Quattor: An administration toolkit for optimizing resources

Marco Emilio Poleggi - CERN/INFN-CNAF

German Cancio - CERN

{Marco.Poleggi,German.Cancio}@cern.ch

Outline

quattor

- Design concepts
- Configuration management
- Configuration deployment
- Target node administration

Design concepts: goal

quattor

Taking care of the *configuration, installation* and *management* of target fabric nodes

- A **Configuration Database** holds the “desired state” of all fabric elements, arranged in template hierarchies
  - Cluster information: name and type, operating system, architecture, etc.
  - Node setup: hardware (HD, RAM, network...), software packages, system services, etc.
- On target nodes, autonomous management agents are responsible of
  - Base installation
  - Software installation and management
  - Service (re-)configuration

Design concepts: features

quattor

- Centralized control of configuration data:
  - Unique configuration storage
  - Automatic notification of changes to the target nodes
- Autonomous target nodes:
  - Pull-based re-configuration (no remote scripts, no network file systems)
  - Local configuration files
- Reproducibility:
  - Idempotent, atomic operations
- Scalability:
  - Through load balancing, proxy caching, scalable protocols: *O(10k)* nodes!
- Based on well-known standards:
  - HTTPS, XML, RPM/PKG, SysV init scripts, etc.
- Portability:
  - Linux and Solaris currently supported

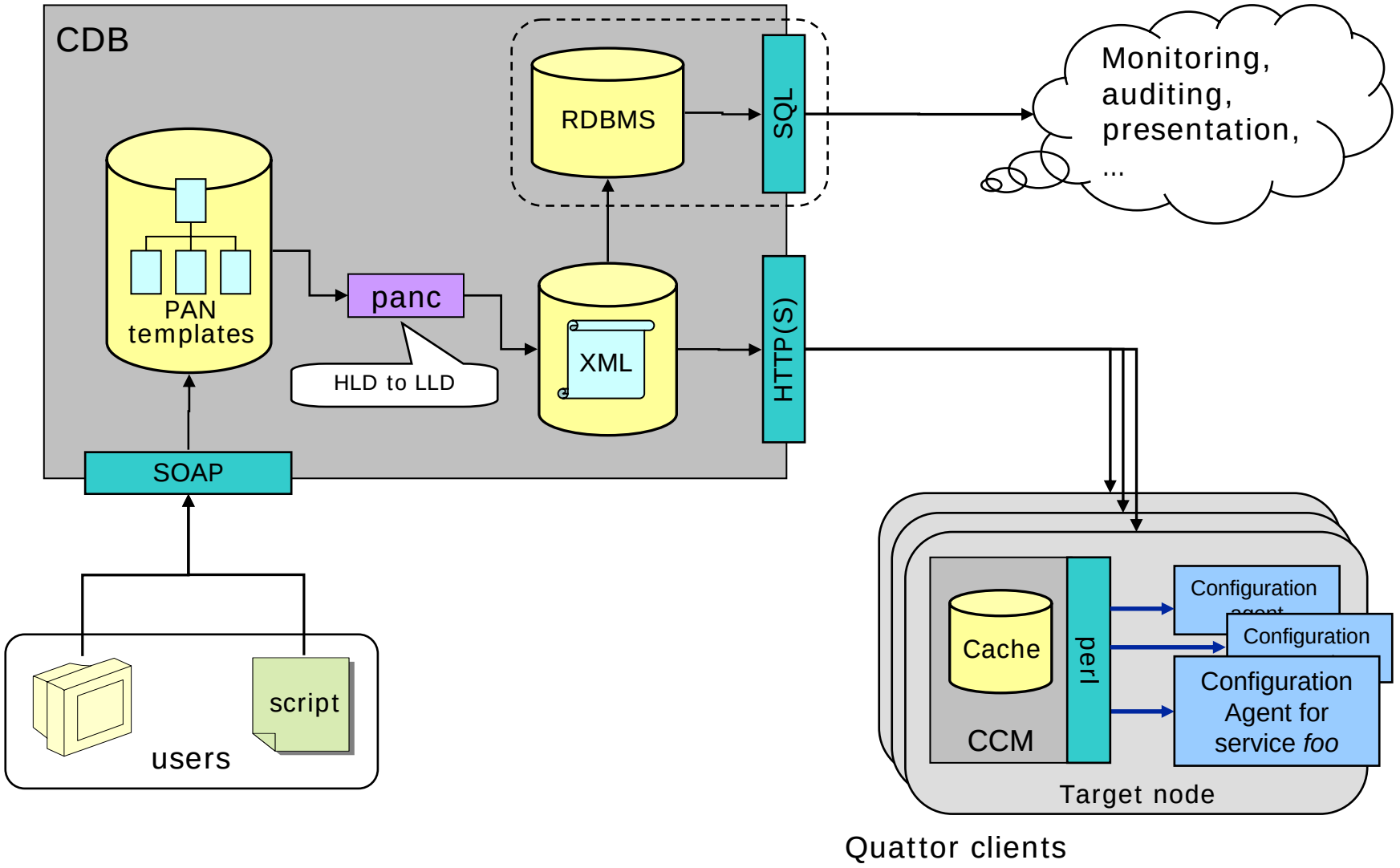
Outline

quattor

- Design concepts
- Configuration management
  - *Configuration Database* (CDB)
- Configuration deployment
- Target node administration

Configuration Management Infrastructure

quattor



Configuration Database (CDB)…

quattor

- Keeps complete configuration information in a unique repository
  - Uses *namespaces* for a hierarchical management – NEW!
  - Uses *Access Control Lists* (ACLs) to restrict user's scope – NEW!
  - *Remote* interaction but *centralized* control
    - All changes are tested on the server before commit
- Data consistency is enforced by a transactional mechanism
  - Concurrent operations are isolated in user *sessions*
    - Allows disconnected operations
  - All changes are done in transactions along a user session
  - Conflicts of concurrent modification of the same template are detected
- Configuration is validated and kept under version control
  - Built-in validation, e.g. types
  - User-defined validation, e.g. range of high-level parameters

...Configuration Database (CDB)

quattor

- Going back to previous versions of the configuration is possible
  - Full history is kept in CVS
- Support for different user authentication mechanisms: *X.509* – NEW!, *Kerberos5* – NEW!, encrypted passwords. Easily extensible!
- Optional SQL module for feeding data to Oracle and/or MySQL

Subversion-based alternative (*SCDB*)

- Developed at *LAL*
- Decentralized control on user workstations
  - Local data check-out
  - Testing and validation
- Central *Subversion* (instead of CVS) repository

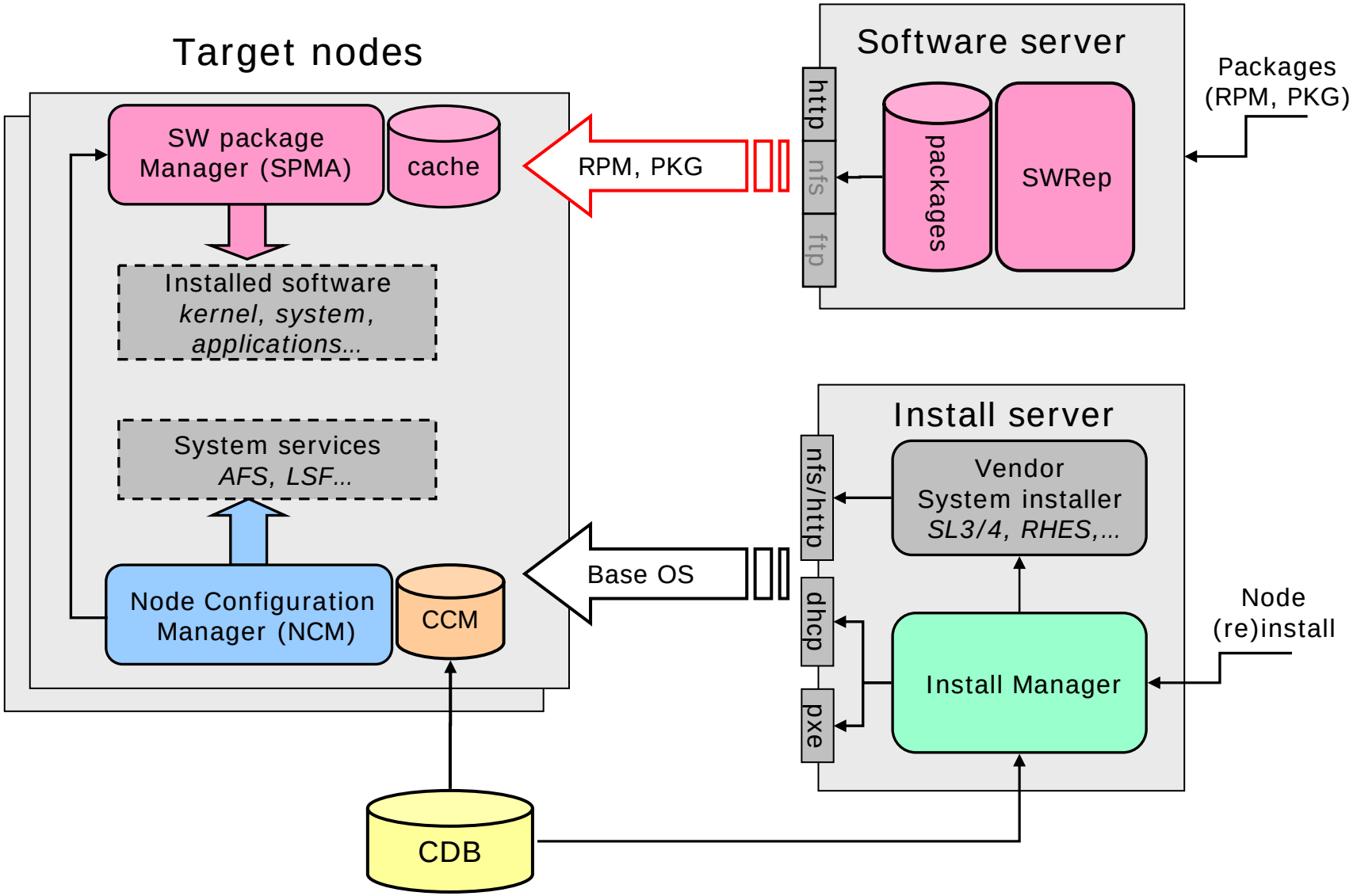
Outline

quattor

- Design concepts
- Configuration management
- Configuration deployment
  - *Automated Installation Infrastructure* (AII)
  - *Configuration Cache Manager* (CCM)
  - *Node Configuration Manager* (NCM)
- Target node administration

Managing target nodes (clients)

quattor



Automated Installation Infrastructure (AII)

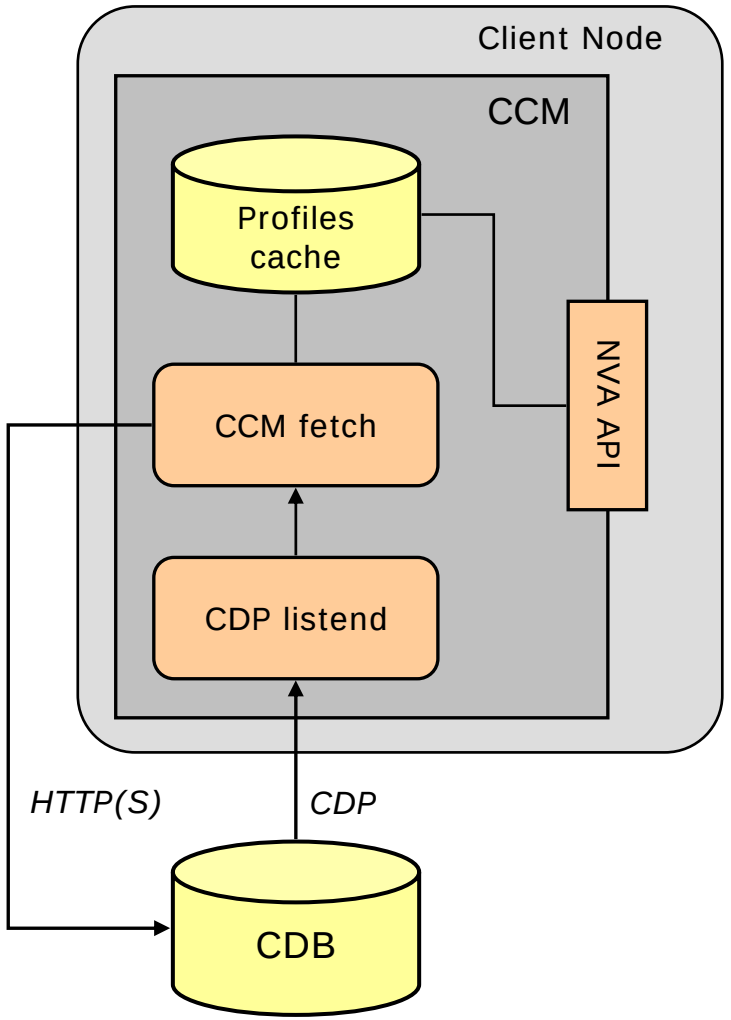
quattor

- Sits on top of the standard vendor installer, and configures it
  - Which OS version to install
  - Network and partition information
  - Core packages
  - Custom post-installation instructions
- Two-phase process
  1. Base OS installation
  2. Quattor client SW installation and reconfiguration
- Automated generation of control file (*KickStart*)
- It also takes care of managing *DHCP* (and *TFTP/PXE*) entries
- Configuration information is taken from CDB or via command line
- Available for RedHat-based Linuxes, through the *Anaconda* installer

Configuration Cache Manager (CCM)

quattor

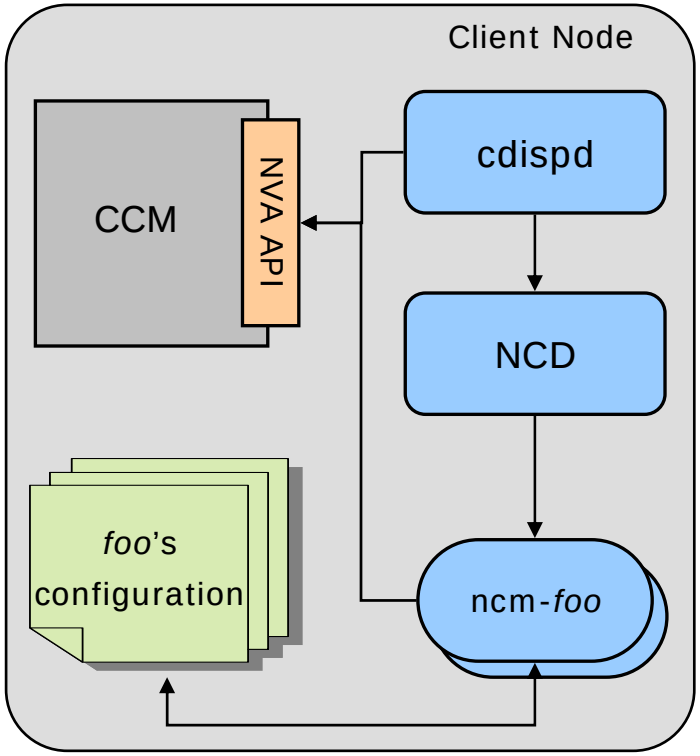
- Runs on every managed node
- Provides a local interface to the node's configuration information (*NVA API*)
- Information is downloaded from CDB and *cached*:
  - Faster access to the configuration
  - Avoids peaks on CDB servers
  - Supports disconnected operations
  - Synchronization with CDB through notification/polling: *Configuration Distribution Protocol* (CDP)
- Client authentication trough *X.509* – NEW!



Node Configuration Manager (NCM)...

quattor

- *NCM* is responsible for ensuring that reality on a node reflects the *desired* state in CDB
- Service-specific plug-ins, called *ncm-components,* make the necessary changes:
  - (Re-)generate local configuration files, e.g. `/etc/sshd/sshd_config`
  - Restart/reload daemons via *SysV*-style scripts
  - Resolve configuration dependencies, e.g. configure *network* before *sendmail*
- Triggering of ncm-components
  - invoked on boot
  - via `cron`
  - upon changes in CDB



... the NCM framework

quattor

- Is extensible
  - Several ncm-components already available for system and Grid/LCG services
    - QWG's suite
    - *ncm-yaim*
  - minimal *Perl* skills required for writing new components ;-)
- Has Library support for easy development
  - configuration data access
  - file manipulation
  - process management
  - exception handling
- Features commands for querying the node's configuration

Outline

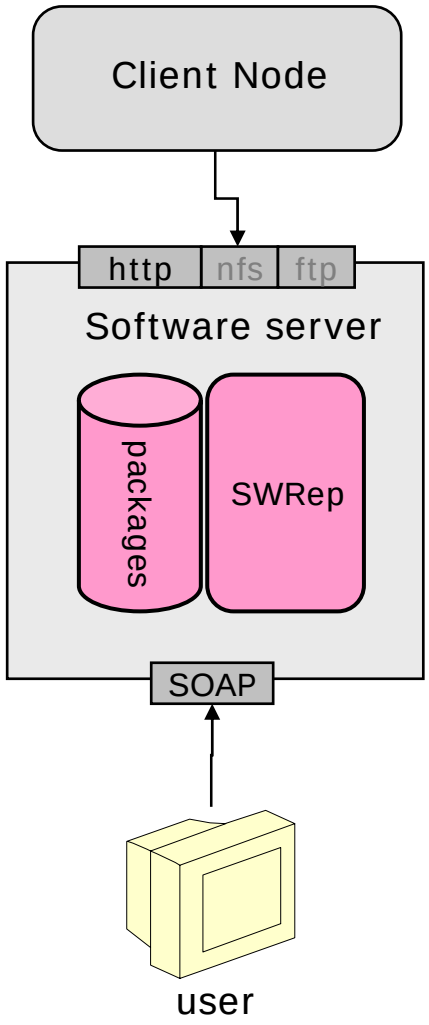
quattor

- Design concepts
- Configuration management
- Configuration deployment
- Target node administration
  - *SoftWare Repository* (SWRep)
  - *Software Package Management* (SPM)

Software Repository (SWRep)…

quattor

- Hierarchical storage structure
  - Platforms: e.g., *i386, x86_64*
  - Areas: e.g., `/edg/quattor`
- Extensible to accommodate different packagers: RedHat's *RPM*, Solaris' *PKG*, Debian's *pkg*, etc.
  - Multiple package versions support
- User management via ftp-like commands
  - SOAP-based interface – NEW! - with the same plug-in-based authentication as for CDB (*X.509*, etc.- NEW!)
  - ACL-based mechanism to grant/deny modification rights for package "areas"



...Software Repository (SWRep)

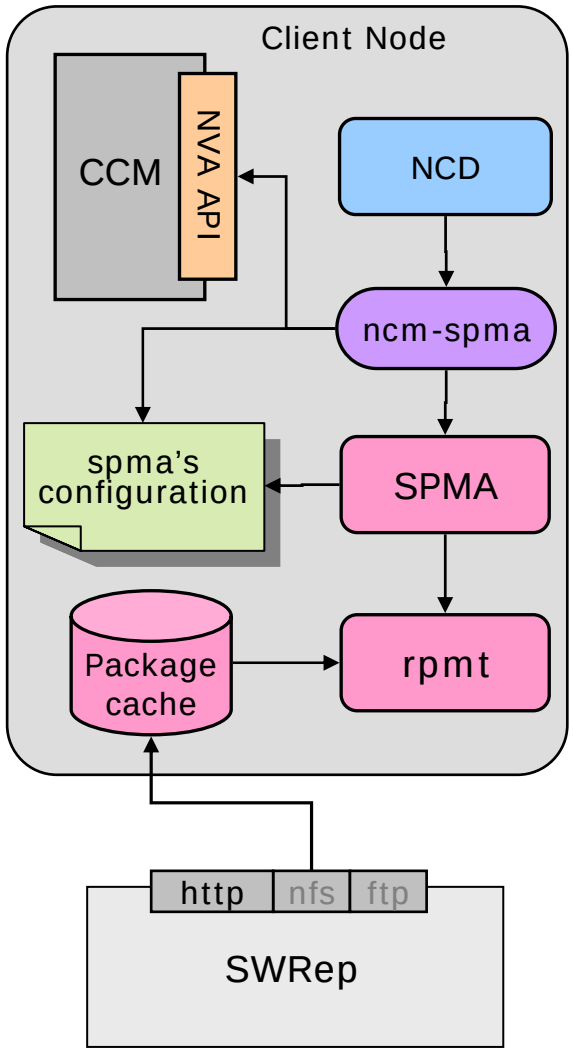
quattor

- Client access via standard protocols
  - HTTP, AFS/NFS, FTP
- Based upon off-the-shelf software
  - *Apache* Web server as file repository
  - *rsync* for mirroring/redundancy
- Scalability:
  - Up to *~800* nodes with single-server set-up
  - *O(10k)* nodes with proxy-caches + load balancing (see CERN-CC reverse proxy network)
    - *Squid* support through an NCM component – NEW!

Software Package Management (SPM)…

quattor

- The *SPM* subsystem manages *all* or a *subset* of packages on the nodes
  - *Full-control* mode: wipe out unknown packages, (re-)install missing ones. Typical mode for production nodes
  - *Non-intrusive* mode: configurable management to allow user-installed packages with priority control. Typical mode for development/desktop nodes
- *SPMA* (SPM *Agent*) is a package *manager* (does a lot more than upgrading!)
  - Multiple versions of the same package can be installed
  - Upgrade/downgrade control
  - Transactional behavior through *rpmt* – NEW Python implementation



...Software Package Management (SPM)

quattor

- Portable, thanks to an extensible plug-in-based framework
  - Plug-ins currently available for Linux *RPM* and Solaris *PKG*
- Scalability is assured by
  - Standard protocols
  - Time smearing
  - Package pre-caching
  - Forward/reverse proxy-cache support
- Support for multiple repositories
- Configurable remotely via CDB, or locally

http://quattor.org