Deploying Foreman in Enterprise Environments 2.0

best practices

and lessons learned…

Nils Domrose

Cologne, August, 4 2014

About me

inovex

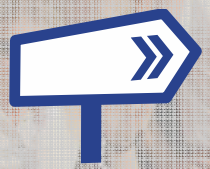
- senior linux systems engineer at inovex GmbH
- worked as a network engineer, software developer and systems engineer
- using foreman for about 1 year
- using bare-metal deployment for ages
- life is short – let's focus on interesting stuff!

About inovex

We use technology to make our customers happy. And ourselves.

inovex

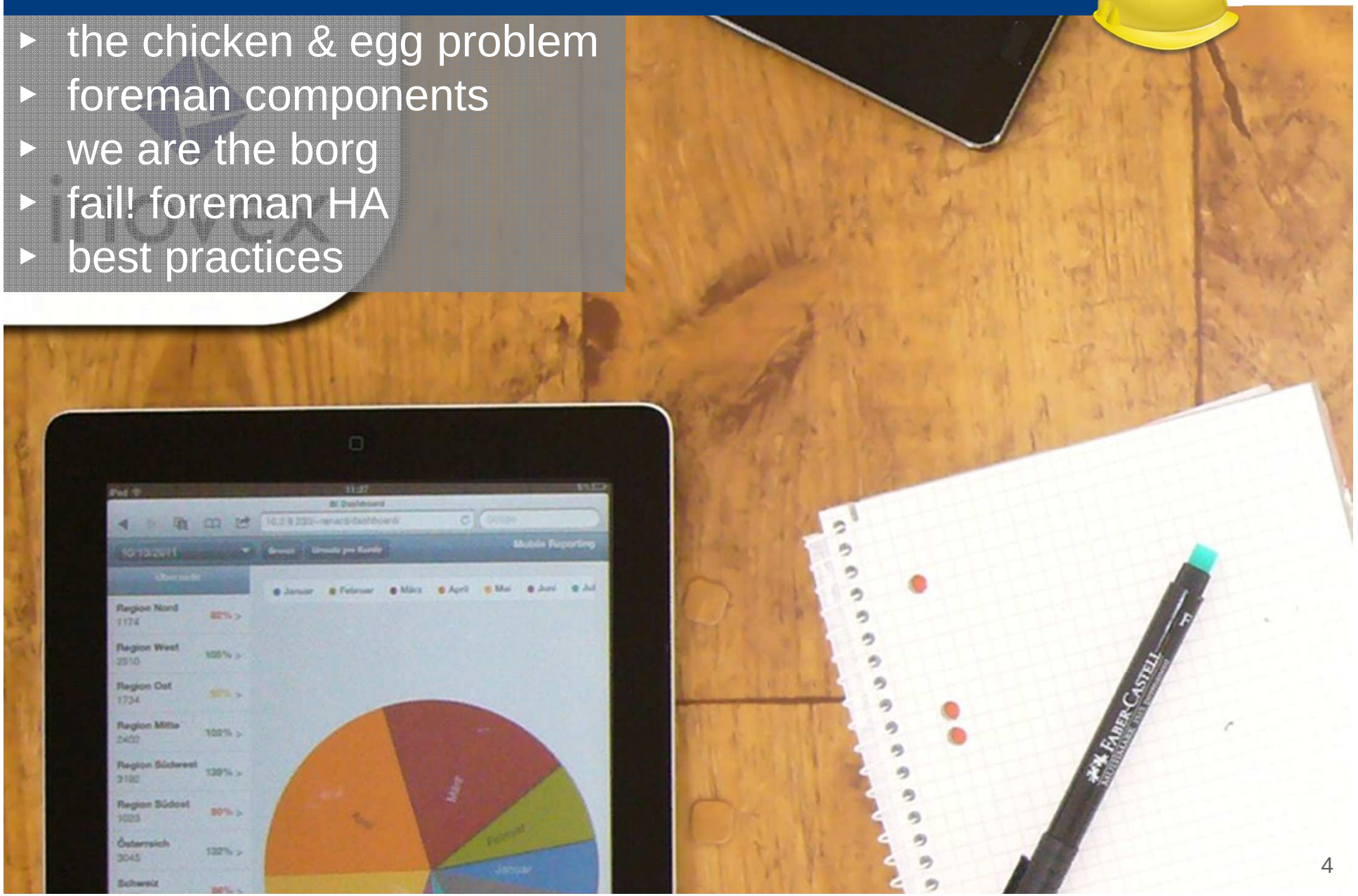
- http://www.inovex.de
- offices in [‘Pforzheim’, ‘Karlsruhe’, ‘Cologne’, ‘Munich’]
- we have open positions...



1	2	3	4	5
SHAPE Consulting	BUILD Application Development	RUN IT Engineering & Operations	TRACK Business Intelligence	TEACH Academy

Agenda

- the chicken & egg problem
- foreman components
- we are the borg
- fail! foreman HA
- best practices

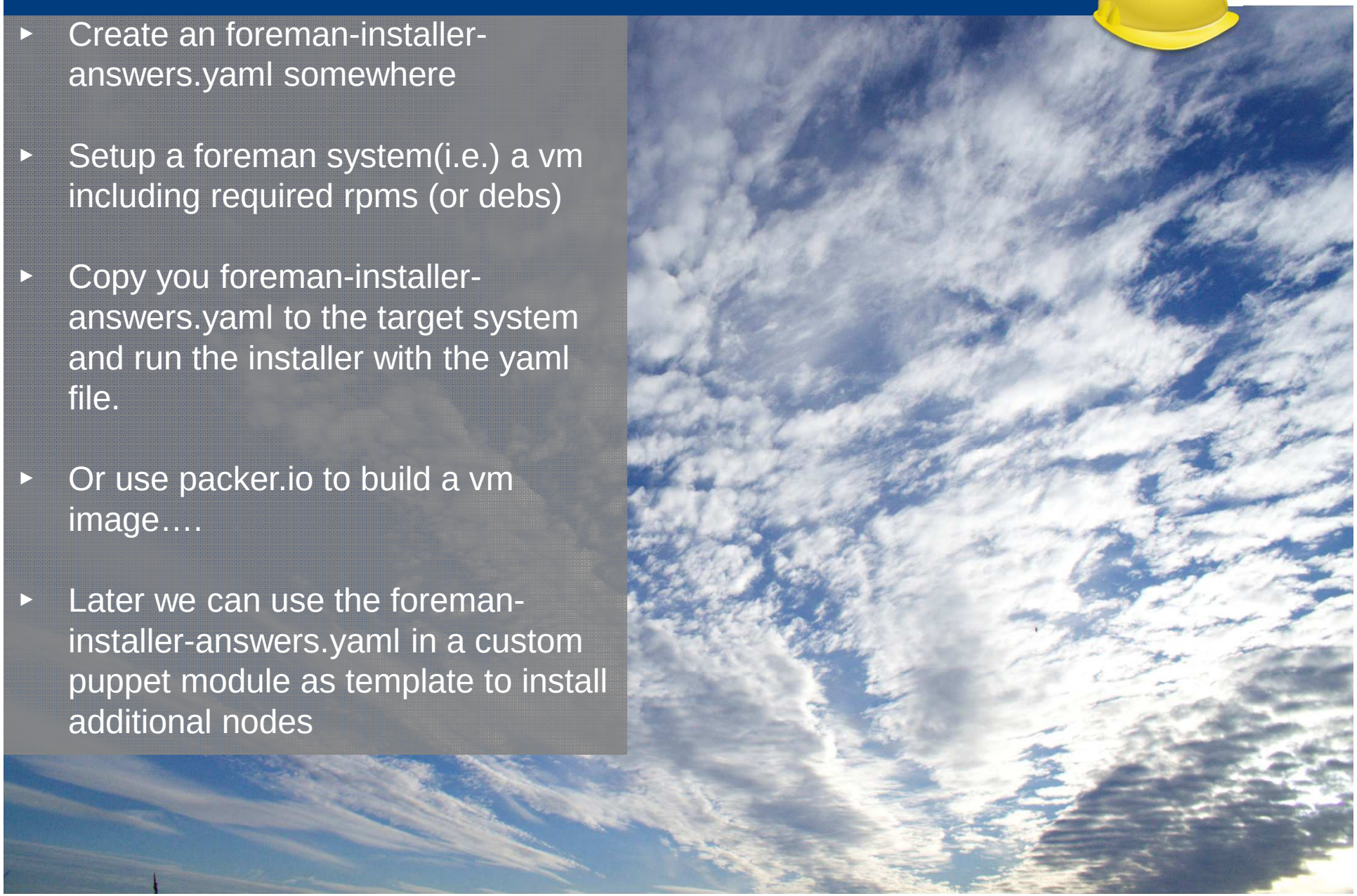


The Chicken & Egg Problem

...deploying your deployment infrastructure

inovex

- Create an foreman-installer-answers.yaml somewhere
- Setup a foreman system(i.e.) a vm including required rpms (or debs)
- Copy you foreman-installer-answers.yaml to the target system and run the installer with the yaml file.
- Or use packer.io to build a vm image….
- Later we can use the foreman-installer-answers.yaml in a custom puppet module as template to install additional nodes



The Chicken & Egg Problem

...deploying your deployment infrastructure

inovex

```
$foreman_answersfile = '/etc/foreman/foreman-installer-answers.yaml'

file {$foreman_answersfile:
  owner   => 'root',
  group   => 'root',
  mode    => '0600',
  content => template('foreman/foreman-installer-answers.yaml.erb'),
}

exec {'foreman-installer':
  command     => '/usr/bin/foreman-installer –d'
  logoutput   => on_failure,
  require     => File[$foreman_answersfile],
  ...
```

The Chicken & Egg Problem

Or use kafo...

inovex

- kafo?
- kafo! - imagine master-less puppet plus config data in yaml (like hiera)
- can’t tell whether it’s the chicken or the egg or both but it’s cool ☺
- stores parameters for your modules in answers.yaml
- simply copy your modules into the installer’s modules directory
- or even use puppet-librarian to manage your modules (poor man’s git-submodules)

https://github.com/rodjek/librarian-puppet

https://github.com/theforeman/kafo

The Chicken & Egg Problem

kafo - getting started

inovex

- Create a new installer project:

```
`--> i=my-installer; mkdir $i; cd $i; kafofy -n $i

using ./config/my-installer.yaml as default config file
... creating config file ./config/my-installer.yaml
... creating bin/my-installer
Your directory was kafofied
Now you should:
  1. upload your puppet modules to modules directory (you can use librarian-puppet project)
  2. create default ./config/answers.yaml or modify ./config/my-installer.yaml to load another answer file
  3. run bin/my-installer to install your modules
```

The Chicken & Egg Problem

get the modules

inovex

- delete the modules directory

```
root@foreman:/var/tmp/my-installer# rm -Rf modules/
```

- create a Puppetfile

```
root@foreman:/var/tmp/my-installer# cat Puppetfile
forge "http://forge.puppetlabs.com"

mod 'puppetlabs/stdlib'

mod 'ntp',
  :git => 'git://github.com/puppetlabs/puppetlabs-ntp.git'
```

- run the librarian

```
root@foreman:/var/tmp/my-installer# librarian-puppet install --clean
root@foreman:/var/tmp/my-installer# ll modules
total 16
drwxr-xr-x 4 root root 4096 Jan 29 12:25 ./
drwxr-xr-x 7 root root 4096 Jan 29 12:17 ../
drwxr-xr-x 7 root root 4096 Jan 29 12:25 ntp/
drwxr-xr-x 6 root root 4096 Jan 29 12:24 stdlib/
```

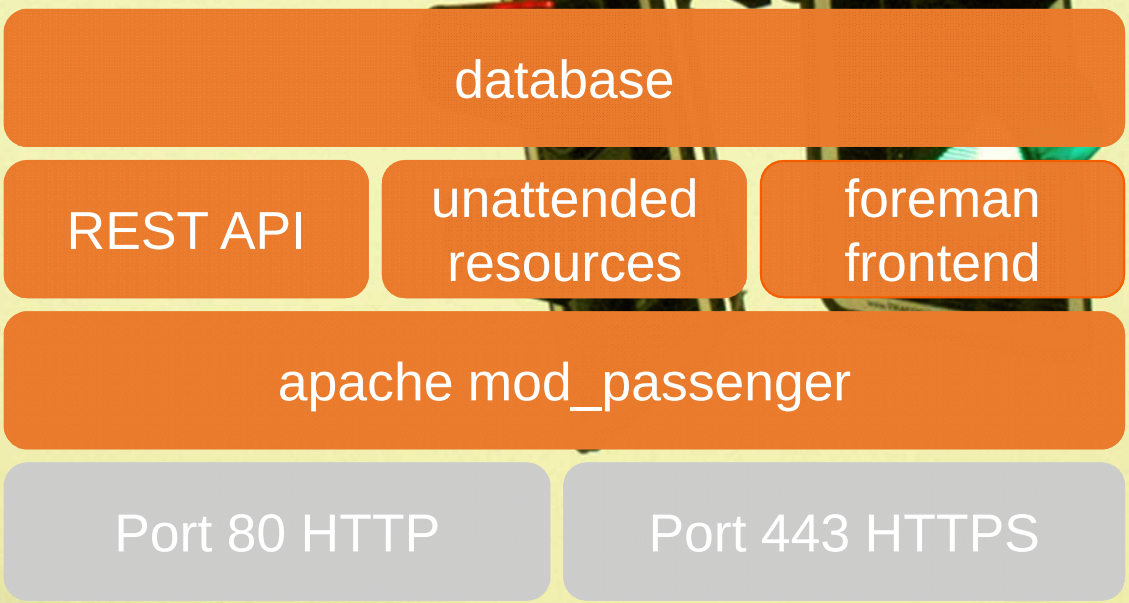
- next: package it, install it

Foreman Components

foreman

inovex

- rails application
- apache mod_passenger
- Webinterface
- REST API
- unattended resources – rendered templates

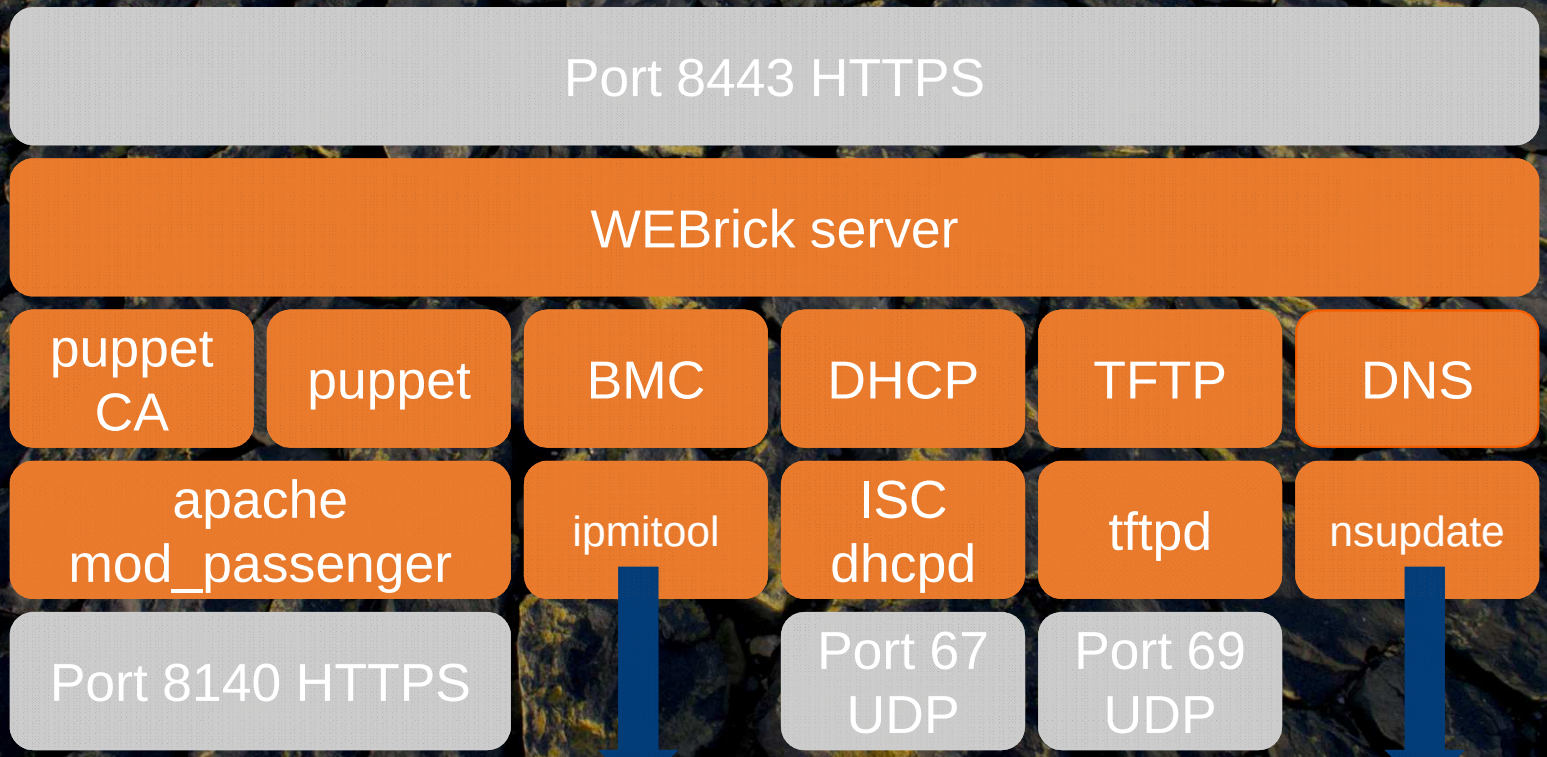


Foreman Components

smart-proxy

inovex

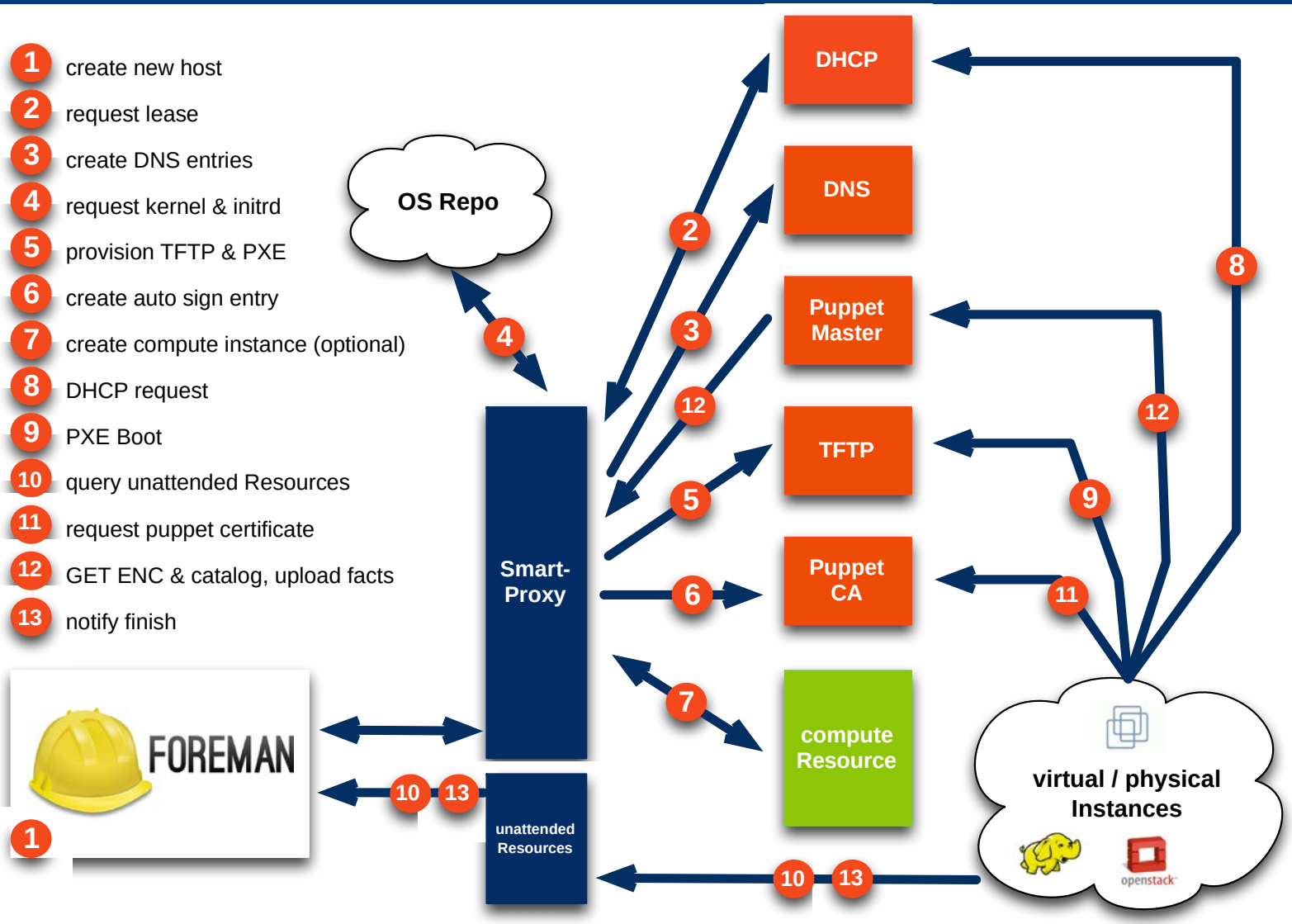
- WEBrick based REST server to manage supporting infrastructure



Installation process using the foreman

...let‘s deploy some boxes

inovex

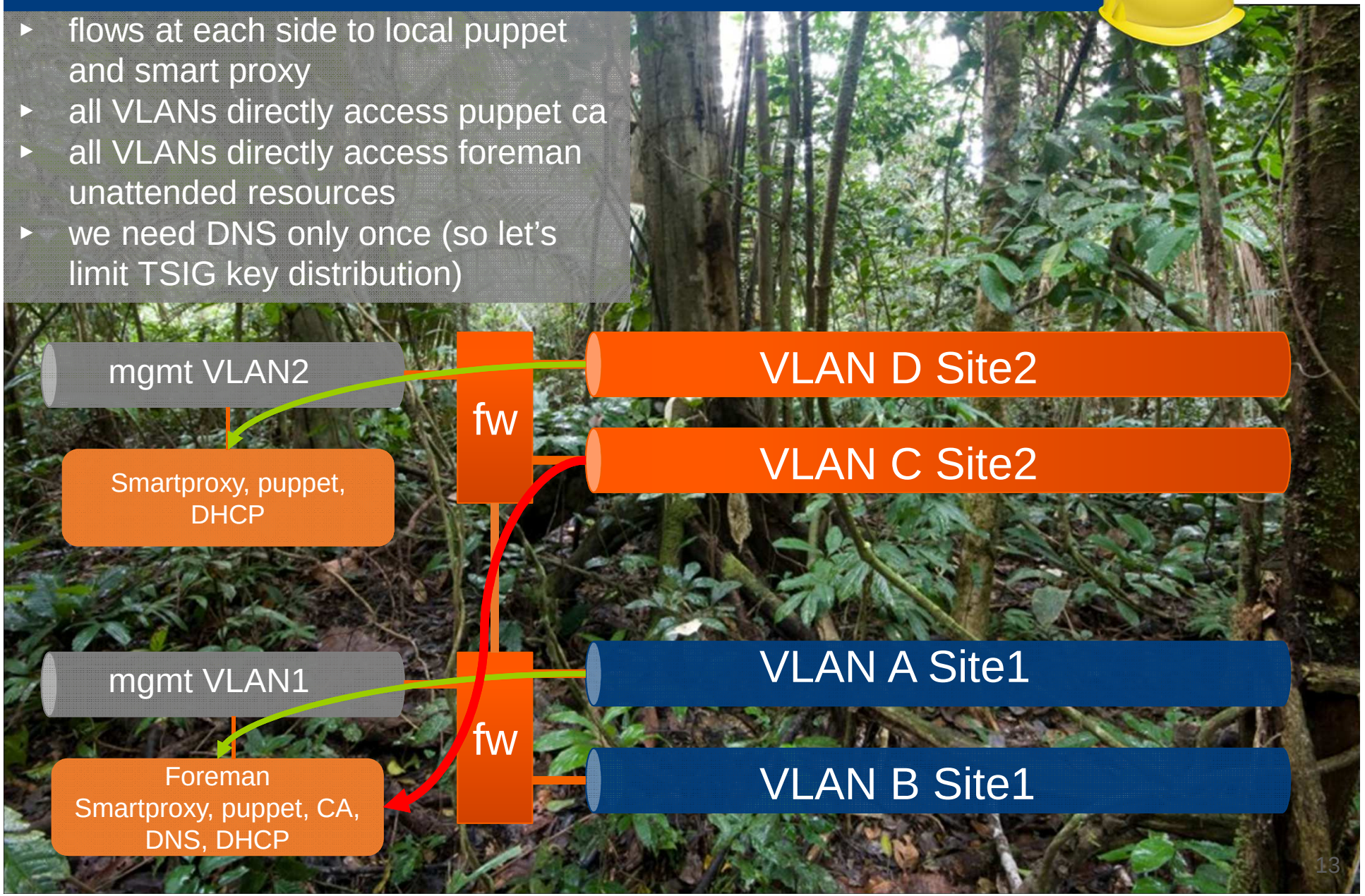


Foreman Components

placement of foreman components

inovex

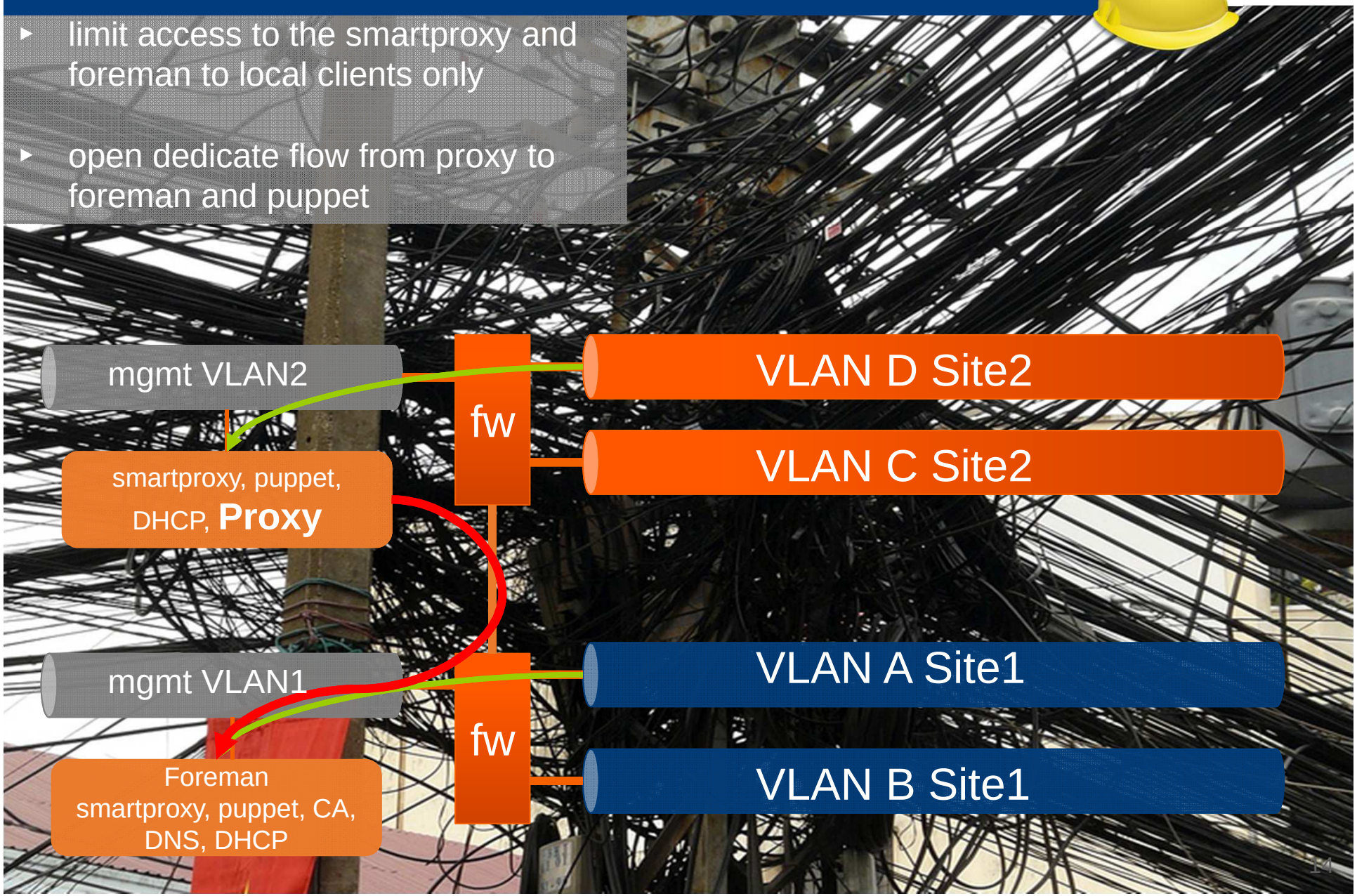
- flows at each side to local puppet and smart proxy
- all VLANs directly access puppet ca
- all VLANs directly access foreman unattended resources
- we need DNS only once (so let's limit TSIG key distribution)



Foreman Components

optimizing comms flows

- limit access to the smartproxy and foreman to local clients only
- open dedicate flow from proxy to foreman and puppet



Foreman Components

optimizing comms flows

inovex

- install apache mod_rewrite on remote smartproxy
- rewrite/proxy unattended resources
- rewrite/proxy puppet CA resources (already implemented in foreman installer puppet-puppet module via $server_ca_proxy:: by ewoud)

```
...
rewrite_rule => [
  '^/unattended/(.*)$  http://foreman.mysite.com/unattended/$1  [P]',
]
rewrite_rule => [
  '^/([^/]+/certificate.*)$ https://puppetca.mysite.com:8140/$1',
]
...
```

Foreman Components

foreman customization

inovex

- we must identify proxied servers
- foreman url must be customized
- templates must be adapted

- enable tokens (default in 1.4)

token_duration	60

- create parameter in hostgroups for each segment (unattended_url in 1.4 does not seam to help here…)

Host group parameters

Global	foreman_base	http://foreman_remote.mysite.com

+ Add Parameter

- adapt templates

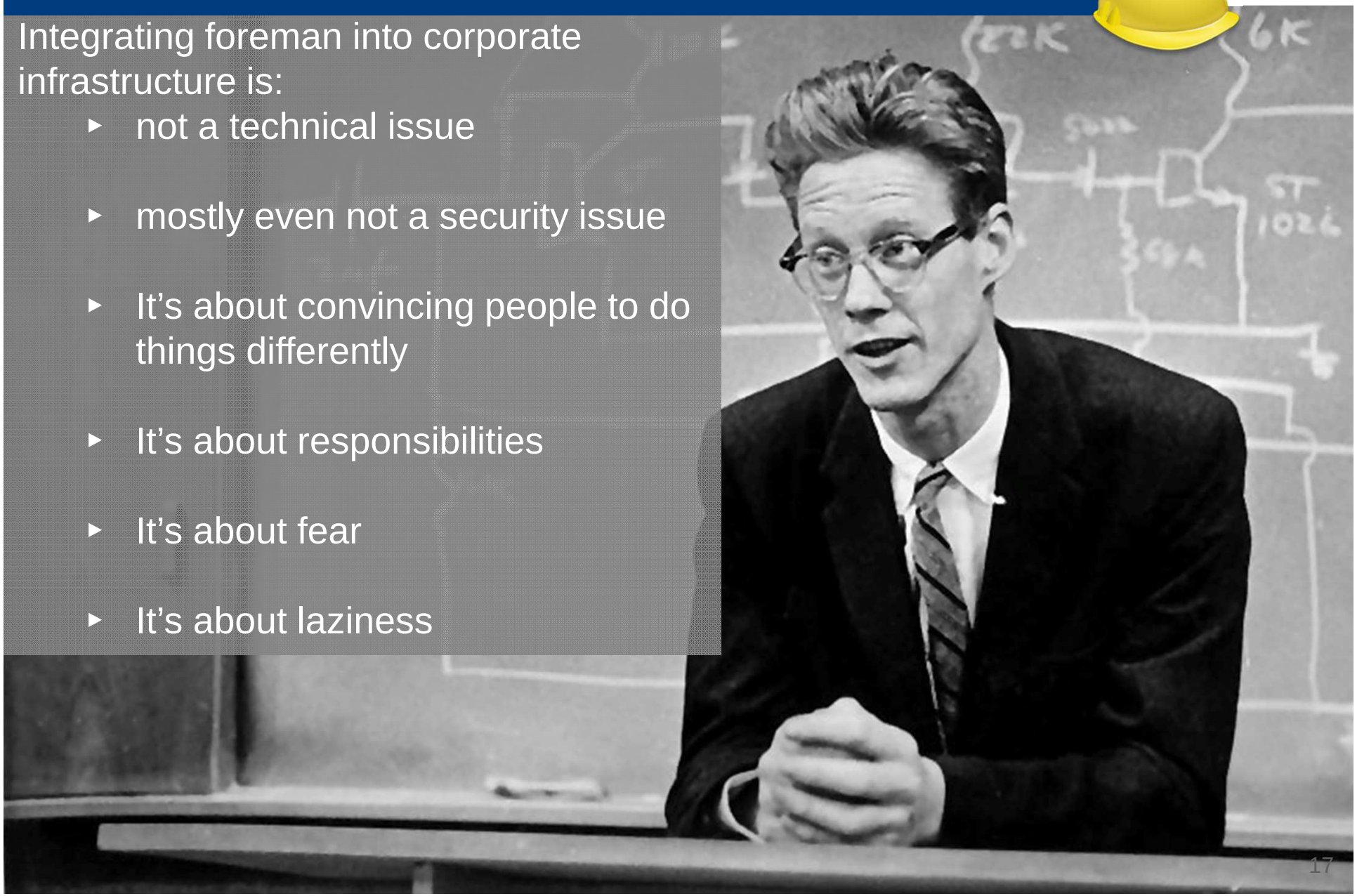
```
ks=<%= @host.params['foreman_base']%>/unattended/provision?token=<%= @host.token %>

ks=<%= foreman_url("provision")%> ksdevice=bootif network kssendmac
```

Resistance is futile

Integrating foreman into corporate infrastructure

Integrating foreman into corporate infrastructure is:

- not a technical issue
- mostly even not a security issue
- It’s about convincing people to do things differently
- It’s about responsibilities
- It’s about fear
- It’s about laziness



Resistance is futile

Let’s assimilate the corporate DHCP

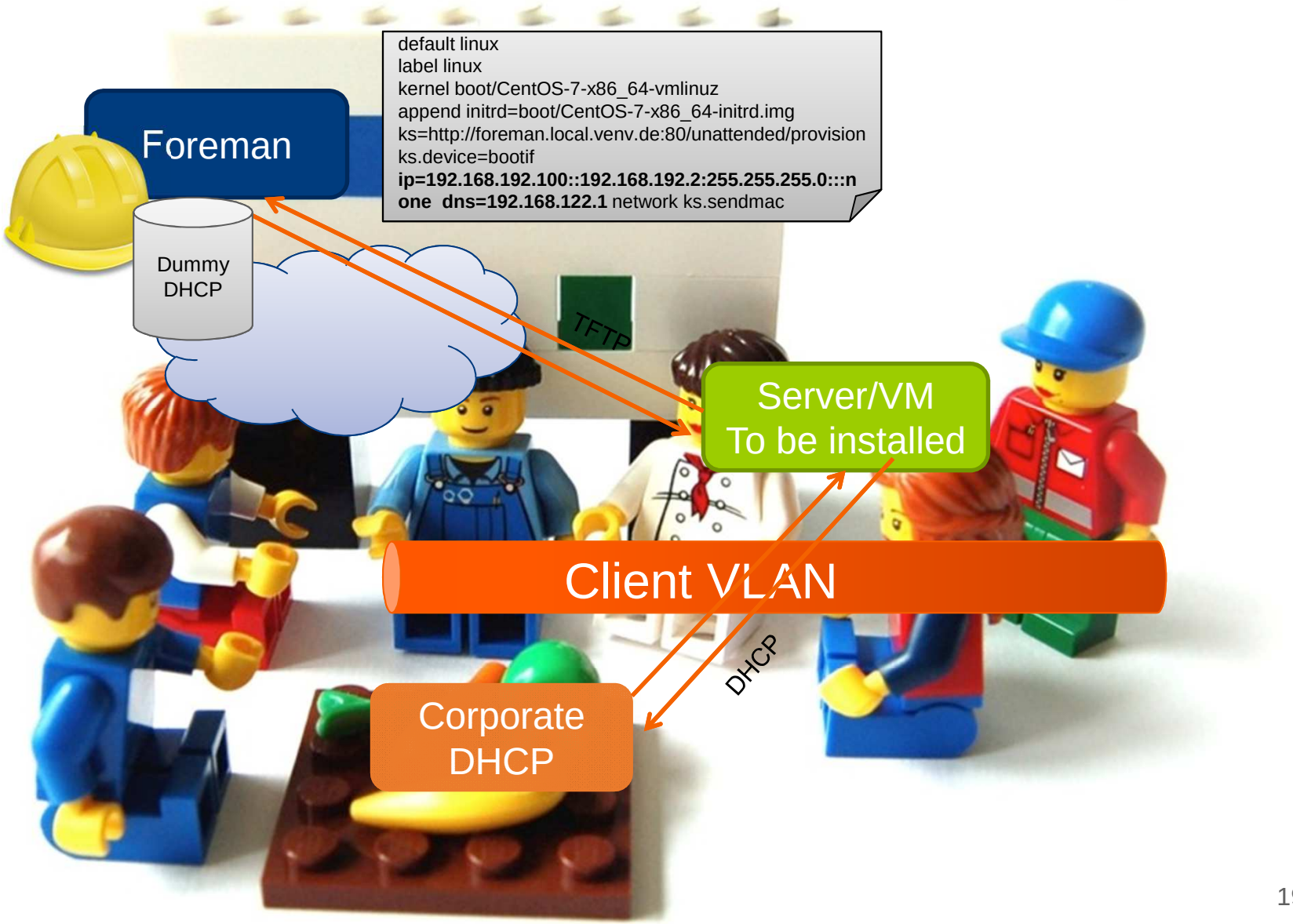
inovex

The corporate DHCP does not provide a proper API and changes are distributed across the DHCP infrastructure in a slow, asynchronous manner…

- Use a stub ISC-DHCP to manage your pools (I need some time to write+submit a stub dhcp module ☺)
- Ask for a small dynamic pxe-boot range in every subnet
- Ask for a next-server entry in the pxe-boot range
- Create a subnet on your Stub server an a range to assign addresse from

Resistance is futile

Let’s assimilate the corporate DHCP



Resistance is futile

Let’s assimilate the corporate DHCP

inovex

You must adapt you PXE templates in order to provide a static network config to darcut!

```
default linux
label linux
kernel <%= @kernel %>
append initrd=<%= @initrd %> ks=<%= foreman_url("provision")%> ks.device=bootif ip=<%= @host.ip %>::<%= @host.subnet.gateway %>:<%= @host.subnet.mask %>:::none  dns=<%= @host.subnet.dns_primary %> network ks.sendmac
```

Resistance is futile

The one with static DNS

inovex

You kindly ask for TSIG keys to create your A and PTR records automatically but the DNS is managed manually and there is no TSIG key although dynamic updates are enabled…

Make a deal: you won't talk about the missing TSIG keys and get your key or keytab and permissions right after the guy fixed the security issue.

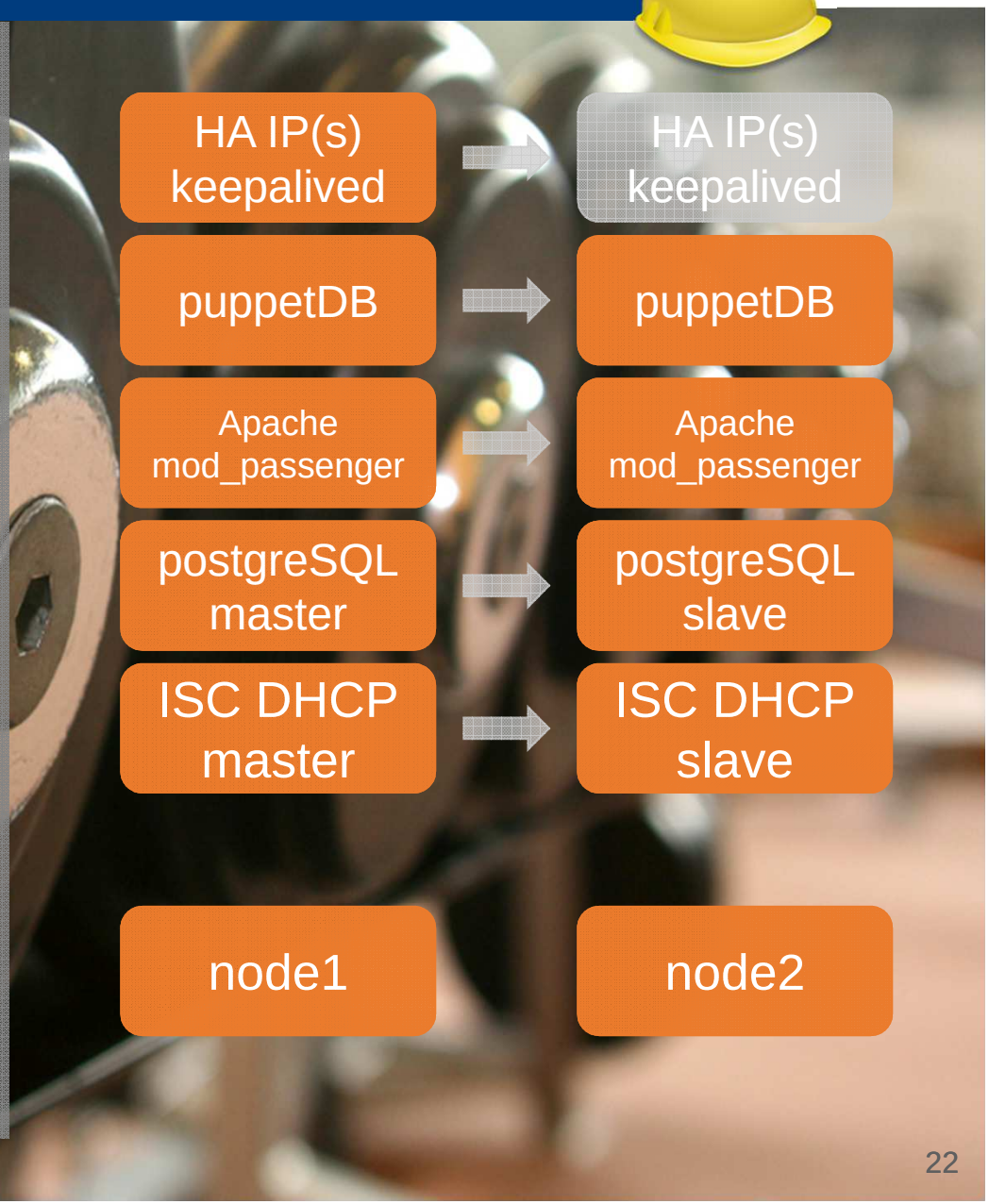
Ask for dedicated zones and for credentials which allow you to manage your zones using nsupdate

- Alternatively ask for delegation of your zones and setup a dedicated DNS server (protected by keys)

FAIL!

Foreman HA

- At least two nodes, each one running postgreSQL database
- If you scale out, you might setup a nice pg-pool2 cluster
- puppetmasters could be ran active active as well as the foreman rails application and puppetDB (keep memcached in mind)
- In addition to keepalived you can setup a software loadbalancer (i.e. ha-proxy) to balance traffic and utilize both servers for puppet and foreman traffice
- DO NOT use VIPs as ip-helper – most firewalls don't like async DHCP traffic!



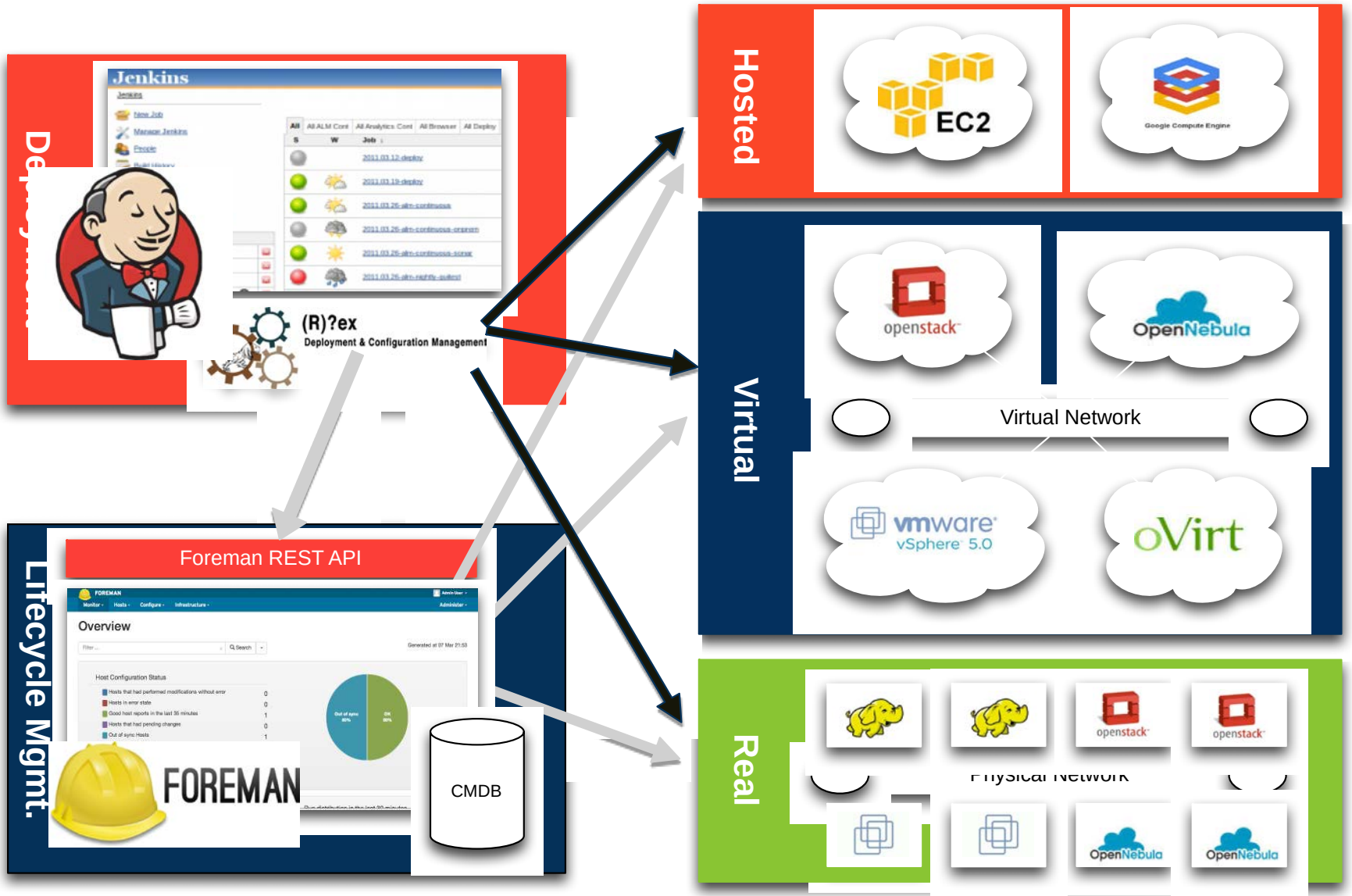
Best practices

inovex

- use separate IP addresses for each service so you can split up stuff later on
- If you can’t use DHCP use the bootdisk plugin
- LDAP or kerberos5 for user authentication
- use Locations or Organizations for filtering
- use the column plugin for better overview
- ENC and smart variables are nice – ever configured dhcp pool as yaml hash? Use smart variables as switches, and for simple datatypes only - do the complex stuff in service modules.

The big picture

inovex



Centos 7

...secret guide to world domination

inovex

- Can be deployed using the community templates . Or by adapting custom templates
- Currently not yet running out of the box on CentOS 7 - support will be added in 1.6.0
- Some minor problems around selinux – setting permissive mode allows foreman to run
- Foreman Installer relies on facts, facter relies on host and ifconfig binaries – both not installed by default.

https://github.com/theforeman/community-templates/

Centos 7

...secret guide to world domination

inovex

- Mind firewalld

```
`--> firewall-cmd --zone=public --add-port=8140/tcp
`--> firewall-cmd --zone=public --add-port=8140/tcp --permanent
```

Wishlist

...secret guide to world domination

inovex

- integrate the reverse proxy feature into smart-proxy (there is a ticket for that somewhere)
- implement iso image provisioning based vm installation in conjunction with the bootdisk plugin
- support more configuration management stacks
- better support for discovery, ipmi and hardware provisioning (like HW-Raid) – on-going

Thank You!

Contact

Nils Domrose
Senior Systems Engineer

inovex GmbH
Office cologne
Schanzenstr. 6-20
51063 Köln

nils.domrose@inovex.de