



# Software Management with Quattor

German Cancio

CERN/IT





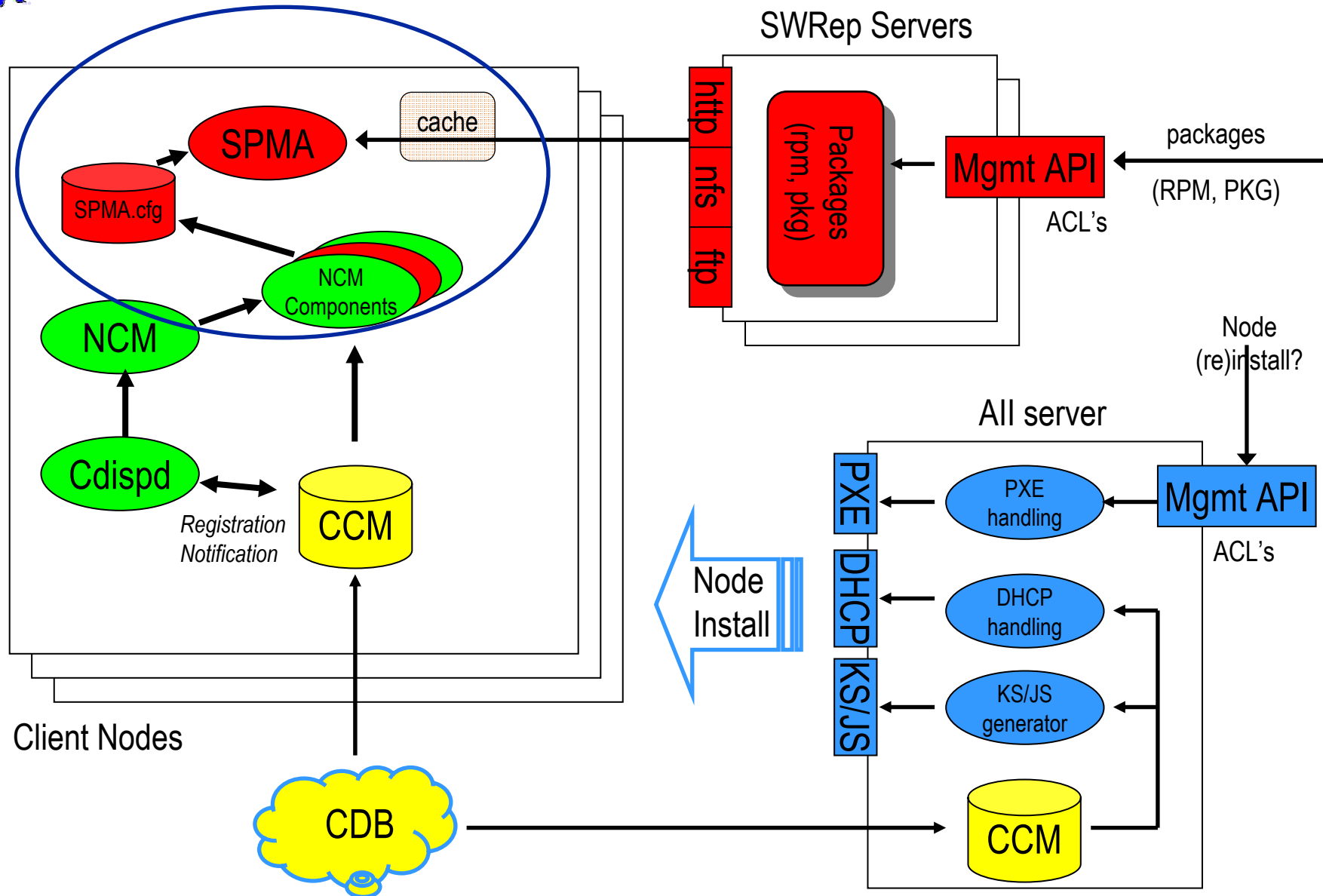
# Outline



- ◆ Context
- ◆ SWRep and SPMA
- ◆ SWRep/SPMA vs. APT/yum
- ◆ Two easy SPMA exercises



# SPMA environment





# SWRep (Software Repository)



- ◆ Universal repository for storing Software:
  - Extendable to multiple platforms and packagers (RH Linux RPM.. Solaris PKG, others like Debian pkg)
  - Stores multiple package versions/releases per platform
- ◆ Management (“product maintainers”) interface:
  - ACL based mechanism to grant/deny modification rights (packages associated to “areas”)
- ◆ Client access: via standard protocols
  - HTTP, AFS/NFS, FTP
- ◆ Replication for load balancing/redundancy: using standard tools
  - Apache mod\_proxy (as used at CERN, [slides here](#))
  - rsync



## SPMA (Software Package Manager Agent)



- ◆ Can manage either *all* or a *subset* of packages on the nodes
  - On production nodes: *full control* - wipe out unknown packages, (re)install missing ones.
  - On development nodes: non-intrusive, configurable management of system and security updates.
- ◆ Package *manager*, not only *upgrader*
  - Can roll back package versions
  - Transactional verification of operations
- ◆ Scalability:
  - Supports HTTP (also FTP, AFS/NFS) and forward/reverse proxies
  - Time smearing and package pre-caching
- ◆ Portability via generic plug-in framework
  - System packager specific transactional interface (RPMT, PKGT)
- ◆ Multiple repositories can be accessed (eg. division/experiment specific)



## SPMA (II)



- ◆ SPMA functionality:
  1. Compares the packages currently installed on the local node with the packages listed in the configuration profile
  2. Computes the necessary install/deinstall/upgrade operations
  3. Invokes the packager (rpmt/pkgmt) with the right operation transaction set
  
- ◆ The SPM is driven via a local configuration file
  - An NCM *component* (ncm-spma) generates/updates this configuration file out of CDB information
  - `/var/lib/spma-target.cf`



## SPMA (III)



### RPMT

- ◆ RPMT (RPM transactions) is a small tool on top of the RPM libraries, which allows for multiple simultaneous package operations resolving dependencies (unlike plain RPM)
  - Example: 'upgrade X, deinstall Y, downgrade Z, install T' and verify/resolve appropriate dependencies
- ◆ Does use basic RPM library calls, no added intelligence
- ◆ Ports available for RH7 (RPM 4.0) and SL3 (RPM 4.2)
  - Note: due to a bug in RPM libraries, RPMT requires RPMlib 4.2.1 (see [Savannah #5005](#)). A workaround will be provided soon.
  - CERN Scientific Linux ships RPM 4.2.1 instead of 4.2.2 for this reason.



# SPMA/SWRep vs. other package management tools



- ◆ Quattor does not impose using any packaging tool.
  - **SPMA**
  - **APT** (and **yum**, not described here as very similar to APT)
- ◆ SPMA advantages:
  - **Declarative**: Keep list of packages in CDB templates. Packages are updated/downgraded as required.
    - SPMA behaves stateless.
  - Every node may have a completely **different setup** in terms of installed packages and versions.
    - Not "go for the latest"
  - Explicit **separation of Package depot** (SWRep) and **configuration** (in CDB)
    - You can have multiple versions (old, production, new, beta) of packages in the repository without causing client updates
  - SPMA supports **rollbacks**. Just change the packages/versions in CDB and SPMA will take care to downgrade/upgrade/install/remove whatever is required.
  - SPMA supports multiple **simultaneous package versions** on one node.





# SPMA/SWRep vs. other package management tools



## ◆ APT advantages:

- **Standard tool** shipped with Scientific Linux
    - LCG and Quattor offer APT/yum repositories for downloading software
  - **Dependency resolution**
    - But may decide to resolve differently than what you want! Eg. More than one package 'providing' a dependency.
    - Requires to set 'priorities' on repositories.
  - Nice **GUI** (synaptic), easier to use.
- ◆ APT and SPMA/SWRep should not be used in parallel on the same node.
- However, it is possible to bootstrap the Quattor server with APT, and then use SPMA to manage your farms and servers
- ◆ The choice between SPMA and APT/yum will depend on the complexity of your environment and/or the control level you need.
- CERN example: SPMA for Computer Centre ("full" mode for batch nodes, "light" mode for development nodes)
  - APT for desktops



# SPMA Exercises



## SPMA exercises



- ◆ Quick introduction to the PAN package templates
- ◆ Adding/removing packages to the node
- ◆ Local packages, and user priority switch
- ◆ Handling simultaneous package versions with the SPMA

### References:

- [Yesterday's slides \(for CDB and SWRep operations\)](#)
- `man spma`
- `spma --help`
- `vi /etc/spma.conf`
- [A read-only access to a SWRep server on 'lxb0613.cern.ch' is configured, as user 'tutorial' \(`su -l tutorial`\)](#)
- `man swrep-client`



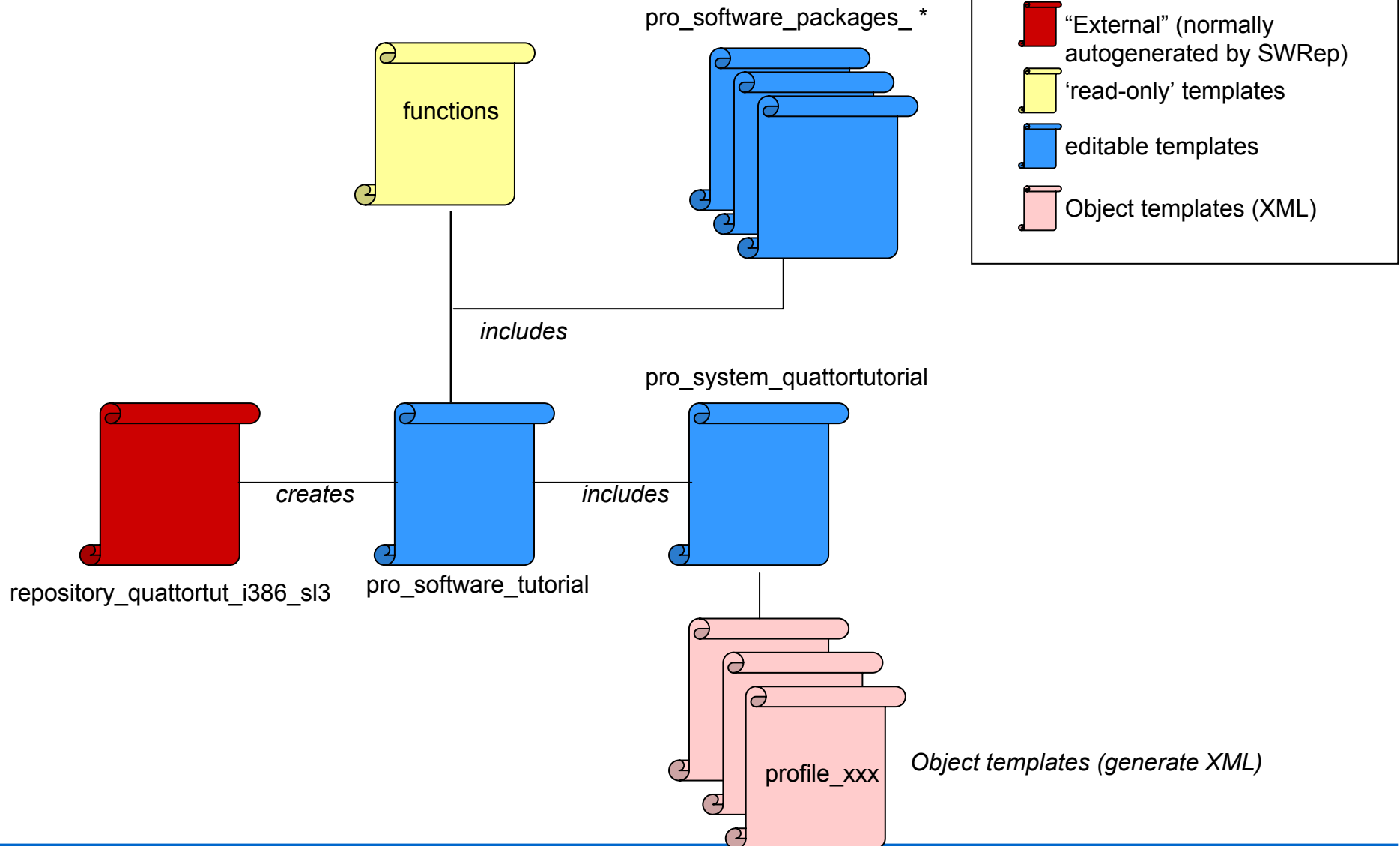
## SPMA setup: package templates (I)



- ◆ The Pan templates for the SPMA contain the list of packages to be installed by the SPMA.
  - **check** `pro_software_quattortutorial.tpl` **which includes:**
    - `pro_software_packages_scientificlinuxcern_3_03`
    - `pro_software_packages_quattor_sl`
    - `pro_software_packages_quattortutorial`
- ◆ By overlaying the templates using 'include', we get the desired list of packages in the configuration profile.



# Template structure for software packages





## SPMA setup: package templates (II)



Some PAN functions are provided for **manipulating the package list in the profile**, which are used in the templates:

- ◆ `pkg_add ("packagename", "version-release", "arch");`
  - Adds a package to the profile.
- ◆ `pkg_del ("packagename", ["version-release"]);`
  - Removes a package version (or all if none specified) from the profile .
- ◆ `pkg_repl ("packagename", "new ver-rel", "arch" [, "old ver-rel"]);`
  - Replaces the package version 'old' by 'new' in the profile. If 'old' is not specified, it replaces all other versions.
- ◆ It is important to understand that the 'add', 'del' and 'repl' functions do only modify the final **list** of desired packages.
  - Eg. 'pkg\_del' does **not** instruct to delete any package, but removes it from the list of desired packages
  - Useful when modifying inherited profiles
- ◆ For the exercises, we will use only the 'pkg\_add' function.
- ◆ More information about these functions: see `pro_declarations_functions_general.tpl`



## Running SPMA



- ◆ The SPMA configuration is generated/updated by running an NCM component
  - `ncm-ncd --configure spma` : *updates* `/var/lib/spma-target.cf` *and* `/etc/spma.conf` *if needed*
  - The NCM component updates the SPMA config files, and can automatically run the SPMA if there were changes (this is deactivated for today's tutorial).
- ◆ SPMA is run as follows:
  - `# spma [options]`
- ◆ Most important SPMA options:
  - `--noaction`
  - `--verbose`
  - `--userpkgs (yes/no)` *run in 'light' or 'full' mode*
  - `--userprio (yes/no)` *upgrade/downgrade packages installed by user*
- ◆ The **typical sequence** for this tutorial will be:
  1. *update templates in CDB via* `cdbop`
  2. *run* `ncm-ncd --configure spma && spma --noaction` (*check what changes*)
  3. *run* `spma`



## SPMA exercise 1



### Exercise 1: Adding/removing packages to the node

i. Start cdbop, "get" the  
`\pro_software_packages_scientificlinuxcern_3_03'`  
template, and add to it the following RPM package:

- **Name:** `uucp` **version-release:** `1.06.1-47` **arch:** `i386`
- **Tip:** follow the syntax of the other packages

ii. "update" CDB via cdbop, and run "commit"

iii. Now, update the SPMA configuration file:

```
ncm-ncd --configure spma
```

iv. And now run the SPMA:

```
spma --noaction followed by spma
```





## SPMA exercise 1. (contd)



- v. What happens if you re-run the SPMA again?
- vi. What happens if you remove the package by hand (as root) (`rpm -e uucp`), and run the SPMA afterwards?
- vii. Now comment out the line added to the template in i.) and add another line for adding the following package: 'squid'.

Use the 'swrep-client' tool for finding out which are the version(s) of squid on the SWRep. You need to run this as user 'tutorial', not 'root'.

Then, re-run cdbop, ncm-ncd, and spma.

### Notes:

- ◆ The generated SPMA local configuration file (containing the 'target' package list) is kept in `/var/lib/spma-target.cf`
- ◆ SPMA log file: `/var/log/spma.log`



## SPMA exercise 2.



### Exercise 2: Local packages

i. change the configuration of SPMA (in `/etc/spma.conf`) to allow packages directly installed by the user: `set 'userpkgs = yes'`

Or, better, automatically configure this entry using the 'SPMA' NCM component configuration in CDB:

```
"/software/components/spma/userpkgs" = "yes" ;  
(set this in profile_lxb<xxxx>.tpl)
```

followed by an `ncm-ncd --configure spma`

ii. Install (by hand) the following package: `xpdf-2.02-9.3.i386.rpm`

```
rpm -ivh http://lxb0613.cern.ch/swrep/i386_sl3/xpdf-2.02-9.3.i386.rpm
```

or via apt-get: `apt-get install xpdf`

iii. Run SPMA.

iv. Change the configuration of SPMA back to `'userpkgs = no'` (default).

v. Run SPMA again.



## Other SPMA exercises



To test out at your home institute:

- ◆ Check `pro_declarations_functions_general.tpl`:
  - Using the `pkg_repl` and `pkg_del` functions (check)

Check `'man spma'`:

- Using the SPMA pre-caching facility
- Using the 'userprio' switch
- Using proxy servers.



 **quattor**

**<http://quattor.org>**