A Validation Framework for the Long Term Preservation of High Energy Physics Data

Securing HEP data for future analysis

DPHEP

LOPS @ ICDE
March 31st, 2014
Holiday Inn Chicago Mart Plaza

David South (DESY), Dmitri Ozerov (DESY)
on behalf of the DESY-DPHEP Group
arXiv:1310.7814

HELMHOLTZ | ASSOCIATION · DESY IT · ZEUS · DESY LIBRARIES INSPIRE · H1 · hermes · DESY

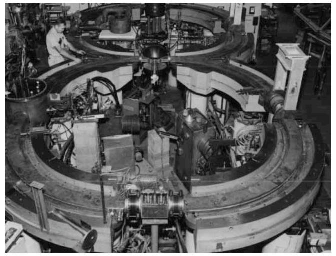
Experimental particle physics in the collider era

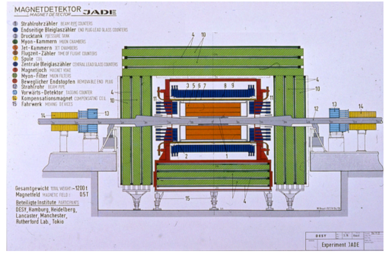
- A wide variety of physics results from many, often very different experiments

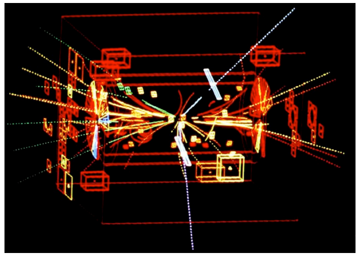
- Energy frontier probed with increasingly complex accelerator installations
  - From single room colliders in the late 1950s to installations measured in kilometres
  - Results from newer experiments typically, but not always, supersede those of similar older ones

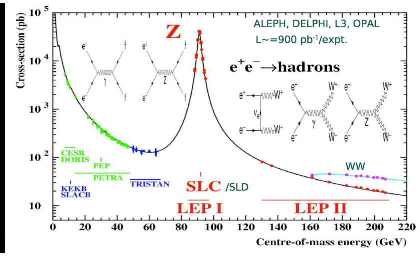
- Growth in size of the international collaborations, increase in the diversity of the data management

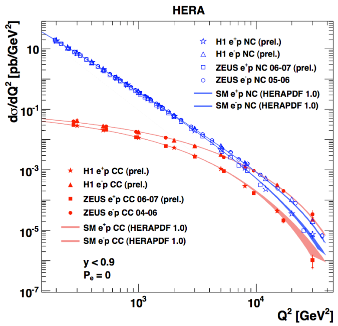
- We are now in the age of the Higgs boson and the LHC, with Petabytes of data
  - Belle 2, HL-LHC, and other projects such as the ILC or the next e-p/A collider still to come

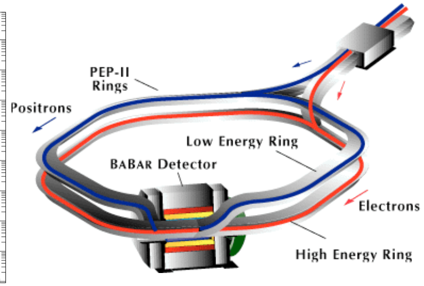


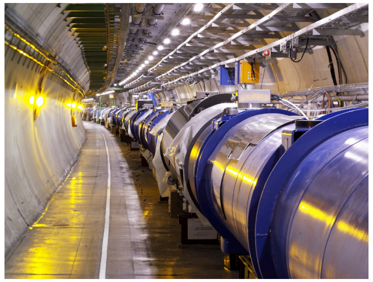






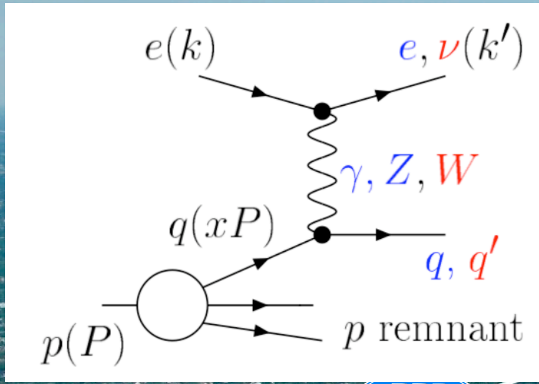


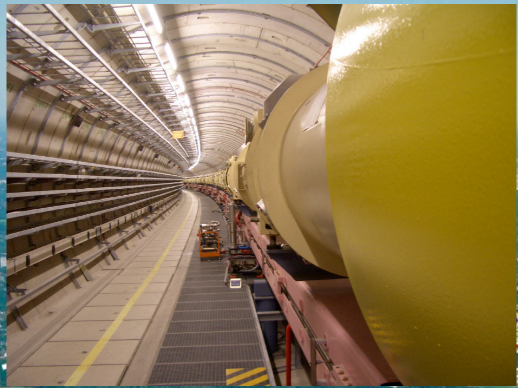


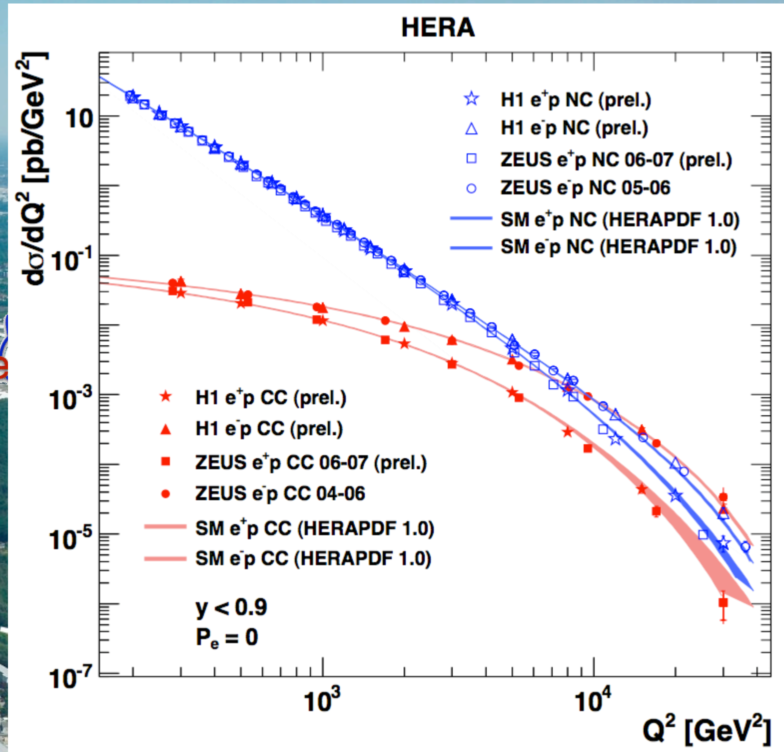


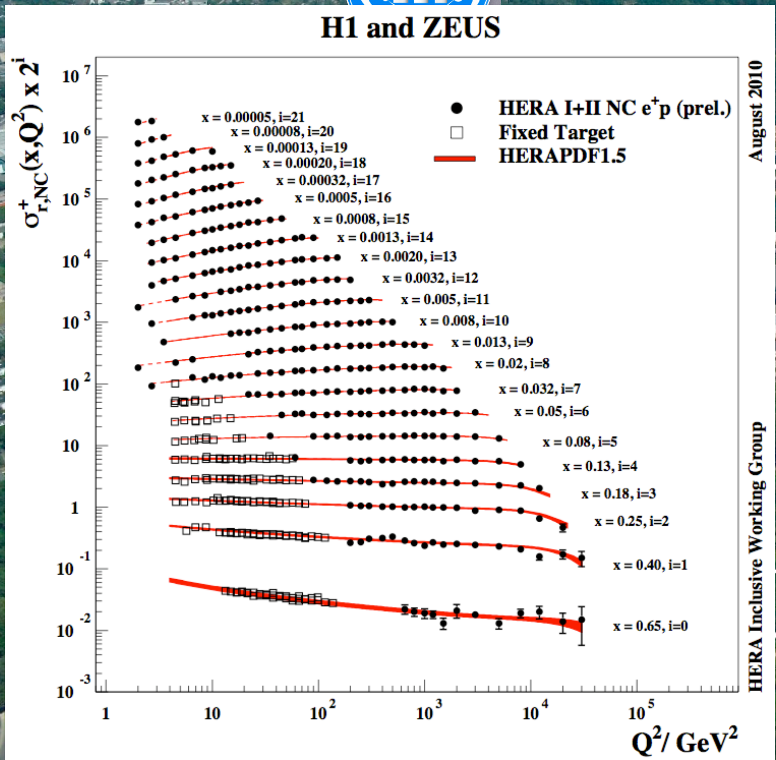
1992: Hadron-Electron Ring Accelerator (HERA) @ DESY

1992: Hadron-Electron Ring Accelerator (HERA) @ DESY



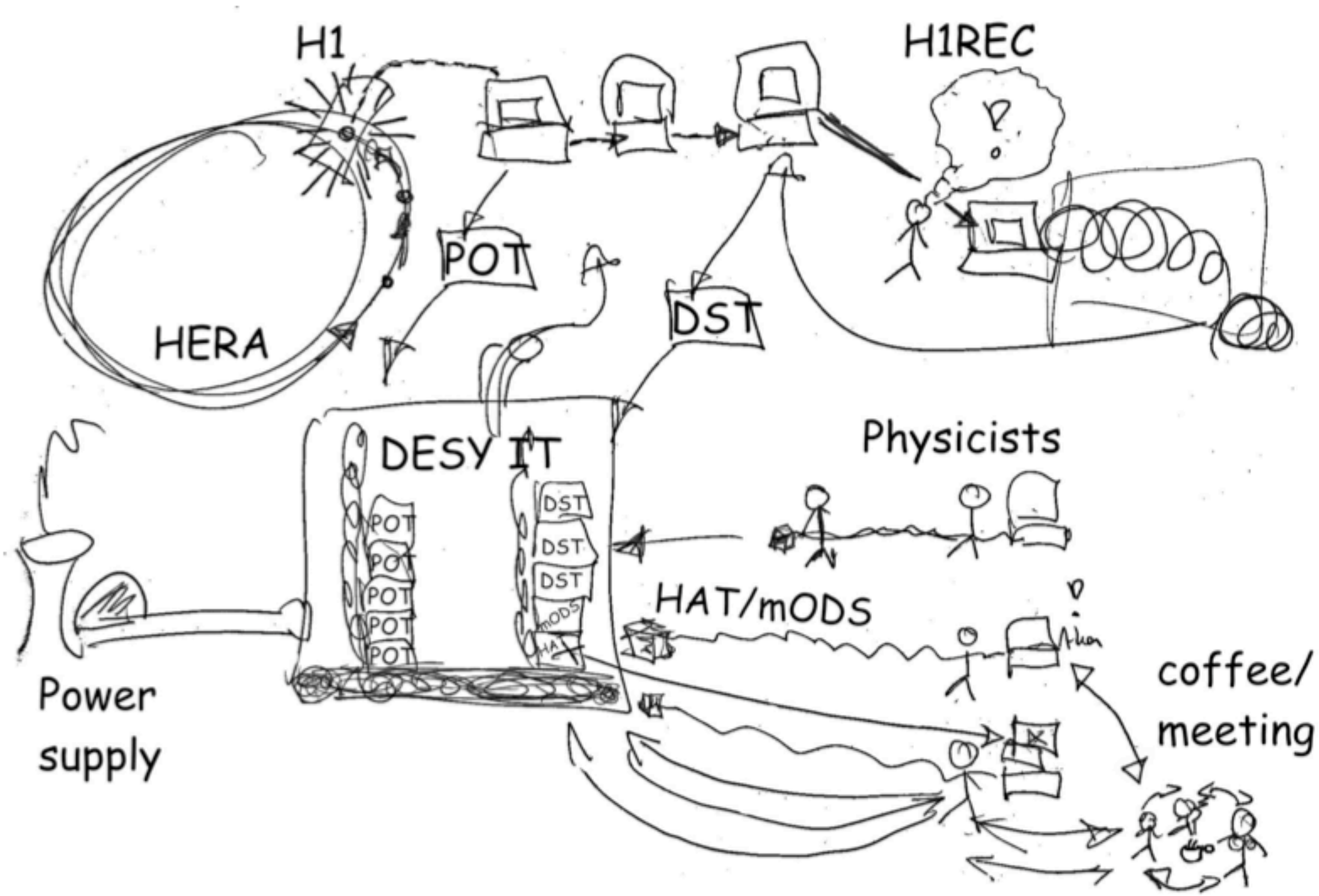




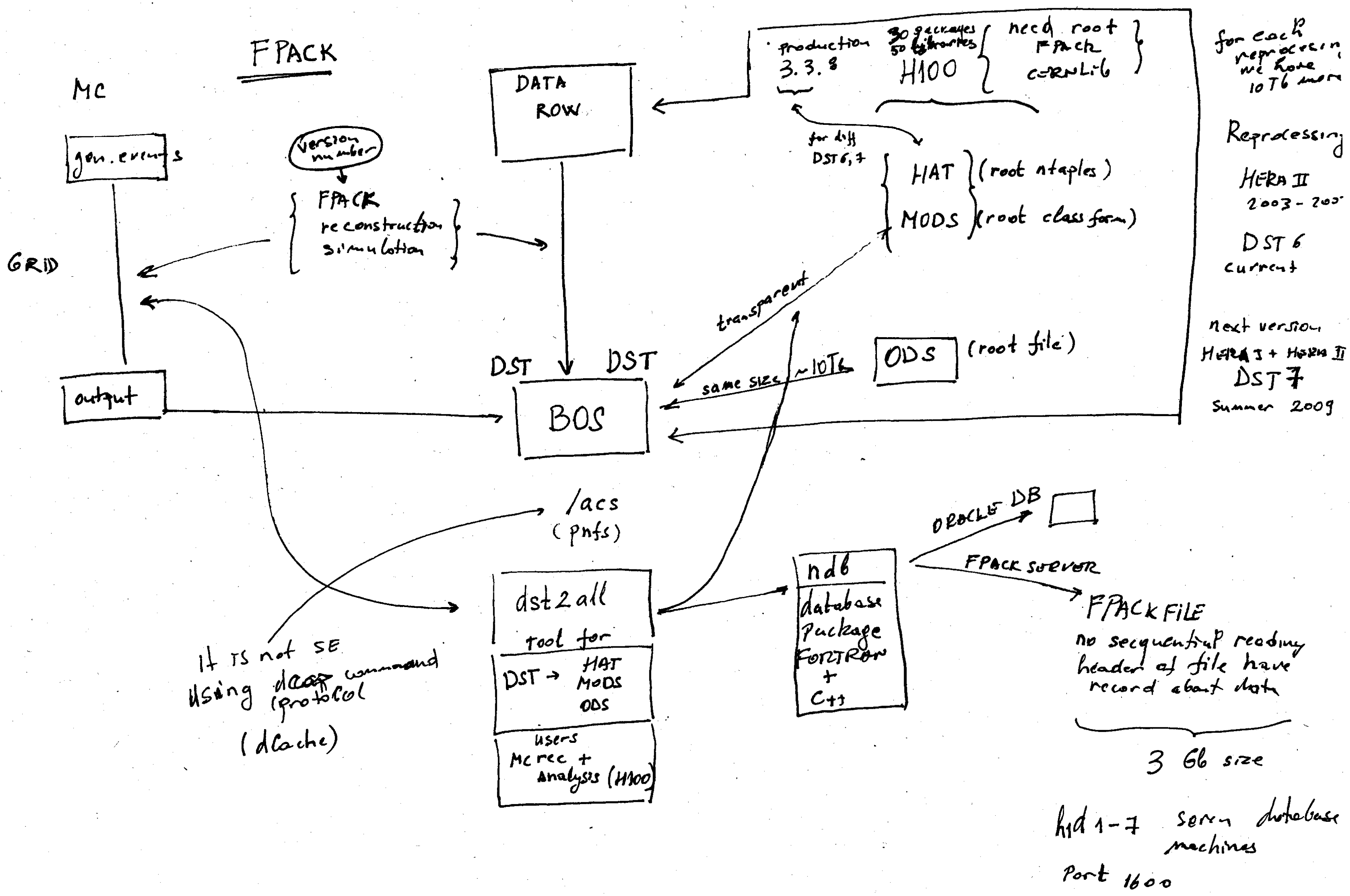


- The world's only electron-proton collider, collisions at H1 and ZEUS 1994-2007
- Precise picture of the proton, crucial measurements for hadron colliders

Workflow of a high energy physics experiment (1)

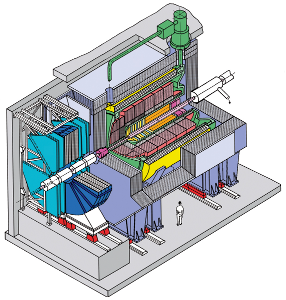


Workflow of a high energy physics experiment (2)



Workflow of a high energy physics experiment (3)

+

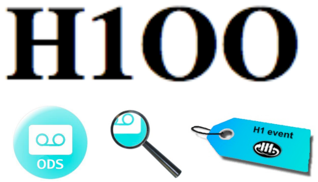


HERA delivered $e^{\pm}p$ collisions 1992-2007 and the H1 Collaboration collected 0.5 fb^{-1} of data, ~ 10^9 events

119.977679	129.534731	124.739135	176.316414
130.46875	135.839924	130.84732	168.289658
135.895502	149.510531	140.795689	120.686833
134.127052	140.495868	132.823819	206.138393
129.851598	137.880438	124.888856	189.675642
123.797241	131.84633	126.146789	202.496855
118.435374	130.691651	112.877008	140.366234
112.401212	121.561443	114.237637	125.298579
112.388488	128.496503	113.302591	192.223669
129.011813	138.880759	128.517198	108.701884
127.077465	139.289941	129.528986	127.406576
124.9785	135.363241	127.454638	129.669126
124.294035	133.242253	124.704841	244.567067
125.663717	135.159011	125.476984	169.271991
123.704853	127.612613	124.25382	170.401964
118.926697	122.818967	115.379664	134.970308
116.588208	121.798711	116.018173	323.148148
119.458869	124.788744	119.103839	204.736734
120.081967	124.847434	120.425321	289.50681
123.462329	127.367029	123.298233	287.632974
124.442179	128.115374	125.592252	362.764329
125.490169	128.448761	124.411031	382.978361
124.446597	128.898705	126.602473	358.369956

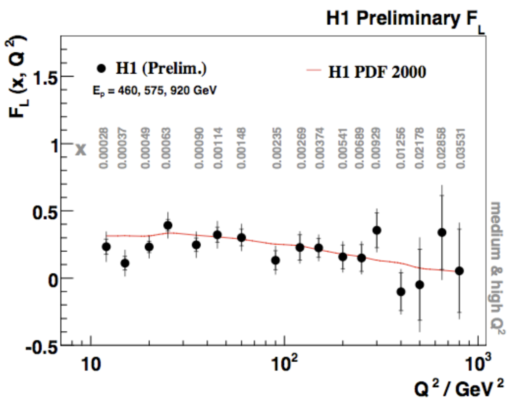
The raw data output from the detector is written to tape

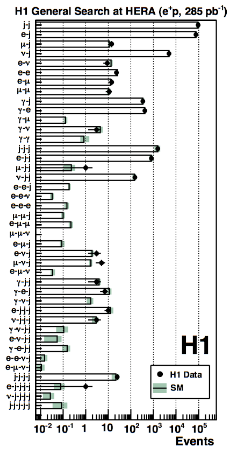
Raw data transformed into DST format using Fortran based software, regular re-processing



Analysis level data format and software written in C++ and based on ROOT

Regular common data and MC production, calibrations and analysis performed using central computing resources

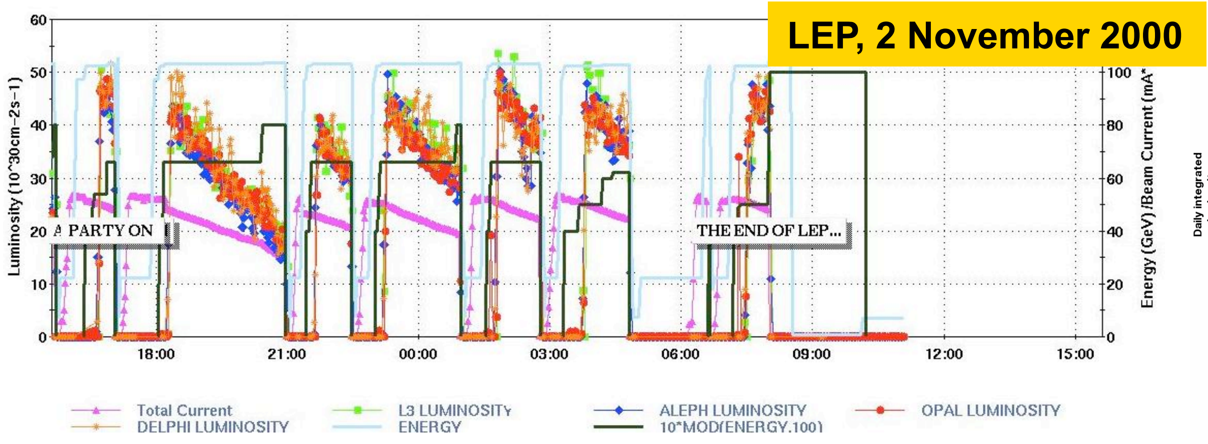


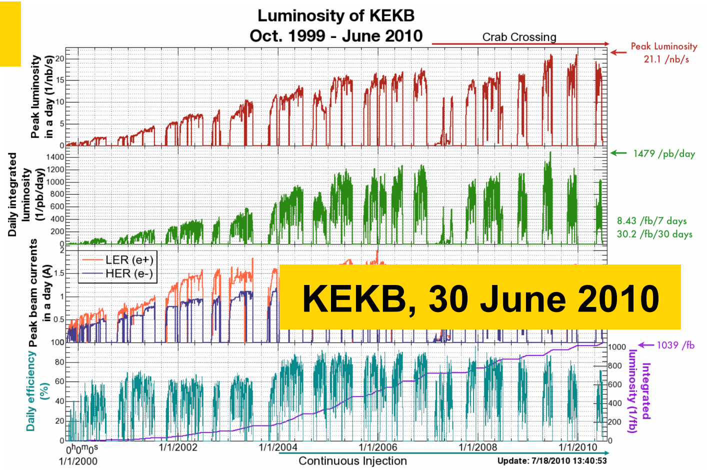


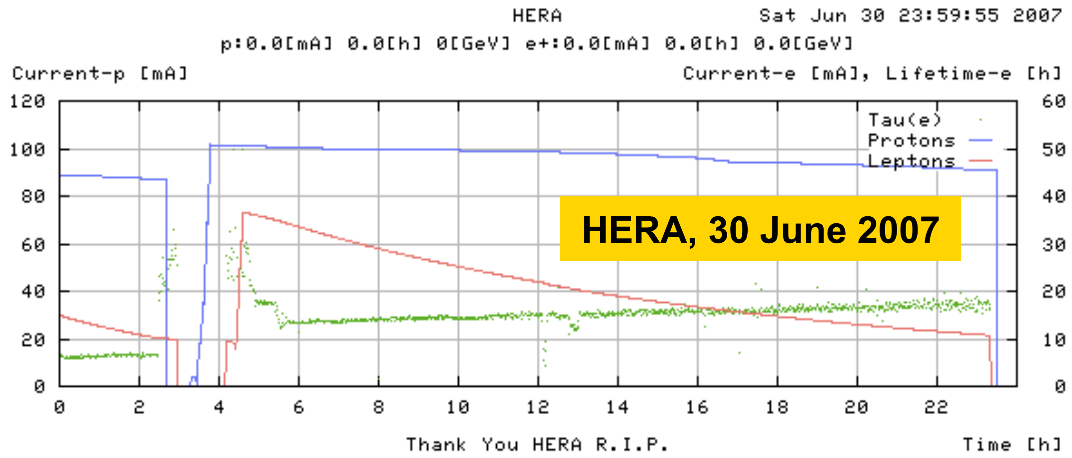
H1 publishes physics results

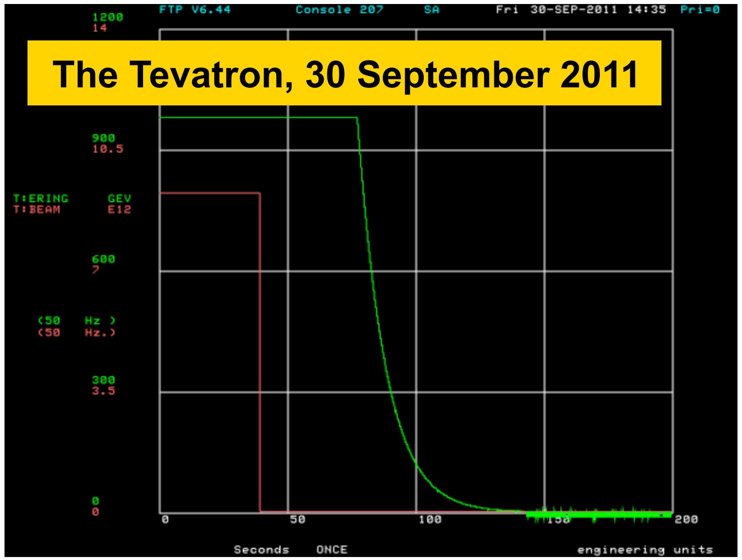
DESY

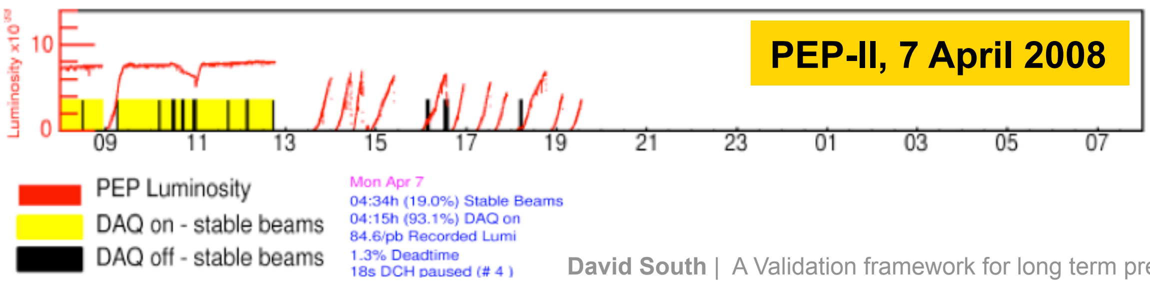
The last years have seen the end several older experiments





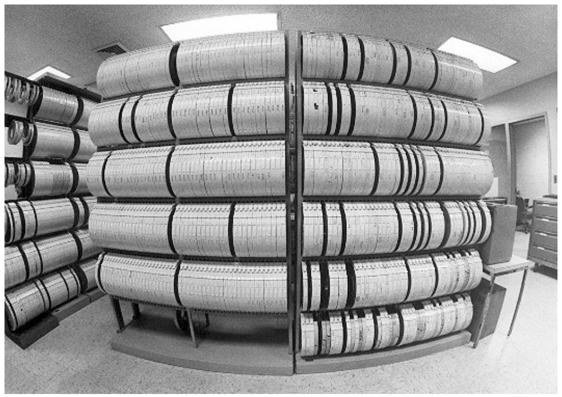






What do you do when the collisions have stopped ?

- Finish the analyses! But then what do you do with the data?
  - Until recently there was no clear policy on this in the HEP community
  - In the main, older HEP experiments have simply lost the data
- Data preservation, including long term access, is generally not part of the planning, software design or budget of an experiment
  - So far, HEP data preservation initiatives have been in the main not planned by the original collaborations, but rather the effort a few knowledgeable people
- The conservation of tapes is not data preservation!
  - *"We cannot ensure data is stored in file formats appropriate for long term preservation"*
  - *"The software for exploiting the data is under the control of the experiments"*
  - *"We are sure most of the data are not easily accessible!"*

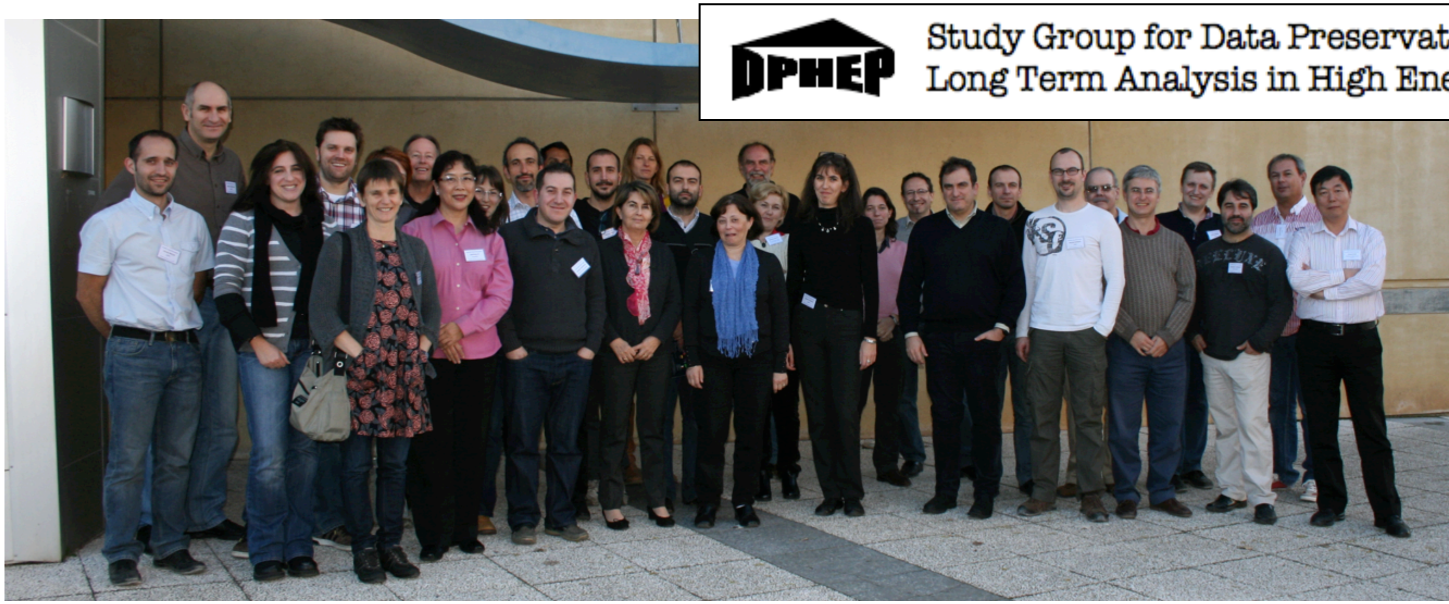


The difficulties of data preservation in HEP

- Experiments are generally interested in the here and now
  - Until recently the issue of data preservation was not really considered at the LHC
- Handling HEP data involves large scale traffic, storage and migration
  - The increasing scale of the distribution of HEP data and evolving access methods may complicate the task
- Who is responsible? The experiments? The computing centres?
  - Problem of older, unreliable hardware: unreadable tapes after 2-3 years
  - The software for accessing the data is usually under the control of the experiments
- Key resources, funding and expertise, decrease after data taking stops
- And importantly: *Who says we want to do all this anyway ?*
  - Is the potential benefit really worth the cost and effort? And how much does it cost?
  - **Can the relevant physics cases be made?**

DPHEP: An international Study Group on data preservation

DPHEP Study Group for Data Preservation and Long Term Analysis in High Energy Physics



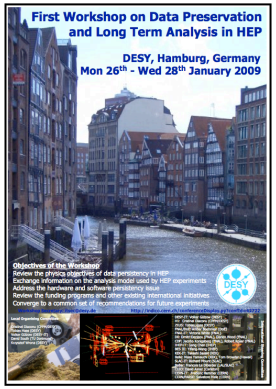
- First contacts established 2008, endorsed as an ICFA panel 2009
  - Group since grown to over 100 contact persons
  - Initial make up of the group was driven by the coincidence of the end of data taking at several large colliders – SLAC, HERA, Tevatron
  - Has grown to include many others including the LHC experiments
- Steering Committee with representatives from all members
- International Advisory Committee

DESY

DPHEP: All major High Energy Physics players represented

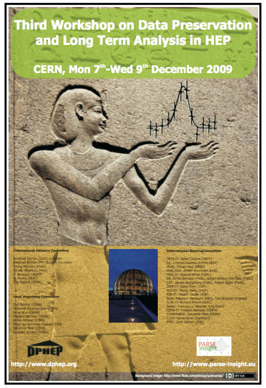
DPHEP: Series of workshops held since 2009

> Series of DPHEP workshops held since 2009



Jan 2009: DESY

May 2009: SLAC



Dec 2009: CERN

Jul 2010: KEK

May 2011: Fermilab

Nov 2012: CPPM

> Dedicated sessions at Computing in High Energy Physics conferences

> More recently: meetings focussing on dedicated topics such as “Costs of Curation” and “DPHEP Portal” (planned for later this year)

> 2009: Short publication, four key areas of the study group: Physics Case for Data Preservation, Preservation Models, Technologies, Governance

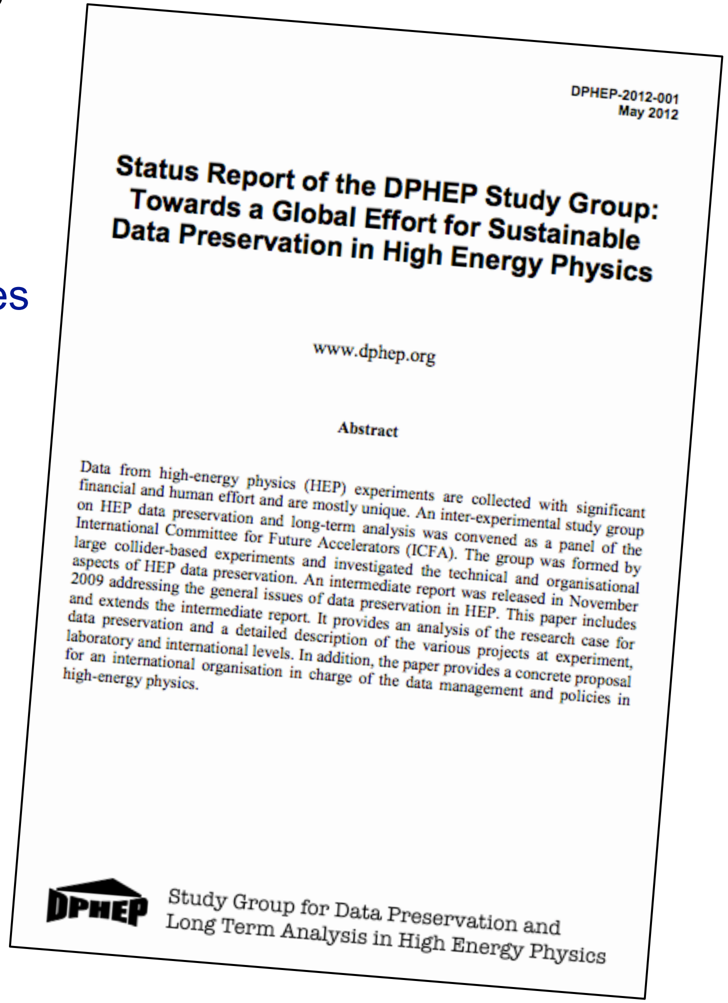
arXiv:0912.0255

DESY

DPHEP: Publication of the key findings

> 2012: Full status report of the activities of the DPHEP Study Group, including:

- Tour of data preservation activities in other fields
- An expanded description of the physics case
- Defining and establishing data preservation principles
- Updates from the experiments and joint projects
- FTE estimates for these and future projects
- Next steps to establish fully DPHEP in the field



DPHEP-2012-001
May 2012

Status Report of the DPHEP Study Group: Towards a Global Effort for Sustainable Data Preservation in High Energy Physics

www.dphep.org

Abstract

Data from high-energy physics (HEP) experiments are collected with significant financial and human effort and are mostly unique. An inter-experimental study group on HEP data preservation and long-term analysis was convened as a panel of the International Committee for Future Accelerators (ICFA). The group was formed by large collider-based experiments and investigated the technical and organisational aspects of HEP data preservation. An intermediate report was released in November 2009 addressing the general issues of data preservation in HEP. This paper includes and extends the intermediate report. It provides an analysis of the research case for data preservation and a detailed description of the various projects at experiment, laboratory and international levels. In addition, the paper provides a concrete proposal for an international organisation in charge of the data management and policies in high-energy physics.

DPHEP Study Group for Data Preservation and Long Term Analysis in High Energy Physics

arXiv:1205.4667

Building the physics case: Reasons to preserve HEP data

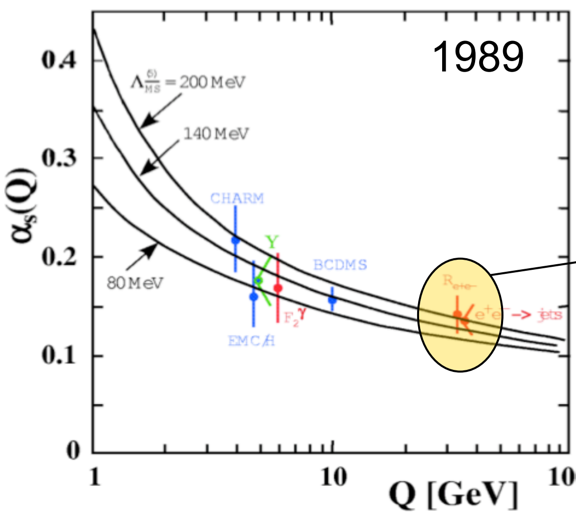
> Long term completion and extension of an existing physics program
  - Up to 25% of papers are finalised in the "archival mode". Publications still coming!
  - Gain in scientific output of the experiments

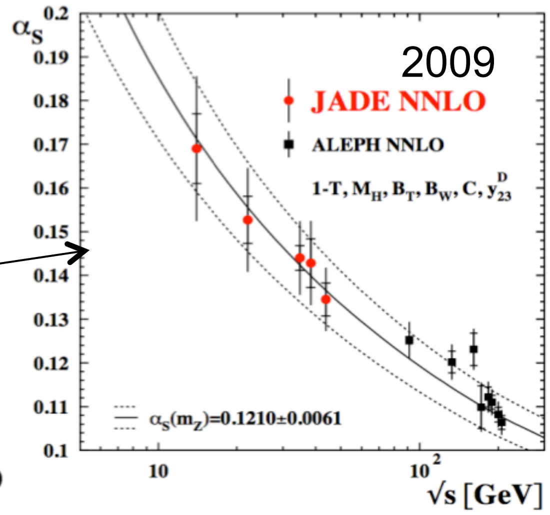
> Cross-collaboration and combinations of physics results
  - During the active lifetime of similar experiments at one facility: LEP, HERA, TeVatron
  - And later across larger boundaries: Belle/BaBar, TeVatron/LHC

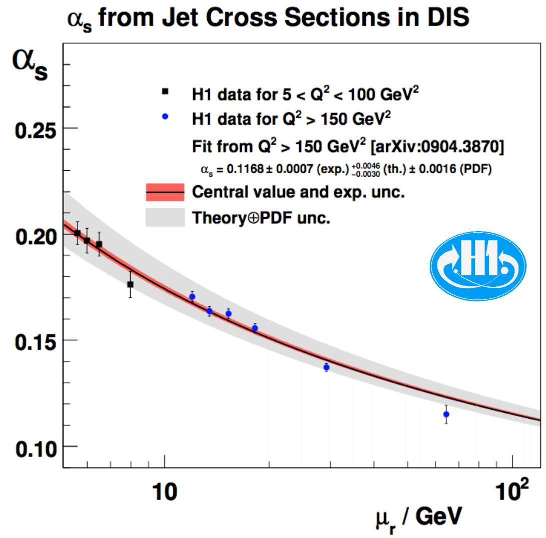
> Revisit old measurements or perform new ones
  - Access to newly developed techniques, comparisons to new theoretical models
  - Unique data sets available in terms of energy, initial states

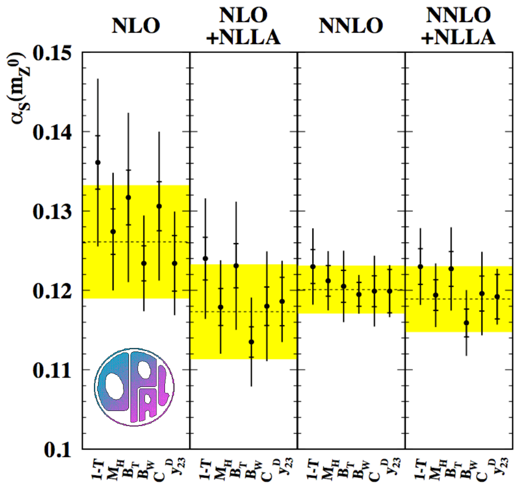
> Use in scientific training, education, outreach
  - Simplified formats: associated exercises to perform e.g. composite-particle reconstruction, finding signals in the background, ...

Example: Revisit old measurements or perform new ones









- Access to newly developed techniques, comparisons to new theoretical models
  - History may be repeated with the HERA α_s measurements
- Unique data sets are available in terms of initial state particles and energy
  - If no LHeC or alternative, all he have are the HERA $e^{\pm}p$ data
  - Tevatron $p\bar{p}$ are also unique: A_{FB}, high-x jets, ...
  - Fixed target experiments, ... others, ...

Ok, so it's worth doing. Define some preservation models

> First you need to ask the question: What is HEP data?

- Digital information, software, meta-data, documentation, publications, expertise, ...
- Define preservation levels according to use case and expectation

Increasing cost, complexity and benefits ↓

	Preservation Model	Use Case	
1	Provide additional documentation	Publication related info search	**Documentation**
2	Preserve the data in a simplified format	Outreach, simple training analyses	**Outreach**
3	Preserve the analysis level software and data format	Full scientific analysis, based on the existing reconstruction	**Technical Preservation Projects**
4	Preserve the reconstruction and simulation software as well as the basic level data	Retain the full potential of the experimental data	

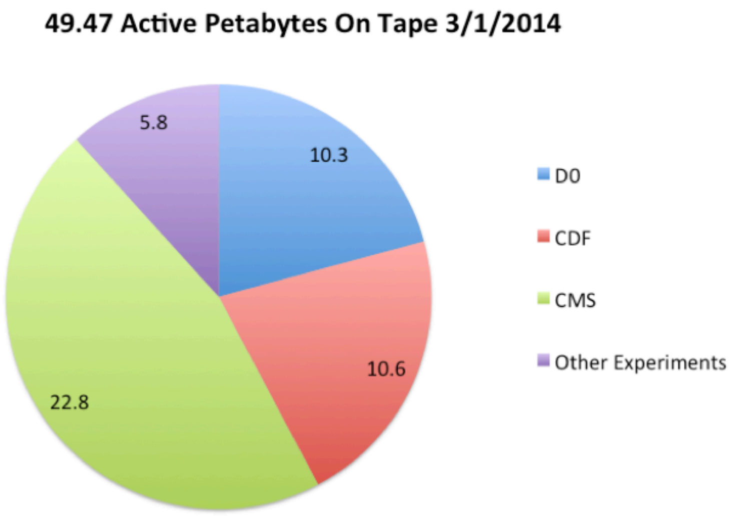
> Originally idea was more of a progression, almost like an inclusive level structure, but now seen as complementary initiatives

> Three levels representing three areas:

- Documentation, Outreach and Technical Preservation Projects

The HEP data themselves are not the problem

- What is clear from estimates of data size, is not prohibitive: numbers between 0.5 to a 10 PB
- Computing centres are, at least by volume arguments, able to store the data
  - Data preservation in HEP is not about the data!



- Regular migration of the data to latest technologies should be considered and carefully planned
- Currently employed storage systems may not be suited for archival storage: Requires regular integrity checks, rather than occasional use
  - Any archival system should be able to absorb future technological evolutions
- What is much more crucial is the preserving the software and environment, which allows analysis to continue into the future
  - ***This is the challenge facing HEP experiments***

Isn’t it obvious, virtualisation will solve everything?

My first and very naïve ansatz

- OK, why don’t we just put everything an a virtual machine?
  - Data archival is done elsewhere, just need “to plug that into the VM”
  - Your VM contains everything you need to develop and run code and analysis
- The problem would then be reduced to maintain virtual images, and maintain their ability to run. In the Cloud era, seems like a trivial task
- Problems: Everything in IT is a moving target:
  - Will your network always be the same?
  - Will your access protocol always be the same?
  - Are you sure you do not need new software (e.g. MC generators) that require a new OS?
  - Are you sure your i386/SL4 VM will produce the same results when emulated on a quantum computer in NN years?
  - What about service you need, like CondDB,…
- Naïve virtualization will not work… but still, virtualization can help

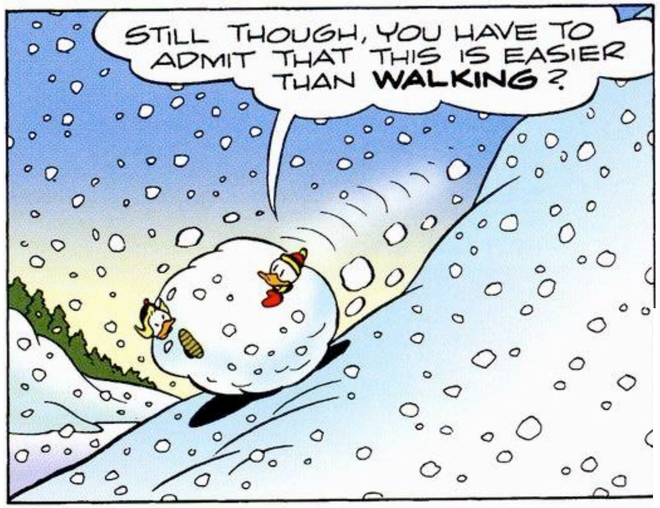
Yves Kemp | Long Term Data Preservation and Virtualization | 22.6.2010 | **Page 5**

Yves Kemp (DESY-IT)

Freezing vs rolling

> **Pro Freezing**
- One-time effort, very small maintenance outside of analysis phase
- Also allows software w/o code (but might fail with DRM / licensing issues)

> **Cons Freezing**
- Rely on certain standards and protocols that may evolve
- Potential performance problems



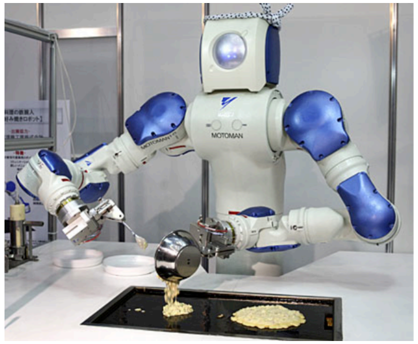
> **Pro Test-driven migration**
- Usability and correctness of code is guaranteed at every moment
- Data accessibility and integrity can be checked as well
- Fast reaction to standard/protocol changes
- General code quality can improve, as designed for portability and migration

> **Cons Test-driven migration**
- Needs long-time intervention, more man-power and resources needed
- Some knowledge of the frameworks must be passed to maintainers

Yves Kemp (DESY-IT)

Pizza Preservation

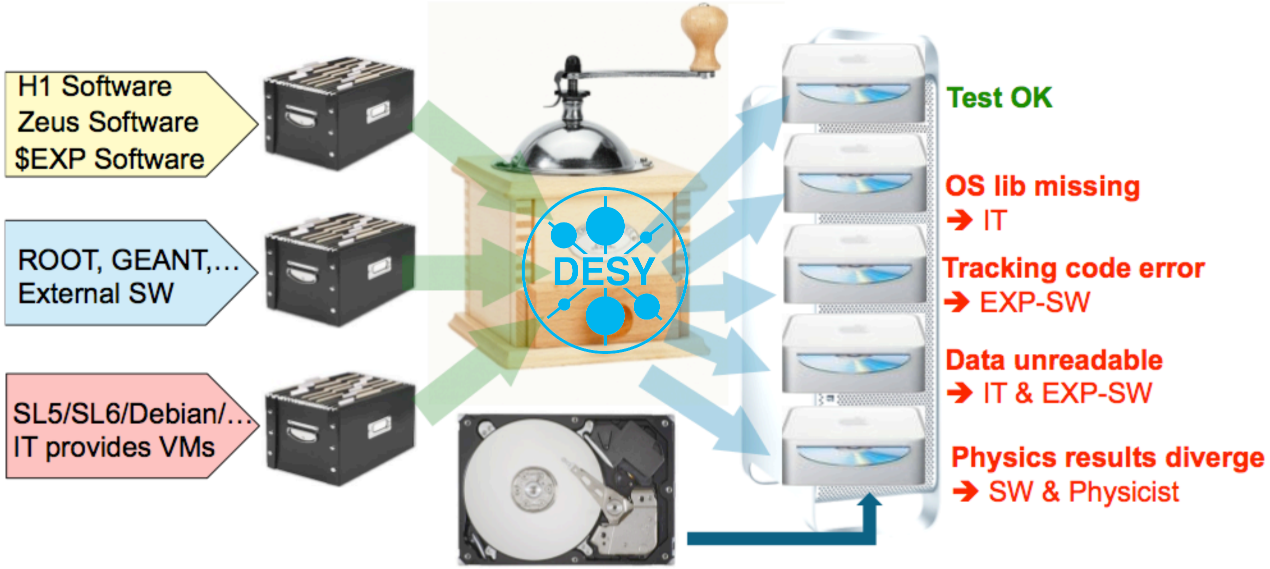
- Couple of days
  - Fridge
- Couple of month
  - Deep freezer
- Couple of years???
  - Preserve the recipe
  - Practice it often: You will not forget the recipe and you can detect variations in external dependencies



- Whilst freezing the software and environment is easy to do, long term use and correctness of the results not guaranteed
  - Naïve assumption virtualisation solves everything may break down at the first security issue
- Freezing software is *OK* if the timeline and scope are reduced, but if changes are needed this is more difficult the longer software is frozen
- Better to cook the same recipe again and again (and maybe even allow it to be improved), validating the output *automatically*
  - Virtualisation can help!

Yves Kemp (DESY-IT)

The Software Preservation System @ DESY



- Automated validation system to facilitate future software and OS transitions
  - Uses virtualisation techniques to repeatedly run well defined tests
  - Perform checks of different and evolving environments (OS, s/ware configuration)
  - Stand alone system: No hidden dependencies or /afs access etc: rigorous testing
  - Automatically check these results against predefined, default values
  - Notify when test results differ from these values
  - Separate responsibilities of IT and the experiments

The Software Preservation System @ DESY

- Automated validation system to facilitate future software and OS transitions
  - Uses virtualisation techniques to repeatedly run well defined tests
  - Perform checks of different and evolving environments (OS, s/ware configuration)
  - Stand alone system: No hidden dependencies or /afs access etc: rigorous testing
  - Automatically check these results against predefined, default values
  - Notify when test results differ from these values
  - Separate responsibilities of IT and the experiments

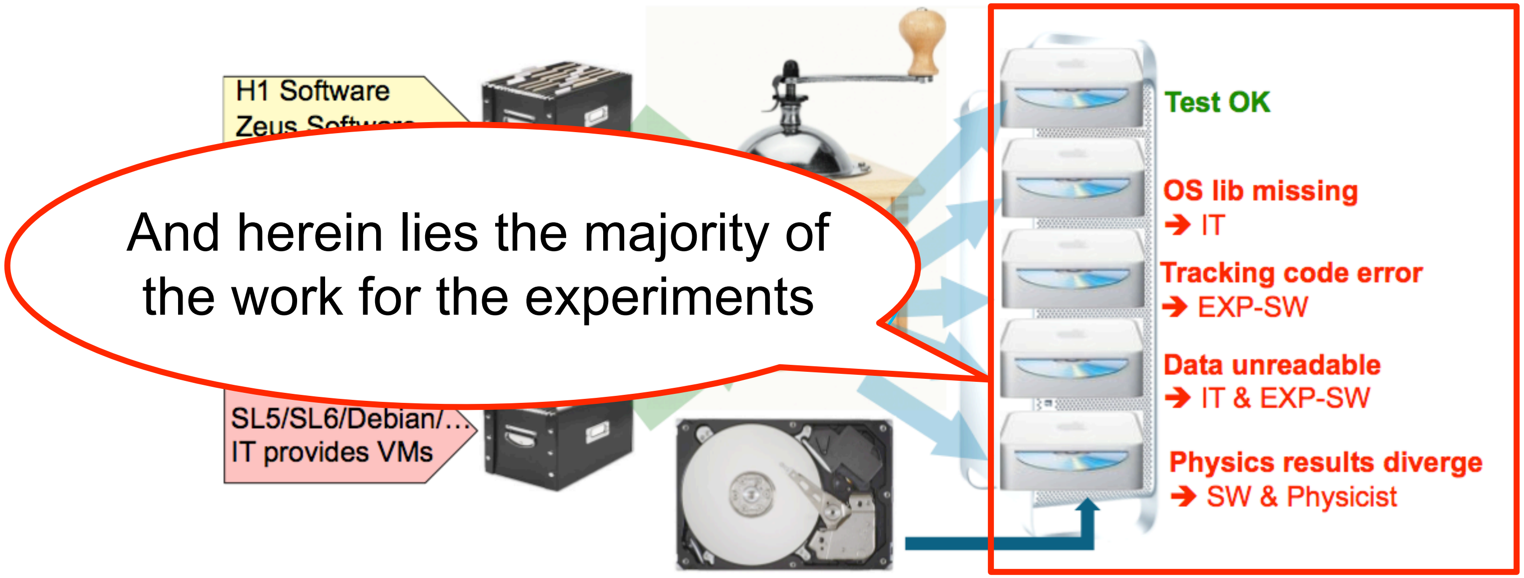
The Software Preservation System @ DESY

> Automated validation system to facilitate future software and OS transitions
- Uses virtualisation techniques to repeatedly run well defined tests
- Perform checks of different and evolving environments (OS, s/ware configuration)
- Stand alone system: No hidden dependencies or /afs access etc: rigorous testing
- Automatically check these results against predefined, default values
- Notify when test results differ from these values
- Separate responsibilities of IT and the experiments

The Software Preservation System @ DESY

- Automated validation system to facilitate future software and OS transitions
  - Uses virtualisation techniques to repeatedly run well defined tests
  - Perform checks of different and evolving environments (OS, s/ware configuration)
  - Stand alone system: No hidden dependencies or /afs access etc: rigorous testing
  - Automatically check these results against predefined, default values
  - Notify when test results differ from these values
  - Separate responsibilities of IT and the experiments

The Software Preservation System @ DESY

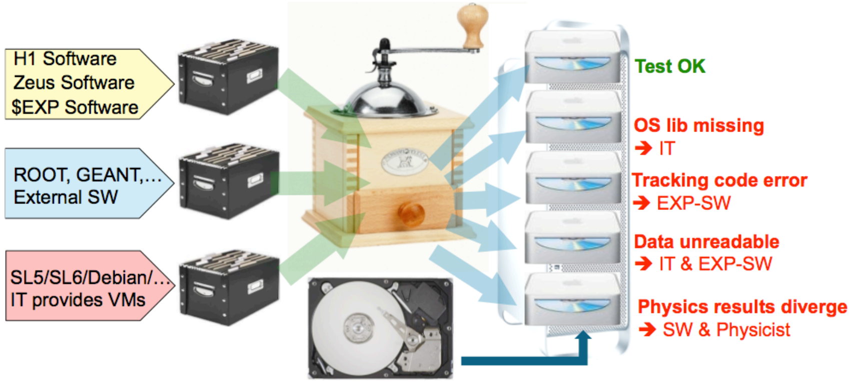


- Automated validation system to facilitate future software and OS transitions
  - Uses virtualisation techniques to repeatedly run well defined tests
  - Perform checks of different and evolving environments (OS, s/ware configuration)
  - Stand alone system: No hidden dependencies or /afs access etc: rigorous testing
  - Automatically check these results against predefined, default values
  - Notify when test results differ from these values
  - Separate responsibilities of IT and the experiments

Inputs to the preservation system and work flow

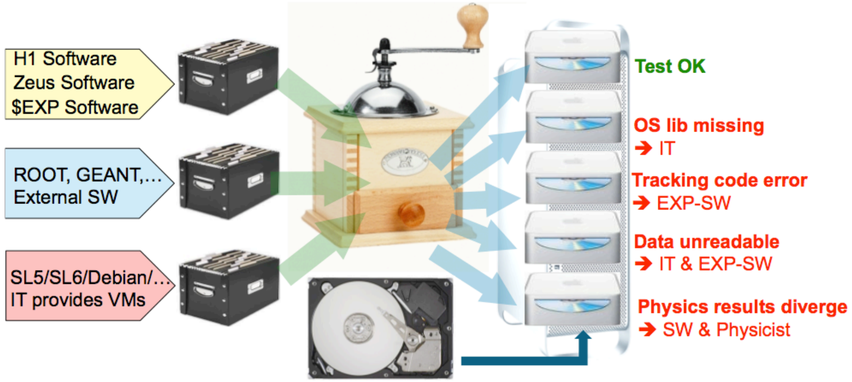
> 1) Experimental software is consolidated and migrated to the most recent OS

- Unnecessary dependencies are removed, necessary ones incorporated into the system
- Analysis and data validation tests defined and prepared, examining each part of the experimental software required



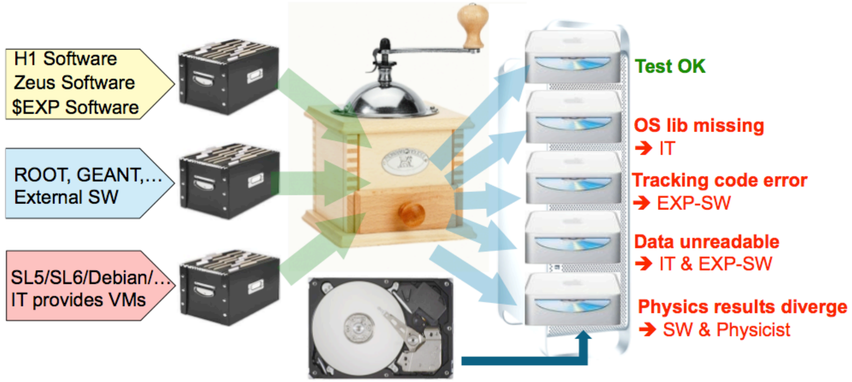
Inputs to the preservation system and work flow

- 1) Experimental software is consolidated and migrated to the most recent OS
  - Unnecessary dependencies are removed, necessary ones incorporated into the system
  - Analysis and data validation tests defined and prepared, examining each part of the experimental software required



- 2) Regular build of the experimental software is done automatically according to the current prescription of the working environment and the validation tests are performed
  - At regular intervals, new OS and software versions will then be integrated into the system, under the supervision of experts from the host IT department

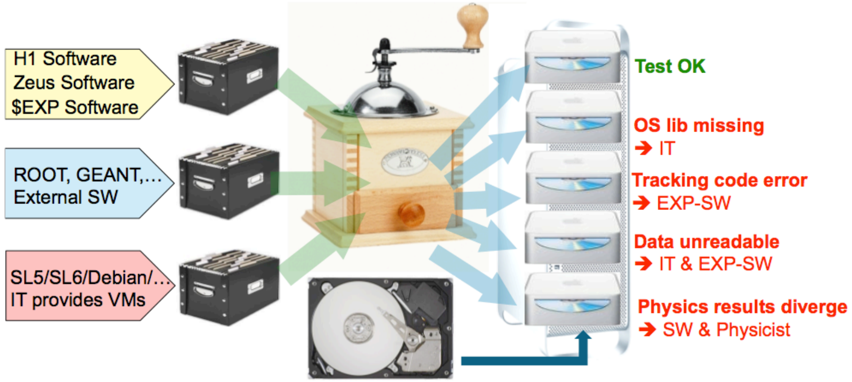
Inputs to the preservation system and work flow



- 1) Experimental software is consolidated and migrated to the most recent OS
  - Unnecessary dependencies are removed, necessary ones incorporated into the system
  - Analysis and data validation tests defined and prepared, examining each part of the experimental software required
- 2) Regular build of the experimental software is done automatically according to the current prescription of the working environment and the validation tests are performed
  - At regular intervals, new OS and software versions will then be integrated into the system, under the supervision of experts from the host IT department
- 3) Validation successful: no further action to be taken; test fails, differences compared to the last successful test are examined and problems identified
  - Interventions then required either by the host of the validation suite or the experiment themselves, depending on the nature of the reported problem

Inputs to the preservation system and work flow

- 1) Experimental software is consolidated and migrated to the most recent OS



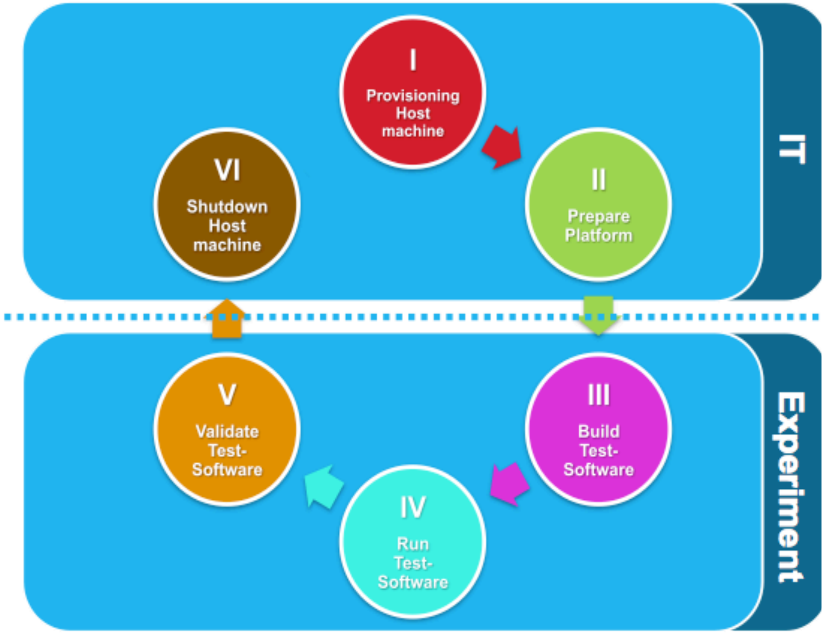
  - Unnecessary dependencies are removed, necessary ones incorporated into the system
  - Analysis and data validation tests defined and prepared, examining each part of the experimental software required

- 2) Regular build of the experimental software is done automatically according to the current prescription of the working environment and the validation tests are performed
  - At regular intervals, new OS and software versions will then be integrated into the system, under the supervision of experts from the host IT department

- 3) Validation successful: no further action to be taken; test fails, differences compared to the last successful test are examined and problems identified
  - Interventions then required either by the host of the validation suite or the experiment themselves, depending on the nature of the reported problem

- 4) Final phase: when person-power becomes an issue or an unsolvable problem occurs: preserve final image, finite lifetime

The `sp-system`: Towards the full implementation



- Pilot project in 2010
  - Single configuration, simple tests
- Full implementation now at DESY
- Common baseline of Scientific Linux 5 32-bit achieved in 2011 by all expts
  - Sound starting point for validation
- Multiple OS configurations available within `sp-system`:
  - **sl5.6/64**(gcc4.4), **sl5.7/32**(gcc4.4), **sl5.7/64**(gcc4.1), **sl5.7/32**(gcc4.1), **sl6.2/64**(gcc4.4)
- In addition, to multiple ROOT (analysis software) versions
  - **5.26**.00d, **5.28**.00c, **5.30**.05, **5.32**.00, **5.34**.01
- 64-bit systems a major step toward migrations to future OS and hardware
  - SL6 will only be supported in 64 bit variant at DESY
  - nfs4.1 technology, to be used in dCache, native only in SL6.2/64 or higher

First sketch of H1 tests

++++++++++++++
++++++++++++++
++h1 executables
++++++++++++++
++++++++++++++

antlr
batch_kinit
cartl
chk_tree
dlg
fpack
fplist
fpmerge
fpsubset
h1fttemu
h1geaonly
h1ieeefp.o / h1ieeefp.cpp
h1rec
h1sim
h1simcheck
h1simrec
hostn
l45his
l4his
l4m
l4s
look
ltab
ndbint
nqs2pbs
pbs_tclsh
pbs_wish
pbsdsh
pbsnodes
printjob
printtracking
qalter
qdel
qdisable
qenable
qhold
qmgr
qmove
qmsg
qorder
qrerun
qrls
qrun
qselect
qsig
qstart
qstat
qstop
qsub
qterm
refresh
refresh_init
tracejob
xpbs
xpbsmon

++++++++++++
++++++++++++
55 ++h1 libraries
++++++++++++
++++++++++++

#cernlib-gcc44
libLHAPDF.so
libariadne412.a
libbases.a
libbos.a
libcascade2.a
libdatman.a
libdiffvm.a
libfpack.a
libfpack.so
libgksdummy.a
libh1bstrec.a
libh1eclass.a
libh1fttemu.a
libh1geang.a
libh1geanh.a
libh1geant.a
libh1l4.a
libh1look.a
libh1mcutil.a
libh1ndb.a
libh1phan.a
libh1qt.a
libh1rec.a
libh1sim.a
libh1trig.a
libh1util.a
libheracles*.a
libheracles*.so
libhztool.a
libjetset74.a
liblook.a
libpythia62.a
libpythia64.a
librapgap31.a
libshift.a

+++++++++++++++++
+++++++++++++++++
36 ++h1oo packages 51
+++++++++++++++++
+++++++++++++++++

H1Analysis
H1AnalysisExample
H1Arrays
H1Banks
H1Benchmarks
H1Binning
H1Bos2oop
H1CalcPointers
H1CalcWeights
H1Calculator
H1CalibTrigger
H1CaloTrigger
H1Clusters
H1Cuts
H1ElecCalibration
H1Examples
H1Filler
H1Finder
H1Geom
H1HadronicCalibration
H1Hat
H1HatFiller
H1HfsFinder
H1JetFinder
H1Macros
H1Mods
H1MuonFinder
H1NonepBgFinder
H1OOBanks
H1Ods
H1PartEmFinder
H1PhysUtils
H1Pointers
H1QCDFunc
H1Red
H1SVFit
H1Selection
H1Skeleton
H1SoftLeptonId
H1Steering
H1SubDetInfo
H1Tools
H1Tracks
H1TrkFinder
H1UserCim
H1UserDstar
H1UserFtt
H1UserLifetime
H1Wrappers
oo_tools
#share

+++++++++++++++++
+++++++++++++++++
++h1oo binaries
+++++++++++++++++
+++++++++++++++++

AnalysisExample
AnalysisExampleExtraction
AnalysisExamplePlots
H1Bos2oop
H1Makeptr
Lt2Root
MakeInputTable
TestQCDFunc
batchAnalysis
boosted_jets
checkcim
cintsteering
clusters_ods
copyMyEvents
create_eventlist
dbaccess
deleteJobs
dst2all
dst2ods
dstar_mods
empz_hat
h1red
h1root
jpsi_mods
kaonfind_ods
l1te_hat
lumicalc
mergeAnalysis
mymkcim
ods2modshat
oolist
oolumi
oomclumi
oomove
oosubset
read_dstartree
read_eventlist
read_ods
read_usertree
rerun_finder
rerun_rec
resubChains
snapshot
steermanager
test_binning
write_eventlist

+++++++++++++++++
+++++++++++++++++
46 ++h1oo libraries
+++++++++++++++++
+++++++++++++++++

libH1Analysis.so
libH1AnalysisExample.so
libH1Arrays.so
libH1Benchmarks.so
libH1Binning.so
libH1CalcPointers.so
libH1CalcWeights.so
libH1Calculator.so
libH1CaloTrigger.so
libH1Clusters.so
libH1Cuts.so
libH1ElecCalibration.so
libH1Filler.so
libH1Filler_odsonly.so
libH1Finder.so
libH1Geom.so
libH1HadronicCalibration.so
libH1Hat.so
libH1HatFiller.so
libH1HfsFinder.so
libH1JetFinder.so
libH1MagfieldOO.so
libH1Mods.so
libH1MuonFinder.so
libH1NonepBgFinder.so
libH1OOBanks.so
libH1Ods.so
libH1PartEmFinder.so
libH1PhysUtils.so
libH1Pointers.so
libH1QCDFunc.so
libH1Red.so
libH1RedLook.so
libH1Red_bos.so
libH1SVFit.so
libH1Selection.so
libH1Skeleton.so
libH1SoftLeptonId.so
libH1SoftLeptonId_impl...so
libH1Steering.so
libH1SubDetInfo.so
libH1Tools.so
libH1Tracks.so
libH1TrkFinder.so
libH1UserCim.so
libH1UserDstar.so
libH1UserDstar_fill.so
libH1UserFtt.so
libH1UserFtt_Filler.so
libH1UserLifetime.so
libH1UserTiming.so
libH1UserTiming_fill.so
libH1Wrappers_bos.so
libH1Wrappers_fastjet.so
libH1Wrappers_geom.so

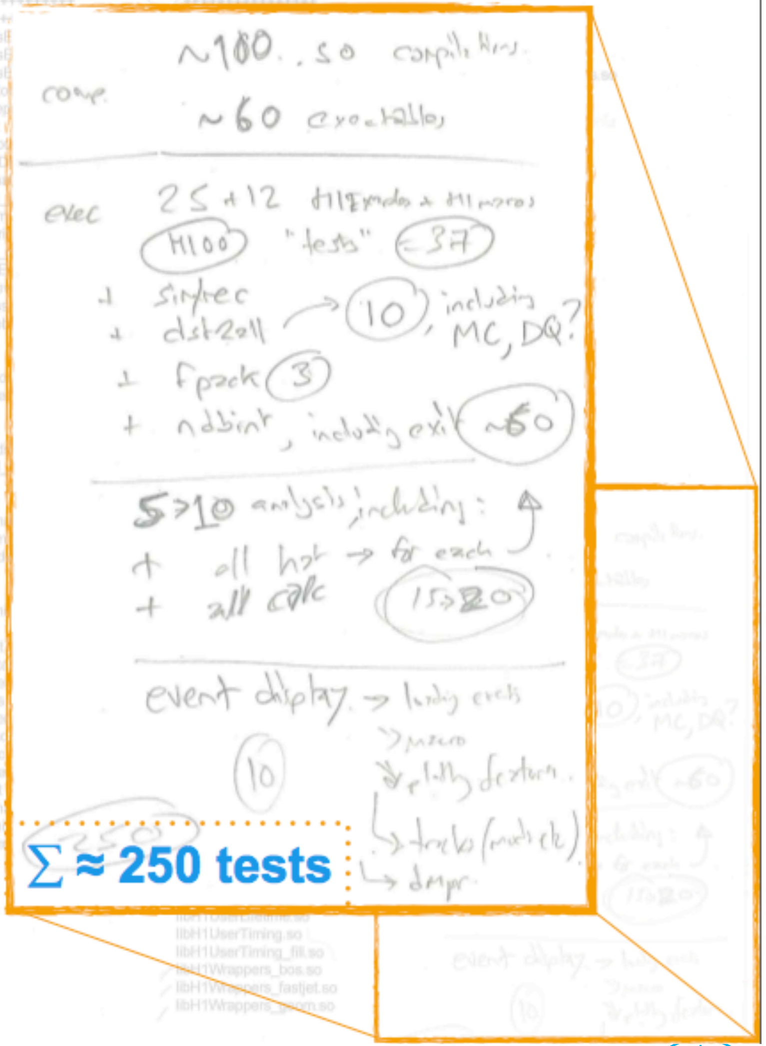
74

libH1Wrappers_lumi.so
libH1Wrappers_ndb.so
libH1Wrappers_neurobayes.so
libSISConePluginOO.so
libUser.so
libbosutil.so
libcernlibOO.so
libfastjetOO.so
libfortran.so
libfortranpatchOO.so
libfortranshared.so
libfortranstat.a
libfpackOO.so
libh1ndbOO.so
libh1recOO.so
libmdbdummy.so
libneurobayesOO.so
libsisconeOO.so
libutildummy.so

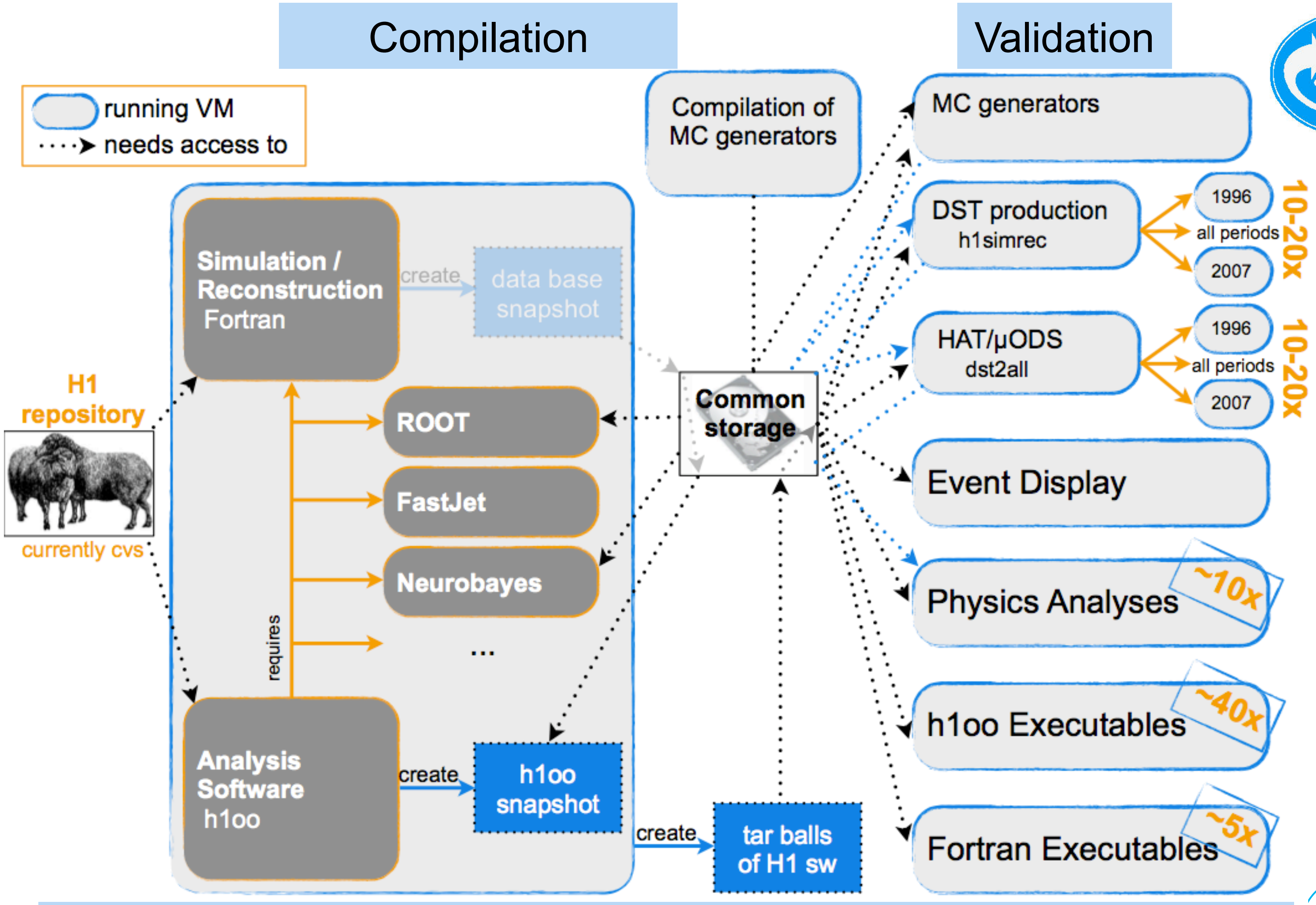
DESY

First sketch of H1 tests

- Validate compilation of
  - ~100 (shared) library objects
  - ~60 executables
    - MC generators not yet included
    - most important:
      - simrec - reconstruction / dst production
      - dst2all - h1oo file production
- Validate correct running of
  - 37 x h1oo
  - 4 x fpack, ndb
  - ≥10 x h1simrec → dst2all → analysis
    - One test for every run period + MC / (DQHat)
  - … Let's say about 60 executables
- Run and validate physics analyses
  - (At least) one test for every run period
    - Inclusive & all HAT/H1Calculator variables
  - 5-10 'real' physics analyses
- Check event display



Example structure of experimental tests: H1 (Level 4)



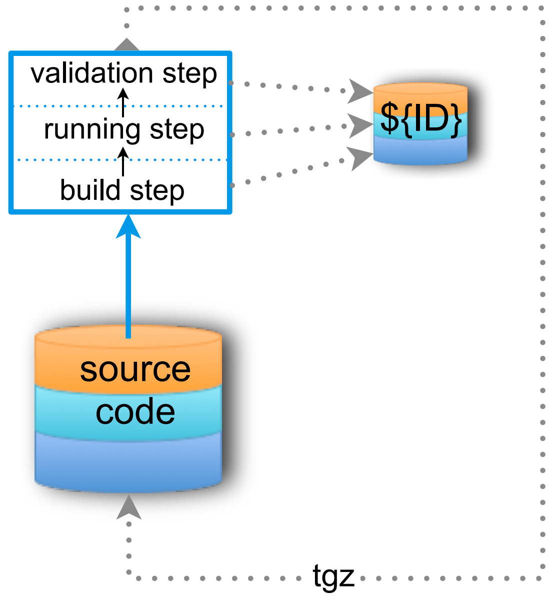
Including compilation of individual packages: about 250 tests planned by H1

Running jobs in the `sp-system`

> Initial step

- Compilation of analysis (level 3) and sim/rec (level 4) software
- ***Or:*** use tar-balls with pre-compiled software
- Provide access to software

  Copy tar-balls to persistent storage
- All output kept in directory with unique name



Running jobs in the `sp-system`

> Initial step

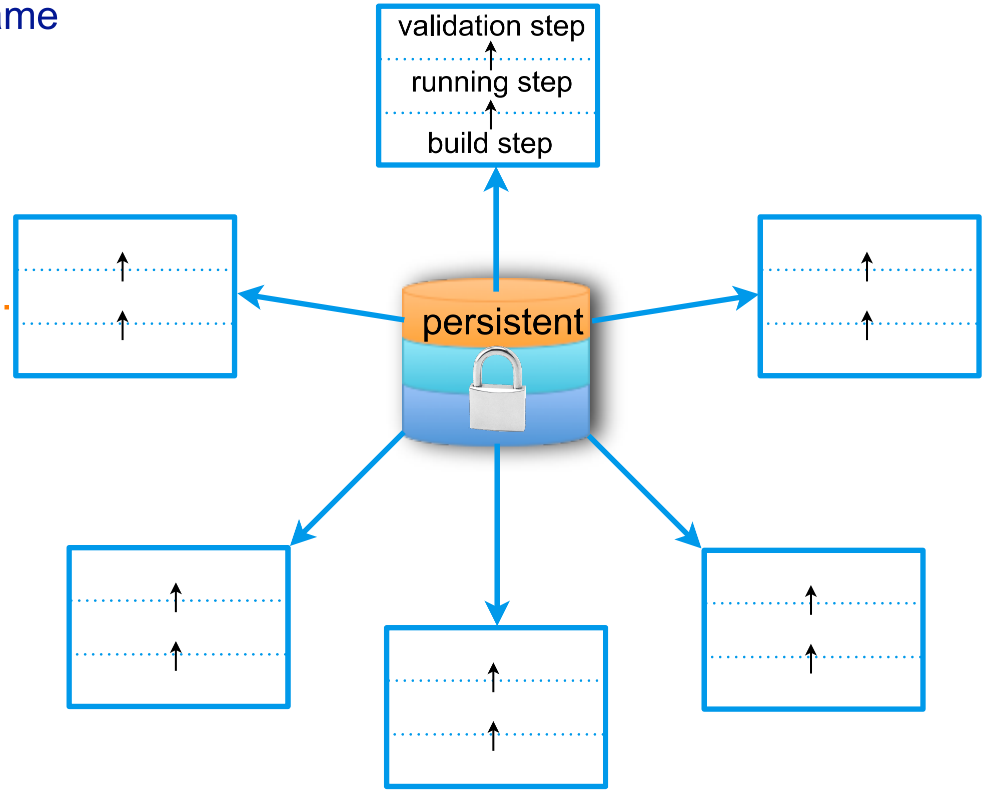
- Compilation of analysis (level 3) and sim/rec (level 4) software
- ***Or:*** use tar-balls with pre-compiled software
- Provide access to software

  Copy tar-balls to persistent storage
- All output kept in directory with unique name

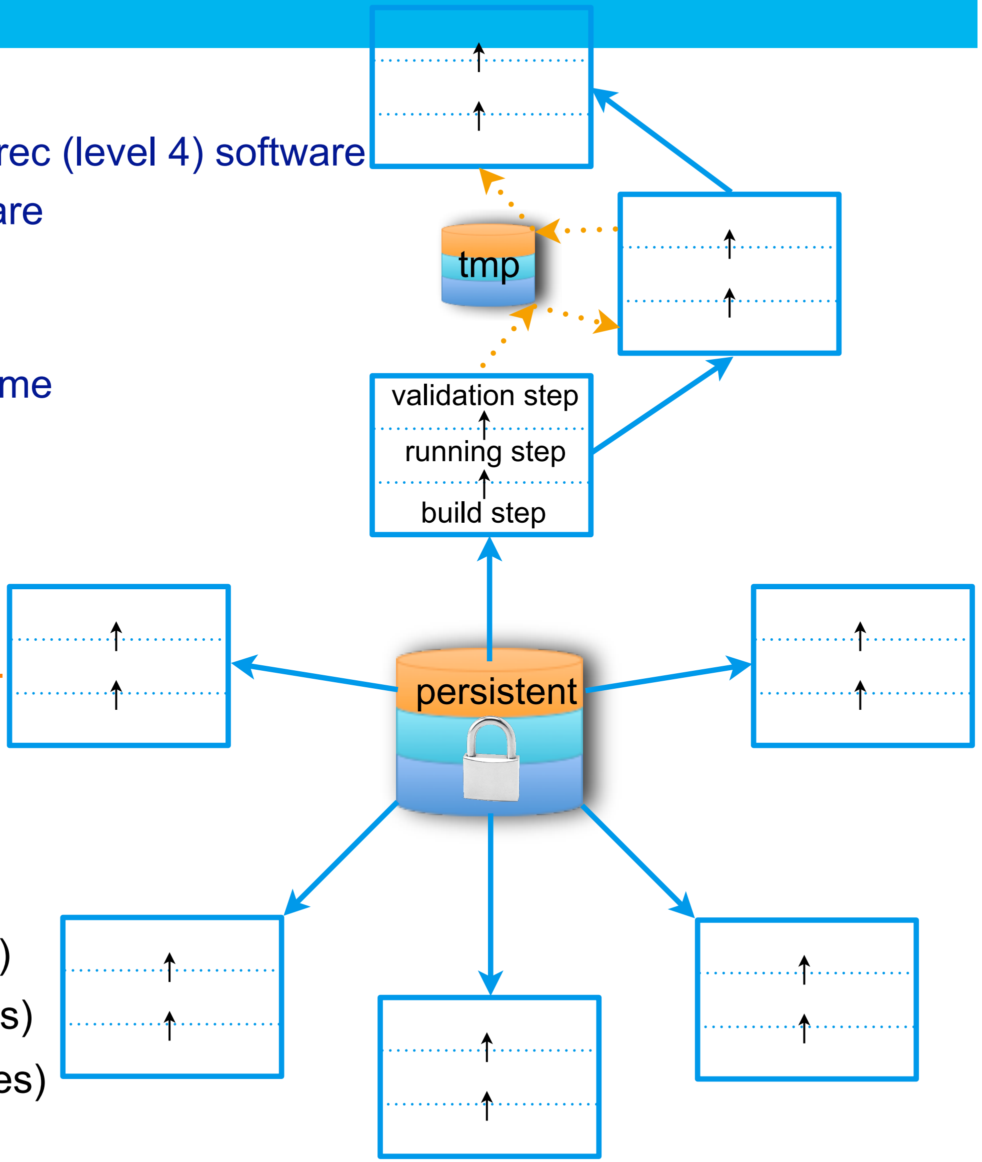
> Run parallel tests

- Set up software environment
- Validate binaries with persistent input

  e.g. event display, database access, ...



Running jobs in the `sp-system`



- **Initial step**
  - Compilation of analysis (level 3) and sim/rec (level 4) software
  - ***Or:*** use tar-balls with pre-compiled software
  - Provide access to software
    - Copy tar-balls to persistent storage
  - All output kept in directory with unique name
- **Run parallel tests**
  - Set up software environment
  - Validate binaries with persistent input
    - e.g. event display, database access, ..
- **Run sequential tests**
  - Set up software environment
  - Validate file production
    1. MC generation (produce gen files)
    2. Reconstruction (gen. files → DSTs)
    3. Analysis level (DSTs → ROOT files)
  - Tests use output of previous test as input
- Results remain accessible or can be reproduced with identical results

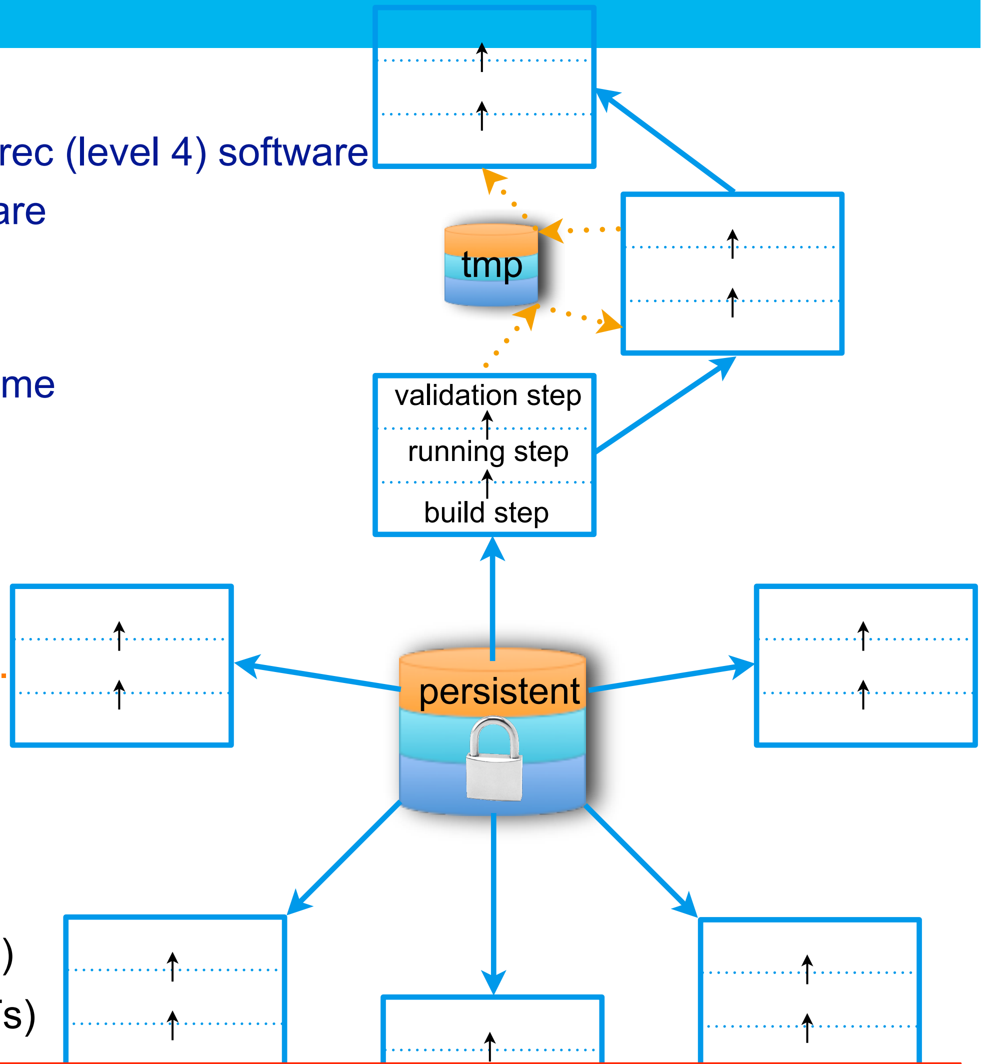
Running jobs in the `sp-system`

- Initial step
  - Compilation of analysis (level 3) and sim/rec (level 4) software
  - ***Or:*** use tar-balls with pre-compiled software
  - Provide access to software
    - Copy tar-balls to persistent storage
  - All output kept in directory with unique name

- Run parallel tests
  - Set up software environment
  - Validate binaries with persistent input
    - e.g. event display, database access, ..

- Run sequential tests
  - Set up software environment
  - Validate file production
    1. MC generation (produce gen files)
    2. Reconstruction (gen. files → DSTs)
    3. Analysis level (DSTs → ROC
  - Tests use output of previous test as

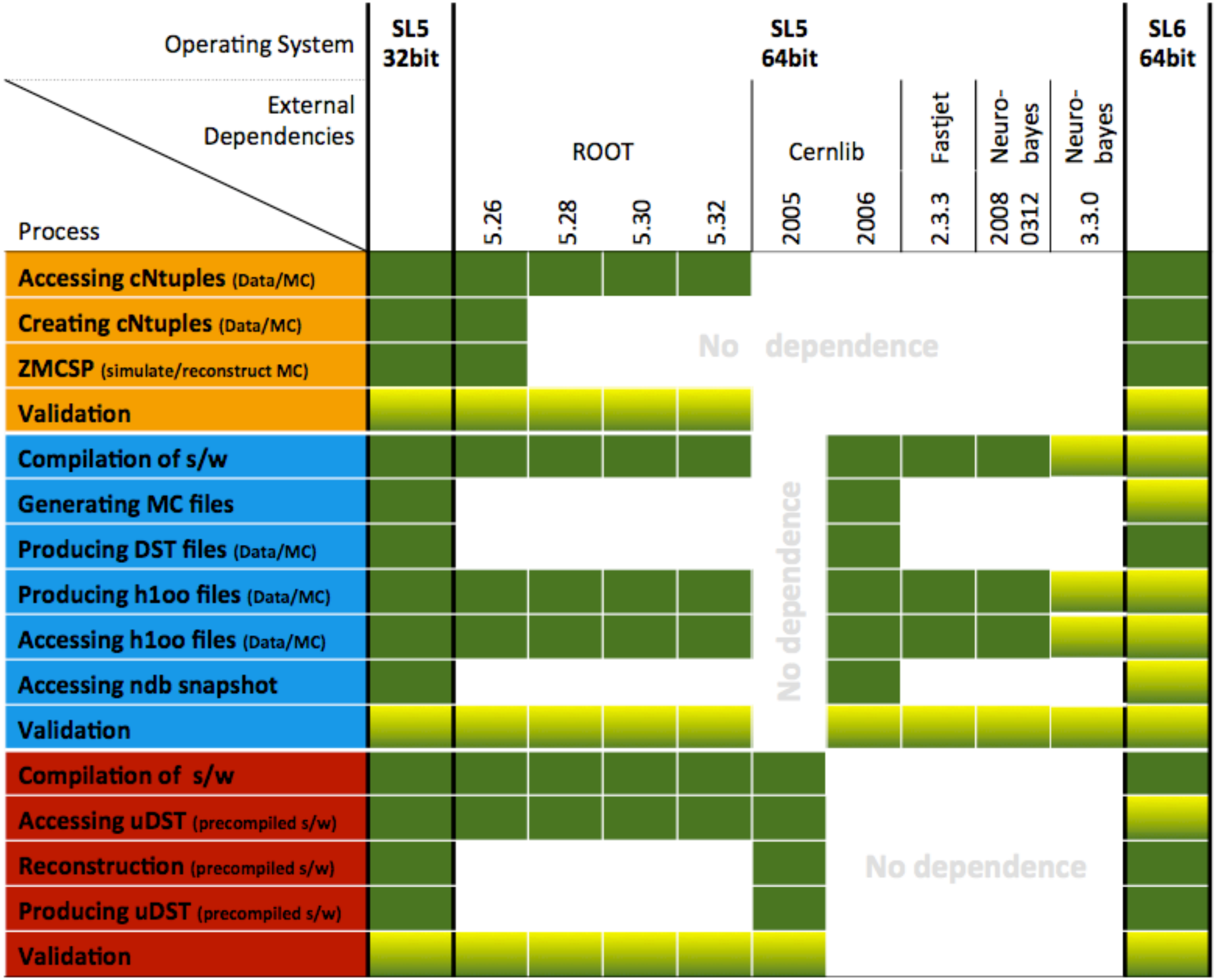
- Results remain accessible or can



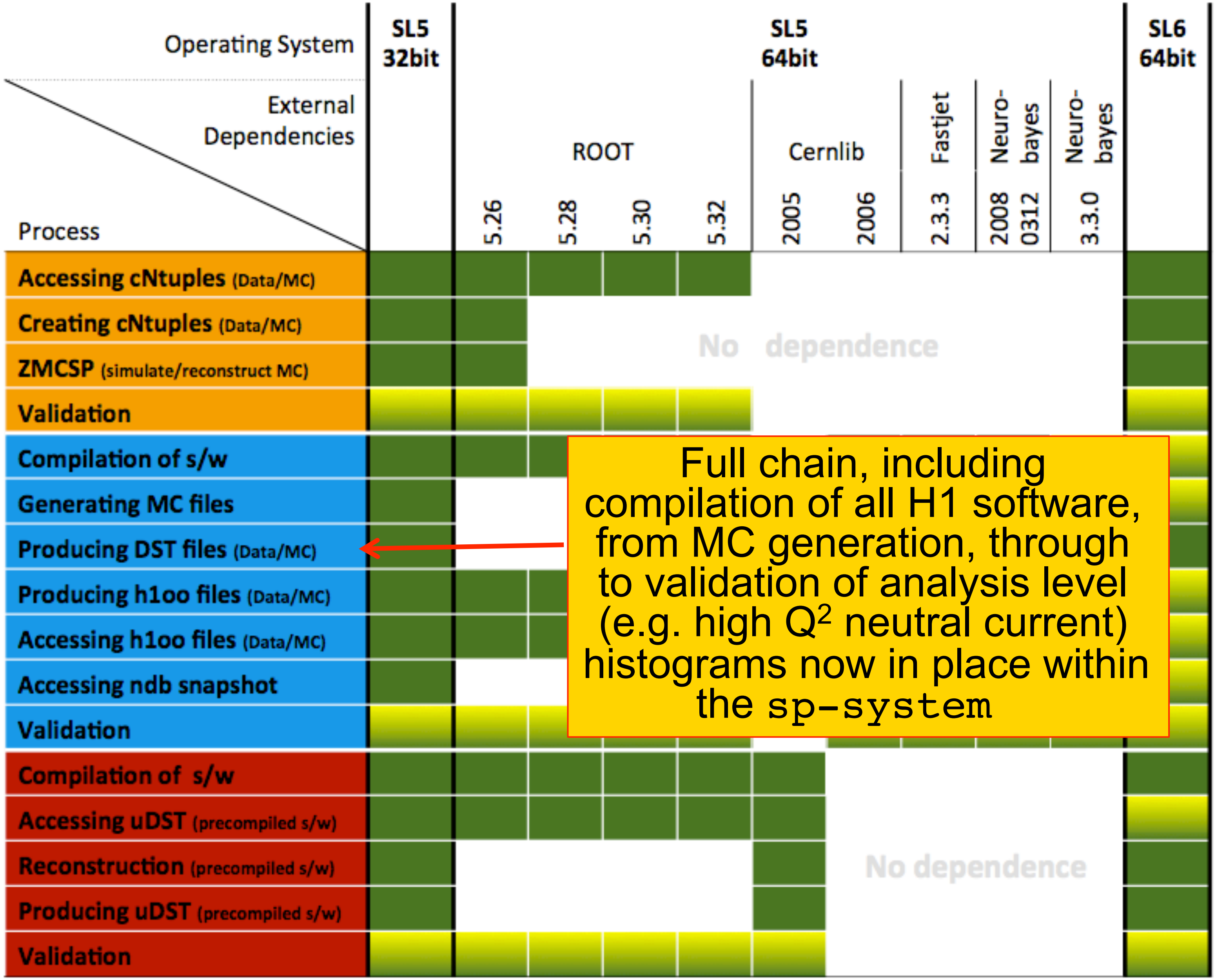
➔ It is essential to have robust definition of complete set of experimental tests

The nature and number dependent on desired preservation level

Putting it all together

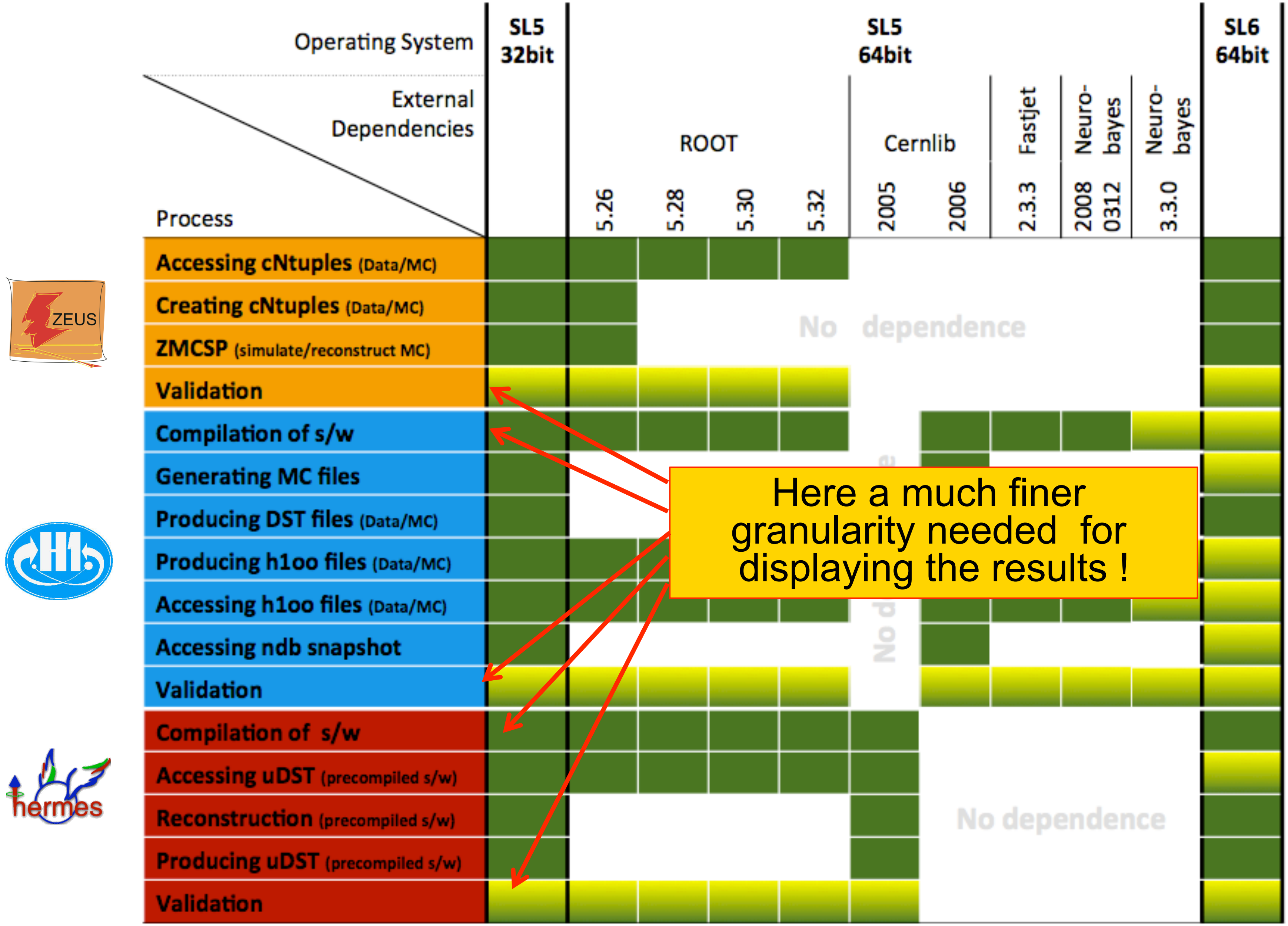


Putting it all together



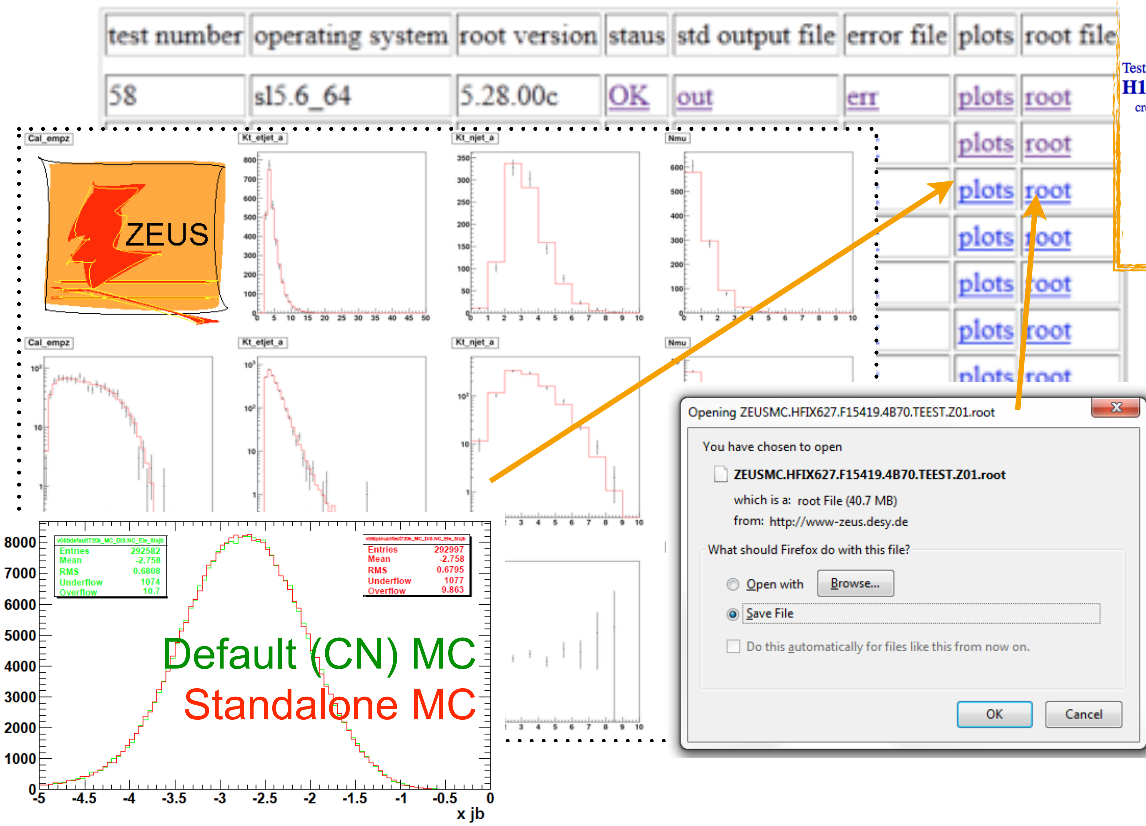
Full chain, including compilation of all H1 software, from MC generation, through to validation of analysis level (e.g. high Q^2 neutral current) histograms now in place within the `sp-system`

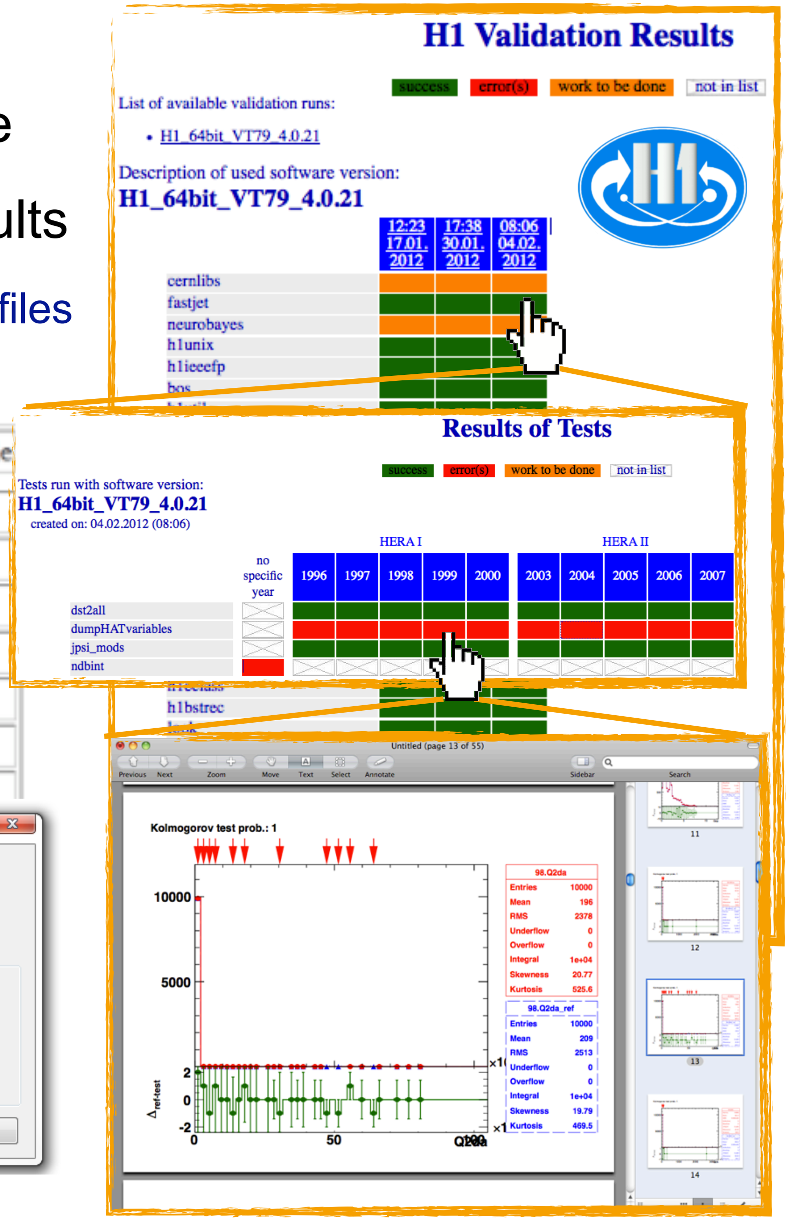
Putting it all together



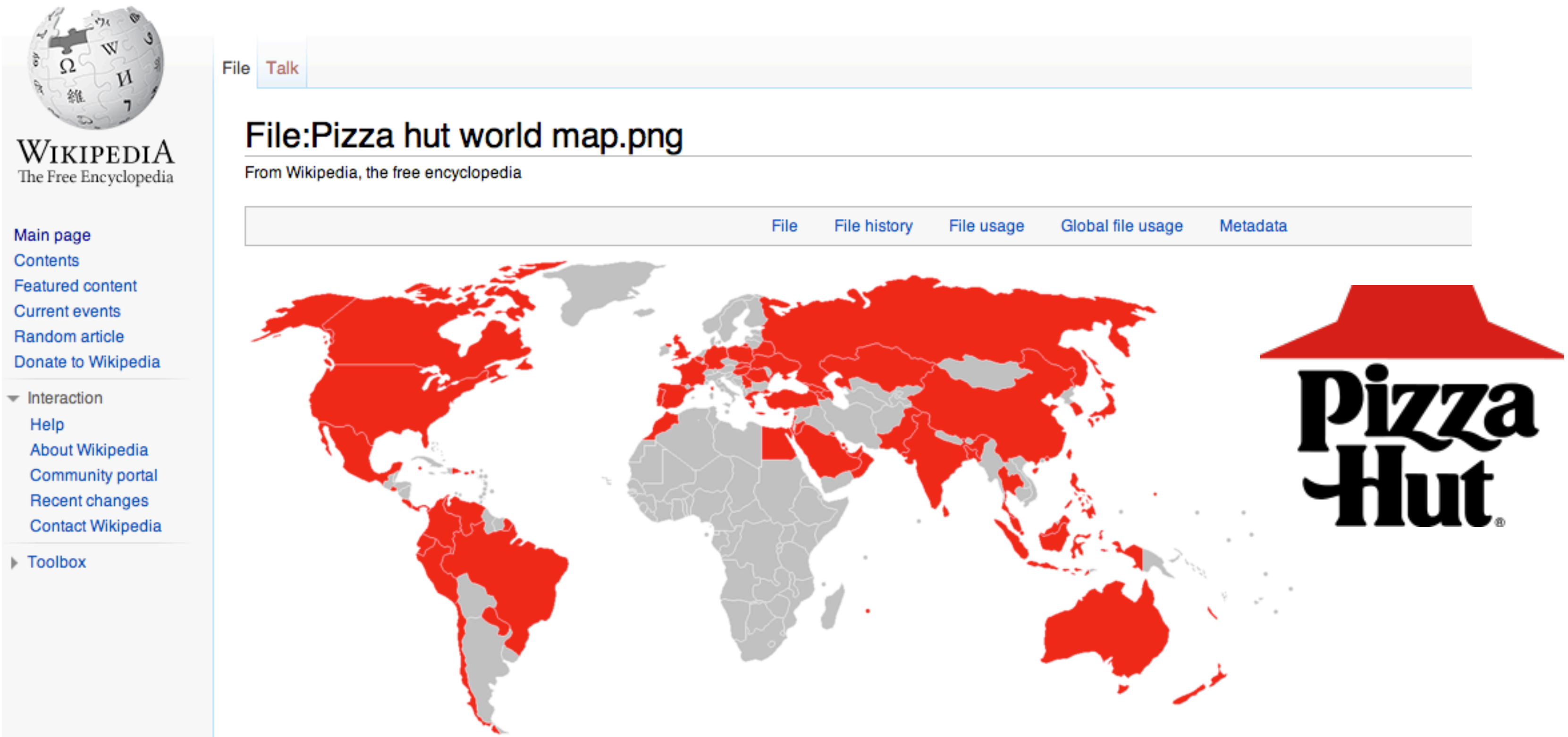
Digesting the validation results

- Display the results of the validation in a comprehensible way: web based interface
- The test determines the nature of the results
  - Could be simple yes/no, plots, ROOT files, text-files with keywords or length, ...





Deployment



- The whole point of the `sp-system` is ***not*** to provide a future resource for the experiments, but rather to provide a recipe which can be deployed
  - At DESY, this means for example exploring alternative resources such as the local BIRD cluster, the National Analysis Facility (dedicated to LHC, unlikely) or the Grid

Current status and future of the project

- In total more than 300 runs over sets of pre-defined tests have been performed within the `sp-system` by the HERA experiments
  - The experiments are in the process of migrating to SL6/64bit, and tests performed so far have already identified and helped to solve several long-standing bugs
  - The next challenges include Scientific Linux 7 compatibility issues with ROOT 6

- The process of defining and implementing the complete set of validation tests for the whole chain of software to be preserved is still on-going and is expected to take another year
  - Crucial period, future analysis of data from HERA dependent on long term access

- The concept of the `sp-system` at DESY is applicable to other HEP experiments, including those at the LHC
  - Additionally, the `sp-system` is also able to host validation initiatives from other experiments and the light structure allows migration of current tests into it

Summary

- DPHEP is established in the field of high energy physics for matters concerned with data preservation
  - Group is now in transition to a new Collaboration structure, first signatories in 2013
  - Many successful cross-experimental projects

- Future analysis of HEP data relies not only on reliable, long term data access but also on keeping the software and environment alive
  - We believe employing migration and in-depth validation provides greater longevity compared to basic virtualisation solutions – although virtualisation is still a crucial tool!
  - This includes compilation and examines all facets of the experiment software stack

- The solution offered, which uses virtualisation as a tool to provide a coherent, multi-versioned environment, is applicable to other HEP experiments and other scientific fields

arXiv:1310.7814