Digital Forensics

Framework

Ossir – Groupe Paris
09/11/2010

Tables des matières

- Présentation d'ArxSys
- Présentation générale de DFF
- Présentation de l'API de DFF
- Démo
- Questions

Tables des matières

ArxSys

- Issue d'un projet de fin d'études
- Jeune Entreprise Innovante créée en 2009
- Éditeur de logiciel open-source d'investigation numérique

PARIS DEVELOPPEMENT

IRCGN

DFRWS

prix de l'innovation des assises 2010

oséo

Tables des matières

Pourquoi ?

- **Analyses forensiques :**

  Problèmes : *Incidents, Fraudes, ...*

  Réponses à apporter: *Qui ? Quand ? Quoi ? Ou ? Comment ?*

- **Besoin d'outils :**

  Open Source

  Multiplateforme

  Évolutif / Modulaire

Pour qui ?

Tous ceux qui font de l'investigation numérique, en particulier :

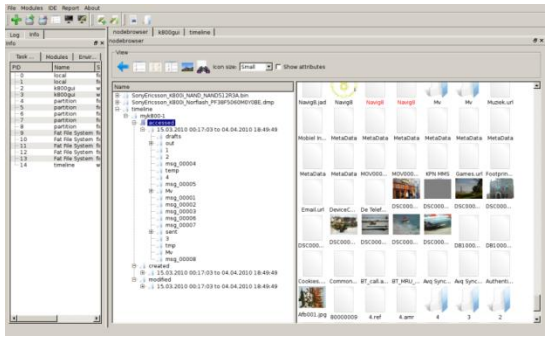
- Équipes sécurité des entreprises
- Enquêteurs & experts judiciaires
- Consultants en sécurité
- Étudiants

Comment ?

En entrée :

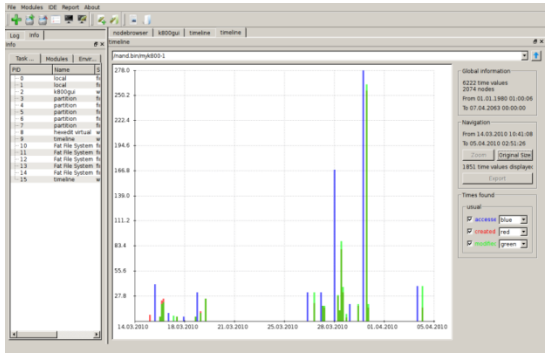
Supports : Disques Durs, Mémoires Flash / RAM

- Copies (Analyses mortes)
- Périphériques (Analyses “live”)
- Chargement d’une sauvegarde



Interfaces :

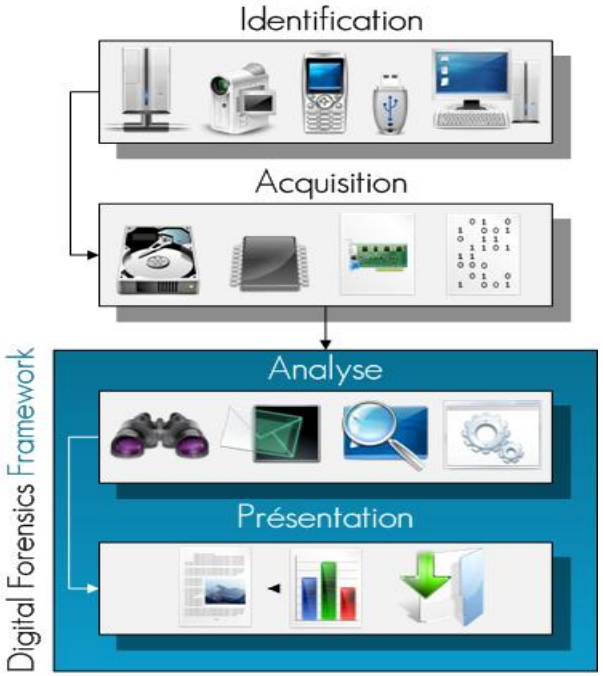
- Shell
- Graphique
- Interpréteur



Comment ?

Analyse : (Modules)

- Nodes : File System, VM, Volatility, phones, carver, ...
- Metadonnées : File System, embarquées, ...
- Visualiseurs : hexview, images, vidéos, ...
- Graph : timeline, type de fichiers, ...
- Réduction : filtres, hash, dictionnaires, …



Sortie : (Nodes, Favoris)

- Extraction
- Génération de rapports
- Sauvegarde

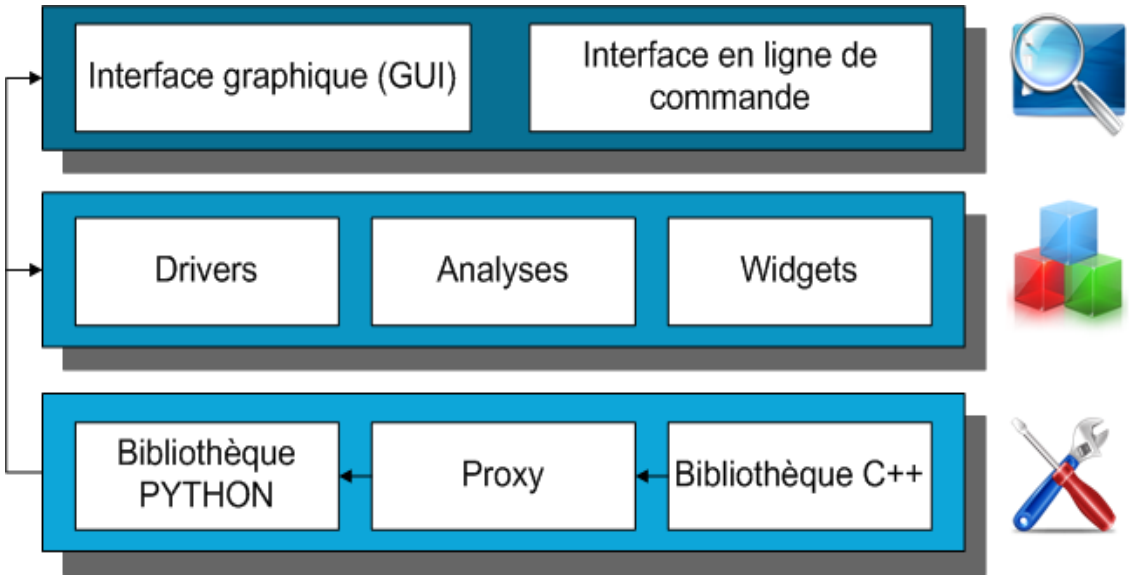
Tables des matières

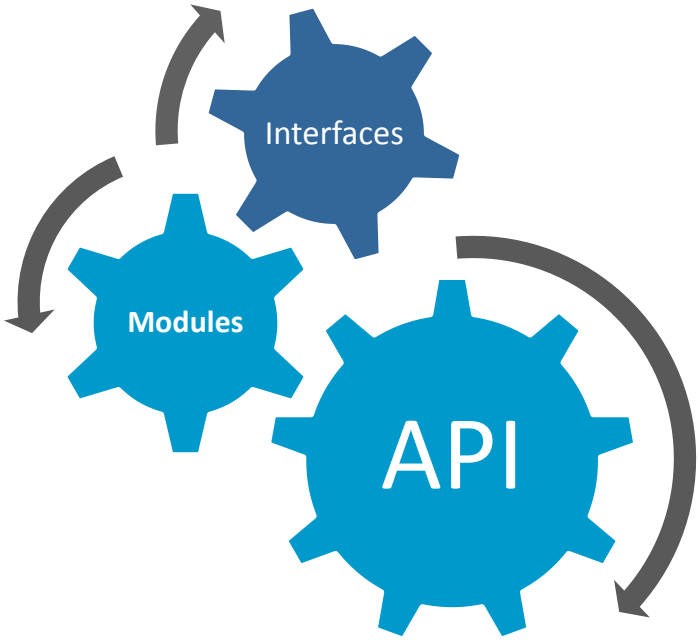
Tour d'horizon technologique

- Framework orienté objet
- Développé en Python et C++
- Interconnexion des langages via swig
- Interface Utilisateur Graphique en PyQt
- Chaîne de compilation avec Cmake

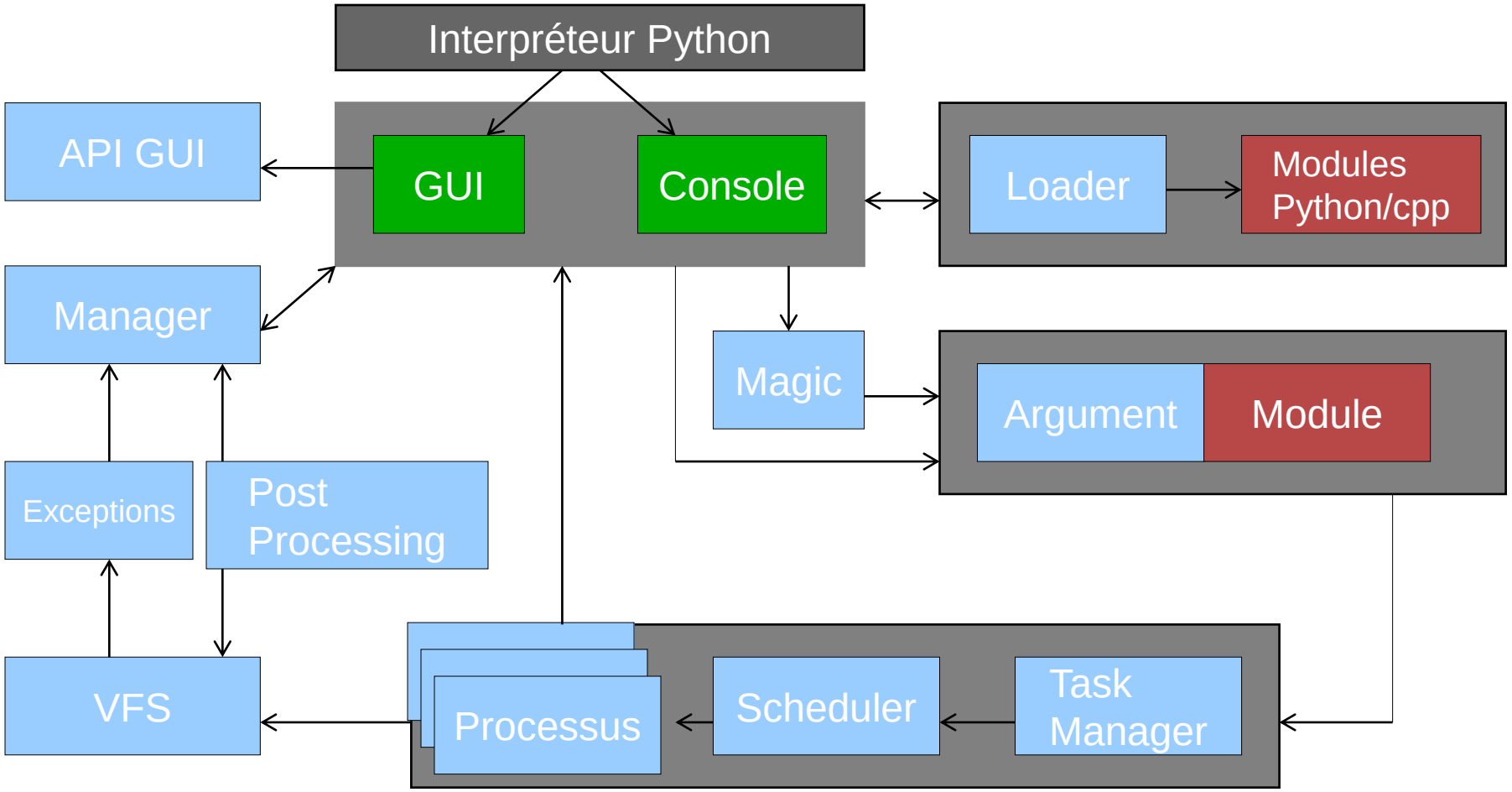
DFF

Architecture





Exécution d'un module



Système de fichiers virtuels

- Maintien de l'arborescence générée par les modules de reconstruction
- Instance unique (singleton)
- Empilable
- Intègre un mécanisme d'évènements et de callbacks

Nodes / VFiles

- Représente une donnée au sein du VFS
- Encapsule les métadonnées
- Fournit le « mapping » d'un fichier
- Propose les interfaces classiques de Nodes dans les VFS
  - Parents, fils, nombres de fils, …
  - Méthodes d'ajout / suppression

- Node.open() → Vfile
- VFile == Objet file de python
- VFile fournit des algorithmes de recherche

FSO / MFSO
[Mapped] File System Object

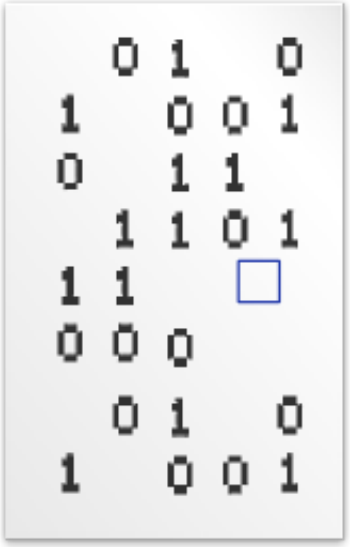
- FSO
  - Classe mère des modules de reconstruction
  - Fournit les interfaces d' I/O avec les modules
    - Vopen(), vseek(), vread(), …
- MFSO
  - Simplifier le développement pour certains types de modules
    - Manipulations des fichiers
    - Gestion du cache
    - Gestion des allocations / désallocations
  - Fournir des interfaces spécifiques pour la visualisation avancée
    - Représentation des « mappings » multicouche

File Mapping

- Principalement utilisé par les modules de types MFSO
- Liste chaînée triee des blocs alloués (chunks)
- Relation blocs / offsets
- Agrégation de plusieurs Nodes
- Gestion de “shadow” blocs

File Mapping (exemple)

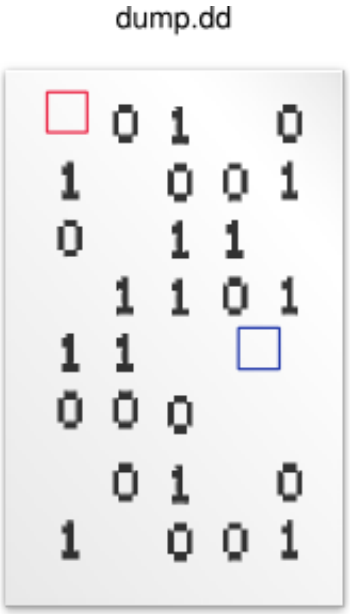
dump.dd



foo.bar

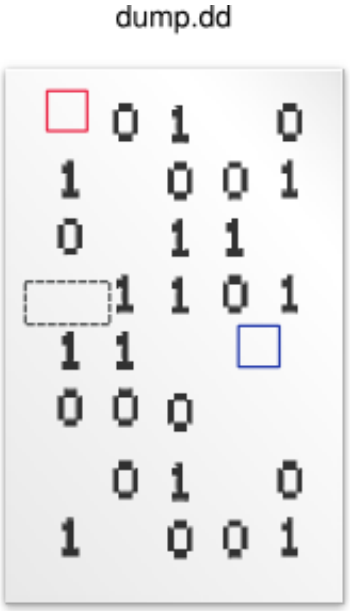
1) push(0, 512, dump.dd, 12348745)

File Mapping (exemple)



1) push(0, 512, dump.dd, 12348745)
2) push(512, 512, dump.dd, 10240)

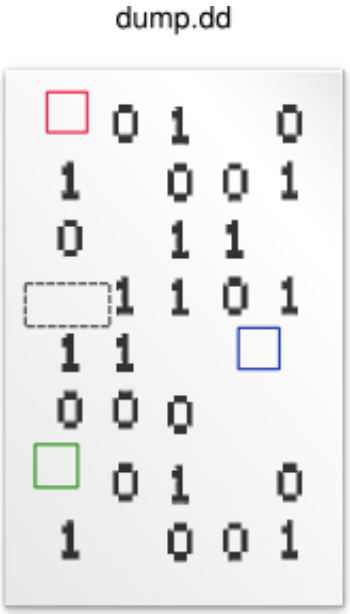
File Mapping (exemple)



1) push(0, 512, dump.dd, 12348745)
2) push(512, 512, dump.dd, 10240)

[…]

File Mapping (exemple)

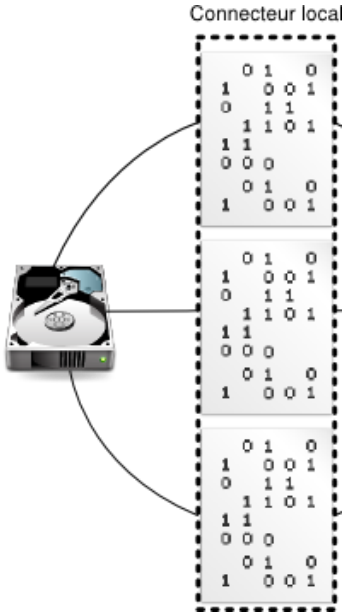


1) push(0, 512, dump.dd, 12348745)
2) push(512, 512, dump.dd, 10240)

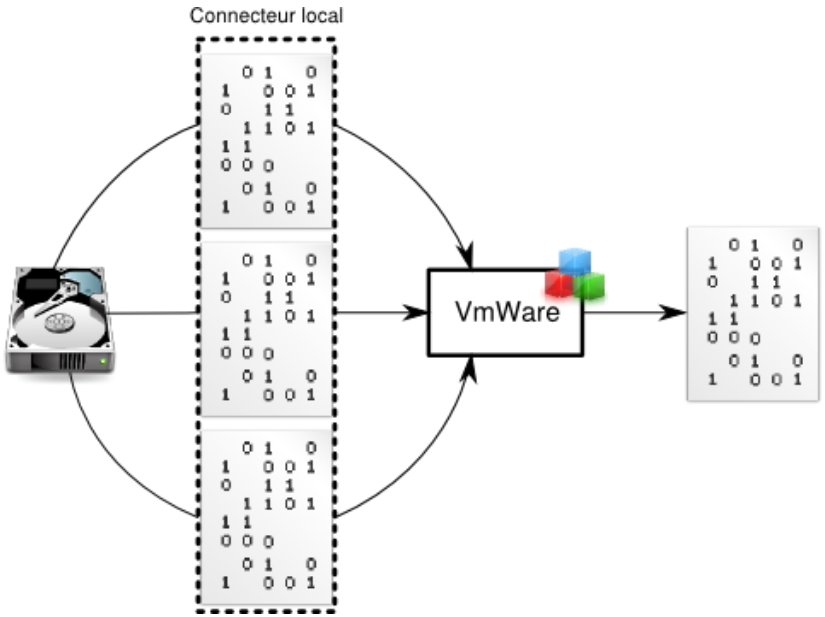
[…]

N) push(1310720, 42, dump.dd, 4965478)

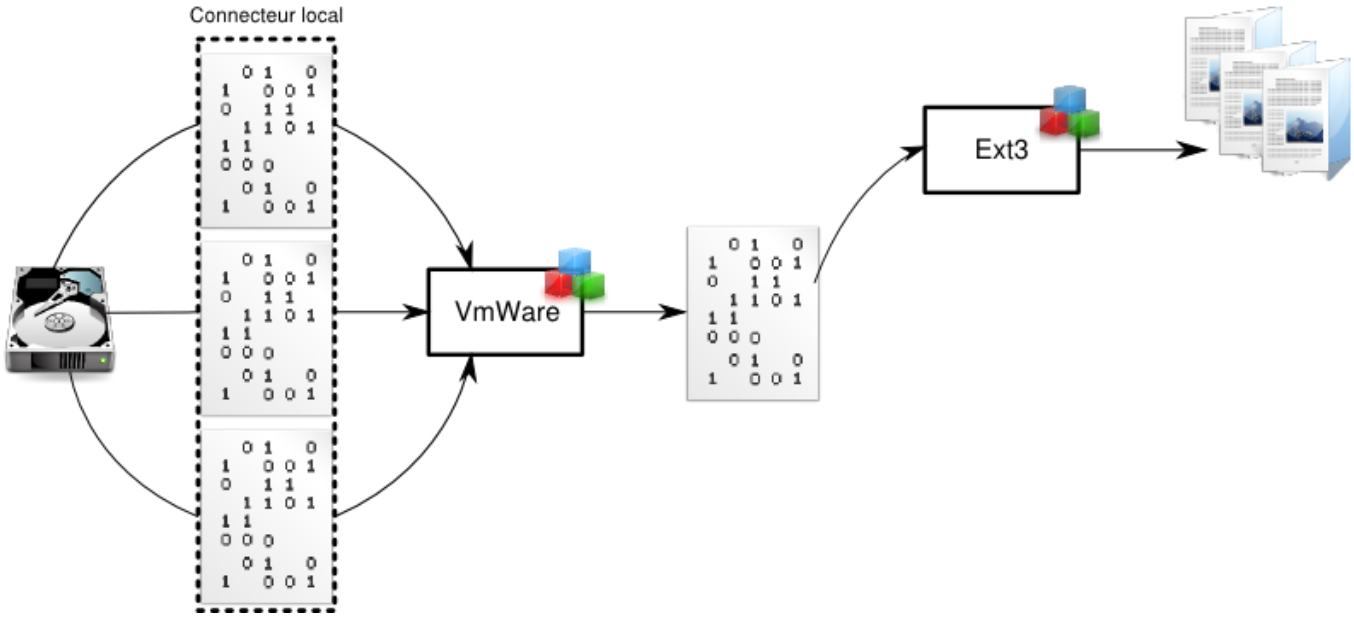
Résumé



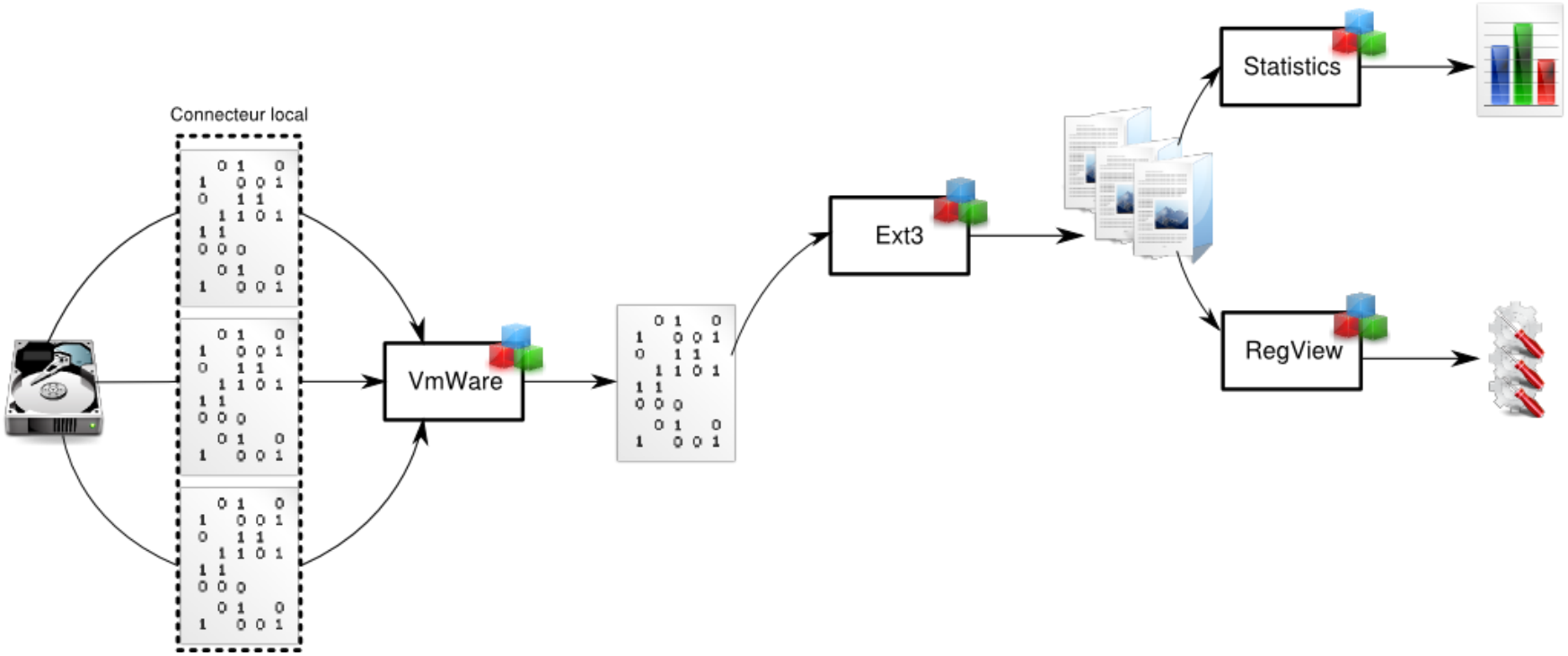
Résumé



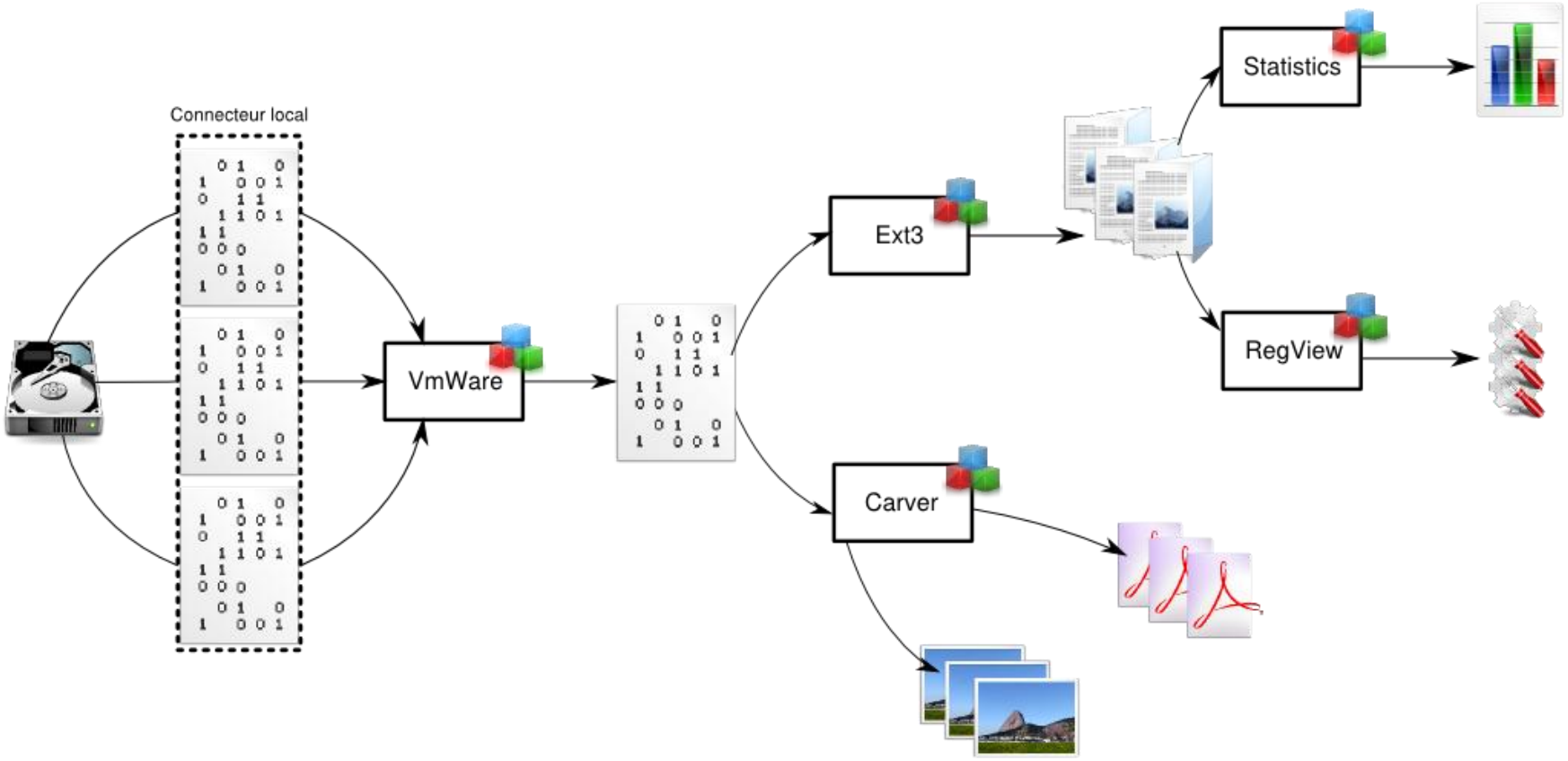
Résumé



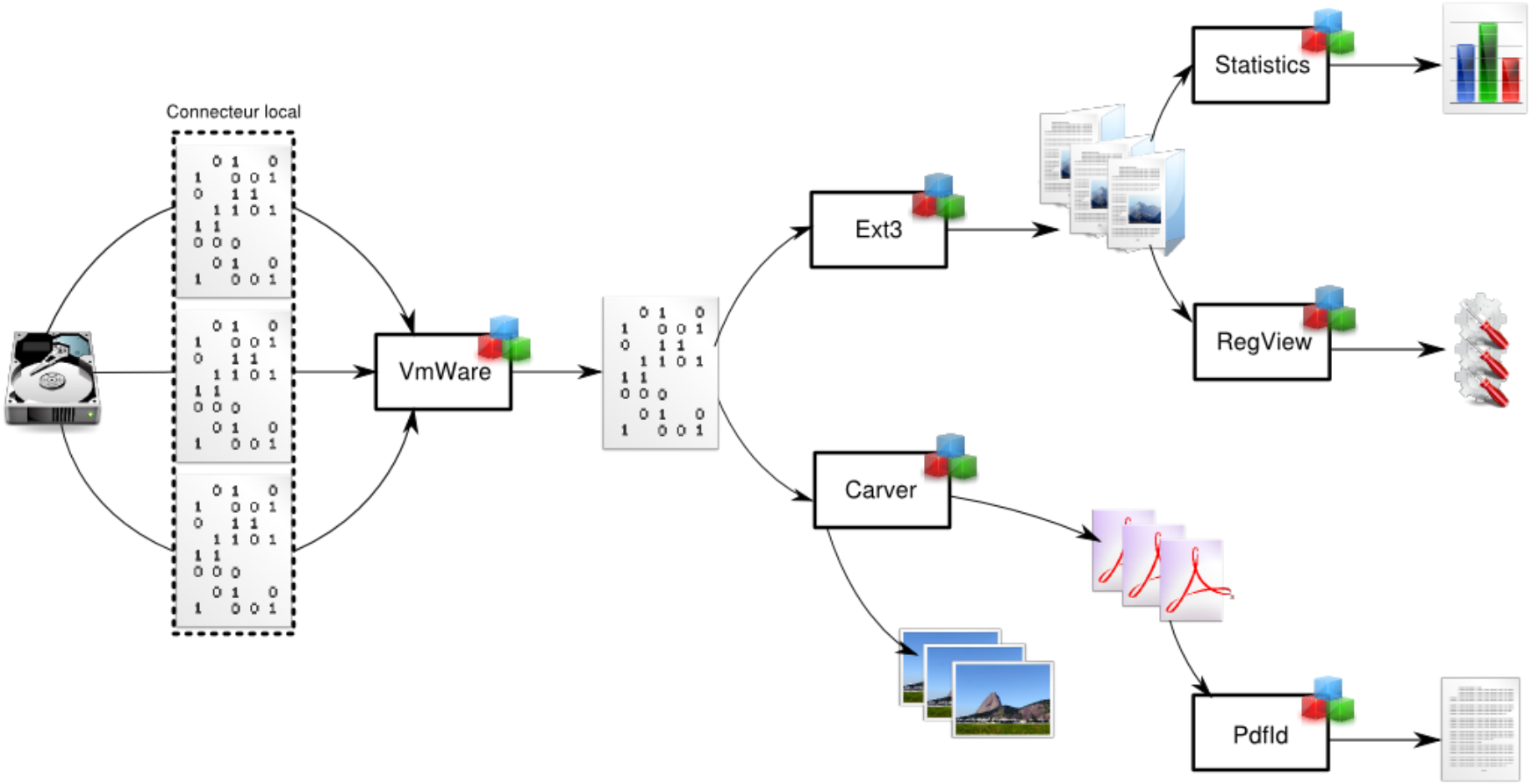
Résumé



Résumé



Résumé


Connecteur local
VmWare
Ext3
Statistics
RegView
Carver
PdfId

Tables des matières

Questions

Contacts :

Frederic Baguelin : fba@arxsys.fr

Solal Jacob : sja@arxsys.fr