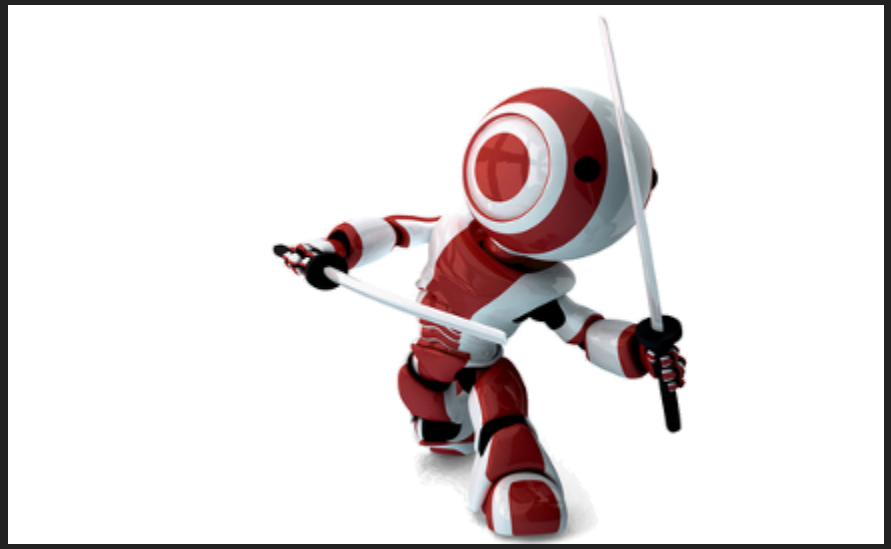
SUPERVISION AVEC SHINKEN



- Présentation xstra 26/1/16
- Olivier Benzerara

INTRODUCTION

- Historique
  - Développement au point mort de nagios
  - Un architecture obselète, non redimensionnable
  - Une base de code vieillissante
  - un existant incontournable (plugins)
- Genèse
  - Un proof of concept [Gabès(2010)]
  - une ré-implémentation de nagios et non un fork
  - Une base de code agile en python
  - un développement plus ouvert (github)

CAHIER DES CHARGES

- Test des protocoles standards (HTTP, SMTP, IMAP, LDAP...)
- Intégration de test locaux (Sondes, jobs cron)
- Coupler supervision/metrologie (minimiser les requetes)
- Interface Web de visualisation (Auth Apache)
- Minimiser les alertes (dépendances, escalades)
- Configuration en mode texte (versions)
- Produit Open-Source, modulaire
- Maitrise du code source (python)

PARADIGM

- La supervision technique (système) *host*
  - Elle va consister à surveiller le réseau, l'infra,...
- La supervision applicative *service*
  - Cette partie consiste à surveiller les applications.
- La supervision métier *business rules*
  - Consiste à surveiller les processus métiers.
- La supervision de la sécurité *event*
  - Surveillances des attaques contre le système

FONCTIONNALITÉS

Base : idem Nagios Core (mais modulaire)

Packs « natifs » -> **module**

- Notion d''impact (criticité)
- Business Process
- WebUI
  - Dashboard
  - Visualisation perfdata...
- LiveStatus
- Snapshots
- Découverte automatique

Packs « contributions » -> **pack**

- VMWare (migration aware)
- check_nwc_health (SNMP)

APPROCHE DEVOPS

- Coeur de la supervisation en package RPM/DEB
- Modules et packs additionels
- CLI pour la gestion ("shinken install webui2")
- un site web **shinken.io** contenant + de 142 packages additionels

ETAT DES LIEUX

	Comptabilité avec VMware	Paramétrage, Modularité, ajout des plugins	Déploiement sur système distribué et hétérogène	Réactivité de la communauté et équipe de développement	Fonctionnalité de supervision (quels équipements et quels facteurs sur ces équipements)	Performance	Interface web, vue custom	Platform (Win/Lin)
Nagios	+	-+++	++	+-	+++	++ (rapide car écrit en C)	++ (interface de base)	+ (Linux)
Shinken	+	-+++	+++ (idéal pour distribuer)	++	+++	+++ (efficace car optimisé)	++ (plus moderne)	++
EON Eyes Of Network (nagios+cacti+..)	+	+-++	++	-+	+++	+ +	++	- (os dédié centos)
Centreon (Nagios)	+	-++	++	++	+++	+ +	++	
ZABBIX	+	-+-	++	++	+++	++	++	++
openNMS	+	-+-	++	-+	++	+	+-	++
VIGILO (Nagios)	+	-+-	++	--	+++	+	+ +	+ (Linux)

DAEMON

- La structure de supervision a été decoupé en plusieurs services pour les raisons suivantes:
  - Architecture multisite
  - Haute disponibilité
  - monté en charge
  - qualité de service

ARBITER

Lit la configuration, la sépare et l'affecte au(x) scheduler(s). Il vérifie régulièrement que le(s) scheduler(s) fonctionnent correctement et affecte la configuration d'un scheduler en panne à un autre scheduler. Il reçoit les actions de supervision (commentaire, acquittement, …) et les dirige vers le scheduler correspondant.

SCHEDULER

Gère une partie de la configuration. Dispatche les tests à effectuer sur les pollers et les actions sur les reactionners. Le scheduler est aussi responsable du traitement du résultat, de la corrélation et de l'activation des actions (notifications, reprise sur incident).

POLLER

Lance les tests et transfère le résultat au scheduler. Un scheduler peut être associé à un domaine de supervision spécialisé (Windows par exemple) ou à un environnement (poller dédié à un site géographique par exemple).

REACTIONNER

Responsable de la notification et des reprises sur incident (« event handler »).

BROKER

responsable des données et de leur stockage. Il existe plusieurs brokers, chacun répondant à un besoin particulier : un broker dédié aux données de performance (pour Graphite par exemple) ou un broker dédié au stockage des données de supervision temps réel pour une interface web particulière.

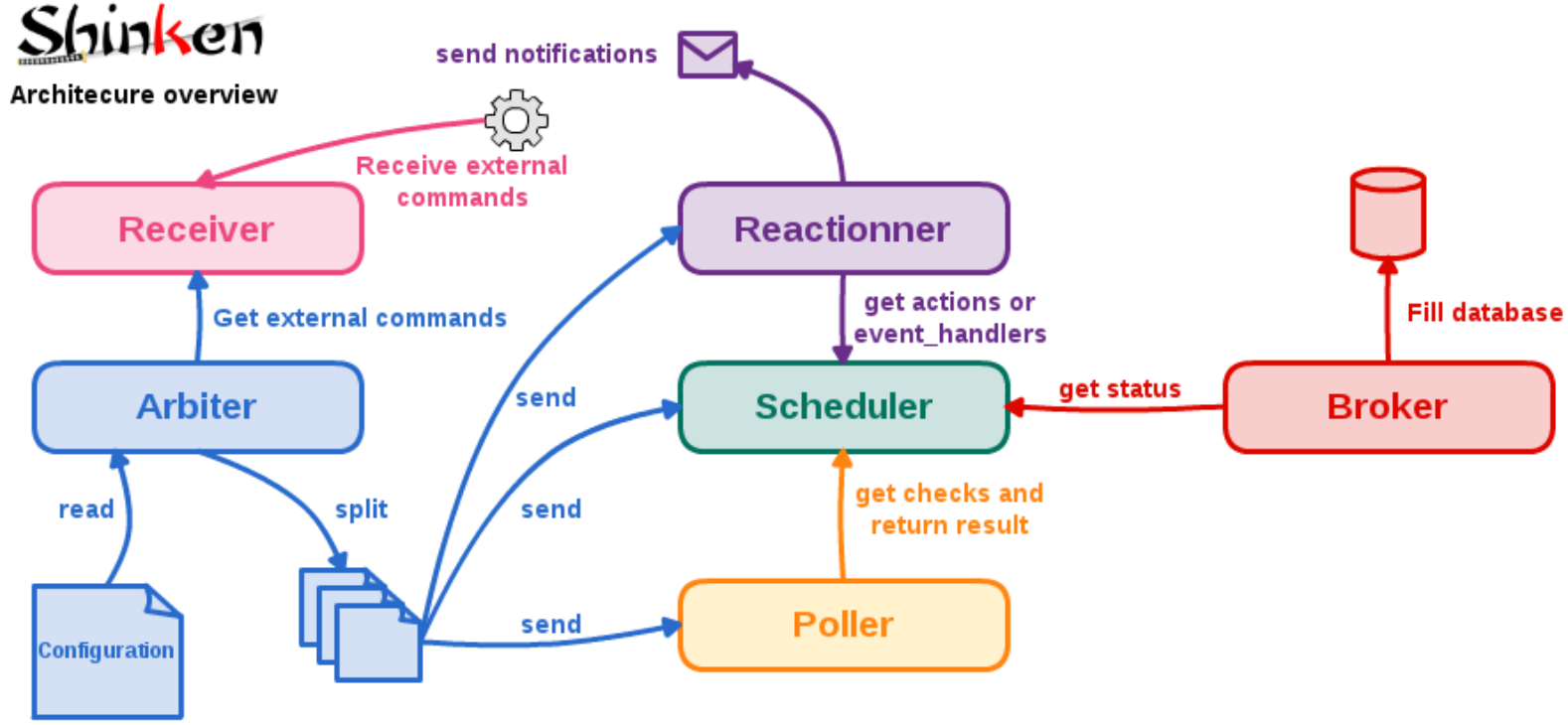
RECEIVER

reçoit les données de supervision passives et les transfert au scheduler correspondant.

TOPOLOGIE

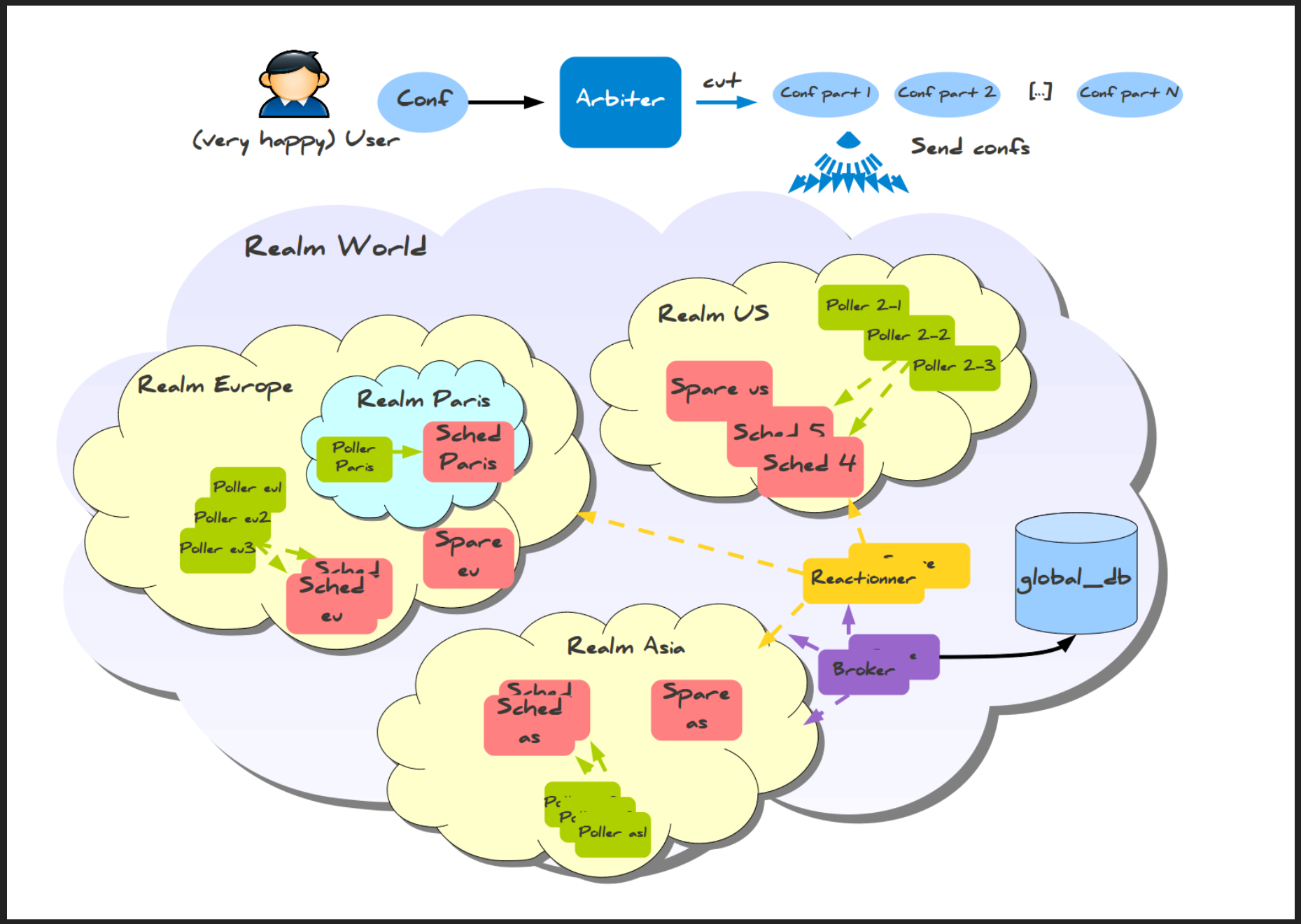
Shinken

Architecure overview

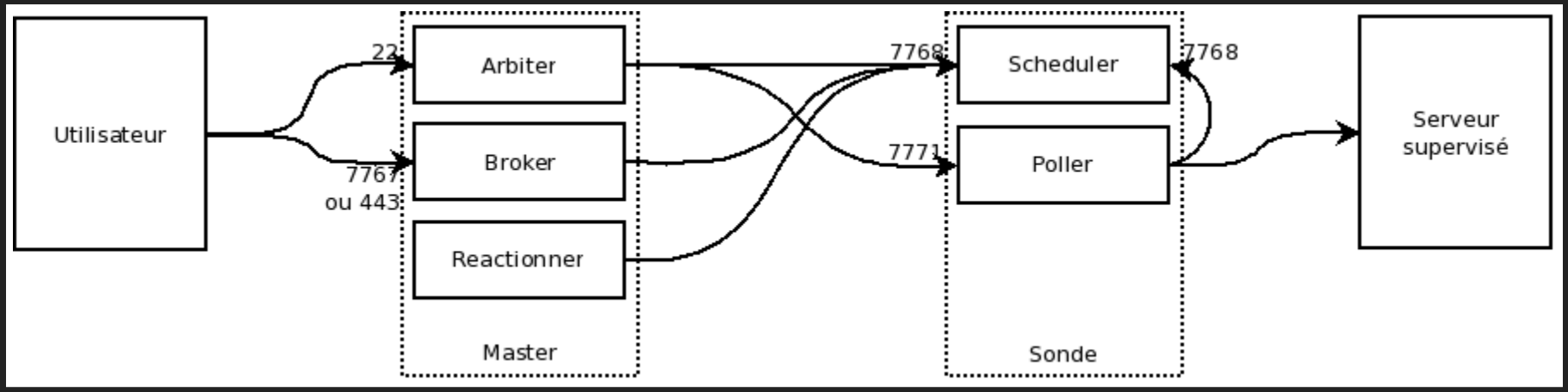


- Arbiter : manage and dispatch configuration, check health of overall architecture, receive external commands (acknowledgement for exemple)
- Scheduler : manage scheduling queue
- Poller : execute checks and return results to the scheduller
- Broker : consume status and output them in different ways (fill database, flat files ....). Provide an API to external tools such as monitoring console.
- Reactionner : send notification the way you want (email,sms, write rss, instant messenging) or execute event_handlers.
- Receiver : receive external commands such as passive checks result or acknowledgement (this module is optional).

REALM


(very happy) User
Conf
Arbiter
cut
Conf part 1
Conf part 2
[...]
Conf part N
Send confs
Realm World
Realm US
Poller 2-1
Poller 2-2
Poller 2-3
Spare us
Sched 5
Sched 4
Realm Europe
Realm Paris
Poller Paris
Sched Paris
Poller eu1
Poller eu2
Poller eu3
Sched eu
Spare eu
Reactionner
global_db
Realm Asia
Sched as
Spare as
Broker
Poller as1

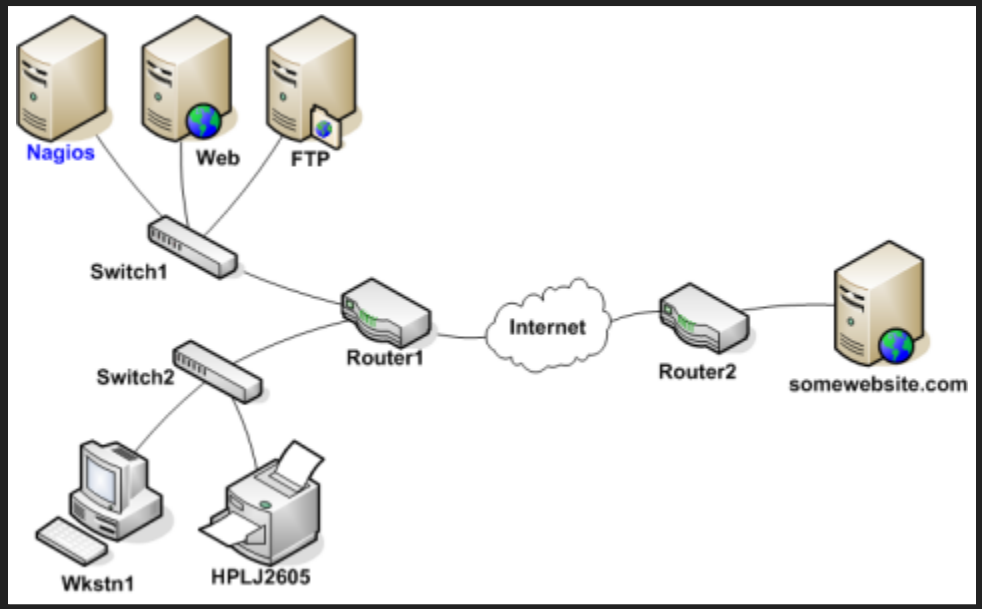
ARCHITECTURE DISTRIBUÉE



ACQUISITION DES HOSTS

- Network discovery
- importation de GLPI
- Migration de nagios

DÉPENDENCES




```
define host{
  host_name     Web
  parents     Switch1
}
```

BUSINESS RULES

```
define service {
       host_name meta
       service_description Web cluster
       check_command bp_rule!g:web,g:HTTPS?
       business_rule_service_notification_options n
       ...
       }
```

NOTIFICATIONS AND ESCALATIONS

```
define escalation{
    escalation_name   To_level2
    first_notification_time    60
    last_notification_time     120
    contact_groups    level2
}

define escalation{
    escalation_name     To_level_3
    first_notification_time    120
    last_notification_time     0
    contact_groups    level3
}
```

NOTIFICATIONS AND ESCALATIONS

```
define contact{
  contact_name                    happy_admin
  alias                           happy_admin
  email                           admin@localhost
  pager                           +33699999999
  notificationways                email_in_day,sms_the_night
}
```

MONITORING

- Monitoring Active Directory
- Monitoring Asterisk servers
- Monitoring DHCP servers
- Monitoring IIS servers
- Monitoring Linux devices
- Monitoring Linux devices
- Monitoring Linux devices via a Local Agent
- Monitoring Linux devices via SNMP
- Monitoring Routers and Switches
- Monitoring Network devices

MONITORING

- Monitoring Oracle databases
- Monitoring MySQL databases
- Monitoring Printers
- Monitoring Publicly Available Services
- Monitoring VMware hosts and machines
- Monitoring Microsoft Exchange
- Monitoring Microsoft SQL databases
- Monitoring Windows devices
- Monitoring Windows devices via NSClient++
- Monitoring Windows devices via WMI

FONCTIONNALITÉS EVOLUÉES

- External Commands
- Event Handlers
- Volatile Services
- Service and Host Freshness Checks
- Distributed Monitoring
- Redundant and Failover Network Monitoring
- Detection and Handling of State Flapping
- Notification Escalations
- On-Call Rotations

FONCTIONNALITÉS EVOLUÉES

- Monitoring Service and Host Clusters
- Host and Service Dependencies
- State Stalking
- Performance Data
- Scheduled Downtime
- Adaptive Monitoring
- Predictive Dependency Checks

INTERFACE GRAPHIQUE

- Thruk,webui,pnp4nagios,...
- graphite,rrdtools
- API REST

CONCLUSION

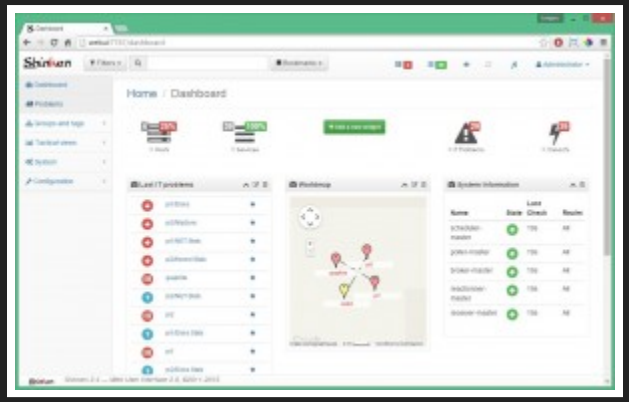
- Plus
  - Développeur francais
  - développement opensource (github)
  - communauté réactive (irc)
  - richesse des outils et fonctionalités
  - code modulaire et éditable
  - webui agréable / métrologie

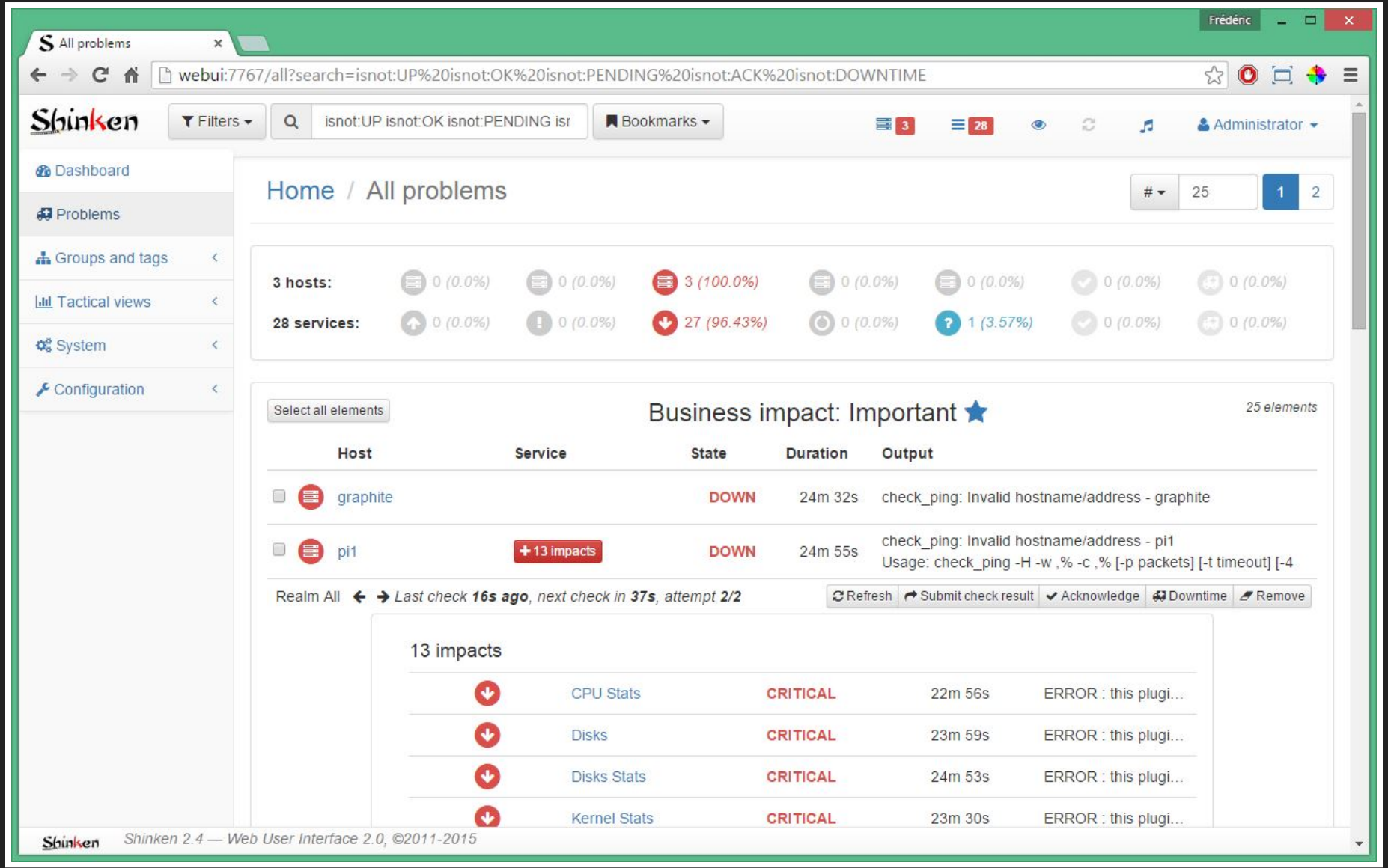
CONCLUSION

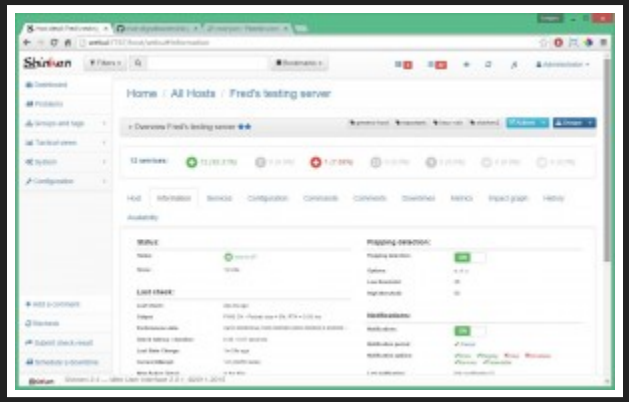
- Moins
  - maturite (2011)
  - Installation modulaire
  - communaute limite
  - approche devops , cela plait ou pas ?
  - business model ?

QUESTIONS ?

WEBUI




Frédéric
All problems
webui:7767/all?search=isnot:UP%20isnot:OK%20isnot:PENDING%20isnot:ACK%20isnot:DOWNTIME
Shinken
Filters
isnot:UP isnot:OK isnot:PENDING isr
Bookmarks
3
28
Administrator
Dashboard
Problems
Groups and tags
Tactical views
System
Configuration
Home / All problems
25 1 2
3 hosts: 0 (0.0%) 0 (0.0%) 3 (100.0%) 0 (0.0%) 0 (0.0%) 0 (0.0%) 0 (0.0%)
28 services: 0 (0.0%) 0 (0.0%) 27 (96.43%) 0 (0.0%) 1 (3.57%) 0 (0.0%) 0 (0.0%)
Select all elements
Business impact: Important
25 elements
Host Service State Duration Output
graphite DOWN 24m 32s check_ping: Invalid hostname/address - graphite
pi1 +13 impacts DOWN 24m 55s check_ping: Invalid hostname/address - pi1
Usage: check_ping -H -w ,% -c ,% [-p packets] [-t timeout] [-4
Realm All Last check 16s ago, next check in 37s, attempt 2/2
Refresh Submit check result Acknowledge Downtime Remove
13 impacts
CPU Stats CRITICAL 22m 56s ERROR : this plugi...
Disks CRITICAL 23m 59s ERROR : this plugi...
Disks Stats CRITICAL 24m 53s ERROR : this plugi...
Kernel Stats CRITICAL 23m 30s ERROR : this plugi...
Shinken Shinken 2.4 — Web User Interface 2.0, ©2011-2015


Home / All Hosts / Fred's testing server