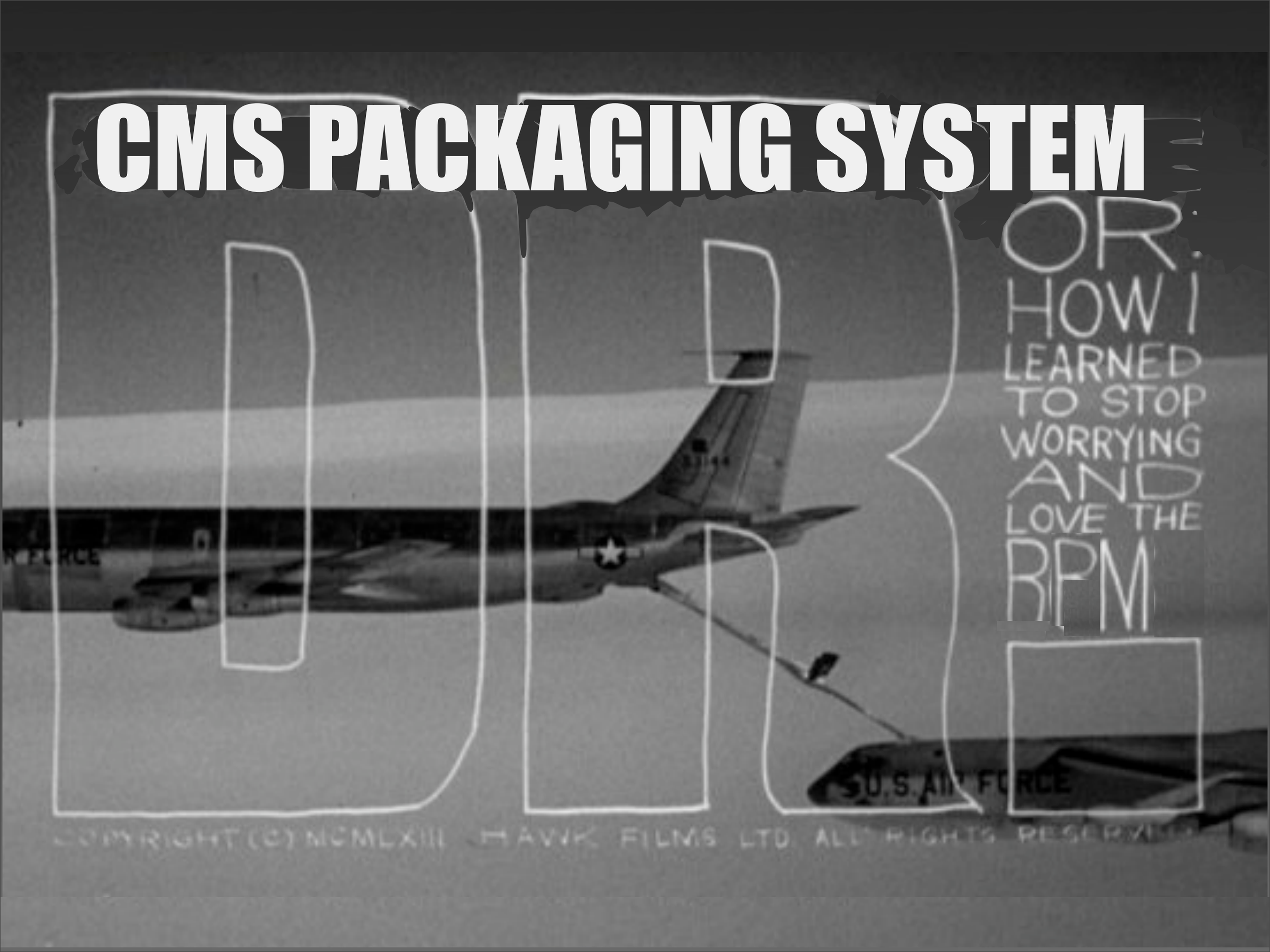
CMS PACKAGING SYSTEM



Why bother?
Motivations for a software packaging and distribution system

Offline manager:

- Quick turnaround from end of integration to release
- Release validability
- Release reproducibility
- All the offline software, not just the simulation/reconstruction one

Computing manager:

- Quick turnaround from software release to deployment.
- Easy deployment procedure.
- Robust distribution system.
- Automated procedure.
- Avoid single point of failure.
- Ability to exploit new hardware.

Tier-N admin:

- Easy installation procedure.
- As automated procedure as possible...
- ... yet allowing the possibility to follow it step by step.
- Standard, well known, tools.

Release integrator:

- Easy procedure to build software
- Easy procedure to build/update externals
- Easy multi-platform builds
- No additional steps for packaging
- Reproducible builds
- Everything under control

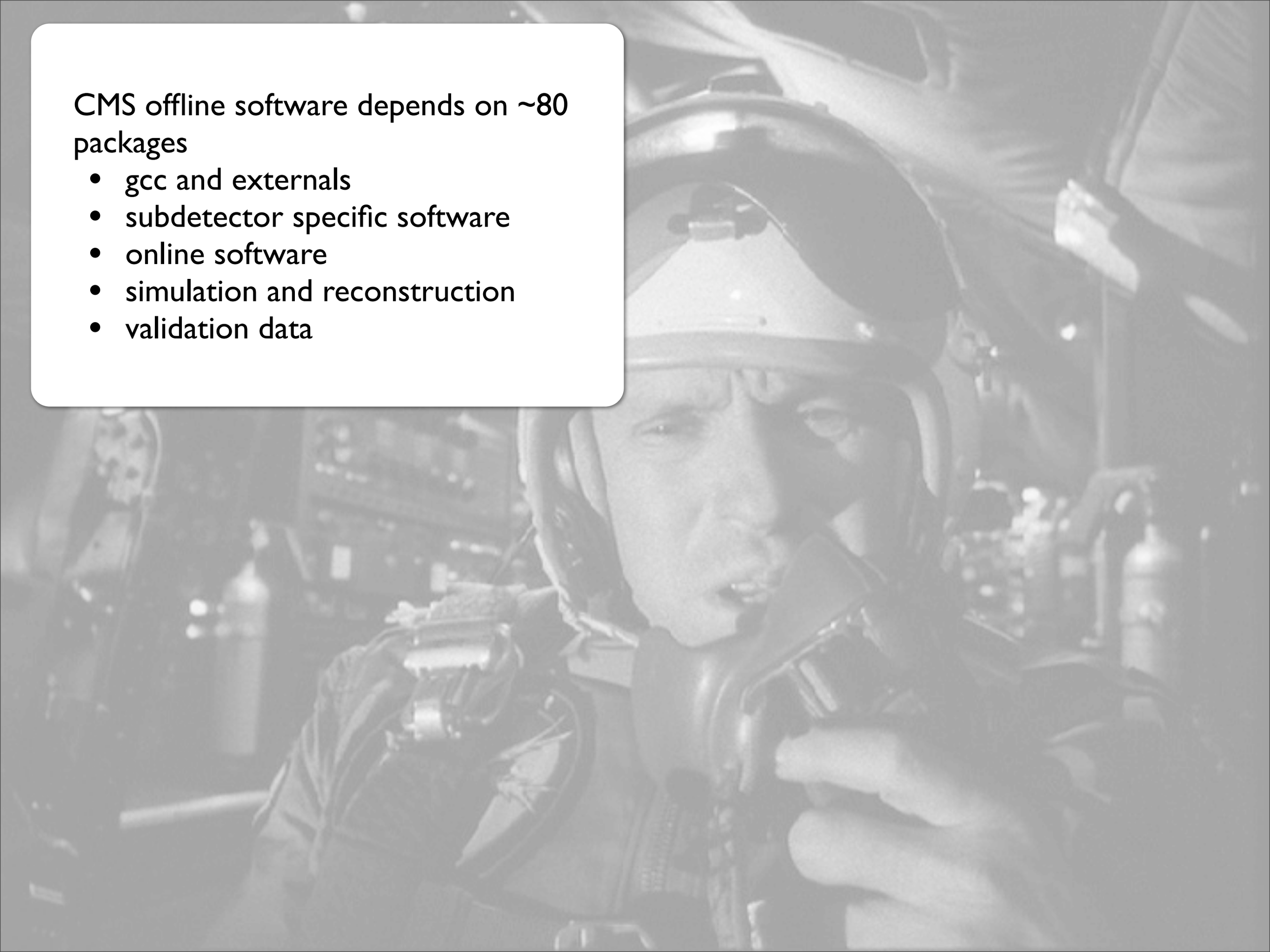
Physicists point of view

- Software availability
- Possibility to install it on his notebook/desktop/cluster
- Easy installation procedure
- Release reproducibility

...we all know it's needed...
...we all know it's painful...
...**very** painful for HEP software...

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data



CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data

1.7GB of packages
3.5GB of installed software

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data

1.7GB of packages
3.5GB of installed software

1 Tier 0
7 Tier 1
49 Tier 2
~8K worker nodes
3K people

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data

1.7GB of packages
3.5GB of installed software

1 Tier 0
7 Tier 1
49 Tier 2
~8K worker nodes
3K people

3 supported platforms:

- slc3_ia32_gcc323
- slc4_ia32_gcc345
- slc4_ia32_gcc345 on 64 bit hw

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data

1.7GB of packages
3.5GB of installed software

Several “wanted” platforms

1 Tier 0
7 Tier 1
49 Tier 2
~8K worker nodes
3K people

3 supported platforms:

- slc3_ia32_gcc323
- slc4_ia32_gcc345
- slc4_ia32_gcc345 on 64 bit hw

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data

1.7GB of packages
3.5GB of installed software

Several "wanted" platforms

- slc4_ia32_gcc4XX

1 Tier 0
7 Tier 1
49 Tier 2
~8K worker nodes
3K people

3 supported platforms:

- slc3_ia32_gcc323
- slc4_ia32_gcc345
- slc4_ia32_gcc345 on 64 bit hw

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data

1.7GB of packages
3.5GB of installed software

Several "wanted" platforms

- slc4_ia32_gcc4XX
- slc4_amd64_gccXXX

1 Tier 0
7 Tier 1
49 Tier 2
~8K worker nodes
3K people

3 supported platforms:

- slc3_ia32_gcc323
- slc4_ia32_gcc345
- slc4_ia32_gcc345 on 64 bit hw

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and reconstruction
- validation data

1.7GB of packages
3.5GB of installed software

Several "wanted" platforms

- slc4_ia32_gcc4XX
- slc4_amd64_gccXXX
- osx104_ia32_gcc401/ osx104_pcc32_gcc401

1 Tier 0
7 Tier 1
49 Tier 2
~8K worker nodes
3K people

3 supported platforms:

- slc3_ia32_gcc323
- slc4_ia32_gcc345
- slc4_ia32_gcc345 on 64 bit hw

CMS offline software depends on ~80 packages

- gcc and externals
- subdetector specific software
- online software
- simulation and recon
- validation data

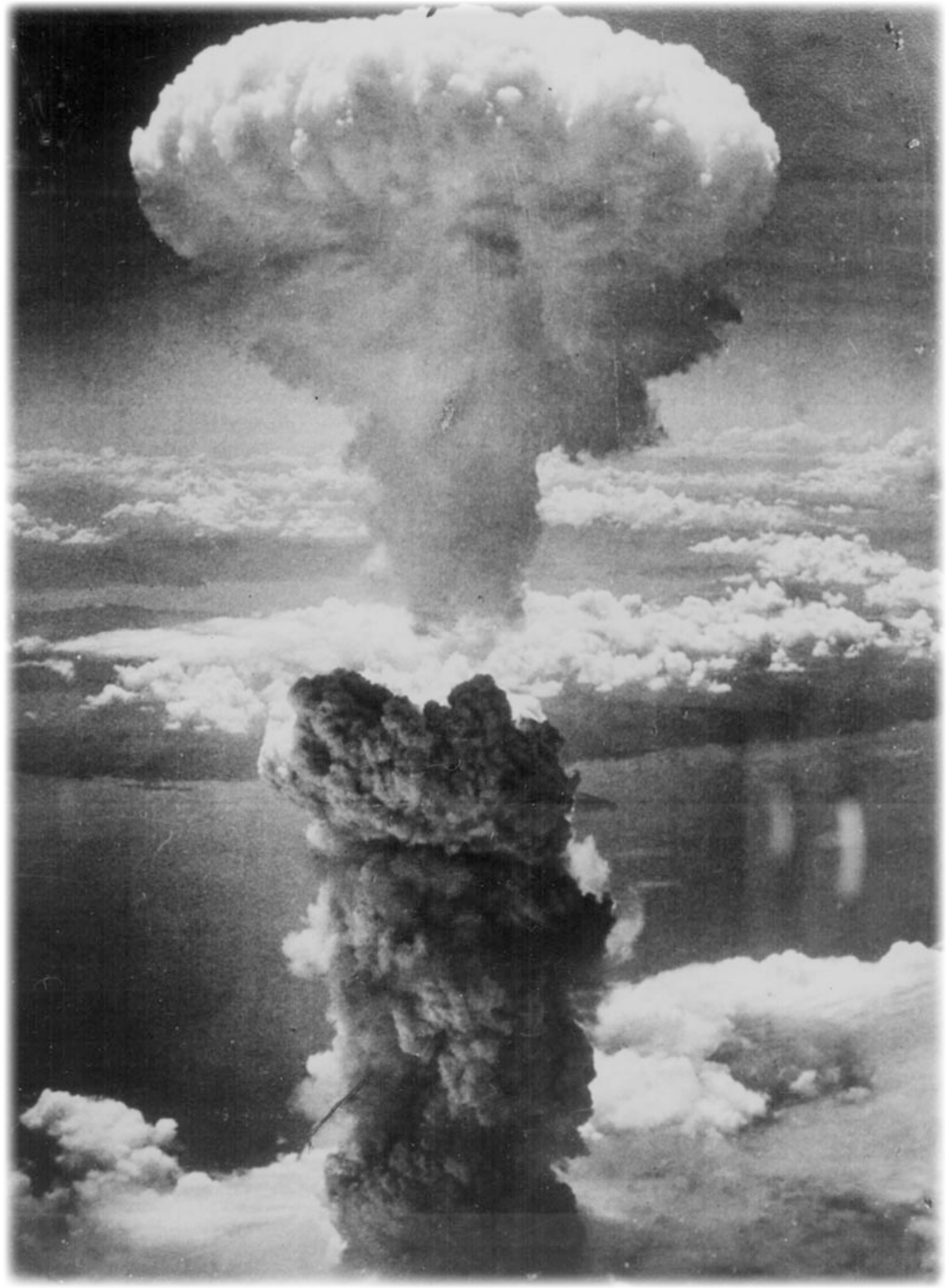
1.7GB of packages
stalled software

Several "wanted" platf
- slc4_ia32_gcc4XX
- slc4_amd64_gccX
- osx104_ia32_gcc401/ osx104_pcc32_gcc401

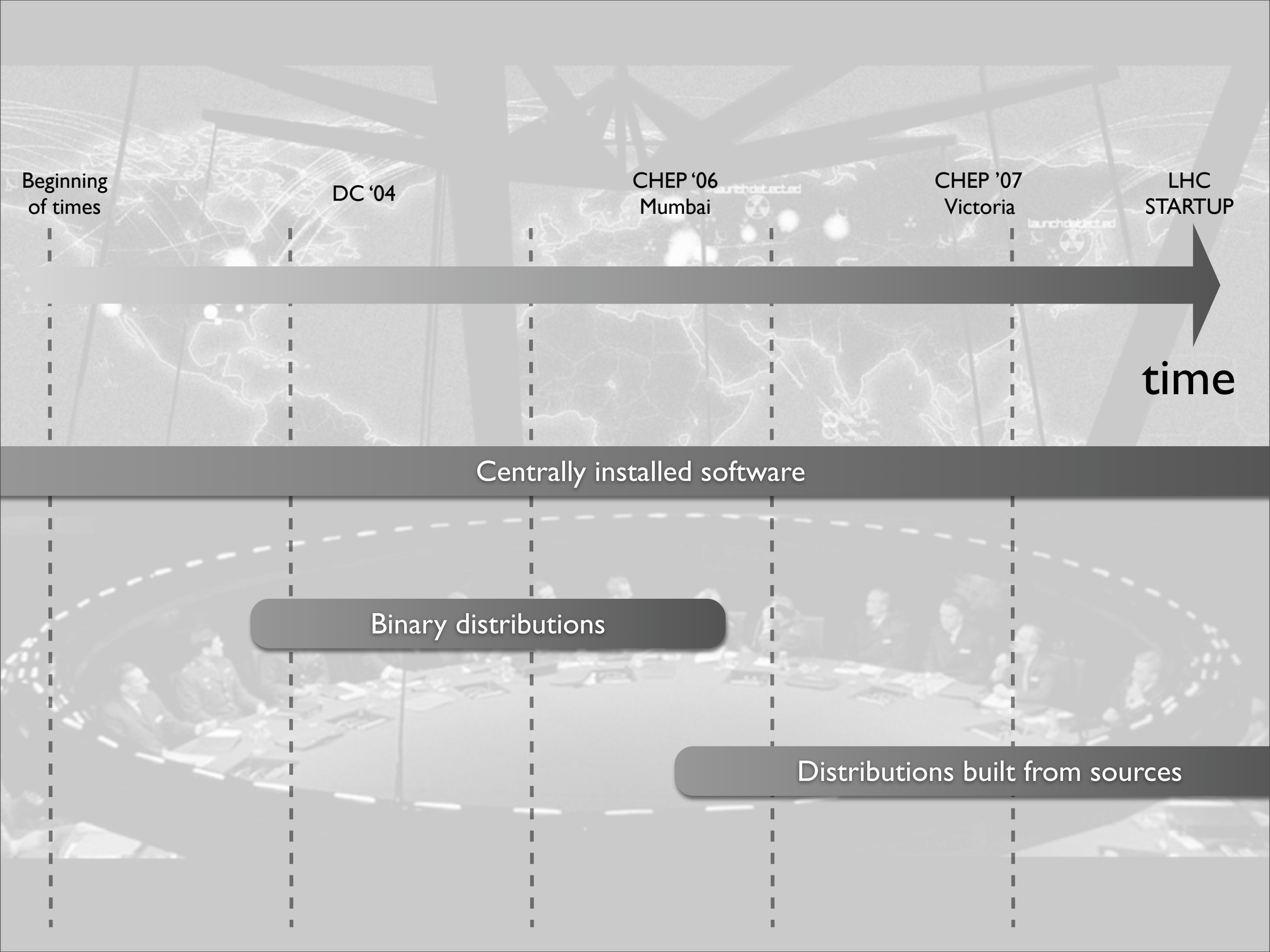
0
1
2
~8K worker nodes
3K people

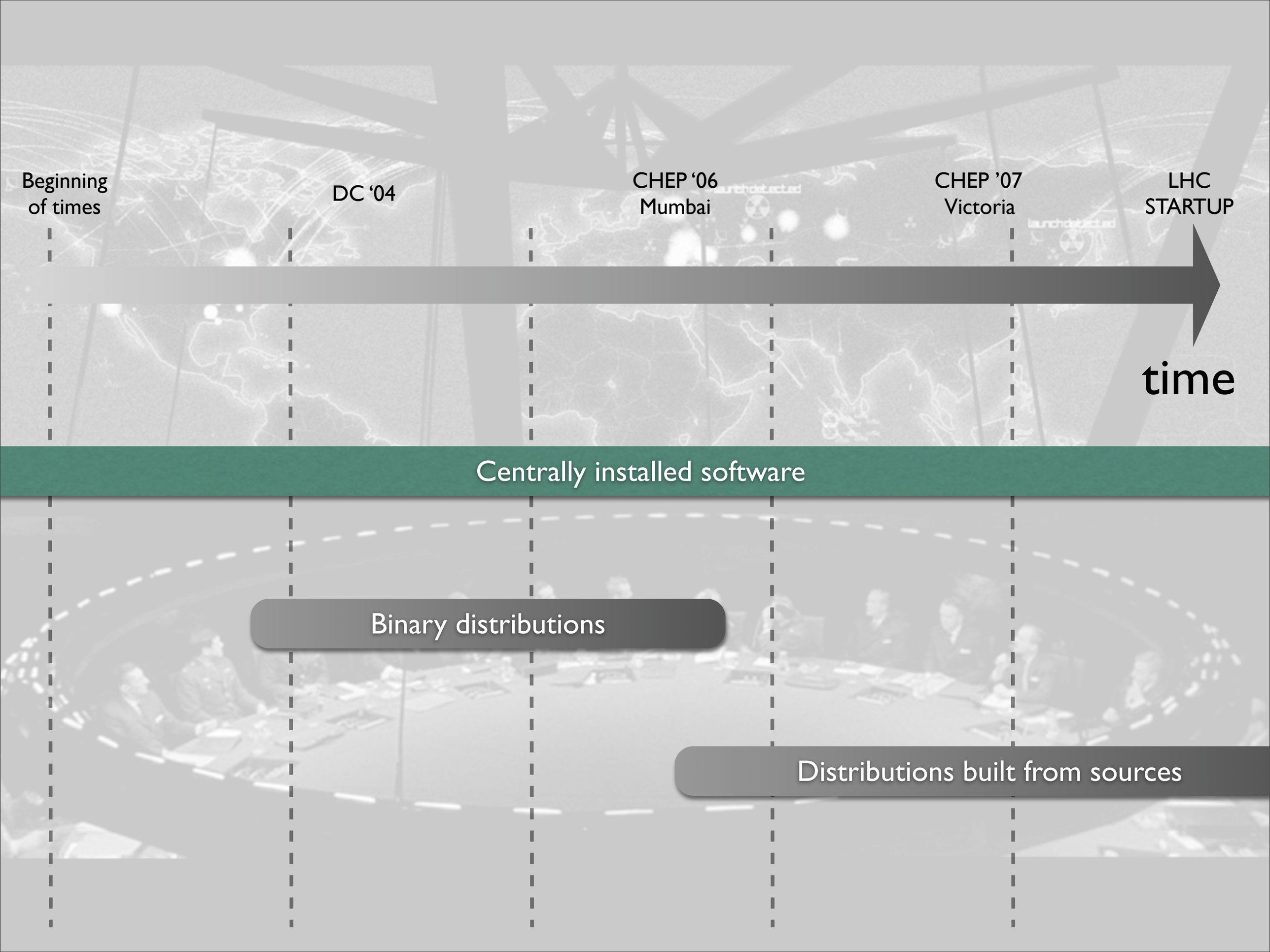
3 supported platforms:

- slc3_ia32_gcc323
- slc4_ia32_gcc345
- slc4_ia32_gcc345 on 64 bit hw



This talk will show you how it is possible to build and distribute CMS software from scratch in a few of hours.


Beginning of times
DC '04
CHEP '06 Mumbai
CHEP '07 Victoria
LHC STARTUP
time
Centrally installed software
Binary distributions
Distributions built from sources


Beginning of times
DC '04
CHEP '06 Mumbai
CHEP '07 Victoria
LHC STARTUP
time
Centrally installed software
Binary distributions
Distributions built from sources

Central installation

- externals configuration for a release is decided
- externals magically appear on AFS
- experiment software is configured to use the centrally installed externals
- software is built over AFS
- software is available over AFS
- software is validated at CERN

Central installation

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

software is available over AFS

software is validated at CERN

Solved issues:

✓ Software is actually built and available

Central installation

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

software is available over AFS

software is validated at CERN...ONLY!

Pending issues:

➡CERN only

Central installation

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

software is available over AFS

software is validated at CERN...ONLY!

Pending issues:

➡ Slow (AFS!)

Central installation

externals configuration for a release is decided (by LCG)

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

software is available over AFS

software is validated at CERN...ONLY!

Pending issues:

➡No control on the release procedure

Central installation

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

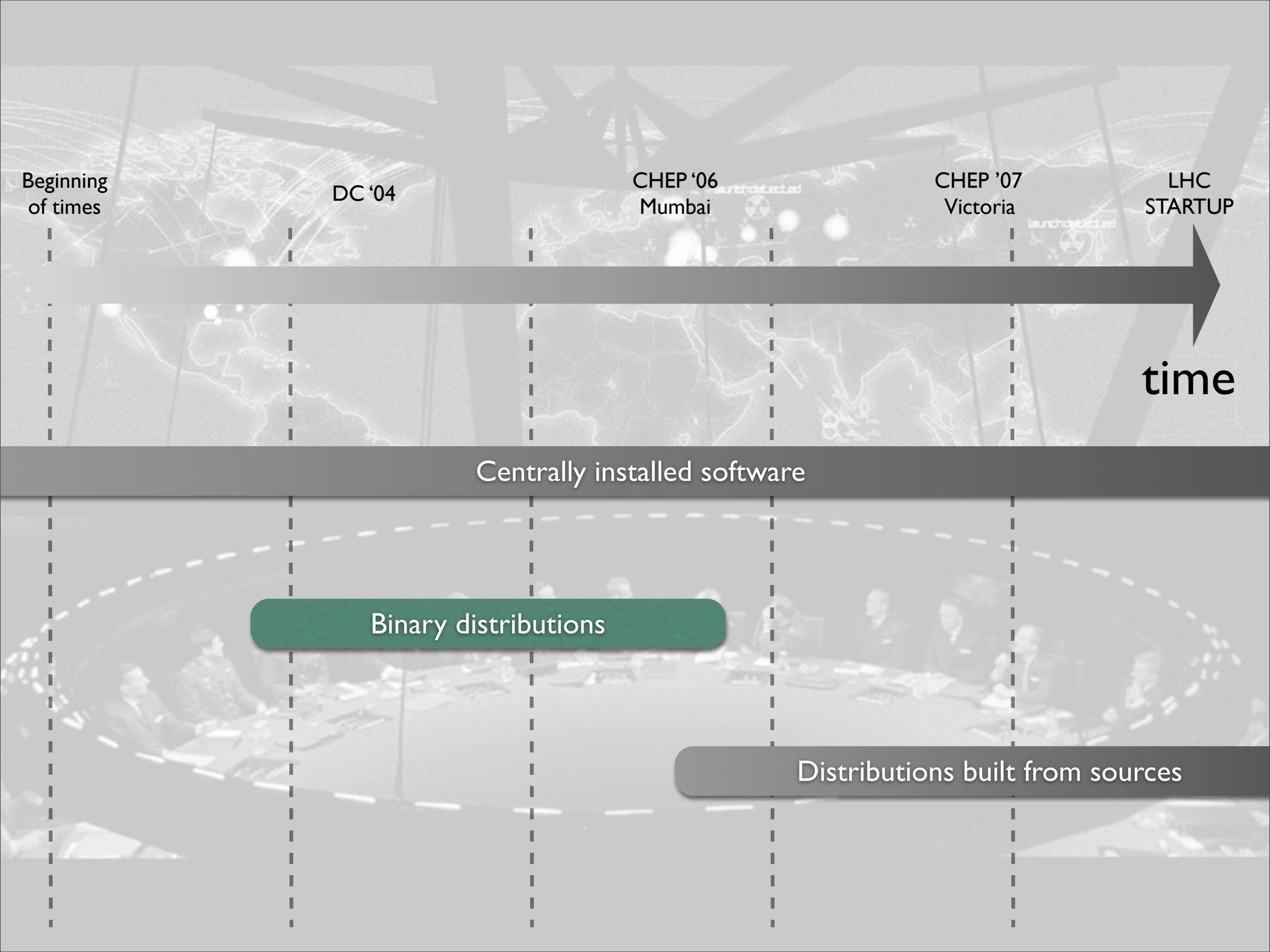
software is built over AFS

software is available over AFS

software is validated at CERN...ONLY!

Pending issues:

➡ No reproducibility


Beginning of times
DC '04
CHEP '06 Mumbai
CHEP '07 Victoria
LHC STARTUP
time
Centrally installed software
Binary distributions
Distributions built from sources

Binary packaging

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

software is available over AFS

software is validated at CERN

AFS area is packaged "as is"

Packages are distributed to other sites

Two additional steps added

Binary packaging

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

software is available over AFS

software is validated at CERN...ONLY!

AFS area is packaged "as is"

Packages are distributed to other sites

✓ Software can be installed outside CERN

Binary packaging

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

software is available over AFS

software is validated at CERN

AFS area is packaged "as is"

Packages are distributed to other sites

➡ All other issues are still there!

Binary packaging

externals configuration for a release is decided

externals magically appear on AFS

experiment software is configured to use the centrally installed externals

software is built over AFS

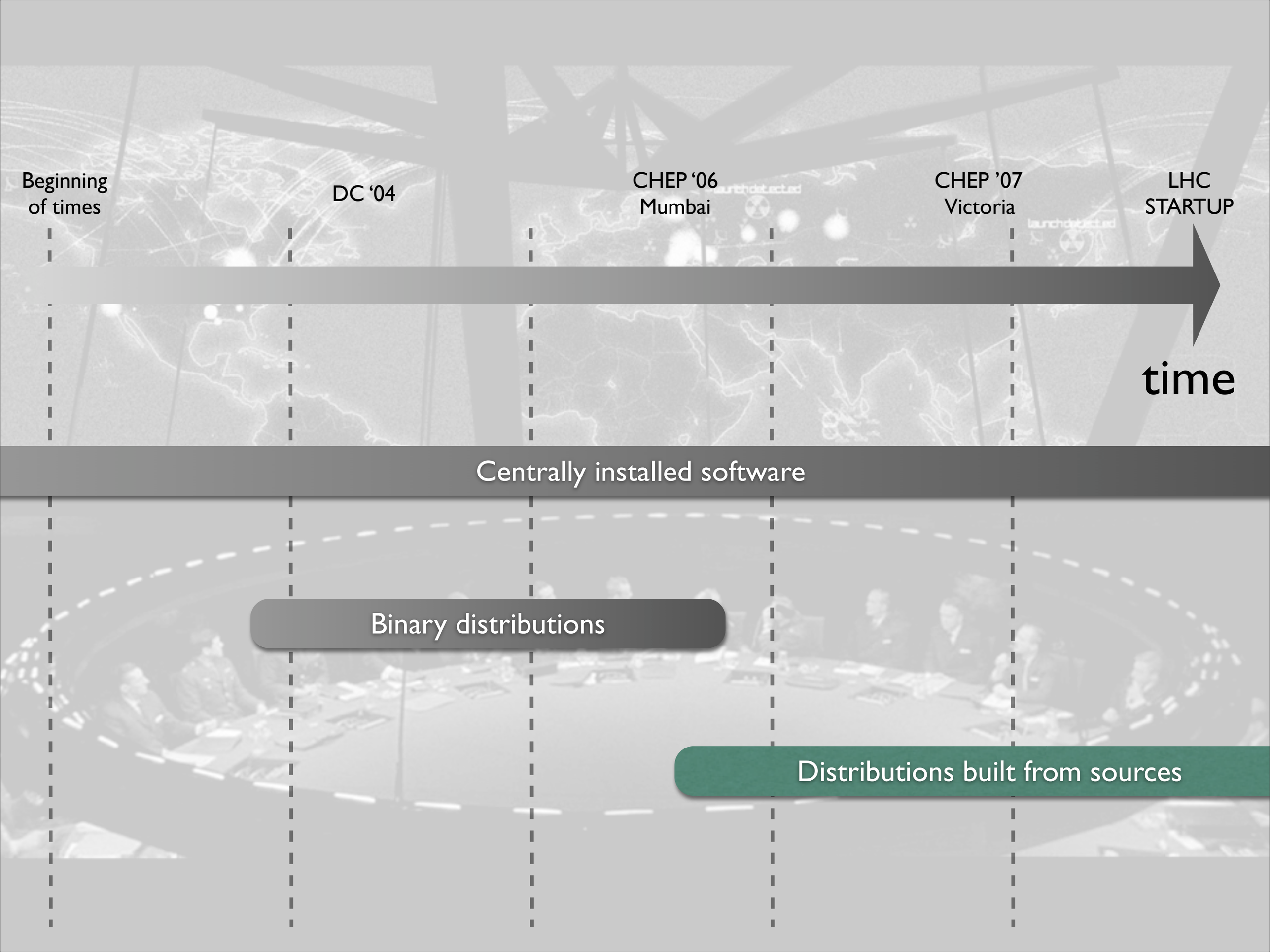
software is available over AFS

software is validated at CERN

AFS area is packaged "as is"

Packages are distributed to other sites

➡ What gets build might be different than what is being packaged.


Beginning of times
DC '04
CHEP '06 Mumbai
CHEP '07 Victoria
LHC STARTUP
time
Centrally installed software
Binary distributions
Distributions built from sources

Distribution built from sources

externals configuration for a release is decided (by CMS!)

Externals are build from source, by a script, from a set of written, CVS maintained, recipes

Resulting build is packaged, test installed and hidden dependencies are spotted immediately

Distribution kit is made available and it is installed and validated at different sites, at the same time.

Packaging is just a by-product of the build procedure

Distribution built from sources

externals configuration for a release is decided (by CMS!)

Externals are build from source, by a script, from a set of written, CVS maintained, recipes

Resulting build is packaged, test installed and hidden dependencies are spotted immediately

Distribution kit is made available to everyone.

Solved issues:

✓ Full control on what goes into a release

Distribution built from sources

externals configuration for a release is decided (by CMS!)

Externals are build from source, by a script, from a set of written, CVS maintained, recipes

Resulting build is packaged, test installed and hidden dependencies are spotted immediately

Distribution kit is made available to everyone.

Solved issues:

✓ Full reproducibility of the release

Distribution built from sources

externals configuration for a release is decided (by CMS!)

Externals are build from source, by a script, from a set of written, CVS maintained, recipes

Resulting build is packaged, test installed and hidden dependencies are spotted immediately

Distribution kit is made available to everyone.

Fast!

✓ No AFS

✓ Under experiment control

Distribution built from sources

externals configuration for a release is decided (by CMS!)

Externals are build from source, by a script, from a set of written, CVS maintained, recipes

Resulting build is packaged, test installed and hidden dependencies are spotted immediately

Distribution kit is made available to everyone.

✓ Everyone is equal: CERN workflow is no different from user installing on his laptop.

How to do it:

- Packaging tool.
- Build instructions for all your externals.
- An helper script to resolve dependency tree of your software, fetch the sources, launch the build.
- Lots of web space to store your packages.
- A software distribution tool.
- Someone who is not scared about tackling the job.

How we did it

- **RPM** as a package manager
- **apt-get** as a distribution manager
- In house glue-script "**cmsBuild**", part of **PKGTOOLS**
  - Configuration creation (with automatic dependencies analysis)
  - Fetching of the sources.
  - Generation of spec files from simplified syntax.
  - Generation of environment scripts.
  - rpmbuild invocation

How we did it (2)

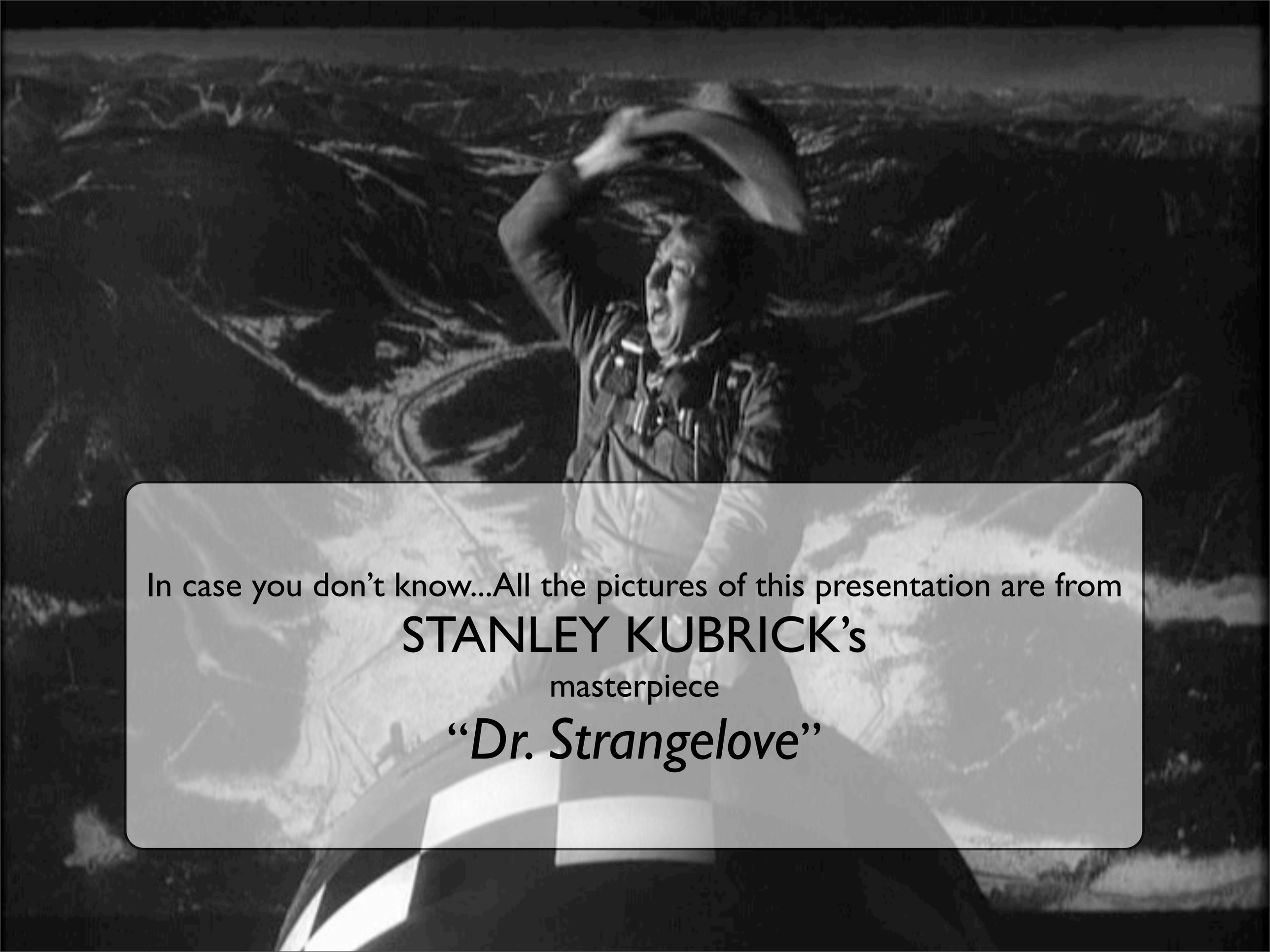
- SPEC files for all the software used by CMS (apart from grid UI stuff).
  - About 100 external packages!!
- In house script to manage apt-get repositories:
  - Standard apt-get repositories
  - Per developer test repositories integrated (by reference) with the official one, yet fully separated.

Achievements

- CMS in the last 2 years went from 1 distribution kit per month to 2 per day.
- Minimal system dependencies for the distribution kit (perl, X11, glibc, bash, kernel)
- All the external software used by CMS is under CMS control. CMS has the knowledge on how to reproduce any give release from scratch in a matter of hours.
- Currently building all the externals from scratch on a quad processor box takes about 4 hours, CMSSW 2 and1/2 hours.
- CMS software is available for installation at the same time at CERN, FNAL, T2 Madagascar, on the user laptop.

Future

- Next release of PKGTOOLS coming soon:
  - Focused to make the release integrator happy
  - will simplify / integrate / fortify "configuration management" phase.
  - parallel build of non dependent packages
  - distcc
  - more platforms
- Early tests (with a quad core Xeon) show that we can seriously aim to build everything from scratch to CMSSW, in less than 2 hours (yes, with **ALL** the externals).



In case you don't know...All the pictures of this presentation are from

STANLEY KUBRICK's

masterpiece

"*Dr. Strangelove*"

Furthermore, it should be noted that none of the characters portrayed in this pdf are meant to represent any real persons living or dead.

Fermorhurte, it shuold be ntoed taht nnoe of the cacterhars payeortrd in tihs pdf are manet to resenepert any rael ponerss linivg or daed.

“Peace on Earth”