



**HAL**  
open science

# Conception et réalisation d'une solution d'automatisation des opérations d'administration pour infrastructures de stockage de données SAN au sein du groupe Banque Populaire

Julien Brusset

## ► To cite this version:

Julien Brusset. Conception et réalisation d'une solution d'automatisation des opérations d'administration pour infrastructures de stockage de données SAN au sein du groupe Banque Populaire. Technologies Émergentes [cs.ET]. 2016. dumas-01662500

**HAL Id: dumas-01662500**

**<https://dumas.ccsd.cnrs.fr/dumas-01662500>**

Submitted on 13 Dec 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS  
CENTRE REGIONAL ASSOCIE DE TOULOUSE

---

**MEMOIRE**

présenté en vue d'obtenir

le **DIPLOME D'INGENIEUR CNAM**

**SPECIALITE : Informatique**

**OPTION : Informatique, Réseaux, Systèmes et Multimédia (IRSM)**

par

**Julien BRUSSET**

---

Conception et réalisation d'une solution d'automatisation des opérations d'administration pour infrastructures de stockage de données SAN au sein du groupe Banque Populaire

**Soutenu le 14 janvier 2016**

---

**JURY**

<b>PRESIDENT :</b>	<b>M. Yann Pollet</b>	<b>Professeur des Universités</b>
<b>MEMBRES :</b>	<b>M. Thierry Millan</b>	<b>Maitre de conférences</b>
	<b>M. Bertrand Raiff</b>	<b>Maitre de conférences</b>
	<b>Mme Delphine Dupin</b>	<b>Expert système</b>
	<b>M. Eric Sebban</b>	<b>Responsable de service</b>

## **Remerciements**

Je tiens tout d'abord à remercier Cécile, ma compagne, pour son soutien lors des nombreuses années de préparation de ce diplôme, sa présence et le réconfort qu'elle a su m'apporter lors des moments de doute.

Je remercie également tous ceux, professeurs, personnels administratifs et auditeurs, qui m'ont accompagné ou que j'ai croisés pendant ces dix années passées au sein du CNAM.

Je remercie Mme Delphine Dupin et M. Eric Sebban pour m'avoir donné l'opportunité de travailler sur un projet aussi intéressant et pour toute l'aide qu'ils m'ont apportée au cours de la réalisation de ce mémoire.

Enfin, je dédie ce travail à la mémoire de mon père, Jacques Brusset.

## Liste des abréviations

API	Application Program Interface
BASH	Bourne Again Shell
BPCE	Banque Populaire / Caisse d'Épargne
CPU	Central Processing Unit
DAE	Disk Array Extension
EFD	Enterprise Flash Disk
FAST	Fully Automated Storage Tiering
FC	Fibre Channel
HACMP	High-Availability Cluster MultiProcessing
HTML	Hypertext Markup Language
HTTP	Hypertext Transfert Protocol
IBP	Informatique Banque Populaire
ISL	Inter-Switch Link
JSON	Javascript Object Notation
LUN	Logical Unit Number
MAC	Media Access Control
NAS	Network Attached Storage
OS	Operating System
PVM	Python Virtual Machine
RAID	Redundant Array of Independent Disks
RAIN	Redundent Array of Independent Nodes
REST	Representational State Transfer
RHEL	Redhat Enterprise Linux
SAN	Storage Area Network
SATA	Serial Advanced Technology Attachment
SOAP	Simple Object Access Protocol
SDN	Software Defined Network
SDS	Software Defined Storage
SMI/S	Storage Management Initiative Specification
SNIA	Storage Network International Alliance
SPS	Standby Power Supply
SSD	Solid State Disk

SSH	Secure Shell
UPS	Uninterruptible Power Supply
VBS	Visual Basic Script
VIO	Virtual IO
VSAN	Virtual SAN
XML	Extensible Markup Language
WORM	Write Once Read Many
WWN	World Wide Name

## **Glossaire**

Cloud Computing : Exploitation de la puissance de calcul ou de stockage de serveurs informatiques distants par l'intermédiaire d'un réseau, généralement Internet.

Hyperviseur : Plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une machine physique en même temps.

Instance : Synonyme de machine virtuelle employé dans le monde du Cloud Computing.

Network Attached Storage (NAS) : Serveur de fichiers autonome, relié à un réseau dont la principale fonction est le stockage de données en un volume centralisé pour des clients réseau.

Orchestration : Processus automatique d'organisation, de coordination, et de gestion de systèmes informatiques complexes, de middleware, et de services.

Storage Area Network (SAN) : Réseau spécialisé permettant de mutualiser des ressources de stockage.

Virtualisation : Ensemble de techniques permettant d'héberger plusieurs systèmes d'exploitation sur une même couche physique.

Virtual Machine (VM) : Acronyme désignant les machines virtuelles hébergées sur un hyperviseur.

# Table des matières

Remerciements .....	2
Liste des abréviations .....	3
Glossaire .....	5
Table des matières .....	6
Administration des infrastructures de stockage, un enjeu stratégique .....	10
<b>I CONTEXTE DE L'ETUDE .....</b>	<b>12</b>
I.1 PRESENTATION DE LA SOCIETE I-BP (INFORMATIQUE BANQUE POPULAIRE) .....	12
I.1.1 Le groupe BPCE .....	12
I.1.2 La société i-BP dans le groupe BPCE .....	13
I.1.3 Statuts .....	14
I.1.4 Implantation géographique .....	14
I.1.5 Organisation générale de la société i-BP .....	14
I.1.5.1 L'équipe VSS .....	15
I.2 AVANT-PROPOS : LES RESEAUX DE STOCKAGE DE DONNEES SAN .....	16
I.2.1 Qu'est-ce qu'un réseau de stockage SAN ? .....	16
I.2.2 Caractéristiques des réseaux SAN .....	16
I.2.2.1 Extensibilité .....	16
I.2.2.2 Disponibilité .....	16
I.2.2.3 Hétérogénéité .....	17
I.2.3 Spécificités des réseaux SAN .....	17
I.2.3.1 Mécanisme d'adressage .....	17
I.2.3.2 Zones logiques .....	18
I.2.3.3 Unité d'allocation de volumétrie .....	18
I.2.4 Une administration à parfaire .....	18
I.3 LES ENJEUX DE L'ADMINISTRATION DES INFRASTRUCTURES SAN .....	19
I.3.1 Complexité .....	19
I.3.2 Sécurité et fiabilité .....	19
I.3.3 Charge de travail .....	19
I.3.4 Traçabilité .....	20
I.3.5 Maintenance et évolutivité .....	20
I.4 PRESENTATION DU PROJET .....	21
I.4.1 Périmètre technique existant .....	21
I.4.1.1 Services de stockage de données .....	21
I.4.1.2 Répartition géographique du centre de données principal .....	21
I.4.2 Renouvellement technologique des infrastructures de stockage .....	22
I.4.3 Architecture de la solution technique retenue .....	22
I.4.4 Définition des standards et outils d'administration .....	24
<b>II ETAT DE L'ART DES SOLUTIONS DE « SOFTWARE DEFINED STORAGE » .....</b>	<b>25</b>
II.1 PRESENTATION DU « SOFTWARE DEFINED STORAGE » .....	25
II.1.1 Origine de l'appellation .....	25
II.1.1.1 Virtualisation du stockage, base du « Software Defined Storage » .....	25
II.1.1.1.1 Les bénéfices de la virtualisation du stockage de données .....	26
II.1.1.1.2 Les risques liés à la virtualisation du stockage de données .....	28
II.1.1.2 Le concept de « Software Defined Storage » .....	30
II.1.1.3 Spécificités de l'approche « Software Defined Storage » .....	31
II.1.1.4 Avantages des solutions de type SDS pour les entreprises .....	32

II.1.1.4.1	Rationalisation des modes d'administration.....	32
II.1.1.4.2	Automatisation et prise en compte de la logique métier .....	33
II.1.1.4.3	Unification des modes d'accès « bloc » et « fichier » .....	33
II.1.1.4.4	Amélioration de l'interconnexion .....	34
II.2	LES PRODUITS SDS DU MARCHE .....	35
II.2.1	EMC <sup>2</sup> VIPR.....	35
II.2.1.1	Description .....	35
II.2.1.2	Architecture de la solution .....	35
II.2.1.3	Fonctionnalités .....	36
II.2.1.4	Orientation et évolution.....	37
II.2.2	OpenStack Cinder .....	37
II.2.2.1	Description .....	37
II.2.2.2	Architecture de la solution .....	37
II.2.2.3	Fonctionnalités .....	38
II.2.2.4	Orientation et évolution.....	39
II.2.3	VMWare Virtual SAN .....	39
II.2.3.1	Description .....	39
II.2.3.2	Architecture de la solution .....	39
II.2.3.3	Fonctionnalités .....	40
II.2.3.4	Orientation et évolution.....	42
II.3	CRITIQUE DU MODELE « SOFTWARE DEFINED STORAGE » .....	43
II.3.1.1	Une maturité en devenir .....	43
II.3.1.2	Coûts prohibitifs .....	43
II.3.1.3	Un concept encore flou.....	44
II.4	CONCLUSION .....	45
<b>III</b>	<b>ANALYSE ET CONCEPTION D'UNE SOLUTION D'AUTOMATISATION .....</b>	<b>46</b>
III.1	ANALYSE DU BESOIN .....	46
III.1.1	L'existant et les défauts de l'existant.....	46
III.1.1.1	L'existant .....	46
III.1.1.2	Les défauts de l'existant.....	46
III.1.2	Cahier des charges .....	48
III.1.2.1	Fonctionnalités attendues.....	48
III.1.2.2	Contraintes et exigences.....	49
III.2	PLANIFICATION DU PROJET .....	52
III.2.1	Lotissement du projet.....	52
III.2.1.1	Lot A – Modules d'administration VPLEX.....	52
III.2.1.2	Lot B – Modules d'administration VMAX.....	52
III.2.1.3	Lot C – Modules d'administration CISCO .....	52
III.2.2	Sélection du cycle de développement.....	52
III.2.3	Planification .....	53
III.2.3.1	Ordonnancement des tâches du projet .....	54
III.2.3.2	Diagramme de GANTT .....	54
III.3	SPECIFICATIONS TECHNIQUES DU PROJET .....	55
III.3.1	Spécifications techniques VPLEX.....	55
III.3.1.1	Présentation du dispositif.....	55
III.3.1.1.1	Modules d'entrée/sortie Front-End.....	56
III.3.1.1.2	Modules d'entrée/sortie Back-End.....	57
III.3.1.1.3	Modules d'entrée/sortie WAN COM.....	57
III.3.1.2	Éléments de configuration VPLEX.....	57
III.3.1.2.1	Gestion de la volumétrie .....	57
III.3.1.2.2	Distributed device .....	58
III.3.1.2.3	Storage View.....	59
III.3.1.3	Normalisation des éléments de configuration.....	59
III.3.1.3.1	Structure des volumes de données .....	59
III.3.1.3.2	Nomenclature des objets VPLEX.....	60

III.3.1.4	Administration de la solution.....	60
III.3.1.4.1	Unisphere for VPLEX.....	60
III.3.1.4.2	VPLEX CLI (Command Line Interface).....	61
III.3.1.4.3	API VPLEX.....	61
III.3.2	Spécifications techniques VMAX.....	62
III.3.2.1	Présentation du dispositif.....	62
III.3.2.1.1	La baie système.....	62
III.3.2.1.2	Les baies de stockage ou «Storage bay ».....	63
III.3.2.2	Fonctionnalités implémentées.....	65
III.3.2.2.1	Configuration de type « Open ».....	65
III.3.2.2.2	Virtual Provisionning.....	65
III.3.2.2.3	FAST/VP.....	66
III.3.2.2.4	Timefinder SNAP/VP.....	68
III.3.2.3	Éléments de configuration VMAX.....	69
III.3.2.3.1	Auto-provisionnement VMAX.....	69
III.3.2.4	Normalisation des éléments de configuration.....	71
III.3.2.5	Administration de la solution.....	71
III.3.2.5.1	Solutions Enabler.....	71
III.3.2.5.2	Unisphere for VMAX.....	71
III.3.2.5.3	API VMAX.....	72
III.3.3	Spécifications techniques CISCO.....	73
III.3.3.1	Présentation du dispositif.....	73
III.3.3.1.1	Présentation du SAN Front-End.....	73
III.3.3.1.2	Topologie physique du réseau SAN Front-End.....	74
III.3.3.1.3	Configuration matérielle Cisco SAN Front-End.....	74
III.3.3.2	Éléments de configuration.....	76
III.3.3.2.1	Attribution des VSAN ID.....	76
III.3.3.2.2	Constitution des zones.....	76
III.3.3.3	Normalisation des éléments de configuration.....	77
III.3.3.4	Administration de la solution.....	77
III.3.3.4.1	Cisco DCNM (Data Center Network Manager).....	77
III.3.3.4.2	Cisco CLI.....	78
III.3.3.4.3	API Cisco.....	78
III.3.4	Choix du langage de programmation : Python.....	79
III.3.4.1	Historique du langage Python.....	79
III.3.4.2	Quelle version de Python choisir ?.....	79
III.3.4.3	Fonctionnalités du langage Python.....	80
III.3.4.4	Fonctionnement de l'interpréteur Python.....	80
III.3.4.5	Avantages de Python sur les autres langages de scripting.....	81
III.3.4.6	Implémentations du langage Python.....	82
III.3.4.7	Performance du langage Python.....	82
III.3.4.7.1	Cython.....	83
III.3.4.7.2	Numba.....	83
III.4	CONCEPTION DE LA SOLUTION D'AUTOMATISATION.....	85
III.4.1	Lot A – Conception des modules d'administration VPLEX.....	85
III.4.1.1	Cas d'utilisation.....	86
III.4.1.1.1	Création de volumes locaux.....	87
III.4.1.1.2	Création de volumes distribués.....	89
III.4.1.1.3	Création de Storage View.....	90
III.4.1.1.4	Ajout de volumes à une Storage View existante.....	92
III.4.1.1.5	Ajout d'initiateurs à une Storage View existante.....	93
III.4.2	Lot B – Conception des modules d'administration VMAX.....	95
III.4.2.1	Cas d'utilisation.....	95
III.4.2.1.1	Création de devices Symmetrix.....	95
III.4.2.1.2	Ajout des devices à un Storage Group.....	97
III.4.3	Lot C – Conception des modules d'administration Cisco.....	99
III.4.3.1	Cas d'utilisation.....	99
III.4.3.1.1	Création d'alias.....	100
III.4.3.1.2	Création de zone.....	101
III.4.3.1.3	Renommer un alias.....	104

III.4.3.1.4	Configuration de Port FC.....	105
III.5	REALISATION DE LA SOLUTION D’AUTOMATISATION .....	108
III.5.1	Préparation de l’environnement de développement.....	108
III.5.1.1	Mise à disposition des serveurs.....	108
III.5.1.2	Accès et habilitations .....	109
III.5.1.3	Ouvertures de flux.....	109
III.5.1.4	Choix de l’environnement de développement intégré (EDI) .....	110
III.5.1.5	Serveur de gestion de versions.....	110
III.5.1.6	Organisation du projet.....	111
III.5.1.7	Générateur de documentation .....	112
III.5.2	Réalisation de l’interface homme-machine .....	113
III.5.2.1	Diagramme de classes.....	113
III.5.2.1.1	L’interface HumanInterface.....	114
III.5.2.1.2	La classe CommandLineInterface.....	114
III.5.2.1.3	La classe GraphicUserInterface .....	115
III.5.3	Réalisation du lot A – VPLEX .....	116
III.5.3.1	Diagramme de classes.....	116
III.5.3.1.1	La classe VplexRestfulCommunicator.....	116
III.5.3.1.2	La classe VplexJSONParser.....	117
III.5.3.1.3	La classe VplexArray.....	117
III.5.3.1.4	La classe VplexArrayControler .....	118
III.5.4	Réalisation du lot B – VMAX .....	120
III.5.4.1	Diagramme de classes.....	120
III.5.4.1.1	La classe VmaxXMLParser .....	120
III.5.4.1.2	La classe VmaxArray.....	121
III.5.4.1.3	La classe VmaxArrayControler .....	121
III.5.5	Réalisation du lot C – Cisco .....	123
III.5.5.1	Diagramme de classes.....	123
III.5.5.1.1	La classe WWN .....	123
III.5.5.1.2	L’interface CiscoCommand et ses réalisations .....	124
III.5.5.1.3	L’interface ISANSwitch .....	124
III.5.5.1.4	Les classes abstraites AbstractCiscoSanSwitch et AbstractBrocadeSanSwitch .....	125
III.5.5.1.5	La classe CiscoNPIVSanSwitch .....	125
III.5.5.1.6	La classe CiscoNPVSanSwitch.....	126
III.5.5.1.7	La classe CiscoSanControler .....	126
III.5.6	Réalisation des lanceurs applicatifs .....	127
III.5.7	Réalisation des tests unitaires .....	129
III.6	BILAN.....	130
III.6.1	Rappel des objectifs du projet.....	130
III.6.2	Bilan du projet .....	130
III.6.3	Perspectives d’évolution .....	131
	Vers une transformation des modes d’administration en informatique ?.....	133
	Bibliographie .....	135
	Liste des figures.....	136
	Liste des tableaux .....	138

## **Administration des infrastructures de stockage, un enjeu stratégique**

Avec l'informatique, tout est automatique ! Voilà un lieu commun répandu s'il en est. Bien évidemment, il est absolument indéniable que l'essor de l'informatique a permis de traiter, et ce de manière automatique, un très grand nombre de tâches qui auparavant relevaient d'interventions humaines. Après tout, étymologiquement, le mot informatique ne signifie-t-il pas : automatisation de l'information ?

Cependant, il existe au moins une activité parmi la pléthore des métiers de l'informatique pour laquelle, paradoxalement, les évolutions technologiques de ces dernières années tendent à complexifier la tâche au quotidien. Cette activité est celle des ingénieurs dont la tâche est d'administrer les infrastructures spécialisées dans le stockage de données.

Avec l'explosion des nouvelles technologies, et d'Internet en particulier, les besoins en matière de stockage de données ne cessent de croître et les infrastructures devant les accueillir de se complexifier. La faute, notamment, à la multiplication d'architectures techniques et d'équipements spécialisés permettant d'accroître la capacité de stockage mais également de tendre vers toujours plus de sécurité et de disponibilité.

Cet empilement de strates, sorte de tour de Babel technologique, composé d'équipements issus de constructeurs différents, n'utilisant pas les mêmes modes d'administration et ne reposant pas sur les mêmes concepts, contraignent les ingénieurs stockage à jongler entre une multitude d'outils supposés simplifier leur activité au quotidien pour finalement mieux la complexifier.

Et que dire également de l'environnement économique dans lequel évoluent les sociétés de nos jours ? En ces temps de crise, les entreprises réduisent leurs embauches ainsi que le nombre de leurs prestataires de services. Les équipes sont donc contraintes à produire plus sans pour autant toujours en avoir les ressources humaines. Ce paradoxe pose inévitablement la question de la qualité, mais aussi de la sécurité des opérations d'administration.

Face à toutes ces problématiques, comment les ingénieurs stockage peuvent-ils garantir le niveau de fiabilité, de rapidité et de sécurité nécessaire à l'exercice de leur profession ? Avec quels moyens, autant organisationnels que techniques, peut-on atteindre ces objectifs ? C'est à toutes ces interrogations que le présent document va tenter de répondre.

Cette étude se compose de trois parties :

La première partie est consacrée au contexte de l'étude et regroupe toutes les informations relatives à la structure du groupe Banque Populaire ainsi qu'une présentation détaillée du projet ayant permis de réaliser ce document.

La deuxième partie, plus théorique, aborde l'état de l'art des solutions dites de « Software Defined Storage », ces nouveaux outils permettant d'assurer une administration automatisée des infrastructures de stockage de données, leurs concepts ainsi que leurs avantages et leurs inconvénients dans le cadre d'une utilisation en production.

Enfin, la troisième partie est, quant à elle, dévolue à la conception, la réalisation puis à l'implémentation de la solution présentée lors de la phase d'analyse. L'accent sera mis sur les problématiques inhérentes aux déploiements en environnements de production et aux besoins d'évolutivité qu'impose une telle solution.

# I Contexte de l'étude

## I.1 Présentation de la société i-BP (Informatique Banque Populaire)

### I.1.1 Le groupe BPCE

La société i-BP, société où le travail relatif à ce mémoire a été accompli, fait partie du groupe BPCE (Banque Populaire – Caisses d'Épargne), deuxième groupe bancaire en France. L'architecture de ce groupe repose sur trois niveaux distincts comprenant :

- Les deux réseaux coopératifs principaux, actionnaires à parité du groupe BPCE, que sont les 19 Banques Populaires et les 17 Caisses d'Épargne
- Un organe central chargé de la définition de la stratégie, de la gestion des politiques communes et des missions de contrôle
- Des filiales (Natixis, Crédit Foncier de France, Banque Palatine, etc.) qui portent les métiers que les Banques Populaires et les Caisses d'Épargne ne peuvent exercer seules, pour des raisons de taille critique, d'expertises ou de présence à l'international

Le tableau ci-dessous présente l'importance du groupe BPCE dans le paysage bancaire français au travers de quelques chiffres clés :

Chiffres clés
<b>36 millions</b> de clients
<b>8,8 millions</b> de sociétaires
<b>115 000</b> collaborateurs
<b>8 000</b> agences bancaires

Tableau 1 – Groupe BPCE, chiffres clés

L'organisation du groupe BPCE est illustrée grâce à la figure ci-dessous :

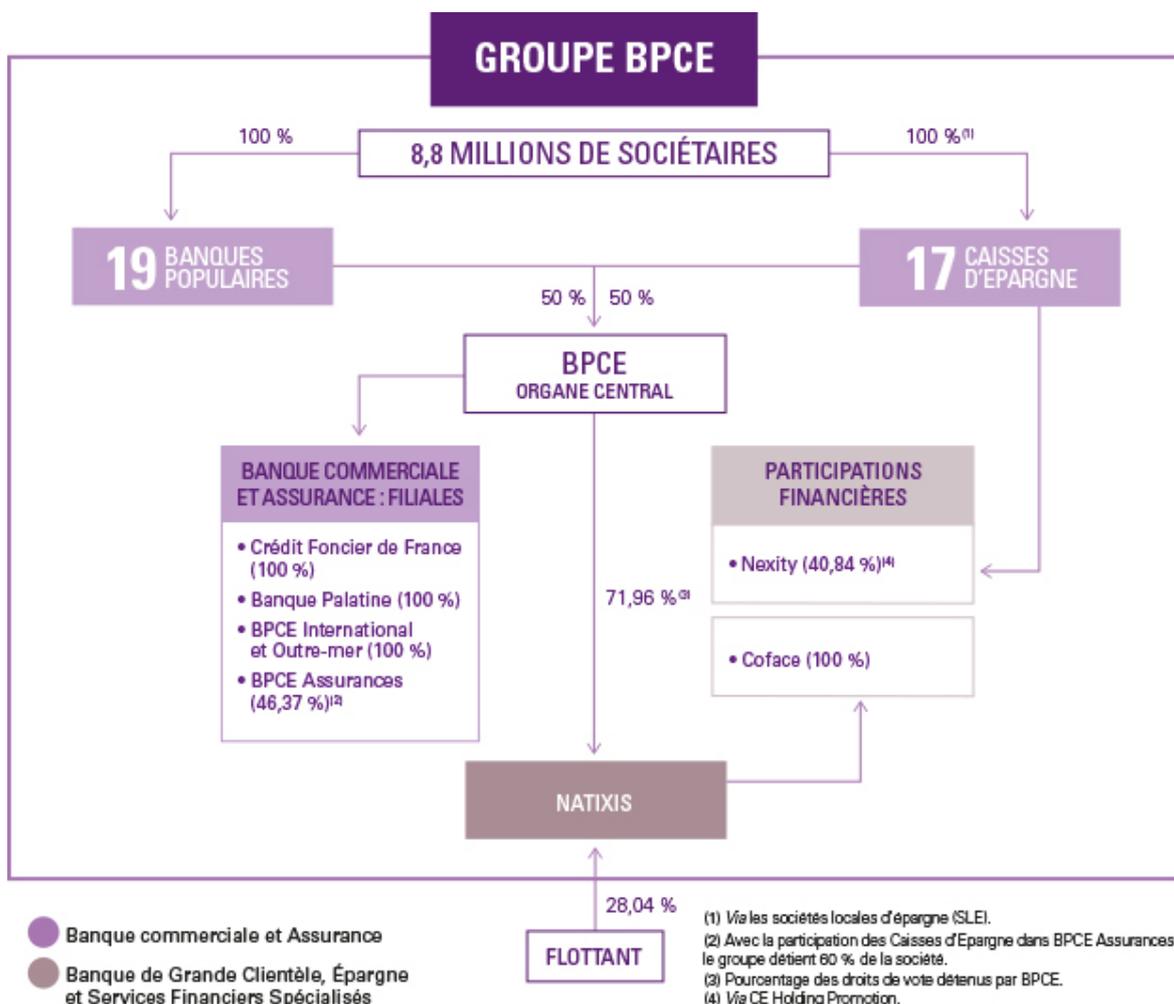


Figure 1 – Organisation du groupe BPCE

### I.1.2 La société i-BP dans le groupe BPCE

Entité du groupe BPCE, la société i-BP est en charge du développement et de l'infogérance informatique des 22 banques suivantes :

- 16 Banques Populaires régionales
- 3 Crédits Maritimes Mutuels
- La Banque de Savoie
- Le Crédit Commercial du Sud-Ouest-Pelletier
- La Banque Chaix

### I.1.3 Statuts

La société i-BP est une société anonyme (SA), à conseil d'administration, créée le 1<sup>er</sup> janvier 2001, au capital de 89 733 150 € ayant son siège à Montigny le Bretonneux (78180, Immeuble le Futura, 23 Place de Wicklow, SIREN 435 377 684 RCS Versailles).

Son activité principale consiste à, je cite : « commercialiser tous travaux et prestations de services informatiques ainsi que plus généralement de traitement de l'information ».

### I.1.4 Implantation géographique

La société i-BP est implantée sur 7 sites, regroupés en 4 établissements :

Etablissements	Sites
<b>i-BP Dijon</b>	Longvic (21)
<b>i-BP Nantes</b>	Saint-Herblain (44) Lyon-Gerland (69)
<b>i-BP Toulouse</b>	Balma (31) Perpignan (66)
<b>i-BP Versailles</b>	Paris (75) Saint Quentin en Yvelines (78)

Tableau 2 – Implantation géographique de la société i-BP

### I.1.5 Organisation générale de la société i-BP

L'organisation interne d'i-BP se structure en 7 directions principales rattachées au Directeur Général.

- Direction maîtrise d'ouvrage (MOA)
- Direction de l'édition logicielle (DEL)
- Direction services et architecture (DSA)
- Direction de l'infogérance (DDI)
- Direction générale adjointe (DGA)
- Direction des ressources humaines (DRH)
- Direction de l'audit (AUD)

Ces directions se structurent selon l'organigramme présenté ci-dessous :

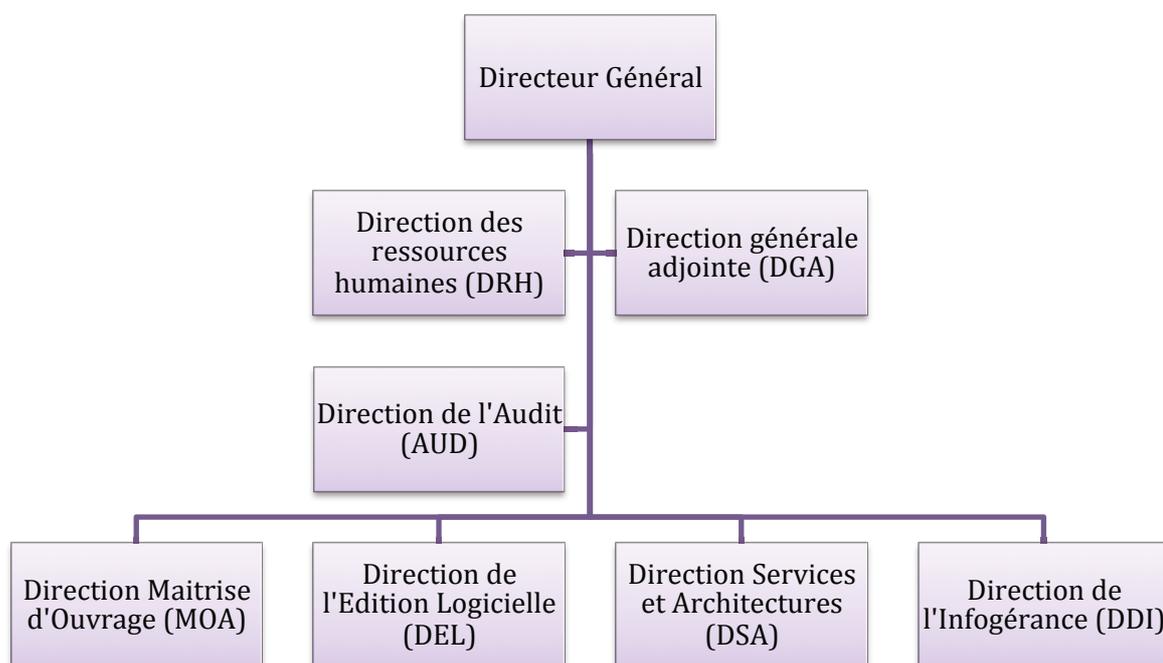


Figure 2 – Organisation générale de la société i-BP

### I.1.5.1 L'équipe VSS

Le présent stage a lieu au sein de l'équipe VSS (pour Virtualisation, Stockage et Sauvegarde).

Cette équipe de 15 personnes, sous la direction de M. Eric Sebban, regroupe les ingénieurs de production de niveau 2 et 3 en charge de l'administration et des évolutions d'infrastructures autour des périmètres de la sauvegarde, la virtualisation et le stockage de données. Cette équipe prend place au sein de la direction de l'infogérance (DDI).

Mme Delphine Dupin, experte stockage de niveau 3, est la tutrice désignée pour encadrer ce stage.

## **I.2 Avant-propos : les réseaux de stockage de données SAN**

### **I.2.1 Qu'est-ce qu'un réseau de stockage SAN ?**

Un réseau de type SAN (pour Storage Area Network) mutualise les ressources de stockage de données au sein d'une architecture spécialisée. Ce type de réseau rend possible la mise en relation, via un accès bas niveau, d'une machine cliente avec une ou plusieurs ressources disque distantes. On parle alors d'accès en mode bloc.

Formellement, pour le client de ces ressources, on parle alors d'initiateur, il n'y a pas de différence entre un volume de données local et un volume de données distant.

Ce type de dispositif technique permet de centraliser les données au sein de baies de disques tout en déchargeant les réseaux locaux d'entreprise des activités liées au stockage des données, mais également de la sauvegarde.

### **I.2.2 Caractéristiques des réseaux SAN**

#### **I.2.2.1 Extensibilité**

Les réseaux SAN se caractérisent en premier lieu par une forte capacité à étendre leur capacité de stockage. A l'aide des topologies de type « fabric » utilisées dans l'univers des réseaux SAN, il est aisé d'ajouter de nouveaux éléments comme notamment des baies de disques permettant de stocker toujours plus de données.

#### **I.2.2.2 Disponibilité**

Elément crucial s'il en est au sein d'un réseau dont la vocation première est le stockage de l'information : la disponibilité, c'est-à-dire la capacité à accéder aux données sans interruption de service possibles. Dans un réseau de type SAN, cette propriété revêt deux aspects.

Le premier d'entre eux est lié à la structure même des moyens d'interconnexions SAN ainsi qu'aux standards d'architecture en vigueur. Il est, en effet, la norme de constituer au moins deux « fabrics » totalement indépendantes et disjointes l'une de l'autre afin de réaliser le cœur de réseau SAN. La disponibilité des données est alors assurée par la redondance des chemins permettant l'accès à celles-ci.

Second aspect lié à la disponibilité des données : les mécanismes de redondance utilisés au sein des baies de stockage. Ces équipements permettent, notamment à l'aide de dispositifs tels que les grappes de disques RAID, de tolérer la perte physique d'un ou plusieurs composants sans impact sur l'activité de production.

Notons également, toujours au sujet de la disponibilité des données, la capacité des baies de stockage de dernière génération à répliquer d'importants volumes de données sur de grandes distances permettant ainsi d'accroître cette disponibilité en introduisant un facteur de résilience lié à la perte éventuelle d'une zone géographique donnée.

### **I.2.2.3 Hétérogénéité**

Les réseaux SAN étant totalement agnostiques au regard du type de données transportées, ces derniers peuvent donc véhiculer des blocs produits par toutes sortes de systèmes d'exploitation. Il est donc possible, au sein d'un même SAN, de stocker des données en provenance de systèmes d'exploitation Unix ou Windows. Il est également possible de véhiculer des échanges issus du monde mainframe IBM.

## **I.2.3 Spécificités des réseaux SAN**

Les réseaux SAN disposent de spécificités atypiques au regard des autres moyens d'interconnexions existants et notamment des réseaux locaux. Cette section va présenter certaines de ces spécificités.

### **I.2.3.1 Mécanisme d'adressage**

A l'instar d'un réseau de type Ethernet qui utilise une adresse MAC pour désigner de manière explicite un équipement connecté sur ce dernier, un réseau SAN utilise un mécanisme analogue. Les adresses MAC sont ici remplacées par ce que l'on nomme un WWN (World Wide Name).

Un WWN est une adresse codée sur 64 bits qui, tout comme une adresse MAC, comporte une portion réservée au constructeur ayant produit l'équipement ainsi qu'une partie dévolue à l'identification universelle du matériel.

### **I.2.3.2 Zones logiques**

Sans configuration logicielle préalable, les WWN présents au sein d'un réseau SAN ne peuvent communiquer entre eux. Il s'agit là d'un point de divergence vis-à-vis des réseaux de type Ethernet et qui constitue une forme de mécanisme de sécurité. Afin qu'une machine reçoive bien la volumétrie qu'elle est supposée détenir, on crée une zone logique qui permet de déclarer de manière explicite qui voit quoi au sein d'un réseau SAN. Cette étape est nommée « zoning ».

### **I.2.3.3 Unité d'allocation de volumétrie**

Les réseaux SAN et les baies de stockage qui y sont raccordées permettent de distribuer des unités de stockage disque de tailles et de construction variables que l'on nomme LUN (pour « Logical Unit Number »). Ce sont ces LUN qui seront attribués à un serveur afin qu'il puisse y stocker ses données. Une part non négligeable du temps de travail d'un ingénieur stockage est dévolue à la création, puis à la livraison de ces LUN.

### **I.2.4 Une administration à parfaire**

L'administration d'un réseau de type SAN, à ceci de particulier qu'elle repose sur la gestion de multiples équipements. En premier lieu, les commutateurs SAN qui constituent le réseau en tant que tel, mais aussi et surtout les baies de stockage qui sont les équipements qui vont distribuer les LUN au sein du réseau. Ces baies fournissent également les services réplication de données attendus d'un tel matériel.

Cet assemblage de technologies étant rarement le fruit d'un seul et même constructeur, les moyens permettant l'administration de ces équipements ne reposent, traditionnellement, pas sur les mêmes principes ni sur des outils communs. Cette situation tend à rendre la tâche quotidienne des ingénieurs stockage plus compliquée mais également plus risquée : la multiplication des interventions humaines, par nature faillibles, augmente sensiblement le risque de l'activité d'administration.

## **I.3 Les enjeux de l'administration des infrastructures SAN**

### **I.3.1 Complexité**

L'administration d'une infrastructure de stockage SAN est une chose complexe. Il y a d'une part la complexité induite par la multiplicité des équipements à gérer, mais également les différences existant entre ces équipements (différentes versions d'un produit, différents constructeurs, différents modes d'administration, etc.) qui sont autant d'aspects qui rendent la tâche de l'ingénieur stockage ardue.

La conception d'un système d'automatisation et de gestion d'infrastructure doit prendre en compte cette problématique : comment peut-on réduire la complexité inhérente aux infrastructures de stockage de données ?

### **I.3.2 Sécurité et fiabilité**

Le stockage des données d'une entreprise revêt une importance capitale pour celle-ci. Cette assertion est d'autant plus vraie lorsque l'entreprise en question appartient au secteur bancaire. Or, les opérations d'administration effectuées quotidiennement par les équipes exploitant les équipements de stockage SAN peuvent, en cas d'erreur humaine par exemple, avoir des conséquences aussi immédiates que dramatiques : suppression des volumes de données d'un serveur, écrasement des données suite à l'utilisation inappropriée des capacités de copies d'une baie de disques, etc. Ainsi, comment peut-on sécuriser l'exploitation des équipements de production et fiabiliser ces processus ?

### **I.3.3 Charge de travail**

La charge de travail moyenne des acteurs du stockage de données en entreprise ne cesse d'augmenter au gré des besoins, toujours croissants, en espace disque. Cette demande va de pair avec l'installation de nouveaux équipements qui génèrent à leur tour une charge de travail plus importante du fait de leur exploitation. La tendance conjoncturelle tend également à augmenter ce phénomène : en ces temps de crise, les entreprises réduisent leurs embauches ainsi que le nombre de leurs prestataires de services. Les équipes sont donc contraintes à produire plus sans pour autant toujours en avoir les ressources humaines. Aussi, dans le cadre d'un tel projet, quels moyens autant techniques qu'organisationnels peut-on mettre en œuvre afin de réduire la charge de travail globale ?

### **I.3.4 Traçabilité**

Du fait de la nature stratégique de l'activité du stockage de données, la traçabilité des opérations d'administration devient, elle aussi, stratégique. En cas d'erreur ou d'anomalie, il est crucial de disposer d'informations précises et fiables permettant d'analyser « post mortem » les causes profondes de ces problèmes dans le cadre d'un processus d'amélioration continue. Tout système permettant l'exploitation de ce type d'équipement doit prendre en compte cet impératif : comment garantir la traçabilité des opérations d'exploitation ?

### **I.3.5 Maintenance et évolutivité**

La conception d'un système de gestion automatique ainsi que les processus qui l'accompagnent ne représentent qu'une partie de la solution au problème initial. Bien trop souvent, la conception de ce type d'outil reste uniquement centrée sur la conception en tant que telle sans faire cas de l'après. Que se passe-t-il une fois le système en production ? Celui-ci est-il facilement maintenable, peut-on y rajouter des fonctionnalités sans pour autant tout revoir ? Ce projet est-il seulement documenté comme il se doit ?

## **I.4 Présentation du projet**

### **I.4.1 Périmètre technique existant**

La société i-BP assure, comme indiqué plus haut dans le présent document, une activité d'infogérance des moyens informatiques du groupe Banque Populaire. Cette section dresse l'inventaire de ces moyens ainsi que leur positionnement géographique.

Les éléments présentés dans cette section ont été établis sur un relevé de situation du parc effectué en janvier 2013, soit quelques mois avant le démarrage du projet.

#### **I.4.1.1 Services de stockage de données**

Les grandes fonctions du stockage distribué de la société i-BP assurées au sein de ses centres de données sont :

- L'hébergement des entrepôts de données des hyperviseurs de l'infrastructure virtuelle (85% des serveurs du parc sont des machines virtuelles)
- L'accueil de LUN pour des serveurs physiques (de type SQL Server, Oracle, etc.) en accès bloc
- Des services de fichiers Windows pour la bureautique et des applications
- Des applications d'archivage qui s'appuient sur du stockage WORM avec traçabilité
- Des baies de disques dédiées au stockage de la messagerie e-mail
- Des services de protection de données basés sur des prises d'image et de la réplication IP
- Des capacités de stockage pour le tampon de sauvegarde

#### **I.4.1.2 Répartition géographique du centre de données principal**

Les environnements de la société i-BP sont principalement hébergés dans le centre de données ALBIANT-IT, situé à Castres (81), sur deux sites distants d'environ deux kilomètres. Le premier de ces sites est appelé SAPHIR et le second TOPAZE.

Les deux sites accueillent des environnements de production, d'intégration et de développement. Ils sont tous deux « actifs », et la charge est distribuée équitablement entre eux.

#### **I.4.2 Renouvellement technologique des infrastructures de stockage**

Fin 2012, la société i-BP constatant un début de saturation de ses infrastructures de stockage de données NetApp, décision fut prise d'initier un ensemble de réflexions ayant pour objet de préciser les objectifs et enjeux d'un projet de renouvellement.

Ce travail permit notamment la définition d'un ensemble de besoins comparables entre les diverses entités du groupe BPCE en matière d'infrastructures informatiques, permettant ainsi d'envisager une convergence des investissements.

Début 2013, une consultation ayant pour objet le renouvellement de l'infrastructure de stockage de données SAN fut portée par i-BP et Natixis. Les exigences formulées lors de cet appel d'offres furent les suivantes :

- La solution retenue devait offrir une capacité de stockage utile permettant de couvrir les besoins de la production pour les 3 années à venir.
- La solution retenue devait disposer de fonctionnalités de type métro cluster permettant de créer un miroir des données avec bascule automatique entre les sites de SAPHIR et TOPAZE.
- Le coût devait être compétitif.
- En exigence optionnelle, la solution retenue devait fournir via ses outils d'administration, des possibilités d'industrialisation des opérations d'exploitation récurrentes.

A l'issue de cette consultation, c'est la solution proposée par le constructeur EMC<sup>2</sup> qui fût retenue (les autres concurrents à cet appel d'offres étaient alors HP, Hitachi, IBM et NetApp).

Fin 2013, les premières étapes de ce projet de renouvellement furent lancées.

#### **I.4.3 Architecture de la solution technique retenue**

L'architecture technique sélectionnée par la société i-BP à l'issue de la consultation évoquée plus haut reconduit l'installation des nouveaux équipements sur les sites de SAPHIR et TOPAZE.

Ce socle technique va permettre aux différentes entités applicatives hébergées par i-BP d'obtenir du stockage performant et sécurisé à l'aide de mécanismes de réplication de données.

Par ailleurs, des fonctionnalités de virtualisation de la donnée font partie intégrante de cette nouvelle architecture. Ainsi, il sera possible d'effectuer des migrations de données « à chaud » tout en conservant un haut niveau de disponibilité de la solution. Pour atteindre ces objectifs, les équipements EMC<sup>2</sup> suivants ont été retenus :

- Des métro clusters VPLEX, permettant de fournir les fonctionnalités de virtualisation
- Des baies de disques de type VMAX permettant d'assurer le stockage de masse des données
- Un réseau SAN dédié, dit « back-end », assurant l'interconnexion des équipements précédemment cités
- Un réseau SAN dédié, dit « front-end », assurant l'interconnexion des hyperviseurs, serveurs physiques et autres équipements utilisant le protocole Fibre Channel pour accéder aux données

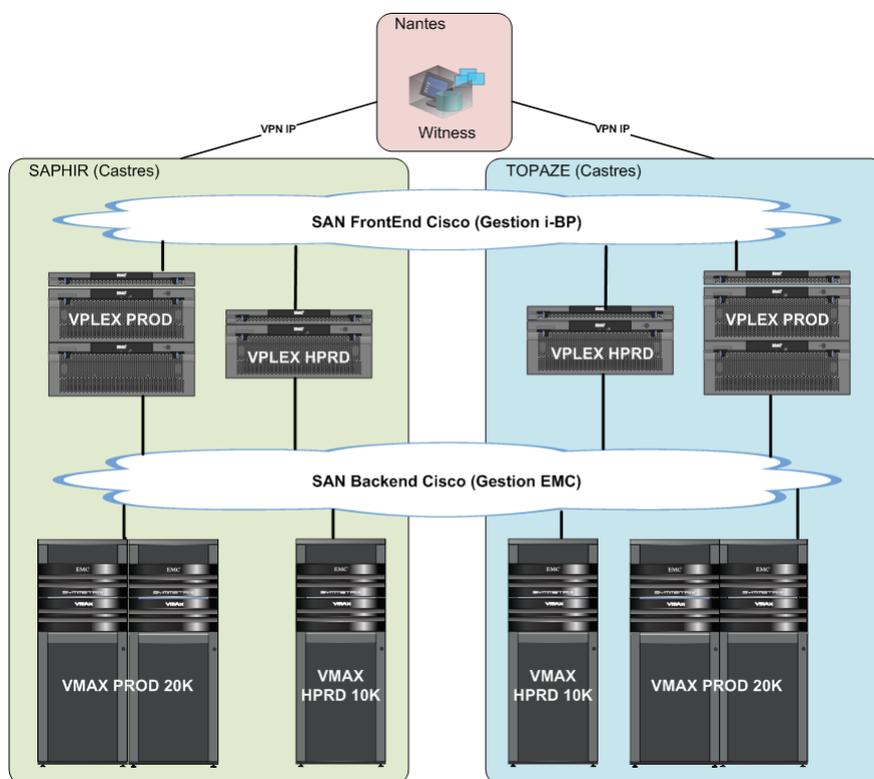


Figure 3 – Architecture générale de la solution EMC<sup>2</sup> (sites de SAPHIR et TOPAZE)

La totalité de la volumétrie sera virtualisée et présentée aux serveurs au travers des équipements EMC<sup>2</sup> VPLEX.

#### **I.4.4 Définition des standards et outils d'administration**

Le bouleversement technologique d'ampleur que constitue le renouvellement intégral de l'infrastructure de stockage SAN, qui plus est à l'aide de matériels provenant d'un constructeur différent, s'accompagne inmanquablement de la nécessité de refondre les standards de production et pratiques d'exploitation actuellement en vigueur.

Cette nécessité est sous-tendue par le besoin de tirer parti des nouvelles fonctionnalités de l'infrastructure cible ou la prise en compte de certaines spécificités propres au constructeur de ce matériel.

La société i-BP, consciente de cette problématique, souhaite également se doter des outils et méthodes qui permettront de réduire le coût d'exploitation de ses infrastructures SAN tout en augmentant le niveau de sécurité et de fiabilité.

La société i-BP souhaite donc tendre vers une gestion industrielle de ses moyens de stockage de données. C'est dans cette perspective que l'acquisition et la mise en œuvre de solutions de type « Software Defined Storage » ou SDS furent envisagées.

## **II Etat de l'art des solutions de « Software Defined Storage »**

### **II.1 Présentation du « Software Defined Storage »**

#### **II.1.1 Origine de l'appellation**

De nos jours le concept de virtualisation est à présent bien connu et largement répandu dans les centres de données du monde entier. Il suffit de constater le succès des solutions de machines virtuelles telles que VMWare vSphere ou Citrix XenServer pour s'en persuader.

Cependant, aussi évoluées qu'elles soient, la virtualisation apportée par ces solutions n'est que partielle et ne concerne qu'une partie des ressources que l'on trouve habituellement dans les centres de données. En effet, si l'utilisation des machines virtuelles et des hyperviseurs permettent la virtualisation de la puissance processeur et de la mémoire vive afin d'en optimiser l'utilisation, il n'en va pas de même du réseau et du stockage de données les deux autres grandes ressources que l'on retrouve au sein de ces fermes de serveurs.

Depuis 2013 de nouveaux concepts sont apparus et tentent de combler ce manque en promettant de virtualiser les réseaux grâce aux logiciels de type « Software Defined Network » (SDN) ainsi que le stockage de masse et les réseaux SAN en général à l'aide des « Software Defined Storage » (SDS). C'est à cette dernière déclinaison que nous allons nous intéresser plus avant dans ce chapitre.

Dans les deux cas, l'objectif est la virtualisation de ces ressources afin d'en rendre la configuration et la consommation plus efficace, mais aussi pour simplifier leur gestion à l'échelle du centre de données.

##### **II.1.1.1 Virtualisation du stockage, base du « Software Defined Storage »**

De nos jours, si la virtualisation est un phénomène bien connu et très répandu dans les entreprises, ce concept est presque systématiquement associé à la virtualisation des systèmes d'exploitation et des ressources associées soit la CPU et la mémoire vive principalement.

Qu'est-ce que la virtualisation ? Pour résumer la virtualisation en une phrase nous pourrions dire que : virtualiser c'est décorréliser les ressources physiques des ressources logiques. Concernant le stockage de données : virtualiser, c'est donner au système d'exploitation une vision logique de l'espace de stockage. Cette vision dissocie complètement la réalité physique des systèmes matériels sur lesquels sont effectivement stockées les données des aspects logiques vus par l'utilisateur.

La virtualisation du stockage de masse est au cœur de toute solution de type « Software Defined Storage » et bien que les implémentations réalisées sur la base de ce paradigme diffèrent les unes des autres, il est néanmoins possible de dégager les aspects communs suivants.

#### ***II.1.1.1.1 Les bénéfices de la virtualisation du stockage de données***

##### **II.1.1.1.1.1 Organisation des données**

Principale fonctionnalité d'un système de virtualisation du stockage : l'utilisation de ce type de technologie est avant tout la promesse d'une plus grande souplesse dans l'organisation et la distribution des espaces disques.

Grâce à sa position centrale au sein de l'infrastructure le dispositif de virtualisation, alimenté en volumétrie par les baies de disques du réseau SAN, rend possible la création de volumes de données composites en s'affranchissant des limites technologiques rencontrées jusqu'alors.

Avant l'apparition des mécanismes de virtualisation, la relation qui unissait un serveur et la baie de stockage hébergeant ses volumes de données n'était pas sans rencontrer des limitations. En effet, chaque baie de disques disposant de ses propres contraintes techniques, il était impossible d'étendre la capacité de stockage d'un serveur de manière illimitée. A l'ère du « Big Data » et de l'explosion des besoins en matière de stockage, ce type de réponse n'est plus envisageable.

L'utilisation des techniques de virtualisation permet de répondre à cette problématique en rendant possible la création d'espaces de stockage répartis sur de nombreuses baies de disques à travers le réseau SAN tout en masquant cette complexité aux serveurs les utilisant.

Les opérations de redimensionnement des volumes de données sont des actions qui peuvent également, selon les systèmes d'exploitation, poser problème. Cette problématique est résolue par l'usage des dispositifs de virtualisation qui fournissent les outils permettant de traiter ces opérations de manière transparente et homogène.

#### **II.1.1.1.2 Migration de données et modification de l'infrastructure de stockage**

Les infrastructures de stockage de données, comme toute autre infrastructure informatique, sont amenées à évoluer dans le temps. Elles changent d'ampleur, se transforment et sont soumises, bien souvent, à un cycle de vie correspondant à leur amortissement budgétaire et à leur obsolescence plus ou moins programmée.

En résulte l'apparition cyclique de projets de migration ayant pour objectif le renouvellement de tout ou partie de ces infrastructures. Ces opérations sont lourdes et potentiellement délicates. Elles peuvent perturber le fonctionnement des serveurs de l'entreprise dont, de plus en plus souvent, l'arrêt n'est pas envisageable car l'activité se poursuit 24h/24h.

Les dispositifs de virtualisation du stockage permettent de traiter ces aspects en proposant un ensemble de mécanismes (le plus souvent propriétaires) qui rendent possible la migration de données, l'ajout ou la suppression de baies de disques « à chaud » et donc sans interruption de service.

#### **II.1.1.1.3 Réplication de données**

Les mécanismes de réplication de données sont essentiels pour assurer les plans de reprise d'activité des entreprises en cas de désastre impactant un site de production. Ces mécanismes sont d'ailleurs largement présents au sein des baies de disques du marché.

Cependant, les processus de réplication de données utilisés au sein des dispositifs de virtualisation tentent d'aller plus loin en proposant la création de volumes de données actifs/actifs s'étendant sur plusieurs sites géographiques. Ainsi, la perte totale d'un site de production peut être intégralement compensée sans impact sur l'activité de production.

#### **II.1.1.1.4 Simplification du management de l'infrastructure**

Comme évoqué au chapitre I.2.4, l'administration d'un réseau SAN est une chose complexe qui fait intervenir un nombre très important d'outils et de mode d'administration divers.

L'utilisation des outils de virtualisation permet de réduire, généralement, cette complexité en proposant une interface d'administration unique et une vue de bout en bout de l'infrastructure de stockage. Cette interface devient, en quelque sorte, le point focal d'administration pour les équipes d'exploitation.

#### ***II.1.1.1.2 Les risques liés à la virtualisation du stockage de données***

##### **II.1.1.1.2.1 Complexification de l'infrastructure**

Cela peut sembler paradoxal, mais l'utilisation d'un système de virtualisation du stockage de masse induit inévitablement une part de complexité supplémentaire au sein de l'infrastructure SAN.

Cette complexité ne résulte pas de l'utilisation du dispositif en tant que tel puisqu'il est acquis qu'un système de virtualisation, grâce à l'agilité qu'il confère aux administrateurs, tend à faciliter la gestion d'un réseau SAN. Cette complexité est en réalité causée par la présence d'un équipement supplémentaire, le système de virtualisation donc, au sein du chemin de données.

En effet, comme toute infrastructure informatique, les réseaux SAN n'échappent pas aux demandes de mises en conformité pour des raisons liées aux activités de support. Cela suppose que chaque équipement présent au sein du réseau apparaisse au sein de la matrice de compatibilité correspondante, matrice qui s'étend aussi bien aux commutateurs SAN qu'aux baies de disques et autres serveurs présents sur le réseau de stockage.

Ainsi, les activités de maintien en conditions opérationnelles, comme les montées de versions notamment, sont fortement impactées par le rallongement du chemin de données. La mise en œuvre d'une matrice de compatibilité recouvrant la totalité du périmètre est donc plus complexe.

#### **II.1.1.1.2.2 Un retour arrière problématique**

Nous l'avons évoqué plus haut dans ce chapitre, les dispositifs de virtualisation du stockage de masse se caractérisent, notamment, par les services qu'ils apportent sur l'organisation et la construction des volumes de données.

Nombre de ces mécanismes rendent possible la création de pools de volumétries. Ces pools sont constitués par l'agrégation des volumétries issues des baies de disques présentes sur le réseau SAN. Ces volumétries sont ensuite redécoupées et recombinaées selon les besoins. Ces fonctionnalités sont particulièrement prisées car elles permettent d'optimiser au mieux l'espace disque en s'affranchissant des limitations techniques rencontrées sur une baie de disques unique.

Il est intéressant de noter que ces possibilités ne sont pas sans soulever quelques questions liées à la stratégie de l'entreprise. Ces mécanismes de redécoupage du stockage étant issus d'implémentations propriétaires, et par définition non compatibles entre eux, la question d'une éventuelle sortie du modèle de virtualisation que ce soit pour une solution concurrente ou pour revenir à une architecture non virtualisée est problématique.

Dans un cas comme dans l'autre la solution passera par un projet de migration global qui ne pourra s'appuyer les mécanismes de virtualisation actuellement en production, ce qui sera inévitablement complexe et coûteux en investissement humain.

#### **II.1.1.1.2.3 Augmentation des temps de latence**

Les performances sont un aspect essentiel d'un réseau SAN. De ses performances dépendra en très grande partie le bon fonctionnement des applications et des serveurs utilisant ses services.

Lorsque l'on parle de performance pour un réseau véhiculant les données issues du stockage de masse il est principalement question des temps de latence. Un temps de latence est la métrique désignant le temps qui s'écoule entre le début d'une requête d'entrée/sortie et le moment où cette dernière est satisfaite. L'unité utilisée est la milliseconde.

Les temps de latence sont facteur non seulement des performances intrinsèques des équipements constituant le réseau mais également de la profondeur du chemin de données,

soit le nombre d'équipements traversés pour atteindre l'emplacement où sont stockées les données.

Face à ce constat, une évidence s'impose : toute prolongation de ce chemin, et c'est le cas lorsque l'on ajoute un dispositif de virtualisation du stockage, entraînera mécaniquement une augmentation des temps de latence au sein du réseau.

Fort heureusement, cette réalité peut être nuancée grâce aux mécanismes de cache présents au sein des équipements de virtualisation et qui permettront de compenser, en partie, cette augmentation des temps de latence. Cependant, l'impact sur les performances de l'utilisation des techniques de virtualisation ne peut être ignoré et doit être pris en compte.

### **II.1.1.2 Le concept de « Software Defined Storage »**

Le rapport [1] publié par la SNIA (Storage Networking Industry Association) décrivant les fondements des solutions de type SDS confère à ces derniers les caractéristiques essentielles suivantes :

- Automatisation : simplification de l'administration de l'infrastructure par l'automatisation des tâches récurrentes
- Interfaces standard : API permettant l'administration, la création et la maintenance de ressources liées au stockage
- Virtualisation des chemins d'accès aux données : utilisation des techniques de virtualisation du stockage et support des modes objets, fichiers et blocs
- Evolutivité : Capacité à étendre l'infrastructure de stockage sans interruption de service et sans impacter les performances

Ces caractéristiques ne sont pas une nouveauté dans l'industrie du stockage de données et un observateur attentif ne manquera pas de relever que déjà en 2009 dans [2] il était question de tous ces aspects.

Aujourd'hui, les techniques de virtualisation évoquées au chapitre précédent sont très largement utilisées au sein des réseaux SAN. Elles permettent d'apporter un premier niveau d'abstraction au stockage de la donnée en masquant son implémentation physique et en conférant à l'ensemble une certaine agilité, notamment pour les problématiques liées aux migrations de données.

L'évolutivité des solutions de stockage est, elle aussi, une propriété largement répandue de nos jours. Qu'il s'agisse d'étendre le réseau SAN lui-même par l'adjonction de nouveaux directeurs ou des baies de stockage par l'ajout de disques, les mécanismes permettant d'étendre la capacité des solutions de stockage en environnement professionnel sont bel et bien devenus un standard indispensable à toute production informatique.

Alors si une majorité des caractéristiques que l'on retrouve dans les solutions de type SDS existent déjà sous une forme ou sous une autre au sein des réseaux SAN actuels, quelle plus-value ces solutions peuvent-elles apporter ?

### II.1.1.3 Spécificités de l'approche « Software Defined Storage »

Les solutions de type « Software Defined Storage », au-delà des aspects hardware évoqués plus haut, sont avant tout des constructions logicielles et plus précisément des suites logicielles de composants agissant à divers niveaux au sein de l'infrastructure de stockage.

En premier lieu, et comme l'illustre la figure ci-dessous, toute solution de type SDS à l'instar d'une solution orientée « Software Defined Networking » tente de dissocier ce que l'on nomme la couche de contrôle de la couche stockage à proprement parler.

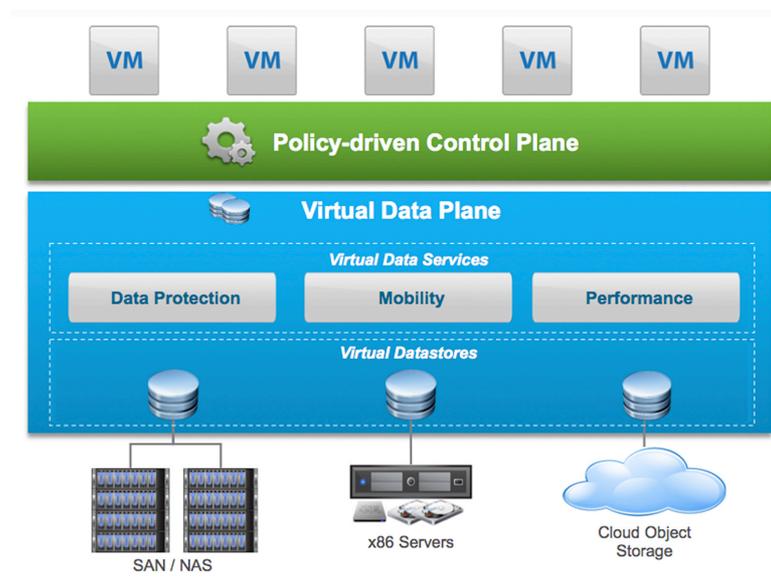


Figure 4 – Modèle en couche d'une solution de type SDS (source : [3])

La couche de stockage représente un premier niveau d'abstraction vis-à-vis des ressources physiques. C'est à ce niveau que les ressources de stockage sont administrées à proprement parler et où les mécanismes de virtualisation (évoqués au chapitre II.1.1.1) sont mis en œuvre, ce qui confère à la solution SDS sa souplesse.

La couche de contrôle, que l'on pourrait qualifier également de couche de pilotage, fournit les interfaces permettant l'administration globale de la solution. C'est ici que l'on définit la stratégie de l'entreprise en matière de stockage, que ce soit par la création d'exigences en matière de qualité de service ou de standards de production. L'automatisation du « provisioning », activité liée à la création et la livraison de volumes de données virtuels, doit également être possible ainsi que l'exploitation des services de protection de données (réplication, clone, migration, etc.).

Pour finir, nombre de solutions de type SDS tentent d'apporter, en plus des fonctionnalités évoquées ci-dessus, un ensemble de services avancés dans le but de conférer toujours plus d'agilité à ses utilisateurs. Ainsi, la quasi-totalité des SDS du marché fournit un ensemble d'API permettant l'interconnexion avec d'autres types de produits comme des solutions de Cloud Computing.

#### **II.1.1.4 Avantages des solutions de type SDS pour les entreprises**

##### ***II.1.1.4.1 Rationalisation des modes d'administration***

Pourquoi une entreprise devrait-elle opter pour une gestion de type « Software Defined » de ses ressources de stockage ? Comme évoqué dans [4] les centres de données d'aujourd'hui sont confrontés à un besoin toujours plus important en matière de réactivité. Les mises en production se succèdent, la faute à des contraintes de « time to market » toujours plus présentes.

Dans le même temps, la complexité des réseaux augmente elle également, ces derniers incorporant sans cesse de nouvelles fonctionnalités, de nouveaux concepts, ce qui complexifie leur administration et donc accroît le temps nécessaire à la livraison des espaces disques indispensables à l'activité de production.

L'approche SDS est la promesse, en s'appuyant sur les techniques de virtualisation évoquées plus haut, de réduire le temps nécessaire à l'administration des infrastructures par

la mise à disposition de mécanismes et d'interfaces unifiées capables de gérer la totalité des équipements présents au sein du réseau. Et ce quel que soit le fournisseur de ces équipements et quelles que soient les différences technologiques pouvant exister entre ces matériels. Ce qui constitue indéniablement un gain de temps très conséquent.

#### ***II.1.1.4.2 Automatisation et prise en compte de la logique métier***

Toutes les entreprises doivent stocker leurs données et toutes les entreprises ont besoin de sécuriser ces données, mais toutes ne le font pas la même façon.

Certaines entreprises utilisent plusieurs sites pour héberger leurs infrastructures et utilisent des moyens techniques pour répliquer les données entre ces sites. Certaines entreprises utilisent abondamment les machines virtuelles et les répartissent géographiquement ou fonctionnellement selon des règles qui leur sont propres.

Il en va de même pour les besoins en matière de stockage et de la façon dont ces ressources seront présentées et consommées. L'ensemble de ces règles d'administration peut être désigné sous l'appellation de « logique métier ».

La livraison de volumétries s'appuie sur un schéma, propre à chaque entreprise, constitué d'un ensemble d'étapes et de règles plus ou moins formelles ayant pour objet la standardisation et la bonne utilisation des ressources de stockage. Cela se traduit par une série d'actes d'administration à effectuer, en séquence, sur tous les équipements de la chaîne de données pour arriver au but escompté.

L'exécution manuelle de ces actes d'administration est une procédure chronophage et potentiellement risquée. Une solution de type SDS doit permettre, grâce aux services de sa couche de pilotage, d'automatiser toutes ces tâches et de masquer leur complexité sous-jacente. En résultent des opérations standardisées et une réduction drastique du coût d'exploitation de la solution de stockage.

#### ***II.1.1.4.3 Unification des modes d'accès « bloc » et « fichier »***

En matière de stockage de données, deux modes d'accès aux ressources sont très largement présents au sein des centres de données. Il s'agit du mode dit « bloc », qui correspond aux réseaux SAN, et du mode dit « fichier » qui correspond aux dispositifs de type NAS.

Depuis que ces technologies existent, leur exploitation a toujours fait l'objet d'équipements et de modes d'administration disjoints. L'une de promesse de l'approche SDS, évoquée dans [1], est la convergence de ces modes d'administration au travers d'une interface unifiée.

#### ***II.1.1.4.4 Amélioration de l'interconnexion***

Les solutions de stockage professionnelles ont toujours brillé par leurs carences en matière de vecteurs d'interconnexion et ont toujours eu l'image de réseaux fermés et propriétaires. C'est toujours en partie vrai, mais l'avènement de phénomènes comme le Cloud Computing tend à changer la donne : de plus en plus de constructeurs dotent leurs équipements d'API utilisables dans le cadre d'une administration quotidienne.

Malheureusement, ces API ne sont pas compatibles entre elles et n'utilisent, bien souvent, pas les mêmes modes de communication (REST, SMI/S, SOAP, etc.). Les logiciels de type SDS, en tant qu'éléments fédérateurs au sein du réseau de stockage, permettent de contourner cette problématique de compatibilité en présentant « au reste du monde » une API unique.

Cette API pourra par la suite être utilisée pour élever le niveau d'automatisation de l'infrastructure grâce au développement de scripts notamment. C'est également là l'opportunité de fédérer la couche de stockage de l'entreprise au sein de produits tiers, comme des solutions de Cloud Computing pour ne citer que cet exemple.

## **II.2 Les produits SDS du marché**

Nous l'avons vu, le concept de SDS est quelque chose de récent. Malgré cela, il existe d'ores et déjà un bon nombre de produits sur le marché qui revendique cette appellation. Cette section va présenter certains d'entre eux, du moins les plus pertinents dans le cas de la présente étude, afin d'en lister les principales fonctionnalités et le cas échéant disparités.

### **II.2.1 EMC<sup>2</sup> VIPR**

#### **II.2.1.1 Description**

Disponible en version 1.0 depuis le 27 septembre 2013, VIPR (que l'on prononce « viper ») est la solution de « Software Defined Storage » de la société EMC<sup>2</sup>, société comptant parmi les leaders mondiaux des produits de stockage à destination des moyennes et grandes entreprises.

A l'instar des autres solutions de SDS et conformément aux principes même de ce concept, VIPR tente de répondre au besoin de virtualisation des entreprises en scindant les aspects relatifs à l'administration du stockage de la partie configuration et gestion de la stratégie de stockage de l'entreprise.

#### **II.2.1.2 Architecture de la solution**

Contrairement à certains produits du marché, VIPR est une solution purement logicielle qui ne s'appuie sur aucun composant matériel pour assurer son fonctionnement.

L'intégralité des composants de VIPR est hébergée et fonctionne au sein de machines virtuelles dédiées à cet usage.

Comme l'illustre la figure ci-dessous, l'interconnexion avec le stockage piloté par VIPR, qu'il soit de marque EMC<sup>2</sup> ou de constructeurs tiers, est réalisée à l'aide des API proposées par ces équipements. Traditionnellement : REST, SMI/S ou SOAP.

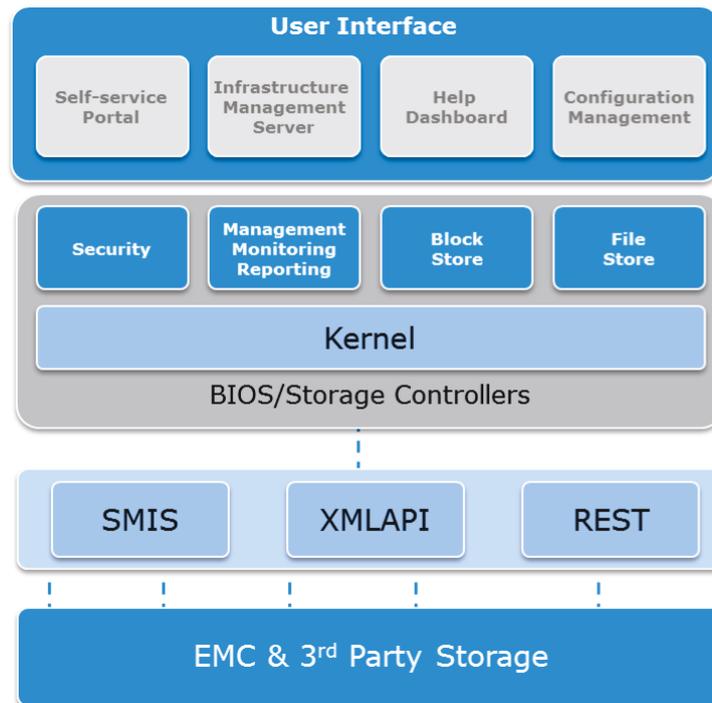


Figure 5 – Architecture générale de la solution EMC<sup>2</sup> VIPR

### II.2.1.3 Fonctionnalités

Au premier rang de ses fonctionnalités, VIPR propose de fédérer les espaces de stockage des équipements qui lui sont présentés afin de constituer un pool de ressources unique (ou de multiples pools selon le besoin). Il est alors possible, à l'aide des mécanismes d'automatisation fournis par ce produit de créer à la volée un ensemble de volumes de données répondants à de multiples critères, comme la taille des volumes bien évidemment, mais également la qualité de service auxquels ils doivent répondre.

Le produit VIPR promet de simplifier la gestion de l'infrastructure de stockage en fournissant une interface d'administration unifiée accessible selon plusieurs vecteurs : navigateur Web, ligne de commande ou API REST.

En parallèle des services liés à la gestion du stockage pur, VIPR dispose de fonctionnalités de passerelle permettant à des solutions tierces d'utiliser ses mécanismes à des fins d'intégration. Il est ainsi envisageable d'inclure les équipements gérés par VIPR au sein d'infrastructures de Cloud Computing.

#### **II.2.1.4 Orientation et évolution**

L'orientation de ce produit est très clairement centrée autour des aspects du stockage, « Software Defined Storage » oblige, sans tenter d'étendre son emprise sur le reste du centre de données. Le développement est actif et de nouvelles versions voient régulièrement le jour pour tenter d'ajouter de nouvelles fonctionnalités et de supporter de nouveaux équipements.

Tout récemment, EMC<sup>2</sup> a annoncé libérer une partie du code source de VIPR et la création d'un projet open source destiné à promouvoir l'approche SDS et, à n'en pas douter, augmenter la visibilité et la notoriété de ce type de produit [5].

Précision importante, dans la version 1.0, VIPR ne disposait pas des composants et connecteurs nécessaires au pilotage d'un équipement tel que le VPLEX, élément central de la nouvelle infrastructure de stockage acquise par la société i-BP.

### **II.2.2 OpenStack Cinder**

#### **II.2.2.1 Description**

Openstack est une solution open source de Cloud Computing de type IaaS (pour « Infrastructure As A Service ») à présent bien connue et disposant d'une forte communauté. Comme tout produit de Cloud Computing, Openstack est avant tout une pile logicielle comprenant de très nombreux services permettant de virtualiser tous les services que l'on trouve habituellement dans un centre de données.

Cinder, quant à lui, est le composant dévolu à la gestion des espaces disques et à la production des volumes de données en mode bloc au sein de l'infrastructure Openstack.

#### **II.2.2.2 Architecture de la solution**

Comme évoqué plus haut, Cinder n'est qu'un des services de la suite Openstack. Ce dernier a pour mission de servir de couche d'abstraction entre le service responsable de l'orchestration et de la gestion des machines virtuelles, le service « Nova », et les équipements en charge du stockage des données en mode bloc.

Le service Openstack Cinder incorpore un ensemble de pilotes lui permettant d'interagir avec les différentes baies de disques du marché. La figure ci-dessous illustre ce mode de

fonctionnement ainsi que les échanges prenant place entre les différents services et équipements.

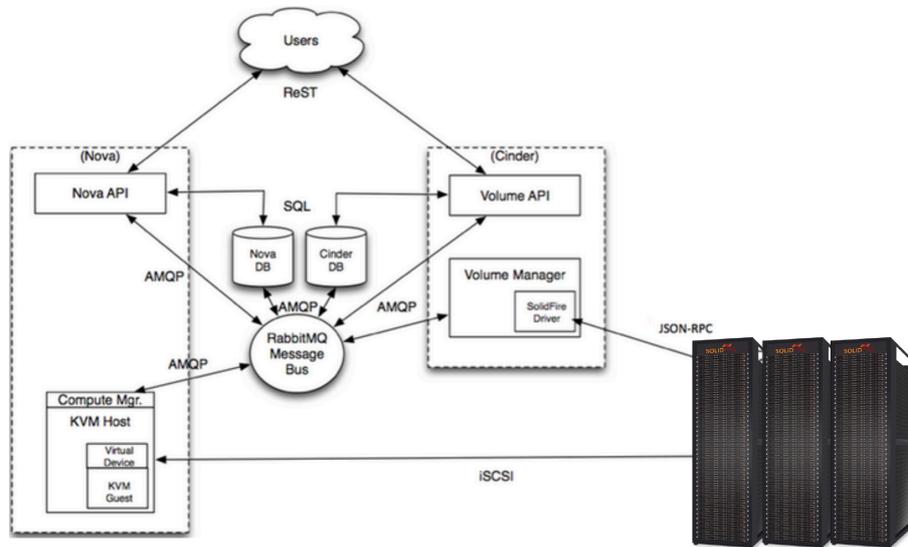


Figure 6 – Architecture du service Openstack Cinder

### II.2.2.3 Fonctionnalités

Openstack Cinder est-il un produit de type « Software Defined Storage » ? La question peut se poser au regard des fonctionnalités proposées et se pose d'ailleurs au sein même de la communauté Openstack comme le démontre [6]. C'est pourtant bien ainsi qu'il est présenté par la fondation Openstack [7].

Certes, la finalité de ce système reste le pilotage d'une infrastructure de stockage afin d'en faciliter l'exploitation tout en fournissant un ensemble complet d'API permettant l'interconnexion avec d'autres systèmes. Il s'agit là en effet de propriétés que l'on retrouve au sein des produits « Software Defined Storage ».

Cependant, en accord avec [8], l'auteur de la présente étude pense que cela ne suffit pas pour faire de ce produit une solution de « Software Defined Storage » dans l'acceptation que l'on se fait actuellement de ce type de produit.

Comme indiqué plus haut dans ce document, les solutions de « Software Defined Storage », pour obtenir ce titre, doivent clairement dissocier les aspects liés à la gestion du

stockage de la définition des processus et des classes de services devant régir l'infrastructure.

Ce second point ne fait pour l'heure l'objet d'aucune implémentation au sein d'Openstack Cinder.

#### **II.2.2.4 Orientation et évolution**

Openstack, compte tenu de sa communauté très active et de l'engouement actuel pour les solutions de Cloud Computing, est très clairement amené à poursuivre son évolution. Néanmoins, il restera une solution viable uniquement dans le cadre d'une adoption massive du modèle Cloud Computing là où les produits de type SDS laissent plus de libertés.

Point important, malgré le support de très nombreux équipements sous la forme de pilotes Cinder, le support natif des équipements de type EMC<sup>2</sup> VPLEX, équipements au cœur de la stratégie de stockage de la société i-BP, n'est pour l'heure pas assuré.

### **II.2.3 VMWare Virtual SAN**

#### **II.2.3.1 Description**

Solution de « Software Defined Storage » développée par la société VMWare, connue notamment pour ses produits de virtualisation et actuellement en position de leader sur ce marché, VSAN (pour « Virtual SAN ») est disponible depuis le mois d'août 2013.

Ce produit, bien qu'appartenant à la famille des logiciels de type SDS, aborde le problème de l'automatisation et de la gestion des ressources de stockage selon une approche différente de ses concurrents en mettant les hyperviseurs (les logiciels permettant l'hébergement des machines virtuelles) au cœur du dispositif.

#### **II.2.3.2 Architecture de la solution**

L'approche utilisée par « Virtual SAN » pourrait être résumée ainsi : supplanter les réseaux de stockage SAN traditionnels en mettant à profit les espaces de stockage interne des hyperviseurs d'ores et présents chez le client, les agréger et ainsi constituer une entité logique, sorte de réseau SAN virtuel.

Contrairement à d'autres produits du marché et notamment ceux abordés jusqu'à présent dans cette étude, VSAN n'utilise pas de pile logicielle dédiée pour assurer son fonctionnement. L'intégralité des composants permettant de créer et de gérer un SAN virtuel est intégrée dans le noyau de l'hyperviseur VMWare ESXi.

Le produit VMWare VSAN s'appuie sur une architecture de type « scale out » ce qui signifie que l'extensibilité de la solution est assurée par l'ajout de nouveaux équipements au sein du cluster existant. Ainsi, étendre la capacité d'un SAN virtuel revient à ajouter de nouveaux hyperviseurs au SAN actuel, comme l'illustre la figure ci-dessous.

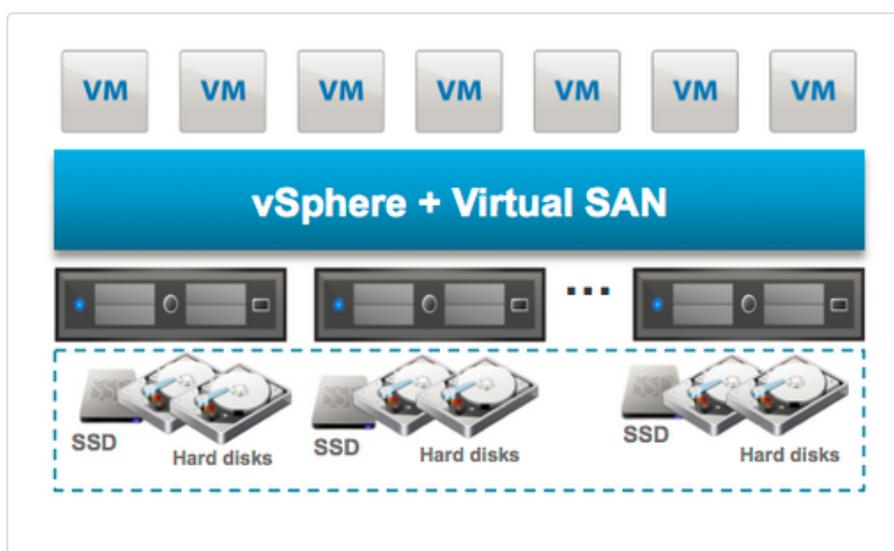


Figure 7 – Illustration de l'approche "scale out" du produit VMWare VSAN (source [6])

### II.2.3.3 Fonctionnalités

Nous l'avons vu, le produit VSAN s'adresse en priorité à des clients possédant des environnements massivement virtualisés et souhaitant s'appuyer sur cette infrastructure pour y bâtir un réseau SAN que l'on pourrait qualifier d'alternatif.

Les machines virtuelles d'une infrastructure VMWare sont stockées au sein d'entités logicielles que l'on nomme entrepôts de données. Ces entrepôts prennent la forme d'espace de stockage provenant traditionnellement d'un réseau SAN.

Dans une solution de type VSAN, le réseau SAN en question est constitué de la somme des espaces disques locaux des hyperviseurs (les hôtes hébergeant les machines virtuelles) du

cluster. Une telle solution permet de créer des entrepôts de données distribués dont les données sont de fait réparties entre les espaces de stockage de chaque hyperviseur.

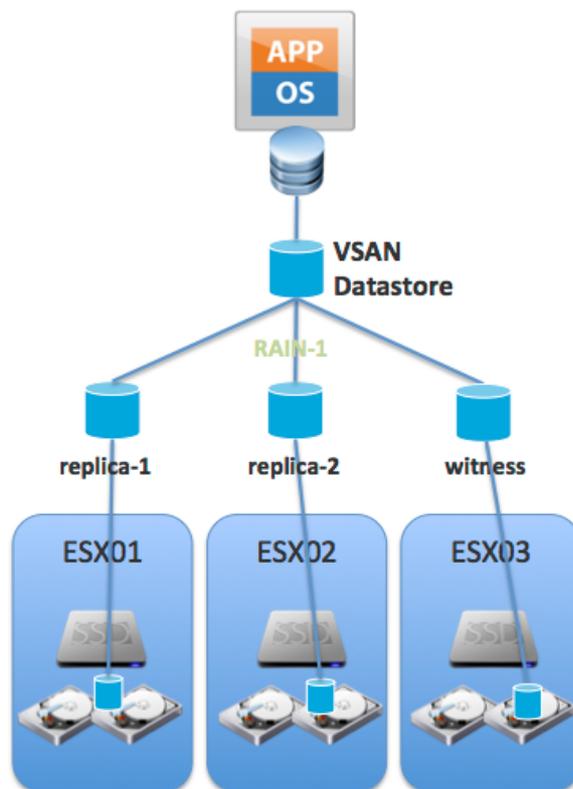


Figure 8 – Composition d'un entrepôt de données sous VMWare Virtual SAN

La figure ci-dessus illustre le principe d'entrepôt de données virtuel réparti sur trois machines physiques exécutant le logiciel hyperviseur VMWare ESXi. Ces serveurs sont équipés de disques durs internes mis à contribution pour le stockage de cet entrepôt de données.

On notera que les services de protection habituellement embarqués au sein de baies de disques, comme notamment le système RAID, sont remplacés ici par une implémentation similaire appelée RAIN (Redundant Array of Independent Nodes) directement située au sein de l'hyperviseur.

Comme tout système de type « Software Defined Storage », VSAN permet de mettre en œuvre un ensemble de politiques dites fines agissant au niveau machine virtuelle. Ces politiques décrivent le fonctionnement du centre de données virtualisé, notamment sous forme de classes de services. Ces possibilités garantissent des niveaux de qualité de service selon des critères de capacité, de disponibilité ou bien encore de performance.

Il est ainsi envisageable, à titre d'exemple, de créer une politique ne s'appliquant qu'aux machines virtuelles de production nécessitant des temps de latence disque très faibles afin que les données issues de ces dernières soient automatiquement déplacées sur les disques durs appropriés. Par exemple, des disques de type Flash/SDD.

Inversement, des machines utilisées dans le cadre d'une maquette et donc de moindre importance verraient leurs données placées automatiquement sur des disques durs considérés comme lents : des disques durs à interface SATA.

#### **II.2.3.4 Orientation et évolution**

L'intérêt de la solution VMWare VSAN est clairement de concurrencer le modèle que l'on pourrait qualifier de « standard » : celui des grandes entreprises ayant recours à des réseaux SAN et s'appuyant sur des baies de disques propriétaires et coûteuses. A l'opposé, VSAN propose de mettre à profit du matériel banalisé et peu onéreux. En cela, ce type de produit s'inscrit très clairement dans la lignée des solutions dites de stockage convergé ou ultra convergé.

A noter que cet outil, pour intéressant qu'il soit, s'adresse plus particulièrement à des clients qui n'utilisent pas d'équipements à haute densité de type serveurs lames. En effet, si le but de VSAN est d'agréger l'espace disque local de tous les hyperviseurs du réseau, force est de constater que les serveurs lames ne représentent pas les meilleurs candidats : le nombre d'emplacements disque disponibles sur ce type de châssis étant bien trop limité pour constituer une alternative intéressante.

Bien que le développement de cet outil soit très actif et qu'il bénéficie de surcroît de l'appui d'une société de premier plan tel que VMWare, il n'est pas exempt de défauts, potentiellement rédhibitoires. Le plus important étant que VSAN n'est pas encore en mesure de supporter la mise en œuvre de clusters étendus sur plusieurs sites géographiques. Or la société i-BP dispose de tels clusters au sein de son infrastructure.

## **II.3 Critique du modèle « Software Defined Storage »**

Bien que le concept de SDS semble être à première vue une solution idéale pour les réseaux d'entreprises dont la gestion du stockage de données est essentielle, il n'en reste pas moins qu'il s'agit là d'un phénomène en grande partie marketing et qui souffre de nombreux manques.

### **II.3.1.1 Une maturité en devenir**

S'il y a bien une ressource stratégique au sein des fermes de serveurs, c'est bien les données sur lesquelles reposent tous nos systèmes d'information modernes et, par extension, les moyens mis en œuvre pour les conserver. En résulte une crainte bien compréhensible lorsqu'une technologie émerge et tend à redéfinir des concepts et pratiques jusque-là bien établis. Après tout, la production informatique c'est avant tout une recherche de stabilité plus que de changement.

La compétition que se livrent les divers éditeurs du marché sur un segment aussi stratégique est féroce. Les effets d'annonce se succèdent et la pression du « time to market », qui touche tous les acteurs de l'industrie, n'est pas sans conséquence sur la qualité des produits livrés. D'autant plus lorsqu'il s'agit d'une nouvelle gamme de solutions.

C'est ce que l'on peut craindre concernant l'avènement des produits de type « Software Defined Storage » : pour occuper au plus vite ce segment de marché, les premières moutures de ce type de solution souffrent de sévères limitations techniques et, à n'en pas douter, de bogues plus ou moins importants.

### **II.3.1.2 Coûts prohibitifs**

Les technologies de type « Software Defined Storage » sont relativement récentes. Le modèle économique pour cette catégorie de produits n'est pas encore clairement défini. C'est pour cette raison que l'on constate un flottement dans leur tarification ainsi qu'un effet « nouveauté = prix élevé » assez dommageable.

Prenons l'exemple du produit VMWare VSAN évoqué plus haut. Ce dernier est facturé 2495\$ par processeur. Un serveur type utilisé dans le cadre d'une infrastructure virtualisée compte au moins deux sockets physiques soit un coût de 4990\$ pour ce serveur. Dans

l'hypothèse d'un total de 100 serveurs présents au sein du centre de données, ce qui n'a rien d'excessif, le coût d'une solution VSAN s'élève donc à 499000\$. Cela n'a rien d'anecdotique.

### **II.3.1.3 Un concept encore flou**

Enfin, il n'est pas déraisonnable de s'interroger sur le concept même de « Software Defined Storage » et sur sa pertinence. Même si ce dernier semble apporter de nombreuses réponses aux problèmes opérationnels rencontrés lors de l'administration d'infrastructures de stockage de données, comme notamment la réduction de la complexité des opérations d'administration, il n'en reste pas moins que la prudence reste de mise.

Il s'agit d'éviter les effets de mode pour s'inscrire dans une démarche de veille technologique : suivre l'évolution de ces concepts prometteurs pour constater s'ils finiront, le cas échéant, par s'encrer au sein des centres de données.

## **II.4 Conclusion**

A l'issue de cette étude un constat s'impose : les outils de type « Software Defined Storage » actuellement présents sur le marché ne disposent pas encore de la maturité requise et ne permettent pas d'atteindre le niveau d'exigence souhaité par la société i-BP.

Face à ce constat, décision fut prise de saisir l'opportunité que représente ce projet pour étudier la viabilité d'une solution alternative réalisée en interne, certes, moins ambitieuse mais correspondant mieux aux méthodes et standards actuellement en vigueur au sein de la production de la Banque Populaire.

## **III Analyse et conception d'une solution d'automatisation**

### **III.1 Analyse du besoin**

#### **III.1.1 L'existant et les défauts de l'existant**

##### **III.1.1.1 L'existant**

L'infrastructure de stockage de données en mode bloc de la société i-BP est constituée, du moins avant l'introduction du matériel EMC<sup>2</sup> évoqué plus haut dans ce document, exclusivement de contrôleurs de marque NetApp.

La quasi-totalité de cette infrastructure est opérée manuellement à l'aide des outils fournis par le constructeur. Ce qui signifie que les nombreuses étapes nécessaires à la création, la configuration et la présentation d'un volume de données à un serveur résultent d'opérations faisant l'objet d'un traitement manuel. Il n'existe ni automatisation ni ordonnancement de ces opérations.

Seuls existent quelques scripts élémentaires développés à l'aide des langages et interpréteurs VBS/BASH. Ils assurent une supervision sommaire des équipements concernés pour mieux alerter les administrateurs en cas de défaillances techniques ou de comportements anormaux.

Les référentiels issus de la gestion de cette infrastructure (fichiers d'inventaire, mains courantes, etc.) font l'objet, là encore, d'un traitement manuel de la part des administrateurs.

Les demandes de mises à disposition de volumétries sont adressées à l'équipe VSS par le biais d'un site web dédié à cet usage, le « workflow GRIS » (pour Gestion des Ressources Internes Stockage). Le temps de traitement toléré pour une telle demande est de trois jours ouvrés à compter du jour de réception.

##### **III.1.1.2 Les défauts de l'existant**

L'administration des infrastructures de stockage à l'aide des outils standards fournis par les constructeurs (EMC<sup>2</sup>, Cisco, etc.) ne représente pas une solution acceptable sur le long terme pour les raisons suivantes :

- L'absence de console d'administration unifiée oblige les opérateurs à « jongler » entre les différentes interfaces des constructeurs. En résulte un temps de traitement total conséquent et qui ne permet pas de répondre rapidement aux sollicitations des autres équipes.
- Le traitement manuel des opérations en plus d'être lent est inévitablement une pratique risquée : elle rend possible une erreur de manipulation avec toutes les conséquences que cela peut avoir.
- Les aspects métiers i-BP ne peuvent pas faire l'objet d'une prise en compte par ces outils. Nomenclatures, modes de présentations et d'utilisations : tout doit faire l'objet d'un traitement manuel sans possibilités d'automatisation.

Les scripts développés par i-BP actuellement en production sur l'infrastructure de stockage NetApp, bien que peu nombreux et n'ayant pas vocation à assurer l'administration complète de cette infrastructure, ne sont pas sans soulever certaines problématiques :

- Les algorithmes mis en œuvre dans ces scripts sont basiques. Il s'agit plus d'enchaînements de commandes de base que d'algorithmes à proprement parler. En résulte une incapacité à détecter les anomalies en cours de traitement et à y apporter un traitement adéquat.
- Les langages et interpréteurs utilisés (VBS et BASH) ne permettent pas d'envisager de portabilité d'un système d'exploitation à un autre et limitent donc leur utilisation à une plate-forme donnée. Par ailleurs, ces outils sont obsolètes au regard des alternatives existantes.
- Les scripts existants ayant été développés au coup par coup, selon les besoins, ces derniers n'utilisent ni les mêmes approches ni les mêmes concepts et ne s'appuient sur aucune base commune. Leur maintenance s'avère donc très complexe.
- Corollaire du point précédent, l'absence de documentation constatée complexifie non seulement les activités de maintenance du code, mais nuit également aux transferts de compétences.
- Enfin, le périmètre couvert par ces outils est exclusivement orienté sur des technologies NetApp sans adaptation possible à d'autres matériels.

### **III.1.2 Cahier des charges**

La société i-BP souhaite donc remédier aux manques et problèmes évoqués ci-dessus par la création d'une solution développée en interne et dont les fonctionnalités attendues sont les suivantes :

#### **III.1.2.1 Fonctionnalités attendues**

Afin de réduire le temps consacré à l'administration de ses infrastructures de stockage de données, d'augmenter le niveau de sécurité et de gagner en réactivité, la société i-BP souhaite se doter d'un ensemble d'outils rendant possible l'automatisation des tâches les plus couramment effectuées par ses équipes d'ingénieurs stockage.

L'objectif n'est pas de couvrir l'intégralité des actes de configuration et d'administration possibles. Seules les opérations liées à la création, la configuration et la livraison de nouvelles volumétries seront prises en charge par la solution. Ce type d'actions représente environ 80% des demandes traitées par l'équipe VSS-Stockage.

Les opérations liées aux travaux de dé-commissionnement de serveurs et à la suppression des ressources de stockage associées sont suffisamment rares pour être considérées comme marginales. Elles ne feront pas l'objet d'un développement spécifique. La société i-BP se laisse néanmoins la possibilité d'étudier la possibilité d'étendre le périmètre de ces outils dans les mois et années à venir.

Dans un souci de cohérence et de fiabilité, seules les opérations conformes aux standards et bonnes pratiques actuellement en vigueur au sein de la société i-BP pourront faire l'objet d'un automate dédié. Il est ainsi acquis que les demandes sortant de ce cadre feront l'objet d'une demande de dérogation et, sur acceptation de celle-ci, d'un traitement manuel de la part des intervenants.

Bien que le but principal de ce projet soit d'automatiser les opérations liées à l'administration de son infrastructure de stockage, la société i-BP souhaite que les outils à venir puissent laisser à l'appréciation des opérateurs certains détails. C'est le cas notamment des problématiques de répartition de charges spécifiques à ce type d'équipement.

Le respect des nomenclatures i-BP devra faire l'objet d'une observance stricte. Tous les éléments de configuration créés ou altérés par les outils et les automates (volumes VPLEX, zones Cisco, etc.) devront se conformer aux règles préalablement édictées.

La traçabilité des opérations étant un aspect particulièrement important, un système de journalisation complet devra être mis à disposition. Il permettra de conserver les éléments suivants sans limites de temps :

- Nature de l'opération effectuée
- Horodatage
- Identité de la personne ayant lancé l'opération
- Equipements concernés
- Paramètres passés à l'automate (numéro de série, baie de disques, etc.)
- Etat du traitement (réussite, échec, anomalie, etc.)
- Synthèse des opérations réellement effectuées sur l'infrastructure

La prise en charge par les automates des différents équipements de l'infrastructure de stockage devra se faire au travers de fichiers de configuration dédiés à cet usage. Ces derniers devront rendre possible l'ajout de nouveaux équipements de manière transparente et dynamique.

Dans un souci de maintenabilité de la solution, les composants logiciels développés dans le cadre de ce projet devront être conçus dans un souci de réutilisabilité maximum.

### **III.1.2.2 Contraintes et exigences**

Le développement de cette solution d'automatisation devra être réalisé dans le cadre de l'enveloppe budgétaire allouée au projet de renouvellement des infrastructures de stockage i-BP.

Les livrables produits à l'issue de ce projet devront faire l'objet d'une documentation complète (modélisation, code source, procédures, dossier d'exploitation, etc.) au format actuellement en vigueur. Une recette de ces documents devra être effectuée par l'équipe projet.

Afin de faciliter autant que possible la manipulation et l'exploitabilité des automates à venir, la société i-BP souhaite que ces derniers prennent la forme de scripts pleinement utilisables en ligne de commande.

Ce mode de fonctionnement, bien qu'il puisse paraître simpliste au regard des outils en mode graphique fournis par les constructeurs des équipements concernés, apporte un grand nombre d'avantages qui se révèlent déterminants pour une utilisation en environnement de production :

En premier lieu, l'utilisation de scripts pour réaliser les opérations d'administration d'une infrastructure de stockage se révèle à l'usage être un gain de temps très important. L'usage des consoles en mode graphique impose le recours à la souris et de naviguer dans un nombre conséquent de menus pour réaliser une à une les opérations attendues. Un script permet en une simple ligne de commande de remplacer cette navigation décousue.

Plus important encore, le développement d'un script rend possible l'inclusion des spécificités métiers de l'entreprise au sein de son algorithme. Ce qui permet de réaliser une chaîne d'opérations potentiellement complexes, spécifiques au fonctionnement de la production i-BP, à l'aide d'un seul automate.

L'utilisation de scripts en ligne de commande ouvre également la possibilité d'exploiter ces derniers dans le cadre d'un ordonnancement. Ce qui n'est traditionnellement pas possible avec les outils standards type GUI.

Cette approche permet également de rationaliser les modes d'administration disponibles au sein de l'infrastructure de stockage en concentrant ces derniers sur un serveur unique. Cela permet de gagner en rapidité : il est inutile de passer d'un serveur d'administration à un autre. Les ouvertures de flux et les habilitations nécessaires aux divers intervenants en sont simplifiées.

Enfin, et il s'agit là d'un point non moins important : l'utilisation de scripts permet de prendre en charge les aspects liés à la nomenclature des objets présents au sein du réseau de stockage. Chaque production disposant de ses propres normes et conventions pour ce qui est des noms donnés aux divers éléments de configuration de son réseau.

Le respect de ces normes est un point capital pour assurer une bonne exploitation de l'infrastructure dans le temps, et une chose est certaine : les êtres humains ne respectent pas toujours ces normes. Les automates déterministes et les scripts, eux, ne se trompent jamais.

## **III.2 Planification du projet**

Afin que le projet puisse se dérouler dans des bonnes conditions et, in fine, produise les livrables attendus par la société i-BP, il convient de lui conférer un cadre. Cette section va s'attacher à décrire ce cadre.

### **III.2.1 Lotissement du projet**

La nouvelle infrastructure de stockage SAN acquise par la société i-BP est une architecture composée de baies de disques et de boîtiers de virtualisation de type et de spécificités différents. Dans un souci de cohérence et d'organisation, les livrables relatifs à ces équipements feront l'objet d'un lotissement spécifique.

Chaque lot vise à produire une interface d'administration complète pour l'équipement concerné. Cette interface fonctionne de manière autonome et indépendante.

#### **III.2.1.1 Lot A – Modules d'administration VPLEX**

Le premier lot doit produire un ensemble d'outils permettant l'administration complète et automatisée de la plate-forme EMC<sup>2</sup> VPLEX en tenant compte de contraintes et des exigences d'exploitation exprimées par la société i-BP.

#### **III.2.1.2 Lot B – Modules d'administration VMAX**

Le deuxième lot, quant à lui, doit produire un ensemble d'outils rendant possible l'administration complète et automatisée des baies de disques EMC<sup>2</sup> VMAX. Là encore, un souci tout particulier sera porté aux contraintes et aux standards d'exploitation d'i-BP.

#### **III.2.1.3 Lot C – Modules d'administration CISCO**

Le troisième et dernier lot vise la production d'automates et d'outils permettant d'administrer une infrastructure SAN composée de directeurs Cisco MDS 9000.

### **III.2.2 Sélection du cycle de développement**

En accord avec l'équipe projet i-BP, le cycle de développement utilisé sera une approche de type « cycle en V ».

Le cycle en V se compose de sept étapes que l'on peut regrouper en trois grandes catégories que sont la phase de spécification, la phase de réalisation et la phase de validation.

La phase de spécification permet à la maîtrise d'ouvrage de formuler précisément ses besoins et attentes d'un point de vue fonctionnel. C'est notamment au cours de cette phase que le cahier des charges est rédigé puis validé par le client.

La deuxième phase, dite, de réalisation, regroupe les étapes de modélisation logicielle ainsi que le développement (ou codage) à proprement parler.

Enfin, la phase de validation permet, comme son nom le laisse supposer, la validation du développement préalablement accompli à l'aide notamment de tests unitaires et d'intégration. La conduite de tests de recette complets permet au client de valider tout ou partie des livrables attendus.

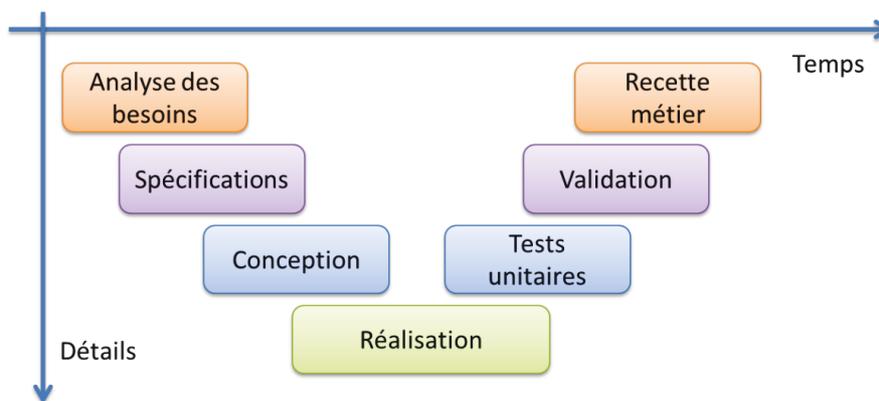


Figure 9 – Schéma illustrant le cycle de développement « en V »

### III.2.3 Planification

Afin de pouvoir fournir les livrables dans les temps il est nécessaire de dresser un inventaire des tâches du projet, de les ordonnancer, puis de définir un planning.

### III.2.3.1 Ordonnancement des tâches du projet

	Nom de la tâche	Durée
1	Etude préliminaire	15 jours
2	Etude de faisabilité	15 jours
3	Phase de spécification du lot A - VPLEX	10 jours
4	Conception et réalisation du lot A - VPLEX	35 jours
5	Tests unitaires et validation du lot A - VPLEX	5 jours
6	Phase d'intégration du lot A - VPLEX	3 jours
7	Mise en production du lot A - VPLEX	2 jours
8	Phase de spécification du lot B - VMAX	8 jours
9	Conception et réalisation du lot B - VMAX	30 jours
10	Tests unitaires et validation du lot B - VMAX	5 jours
11	Phase d'intégration du lot B - VMAX	2 jours
12	Mise en production du lot B - VMAX	2 jours
13	Phase de spécification du lot C - CISCO	10 jours
14	Conception et réalisation du lot C - CISCO	35 jours
15	Tests unitaires et validation du lot C - CISCO	5 jours
16	Phase d'intégration du lot C - CISCO	3 jours
17	Mise en production du lot C - CISCO	2 jours
18	Rédaction et livraison des livrables du projet	15 jours
19	Organisation des formations	10 jours

Figure 10 – Liste détaillée des tâches du projet et des charges associées

### III.2.3.2 Diagramme de GANTT

Le diagramme de GANTT ci-dessous nous permet de visualiser dans le temps les différentes tâches composant le projet ainsi les dépendances liant ces tâches entre elles.

La figure ci-dessous présente le diagramme de GANTT du projet :

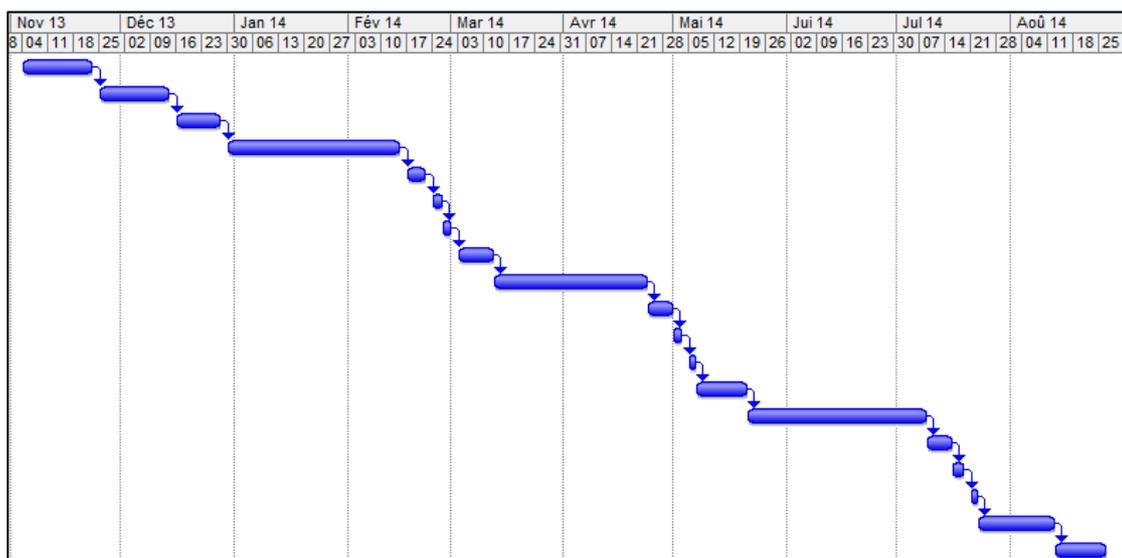


Figure 11 – Diagramme de GANTT du projet

### III.3 Spécifications techniques du projet

#### III.3.1 Spécifications techniques VPLEX

##### III.3.1.1 Présentation du dispositif

L'équipement appelé VPLEX est responsable de la virtualisation du stockage des données au sein de la nouvelle architecture de stockage i-BP. Ce dernier s'appuie sur les volumétries issues des baies de disques VMAX pour réaliser son travail de virtualisation.

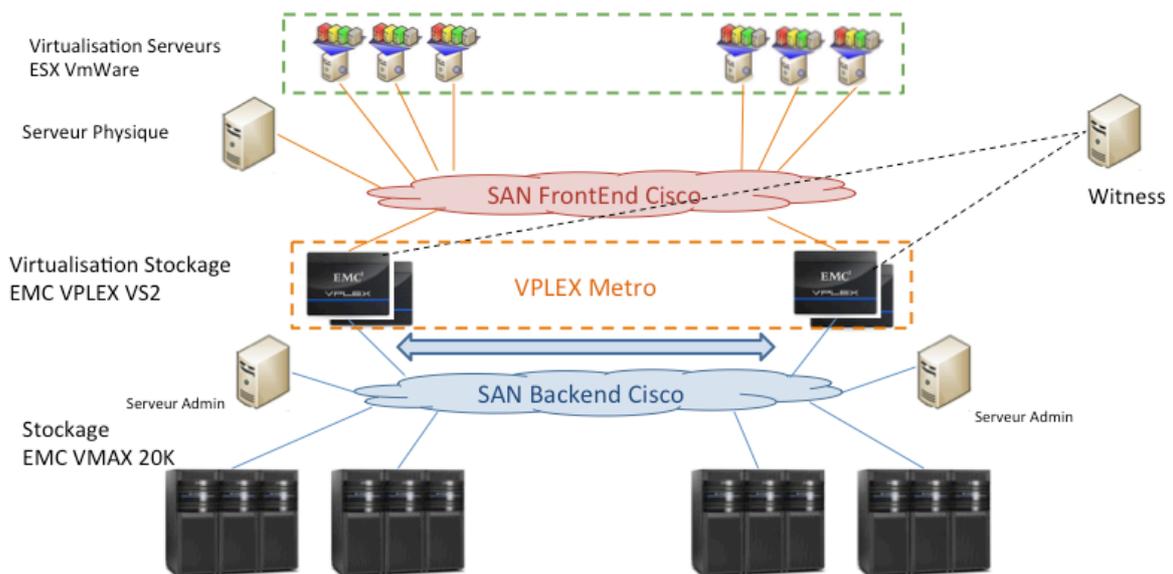


Figure 12 – Architecture globale de la solution EMC<sup>2</sup> VPLEX/VMAX

Pour chaque environnement (hors production et production) un VPLEX est positionné sur le site de SAPHIR et un second sur le site de TOPAZE. Ces équipements apparaissent dans l'encadré orange sur la figure 12 ci-dessus. Ces deux VPLEX sont liés entre eux et forme une entité logique appelée métro cluster.

Un VPLEX se compose d'un, deux ou quatre moteurs. Le moteur est responsable de la fédération des flux de données d'E/S (voir figure 13). Il se connecte aux hôtes et au stockage via des connexions Fibre Channel pour le transport des données.

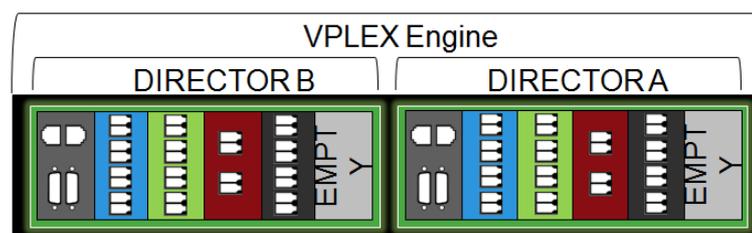


Figure 13 – Vue schématique d'un moteur (ou engine) VPLEX

Un VPLEX équipé d'un seul moteur est doté des composants principaux suivants :

- deux directeurs, désignés A et B, qui exécutent le logiciel « GeoSynchrony » et qui se connectent au stockage, aux hôtes et aux autres directeurs du cluster via des connexions Fibre Channel et Gigabit Ethernet
- une alimentation de secours, pour alimenter le moteur pendant les coupures d'alimentation temporaires
- deux modules d'administration munis d'interfaces pour gérer à distance le moteur VPLEX

Chaque VPLEX se compose également, quel que soit son nombre de moteurs :

- d'un serveur d'administration, qui gère le VPLEX et fournit une interface depuis une station de gestion à distance
- d'une armoire EMC<sup>2</sup> de 40U standard permettant de loger l'équipement du cluster dans son intégralité

En outre, les VPLEX constitués de plusieurs moteurs contiennent également :

- deux commutateurs Fibre Channel servant à la communication inter directeur entre différents moteurs
- deux onduleurs fournissant une alimentation de secours aux commutateurs Fibre Channel et permettant au système de fonctionner en cas de coupure d'alimentation temporaire

La prévention des situations de type « split-brain » et le bon fonctionnement du métro cluster VPLEX sont assurés par un composant logiciel autonome appelé « Witness » embarqué au sein d'une machine virtuelle située sur le site i-BP de Nantes.

#### ***III.3.1.1 Modules d'entrée/sortie Front-End***

La connectivité des serveurs vers les clusters VPLEX est assurée par les ports Front-End des directeurs contenus dans les moteurs. Les ports Front-End sont consacrés à l'activité E/S serveurs. Chaque directeur possède un module Front-End en slot A0 pour le directeur A et B0 pour le directeur B (cf. figure 13 en bleu).

### ***III.3.1.1.2 Modules d'entrée/sortie Back-End***

Les modules E/S Back-End sont présents dans chaque directeur, soit deux par moteur. Ils sont chargés de la connectivité du VPLEX vers les ports stockage disques, dans le cas présent vers les baies VMAX. Chaque module E/S Back-End possède 4 ports Fibre Channel en slot A1 pour le directeur A et B1 pour le directeur B (cf. figure 13 en vert).

### ***III.3.1.1.3 Modules d'entrée/sortie WAN COM***

Les modules E/S WAN COM sont présents dans chaque directeur, soit deux par moteur. Ils sont chargés de la connectivité inter cluster VPLEX dans les configurations synchrones ou asynchrones. Les modules E/S Fibre Channel sont composés de 4 ports situés en slot A2 pour le directeur A et B2 pour le directeur B (cf. figure 13 en rouge).

## **III.3.1.2 Eléments de configuration VPLEX**

### ***III.3.1.2.1 Gestion de la volumétrie***

La première phase est à effectuer au niveau de la ou des baies de disques alimentant le VPLEX en volumétrie. Dans le cas de la présente architecture une ou plusieurs baies de type VMAX. Au travers des ports Back-End des moteurs, le volume de données est présenté au cluster VPLEX. Ce volume de données ainsi provisionné vers le cluster VPLEX est nommé « Storage Volume ».

La deuxième phase se situe au niveau du cluster VPLEX : le « Storage Volume » est encapsulé par ce dernier grâce à une action appelée « Claiming ». Un « Extent », premier élément de configuration VPLEX, est ensuite créé sur le « Storage Volume ». Un « Extent » peut être constitué d'une partie ou de l'ensemble du « Storage Volume ».

Au niveau supérieur, un « Device » est créé en s'appuyant sur les « Extents » existants. Il est possible de combiner un ou plusieurs « Extents » au sein d'un « Device » unique en appliquant un type de RAID spécifique.

La présentation de la volumétrie aux serveurs zonés sur le VPLEX devient effective lors de la création d'un volume virtuel sur un « Device » préalablement créé et à l'inclusion de ce dernier au sein d'une entité logique appelée « Storage View ».

Ce fonctionnement ainsi que la structure hiérarchique des divers éléments de configuration VPLEX sont illustrés dans la figure suivante :

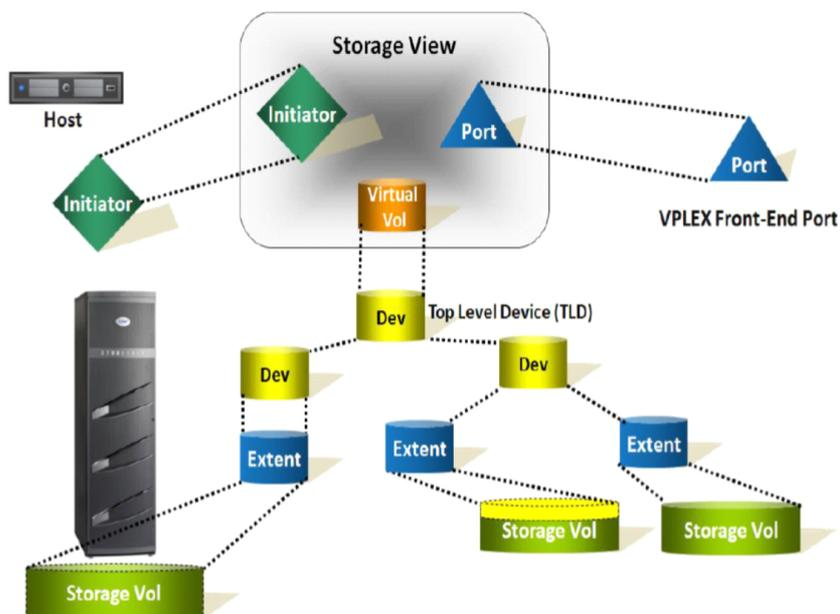


Figure 14 – Hiérarchie des éléments de configuration au sein d'un cluster VPLEX

### III.3.1.2 Distributed device

Un « Distributed Device » est un « Device » qui est présent sur les deux VPLEX du cluster en accès simultané actif/actif. Le « Distributed Device » est composé de volumétries appartenant aux baies de stockage situées sur 2 sites différents. En voici un exemple :

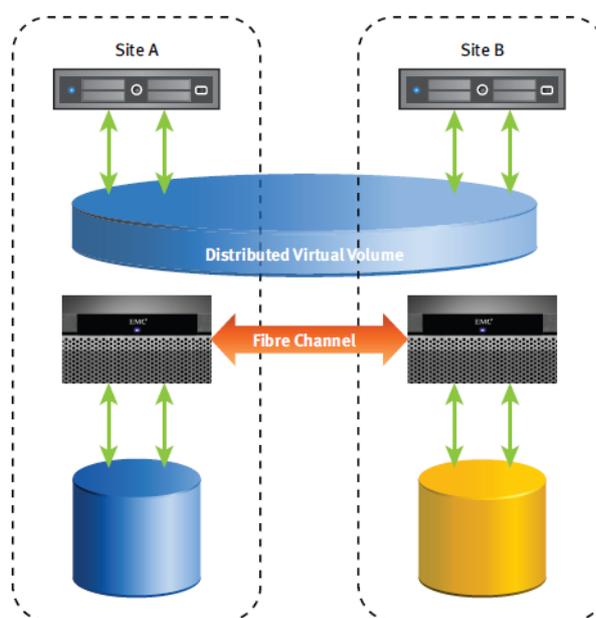


Figure 15 – Représentation d'un « Distributed Device » au sein d'un métró cluster VPLEX

### III.3.1.2.3 Storage View

La « Storage View » est l'élément de configuration par lequel les volumes de données sont présentés de manière effective aux serveurs les utilisant.

Une fois zonées vers les ports Front-End du VPLEX (action effectuée en intervenant sur les commutateurs Cisco Front-End), les cartes HBA des serveurs sont enregistrées dans la configuration de ce dernier. Cette opération permet de mémoriser les WWN des cartes HBA du serveur en leur associant un nom ainsi qu'un type correspondant au système d'exploitation de cette machine.

La « Storage View » est composée des éléments suivants :

- Les WWN des cartes HBA du serveur
- Des ports Front-End vers lesquels les HBA serveurs sont zonés
- Un ou plusieurs volumes virtuels à présenter au serveur

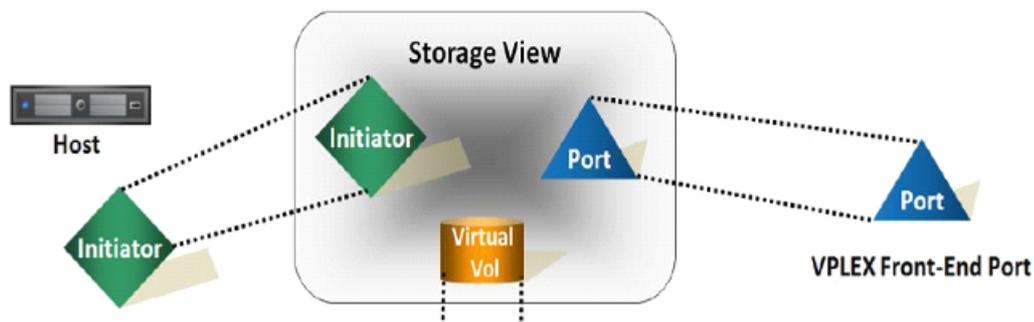


Figure 16 – Eléments constitutifs d'une « Storage View » VPLEX

### III.3.1.3 Normalisation des éléments de configuration

#### III.3.1.3.1 Structure des volumes de données

Comme indiqué précédemment, les volumes de données VPLEX sont composés de plusieurs éléments logiques qui peuvent être combinés selon de multiples configurations.

Afin de simplifier autant que possible l'administration du VPLEX et d'assurer une bonne lisibilité à l'ensemble, la règle du « 1 pour 1 » est appliquée. Cette règle stipule que chaque élément de configuration de rang supérieur ne doit s'appuyer que sur un seul élément de configuration de rang inférieur.

### III.3.1.3.2 Nomenclature des objets VPLEX

La nomenclature des différents éléments de configuration VPLEX devra respecter les règles suivantes :

Élément de configuration	Règle de nommage
Storage Volume	SVOL_<VMAX>_<HVE>
Extent	EXTT_<VMAX>_<HVE>
Device	LDEV_<VMAX>_<HVE>
Distributed Device	DDEV_<VMAX1>_<VMAX2>_<HVE>
Virtual Volume	<B G S>_<NOM_DEVICE>_<HVE>
Storage View	VIEW_<SERVEUR CLUSTER>

Tableau 3 – Nomenclature des différents éléments de configuration VPLEX

### III.3.1.4 Administration de la solution

#### III.3.1.4.1 Unisphere for VPLEX

L'administration d'un cluster VPLEX est assurée à l'aide d'un outil installé sur le « management server » de ce dernier et que l'on nomme « Unisphere for VPLEX ». Ce produit prend la forme d'une application Web accessible via un navigateur internet.



Figure 17 – Fenêtre principale de la plateforme d'administration «Unisphere for VPLEX»

### III.3.1.4.2 VPLEX CLI (Command Line Interface)

Outre l'utilisation de l'interface graphique, il est également possible d'administrer un cluster VPLEX à l'aide d'un ensemble de lignes de commandes installées sur le « management server » de ce dernier.

Cette solution logicielle d'administration est connue sous le nom de « VPLEX CLI » et prend la forme d'un système de fichiers virtuels dans lequel il est possible de visualiser les éléments de configuration comme s'il s'agissait de simples fichiers.

Toute opération d'administration effectuée via l'interface graphique « Unisphere for VPLEX » est également réalisable en ligne de commande, l'inverse n'est pas vrai. Certaines fonctionnalités clés ne sont pas encore disponibles via le GUI.

### III.3.1.4.3 API VPLEX

Afin de permettre le développement d'outils d'administration personnalisés, le VPLEX dispose également un ensemble d'API accessible par le « management server ».

Comme documenté dans [10], ces API fonctionnent selon le principe des API REST : les appels sont envoyés à l'aide du protocole HTTP et des requêtes GET/PUT/POST. Les réponses produites par le VPLEX utilisent le format de représentation de données JSON. La figure ci-dessous est un exemple de flux JSON VPLEX :

```
{
  "response": {
    "context": [
      {
        "name": "cluster-1",
        "parent": "/clusters",
        "type": "cluster",
        "allow-auto-join": 0,
        "auto-expel-period": 0,
        "auto-join-delay": 0,
        "cluster-id": 1,
        "connected": true,
        "default-cache-mode": "synchronous",
        "default-director": "director-1-1-A",
        "director-names": ["director-1-1-B", "director-1-1-A"],
        "health-indications": [],
        "health-state": "ok",
        "island-id": 1,
        "operational-status": "ok",
        "transition-indications": [],
        "transition-progress": [],
        "children": ["cluster-connectivity", "consistency-groups", "devices",
                   "exports", "storage-elements", "system-volumes",
                   "uninterruptible-power-supplies", "virtual-volumes" ]
      }
    ],
    "message": null,
    "exception": null,
    "custom-data": null
  }
}
```

Figure 18 – Réponse à une requête d'interrogation REST de type GET au format JSON

### **III.3.2 Spécifications techniques VMAX**

#### **III.3.2.1 Présentation du dispositif**

Le produit VMAX est une baie de disques issue de la gamme EMC<sup>2</sup> SYMMETRIX et qui représente le segment dit « haut de gamme ». Au sein de l'infrastructure de stockage i-BP, les VMAX ont pour rôle d'alimenter en volumétrie les dispositifs de virtualisation que sont les VPLEX. Ils assurent également les fonctions liées à la gestion de la qualité de service et aux techniques de copie de données.

Ces baies de disques sont réparties de la manière suivante :

- En hors production, une baie VMAX 10K sur chaque site (SAPHIR et TOPAZE)
- En production, deux baies VMAX 20K sur chaque site (SAPHIR et TOPAZE)

Un équipement VMAX est constitué d'une baie système et d'une ou plusieurs baies de stockage contenant les disques durs.

La baie système contient les éléments suivants :

- Des moteurs et les alimentations nécessaires
- La distribution de l'alimentation électrique pour les moteurs
- Les alimentations de secours (SPS)
- Le « Service Processor », un dispositif d'administration, avec son écran, son clavier, sa souris ainsi qu'une alimentation électrique de secours (UPS).

Les baies de stockage, ou « storage bay », contiennent :

- Les alimentations pour elles-mêmes
- 16 DAE, ou « Disk Array Enclosure », des emplacements où viennent se loger les disques durs et pouvant en recevoir 15 par DAE
- Les disques durs

##### ***III.3.2.1.1 La baie système***

La baie système est le cœur du système VMAX. Elle fait l'interface entre l'environnement extérieur, le VPLEX dans le cas présent, et les disques physiques contenus dans les baies de stockage. Comme indiqué plus haut cette baie système contient également les moteurs VMAX.

Chaque moteur VMAX contient deux directeurs. Chacun de ces directeurs est composé :

- De 16, 32 ou 64 Go de mémoire physique
- Deux modules Back-End qui assurent la connexion aux baies de disques
- Deux modules Front-End qui assurent la connectivité vers le VPLEX. Plusieurs combinaisons en types et en nombre de ports existent pour ces modules E/S.

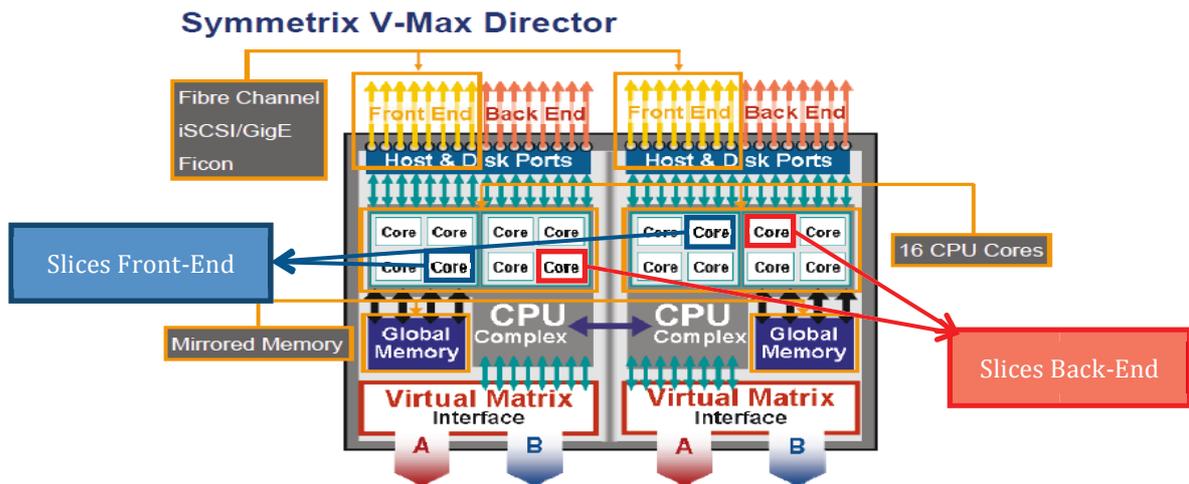


Figure 19 – Vue logique d'un directeur VMAX

### III.3.2.1.2 Les baies de stockage ou «Storage bay»

Les baies de stockage sont les baies qui vont recevoir physiquement les disques durs du dispositif. Ces baies ne sont raccordées qu'à la baie système du VMAX. A pleine capacité chacune de ces baies peut embarquer 240 disques.

#### III.3.2.1.2.1 Gestion des disques par Octant

La gestion des disques dans un VMAX se fait par octant. Cela signifie qu'un ensemble de disques issus d'une baie de stockage est pris en charge par un moteur spécifique.

Comme l'indique la figure ci-dessous, chaque groupe de disques d'une couleur donnée est géré par le moteur de la même couleur.

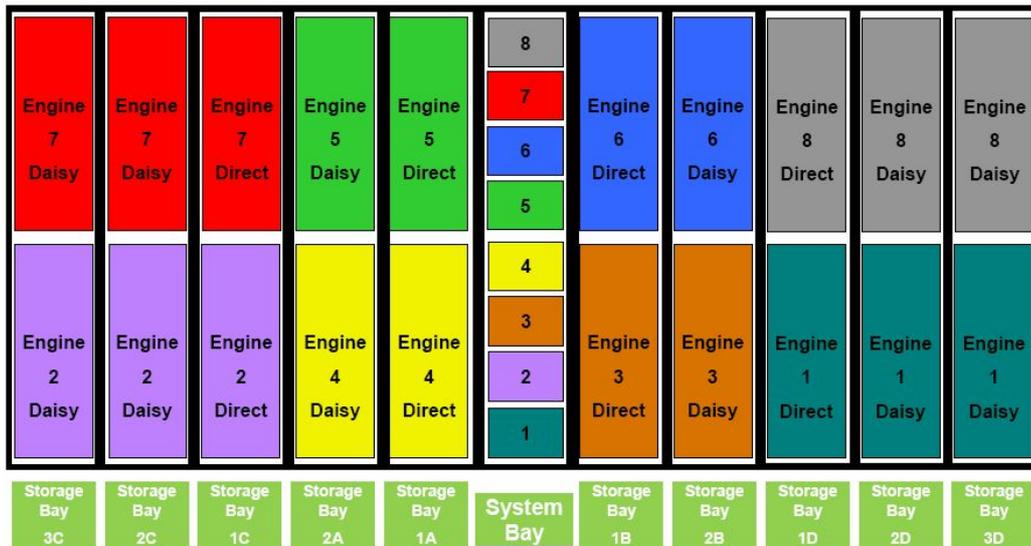


Figure 20 – Gestion des disques en octant d'un VMAX

### III.3.2.1.2.2 Disques

Comme le montre la figure ci-dessous, un VMAX peut embarquer plusieurs technologies de disques durs, présents en quantité variable selon les usages et besoins.

Drive Type <sup>1</sup>	Drive size	Rotational Speed (RPM)	Formatted capacity		
			Open Systems	Mainframe	IBM i
Fibre Channel 146 GB	3.5"	15K	143.53 GB	139.34 GB	141.78 GB
Fibre Channel 300 GB	3.5"	15K	288.19 GB	279.77 GB	288.23 GB
Fibre Channel 450 GB	3.5"	15K	432.29 GB	419.65 GB	435.14 GB
Fibre Channel 600 GB	3.5"	15K	576.39 GB	559.54 GB	580.18 GB
Fibre Channel 300 GB	3.5"	10K	288.19 GB	279.77 GB	288.23 GB
Fibre Channel 400 GB	3.5"	10K	393.84 GB	382.33 GB	389.80 GB
Fibre Channel 450 GB	3.5"	10K	432.29 GB	419.65 GB	435.14 GB
Fibre Channel 600 GB	3.5"	10K	576.39 GB	559.54 GB	580.18 GB
SATA II 1000 GB	3.5"	7.2K	984.81 GB	956.02 GB	985.05 GB
SATA II 2 TB	3.5"	7.2K	1882.72 GB	1827.67 GB	1912.14 GB
SAS 300 GB	3.5"	10K	288.19 GB	279.77 GB	288.23 GB
SAS 450 GB	3.5"	10K	432.29 GB	419.65 GB	435.14 GB
SAS 600 GB	3.5"	10K	576.39 GB	559.54 GB	580.18 GB
Flash 100 GB	3.5"	n/a	98.49 GB	95.61 GB	98.51 GB
Flash 200 GB	3.5"	n/a	196.97 GB	191.21 GB	197.01 GB
Flash 400 GB	3.5"	n/a	393.84 GB	382.33 GB	389.80 GB

1. A maximum of 1024 hypervolumes on each physical drive.

Figure 21 –Types de disques durs supportés par un VMAX

Les VMAX de la société i-BP contiennent des disques durs de technologies Fibre Channel, SATA et Flash.

### **III.3.2.2 Fonctionnalités implémentées**

Les fonctionnalités suivantes sont implémentées dans les VMAX de l'architecture i-BP :

- Configuration de type « Open »
- Virtual Provisioning
- Fast/VP
- Timefinder VP/Snap

#### ***III.3.2.2.1 Configuration de type « Open »***

Les baies EMC<sup>2</sup> VMAX supportent plusieurs types d'environnements et permettent de travailler avec des Mainframes, des IBM type iSeries ou encore des serveurs compatibles Intel x86.

Dans le cas présent, toutes les baies VMAX présentent sur les sites de SAPHIR et de TOPAZE sont dotées de disques formatés pour fournir une volumétrie exclusivement destinée à des environnements de type « Open » (Intel x86).

#### ***III.3.2.2.2 Virtual Provisioning***

Le « Virtual Provisioning » est l'implémentation EMC<sup>2</sup> d'une technologie connue sous le nom de « Thin Provisioning ».

Le « Thin Provisioning » est une technique permettant de présenter à un serveur une volumétrie dont l'espace physique sera alloué à la demande, au fil des requêtes d'écriture. Cette volumétrie physique est approvisionnée grâce à des agrégats de volumes physiques appelés pools.

Le mécanisme de « Thin provisioning » permet de sur-allouer de l'espace disque en présentant à un serveur davantage de volumétrie qu'il n'y en a physiquement de disponible dans la baie. Cette technique a l'avantage d'optimiser l'utilisation des ressources physiques disponibles.

Il est donc particulièrement important de surveiller attentivement et très régulièrement le niveau d'utilisation des pools pour éviter les situations de saturation de la capacité de stockage.

Si les pools venaient à épuiser leurs capacités physiques, toute requête d'écriture émanant d'un serveur ne pourrait être satisfaite, entraînant des erreurs applicatives et dans certains cas des crashes voire des corruptions de données.

Il est donc primordial de mettre en œuvre un ensemble de procédures et de mécanismes d'approvisionnement selon l'évolution de la consommation des pools.

#### **III.3.2.2.2.1 Avantages**

Le principal avantage du « Virtual Provisioning » réside dans le fait qu'il permet de réduire les coûts d'administration de l'infrastructure de stockage :

- Le stockage nécessaire à un serveur peut être préprovisionné sans que le stockage physique soit disponible dans la baie
- Le mode de gestion par pool permet d'optimiser le taux d'utilisation du stockage et de réduire les pertes

La réduction du coût lié au stockage physique est également un élément à prendre en compte. En mode « Thin provisioning » seul l'espace réellement utilisé est consommé. Or, on constate régulièrement un taux d'utilisation des systèmes de fichiers compris entre 50 et 80 %. L'espace disque non consommé peut être ainsi utilisé par les autres serveurs du parc.

Une fois le mode « Virtual Provisioning » est mis en œuvre, l'activité principale de l'administrateur est de surveiller le niveau d'utilisation des pools.

#### **III.3.2.2.3 FAST/VP**

FAST/VP est un mécanisme interne aux baies VMAX, qui automatise :

- L'identification des données applicatives particulièrement actives (ou inactives)
- Le déplacement de la volumétrie correspondante sur différentes technologies de disques (appelés « Tiers »)

Le dispositif FAST/VP permet de maintenir une bonne qualité de service dans le temps en déplaçant automatiquement les données vers la technologie de stockage la plus appropriée. Cela permet d'optimiser les performances ainsi que les coûts.

Pour ce faire, FAST/VP surveille les charges d'entrées/sorties au niveau des volumes de données pour identifier ceux qui présentent un niveau d'utilisation important et qui bénéficieraient d'un déplacement sur des disques de haute performance.

A l'opposé, FAST/VP identifie également les données les moins sollicitées et peut les déplacer sur des disques plus capacitifs mais moins performants.

Cette activité d'analyse des données est fondée sur des politiques (ou « polices ») qui associent un groupe de volumes à différentes technologies de disques et type de protection RAID.

La décision de déplacer telle ou telle donnée est basée sur l'échantillonnage d'une période d'activité définie par l'utilisateur : c'est la fenêtre d'analyse. Les déplacements de données sont réalisés durant une période de temps elle aussi définie par l'utilisateur : la fenêtre de mouvement.

Les mouvements de données sont réalisés à chaud, sans interruption de service ni impact substantiel sur l'activité.

#### **III.3.2.2.3.1 Avantages**

Les avantages liés à l'utilisation de la technologie FAST/VP d'un point de vue opérationnel sont :

- La gestion automatique et continue des aspects de l'optimisation des performances
- De meilleurs temps de réponse grâce à l'utilisation des disques Flash (EFD/SSD)
- La réduction des coûts de stockage à l'aide des disques SATA

#### **III.3.2.2.3.2 Composants FAST/VP**

Les trois principaux éléments à l'œuvre dans FAST/VP sont :

- Le « Thin Tier » : une ressource de stockage partagée, dotée de caractéristiques communes
- Les politiques FAST/VP : des règles qui gèrent le positionnement et le déplacement des données entre les différents niveaux de stockage

- Le « Storage Group » : un regroupement logique de volumes en vue d'une gestion commune, assujetti à une politique FAST/VP.

La figure ci-dessous illustre les relations existantes entre les divers éléments de configuration de la technologie FAST/VP :



Figure 22 – Composants utilisés dans une configuration de type FAST/VP

#### III.3.2.2.4 Timefinder SNAP/VP

Avec l'introduction des technologies de « Virtual Provisionning » dans les baies de la gamme SYMMETRIX, EMC<sup>2</sup> a aussi fait évoluer sa solution de copie par instantanés, dénommée « Timefinder VP Snap ».

La technologie de copie par instantané permet de prendre des images des volumes de données sans consommer pour autant la même capacité que le volume source. Seules les données modifiées depuis la prise d'image feront l'objet d'une sauvegarde. Les deux volumes impliqués dans la session « VP Snap » doivent être de type « Thin ». Il est possible de copier plusieurs fois un même volume source : cela engendre la mutualisation des données consommées dans le pool physique sous-jacent.

Le schéma ci-dessous illustre la relation liant un volume de données source d'une session « Snap VP » et ses trois copies ainsi que l'impact sur la capacité de stockage physique :

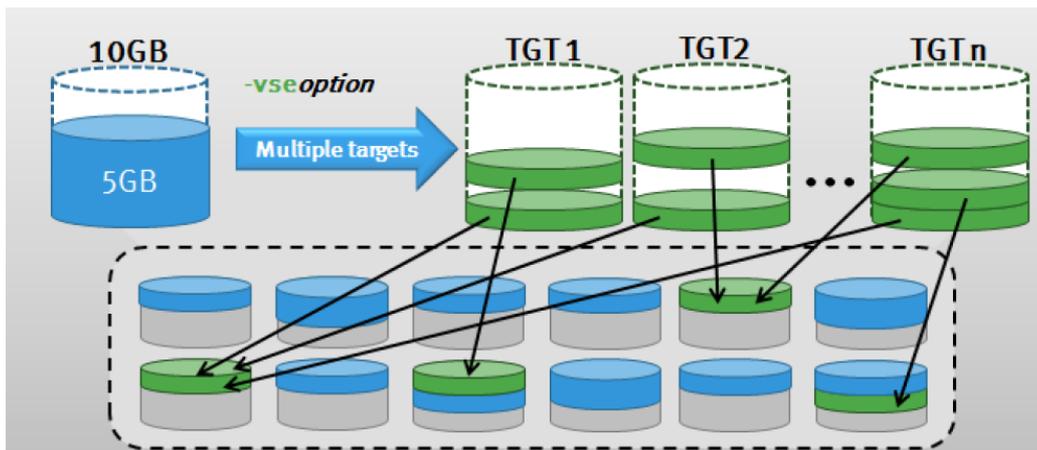


Figure 23 – Schéma de principe de la fonctionnalité « Timefinder SNAP/VP »

### III.3.2.3 Eléments de configuration VMAX

#### III.3.2.3.1 Auto-provisionning VMAX

L'« Auto-Provisionning » est un nouveau mécanisme de présentation des volumes de données aux serveurs, introduit par les baies VMAX.

Il repose sur la définition d'enveloppes logiques par type d'éléments de configuration qui, reliées entre elles, assurent la visibilité des volumes aux serveurs.

Ces enveloppes sont de trois types :

- Le « Storage Group » : il regroupe les volumes qui seront présentés simultanément à un serveur ou à un ensemble de serveurs
- L'« Initiator Group » : il regroupe les WWN à qui l'on donnera simultanément à voir la volumétrie
- Le « Port Group » : il regroupe les ports Front-End qui présenteront la volumétrie aux HBA

Ces groupes sont ensuite liés entre eux par une enveloppe complémentaire : la « Masking View ». C'est cette dernière enveloppe qui permet de déclarer le chemin d'accès aux données.

Il existe une forme particulière de « Storage Group » appelée « Storage Group cascadié ». Cet objet permet de créer un « Storage Group » qui contient d'autres « Storage Groups ».

Cela peut s'avérer utile pour associer différentes politiques FAST/VP à un sous-ensemble de volumes vus par un même serveur.

Le même concept existe au niveau « Initiator Group ». Celui-ci permet de grouper dans un « Initiator Group », d'autres « Initiators Group ». Par exemple, pour un cluster, il est envisageable de créer un IG cascadié et d'y inclure les IG de chaque nœud du cluster.

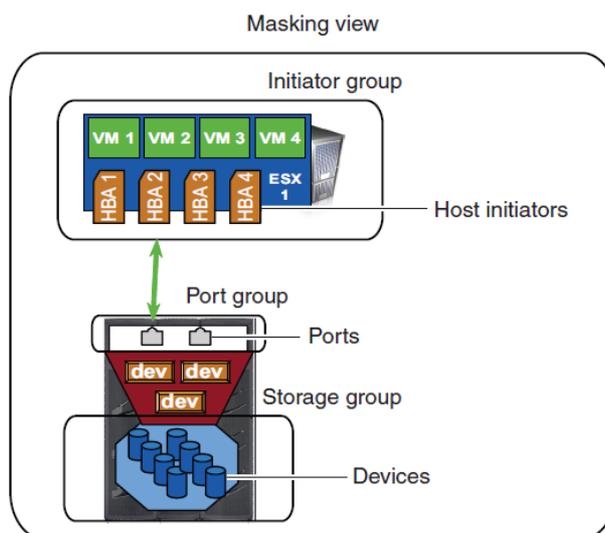


Figure 24 –Illustration des différents groupes logiques VMAX

Du fait de la définition des groupes et de la création d'une « Masking View », la baie VMAX se chargera automatiquement de mapper les volumes du « Storage Group » sur les ports du « Port Group ». Elle masquera ensuite les WWNs des HBA sur les ports du « Port Group » pour autoriser l'accès aux volumes du « Storage Group ».

L'«Auto-Provisioning » permet d'accélérer les opérations d'exploitation : dans le cas d'un ajout de volume à un serveur préalablement configuré, il suffit d'ajouter ce dernier à la « Masking View » existante.

L'« Auto-Provisioning » permet également de gérer fonctionnellement les affectations. Il est ainsi possible, dans le cadre d'un cluster applicatif de définir des « Storage Groups » locaux, propres à chaque serveur et des « Storage Group » fonctionnels présentant les volumes simultanément aux différents serveurs du cluster.

### III.3.2.4 Normalisation des éléments de configuration

Dans le contexte i-BP, la nomenclature suivante est utilisée :

Groupe	Nomenclature
<b>Storage Groups</b>	SG_<HOSTNAME>
<b>Initiators Group</b>	IG_<HOSTNAME>_F<1 2>
<b>Ports Group</b>	PG_<HOSTNAME>_<DIRPORT>
<b>Masking View</b>	MV_<HOSTNAME>

Tableau 4 – Nomenclature des éléments de configuration VMAX

### III.3.2.5 Administration de la solution

#### III.3.2.5.1 Solutions Enabler

Les baies VMAX peuvent être administrées à l'aide d'une suite logicielle composée de lignes de commandes que l'on appelle « Solutions Enabler ».

Cette suite logicielle a la particularité de piloter les baies VMAX en mode « in-band » ce qui signifie que les ordres générés par les commandes « Solutions Enabler » transitent directement via la liaison Fibre Channel reliant le serveur d'administration à la baie qu'il administre.

Cette communication est réalisée par le truchement de disques particuliers appelés « GateKeeper ». Ces derniers sont présentés au serveur directement depuis la baie VMAX lors de la phase de configuration initiale.

La suite « Solutions Enabler » peut également fonctionner en mode proxy afin de relayer les ordres de configuration de serveurs distants ne disposant pas d'une liaison directe avec la baie VMAX à administrer.

#### III.3.2.5.2 Unisphere for VMAX

Outre l'administration en mode ligne de commande, il est également possible de configurer une baie VMAX à l'aide d'un logiciel accessible via un navigateur Internet : « Unisphere for VMAX ».

Pour assurer son fonctionnement, ce logiciel s'appuie sur les fondations de « Solutions Enabler ». Ainsi, l'installation de « Solutions Enabler » est un prérequis indispensable à l'installation de l'outil « Unisphere ».

### III.3.2.5.3 API VMAX

Enfin, il est également possible d'industrialiser l'administration d'une baie de disques de type VMAX à l'aide des API conçu à cet effet.

Ces API ont la particularité de reposer sur les commandes natives utilisées par la suite « Solutions Enabler » accompagnée de la possibilité d'échanger des flux XML comme indiqué dans [10].

```
symcfg list -out xml OR symcfg list -out xml_element

<?xml version="1.0" standalone="yes" ?>

<SymCLI_ML>
  <Symmetrix>
    <Symm_Info>
      <symid>000190102055</symid>
      <attachment>Local</attachment>
      <model>DMX3-24</model>
      <microcode_version>5772</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <devices>683</devices>
      <physical_devices>94</physical_devices>
    </Symm_Info>
  </Symmetrix>
  <Symmetrix>
    <Symm_Info>
      <symid>000190300215</symid>
      <attachment>Local</attachment>
      <model>DMX3-6</model>
      <microcode_version>5772</microcode_version>
      <cache_megabytes>32768</cache_megabytes>
      <devices>239</devices>
      <physical_devices>2</physical_devices>
    </Symm_Info>
  </Symmetrix>
  <Symmetrix>
    <Symm_Info>
      <symid>000190300237</symid>
      <attachment>Remote</attachment>
      <model>DMX3-6</model>
      <microcode_version>5773</microcode_version>
      <cache_megabytes>16384</cache_megabytes>
      <devices>3756</devices>
    </Symm_Info>
  </Symmetrix>
</SymCLI_ML>
```

Figure 25 – Exemple de flux XML utilisé par la suite « Solutions Enabler »

Les flux XML ainsi produits peuvent ensuite être analysés puis traités par la quasi-totalité des langages de programmation du marché.

### **III.3.3 Spécifications techniques CISCO**

#### **III.3.3.1 Présentation du dispositif**

La société i-BP souhaite remplacer son infrastructure SAN vieillissante de marque Brocade par une nouvelle infrastructure de marque Cisco.

Cette nouvelle infrastructure est appelée SAN Front-End, car elle se situe en amont de la nouvelle solution de stockage VPLEX/VMAX. Tous les serveurs du parc souhaitant accéder à de la volumétrie issue de cette nouvelle solution de stockage doivent être raccordés à ce réseau SAN.

##### ***III.3.3.1.1 Présentation du SAN Front-End***

L'architecture mise en œuvre est constituée d'une combinaison de quatre commutateurs Fibre Channel Cisco MDS9710 et de quatre commutateurs Fibre Channel Cisco MDS9148S en architecture « Core/Edge ». Ils sont repartis entre deux « fabrics » indépendantes étendues entre les sites de SAPHIR et de TOPAZE.

Ces « fabrics » se nomment « fabric 1 » et « fabric 2 ».

Les MDS9710 constituent les commutateurs dits « Core », les MDS9148S constituent les commutateurs dits « Edge ». Les VPLEX sont raccordés à chaque « fabric » dans un souci de redondance maximale.

Les connexions inter-commutateurs, appelées ISL, entre les salles de SAPHIR et TOPAZE s'effectuent via des fibres monomodes (quatre liens de 16Gbps chacun). Les autres connexions ISL s'effectuent en fibres multimodes, pour les connexions intra site.

Les commutateurs MDS9710 « Core » accueillent les connexions des VPLEX, des serveurs de sauvegarde et des autres équipements du réseau qui génèrent un trafic particulièrement conséquent.

Sur ces équipements physiques, plusieurs SAN virtuels sont créés, appelés VSAN. Ils permettent de créer une isolation logique des environnements disques et sauvegarde.

### III.3.3.1.2 Topologie physique du réseau SAN Front-End

La topologie physique du SAN Cisco Front-End retenue est la suivante :

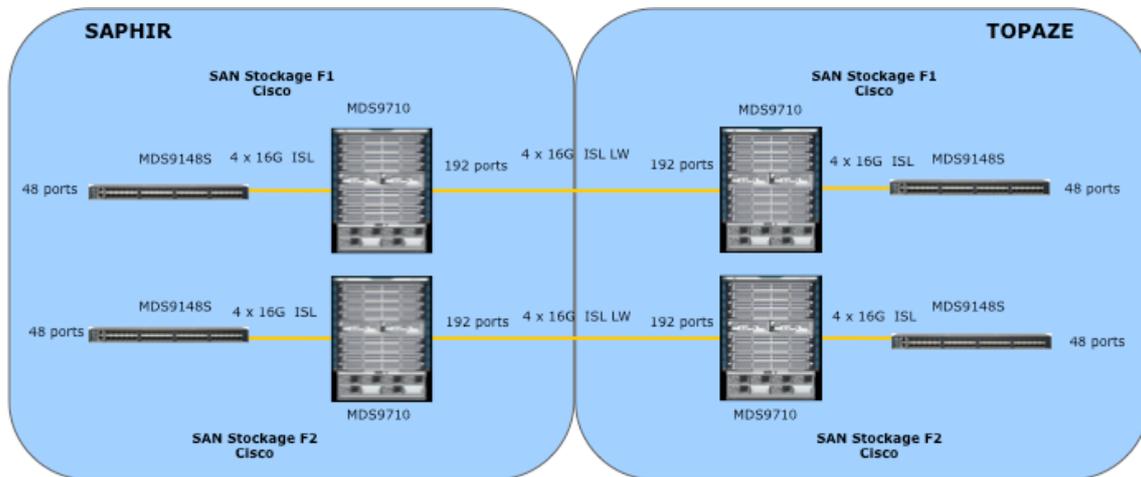


Figure 26 – Architecture physique du SAN Cisco Front-End

### III.3.3.1.3 Configuration matérielle Cisco SAN Front-End

Les commutateurs MDS9710 comprennent tous :

- 4 cartes 48 ports
- 184 x 2/4/8-Gbps FC shortwave SFP LC
- 4 x 4/8/16-Gbps FC shortwave SFP LC
- 4 x 4/8/16-Gbps FC longwave SFP LC

Les commutateurs MDS9148S comprennent tous :

- 44 x 2/4/8-Gbps FC shortwave SFP LC
- 4 x 4/8/16-Gbps FC shortwave SFP LC

Le commutateur FC Cisco MDS 9710 est un commutateur haute disponibilité avec dix cartes de 48 ports dans un châssis de 14U :



Figure 27 – Vue physique d’un commutateur Cisco MDS9710

Le commutateur FC Cisco MDS 9148S réunit la facilité d’utilisation, la flexibilité, la haute disponibilité et la sécurité dans une unité compacte de 1U seulement.

Avec une capacité d’extension de 16 à 48 ports FC 2/4/8/16 Gbps. Le Cisco MDS 9148S offre aussi bien les densités exigées par les commutateurs SAN des départements d’entreprise que par les commutateurs d’accès sur des architectures SAN de type « Core/Edge ». Le MDS 9148S utilise le logiciel Cisco MDS 9000 NXOS :



Figure 28 – Vue physique d’un commutateur Cisco MDS9148S

### III.3.3.2 Eléments de configuration

#### III.3.3.2.1 Attribution des VSAN ID

La définition des VSAN est faite telle que décrite dans le tableau ci-dessous :

				VSAN	Default VSAN	IB-DSK1/2	IB-BCK1/2
Salle	Chassis	Fabric	Nom		VSAN 1	VSAN 100	VSAN 500
SAPHIR	IBSAN100	1	IBSAN100			1	10
SAPHIR	IBSAN200	2	IBSAN200		1	10	50
SAPHIR	IBSAN110	1	IBSAN110		2	11	51
SAPHIR	IBSAN210	2	IBSAN210		2	11	51
TOPAZE	IBSAN101	1	IBSAN101		3	12	52
TOPAZE	IBSAN201	2	IBSAN201		3	12	52
TOPAZE	IBSAN111	1	IBSAN111		4	13	53
TOPAZE	IBSAN211	2	IBSAN211		4	13	53

Tableau 5 – Assignment des VSAN ID par commutateur

#### III.3.3.2.2 Constitution des zones

La connectivité entre les équipements raccordés à une « fabric » est assurée à l'aide d'entités appelées « zones ». Ces zones peuvent contenir un nombre variable d'alias. Les alias sont les enveloppes contenant les adresses WWN des serveurs et autres équipements du réseau.

Afin de structurer au mieux la configuration du SAN, il a été décidé que :

- Les WWN d'un serveur ne peuvent être utilisés que dans un seul alias par « fabric »
- Les alias peuvent être utilisés dans un nombre quelconque de zones
- Une zone ne peut être constituée que de deux membres (deux alias au maximum)

### III.3.3.3 Normalisation des éléments de configuration

Dans le cadre de ce projet, la normalisation des objets SAN a été établie selon la norme suivante :

Élément	Nomenclature
Zoneset	<Z1 Z2>_Zoneset
Zone	<Z1 Z2>_<ALIAS1>_<ALIAS2>
Alias	<SERVEUR>_HBA<1 2>

Tableau 6 – Nomenclature des éléments liés au réseau SAN Front-End

### III.3.3.4 Administration de la solution

#### III.3.3.4.1 Cisco DCNM (Data Center Network Manager)

L'administration de l'infrastructure SAN est possible en mode graphique à l'aide de l'outil Cisco DCNM.

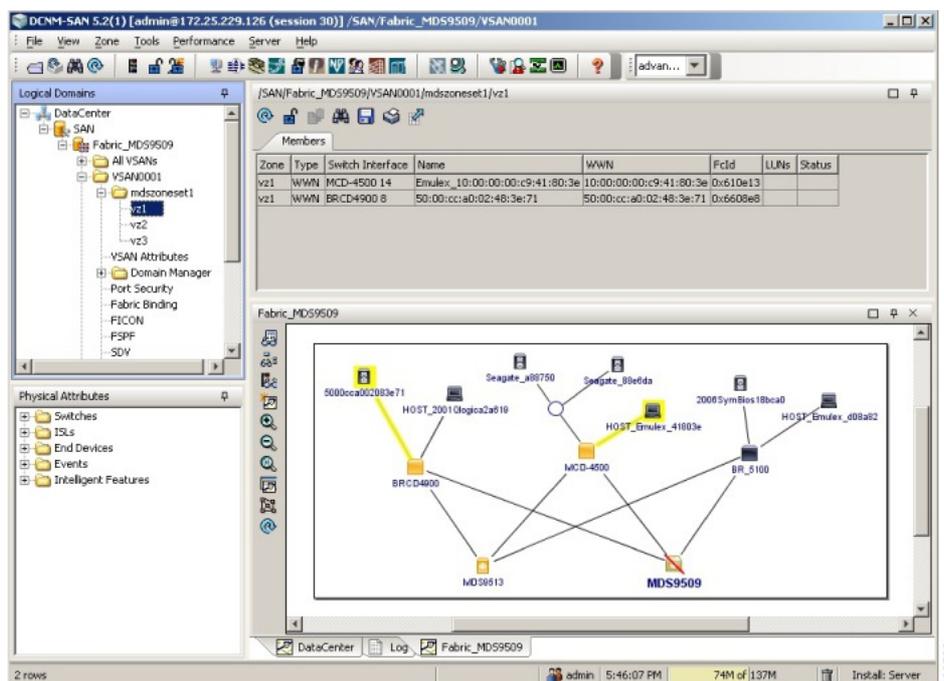


Figure 29 – Fenêtre principale de l'outil d'administration Cisco DCNM

DCNM permet de découvrir plusieurs commutateurs simultanément ainsi que la topologie du SAN par parcours des liaisons inter-commutateurs et analyse des configurations de ces derniers.

Dans le contexte i-BP, DCNM est utilisé pour assurer :

- La configuration des commutateurs en mode graphique
- Le diagnostic en cas d'incident
- L'activation des modifications de zoning
- Le suivi de performance des commutateurs
- La réception des alertes envoyées par les commutateurs
- Le stockage des configurations des commutateurs

#### ***III.3.3.4.2 Cisco CLI***

De loin le mode d'administration le plus utilisé dans le monde Cisco, il est également possible de se connecter directement sur les commutateurs à administrer afin d'utiliser la ligne de commande de ces derniers. Cet accès s'effectue au travers d'une liaison de type SSH.

La ligne de commande utilisée par NXOS, le logiciel utilisé au sein des commutateurs Cisco de la famille MDS, est très voisine de son homologue dans le monde IP et permet de contrôler tous les aspects relatifs à la « fabric » Fibre Channel.

#### ***III.3.3.4.3 API Cisco***

Contrairement aux commutateurs Nexus utilisés dans le monde IP, la famille des commutateurs MDS ne met pas à disposition d'API.

Fort heureusement, les équipements Cisco étant particulièrement bien représentés dans le monde professionnel et la communauté open source étant très active sur ces sujets, il existe de nombreuses alternatives libres rendant possible l'analyse et l'extraction de données issues de la configuration de tels équipements.

### **III.3.4 Choix du langage de programmation : Python**

Pour réaliser ce projet, il a fallu choisir un langage de programmation parmi ceux qui sont disponibles sur le marché. C'est le langage Python qui fût retenu. Les caractéristiques de ce langage, ses spécificités et ses avantages sont abordés dans la suite de cette section.

#### **III.3.4.1 Historique du langage Python**

Apparu en 1990, le langage de programmation Python est un langage de type interprété, multiparadigmes et multiplateformes créé par Guido Van Rossum.

Ce langage s'est rapidement imposé dans l'industrie grâce à ses fonctionnalités avancées, sa large bibliothèque de modules tiers et sa syntaxe épurée au point d'être aujourd'hui devenu incontournable.

Python est omniprésent au sein de toutes les distributions Linux et est largement utilisé pour la réalisation d'outils d'administration système ou de scripts permettant d'assurer les tâches de maintenance courantes.

D'autres domaines utilisent très largement Python. C'est le cas du développement Web où de très nombreux cadres existent (Django, Web2py, Flask, etc.) et sont employés pour la création de sites Web dynamiques. Le calcul scientifique fait également un usage abondant du langage Python grâce à ses modules spécialisés dans ce domaine, « numpy » notamment, et sa capacité à gérer les nombres complexes de manière native.

#### **III.3.4.2 Quelle version de Python choisir ?**

A ce jour deux versions majeures de Python coexistent : la version 2 et la version 3. La version 2 représente la branche historique, branche que ne reçoit à ce jour plus de mises à jour majeures ou d'ajout de fonctionnalités. Seuls les problèmes de sécurité et de performance font l'objet de correctifs.

La version 3 quant à elle, apparue en 2009, est issue d'un travail de réflexion autour du langage en lui-même et de son évolution. Cette version majeure a été l'occasion de supprimer du langage certains types de base, de refondre la gestion des chaînes de caractères (utilisation de l'UNICODE par défaut) et de modifier le fonctionnement de

certaines fonctions. De fait, cette évolution du langage a rendu les codes sources développés en version 2 partiellement incompatibles avec la version 3.

Pour ce projet, la version de Python retenue pour le développement est la version 3. L'utilisation de la version 2 ne se justifiant que pour des projets nécessitant l'emploi de modules indispensables au développement et non encore portés en version 3

### III.3.4.3 Fonctionnalités du langage Python

Le langage Python est un langage orienté objet, impératif et disposant d'un système de typage fort dynamique. La gestion de la mémoire est assurée par un ramasse-miettes et la gestion des erreurs au travers d'une arborescence d'exceptions.

Python facilite également la manipulation des données grâce à ses nombreux types par défaut qui comprennent (liste non exhaustive) :

- Les types numériques de base que l'on qualifiera de standards (int, float, complex)
- Les chaînes de caractères au format UNICODE
- Les tables de hachages ou « dictionnaires » selon l'appellation Python
- Les ensembles
- Les listes
- Les piles et files

### III.3.4.4 Fonctionnement de l'interpréteur Python

Comme évoqué précédemment, Python est un langage interprété dont le fonctionnement peut être illustré par la figure ci-dessous :

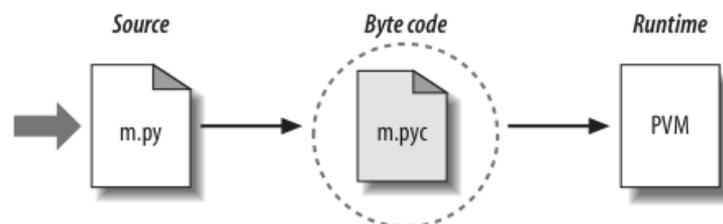


Figure 30 – Fonctionnement de l'interpréteur Python (source : [10])

Les fichiers contenant le code source Python portent l'extension « .py ». Lorsque ces fichiers sont traités pour la première fois par l'interpréteur, celui-ci analyse leurs contenus, puis transforme les instructions qui y figurent en ce que l'on nomme un « byte code » qui

est une forme de code intermédiaire situé entre les instructions-machine et le code source en lui-même.

Le « byte code » produit est stocké au sein de fichiers portant l'extension « .pyc ». Tant que le code source ayant produit ce « byte code » n'est modifié, les fichiers « .pyc » correspondants ne seront pas recréés. Cela permet d'améliorer les performances des exécutions successives des scripts Python.

Pour terminer, le « byte code » est à son tour analysé par la machine virtuelle Python, la PVM, qui se charge d'exécuter de manière effective les instructions présentes dans le « byte code ».

On relèvera que ce mode de fonctionnement (code source, « byte code », machine virtuelle) est commun à de nombreux langages. C'est notamment le cas du langage Java.

#### **III.3.4.5 Avantages de Python sur les autres langages de scripting**

Le langage Python dispose d'un très grand nombre d'avantages qui en font un très bon choix pour ce type de projet.

En premier lieu, Python est un langage dont la syntaxe épurée et ses choix de design en font un langage très puissant, simple à apprendre et dont le code source reste particulièrement lisible, le rendant très proche d'un langage naturel.

Les développements utilisant le langage Python sont traditionnellement plus rapides que ceux qui sont réalisés à l'aide de langages tels que le C ou le Java. Le langage Python étant bien moins contraignant dans son utilisation que les autres langages, le gain de temps peut s'avérer substantiel.

Python est également un langage hautement portable et multi plates-formes. Il est donc possible de l'utiliser sur un large panel de systèmes d'exploitation comme en témoigne la liste suivante :

- Linux et les systèmes UNIX (AIX/HP-UX/Solaris/BSD)
- Microsoft Windows
- Apple OSX

- IBM AS400
- Be/OS
- Terminaux mobiles (Android, Symbian OS)

Grâce à sa structure modulaire et sa communauté active Python dispose d'une bibliothèque de modules, accessibles via le portail « PyPi », très conséquente pouvant répondre à tous les besoins.

Enfin, Python est un langage de choix lorsqu'il s'agit de manipuler les divers formats de représentation et de sérialisation de données comme XML et JSON pour ne citer que ces derniers. Les modules permettant de traiter ces formats font partie de la bibliothèque standard de Python et il n'est donc pas nécessaire d'avoir recours à un module tiers.

#### **III.3.4.6 Implémentations du langage Python**

Le langage Python dispose de plusieurs implémentations. L'implémentation de référence, et donc la plus répandue et la plus connue, est celle accessible directement depuis le site officiel du projet Python. Elle est désignée par l'appellation « CPython ». Comme son nom le laisse supposer, il s'agit d'une implémentation s'appuyant sur le langage C.

Il existe également une implémentation qui utilise Java en lieu et place du langage C. Cette mouture est connue sous le nom de « Jython ». Cette version se distingue de la précédente par son mode de fonctionnement. En effet, « Jython » est composé de classes Java qui interprètent et compilent le code source Python en « byte code » Java. Cette implémentation de Python est utilisée dans le cadre de l'écosystème Java pour réaliser des applets.

Enfin, il existe une implémentation réalisée à l'aide du langage C# de Microsoft et que l'on nomme « IronPython ». L'idée ici est de réaliser un pont entre l'univers Python et les bibliothèques et applications se basent sur le cadriciel Microsoft .NET.

#### **III.3.4.7 Performance du langage Python**

Qu'en est-il des performances générales du langage Python ? Indéniablement, en tant que langage interprété, les performances de ce dernier sont très en deçà des langages compilés comme le langage C ou le C++.

Cependant, il est possible d'augmenter les performances du langage Python de manière très significative pour tendre vers la performance des langages compilés.

### III.3.4.7.1 Cython

Cython est un compilateur permettant de convertir du code Python en code C, code qui est par la suite compilé. Cette astuce permet donc de l'affranchir de la machine virtuelle Python puisque le code ainsi obtenu est totalement indépendant de celle-ci : il s'agit d'un binaire autonome.

Malgré l'avantage que représente Cython en matière de gain de performances, force est de constater que le code Cython n'est plus réellement du Python. En effet, pour pouvoir convertir le code source Python en code source C, Cython doit déterminer les types utilisés par chaque variable. Le langage Python étant un langage à typage fort dynamique, c'est à l'exécution du code que ce type sera connu de l'interpréteur.

Cython intervient sur ce point en altérant la syntaxe native de Python pour rendre possible la déclaration des types de chaque variable :

```
def primes(int kmax):
    cdef int n, k, i
    cdef int p[1000]
    result = []
    if kmax > 1000:
        kmax = 1000
    k = 0
    n = 2
    while k < kmax:
        i = 0
        while i < k and n % p[i] != 0:
            i = i + 1
        if i == k:
            p[k] = n
            k = k + 1
            result.append(n)
        n = n + 1
    return result
```

Figure 31 – Exemple de code Cython permettant de déclarer le type des variables

### III.3.4.7.2 Numba

Numba utilise une approche moins intrusive que celle de Cython. L'idée principale est ici de compiler directement en mémoire des portions de code Python.

Pour ce faire, Numba s'appuie sur le compilateur LLVM (pour « Low Level Virtual Machine ») ainsi que sur un module directement importable dans le code source Python et fournissant un ensemble de « décorateurs » permettant de désigner les portions de code à compiler en mémoire.

### **III.4 Conception de la solution d'automatisation**

Avant d'entamer la phase de réalisation à proprement parler il convient d'effectuer la modélisation de la solution en tant que telle. Cette étape a pour objet de présenter la structure fonctionnelle et technique de la solution considérée dans son ensemble.

La modélisation évoquée dans la présente section respectera le lotissement du projet vu précédemment au chapitre III.2.1.

Pour chaque lot, les cas d'utilisation seront énumérés. Ces diagrammes permettront de mettre en avant les principales fonctionnalités attendues, les interactions entre cas d'utilisation et la place de l'utilisateur dans le dispositif.

Chaque cas d'utilisation sera ensuite détaillé à l'aide d'un diagramme d'activités permettant de mettre en évidence les différentes étapes et contraintes de ce dernier.

#### **III.4.1 Lot A – Conception des modules d'administration VPLEX**

Les modules d'administration VPLEX ont pour objet de permettre aux ingénieurs stockage de l'équipe VSS d'administrer ces équipements et d'assurer les tâches d'exploitation récurrentes. L'automatisation des cas d'utilisation décrits ci-dessous représente un gain de temps très important car ils constituent l'essentiel du travail et sont, lors d'un traitement manuel, particulièrement chronophage.

Cependant, en adéquation avec le cahier des charges vu précédemment, certaines fonctionnalités ou scénarios ont été volontairement supprimés et ne feront pas l'objet de travaux d'industrialisation.

En effet, bien que ces fonctionnalités ne soient pas dénuées d'intérêt leur utilisation est considérée comme étant trop marginale pour justifier le développement d'outils censés réduire leur coût d'exploitation.

C'est notamment le cas des opérations liées aux travaux de dé-commissionnement de volumétries qui représentent moins de 15% des demandes adressées à l'équipe VSS ou bien encore les opérations liées aux migrations de données.

### III.4.1.1 Cas d'utilisation

La figure ci-dessous représente les cas d'utilisation associés à l'administration d'un VPLEX dans le contexte d'une exploitation au sein de la société i-BP :

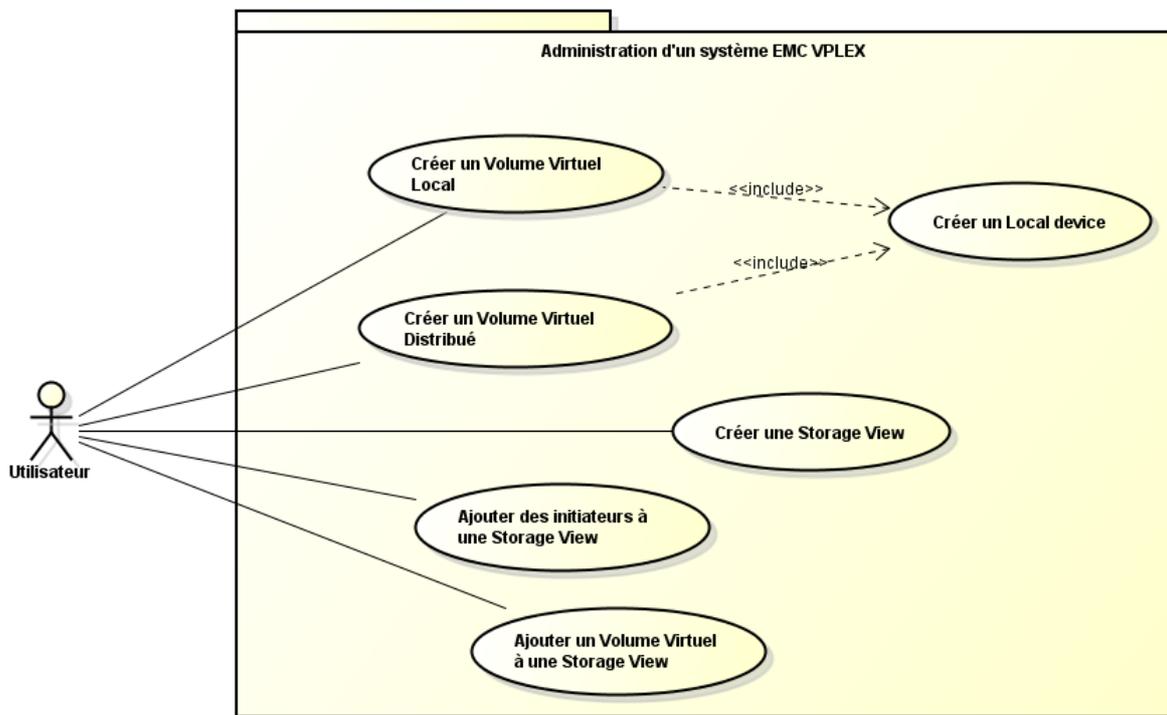


Figure 32 – Cas d'utilisation VPLEX

L'essentiel de ces cas d'utilisation est directement accessible par l'utilisateur de la solution et constitue la modélisation d'opérations d'administration courantes.

Certains de ces cas d'utilisation ne sont pas en contact direct avec l'utilisateur. Ces cas d'utilisation, qualifiés d'internes, ne permettent pas à eux seuls de mener à bien une opération d'administration viable et utile pour l'utilisateur final.

Cependant, ces derniers constituent un élément indispensable à la réalisation de scénarios plus complexes. C'est pourquoi leur factorisation est profitable.

L'administration d'un équipement de type VPLEX a cela de spécifique que cette activité peut s'effectuer à plusieurs niveaux. En effet, comme évoqué précédemment dans les spécifications techniques le concernant, un métro cluster VPLEX est une fusion logique de deux clusters physiques présents en des lieux géographiques distincts.

En résultent, des actes d'administration ayant des portées et impacts variables selon le contexte. Certaines opérations n'agiront que sur un cluster spécifique, d'autres en revanche auront une portée globale et impacteront de manière symétrique les deux clusters du produit.

Cette spécificité se retrouve dans les cas d'utilisation du lot A et apparaîtra plus clairement à l'aide des diagrammes d'activités qui mettent en évidence cette répartition duale.

#### ***III.4.1.1.1 Création de volumes locaux***

La principale fonction d'un équipement de type VPLEX est la création d'espaces de stockage virtualisés, appelés « volumes virtuels ». Ces volumes virtuels peuvent être de type distribué ou local. Cela signifie que le stockage employé pour constituer ce volume peut être situé sur un ou deux sites géographiques.

Un volume virtuel de type local s'appuie donc sur un stockage non répliqué, issu du site où se situe son cluster VPLEX d'appartenance.

Cette solution est bien évidemment non satisfaisante dans le cadre d'une application critique nécessitant une haute disponibilité. Cependant, ces volumes constituent une alternative intéressante et peu coûteuse pour des environnements n'ayant pas ce degré d'exigence.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'environnement associé au cluster VPLEX (production, hors production, etc.)
- L'identifiant unique, « device WWN », à utiliser
- La localisation de ce « device » sous la forme du cluster VPLEX de rattachement

Ce cas d'utilisation prend en charge de manière automatique et transparente les opérations intermédiaires et indispensables à la création d'un volume virtuel : phase de « claim » de la volumétrie, création d'un « Storage Volume », création d'un « Extent », création d'un « Device » pour finir par la création d'un volume virtuel.

L'entité volume virtuel résultante de cette opération est bel et bien locale au cluster de rattachement. La visibilité du volume sur le cluster opposé est rendue possible par l'activation de la fonctionnalité « access anywhere » ou « flag remote ».

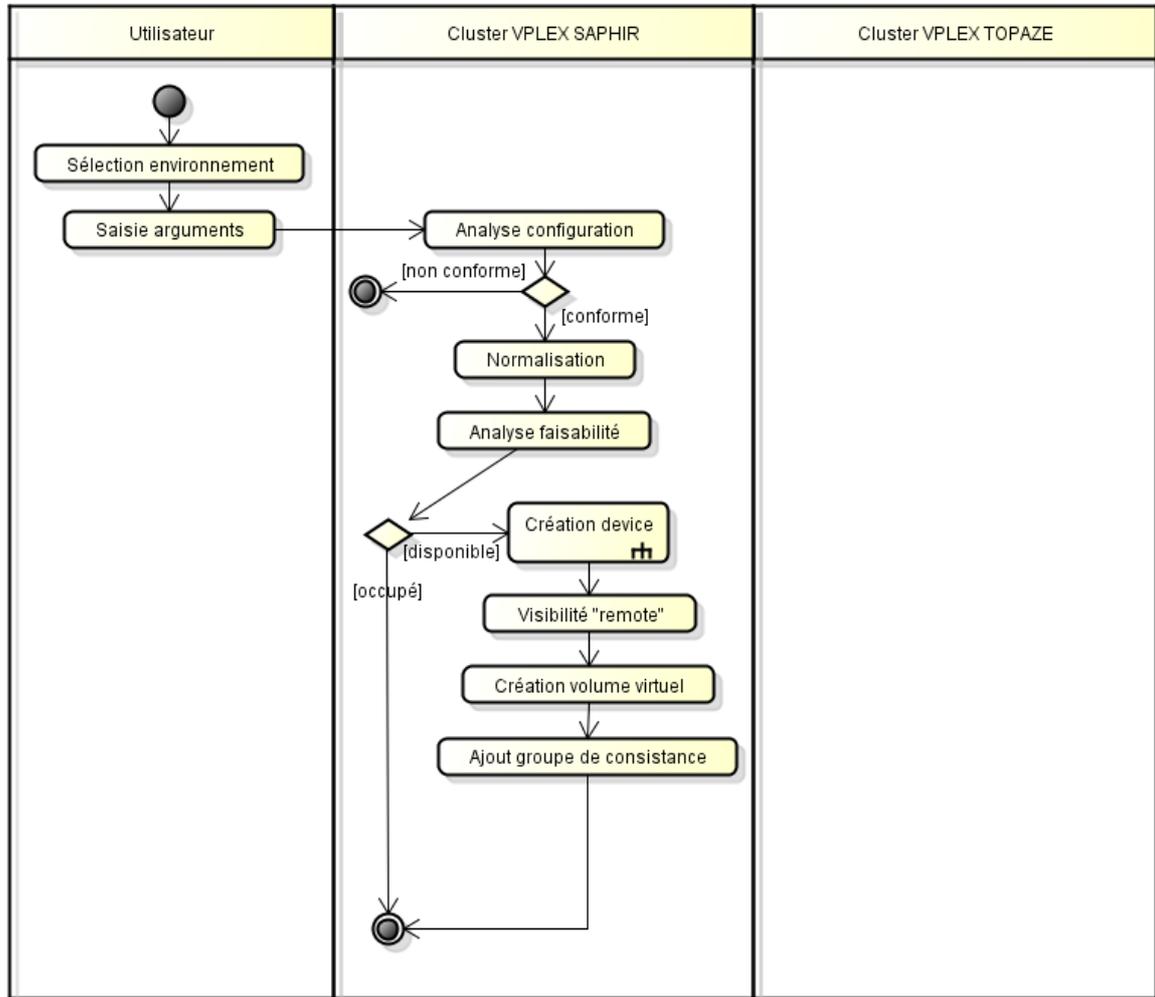


Figure 33 – Diagramme d'activités « Création d'un volume local VPLEX »

La constitution interne du volume virtuel local ainsi créé correspond aux exigences précédemment évoquées : la règle du « 1 pour 1 » (cf. III.3.1.3.1) est respectée.

Le cas d'utilisation prend en charge les étapes relatives à la normalisation évoquée dans le chapitre relatif aux spécifications techniques VPLEX de manière automatique.

Les contrôles de sécurité élémentaires (ex : le volume de données est-il disponible ?) font également partie des étapes indispensables et non facultatives du traitement. En cas

d'anomalie, le traitement s'interrompt immédiatement en communiquant explicitement les raisons de cet échec.

### III.4.1.1.2 Création de volumes distribués

Les volumes distribués représentent une forme particulière de volumes. Ils s'appuient sur la topologie métré cluster du VPLEX pour produire une structure de type « actif/actif ». Ils disposent d'un stockage réparti sur les deux sites géographiques et d'un mécanisme de réplication permettant de gérer la consistance des données.

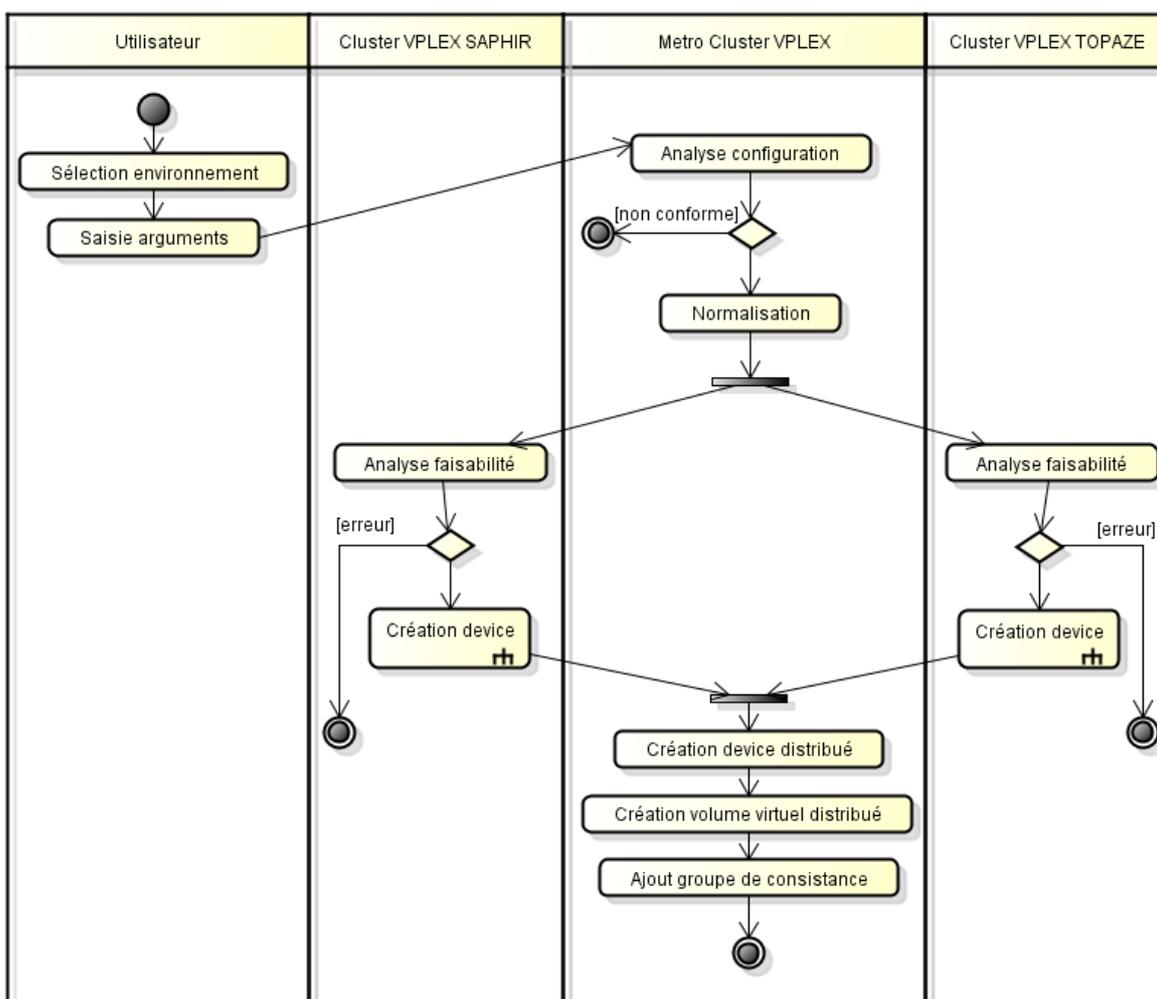


Figure 34 – Diagramme d'activités « Création d'un volume distribué VPLEX »

Ce type de volume est particulièrement indiqué pour les applications nécessitant un stockage performant et résilient : la perte d'un site géographique complet comme, par exemple, le centre de données de SAPHIR est totalement compensé par celui de TOPAZE et le temps d'interruption est nul.

Fonctionnellement, le cas d'utilisation décrivant la création d'un volume distribué ne diffère pas fondamentalement de celui d'un volume local. Il peut même être perçu comme le même cas d'utilisation exécuté deux fois puisqu'il est nécessaire de disposer d'un « device » sur chaque site géographique.

Cependant, le mécanisme de réplication s'appuie pour fonctionner sur un « device » distribué englobant les « devices » conventionnels créés sur chaque cluster VPLEX. Le volume virtuel ainsi créé ne dispose pas d'une visibilité locale mais bel et bien globale.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'environnement associé au cluster VPLEX (production, hors production, etc.)
- L'identifiant unique, « device WWN », à utiliser sur le site de SAPHIR
- L'identifiant unique, « device WWN », à utiliser sur le site de TOPAZE

Le cas d'utilisation prend en charge la création du « device » sur le site de SAPHIR, sur celui de TOPAZE et toutes les étapes intermédiaires jusqu'à la création d'un volume virtuel distribué opérationnel et conforme aux normes i-BP.

A nouveau, en cas d'anomalie lors du traitement, celui-ci s'interrompt en notifiant les raisons précises de cet échec à l'opérateur du script.

#### ***III.4.1.1.3 Création de Storage View***

Les Storage View sont les éléments de configuration permettant de rendre effective la présentation des volumétries contenues dans ces vues aux serveurs qui y sont rattachés.

Les vues disposent d'une visibilité au niveau du cluster VPLEX. Il n'existe pas de vue de type distribuée. Pour une utilisation sur deux clusters, une vue identique doit être créée sur chaque cluster de la solution.

Ce cas d'utilisation prend en charge cette fonctionnalité et rend possible la création d'une vue de manière symétrique sur les clusters VPLEX.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'environnement associé au cluster VPLEX (production, hors production, etc.)
- Du nom des vues à créer sur les deux clusters du VPLEX
- La liste des initiateurs qui seront utilisés en fin de traitement pour peupler ces vues

Le traitement de ce cas d'utilisation prend en charge le contrôle de l'unicité des vues (il ne peut exister au sein d'un même cluster VPLEX deux vues portant le même nom).

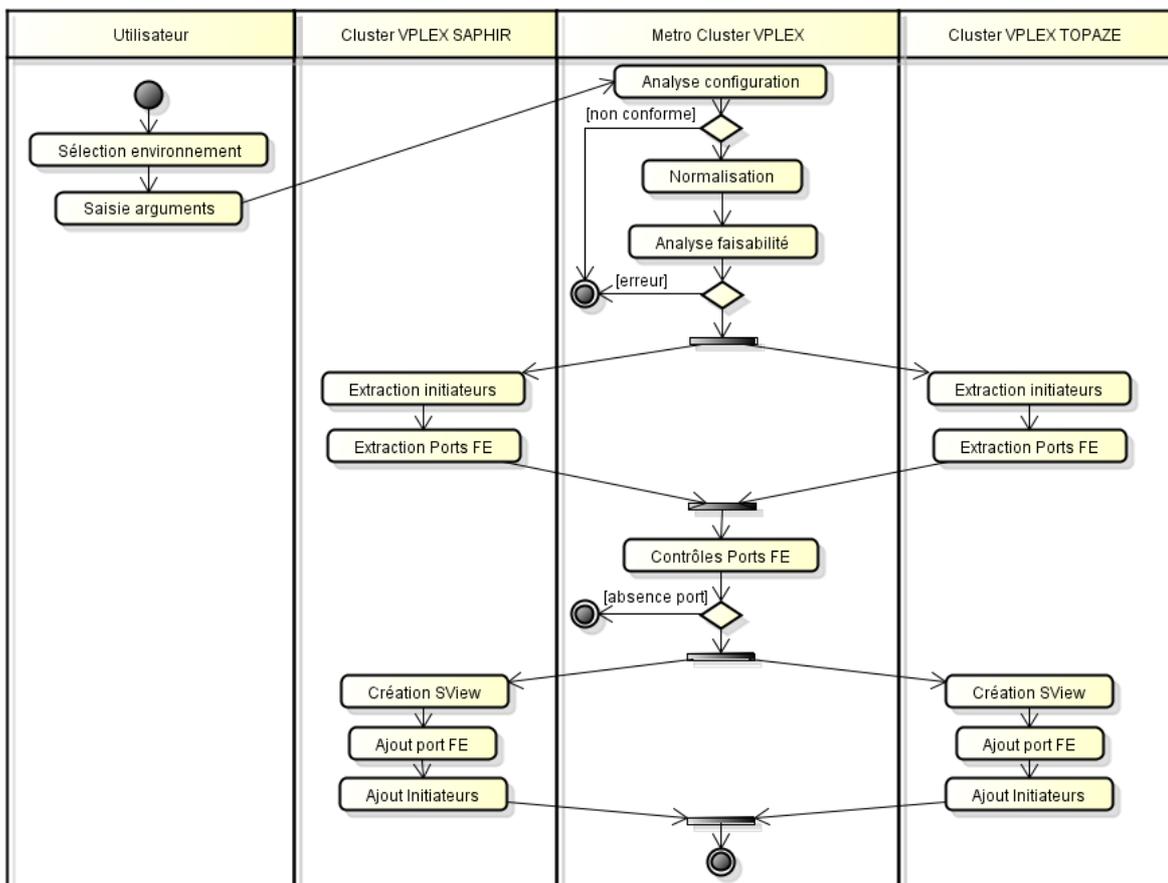


Figure 35 – Diagramme d'activités « Création d'une Storage View VPLEX »

Une vue ne peut être créée sans utiliser un ou plusieurs ports Front-End. Ce cas d'utilisation prévoit l'extraction des ports utilisés par les initiateurs désignés par l'utilisateur. Ces ports seront ensuite employés pour créer la vue, et ce de manière symétrique sur chaque cluster VPLEX.

### III.4.1.1.4 Ajout de volumes à une Storage View existante

L'ajout des volumes virtuels aux « Storage View » représente un des cas d'utilisation les plus importants de la solution, car c'est sans doute celui qui est plus intéressant et plus régulièrement utilisé.

La livraison de la volumétrie encapsulée par la solution VPLEX aux serveurs en ayant besoin est rendue possible par l'ajout de volumes aux vues.

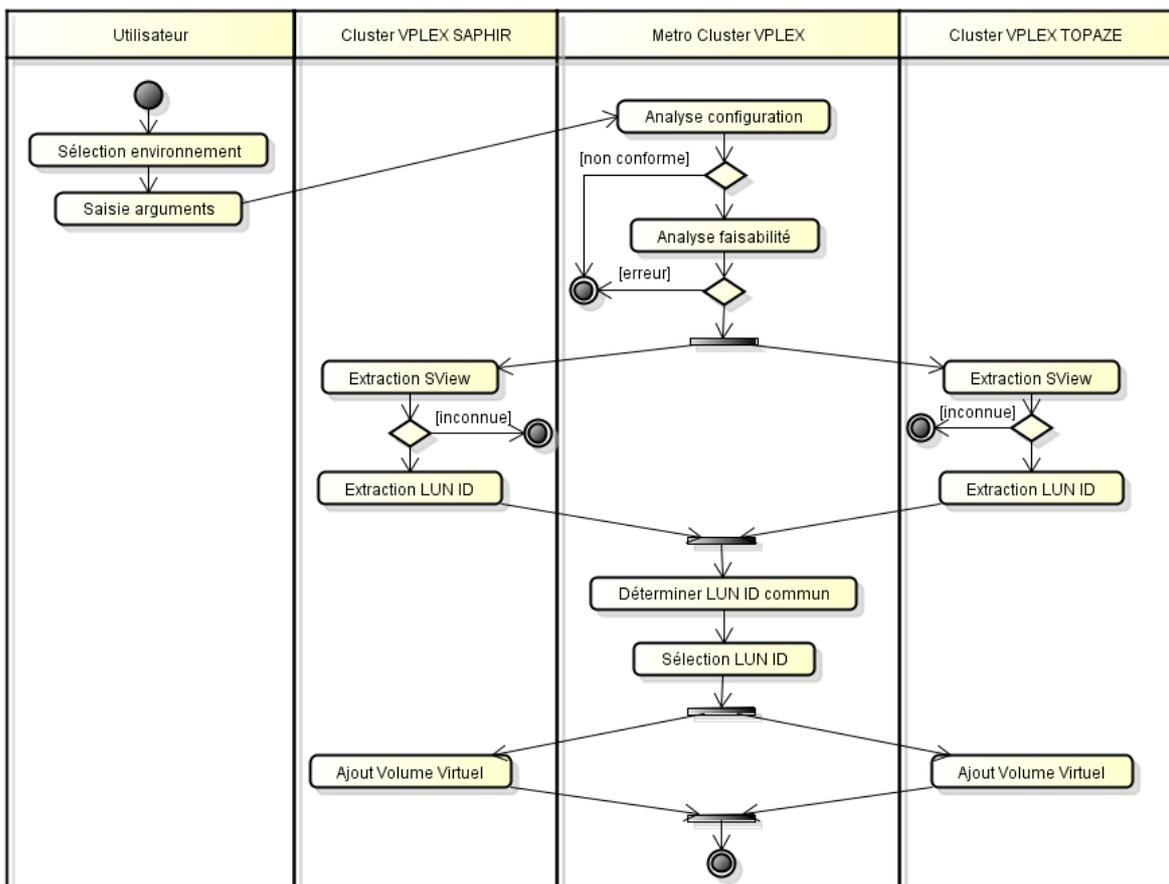


Figure 36 – Diagramme d'activités « Ajout d'un volume virtuel à une vue VPLEX »

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'environnement associé au cluster VPLEX (production, hors production, etc.)
- Du nom de la vue où insérer de la volumétrie
- De la liste des volumes virtuels à utiliser
- En option, la liste des LUN-ID à utiliser pour chaque volume virtuel

La vue où ajouter la volumétrie doit exister au préalable ainsi que les volumes virtuels désignés par l'utilisateur. Cette même contrainte s'applique aux volumes virtuels. Ces vérifications sont effectuées au cours du traitement.

De manière optionnelle l'utilisateur peut choisir de spécifier les LUN-ID qui seront utilisés lors de l'ajout de chaque volume à la vue. Cet identifiant unique fait partie intégrante du protocole SCSI et permet d'identifier de façon sûre et non équivoque chaque composant de la chaîne.

En l'absence de choix effectué par l'utilisateur, le traitement se chargera de déterminer automatiquement les prochains LUN-ID disponibles pour affectation. La sélection devra suivre la contrainte suivante : chaque LUN-ID éligible devra être disponible aussi bien dans la vue du cluster de SAPHIR que dans la vue de TOPAZE. La symétrie des LUN-ID est un élément très important de cohérence au sein des infrastructures de stockage.

En cas d'anomalie ou d'incohérence, le traitement doit s'interrompre avant d'avoir entamé la moindre action. Les contrôles de faisabilité devront s'effectuer en prétraitement, il est en effet très dangereux pour ce type de cas d'utilisation de ne traiter qu'une partie de l'algorithme comme, par exemple, ajouter de la volumétrie sur SAPHIR uniquement sans que cela soit possible sur le site de TOPAZE.

#### ***III.4.1.1.5 Ajout d'initiateurs à une Storage View existante***

A l'instar des volumes virtuels, il est possible d'ajouter un ensemble d'initiateurs à une vue de stockage existante. C'est notamment le cas lorsqu'un cluster ESX reçoit de nouveaux nœuds.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'environnement associé au cluster VPLEX (production, hors production, etc.)
- Du nom de la vue où insérer les nouveaux initiateurs
- De la liste des initiateurs à utiliser

De manière analogue au traitement réalisé lors de la création de la « Storage View », ce cas d'utilisation requiert la connaissance des ports Front-End utilisés par les initiateurs à ajouter à la vue.

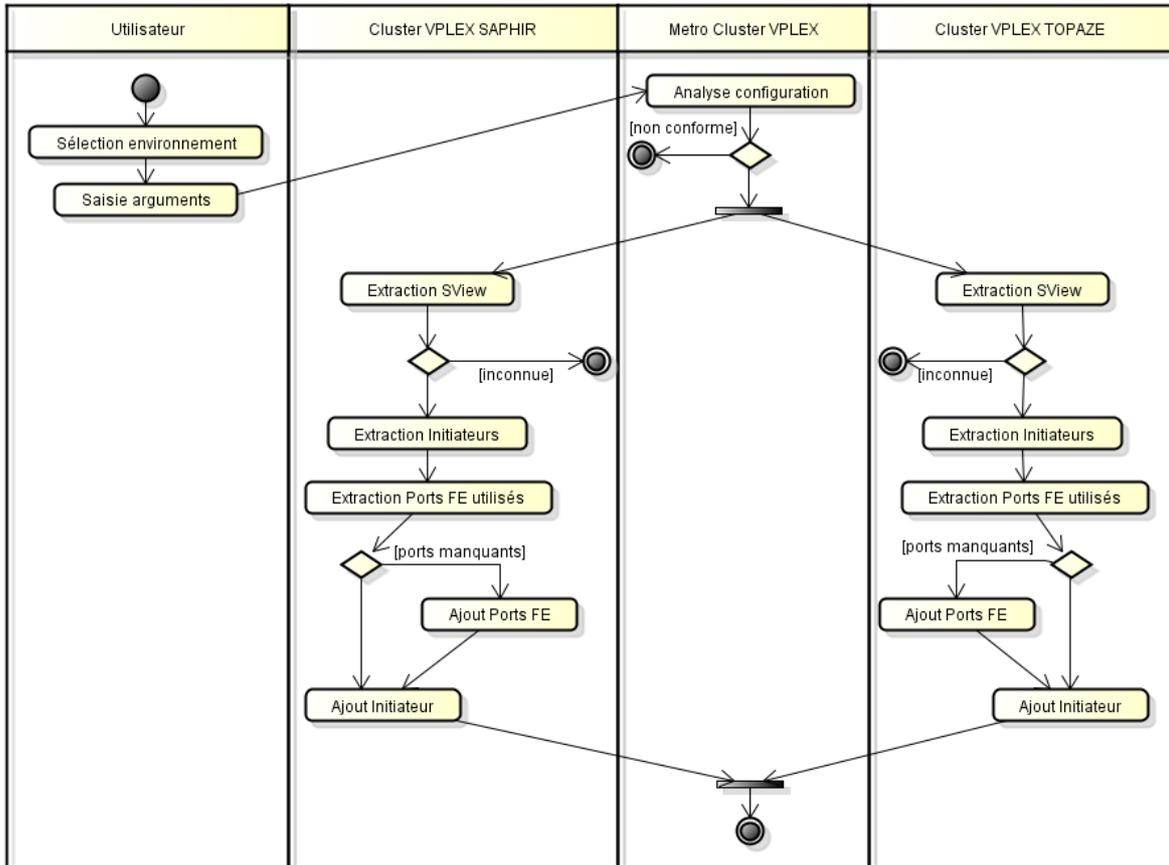


Figure 37 – Diagramme d'activités « Ajout d'un initiateur à une vue VPLEX »

Pour chaque initiateur, les ports « Front-End » utilisés lors des phases de zoning sont donc extraits et comparés à ceux présents au sein de la vue de destination. En cas d'absence, ce cas d'utilisation prévoit l'ajout des ports « Front-End » manquants.

A nouveau, ce traitement est absolument symétrique entre les clusters VPLEX de SAPHIR et de TOPAZE. En fin de traitement le nombre d'initiateurs compris au sein des vues de chaque cluster doit être rigoureusement le même.

## III.4.2 Lot B – Conception des modules d’administration VMAX

### III.4.2.1 Cas d’utilisation

Les cas d’utilisation du lot B permettent de décrire l’exploitation d’une baie VMAX dans le contexte d’une utilisation exclusive avec le produit VPLEX. Ce qui signifie que du point de vue VMAX, le seul client autorisé à accéder à ses ressources sera le cluster VPLEX auquel il est raccordé.

Les baies VMAX peuvent être donc considérées, selon ce schéma, comme des équipements passifs, ne servant qu’à alimenter le VPLEX en volumétrie.

Cette architecture a un impact direct sur la complexité et le nombre des cas d’utilisation VMAX. La présence du VPLEX et de ses fonctionnalités rend inutiles, voire obsolètes, certaines fonctionnalités propres aux baies VMAX.

Pour ces raisons, les cas d’utilisation VMAX sont dans ce contexte limités et peu complexes comme en témoigne la figure ci-dessous :

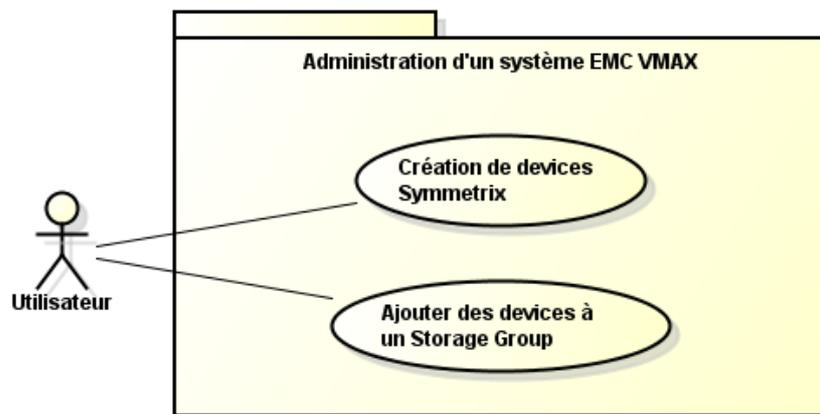


Figure 38 – Cas d’utilisation VMAX

#### III.4.2.1.1 Création de devices Symmetrix

Comme évoqués lors du chapitre III.3.2, les baies VMAX sont les dispositifs qui hébergent de manière réelle et effective les volumétries là où le VPLEX n’assure qu’une fonction d’encapsulation et de virtualisation des ressources fournies depuis les baies VMAX.

En conséquence, c'est sur les baies VMAX que les espaces disques destinés à être présentés au VPLEX pour virtualisation sont créés.

Le présent cas d'utilisation permet de générer toutes les volumétries nécessaires aux besoins de production et d'y appliquer les contraintes de qualité de services souhaités.

L'utilisateur désirant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'identifiant de la baie VMAX concernée
- La capacité du « device » à créer
- Le nombre de « devices » souhaités

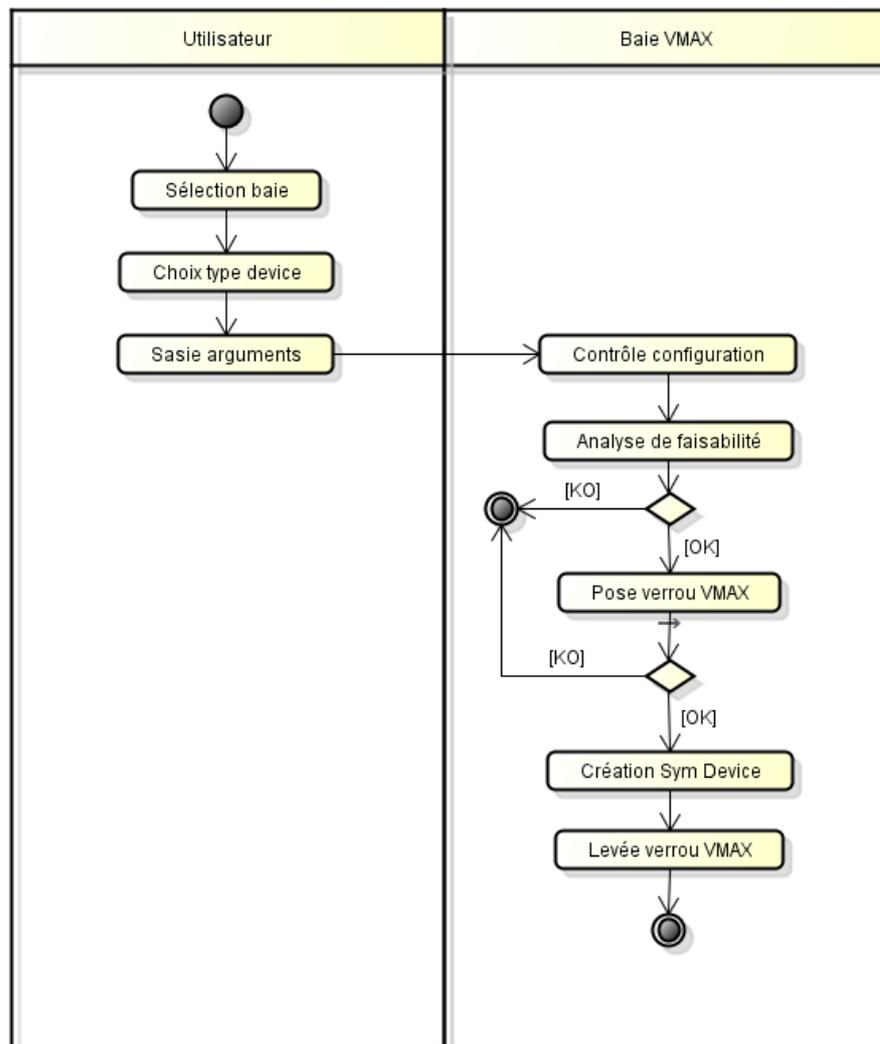


Figure 39 – Diagramme d'activités « Création d'un device VMAX »

Le traitement prend en charge les vérifications permettant de s'assurer de la capacité de la baie de disques à livrer les volumétries demandées. Ces vérifications comprennent l'espace disque disponible au sein des pools de stockage et le contrôle du seuil de sur-allocation.

La création de volumétries étant un mécanisme intrusif et impactant pour la configuration de la baie VMAX ce dernier est encadré par un verrou permettant un accès exclusif à ce processus.

En cas d'anomalie lors du traitement, le verrou est relâché et le message d'erreur en provenance du microcode de la baie fait l'objet d'une communication claire et précise auprès de l'utilisateur des causes de cet échec.

#### ***III.4.2.1.2 Ajout des devices à un Storage Group***

Ce dernier cas d'utilisation rend possible la présentation des volumétries créées depuis le VMAX au VPLEX tout en y appliquant les impératifs de qualité de service requis.

La présentation de volumétrie depuis une baie VMAX vers un équipement externe repose sur les mécanismes « d'auto-provisionnement » abordés au chapitre III.3.2.3.1.

Dans le cadre de ce projet, l'intégralité de la volumétrie VMAX ayant pour vocation à être gérée par le métro cluster VPLEX, les vues de stockage ou « Masking Views » sont préconfigurées lors de l'intégration des baies. Le présent traitement se bornera donc à peupler les groupes adéquats existants.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'identifiant de la baie VMAX concernée
- Le ou les volumes à présenter au VPLEX
- La qualité de service à appliquer aux volumes : « BRONZE/SILVER/GOLD »

Un volume ne pouvant, dans cette topologie, être présenté plus d'une fois, le cas d'utilisation prend en charge les contrôles de sécurité permettant de s'assurer de la disponibilité des volumes à ajouter.

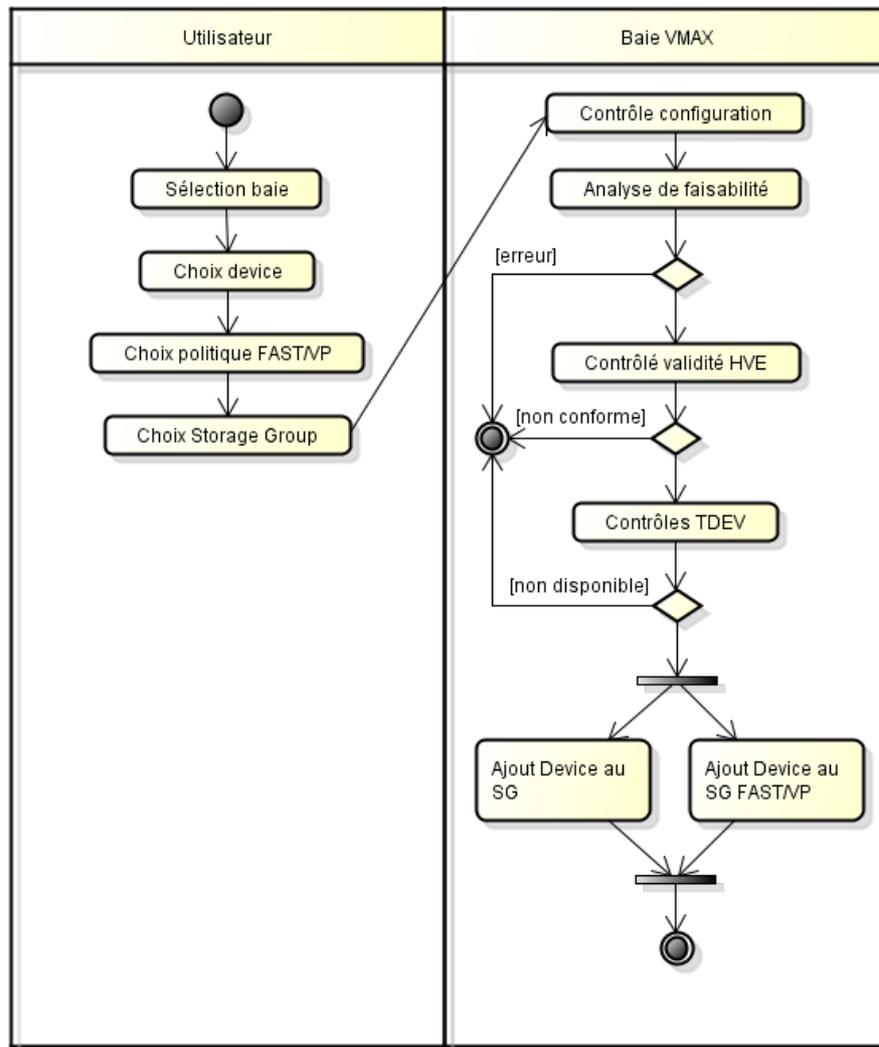


Figure 40 – Diagramme d'activités « Ajout d'un device à un Storage Group VMAX »

Un contrôle de sécurité symétrique sera opéré sur les aspects relatifs à la qualité de service opérée par la baie VMAX au travers de son mécanisme FAST/VP : un volume de données ne peut être assujetti à plus d'une politique FAST/VP.

A l'issue de ce cas d'utilisation, les volumes désignés par l'utilisateur sont présentés de manière effective au cluster VPLEX du site géographique hébergeant la baie VMAX.

### III.4.3 Lot C – Conception des modules d’administration Cisco

#### III.4.3.1 Cas d’utilisation

Les modules d’administration Cisco ont pour objet de faciliter autant que possible l’exploitation de l’infrastructure SAN « Front-End » du projet.

Ce réseau SAN sert à interfacier les métró clusters VPLEX avec le « reste du monde » à savoir les serveurs et autres équipements Fibre Channel déployés au sein des centres de données Banque Populaire de SAPHIR et TOPAZE.

A l’instar des cas d’utilisation du périmètre VPLEX, certains cas d’utilisation du lot C sont internes et n’ont d’utilité que lorsque ces derniers sont employés au sein des cas d’utilisation directement manipulables par l’utilisateur.

La figure ci-dessous liste des cas d’utilisation identifiés pour le lot C :

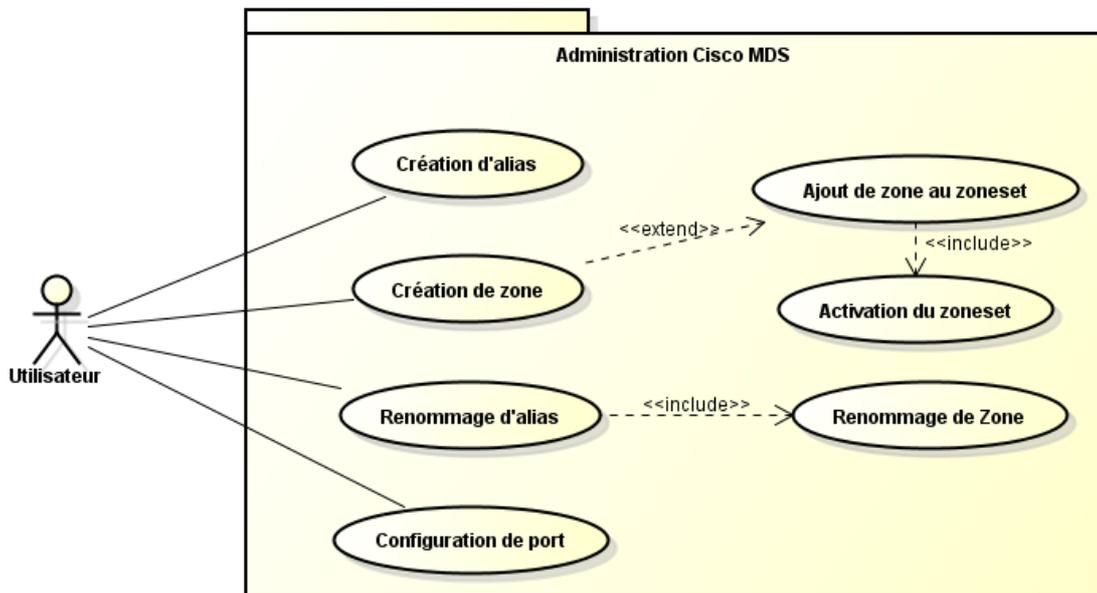


Figure 41 – Cas d’utilisation Cisco MDS

Les cas d’utilisation à destination des directeurs Cisco peuvent agir à deux niveaux distincts. Certains opèrent simplement à l’échelle d’un commutateur spécifique. D’autres opèrent à l’échelle de la « fabric » entière et altèrent donc la configuration globale de cette dernière.

### III.4.3.1.1 Création d'alias

La création d'un alias représente la première étape permettant de zoner un nouvel équipement au sein d'une « fabric » Fibre Channel. Cette opération consiste à appliquer un nom ou une description humainement compréhensible à un ensemble de WWN (les identifiants uniques d'une machine au sein d'un réseau FC).

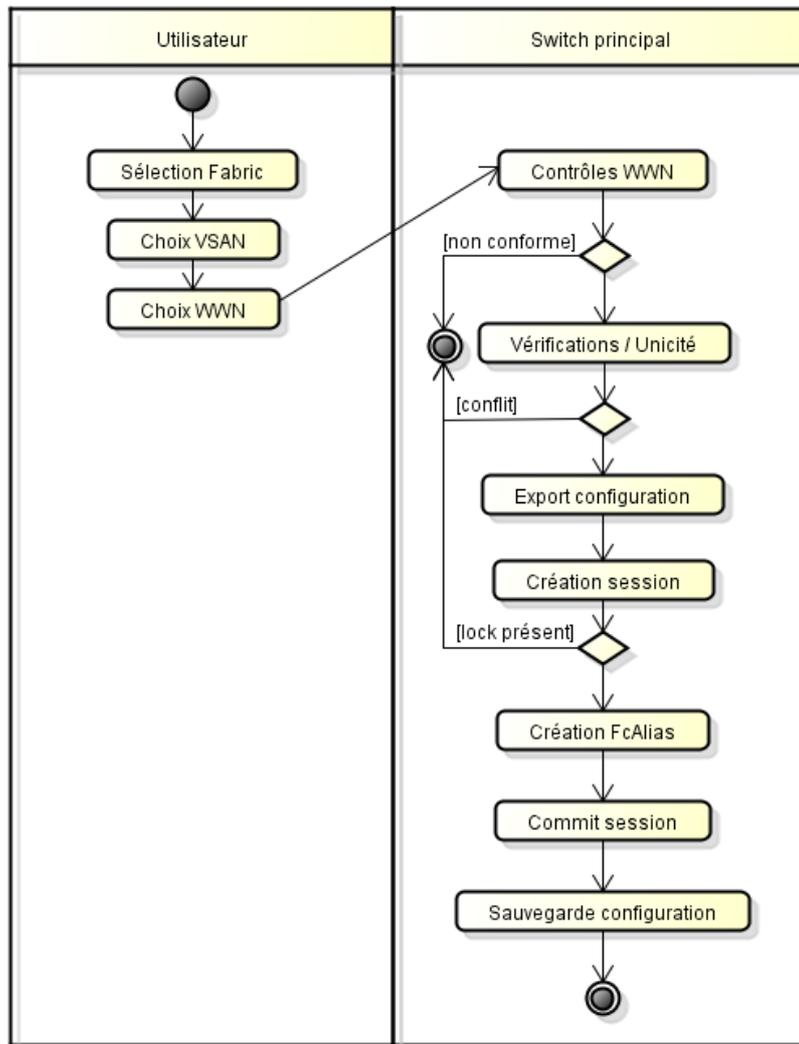


Figure 42 – Diagramme d'activités « Création d'un alias Cisco »

Les directeurs Cisco s'appuyant sur la technologie des VSAN (qui rend possible le découpage d'une infrastructure physique en de nombreuses infrastructures logiques à des fins d'isolation) le cas d'utilisation doit être en mesure de gérer cette spécificité.

La création d'un alias fait partie des opérations ayant un impact global au sein d'une « fabric ». Ce nouvel alias sera inséré au sein du VSAN désigné par l'utilisateur et sa définition sera distribuée auprès de chaque directeur composant cette « fabric ».

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'identifiant de la « fabric » concernée
- Le nom à attribuer à l'alias
- Les WWN devant être inclus dans l'alias
- Le VSAN où procéder à cette création d'alias

En conformité avec les règles d'exploitation, un WWN ne peut être utilisé dans plus d'un alias. De plus, la détection des conflits de nommage fait partie intégrante des mécanismes de sécurité de ce cas d'utilisation.

Le traitement prend également en charge la gestion du verrou exclusif apporté par la fonctionnalité appelée « zoning enhanced Cisco ». Ainsi, la création, l'enregistrement ou le rejet d'une session est traité de manière automatique.

L'enregistrement de la configuration de la « fabric » fait l'objet d'une méthode spécifique permettant de communiquer l'ordre de sauvegarde à tous les directeurs l'a composant, et ce, de manière simultanée.

En fin de traitement, le cas d'utilisation communique auprès de son utilisateur le bon déroulement ou, le cas échéant, les raisons de l'échec de ce dernier.

#### ***III.4.3.1.2Création de zone***

La mise en relation d'une source et d'une cible au sein d'un réseau Fibre Channel est réalisée à l'aide d'objets appelés « zones ». Ce cas d'utilisation permet de créer de nouvelles zones au sein des directeurs Cisco.

La création d'une zone fait partie des opérations ayant un impact global au sein d'une « fabric ». Cette nouvelle zone sera insérée au sein de la configuration, le « zoneset », laquelle sera distribuée auprès de chaque directeur composant la « fabric ».

A l'instar du cas d'utilisation relatif aux alias, les créations de nouvelles zones s'appuient sur le mécanisme d'isolation par VSAN.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'identifiant de la « fabric » concernée
- La liste des alias devant être zonés entre eux
- Le VSAN où procéder à cette création de zone

Les règles d'exploitation actuellement en vigueur n'autorisant pas l'utilisation de plus de deux alias au sein d'une même zone, le traitement inclut un contrôle de sécurité permettant de s'assurer de la conformité de la zone à créer.

Le nom de la zone est la résultante de la concaténation des noms des alias à insérer au sein de cette zone préfixée par « Z1 » ou « Z2 » selon si l'utilisateur opère sur la « fabric » n°1 ou la « fabric » n°2. Cette étape est traitée de manière automatique et sans intervention humaine.

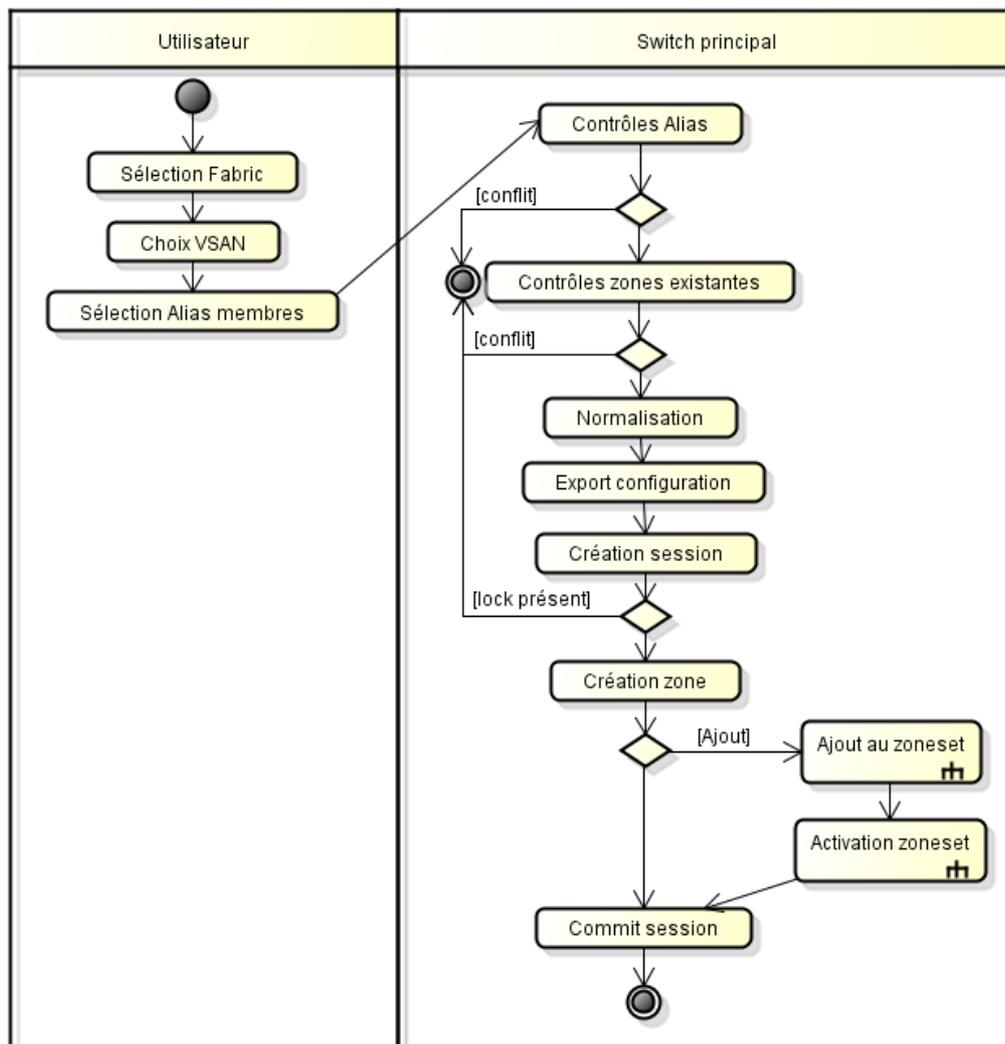


Figure 43 – Diagramme d'activités « Création d'une zone Cisco »

Le traitement prévoit l'appel aux cas d'utilisation internes permettant l'ajout de la zone en cours de création au « zoneset », la configuration de la « fabric », suivi de son activation.

Le traitement prend également en charge la gestion du verrou exclusif apporté par la fonctionnalité appelée « zoning enhanced Cisco ». Ainsi la création, l'enregistrement ou le rejet d'une session est traité de manière automatique.

L'enregistrement de la configuration de la « fabric » fait l'objet d'une méthode spécifique permettant de communiquer l'ordre de sauvegarde à tous les directeurs l'a composant, et ce, de manière simultanée.

En fin de traitement, le cas d'utilisation communique auprès de son utilisateur le bon déroulement ou, le cas échéant, les raisons de l'échec de ce dernier.

### III.4.3.1.3 Renommer un alias

Les activités liées aux changements de noms des équipements et donc des alias les désignant constituent des opérations chronophages et sources d'incohérences suite aux oublis malheureusement trop fréquents.

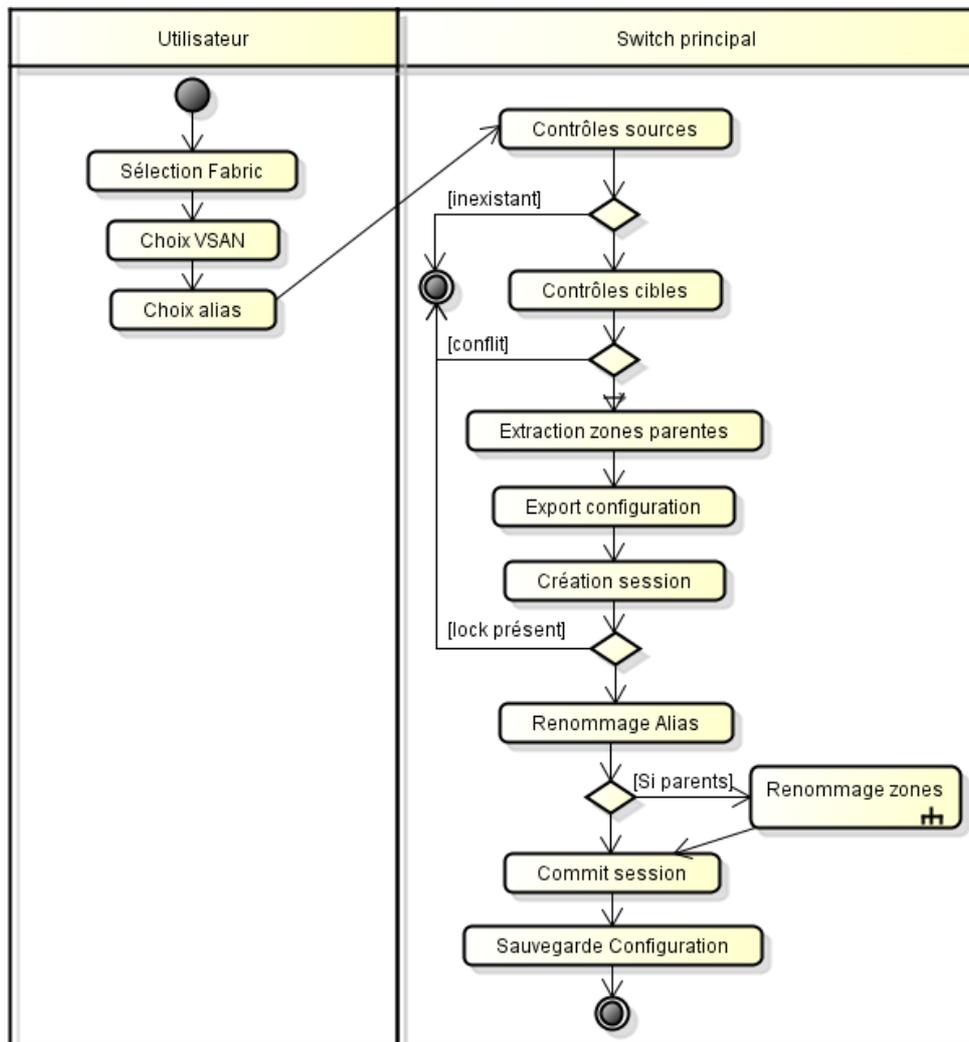


Figure 44 – Diagramme d'activités « Renommer un alias Cisco »

Ce cas d'utilisation permet de changer le nom d'un alias actuellement en production ainsi que le nom des zones pouvant l'utiliser.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- L'identifiant de la « fabric » concernée
- Le nom de l'alias à renommer
- Le nouveau nom de l'alias

- Le VSAN où procéder à cette opération

Le cas d'utilisation prend à sa charge les contrôles de cohérences et d'unicité pour éviter toute collision dans les espaces de nommages.

A l'instar des cas d'utilisation évoqués jusqu'à présent, ce dernier prend également en charge la gestion du verrou exclusif apporté par la fonctionnalité appelée « zoning enhanced Cisco ». Ainsi, la création, l'enregistrement ou le rejet d'une session est traité de manière automatique.

L'enregistrement de la configuration de la « fabric » fait l'objet d'une méthode spécifique permettant de communiquer l'ordre de sauvegarde à tous les directeurs l'a composant, et ce, de manière simultanée.

#### ***III.4.3.1.4 Configuration de Port FC***

Ce cas d'utilisation permet de procéder à la configuration d'un port Fibre Channel, étape préalable indispensable à son utilisation. En effet, par défaut, tous les ports des directeurs composant la « fabric » sont désactivés et sans configuration.

Ce traitement est le seul cas d'utilisation présent dans le lot C à ne pas agir au niveau de la « fabric » mais bien à l'échelle du directeur désigné par l'utilisateur.

L'utilisateur souhaitant avoir recours à ce cas d'utilisation doit disposer des informations suivantes :

- Le nom du directeur Cisco hébergeant le port à configurer
- La désignation du port à configurer
- La description à attribuer au port
- Le VSAN où basculer le port
- En option, la vitesse négociée du port

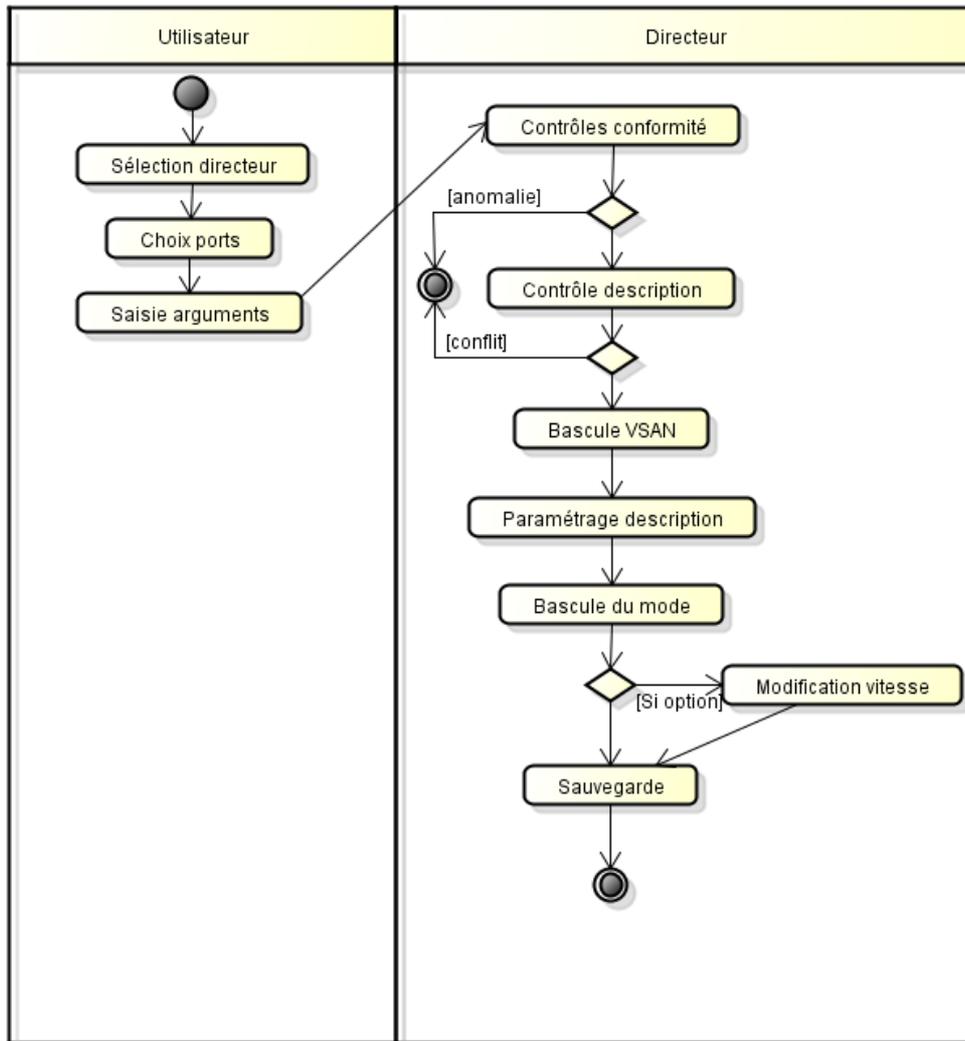


Figure 45 – Diagramme d'activités « Configuration d'un port FC Cisco »

Par défaut, tous les ports non configurés d'un directeur sont désactivés, sans description et appartiennent au VSAN n°1. Le présent cas d'utilisation contient des traitements permettant de contrôler la conformité d'un port aux règles d'exploitation actuellement en vigueur. En cas d'écart, l'utilisateur est notifié des anomalies constatées.

Sur validation des contrôles de conformité, le port à configurer est basculé dans le VSAN de destination, il se voit appliquer une description correspondant à l'équipement qui y est raccordé puis le port est finalement activé.

De manière optionnelle et sur choix de l'utilisateur, la vitesse du port peut être « forcée » pour éviter des aléas liés à l'auto-négociation des capacités.

La sauvegarde de la configuration n'intervient qu'au niveau local, la configuration d'un port n'impactant pas le « zoneset » de la « fabric ».

En fin de traitement, le cas d'utilisation fait état de la réussite ou de l'échec de l'action et en informe l'utilisateur. En cas d'activation réussie, le ou les WWN connectés au port nouvellement configuré sont affichés.

### **III.5 Réalisation de la solution d'automatisation**

La phase de réalisation va s'attacher à mettre en œuvre les spécifications évoquées lors de la phase précédente, c'est-à-dire la phase de conception. Cette section s'articule de la manière suivante :

Tout d'abord, nous verrons les étapes relatives à la mise à disposition de l'environnement de développement et les outils utilisés pour assurer ce développement.

Ensuite, pour chaque lot, nous présenterons le diagramme de classes général de la solution puis nous détaillerons le fonctionnement de chacune de ces classes, leurs rôles dans la solution et les interactions pouvant exister entre elles.

Pour finir, nous aborderons les aspects relatifs à la création des interfaces homme-machine, ou IHM, que les utilisateurs seront amenés à manipuler.

#### **III.5.1 Préparation de l'environnement de développement**

##### **III.5.1.1 Mise à disposition des serveurs**

Pour assurer le développement de cette solution d'administration et d'automatisation des infrastructures de stockage, un ensemble de serveurs est nécessaire pour assurer l'hébergement des divers composants logiciels de l'architecture.

Dans un souci d'isolation, ces serveurs sont dédiés exclusivement à ce projet et comprennent :

- Une machine de développement, nommée IB01163V, issue de l'environnement de développement à la demande. Cette machine dispose de tous les outils nécessaires au développement (IDE, interpréteur Python, etc.) et fonctionne sous Microsoft Windows 7. Ce serveur est une machine virtuelle
- Un serveur de maquette et d'intégration, nommé IBUNSE11, où sont mis en validation les modules fraîchement développés. Cette machine héberge également le système de gestion de versions du projet. Ce serveur est une machine virtuelle qui utilise RedHat Linux RHEL6
- Un serveur de production, nommé IBUNSE10, utilisé lors de l'exploitation quotidienne. Seules les versions stabilisées du code sont installées sur cette machine. Ce serveur est une machine virtuelle qui utilise RedHat Linux RHEL 6

### III.5.1.2 Accès et habilitations

L'accès aux machines virtuelles Linux s'effectue au travers de connexions de type SSH via l'interface réseau d'administration de ces serveurs.

Ces serveurs utilisent le démon SSSD Linux pour assurer l'interconnexion avec le domaine Active Directory de la société. Il est ainsi possible à tous les utilisateurs du groupe « VSS Stockage » de se connecter à ces machines tout en disposant du niveau d'habilitation adéquat.

L'accès à la machine de développement, qui est une machine virtuelle utilisant Microsoft Windows 7 pour assurer son fonctionnement, s'effectue au travers d'un portail Citrix XenApp.

### III.5.1.3 Ouvertures de flux

Les serveurs étant situés dans des réseaux différents, isolés entre eux par des firewalls pour des raisons de sécurité évidentes, des ouvertures de flux sont nécessaires.

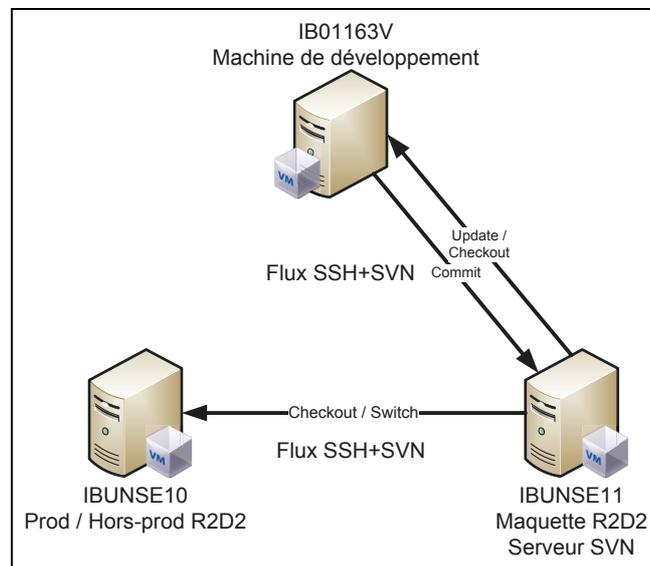


Figure 46 – Ouvertures de flux nécessaires entre les serveurs de la solution

Les échanges entre les serveurs se limitant à des flux de type SSH, seul ce protocole est autorisé à transiter entre ces machines. Le système de gestion de versions est également configuré pour que ses échanges soient transportés à l'aide de ce protocole.

### III.5.1.4 Choix de l'environnement de développement intégré (EDI)

Bien que le langage Python soit réputé simple d'emploi et qu'il soit possible d'assurer un développement à l'aide d'un simple éditeur de texte, il ne saurait être envisageable pour un développement plus ambitieux de s'abstenir d'utiliser un environnement de développement intégré ou EDI. L'EDI sélectionné pour ce projet est « Pycharm » édité par la société JetBrains.

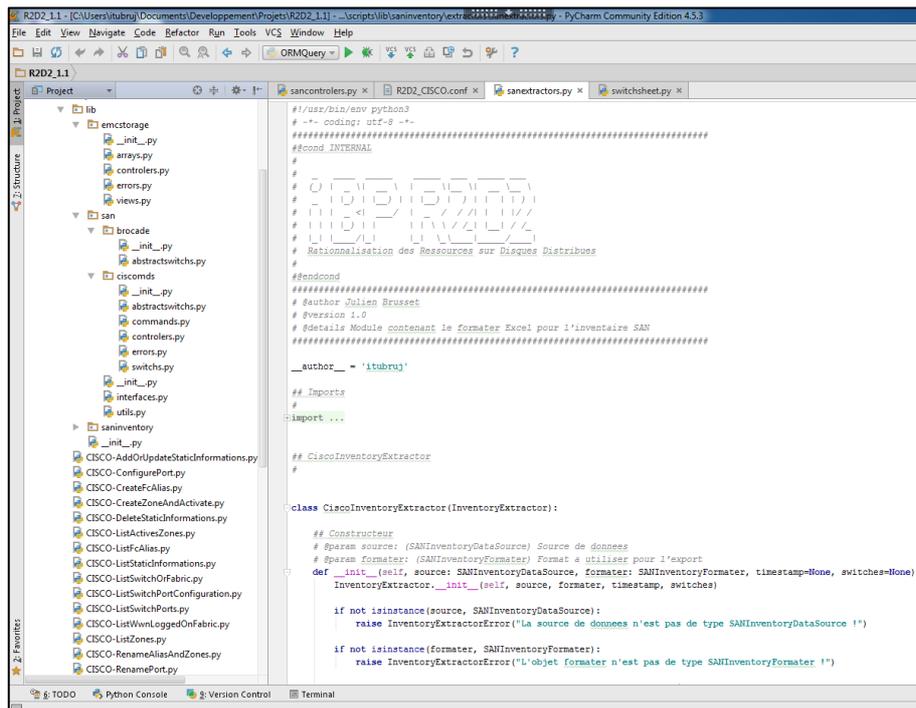


Figure 47 – L'environnement de développement intégré Pycharm

Ce dernier compte de nombreux atouts qui font de lui une solution de choix pour ce projet :

- Gestion native du langage Python (syntaxe, particularités, etc.)
- Support des principaux systèmes de gestion de versions du marché
- Débogueur intégré et performant
- Gestion des environnements de développement virtuels
- Installation et maintenance des modules tiers

### III.5.1.5 Serveur de gestion de versions

La gestion dans le temps du code source d'un projet requiert l'utilisation d'un logiciel dévolu à cette tâche et que l'on nomme un gestionnaire de versions. Dans le cadre de ce projet, c'est le logiciel Subversion qui assure cette fonction.

Subversion est un gestionnaire de versions centralisé s'appuyant sur un dépôt unique. Ce dépôt est organisé de la manière suivante :

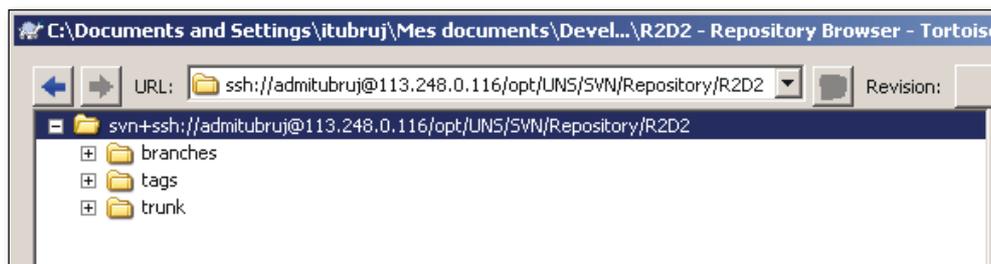


Figure 48 – Architecture du dépôt Subversion du projet

Le dépôt adopte une approche classique dans son organisation. On sépare le tronc principal, les branches qui servent à isoler les développements spécifiques, et les étiquettes qui permettent de « figer dans le temps » une version donnée en prévision d'une mise en production prochaine.

### III.5.1.6 Organisation du projet

Afin d'assurer une bonne maintenabilité ainsi qu'une meilleure lisibilité de l'ensemble du code source, il est important de structurer ce dernier de manière cohérente. La figure ci-dessous reprend l'organisation générale du code source :

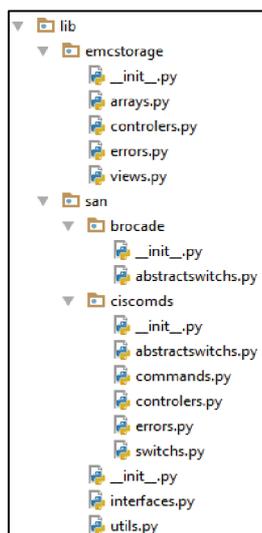


Figure 49 – Organisation des paquetages du projet

L'arborescence proposée permet d'isoler les divers modules développés dans le cadre du projet afin d'accroître la réutilisabilité de ces derniers. Ainsi, les paquetages relatifs à

l'administration du périmètre EMC<sup>2</sup> sont isolés de ceux intervenant au niveau du réseau SAN Cisco.

Afin de faciliter les développements ultérieurs, les comportements métiers communs ont été factorisés au sein d'interfaces et l'arborescence du code source permet d'ores et déjà l'ajout de futurs modules.

### **III.5.1.7 Générateur de documentation**

Une bonne documentation des API est un point crucial pour tout projet développement. C'est la garantie de faciliter les développements futurs en permettant aux développeurs d'obtenir une meilleure compréhension de l'ensemble.

Malheureusement, maintenir ce type de documentation peut s'avérer fastidieux. C'est pourquoi il est très utile de s'appuyer sur un générateur de documentation qui va créer et surtout de maintenir à jour les documents relatifs aux API en s'appuyant sur les commentaires présents dans le code source.

Pour assurer cette tâche, le logiciel Doxygen a été sélectionné. Il permet de créer des documentations aux formats HTML ou LateX. Il est également possible de générer des « man pages » exploitables sur tous les systèmes Unix.

### III.5.2 Réalisation de l'interface homme-machine

L'interface homme-machine, comme son nom l'indique, représente l'ensemble des moyens permettant à un être humain d'interagir avec les cas d'utilisation modélisés par la présente solution.

Comme évoqué dans le cahier des charges, la société i-BP souhaite que cette IHM prenne la forme d'une interface utilisable en ligne de commande. En effet, les outils d'administration utilisés quotidiennement par les équipes d'administration revêtent traditionnellement cette apparence, car ils permettent de traiter plus rapidement les demandes en « rejouant » les jeux de commandes précédemment utilisés.

De plus, ce type d'interface peut aisément prendre place au sein d'un traitement « batch » exécuté via un ordonnanceur de tâches sans qu'aucune interaction humaine ne soit nécessaire.

#### III.5.2.1 Diagramme de classes

La figure ci-dessous présente les différentes classes contenues dans le paquetage « views » ainsi que les relations qui lient ces dernières :

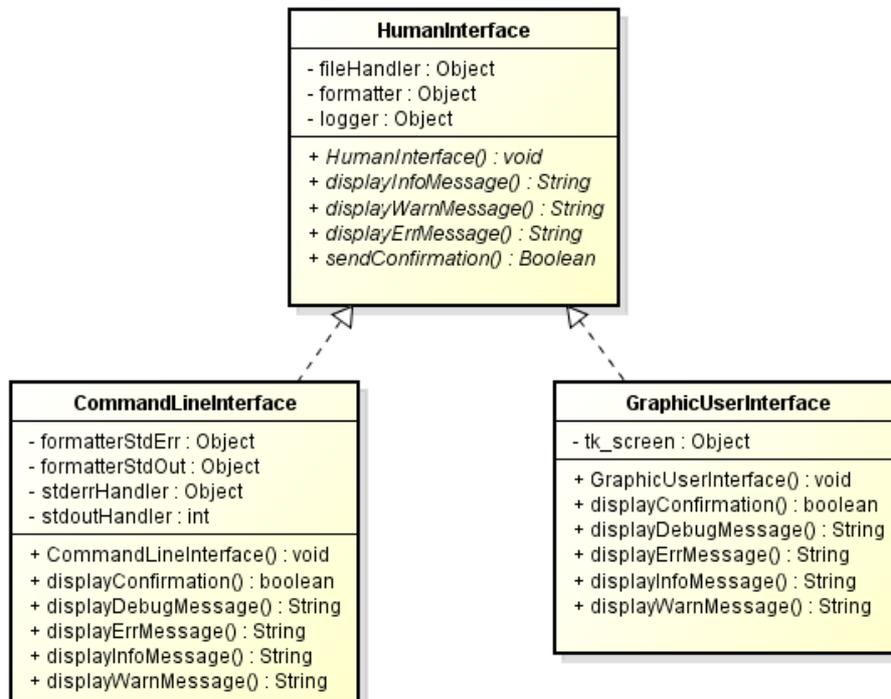


Figure 50 – Diagramme de classes du module « views »

### ***III.5.2.1.1 L'interface `HumanInterface`***

L'interface « `HumanInterface` » permet de fixer les comportements attendus des modules de type IHM. Cette interface déclare les méthodes et les attributs devant être surchargés dans les classes concrètes.

L'utilisation d'une interface permet d'utiliser les propriétés de polymorphisme du langage Python et rend le développement de nouvelles IHM plus aisé en permettant d'interchanger celles-ci sans aucun impact sur les autres classes du modèle.

### ***III.5.2.1.2 La classe `CommandLineInterface`***

Classe principale de ce paquetage, la classe « `CommandLineInterface` » fournit comme son nom se laisse deviner les méthodes nécessaires à l'interaction homme-machine en mode ligne de commande.

La journalisation des activités liées à l'administration des infrastructures de stockage étant un point essentiel, cette classe prend en charge la journalisation systématique de tous les flux échangés entre l'application et son utilisateur.

Les échanges en question sont stockés sous forme de fichiers journal contenant toutes les informations nécessaires pour permettre la traçabilité des actions ainsi que les activités liées au débogage en cas d'incident.

La classe « `CommandLineInterface` » repose en grande partie sur le module « `logger` » de la bibliothèque standard Python. Ce module permet de manipuler les flux d'E/S en les catégorisant selon la criticité de l'information véhiculée. Quatre catégories existent par défaut : « `INFO` », « `WARNING` », « `CRITICAL` », « `DEBUG` ».

La création de « `handlers` » permet de configurer les flux E/S et d'affecter un comportement différent à chacun d'entre eux selon leur catégorie d'appartenance. Ainsi, il est envisageable d'ajouter des informations pour un flux destiné à être consigné dans un fichier de journal, comme un horodatage par exemple, sans affecter le fonctionnement global de la méthode à la manière d'un décorateur.

### ***III.5.2.1.3 La classe `GraphicUserInterface`***

Le langage Python fournit, dans sa bibliothèque standard, les outils permettant de créer des IHM de type interface graphique ou GUI (Graphic User Interface). Le paquetage le plus connu pour cela est « *tkinter* ».

La classe « *GraphicUserInterface* » rend possible l'utilisation des méthodes définies au sein de l'interface « *HumanInterface* » dans un contexte GUI via l'utilisation du module Python standard « *tkinter* ».

L'interface homme-machine principale utilisée par le présent projet étant une interface de type « commande en ligne » (pour les raisons évoquées ci-dessus) aucun cas d'utilisation n'aura recours, dans un premier temps, au service de la classe « *GraphicUserInterface* ».

Néanmoins, cette possibilité est laissée à titre d'option et permettra de faciliter de futurs développements nécessitant l'usage d'une interface graphique.

### III.5.3 Réalisation du lot A – VPLEX

#### III.5.3.1 Diagramme de classes

La figure ci-dessous présente les différentes classes contenues dans le paquetage VPLEX ainsi que les relations qui existent entre elles :

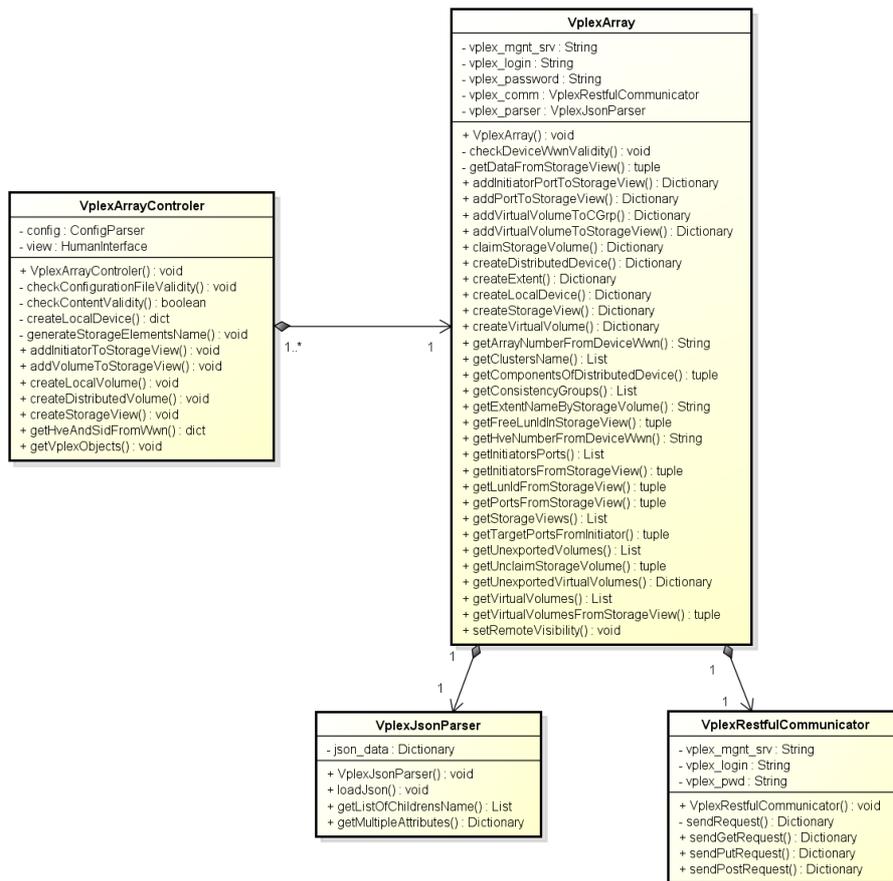


Figure 51 – Diagramme de classes du module VPLEX

#### III.5.3.1.1 La classe *VplexRestfulCommunicator*

Comme évoqué précédemment dans le chapitre traitant des spécifications techniques VPLEX, les échanges de données avec un équipement de cette famille s’effectuent par le biais de requêtes de type REST.

La classe « *VplexRestfulCommunicator* » prend à son compte la matérialisation de ces échanges en servant d’intermédiaire entre le VPLEX et les autres classes du modèle.

A l'aide d'identifiants spécifiés à la création de l'objet, l'instance de cette classe est en mesure de se connecter au serveur de management VPLEX et de lui transmettre des requêtes REST à l'aide des méthodes :

- `sendGetRequest()` : pour les requêtes de type HTTP/GET
- `sendPutRequest()` : pour les requêtes de type HTTP/PUT
- `sendPostRequest()` : pour les requêtes de type HTTP/POST

En retour à ces sollicitations, le VPLEX sérialise les données au format JSON. La classe « *VplexRestfulCommunicator* » se charge alors de retourner ce flux aux objets appelants pour analyse et traitement.

#### ***III.5.3.1.2 La classe VplexJSONParser***

Cette classe a pour but d'analyser les flux de données retournés par les requêtes REST VPLEX pour en extraire les informations utiles au traitement.

L'utilisateur de cette classe peut soumettre un flux JSON assorti d'une liste d'attributs à rechercher dans ce flux. Charge alors à la classe « *VplexJSONParser* » de parcourir la structure JSON et d'en extraire les données recherchées.

Ces dernières sont ensuite insérées au sein d'une structure de données standard afin d'en faciliter le traitement par les autres classes de la solution.

#### ***III.5.3.1.3 La classe VplexArray***

La classe « *VplexArray* » a pour vocation de modéliser un cluster VPLEX en tant qu'objet autonome. Elle simplifie les échanges entre un VPLEX et son utilisateur en faisant abstraction de la complexité sous-jacente à ces échanges.

Cette classe contient toutes les méthodes nécessaires à la réalisation des opérations d'administration courantes dans le cadre de l'exploitation d'une solution VPLEX en production.

Pour assurer son fonctionnement, la classe « *VplexArray* » s'appuie sur les classes « *VplexRestfulCommunicator* » et « *VplexJSONParser* » afin de déléguer les tâches liées à

l'échange de données avec le cluster VPLEX et à l'extraction des données nécessaires au traitement.

Chaque méthode de la classe « *VplexArray* » contient la logique métier nécessaire et suffisante à la sollicitation du cluster VPLEX, à l'extraction, puis à l'interprétation des données issues de cette sollicitation.

Deux catégories de méthodes se dégagent de cette classe :

- Les méthodes d'interrogation
- Les méthodes de configuration

Les premières permettent d'extraire des données du VPLEX afin d'obtenir des informations sur l'état de ce dernier. Ces méthodes sont particulièrement utiles pour implémenter des contrôles de sécurité en vérifiant tel ou tel point de configuration du VPLEX.

Les secondes permettent, comme le nom le laisse supposer, d'effectuer des actes de configuration sur le cluster VPLEX correspondant à l'instance de cette classe.

#### ***III.5.3.1.4 La classe *VplexArrayControler****

Cette classe assure le travail d'orchestration au sein du modèle. Elle contient la logique métier nécessaire et suffisante au traitement de scénarios structurés.

Ici, le terme contrôleur fait écho au composant du même nom employé dans les solutions de type SDS évoquées dans la section état de l'art du présent document. Ce composant joue le rôle de chef d'orchestre en pilotant les ressources qui sont mises à sa disposition.

Cette classe s'appuie sur la classe « *VplexArray* » pour assurer son fonctionnement et analyse le fichier de configuration qui lui est transmis afin d'en extraire les informations permettant l'instanciation des objets de type « *VplexArray* ». Ces informations comprennent les adresses IP des serveurs de management ainsi que les identifiants permettant de s'y connecter.

La classe « *VplexArrayControler* » effectue également les opérations relatives à la journalisation des opérations d'administration.

### III.5.4 Réalisation du lot B – VMAX

#### III.5.4.1 Diagramme de classes

La figure ci-dessous présente les différentes classes contenues dans le paquetage VMAX ainsi que les relations qui existent entre elles :

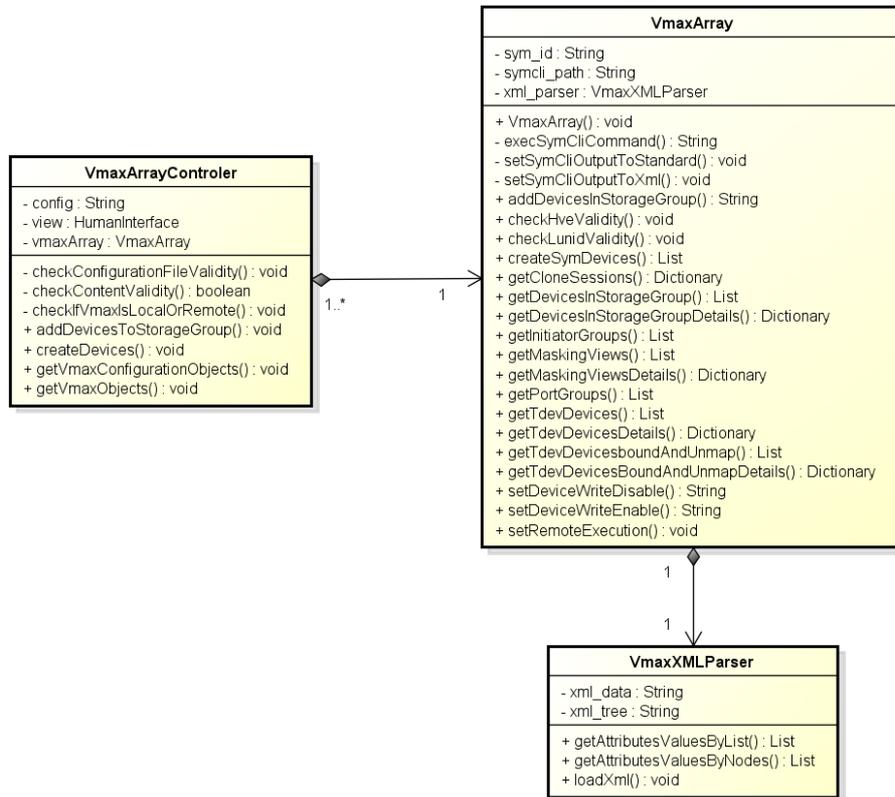


Figure 52 – Diagramme de classes du module VMAX

##### III.5.4.1.1 La classe *VmaxXMLParser*

Comme évoqué dans le chapitre III.3.2.5.3, les API implémentées au sein des baies de disques VMAX reposent sur l'utilisation d'un jeu de commandes en ligne, appelé « Solutions Enabler », ainsi que sur des échanges de flux XML.

La classe « *VmaxXMLParser* » prend à sa charge l'analyse de ces flux XML afin d'en extraire les éléments utiles au traitement. Pour assurer son fonctionnement, elle s'appuie sur le module standard Python « ElementTree » qui permet de parcourir les structures de données arborescentes.

Grâce aux méthodes qu'elle propose, cette classe fournit à son utilisateur le moyen de spécifier les attributs recherchés et de recevoir les valeurs sous forme d'une structure de données Python standard.

#### ***III.5.4.1.2 La classe *VmaxArray****

La classe « *VmaxArray* » modélise une baie de disques VMAX en tant qu'objet autonome et fournit un ensemble de méthodes qui rendent possible son administration.

Ces méthodes contiennent la connaissance métier nécessaire et suffisante à l'envoi de requêtes à la baie VMAX au travers de son jeu de commandes « Solutions Enabler » puis s'appuient sur la classe « *VmaxXMLParser* » pour assurer l'interprétation des flux qui lui sont renvoyés.

Tout comme pour le modèle VPLEX, deux catégories de méthodes se dégagent de cette classe :

- Les méthodes d'interrogation
- Les méthodes de configuration

Les méthodes d'interrogation ont pour vocation l'extraction de données de la baie VMAX. Les données extraites sont ensuite employées comme aide à la décision. Ces méthodes constituent la base des mécanismes de contrôle de sécurité implémentés dans le modèle.

Les méthodes de configuration permettent d'altérer la configuration de la baie VMAX et d'en assurer l'administration.

#### ***III.5.4.1.3 La classe *VmaxArrayControler****

A l'instar de la classe contrôleur évoquée dans la section VPLEX, la classe « *VmayArrayContoler* » assure le rôle de module d'orchestration au sein du modèle VMAX.

Cette dernière présente au travers de ses méthodes un ensemble de scénarios complets et structurés qui représentent la réalisation concrète des cas d'utilisation évoqués lors de la phase de conception.

Cette classe assure également le rôle d'analyseur syntaxique du fichier de configuration. Ce fichier lui est transmis lors de l'instanciation et contient tous les éléments externes nécessaires au traitement comme les numéros de série des baies, où stocker les fichiers de journalisation, etc.

La classe « *VmaxArrayControler* » instancie des objets de type « *VmaxArray* » afin d'y transférer ses ordres d'administration selon la séquence définie dans les scénarios contenus dans ses méthodes.



La classe utilitaire « *WWN* » a pour but de faciliter la manipulation de ces adresses en factorisant les actions s’y rapportant, comme notamment les motifs de reconnaissance ou les opérations de comparaison.

#### ***III.5.5.1.2 L’interface CiscoCommand et ses réalisations***

L’interface « *CiscoCommand* » a pour but de définir le comportement commun des opérations de configuration courantes au sein du modèle Cisco.

Cette interface déclare un ensemble de méthodes et d’attributs qui sont disponibles pour toutes les opérations de configuration prévues dans le paquetage Cisco et qui permettent ainsi d’utiliser les capacités de polymorphisme du langage Python.

Cette interface est utilisée par trois réalisations concrètes qui sont :

- La classe « *CiscoFcAlias* » permettant la déclaration et la création d’alias
- La classe « *CiscoZone* » permettant la déclaration et la création de nouvelles zones
- La classe « *CiscoFcPort* » qui permet la configuration des ports FC d’un directeur

En plus de ces méthodes et attributs communs, les classes évoquées ci-dessus embarquent des méthodes et attributs propres à leur traitement.

#### ***III.5.5.1.3 L’interface ISANSwitch***

L’interface « *ISANSwitch* » est l’interface racine de ce modèle. Elle permet de définir le comportement minimal attendu d’un commutateur SAN pour qu’il puisse être considéré comme tel.

Cette abstraction est volontairement très succincte afin qu’elle puisse correspondre à des commutateurs qui ne sont pas de marque Cisco et qui, bien que le présent projet ne prévoit pas d’en inclure dans son périmètre, seraient amenés à être utilisés dans un futur proche. C’est le cas notamment des commutateurs de marque Brocade.

Cette interface s’attache donc à décrire les comportements essentiels de ces équipements.

#### **III.5.5.1.4 Les classes abstraites *AbstractCiscoSanSwitch* et *AbstractBrocadeSanSwitch***

La classe « *AbstractCiscoSanSwitch* » regroupe toutes les méthodes et attributs communs aux commutateurs de marque Cisco quel que soit leur rôle au sein du réseau Fibre Channel (commutateur de cœur, de frontière, embarqué, etc.).

Cependant, ce corpus de méthodes ne suffit pas à modéliser un commutateur dans son intégralité. Certains aspects, spécifiques au positionnement du commutateur dans le réseau, doivent compléter ce socle commun.

C'est pour cette raison que la présente classe est déclarée comme étant abstraite afin d'en empêcher l'instanciation directe.

La classe abstraite « *AbstractBrocadeSanSwitch* » quant à elle n'est, pour l'heure, qu'une coquille vide incluse dans le projet afin de faciliter les développements à venir dans l'éventualité de la mise en production de commutateurs de marque Brocade.

#### **III.5.5.1.5 La classe *CiscoNPIVSanSwitch***

Cette classe concrète s'appuie sur les bases définies par la classe abstraite « *AbstractCiscoSanSwitch* » pour produire un objet autonome capable d'administrer un commutateur réel dans sa totalité.

Cette classe ambitionne plus particulièrement de manipuler des commutateurs de cœur ou de frontière soit des commutateurs constituant l'épine dorsale d'une architecture de réseau SAN.

C'est sur ce type d'équipement que les opérations liées à la configuration de la « fabric » peuvent prendre place. Les créations d'objets tels que les alias, les zones pour ne citer que ces exemples doivent être effectués sur ce type de commutateurs, car ce sont les seuls à être en mesure d'altérer la configuration globale de la « fabric ».

Cette classe implémente donc un ensemble de méthodes propres à ces responsabilités ainsi que des contrôles de sécurité permettant de s'assurer du rôle réel effectué par le commutateur au sein du réseau.

### ***III.5.5.1.6 La classe CiscoNPVSanSwitch***

Certains commutateurs utilisés au sein d'infrastructures virtuelles disposent de fonctionnalités particulièrement réduites et n'exercent, tout au plus, qu'un rôle de routage des trames Fibre Channel vers le cœur de réseau SAN. C'est le cas des commutateurs en mode NPV.

Contrairement aux commutateurs de type NPIV évoqués au chapitre précédent, les commutateurs NPV ne disposent pas d'un accès à la configuration de la « fabric ». Ces équipements sont souvent assimilés à un rôle de passerelle dans le monde Fibre Channel.

Pour cette raison, la classe « *CiscoNPVSanSwitch* » ne dispose que des méthodes permettant de modifier la configuration locale du commutateur en respectant les contraintes spécifiques à ces équipements.

### ***III.5.5.1.7 La classe CiscoSanControler***

La classe « *CiscoSanControler* » assure le rôle d'élément orchestrateur au sein du modèle. Tout comme pour les classes contrôleur évoquées jusqu'à présent, cette dernière présente au travers de ses méthodes un ensemble de scénarios complets et structurés qui représentent la réalisation concrète des cas d'utilisation évoqués lors de la phase de conception.

Cette classe assure également le rôle d'analyseur syntaxique du fichier de configuration qui lui ait transmis lors de l'instanciation et qui contient tous les éléments externes nécessaires à son traitement. Les différents commutateurs du réseau SAN Front-End y sont consignés.

La classe « *CiscoSanControler* » peut instancier au cours de son traitement des objets de type « *CiscoNPIVSanSwitch* » ou « *CiscoNPVSanSwitch* » selon les informations contenues dans le fichier de configuration.

### III.5.6 Réalisation des lanceurs applicatifs

Les lanceurs applicatifs sont les scripts Python qui seront manipulés par les utilisateurs pour accéder aux différents cas d'utilisation modélisés par la solution. Il y a donc un lanceur par cas d'utilisation décrit dans ce document.

Ces lanceurs prennent la forme de scripts Python utilisables en ligne de commande. Chaque lanceur peut être utilisé avec un nombre variable d'arguments, ou commutateurs, permettant aux utilisateurs d'effectuer des choix ou activer certaines fonctionnalités optionnelles.

Ces scripts assurent les contrôles syntaxiques permettant de valider les entrées des utilisateurs et fournissent de l'aide contextuelle lorsqu'ils sont appelés à l'aide du commutateur adéquat.

La figure ci-dessous illustre l'utilisation du lanceur associé au cas d'utilisation Cisco « Création d'un alias Cisco » :

```
[admitubruj@ibunsell scripts]$ CISCO-CreateFcAlias.py -h
usage: CISCO-CreateFcAlias.py [-h] fabric name vsan WWN [WWN ...]

R2D2 - CISCO - Creation d'un nouvel FcAlias

positional arguments:
  fabric      Fabric SAN definit dans la configuration
  name       Nom a donner a l'alias
  vsan       VSAN dans lequel declarer l'alias
  WWN       Un ou plusieurs WWN (separateur entre WWN: espace)

optional arguments:
  -h, --help  show this help message and exit
[admitubruj@ibunsell scripts]$
```

Figure 54 – Exemple de lanceur applicatif utilisé pour la création d'un alias Cisco

Le module Python « *argparse* », appartenant à la bibliothèque standard du langage, rend possible la création de telles interfaces en déclarant à la volée les arguments attendus au cours du traitement ainsi que leur caractère obligatoire ou facultatif.

Une fois les contrôles syntaxiques validés et les arguments récupérés, chaque lanceur procède à l'instanciation du contrôleur adéquat et appelle la méthode concernée par le cas d'utilisation en lui fournissant le ou les arguments spécifiés par l'utilisateur.

La suite du traitement est alors entièrement à la charge du contrôleur.

### III.5.7 Réalisation des tests unitaires

Afin de faciliter et de fiabiliser le développement tout en rendant la conduite des tests de non-régression possibles, il est primordial de procéder à la création de tests unitaires tout au long du développement.

Ces tests ont pour objet d'évaluer le bon fonctionnement de chaque classe et de chaque méthode de manière précise et en dehors de tout contexte d'utilisation. Il est ainsi possible de vérifier le bon comportement d'une méthode dans tous les cas de figure envisageables.

La conduite de tels tests est une étape cruciale permettant, notamment, de vérifier le bon fonctionnement de la solution après des modifications sur le périmètre technique comme, par exemple, les montées de versions applicatives.

Le langage Python dispose d'un module consacré aux tests unitaires dans sa bibliothèque standard, ce dernier s'appelle « *unittest* ». Ce module facilite grandement le développement d'objets ayant pour vocation la conduite de tests.

```
import unittest
import lib.emcstorage.arrays

## Classe VplexGetRequests
# Cette classe va initier tous les tests relatifs aux REST interrogations VPLEX
#
class VplexGetRequests(unittest.TestCase):

    ## Methode d'initialisation des tests
    #
    def setUp(self):
        self.vplex_mgnt = "113.92.107.160"
        self.vplex_login = "service"

        #Creation de la reference vers l'objet VPLEX ARRAY
        self.vplexArray = lib.emcstorage.arrays.VplexArray(self.vplex_mgnt,
                                                            self.vplex_login,
                                                            self.vplex_pass)
```

Figure 55 – Exemple de test unitaire utilisant le module Python « *unittest* »

Les tests unitaires prennent la forme de classes héritant de la classe « *TestCase* ». Les méthodes composant cette classe matérialisent les tests qui seront exécutés, en séquence, pour évaluer la conformité de l'objet à tester.

Chaque test unitaire, comme son nom l'indique, vise l'évaluation d'un composant précis du système. En cas de comportant anormal, la méthode renvoie alors un code d'erreur clairement identifiable ainsi que les conditions ayant conduit à cette erreur.

## **III.6 Bilan**

### **III.6.1 Rappel des objectifs du projet**

Les récentes évolutions des infrastructures SAN ont, certes, permis d'étendre de manière conséquente les capacités de stockage ainsi que la qualité de service de ces dernières, mais c'est au prix d'une augmentation importante de leur complexité. Ceci est d'autant plus vrai que les diverses technologies présentes sur le marché, en très grande partie propriétaires, ne sont encore que trop peu souvent compatibles entre elles.

En résulte une inflation très significative du temps passé à en assurer l'exploitation et la maintenance ainsi que du risque lié à ces pratiques.

Consciente de ces problématiques la société i-BP, à l'occasion d'un projet de renouvellement de son infrastructure de stockage de données SAN, souhaite adresser ces sujets de manière durable. C'est également l'occasion de se préparer aux transformations à venir du système d'information et à l'introduction probable des technologies du Cloud Computing.

Dans cette perspective, un projet est initié dont l'un des objectifs principaux est de déterminer quels sont les moyens et techniques permettant de réduire le risque lié aux activités d'administration et de maintenance. Le second objectif, tout aussi important que le premier, est la réduction du coût total d'exploitation de la plateforme.

Ainsi, le but de ce projet pourrait être résumé en cette simple phrase : identifier et mettre en œuvre les méthodes et les techniques permettant de tendre vers une gestion industrielle des moyens de stockage de la donnée.

### **III.6.2 Bilan du projet**

L'intégration des pratiques et des automates développés au cours de ce projet au sein du système d'information a un impact très positif sur l'activité d'administration des infrastructures de stockage de données.

Tout d'abord, le gain en productivité est très perceptible. Le temps moyen consacré par acte d'administration sur les plateformes SAN, VMAX, VPLEX, qui pouvait représenter

jusqu'à une heure de travail selon la complexité de l'opération, est à présent réduit à quelques secondes. Le temps ainsi épargné est réinvesti au profit de tâches plus nobles et auparavant délaissées faute de temps. L'audit des performances du réseau SAN est l'une de ces tâches trop souvent mises de côté et qui peuvent à présent faire l'objet d'un traitement régulier. Cela facilite grandement la conduite de travaux d'optimisation et permet de mieux anticiper l'apparition de goulets d'étranglement.

La fiabilité des processus, élément central de toute activité de production informatique, a également été améliorée par l'utilisation des automates fournis par le projet. Le respect des nomenclatures est assuré de facto et permet de disposer d'une information complète et fiable de chaque élément de configuration de l'infrastructure. Les erreurs de manipulation dont les conséquences peuvent être catastrophiques ont été éliminées par l'introduction de mécanismes de sécurité stricts systématisant les points de contrôle essentiels.

Elément dont l'importance va grandissante et d'autant plus en environnement bancaire : la sécurité opérationnelle est renforcée grâce aux mécanismes de journalisation d'activité présents au sein des automates. Ils permettent d'assurer la traçabilité complète des actions effectuées au sein de l'infrastructure sans limites de temps.

Ce projet a également permis, dans une certaine mesure, de challenger les constructeurs dans leurs offres logicielles en opposant le développement d'outils sur-mesure, répondant très exactement aux attentes et au niveau de spécifications attendues par la société i-BP à des produits, certes, prometteurs mais dont les concepts et surtout la fiabilité restent à être démontrés.

### **III.6.3 Perspectives d'évolution**

Les perspectives d'évolution sont nombreuses, car toutes les problématiques que soulève un tel projet qu'elles soient actuelles ou à venir ne peuvent être adressées au cours d'un projet unique.

En premier lieu, les transformations à venir au sein du groupe BPCE, dont les diverses entités tendent à fusionner entre elles, vont considérablement augmenter le nombre d'équipements dont il faudra assurer l'exploitation. Il est question ici de tripler le parc informatique actuel. Cette évolution va donner l'occasion de mettre à l'épreuve l'actuelle

solution d'industrialisation face aux enjeux de la montée en charge et de la densification des moyens de stockage de la donnée.

Cette augmentation du périmètre s'accompagnera également d'une diversification des équipements en introduisant de nouvelles baies de disques issues de constructeurs autres qu'EMC<sup>2</sup>. Ces équipements ne reposeront ni sur les mêmes concepts ni sur les mêmes modes d'administration que l'actuelle plateforme. C'est pourquoi cette problématique de diversification est également une question centrale qu'il conviendra d'adresser rapidement.

Il sera intéressant d'adapter les outils développés au cours de ce projet et ainsi mettre à l'épreuve leur conception logicielle : ces derniers étant conçus autour de mécanismes devant faciliter le support de nouveaux équipements.

L'augmentation de périmètre soulève également des problématiques liées à la collecte des données techniques et d'inventaires des composants présents au sein de l'infrastructure. A ce jour aucun logiciel du marché ne permet d'adresser ce sujet de manière globale, c'est-à-dire toute technologie et toutes marques confondues. Tout du moins pas avec le niveau de spécifications attendu par la société i-BP.

Ce constat relance l'idée de concevoir en interne une solution alternative répondant précisément à ce niveau d'exigence. Solution qui pourrait être bâtie, en grande partie, sur les fondations du présent projet.

Enfin, et même si cela a mis du temps à arriver au sein de grandes entreprises comme la Banque Populaire, les sujets liés au Cloud Computing sont à présent au cœur des préoccupations. Ils sont la promesse d'une informatique plus flexible répondant très exactement aux besoins de la DSI en un mot : plus agiles.

La société i-BP a initié tout récemment un projet de réflexion pour la mise en œuvre prochaine d'un Cloud privé de type IaaS (Infrastructure As A Service) au sein de son infrastructure technique. Les outils et solutions d'industrialisation développés au cours de ce projet s'inscrivent totalement dans cette démarche et pourraient constituer une base de départ pour une évolution prochaine des moyens et modes d'administration vers le « tout Cloud ».

## **Vers une transformation des modes d'administration en informatique ?**

L'industrie du stockage de données de masse est en pleine mutation. D'infrastructures propriétaires et complètement fermées elles sont en passe de s'ouvrir en proposant des moyens d'interconnexions jusqu'à présent inédits.

De nouveaux outils et de nouvelles pratiques apparaissent, sous l'impulsion de phénomènes comme le Cloud Computing, qui nous promet de virtualiser et d'automatiser toutes ces étapes qui étaient jusqu'à présent l'objet d'interactions humaines : par nature faillibles et coûteuses en temps d'exploitation.

Force est de constater que ces nouveaux outils ne sont pas encore arrivés à maturité. Bogues, architectures complexes, fonctionnalités incomplètes, omniprésence du marketing, autant d'aspects qui ne plaident pas pour une utilisation de ces produits en production où le besoin de stabilité et de maîtrise est primordial.

Cependant, ce besoin d'industrialisation des opérations d'administration existe bel et bien. Pressées de toutes parts pour produire toujours plus vite et réduire toujours plus leurs coûts d'exploitation, les entreprises ne peuvent ignorer ce phénomène qui devient chaque jour plus important pour leur compétitivité.

Des solutions existent pourtant, cette étude l'a démontré : il est possible d'adresser ces problématiques et ainsi challenger les éditeurs dans leurs offres logicielles en opposant le développement d'outils internes répondant précisément aux attentes de l'entreprise aux produits proposés par ces derniers.

Non content de répondre à un besoin imminent, ce type de démarche ouvre de nouvelles perspectives. En améliorant sensiblement les processus de l'entreprise et en introduisant de nouvelles méthodes de travail. Il est alors envisageable de délaissier les opérations purement techniques pour se focaliser sur l'essentiel et ainsi mieux préparer l'évolution du système d'information.

Cependant, ces types de solutions que l'on qualifiera d'alternatives n'ont pourtant pas pour vocation à être pérennes au-delà de quelques années. Elles constituent avant tout un moyen

pour se donner du temps. Le temps de la réflexion tout d'abord, mais aussi et surtout le temps de voir arriver à maturité de nouvelles pratiques et technologies qui vont sans nul doute transformer durablement l'industrie du stockage de données dans les années à venir.

## Bibliographie

- [1] Storage Networking Industry Association. (2014, avril) [www.snia.org](http://snia.org/sites/default/files/SNIA%20Software%20Defined%20Storage%20White%20Paper-%20v1.0k-DRAFT.pdf). [Online]. <http://snia.org/sites/default/files/SNIA%20Software%20Defined%20Storage%20White%20Paper-%20v1.0k-DRAFT.pdf>
- [2] EMC Education Services, *Information Storage and Management*, Wiley Publishing, Ed.: G.Somasundaram, 2009.
- [3] CostaCloud. (2015, janvier) Storage: Virtualized vs Software Defined. [Online]. <http://costacloud.com/blog/storage-virtualize-vs-software-defined/>
- [4] Lucien Avramov, *The Policy Driven Data Center with ACI*, CiscoPress, Ed. US: CiscoPress, 2015.
- [5] EMC2. (2015, mai) CoprHD Project. [Online]. <http://coprhd.github.io>
- [6] OpenStack Foundation. (2014, novembre) The Role of Software Defined Storage in OpenStack Cinder. [Online]. <https://www.openstack.org/summit/openstack-paris-summer-2014/session-videos/presentation/the-role-of-software-defined-storage-in-openstack-cinder>
- [7] Openstack foundation. (2015, janvier) OpenStack Block Storage Cinder. [Online]. <https://wiki.openstack.org/wiki/Cinder>
- [8] Vedams engineering services. (2015, janvier) Is OpenStack Cinder a Software Defined Storage Solution ? [Online]. <http://www.vedams.com/blog/is-openstack-cinder-a-software-defined-storage-solution/>
- [9] Francis MILLOT. (2014, mai) Virtual SAN - Design et dimensionnement. [Online]. <http://www.virtu-desk.fr/blog/infrastructures-hyper-convergentes/calcul-de-stockage-d-une-infrastructure-virtual-san.html>
- [10] EMC2. (2013) VPLEX Element Manager API Guide. PDF.
- [11] EMC2, *SolutionsEnabler Symmetrix ArrayManagement Guide*, 7th ed. Boston: EMC2, 2013.
- [12] Mark Lutz, *Learning Python*, 5th ed., O'Reilly, Ed.: O'Reilly Media, 2013.

## Liste des figures

Figure 1 – Organisation du groupe BPCE.....	13
Figure 2 – Organisation générale de la société i-BP .....	15
Figure 3 – Architecture générale de la solution EMC <sup>2</sup> (sites de SAPHIR et TOPAZE).....	23
Figure 4 – Modèle en couche d’une solution de type SDS (source : [3]) .....	31
Figure 5 – Architecture générale de la solution EMC <sup>2</sup> VIPR .....	36
Figure 6 – Architecture du service Openstack Cinder.....	38
Figure 7 – Illustration de l’approche "scale out" du produit VMWare VSAN (source [6]).....	40
Figure 8 – Composition d’un entrepôt de données sous VMWare Virtual SAN.....	41
Figure 9 – Schéma illustrant le cycle de développement « en V ».....	53
Figure 10 – Liste détaillée des tâches du projet et des charges associées .....	54
Figure 11 – Diagramme de GANTT du projet .....	54
Figure 12 – Architecture globale de la solution EMC <sup>2</sup> VPLEX/VMAX .....	55
Figure 13 – Vue schématique d’un moteur (ou engine) VPLEX .....	55
Figure 14 – Hiérarchie des éléments de configuration au sein d’un cluster VPLEX.....	58
Figure 15 – Représentation d’un « Distributed Device » au sein d’un métré cluster VPLEX.....	58
Figure 16 – Eléments constitutifs d’une « Storage View » VPLEX .....	59
Figure 17 – Fenêtre principale de la plateforme d’administration «Unisphere for VPLEX» .....	60
Figure 18 – Réponse à une requête d’interrogation REST de type GET au format JSON .....	61
Figure 19 – Vue logique d’un directeur VMAX .....	63
Figure 20 – Gestion des disques en octant d’un VMAX .....	64
Figure 21 –Types de disques durs supportés par un VMAX.....	64
Figure 22 – Composants utilisés dans une configuration de type FAST/VP .....	68
Figure 23 – Schéma de principe de la fonctionnalité « Timefinder SNAP/VP ».....	69
Figure 24 –Illustration des différents groupes logiques VMAX .....	70
Figure 25 – Exemple de flux XML utilisé par la suite « Solutions Enabler » .....	72
Figure 26 – Architecture physique du SAN Cisco Front-End.....	74
Figure 27 – Vue physique d’un commutateur Cisco MDS9710 .....	75
Figure 28 – Vue physique d’un commutateur Cisco MDS9148S .....	75
Figure 29 – Fenêtre principale de l’outil d’administration Cisco DCNM.....	77
Figure 30 – Fonctionnement de l’interpréteur Python (source : [10]).....	80
Figure 31 – Exemple de code Cython permettant de déclarer le type des variables .....	83
Figure 32 – Cas d’utilisation VPLEX .....	86
Figure 33 – Diagramme d’activités « Création d’un volume local VPLEX ».....	88
Figure 34 – Diagramme d’activités « Création d’un volume distribué VPLEX » .....	89
Figure 35 – Diagramme d’activités « Création d’une Storage View VPLEX » .....	91
Figure 36 – Diagramme d’activités « Ajout d’un volume virtuel à une vue VPLEX » .....	92
Figure 37 – Diagramme d’activités « Ajout d’un initiateur à une vue VPLEX » .....	94
Figure 38 – Cas d’utilisation VMAX .....	95
Figure 39 – Diagramme d’activités « Création d’un device VMAX » .....	96
Figure 40 – Diagramme d’activités « Ajout d’un device à un Storage Group VMAX ».....	98
Figure 41 – Cas d’utilisation Cisco MDS.....	99
Figure 42 – Diagramme d’activités « Création d’un alias Cisco ».....	100
Figure 43 – Diagramme d’activités « Création d’une zone Cisco ».....	103
Figure 44 – Diagramme d’activités « Renommer un alias Cisco ».....	104
Figure 45 – Diagramme d’activités « Configuration d’un port FC Cisco ».....	106
Figure 46 – Ouvertures de flux nécessaires entre les serveurs de la solution .....	109
Figure 47 – L’environnement de développement intégré Pycharm .....	110
Figure 48 – Architecture du dépôt Subversion du projet .....	111
Figure 49 – Organisation des paquetages du projet.....	111
Figure 50 – Diagramme de classes du module « views ».....	113
Figure 51 – Diagramme de classes du module VPLEX.....	116
Figure 52 – Diagramme de classes du module VMAX.....	120
Figure 53 – Diagramme de classes du module Cisco.....	123

Figure 54 – Exemple de lanceur applicatif utilisé pour la création d'un alias Cisco .....	127
Figure 55 – Exemple de test unitaire utilisant le module Python « <i>unittest</i> » .....	129

## Liste des tableaux

Tableau 1 – Groupe BPCE, chiffres clés.....	12
Tableau 2 – Implantation géographique de la société i-BP.....	14
Tableau 3 – Nomenclature des différents éléments de configuration VPLEX.....	60
Tableau 4 – Nomenclature des éléments de configuration VMAX.....	71
Tableau 5 – Assignation des VSAN ID par commutateur.....	76
Tableau 6 – Nomenclature des éléments liés au réseau SAN Front-End.....	77

# **Conception et réalisation d'une solution d'automatisation des opérations d'administration pour infrastructures de stockage de données SAN**

**Mémoire d'Ingénieur C.N.A.M., Toulouse 2015**

---

## **RESUME**

Les récentes évolutions des infrastructures de stockage de données SAN ont permis d'augmenter très sensiblement les capacités de stockage ainsi que la qualité de service de ces dernières, mais c'est au prix d'une augmentation significative de leur complexité. C'est d'autant plus vrai que les diverses technologies présentes sur le marché, en très grande partie propriétaires, ne sont encore que trop peu souvent compatibles entre elles.

Le but de cette étude est de déterminer quels sont les moyens permettant de réduire les risques liés à l'administration des infrastructures de stockage de la donnée tout en réduisant coût d'exploitation total lié à cette activité.

**Mots clés : SAN, infrastructure, stockage, automatisation.**

---

## **SUMMARY**

The recent developments of the SAN data storage products have increased significantly the storage capacity and service quality but it is at the price of a very significant increase in their complexity. This is especially true as the various technologies on the market, in large part closed, are still all too rarely compatible.

The purpose of this study is to identify the ways to reduce the risks associated with the administration of the data storage infrastructure as well as reducing total operating cost of this activity.

**Key words : SAN, infrastructure, storage, automation.**