



---

# Interfacing infrastructure as code with non-expert users

---

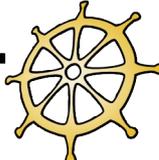
**Jonathan CLARKE** – [jcl@normation.com](mailto:jcl@normation.com)  
[@jooooooooon42](https://twitter.com/jooooooooon42) (that's 7 'o's)



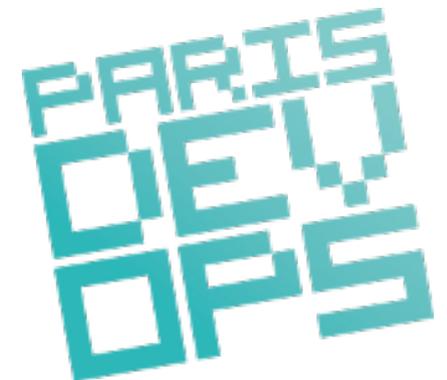
# Jonathan Clarke

**Origins:** Sysadmin, infrastructure management

**Work at** Normation 

**Creator of** Rudder  and ncf.io

**Co-organizer of events:**



# Automated configuration rocks!



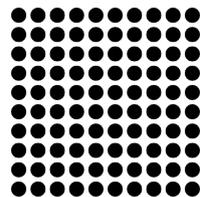
## Save time

Deploy faster & be more responsive to changes



## Improve reliability

Avoid manual errors, harmonize configurations



## Scalable

Manage 1 to > 100000 servers the same way

# Automated configuration

The proper way to manage systems



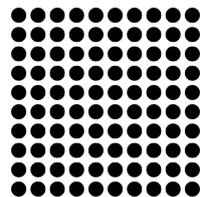
## Save time

Deploy faster & be more responsive to changes



## Improve reliability

Avoid manual errors, harmonize configurations



## Scalable

Manage 1 to > 100000 servers the same way

## “Typical” IT team



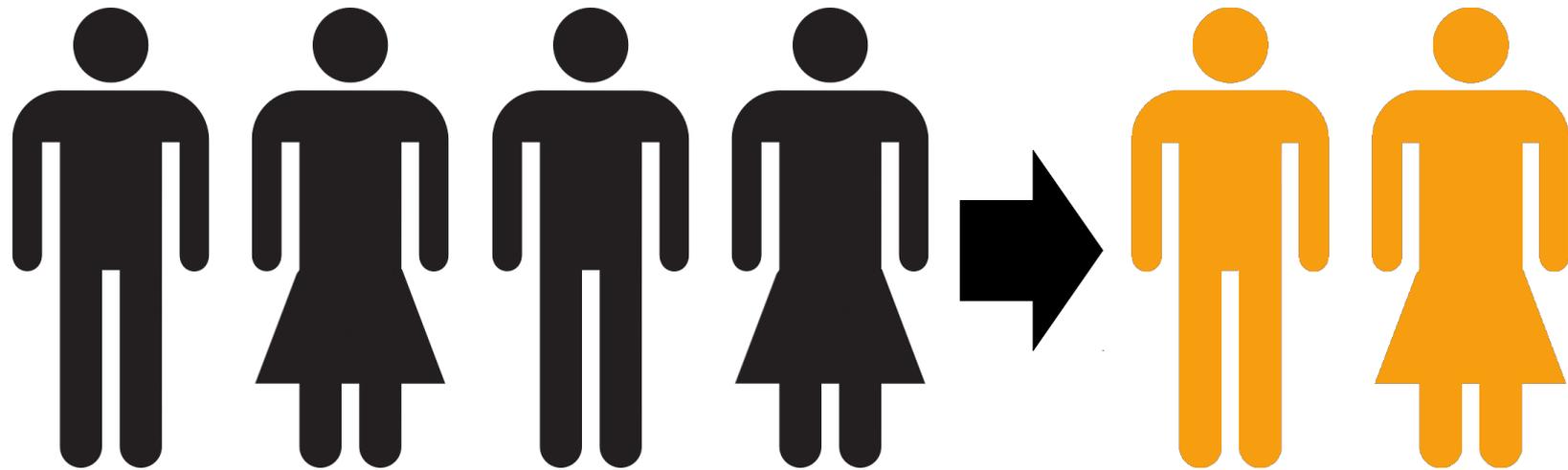
Develop applications and deploy them,  
manage IT operations, security, etc etc etc ...

# “Typical” IT team adopts CM



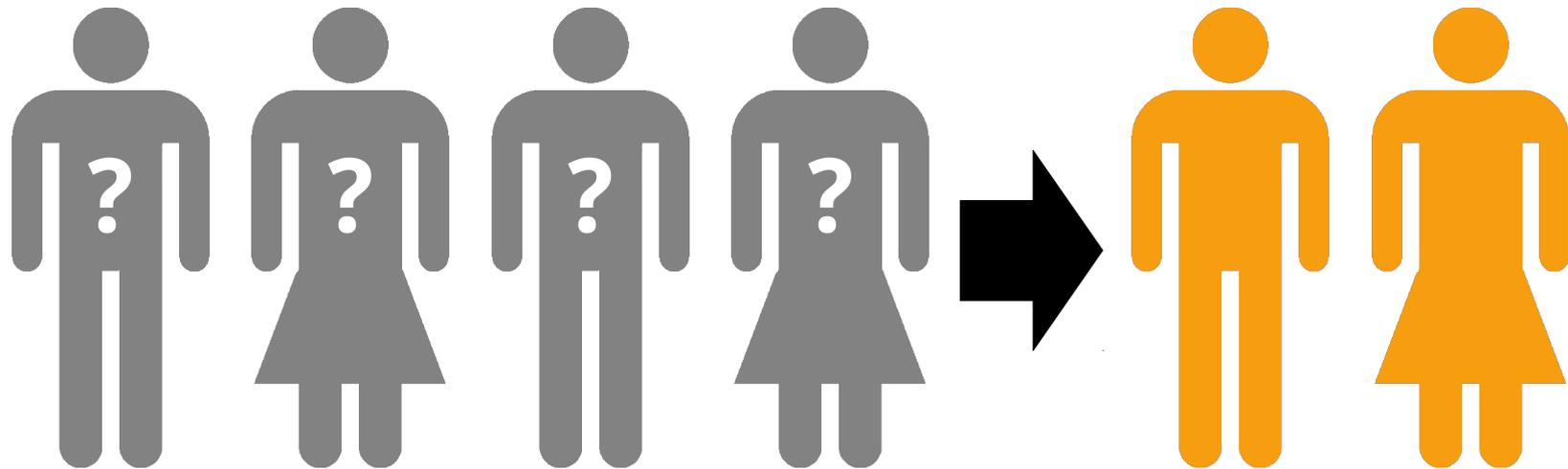
A minority of the team will learn a tool and it's language. Let's call them “the experts”.

# What about the others?



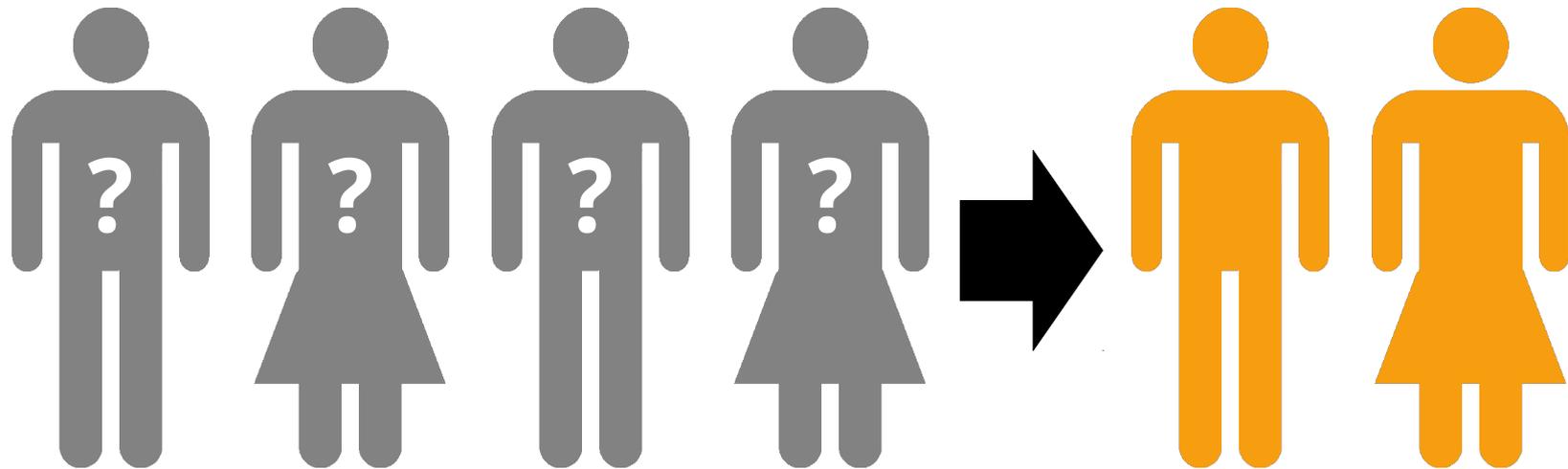
Changes, deployments, new features...  
all end up **requiring** <your chosen CM tool>,  
therefore the “experts”.

# What about the others?



They have a choice: learn or get left behind.

# Does this look good?



Are we actually standing in the way of automated configuration by pushing ahead?

# What are we asking users to do?

## BEFORE

```
ssh box.company.com
rpm -i software.rpm
vi /etc/software.conf
service software restart
```

## AFTER

```
package { 'apache' :
  ensure => present,
  name    => $::operatingsystem ? {
    /( ?i:Ubuntu|Debian|Mint)/ => 'apache2',
    default                    => 'httpd',
  }
  (or similar!)
}
```

and more, and git/svn, etc...

# Getting everyone on board for infrastructure as code is hard



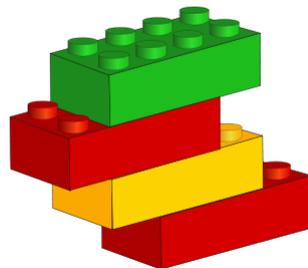
## Steep learning curve

New concepts, non obvious syntaxes, paradigm, ...



## Lack of motivation

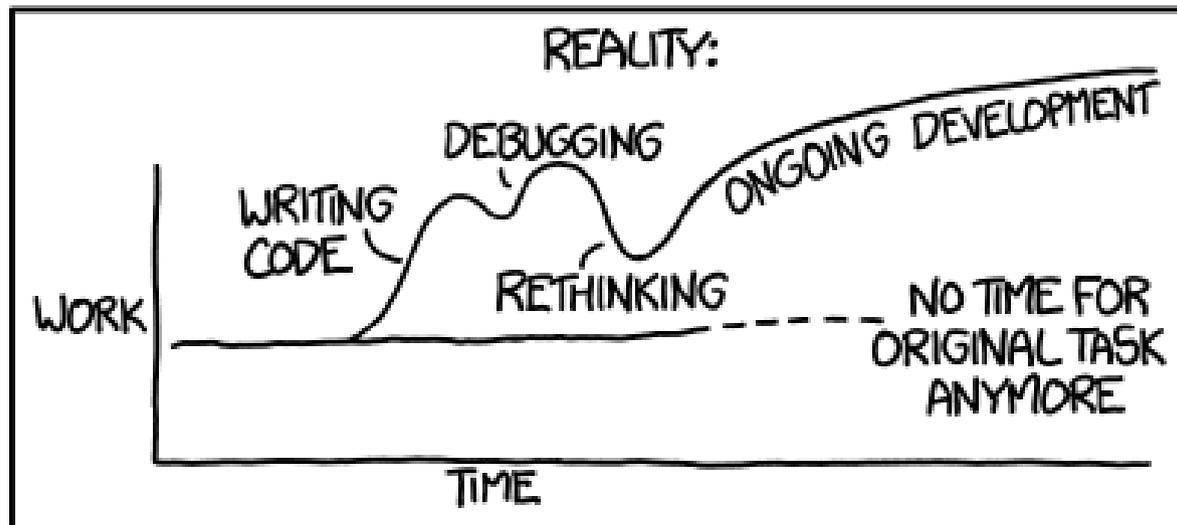
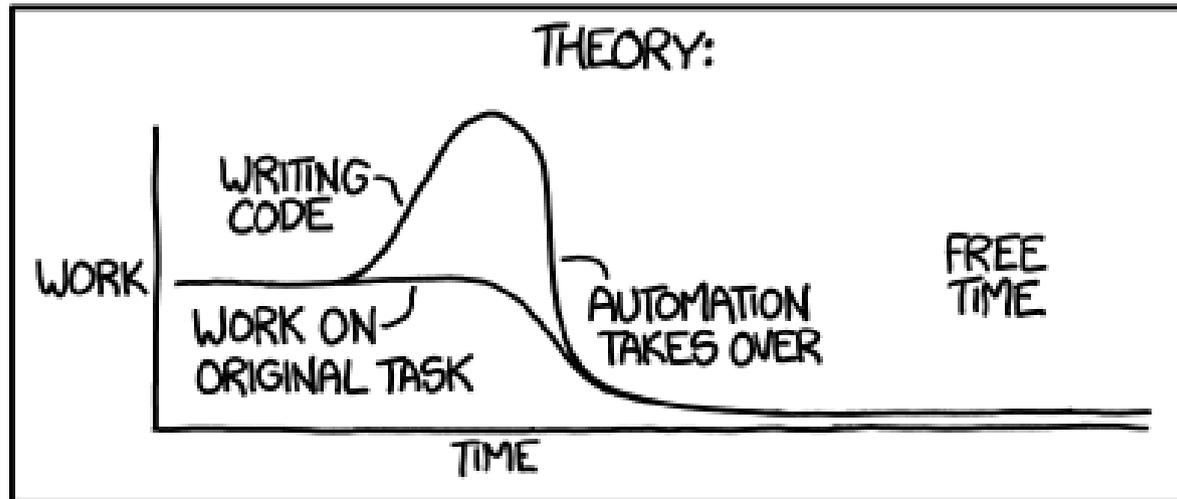
"What do I have to gain from using this tool?"



## Frustration

"I can do it quicker by hand or with a shell script"

"I SPEND A LOT OF TIME ON THIS TASK.  
I SHOULD WRITE A PROGRAM AUTOMATING IT!"





Example of getting non-experts to use complex stuff

# Let's make it easier for everyone to use CM!



## Steep learning curve

New concepts, non obvious syntaxes, paradigm, ...

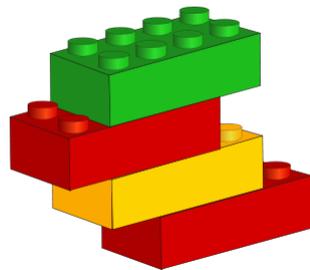
→ Provide simpler interfaces!



## Lack of motivation

"What do I have to gain from using this tool?"

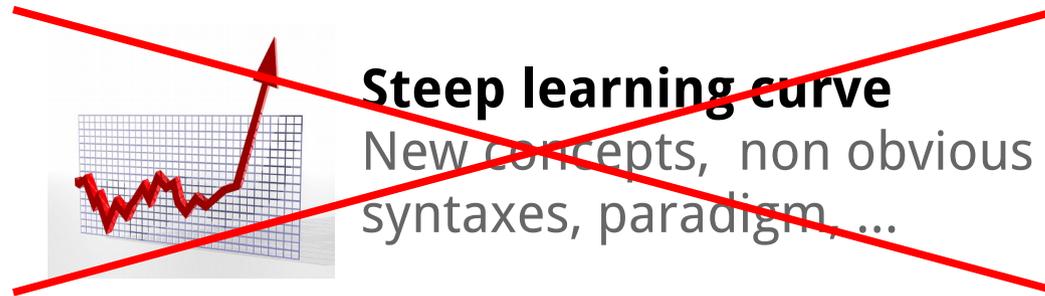
→ Show the benefits!



## Frustration

"I can do it quicker by hand or with a shell script"

→ Enable quick wins!



1) Separate **content** and **controls**

2) Provide access to key parameters without hacking {CFEngine,Puppet,Chef,etc} code



## ~~Lack of motivation~~

~~"What do I have to gain from using this tool?"~~

1) Show the benefits to all users

2) Provide nice **reports** showing what works, how many machines are impacted



- 1) Make it **easy** and **quick** to achieve success
- 2) Provide ready-to-use configuration techniques and share in-house ones simply



# Idea for IT team and CM stack



# Idea for IT team and CM stack



# An example based on ncf and Rudder

ncf.io

Rudder 

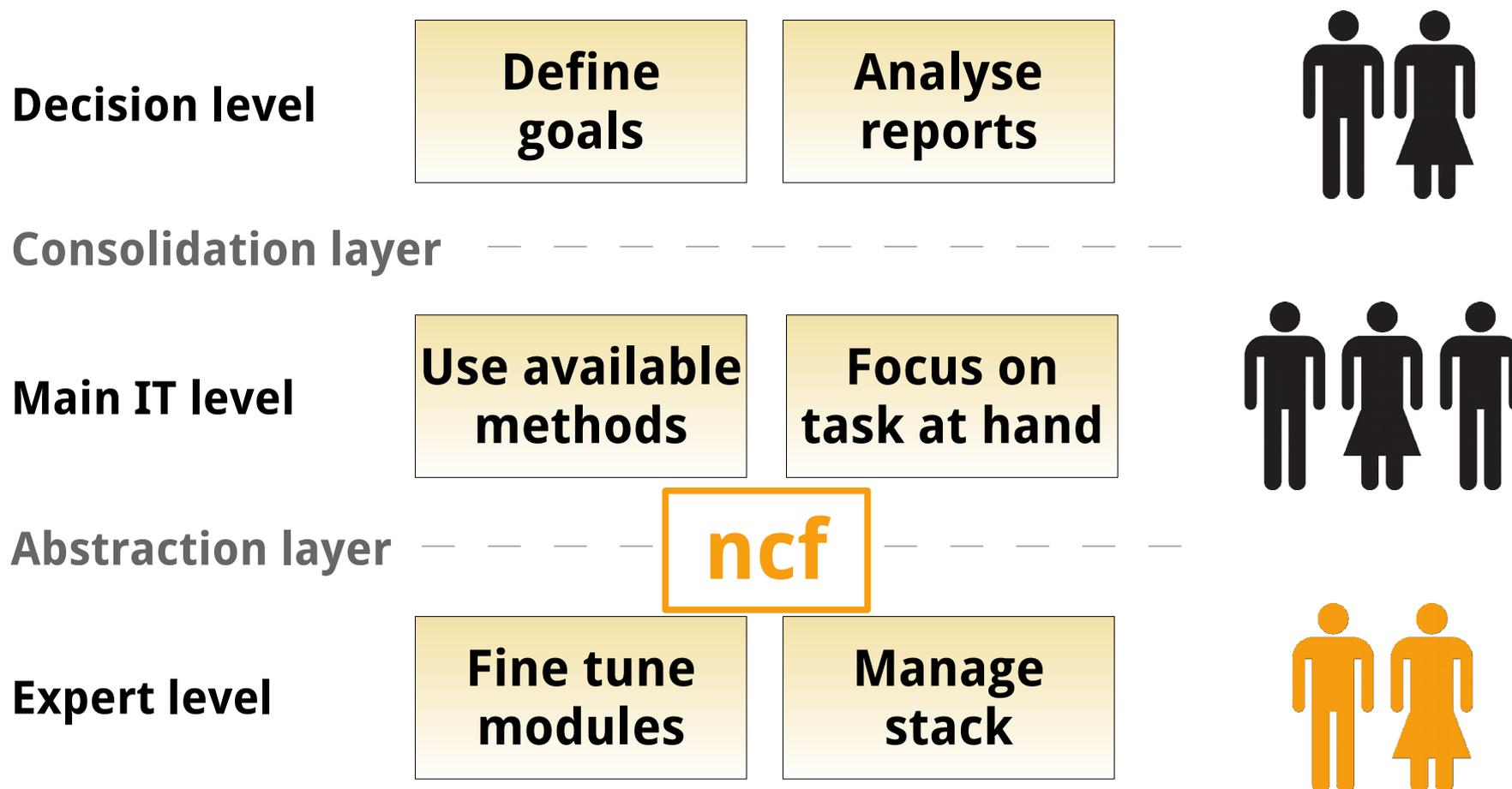
[www.rudder.cm](http://www.rudder.cm)

Open source - GPLv3

## ncf

ncf is a **framework** that runs in **pure CFEngine** language, to help **structure** CFEngine policy and provide **reusable, single purpose** components distributed under the GPLv3 license.

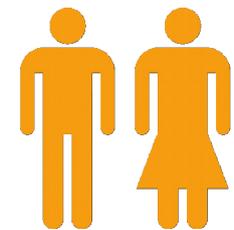
# Idea for IT team and CM stack



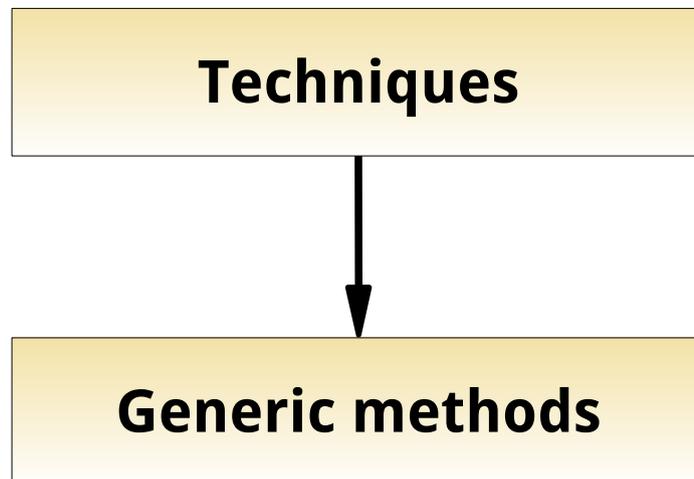
# A layered approach

**Generic methods**

Unit tasks: “Copy file”,  
“Install package”, ...  
**Implement the “how”**



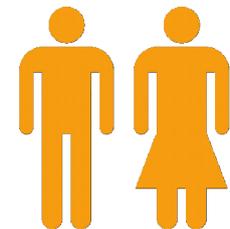
# A layered approach



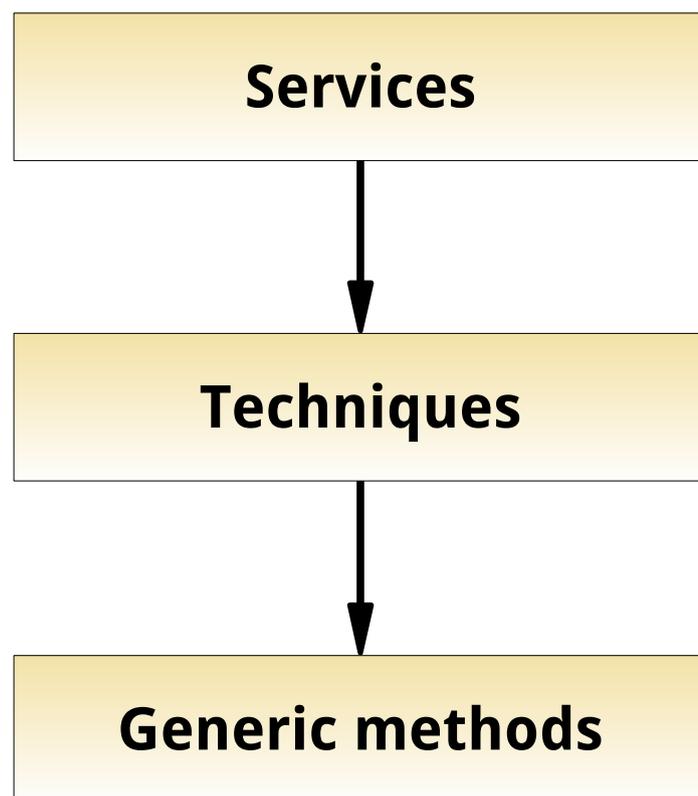
Use the provided methods, keep focus  
**Supply "data"**



Unit tasks: "Copy file",  
"Install package", ...  
**Implement the "how"**



# A layered approach



Get an overview  
Benefit from CM



Use the provided  
methods, keep focus  
**Supply "data"**



Unit tasks: "Copy file",  
"Install package", ...  
**Implement the "how"**



# Example === 1000 words

## With ncf:

```
bundle agent ntp {
  methods:
    "package" usebundle => package_install("ntpd");
    "config" usebundle => file_from_template("ntp.conf", "/etc/ntp.conf");
    "reload" usebundle => service_restart("ntp"),
      ifvarclass => "file_from_template__etc_ntp_conf_repaired";
    "running" usebundle => service_ensure_running("ntp");
}
```

# Simple web interface – ncf builder

ncf builder Create and modify your own ncf technique

LIBRARY Techniques

Filter New

- Bla Technique for evaluation of parsingness >
- Message Of The Day (MOTD) >
- NTP <**
- OpenSSH server >

TECHNIQUE NTP

General information

Name: NTP

Description: Install, configure and ensure ntpd is running. Uses a template file for configuration.

Version: 0.1

Package install *i* + - >

ntp  
Install or update a package in its latest version available

File from template *i* + + - >

/etc/ntp.conf  
This is a bundle to build a file from a template

Service restart *i* + - > <

ntp  
Restart a service using the appropriate method

Service ensure running *i* + - >

+ Add methods Delete Reset Save

METHOD Service restart Reset

Conditions

CFEngine class context:

file\_from\_template\_\_etc\_ntp\_conf\_repaired

Parameters

Name of the service to restart in /etc/init.d:

ntp

Result classes defined by this method

Success: service\_restart\_ntp\_kept

Repaired: service\_restart\_ntp\_repaired

Error: service\_restart\_ntp\_error

# Rudder

Rudder **sets up** an environment with sane defaults to **automate configuration**, building on **ncf** methods **without any code** and including **automatic feedback**.

# Simplified configuration

Information Parameters Target Rules

## Parameters

▼ Section: General

Which SSH configuration file should be edited?:

---

▼ Section: Service settings

Address family used by sshd:

Do you want to define the Ports ? :

▼ SSH ports #1

Port number that SSH listens on:

Supported protocol version(s):

Maximum open sessions per network connection:

---

▼ Section: Authentication settings

Allow challenge-response authentication (e.g. via PAM):  Yes  No  Don't change

Allow password authentication:  Yes  No  Don't change

Allow public key authentication:  Yes  No  Don't change

Allow empty passwords for password authentication:  Yes  No  Don't change

Allow root to login using SSH:

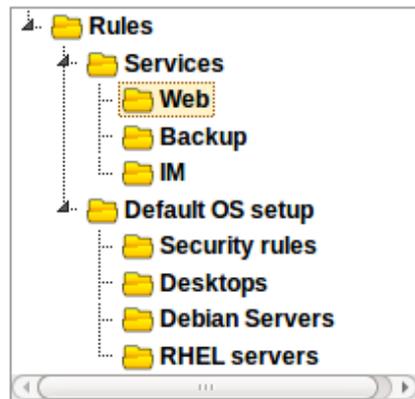
Maximum authentication attempts per connection:

Time before disconnect if not loaded in (seconds):

# Built-in reporting

## CATEGORIES

New Category



## RULES » Services » Web

New Rule

Display Rules from subcategories

Show 10 entries

Search

Name	Category	Status	Compliance	
<a href="#">Web database</a>	Services » Web	In application	0%	Edit
<a href="#">Web front</a>	Services » Web	In application	100%	Edit

Showing 1 to 2 of 2 entries (filtered from 5 total entries)

First Previous 1 Next Last

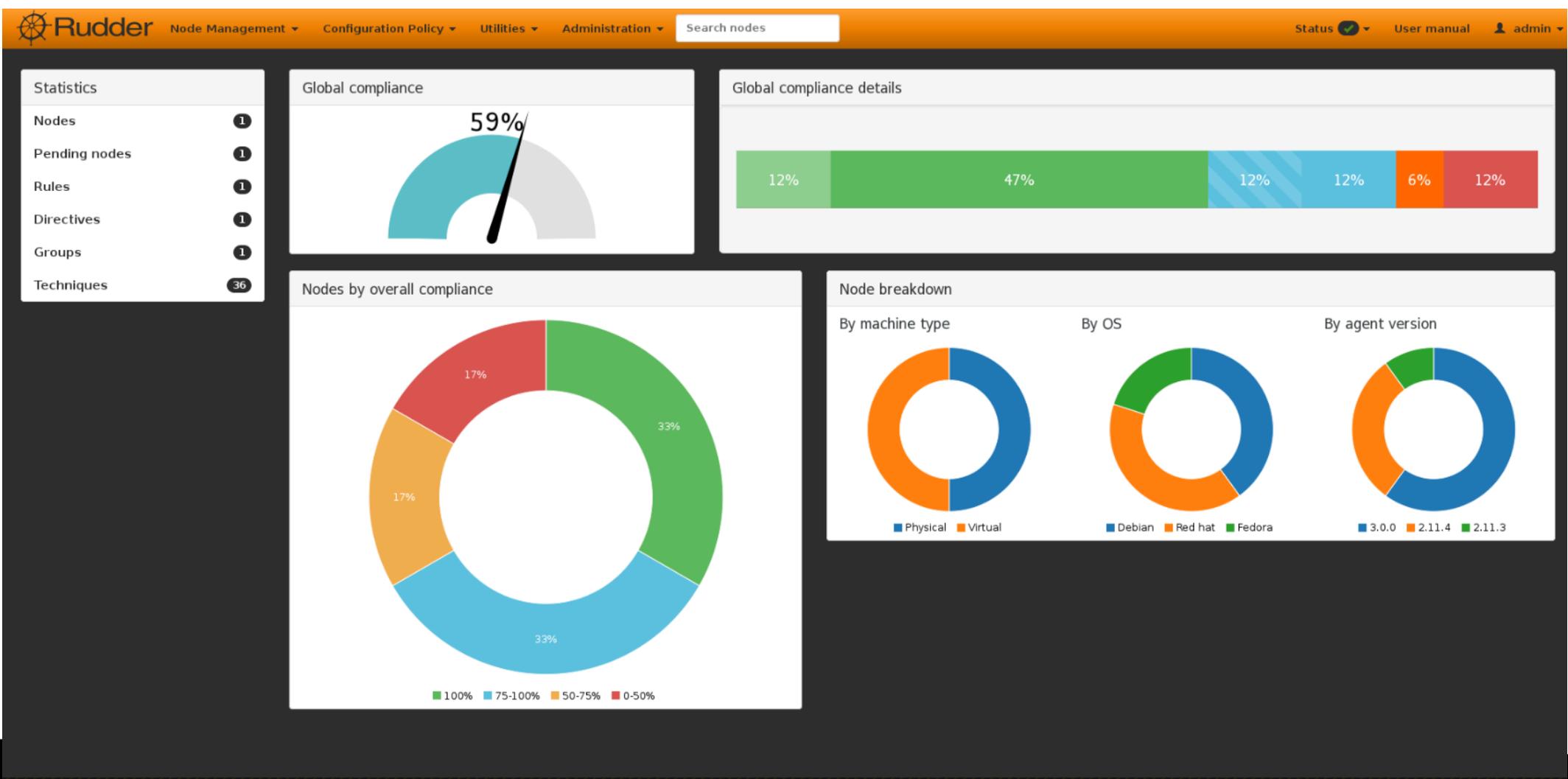
# Built-in reporting

Directive	Status	Compliance
▼ <b>Configure database</b> ⓘ ⚙	Repaired	0%
Component	Status	Compliance
▶ File	Repaired	0%
▶ Post-modification hook	Success	100%
▶ Line deletion regular expressions	Success	100%
▶ Permission adjustment	Success	100%
▶ Line replacement regular expressions	Success	100%
▶ Enforce content by section	Success	100%
▶ <b>Database</b> ⓘ ⚙	Success	100%

Showing 1 to 2 of 2 entries

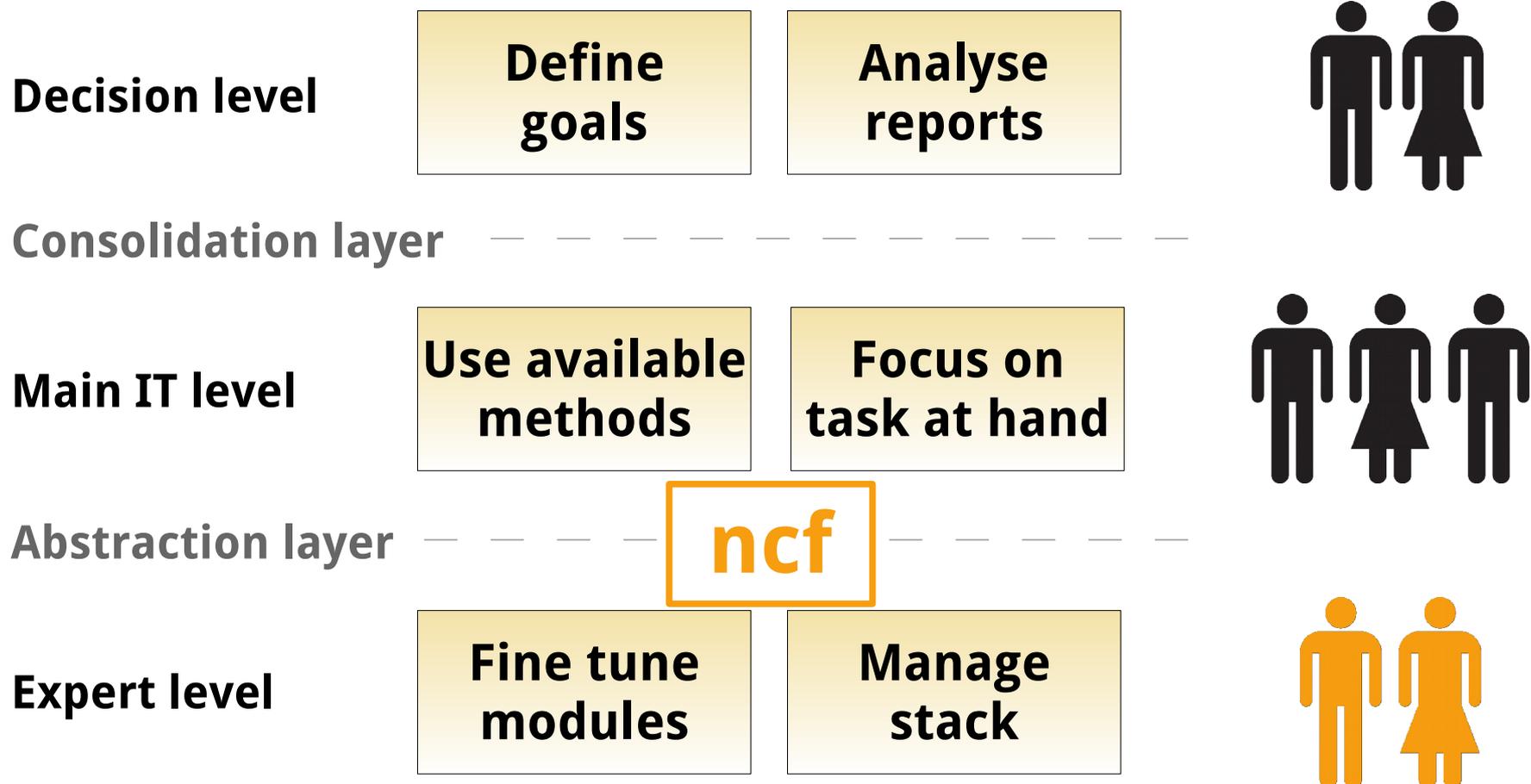
First Previous 1 Next Last

# Dashboard overview

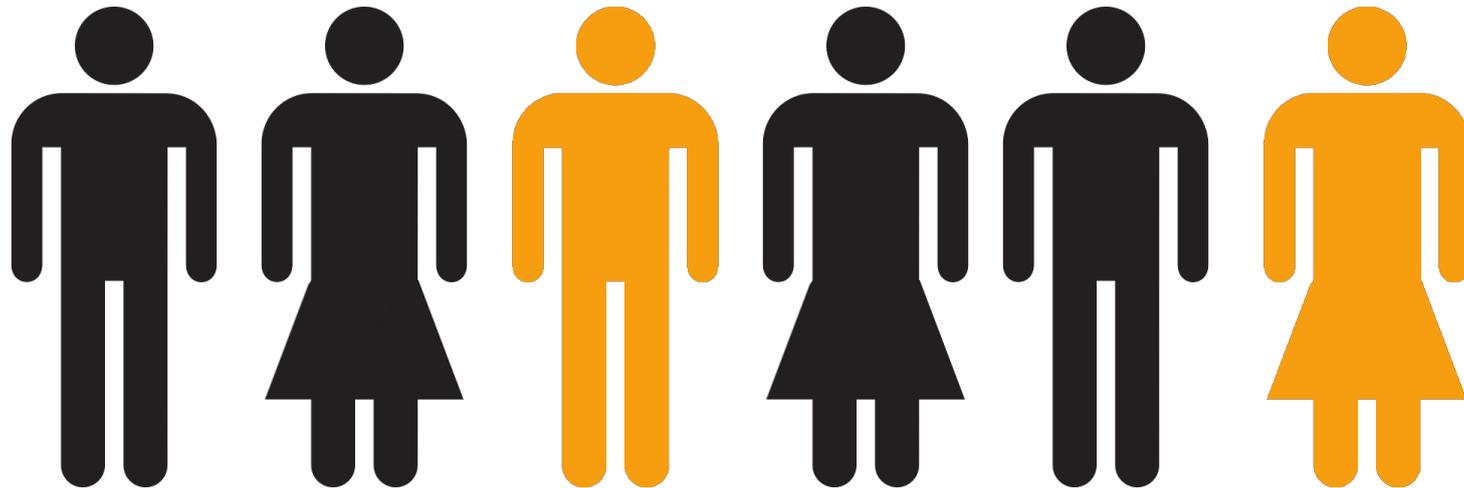


# Idea for IT team and CM stack

R  
u  
d  
d  
e  
r



# Next IT team?





---

# Questions?

---

Check it out on:  
<http://www.ncf.io/>

Check it out on:  
<http://rudder.cm/>

**Jonathan CLARKE** – [jcl@normation.com](mailto:jcl@normation.com)  
**@jooooooooon42** (that's 7 'o's)

