Incident Response & Malware Analysis

~~Encore une histoire de boules~~

Bruno Dorsemaine

<contact@lpecheur.fr>

@l_pecheur

OSSIR Paris

9 septembre 2014

Please send a copy of this presentation to McAfee.

Introduction

- Stage de fin de master
- Orange/DSI Groupe
- Quarkslab

1 Introduction

2 Présentation générale
- Aperçu
- Architecture

3 Les scans

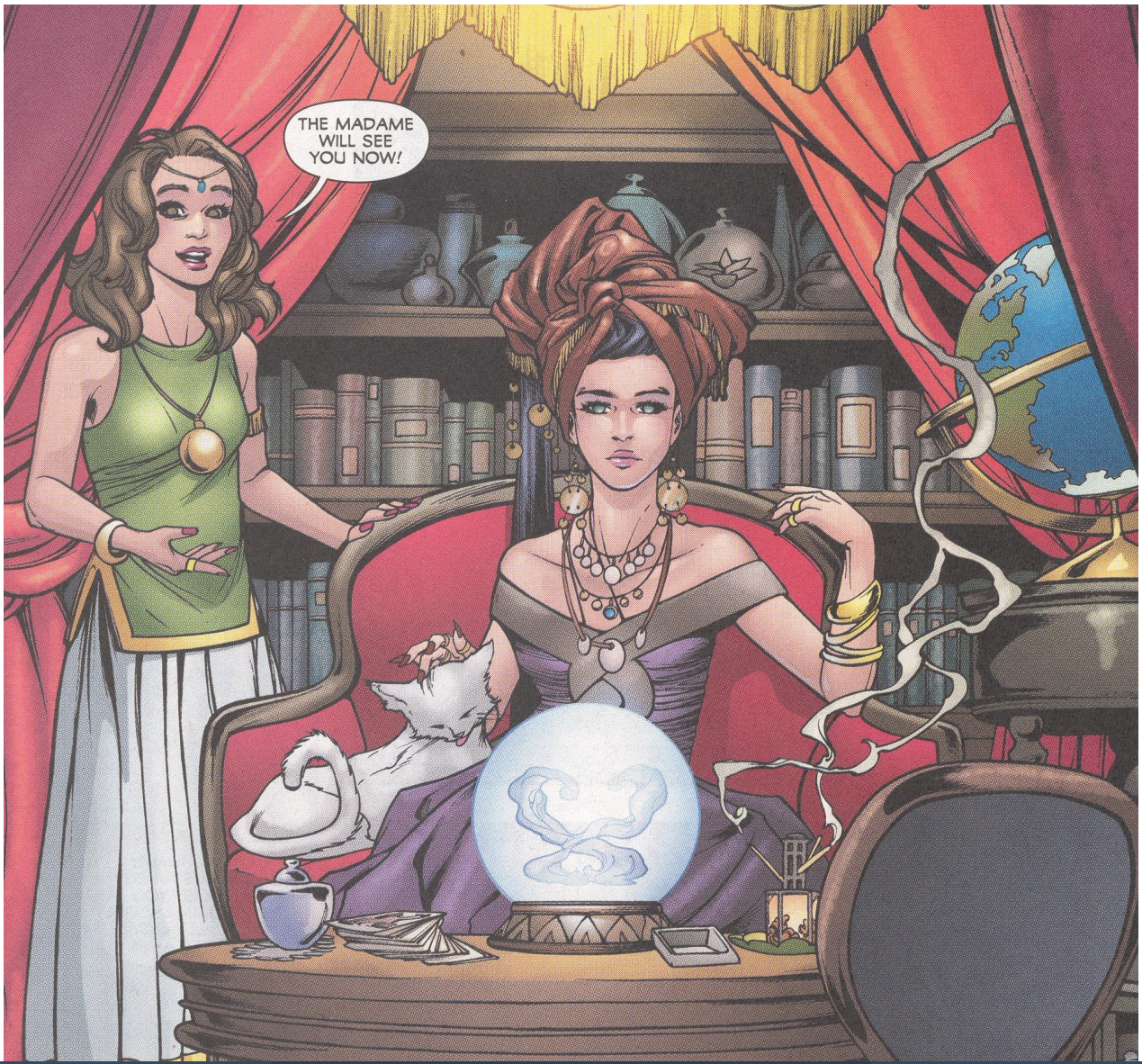
4 Le frontend

5 Le brain

6 Les probes

7 Conclusion

Ce qu'IRMA n'est pas



Mais qu'est-ce que c'est alors ?

Le projet

- Incident Response & Malware Analysis
- Airbus Group, CEA, DCNS, Govcert.lu, Orange & Quarkslab
- Licence Apache 2

Mais qu'est-ce que c'est alors ?

Le projet

- Incident Response & Malware Analysis
- Airbus Group, CEA, DCNS, Govcert.lu, Orange & Quarkslab
- Licence Apache 2

Et qu'est-ce que ça fait sinon ?

- Scans
- Contrôle sur les données soumises

Pourquoi IRMA ?

Produits	Web	Standalone	Libre	Contrôle sur les données
VirusTotal	✓	✗	✗	✗
Metascan	✓	✓	✗	?
AVCaesar	✓	✗	✗	✗
IRMA	✓	✓	✓/✗	✓

Pourquoi IRMA ?

Produits	Web	Standalone	Libre	Contrôle sur les données
VirusTotal	✓	✗	✗	✗
Metascan	✓	✓	✗	?
AVCaesar	✓	✗	✗	✗
IRMA	✓	✓	✓/✗	✓

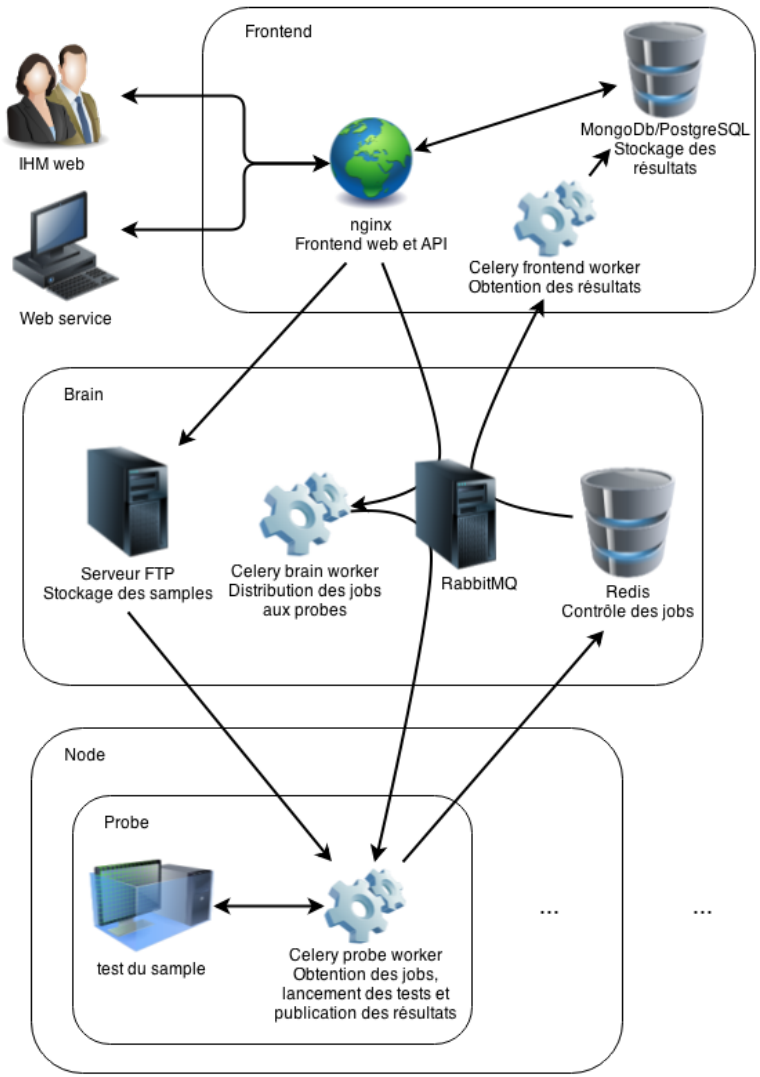
Objectifs du projet

- Alternative libre
- Développer une communauté
- Simple à utiliser
- Facile à étendre et personnalisable

Les technos

- Celery
- MongoDB
- PostgreSQL
- Redis
- libvirt/kvm
- Debian
- Windows 7
- nginx, node.js et Bottle
- Python

Architecture



Le Frontend

- IHM web, API
- Stockage des résultats et des samples
- Lancement de scans et affichage des résultats

Le Brain

- Gestion des jobs
- Quotas par utilisateurs

Les Probes

- Scans & analyses

1 Introduction

2 Présentation générale

3 Les scans
- Démo
- Comment ça marche ?

4 Le frontend

5 Le brain

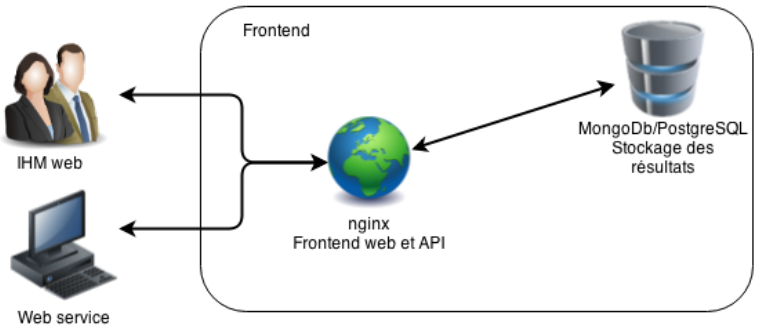
6 Les probes

7 Conclusion

Démo time !

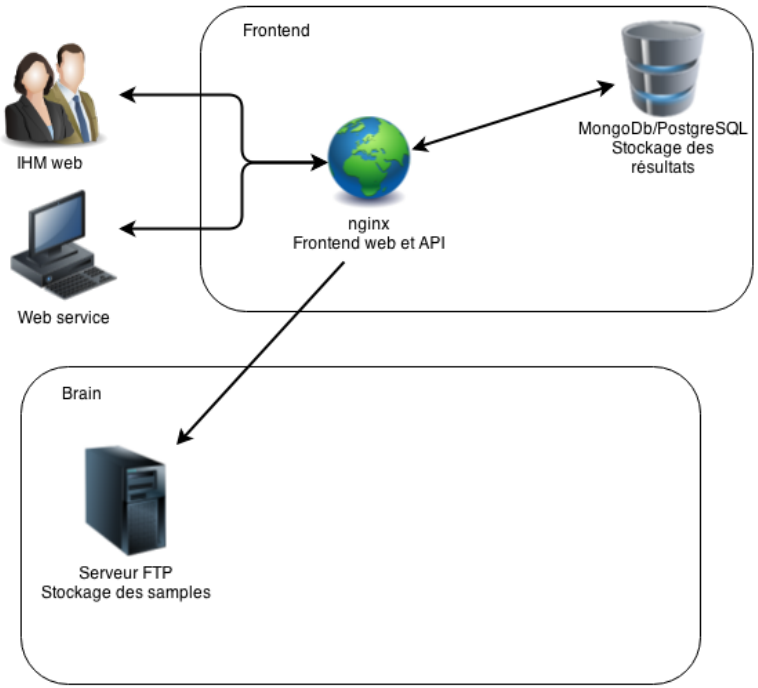
Démo !

Comment ça marche ?



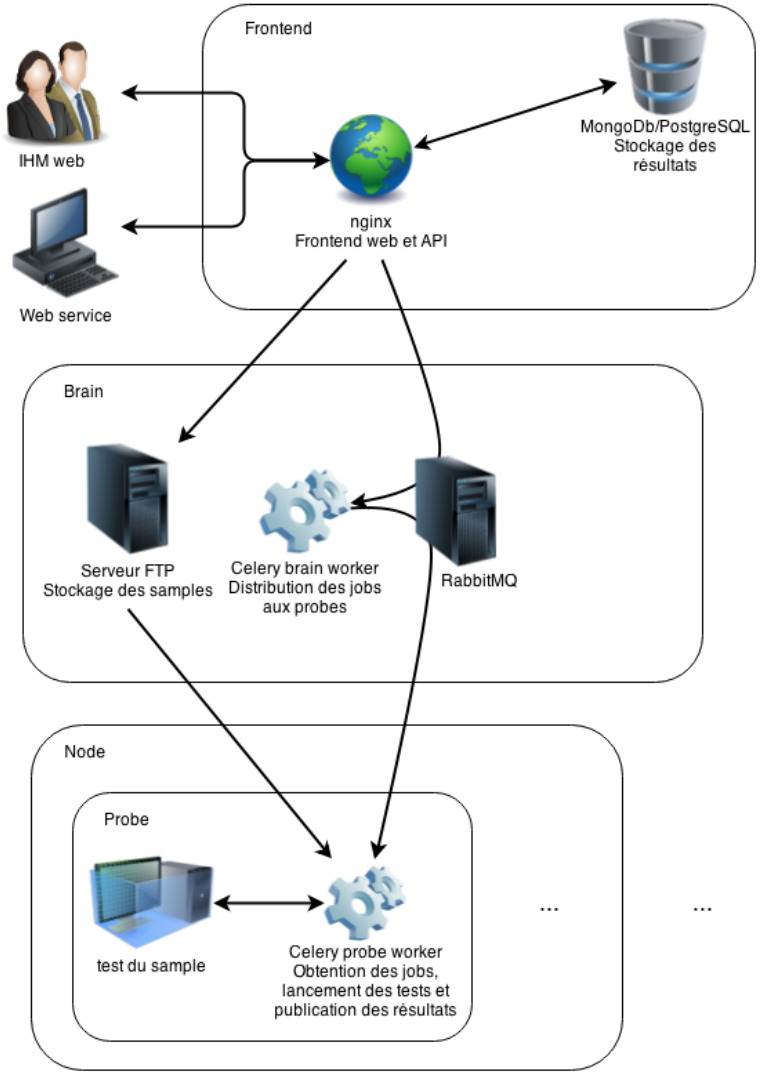
1. Création du scan dans le système

Comment ça marche ?



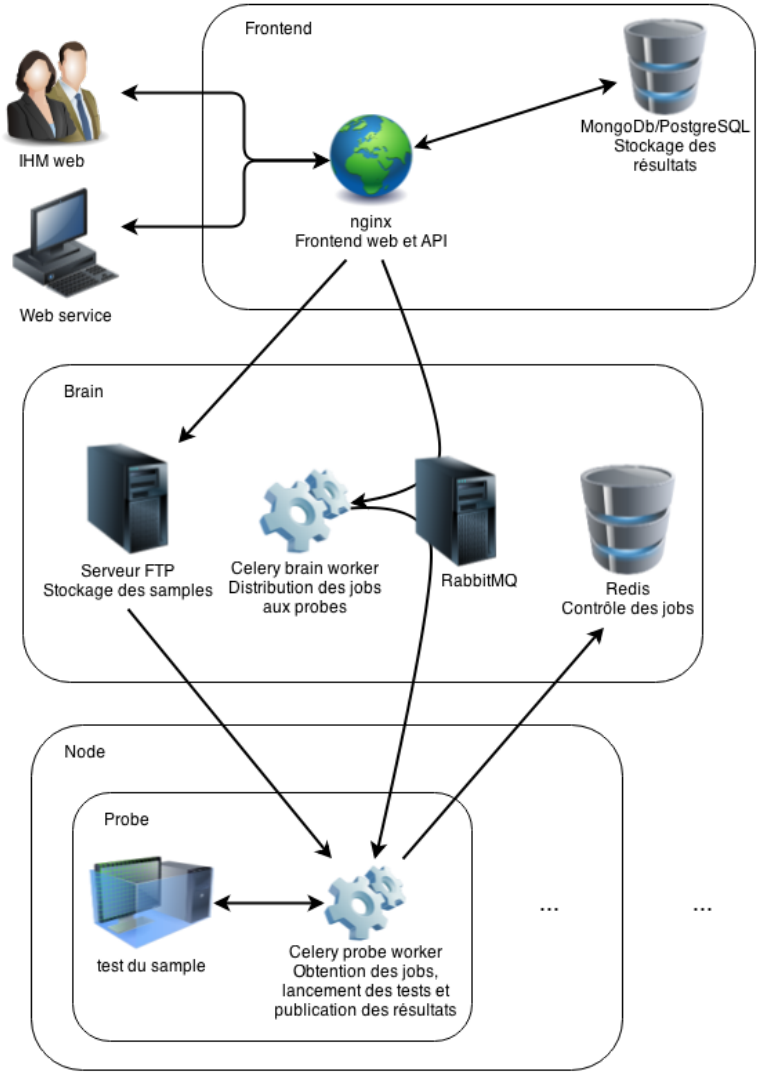
1. Création du scan dans le système
2. Envoi des fichiers sur le brain

Comment ça marche ?



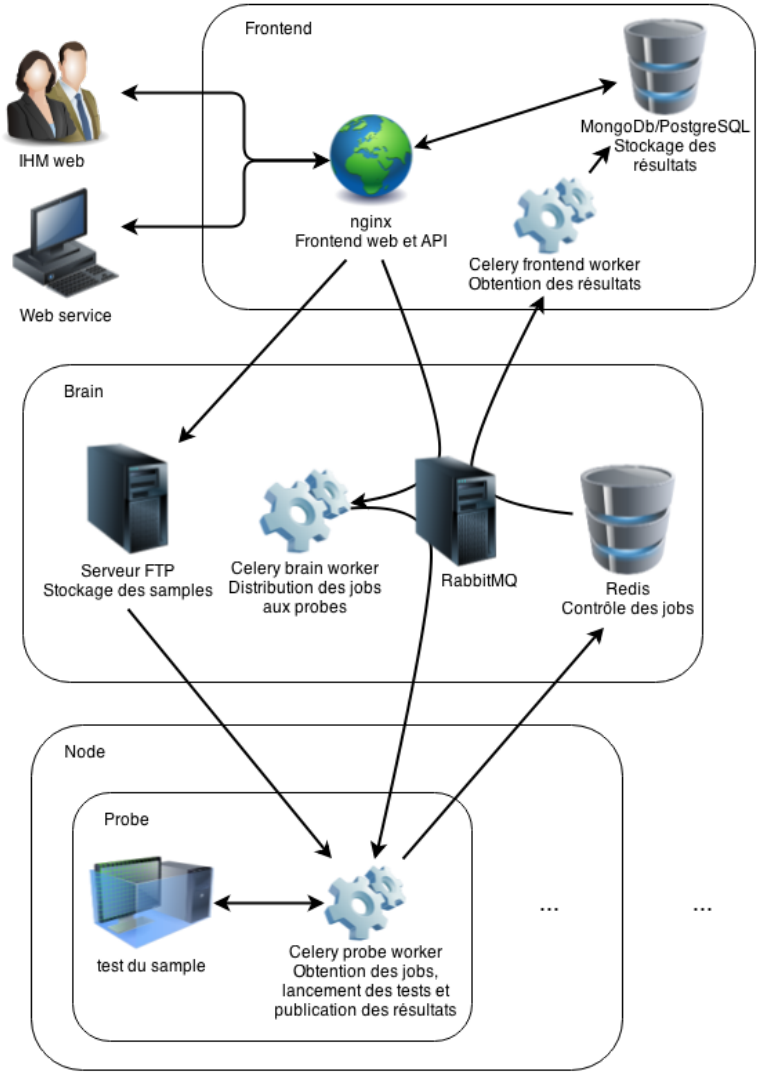
1. Création du scan dans le système
2. Envoi des fichiers sur le brain
3. Lancement des jobs sur les probes

Comment ça marche ?



1. Création du scan dans le système
2. Envoi des fichiers sur le brain
3. Lancement des jobs sur les probes
4. Mise à jour de l'état des jobs

Comment ça marche ?



1. Création du scan dans le système
2. Envoi des fichiers sur le brain
3. Lancement des jobs sur les probes
4. Mise à jour de l'état des jobs
5. Récupération de l'état des jobs et des résultats

1 Introduction

2 Présentation générale

3 Les scans

4 Le frontend
- L'API
- Le stockage des résultats

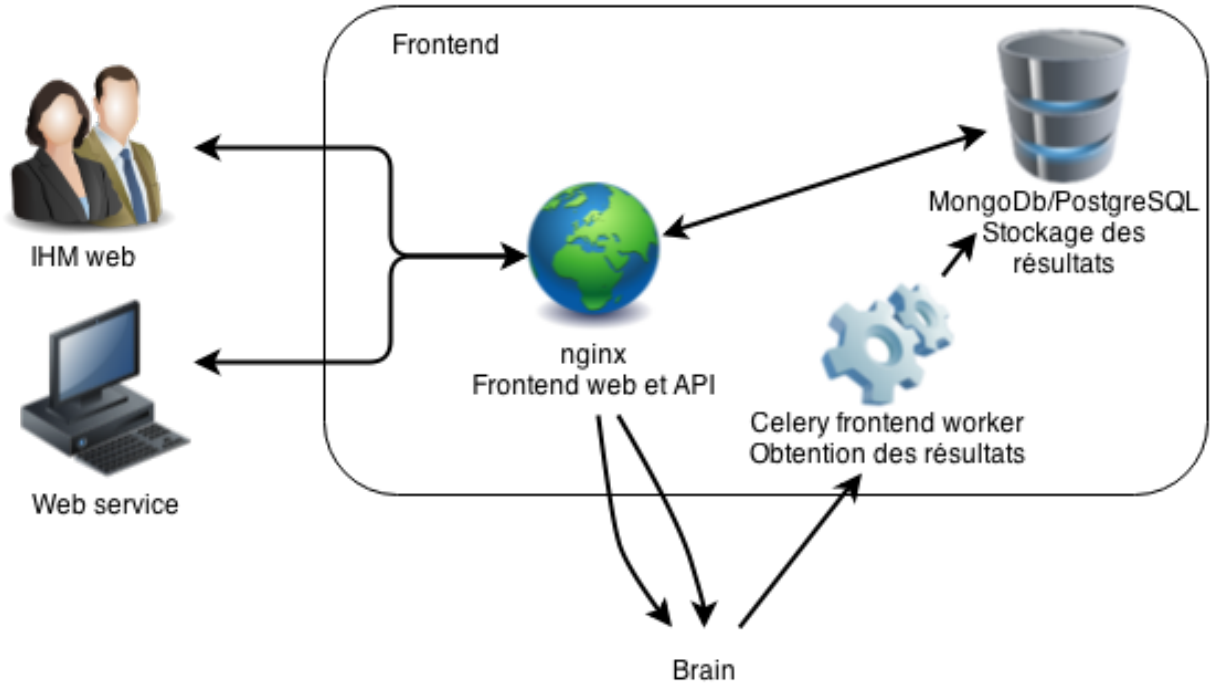
5 Le brain

6 Les probes

7 Conclusion

En détails

- Partie visible par l'utilisateur
- Lancement des scans
- Conservation des résultats et des samples
- Affichage des résultats



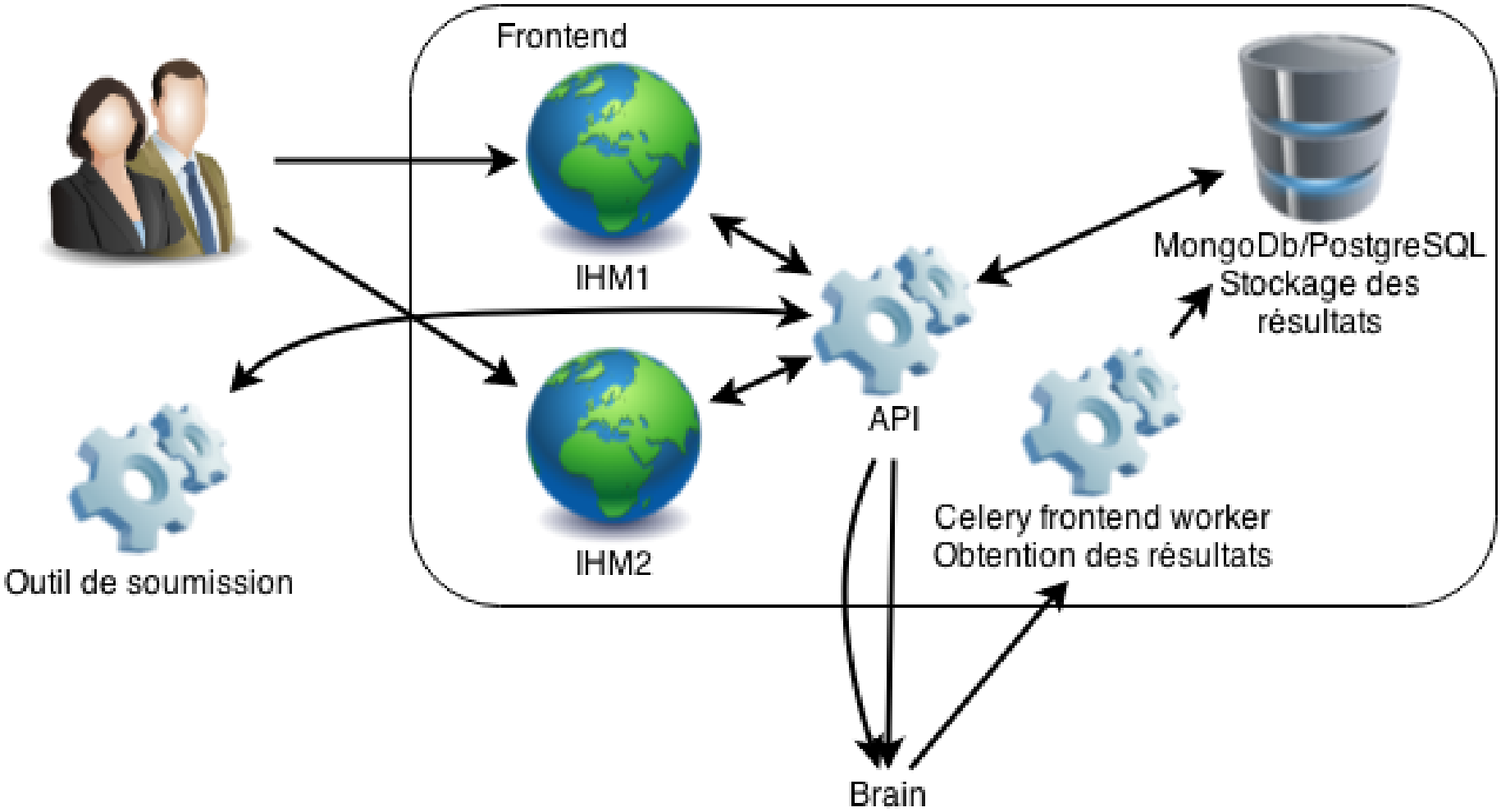
L'API

- Scans
  - création / ajout de samples / annulation
  - statut
- Résultats
  - par hash
  - par scan

Pourquoi c'est intéressant ?

- Ajouter des IHM
- Ajouter des moyens de soumission
- Interfaçage avec un autre outil

Pourquoi c'est intéressant ?



Le stockage des résultats

Données conservées

- Samples
- Historique des scans
- Résultats de chaque scan
- Derniers résultats pour chaque fichier
- Méta-données des scans

Le stockage des résultats

Données conservées

- Samples
- Historique des scans
- Résultats de chaque scan
- Derniers résultats pour chaque fichier
- Méta-données des scans

Différents choix

- NoSQL
- SQL
- Hybride

Premier modèle : NoSQL

Les plus

- Rapide
- Flexible

Les moins

- Relationnel
- Duplication

Premier modèle : NoSQL

Les plus

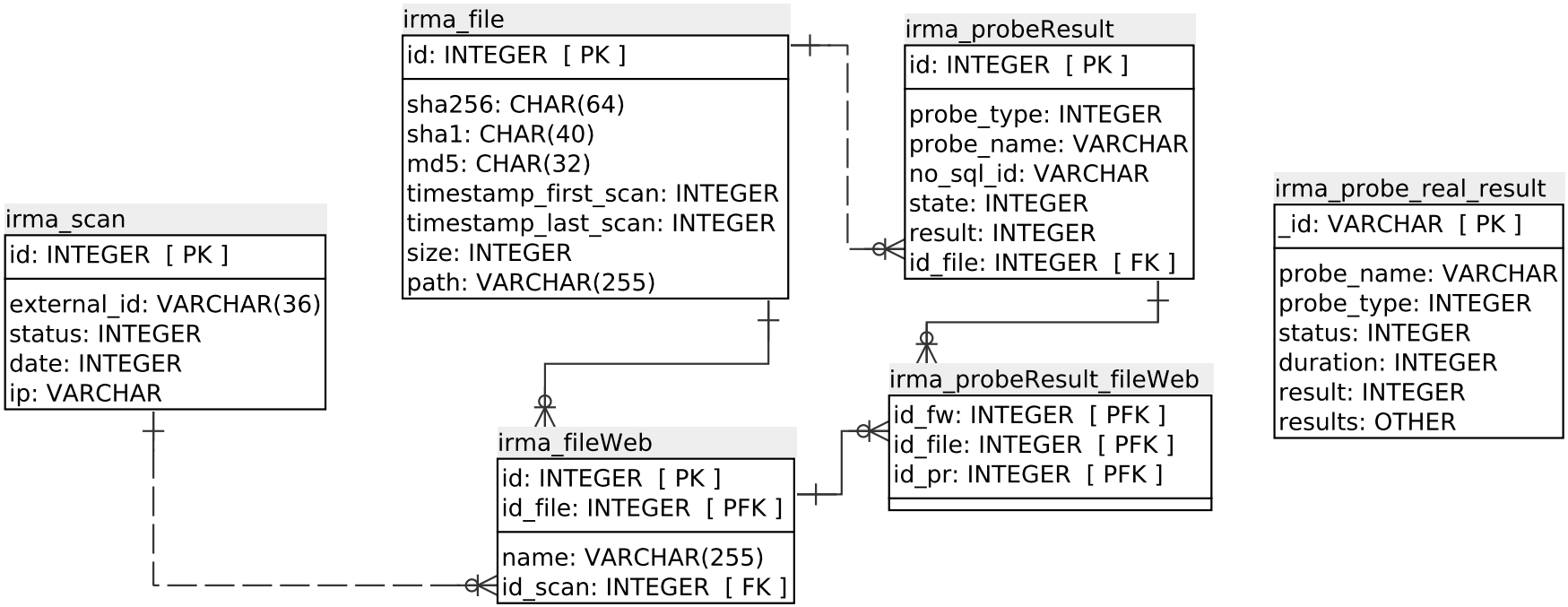
- Rapide
- Flexible

Les moins

- Relationnel
- Duplication

Très pratique en phase de développement mais limitant pour la suite.

Modèle actuel : SQL + NoSQL



Modèle actuel : SQL + NoSQL

Les plus

- Relationnel
- Très peu de duplication
- Rapide

Les moins

- Cohérence SQL/NoSQL à gérer côté code
- 2 SGBD à administrer

Modèle actuel : SQL + NoSQL

Les plus

- Relationnel
- Très peu de duplication
- Rapide

Les moins

- Cohérence SQL/NoSQL à gérer côté code
- 2 SGBD à administrer

Modèle souple pour conserver les résultats et performants pour les recherches et les statistiques.

1 Introduction

2 Présentation générale

3 Les scans

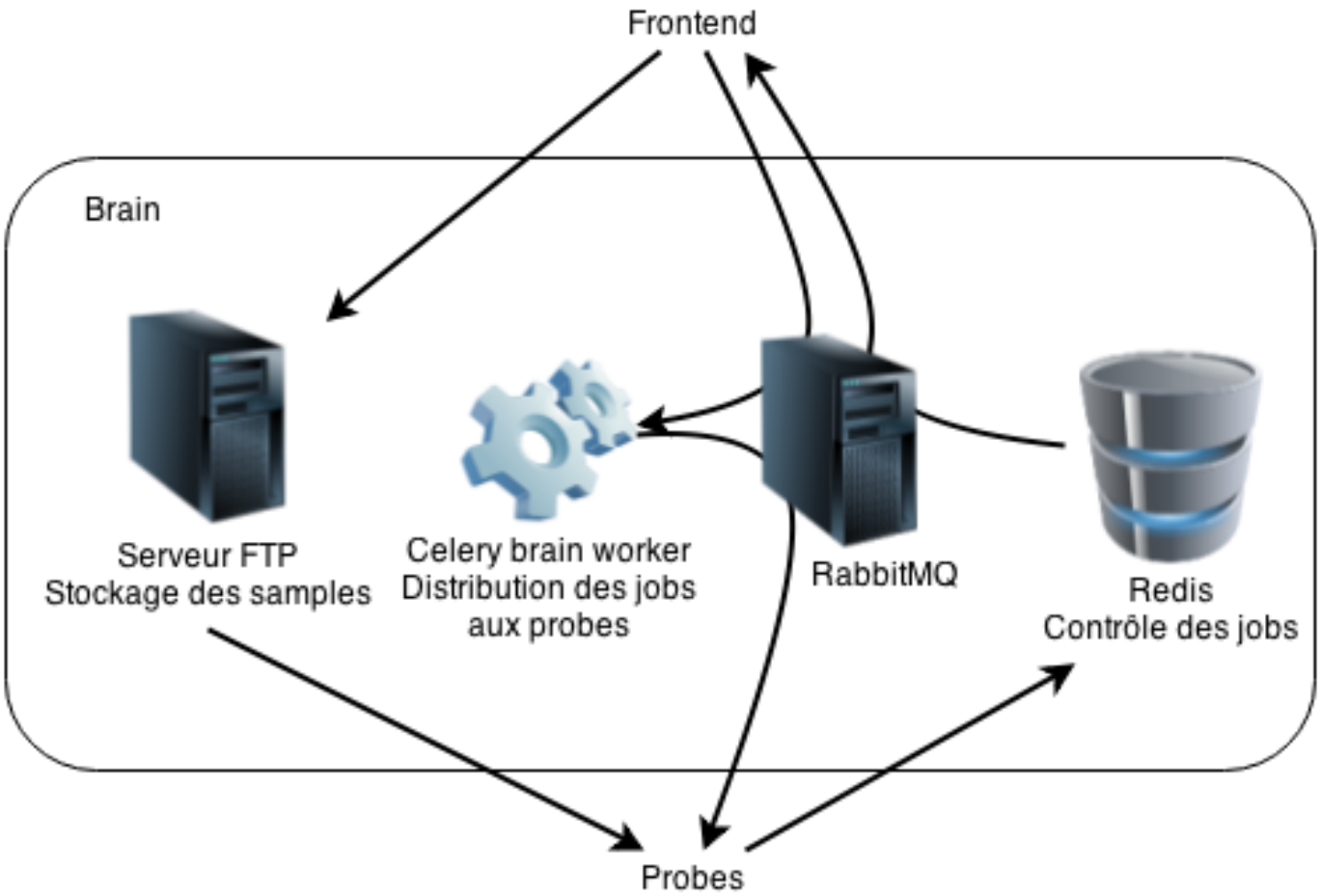
4 Le frontend

5 Le brain
- Celery/RabbitMQ
- Redis

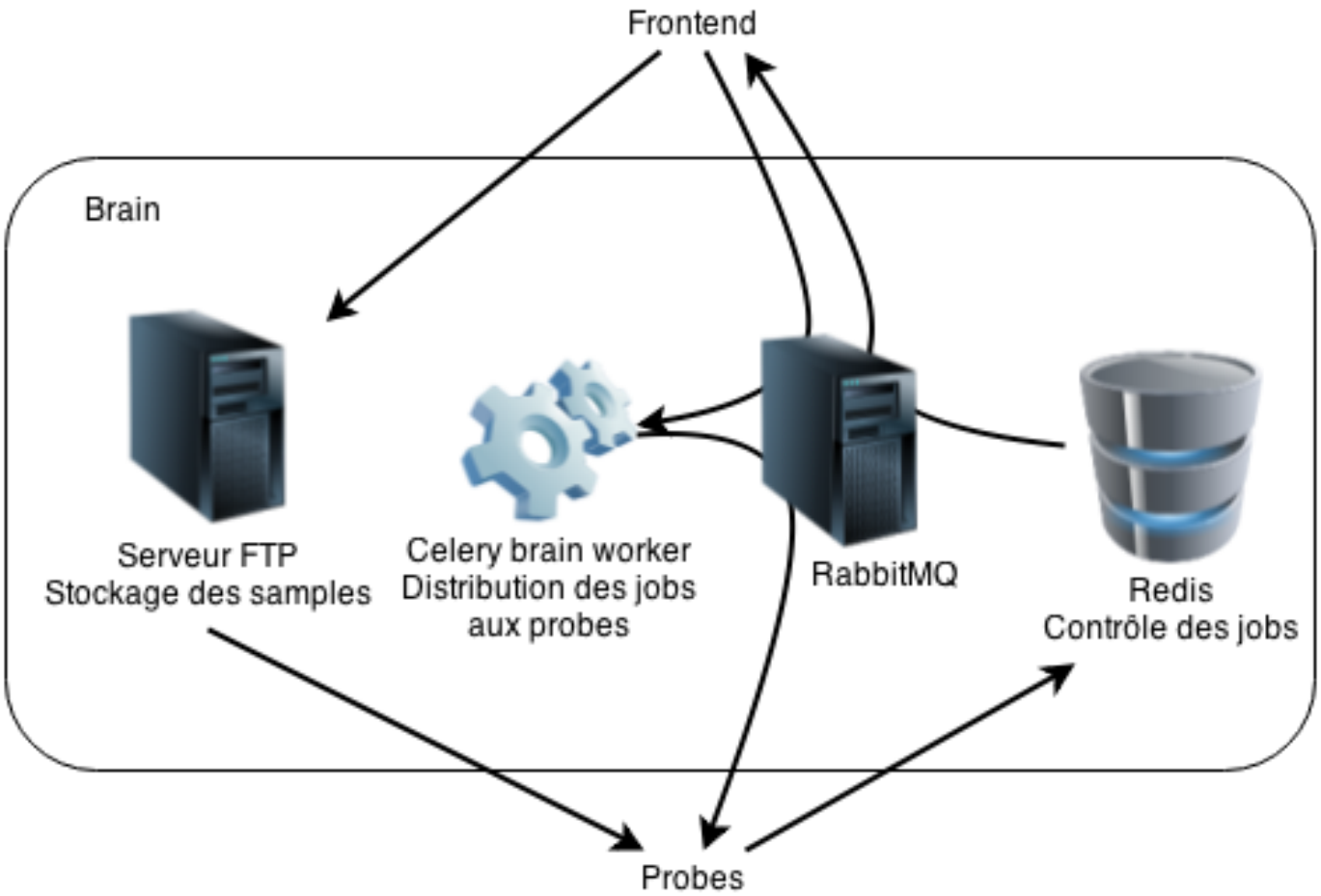
6 Les probes

7 Conclusion

En détails


Frontend
Brain
Serveur FTP
Stockage des samples
Celery brain worker
Distribution des jobs
aux probes
RabbitMQ
Redis
Contrôle des jobs
Probes

En détails



Pas de brain, pas d'IRMA.

Celery/RabbitMQ

Celery/RabbitMQ

C'est quoi alors ?

- RabbitMQ : transmission de messages (producteur/consommateur)
- Celery : surcouche pour RabbitMQ

Celery/RabbitMQ

C'est quoi alors ?

- RabbitMQ : transmission de messages (producteur/consommateur)
- Celery : surcouche pour RabbitMQ

Pourquoi ?

- Communication entre les différents composants
- Asynchrone

Redis

Késako ?

- SGBD
- clé / valeur

Redis

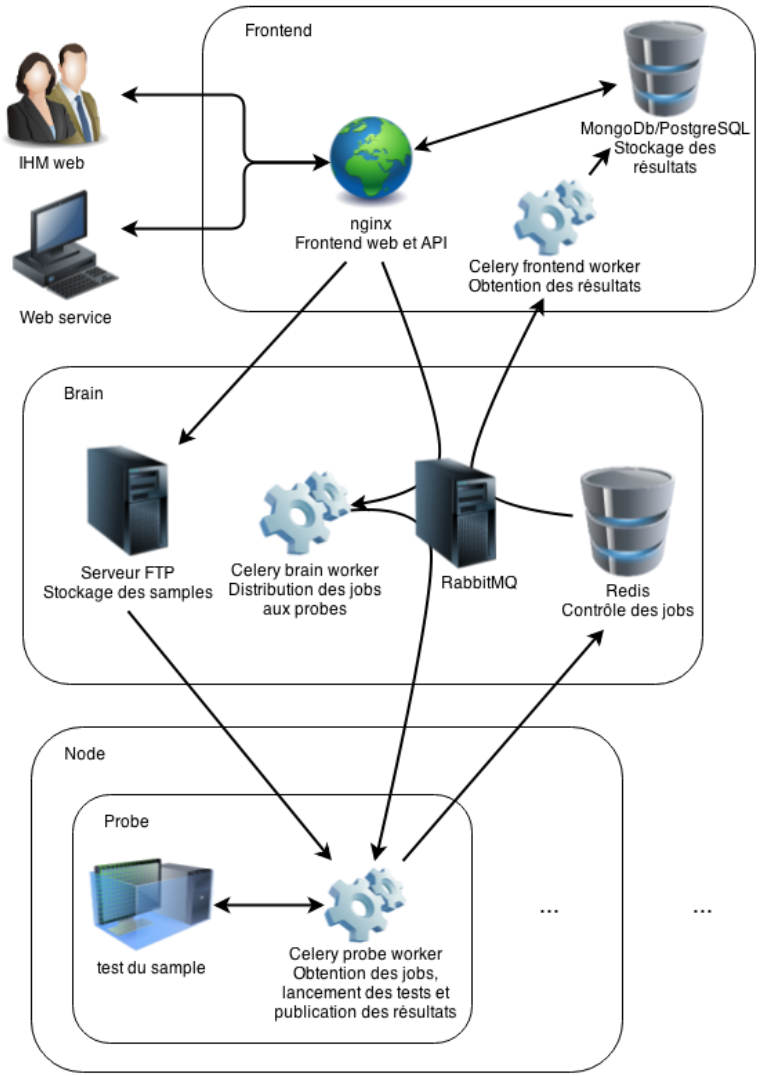
Késako ?

- SGBD
- clé / valeur

Pourquoi faire ?

- État des jobs affectés aux probes
- Correspondance identifiant du scan $\Leftrightarrow$ ensemble des jobs associés via le code du brain
- Clé $\Rightarrow$ identifiant de la tâche Celery associée au job
- Valeur $\Rightarrow$ état du job et résultat s'il est terminé

Retour sur l'archi



1 Introduction

2 Présentation générale

3 Les scans

4 Le frontend

5 Le brain

6 Les probes
- Les différentes probes
- L'ajout d'une probe

7 Conclusion

En détails

- Une probe $\Rightarrow$ une analyse spécifique
- GNU/Linux et/ou Windows
- Moteurs antiviraux
- Autres outils

Les différentes probes

Moteurs antiviraux MS Windows

- McAfee VirusScan Command Line
- Sophos
- Kaspersky Internet Security
- Symantec Endpoint Protection

Moteurs antiviraux GNU/Linux

- ClamAV
- Comodo Antivirus for Linux
- Eset Nod32 Business Edition
- F-Prot
- McAfee VirusScan Command Line

Les différentes probes

VirusTotal

- Probe GNU/Linux & MS Windows
- Envoi du hash uniquement

PE File Analyzer

- Probe GNU/Linux & MS Windows
- Vient de Cuckoo Sandbox
- Renvoie les sections, imports, exports... du binaire

National Software Reference Library

- Probe GNU/Linux
- Hashes de fichiers de différentes versions de MS Windows
- Maintenu par le NIST

Activation d'une probe déjà supportée

En général

1. Installation des binaires/libraries
2. Un tout petit peu de configuration (ou pas)
3. Redémarrage de Celery
4. Détecté par IRMA

Activation d'une probe déjà supportée

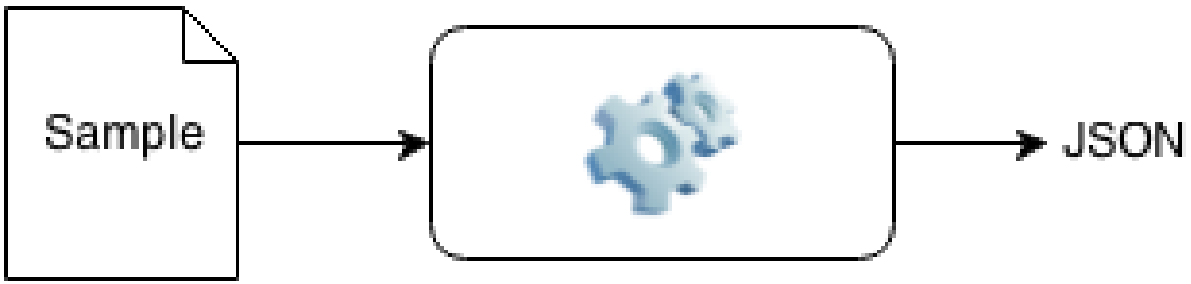
En général

1. Installation des binaires/libraries
2. Un tout petit peu de configuration (ou pas)
3. Redémarrage de Celery
4. Détecté par IRMA

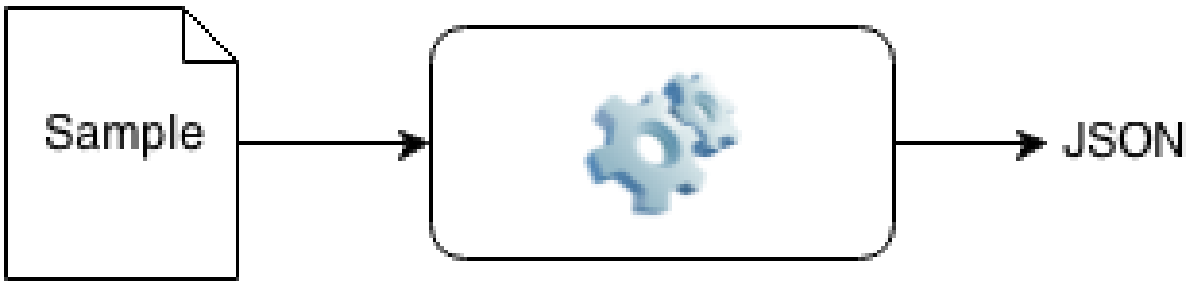
Exemple : ClamAV

1. `$ sudo apt-get install clamav-daemon`
2. `$ sudo freshclam`
   `$ sudo service clamav-daemon restart`
3. `$ sudo service celeryd.probe restart`

Création d'une probe



Création d'une probe



Architecture commune & générique

Création d'une probe

Le JSON

- Code de retour
- Nom de la probe
- Résultats
- Version de l'outil
- Durée
- Type de probe

Création d'une probe

Le JSON

- Code de retour
- Nom de la probe
- Résultats
- Version de l'outil
- Durée
- Type de probe

Au besoin, modification du frontend pour l'affichage

Détection des probes

- Système de plug-ins
- Automatique
- Basé sur la satisfaction des dépendances de chaque probe
- Configuration automatique de la *queue* Celery

Détection des probes : ClamAV

Dépendance de ClamAV : `clamdscan`

Détection des probes : ClamAV

Dépendance de ClamAV : `clamdscan`

1. Satisfaction de la dépendance

```
$ sudo apt-get install clamav-daemon
$ sudo freshclam
$ sudo service clamav-daemon restart
```

Détection des probes : ClamAV

Dépendance de ClamAV : `clamdscan`

1. Satisfaction de la dépendance

```
$ sudo apt-get install clamav-daemon
$ sudo freshclam
$ sudo service clamav-daemon restart
```

2. Détection et configuration automatique

```
$ sudo service celeryd.probe restart
```

Les dissecteurs

Chaque outil $\Rightarrow$ un format de sortie spécifique.

Les dissecteurs

Chaque outil $\Rightarrow$ un format de sortie spécifique.

Résultats conservés

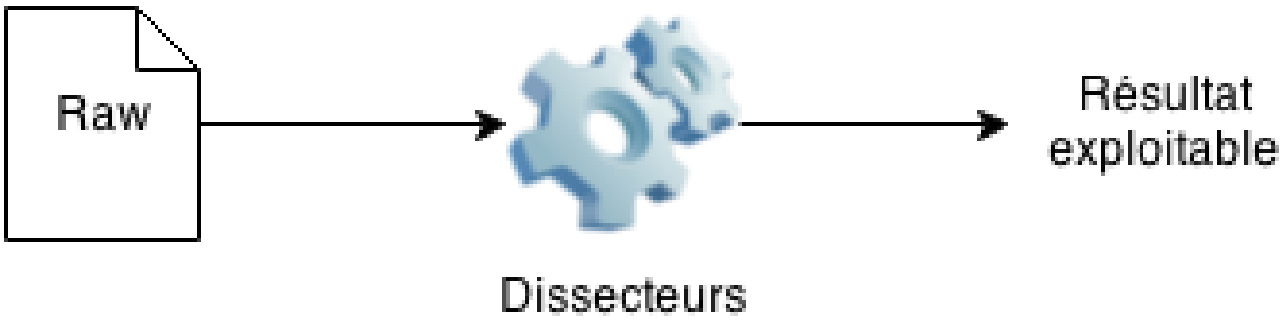
- le *raw*
- quelque-chose d'exploitable directement

Les dissecteurs

Chaque outil $\Rightarrow$ un format de sortie spécifique.

Résultats conservés

- le *raw*
- quelque-chose d'exploitable directement



Les dissecteurs : ClamAV

Raw d'un scan avec ClamAV :

```
root@debian:/# clamdscan /home/ftpuser/frontend/53ff00506e95520b8fa39a90/2d80c5f
0793c5520d2780157f296761972f7b02039585b14474ae7d9668f32f8
/home/ftpuser/frontend/53ff00506e95520b8fa39a90/2d80c5f0793c5520d2780157f2967619
72f7b02039585b14474ae7d9668f32f8: Win.Trojan.Agent-604924 FOUND

----------- SCAN SUMMARY -----------
Infected files: 1
Time: 1.933 sec (0 m 1 s)
root@debian:/# _
```

Les dissecteurs : ClamAV

Raw d'un scan avec ClamAV :

```
root@debian:/# clamdscan /home/ftpuser/frontend/53ff00506e95520b8fa39a90/2d80c5f
0793c5520d2780157f296761972f7b02039585b14474ae7d9668f32f8
/home/ftpuser/frontend/53ff00506e95520b8fa39a90/2d80c5f0793c5520d2780157f2967619
72f7b02039585b14474ae7d9668f32f8: Win.Trojan.Agent-604924 FOUND

----------- SCAN SUMMARY -----------
Infected files: 1
Time: 1.933 sec (0 m 1 s)
root@debian:/# _
```

Dissecteur :

```
r'(?P<file>.*): (?P<name>[^\s]+) FOUND'
```

Les dissecteurs : ClamAV

Raw d'un scan avec ClamAV :

```
root@debian:/# clamdscan /home/ftpuser/frontend/53ff00506e95520b8fa39a90/2d80c5f
0793c5520d2780157f296761972f7b02039585b14474ae7d9668f32f8
/home/ftpuser/frontend/53ff00506e95520b8fa39a90/2d80c5f0793c5520d2780157f2967619
72f7b02039585b14474ae7d9668f32f8: Win.Trojan.Agent-604924 FOUND

----------- SCAN SUMMARY -----------
Infected files: 1
Time: 1.933 sec (0 m 1 s)
root@debian:/# _
```

Dissecteur :

```
r'(?P<file>.*): (?P<name>[^\s]+) FOUND'
```

Résultat :

- `file` ⇒ /home/ftpuser/frontend/53ff00506e95520b8...
- `name` ⇒ Win.Trojan.Agent-604924

1 Introduction

2 Présentation générale

3 Les scans

4 Le frontend

5 Le brain

6 Les probes

7 Conclusion

Bientôt dans IRMA

Malware Analysis

- Nouvelles probes (comportementales...)
- Soumissions via URL
- Ajout d'un moteur de recherches
- ...

Bientôt dans IRMA

Malware Analysis

- Nouvelles probes (comportementales...)
- Soumissions via URL
- Ajout d'un moteur de recherches
- ...

Incident Response

- Scans privés
- Génération automatique de rapports
- Suivi de propagation d'un fichier
- Agent
- ...

Liens utiles

Documentation & installation

- irma.quarkslab.com
- Machines virtuelles de démo : irma.quarkslab.com/install.html

Code

- github.com/quarkslab/irma-frontend
- github.com/quarkslab/irma-brain
- github.com/quarkslab/irma-probe

Contact

- @qb_irma
- #qb_irma sur Freenode

Conclusion

IRMA en résumé

- Alternative à VirusTotal
- Simple d'utilisation
- Facile à installer
- Apache 2
- Extensible
- Une communauté à développer

Conclusion

IRMA en résumé

- Alternative à VirusTotal
- Simple d'utilisation
- Facile à installer
- Apache 2
- Extensible
- Une communauté à développer

Agenda

15 octobre, HITB Malaysia

HITB LAB : IRMA – An Open Source Incident Response & Malware Analysis Platform par A. Quint & F. Lone-Sang

Incident Response & Malware Analysis

~~Encore une histoire de boules~~

Bruno Dorsemaine

<contact@lpecheur.fr>
@l_pecheur

OSSIR Paris

Merci à Google pour toutes ces images sur lesquelles je n'ai pas de droits :)