

Preparing experiments' software for long term analysis and data preservation – a view from DESY-IT

Yves Kemp & Dmitry Ozerov
DESY IT
CHEP 2012, 05/22/2012



ICFA Study Group on Data Preservation and Long Term Analysis in HEP

- > High Energy Physics experiments initiate with this Study Group a common reflection on data persistency and long term analysis in order to get a common vision on these issues and create a multi-experiment dynamics for further reference.
- > The objectives of the Study Group are:
 - Review and document the physics objectives of the data persistency in HEP.
 - Exchange information concerning the analysis model: abstraction, software, documentation etc. and identify coherence points.
 - Address the hardware and software persistency status.
 - Review possible funding programs and other related international initiatives.
 - Converge to a common set of specifications in a document that will constitute the basis for future collaborations.
- > Since August 2009, the Study Group is endorsed by ICFA (International Committee for Future Accelerators).



Preserving the ability to perform analysis

- > Of course you need the data to be preserved in first place
 - So complicated I assume someone else already does this ☺
- > Then you need access to this data
 - Here is where your IT department might come in – they drive the infrastructure
 - Will come to some of these in the second part
- > Your (old) code must run on (modern) machines
 - ... maybe incorporate new code and methods
 - Will come to this in the first part
- > And you will have “external” dependencies
 - Other software products, Databases (conditions...)
 - Will be covered both in the first in second part
- > The third part will be an example of setting up a scheme that we use for HERA experiments at DESY



Naïve approach: Virtualization does it all.

> Naïve scenario:

- You have your code in a **self-contained virtual machine**.
- You can **compile new code** in this VM.
- You **read data** from the central storage.
- You perform **all of your analysis** in this virtual machine.

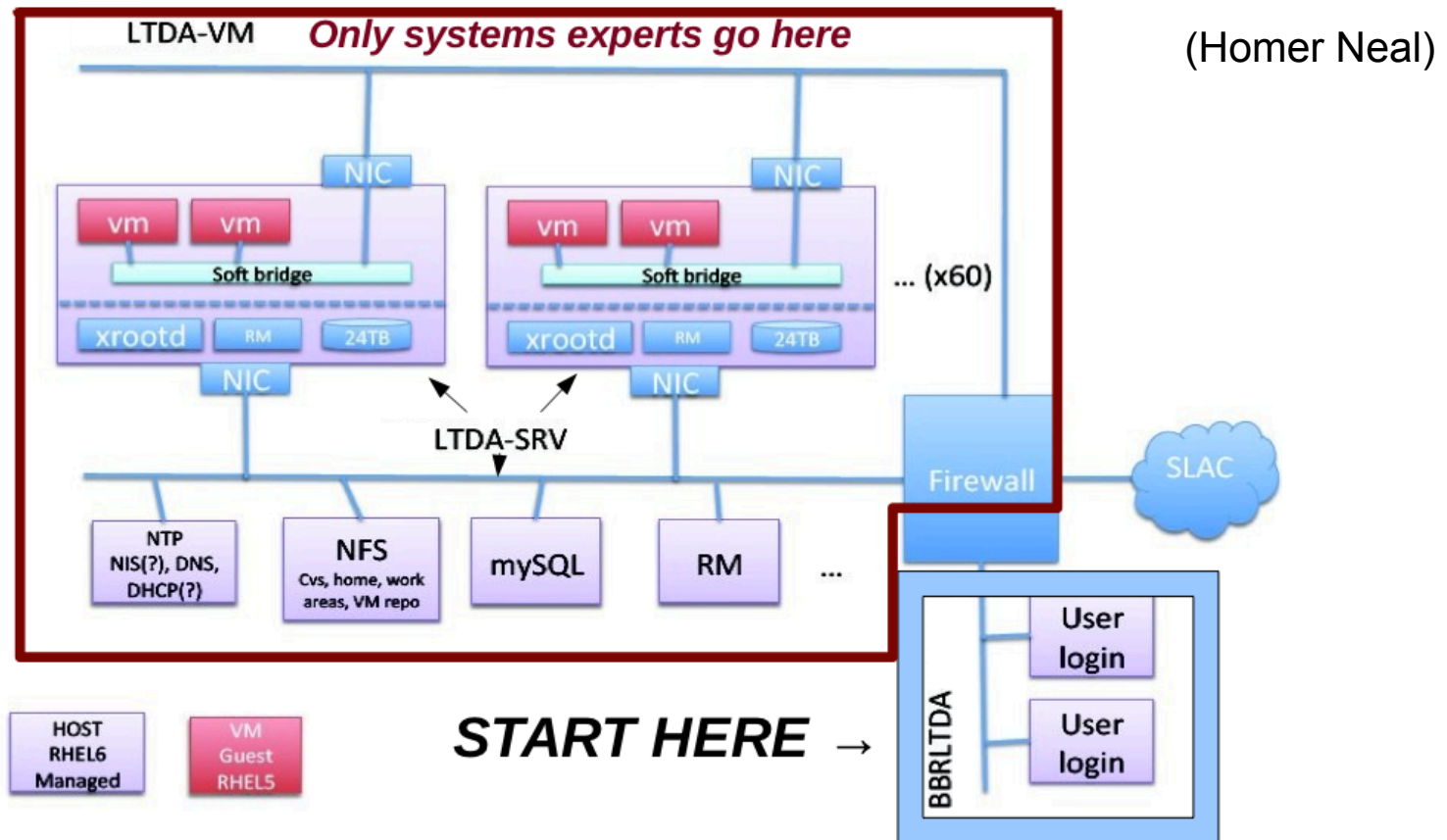
> Questions:

- Is the environment really so isolated and self-contained?
- Will you always be able to compile new code on an old OS?
- Your VM lives in a network: Will it integrate when needed?
- Will your storage have the same entry doors when needed?
- Is one VM enough for the analysis, or will you need plenty of these VMs?



Virtualization is an important tool – when used well

- You can plan a long-term analysis facility that handles these issues
 - BaBar plans to use the following system until 2018:



- Virtualization is part of the game here – but it certainly is not the naïve approach described before

Another approach using Virtualization: Pizza Preservation



How to preserve a pizza?

- > 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

> Pro's and con's of simply freezing... personal summary

- + Easy to do (manpower), easy to do (time)
- Operability of the software and correctness of results not guaranteed
- Changes if needed will become more difficult the longer SW is frozen

> Freezing SW OK if timeline and scope reduced

- E.g. makes perfectly sense for BaBar SW and analysis as BaBar: SuperB on the horizon

> ... but this is probably not the case for HERA: No successor experiment foreseen

- So, cook the same recipe ever and ever again, and validate the output - automatically



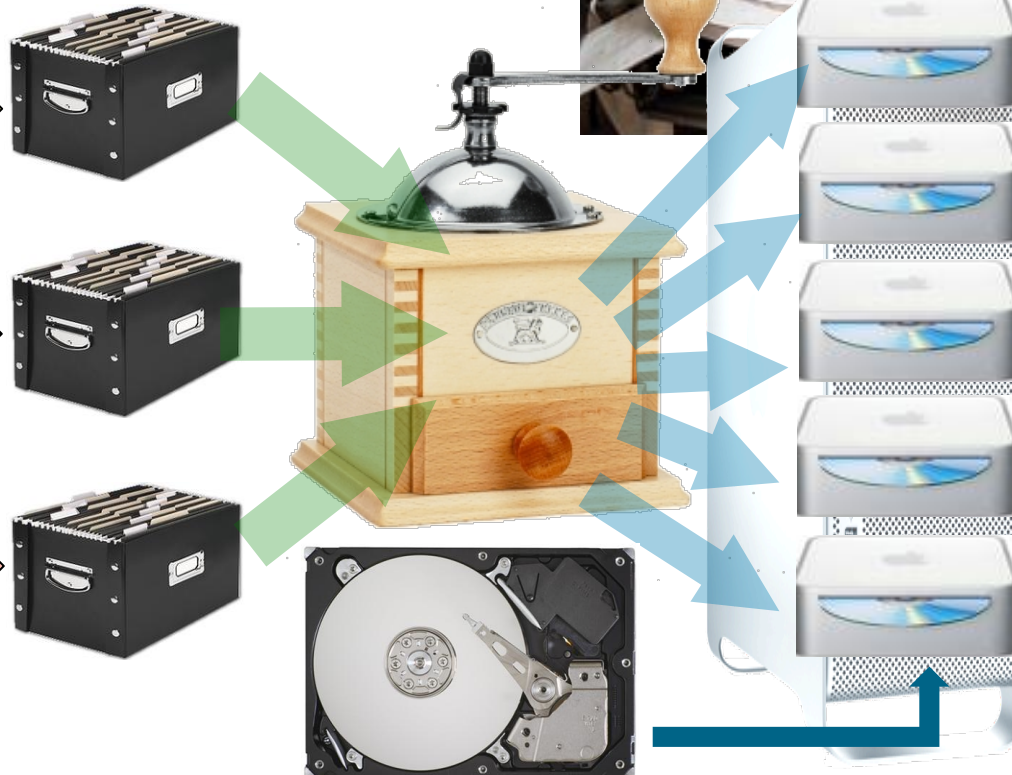
From Pizza to HEP analysis: The coffee-mill idea

**Virtualization is state-of-the-art
technology for automation of
the whole process**

H1 Software
Zeus Software
\$EXP Software

ROOT, GEANT,...
External SW

SL5/SL6/Debian/...
IT provides VMs



Test OK

**OS lib missing
→ IT**

**Tracking code error
→ EXP-SW**

**Data unreadable
→ IT & EXP-SW**

And cloud ... another buzz-word?

> If virtualization is used as a technology to

- Either decouple (new) hardware from (old) Operation System
- Or as a means to automate a process

> The back-end can be a cloud

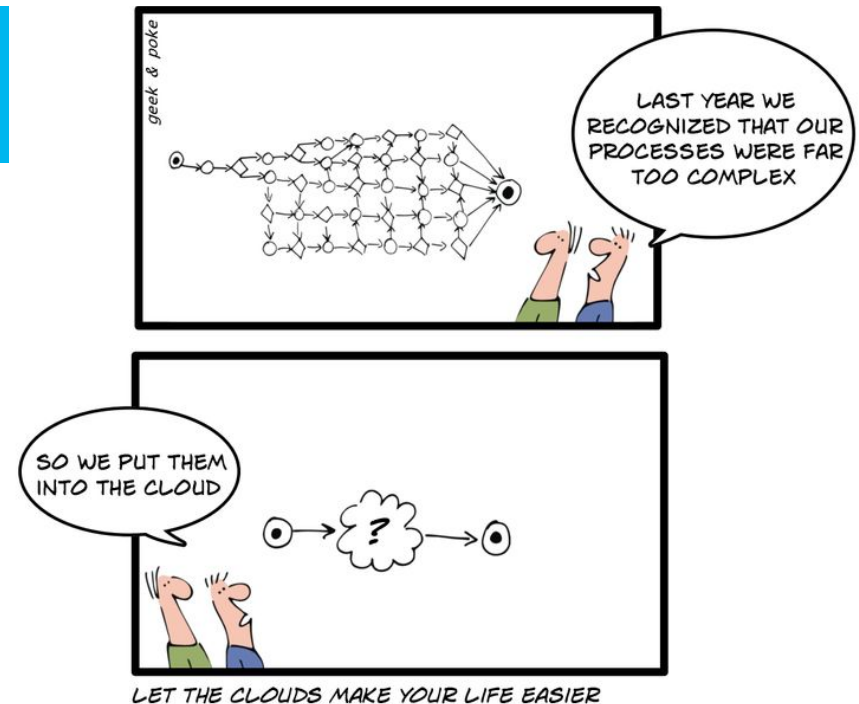
- Be it in your Computing Center
- Be it in the Computing Center of someone you pay for

> Just be careful: You need the data too!

- If the Cloud is in your Computing Center: (Relatively) Easy
- At \$AMAZON: Needs more thought, and potentially some money

> Interfaces to Cloud might/will change. Change is not in your hands.

> Clouds can help you – but are not a silver bullet per se

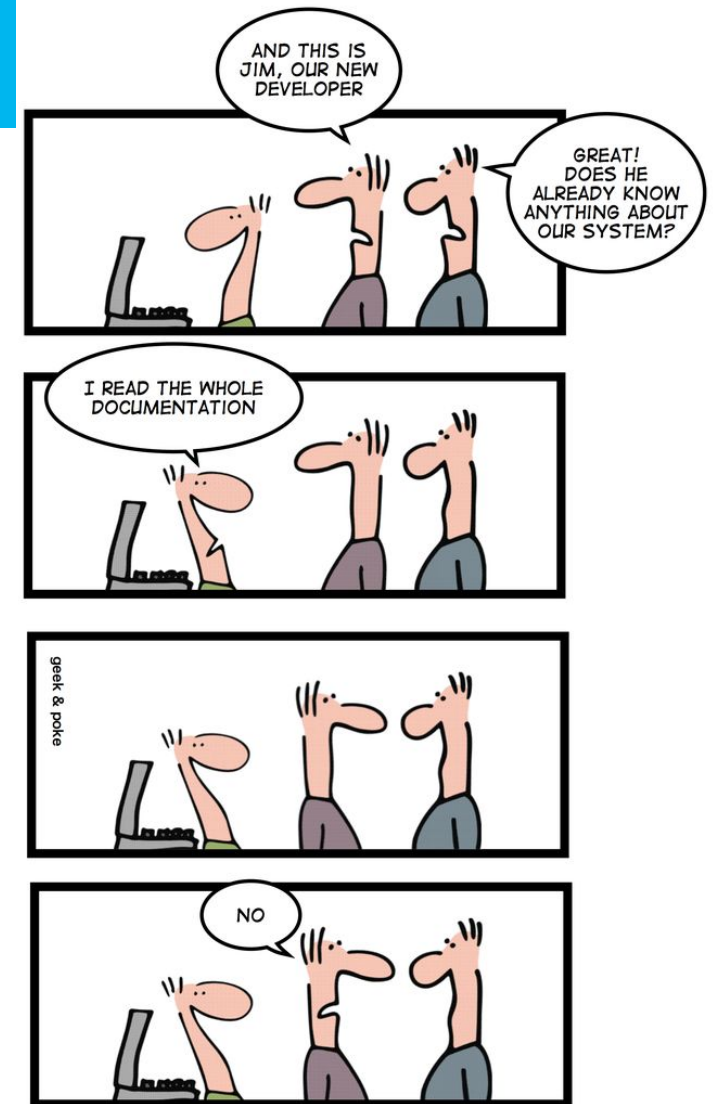


<http://geekandpoke.typepad.com/geekandpoke/2009/03/let-the-clouds-make-your-life-easier.html>



... all these are just tools

- > The most important questions to be answered by experiments are:
 - Is the code well documented?
 - Is the code adaptable to new environments?
 - Are the workflows well documented?
 - Are the external dependencies well known?
 - Are there checks that verify correctness of data and algorithms?
- > Virtualization and Clouds can assist preparing for long term analysis – but not replace the preparation



<http://geekandpoke.typepad.com/geekandpoke/2012/04/the-new-developer.html>

IT view: Experiments move from Active to “legacy”

Active period:

- Large user community
- Large fraction of used resources (“main usage in IT”)
- Development ongoing and adaption to changes
- Many responsible people both in Experiments and IT
- Financial support clear



© Warner Bros.



© Warner Bros.

Legacy, long-term-analysis period:

- Small community, only sporadic users
- Minor fraction of used resources
- Little development and adaption
- Small number of responsible people
- Financial support unclear



Persistent Lab department vs. “transient” experiments

- > Some lab departments usually are organized “persistently”
 - IT department, Library
- > Can take over some parts of experiments
 - Like hosting documentation – paper docu, web pages, ...
 - Digitizing and indexing information
 - Archiving experiments data
 - ...
- > Helping experiments to consolidate know-how and preserve it for the future
 - DESY-IT and Library strongly involved in HERA DPHEP activities
- > Prerequisite: There is a lab behind an experiment, and the lab shows some interest in the legacy experiment?



Resources for legacy experiments – wishes from IT

> Be as mainstream as possible

- OS: Be somewhat reactive and move with other user communities
- Dependencies should also be “mainstream” (e.g. no need for deprecated versions ...)

> Be as flexible as possible

- Things will change in the infrastructure: Server names, IP addresses and FW rules, ...
- Be able to adapt to these changes

> Use what is there

- Today the working horse in HEP is the Grid
- Tomorrow maybe Cloud – maybe not / what will come after Cloud?
- Be flexible and use resources that are there



... coming back to the pizza preservation

- Idea:**
- > Use virtualization techniques to repeatedly run well defined tests
 - > Perform checks against different and evolving environments
 - > Automatically check these results against predefined values
 - > Only notify when test results differ from these values
 - > Separate IT and Experiments duties

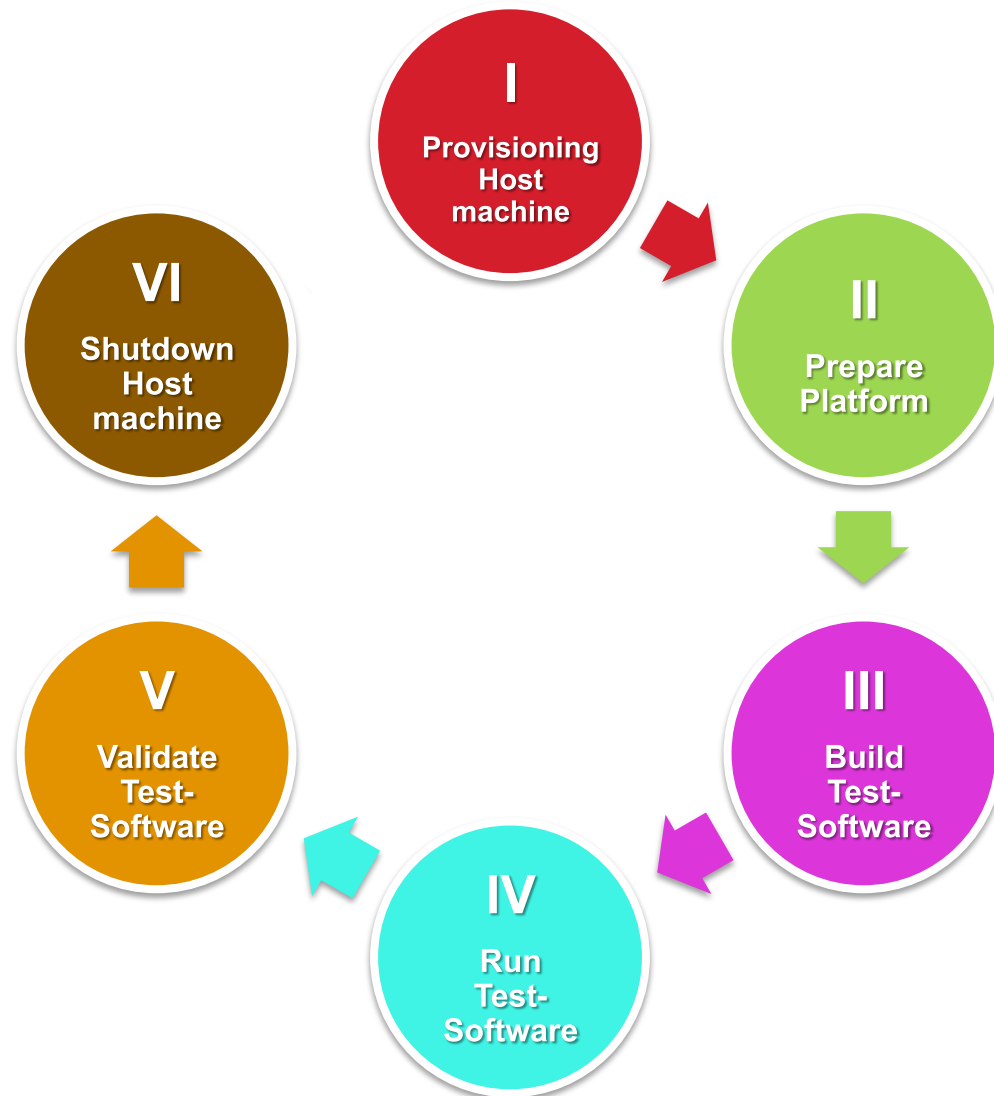


© AFP

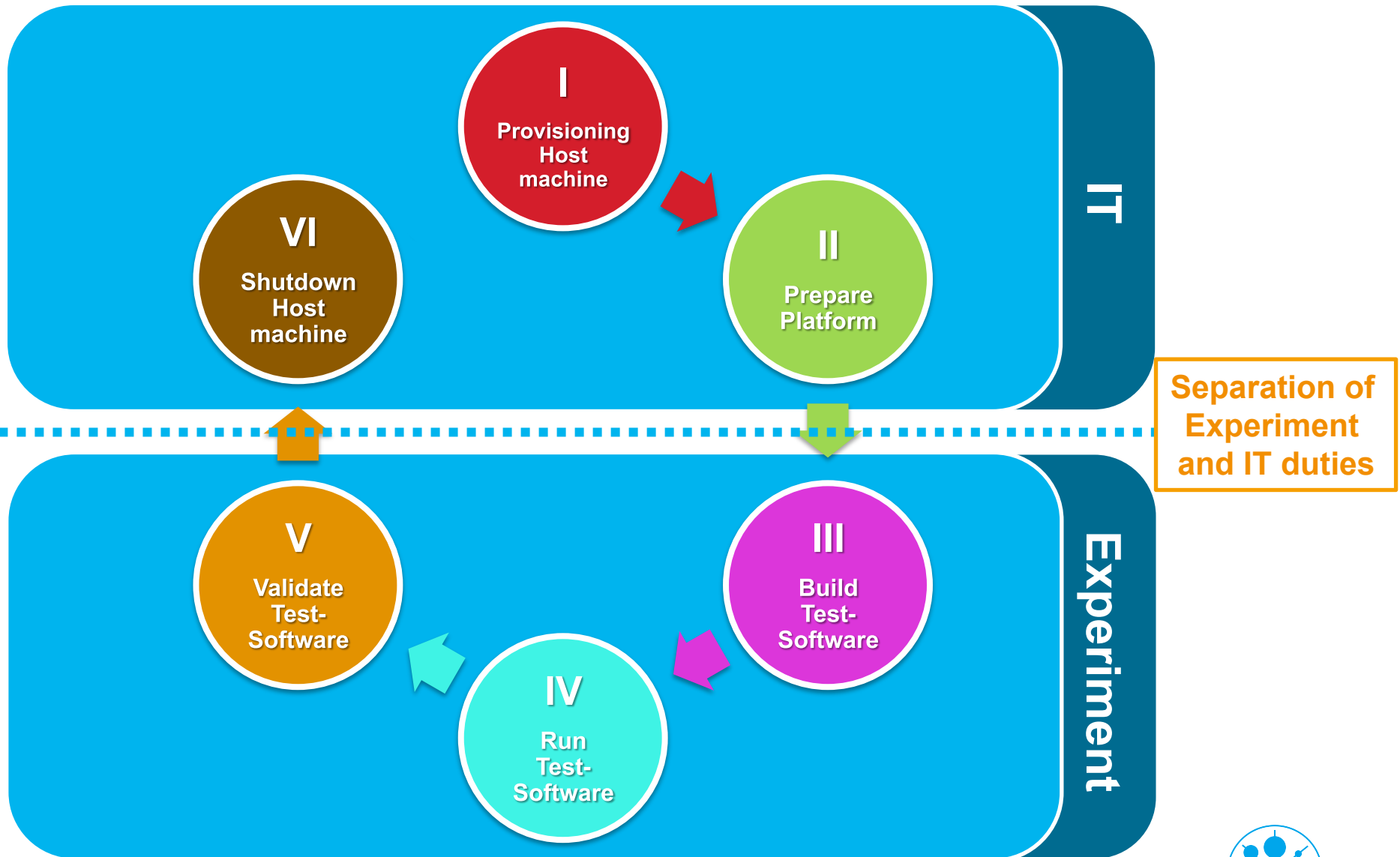
Implementation at DESY for HERA experiments:

- > Done by master student Marco Strutz
- > Enables preparation of a VM, complete run, and controlled shutdown
- > Enables separation between IT and experiments

The Generic Recipe (Atomic Test Life-Cycle)



... and the two cooks



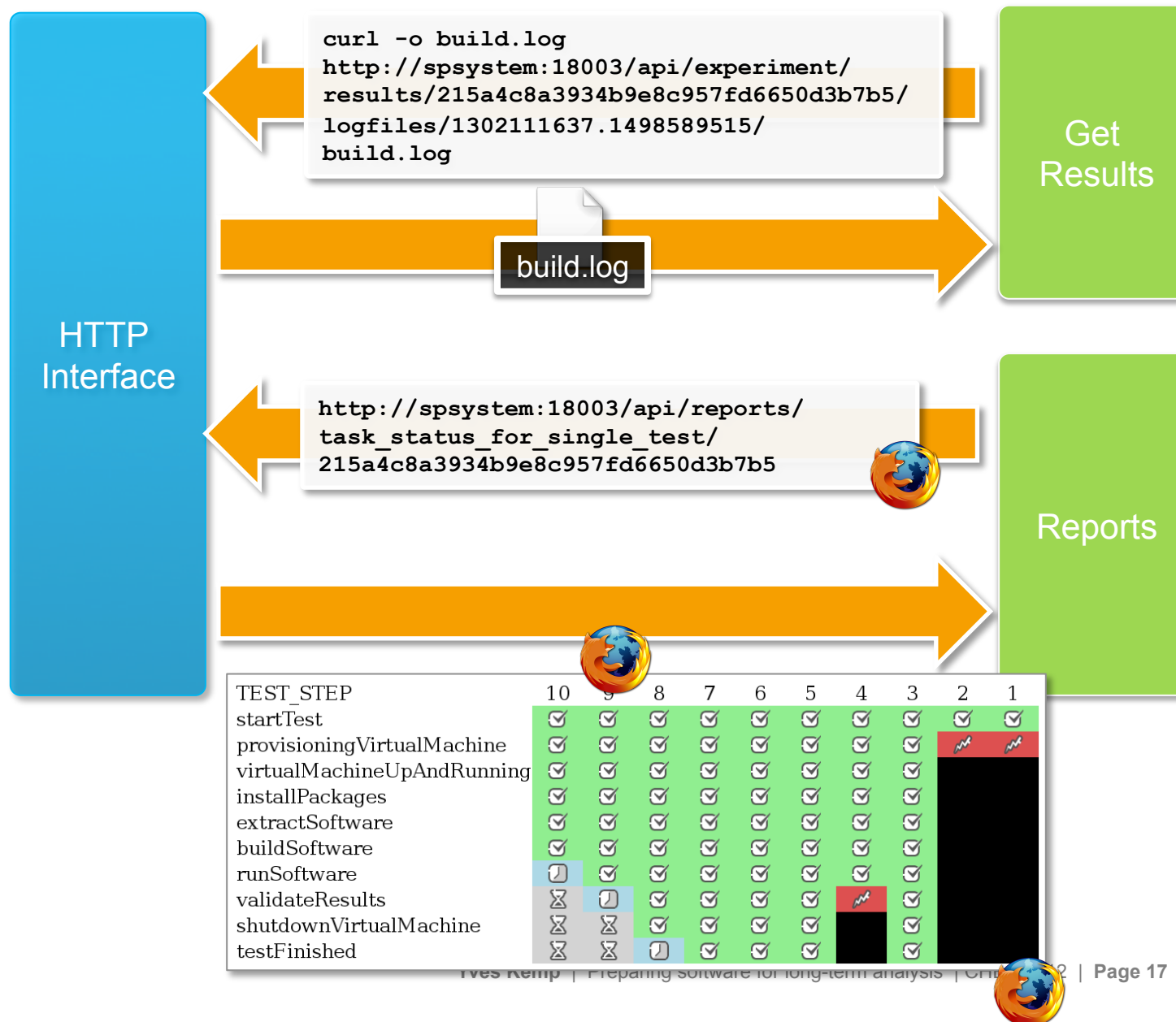
... and very short overview about what one needs to specify

I have some software.
What do I need to
provide you to use
your system?



- > The code:
 - E.g. Some ROOT internal test
- > A build.sh script
 - E.g. compile ROOT
- > A run.sh script
 - E.g. Run ROOT internal Stress Test
- > A validation.sh script
 - E.g. Validate the output of the Stress Test
- > Additional packages in the VM image
 - E.g. gcc in version 4.N.N
- > Information about the desired VM image
 - E.g. SL5.N 64bit

Get results from the test run



```
+ wget -q http://www.desy.de/~johndoe/virt/sw-test.tgz
+ tar xvfz sw-test.tgz
tar: sw-test.tgz: Cannot open: No such file or directory
tar: Error is not recoverable: exiting now
tar: Child returned status 2
tar: Exiting with failure status due to previous errors
+ chmod 755 swmc_run
chmod: cannot access `swmc_run': No such file or directory
+ ./swmc_run config1234 seed 9876
/home/dpheap/application.sh: line 9: ./swmc_run: No such file or directory
+ ls -ltra
total 32
-rw-r--r--. 1 dpheap dpheap 124 Mar 31 2010 .bashrc
-rw-r--r--. 1 dpheap dpheap 176 Mar 31 2010 .bash_profile
-rw-r--r--. 1 dpheap dpheap 18 Mar 31 2010 .bash_logout
drwxr-xr-x. 2 dpheap dpheap 4096 Mar 31 2010 .gnome2
drwxr-xr-x. 4 dpheap dpheap 4096 May 13 2010 .mozilla
drwxr-xr-x. 3 root root 4096 Jun 7 2010 ..
drwx-----. 4 dpheap dpheap 4096 Mar 19 19:04 .
-rwxr-xr-x. 1 dpheap dpheap 167 Mar 19 19:04 application.sh
```

Task	1	2	3	4	5	6	7	8	9	10
buildSoftware	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
runSoftware	⌚	✓	✓	✓	✓	✓	✓	✓	✓	✓
validateResults	⌚	⌚	✓	✓	✓	✓	✓	✓	⚡	✓
shutdownVirtualMachine	⌚	⌚	✓	✓	✓	✓	✓	✓	⬛	✓
testFinished	⌚	⌚	⌚	✓	✓	✓	✓	✓	✓	✓



First results and lessons -

- > Framework fits well for repeated similar and isolated actions
- > Not well suited for development phase itself – no fast turnaround
- > Experiments needed dedicated storage
 - Common input to virtual machines
 - Transient storage for inter-VM exchange in complex workflows
 - Easy-access store for logs and result files
- > Decision: Do not use this framework during test development by HERA experiments
 - Instead use a similar framework with fast turnaround – without the automation
 - With a simple method of storing information: Local storage to VM
- > Future: When tests are written and finalized, decide on details of a future automated test framework with storage



Short description of the current framework for HERA@DESY

- > Always-On VMs with vanilla environment provided by IT
 - No VM configuration and start-up delay
 - Automation only to a small extent
- > Job submission via a batch-like interaction
 - No direct “ssh” to VM
 - All steps are detailed like in a “pizza recipe”
- > No HERA experiment specific on VM
- > No access to AFS that “hides” dependencies
- > Common internal storage for exchange between different tests and input for large files
 - Storage local to the VM





Status of HERA Experiments' Software



Process \ External Dependencies	SL5 32bit					SL5 64bit						SL6 64bit
	← Reference	ROOT				Adamo	Cernlib		Fastjet	Neuro-bayes	Neuro-bayes	
		5.26	5.28	5.30	5.32		2005	2006	2.3.3	2008 0312	3.3.0	
Accessing common ntuples												
ZMCSP (simulate/reconstruct MC)						No dependence						
Creating common ntuples												
Validation												
Compilation of s/w						No dependence						
Generating MC files							Use newer version					
Producing DST files												
Producing h1oo files												
Accessing h1oo files												
Accessing ndb snapshot												
Validation												
Compilation of s/w												
MC generation & digitisation												
Reconstruction												
Producing uDST												
Analysing uDST (Fortran, HANNA++)												
Validation												



(Michael Steder)



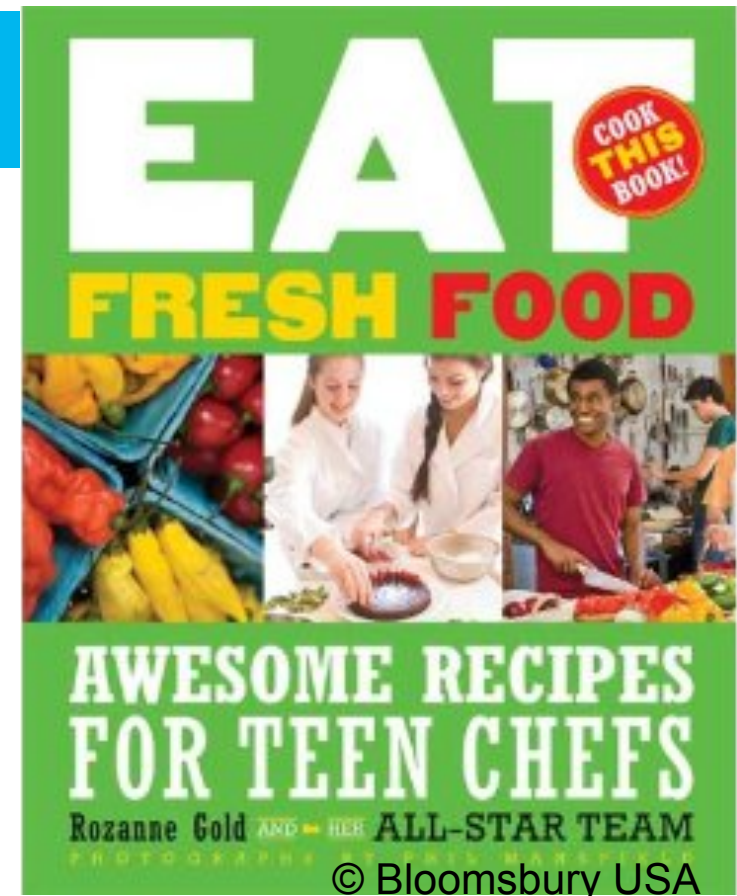
Summary

Preparing experiments for long term analysis is challenging – even if we let aside the whole data preservation itself.

Virtualization and Cloud technologies can help to some extent

Start as early as possible, best keep the software alive

Legacy experiments should make it as easy as possible to IT departments to support them



- 607 - Plenary talk by David South Wednesday 9:30-10:00
- 464 – Poster by M. Steder on Thursday (H1)
- 482 – Poster by E. Avetysian Thursday (HERMES)
- 462 – Poster by K. Wichmann on Thursday (Zeus)
- DPHEP Session on Thursday 13:30-19:25 in Room 808

