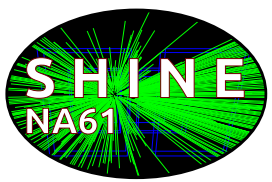
Virtualised data production infrastructure for NA61/SHINE based on CernVM

Dag Toppe Larsen

15. September 2013

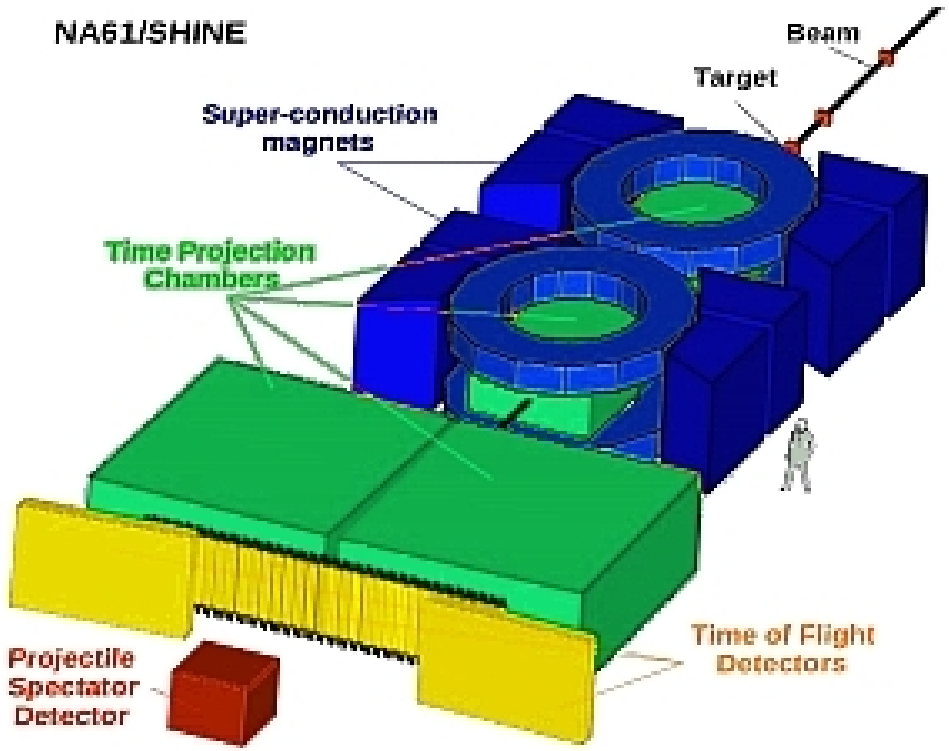


Outline

- NA61/SHINE experiment
  - Data production
- Motivation & "vision"
- Underlying CernVM technologies
- New data production infrastructure
  - Database
  - Scripts
  - Prototype web interface
- Outlook

NA61/SHINE experiment

- Fixed-target experiment at the CERN SPS
- Continuation of NA49 experiment
- Current physics goals:
  - Energy scan for onset of deconfinement and the critical point
  - Measure reference neutrino production
  - Measure properties of atmospheric interactions



NA61/SHINE experiment set-up

NA61/SHINE data production terminology

- Reaction: Unique combination of target type, beam type, beam momentum and year of data taking
- Run: Time-limited sub-set of reaction, O(1h)
- Chunk: Size-limited sub-set of run, O(1GB)
- Raw file: File containing a chunk
- Global key: Unique identifier for set of calibration data
- Production: Processing of a reaction using a unique combination of global key, software versions, *etc.*
- Mass production: Processing of full reaction data set
- Test production: Processing of small sample of data set

NA61/SHINE data production frameworks

- Currently two software frameworks for NA61/SHINE
- "Legacy" framework
  - Inherited from NA49
  - Based on proprietary PGI-Fortran
  - Stand-alone clients executed in sequence
- Shine framework
  - New software
  - Based on ROOT/C++
  - Single executable with libraries
- Legacy is in process of being phased out in favour of Shine
  - For now data production infrastructure must support both

Motivation for new data production infrastructure

- NA61/SHINE data production traditionally initiated manually
  - Configuration files modified manually to reflect desired reaction, software version, calibration versions, production mode, *etc.*, then submitted
  - Various scripts run manually to check integrity of production after batch processing
  - Bookkeeping database updated manually using web interface
  - Time-consuming and prone to human error

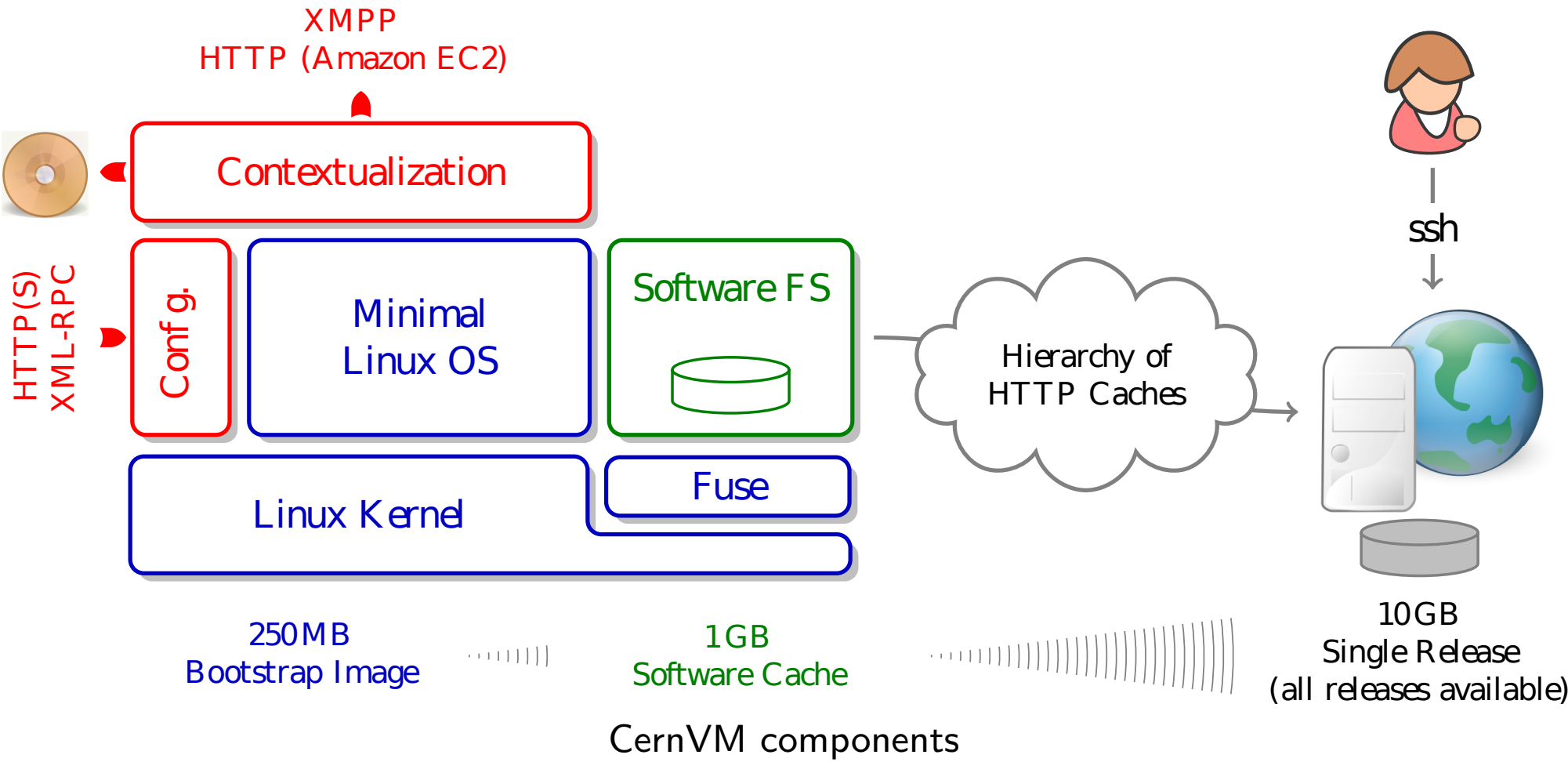
- A more automatic approached desired
  - Take advantage of latest developments in cloud-based distributed computing
  - Allow for data preservation

Data production "vision"

- User logs in to data production web interface
- User chooses reaction of interest
- System shows existing productions for reaction, and offers to start new production

- In case new production is desired
  - User selects desired production parameters (*e.g.* software version, calibration version, mass or test production, *etc.*)
  - User confirms start of production
  - System checks whether production already exists, and that user has sufficient privileges to start production
  - System sets up processing environment and submits jobs
  - System checks status of jobs and automatically resubmits failed jobs
  - System declares production as completed

CernVM eco-system



CernVM components

→ Together these make CernVM a good choice for platform for NA61/SHINE virtualised data production infrastructure

CernVM components

- CernVM virtual machine
  - Linux distribution for VMs
  - Supports most hypervisors
  - Both for desktop and batch
  - Based on Scientific Linux 5
  - About 300MB compressed image size
- CernVM on-line
  - Framework allowing for virtual clusters to be deployed across several physical clouds & sites
  - Can be used as an alternative approach to the grid for distributed computing
- CernVM file system
  - For distribution of experiment software & calibration data
  - Files compressed & hash key generated
  - Distributed globally via HTTP and hierarchy of cache-servers
  - On-demand file download & semi-permanent caching
- CernVM contextualisation
  - Contextualisation used to configure VM for specific experiment/task
  - Automatically for batch systems via EC2 interface
  - Manually through integrated web interface for users' desktops

Data preservation perspective

- Not enough to preserve data, software must also be preserved
  - Compiling and running "old" software
    depend on "old" compilers and system libraries, which
    depend on "old" OS, which
    depend on "old" hardware
  - At some point "old" hardware will not be available any more...
- Porting software to new systems time consuming and can introduce new bugs
- Virtualisation may offer a new approach
  - Allows for preservation of hardware in "software"
  - Virtual machines can emulate "old" hardware, which allows "old" OS and software to run unmodified
  - CernVM has unique build system allowing "old" CernVM disk images to be (re)built with support for modern technology (*e.g.* hypervisors and file transfer protocols)

Virtualised data production infrastructure overview

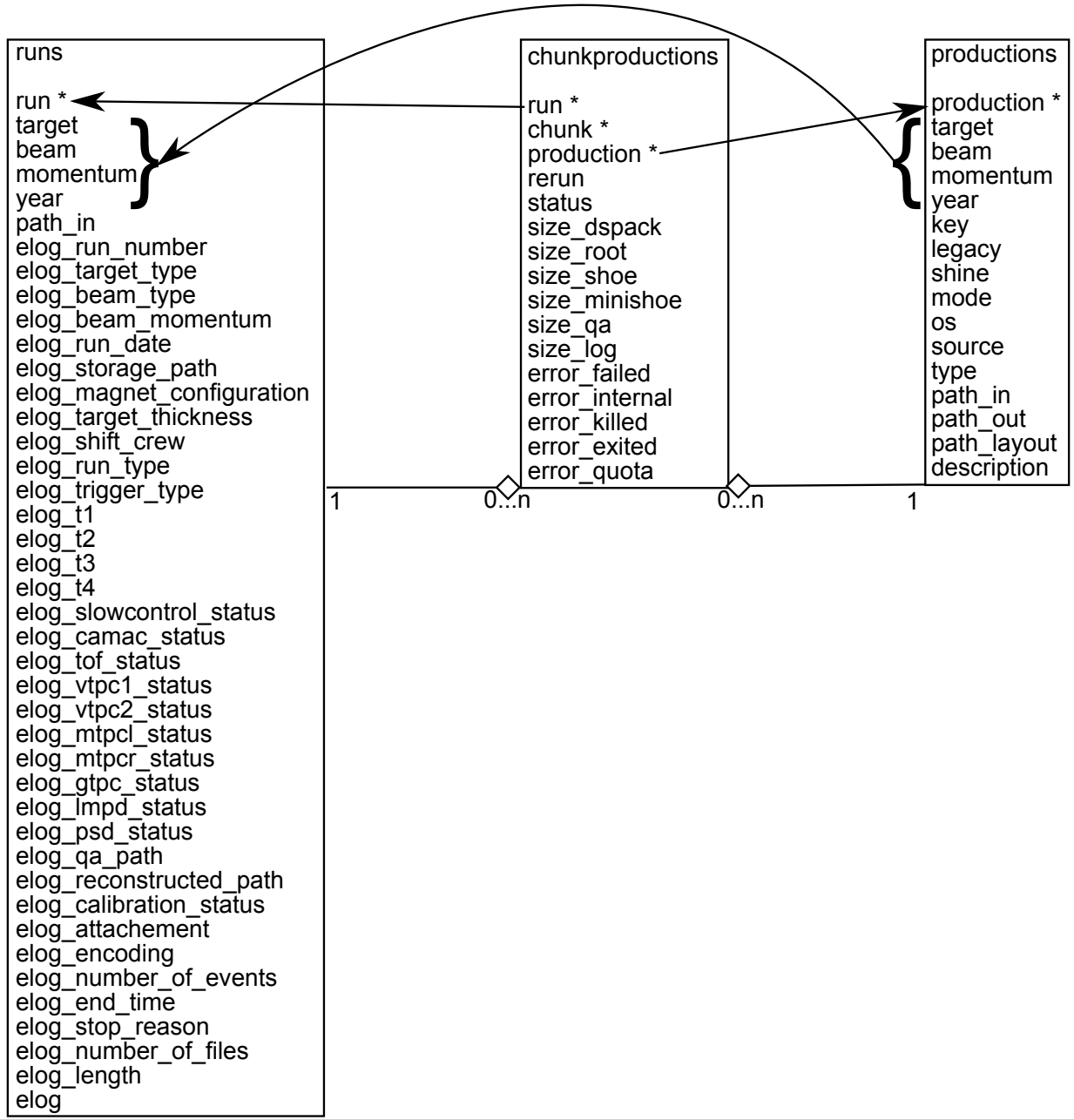
- All NA61/SHINE software and calibration data needed installed on CVMFS
  - Modifications to legacy framework required
  - Shine framework worked out-of-the-box
  - Enables CernVM-based processing on both virtual clusters and personal VMs
- Allows for virtualisation-based data preservation
- Processing chain based on Condor (batch system used by CernVM)
- SQLite-based production database to store all production information

Virtualised data production infrastructure overview

- xRootd used for data transfer
  - Allows/simplifies processing outside CERN
- Currently running on CERN OpenStack infrastructure
  - Integration with CernVM on-line will allow non-CERN based processing
- Scripts to automatically submit/check data productions
  - Web front-end to scripts under development
- Large-scale production underway to verify consistent results between CernVM and Lxbatch productions
  - Preliminary result validation is encouraging

Production database schema

- runs
  - One entry per run
  - $O(10^4)$ entries
  - Information for run, including from eLog
- productions
  - One entry per unique production
  - $O(10^2)$ entries
  - Information for production, *i.e.* unique set of parameters
- chunkproductions
  - One entry per produced chunk
  - $O(10^6)$ entries
  - Error checking



Production scripts

- Responsible for interactions between different components
- Typically communicate both with production database and some other component
- Can be executed both from command line by user or used via web interface front-end
- One master command “produce” that user/web interface interacts with

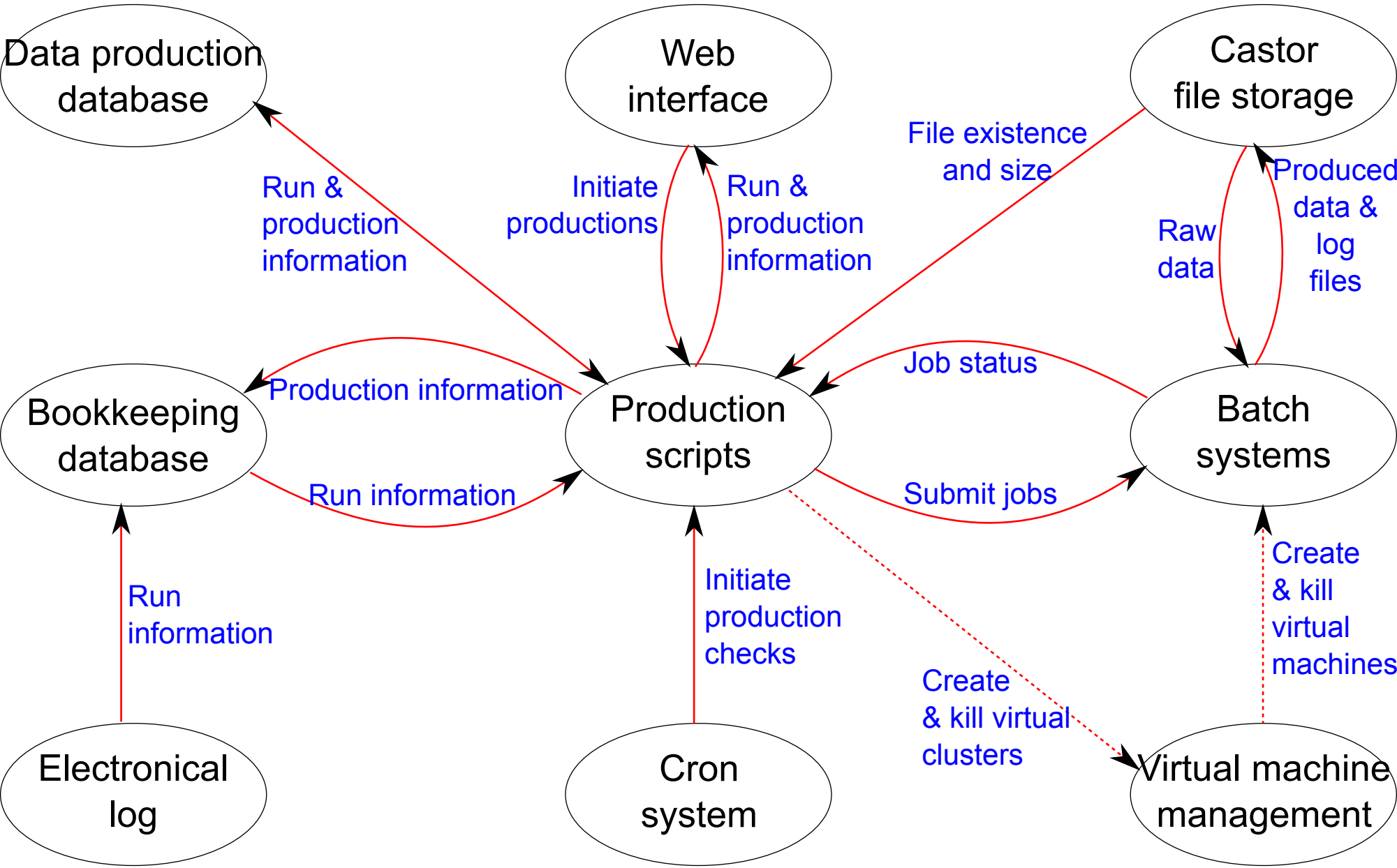
Master production script

```
-bash-4.1$ ./produce
Usage:
./produce <command>
<command> one of:
reactions     - list all reactions in database
productions   - list all productions in database
./produce <command> <path_in>
<command> one of:
regreaction   - register all reactions found at path_in in database
./produce <command> <target> <beam> <momentum> <year> [<key> <legacy> <shine> <mode> <source> <path_in>
 <description>]
<command> one of:
regproduction - register new production in database
produce       - start new production
check         - check production for errors and update database
summary       - production summary
reproduce     - reprocess chunks with with errors for production
okchunks      - list all OK chunks for production
```

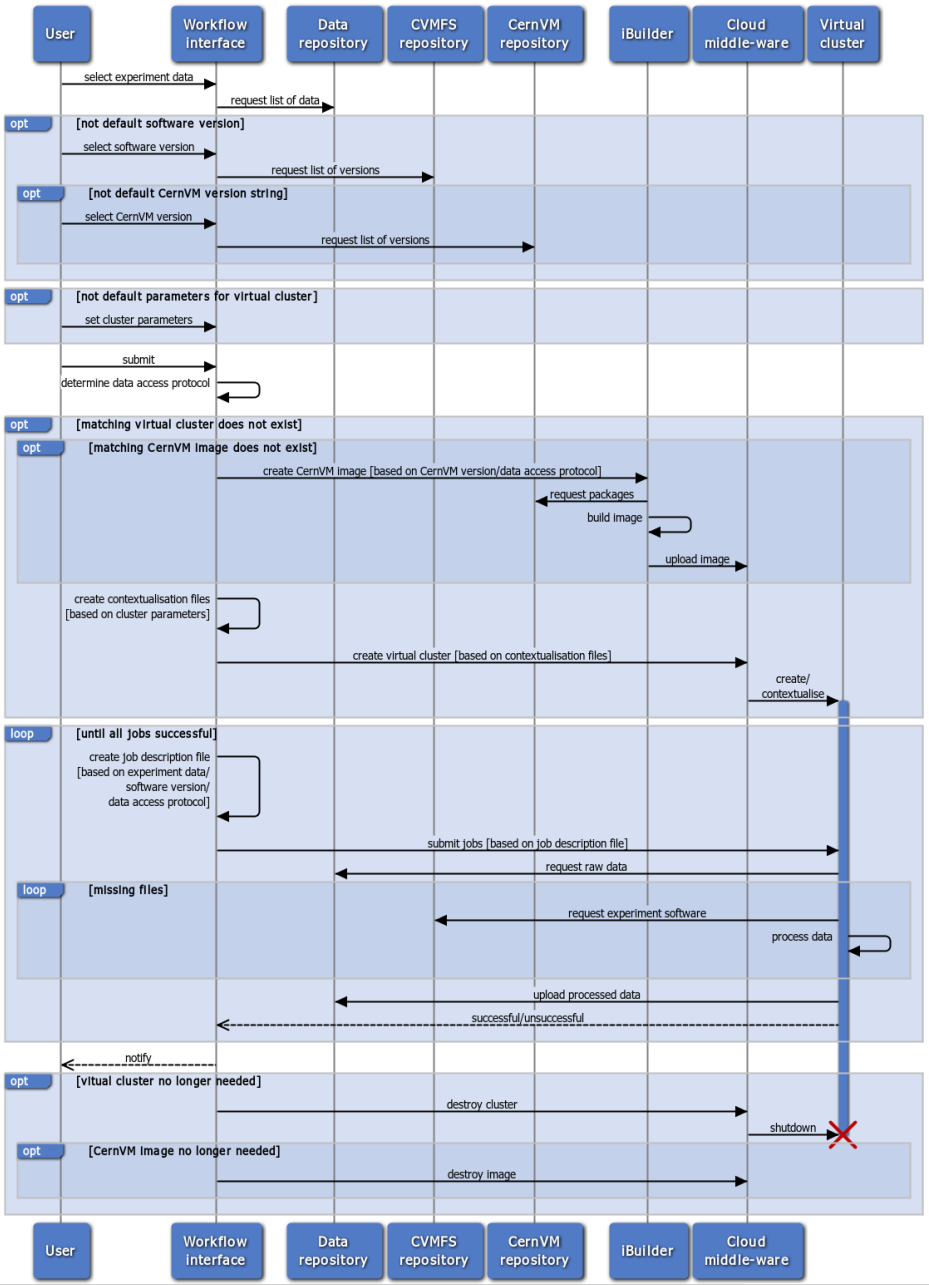
Examples:

```
produce reactions
produce productions
produce regreaction /castor/cern.ch/na61/11/Be/Be160
produce regproduction Be Be 158 11 040 13e v0r5p0 pp phys def "A new production"
produce produce Be Be 158 11 040 13e v0r5p0 pp phys
produce check Be Be 158 11 040 13e v0r5p0 pp phys
produce summary Be Be 158 11 040 13e v0r5p0 pp phys
produce reproduce Be Be 158 11 040 13e v0r5p0 pp phys
produce okchunks Be Be 158 11 040 13e v0r5p0 pp phys
```

Overall component interaction overview



Job submission component interaction sequence



- System queries databases to obtain list of reactions and production parameters
- User selects reaction and production parameters and submits reaction
- System creates job files and distribute to cluster
- Worker nodes obtain raw files, process and return produced data
- System checks for errors, resubmits failed jobs

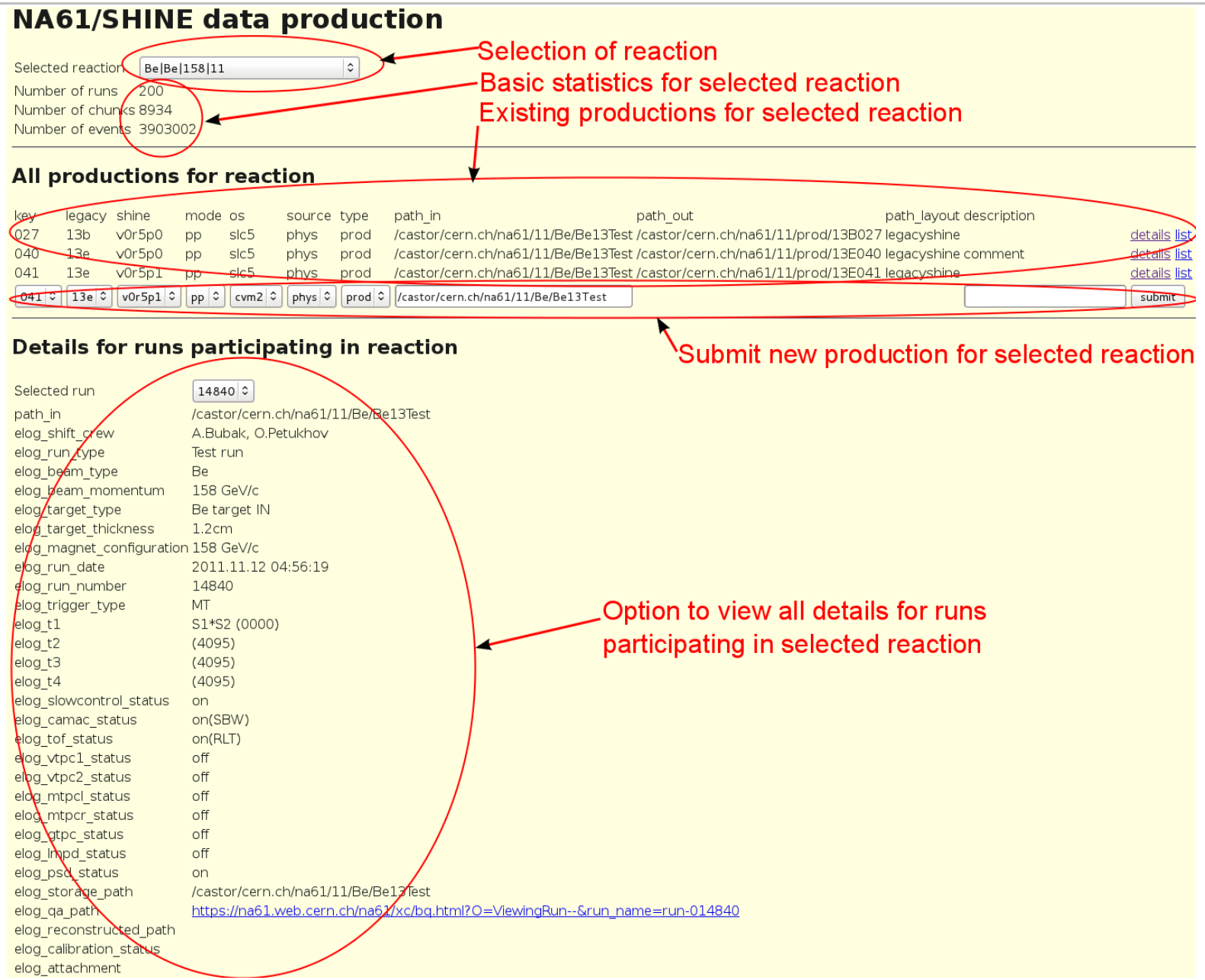
File system enforcement of uniqueness of production

- Production is defined as unique combination of reaction (target, beam, momentum & year) and reconstruction conditions (global key, legacy version, shine version, mode, OS, source & type)
  - legacy field may be empty as one moves from legacy to Shine
- There should be only one set of produced data corresponding to production
  - Trying to enforce this on file system level by encoding unique parameters in data path:
    <type>/
    <target>_<beam>_<momentum>_<year>/
    <key>_<legacy>_<shine>_<mode>_<os>_<source>/
    <output-file-type>/
  - Example:
    prod/Be_Be_158_11/040_13e_v0r6p0_pp_cvm2_phys/shoe.root/
- Also believed to be an intuitive representation of data

Web interface prototype

- Web interface for production framework under development
  - Front-end for production scripts
  - Based on XHTML/CSS
- Focus on usability under different use cases
  - "Reaction" and "production" believed to be the main entities
  - User selects reaction from list; system displays all productions for reaction
  - User can either choose to view further details for reactions and participating runs, or start new production
- Currently information can only be viewed, actual production initiation under implementation
- Information for past productions will be imported

Web interface prototype



Outlook

- Mass production of full reaction ongoing on CernVM cluster
  - Will be validated against data produced on traditional CERN Lxbatch
  - Will gradually increase utilisation of virtual cluster for standard data production once validity is confirmed
- In parallel, improvements to web interface, bug fixes, *etc.*
- System expected to be production ready this year
  - Likely to coincide with switch to new Shine data production framework
- Switch to μCernVM 3?
- Integrate CernVM on-line and distribute computing to non-CERN sites