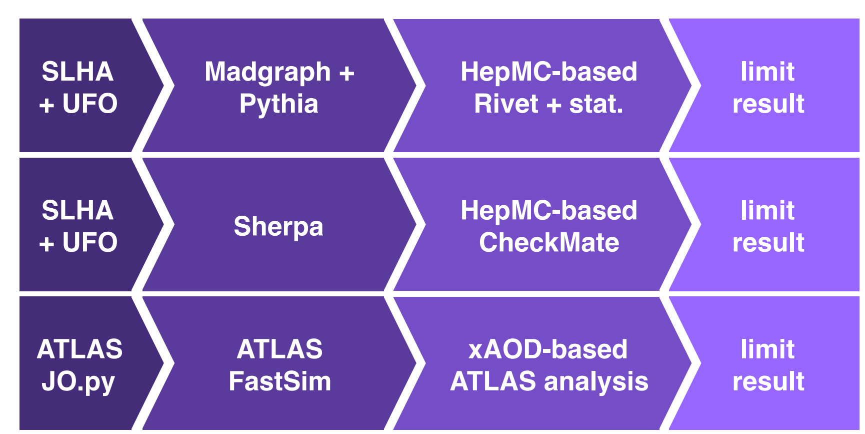
Otherwise, we can use *graph interfaces* between workflows to build more complex workflows by **mixing and matching** separately preserved workflows.

- possible, because yadage workflows are machine readable. can programmatically create new workflows.

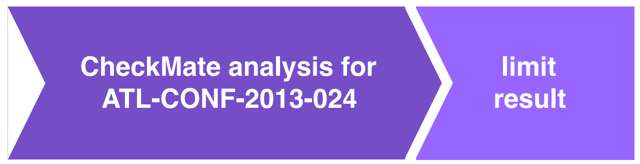
Allows us to build up a **comprehensive catalogue** of reinterpretation workflows quickly.

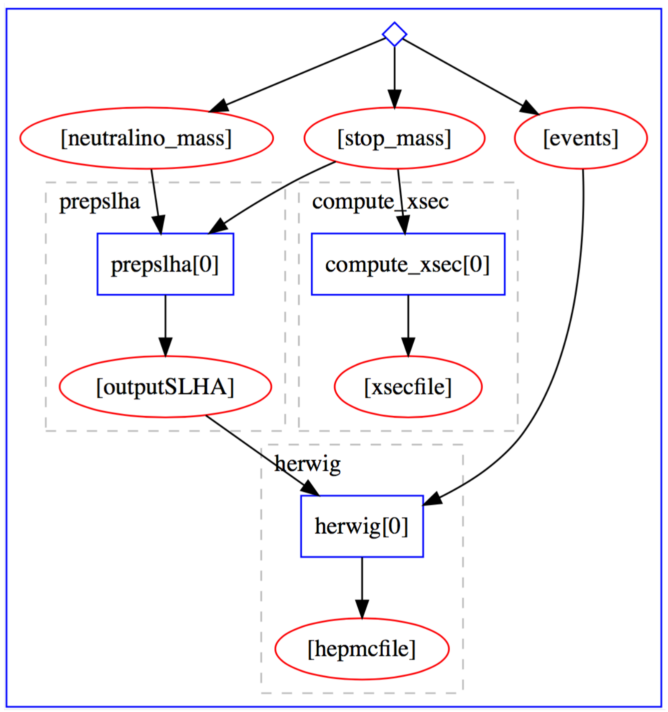
- match entire CheckMate/Rivet catalogues with generic HepMC producing workflows.
- only require analysis teams to capture their immediate workflow based on derived xAOD ntuples, upstream **ntuple production workflow can be handled separately.**



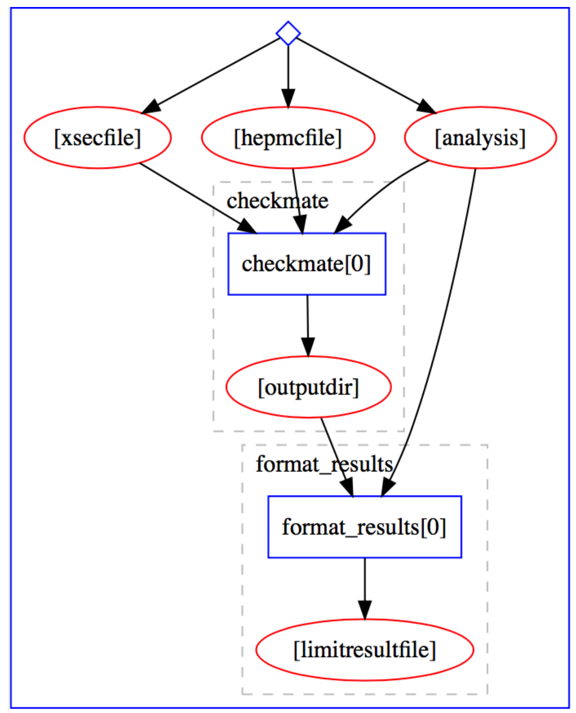


Example: Herwig upstream with CheckMate downstream

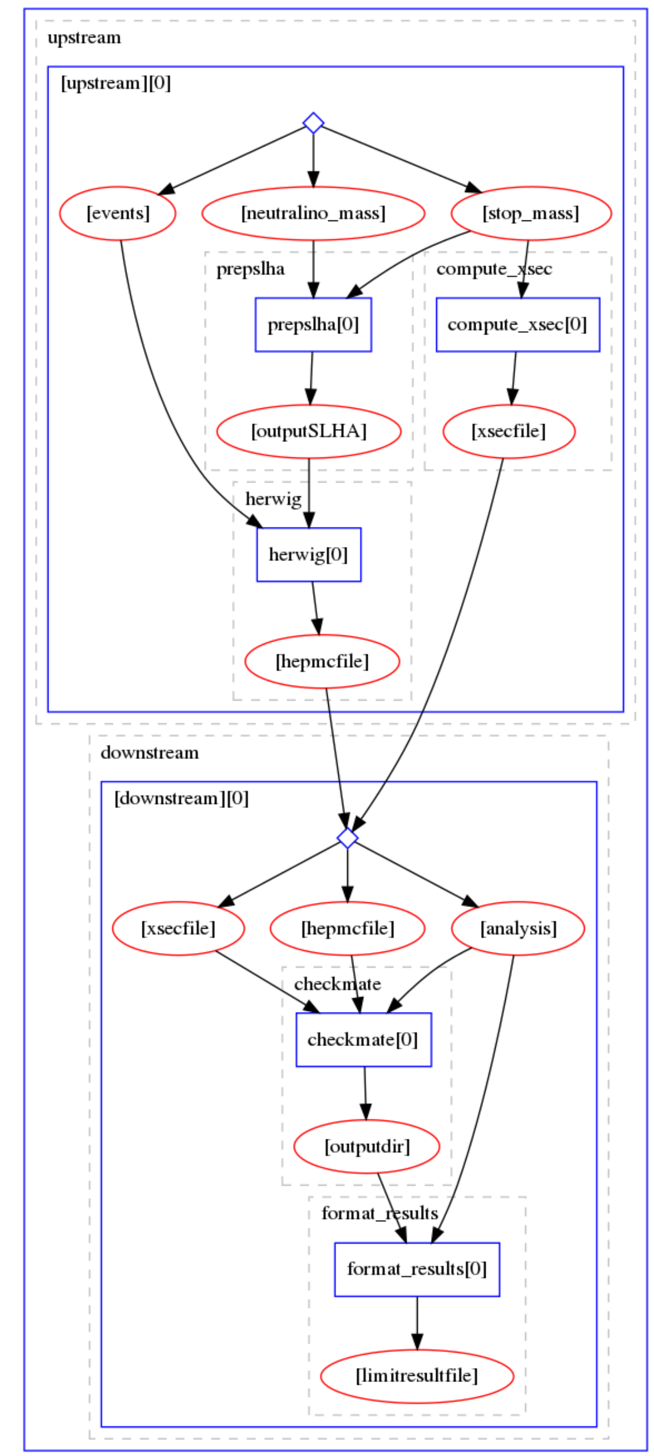




Herwig workflow



CheckMate workflow

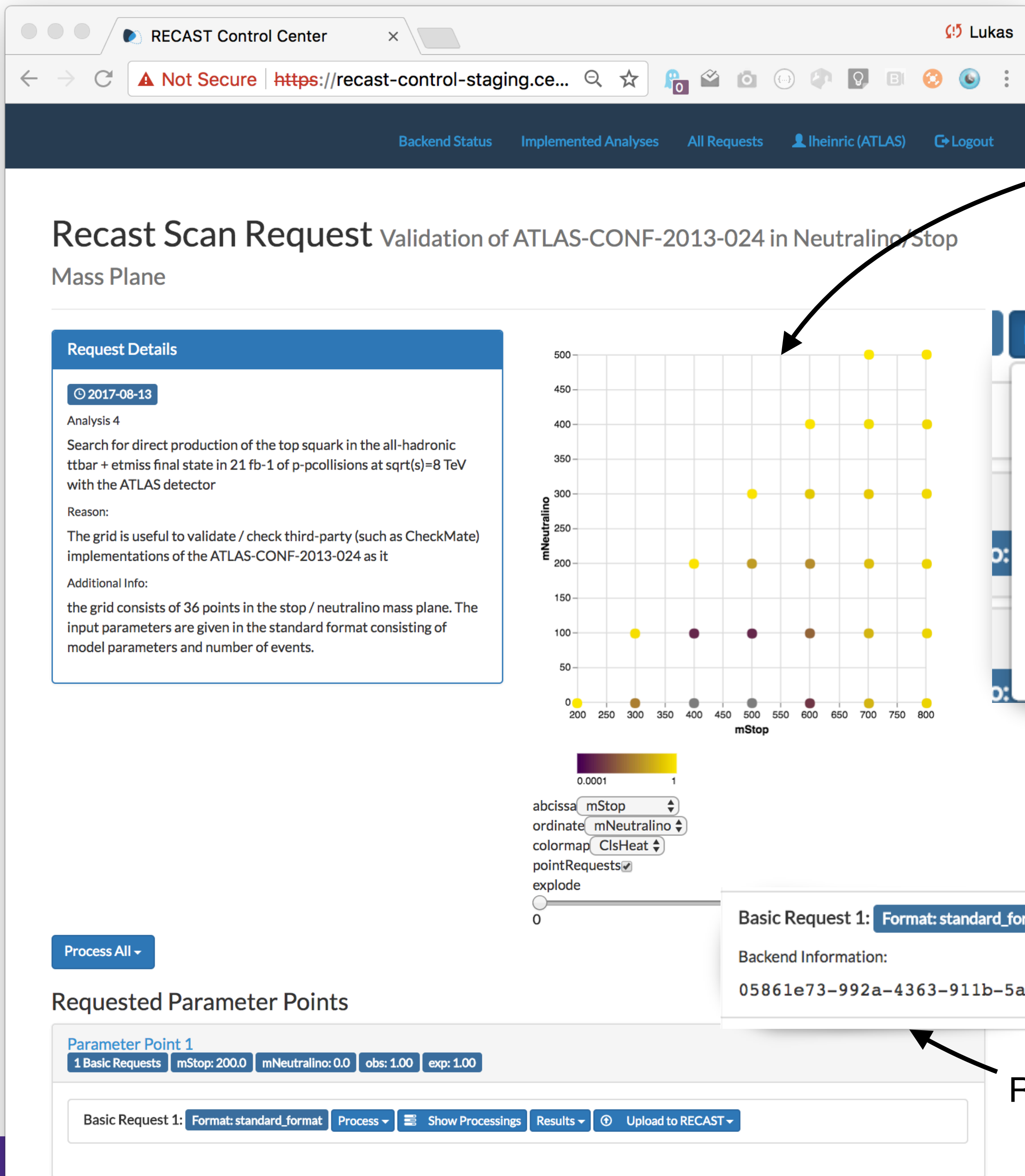


computed combined workflow

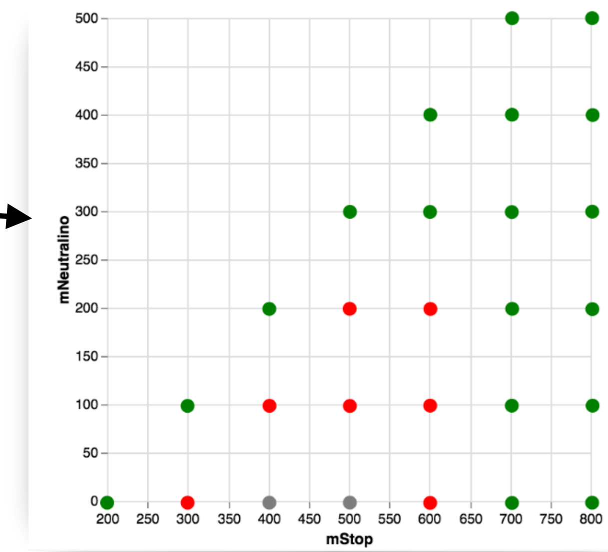
upstream and downstream workflow can be developed and maintained independently, on mutually exclusive software stack. Combination works on the semantic workflow level / between files on disk.

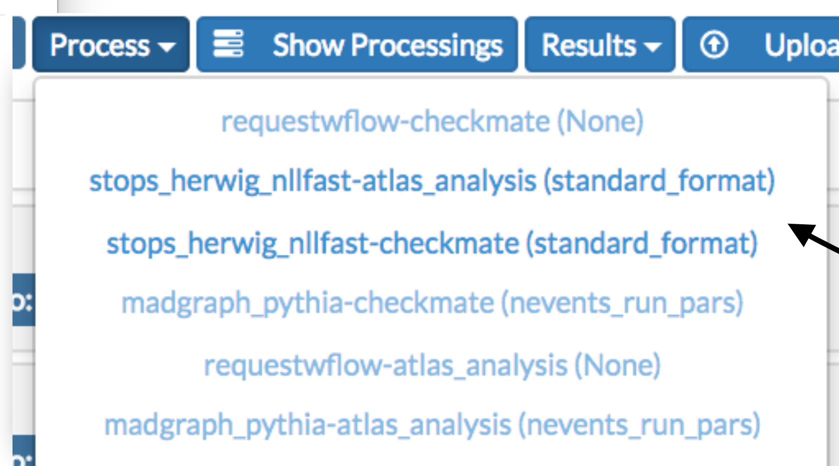
Processing Requests from the Control Center

web interface showing request details and result visualization. Offers all possible workflow configurations given request format and anlysis details.



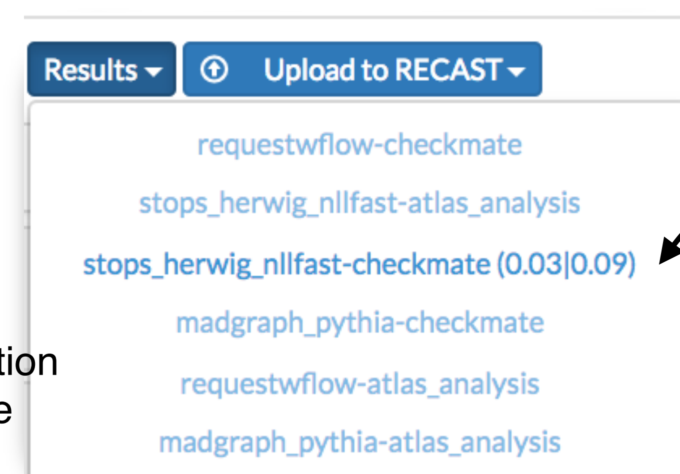
toggle scan scatter plot between Cls heat map and 0.05 exclusion threshold





possible processing workflows for this request format and analysis

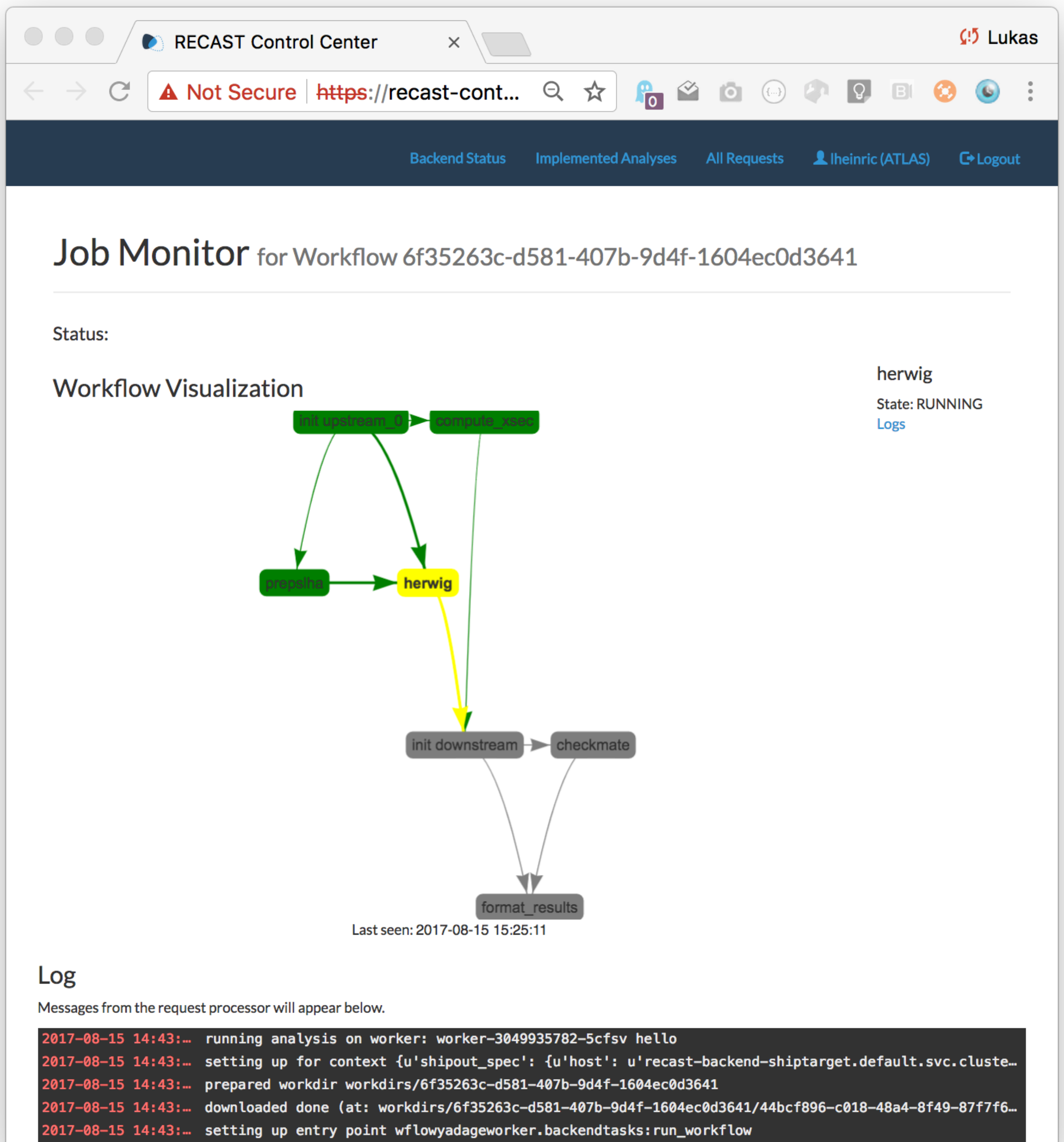
REANA backend information can retry jobs on failure

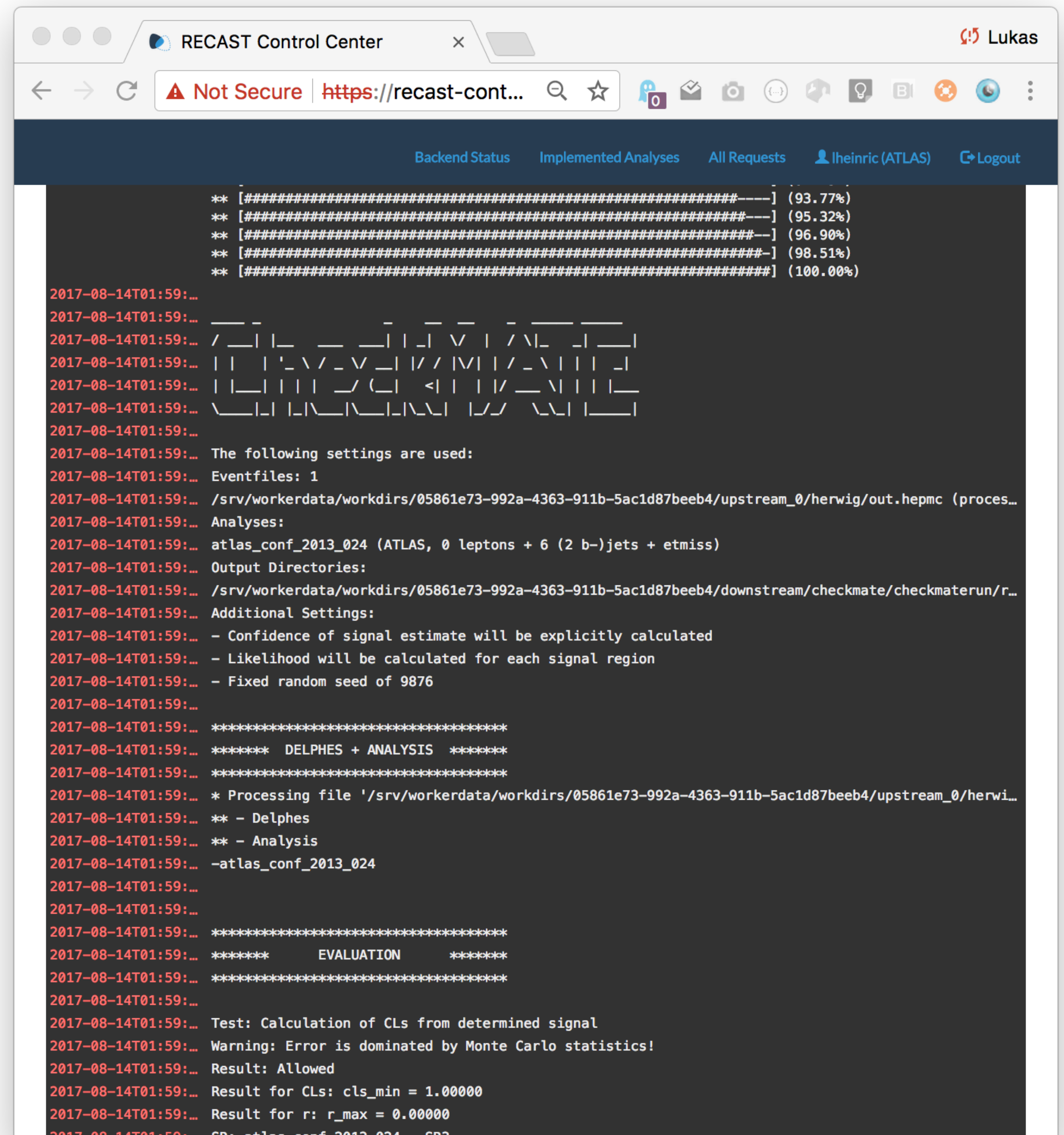


CLs obs|CLs exp result for this implementation

Live Monitoring Workflow execution

Once, submitted, web-based workflow monitoring — live streaming visualization and logging. Live node-level logging via LogStash (indexable via ElasticSeach for anomaly detection) — similar feel to Travis / GitLab CI job monitoring.





Within ATLAS we're working towards increasing implementation

- multiple examples of containerized workflows within SUSY and Exotics groups captured via Continuous Integration
- Using REANA prototype deployment to run validation and reinterpretation of these analyses. Cluster Size: O(1k) VCPUs
  - sufficient scale to run large number ntuple→limit reinterpretations per day.

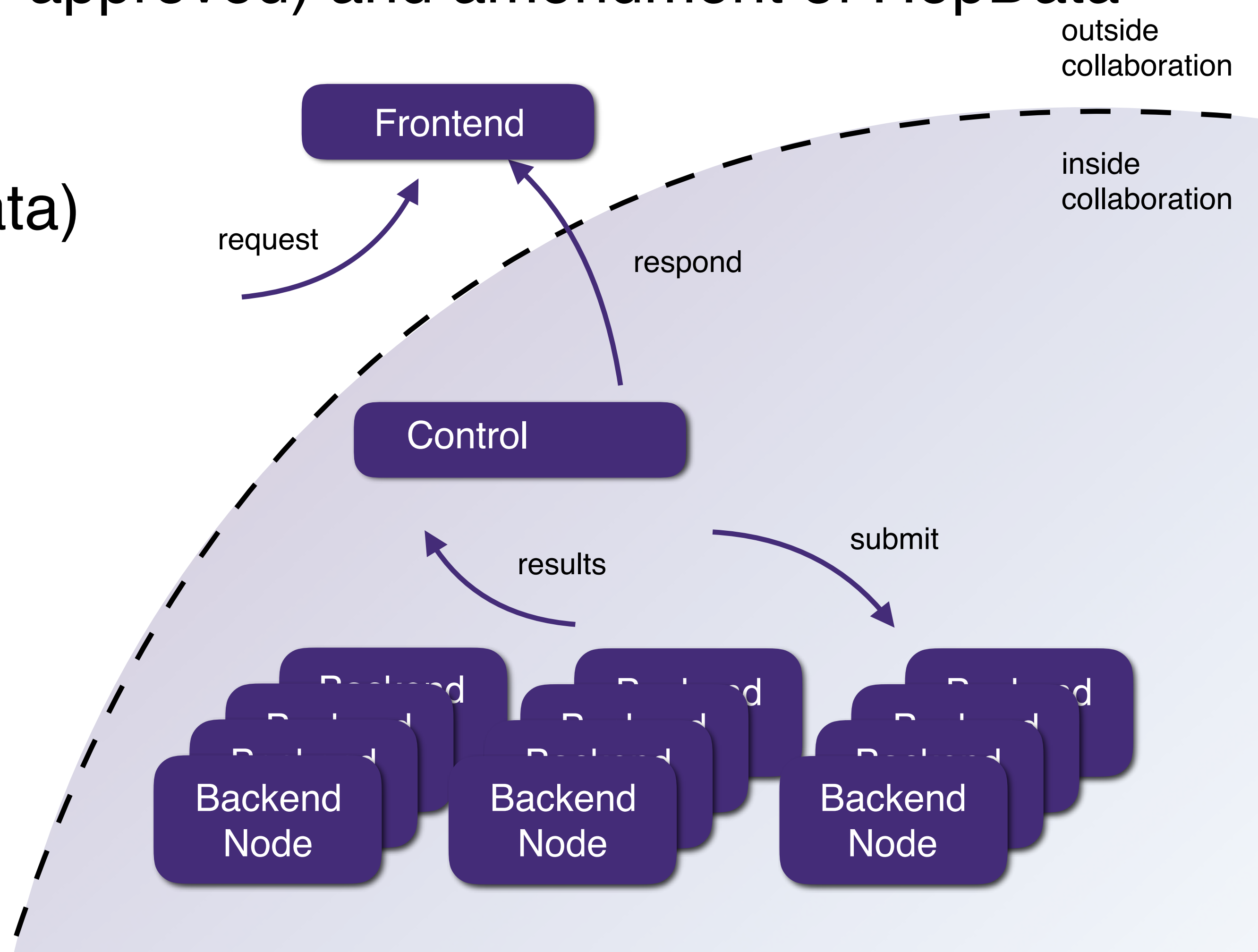
  - in principle full chain (incl simreco) scan possible on O(week)
- proof-of-concept combined upstream MC generation / downstream analysis workflow done

- for now request and processing privileges only ATLAS internal

Future Opportunities:

When implementation of analysis preservation in ATLAS advances and if it proves robust, there is an opportunity to open request privileges to pheno-community

- reinterpretations-as-a-service
- allow scan definition / suggestion by people outside of collaboration, provide e.g. parameter cards based on pre-selection based on external constraints (DM relic density, Higgs Mass, etc..)
- fast-track approval (analysis was already approved) and amendment of HepData records
- Track citation of request for theorist,
  of response for collaborations (citable data)



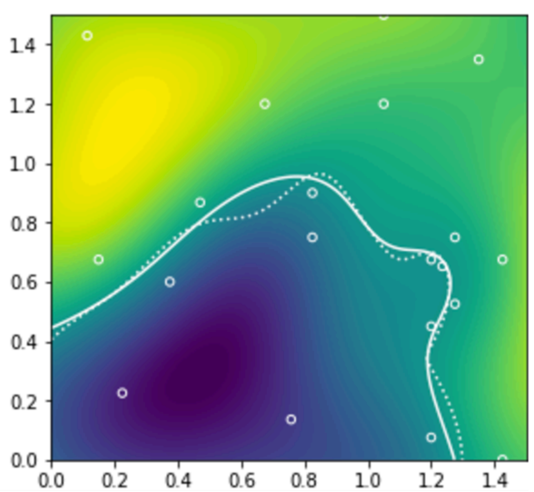
Smart Parameter Scans via Bayesian Optimization

reinterpretation of analysis wrt. to parametrized BSM model = the definition of an expensive multi-variate function. Cartesian grids suffer from curse of dimensionality, need to choose points wisely to find expulsion contour with minimum number of points

$$\mathrm{CL}_s = f(m_1, m_2, g_2, g_2, \cdots | \mathcal{D}_{13\mathrm{TeV}})$$

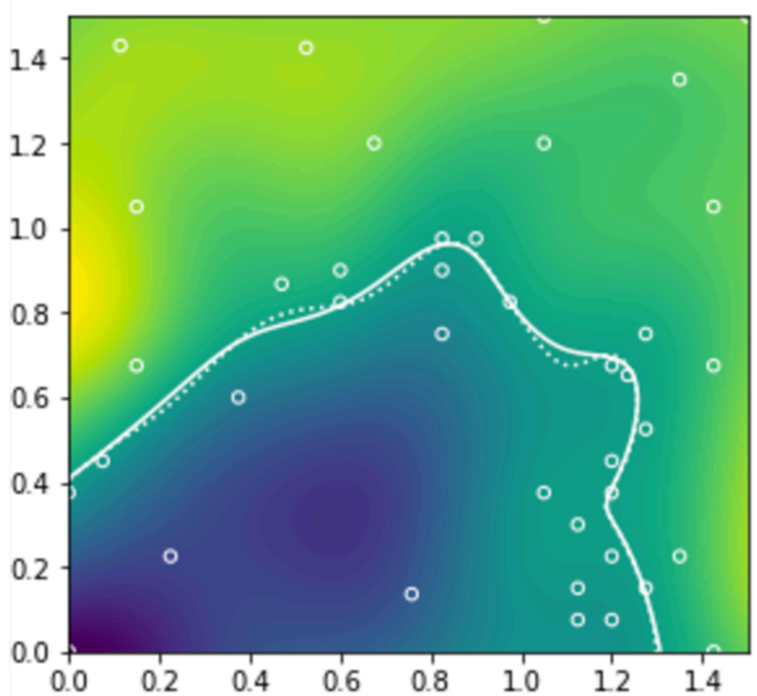
Idea: Bayesian Optimization with objective to minimize overall differential entropy of point classification p.d.f. (i.e. certainty about contour)

Can find much better contour interpolation with ***fewer*** *(but iterative)* simulated points.

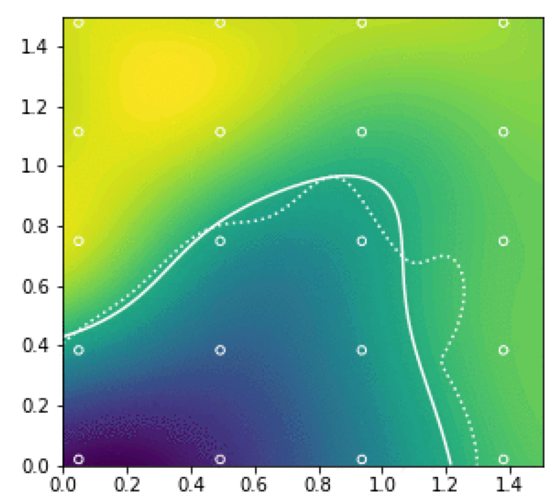


20 points (4 random+ 16 BO)
basic shape already well-known

animated gif



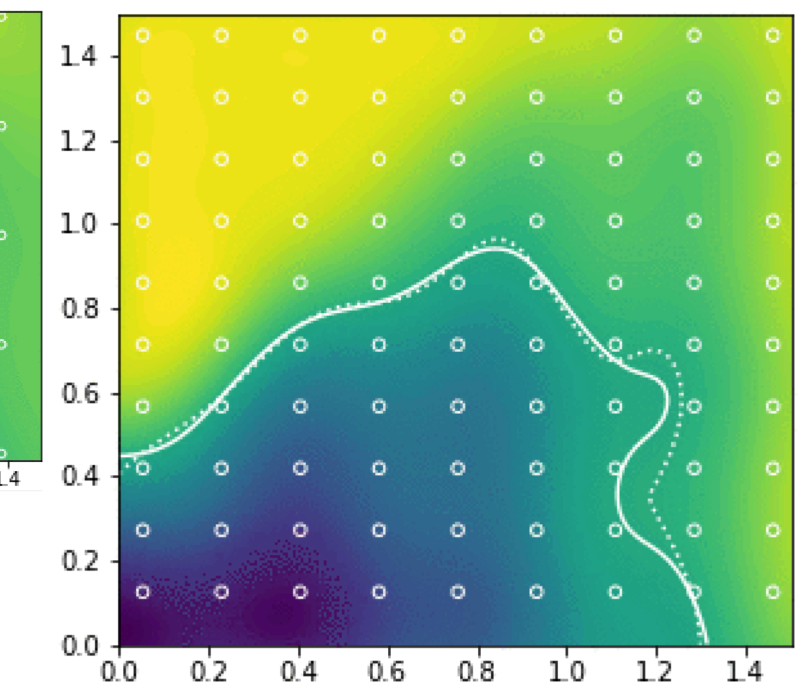
40 points
(mostly detailed refinement)



20 points (5x4) grid

even with dense grid
key contour features
may be missed

animated gif



90 points in 9x10 grid

- ATLAS analysis preservation focuses on analysis re-usability via containerized declaratively defined analysis workflows
- push to minimize workload on analysis team to capture analysis
  - capture during development
  - capture only code analysis-related workflow
  - easy human-readable text-based workflow capture
- Close collaboration with CAP and REANA efforts to drive analysis re-use applications
  - native cloud-based solution on modern infrastructure.
- First application: collaboration internal re-interpretation of SUSY and Exotics analyses
- Exciting Opportunities on the horizon for more systematic approach to reinterpretation