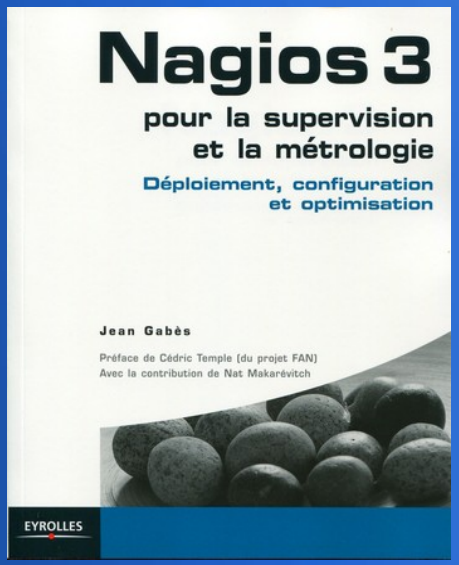
SophiaConf 2012

Jean Gabès

Qui suis-je ?

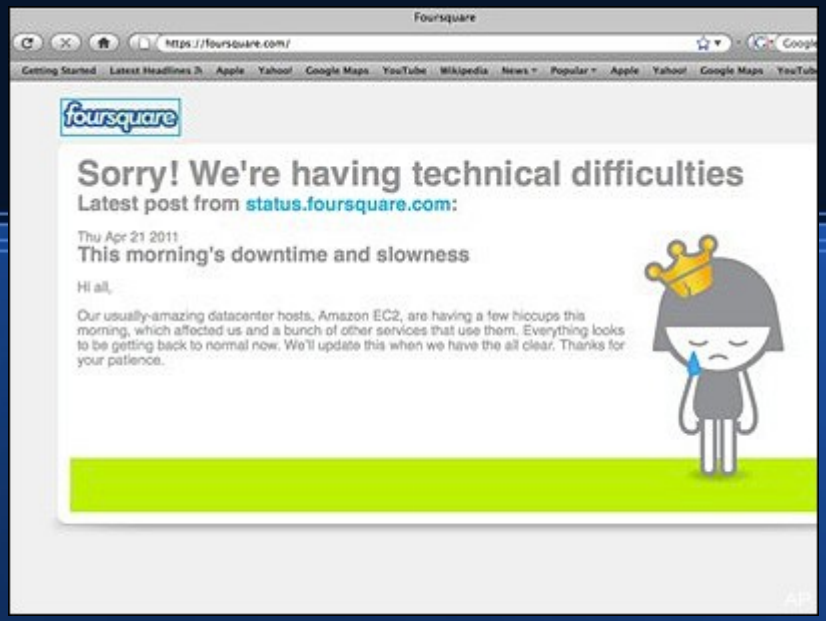
Jean Gabès

Administrateur système sur Bordeaux,
auteur du livre Nagios3 aux éditions Eyrolles
et de Shinken

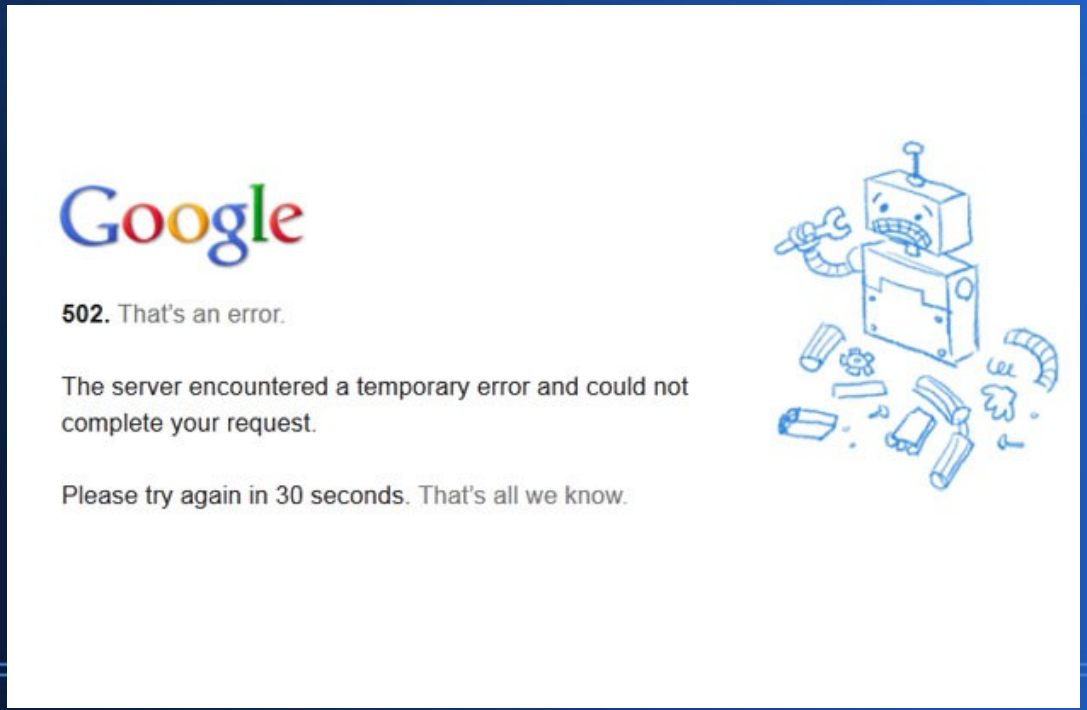


Pourquoi superviser ?

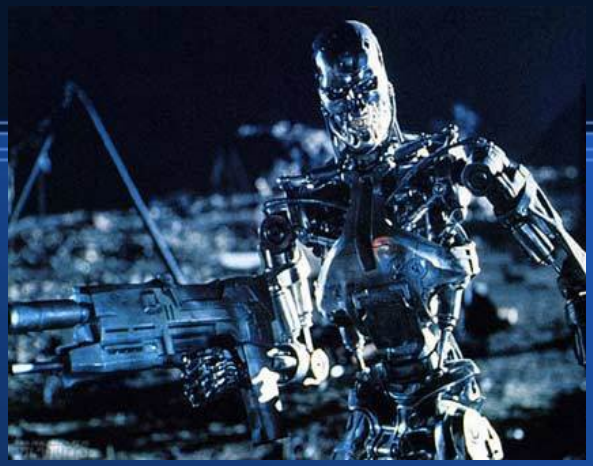
Quand l'IT va mal, le business va mal...

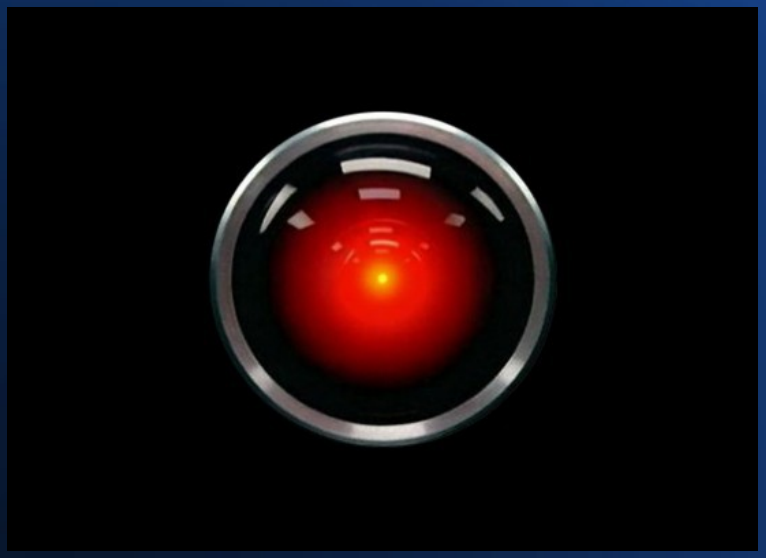

Foursquare
https://foursquare.com/
foursquare
Sorry! We're having technical difficulties
Latest post from status.foursquare.com:
Thu Apr 21 2011
This morning's downtime and slowness
Hi all,
Our usually-amazing datacenter hosts, Amazon EC2, are having a few hiccups this morning, which affected us and a bunch of other services that use them. Everything looks to be getting back to normal now. We'll update this when we have the all clear. Thanks for your patience.

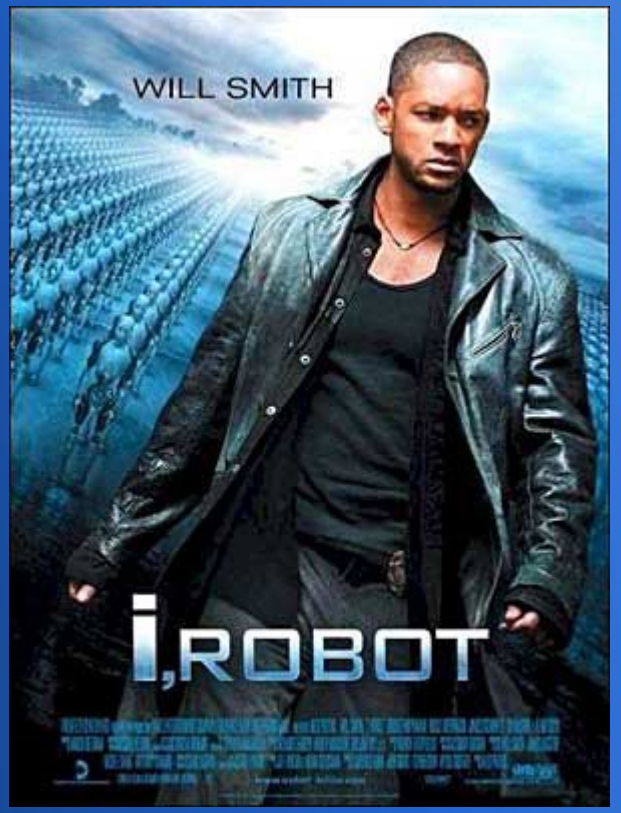

Online Services


Google
502. That's an error.
The server encountered a temporary error and could not complete your request.
Please try again in 30 seconds. That's all we know.

Ou voire pire....






WILL SMITH
I,ROBOT

Pour sauver le ~~monde~~ business :
Les outils de supervision !

Bon il y en a pleins...

Concernant la supervision pure IT, Nagios™® est la référence de ces 10 dernières années ...

Concernant la supervision pure IT, Nagios™® est la référence de ces 10 dernières années ...

… grâce à beaucoup de modules

- Mod_gearman : Distribution de la charge sur le LAN
- LiveStatus : accès aux données
- Thruk/Multisite/NagVis : vue temps-réel
- PNP, Graphite : graphiques

Et des plugins de supervisions

```
$ check_disks -w 90% -c 95%
Disks OK | /=50% /var=80% /data=35%

$ echo $?
0
```

Plugins & modularité sont bien !

Mais si ce n'était plus suffisant ?

L'IT grossit de jours en jours

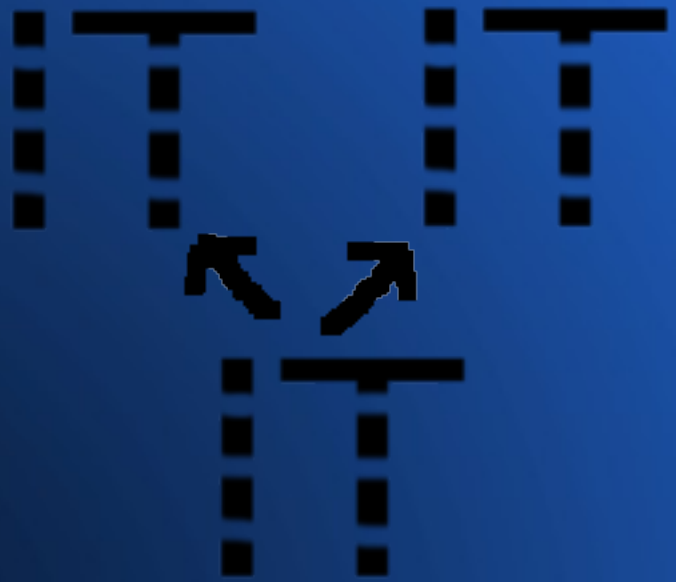
IT

IT

IT

Avec plein de couches (physique, réseaux, virtuel, …)

Rempli de clusters

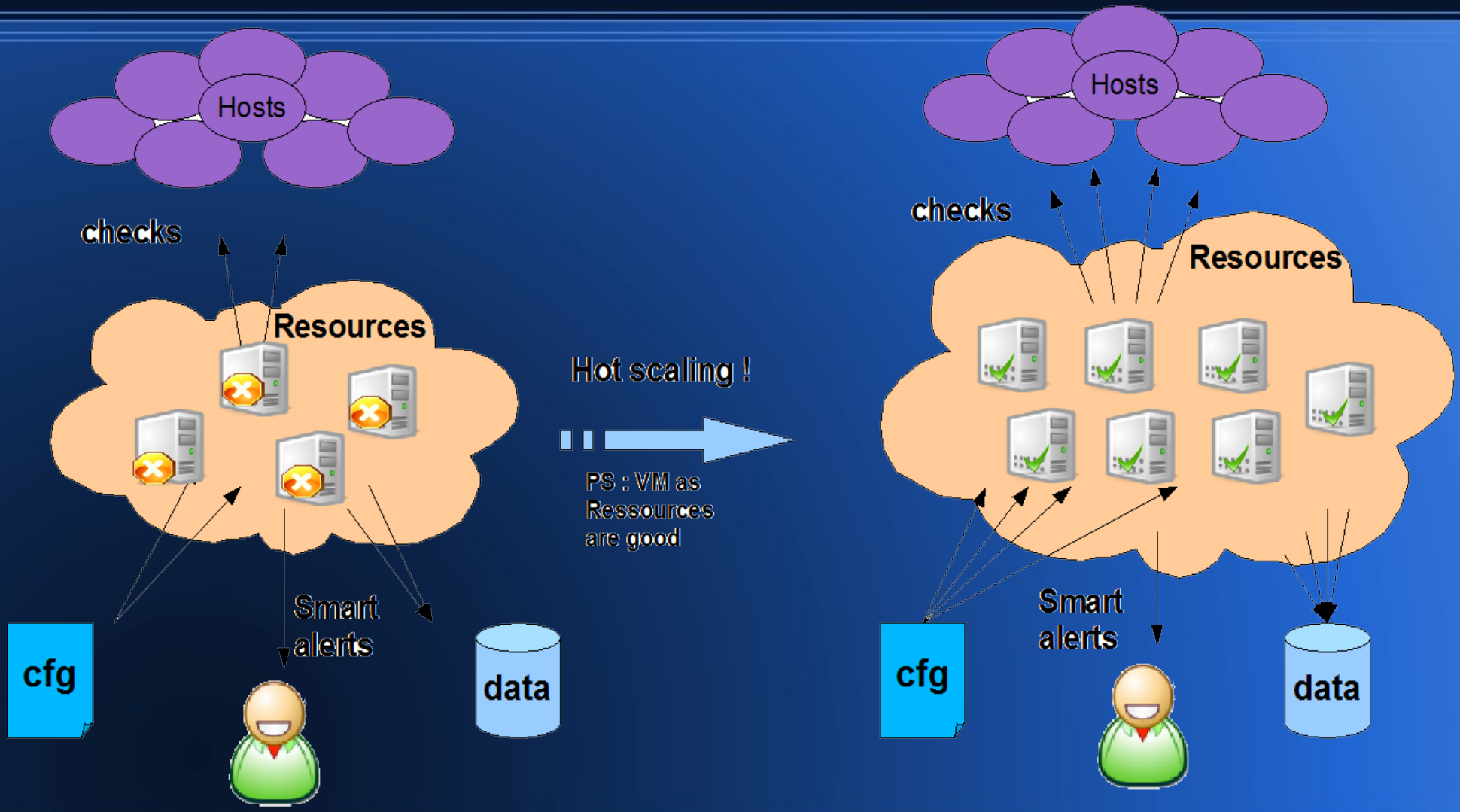


Et de sites distants

Difficultés classiques dans la supervision IT

- Trop de charge
- Gestion de la configuration
- Perte de site distant ?
- Haute disponibilité

Architecture idéale



Oui vous pouvez bricoler autour d'un Nagios™®
pour cela ...

… ou vous pouvez juste utiliser Shinken :)

Shinken est une réécriture complète de Nagios™® (C) en Python

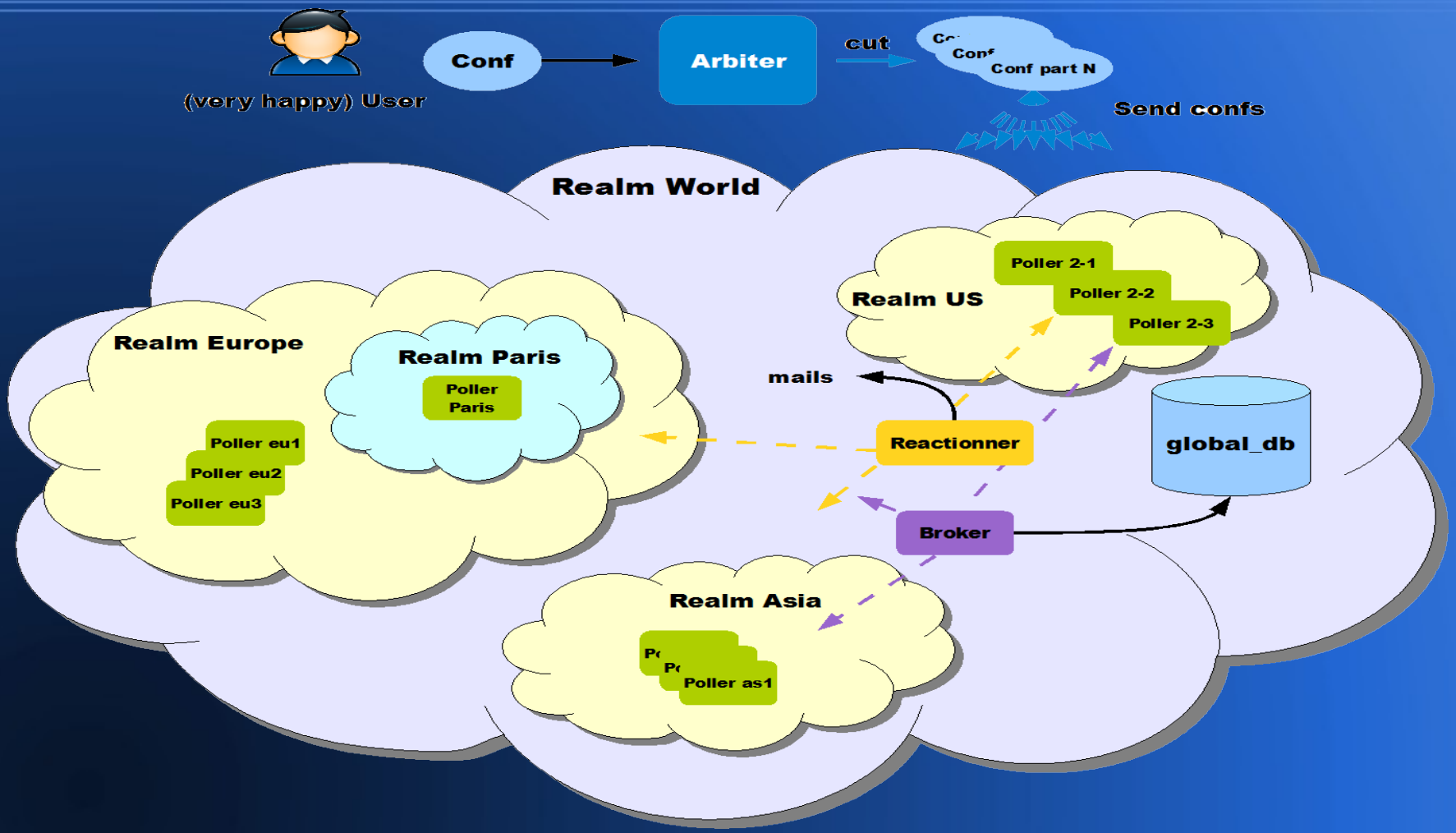
Attendez ! Python c'est lent !

Ok, on bench. Xeon 4cores@2.9ghz, 12Go ram, dummy check

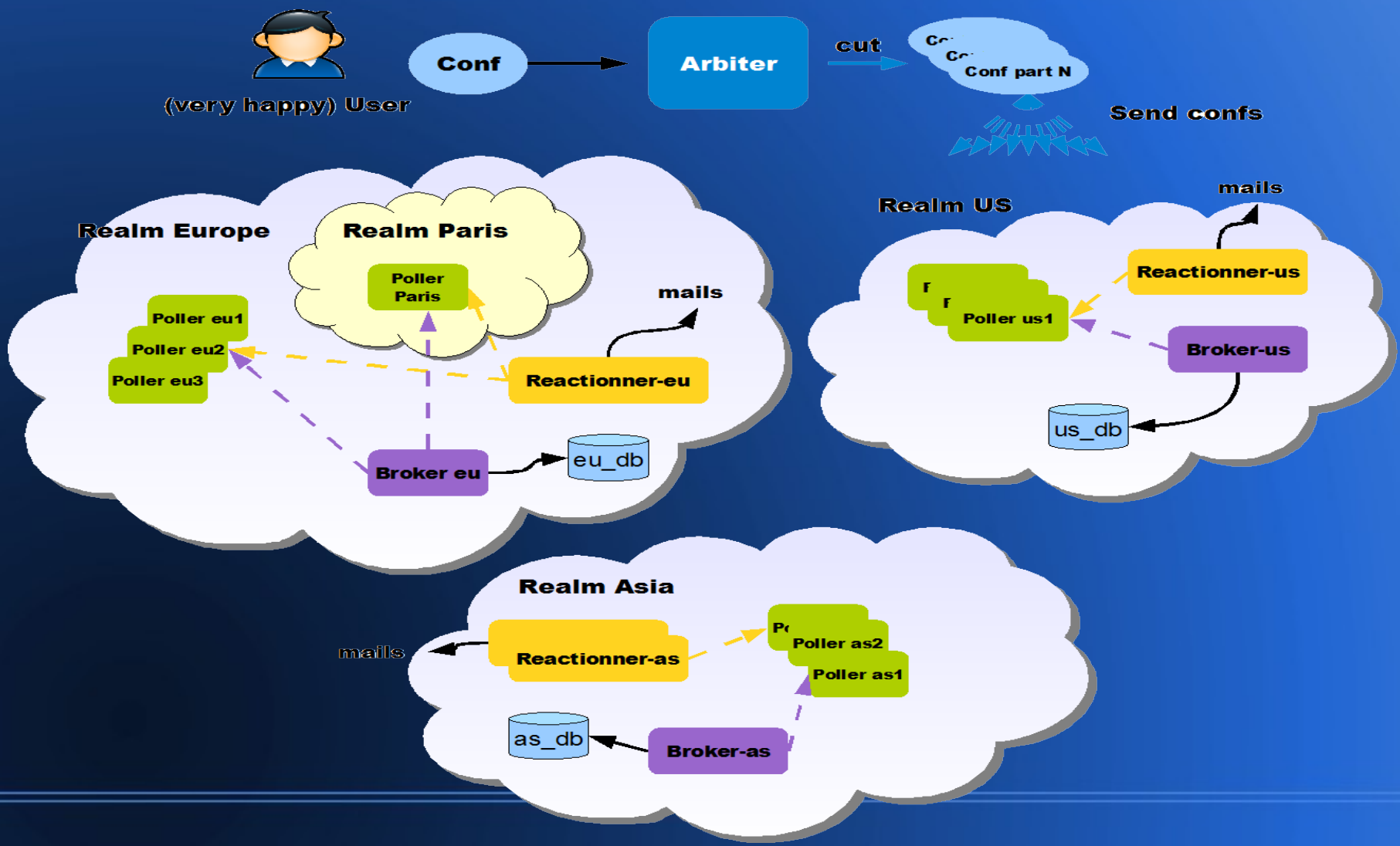
(en nombre max de checks en 5min)

- Nagios 3 (C) <=> icinga 1.6 (C) = 30K 
- Centreon-engine (C) = 25K 
- Shinken = 120K 

De grandes possibilités d'architecture



De grandes possibilités d'architecture



Les soucis de supervision IT sont réglés, mais quid des problèmes des admins de 2012 ?

La virtualisation est partout

(OK peut être moins dans le futur avec la nouvelle politique de licence d'ESX5...)

Si un ESX crash, vous n'avez pas envie de recevoir 20+ alertes pour les VM dessus !

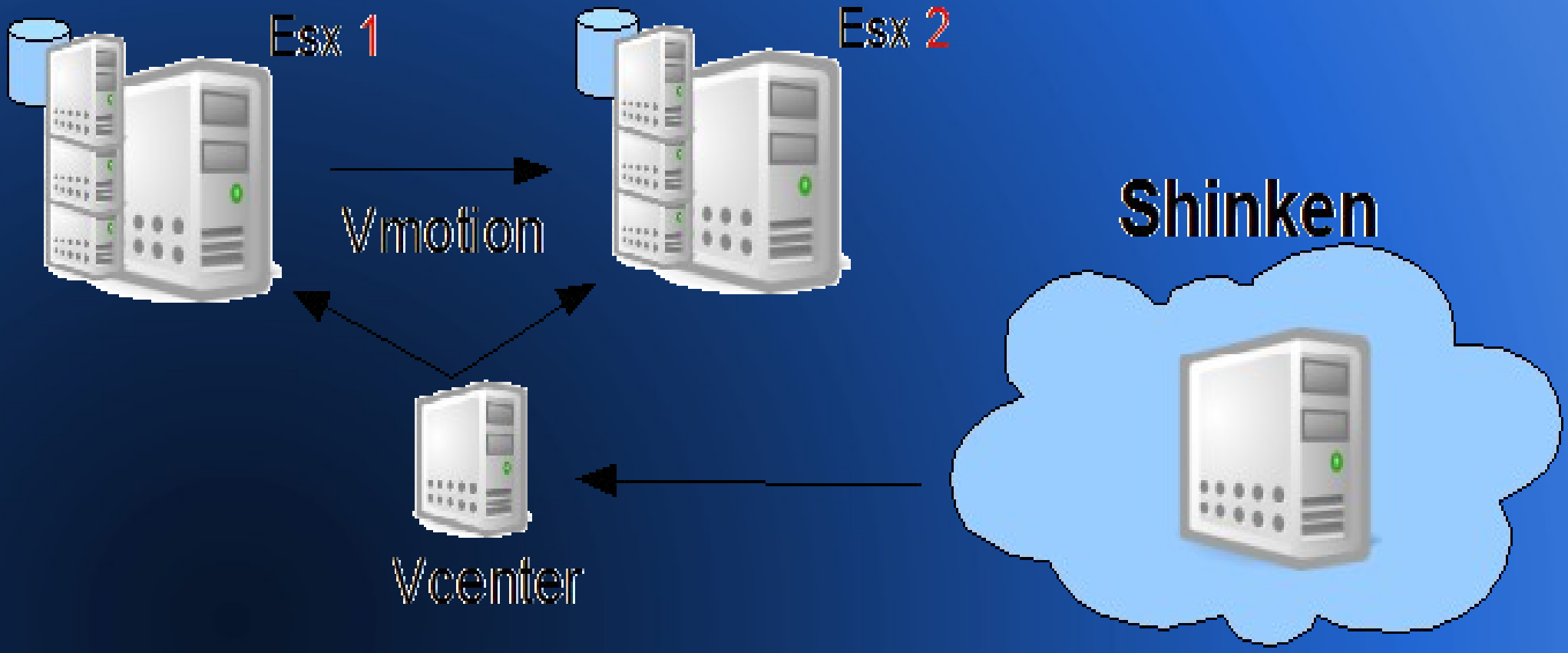
Que l'alerte de l'ESX. Simple : dépendance d'hôte :)

Mais les admins Vmware sont des personnes étranges

Ils « VMotion » des VMs aussi souvent qu'un codeur Perl tapes $_

Donc oubliez la configuration en fichiers plats manuels :)

Ils suffit d'utiliser le module Vmware™® pour Shinken


Esx 1
Esx 2
Vmotion
Shinken
Vcenter

OK, et si on s'attaquait a l'un des pires soucis des admins ?

Pas la pénurie de café/bière...

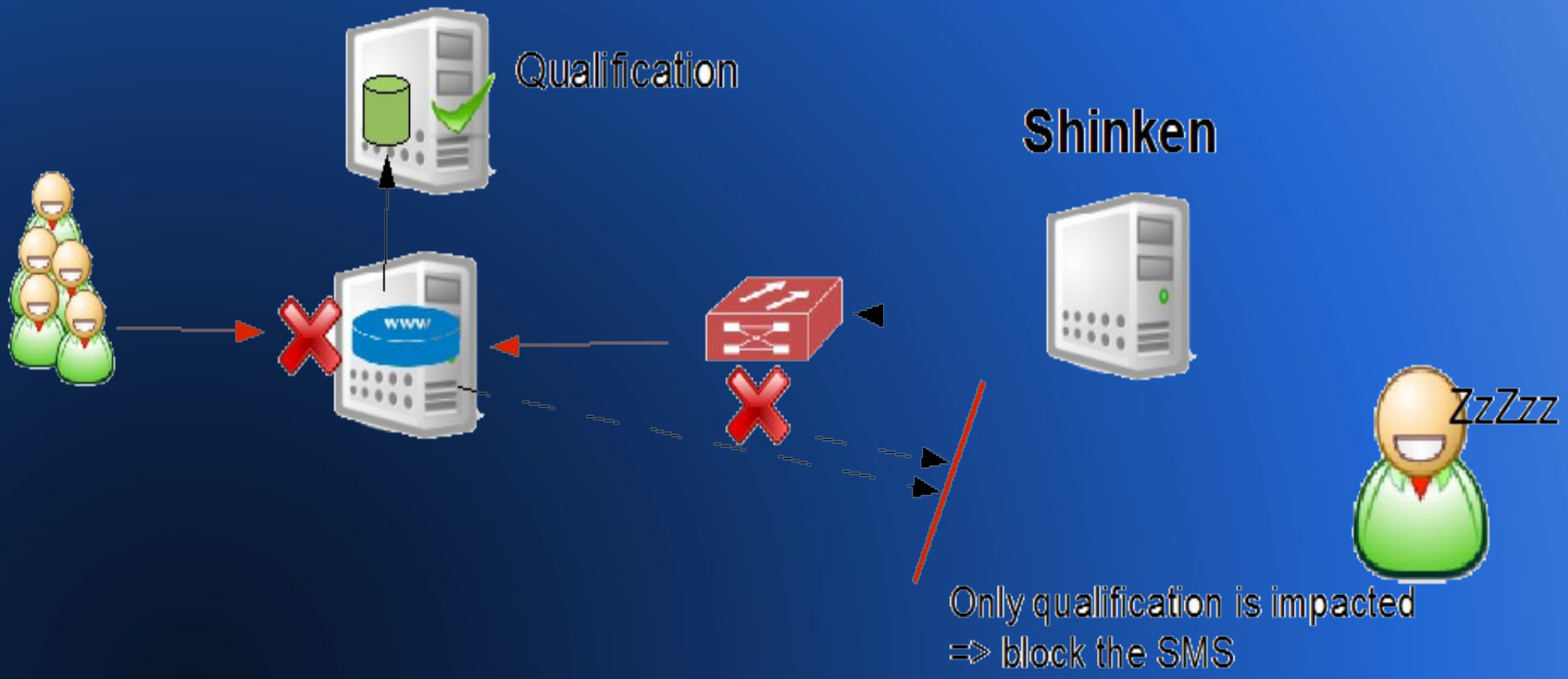
Les fausses alertes !

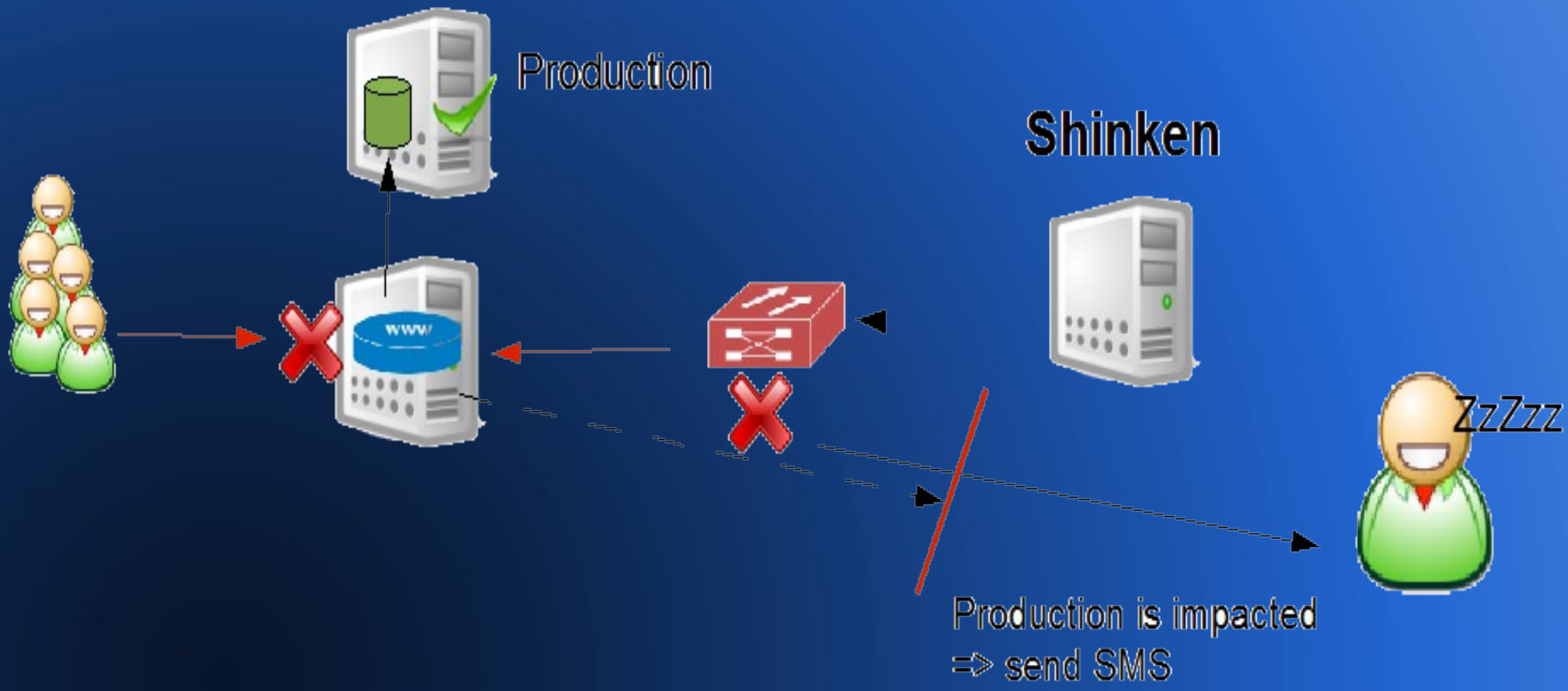
Exemple : une erreur critique sur de la qualification, est-ce vraiment si critique ?

Cas réel : un switch de production casse une application de qualification

Est-ce que l'on doit réveiller le pauvre admin à 3h du mat pour ça ?? Non !

Se concentrer sur l'analyse des problèmes sources, et l'impact au niveau « business » des applications


Qualification
Shinken
ZzZzz
Only qualification is impacted
=> block the SMS


Production
Shinken
www
ZzZzz
Production is impacted
=> send SMS

La différence entre problèmes sources et impacts est centrale dans Shinken

Et entre les niveaux d'importances business, plus que juste warning/critical

Ok pour les alertes. Mais quid des interfaces graphiques ?

Shinken WebUI :

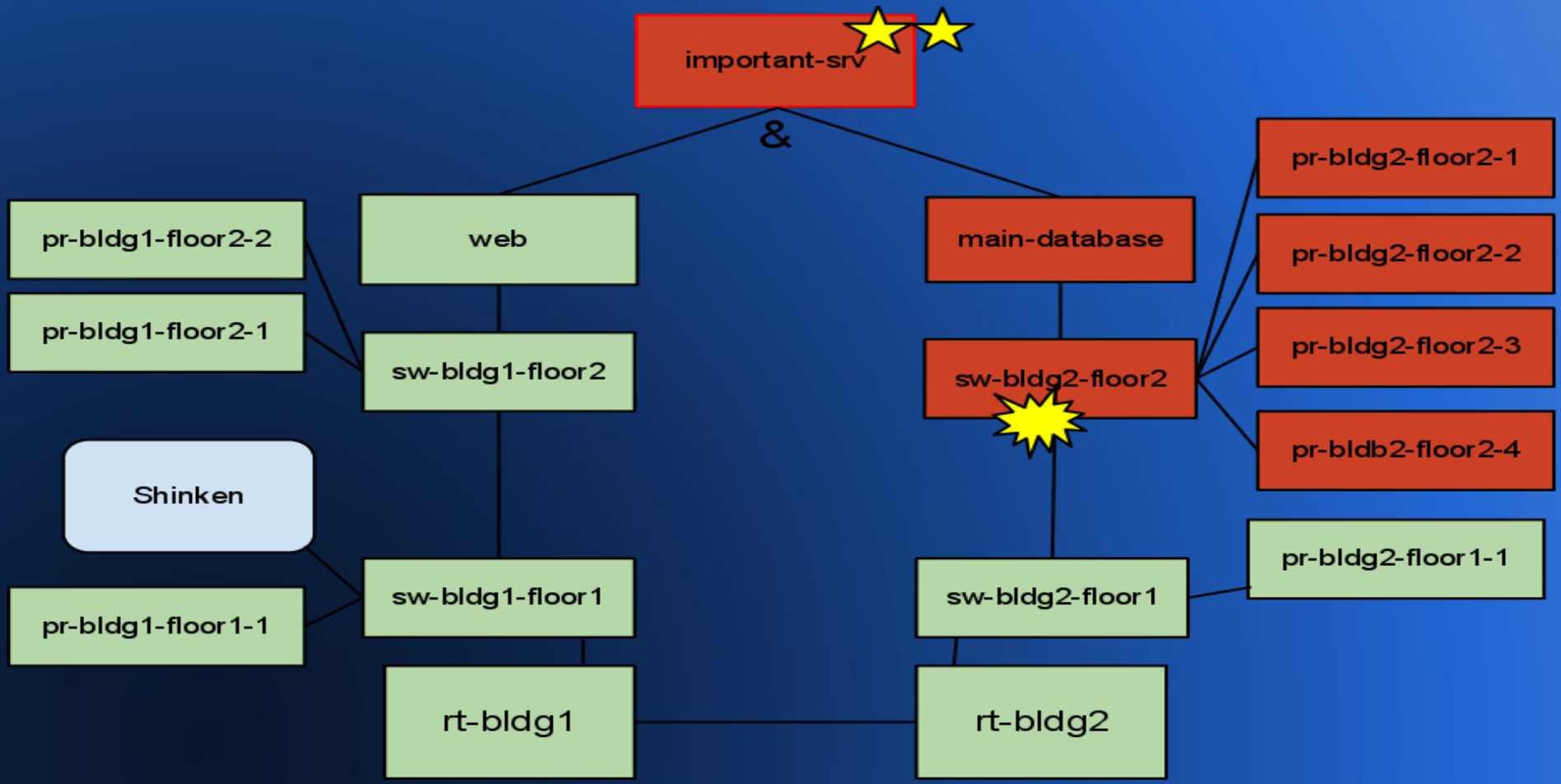
- Problèmes sources & impacts sur des vues différentes
- TOUT est trié suivant l'importance business
- Visualisation des dépendances sous forme d'arbres et de graphes
- HA et agrégation de données
- Pas de base de données !

- Très “visuelle”
- HTML5 (désolé pour IE6...)
- Limitation (volontaire...) des informations affichées
- Modulaire (PNP, graphite)
- Même votre boss va la comprendre...

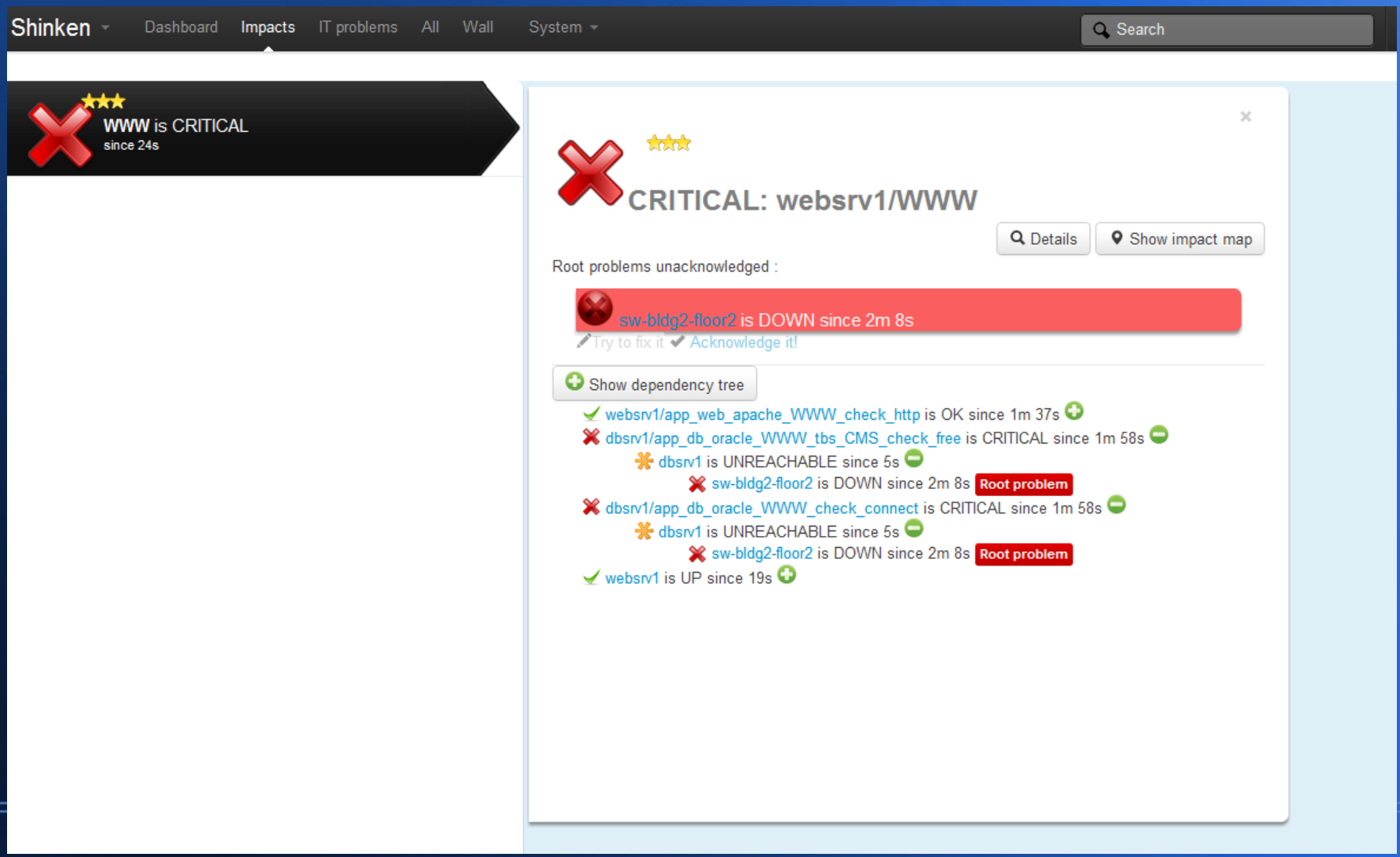
Deux types d'utilisateurs (incompatibles)

- Responsables : visualisation des impacts sur les applications end-users (et pourquoi elles sont tombées)
- Admins : visualisation des éléments IT qui sont les problèmes sources

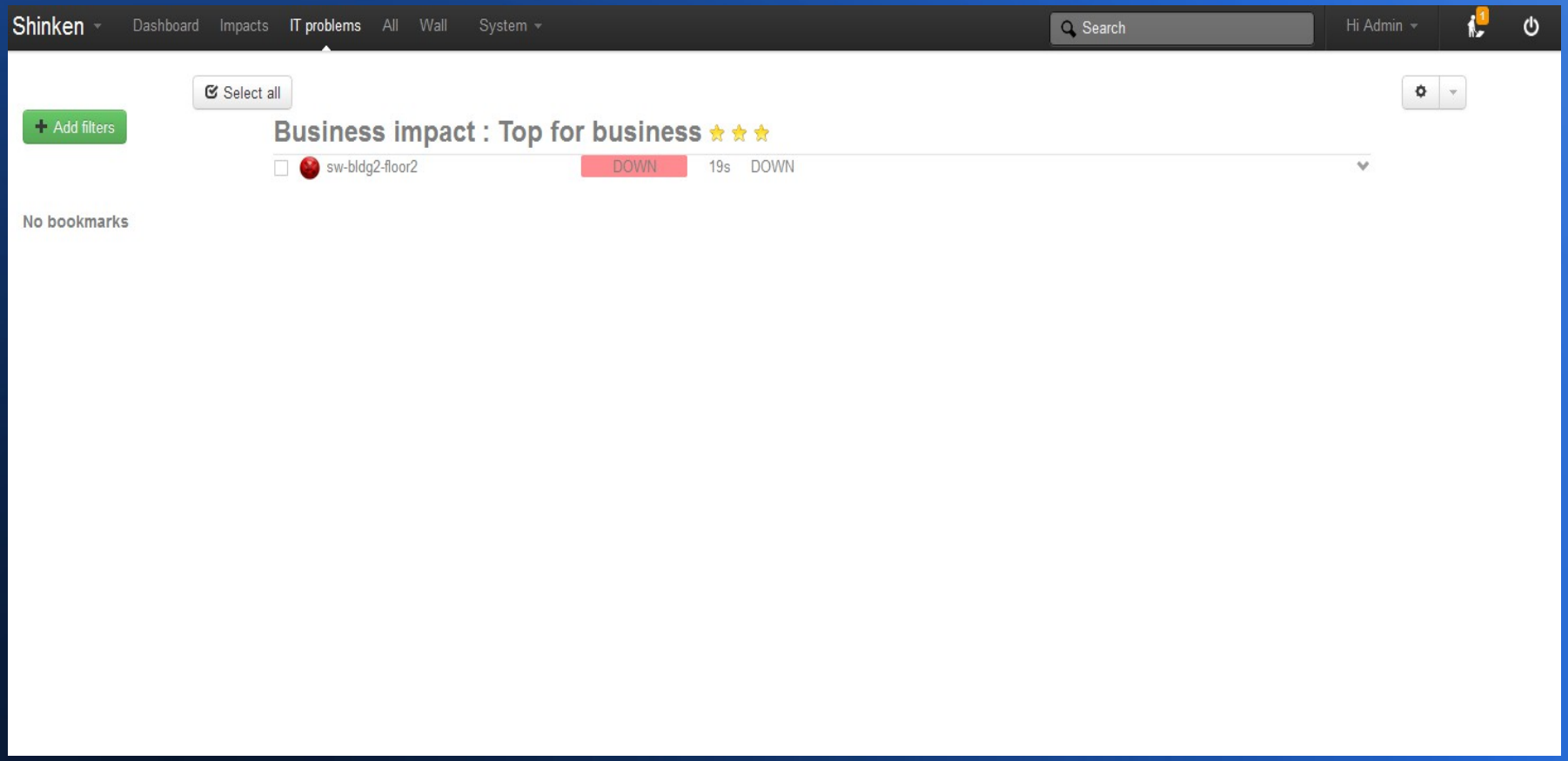
- Vues Problèmes sources VS impacts
- Personne ne veux voir les deux
- Le tout trié suivant l'importance métier


important-srv
&
web
main-database
pr-bldg1-floor2-2
pr-bldg1-floor2-1
sw-bldg1-floor2
sw-bldg2-floor2
pr-bldg2-floor2-1
pr-bldg2-floor2-2
pr-bldg2-floor2-3
pr-bldb2-floor2-4
Shinken
pr-bldg1-floor1-1
sw-bldg1-floor1
sw-bldg2-floor1
pr-bldg2-floor1-1
rt-bldg1
rt-bldg2

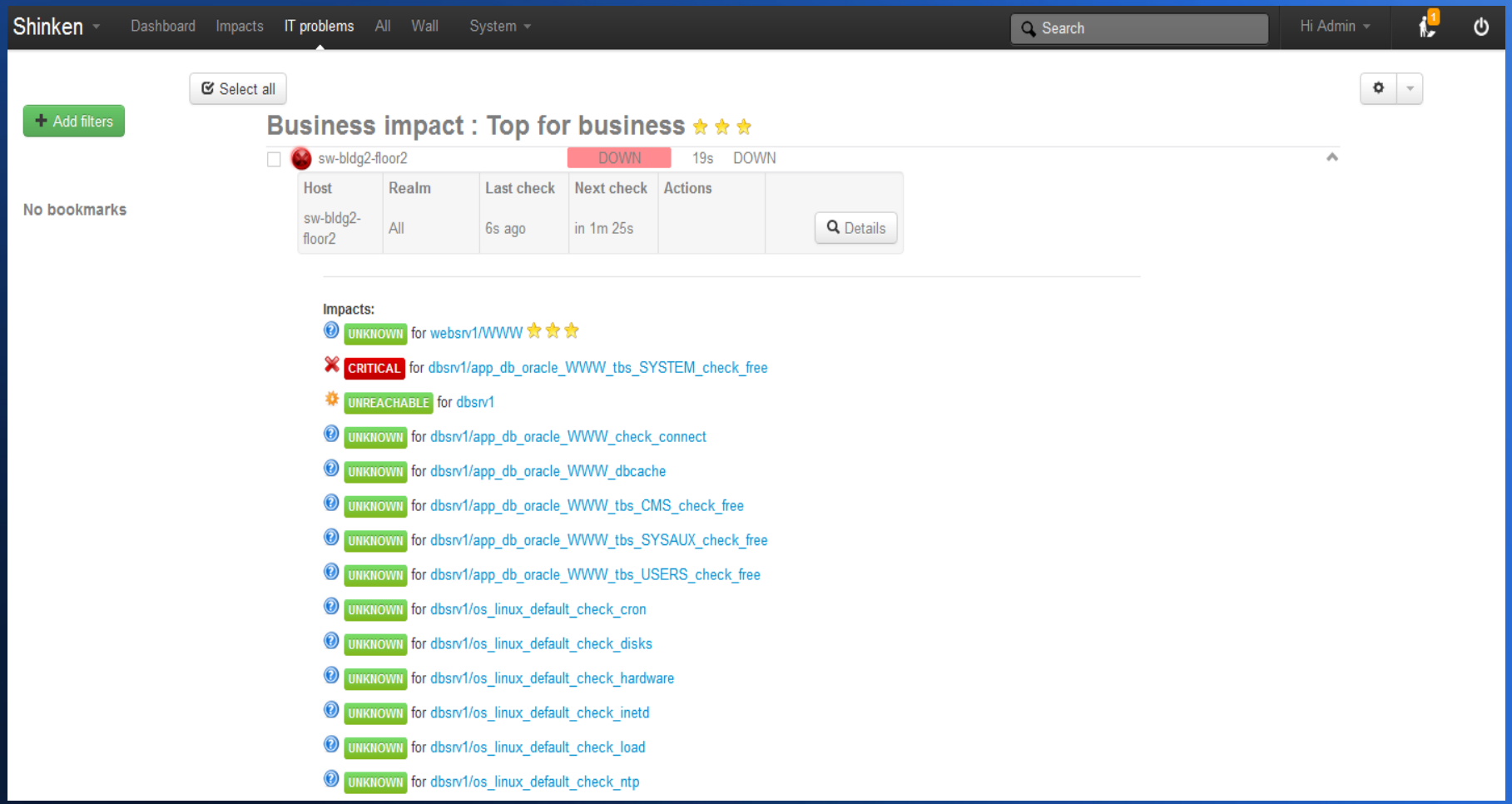
Vue du responsable



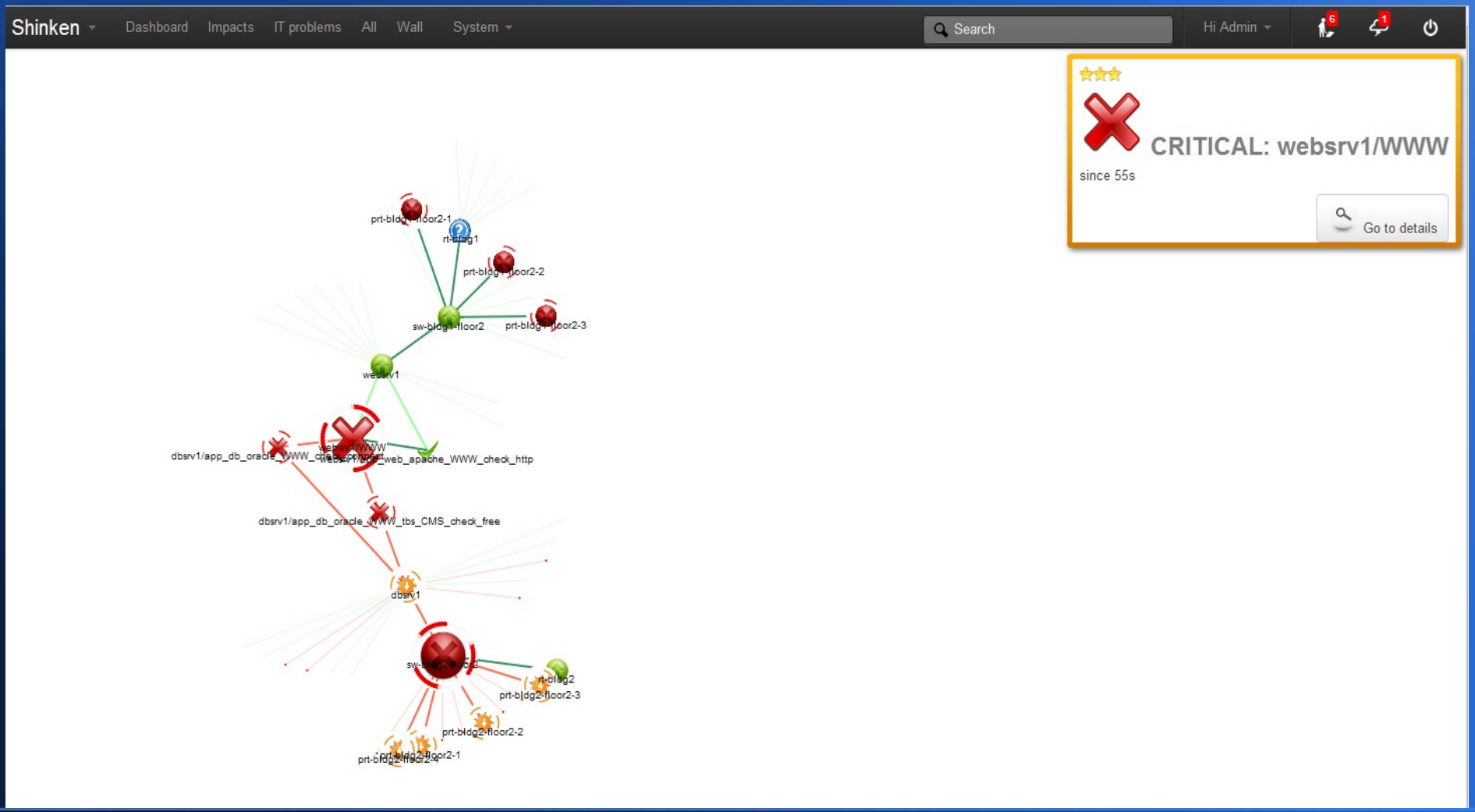
Vue d'un administrateur



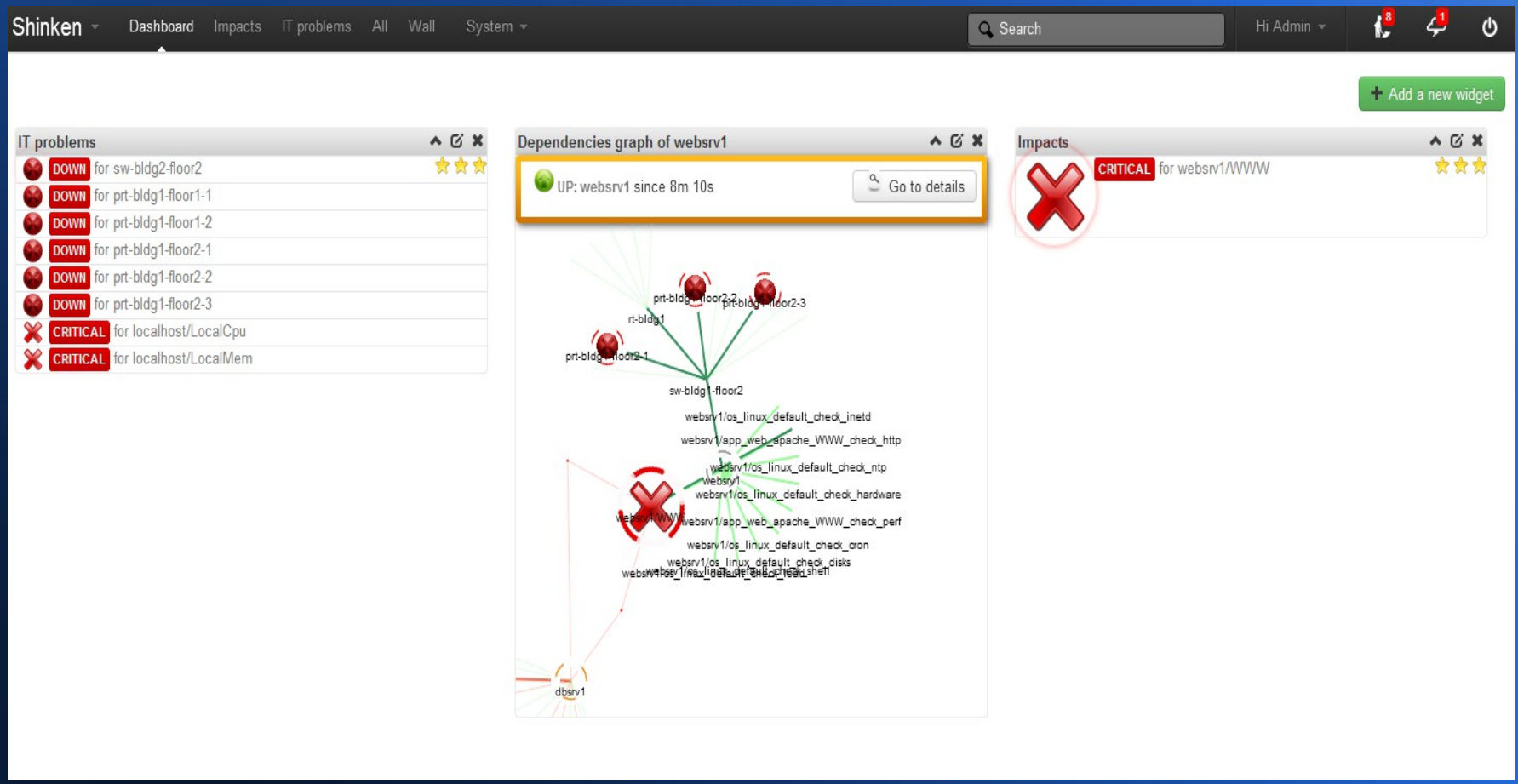
Et si l'admin veut voir pourquoi c'est si important...



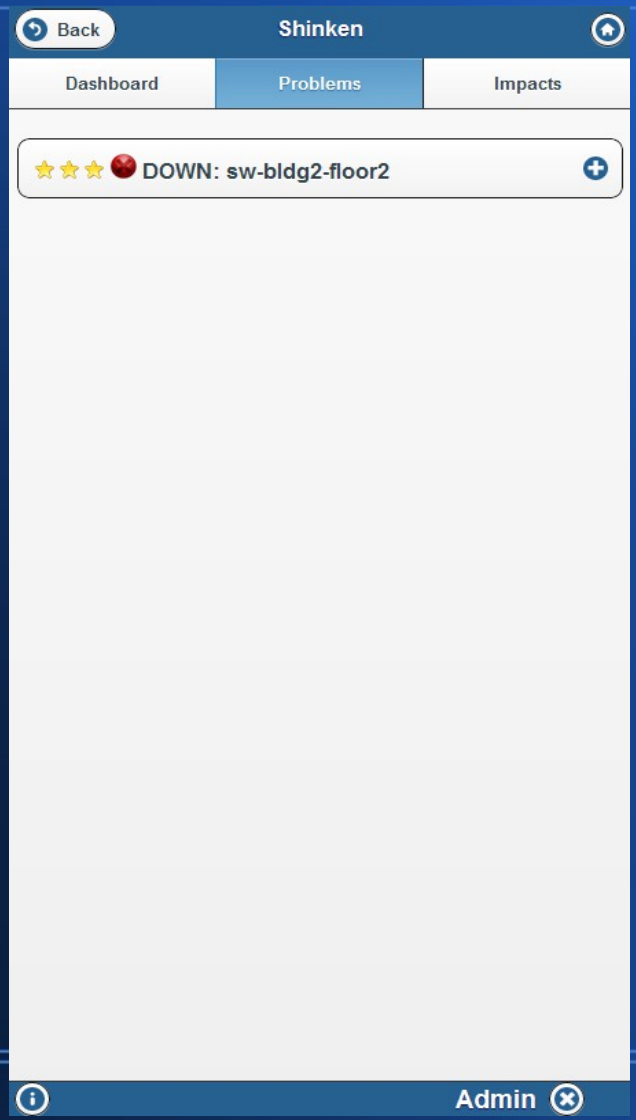
Les deux vont comprendre un graphe



Et chacun peut avoir SON dashboard



Et sa version « mobile »


Back
Shinken
Dashboard
Problems
Impacts
DOWN: sw-bldg2-floor2
Admin

Demo : http://demo-shinken.web4all.fr/

OK, on voit seulement ce que l'on souhaite voir.
Bien.

Mais une lourde tâche reste : rajouter nos serveurs dans l'outil!

Les templates de configuration de Nagios™® sont indispensables!

Mais pas suffisant. Il reste encore trop d'éléments à définir

Shinken permet d'accrocher des vérifications par des expressions complexes du genre « Linux&Prod »

Ceci permet de n'avoir qu'à « tagger » ses machines au lieu de multiplier les groupes et sous groupes de serveurs

Dans Nagios™®, on considère que 80% de la configuration consiste en l'écriture des services.

Shinken propose des solutions pour limiter fortement ces 80%.

Exemple : clé duplicate_foreach
Génère un service par “propriété” d'un hôte.

```
Define host{
host_name  srv-lin-1
Use        linux
_disks     /, /var, /data
}
```

```
Define service {
    host_name          linux
    Register           0
    Description        Disk $KEY$
    check_command      check_disk!$KEY$
}
```

Fait prouvé : un bon informaticien est fainéant

Fait sûrement vrai : les admins sont de bons informaticiens!

Un admin préfère éviter de :

- Écrire un plugin de 0
- Tagger manuellement ses machines
- Écrire la configuration de nouveaux types de serveurs/applications

- Plugins : Merci Monitoring-exchange.org!
- Tagging : merci skonf discovery
- Nouvelle configuration : merci les packs Shinken

Pourquoi tagger manuellement ses serveurs quand on peut juste écrire des règles pour le faire?

Module de découverte de Shinken!

- Runners : scripts qui “scanne” et qui sort des données
- Rules : lit les données et génère de la configuration

Ex : nmap runner scanne un serveur et exporte des données

$ nmap_discovery_runner.py -t localhost

localhost::isup=1
localhost::os=linux
localhost::osversion=2.6.x
localhost::osvendor=linux
localhost::macvendor=hp
localhost::openports=22,80,3306
localhost::fqdn=localhost
localhost::ip=127.0.0.1

Exemple de règle qui détecte et tag “linux”

```
define discoveryrule {
        discoveryrule_name      Linux
        creation_type           host

        os                      linux       ; what we match
        +use                    linux       ; what we wrote in the object, here
                                            ; append the linux template
}
```

localhost : use ssh,mysql,http,linux

Découverte multi-niveaux :

- 1 Si on match une donnée
- 2 On lance un nouveau runner
- 3 On ré-applique les règles
- 4 GOTO 1

Ex : découverte des shares Windows

```
define discoveryrun {
        discoveryrun_name           WindowsShares
        discoveryrun_command        discovery_windows_share

        # And scan only windows detected hosts!
        os                          windows
}
```

Résultat

```
define host {
        host_name               win-srv
        use                     windows

        _shares                 Work,Public,Private
}
```

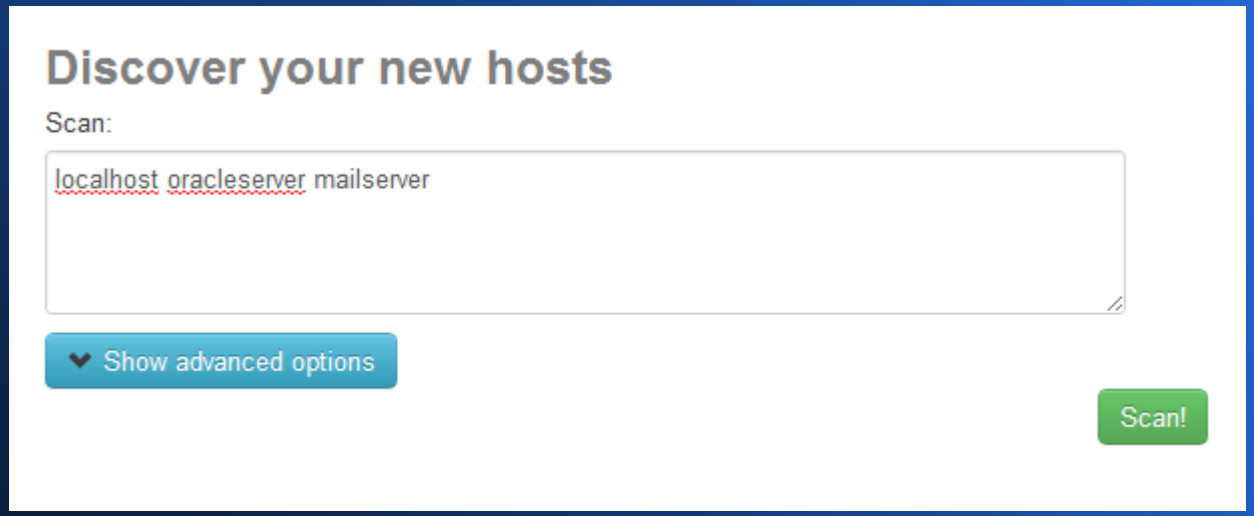
Lancement en CLI :

shinken-discovery -c etc/discovery.cfg --db Mongodb -m 'NMAPTARGET=localhost'

Ou mieux : l'UI sKonf!

sKonf :

- UI pour la gestion facile de sa configuration
- Découverte ou configuration classique
- Gère les paramètres spécifiques de Shinken
- Stade Beta avancé


Discover your new hosts
Scan:
localhost oracleserver mailserver
Show advanced options
Scan!

Here are the results :

oracleserver Hostname

oracle × linux × ssh × generic-host × Tags

✔ Validate

mailserver Hostname

windows × windows2008 × http × https × smtp × imap × imaps × generic-host × Tags

✔ Validate

localhost Hostname

mysql × linux × http × ssh × generic-host × Tags

✔ Validate

Generic | Macros | Notifications | Dependencies | Advanced

Hostname: oracleserver

Display name: oracleserver

Address:

Tags: oracle × linux × ssh × generic-host ×

Maintenance Period:

Check Period:

Check Command:

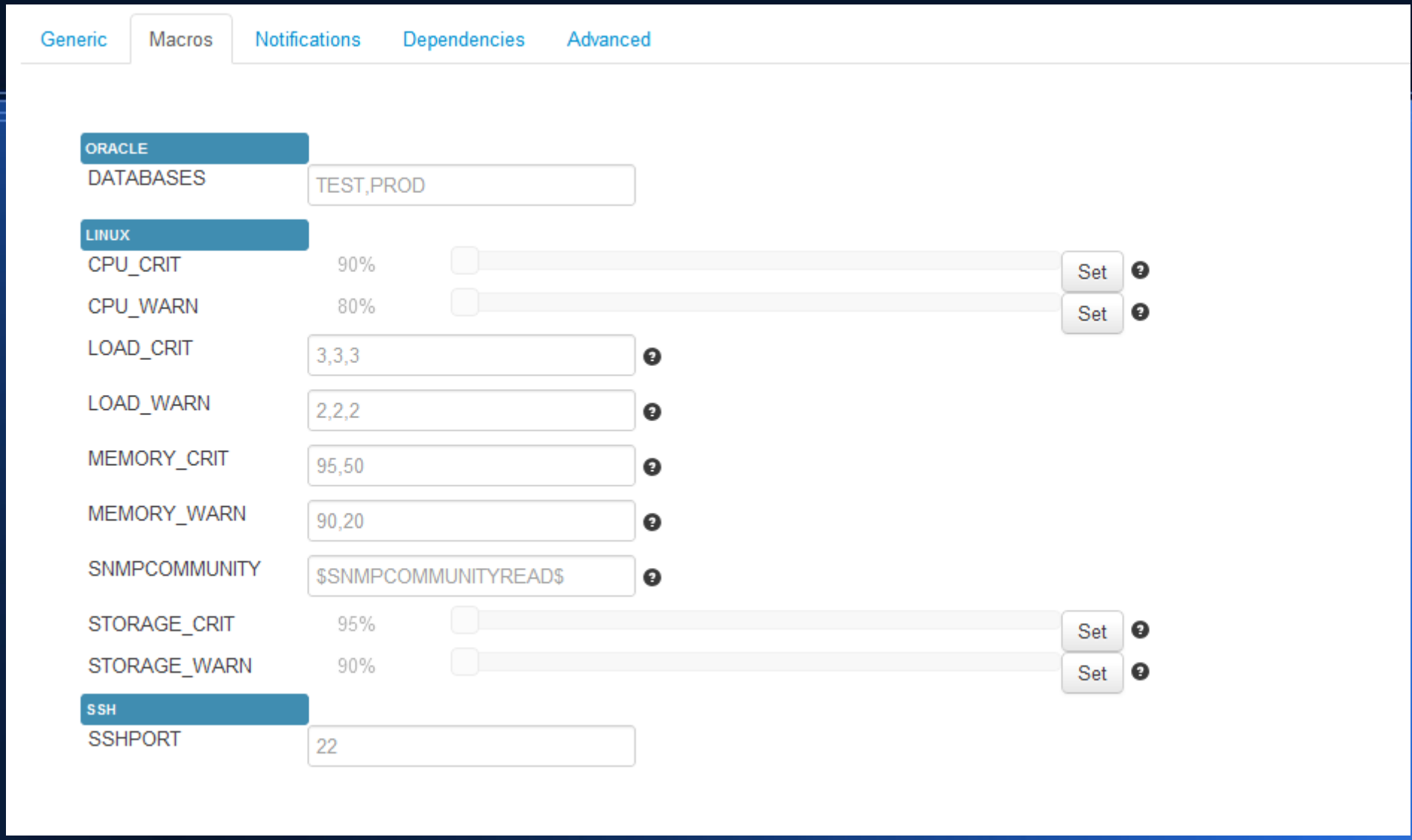
Args:

Max Check Attempts:

Normal Check Interval*
60 seconds

Active Checks Enabled: On | Off | Unset

Passive Checks Enabled: On | Off | Unset


Generic
Macros
Notifications
Dependencies
Advanced
ORACLE
DATABASES
TEST,PROD
LINUX
CPU_CRIT
90%
Set
CPU_WARN
80%
Set
LOAD_CRIT
3,3,3
LOAD_WARN
2,2,2
MEMORY_CRIT
95,50
MEMORY_WARN
90,20
SNMPCOMMUNITY
$SNMPCOMMUNITYREAD$
STORAGE_CRIT
95%
Set
STORAGE_WARN
90%
Set
SSH
SSHPORT
22

Packs ?

Packs = fichier zip avec tout ce dont vous avez besoin sur un sujet particulier (comme Linux, Windows ou EMC)

- Fichiers cfg (templates, commands, services, discovery, …)
- Templates de graphiques (PNP ou Graphite)
- Images (pour que votre UI soit jolie :))
- Fichier .pack file (json, descriptif)
- Pas (encore) d'installation de plugin

Heu... pourquoi créer un .zip avec tout ça?

- A. Le déplacer dans /dev/null
- B. Le partager!
- C. Me l'envoyer pour que je lance un outil “open core” comme NagiosXI
- D. Obiwan Kenobi

Partager via community.shinken-monitoring.org (démon hostd, disponible dans les sources Shinken)

Créer un tel fichier zip est aussi facile que de lancer :

$ shinken-packs -c -p /path/to/linux.pack

Et pour l'envoyer :

$ shinken-packs -u -k APIKEY -z /tmp/linux.zip

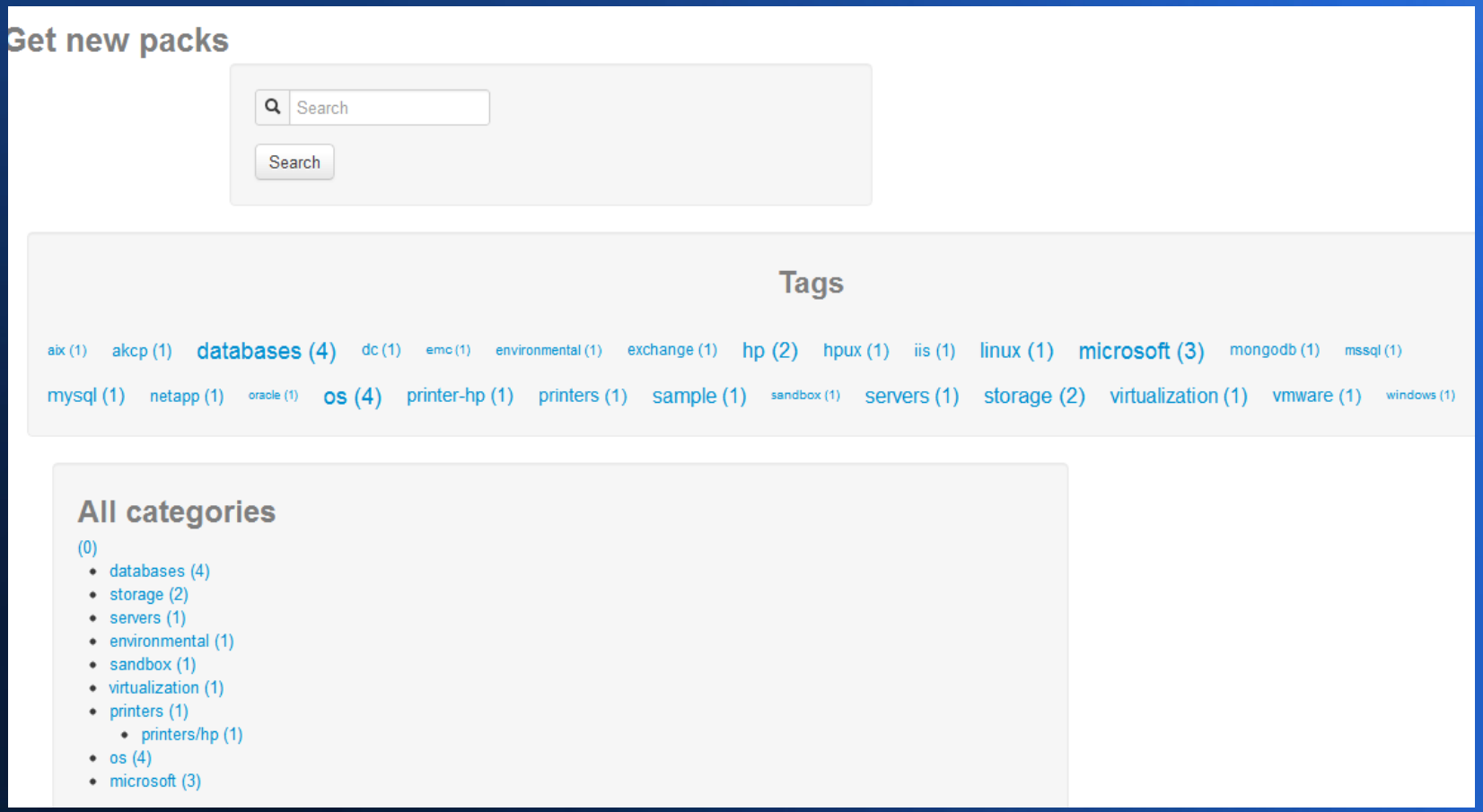
(une clé api est générée dès que vous êtes enregistrés sur le site)

Et pour les récupérer?

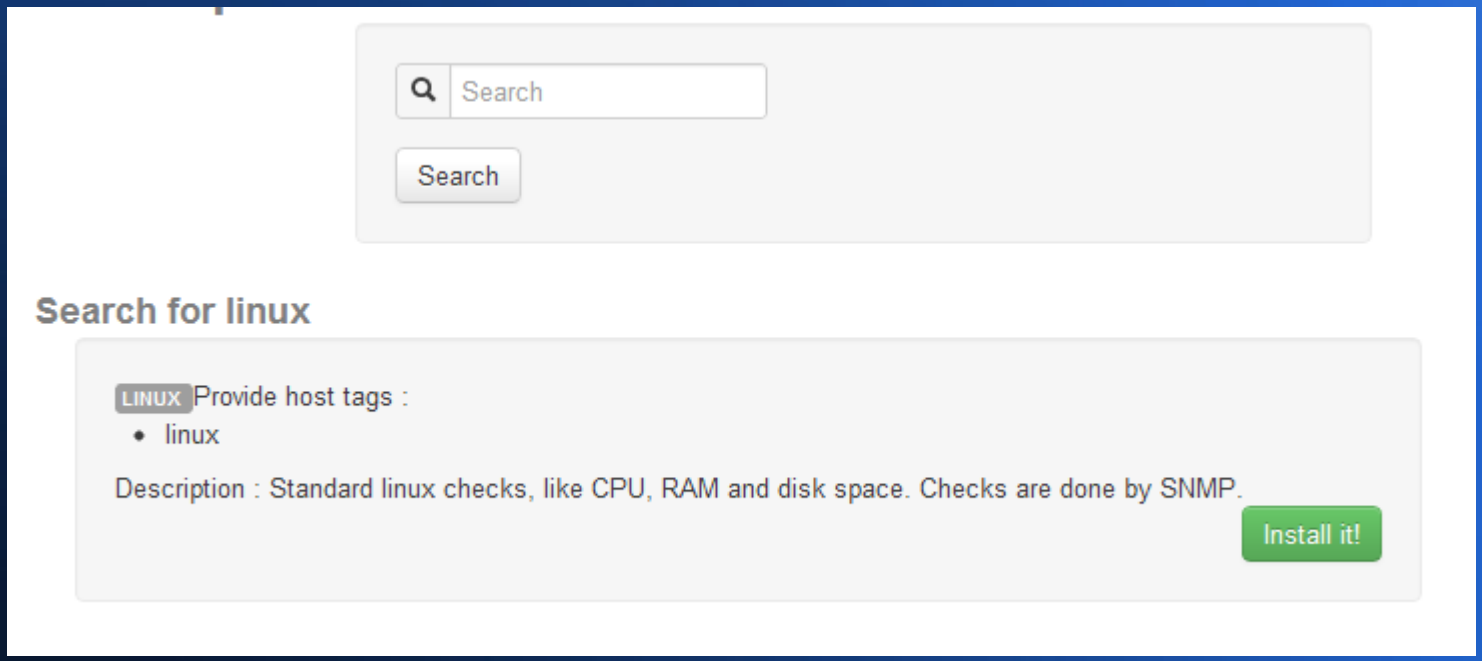
Sur le site :


Pack linux
Publisher: naparuba
Description: Standard linux checks, like CPU, RAM and disk space. Checks are done by SNMP.
Documentation: Link
Download it!
Host tags
LINUX
Services linked
Load
Disks
Memory
Cpu
NetworkUsage

Ou directement depuis l'UI sKonf :



Depuis sKonf :



Au final ?

- L'architecture est adaptable aux grands environnements
- Beaucoup d'améliorations par rapport à Nagios™®
- La WebUI est géniale, sKonf le sera prochainement
- Des triggers à la Zabbix arrivent! (corrélation&KPI)
- Lancement de services professionnels autour du projet :)

Merci

Des questions?