Devolved management of distributed infrastructures with Quattor

06/08 – LNGS L'Aquila

Marco Emilio Poleggi – *INFN-CNAF*

Marco.Poleggi@cnaf.infn.ch

Outline

- Background
- Devolved management workflow
- Distributed infrastructures with QWG
- Integration with other tools
- Conclusion

Background…

- Grid computing is changing the way resources are used
  - From big data centers to geographically distributed “federated” sites
  - Focus on the tools for managing the “fabric” (HW/SW). Requisites
    - <u>Scalability</u> to deal with large installations
    - <u>Flexibility</u> to accommodate heterogeneous frameworks
    - <u>Modularity</u> to optimize configuration data usage
  - Performance to flexibility trade-off
    - More actors on the scene: different responsibilities
    - Sites want local autonomy but have reduced knowledge of the configuration system
    - Share common configuration parts
    - Allow local policies

...Background

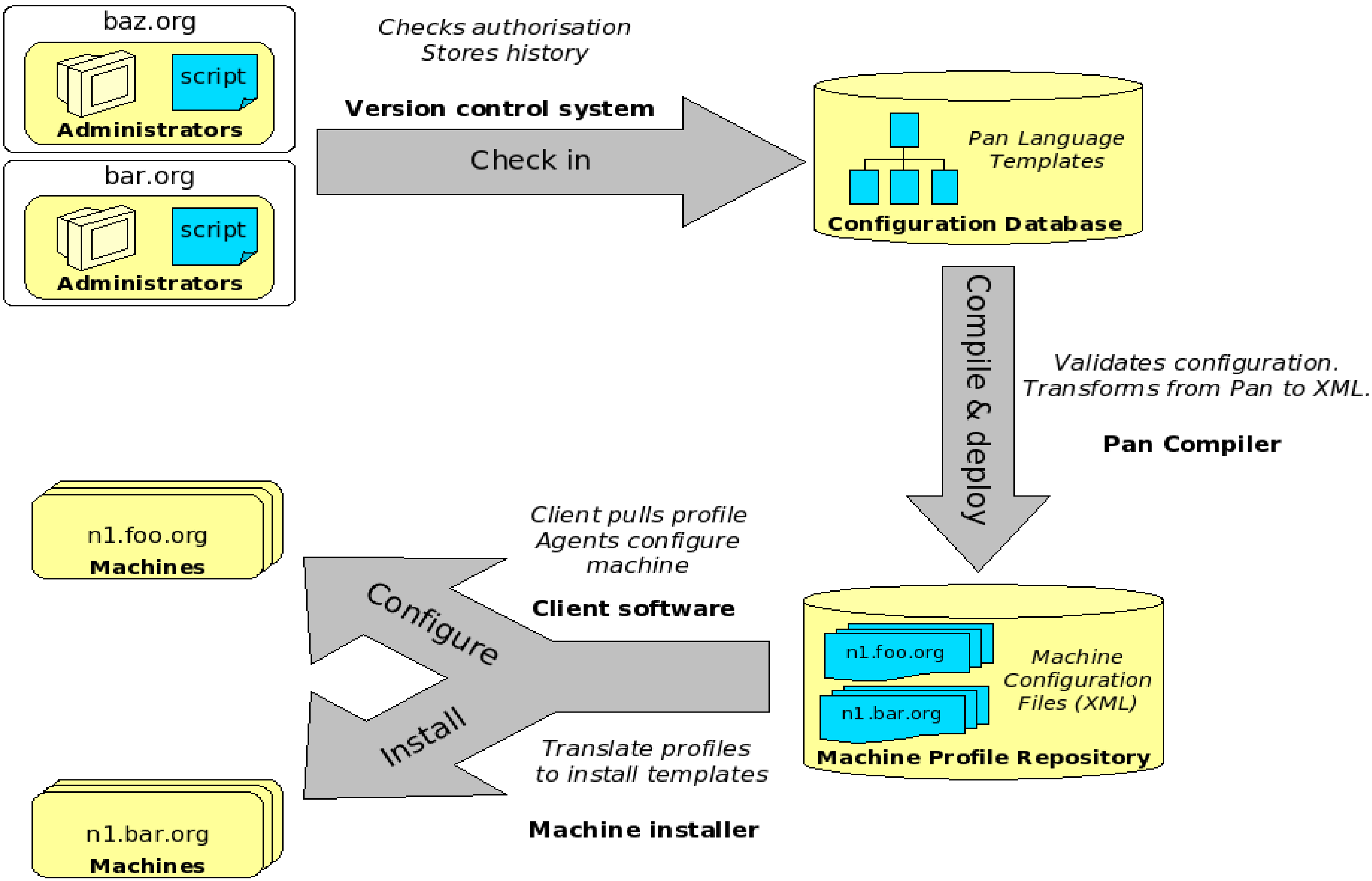
- Quattor was developed to meet the above requirements
  - Uses a high-level *declarative* configuration language – *Pan*
    - Hierarchical schema
    - Provides modularization for data reuse and customization
    - Supports pre-deployment checks through *validation*
  - Is flexible in service deployment
    - Version control systems; configuration and software repositories; software management agent
  - Uses open standards such as HTTPS, X.509 and XML

Table 1: Quattor deployments

Metric	BEGrid	CERN	CNAF	Grid-Ireland	GRIF	Nikhef	UAM
Managed machines	260	8000	800	417	575	301	553
Administrators	8	100	10	11	25	4	3
Physical sites	6	1	1	18	6	1	1

Devolved management workflow...

quattor



...Devolved management workflow...

quattor

- Configuration management system
  - Subsystem deployment can be
    - *Centralized* for strict operation control on the server
    - *Distributed* for more flexibility (profile compilation on the clients)
  - Authentication via X.509/KRB5/encrypted passwords
  - Authorization via access control lists (ACLs)
- Automatic installation of managed nodes (all operations can be done remotely)
  - Retrieves information from machine profiles
  - Configures DHCP and PXE
  - Generates Kickstart files

...Devolved management workflow

- Node configuration management
  - Notification of changes + download of fresh profiles
  - Autonomous agents (“components”) triggered by changes in specific parts of the configuration schema
    - Deployment can be also manual
    - Pre/post runtime dependencies ensure correct service start up
  - Idempotent (repeated actions have the same effect)
- Software management
  - Separation of *repository* and *configuration*
    - Different repositories accessed via HTTP
    - Package lists in Pan templates
  - Modes
    - *Strict* -- install only listed packages, remove manual installations
    - *Flexible* -- allow multiple versions, respect manual installation
  - Rollbacks can be easily performed

Pan language...

quattor

- Definition of configuration information with an associated schema

```
'/hardware/network/hwaddr' = '00:11:22:33:44:55:66';
```

- Validation: allows to check constraints before deployment

```
# Validate MAC addresses. Allows either hyphens or colons as
# separators.
function is_hwaddr = {
  match(ARGV[0],
    '^[\dA-Fa-f]{2}([:-])[\dA-Fa-f]{2}(\1[\dA-Fa-f]{2}){4}$');
};
# Define type using is_hwaddr() function.
type type_hwaddr = string with is_hwaddr(SELF);
# Bind this to a particular path. Path must be a list of MAC
# addresses.
bind '/hardware/network/hwaddr' = type_hwaddr[];
```

...Pan language

- Modularization and configuration reuse:
  - Structure templates

```
structure template hardware/cpu/intel_xeon_E5420;
  'vendor' = 'Intel';
  'model' = 'Intel(R) Xeon(R) E5420 2.50GHz';
  'speed' = 2500*MHz;
  'arch' = 'x86_64';
```

This would then be incorporated into a machine configuration as follows:

```
'/hardware/cpus' = list(
  create('hardware/cpu/intel_xeon_E5420'),
  create('hardware/cpu/intel_xeon_E5420'));
```

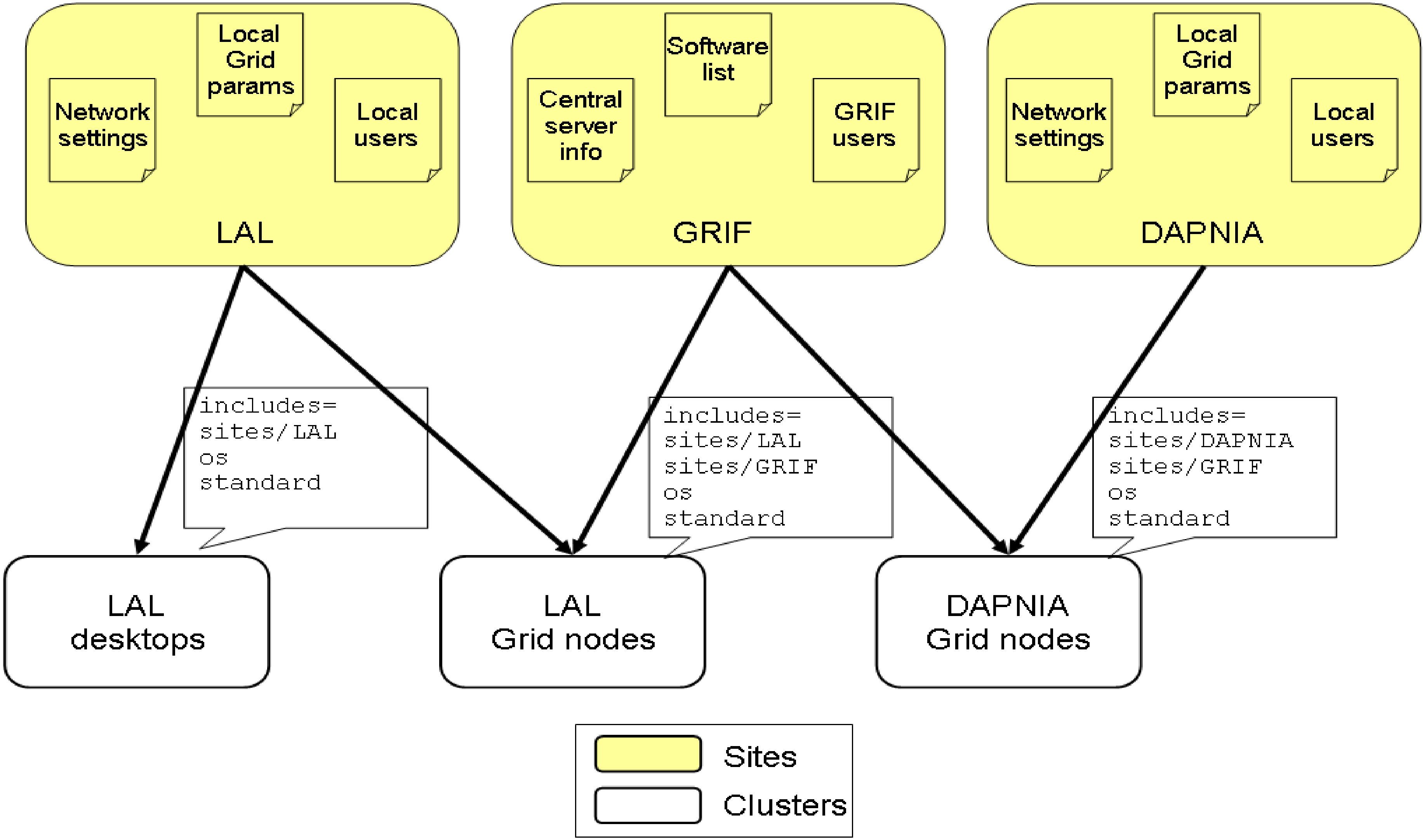
  - *Namespacing* + *loadpath* permit to import different configuration parts depending on some parameters
    - “Modules” are located in different namespace hierarchies
    - The desired namespace is selected by the loadpath value

Distributed infrastructures with QWG...

quattor

- *QWG templates* are a full Quattor “distribution” for grid services
  - Large base of shared configuration.
  - Minimal set of parameters to adapt the services to the local environment.
  - Variables for conditional *includes* allow local customizations.
  - The configuration is based on the concepts of *site* and *cluster*.
    - A *cluster* is an arbitrary grouping of machines that share configuration information (for example, “compute nodes” or “grid servers”).
    - A *site* is a logical group that defines a set of configuration elements to be shared by different clusters.
- Typically, a single Quattor instance is used to manage the physical sites.

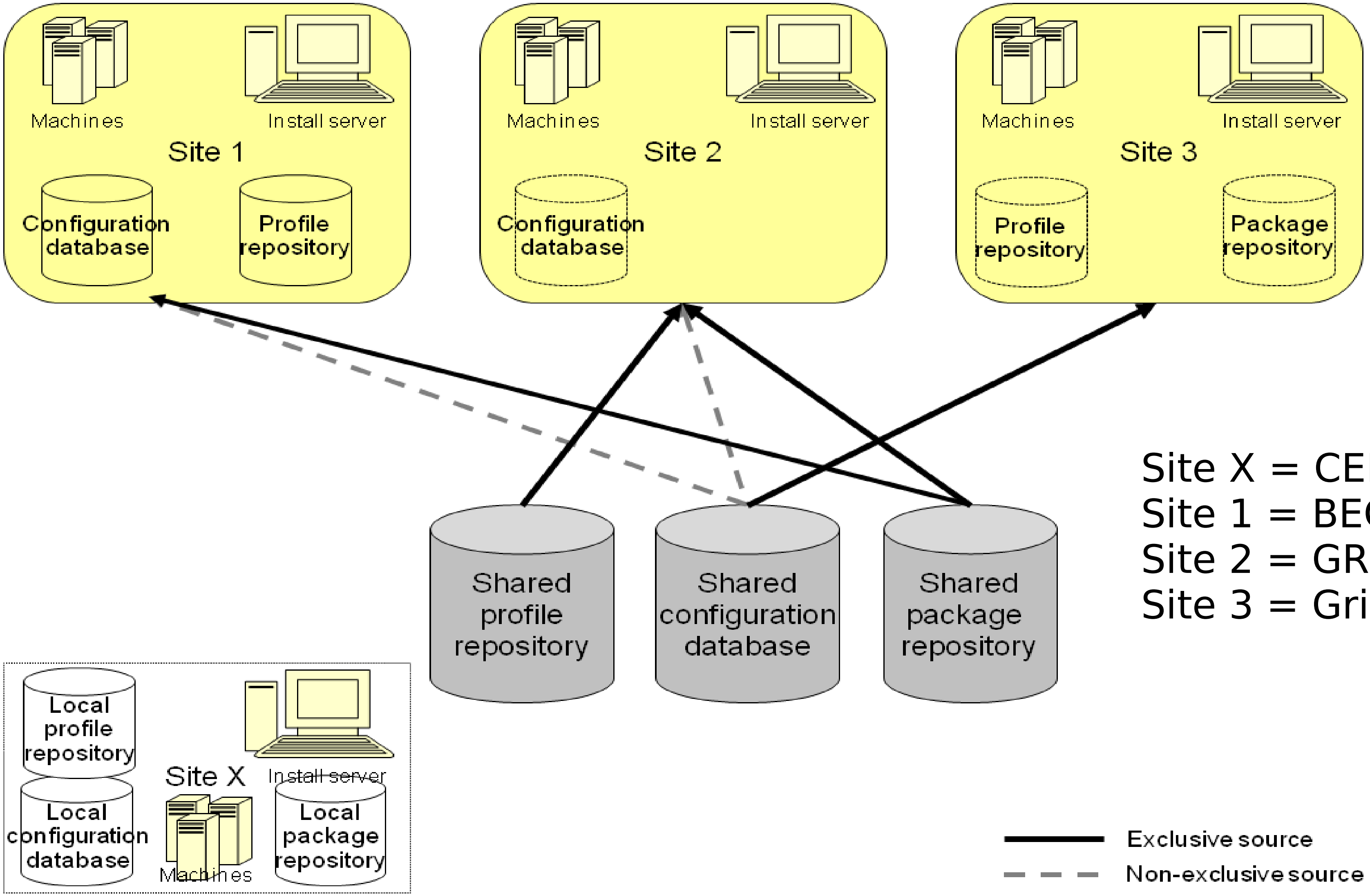
- Sites and clusters configuration at GRIF



...Distributed infrastructures with QWG

quattor

- Physical deployments of Quattor services



Site X = CERN, CNAF
Site 1 = BEGrid
Site 2 = GRIF
Site 3 = Grid-Ireland

Management: demanded vs stand-alone

quattor

- Demanded means: a “father” site acts as a service provider to “children” sites
  - Half way between centralized and stand-alone models
- Advantages
  - The configuration base is maintained by developers at the father site
    - children sites get updated automatically
    - Reduced knowledge of configuration's “guts” at children sites ~ reduced manpower
  - Children sites are autonomous for
    - Local customizations (both software and configuration)
    - Also operations, depending on deployment's set-up
- Drawbacks
  - Direct responsibility at the father site for the shared configuration deployment ~ increased manpower

Integration with other tools

quattor

- Quattor is not so invasive as it may seem
  - Only what's expressly defined in the configuration schema is touched on the live system, the rest is ignored
- Integration means:
  - Preparing a schema and developing Pan templates describing the service
  - Developing “components” to enact the service configuration
- Currently there is support for
  - Monitoring systems (Lemon and Nagios)
  - Virtualization tools (Xen and OpenVZ)
- Moreover, Quattor allows peaceful coexistence with Windows desktops ;-)

Conclusion

- Quattor has demonstrated effectiveness and flexibility in a wide spectrum of site configurations
  - The benefits are clear especially when managing large “farms”
  - The learning curve is rather steep, though the community's support alleviates the pain ;-)
- CNAF is converging towards a well structured configuration system for the Tier-1, partially based on QWG templates
- The industry has recently shown interest in Quattor
- There's still room for improvements
  - An authorization mechanism in the Pan language is desired

quattor

http://quattor.org/