

Recia

RESSOURCES NUMÉRIQUES PUBLIQUES
en Centre-Val de Loire

Gestion des infrastructures d'établissements scolaire avec SaltStack et iTop

**Le groupement informatique au service de l'action
publique en Centre-Val de Loire**

**OPEN
SOURCE
EXPERIENCE**

Je me présente

- Vincent MONTAGNE
- 35 ans
- Administrateur systèmes et réseaux
- GIP RECIA



Le GIP RECIA

UN ACTEUR PUBLIC DU TERRITOIRE RÉGIONAL

Opérateur Public de Services Numériques (OPSN) : un groupement informatique au service de l'action publique

20 ANS D'EXPÉRIENCE DANS LE DOMAINE DU NUMÉRIQUE EN RÉGION

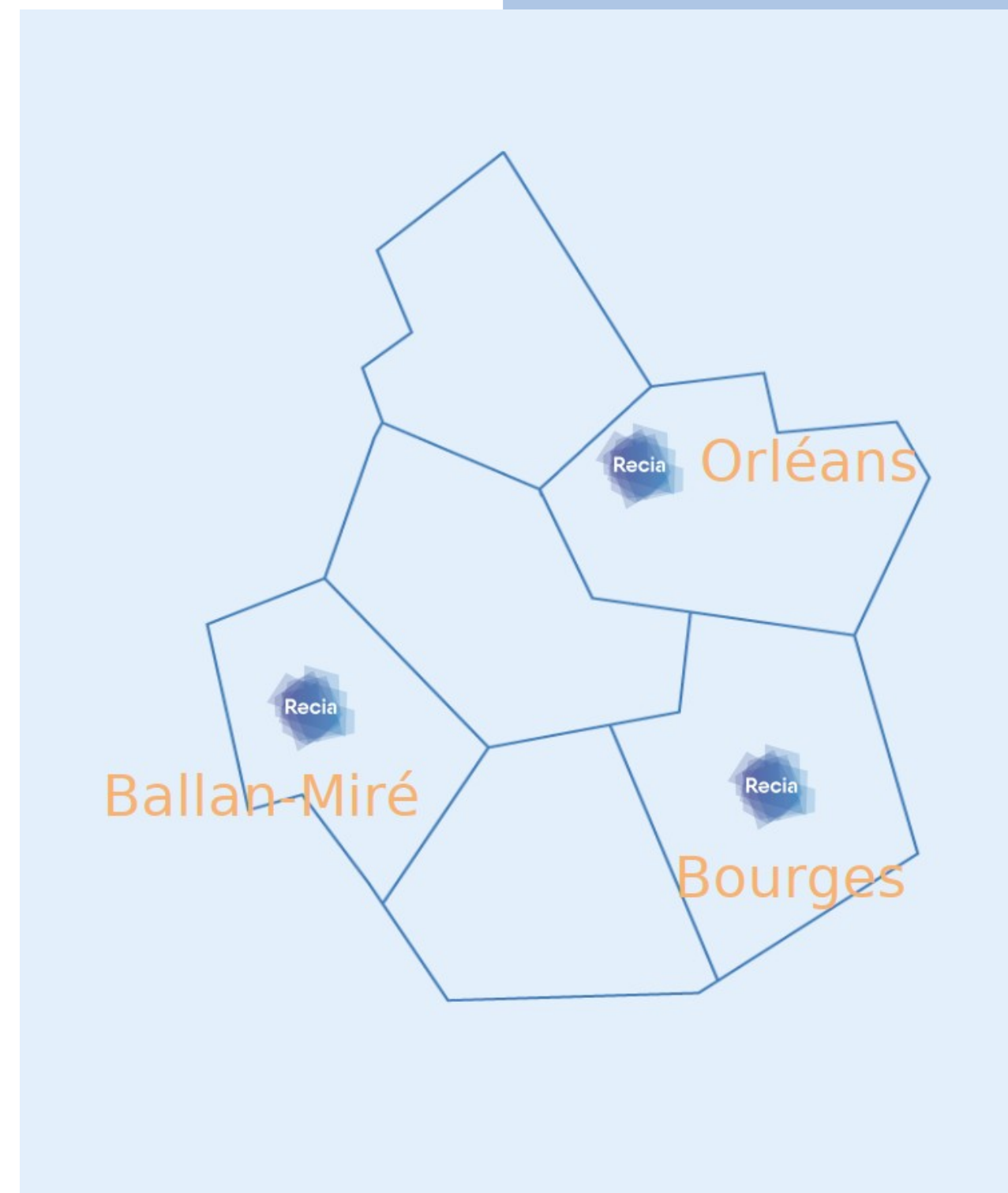
Créé en 2003 à l'initiative de l'État et de la Région Centre-Val de Loire.

DES MISSIONS D'AMENAGEMENT, D'ANIMATION ET D'ACCOMPAGNEMENT DES TERRITOIRES

- CONTRIBUER À LA STRATÉGIE NUMÉRIQUE RÉGIONALE
- ANIMER LES RÉSEAUX D'ACTEURS DU NUMÉRIQUE
- DÉVELOPPER DES SERVICES OPÉRATIONNELS

UNE PROXIMITÉ TERRITORIALE ET UNE IMPLICATION NATIONALE

- Une implantation sur 3 départements
- Participation active aux réseaux d'échanges nationaux
- Acteur de la communauté des logiciels libres



DECLIC

avioca

ADULLACT

esup
portail

afigéo
Construire l'information géographique de demain

5 décembre 2024

Open Source Experience

RÉGION
CENTRE
VAL DE LOIRE

PRÉFÈTE
DE LA RÉGION
CENTRE-VAL
DE LOIRE
Liberté
Égalité
Fraternité

Le GIP RECIA

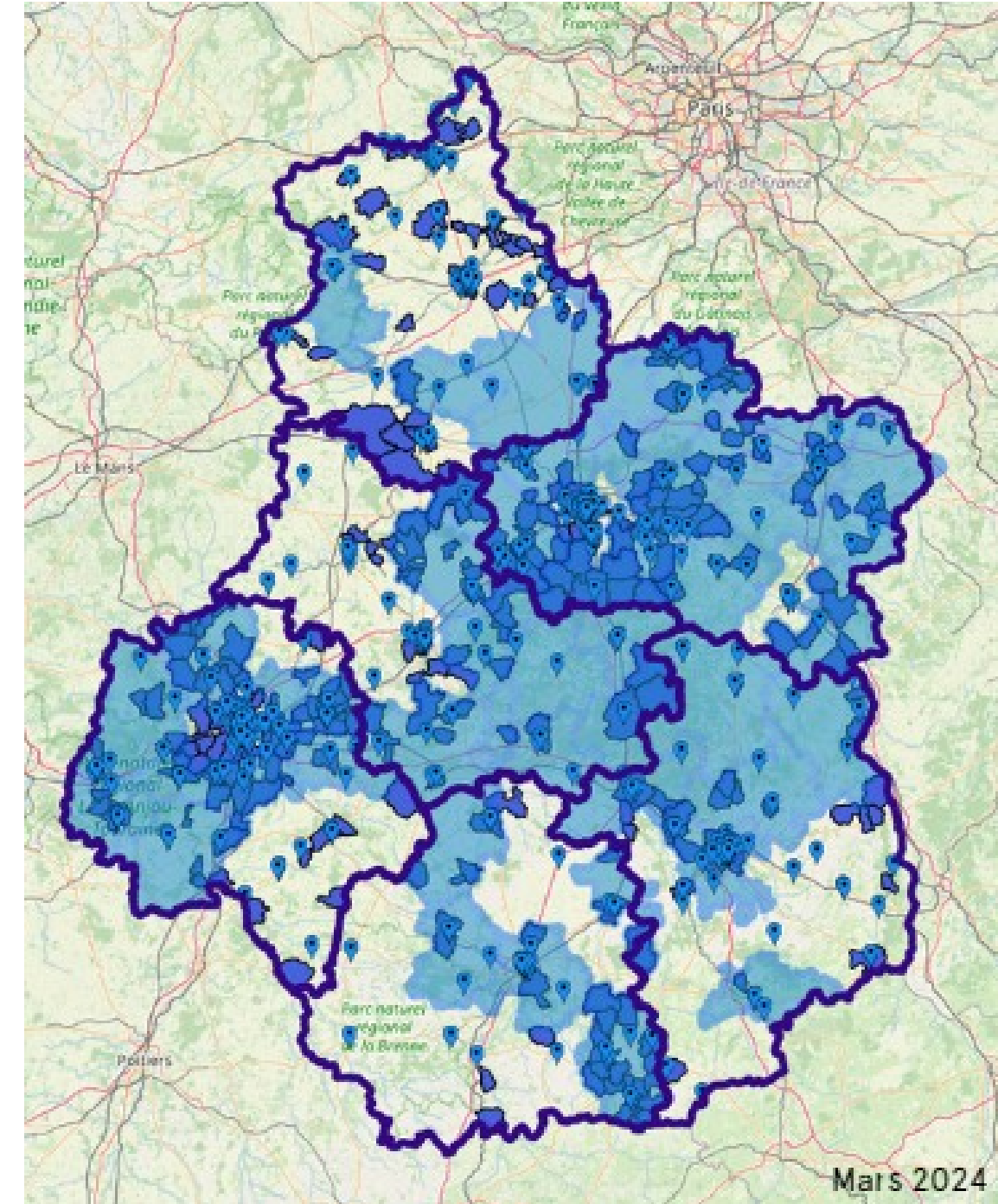


+ de 720 membres
représentants



+ de 980 structures
publiques

DES MEMBRES ET DES PARTENAIRES INSTITUTIONNELS RECONNUS :



Sommaire

- Présentation du contexte
- Situation d'origine
- Nouvelle solution avec Saltstack
- Présentation Saltstack
- Liaison CMDB itop
- Conclusion



Contexte

- Gestion d'établissements scolaires en région centre (~ 190)
 - Lycées
 - Lycées agricoles
 - CFA
 - EF2S
 - Collèges du Cher
- Gestion des infrastructures
 - Réseau
 - Serveur
- Gestion des stations
- Utilisation principale d'outils libres



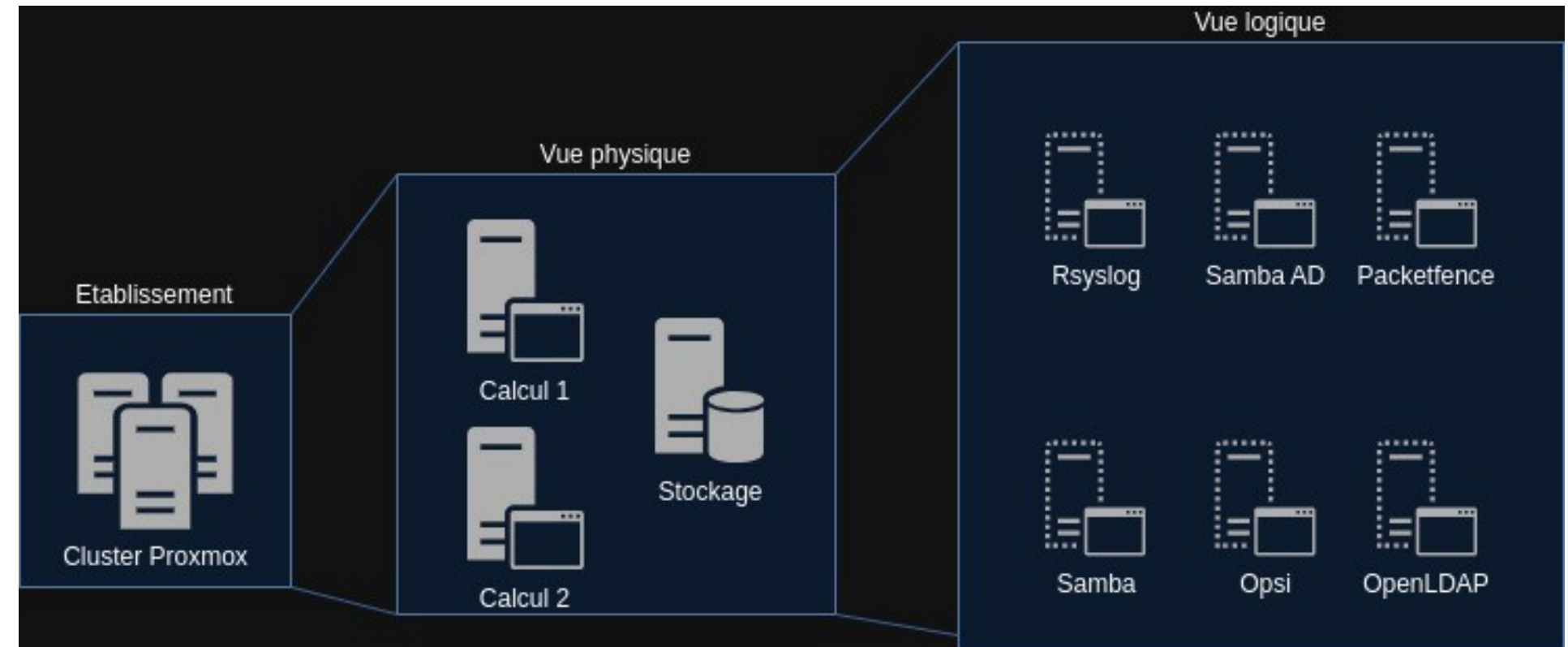
Mon service

- Service GSI (Gestion des Systèmes Informatiques)
- 3 pôles
 - IDMS (Intégration, Déploiement Maintenance Stations)
 - 8 personnes
 - CDS (Centre de services - Télégestion)
 - 16 personnes
 - ISSR (Infrastructures, Sécurité, Systèmes et Réseaux)
 - 10 personnes
 - Je suis ici !



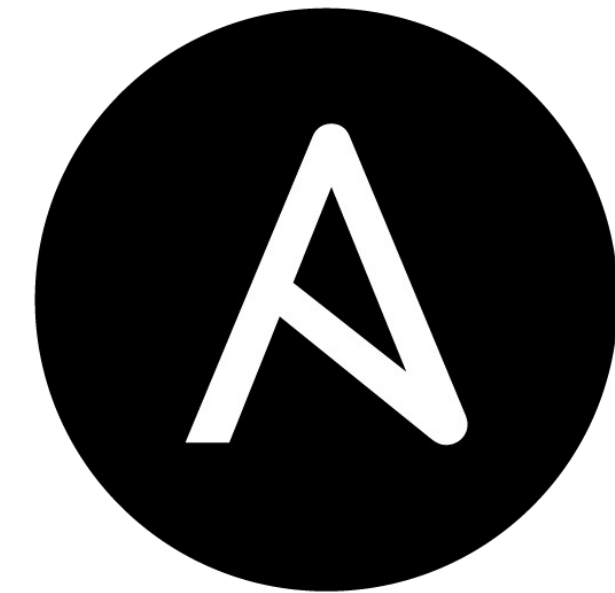
Infrastructure type

- Un cluster proxmox
 - 2 serveurs orientés calcul
 - 1 serveur orienté stockage
- Des KVM ou LXC sous debian
 - AD : Samba
 - LDAP : OpenLDAP
 - Serveur de fichier : Samba
 - Serveur de gestion des stations : OPSI
 - NAC : Packetfence
 - Serveur de sauvegarde : rdiff-backup
 - Etc ...



Historique

- Déploiement et gestion par Ansible
- Difficultés à déployer les configurations sur tous les serveurs
- Lenteurs des accès SSH
- Décalage des configurations entre les serveurs
 - Différents problèmes
 - Incident réseau opérateur
 - Coupure électrique
 - Différences de version



ANSIBLE

Remplacement d'Ansible

- Utilisation de Saltstack
 - Configuration des hyperviseurs Proxmox
 - Déploiement des KVM et LXC
 - Configuration de l'OS invité (Debian)
 - Installation et configuration des services / logiciels

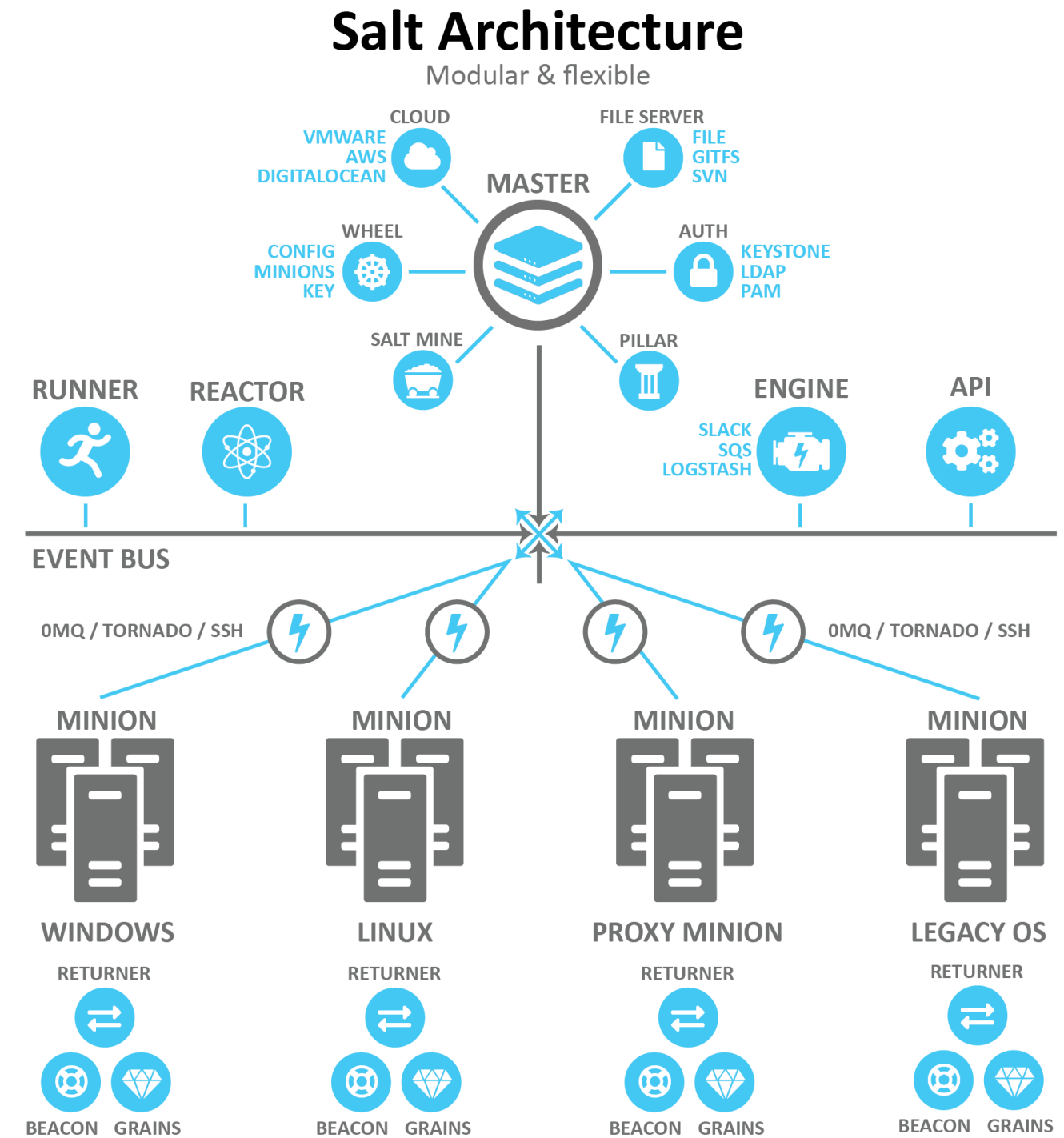


Présentation de Saltstack

- Logiciel de gestion de configuration et d'exécution de commandes distribué
- Open source
- Développé en Python
- Créé en 2011
- Racheté par VMware, maintenant chez Broadcom
- Avantages :
 - Rapide
 - Flexible

Principe de fonctionnement

- Client / Serveur
 - Minion / Master
- Bus ZeroMQ
- Connexion permanente
- Chiffrement des échanges
 - Systèmes de clé entre master et minion
- Différents types de modules suivant le mode de fonctionnement



Notre fonctionnement

- 2 environnements indépendants
 - Développement
 - Production
- 1 dépôt git par environnement
- Approbation nécessaire pour merge sur le dépôt production
- Les dépôts contiennent
 - Fichier de configuration du master
 - Déclaration pillars
 - States
 - Modules personnalisés

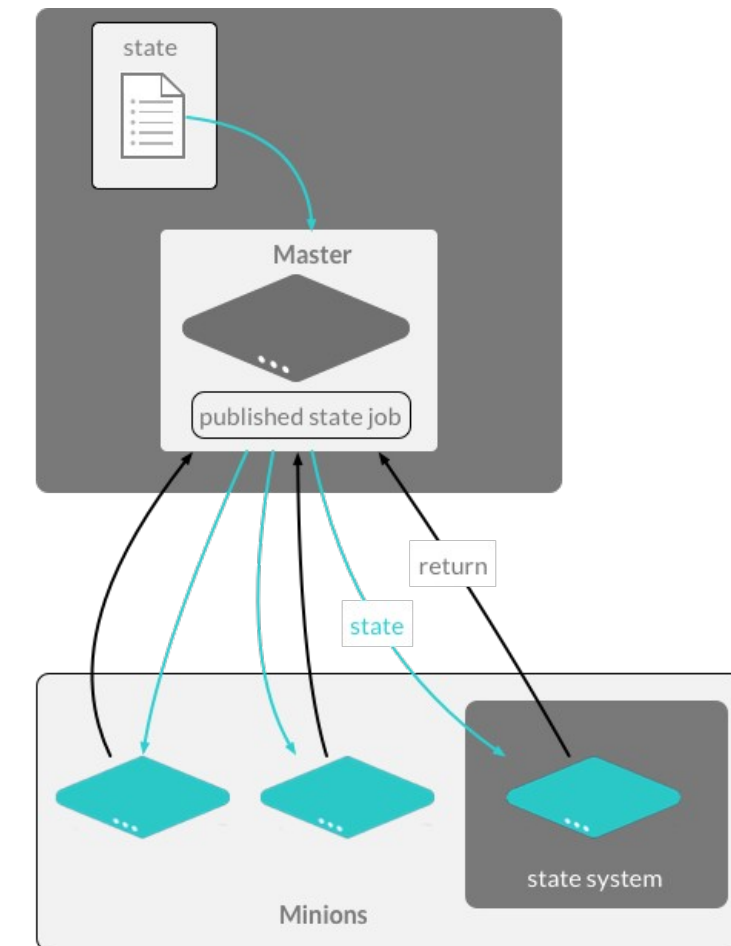


Modules

- Différents types de modules
 - Exécution
 - Exécute des actions sur le minion
 - States
 - Application d'un état sur le minion
 - Returner
 - Différents affichages de retour de commande
 - Grains
 - Fonction de génération des grains (variables) par le minion
 - Pillar
 - Extension des pillars via des sources externes (ex : Stack)
 - Render
 - Différents systèmes de rendu des fichiers (par défaut : jinja)
 - ...

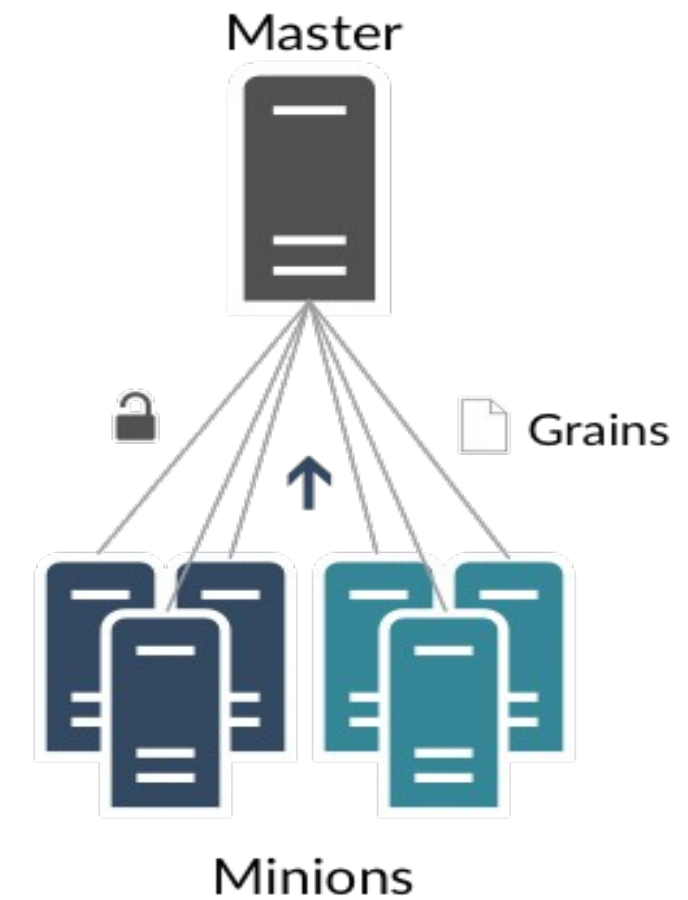
States

- Définissent l'état souhaité d'un produit / service / logiciel
- Utilise une syntaxe claire et lisible
- Utilise Jinja
- Fichier
 - syntaxe YAML
 - extension .sls
- Personnalisation du comportement via variables
 - Grains
 - Pillars



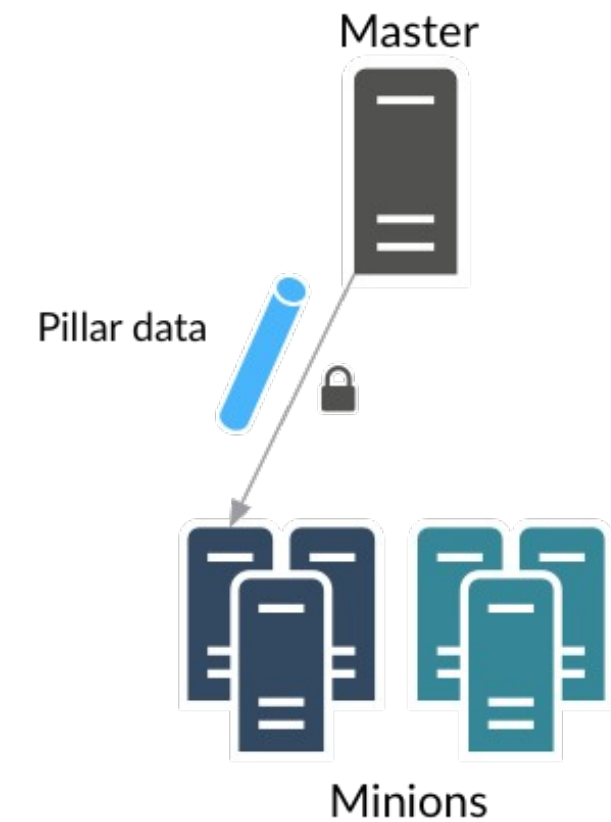
Grains

- Informations générées et stockées par le minion
- Données qui changent très peu
- Principalement des propriétés systèmes
- Exemple : os, kernel, ip_interfaces
- Possibilité d'ajouter des grains customs



Pillars

- Définis en yaml sur le master
- Générés par le master
- Mis à disposition pour le minion ciblé uniquement
- Variables définies par les admins
- Utilisés dans les states
- Extension possible



Gestion des particularismes

- Utilisation des pillars saltstack
- Extension pillarStack
 - Fusion des fichiers de pillars en suivant une stratégie
 - Ordre de fusion défini dans un fichier de configuration
 - Template Jinja
 - Différentes variables mises à disposition
 - Grains
 - Pillar
 - Stack



Exemple PillarStack

Fichier de configuration

```
$ cat /path/to/master.d/ext_pillar.conf
ext_pillar:
- stack: /path/to/stack.cfg
```

Fichier de définition de pillarstack

```
$ cat /path/to/stack/config.cfg
commun.yml
type_etab/{{ pillar.get('type_etab') }}.yml
code_etab/code_{{ pillar.get('code_etab') }}.yml
minions/{{ minion_id }}.yml
```

Arborescence

```
$ tree /path/to/stack/
/path/to/stack/
├── config.cfg
├── commun.yml
├── type_etab/
│   ├── len.yml
│   └── cen.yml
├── code_etab/
│   ├── code_xxx.yml
│   └── code_yyy.yml
└── minions/
    ├── srv-exemple-01.yml
    └── srv-exemple-02.yml
```

Informations complémentaires

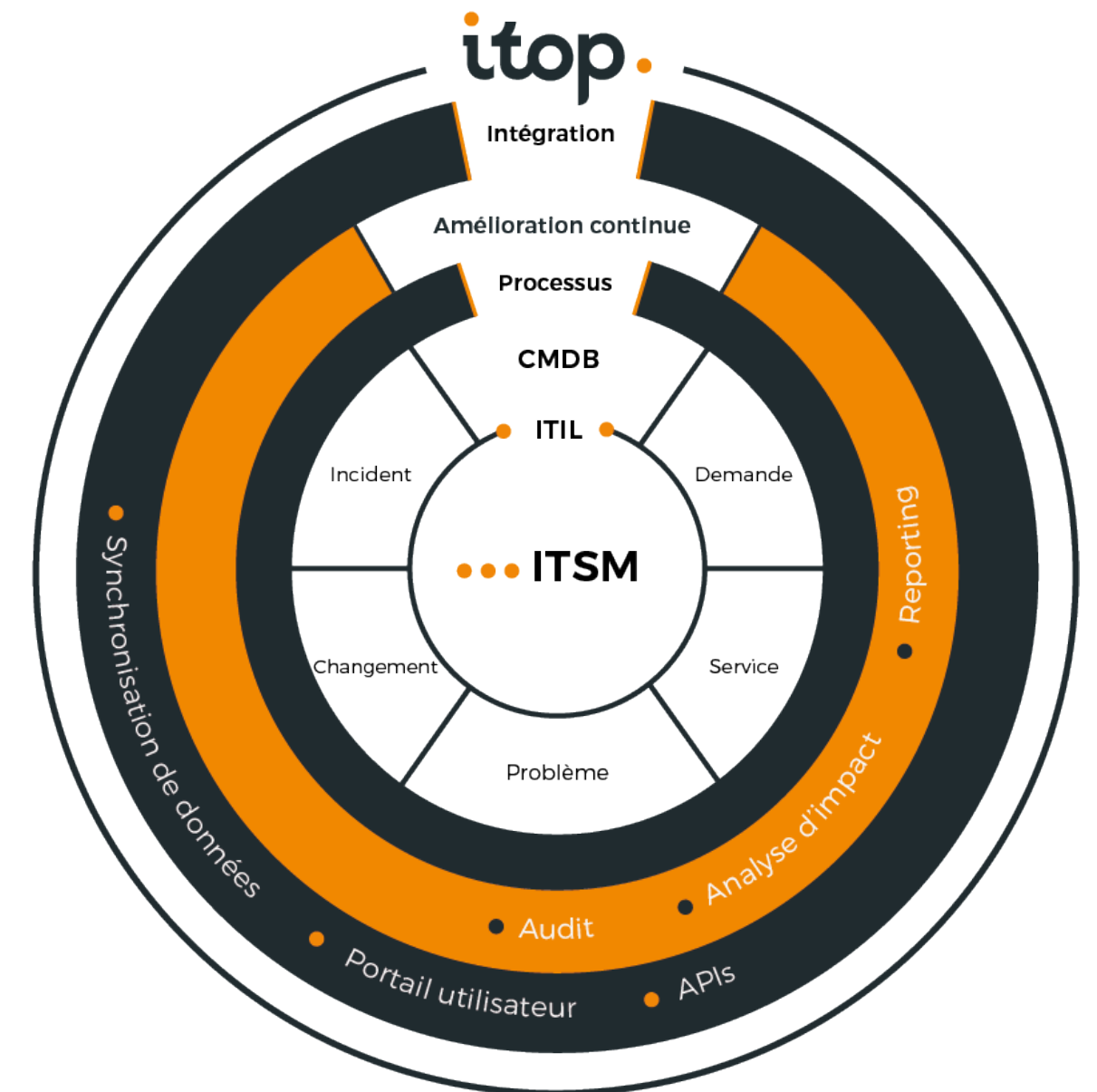
- Nombreuses informations présentes dans notre CMDB Itop
- Serait utile en tant que variable / pillar
- Trop lourd à éditer et maintenir manuellement
- Mise à jour dynamique suivant les changements dans Itop
- API disponible sur Itop



Itop

- iTop est une plateforme complète de gestion des services informatiques, basée sur le web et en open source, qui comprend :
 - une CMDB entièrement personnalisable,
 - un système de helpdesk
 - et des outils de reporting.
- iTop est conçue par la société éditrice Combodo
- créée en 2010

itop.
easy service management



Création de pillars automatique

- Création de pillars pour chaque établissement depuis les informations présentes dans Itop
- Script python
 - Récupère les informations via API Itop
 - Traite les données
 - Génère des fichiers de pillars en yaml
- CI/CD gitlab
 - Exécute le script
 - Merge request
 - si des valeurs ont changé
 - Vers environnement de développement



Résultat

- Un fichier par établissement
- Chargé par la PillarStack
- Données accessibles dans la suite de la pillarstack
 - variable stack
- Valeurs peuvent être surchargées si besoin

```
-----
centralized:
-----
  packetfence_wifi:
vlangs:
-----
  2:
-----
    broadcast:
    cidr:
    dhcp:
-----
    manage_by:
    options:
-----
      comment:
      domain_name_servers:
      includes:
      netmask:
      routers:
-----
    gateway:
    mask:
    name:
    nameservers:
    network:
    ntps:
    routable:
    searchdomains:
    subnet:
...
servers:
-----
  ad:
-----
    admin_password:
    base_dn:
    bind:
-----
    forwarder:
    domain:
    ip:
    ntp:
    realm:
    reverse_zones:
  ebpd:
-----
    ip:
  slapd:
-----
    admin_account:
    admin_password:
    ip:
    suffix:
...
```



Modification datamodel Itop

- Pas de modifications des classes par défaut
- Ajout de classe recia spécifique à nos besoins
 - Attribut d'organisation
 - Type de toip
 - Serveur de gestion radius
 - ...
 - Attributs de VLAN
 - Serveur NTP
 - Serveur dns
 - ...
- Liaison des classes recia aux classes existantes uniquement si besoin
- Ne modifie pas le fonctionnement pour les autres utilisateurs
- Possible grâce à l'option designer (non disponible en community)

Remontées d'informations dans itop

- State salt de récupération d'information sur les serveurs
- Envoie des informations dans une base redis
- Script python de corrélation des données
- Mise à disposition des données dans une base de données
- Utilisation des données pour synchronisation itop
- Exemple :
 - Récupération des vmid
 - Connu de l'hyperviseur mais pas de la VM



Solution aboutie

- Saltstack avec application toutes les heures sur les minions
- PillarStack pour configuration des spécificités
- Récupération des valeurs de la CMDB et injection dans les pillars
- Déploiement des VMs
- States Saltstack pour la configuration des applications
- Remontée d'informations dans Itop pour compléter la CMDB

Recia

RESSOURCES NUMÉRIQUES PUBLIQUES
en Centre-Val de Loire

Merci de votre attention

**Le groupement informatique au service de l'action
publique en Centre-Val de Loire**

**OPEN
SOURCE
EXPERIENCE**

