



Piloter vos services informatiques avec Rudder



Nicolas Charles

<nch@normation.com>



Qui suis-je ?

Nicolas Charles

Développeur Scala

Expert CFEngine

CFEngine Community Champion

CEO de Normation

Développeur de la solution Rudder



- Expert en gestion de configuration
 - Partenaire CFEngine AS
 - Formateur pour CFEngine AS
- Editeur de la solution Rudder
 - Basé sur des outils éprouvés et robustes
 - Simplicité de déploiement et d'utilisation
 - **Open source**



Sommaire

- 1) Introduction : gestion de configuration ?
- 2) Démo
- 3) Concepts de Rudder
- 4) Première approche
- 5) Démo !



Gestion de configuration

Les principes par l'exemple...



Le serveur a crashé

Reinstallez en un, on ne peut pas travailler sans

D'accord, ca sera fait dans deux jours

Gestion de Configuration

Il y a un patch de sécurité critique à déployer sur tous nos serveurs !

Installez le vite

D'accord, je mets toute l'équipe dessus.



Reproductibilité

Industrialisation

Automatisation

Gestion de Configuration



Comment configure-t-on
le service X?

*Demande à Jean,
c'est l'expert du sujet*

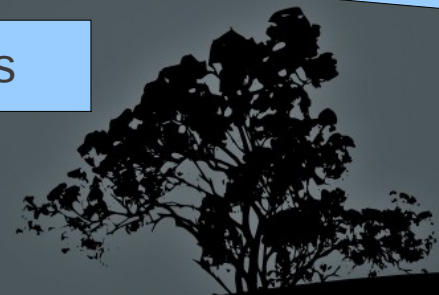
Mais il a démissionné ...

Gestion de Configuration

Hum, ce serveur remonte des erreurs
depuis quelques semaines.

*Ho? Je crois que Michel a changé
quelque chose dessus récemment...
Il vous dira quoi.*

Flute, il est en congés



Documentation

Historisation

**Capitalisation
du savoir**

Gestion de Configuration



On vient de voler nos données en utilisant une vulnérabilité dans un module dont on n'avait pas besoin...

Euh, oui, mais nous l'avons réactivé pour résoudre un problème et nous avons oublié de le désactiver après...
Désolé...

Je pensais que les spécifications indiquaient qu'il devait être désactivé ?

Gestion de Configuration



Gestion de Configuration

Vigilance

```
graph TD; A[Vigilance] --> B[Réparations automatiques]; A --> C[Alertes];
```

Réparations
automatiques

Alertes

Je ne comprend pas comment ce serveur est configuré. Il ne correspond pas à nos règles

Oh, c'est un serveur historique...

Gestion de Configuration

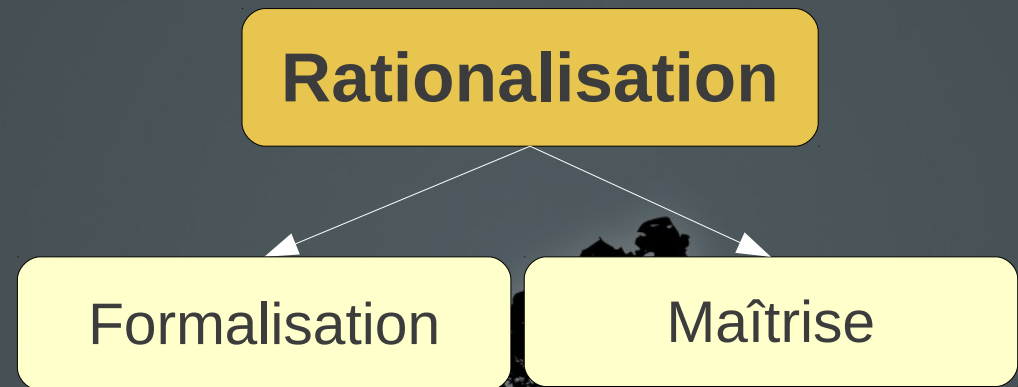
Heu, c'est un ensemble de petites choses, ici et là

Donnez moi des détails sur notre politique de sécurité

Euh ...

*Ah... Bien..
Dites moi: est-elle bien appliquée sur tous nos serveurs sensibles ?*

Gestion de Configuration



Reproductibilité

Industrialisation

Documentation

Historisation

Automatisation

**Capitalisation
du savoir**

Gestion de Configuration

Vigilance

Rationalisation

Réparations
automatiques

Alerte

Formalisation

Maîtrise



Présentation des concepts



Rudder : concepts

- Objectif de Rudder :
 - Démocratiser la gestion de configuration
 - Rendre accessible à tous...
 - ...quitte à perdre un peu de souplesse
 - Sans sacrifier l'efficacité



Rudder : concepts

- Démocratiser la gestion de configuration
 - Application des politiques de configuration paramétrables ...
 - Fournies par Normation et la communauté
 - Qui peuvent être enrichie par les utilisateurs
 - ... sur des groupes de serveurs
 - en utilisant simplement une interface Web



Rudder : concepts

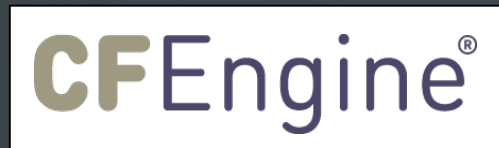
- Fondamentaux :
 - Basé sur des outils éprouvés et robustes
 - Profiter de la puissance des outils à la pointe
 - Simplicité de déploiement et d'utilisation
 - Abstraction de la complexité pour l'utilisateur
 - Open source



Rudder : concepts



Conçu spécifiquement
pour la gestion
de configuration



Basé sur CFEngine,
standard depuis 1993



Inventaire
automatique



open source
Open Source



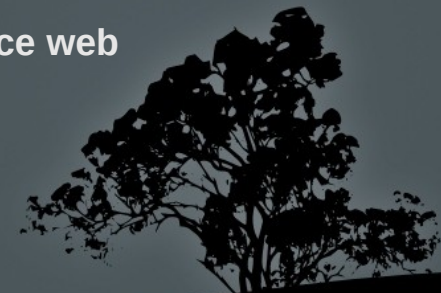
Bibliothèque de *best
practices* incluse



Reporting graphique



Interface web



Rudder : concepts

Multi-plateforme



Rudder : concepts

Multi-OS Multi-distribution

Rendre "transparent" les différences

Chaque configuration type propose des options communes, quelque soit l'OS cible

Adapté aux environnements hétérogènes



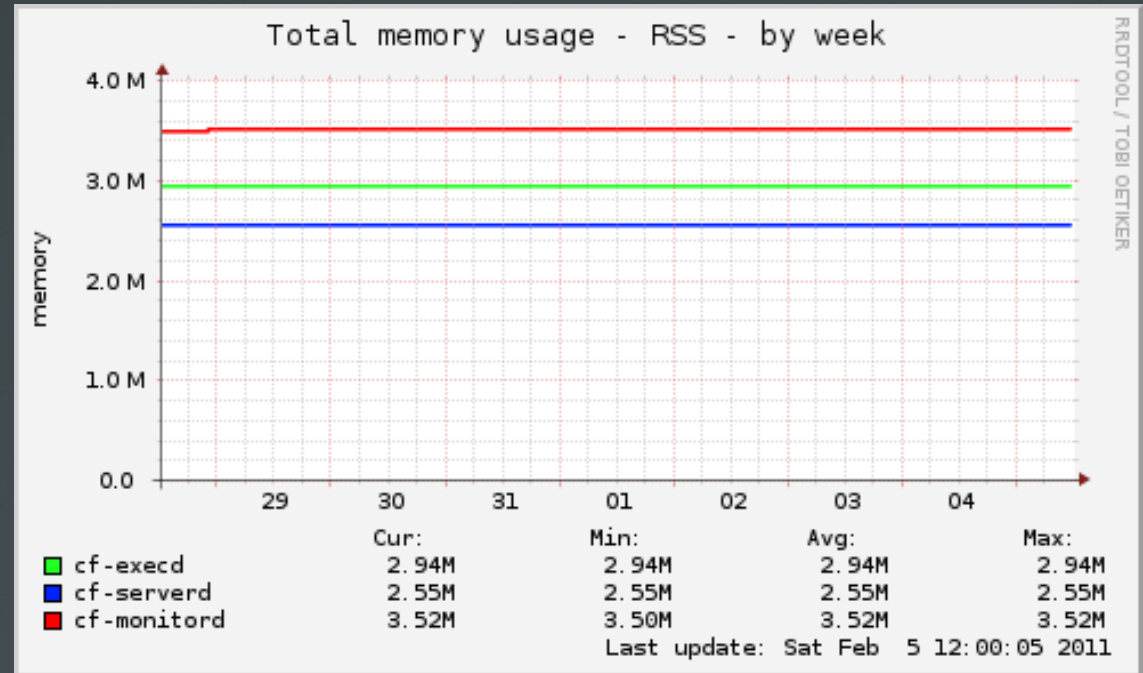
Rudder : concepts

Agent léger, peu intrusif

Non-intrusif

Dépendances simples :

- BerkeleyDB
- OpenSSL
- PCRE



Consommation mémoire des démons (noeuds)

Rudder : concepts

Possibilité de ne gerer que quelques noeuds/services

Choix des noeuds gérés

Choix des politiques appliquées aux noeuds

- Possibilité de ne rien gerer, juste avoir l'inventaire
- Rajout étape par étape des noeuds/politiques

Coexistence possible avec d'autres outils de gestion de configuration

**Déploiement
progressif**



Rudder : concepts

Large bibliothèque de *templates* de configuration livrées avec Rudder

Configurations type système :

- DNS, NTP, hosts, fstab, MOTD
- Serveurs SSH, Apache, NFS
- Gestion d'utilisateurs, de groupes, de sudoers, clés SSH

Bibliothèque
de
configurations



Rudder : concepts

Multi-plateforme



Agent léger, peu intrusif

Adapté aux environnements hétérogènes

Autonome
Tolérant aux pannes

Bibliothèque de configurations

Déploiement progressif

Rudder
Drift assessment



Rudder : concepts

- 3 types d'utilisateurs :
 - Experts qui peuvent créer des politiques pour des besoins spécifiques
 - Administrateurs systèmes qui configurent les politiques et les appliquent
 - Tous ceux qui peuvent s'intéresser à la conformité du SI

Pas de formation préalable nécessaire





Démonstration



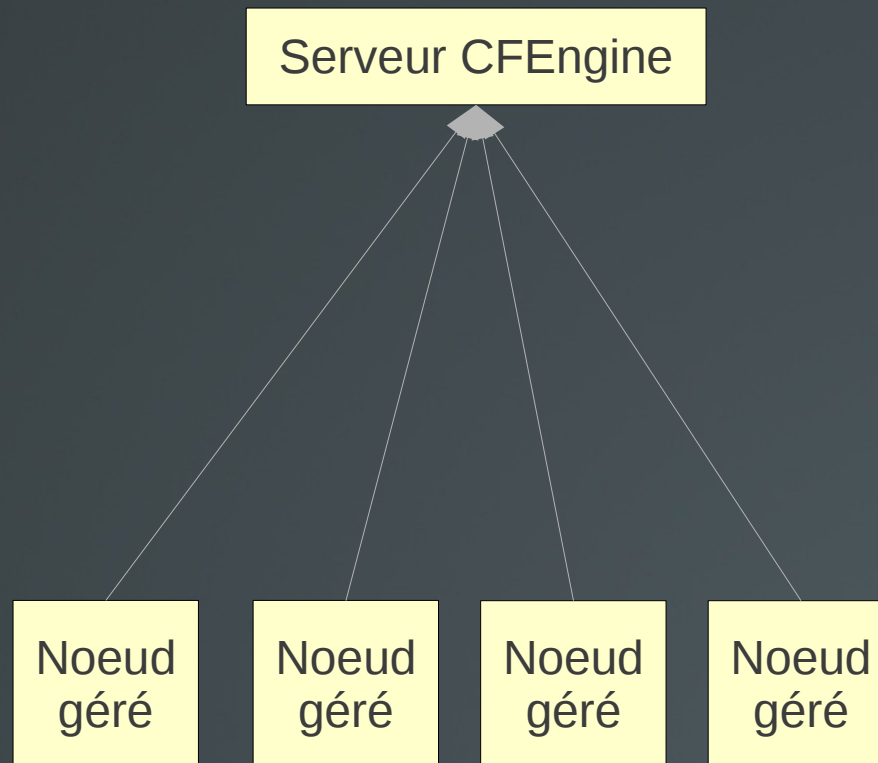


Rudder : première approche



Rudder : première approche

- Architecture CFEngine typique

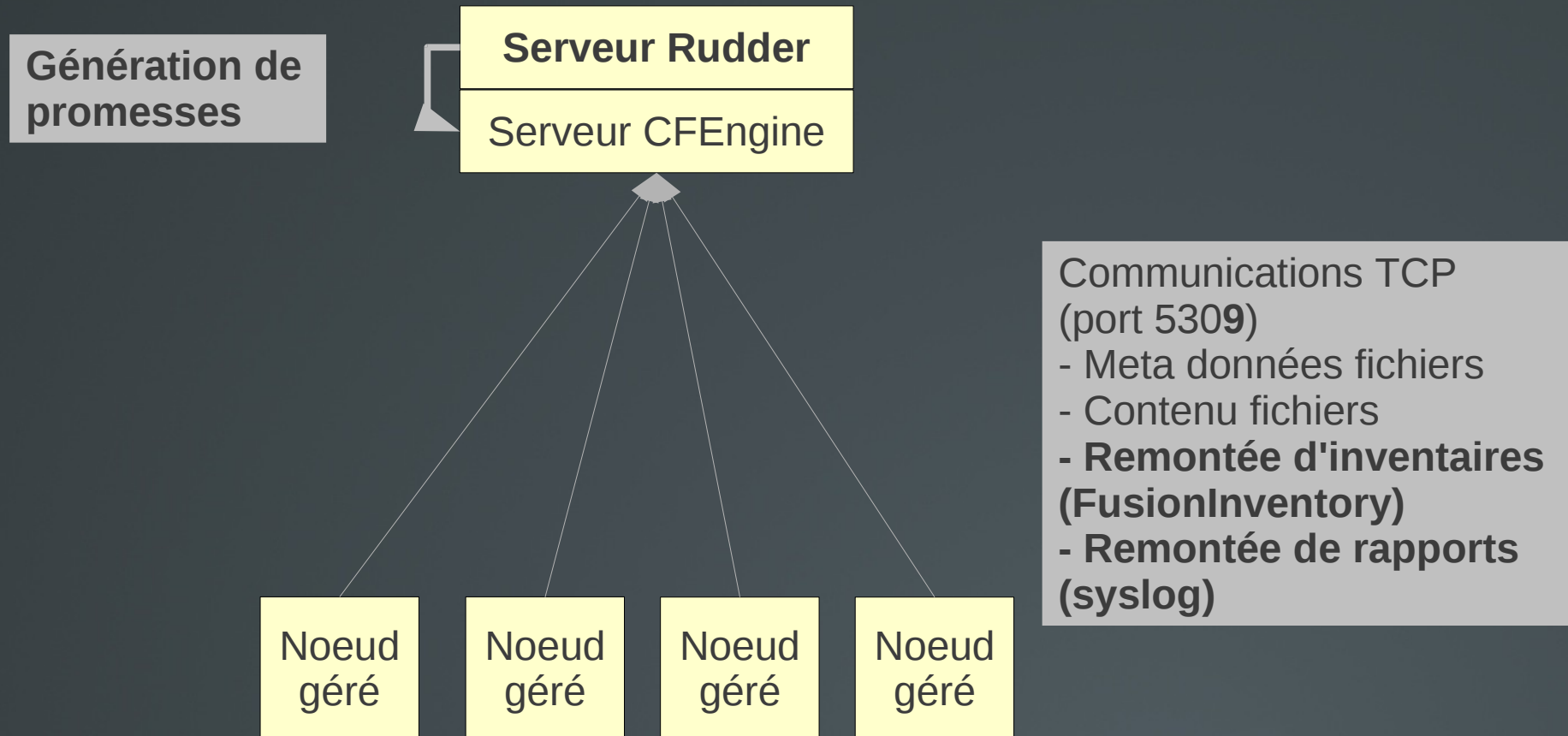


Communications TCP
(port 5308)
- Meta données fichiers
- Contenu fichiers



Rudder : première approche

- Architecture Rudder



Prérequis

- Des serveurs à gérer : "noeuds"
 - Caractéristiques
 - Quelques Mo de RAM disponibles
 - Quelques dizaines de Mo d'espace disque
 - Dépendances : libssl, libpcrc, BerkeleyDB
- Un serveur Rudder
 - GNU/Linux uniquement
 - Debian 5 ("lenny") ou 6 ("squeeze") ou SLES 11
 - Minimum 1 Go de RAM
 - Machine virtuelle Java (JRE)



Installation

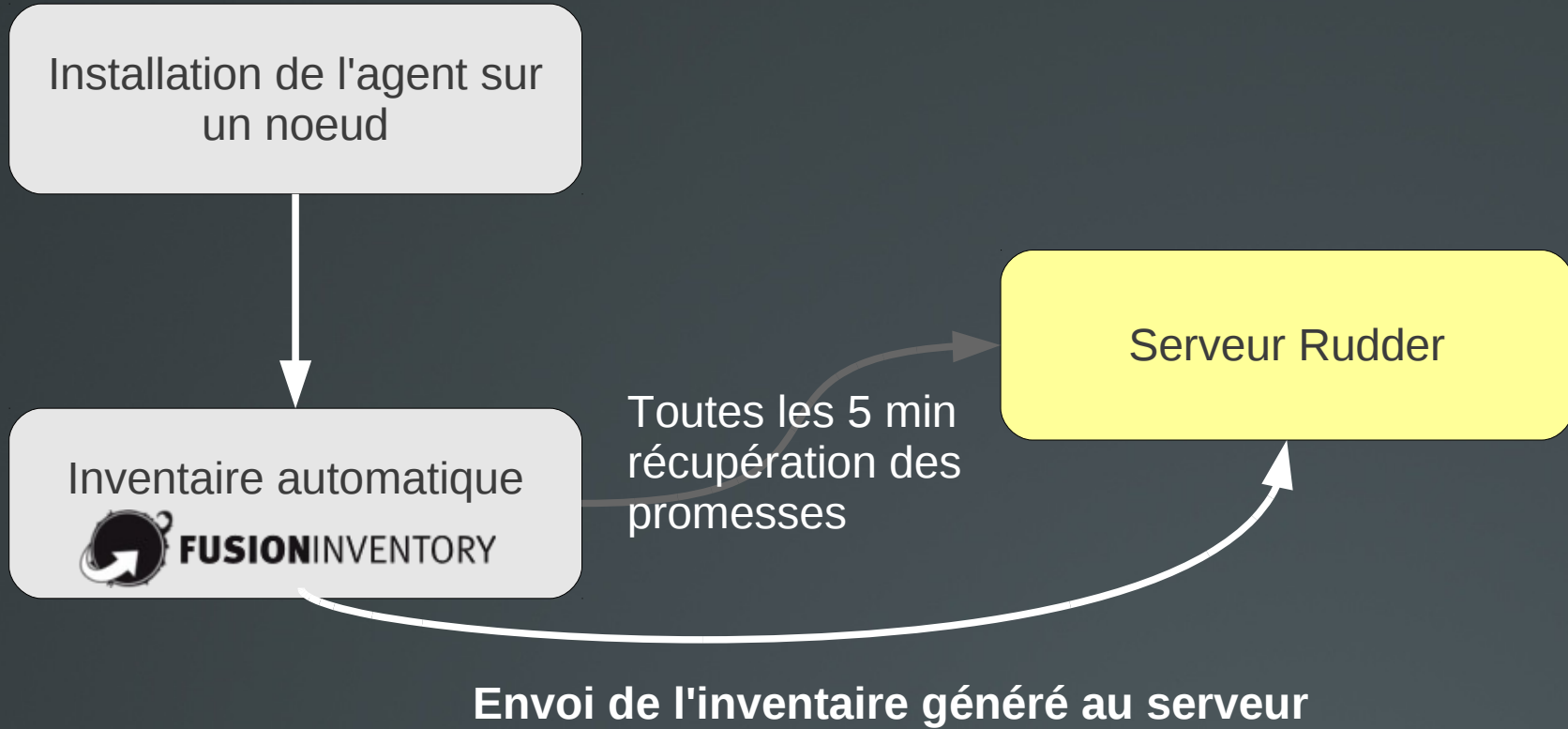
- Documentation en ligne
 - <http://www.rudder-project.org/foswiki/Download/>
 - Pour Debian :
 - Serveur Rudder :

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

- Noeud à gérer :

```
# echo 'deb http://www.rudder-project.org/apt-2.3/ squeeze main contrib non-free'  
>> /etc/apt/sources.list  
# aptitude update  
# aptitude install rudder-agent  
# echo "adresse du serveur" > /var/rudder/cfengine-community/policy_server.dat
```

Cycle de vie - Installation



Cycle de vie - Acceptation

Envoi de l'inventaire généré au serveur



Serveur Rudder
Inscription dans la base

Acceptation du
noeud par
l'utilisateur

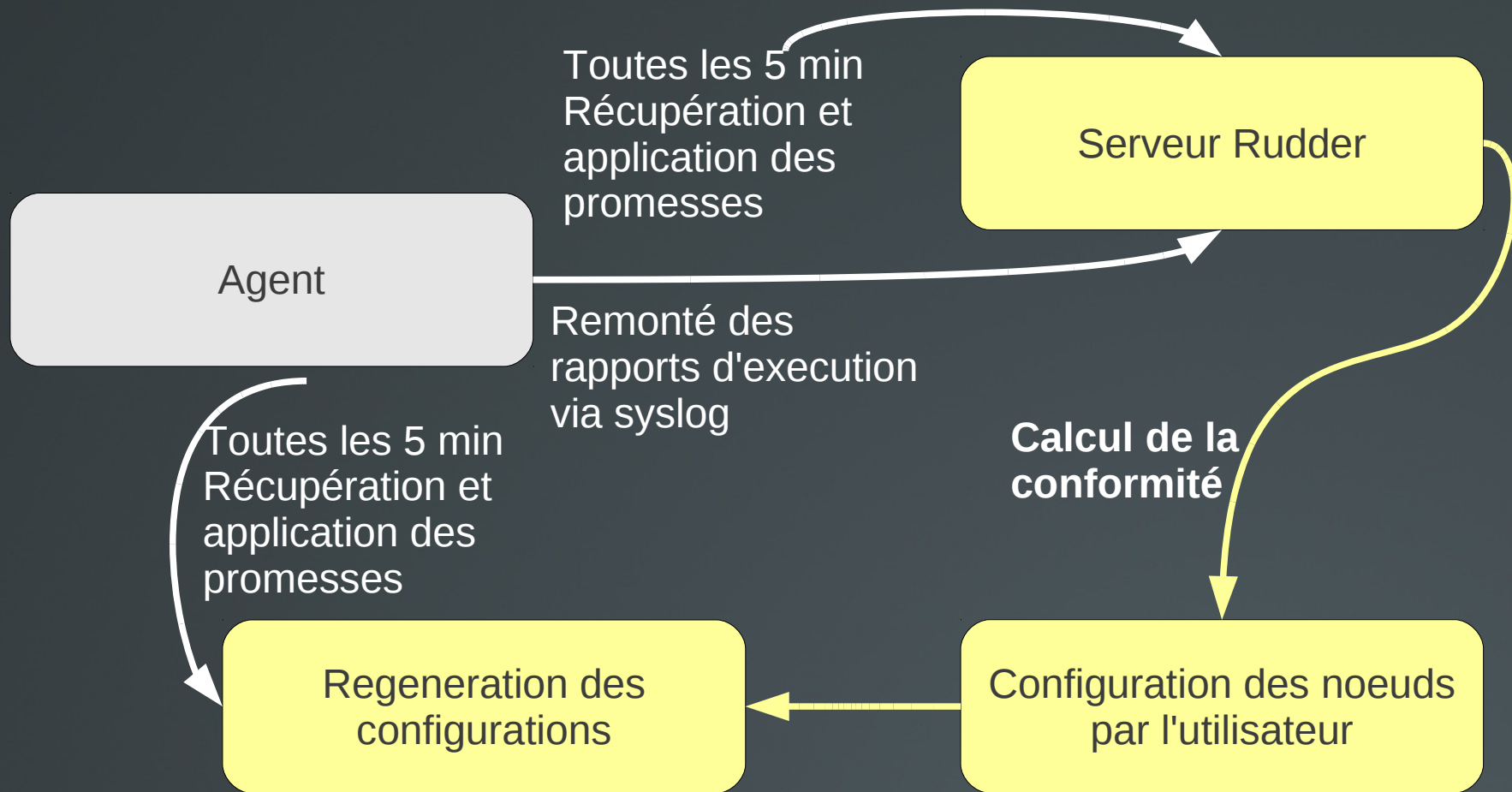
Génération des
promesses du noeud

Agent

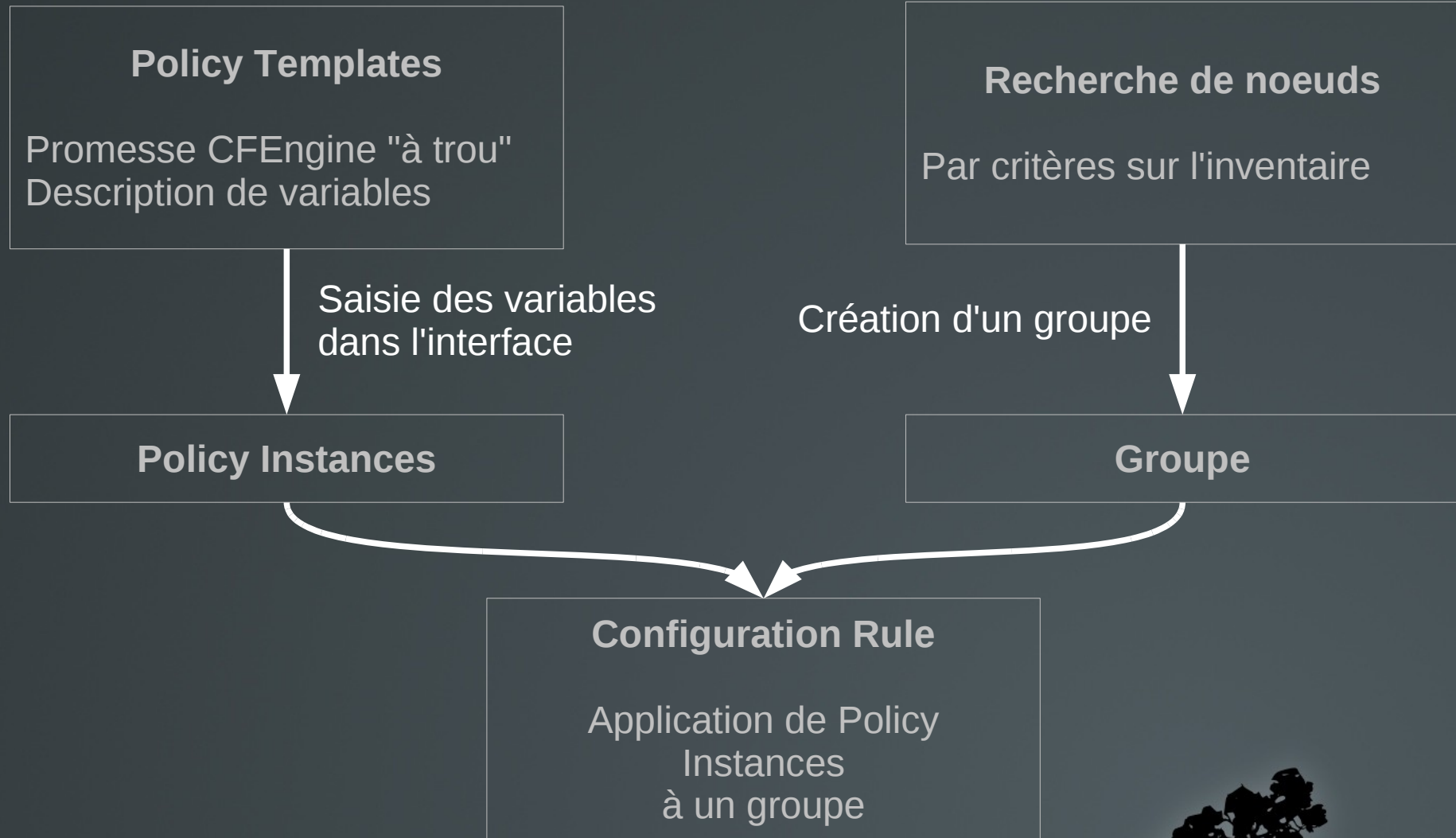
Toutes les 5 min
récupération des
promesses



Cycle de vie – Régime de croisière



Rudder : configurer des noeuds



Tracabilité

- Chaque modification est tracée
 - Toutes les actions utilisateurs
 - Login/Logout
 - Modification de configuration, groupe, ...
 - Déploiement manuel
 - Toutes les actions automatiques
 - Changement d'inventaire de noeud
 - Changement de politiques appliquées
 - Déploiement automatique
- Les diffs de toutes les modifications sont disponibles





Rudder : étendre les fonctionnalités



Roadmap

- 2.4 : Fin Janvier
 - Import/Export des Policy Instances, Groupes et Configuration Rule configurations entre serveurs Rudder
 - Rajout/améliorations de Policy Templates
 - Suppression de noeuds
 - Approbation des modifications avant déploiement
- 2.5 : Mi-2012
 - Amélioration du formulaire de Policy Instances
 - Amélioration du reporting
 - API d'exposition de données



Plugin

- L'application web peut être étendue par plugin
 - Découverte automatique au démarrage
- Un exemple d'implémentation open-source
 - <https://github.com/Normation/rudder-plugin-helloworld>
- Des plugins propriétaires payants
 - Rapports PDF
 - Support complet de Windows



Contribuer

- Sources en ligne sur GitHub
- Documentation sur le wiki
 - <http://rudder-project.org>
- Communauté open source naissante
 - Mailing lists
 - rudder-users@lists.rudder-project.org
 - rudder-dev@lists.rudder-project.org
 - IRC : #rudder sur FreeNode



Policy Templates

- Deux parties

Templates .st

Templates contenant du code
CFEngine
Utilise StringTemplate pour
remplacer les variables

Policy.xml

Descripteur de la Policy Template
Nom, Description, Compatibilité
(OS, ..)
Variables : nom, type, valeur par
defaut
Templates presents, et leur
destination



Policy Templates

- policy.xml - Informations

Policy.xml

```
<POLICY name="MOTD configuration">
  <DESCRIPTION>This policy template will check if the required Message Of
  The Day is present on the system.</DESCRIPTION>
  <COMPATIBLE>
    <OS version=">= 4 (Etch)">Debian</OS>
    <OS version=">= 4 (Nahant)">RHEL / CentOS</OS>
    <OS version=">= 10 SP1 (Agama Lizard)">SuSE LES / DES / OpenSuSE</OS>
    <AGENT version=">= 3.1.5">cfengine-community</AGENT>
  </COMPATIBLE>

  <MULTIINSTANCE>>false</MULTIINSTANCE>
```



Policy Templates

- policy.xml – Informations pour CFEngine

Policy.xml

```
<BUNDLES>
  <NAME>check_motd_configuration</NAME>
</BUNDLES>

<TMLS>
  <TML name="motdConfiguration"/>
</TMLS>
```



Policy Templates

- policy.xml – Variables pour Rudder

Policy.xml

```
<SECTIONS>
  <SECTION name="MOTD entry" multivalued="false">
    <INPUT>
      <NAME>MOTD</NAME>
      <DESCRIPTION>Message of the day (MOTD) to display</DESCRIPTION>
      <CONSTRAINT>
        <TYPE>textarea</TYPE>
      </CONSTRAINT>
    </INPUT>
    <SELECT1>
      <NAME>MOTD_EMPTY</NAME>
      <DESCRIPTION>Enforce this MOTD only</DESCRIPTION>
      <LONGDESCRIPTION>This option will remove any existing messages in the motd file and
replace them with the text provided above</LONGDESCRIPTION>
      <ITEM>
        <LABEL>Yes</LABEL><VALUE>>true</VALUE>
      </ITEM>
      <ITEM>
        <LABEL>No</LABEL><VALUE>>false</VALUE>
      </ITEM>
      <CONSTRAINT>
        <DEFAULT>>false</DEFAULT>
      </CONSTRAINT>
    </SELECT1>
  </SECTION>
</SECTIONS>
```

Policy Templates

- motdConfiguration.st - variable

motdConfiguration.st

```
bundle agent check_motd_configuration {
  classes:
    "motd_absent" not => fileexists("/etc/motd");

  vars:
    "motd" string => "&MOTD&";
    "motd_empty" string => "&MOTD_EMPTY&";
    "motd_uuid" string => "&TRACKINGKEY&";

  debian::
    "motd_file" slist => { "/etc/motd", "/etc/motd.tail" };

  !debian::
    "motd_file" slist => { "/etc/motd" };
}
```

Policy Templates

- motdConfiguration.st - actions

motdConfiguration.st

files:

!windows::

 "\$(motd_file)"

 create => "true",
 edit_defaults => rudder_empty_select("\$

(motd_empty)"),

 perms => mog("644", "root", "root"),
 edit_line => insert_lines("\$(motd)"),
 classes => kept_if_else("motd_file_kept",

"motd_file_edited", "motd_file_failed");



Policy Templates

- motdConfiguration.st - reporting

motdConfiguration.st

reports:

```
motd_file_kept.!motd_file_edited::
```

```
    "@@motdConfiguration@@result_success@@$(motd_uid)
@@motdConfiguration@@None@@$(g.execRun)##$(g.uid)@#The MOTD is in
conformance with the policy";
```

```
motd_file_edited::
```

```
    "@@motdConfiguration@@result_repaired@@$(motd_uid)
@@motdConfiguration@@None@@$(g.execRun)##$(g.uid)@#The MOTD was
successfully edited to match the policy";
```

```
motd_file_failed::
```

```
    "@@motdConfiguration@@result_error@@$(motd_uid)
@@motdConfiguration@@None@@$(g.execRun)##$(g.uid)@#Could not edit the
MOTD";
```

```
!windows.motd_absent::
```

```
    "@@motdConfiguration@@log_repaired@@$(motd_uid)
@@motdConfiguration@@None@@$(g.execRun)##$(g.uid)@#The MOTD file was
absent, I will create it";
```



Merci de votre attention !

Restez en contact...

Normation 

The logo for Normation consists of a cluster of yellow, hexagonal shapes arranged in a roughly triangular pattern.

Nicolas Charles

Mail: nch@normation.com

Twitter: [nico_charles](https://twitter.com/nico_charles)

