LHCb Analysis Preservation Roadmap

Sebastian Neubert

Uni Heidelberg

9th LHCb Computing Workshop, May 2017

Overview over roadmap document

Draft available on authorea:

https://goo.gl/ngAzhn

- Part I lays out scope of the problem and the LHCb philosophy on AP
- Shows pragmatic solutions
- Addresses PWGs, analysts and computing team
- Part II with recommendations on tools, technologies and tutorials will be written until the end of the year

Table of contents:

Motivation for Analysis Preservation

Reproducibility is a fundamental scientific requirement

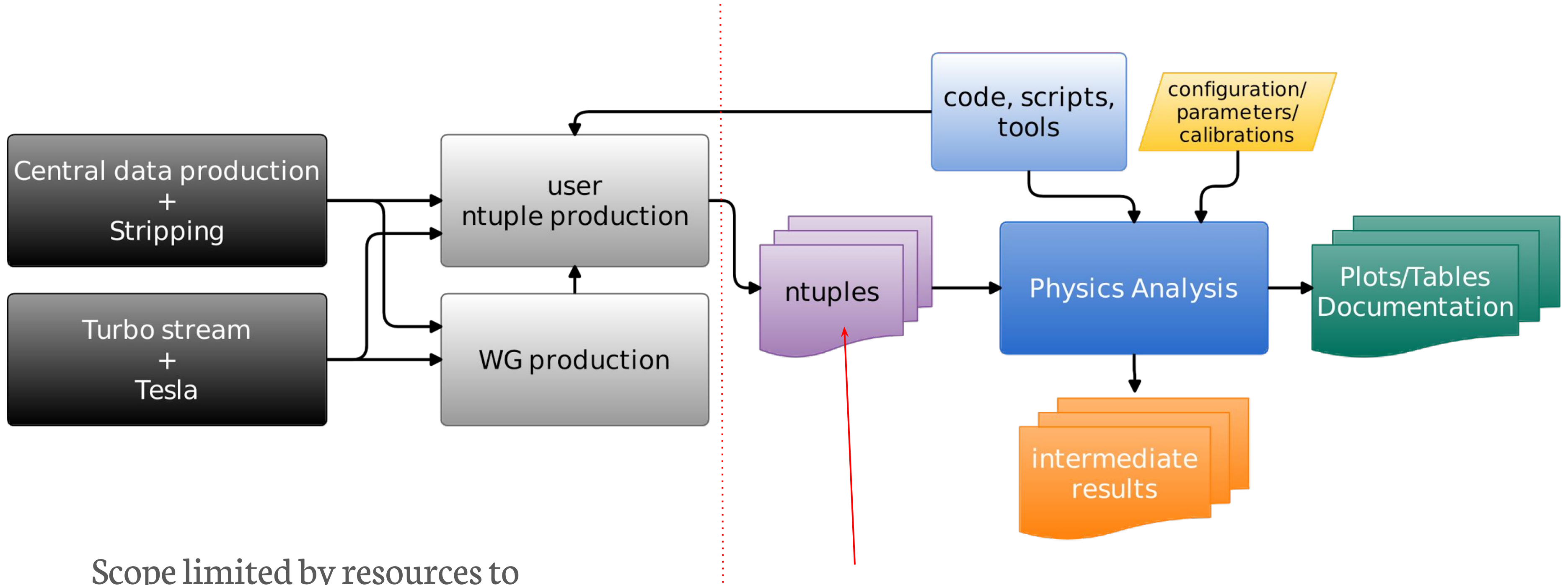
HEP has special responsibilities, due to large/long term projects

HEP AP addresses several problems of knowledge transfer:

- Collaborative working
- Knowledge preservation during review
- Knowledge transfer to other analysis teams
- Knowledge transfer to future generations

Complex analyses -> analysis reproducibility = code & data preservation

Scope of Analysis Preservation in LHCb



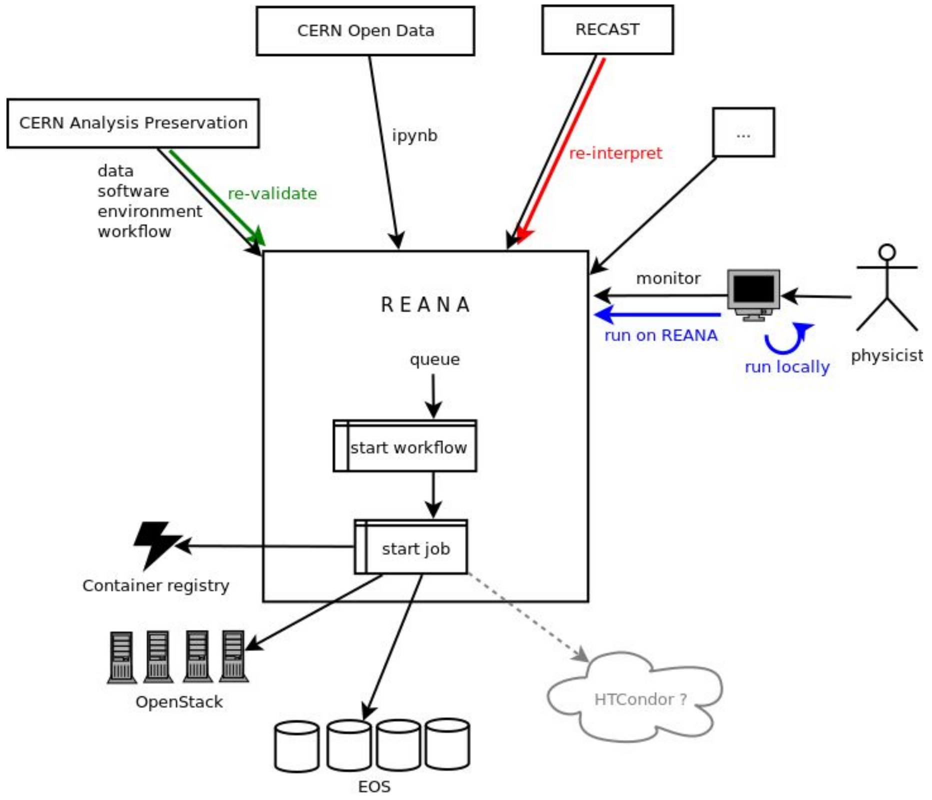
Scope limited by resources to reproduce input data (MC/reco/stripping)

ntuples provide natural interface point

Preserving = Re-running

Things we are supposed to deliver to CAP/REANA

- Metadata
  - See LHCb working-group database
  - Can be used to store information on
    - Central production steps
    - Gaus/Brunel/DaVinci... versions
- Input ntuples
  - Currently 1TB per analysis planned
- Full analysis code
  - In gitlab repo
  - Including workflow description
- Container image
  - Contains software env
  - For running analysis in REANA



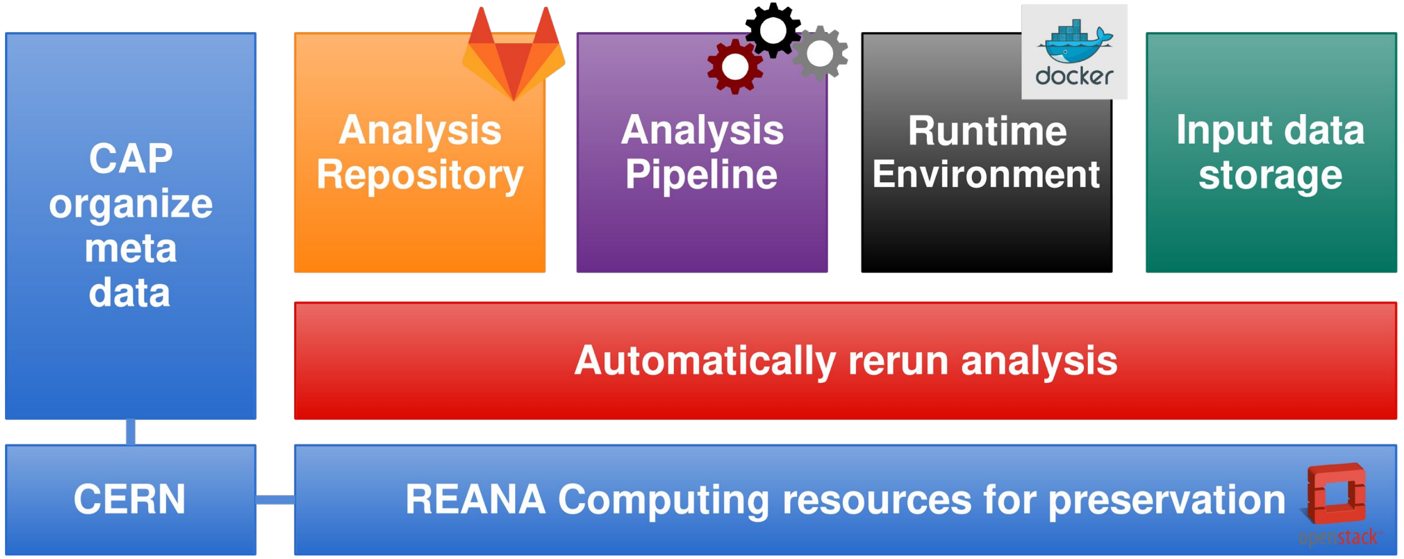
Analysis Preservation for the People

AP needs to be useful during everyday analysis work

- Provide analysts freedom to design their analysis and choose the best tools for the job
- Corollary: linux shell is what we assume as smallest common denominator
- Tools for preservation have to help already during design, implementation and review

AP practices most effective when they are adopted early in the analysis development

Four Domains: Modular Analysis Preservation



- AP can be adopted step-by-step
- Synergies between the domains

Domain	Minimal Recommendation	Needed for CAP	Best Practices
Analysis Repo	Complete analysis code is on gitlab.cern.ch		Project is hosted in WG gitlab-group
			link modules into one master repo e.g. using git submodule mechanism
			use fork/merge development model
			use a separate repo for ANA and results
Analysis Pipeline	Full instructions how to run	Script to automatically run the complete analysis	Use a dedicated pipeline tool (recommendations in part II)
	analysis in ANA note	Analysis can be run in batch mode without UI support	Use a gitlab-ci server to re-run analysis on change
		Every plot//table is produced without further human intervention	Use a dependency-checking pipeline, which only recomputes necessary steps
			Use the same pipeline tool for local and on-the-cloud running
Runtime Environment	Full instructions how to install	Docker image containing everything the analysis needs	Use provided LHCb docker images as base
	all dependencies in ANA note	Dockerfile committed to analysis repo	add custom software on top of base image
		Docker image hosted on gitlab registry at CERN	use package manager (i.e. conda) to manage software inside your docker image
			use same docker-image on your continuous integration server
Input data	All input files are kept on eos, readable to collaboration		Input files are kept in WG directory on eos

Analysis Repository

on gitlab: repo = project

Goals:

- Preserve analysis tools and logic
- Facilitate collaboration
- Enable reuse of tools

The repo really should be central point:

- Version control
- Collaborative tool
- Documentation/TWiki
- CI server control

Minimal recommendation:

- complete* analysis code on gitlab.cern.ch

Best practices:

- physics-WG gitlab-group
- fork&merge workflow
- analysis can be modularized
  - one master repo
- use a separate repo for results and ANA
  - Will make it easier to setup continuous integration

* producing all plots/tables in the ANA

Sidenote: modularizing projects on gitlab

- Teams might split responsibilities for different parts of the analysis
- Tools can be shared between several analyses

Recommendation:

- One master repo
- include modules into the master
  - As submodules
  - Or subtrees

Git offers two mechanisms to handle this:

- Git submodule
  - Creates a “pointer” to another repo inside your repository
- Git subtree
  - A set of script to merge another repo into your project as a subdirectory

Huge internet debate which one is preferable: choose yourself, both are working solutions

Nice article here

Analysis Pipeline

Only those parts of the analysis, which are automated can be preserved

- **Automation is machine-readable documentation**
- **Exact definition of analysis flow**
- **Enables automatic testing**
  - **“nightlies”**

Ability to consistently rerun the analysis, significantly lowers barrier to implement reviewer’s requests

Minimal recommendation:

- Section in ANA note describing how to run the complete analysis

Best practices:

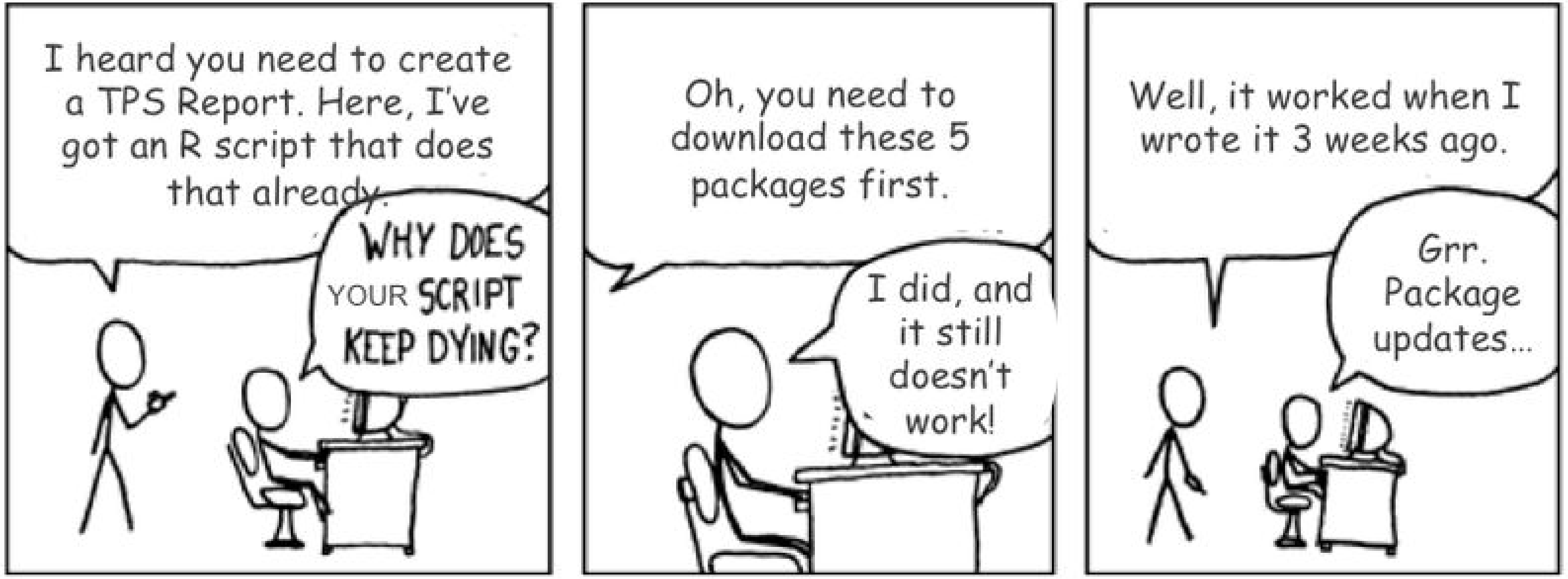
- Analysis completely scripted
  - No UI, no manual settings, no copy-paste
- Use a dependency-checking pipeline tool
- Use a continuous-integration server
- Start using a pipeline early in the analysis design to profit during review

Pipeline tools under evaluation

See Chris' talk in the afternoon

Tool	Install	Doc	Key Features	Missing Features	Complexity	Community support
Snakemake	pip3 instal or conda, needs python 3	Very good	Python in pipeline specs, can execute shell commands, and python functions Xrootd support	Dependency tracking on git, python3 may be a problem	medium	good
Yadage	Pip install	Ok	Dynamic DAG topologies, submits to Kubernetes cluster	Caching	medium	Small com But: CERN
Luigi	pip install	Excellent	Inherit from Python-classes; Full python-power	Caching, Dependencies hard coded (see sciluigi for fix)	high	Very good
Fabricate	One file pip install	Ok	Automatic dependency discovery	??	low?	Small com
CWLtool	pip install	Good	Common workflow language, several implementations are available	??	medium	good
GNU Make	yum -y install make	Excellent	Well known	Wicked syntax, Problematic if a job crashes (can leave corrupt data files)	Low - high	Excellent

Runtime Environment


I heard you need to create a TPS Report. Here, I've got an R script that does that already.
WHY DOES YOUR SCRIPT KEEP DYING?
Oh, you need to download these 5 packages first.
I did, and it still doesn't work!
Well, it worked when I wrote it 3 weeks ago.
Grr. Package updates...

Runtime Environment

Re-running the analysis requires the full runtime environment

- Requirement: Allow analysts to use all tools they need
- Linux containers provide technology to capture runtime-env
- Capture both an image of the env and rules how to recreate it
- REANA will use cloud infrastructure and accept containerized analyses

Minimal recommendation:

- Provide ANA with installation instructions for all dependencies

Best practices:

- Use base images provided by collaboration
- Image contains everything needed
  - CVMFs mount not useable for AP!
- Use package manager inside container
- Dockerfile kept in analysis repository
- Container image on gitlab registry

Analysis software stack

Basic container images should be provided by collaboration

- ROOT, roofit, roostats, xrootd
- Anaconda (scipy, numpy...)

- Lightweight
  - Not the full LHCb stack
- Mounting CVMFs to make software available defeats purpose of AP

Additional specialized tools installed by analysis team

- Docker allows to "inherit" from base images
- Need to provide definite versions
  - Proper Dockerfile
  - Rebuilding the container needs to guarantee same versions
- Prepuild images will be ingested and preserved by CAP
- Analysis team / WGs responsible for custom add-ons

Input data

- Ntuples
  - Or alternative input (readable without LHCb stack!)
- Calibration data
  - e.g. PIDCalib output
- Parameters
  - e.g. TMVA Classifier XMLs

Need to manage access credentials

See talk by Luca&Massimo

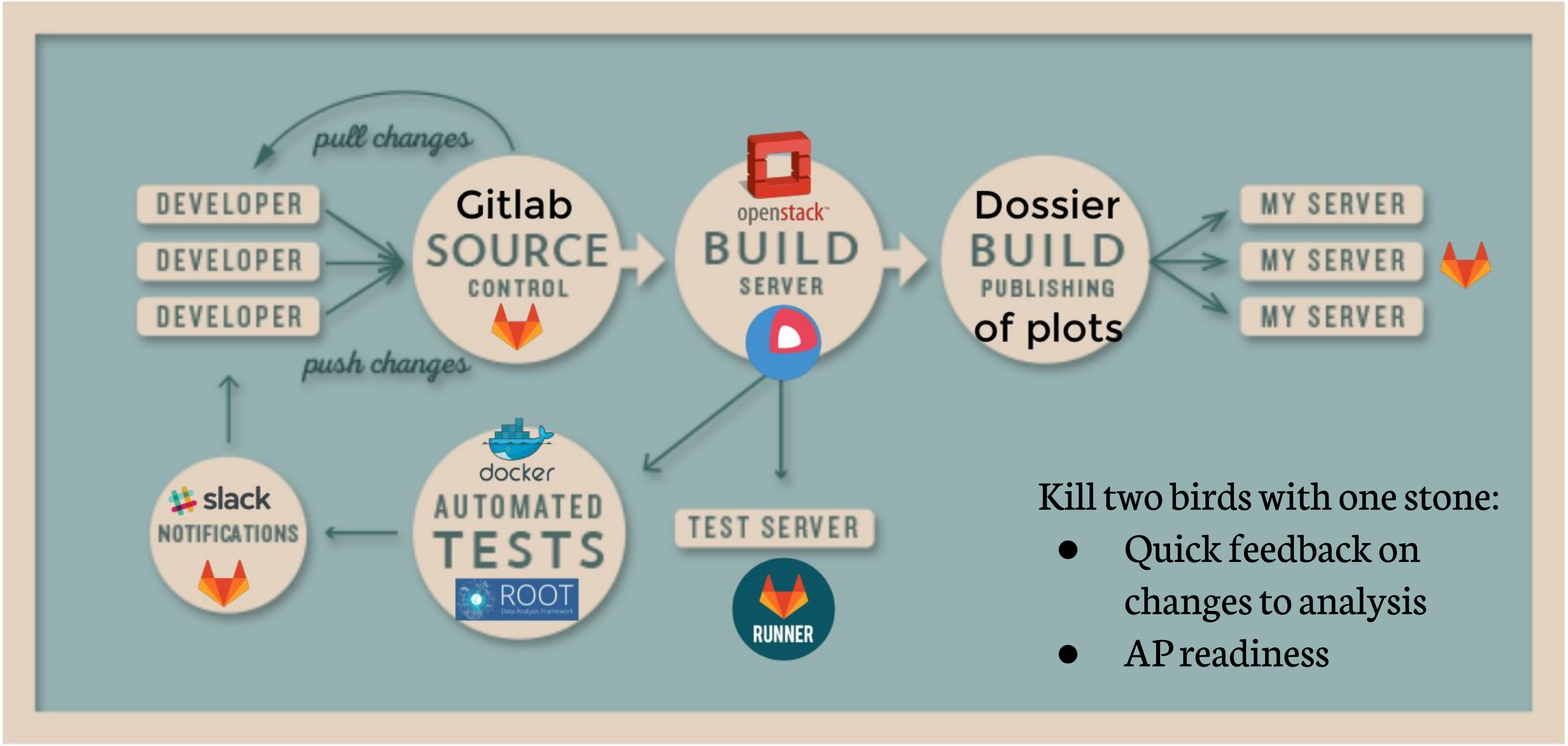
Minimal recommendation:

- All input data stored on EOS

Best practices:

- Use working group directories
- Calculate as much as possible in the analysis pipeline
- Use caching of intermediate results
- Expensive intermediate results can be treated as input-data

Continuous Integration: Always have a running analysis

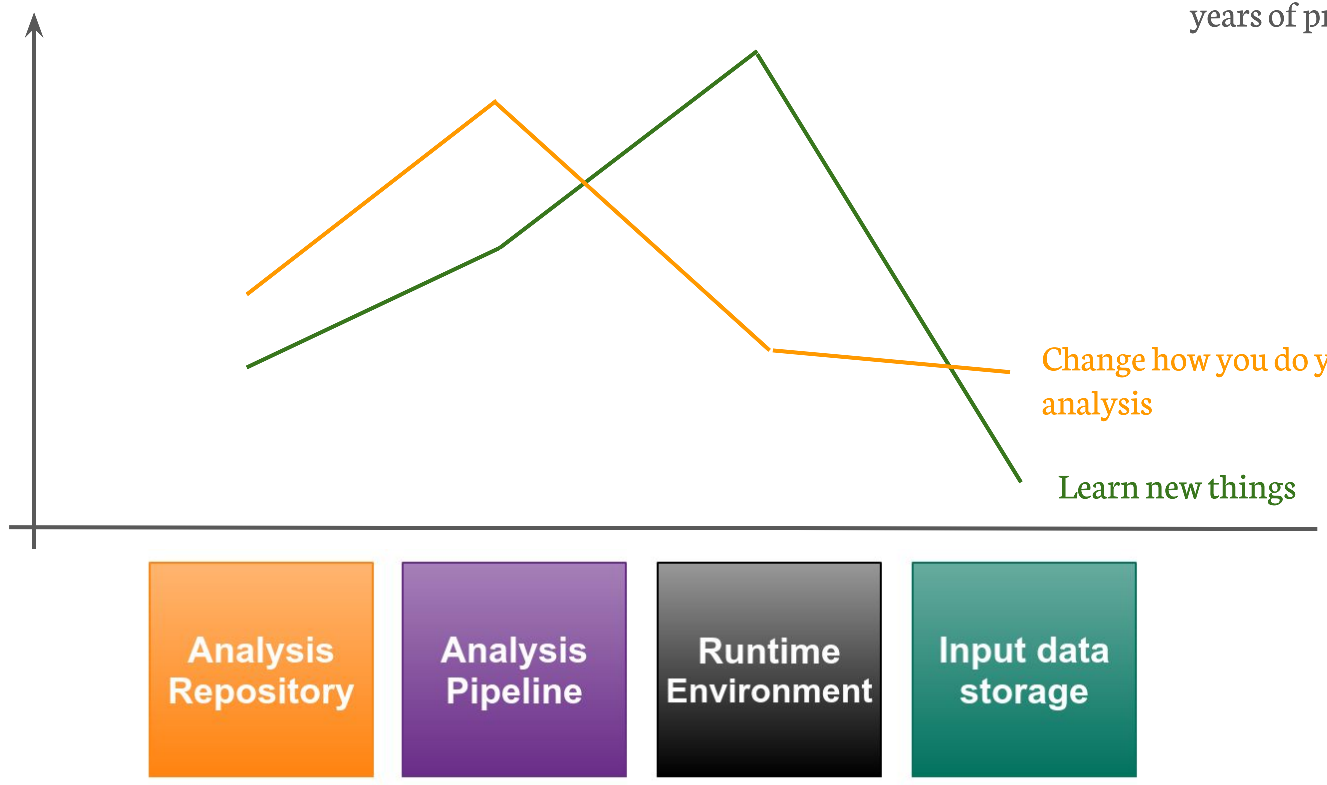


Kill two birds with one stone:

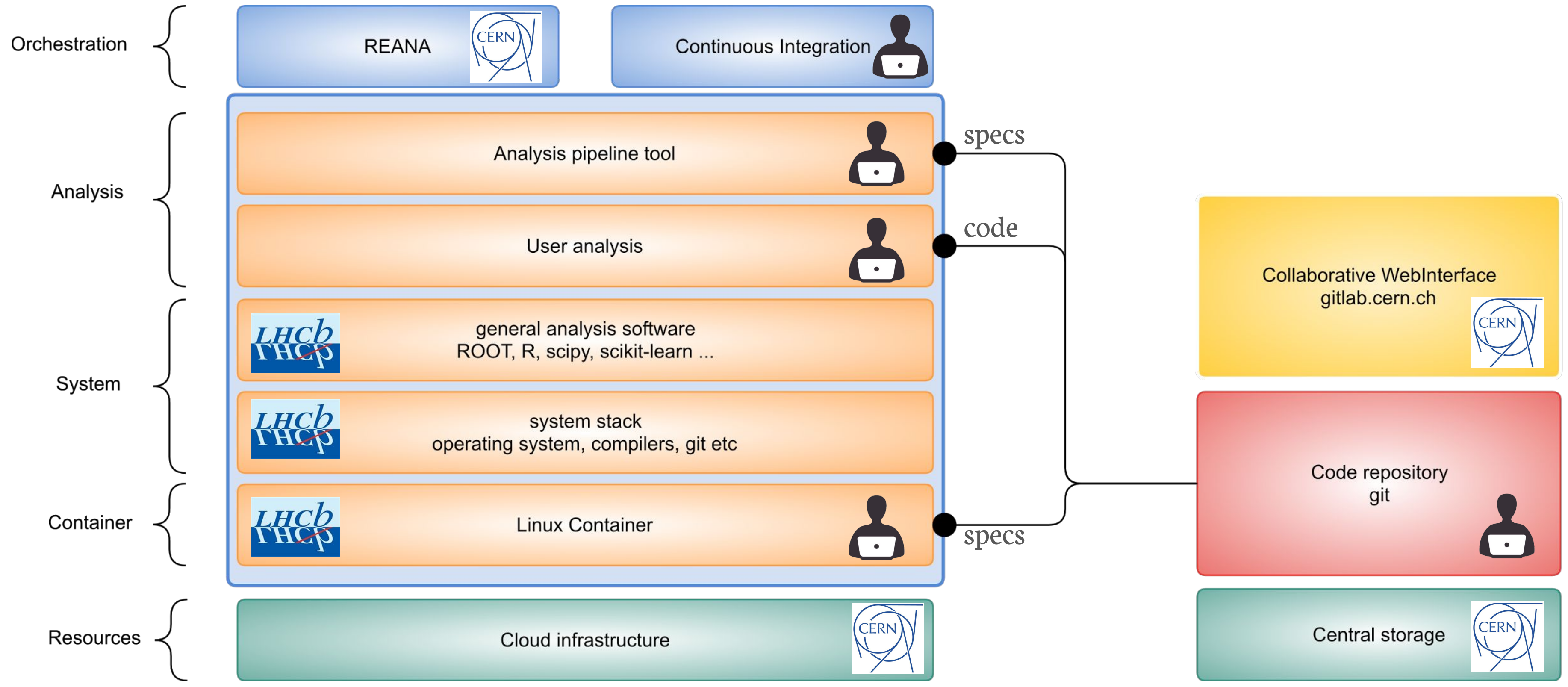
- Quick feedback on changes to analysis
- AP readiness

Required change in user behaviour

based on gut-feeling from 2 years of prototyping



An AP Stack


Orchestration
REANA
CERN
Continuous Integration
Analysis
Analysis pipeline tool
User analysis
System
general analysis software
ROOT, R, scipy, scikit-learn ...
system stack
operating system, compilers, git etc
Container
Linux Container
Resources
Cloud infrastructure
CERN
specs
code
specs
Collaborative WebInterface
gitlab.cern.ch
CERN
Code repository
git
Central storage
CERN

Conclusion

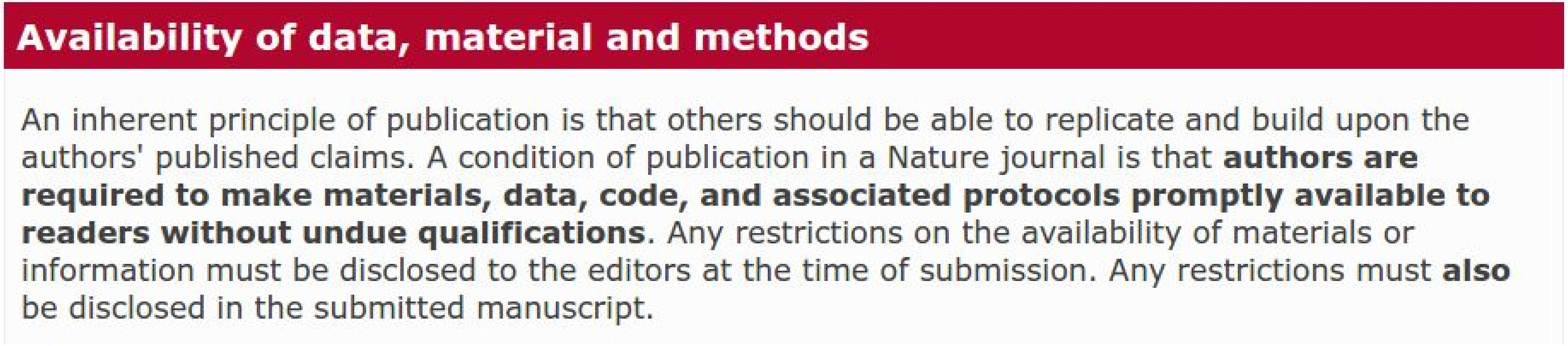
- **Four domains of analysis preservation**
  - Modular approach
  - Adapting to analysts needs
- **Best practices, which can be implemented now**
  - Protocol > technology
  - Practices help with everyday analysis business
- **Technology recommendations / tutorials will be provided in Part II (Fall 2017)**

Gitlab repository as analysis portal

- Collaborative tools include
  - nice README using Markdown
  - Webpage
  - Issue-tracker (JIRA or lightweigt)
  - In-line commenting
  - Merge tool
  - CI server interface
  - Container registry
  - Access control via egroups
- CAP will be able to ingest information from gitlab

Analysis Preservation and Publication Policies

Nature requirements:



Availability of data, material and methods

An inherent principle of publication is that others should be able to replicate and build upon the authors' published claims. A condition of publication in a Nature journal is that **authors are required to make materials, data, code, and associated protocols promptly available to readers without undue qualifications**. Any restrictions on the availability of materials or information must be disclosed to the editors at the time of submission. Any restrictions must **also** be disclosed in the submitted manuscript.