

académie
Dijon 

MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE



Déploiement applicatif RPM – Ansible

31 Mai 2017

Laurent LE MOINE – Pierre-Alain TORET

- **Présentation des équipes**
- **Introduction**
- **Etat des lieux**
- **Objectifs**
- **Généralités sur le déploiement d'applications**
- **STRADA / SES : le déploiement « tout RPM »**
- **Réutilisation des paquets**
- **Diffusion des paquets**
- **Bilan**
- **Actualités : Ansible / Satellite**
- **Conclusion**

→ Equipe nationale de développement / intégration

- Projets principaux : SEM, SES, STRADA, SIEI/RIO
- Missions de l'intégration (1 ETP)
 - Packaging (RPM, sh, wlst, sql ...)
 - Qualification technique
 - Test techniques (robustesse, montée de charge)

→ CRT échanges de données (4 ETP)

- Assistance technique N2/N3 : SEM, SES, SIEI, STRADA
- Ingénierie / veille technologique
- Exploitation nationale (SIEI, RIO, SES, SYSCA, SCB2F, Extractions)

Pourquoi et comment maîtriser le processus de déploiement applicatif ?

Variantes dans le parc applicatif du MEN

- Application nationale centralisée
- Application nationale distribuée en académie
- Application académique centralisée
- Application académique spécifique à un rectorat

+ habitudes et normes des équipes d'exploitation

+ choix d'architecture locaux ou nationaux

→ Beaucoup de situations et d'acteurs différents : risque sur la fiabilité, sur la sécurité et sur l'exploitabilité des applications

Différents acteurs, différentes implications

- Les experts / pôles
 - Versions recommandées / procédures
- Les CPT DNE-B11
 - DAT
- Fourniture d'un modèle de VM par le CRT-H dans le contexte PHM
- Équipe projet
 - Documentations d'installation
 - Scripts d'installation
 - Packages d'installation
- Équipe chargée de l'exploitation
 - Mise en œuvre de tous les éléments précédents, plusieurs fois si nécessaire

→ Sujet transverse projet / production

Où en est-on ?

→ Choix limité de middleware validés par la DNE

- RHEL
- WebLogic / Jboss EAP pour le serveur d'application
- IBM DB2 / PostgreSQL pour la base de données

→ **Malgré l'uniformité, des procédures manuelles subsistent pour l'installation des briques**

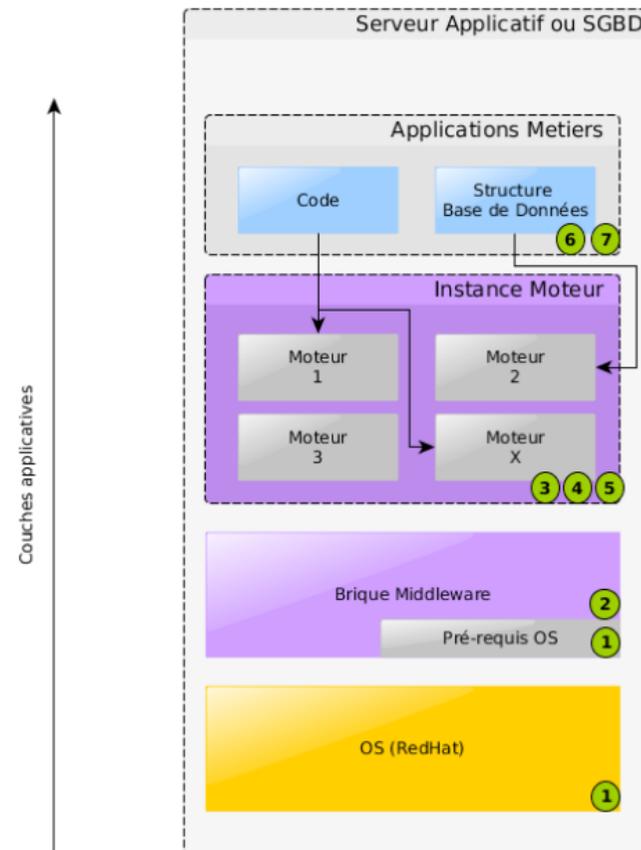
Quel type de solution ?

- **Rapide, reproductible et simple pour l'exploitant**
- **S'adaptant aux besoins des applications nationales**
- **S'adaptant aux architectures cibles (physique / virtuel / conteneur)**

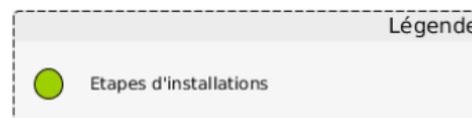
Gains attendus ?

- **Réduction des demandes d'assistance pour les installations**
- **Economies d'échelle (gain de temps pour les équipes projets et pour l'exploitation lors de la création des environnements)**
- **Facilité à fabriquer de nouveaux environnements**

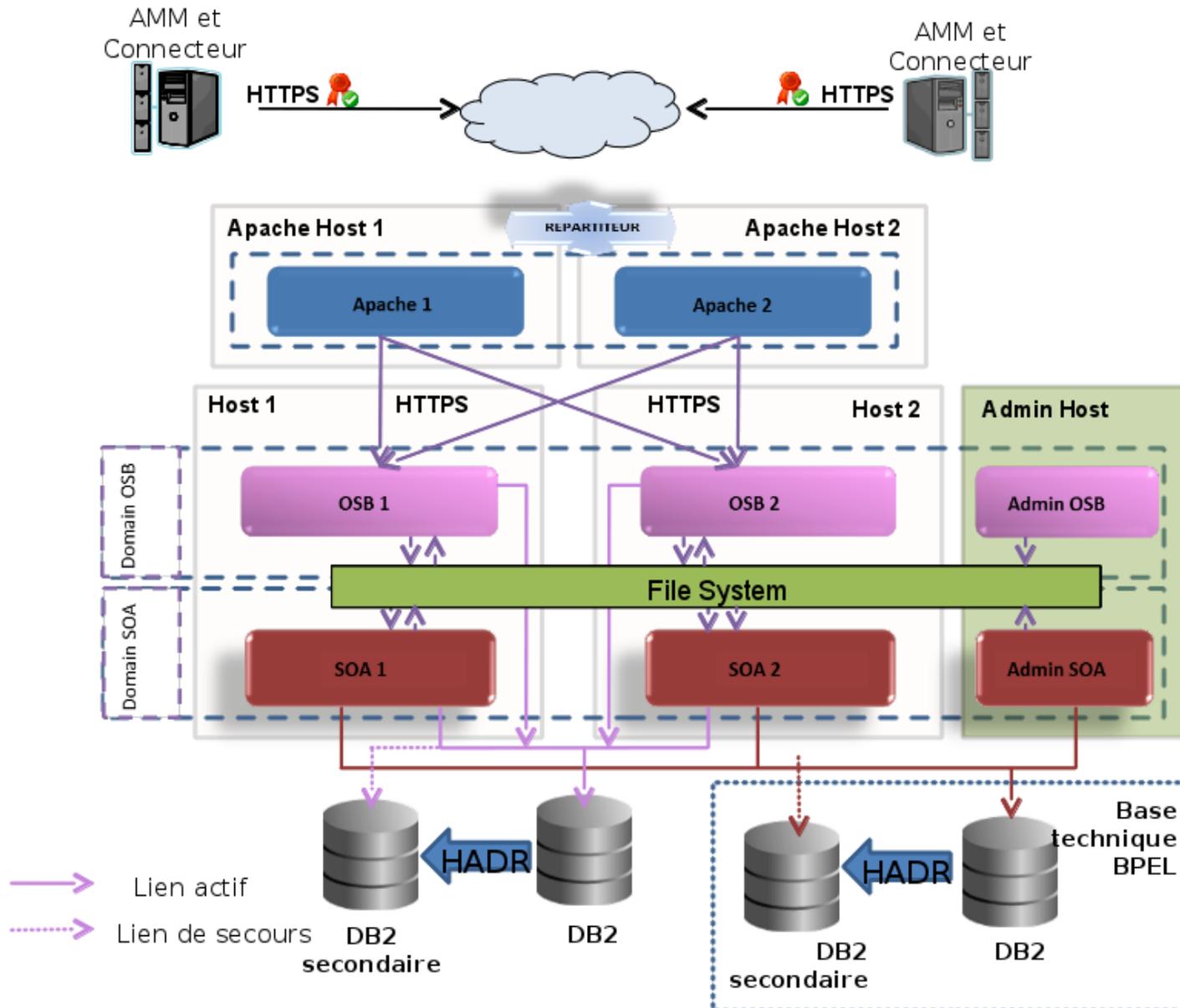
Comment déploie-t-on une application ?



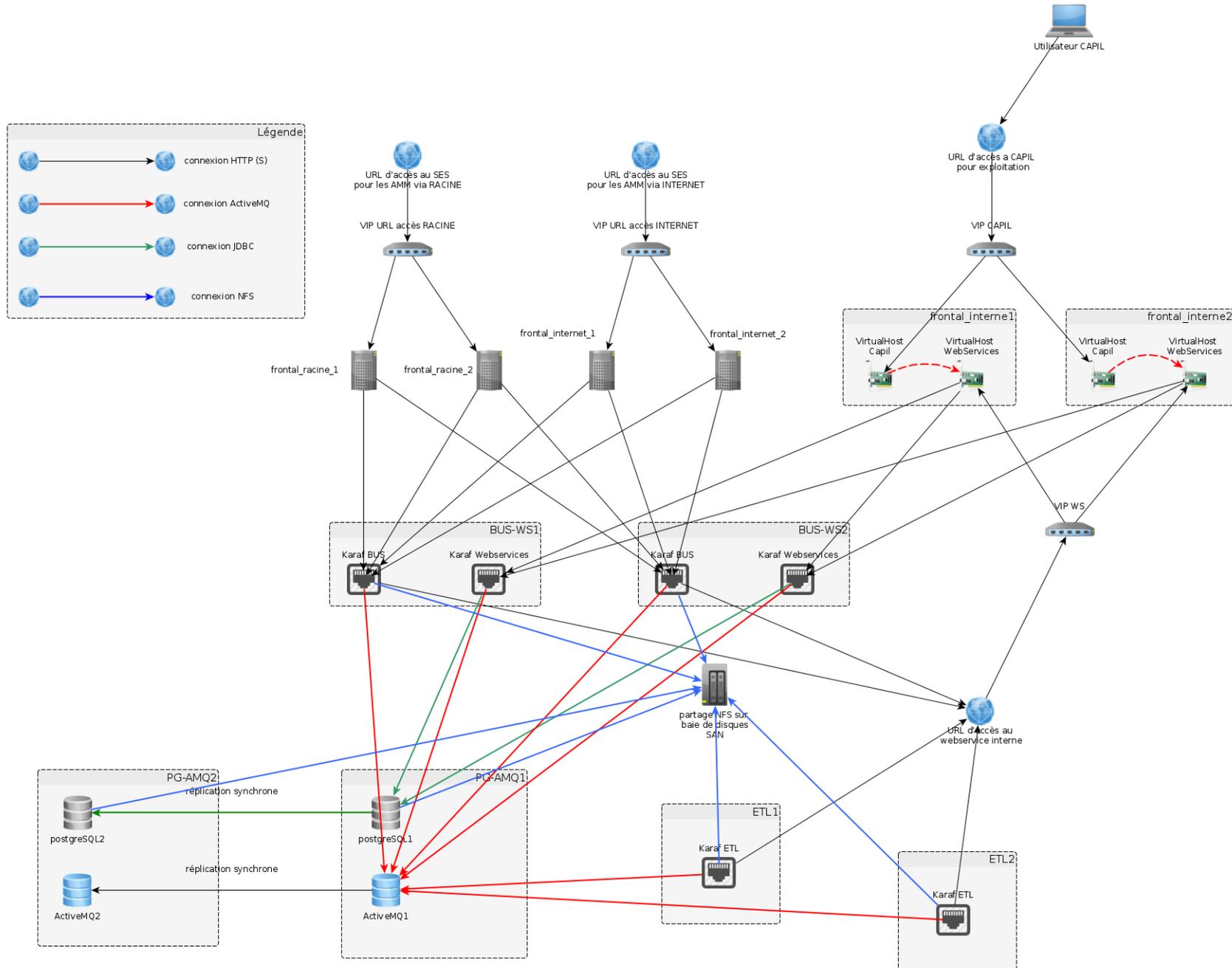
8. les paramètres du code liés à l'environnement (IP, chemins, ...)
7. le code applicatif spécifique à l'instance
6. les paramètres de l'instance (préconisations projet)
5. les paramètres de l'instance (préconisation MEN)
4. (optionnel) une ou plusieurs instances du moteur (instance DB2 ou postgresql, domaine weblogic ou jboss)
3. Le moteur middleware (DB2 / postgresql, weblogic, jboss, apache ...)
2. (optionnel) le framework (JVM si nécessaire)
- 1bis. Le paramétrage de l'OS (spécifique à la brique)
1. Le système d'exploitation (RHEL dans notre contexte)



SES/STRADA : déploiement «tout RPM»



SES/STRADA : déploiement «tout RPM»



Décomposition du déploiement STRADA 1/2

→ Étape 0 : machine virtuelle durcie type PHM

→ Étape 1 / 1bis :

- Création utilisateurs
- Paramétrage OS
- Configuration SELinux
- Pare-feu

→ Étape 2 :

- JVM Oracle

→ Étape 3 :

- Moteurs : DB2 / WebLogic / OSB / SOA

→ Étape 4 :

- Instances DB2 technique (db2isoa) et métier (db2istr)
- Domaines WebLogic OSB / SOA / Astra

Décomposition du déploiement STRADA 2/2

→ Étape 5 :

- Configuration logs WebLogic
- Tuning technique DB2 / WebLogic
- Création de la base technique via (RCU Oracle)

→ Étape 6 :

- Datasources
- Tuning métier DB2
- Configuration HADR et TSA pour DB2

→ Étape 7 : Déploiement du code

- SOA / OSB / Astra
- SQL DB2

→ Étape 8 : paramétrage du code

- Automatique pour DB2 et WebLogic
- Semi-automatisé pour OSB et SOA à cause du langage maison

→ 1-4 → RPM génériques

- 1-3 aucune adhérence au projet
- 4 lié au projet mais il n'y pas d'utilisation de variables

→ 5-*

- Nécessité d'adapter les RPMS au projet à l'environnement

→ **Les RPM ne suffisent plus !**

→ **Utilisation d'un outil de contextualisation**

- Fichier de configuration pour le projet
- Clef/valeur par info projet (IP / comptes / chemins / ...)

→ **Les RPMS livrés sont des templates et sont personnalisés à l'aide de la configuration issue de l'outil de contextualisation**

SES/STRADA : déploiement «tout RPM»

Configuration: /root/ses.conf (sur int-etl1)

Fichier

Systeme d'echange scolarite

- Général
- Apache
- Comptes techniques
- Paramétrage noyaux

BUS			
IP serveur BUS 1	<input type="text" value="192.168.1.202"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
FQDN serveur BUS 1	<input type="text" value="int-bus1.ses.test.integration.vcloud.in.ac-dijon.fr"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
nom court serveur BUS 1	<input type="text" value="int-bus-ws1"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
IP serveur BUS 2	<input type="text" value="192.168.1.201"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
FQDN serveur BUS 2	<input type="text" value="int-bus2.ses.test.integration.vcloud.in.ac-dijon.fr"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
nom court serveur BUS 2	<input type="text" value="int-bus-ws2"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
ETL			
IP serveur ETL 1	<input type="text" value="192.168.1.200"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
FQDN serveur ETL 1	<input type="text" value="int-etl1.ses.test.integration.vcloud.in.ac-dijon.fr"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
nom court serveur ETL 1	<input type="text" value="int-etl1"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
IP serveur ETL 2	<input type="text" value="192.168.1.203"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
FQDN serveur ETL 2	<input type="text" value="int-etl2.ses.test.integration.vcloud.in.ac-dijon.fr"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
nom court serveur ETL 2	<input type="text" value="int-etl2"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
POSTGRESQL			
IP serveur Postgres 1	<input type="text" value="192.168.1.205"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
FQDN serveur Postgres 1	<input type="text" value="int-pg1.ses.test.integration.vcloud.in.ac-dijon.fr"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
nom court serveur Postgres 1	<input type="text" value="int-pg1"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
IP serveur Postgres 2	<input type="text" value="192.168.1.204"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
FQDN serveur Postgres 2	<input type="text" value="int-pg2.ses.test.integration.vcloud.in.ac-dijon.fr"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>
nom court serveur Postgres 2	<input type="text" value="int-pg2"/>	<input type="button" value="Prec"/>	<input type="button" value="Def"/>

Comment contextualiser ?

→ Outil du pôle EOLE : **gen_config 2.3**

→ Niveau projet (masqué)

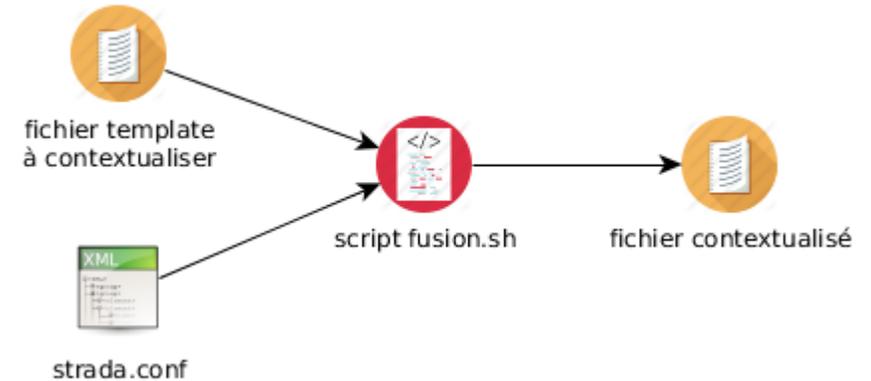
- Chemins applicatifs
- Ports normalisés
- UID / GID + noms

→ Niveau environnement

- IP
- Noms DNS
- Mots de passe

→ Validation des champs

- Format IP
- IP joignable ?
- Résolution DNS



Quels paquets sont réutilisables pour d'autres projets ?

→ **DB2 10.5 WSE**

→ **JDK 1.7.0.update 97**

→ **JDK 1.7.0.update 99**

→ **JDK 1.8.0.update 77**

→ **JDK 1.8.0.update 121**

→ **Oracle WebLogic 10.3.6**

→ **Oracle WebLogic 12.1.2**

→ **Oracle WebLogic 12.2.1**

→ **Oracle DI 10.1.5.0**

→ **Oracle Service Bus 11.1.1.7 et 11.1.1.9**

→ **Oracle SOA Suite 11.1.1.7 et 11.1.1.9**

Comment diffuser nos paquets ?

→ **Création d'un dépôt :**

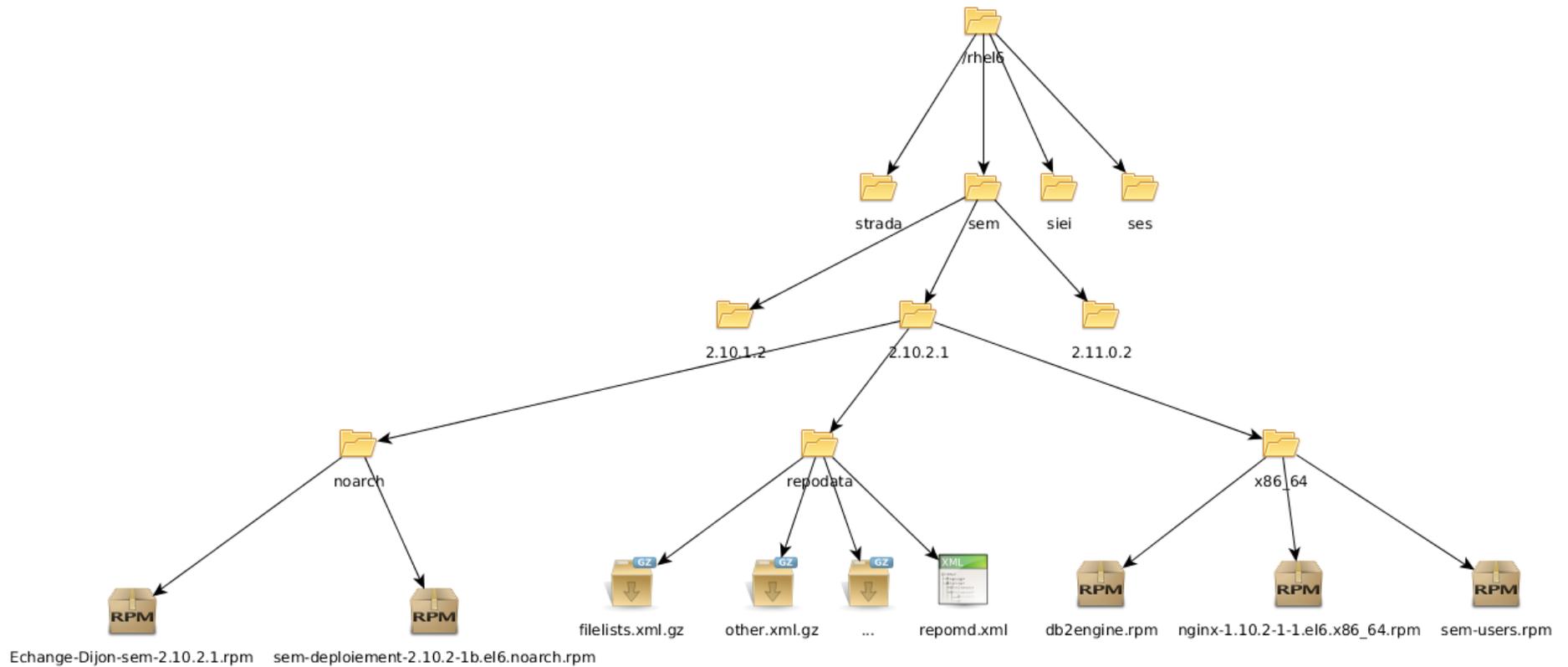
- **Serveur web (apache httpd)**
- **Utilitaire rpm (createrepo)**

→ **Diffusion des paquets pour les projets :**

- **STRADA**
- **SEM**
- **SES**
- **SIEI / RIO**

→ **Dépôt de l'équipe projet : <http://depotrpm.in.ac-dijon.fr/>**

Comment diffuser nos paquets ?



Quelles sont les limites ?

- **Pas de gestion de la mise à jour des paquets STRADA**
 - **Désinstallation gérée partiellement**
- **Usine de construction spécifique à RHEL6**
- **Pas de multi-machine**
 - **Suivre la documentation et passer de serveur en serveur**
 - **Un seul serveur à la fois**
- **Conformité à l'issue de l'installation, rien n'empêche de modifier un fichier par la suite**
- **Nécessite une normalisation des infrastructures**

Quels sont les avantages ?

- **Installation reproductible et beaucoup plus rapide**
 - **Manuel : 7j/h**
 - **RPM : 2j/h**
- **Documentation allégée**
- **Environnements identiques**

Quels apports ?

Basé sur

- Playbook
- Inventaire

Peu de tests menés, la technologie est neuve pour nous, mais prometteuse

Permet le multi-machine

Outil commun intégration / exploitation

Permet un déploiement beaucoup plus rapide que le RPM « manuel »

→ Vers une solution industrielle pour suivre la vitesse de livraison des projets

Exemple de playbook

```
~/ansible/ses$ cat roles/codepg/tasks/main.yaml
---
# ce qui va être joué pour pg1

- name: installe le code pg
  become: yes
  become_user: pgises
  command: ./install
  args:
    chdir: /partage/deploiement/ses-1.0.0/pgises
  notify:
    - restart pg
```

Exemple d'inventaire

```
[sesapacheext]
```

```
apacheext      ansible_host=192.168.1.208      ansible_user=dijon
```

```
[sesbusws]
```

```
sesbusws1      ansible_host=192.168.1.202      ansible_user=dijon
```

```
sesbusws2      ansible_host=192.168.1.201      ansible_user=dijon
```

```
[seset1]
```

```
seset11        ansible_host=192.168.1.200      ansible_user=dijon
```

```
seset12        ansible_host=192.168.1.203      ansible_user=dijon
```

```
[sespgmaster]
```

```
pgmaster       ansible_host=192.168.1.205      ansible_user=dijon
```

Prototype Ansible pour SES

Documentation de la v1.0.0 :

- 33 pages de doc spécifique
- Doc d'installation PostgreSQL DNE-B11
- Doc d'installation du dépôt RPM de Dijon

Prototype Ansible sur 9 vms :

« Automatiser la documentation d'installation »

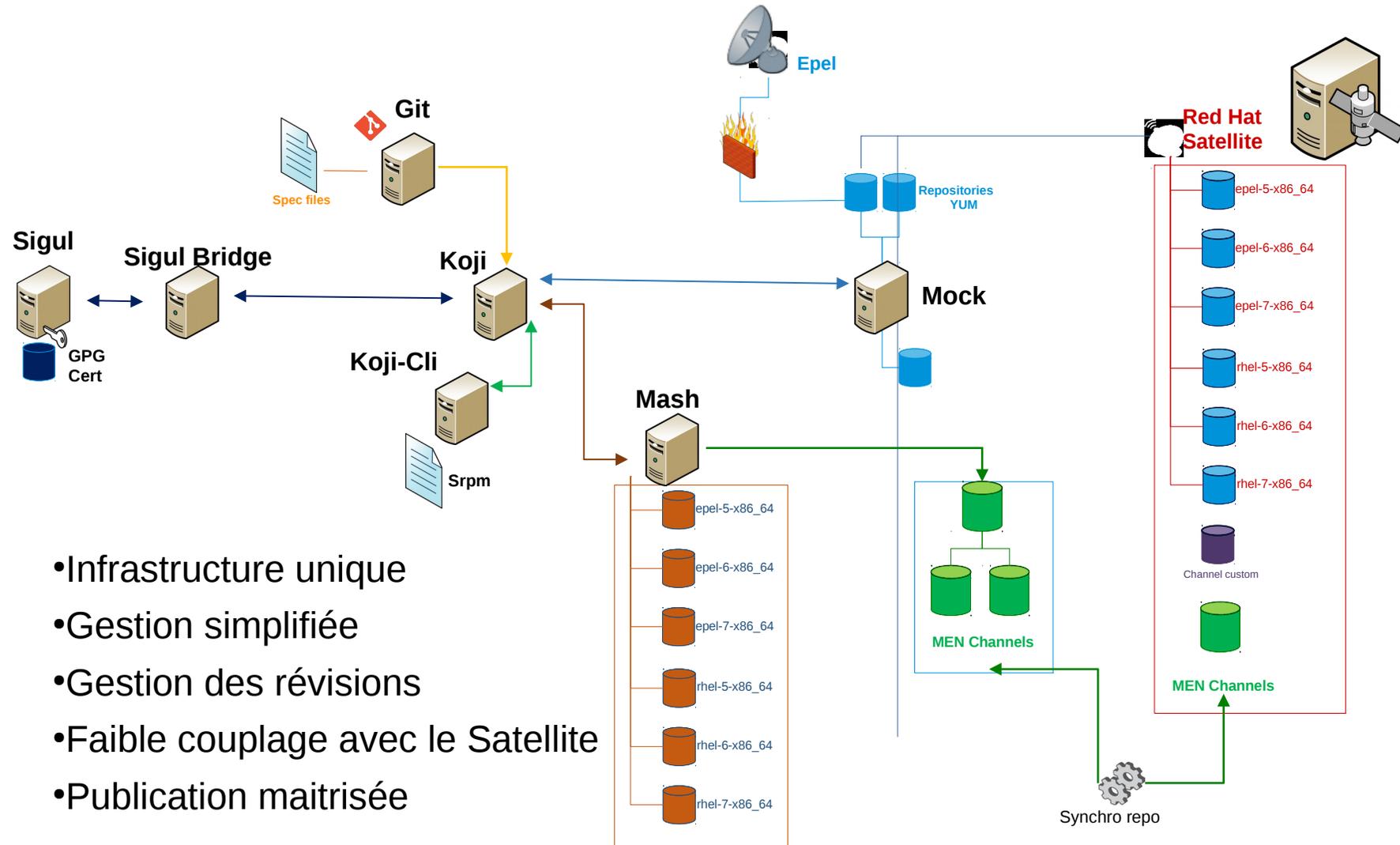
- **15 mins pour déployer un environnement du SES prêt à recevoir des flux**
- **310 lignes de code pour les 22 fichiers yaml**

→ Reste à sortir la partie code des paquets RPM

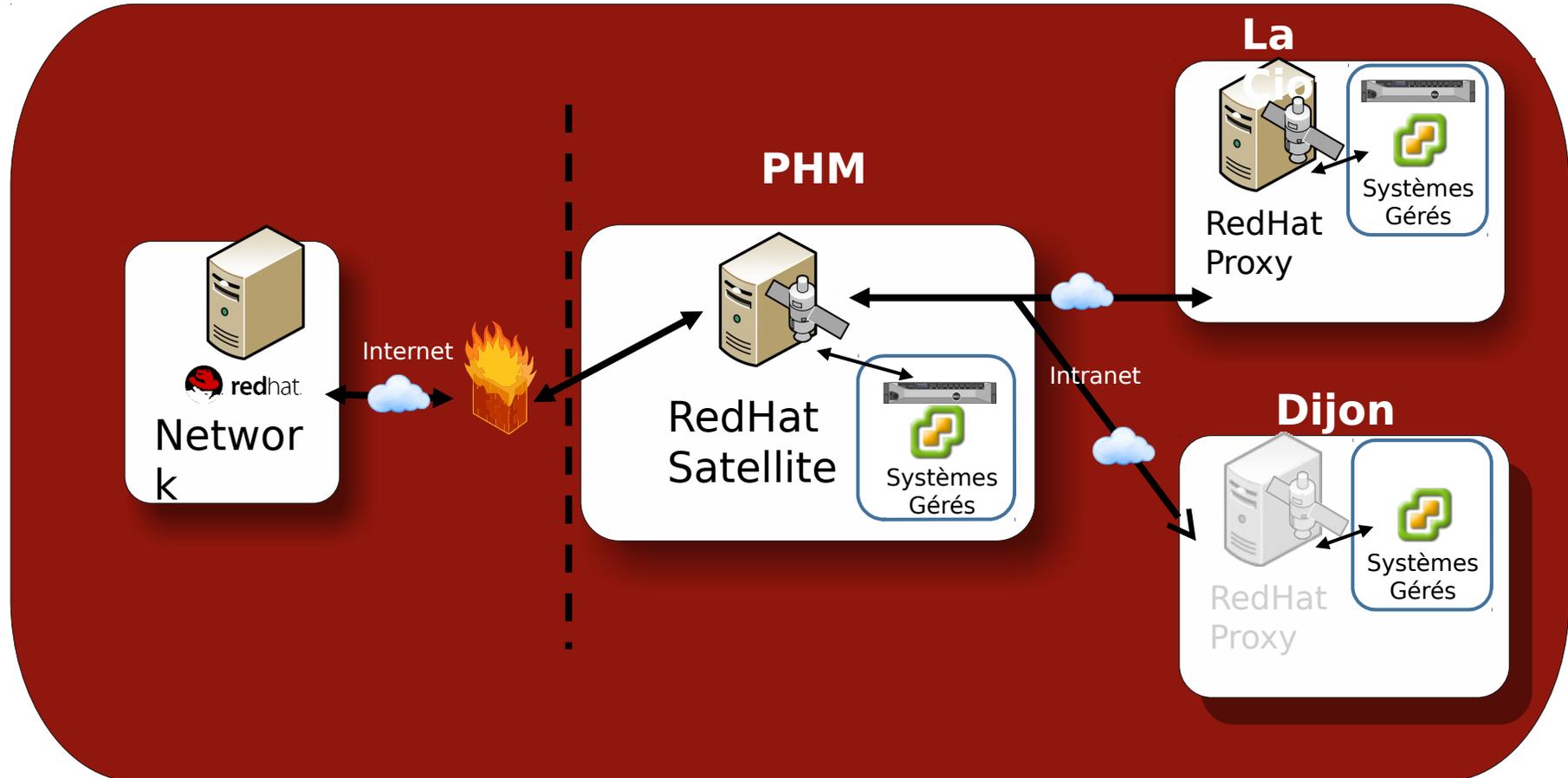
Quels sont les cas d'utilisation ?

Travaux communs avec la PHM pour la distribution de paquets

- **Gel des versions exposées**
- **Maintien de la cohérence avec les paquets Red Hat**
- **Sélection de paquets à exposer aux serveurs**
- **Evite d'avoir à maintenir un dépôt « artisanal »**



- Infrastructure unique
- Gestion simplifiée
- Gestion des révisions
- Faible couplage avec le Satellite
- Publication maîtrisée



- **Nécessité d'avoir un catalogue d'infrastructures standardisées (modèle d'OS)**
 - **Limite les écarts lors du déploiement applicatif (partitionnement, configuration générale)**
- **Fournir des middleware standardisés en RPM réutilisables par tous**
- **Utiliser Ansible pour industrialiser / orchestrer l'installation de bout en bout**
 - **RPM Middleware**
 - **Application**
 - **Configuration**
- **S'assurer à posteriori du maintien en condition de la configuration réalisée le jour de l'installation**
- **Reste à automatiser la partie Réseau (pare-feu / répartisseur de charge)**



» L'école
change avec
le numérique »
#EcoleNumerique

Équipes nationales de Dijon