Configuration management benefits for everyone

Normation
Nicolas Charles <nch@normation.com>
Jonathan Clarke <jcl@normation.com>
<table>
<thead>
<tr>
<th>Speakers</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Nicolas Charles</td>
<td>Jonathan Clarke</td>
</tr>
<tr>
<td>Scala developer</td>
<td>Sysadmin</td>
</tr>
<tr>
<td>Works at Normation</td>
<td>Works at Normation</td>
</tr>
<tr>
<td>Rudder developer</td>
<td>Rudder developer</td>
</tr>
<tr>
<td>CFEngine expert</td>
<td>CFEngine expert</td>
</tr>
<tr>
<td>CFEngine Community</td>
<td></td>
</tr>
<tr>
<td>Champion</td>
<td></td>
</tr>
</tbody>
</table>
Service management

Make sure the service does it's job
- Install & Update
- Configure
- Run

Security
- User accounts
- Password policy
- Backups
- Log everything
- Security patches

Availability
- Limit the impact of a failure
- Scale out
- Plan for disaster recovery

Knowledge
- Document configuration
- Formalize procedures
- Log changes
Configuration management benefits

Automate
- First install + reinstalls
- Update
- Configure

Collaboration
- More knowledge:
  - Centralize information
  - Full change log
- Less documentation:
  - Less written procedures
  - More automation

Regular checks
- Install OK?
- Configuration OK?
- Integrity?

Industrialization
- Re-use (configs, policies...)
- Reporting on config status
- Dashboards

More knowledge:
- Centralize information
- Full change log
Less documentation:
- Less written procedures
More automation
In some situations, configuration management may be too much overhead...

For the all the rest, advantages are undeniable!

But does everyone really benefit?

Managers?  
Junior sysadmins?  
Non specialists?
Goals

Lower the learning curve to use CM

This may mean losing some flexibility but mustn't mean losing efficiency

Share CM benefits with a wider population

Different information and capabilities for different people
Fundamentals

Build on reliable tools
- **CFEngine®**
  - Based on CFEngine
  - Lightweight and powerful
- **FUSION INVENTORY**
  - Automatic inventory

Share
- **Open source**
- **OS-specific packages**
- **Library of infrastructure configurations included**

Improve
- **Web interface**
- **Reporting graphique**
Principle

New nodes

Web interface on Rudder server

View node data

Put nodes in groups

Configure rules on groups

View infrastructure status

Inventory

Managed nodes

CFEngine policy

Reports
Configuration Rules

Predefined templates to manage systems
- Install packages, distribute files
- Manage users, distribute SSH keys
- Configure DNS, NTP, package managers
- Schedule backups
...

Parametrization in the Web Interface
- Forms to change defaults

Conversion into CFEngine Policies
- Applied by CFEngine agents
Current status

Version 2.3 released in October 2011

Web interface to manage nodes and configuration rules

Real time reports on infrastructure status

Policy Templates (currently 33)

All changes logged

Packaged for main Linux distributions
Demonstration
Installing a Rudder server

```bash
# echo 'deb http://www.rudder-project.org/apt-2.3/ squeeze main' >> /etc/apt/sources.list
# aptitude update
# aptitude install rudder-server-root
# /opt/rudder/bin/rudder-init.sh
```

Installing Rudder on a node to manage

```bash
# echo 'deb http://www.rudder-project.org/apt-2.3/ squeeze main' >> /etc/apt/sources.list
# aptitude update
# aptitude install rudder-agent
# echo "server address" > /var/rudder/cfengine-community/policy_server.dat
```
Requirements (node)

Small amount of free RAM (10-20 MB)

Some dependencies
- SSL
- BerkeleyDB
- PCRE
- Syslog

Memory occupation of CFEngine daemons
Rudder architecture

- Based on typical CFEngine architecture
Rudder architecture

- Extra components on the server only

Generate CFEngine policy

Rudder server

CFEngine server

Node

Node

Node

Node

Communications by TCP (port 5309)
- File metadata
- File contents
- Send inventories (FusionInventory)
- Send reports (syslog)
Rudder workflow

**Policy Templates**
- CFEngine syntax
- Variables for web configuration

Enter variables in the web interface

**Policy Instances**

**Configuration Rule**
- Apply Policy Instances to a Group

**Nodes**
- Search criteria on inventory information
  - Hardware / OS / Network
  - Software
  - Node name

Create a group

**Group**
Extend

Write new Policy Templates
- Based on CFEngine 3
- An XML descriptor to set up the web forms
- Configure anything!

Write plugins for the webapp
- Plugins are automatically discovered at startup
- Implementation example:
  https://github.com/Normation/runder-plugin-helloworld
Roadmap

2.4: February 2012
- Import/Export configurations across Rudder servers
- Approval process for changes before deploying them
- More and better Policy Templates
- Deleting nodes
- Simple REST API

2.5: Mid 2012
- Better Policy Configuration display
- More detailed reporting
- Authorizations
Community

- Source code on GitHub
- Documentation wiki
  - http://rudder-project.org
- Small open source community
  - Mailing lists
    - rudder-users@lists.rudder-project.org
    - rudder-dev@lists.rudder-project.org
  - IRC : #rudder on FreeNode
  - Twitter: @RudderProject