Réunion bilatérale technique
Ministère de la Défense - Equipe Noyau

Valeurs Immatérielles Transférées aux Archives pour Mémoire

VITAM

Sommaire

- Contexte du projet
- Sujets techniques
  - Réseaux
  - Système
  - Authentification
  - Haute disponibilité
  - Exploitabilité
  - Transferts de fichiers
  - Gestion de la traçabilité
  - Sécurité des supports

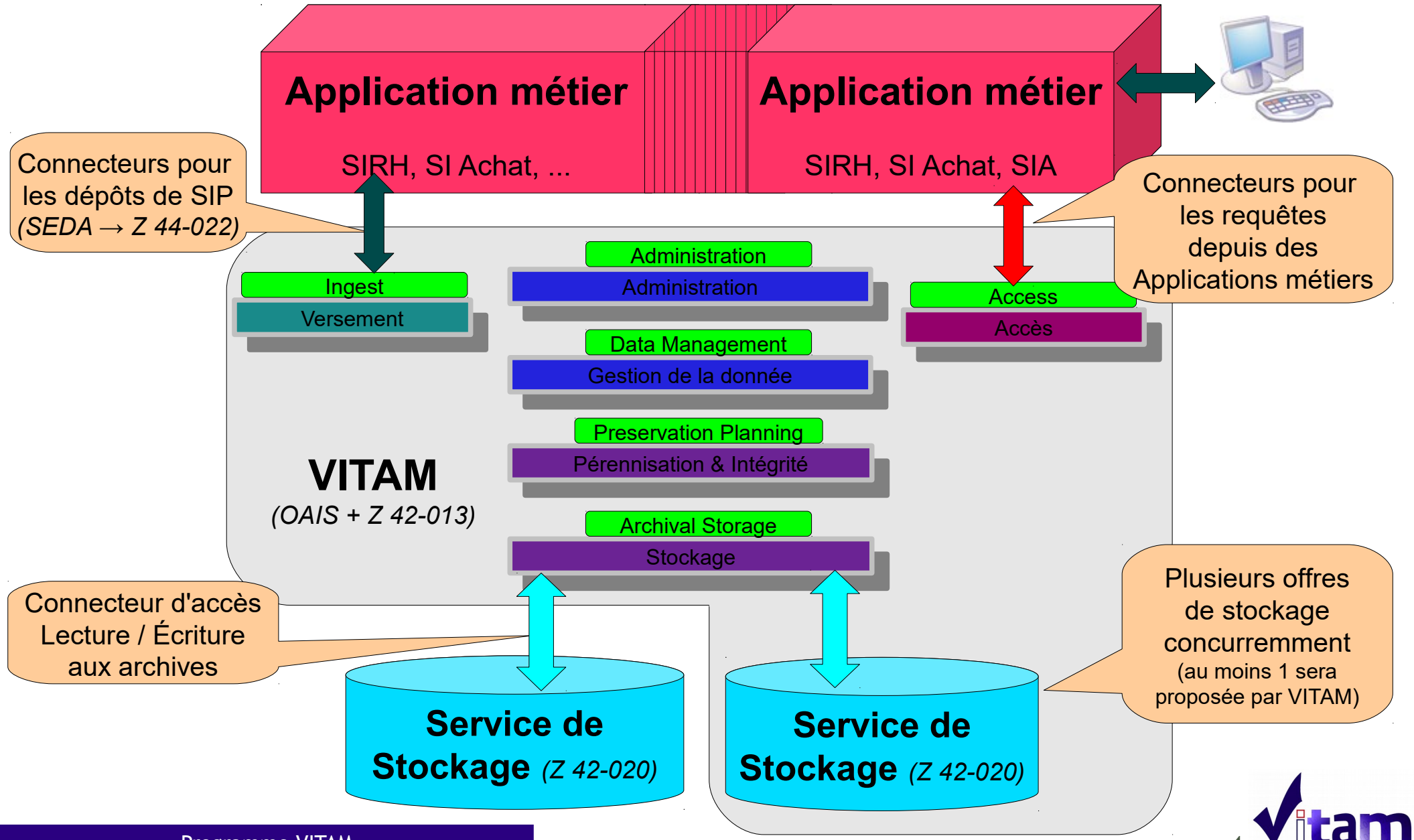
Contexte de la réunion

- Chantiers techniques
  - Des chantiers techniques se déroulent avec tous les acteurs de la sphère publique pour collecter une liste d'entrants qui pourront guider les choix techniques du projet Vitam
  - 5 chantiers
    - *09/12/2015 : Infrastructures existantes chez les différents acteurs*
    - *12/01/2016 : Introduction aux API « externes » pressenties de VITAM*
    - *04/02/2016 : Enjeux de sécurité existants chez les différents acteurs*
    - *17/02/2016 : Présentation approfondie des API « externe » pressenties de VITAM*
    - *15/03/2016 : Approfondissement des sujets d'exploitabilité*

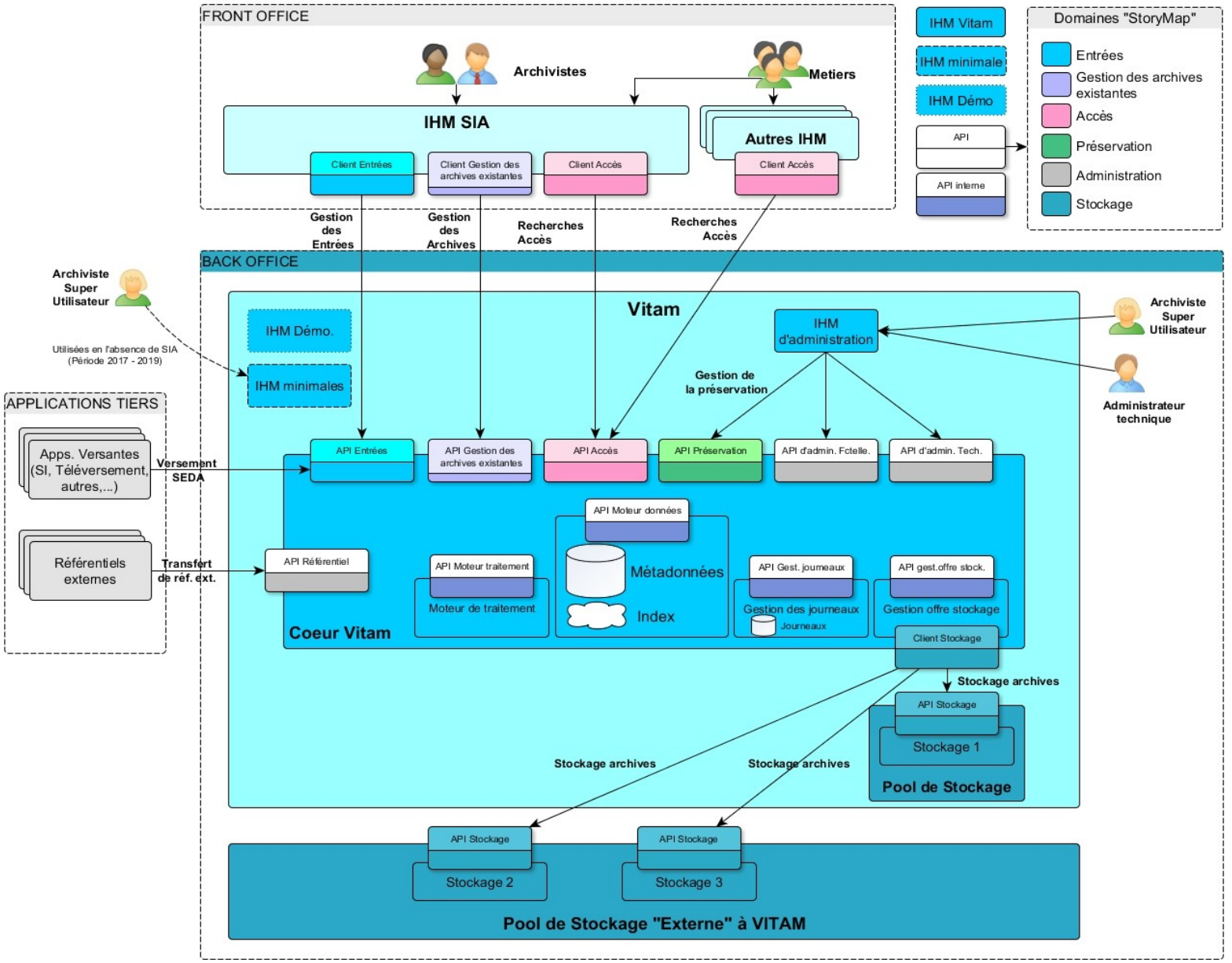
- Du fait de l'importance des 3 ministères porteurs dans le programme Vitam, des bilatérales fonctionnelles et techniques sont prévues pour collecter de manière plus précise les besoins

L'architecture générale de la solution logicielle Vitam

Interfaçage, Indépendance, Réutilisation, Sécurité



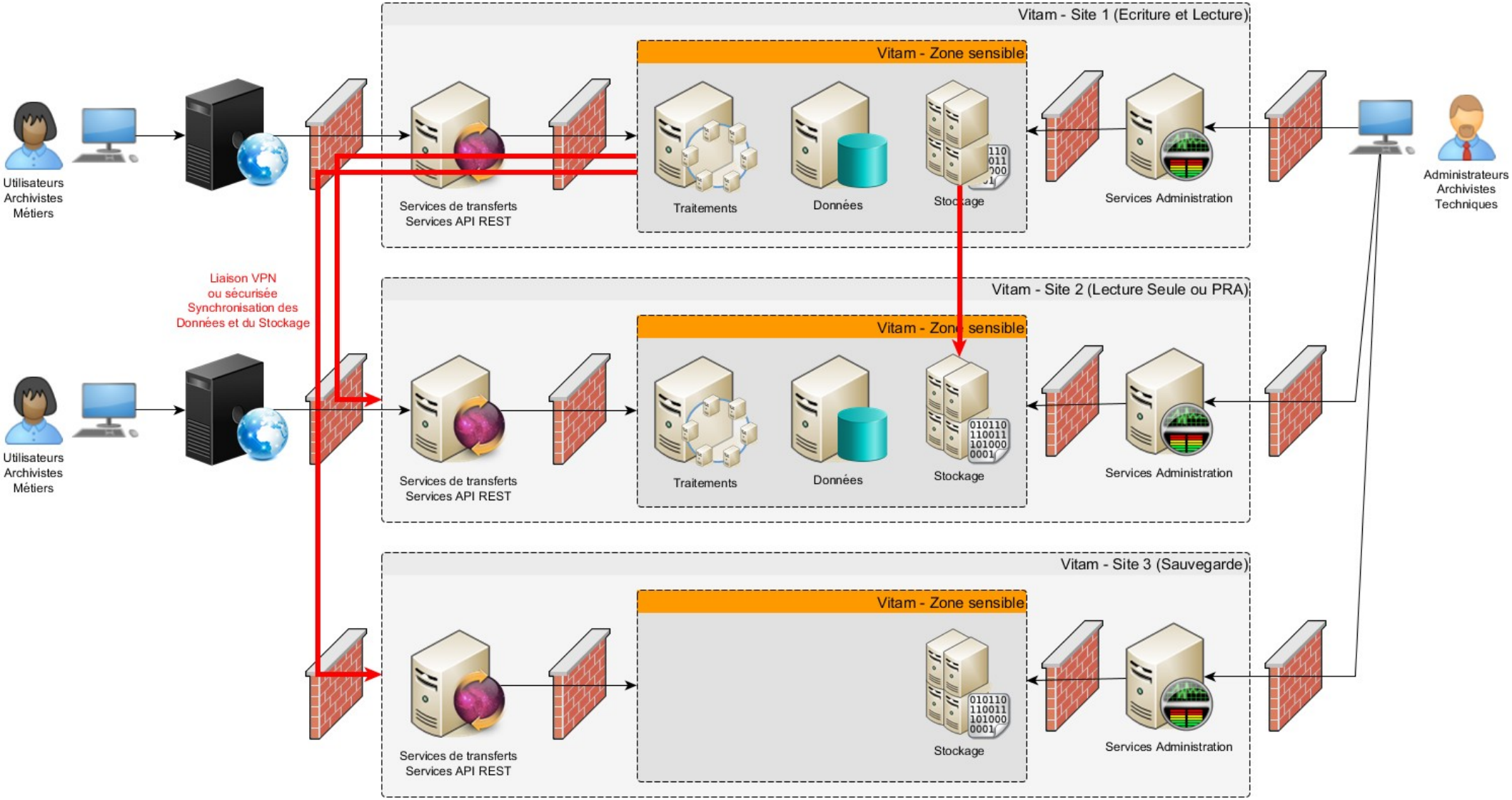
L’architecture fonctionnelle de la solution logicielle Vitam



Réseau - Zoning

- VITAM est un back-office
  - Accès via les applications métiers ou le SIA
  - Pas d'accès directement par les utilisateurs (hors administration)
- Éléments de séparation envisagés
  - En entrée de la DMZ Vitam
    - *API de versement, accès et gestion*
    - *IHM d'administration Vitam*
  - Distinction des rôles
    - *Administrateurs systèmes via outils spécialisés : par site a priori obligatoire*
    - *Administrateurs techniques et fonctionnels via IHM : unifié ?*
  - Les données à protéger sont présentes
    - *Sous forme pérenne dans les offres de stockages présents sur les différents sites*
    - *Sous forme requêtable dans le moteur de données (base de données et indexation)*
    - *Ils ne doivent pas être accessibles depuis l'extérieur (Données) et uniquement par Vitam quant à l'extérieur (Stockage sur autre site)*

Réseau - zoning



Réseau - Zoning

- Il est envisagé d'être compatible avec le modèle de zoning suivant :
  - Zone Frontale/DMZ : héberge les guichets d'accès exposant les API REST externes (vers les applications) + service de transfert
  - Zone « Applicative » : héberge tous les composants applicatifs exposant les API REST internes
    - *Il est souhaitable d'avoir une zone/VLAN réservé pour l'accès aux bases de données qui ne serait accessible qu'à partir de l'API du moteur de données*
  - Zone technique : héberge les rebonds/bastions ainsi que le moteur de déploiement
  - Pour les échanges « stockage » (voir le slide suivant)

Réseau - Zoning
Zoom sur le moteur de stockage

- Le moteur de stockage est sur le site primaire et doit interagir avec les offres de stockages qui sont réparties sur les sites
  - Le moteur de stockage se trouve en zone applicative
  - Les offres de stockage se trouveront dans des zones sur chacun des sites (est-ce la même zone ou une autre?)
- Comment faire la liaison entre les 2 composants
  - En ressortant du datacenter (proxy en zone frontale) puis en rerentrant « par le haut » dans le datacenter cible
  - Par un lien 'direct' entre les 2 zones (un peu comme des liens d'infrastructure)
- Suite à l'atelier 3, les 2 besoins ont été exprimés
  - L'offre de stockage de référence de Vitam devrait être compatible avec les 2 visions

Système

- Virtualisation
  - Quelle solution ?
  - Solution Iaas ?
- OS utilisé : Centos
  - Version de Centos utilisé : 6 ? 7 ?
  - Quelle est la roadmap ?
- Durcissement OS
  - Noyaux durcis ?
  - Mise en œuvre de modules de sécurité linux (LSM) ?
- Élévation de privilèges
  - Découpage des droits entre les administrateurs systèmes (root) et les administrateurs applicatifs
    - *Les populations d'administrateurs systèmes et applicatifs sont-ils différents ?*
  - Authentification en 2 temps ? (nominative puis élévation de privilèges) su ? Sudo ?
  - Dans le cas d'outils de déploiements, force-t-on un utilisateur nominatif à insérer son mot de passe Unix ?

Système - Modèle de déploiement

- Les besoins associés aux activités de déploiement et configuration sont les suivants :
  - Assurer un déploiement cohérent des éléments binaires VITAM sur un environnement donné (pas de multi-environnement à ce niveau là)
  - Instancier la configuration des éléments binaires VITAM pour chaque cible de déploiement
- Les besoins sous-jacents sont :
  - Connaître la topologie de l'environnement VITAM (à déployer et déployé)
  - Connaître l'inventaire des éléments VITAM installés
  - Tirer partie des fonctionnalités d'élasticité fournies par la plateforme sous-jacente si elles sont supportées
- Objectifs :
  - Être capable de répondre à l'installation de « petits » VITAM (jusqu'à 1 VM) comme les VITAM plus conséquents (> 30 – 50 VM)
  - Pour les petits VITAM, on peut faire le parallèle avec devstack ou packstack (distributions facilitant le déploiement d'Openstack y compris sur une VM)

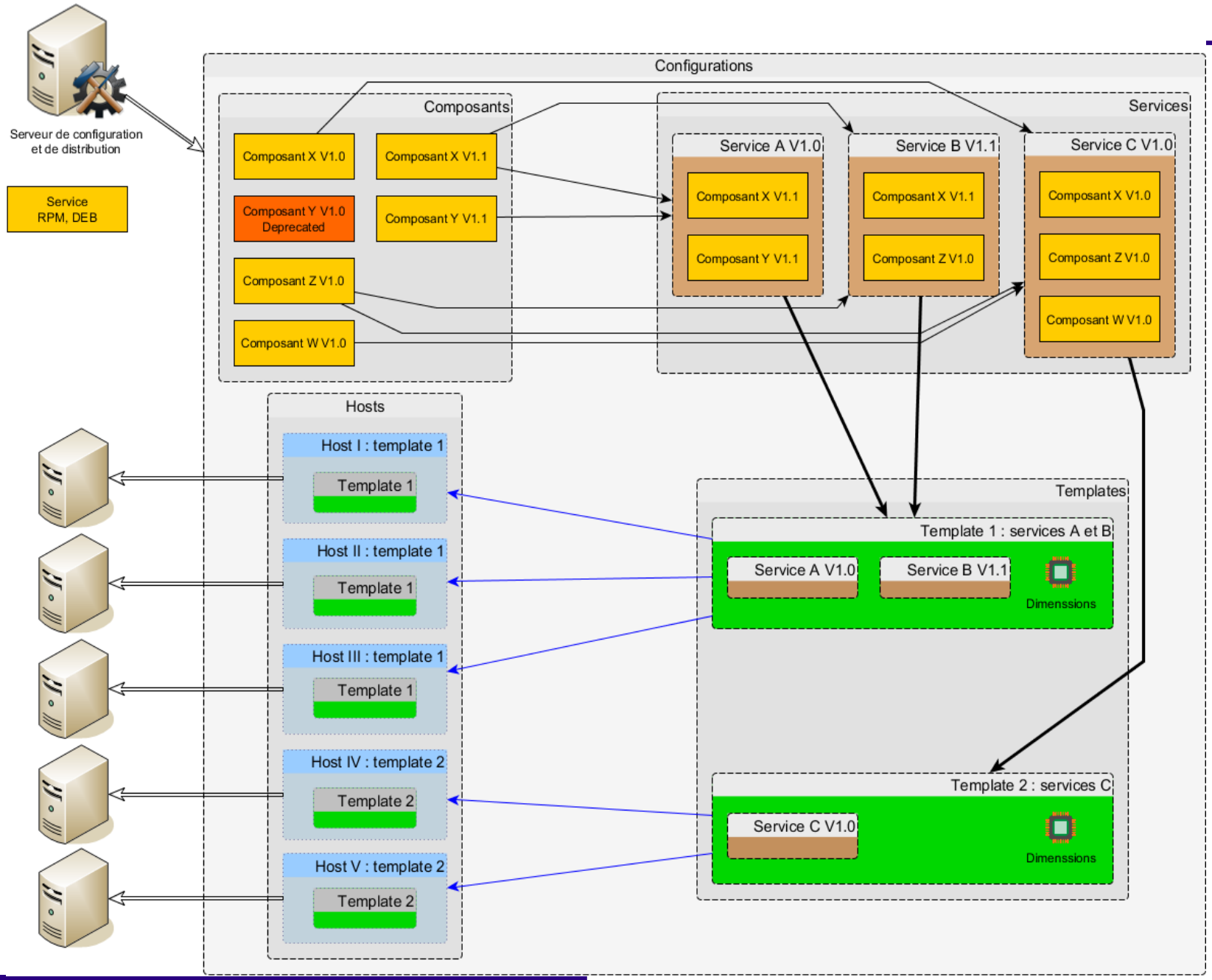
Système - Déploiement

- Système de déploiement
  - Mode « push » pressenti pour des raisons de sécurité : du serveur de déploiement vers les systèmes cibles
  - Limité à un site donné .
    - *Il pourra être nécessaire d'avoir des informations d'autres sites (ex : version des offres de stockage)*
    - *Cet échange de données est pour l'instant considéré comme « manuel » (ex : export par un administrateur technique/système du site distant, envoi du fichier, import dans l'outil de déploiement du site primaire)*
  - Système pressenti : Ansible
- Intégrité des composants livrables
  - Intégrité des binaires livrés ?
  - Signature des packages ? (type GPG)
- Système de configuration
  - Est une sous-partie du processus de déploiement
  - Gestion des secrets
  - Solution envisagé : Basé sur le même outil que le système de déploiement

Objets manipulés

- Modèle de données du déploiement
  - Composants : atome pour l'intégrateur (et pour toute personne désirant construire des services)
  - Services :
    - *Atome de déploiement pour une vision « production »*
    - *Sur un serveur donné, une seule version d'un service peut être déployé*
  - Templates :
    - *Constitue une unité de distribution sur un serveur*
    - *Est un ensemble de services*
  - Plate-forme : Bornes et seuils d'usage de la plate-forme IaaS
  - Servers :
    - *Contient les éléments de paramétrages du serveur (nom, IP)*
    - *Est lié à un template (et un seul)*
  - Infrastructure : nom, version, description, propriétés
    - *Contient les éléments de paramétrages de l'infrastructure (Firewall, load-balancer)*

Déploiement
Principe de configuration



- n Composants => 1 Service

- n Services => 1 Template

- 1 Template => n Hosts

En V3 : élasticité
- 1 Template => 1 Plate-forme (décrira les bornes des Hosts autorisés)

- 1 Plate-forme => n Hosts (dynamiques)

(ou extension de Template à Plate-forme)

Déploiement

- Lors d'un déploiement initial
  - Configuration des Composants et Services fournis par Vitam
  - Ajout possible de Composants
  - Modèle de Templates fournis par Vitam
  - Création de « vos » Templates sur la base d'un ou plusieurs templates de base
  - Création de vos Hosts (quasi automatique en V3)
  - Installation des Hosts (OS, bootstrap Vitam)
  - Lancement du déploiement (binaires et configurations)
- Lors d'une mise à jour
  - Mise à jour des Composants et Services fournis par Vitam
  - Mise à jour de vos Templates (versionning)
  - Déploiement contrôlé par Vitam
    - *Pré-actions globales, par Service, par Composants*
    - *Post-actions globales, par Service, par Composants*

Des règles ou recommandations existantes dans vos SI ?

Méthode de déploiement

- Déploiement « in place » vs « out of place » des binaires
  - In place : On remplace les binaires précédents
  - Out of place : On installe les binaires sur un emplacement différent (quitte à avoir un chemin invariant via un système d'alternatives)
    - *Exemple : OpenJDK, OpenSSL*
- Fichiers de configuration
  - Les fichiers de configuration sont instanciés via un modèle de templating
  - Toute modification doit être faite par le moteur de templating sous peine de ne pas être prise en compte au prochain upgrade
    - *A noter qu'il peut y avoir une historisation du fichier de configuration « divergeant » du standard (comme les rpmsave/rpmnew)*
  - Ces problématiques sont vraies aussi bien en installation « in place » que « out of place »

Des règles ou recommandations existantes dans vos SI ?

Retour arrière

- Composants stateless
  - « In place » : on réinstalle la version précédente (via l'outil de déploiement)
  - « Out of place » : on change le système d'aliasing pour pointer sur la version précédente
- Composants stateful (SGBD)
  - Même problématique dans les 2 stratégies de déploiement
  - Pour les mises à jours sans impacts sur les données, même cas que les composants stateless
  - Pour les mises à jours ayant un impact sur les données
    - *Sauvegarde des données associées au composant (étape longue)*
    - *Upgrade des binaires*
    - *Phase de mise à jour des données*
    - *En cas de rollback, restauration des données puis retour arrière des binaires (cf stateless)*

Implémentation du déploiement

- Système de packaging

	Avantages	Inconvénients
Système de packaging de la distribution (RPM/DEB)	Nativement : · Signature des packages (clé gpg) · Inventaire et de l'intégrité des fichiers installés (rpm -V / dpkg -V) · Dépendances · Gestion unifiée des packages sur un OS	· Installation à faire par root · Le chemin d'installation est fixé au moment du packaging · L'utilisateur propriétaire des fichiers est fixé au moment du packaging · Une seule version d'un package installée à un moment donné · Pour le projet Vitam : Maintenir 2 souches de packages (DEB+ RPM)
Développement spécifique	· Possibilité de faire l'installation avec l'utilisateur applicatif · Choix du chemin d'installation à l'installation · Possibilité d'installer plusieurs versions d'un packages à un moment donné	· Pas de gestion unifiée de packages avec l'OS · Pour le projet Vitam : Système de packaging à développer

Arborescence d'installation

- Si on part sur un système de packaging OS, les chemins d'installation seront fixés
- Pour des questions de points de montage, il nous semble utile de différencier
  - « Configuration et binaires »
  - « Log »
  - « Données »
- 2 approches possibles
  - Respect strict de la FHS
    - *« Configuration et binaires » : /opt/vitam/<composant>*
    - *« Log » : /var/opt/vitam/<composant>/log*
    - *« Données » : /var/opt/vitam/<composant>/lib*
  - Normes hors FHS mais plus « intuitive »
    - *« Configuration et binaires » : 2 possibilités*
      - */vitam/<composant> (avec en dessous bin, etc)*
      - */vitam/produit/<composant> (avec en dessous bin, etc)*
    - *« Log » : /vitam/log/<composant>*
    - *« Données » : /vitam/data/<composant>*

Utilisateurs d'exécution

- Si on part sur un packaging OS, les utilisateurs Linux d'exécution sont fixés lors du packaging

- 2 stratégies sont envisageables :
  - 1 utilisateur « global » (ex : vitam) pour tous les services
    - *Cas général : peu de services déployés sur un serveur donné*
  - 1 utilisateur par service
    - *Permet une segmentation en cas de regroupement de service*
      - *Plus sécurisé mais est-ce justifié ?*
    - *Est plus complexe à administrer*

Authentification

- Protocoles d'échanges entre les SI et Vitam
  - Protocoles envisagés
    - *HTTPS pour les API REST*
    - *Autres protocoles pour des flux de fort dimensionnement (FTPS, SFTP, WAARP, …)*
  - VITAM pourrait proposer
    - *Une interface (au sens logiciel) permettant de s'interfacer avec la solution préconisée par la DSI*
    - *Une implémentation de référence (sans doute authentification par certificat client)*
- Authentification dans les IHM Vitam
  - VITAM pourrait proposer
    - *Une interface (au sens logiciel) permettant de s'interfacer avec la solution préconisée par la DSI*
    - *Une implémentation de référence (solution à définir)*

Authentification

- Protocoles d'échanges dans Vitam (entre les modules)
  - Protocoles envisagés
    - *HTTPS ou HTTP*
      - *Faut-il chiffrer entre la zone « Frontale/DMZ » et la zone applicative ?*
      - *Entre les sites (communication inter-sites), les échanges seront chiffrés, même si mise en place d'un VPN*
      - *Pas de chiffrement entre composants de la zone applicative*
  - Mécanisme d'authentification envisagé
    - *Secret global de plateforme permettant d'identifier la plateforme*

Haute disponibilité/Répartition de charge

- 3 types de besoins de haute disponibilité/répartition de charge
  1. Accès aux « API externes » VITAM à partir des applications Front Office
  2. Accès aux offres de stockages (site distant) à partir du moteur de stockage
  3. Accès entre composants applicatifs
- Pour les 2 premiers besoins, la solution serait une solution « classique » de LB/HA en coupure
  - Sans doute en HTTPS (authentification par certificat client) dans le cas 1 . Si l'affinité de session s'avérait nécessaire (à définir durant les développements), il serait possible de le faire par SSLID
  - Dans le cas de l'implémentation de référence VITAM de l'offre de stockage,cela serait sans doute également par LB/HA sur HTTP ou HTTPS

Haute disponibilité/Répartition de charge

- Accès entre composants applicatifs
  - La solution classique de LB/HA en coupure peut être implémenté de 2 manières :
    - *1 LB/HA « central » qui rend le service pour toute la plate forme Cette solution est relativement facilement exploitable mais est peu scalable car la fonction est centralisée et prend notamment la totalité du trafic de la plate forme*
    - *Des LB/HA « distribués » pour chaque service. Cette solution est plus scalable car le flux réseau est plus réparti (ne passse plus par un seul point) mais est plus difficilement exploitable (une même fonction est répartie à différents emplacements de la plate forme)*

Haute disponibilité/Répartition de charge

- Accès entre composants applicatifs
  - Une solution de type « Service Registry » que l'on retrouve dans les architectures micro-services :
    - *Quand un client désire se connecter, il contacte un composant intermédiaire qui connaît à un moment donné la liste des instances fournissant le service. Puis le client fait un 2ème appel à l'instance du service*
    - *Cette solution permet d'avoir un service centralisé (exploitable) tout en limitant les flux sur ce service .*
- Quels sont vos expériences/retours ?
- Le choix de la solution devrait être fait avant la bêta

Exploitabilité

- Sauvegarde
  - Vu la volumétrie, il n'est pas prévu de « sauvegarde » au sens classique du terme pour les données et méta-données mais plutôt de réplication applicative
  - Par contre, il est nécessaire de sauvegarder, comme pour tout SI, un certain nombre de stock de données (ex : référentiels)
    - *Des scripts d'exports/import seront mis à disposition pour fournir sous forme « fichiers » le contenu des stocks « critiques » à sauvegarder*
    - *Le système de sauvegarde SI peut ensuite sauvegarder ces fichiers*

- Supervision
  - VITAM disposera d'une supervision « applicative » qui sera utilisé pour de la prise de décision interne à Vitam et qui sera partiellement exposé dans les IHM d'administrations
  - Pour permettre une remontée d'informations dans la supervision SI, nous proposons SNMP/API Rest et les logs comme canaux .

Transferts de fichiers

- Besoin d'un outil de transfert de fichiers
  - Dans le cadre des versements d'archives ou de la restitution de masse
  - Securité : Sécurité du transfert, identification du partenaire
  - Exploitation : Suivi du transfert, reprise sur incident
  - Applicatif : Capacité de déclenchement d'actions avant/après/en cas d'erreur

- Solutions pressenties
  - API HTTP REST
  - FTP ou FTPS
  - Optionnellement WAARP

Traçabilité des échanges

- Lors d'une requête, une traçabilité peut être active
  - Tracer l'activité, les données ou objets numériques accédées
    - *Les journaux du SAE Vitam doivent répondre*
- Lors d'une réponse, la réponse peut être à valeur probante
  - Les objets ne sont pas signés dans le SAE car ne résiste pas au temps
    - *Durée de validité d'un certificat, fragilité de la chaîne de confiance, fragilité des algorithmes, ...*
    - *La preuve est donc systémique (droit public) et non par item*
  - L'activité (cycle de vie, événements) à l'intérieur du SAE doit permettre de conserver la valeur probante
    - *La preuve selon la NF Z 42-013 est systémique*
      - *Horodatage et empreinte (hash et/ou chaînage)*
      - *Journalisation séquentielle et horodatée des événements (cycle de vie, événements)*
        - » *Les journaux sont eux-mêmes archivés avec une preuve de non modification (chaînage ou Arbre de Merkle)*
  - Une réponse peut intégrer une signature par le SAE
    - *Elle a une valeur ponctuelle de garantie de la validité des informations transmises par le SAE*

Sécurisation des supports

- Résistance au vol d'informations
  - Le chiffrement est un danger pour la pérennité de l'information
    - *Perte du secret, corruption de l'algorithme*
      - *Mais ce risque est lié à la durée d'utilisation*
    - *Si le chiffrement est applicatif*
      - *La durée est liée à l'application (ici > 20 ans)*
    - *Si le chiffrement est sur les supports*
      - *La durée est liée à la viabilité des supports (< 10 ans)*
  - Le chiffrement est possible sur des supports à durée de vie limitée
    - *Les supports rapides comme les disques sont de bons candidats*
      - *Rapidité des calculs*
      - *Durée de vie inférieure à 10 ans (voire 5 ans)*
      - *Chiffrement porté par le stockage (le changement de stockage change le chiffrement)*
    - *Les supports lents comme les bandes ne sont pas de bons candidats*
      - *Lenteur d'accès, durée de vie potentiellement > 10 ans, chiffrement par le lecteur (rendant difficile la garantie)*
      - *Une sécurité périmètrique est préférable (bunker)*
    - *Note : NF Z42-020 n'impose aucun chiffrement ni signature mais une garantie systémique via les journaux*