

Quattor: Configuration Management at (Large)

Sites

Michel Jouvin

LAL/Orsay

jouvin@lal.in2p3.fr

fOSSa'09, Grenoble

<http://ocs.inrialpes.fr/2009>

November 2009, 17



Outline

- Why Quattor?
- Quattor history
- Quattor goals, architecture and workflow
- Quattor main features
- Configuration examples
- GRIF use case: challenges and achievements
- Quattor project model and licensing
- Quattor community
- Summary

Why Quattor?

- Developed by European Data Grid (2001-04)
 - Efficient management of large fabrics with different types of nodes
 - Independent of grid middleware
- Management tool for reproducible configurations
 - **Installation AND reconfiguration** based on a common configuration description
 - Existing tools doing one or the other : Kickstart, APT, Puppet...
 - Easy configuration cloning and modification rollback
 - Replaying a configuration must produce the same result
 - Configuration validation before deployment
- Service oriented
 - Easy mix of services on a machine
 - As long as the mix is supported (port conflict...)
 - No side effect of one service configuration on another service
 - Configuration consistency of common low level service
 - Accounts, cron, NFS mounts...

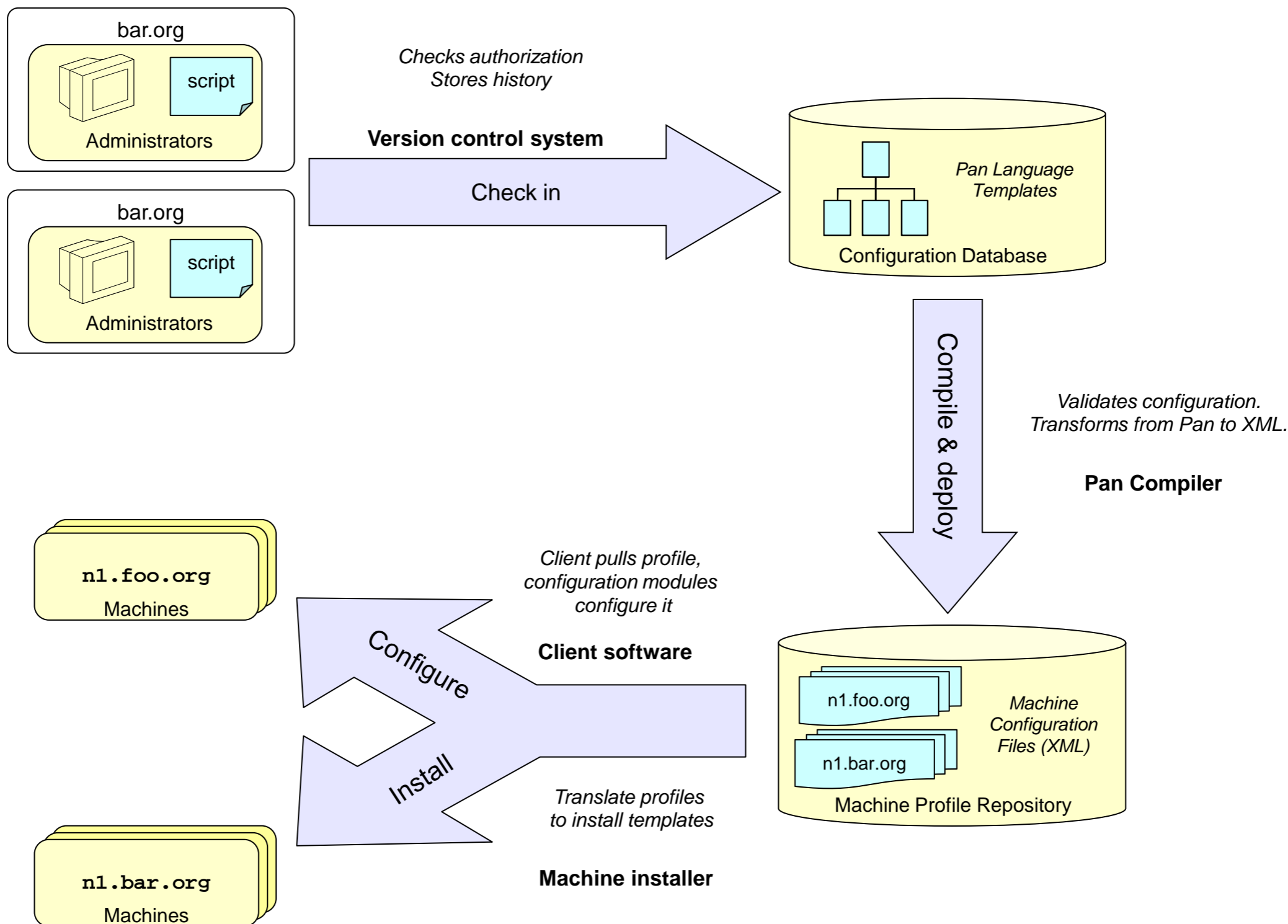
Quattor History

- Quattor started a part of European Data Grid (EDG)
 - First European grid-related project: 2001-2004
 - Funded by EC
 - Large contribution from CERN with its experience
- Since 2004, maintained and developed by the community of people who use it
 - Community increased for ~5 sites to 50+:
 - Many large/very large EGEE sites: CERN, RAL, CNAF, GRIF...
 - Since 1 year, largest site outside HEP: bank (20K nodes)
 - Major contributor, committed to open-source model
- Main core components used today have been developed by the community since 2004
 - PAN compiler, SCDB, QWG templates, Quatview...

Quattor Goals

- Enable different management models
 - Traditional central standalone
 - Distributed sites: several cooperating sites (e.g. GRIF)
 - Devolved management: centrally-assisted model
- High-level configuration descriptions
 - Modular configuration descriptions
- Validate configurations before deployment
 - Ability to describe their consistency constraints
- A toolkit : use/replace a subset of the components
 - Pan language is the only mandatory component
- A central management point
 - No need to connect to machines to trigger an action
- Reduced downtimes
 - Normal administration tasks don't require reinstallations
- Revert to previous states!!!

Quattor Architecture/Workflow



Quattor Main Features...

- Configuration described in term of final state
 - No description of how to reach the state (actions)
- Abstract, service oriented description of configuration
 - Written in PAN (declarative) language and **compiled**
 - Organized in "templates" that can be heavily factorized
 - Russian puppet-like configuration description
 - Configuration consistency checked at compile time
 - Information schema and validation functions
- Configuration stored in a versioned central database
 - Allow very easy rollback of any change
 - 3 implementations with different focus: flexibility, scalability
 - Underlying technologies: CVS, SVN, Git!
 - Different implementations of deployment serialization
- Packages deployed with a special module: SPMA
 - Explicit list of version/arch: easy rollback
 - YUM-based tool to check dependencies before deployment
 - Currently restricted to RPM

... Quattor Main Features

- Configuration implemented through *Quattor Configuration Modules*
 - Configuration Modules are plugins running on client :
 - More than 200 available: ½ core services, ½ gLite services (grid)
 - Extensible framework : easy to add new components
 - Simple API to access the configuration
 - Currently restricted to Perl language
- Manage both installation and configuration changes from the same configuration description
 - Kickstart configuration file **generated** from the configuration description
 - Kickstart does the minimal installation to start the Quattor client
 - Final configuration implemented through standard execution of Quattor Configuration Modules
 - Ability to trigger reinstallation with a shell command

Configuration Deployment

- Each machine as its own low-level profile
 - Reduces security risks or possible interference
 - Profile fetching through http
 - Possible to use https, w/ or w/o authentication (certificates)
 - Not versioned: only the configuration is versioned
 - Easy to rebuild profile corresponding to a previous configuration
 - XML file: not intended to be edited!
- Pull model: client fetches its new configuration
 - Possibility to notify it a new configuration is available
- Received configuration is valid
 - Each profile validated separately, no matter shared templates
 - Deployment will work... except configuration module bugs!
 - If it works on 1 machine, works on all similarly configured
- Configuration is idempotent
 - Produce the same configuration if replayed several times
 - Even if machine is reinstalled (except live data)

QWG Templates

- A set of generic configuration descriptions
 - Covers OS configuration, standard services such as Web servers, Nagios configuration, gLite services
 - An optional part of the toolkit: used by most of the sites
 - Provide a high level of sharing between sites
 - 1 of the most distinctive features of Quattor: **acts as a repository of best-practices**
 - Much more powerful and efficient than sharing recipes
- Some advanced built-in features such as OS errata deployment
 - « Easy » deployment of errata with flexible support of different policies for massive upgrades
 - E.g. staged deployment
 - Seamless handling of kernel upgrades
 - Active kernel removed but only after a successful reboot
- Profile cloning: tweak a reference profile

Template Examples...

- All profiles for a specific machine types identical

```
object template profile_grid23;

include machine_types/lal_wn;

include repository/common;
```

A complete, real
example

- WN type defined using generic templates + variables to set site specific parameters

```
unique template machine_types/lal_wn;

# GRIF specific configuration for a WN
variable WN_CONFIG_SITE = "site/grif/wn";

include machine-types/wn;
```

Another complete,
real example

- IP address, OS version... specified by variables either in the machine profile or shared

```
variable OS_VERSION = nlist(
  escape("grid11.lal.in2p3.fr"), "s1307-i386",
  escape("grid15.lal.in2p3.fr"), "s1440-x86_64",
  escape("grid16.lal.in2p3.fr"), "s1440-x86_64",
  escape("grid17.lal.in2p3.fr"), "s1440-x86_64",
  escape("grid23.lal.in2p3.fr"), "s1440-x86_64",
);
```

... Template Examples

- Machine HW configuration is defined in a site specific template using generic templates

```
structure template hardware/machine/200/48/ibm1/slot30;  
  
"location" = "200_48_ibm1_30";  
"serialnumber" = "KKWMZ7T";  
  
"cpu" = list(create("hardware/cpu/opteron_248"),  
             create("hardware/cpu/opteron_248"),  
             create("hardware/cpu/opteron_248"),  
             create("hardware/cpu/opteron_248"));  
  
"harddisks" = nlist("sda", create("pro_hardware_harddisk_sata", "capacity", 80*GB));  
"ram" = list(create("hardware/ram/generic", "size", 8192*MB));  
  
"cards/nic" = nlist("eth0", create("hardware/nic/tg3"),  
                  "eth1", create("hardware/nic/tg3"));  
"cards/nic/eth0/hwaddr" = "00:11:25:C4:20:DA";  
"cards/nic/eth1/hwaddr" = "00:11:25:C4:20:DB";  
"cards/nic/eth1/boot" = true;
```

Site Parameters

- Site parameters are stored in site-specific templates
 - No modification of generic templates

```
variable SE_ARCH ?= nlist(SE_HOSTS[0], 'multidisk',
                        SE_HOSTS[1], 'multidisk');

variable SE_HOST_DEFAULT ?= SE_HOSTS[0];
variable SE_HOST_DEFAULT_SC3 ?= SE_HOSTS[1];

variable SE_DPM_DISK_HOSTS ?= nlist("grid07."+SITE_DOMAIN, "1");

variable LFC_HOST      ?= "grid07."+SITE_DOMAIN;

variable RB_HOST      ?= "grid09."+SITE_DOMAIN;
variable PX_HOST      ?= "grid02."+SITE_DOMAIN;
variable BDII_HOST ?= "grid01."+SITE_DOMAIN;

# BDII CONFIGURATION -----
# -----

variable BDII_URLS ?= {
  urls = nlist(
    "CE", "ldap://" + CE_HOST + ":" + to_string(GRIS_PORT) + "/mds-vo-name=local,o=grid",
    "LFC", "ldap://" + LFC_HOST + ":" + to_string(GRIS_PORT) + "/mds-vo-name=local,o=grid",
    "LFCLAL", "ldap://" + 'grid14.lal.in2p3.fr' + ":" + to_string(GRIS_PORT) + "/mds-vo-
name=local,o=grid",
    "RB", "ldap://" + RB_HOST + ":" + to_string(GRIS_PORT) + "/mds-vo-name=local,o=grid",
    "PX", "ldap://" + PX_HOST + ":" + to_string(GRIS_PORT) + "/mds-vo-name=local,o=grid",
    "MON", "ldap://" + MON_HOST + ":" + to_string(GRIS_PORT) + "/mds-vo-name=local,o=grid");
```

Managing History (SCDB+Trac)

Added
 Modified
 Copied or renamed

View changes

Old	New		Date	Rev	Chgset	Author	Log Message
<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	03/06/06 14:12:07	@3599	[3599]	jouvin	Fix disk information for grid27
<input checked="" type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/06/06 14:02:11	@3597	[3597]	jouvin	Fix disk information for grid27
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 18:42:31	@3596	[3596]	lpnhe	Modifications following the last updates in the Grif site structure at ...
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 17:51:31	@3594	[3594]	ipno	modif pro_hardware_machine..hp_proliant..
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 17:41:04	@3591	[3591]	ipno	modif pro_hardware_machine..hp_proliant..
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 17:35:47	@3588	[3588]	ipno	modif pro_hardware_machine..hp_proliant..
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 16:05:39	@3585	[3585]	ipno	ajout ipnsedpm.in2p3.fr
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 15:35:16	@3583	[3583]	jouvin	Move grid27 to SL4.2 test cluster
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 15:07:19	@3580	[3580]	jouvin	Rename cluster orme-slc42 slc/ to machine-types/, remove previous ...
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 14:57:54	@3579	[3579]	jouvin	Rename cluster orme-slc42 lal/ to machine-types/
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 14:37:02	@3577	[3577]	jouvin	Definition of machine type pro_lal_desktop
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 14:36:01	@3576	[3576]	jouvin	Definition of machine type pro_lal_desktop
<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	03/03/06 14:20:20	@3575	[3575]	jouvin	...

Managing History (SCDB+Trac)

Changeset 3568

Timestamp: 03/03/06 11:14:27

Author: jouvin

Message: Use an OS version independent template to add openafs client; define auger1 as an xtremweb server

Files:

- [trunk/cfg/clusters/lal-sl420/profiles/profile_auger1.tpl](#) (1 diff)
- [trunk/cfg/os/sl305-i386/os/pro_os_openafs_client.tpl](#)
- [trunk/cfg/os/sl420-i386/os/pro_os_openafs_client.tpl](#)
- [trunk/cfg/sites/lal/machine-types/pro_lal_config_afs_client.tpl](#) (1 diff)


Unmodified
 Added
 Removed
 Modified
 Copied
 Moved

trunk/cfg/clusters/lal-sl420/profiles/profile_auger1.tpl

r3516	r3568	
10	10	
11	11	define variable XW_STARTUP_START = false;
12		#include pro_lal_server_physics_xtremweb;
13		include pro_lal_server;
	12	include pro_lal_server_physics_xtremweb;
14	13	
15	14	

trunk/cfg/sites/lal/machine-types/pro_lal_config_afs_client.tpl

r3371	r3568	
12	12	define variable PKG_ARCH_KERNEL_MODULE_OPENAFS = PKG_ARCH_KERNEL;
13	13	

 Terminé

GRIF Challenges

- Build one homogeneous grid site from 6 sites near Paris
 - Have a predictable behavior of the whole site
 - Easy and transparent relocation of resources and load
 - Same quality of service whatever is the number of administrators at each site
- Leverage administration of the whole site
 - No possibility to hire anybody for the project : required to build a technical team with existing people
 - Very few dedicated grid person
 - Take the most from existing manpower : avoid duplication
- Increase the overall expertise
 - No site large enough to take control of the others
 - Very few grid "experts" at the beginning : 2-3 people
- Integrate operation at each site with non grid systems
 - Generally the same people managing both

GRIF Achievements

- Being 1 site created a driving force for funding mutualization between various groups
- Expertise dissemination is a reality
 - Between 1 and 3 grid admin “experts” per site
- GRIF-wide technical team is a reality
 - Very rich human adventure, **the main GRIF asset**
 - Lots of everyday cross-site interactions
 - 1 face-to-face meeting per month
 - Configuring a service, installing an update... can be handled by one person for the whole GRIF
 - Can change from one time to another
- Each site still has control of its own resources
 - Local customization possible : storage integration, authentication infrastructure, network params...
 - Integration of operations with non grid systems underway at most of the sites.

Quattor Project Model

- Quattor is an open-source project since the very beginning
 - Strong commitment to remain such
- Small but active community adhering to the *agile software approach*
 - Never really formalised...
 - Strong interaction between developers and users
 - In fact most developers are also users (system administrators)
 - Early release/feedback: using SVN branches to identify work in progress
 - “Continuous releases” rather than “big releases”
- Nothing prevent commercial exploitation
 - Lack of SLA-based support identified as a potential barrier for adoption in large context,
 - Despite Quattor unique scalability
 - A SME looking at possibility to offer commercial consultancy and support

Quattor Licensing

- 2 open-source licences used:
 - Components developed during initial phase: EDG licence
 - <http://www.opensource.org/licenses/eudatagrid.php>
 - Since 2004 (end of EDG project): Apache2
- Both known to have compatible terms
 - Allow the reuse of the code to build other products with different (more restrictive) license terms
 - Plan to move everything under Apache2
- No use of code from external software with different licenses

Quattor Community

- Made of everybody using or contributing to Quattor
 - Quattor contributors happened to be also Quattor users: part of Quattor's culture
 - 50+ sites, mainly in Europe but also Asia and Africa
- No formal organisation currently
 - About to study the feasibility and potential benefits of some sort of consortium
 - Part of the QUEST project to be submitted to EC soon...
 - Will study what has been done by other similar projects
- Main activity: a twice-a-year workshop
 - Where people agrees on priority
 - No management structure that can impose priorities
 - Consensus-based decisions contribute to the strength of the community
 - Despite it is not always the fastest method on the short term...
- A few « recognized » leaders but not a formal board

Quattor Project Methodology

- Source code in SVN
 - Hosted at Source Forge (+ LAL for QWG Templates)
 - Everybody can be a committer: just ask...
 - A specific branch dedicated to contributions
 - Branches for (almost) stable code
 - Recommended for production
 - Also "stable" releases: immediately outdated...
- All the documentation in wiki
 - MediaWiki (SourceForge) + Trac (QWG Templates)
 - Everybody invited to contribute: no moderation...
 - ... but people tends to be reluctant
 - IMO main cause: worry to write something inaccurate or not appear an expert
 - No alternative for an up-to-date documentation
 - Personal methodology: answer email requests in the wiki

Widening Quattor Usage

- We believe Quattor has some unique features making it attractive in a lot of contexts
 - Recent (2 years ago!) adoption by a bank confirm this
 - Most current restrictions not coming from the architecture but from the lack of demand or manpower
 - QUEST proposal about to be submitted to EC FP7 would like to address these more pro-actively
- We have mainly an academic background...
- But we don't want to restrict Quattor to this world
 - Not easy to find the right contacts/channels outside this world...
 - Commercial consultancy and support (with clear SLA) often a requirement

Summary

- Quattor is a SW aiming to handle site IT resource configuration, with an emphasis on consistency and validation before deployment
 - Currently restricted to Linux/Red Hat and its derivative
 - Plans to enable support of other Linux/Unix
- Main usage context is EGEE site/resource management but is able to do much more
 - Used in many other contexts: desktop management, virtual machine management...
 - In fact the largest Quattor site today is a large bank...
- Quattor community very active
 - Very committed to the open-source model
 - Thinking about a more formal project organisation for the long-term sustainability of the project
 - Without compromising current flexibility and *agile SW approach*
 - Looking at new resource provisioning models like clouds

Useful Links

- Quattor main site : <http://quattor.org>
 - Hosted at SourceForge
 - Core toolkit, Quattor Configuration modules
 - General documentation and case studies
 - Pan language reference manual
 - Different presentations about Quattor
- QWG site : <http://trac.lal.in2p3.fr/QWG>
 - Documentation about QWG templates
 - OS and core service templates
 - gLite middleware templates
 - Documentation of SCDB
 - Actively updated
 - Eventually, should merge into SourceForge
- PLUME-FEATHER: <http://www.projet-plume.org>
 - <http://www.projet-plume.org/fr/fiche/quattor>