Étude et mise en oeuvre d'une plate-forme de développement applicatif et suivi de sa réalisation

Cédric Esnault

IGN
INSTITUT NATIONAL
DE L'INFORMATION
GÉOGRAPHIQUE
ET FORESTIÈRE

le cnam

CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

Centre régional de NOISY LE GRAND

MÉMOIRE

présenté en vue d'obtenir

LE DIPLÔME D'INGÉNIEUR CNAM

SPÉCIALITÉ : INFORMATIQUE

OPTION : IRSM

par

ESNAULT Cédric

Étude et mise en œuvre d'une plate-forme de développement applicatif et suivi de sa réalisation

Soutenu le 14/01/2016

JURY

PRÉSIDENT :	**Mr Michel CRUCIANU**	**Président**
MEMBRES :	**Mme Geneviève DELOR**	**Directrice du CNAM de Noisy-Le-Grand**
	Mr Eric LEE	**Tuteur CNAM**
	Mr David CAUDRELIER	**Tuteur IGN**
	Mr Olivier GUEGUEN	**Tuteur IGN**

Remerciements

Je tiens à remercier mon tuteur, David Caudrelier pour ses conseils, son soutien et sa bonne humeur durant l'encadrement de ce projet.

Je tiens également à remercier l'ensemble de mes collègues qui m'ont encouragé pendant ces années studieuses et permis de réaliser ce mémoire dans d'excellentes conditions.

Je remercie aussi tous les membres du jury, les enseignants et le personnel du CNAM qui m'ont aidé et aiguillé durant ce long cursus.

Je tiens enfin à remercier mon entourage et tout particulièrement ma compagne pour m'avoir soutenu et épaulé toutes ces années.

Liste des abréviations et acronymes

AD	Action de développement
API	*Application programing interface*
AUFS	*Another union file system*
AWS	*Amazon Web services*
CERN	Conseil européen pour la recherche nucléaire
CMDB	*Configuration management database*
CMS	*Content management system*
CNAM	Conservatoire national des arts et métiers
COPIL	Comité de pilotage
CPU	*Central processing unit*
CSA	*Cloud service automation*
CVS	*Concurrent versioning system*
D2SI	Direction des services et du système d'information
DDNS	*Dynamic domain name service*
DISIC	Direction interministérielle des systèmes d'information
DMZ	*Demilitarized zone*
DPC	Direction des programmes civils
DRE	Direction de la recherche et de l'enseignement
DSI	Direction du système d'information
EDI	Environnement de développement intégré
GED	Gestion électronique de document
GPU	*Graphic processing unit*
HTTP	*Hypertext transfer protocol*
IAAS	*Infrastructure as a service*
IDP	Département des projets d'infrastructures
IFN	Inventaire forestier national
IGN	Institut national de l'information géographique et forestière
INSPIRE	*Infrastructure for spatial information in the european community*
IP	*Internet protocol*
IPAM	*IP address management*
ITIL	*Information technology infrastructure library*
ITRF	*International terrestrial reference frame*
LAN	*Local area network*
LXC	*Linux container*
MAAF	Ministère de l'agriculture des eaux et des forêts

MEDDE	Ministère de l'écologie, du développement durable et de l'énergie
MEN	Ministère de l'éducation nationale
MOA	Maîtrise d'ouvrage
MOE	Maîtrise d'œuvre
MQSUSI	Mission qualité, sécurité et urbanisation du système d'information
OIV	Organisme d'intérêt vital
OO	*Operation orchestrator*
OS	*Operating system*
PAAS	*Plateform as a service*
PME	Petite et moyenne entreprise
POC	*Proof of concept*
PRA	Plan de reprise d'activité
PSSI	Politique de sécurité des systèmes d'information
PXE	*Preboot execution environment*
QSUSI	Qualité, sécurité et urbanisation du système d'information
RAM	*Random access memory*
RETEX	Retour d'expérience
RIE	Réseau interministériel de l'État
ROI	*Return on investment*, retour sur investissement
RSI	Réseaux et systèmes informatiques
SAAS	*Software as a service*
SAI	Service des applications innovantes
SDN	*Software defined network*
SDSI	Schéma directeur des systèmes d'information
SESI	Support et exploitation du système d'information
SGN	Service géodésie et nivellement
SI	Système d'information
SIDT	Service du système d'information et des développements technologiques
SIG	Système d'information géographique
SQL	*Structured query language*
SSI	Sécurité du système d'information
SVN	Subversion
TCP	*Transmission control protocol*
VLAN	*Virtual local area network*
VM	*Virtual machine*
VPN	*Virtual private network*

Glossaire

API	De l'anglais *Application programming* interface : Outil de programmation pour interagir entre plusieurs logiciels.
AWS	*Amazon Web Services :* Nom des services de *Cloud* de la société *Amazon.*
Backport	Ce sont des dépôts logiciels *Debian* contenant des paquets de la prochaine version stable.
Big Data	Littéralement « Grosses données », ensemble de données informatiques très volumineuses nécessitant l'utilisation d'outils spécifiques pour leurs traitements.
CCOP	Salaire chargé+frais de services+frais généraux
CERN	Acronyme de « Conseil européen pour la recherche nucléaire » : Laboratoire européen de recherche sur la physique des particules.
Cloud	Abréviation de l'anglais *Cloud* c*omputing*, signifiant « Informatique dans les nuages ». C'est une délocalisation de l'infrastructure informatique.
CMDB	De l'anglais *Configuration management database :* Outil qui permet d'avoir une connaissance exacte du contenu et des configurations des équipements informatiques.
CMS	De l'anglais *Content managment system :* Outil de gestion de contenu.
CRON	Logiciel de planification de commandes sous *Linux.*
DNS	De l'anglais *Domain name system :* Gestion des noms de domaines
DDNS	De l'anglais *Dynamic domain name system :* Outil de gestion dynamique des noms de domaines associés aux adresses IP.
Devops	Acronyme issu de l'anglais *Development & operations :* Mode de fonctionnement visant à favoriser la coopération entre les développeurs et les exploitants qui ont initialement des objectifs antinomiques (innovation *vs* stabilité).
DMZ	De l'anglais *Demilitarized zone : Z*one réseau isolée par un pare-feu généralement accessible depuis Internet.
DropBox	Littéralement « boite de dépôt »: Logiciel de dépôt centralisé de documents en ligne.
From scratch	Expression signifiant « en partant de rien ».
Hardware	Partie matérielle d'un équipement informatique par opposition à *Software* qui désigne la partie logicielle.
Hyperviseurs	Serveurs physiques dédiés exclusivement à l'hébergement de machines virtuelles.

IDP,IDE,IDI...	Sigles administratifs IGN désignant les départements des services SIDT et SAI.
IPAM	De l'anglais *IP address management :* Outil de gestion des adresses IP.
ITIL	De l'anglais *Information technology infrastructure library*: Ensemble de bonnes pratiques pour la gestion des systèmes d'information.
Méthode agile	Approche itérative et incrémentale, menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients (source Véronique Messager Rota).
Nginx	Logiciel libre de serveur HTTP et de proxy inversé.
Opendata	Donnée numérique diffusée selon une licence ouverte garantissant son libre accès et sa réutilisation par tous, sans restriction technique, juridique ou financière (source Wikipédia).
OpenLDAP	De l'anglais *Open Lightweight directory access protocol :* Implémentation Open source du protocole d'annuaire LDAP.
PKI	De l'anglais *Public key infrastructure* : Outil de gestion de certificats électroniques.
Plugins	Littéralement « Branchées dessus » : Extensions permettant d'ajouter des fonctionnalités à un logiciel.
POC	De l'anglais *Proof of concept* : Maquette dont le but est de montrer la faisabilité d'un projet.
Post-installation	Script qui est lancé après une installation, manuellement ou automatiquement, afin de compléter celle-ci.
Reverse-proxy	Outil permettant de rediriger des flux réseaux vers différentes cibles selon certains critères.
Shell	Désigne une interface « utilisateur » du système d'exploitation.
SQL	De l'anglais *Structured Query Language :* Langage d'exploitation de base de données.
TAG	Littéralement « étiquette » : Dans *Docker*, le *TAG* permet de différencier des images similaires par leur nom.
URL	De l'anglais *Universal resource locator :* Adresse référençant un emplacement sur internet.
VCS	De l'anglais *Version control system :* Désigne les logiciels de gestion de code-source.
VPN	De l'anglais *Virtual private network* : Signifie réseau privé virtuel.
Web	Abréviation de l'anglais *World Wide Web* signifiant toile d'araignée mondiale et désignant l'Internet.

Table des matières

Introduction

L'informatique est un domaine en perpétuelle évolution. Pour garantir une qualité de service à ses clients qui sont de plus en plus connectés, l'entreprise doit se transformer. Elle doit adapter ses produits à leurs usages (périphériques mobiles, objets connectés…). Ces usages sont aujourd'hui responsables de l'évolution du *Web*[1], on parle ici de *Cloud*[2], de *Big Data*[3], de services *Web*. Ces concepts ont été mis en place pour répondre à ce besoin d'accès aux données n'importe où et n'importe quand.

Les performances exponentielles du matériel informatique, la quantité d'objets connectés mais aussi leurs obsolescences rapides font que les usages qui en sont fait changent, ils sont éphémères, rapidement remplacés par d'autres.

Pour répondre à ces changements rapides, il faut être réactif. Historiquement, l'Institut national de l'information géographique et forestière (IGN) est un établissement qui a toujours privilégié la qualité et la stabilité de ses produits, parfois au détriment de la réactivité. Cependant, la mise en place en 2014 d'un incubateur à l'IGN (*http://ignfab.ign.fr*), destiné à promouvoir l'expertise de l'IGN et *l'OpenData*[4] au travers d'une assistance aux *Startups* et aux petites et moyennes entreprises (PME), montre que l'institut souhaite s'inscrire dans ce nouveau mode de fonctionnement.

Pour favoriser le développement rapide de nouveaux produits, il est nécessaire de mettre en place de nouvelles méthodes de conception, issues des méthodes Agiles[5] et de leurs évolutions, adaptées à ce cycle de vie du logiciel. De nouvelles méthodes impliquent de nouveaux outils. Une plate-forme de développement applicatif collaborative semble être un outil parfaitement adapté à la mise en place de ces méthodes.

Dans le cadre de la réalisation du travail de fin d'étude du cycle d'ingénieur du CNAM, j'ai été chargé d'étudier puis de mettre en place une telle plate-forme.

---

[1] **Web,** abréviation de l'anglais *World Wide Web* signifiant toile d'araignée mondiale et désignant l'Internet.
[2] **Cloud,** abréviation de l'anglais *Cloud Computing*, signifiant « Informatique dans les nuages ». C'est une délocalisation de l'infrastructure informatique.
[3] **Big Data**, littéralement « Grosses données », concerne des données informatiques très volumineuses nécessitant l'utilisation d'outils spécifiques pour leurs traitements.
[4] **Opendata**, donnée numérique diffusée selon une licence ouverte garantissant son libre accès et sa réutilisation par tous, sans restriction technique, juridique ou financière (source Wikipédia).
[5] **Méthodes agiles**, approche itérative et incrémentale, qui est menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en compte l'évolution des besoins des clients (source Véronique Messager Rota).

Ce mémoire présentera dans un premier temps le contexte du projet au sein de l'établissement puis son organisation concernant la conception logicielle. Les méthodes et techniques employées pour encadrer ce projet seront ensuite détaillées. Un troisième chapitre décrira finalement les études réalisées, les maquettes mises en place pour éprouver les solutions proposées et enfin la mise en place d'une partie de la plate-forme.

I Contexte du projet

I.1 L'Institut

L'IGN tel qu'il existe actuellement est né en 2012 de la fusion de l'Inventaire forestier national (IFN) et de l'ancien Institut géographique national dont il a gardé le sigle IGN. C'est un établissement public à caractère administratif ayant pour mission d'assurer la production, l'entretien et la diffusion de l'information géographique et forestière de référence en France.

Illustration 1: Logo de l'IGN

L'IGN compte plus de 1600 collaborateurs localisés sur une dizaine de sites en France. Il est placé sous la tutelle des ministres chargés respectivement du développement durable et des forêts. Ses missions sont nombreuses et variées, l'institut est ainsi responsable de l'infrastructure géodésique et des systèmes nationaux de références géographiques, gravimétriques et altimétriques en France. L'IGN constitue, assemble et met à jour régulièrement la couverture du territoire en imagerie aérienne et satellitaire. Cela lui permet de concevoir, de réaliser, de maintenir et de diffuser les bases de données géographiques et forestières ainsi que les fonds cartographiques de référence du territoire. Cette démarche participe à la mise en place de la directive « *Infrastructure for spatial information in the european community* » (INSPIRE) qui vise à favoriser les échanges de données au sein de la Communauté européenne. L'IGN a également une importante mission de recherche et d'enseignement dans ses domaines de compétence. Au niveau international, l'IGN participe activement aux travaux relatifs à la normalisation de l'infrastructure d'information géographique.

En plus de sa mission étatique, l'IGN a aussi une activité commerciale. Il réalise et distribue plusieurs produits issus de ses données, comme les cartes de randonnées IGN au 1/25 000, référence en la matière.

Toutes ces caractéristiques font de l'IGN un interlocuteur privilégié dans l'expertise de la géomatique en France. Un des principaux enjeux pour l'IGN est l'évolution de son activité vers la production d'outils et la fourniture de services d'utilisation des données à destination du plus grand nombre.

On peut estimer que plus de 300 agents sont directement liés au développement et à la maintenance d'outils informatiques propre au domaine de la géomatique.

Du *Géoportail*[6] au calcul de *l'ITRF*[7], en passant par le système d'exploitation du *GéoCube*[8], les équipes de développement de l'IGN sont amenées à utiliser de nombreuses technologies et méthodologies informatiques pour produire leurs outils.

Illustration 2: Le Géoportail

[6] http://www.geoportail.gouv.fr : site web institutionnel de visualisation et de diffusion de données géographiques
[7] http://itrf.ign.fr : système international de référence terrestre (en anglais *International terrestrial reference frame*).
[8] http://loemi.recherche.ign.fr/pdf/brochureGeocube1.pdf : appareil de mesure autonome développé par le laboratoire d'Opto-électronique et de micro-informatique de l'IGN.

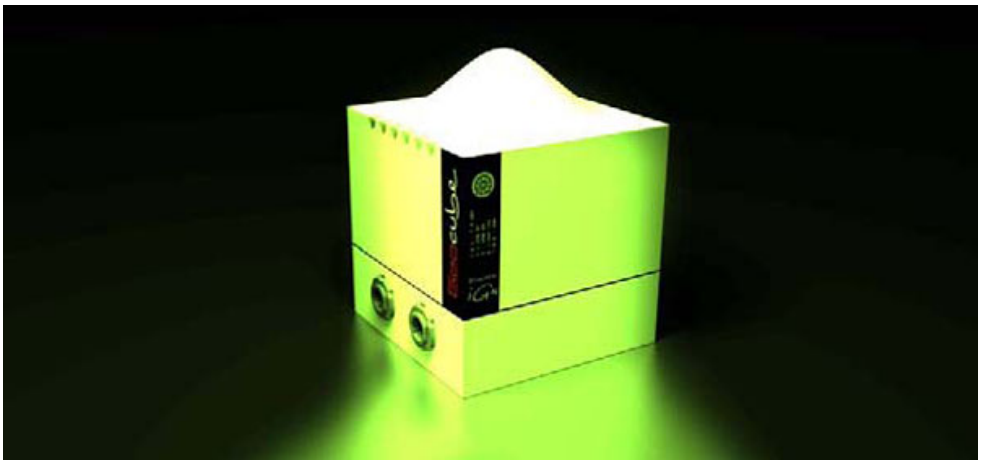


Illustration 3: Le Géocube

I.2 La gouvernance du Système d'information (SI)

En 2013, l'IGN s'est restructuré en créant une Direction des services et du système d'information (D2SI), responsable du SI de l'IGN. Cette Direction est responsable de la conception et de l'évolution du SI en fonction de la stratégie de l'établissement, du besoin des utilisateurs et des avancées technologiques, notamment de celles qui émanent de la recherche. Elle s'assure enfin de la qualité, de la sécurité et de l'efficience du système d'information.

La D2SI s'est engagée dans une démarche ITIL[9] et une de ses premières actions a été de rédiger un Schéma directeur des systèmes d'information (SDSI).

Cette démarche s'accompagne d'une volonté de rationaliser et d'urbaniser les outils et les ressources nécessaires à ses équipes. Dans ce contexte, l'IGN a donc souhaité mettre en place une plate-forme de développement pour atteindre cet objectif.

I.3 La gestion des développements informatiques

La réorganisation de la D2SI a permis entre autres, de regrouper la majeure partie des équipes de développement au sein de deux services principaux, le Service des applications innovantes (SAI) et le Service du système d'information et des développements technologiques (SIDT). En dehors de la D2SI, il reste également des développeurs à la

[9]**ITIL**, de l'anglais *Information technology infrastructure library*, ensemble de bonnes pratiques pour la gestion des systèmes d'information.

Direction de la recherche et de l'enseignement (DRE), au Service géodésie et nivellement (SGN) ainsi qu'à IGN Espace, le service responsable de l'activité Défense et Espace.

I.4 Le département d'accueil

Au sein de la D2SI, le SIDT est responsable du SI dans sa globalité : métiers, système de gestion, infrastructure informatique. Ce service, composé d'environ 90 agents répartis dans 7 départements, assure la maîtrise d'œuvre des développements, la conception, la mise en œuvre et la maintenance des souches logicielles. Il valorise les travaux de recherche, participe aux actions d'innovation, contribue au centre de compétences et d'expertises et assure la veille technologique dans le domaine des SI.

Au sein du SIDT, le département IDP[10] travaille à la fois sur des projets d'évolutions de l'infrastructure mais aussi sur des projets d'évolutions du système d'information, pour la partie liée à l'infrastructure.

Le sujet de projet a été proposé par ce département. Il s'est avéré adapté aux spécificités d'un mémoire CNAM aussi bien en termes de périmètre (technique, organisationnel et fonctionnel) que de durée.

> *Ne pouvant pas quitter totalement mon poste à l'unité Réseaux et systèmes informatiques (RSI) au sein de la DRE, j'ai obtenu de passer 60% de mon temps de travail sur ce projet, support de mon mémoire d'ingénieur CNAM. J'ai donc intégré le département IDP, constitué de trois ingénieurs systèmes & réseaux afin de bien séparer mes deux missions.*

I.5 La gouvernance des projets informatiques à l'IGN

L'organisation de l'IGN prévoit que la maîtrise d'ouvrage (MOA) des projets informatiques soit confiée à la Direction des programmes civils (DPC) et la maîtrise d'œuvre (MOE) à la D2SI.

[10]**IDP** n'est pas un acronyme mais un sigle administratif, il désigne le « Département des projets d'évolution de l'infrastructure ».

À ce titre, chaque projet fait l'objet d'une proposition rédigée (voir chapitre II.2.1) pour présenter l'opportunité du projet (intérêt, gains attendus), une première estimation des coûts, des délais, de la charge de travail et le retour sur investissement (ROI). Cette action de développement est ensuite validée par la DPC puis une organisation de type projet se met en place.

Chaque projet inclut de façon classique un chef de projet D2SI, un Directeur de programme DPC et un Comité de pilotage (COPIL) regroupant des utilisateurs et des représentants côté MOE et MOA.

Dans le cadre de ce projet, qualifié de projet d'infrastructure, la DPC a délégué le rôle de Directeur de projet au chef du SESI en sa qualité de Directeur des projets de refonte de l'infrastructure informatique. (voir organigramme, illustration n°5).

I.6 Les utilisateurs finaux

Si la D2SI est à l'origine du projet, les différentes équipes de développement de l'IGN en sont les utilisateurs et leurs besoins sont divers et variés :

- Le SIDT travaille en grande partie sur des logiciels natifs avec une forte composante en Système d'information géographique (SIG).
- Le SAI quant à lui, travaille avec des partenaires, principalement sur des portails *Web* à dimension géomatique.
- Les chercheurs sont une catégorie à part de développeurs. Ils ont des besoins totalement dépendants de leur sujet de recherche, et ceci dans tous les domaines liés à la géomatique. Concentrés sur le fond de leur recherche, ils sont très demandeurs du partage de connaissance avec les autres développeurs mais sont moins concernés par les aspects de recette et de conditions d'exploitation.

Tous ces besoins se recoupent souvent et il n'est pas rare de voir des équipes de plusieurs unités réaliser le même composant sans recherche de cohérence globale. Les différentes équipes se sont alors concertées et ont émis le besoin de partager leurs acquis et de travailler sur des outils leur permettant d'interagir au niveau du processus de développement. Ce constat est le fondement du projet de plate-forme de développement.

I.7 Le projet

Ce besoin d'outils pour travailler de manière collaborative n'est pas nouveau et les développeurs utilisent déjà de nombreux outils pour les aider dans cette tâche. Les priorités de production des équipes de support informatique font que certains de ces outils ont du être installés et administrés pas les développeurs eux-mêmes. L'absence de concertation pour la réalisation de ces installations a eu pour conséquence une hétérogénéité des outils. De la même manière, les délais de mise à disposition de ressources étant très longs, les développeurs utilisent souvent leur propre machine ou des machines de seconde main pour installer ces outils. Ces mêmes machines sont également utilisées pour tester de nouveaux logiciels mais aussi pour effectuer la qualification et la recette des codes avant la mise en production. Ceci pose des problèmes de maintenance, de performance et d'homogénéité dans le processus de développement.

La mise à disposition de ressources informatiques par les canaux habituels de l'institut n'est actuellement pas compatible avec le cycle de vie très court de ces ressources, il y a donc un écart entre le service rendu et le service attendu.

Les objectifs du projet tels qu'ils sont définis dans le sujet sont :

- Augmenter la productivité, en mutualisant les outils de développement.
- Augmenter l'autonomie des équipes, en supprimant les lourdeurs administratives liées à la fourniture de ressources.

Ces deux objectifs devront être réalisés dans le respect des contraintes identifiées. A l'issue de l'étude préalable, une estimation du ROI sera fournie afin de valider le budget alloué à ce projet. Les ressources humaines nécessaires à sa mise en place ainsi qu'à son exploitation seront évalués et prises en compte, ainsi que la gestion du changement concernant les équipes utilisatrices et les équipes de maintenance. Les solutions proposées seront en accord avec les Politiques de sécurité des systèmes d'information (PSSI) de l'état et plus particulièrement de l'IGN. Notre établissement étant classé Opérateur d'importance vital (OIV), des règles de sécurité spécifiques s'appliquent au niveau de l'exploitation des données géographiques et de la préservation du patrimoine scientifique et technique.

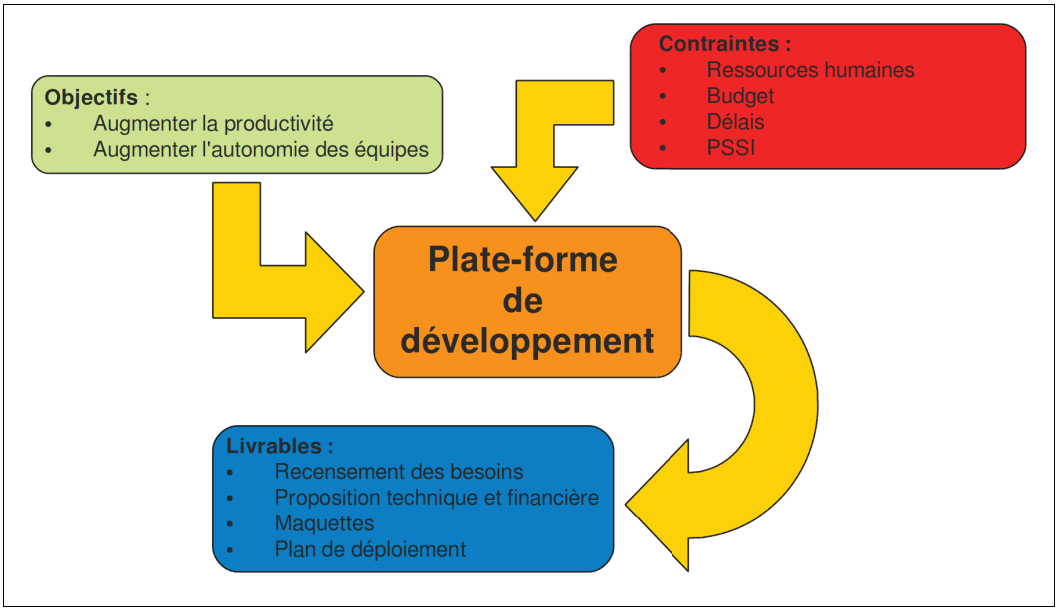


Illustration 4: Mind Map du projet

L'illustration 4 présente le schéma conceptuel du projet de plate-forme de développement.

La gestion de projet mise en place pour mener ce projet va être présentée dans le prochain chapitre. Les différentes phases du projet seront ensuite détaillées dans les chapitres III et IV. Enfin, les perspectives d'évolution de cette architecture seront évoquées.

II Gestion du projet

Ce chapitre aborde le projet du point de vue organisationnel, il décrit les processus et les méthodes utilisées ou mises en places sans aborder l'aspect technique détaillé dans le chapitre III.

II.1 Organigramme

L'organigramme de ce projet est conforme à la gouvernance de projet mise en place à l'IGN. Il indique les intervenants sur ce projet et leurs interactions. Le suivi méthodologique a été réalisé avec la collaboration des différents intervenants concernés par le suivi des projets (commandement D2SI, chef de la mission planification des développements, chef de la mission qualité, sécurité et urbanisation du SI…).

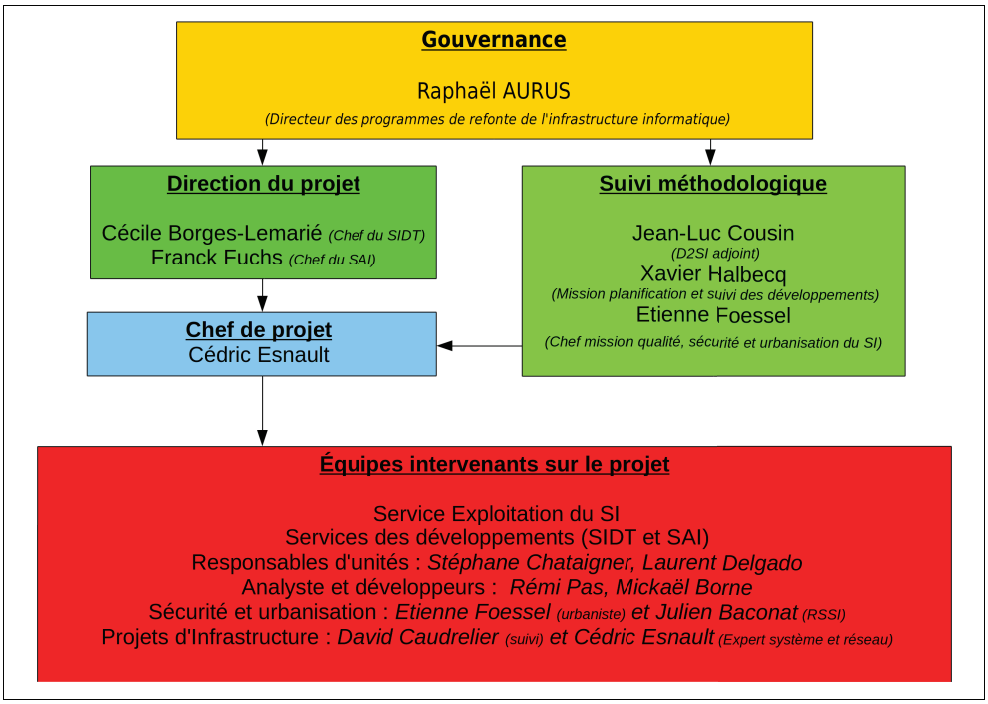


Illustration 5: Organigramme du projet

> *Ce sujet ayant été dimensionné pour mon stage, ma mission était à la fois celle du chef de projet mais aussi celle de l'ingénieur systèmes et réseaux. Mon rôle de chef de projet se concentrait sur le suivi des tâches et de la coordination avec les différents intervenants. Le rôle d'expert technique m'a amené à analyser les problèmes et à trouver les solutions permettant*

d'atteindre les objectifs posés. Si ce second point m'était assez familier, l'aspect management de projet m'a permis de me familiariser un peu plus avec ce volet du métier d'ingénieur.

II.2 Phases du projet

II.2.1 Élaboration du sujet

À partir d'une simple idée exprimée au travers des difficultés rencontrées par les équipes de développement en comité de direction, il faut définir un plan de projet, dimensionné pour un projet de fin d'étude d'ingénieur. Pour réaliser ce plan, une vue d'ensemble de l'organisation et des méthodes de projets à l'IGN est nécessaire. Il faut également tenir compte de la charte des projets IGN qui sert de base à la constitution de tout nouveau projet.

Comme l'indique la gouvernance des projets à l'IGN, le financement et l'acceptation d'un projet de ce type passe par la validation d'une action de développement (AD). La proposition d'AD donne une estimation du budget nécessaire à la réalisation du projet, y compris en moyens humains. Elle présente le projet selon ces différents points :

- Présentation des rôles.
- Fiche budgétaire.
- Présentation du projet.
  - Objectif en production.
  - Travaux existants.
  - Partenariats.
  - Équipements conditionnant le démarrage du projet.
  - Prise en compte des aspects QSUSI.
  - Démarche, points clés et calendrier.

Cette action de développement, disponible en annexe 1 est ensuite validée par le chef du service maître d'œuvre puis par le directeur du projet.

Une fois l'AD validé par la hiérarchie, le sujet peut alors être rédigé en l'adaptant avec les exigences d'un sujet de mémoire d'ingénieur CNAM.

> *La rédaction du sujet de mémoire est une première étape très importante. Elle doit présenter tous les points abordés durant le projet. Il était donc nécessaire de faire une première analyse afin de quantifier ces différentes étapes et de proposer un planning cohérent dans le sujet. Cet exercice aurait été très délicat sans le recul et l'expérience apportée par mon tuteur.*

II.2.2 Définition de la mission

II.2.2.1 Planification du travail

Cette partie consiste à évaluer le travail à effectuer, puis à le découper en tâches potentiellement parallélisables qu'il faut alors assigner aux différents acteurs selon leur nature.

La constitution d'un diagramme de *Gantt* permet de planifier toutes ces étapes et de définir les dates provisoires des comités de pilotage. C'est également une bonne solution pour présenter l'avancement du projet, car il est aisé d'y visualiser les opérations qui prennent du retard et donc qui posent problème.

Le premier planning initial (annexe 3) présenté ici est constitué à partir de l'avant-projet permettant de définir la demande d'action de développement décrite au chapitre II.2.1. Le planning ne contient donc pas la phase de préparation du sujet. Le « t0 » du projet correspond à la validation du projet par la hiérarchie.

Les points clés du planning sont les COPIL, ils vont permettre de cadrer les grandes phases du projet que sont :

- L'étude préalable.
- La réalisation des maquettes.
- La planification d'une solution.

L'évaluation des coûts des différentes tâches ainsi que des délais attendus pour leurs réalisations fait également partie de la planification L'objectif de cette évaluation est de dégager une estimation du *ROI* prévisionnel. L'évaluation de la partie ressource humaine est réalisée à partir des coûts d'imputation des différents agents IGN. Les coûts liés à l'investissement et au fonctionnement sont extraits des marchés publics mis en place à l'IGN pour acquérir logiciels, matériels et unités d'œuvres.

> *Pour réaliser le planning prévisionnel, j'ai demandé des conseils sur les estimations de durées de certaines tâches nouvelles pour moi. L'estimation des coûts est également une tâche très difficile sans une bonne expérience de suivi de projet, notamment en ce qui concerne les ressources humaines.*

II.2.2.2 Coordination du projet

L'organisation du travail fait partie des tâches quotidiennes du chef de projet. On retrouve notamment ces actions :

- Gérer les échanges et les prises de rendez-vous avec les partenaires et les collaborateurs.
- Préparer, piloter les réunions et en faire les comptes-rendus.

II.2.2.3 Communication sur le projet

La communication est un élément primordial dans la réussite d'un projet. Elle a lieu à plusieurs niveaux, tout d'abord avec les collaborateurs directs, acteurs du projet qui ont besoin de connaître l'avancement du projet pour programmer leurs interventions puis avec l'encadrement direct afin d'établir la charge de l'équipe et enfin avec les décideurs à qui il faut rendre compte de l'avancement général et des échéances.

Différentes formes de communications sont mises en place, de façon informelle d'une part, en rencontrant régulièrement les différentes équipes impliquées et en échangeant sur leurs ressentis ou leurs opinions par rapport aux orientations prises par le projet. De façon plus formelle d'autre part, notamment avec l'encadrement, avec plusieurs types de documents.

L'utilisation d'un *reporting* bimensuel avec le tuteur IGN, document très succinct, offre une vue d'ensemble de la situation du projet selon différents aspects ainsi que l'avancement des différentes phases. Une charte de couleur permet de voir immédiatement les points qui posent problème (voir annexe 2).

Un autre mode de communication est aussi utilisé par l'unité IDP sous la forme d'un dépôt *Subversion* (SVN) permettant de gérer le partage de documents de travail relatifs à ses projets et d'en assurer la sauvegarde. De cette manière chaque document utilisé dans le projet (les diagrammes de *Gantt*, les comptes-rendus de réunions, les notes, les schémas, les

documentations techniques…) peut-être consulté par les autres membres de l'équipe. Cela permet d'être informé de l'avancement de chaque projet. Ce dépôt, mis à jour régulièrement ne remplace cependant pas le système de Gestion électronique de document (GED) officiel de l'entreprise dans lequel sont déposés les documents officiels du projet (convocations, comptes-rendus, livrables).

La communication a lieu également avec les partenaires principalement par mail et par téléphone.

La communication avec les décideurs, quant à elle, s'effectue avant tout par la transmission de documents de préparation aux COPIL dont le rôle est décrit au paragraphe II.4.

> *Pendant le projet, j'ai travaillé en étroite collaboration avec l'unité Support et exploitation du système d'information (SESI) dont les équipes doivent s'assurer du bon fonctionnement des systèmes informatique de l'IGN.*
>
> *J'ai également rencontré beaucoup d'agents d'autres services notamment durant les phases de recueil des besoins. Ces rencontres ont été très enrichissantes et m'ont permis de mieux connaître les modes de fonctionnement de ces différents services. Au travers de ces rencontres, j'ai aussi compris les problèmes posés par la gestion des ressources informatiques notamment en ce qui concerne les délais. Les contacts avec les collaborateurs concernés par le suivi du projet tel que la MQSUSI ou la planification des développements ont été eux aussi très utiles. En effet, ces personnes ont posé un regard extérieur sur le projet et m'ont souvent permis d'avancer lors de situations de blocage.*

II.2.2.4 Promotion du projet

Promouvoir un projet comme celui-ci, qui a un impact structurel important, c'est avant tout faire accepter aux différents acteurs que ces modifications sont nécessaires. C'est la conduite du changement. Les deux nouveaux services de développement (SIDT et SAI) sont « jeunes » et issus de la fusion de nombreuses équipes. Ces équipes utilisent des procédures et des méthodes différentes, les évolutions doivent donc être menées avec un maximum de consensus.

Le changement dans un projet d'infrastructure a lieu à plusieurs niveaux. D'un côté, les développeurs doivent s'adapter à de nouveaux outils et certains doivent assumer de nouvelles tâches. De l'autre, les administrateurs systèmes peuvent avoir l'impression que cette automatisation va leur enlever leur part d'expertise et va générer plus d'interventions en maintenance que de bénéfices en temps de mise à disposition. Il faut donc les convaincre de l'amélioration générale que la mise en place du projet peut apporter et préparer les équipes à ce nouvel environnement, en mettant en avant les avantages que chacun pourra en tirer.

Les développeurs n'auront plus de délais avant de mettre en place un nouvel outil, une nouvelle expérimentation. Les administrateurs pourront quant à eux déléguer une partie de leur charge de travail aux développeurs désignés pour l'administration applicative des outils mis en place.

II.3 Budget

Le budget prévisionnel est défini lors de la rédaction de l'AD. Il est programmé par année calendaire et réparti selon trois types de frais : frais de personnels, frais directs et autres dépenses.

En résumé, le budget prévisionnel global pour ce projet est estimé à :

- 76,3k€ en frais de personnels CCOP[11]
- 75k€ en frais d'investissement (matériel, logiciels, licences, supports…)

Ces frais concernent uniquement la mise en place de la plate-forme de développement. Son exploitation relevant d'un autre budget, elle n'est donc pas prise en compte ici.

II.4 Organisation

Pour gérer l'avancement du projet. Il est nécessaire d'avoir au minimum un point hebdomadaire oral avec le tuteur et de réaliser des *reportings* bimensuels pour identifier les dérives éventuelles. Des échanges plus fréquents sont mis en place à l'approche des COPIL pour organiser ces derniers.

Le COPIL permet de regrouper les acteurs principaux, le Directeur du projet, les responsables des services de développement, de l'équipe d'exploitation, la planification et la MQSUSI. Le Directeur du système d'information (DSI) est également invité et valide les livrables fournis en séance.

La tenue des COPIL est indispensable pour pouvoir avancer dans le projet et leurs enjeux sont doubles. Premièrement, ils permettent de présenter l'avancement du projet et d'identifier les problèmes rencontrés depuis le dernier point clé. Deuxièmement, ils doivent permettre de valider les orientations stratégiques du projet. Une bonne préparation et la fourniture d'informations détaillées sont donc primordiales pour permettre aux décideurs présents de faire les bons choix et d'orienter le projet dans la bonne direction.

[11] **CCOP**, salaire chargé+frais de services+frais généraux

Trois comités de pilotages se sont tenus tout au long du projet. Le premier a permis de valider la nature des besoins et de donner le « feu vert » pour la poursuite de l'étude sur deux axes :

- La création de forges logicielles automatisées.
- La fourniture de ressources informatiques par un portail de type *Cloud*.

Le deuxième COPIL s'est conclu par la demande d'une recette pour la solution technique retenue pour la création des forges. Concernant la mise à disposition de ressources informatiques, le COPIL a pris connaissance des scénarios proposés et a demandé le report à 2016 de la réalisation par manque de moyens humains et financiers.

Le troisième COPIL a validé la solution technique de déploiement de forge et a demandé sa mise en production ainsi qu'un cycle de formation pour les utilisateurs. Il marque également la fin du projet.

> *Concernant l'organisation du travail, j'ai dû partager mon temps de travail entre mes deux missions. Mes supérieurs m'ont fait entièrement confiance pour organiser mon emploi du temps pendant toute la durée du projet. En fonction des rendez-vous et des réunions, j'ai privilégié ma présence en début de semaine sur un site et en fin de semaine sur l'autre afin de faciliter les échanges.*
>
> *Depuis la fin du projet, je garde un rôle de correspondant technique et de formateur lors des sessions de formation aux technologies mises en place dans le cadre des forges.*

II.5 Livrables

L'identification des livrables pour chaque phase du projet est importante, car le livrable est un élément tangible qui permet de se concentrer sur les objectifs à atteindre.

II.5.1 Étude préalable

Plusieurs livrables sont prévus pour cette phase :

► Les comptes-rendus des RETEX (Retour d'expérience) du projet interministériel G-Cloud.

► L'étude préalable en elle-même incluant :

▷ La synthèse des besoins et de l'existant.

▷ L'état de l'art.

▷ L'analyse fonctionnelle.

▷ L'étude de coût.

II.5.2 Réalisation des maquettes

Pour la réalisation des maquettes de type POC[12], on considère plusieurs livrables :

► La documentation d'architecture technique à destination de la Mission qualité, sécurité et urbanisation du système d'information (MQSUSI) pour les aspects Sécurité du SI (SSI) et urbanisation.

► La documentation d'installation et d'exploitation à destination des équipes d'exploitation.

► Les manuels « utilisateurs ».

► Les maquettes en elles-mêmes, de façon temporaire et la documentation nécessaire à leur reconstruction ultérieure.

► Le compte-rendu d'évaluation.

> *Plusieurs maquettes ont été réalisées. Ces maquettes, présentées en COPIL ont pour but d'obtenir l'adhésion du commandement en démontrant la faisabilité et l'intérêt de la mise en place de ces solutions.*

II.5.3 Déploiement de la solution choisie

Les livrables attendus pour la mise en place d'une solution retenue sont les suivants

► Le dossier d'architecture technique.

► La documentation d'installation.

[12]**POC**, de l'anglais « *Proof of concept* », désigne une maquette dont le but est de montrer la faisabilité d'un projet.

- La documentation « utilisateur ».
- Le cahier de recette.
- Les supports de formation éventuels.

En complément des livrables habituels, le présent rapport fait office de bilan de projet.

III Étude préalable

Ce chapitre permet de cerner trois éléments fondateurs que sont les objectifs du projet, les besoins des utilisateurs et les solutions disponibles pour répondre à ces besoins.

III.1 Objectifs

La phase de définition du sujet a permis d'échanger avec la D2SI sur ses attentes à propos du projet. L'objectif de l'étude est donc de permettre la mise en place d'une plate-forme de travail, offrant aux acteurs du développement logiciel à l'IGN (analystes, programmeurs, testeurs…), des outils cohérents afin d'augmenter la productivité et l'autonomie des équipes tout en s'adaptant à la diversité des différents projets.

Il n'y a pas de remise en cause des méthodologies propres au processus de développement. L'étude se concentre donc exclusivement sur les moyens permettant d'assister les personnels dans leurs missions.

Les trois étapes de la démarche mise en place sont d'étudier les attentes et les contraintes, de les traduire en besoins fonctionnels puis de trouver les outils les plus adaptés pour atteindre ces objectifs.

III.2 Expression du besoin

III.2.1 Recueil des besoins

III.2.1.1 Préparation

Il convient dans un premier temps d'identifier les agents concernés et de les interviewer pour obtenir ou valider les informations souhaitées.

Les comptes-rendus des différentes réunions de cadrage sur le processus d'amélioration de l'environnement de développement à l'IGN ainsi que les outils en place dans les départements de développement forment les bases pour la constitution d'un questionnaire.

La première partie de ce questionnaire concerne l'évaluation des besoins techniques, elle propose des pistes de réflexion pour exprimer les besoins concernant :

- Les études et les maquettes.
- L'environnement de travail des développeurs.
- La gestion et le partage du code-source produit ou utilisé.
- La gestion et le suivi des projets y compris le suivi des demandes et des incidents.
- L'intégration continue et les tests.
- La qualification.
- La livraison et la recette.
- La maintenance.
- La qualité.
- La formation.
- Les statistiques et le *reporting*.
- La sécurité et la confidentialité.
- La disponibilité.
- La diversité des systèmes cibles et hôtes.
- La formation.
- Tous les besoins qui ne rentraient pas dans ces catégories.

La seconde partie concerne l'identification des différentes contraintes à la fois techniques et administratives que peuvent rencontrer les équipes de développement durant leurs projets.

Ici encore, le questionnaire oriente les personnes sondées vers différentes catégories tout en laissant la liberté de fournir des informations non prévues par celui-ci. Les pistes de réflexion sont les suivantes :

- Contraintes liées à la collaboration avec d'autres organismes.
- Contraintes liées à la sécurité et à la confidentialité.
- Contraintes liées aux délais.

- Contraintes liées aux coûts et aux achats.

Enfin, la troisième partie du questionnaire permet d'identifier les différents acteurs et leur niveau d'interaction avec le processus de développement.

III.2.1.2 Interviews

Le tableau synthétique suivant montre les différents départements interviewés et précise leur type d'activités principales.

Département	Cœur de métier	Type de développement principal
SIDT/IDC	Géomatique et cartographie, traitement de l'information géographique vectorielle	-Code natif Windows de traitement de l'information géographique sous forme vecteur -Industrialisation de sujets de recherches liés aux données vectorielles -Démonstrateurs Web
SIDT/IDD	Archivage et diffusion des données	-Outils liés au système d'archivage et de diffusion (code natif Linux)
SIDT/IDE	Pôle technique du Géoportail	-Technologies du Web (Java, Javascript…) -Code Natif Linux
SIDT/IDF	Inventaire forestier et environnemental	-Technologies du Web -Outils statistiques
SIDT/IDG	Système d'information de gestion	-Code natif Linux (scripting) en environnement de base de données
SIDT/IDI	Image et Lidar, traitement de l'information géographique "raster"	-Code natif Windows et MacOS -Systèmes embarqués
SAI/ISD	Développement de micro-services	-Technologies du Web -Code métier en géomatique
SAI/ISAI	Incubateur	-Tous types
DRE/SRIG	Laboratoires de recherche en géomatique	-Tous types de langages et de supports selon les sujets de recherche. (services Web, calculs mathématiques, systèmes temps réels…)
DRE/ValiLab	Enseignement et valorisation des projets de recherche	-Tous types de langages et d'infrastructures
DPR/SGN/PMT	Développements relatifs à la géodésie et au nivellement	-Code natif (C++, Java, Windows et Linux) et Web

Tableau 1 : Tableau des interviews

Les interviews des équipes de développeurs montrent une grande diversité de besoins qu'il faut synthétiser et classer en vue d'une retranscription en besoins fonctionnels. En dehors de cette constatation, le point flagrant est la volonté unanime de vouloir rationaliser les processus en uniformisant les outils et en mutualisant les réalisations (création d'un socle commun par exemple).

III.2.1.3 Synthèse des interviews

D'une manière générale, les personnes interviewées attendent une grande simplicité de la plate-forme de développement. L'adoption de cette dernière ne se fera qu'à cette condition. Des formations seront à programmer pour les développeurs sur cette plate-forme. L'accompagnement au changement ne devra pas être négligé.

Les gains attendus par la mise en place d'une plate-forme de développement sont avant tout d'ordre organisationnel. Une simplification des accès aux ressources nécessaires pour les développements ainsi qu'une réactivité plus forte sont les deux principales attentes. Une uniformisation des outils et la capitalisation sur les développements et les méthodes sont également souhaitées dans un second temps.

III.2.2 État de l'existant

Pour bien cerner les conditions de travail des développeurs, il est nécessaire de faire un état des lieux des solutions et méthodologies utilisées lors du déroulement d'un projet de développement. On constate là encore une grande diversité de solutions mises en œuvre mais aussi le côté “artisanal” de certains outils mis en place par les développeurs eux-mêmes. Ces outils coûtent en termes de maintenance pour les équipes de développement et les compétences sont la plupart du temps concentrées sur une seule personne.

III.2.2.1 Outils

Le cas de l'équipe “Pôle technique du *Géoportail*” se détache par sa spécificité. En effet, le *Géoportail* est un gros projet dont la production est sous-traitée par un prestataire. Ce dernier propose une architecture de développement sous la forme d'une forge logicielle intégrée, spécifiquement développée pour ce projet et hébergée dans le *Datacenter* du prestataire. Cette forge permet d'accueillir de nouveaux projets annexes au projet *Géoportail* (pour les API[13] notamment) et propose les outils nécessaires au processus de développement dans un mode semi-automatisé. Elle contient donc un gestionnaire de code-source *Mercurial* et un gestionnaire d'incidents et de demandes *Jira.* Ce dernier est également utilisé pour faire les demandes de passages en qualification puis en production qui sont gérées exclusivement par le prestataire.

[13]**API**, de l'anglais *Application programming interface,* outils de programmation pour interagir entre plusieurs logiciels.

La disponibilité de cette forge pose des problèmes de maintenance. En effet, plusieurs projets sans lien concret avec le *Géoportail* ont profité de la présence de cette forge pour en utiliser les services. Cependant, le contrat du prestataire arrivant bientôt à échéance, il faudra migrer tous ces projets sur d'autres forges.

La question de la transition et de la migration revient sur beaucoup de projets, notamment concernant le gestionnaire de code.

Si les agents interviewés sont favorables à une uniformisation des outils, tous s'inquiètent de la migration vers un nouvel outil unique. Ce point a été tranché lors du premier COPIL en indiquant qu'il n'y aurait pas d'assistance pour la migration, vu la charge de travail nécessaire. Les projets en cours se termineront donc avec les outils avec lesquels ils ont été débutés.

On peut établir un listing des applications utilisées dans le cadre des processus de développement en les classifiant dans cinq catégories principales.

► **Gestion du code-source** : Pour la gestion du code-source, le VCS[14] *Mercurial* est utilisé pour le *Géoportail, Svn* et *Git* dans les autres unités. On trouve encore des dépôts sous *Cvs*[15] mais ils sont en cours de migration. Des outils existent pour gérer la transition d'un format vers l'autre sans perte d'information.

► **Gestion de projet** : Au niveau de la gestion de projet, incluant la gestion des demandes et des incidents « *Bugtracking* », *Redmine* est principalement utilisé, mais d'autres logiciels ont été mis en place comme *Jira*, qui est inclus dans la forge du *Géoportail*, ou encore *Mantis* et *Trac.* Les transitions d'un gestionnaire à l'autre sont ici plus complexes étant donné que les logiciels ne gèrent pas tous exactement les mêmes fonctionnalités et qu'il n'existe pas d'outil de transfert d'un moteur vers l'autre.

► **Intégration continue** : Les départements qui ont mis en place un processus d'intégration continue ont tous choisi d'utiliser le logiciel *Jenkins*.

► **Communication** : Pour la communication qu'elle soit interne au projet ou à destination du public (cas des développements Open source), beaucoup de solutions sont utilisées. En interne, les fonctionnalités de communication proposées par les outils de gestion de projet suffisent souvent. Plusieurs CMS[16] sont utilisés pour réaliser des sites web de présentation de projets en exposant la documentation par

[14]**VCS**, de l'anglais *Version control system*, désigne les logiciels de gestion de code-source.
[15]**CVS**, de l'anglais *Concurrent versioning system*, un des premiers gestionnaire de code source collaboratif.
[16]**CMS**, de l'anglais *Content managment system*, outil de gestion de site *Web.*

exemple. Les laboratoires de recherches favorisent l'utilisation de *Wiki* pour mettre en avant le côté collaboratif de cette communication, *DokuWiki* est particulièrement apprécié pour cette tâche.

► **Diffusion** : Ces outils sont particuliers, car ils dépendent principalement des langages ou des OS utilisés. Certains départements utilisent des gestionnaires de dépendance qui opèrent en tant que *Proxy*[17] des dépôts publics pour améliorer les performances (*Composer*, *Sonatype nexus*). Il n'y a cependant pas de réelle volonté de mutualiser ces outils tant ils sont liés au type de projet réalisé.

Les choix concernant les postes de travail n'ont pas été pris en compte. En effet, les départements ne souhaitent pas imposer d'environnement de travail à leurs développeurs estimant que ces derniers sont plus efficaces avec un environnement qu'ils connaissent et maîtrisent, à partir du moment où il est compatible avec les outils collaboratifs utilisés par l'équipe.

III.2.2.2 Ressources

Concernant l'utilisation de ressources informatiques pour les besoins propres aux projets (hors poste de travail), on observe globalement une utilisation détournée intensive des postes de travail et des ressources existantes avec les problèmes d'interactions, de performances et de maintenance que cela implique.

En effet, l'installation de machines virtuelles sur les postes des développeurs pour fournir une application (*composer*, CMS, etc) implique que la machine du développeur soit en permanence allumée. Cela pose des problèmes d'accessibilité lors des dysfonctionnements ou des mises à jour initiées par les équipes de maintenance n'ayant pas connaissance de l'outil en question.

L'utilisation des postes de travail en attente de reversement est courante et ne peut pas être une solution pérenne.

Enfin, la mutualisation des ressources spécifiques à un projet pour un autre projet n'est pas toujours possible et peut poser des problèmes de sécurité et de pérennité des solutions déployées.

[17]**Proxy**, logiciel faisant office d'intermédiaire entre plusieurs machines lors d'un échange de données.

III.2.2.3 Mise en production

L'aspect de la mise en production est également évoqué lors des interviews. Cela se passe généralement par la diffusion des codes validés ou des binaires à d'autres équipes qui se chargent de faire la mise en production (déploiement par les « équipes produit » ou mise en place par les équipes d'exploitation du SI). Il n'y a pas d'équipe en mode « Devops[18] » mis à part dans le pôle « incubateur ». Cette dernière équipe accueille des projets externes et travaille en autonomie totale par rapport au SI de l'IGN. Elle est donc responsable de bout en bout du cycle de vie des solutions développées.

III.2.2.4 Procédures

Le dernier point abordé avec les équipes de développement est d'ordre méthodologique. Il concerne les procédures mises en place pour la demande et la maintenance des outils de développement. La D2SI, dans sa démarche d'adoption des bonnes pratiques ITIL, a mis en place une gestion des demandes et des incidents. Cet outil permet de garantir la traçabilité des demandes de mise à disposition de ressources et de maintenance des outils connus des équipes d'exploitation. En revanche, elle n'apporte pas de réelle amélioration quant à la réduction des délais de mise à disposition des outils.

Ces délais ont des origines organisationnelles et techniques. D'une part, les demandes liées aux processus de développement sont rarement prioritaires sur les demandes liées à la production. Les incidents sur les chaînes de production sont donc traités systématiquement avant les autres, ce qui allonge les délais de traitement des demandes liées aux outils de développement. D'autre part, les équipes d'exploitation sont peu formées aux outils de travail des équipes de développement. L'installation de ces outils ne peut donc se faire qu'en concertation avec les deux équipes ce qui implique là encore des délais pour trouver des créneaux disponibles. La mise en place à l'IGN des recommandations de gestion de mise en production d'ITIL n'a pas encore débuté et les ingénieurs au profil *Devops* manquent pour accompagner la D2SI dans cette démarche.

[18]**Devops**, acronyme issu de l'anglais *Development & operations,* mode de fonctionnement visant à favoriser la coopération entre les développeurs et les exploitants qui ont initialement des objectifs différents (innovation *vs* stabilité).

III.3 Analyse du besoin

Les synthèses des remontées de besoins et de l'existant, permettent de préparer la retranscription du besoin par une analyse fonctionnelle. Pour bien préparer cette analyse, un état de l'art des solutions disponibles sur le marché est nécessaire, ainsi qu'une analyse des retours d'expériences (RETEX) de projets réalisés au sein des différents ministères.

III.3.1 Retours d'expériences *G-Cloud*[19]

La Direction interministérielle des systèmes d'information (DISIC), dont une des missions est de coordonner les projets liés aux technologies de *Cloud computing*, pilote le projet *G-Cloud*. Ce projet gouvernemental a pour objectif de capitaliser sur les retours d'expériences à propos des études et des maquettes réalisées au sein des différents ministères et établissements publics. Plusieurs expérimentations POC sont menées durant la période du projet et leurs résultats permettent d'orienter les choix notamment au niveau technique.

Parmi les projets suivis dans ce groupe de travail, trois expérimentations ont eu d'excellents retours et semblent adaptées aux besoins exprimés lors des premières analyses du projet plate-forme de développement de l'IGN.

III.3.1.1 Ministère de l'éducation nationale (MEN)

Le MEN a mis en production un portail web permettant l'accès sans délai à des machines virtuelles dites « bac-à-sable » pour les centres de formation. L'architecture s'appuie sur la plate-forme de *Cloud* privé *OpenStack* (voir chapitre III.4.2.1) et le MEN a choisi l'assistance de *Mirantis*, société américaine avec une grande expérience du déploiement d'*OpenStack* pour mettre en place ce service. L'accès aux machines se fait via le portail *Web Guacamole* permettant l'accès aux machines virtuelles directement dans le navigateur. Après plusieurs mois d'utilisation, les retours de l'équipe du pôle de compétence du MEN sont très bons, notamment sur la collaboration avec *Mirantis.* Ils insistent sur le fait que le support d'un spécialiste du *Cloud* et particulièrement d'*OpenStack* est indispensable étant donné la complexité des architectures sous-jacentes surtout lors des montées de versions d'*OpenStack*. Ils mettent en garde également sur le fait que le portail *Horizon* d'*OpenStack* n'est pas adapté

[19] ***G-Cloud,*** Gouvernemental Cloud

à un public non administrateur et qu'il faut donc prévoir des alternatives en fonction du public visé.

III.3.1.2 Ministère de l'Intérieur

Le Ministère de l'intérieur a mis en place un service de dépôt de documents en ligne de type « *Dropbox*[20] » accessible par les agents en déplacement. *OpenStack* a également été plébiscité pour leur infrastructure en s'appuyant cette fois ci sur *HP Helion OpenStack*, le produit intégré de la firme américaine *Hewlett Packard*. Les retours sont également très bons. Les ingénieurs du Ministère insistent eux aussi sur la nécessité de se faire accompagner pour mettre en production des infrastructures aussi complexes.

III.3.1.3 DISIC / MEEDE / MAAF

Un dernier projet important a été piloté par la DISIC et réalisé par le Ministère de l'écologie, du développement durable et de l'énergie (MEDDE) et le Ministère de l'agriculture des eaux et des forêts (MAAF). Ce projet, réalisé dans le cadre de la démarche interministérielle G-Cloud vise à montrer la capacité d'interopérabilité des infrastructures *Cloud* basées sur des logiciels libres dans des environnements hétérogènes. Ce POC s'articule également sur des infrastructures *OpenStack.* Les machines sont réparties sur du matériel de marques différentes dans deux *Datacenters*, un au MEDDE à la Défense, l'autre au MAAF à Toulouse. L'ensemble est relié au Réseau interministériel de l'État (RIE) selon des mécanismes de tunnel VPN[21] spécifiques. L'objectif principal de ce POC est de vérifier si la mise en place d'un Plan de reprise d'activité (PRA) sur une infrastructure externe est possible. L'exemple donné est d'avoir une Machine virtuelle (VM) du MEEDE redémarrée sur le *Datacenter* du MAAF et inversement.

Le retour d'expérience sur ce POC est mitigé. La coopération et la communication entre les ministères et les équipes RIE se sont très bien passées et les architectures réseaux et systèmes n'ont pas connu de problèmes majeurs. En revanche, l'implémentation d'*OpenStack* dans sa version « standard » dans une architecture complexe s'est révélée très problématique et les prestataires retenus pour l'implémentation ont été dépassés par cette complexité. Au-delà des conclusions techniques, cette expérimentation démontre la complexité de ces

[20]**Dropbox**, littéralement « boite de dépôt », logiciel de dépôt centralisé de document en ligne.
[21]**VPN**, de l'anglais *Virtual private network*, signifie réseau privé virtuel.

architectures et la difficulté de trouver des expertises dans ce domaine. Ce dernier aspect est primordial dans la recherche de solutions pour fournir des ressources dans le cadre de la plate-forme de développement.

III.3.1.4 Synthèse

La principale observation résultant de ces retours d'expériences se résume en deux points :

- Dans le domaine Open Source, *OpenStack* semble inévitable. Il n'y a effectivement aucun concurrent identifié dont le produit soit suffisamment abouti pour être déployé en production.
- La mise en place et surtout le maintien d'une solution *OpenStack* demandent une expertise poussée dans ce domaine.

III.3.2 Analyse fonctionnelle

III.3.2.1 Classement fonctionnel

Pour établir le classement fonctionnel des besoins, il faut reformuler les besoins exprimés sous forme de fonctionnalités attendues ou encore de cas d'utilisation. Les interviews des équipes d'exploitation et MQSUSI permettent quant à elles de définir les contraintes.

Il devient alors possible de distinguer trois niveaux de fonctions selon leur importance et leur complexité de mise en œuvre :

- Le premier niveau exprime les fonctions indispensables attendues par les équipes et conformes aux contraintes de sécurité, de coûts et de délais.
- Le deuxième niveau regroupe les fonctions importantes qui pourraient être mises en place selon la complexité de mise en œuvre et les moyens disponibles.
- Le dernier niveau comprend les fonctionnalités non indispensables qui ne seront pas mises en place dans un premier temps mais qui doivent tout de même apparaître dans l'étude. La liste des besoins remontés est disponible en annexe 4.

Le diagramme de cas d'utilisation général (illustration 6) d'une plate-forme de développement résume les fonctions de premier niveau attendues. La ligne rouge représente la limite entre les deux axes d'études comme expliqué au chapitre III.3.2.2.

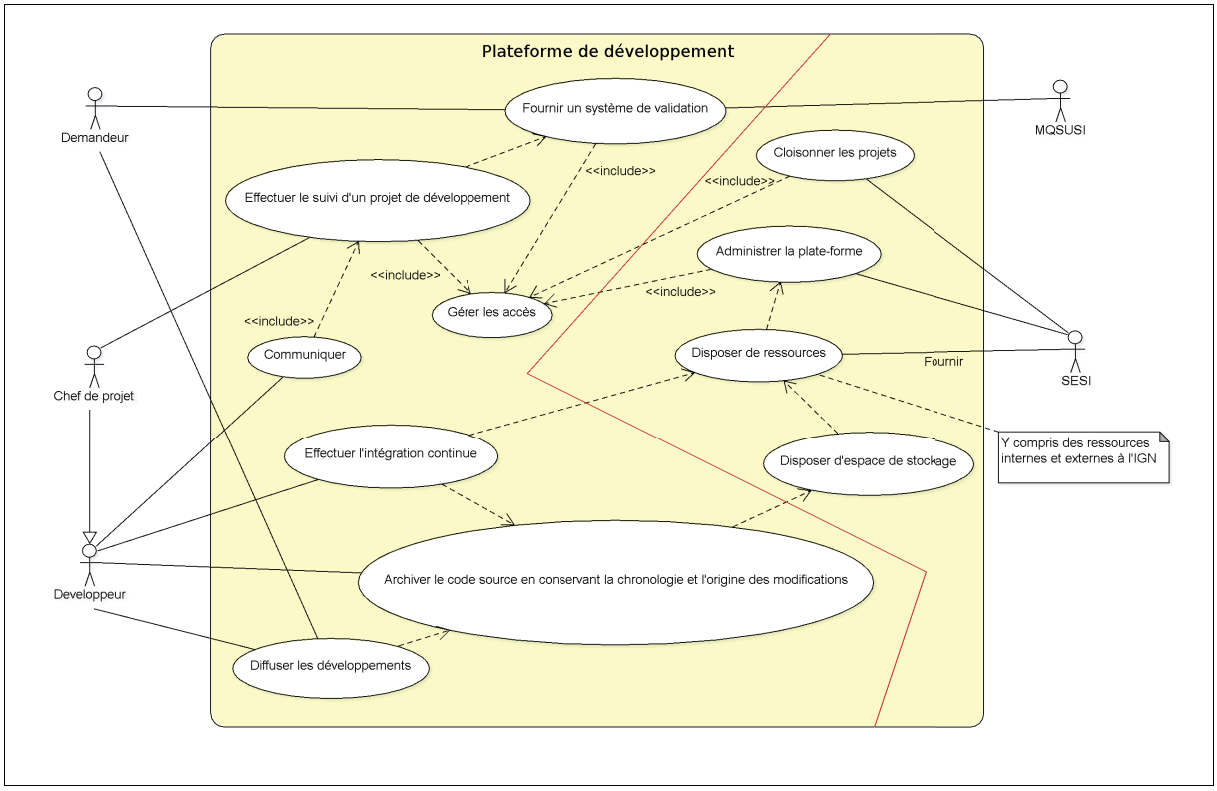


Illustration 6: Diagramme de cas d'utilisation général

Dans chacun de ces cas d'utilisation, les sous fonctions ont été hiérarchisées par ordre de fréquence de demande au sein des équipes tout en prenant en compte la difficulté potentielle de mise en œuvre. Le tableau 2 liste les besoins fonctionnels retenus et validés par le COPIL.

Les principaux besoins se retrouvent dans les deux premiers cas d'utilisation : « Effectuer le suivi d'un projet » et « Archiver le code source … ». La deuxième brique étant dépendante de la troisième « Disposer de ressources », sa mise en place impliquera des solutions techniques complexes. Cette complexité globale aura deux impacts, un premier sur l'infrastructure, un second en termes de sécurité sur la mise en place de la fonctionnalité « Gérer les accès ».

Cas d'utilisation	Besoin couvert	Contraintes/besoins induits
Effectuer le suivi d'un projet de développement	Effectuer le suivi d'un projet Gérer les droits et les accès Disposer d'un système de demande, de report d'incident et de bug Disposer d'un espace d'échange collaboratif Exposer la documentation Avoir des indicateurs de suivi de l'avancement et des ressources	Isoler les projets Authentifier les acteurs Identifier les droits Accès à la plate-forme depuis l'extérieur de l'IGN Disposer de suffisamment de ressources Maîtriser ces ressources Proposer plusieurs langues Proposer un Workflow de validation
Gérer le code source	Conserver l'historique du code source Sauvegarder le code source Collaborer sur une même code source Capitaliser sur les codes sources	Authentifier les acteurs Identifier les droits Accéder à la plate-forme depuis l'extérieur
Développer/ Effectuer l'intégration continue	Mettre en place des maquettes Tester les développements Tester la robustesse Tester sur des terminaux mobiles Proposer des jeux tests Capitaliser sur les jeux tests Capitaliser sur les environnements mis en place Effectuer la maintenance Générer la documentation Historiser les environnements	Authentifier les acteurs Identifier les droits Accéder à la plate-forme depuis l'extérieur Disposer de suffisamment de ressources Accéder à des données IGN Accéder à des ressources externes à l'IGN Disposer de tout type d'environnement Disposer d'un système de gestion des licences Disposer d'une infrastructure WiFi
Diffuser	Mettre en place des démonstrateurs Distribuer les produits finis Exposer les produits sous forme de services Web Réaliser la recette Proposer des environnements de formation Proposer des packages pour les développeurs	Accès à la plate-forme depuis l'extérieur Disposer de suffisamment de ressources
Administrer	Gérer l'infrastructure de la plate-forme de développement	Administrer la plate-forme Authentifier les acteurs Fournir de nouvelles ressources Maîtriser ces ressources Disposer d'un système de gestion des licences Maintenir la plate-forme Assurer une astreinte

Tableau 2 : Classement fonctionnel des besoins

Le cloisonnement ou l'ouverture des projets est un paramètre majeur lors de la mise en œuvre de la solution. L'architecture générale retenue devra donc prendre en compte ce paramètre impliquant fortement la sécurité.

III.3.2.2 Axes d'étude

Parallèlement à ce classement hiérarchique, il est possible de différencier deux axes d'étude dans l'analyse des fonctionnalités retenues. Pour rappel, cette séparation est représentée par le trait rouge sur l'illustration 6 :

- Les fonctions liées au besoin d'outils comme la gestion du code-source.
- Les fonctions liées au besoin de ressources informatiques comme le maquettage.

Certaines fonctions peuvent bien entendu être liées à ces deux aspects.

L'aspect méthodologique du processus de développement est également pris en compte. Cependant, ce dernier est totalement lié à la politique de l'entreprise. La mise en place d'une plate-forme de développement ne doit donc pas influencer cette politique mais bien en être le résultat.

III.3.2.3 Forge logicielle

Au niveau des outils, l'environnement du développeur ayant été exclu de la problématique, on retrouve les cinq fonctionnalités principales identifiées au chapitre III.2.2.1 pour lesquelles il faut fournir des logiciels. L'illustration 7 montre le diagramme de cas d'utilisation propres à ces fonctionnalités.

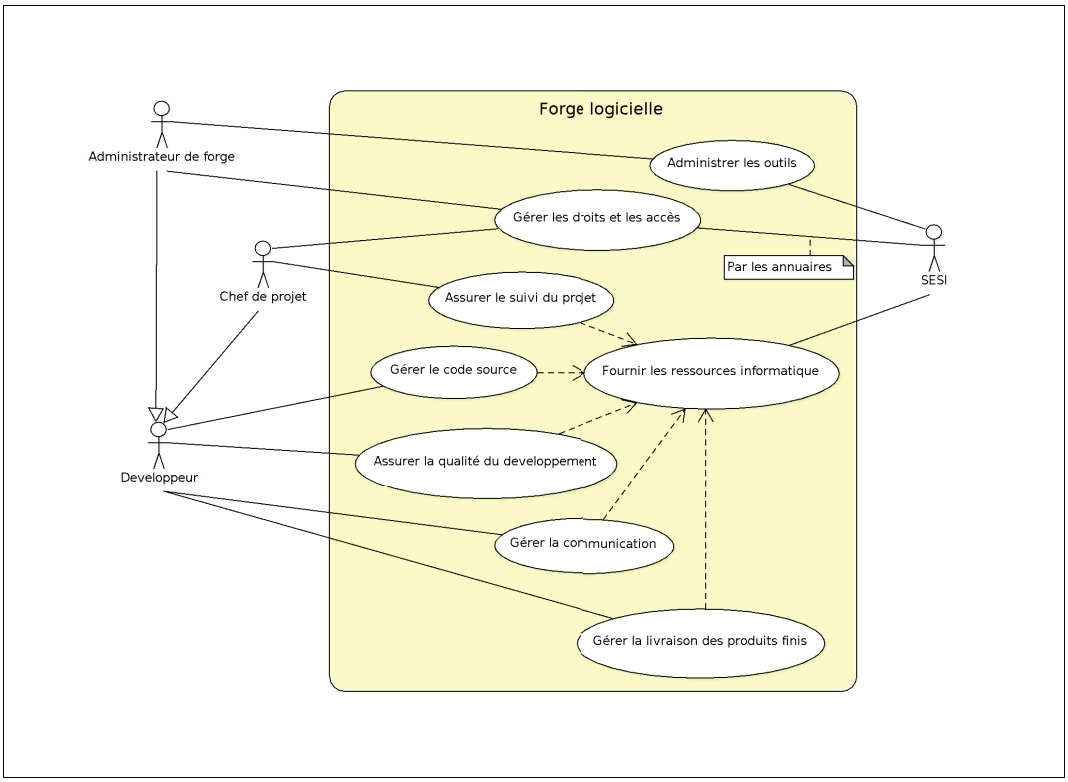


Illustration 7: Cas d'utilisation de la forge logicielle

Pour ces cas d'usages, il faut utiliser différents logiciels qui pourront être regroupés au sein d'une forge logicielle. :

- Assurer le suivi du projet :
  - Par un logiciel de suivi de projet et de *Bugtracking*.
- Gérer l'historique, le partage et la cohérence des codes sources :
  - Par un logiciel de gestion de versions.
- Assurer la qualité et la non-régression des produits livrés :
  - Par un logiciel gérant l'intégration continue.
- Gérer la communication autour du projet :
  - Par un logiciel d'échange type *Wiki.*
- Gérer la livraison et/ou le déploiement des produits finis (binaires, sites *Web*) :
  - Par un gestionnaire de dépôts logiciels.

Il faudra aussi prendre en compte les fonctionnalités suivantes qui s'appliquent à l'ensemble des outils :

- Gérer les droits, les accès et la sécurité.
- Administrer ces outils.

La forge logicielle, si elle peut apporter une certaine autonomie aux développeurs, est toujours dépendante du SESI qui doit toujours fournir les ressources informatiques dont dépendent tous ces logiciels.

III.3.2.4 *Cloud computing*

Sur le plan des ressources informatiques, on identifie dans le diagramme de cas d'utilisation de la forge logicielle (Illustration 7) les fonctionnalités pour lesquelles des ressources informatiques seront nécessaires aux développeurs. Il faut ajouter à cela le besoin de ressource pour réaliser des maquettes, dans ou en dehors de la forge. Ces ressources devront être fournies en limitant au maximum les latences liées aux demandes de mise à disposition. Les besoins en ressources retenus au COPIL (hors logiciels de la forge) sont :

- Faire des maquettes
- Effectuer des tests d'intégration
- Effectuer des tests fonctionnels
- Effectuer des tests de recette
- Héberger des présentations (démonstrateurs)

L'infrastructure mise en place doit pouvoir permettre aux équipes de développement de provisionner des ressources sans intervention des équipes d'exploitation, c'est un des points critiques de la remontée de besoins et donc un des objectifs finaux.

La diversité des OS demandés éliminant les systèmes de « conteneur » (voir chapitre III.4.4) réservés pour l'instant à des OS Linux, cette mise à disposition automatisée passe obligatoirement par un système basé sur des VM. On obtient ainsi une infrastructure de type *Cloud computing* qui est une couche d'abstraction placée au-dessus d'hyperviseurs[22]. Cette infrastructure permet la gestion des instances de machines virtuelles et des contraintes liées à la fourniture de ressources (quota, accès, cycle de vie, connectivité…).

[22]**Hyperviseurs,** serveurs physiques dédiés exclusivement à l'hébergement de machines virtuelles.

Selon le niveau d'abstraction, on parle de :

► **IAAS** (*Infrastructure as a service*) : Cette architecture *Cloud* permet de provisionner des machines virtuelles nues, en précisant ses caractéristiques (nombre de CPU, de RAM, de disques durs systèmes, d'interfaces réseaux…). Libre à l'utilisateur de l'utiliser comme il le souhaite (installer tel ou tel OS, l'allumer, l'éteindre…)

► **PAAS** (*Plateform as a service*) : Cette architecture propose des instances prédéfinies de machines virtuelles voire d'unités de travail (des conteneurs ou autres systèmes d'isolation). Selon le niveau d'abstraction de la plate-forme, l'utilisateur peut donc y déployer des applications ou directement des codes d'application. Le site *heroku.com* permet par exemple de déployer des applications *Web* dans un grand nombre de langages sans intervenir sur l'infrastructure, directement via une API. Les codes sont exécutés par des processus isolés et disponibles à partir d'URL[23] mises à disposition. L'infrastructure d'*Heroku* utilise les services IAAS d'AWS[24] (voir chapitre III.4.1.1).

► **SAAS** (*Software as a service*) : Ce niveau de *Cloud computing* propose une application qui est directement disponible, configurée par l'utilisateur mais hébergée sur l'infrastructure *Cloud* du fournisseur. Un des exemples les plus connus de plate-forme SAAS est le site « Wordpress.com », qui propose d'héberger un site web basé sur le CMS *Wordpress* directement dans leurs *Datacenters*, en seulement quelques clics.

Un deuxième aspect est à prendre en compte quand on parle de *Cloud Computing*, c'est celui de la localisation du *Hardware*[25].

On parle de ***Cloud* public** lorsque les ressources sont gérées par l'hébergeur et les services mis à disposition via internet.

On parle ensuite de ***Cloud* privé** lorsque c'est l'entreprise qui exploite l'infrastructure sous-jacente, celle-ci pouvant être hébergée, le cas échéant, chez un prestataire et accessible au travers de réseaux sécurisés de type VPN.

[23]**URL**, de l'anglais Universal resource locator, il s'agit d'une adresse référençant un emplacement sur internet.
[24]**AWS**, *Amazon Web Services*, nom des services de *Cloud* de la société *Amazon.*
[25]**Hardware**, partie matérielle d'un équipement informatique.

On parle enfin, de ***Cloud* hybride** quand un *Cloud* Privé a la capacité de « déborder » sur un *Cloud* Public pour faire face aux pics de demandes.

> *Durant la période du projet, l'IGN a mis en place un marché d'hébergement en mode Cloud. À ce titre, la D2SI était intéressée par l'analyse des capacités de Cloud hybride des solutions proposées. Les technologies du Cloud Computing sont innovantes et touchent à de nombreux domaines de connaissances aussi bien au niveau réseau qu'au niveau système. Lors de mes recherches sur le sujet, j'ai pu approfondir ces connaissances et me rendre compte de la complexité des architectures déployées dans les environnements de Cloud.*

III.3.2.5 Cloisonnement et confidentialité

Les exigences de sécurité décrites dans la PSSI de l'IGN s'appliquent à la plate-forme de développement. La confidentialité au niveau des projets de développement est aussi régulièrement exigée par nos prestataires extérieurs. Il est alors difficile d'envisager une plate-forme unique répondant à tous les besoins, notamment en termes d'accès, de confidentialité et de protection des données.

L'étude considère donc deux cas (voir illustration 8) :

- Une (ou plusieurs) plate-forme(s) « interne(s) » incluant une version mutualisée des outils mis à disposition.
- Plusieurs plates-formes « externes » où chaque projet est isolé et inclut ses propres outils.

Les différentes plates-formes devront avoir une architecture similaire pour permettre une administration simplifiée, la charge des équipes d'exploitation étant un critère important.

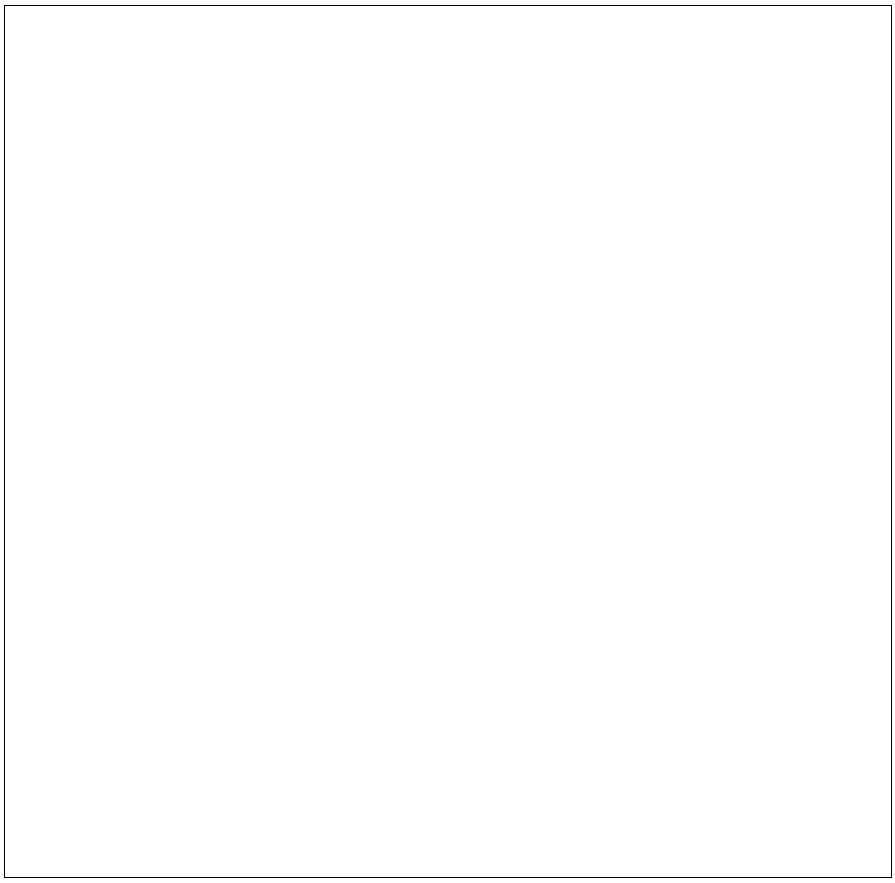


Illustration 8: Cloisonnement des forges

III.3.2.6 Dimensionnement

Afin de dimensionner la plate-forme, il est important de quantifier les différents aspects. Les indicateurs sont le nombre de VM nécessaires pour le besoin de calcul (base commune), la quantité de stockage et le nombre d'utilisateurs. Les aspects réseau, disponibilité et performance sont à prendre en compte dans un second temps.

La quantité de VM nécessaires dans la plate-forme de développement pour couvrir les besoins maximums des services est extraite de la remontée de besoins. Elle est basée à la fois sur l'existant et sur les prévisions de nouveaux projets. Les besoins sont donc exprimés en équivalence de configuration « VM de base » (ex : Une grosse VM réellement utilisée = 2 VM de base …).

Concernant les instances de machines virtuelles, la VM de base est dimensionnée sur le modèle AWS *M3.large* : (2CPU, 8G RAM, 64G DD). Le tableau suivant donne les estimations de besoin en calcul pour les différents départements.

Département	Nb VM	Charge CPU (%)	Durée d'utilisation (h)	Utilisation de calcul GPU
IDC[26]	20	75	12/24	
IDD	15	75	12/24	
IDE	20	75	12/24	
IDF	15	50	12/24	
IDG	25	50	12/24	
IDI	25	50	24/24	OUI
ISA	20	50	12/24	
ISD	20	50	12/24	OUI
DRE	30	50	12/24	OUI

Tableau 3 : Besoins en calcul

Cela donne un total estimé à 190 VM par mois avec une charge moyenne supérieure à 25 % (50 % de CPU sur 12h) avec une dizaine d'instances GPU à la demande. Il est difficile d'évaluer sur un mois la charge GPU, car cette donnée est trop dépendante des types de projets.

Concernant le stockage de données, hors sauvegarde, les départements ont estimé leurs besoins :

- IDC : 2To
- IDD : 1To
- IDE : 2To
- IDI : 10To
- ISD : 2To
- DRE : 22To

Tous les services n'ayant pas pu quantifier leurs besoins, une marge de 15 % est appliquée à cette estimation. Il faut également considérer une augmentation de 5 % par an, la quantité globale de projets n'étant pas censée augmenter fortement. On obtient donc un besoin de 50To de stockage.

[26] **IDC**, **IDE**... identifiants administratifs des départements des services SIDT, SAI et DRE.

Concernant les utilisateurs potentiels de la plate-forme, on recense dans les services un certain nombre de développeurs.

- 80 pour le SIDT
- 40 pour le SAI
- 10 en unité de production
- 70 à la DRE

En ne comptabilisant que les développeurs, on arrive à un total de 200 utilisateurs. Les autres utilisateurs (chefs de services, « clients » …) n'ayant accès qu'à une petite partie de la plate-forme essentiellement au travers de portails web, ils ne sont pas pris en compte.

Ces différentes estimations permettent d'établir une fourchette haute pour le dimensionnement des offres de *Cloud* qui font l'objet d'une analyse dans les chapitres suivants.

III.4 État de l'art

La littérature papier spécifique à la problématique de plates-formes de développement dans son ensemble n'est pas très importante. Elle se limite souvent à l'un ou l'autre des aspects, soit l'utilisation des outils, soit la mise en place de méthodes de gestion de mise en production. L'objectif de la plate-forme n'étant pas le choix des outils mais le moyen de les mettre à disposition, il n'a pas été fait de critique des logiciels choisis, les développeurs étant souvent les mieux placés pour identifier ce qui est adapté à leurs besoins.

L'état de l'art est donc principalement basé sur, d'une part, les offres des grands acteurs de l'informatique que l'on trouve facilement sur internet et d'autre part, sur les expériences réussies des équipes du projet *G-Cloud*.

L'analyse fonctionnelle du besoin a remonté deux axes de travail, le premier sur la fourniture automatique de ressources, le deuxième sur l'automatisation de la création de forges logicielles. Certaines solutions semblent remplir les deux fonctionnalités, elles seront donc évaluées en premier lieu. Le deuxième axe de recherche peut être dépendant du premier, il est donc primordial de trouver une solution qui permette de les instancier indépendamment l'une de l'autre.

III.4.1 Les offres en ligne

On trouve des offres de forges logicielles « prêtes à l'emploi » sur Internet, mais ces dernières ne remplissent qu'une partie des fonctionnalités qui sont attendues. Au travers de sites comme *sourceforge.net* ou *github.com* par exemple, il est même possible d'héberger gratuitement des projets qui seraient distribués sous une licence Open source. Le site *heroku.com* cité en exemple au chapitre III.3.2.4 entre également dans cette catégorie de forge logicielle PAAS.

III.4.1.1 AWS

Les offres en ligne *d'Amazon* sont différentes et seraient potentiellement plus adaptées en couvrant à la fois le besoin d'outils et le besoin de ressources. En effet, la plate-forme AWS propose une quantité importante de micro-services permettant de construire des applications *Cloud* totalement automatisées, incluant toutes les briques nécessaires à la mise en place d'une forge complète quel que soit le type de projet. Il manque cependant un portail de services simplifié, l'utilisation de la plate-forme étant tout de même plus adapté pour des administrateurs ou des développeurs *Devops*.

Avec une tarification « à l'usage », le géant américain, leader mondial des offres de *Cloud computing*, propose une offre alléchante, mais son statut d'entreprise américaine et sa situation géographique en Europe (Irlande et Allemagne) rend ce service incompatible avec la PSSI de l'État pour la gestion de certains de nos projets informatiques.

III.4.1.2 *Cloud* public

Les opérateurs de *Cloud* français comme *Numergy* ou *Cloudwatt* proposent des offres plus bas niveau, principalement IAAS. Ce type d'offre est étudié dans le cadre d'un marché d'hébergement lancé par le D2SI de l'IGN. La plate-forme de développement pourra donc potentiellement en profiter.

Si certains projets, notamment concernant la recherche, pourraient être adaptés à ces offres en ligne, ceci n'est pas envisageable pour la totalité des projets. La position de l'IGN en tant qu'OIV lui impose de ne prendre aucun risque sur l'hébergement de ses données.

En matière de sécurité, l'utilisation d'outils publics peut poser des problèmes. Même si la sécurité propre à l'hébergeur est démontrée et évaluée comme acceptable, certains exemples de cybercriminalité nous obligent à prendre ce genre de solutions avec prudence. (voir exemple en annexe 5)

Une solution unique étant recherchée pour des raisons de simplification, l'étude de ces solutions en ligne n'est pas approfondie.

III.4.2 Les offres Open source

Comme l'ont montré les retours d'expériences du groupe de travail *G-Cloud*, *OpenStack* est la solution incontournable d'une mise en place de *Cloud* privé en entreprise à l'aide d'outils Open source. En effet le seul réel concurrent sur ce domaine est la solution *Cloudstack* de la *Apache software foundation* mais la communauté autour de *Cloustack* est beaucoup moins développée que celle *d'OpenStack.* Il est également beaucoup plus difficile de trouver des distributions « Clé en main »de *Cloudstack.* Dans cette mesure, seules les offres basées sur *OpenStack* sont évaluées.

OpenStack est un projet Open source assez jeune, débuté en 2010 avec une première version de production sortie en 2012.

III.4.2.1 OpenStack

Illustration 9: Logo OpenStack

OpenStack permet de déployer des infrastructures de type *Cloud* en entreprise. Sa fonction principale est donc de fournir à un utilisateur non spécialiste des ressources

informatiques à la demande. Ces ressources se matérialisent par des VM instanciées et gérées par l'outil. *OpenStack* fournit bien sûr également la partie administration propre à sa mécanique interne.

De nombreuses grandes sociétés (HP, IBM, Intel…) participent à son développement. Le nombre de sociétés impliquées dans le développement *d'OpenStack* est impressionnant (voir annexe 6).

Le CERN[27] est un des meilleurs exemples de l'utilisation en production de *Cloud OpenStack*. Pour traiter les données du Large Hadron Collider (LHC), le CERN utilise 4 *Cloud OpenStack* dont le plus gros contient 70 000 cœurs répartis sur plus de 3000 serveurs. Plus d'1 000 000 de VM ont ainsi été créées depuis sa mise en service avec un taux de création/effacement de 100 à 200 VM par heure, tout cela sans interruption avec un accroissement constant du nombre de serveurs disponibles.

Certains fournisseurs de *Cloud* publics comme *Numergy* utilisent également *OpenStack* comme architecture pour fournir leurs VM en mode IAAS.

OpenStack se présente sous la forme d'un ensemble de modules lui permettant de fournir principalement un service de type IAAS mais peut évoluer vers des services PAAS voire SAAS avec des catalogues d'applications comme le module *Murano* par exemple.

Le problème principal *d'OpenStack* pour sa mise en œuvre est sa complexité. Un *Cloud OpenStack* est ainsi constitué de nœuds correspondants à des machines physiques (éventuellement virtuelles) reliées entre elles par plusieurs réseaux qui seront dédiés à des actions spécifiques. Chaque nœud contient un système *Linux* et se voit attribuer un ou plusieurs rôles. Selon les fonctionnalités recherchées, il faut coordonner ces rôles et configurer correctement les différents nœuds. Une installation « *from scratch*[28] » *d'OpenStack* demande donc une importante préparation et une très bonne connaissance de l'architecture et des interactions entre les modules *OpenStack*. Cela demande beaucoup de temps, en cas de problème de configuration, les « *logs* » étant conséquents, il faut savoir faire les bonnes analyses pour cibler la recherche de solution.

L'architecture de base d'un *Cloud OpenStack* est donnée en annexe 7. Sans avoir de serveurs physiques à disposition à cette étape du projet, il est toutefois possible de réaliser une première installation d'un *Cloud OpenStack* sur une infrastructure virtualisée dans un *Cluster*

[27]**CERN**, acronyme de « Conseil européen pour la recherche nucléaire », laboratoire européen de recherche sur la physique des particules.
[28]**From scratch**, expression signifiant « en partant de rien ».

XenServer de la DRE. Cette installation, bien que fonctionnelle, se trouve très limitée au niveau des capacités réseaux offertes par *OpenStack*, certains modules n'étant pas compatibles avec le réseau virtuel proposé par le *Cluster XenServer*. La configuration des différents modules qui composent *OpenStack* demande une grande attention pour comprendre les interactions entre l'ensemble des modules mis en œuvres. Forts de ce constat, de nombreux éditeurs ont mis à disposition des outils pour l'installation et l'administration des infrastructures *OpenStack*. Ces outils permettent l'automatisation d'installations *OpenStack* mais brident inévitablement les capacités de configurations proposées nativement par *OpenStack*.

On peut par exemple citer *DevStack.* Cet outil, fourni par la communauté *OpenStack* permet de tester *OpenStack* sur une seule et même VM. La distribution *OpenStack* proposée par *Mirantis* est facilement configurable avec un outil de déploiement très paramétrable. HP fournit également une distribution automatisée *d'OpenStack* gratuite appelée *Helion Community*.

Si ces distributions permettent de mettre en place des *Cloud OpenStack* opérationnels, il faut bien comprendre que, comme tout système informatique complexe, elles n'empêcheront pas les dysfonctionnements de ces derniers. Une expertise sera donc toujours nécessaire, que ce soit au niveau de l'entreprise ou sous forme de support éditeur pour maintenir ces systèmes en production.

Il faut ensuite se demander si le service rendu par un *Cloud OpenStack* correctement configuré est cohérent par rapport aux besoins émis lors de l'analyse fonctionnelle.

Du point de vue de l'utilisateur, *OpenStack* se présente comme un portail *Web* ou un ensemble d'API permettant la mise à disposition de machines virtuelles selon des modèles pré-établis par l'administrateur du *Cloud*. Si ce portail est assez ergonomique, il est tout de même basé sur des notions plus proches du métier d'administrateur que de celui de développeur.

Il est donc nécessaire d'intercaler une abstraction supplémentaire de type PAAS pour donner l'accès directement aux développeurs.

Ce constat pose un problème, car cette couche intermédiaire est le plus souvent dépendante du type de développement effectué. Il est donc très difficile de trouver un produit qui puisse convenir à toutes les équipes.

III.4.3 Les offres éditeurs

En se basant sur les expériences réussies des équipes participant au groupe de travail *G-Cloud*, on identifie plusieurs éditeurs proposant des solutions qui peuvent répondre à nos besoins. Ces derniers sont donc contactés afin d'étudier leurs offres au niveau de l'automatisation de la fourniture de ressources informatiques.

III.4.3.1 VMWare

L'IGN possède un *cluster* de virtualisation sous *VMWare*, cette société est donc logiquement contactée pour avoir des informations sur les solutions qu'ils proposent. Les retours d'expériences montrent qu'il est indispensable de séparer le *Cluster* de virtualisation des systèmes en production de celui qui serait dédié à une offre *Cloud*. En effet, pour éviter d'introduire des aléas dans le C*luster* dédié à l'infrastructure *Cloud*, il est préférable de n'intervenir sur ce dernier qu'au travers des outils de gestion de *Cloud* et donc d'avoir un C*luster* dédié à cet usage. Il y a donc peu d'intérêt à favoriser cette technologie par capitalisation de l'existant. L'offre proposée par *VMWare* permet la mise en place d'un portail de service pilotant des hyperviseurs via *vSphere*, l'outil de virtualisation phare de l'éditeur.

> *VMWare a mis beaucoup de temps à répondre à mes sollicitations et ne m'a pas proposé d'offre commerciale chiffrée. Les tarifs trouvés sur internet montrent que les solutions Cloud de VMWare sont assez coûteuses étant donné la tarification basée sur le nombre de sockets processeur et les dépendances nécessaires au niveau de la gestion du réseau (SDN[29] NSX). Depuis la fin de l'étude, VMWare à également mis sur le marché une distribution d'OpenStack adaptée à ses outils de virtualisation.*
>
> *J'ai intégré la solution VMWare dans mon étude de coûts (voir chapitre III.6) en me basant sur les tarifs publics sans prendre en compte de prestation d'accompagnement n'ayant pas eu de devis en temps voulu.*

[29]**SDN**, de l'anglais *Software Defined Network* , virtualisation du réseau. NSX est l'outil SDN de VMWare.

III.4.3.2 HP

La société HP France proposent de nombreux produit autour de leurs offres de *Cloud Computing*. Deux de leurs produits sont intéressants dans le contexte du projet de plate-forme de développement.

Le premier se base sur CSA[30], c'est une solution éprouvée chez HP depuis de nombreuses années. Il s'agit d'un portail de service construit autour d'un outil d'automatisation appelé OO[31]. Cet outil est très générique et permet de s'adapter à des infrastructures déjà en place dans l'entreprise. Son rôle est de prendre en compte les différentes briques du SI, puis de les intégrer dans le système via des API ou des *drivers* développés spécifiquement. Un « designer de service » permet ensuite de générer graphiquement des services utilisant les briques du système. Ces services sont ensuite proposés au travers d'un portail W*eb* au design moderne et adaptatif. Dans le cas où nous ne souhaiterions pas réutiliser nos propres *Cluster* de virtualisation (sous *VMWare*), HP propose également sa propre implémentation *d'OpenStack* nommée *HP Helion OpenStack* pour fournir les ressources informatiques.

Ce produit est très séduisant sur le papier mais pose d'importants problèmes d'intégration. En effet, la mise en place d'un tel outil suppose que le SI de l'entreprise soit déjà équipé des briques indispensables au fonctionnement autonome de l'outil (CMDB[32], IPAM[33], DDNS[34]…). Ces briques existent dans le SI de l'IGN sous forme artisanale et il n'y a pas d'API utilisable par un outil tel qu'OO. La mise en place d'une telle solution demande donc une mise à niveau de toute une partie du SI de l'IGN, ce qui n'est pas envisageable dans le cadre de ce projet.

Le deuxième produit proposé par la firme est nommé *HP Helion Development Platform*. Comme son nom l'indique, il s'agit d'un outil basé sur la version HP *d'OpenStack* qui supporte une plate-forme *CloudFoundry*.

CloudFoundry est un logiciel de *Cloud* PAAS assurant le cycle de vie de la phase de développement d'une application. Grâce à une interface en ligne de commande ou via un *plugin*[35] intégré dans leur Environnement de développement intégré (EDI), les développeurs peuvent « pousser[36] » leurs codes dans la plate-forme. À partir de ce moment-là, les

[30]**CSA**, « *Cloud service automation ».*
[31]**OO,** « Operation orchestratot ».
[32]**CMDB**, « Configuration management database ».
[33]**IPAM**, « IP adress management ».
[34]**DDNS**, « Dynamic domaine name service ».
[35]**Plugins,** extensions permettant d'ajouter des fonctionnalités à un logiciel.
[36]**Pousser** est utilisé ici par analogie au verbe « to push » utilisé dans la commande permettant d'effectuer cette action.

ressources nécessaires sont provisionnées (serveur frontal, base de données, « *load balancer*[37] », dépendances…), le code est éventuellement compilé, les tests d'intégration continue sont effectués et la version ainsi préparée est mise à disposition. L'objectif de l'outil est donc de faire en sorte que le développeur ne soit concentré que sur son code et sur le résultat, toute la partie intermédiaire n'étant pas de son ressort, elle est donc totalement automatisée.

Cet outil qui est donc une version sur site (intégrée dans le *Datacenter* de l'entreprise) des solutions *Web* comme *Heroku* ou encore *Google App Engine* est donc parfaitement adapté au développement logiciel tel qu'il est effectué dans certains services à l'IGN. La plupart des langages récents sont pris en charge par la plate-forme ainsi que les bases de données les plus courantes (SQL[38] ou noSQL).

Les limites de cette solution apparaissent suite à la question d'un développeur lors de la présentation d'HP dans nos locaux. En effet, à la question demandant ce qu'il fallait faire pour intégrer dans l'outil la version géographique (*PostGIS*) de la base de donnée *PostgreSQL*, la réponse d'HP est très claire :

- *"C'est impossible sans faire réaliser un développement d'intégration par HP."*

Cette limite est évidement rédhibitoire pour le projet. Si certains développements peuvent être demandés à HP pour les principaux outils spécifiques utilisés par nos développeurs, certains services comme le SAI ont besoin en permanence de nouveaux outils spécifiques, il n'est donc pas concevable d'utiliser ce type de plate-forme dans un cadre aussi hétérogène et évolutif.

III.4.3.3 Mirantis

Mirantis est une société américaine, précurseur de l'industrialisation *d'OpenStack*. La politique de *Mirantis* est de proposer une distribution *d'OpenStack* « *Zero Lock-in* » c'est-à-dire sans contraintes constructeurs ou propriétaires. *Mirantis* s'appuie sur le module *Fuel* qui permet la gestion et la fourniture des serveurs ainsi que le déploiement *d'OpenStack* sur ces derniers à partir d'une interface web dédiée offrant de nombreuses options de paramétrages.

Les détails sur la mise en place de *Cloud OpenStack* à partir de *Fuel* sont fournis en Annexe 8.

[37] **Load Balancer**, outils d'équilibrage de charge.
[38] **SQL**, de l'anglais « *Structured query language* », langage d'exploitation de base de donnée.

L'implémentation de l'outil *Fuel* proposée par *Mirantis* facilite grandement la mise en place et la maintenance des infrastructures *Cloud OpenStack* et reste totalement gratuite. *Mirantis* propose donc une prestation d'accompagnement et de support sur son outil. Cet accompagnement devient indispensable lors de la montée de version *d'OpenStack* qui selon les retours utilisateurs, ne se passe généralement pas sans problème. *Mirantis* fournit également des prestations de développements et de conseils concernant des modules supplémentaires ou permettant d'offrir de nouvelles fonctionnalités à *OpenStack.* Ils proposent aussi d'accompagner les ingénieurs de l'entreprise dans la mise au point de « *drivers* » permettant de prendre en compte des éléments utilisés dans l'entreprise et non pris en charge nativement par *OpenStack* ou par *Fuel.*

L'intégration des modules *OpenStack* par l'outil de *Mirantis* restreint cependant les fonctionnalités à la plate-forme IAAS via Horizon. Comme nous l'avons vu, ce type d'accès n'est pas spécialement adapté aux développeurs mais plutôt à des administrateurs. Une solution de type *Mirantis OpenStack* se limite donc dans un premier temps à la fourniture de ressources en autonomie, la partie forge logicielle devant être assurée par une autre solution. Cette autre solution peut en revanche tout à fait se reposer sur les ressources fournies par le *Cloud OpenStack*.

> *La solution de Mirantis est celle retenue par le COPIL pour la réalisation d'une maquette (voir chapitre III.5.1).*

III.4.4 Forge logicielle

La première partie de l'étude montre qu'une plate-forme de développement PAAS automatisée n'est pas adaptée à la diversité de nos développements mais aussi à l'état d'avancement de l'urbanisation de notre SI. Il faut donc tenter de fournir les outils que les développeurs ont l'habitude d'utiliser sous la forme d'un ensemble pré-configuré et disponible rapidement.

Le premier COPIL a permis de définir les logiciels que les développeurs devaient utiliser prioritairement dans un souci d'harmonisation. Il a également été décidé qu'en fonction des contraintes particulières, d'autres outils peuvent être utilisés mais avec une maintenance non garantie par les équipes de l'exploitation. Le système mis en place doit donc permettre l'ajout de nouveaux outils.

Les outils retenus et la plupart des autres outils liés à cette problématique sont des outils *Web* et sont donc installés sur des serveurs *Linux*. Ce point commun permet d'envisager une mutualisation des ressources. Cette mutualisation permet en effet de fournir une machine virtuelle unique contenant l'ensemble des outils nécessaires.

Certains de ces outils peuvent être installés directement depuis les dépôts officiels de certaines distributions *Linux*, mais ce n'est pas le cas de tous. De plus il y a fréquemment des incompatibilités, ces outils étant tous basés sur les mêmes protocoles du *Web* (par exemple, l'utilisation des ports 80 et 443 pour le protocole HTTP). Il n'est donc pas envisageable de tous les installer sur un système d'exploitation unique.

Pour isoler ces outils, tout en limitant les ressources, il faut faire appel à des technologies de mutualisation comme les conteneurs *Linux*. Cette technologie permet d'isoler un ensemble de processus dans un espace utilisateur dédié. L'isolation est effectuée par des mécanismes propres aux noyaux *Linux*. Ce n'est pas vraiment de la virtualisation car l'ensemble des conteneurs partagent le même noyau mais chaque outil dispose de ses propres processus sans avoir connaissance des autres conteneurs et sans possibilité d'y avoir accès.

Plusieurs technologies de conteneurs *Linux* existent, parmi lesquelles *OpenVZ* et *LXC*[39]. Si ces derniers offrent de bonnes performances, ils restent peu accessibles à des utilisateurs non spécialistes de l'administration système *Linux*. Ce point est donc négatif pour la capacité d'évolution du système.

Une autre technologie de conteneurs *Linux* est apparue récemment (2013) et a très rapidement eu beaucoup de succès au point de devenir incontournable dans toutes les offres de *Cloud* public, il s'agit du système *Docker*.

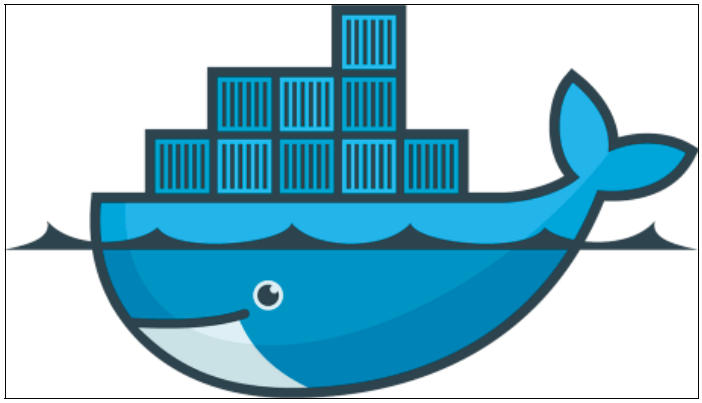


Illustration 10: Logo Docker

[39]**LXC**, de l'anglais « Linux Container »

Docker permet de construire des conteneurs *Linux* autosuffisants et légers, qui peuvent être installés sur différentes cibles de manière identique. Contrairement aux autres solutions de conteneurs *Linux*, *Docker* met en avant des services annexes qui font sa notoriété comme le langage de construction d'image ainsi que le principe de dépôt de ces images.

Une autre force de *Docker* est d'utiliser des systèmes de fichier (AUFS[40] par exemple) lui permettant de se baser sur un agglomérat de strates dont seule la dernière couche est inscriptible (voir illustration 11), les autres étant de ce fait partageables entre les différents conteneurs.

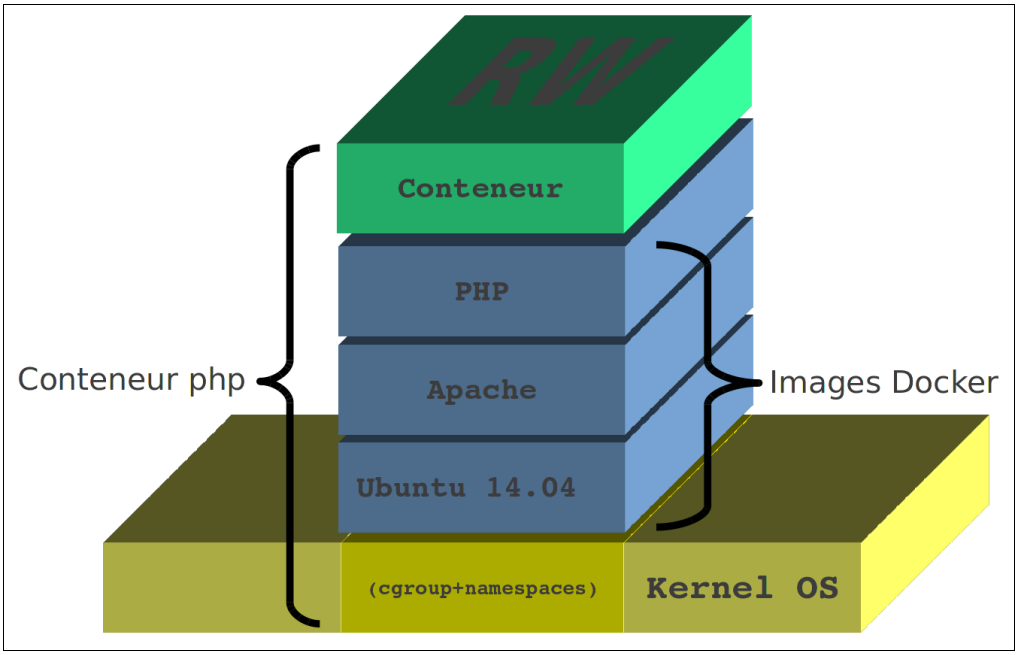


Illustration 11: Principe des couches de Docker

Pour exécuter un conteneur, il faut construire une image contenant les éléments nécessaires à l'application que l'on souhaite faire tourner. *Docker* propose également un service appelé *DockerHub* sur lequel les utilisateurs peuvent déposer leurs images pour que la communauté puisse les réutiliser. *Docker* propose aussi ses propres images « officielles » des applications les plus utilisées.

Une fois l'image créée et/ou récupérée, on instancie un conteneur en ajoutant une couche inscriptible au-dessus de l'image, si l'image intègre un *Shell*[41] (c'est un minimum) on peut s'y connecter lors de l'instanciation (ou plus tard) pour interagir avec l'image. La

[40] **AUFS**, de l'anglais « *Another union file system* ».
[41] **Shell**, désigne une interface utilisateur du système d'exploitation.

commande d'initialisation d'un conteneur propose de nombreuses options pour partager des dossiers ou des services entre le conteneur et la machine hôte ou entre plusieurs conteneurs.

L'intérêt de *Docker* est de proposer un empaquetage léger, réutilisable et pré-configuré d'une application pour pouvoir la déployer à l'identique sur n'importe quel système *Linux,* que ce soit un serveur, un poste de travail ou une instance de *Cloud*. *Docker* apporte une grande souplesse d'utilisation par rapport à des machines virtuelles classiques avec une disponibilité immédiate. En effet, l'OS est déjà démarré, et les conteneurs partagent le même noyau.

En contrepartie, l'isolation entre les applications « conteneurisées » est plus basique mais reste suffisante notamment dans un contexte de développement. D'autre part l'outil n'est disponible que pour *Linux. Microsoft*, en partenariat avec *Docker* travaille sur une version pour *Windows* afin de conteneuriser des applications *Windows*. En attendant, une solution appelée *Boot2docker* permet de lancer des conteneurs *Docker* depuis un environnement *Windows* (dans une VM *Linux* minimaliste).

D'autres sociétés proposent des alternatives à *Docker* comme *Rocket* de *CoreOS* et *Ubuntu Core* proposé par *Canonical*. Ces solutions se posent en concurrentes de *Docker.* Elles manquent aujourd'hui de maturité et n'offrent pas toutes les fonctionnalités proposées par *Docker.*

> *Dans la cadre du projet, le système de forge basé sur Docker décrit au chapitre III.5.2 a donc permis de créer des images pré-configurées des outils retenus tout en laissant la possibilité aux développeurs d'utiliser d'autres outils. Ces images ont ensuite été déployées sur des machines virtuelles en les regroupant selon la nature des projets (forges internes, forges externes...).*

III.5 Maquettes

Le premier COPIL a validé la réalisation des deux maquettes. La première est basée sur la version de *Mirantis d'OpenStack*, en évaluant la possibilité de réutiliser les compétences du SESI concernant le système de virtualisation *VMWare,* ainsi que la possibilité de mettre en place un système de *Cloud* Hybride. La seconde maquette est un prototype de forge basée sur le système de conteneur *Docker.*

L'analyse des besoins et des solutions m'a permis de présenter une étude préalable complète au COPIL afin de pouvoir orienter le projet vers la réalisation des maquettes sous forme de POC aussi bien en ce qui concerne la fourniture de ressource que d'un système de forge automatisé.

III.5.1 Infrastructure *Cloud*

Le budget alloué pour ces maquettes a été estimé au début du projet et cette solution correspondait à l'enveloppe disponible pour la mise en place de l'avant-projet. Après plusieurs contacts avec les ingénieurs de Mirantis, j'ai réalisé les spécifications pour commander le matériel nécessaire. Les demandes d'achats correspondantes ont été fournies par l'intermédiaire de mon tuteur aux services d'achat de l'IGN.

La charge de travail du service achat à la fin de l'année 2014 n'a pas permis, malgré les relances, de réaliser les achats du matériel pour la maquette avant la date limite pour l'année 2014. L'achat a donc été reporté sur le budget 2015. Dans ce contexte, la livraison du matériel n'était pas envisageable avant la fin du mois de janvier 2015. J'ai donc décidé de commencer l'étude de la solution technique de forge pour paralléliser le travail.

Pendant cette phase transitoire d'attente du matériel, j'ai rédigé la documentation technique de mise en place de l'infrastructure OpenStack basée sur la distribution Mirantis. La pré-maquette, réalisée sur le Cluster de virtualisation, utilisée pour faire des démonstrations, notamment lors du premier COPIL, m'a permis d'appréhender l'architecture générale et de prévoir les points de blocages, principalement au niveau réseau.

Les problèmes de budget se sont accentués au début de l'année 2015. Les commandes de matériels prenant du retard, j'ai donc dû trouver des solutions pour obtenir le matériel nécessaire à la réalisation des maquettes.

Après avoir exploré sans succès, les solutions de location, j'ai négocié le prêt de matériels en stock avec le service de l'exploitation pour une durée déterminée avec une restitution du matériel avant l'été 2015.

Le matériel potentiellement disponible, n'étant pas homogène et moins performant que ce qui était prévu dans le cahier des charges, il a fallu adapter les configurations et prévoir certaines limitations dans les expérimentations prévues.

Malgré mes relances, les techniciens n'ont pas pu mettre en œuvre l'infrastructure dans le temps imparti pour pouvoir présenter les résultats de la première maquette lors du deuxième COPIL.

Ce retard pour la mise en place d'une maquette sur du matériel de récupération n'a pas eu une énorme incidence sur le projet, mis à part la perte de montée en compétence obtenue par la collaboration avec les experts de chez Mirantis. Suite à la présentation de l'étude de coûts sur les solutions Cloud envisageables, il a été décidé au COPIL de repousser à 2016 cette partie du projet principalement par manque de moyen humain pour prendre en charge la suite du projet. On peut se rendre compte de l'impact de ce retard sur le planning du projet réalisé présenté en annexe 9.

La maquette en conditions réelles n'a donc pas pu être réalisée, cependant la documentation d'architecture et d'installation de la pré-maquette réalisée sur le cluster de virtualisation VMWare permet d'appréhender correctement la problématique de mise en place d'un tel produit.

III.5.2 Forge

La seconde maquette demandée concerne le système de mise à disposition de forge sous *Docker*. Les objectifs de cette maquette sont les suivants :

- Montrer que l'on peut déployer les applications retenues dans des délais acceptables.
- Toujours pour des raisons de délais, montrer que l'on peut séparer la partie « exploitation de l'infrastructure » réalisée par le SESI de la partie « exploitation des applications » réalisée par les développeurs.
- Montrer que l'on peut capitaliser sur les configurations des outils pour pouvoir les réutiliser sans repasser par cette étape.
- Montrer que la solution respecte les contraintes imposées.

III.5.2.1 Avantages de la solution *Docker*

Les avantages de la solution *Docker* sont nombreux :

- Maintenabilité
  - Maîtrise des outils via un catalogue de conteneurs dans l'entreprise (*Registry*).
  - Économie de VM (1 VM pour l'ensemble des outils d'un projet).
  - Économie d'IP (1 IP pour l'ensemble + alias DNS).
  - Interaction entre les outils. En effet, certaines fonctionnalités ne sont possibles que si les outils partagent la même machine. Ceci est rendu possible avec les mécanismes de communication entre les conteneurs.
- Ergonomie
  - Simplicité de prise en main, *Docker* étant pensé pour une utilisation par des développeurs n'ayant pas forcément des compétences système poussées.
- Évolutivité
  - Possibilité d'avoir une VM avec les outils pré-installés que l'on démarre au besoin.
  - Possibilité d'utiliser d'autres outils conteneurisés toujours dans la même VM.
  - Réutilisation de *Docker* et des conteneurs sur les machines des développeurs pour test et évolution des outils.
  - Utilisation comme « hyperviseur ».
  - Utilisation massive de la technologie par tous les grands acteurs du développement (*Google*, *Ebay*…) est un gage de confiance.

III.5.2.2 Inconvénients

Malgré les nombreux avantages de la solution *Docker*, il existe tout de même quelques inconvénients :

- Technologie récente avec peu de recul (v1.5 à la date de la maquette).
- Isolation plus faible que dans la virtualisation classique.
- Limitation aux systèmes basés sur le noyau *Linux*.
- Formation nécessaire des agents en charge de l'infrastructure et des développeurs.

III.5.2.3 Prise en main de l'outil *Docker*

La prise en main de l'outil est assez rapide, le site de Docker propose de nombreux exemples. Le fonctionnement de *Docker* nécessite un noyau *Linux* 3.8+ qui n'est pas disponible dans la version de production de *Debian* 7 utilisée à l'IGN. Il faut donc utiliser le noyau de la version *Debian 8*, disponible dans les « *Backports*[42] » et utilisable facilement.

L'utilisation d'un *Proxy* d'entreprise pour le trafic *Web* pose fréquemment des problèmes car il faut propager les réglages de *Proxy* à chaque nouveau conteneur et l'adapter aux scripts internes qui sont souvent pensés sans l'utilisation d'un *Proxy*.

La partie la plus complexe de la prise en main concerne la compréhension de la gestion du réseau et des échanges entre les conteneurs, la documentation étant parfois contradictoire. Pour prévoir l'utilisation en DMZ[43] de cette solution, il faut également bien appréhender la partie sécurité liée aux partages entre les conteneurs et la machine hôte.

III.5.2.4 Échange avec les développeurs

Pour rappel, la liste des applications à déployer est la suivante :

- **Gitlab** : un gestionnaire de dépôt *Git*[44].
- **Redmine** : un gestionnaire de projets et de tickets.
- **Dokuwiki** : un moteur de Wiki/forum/blog.
- **Jenkins** : un outil d'intégration continue.

Les configurations communes et minimales de ces applications sont définies en concertation avec les développeurs, responsables d'outils similaires utilisés dans leurs services. Il est décidé de ne pas intégrer de *plugins* dans les images de base, celles-ci étant destinées au plus grand nombre. La gestion des *plugins* reste donc à la charge des développeurs, administrateurs de forge.

[42]**Backports,** dépôts logiciels de la distribution *Linux Debian* contenant des paquets de la prochaine version stable.
[43]**DMZ**, de l'anglais « *Demilitarized zone* », zone réseau isolée par un pare-feu souvent destinée à être accédée depuis Internet.
[44]**Git**, logiciel de gestion de version de code décentralisé, initialement créé pour le noyau *Linux*.

III.5.2.5 Constitution des images *Docker* pré-configurées

Docker propose un système de construction d'image avec une syntaxe très proche du *Shell Linux* dans des fichiers appelés « *Dockerfiles* » (voir exemple en annexe 10). Ce système génère un conteneur temporaire à chaque instruction du fichier, il est de ce fait très efficace, la modification d'une partie du script n'impliquant pas le calcul de la totalité de l'image. En s'inspirant de versions disponibles sur le dépôt public de *Docker* (*https://dockerhub.docker.com*) on peut créer les images pour les différents outils en incluant les réglages propres à l'infrastructure de l'entreprise (serveurs DNS, *Proxy*, authentification des utilisateurs…). Le concept de « TAG[45] » des images *Docker* permet de définir plusieurs images similaires avec des configurations légèrement différentes. On peut avoir ainsi une image de chaque logiciel adaptée aux forges dites « externes » avec des systèmes d'authentification différents par exemple.

III.5.2.6 Création du dépôt d'entreprise

Pour stocker ces images et les rendre accessibles à tous les utilisateurs de l'entreprise, il est nécessaire de mettre en place un dépôt d'entreprise. *Docker* fournit un conteneur contenant le logiciel de stockage et de diffusion d'image pour réaliser cette tâche, il suffit donc de l'instancier sur une machine désignée comme référence pour le dépôt d'image *Docker*.

Pour différencier les images locales des images distantes, le nom de l'image contient la localisation de cette dernière. Si le port (TCP) de destination est précisé, *Docker* cherche une destination locale (*Registry*) dans le cas contraire, il cherche l'image dans le dépôt public de *Docker* (*Dockerhub*). L'exemple suivant est le résultat de la commande docker image qui liste les images présente sur la machine hôte. On peut y voir une image qui vient du dépôt d'entreprise « dockerforge.ign.fr » (dockerforge:5000), une image officielle (java) et une image venant d'un dépôt situé sur le *Dockerhub* (dockerui).

```
rsi@dockerforge:~$ docker images

REPOSITORY                TAG            IMAGE ID       CREATED         VIRTUAL SIZE
dockerforge:5000/jenkins  latest         9bdbc4f2ca13   1 months ago    887.3 MB
java                      openjdk-8-jdk  433801eb0894   1 months ago    816.4 MB
dockerui/dockerui         latest         ee8244dbbfd3   3 months ago    443.4 MB
...
```

[45]**TAG**, littéralement « étiquette ». Dans Docker, le TAG permet de différencier des images similaires par leur nom.

Le registre d'entreprise peut également fonctionner comme un *Proxy*, en mettant en « cache » les images téléchargées depuis le dépôt public de *Docker*. Ce mécanisme permet d'améliorer le temps de téléchargement des images sur les postes internes à l'entreprise.

Le diagramme 12 indique le processus de la construction d'image et d'utilisation du dépôt d'entreprise.

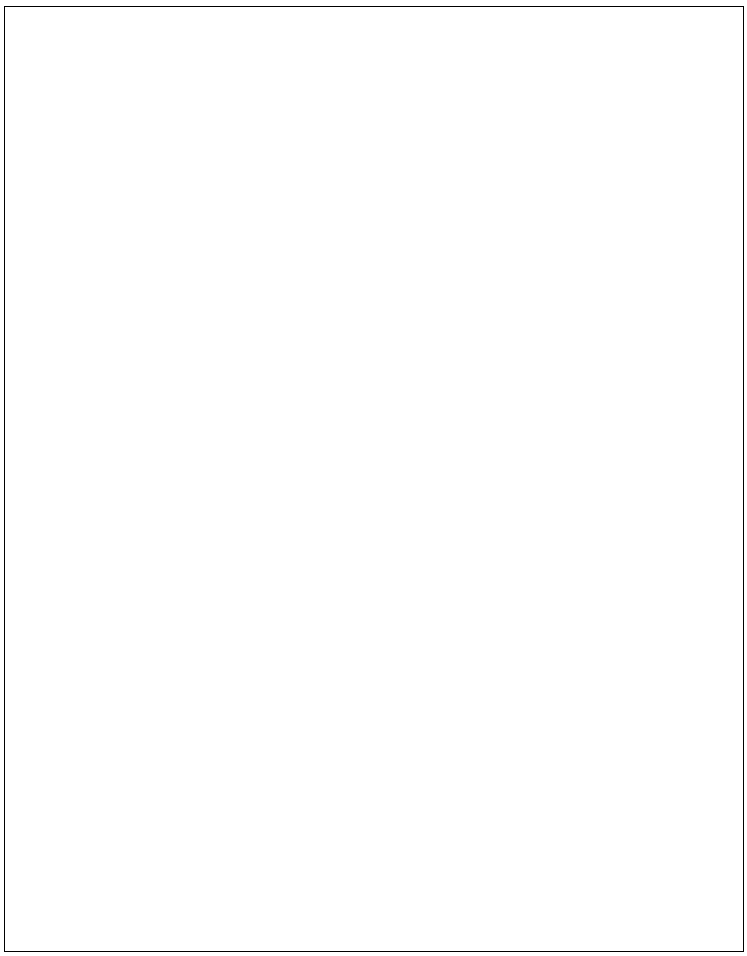


Illustration 12: Construction des images Docker

III.5.2.7 Préparation d'un modèle de machine virtuelle

Pour supporter ces images *Docker,* il faut préparer un modèle de machine virtuelle contenant *Docker* et faire en sorte que la forge puisse être déployée automatiquement par les administrateurs du département d'exploitation. C'est une version très simple de *Debian* conforme aux installations préconisées par le service d'exploitation de l'IGN. Ce modèle est créé dans un des *Clusters VMWare* de production avec pour objectif, à terme d'être migré vers

le système de fourniture de ressource propre à la plate-forme de développement. Ce modèle doit permettre la séparation des actions d'administrations en déléguant l'administration des outils déployés dans la forge aux développeurs administrateurs de forge (voir diagramme 13).

Illustration 13: Séparation de l'administration des forges

Les conteneurs *Dockers* des applications préparées ne sont pas inclus directement dans ce modèle. Pour avoir des logiciels à jour, il est préférable d'utiliser un script de « post-installation[46] ».

III.5.2.8 Création des scripts de déploiement automatisé

Le script d'installation télécharge et démarre la dernière version des outils disponibles dans le dépôt d'entreprise. Il permet également de choisir les « TAG » adaptés à la localisation de la forge (DMZ ou LAN).

Les quatre outils installés proposent leurs services au travers d'une interface *Web,* il y a donc un conflit sur l'utilisation des ports TCP 80 et 443 réservés respectivement aux protocoles HTTP et HTTPS. Une première solution est d'utiliser des ports TCP différents afin de faire cohabiter les applications sur la même machine. On peut donc avoir des accès spécifiques pour les différentes applications :

- https://forge.ign.fr:10080/ donne accès à Gitlab
- https://forge.ign.fr:20080/ donne accès à Redmine
- …

[46]**Post-installation**, script qui est lancé après une installation, manuellement ou automatiquement, afin de compléter celle-ci.

Le port par défaut (80/443) et donc l'url http://forge.ign.fr/ est alors réservé au portail d'accès.

Après échange avec les développeurs, cette solution, bien que fonctionnelle n'est pas satisfaisante car l'utilisation de ports « exotiques » n'est pas toujours prévue par des processus externes pouvant utiliser les API des logiciels installés.

Une autre solution est d'utiliser un conteneur intermédiaire chargé de distribuer les requêtes aux bons outils. Cette architecture, mise en place par le script de « post-installation », garantie d'avoir une VM unique directement utilisable pour l'ensemble des outils. Le diagramme de déploiement suivant (illustration 14) montre l'ensemble des composants mis en œuvre dans une forge.

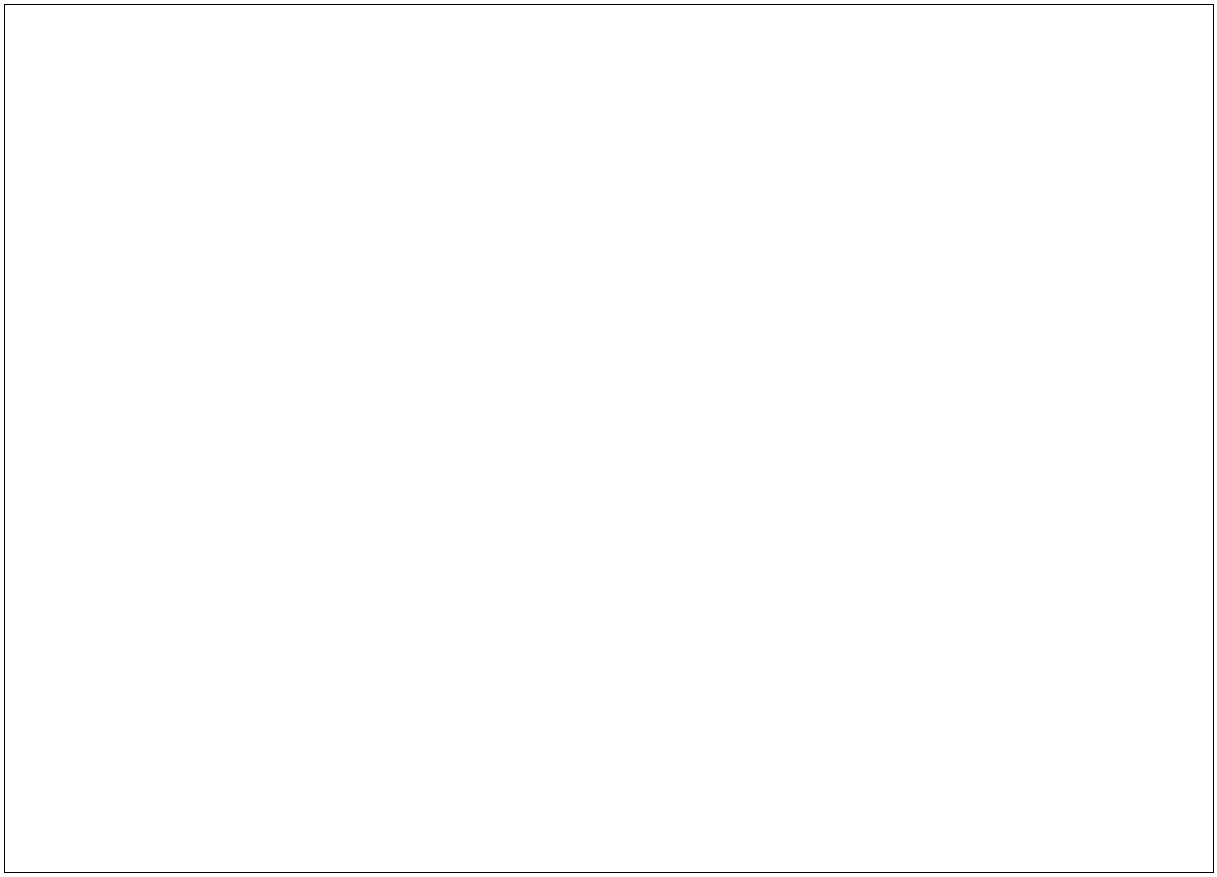


Illustration 14: Diagramme de déploiement d'une forge Docker

Le premier conteneur *Docker* contient un serveur *Nginx*[47] et joue le rôle de « *Reverse-Proxy* », c'est lui qui écoute sur le port 80 et 443 de la VM hôte dont le DNS principal est **forge*.ign.fr**. Ce *Reverse-proxy* fait ensuite une redirection vers les différents outils selon le DNS fourni lors de l'accès (qui sont tous des alias de **forge*.ign.fr**). Ainsi, si on accède à la

[47]**Nginx,** logiciel libre de serveur HTTP et de *Proxy* inversé.

forge avec l'URL *http://gitlab.forge*.ign.fr/* , le *Reverse-proxy* redirige les requêtes HTTP vers le conteneur *GitLab* qui s'exécute dans un des conteneurs de la VM.

Ce *Reverse-proxy* est dynamique. Si on définit un nouvel alias DNS **demo.forge*.ign.fr** et que l'on démarre un conteneur *Docker* (représenté « **???** » sur l'illustration 14) avec un serveur HTTP en lui précisant le DNS ***demo.forge*.ign.fr***, toutes les requêtes HTTP à destination de cette adresse seront automatiquement redirigées vers ce dernier conteneur. De cette façon, on donne la possibilité aux développeurs d'ajouter et/ou de modifier des conteneurs *Docker* dans la forge.

III.5.2.9 Mise en place de protocoles de sauvegarde des données

Le dernier point à mettre en place dans cette maquette concerne la gestion des sauvegardes. *Docker* permet de partager un dossier entre un conteneur et la machine hôte, c'est le concept de *Volume Docker*. Les données de chaque application sont donc partagées avec la machine hôte au travers de *Volumes Docker* nommée « *backup* » (voir illustration 14). Un script CRON[48] effectue la sauvegarde de toutes ces données sous forme d'une archive complète en prévenant l'administrateur en cas d'échec de la sauvegarde. Cette archive peut ensuite être prise en charge par le système de sauvegarde du SESI. Lors de la création d'un nouveau conteneur, si un *Volume Docker* « *backup* » est créé, il est automatiquement sauvegardé par le script. Le fonctionnement du script de sauvegarde est indiqué ci-dessous.

[48]**CRON,** logiciel de planification de commandes sous *Linux*.

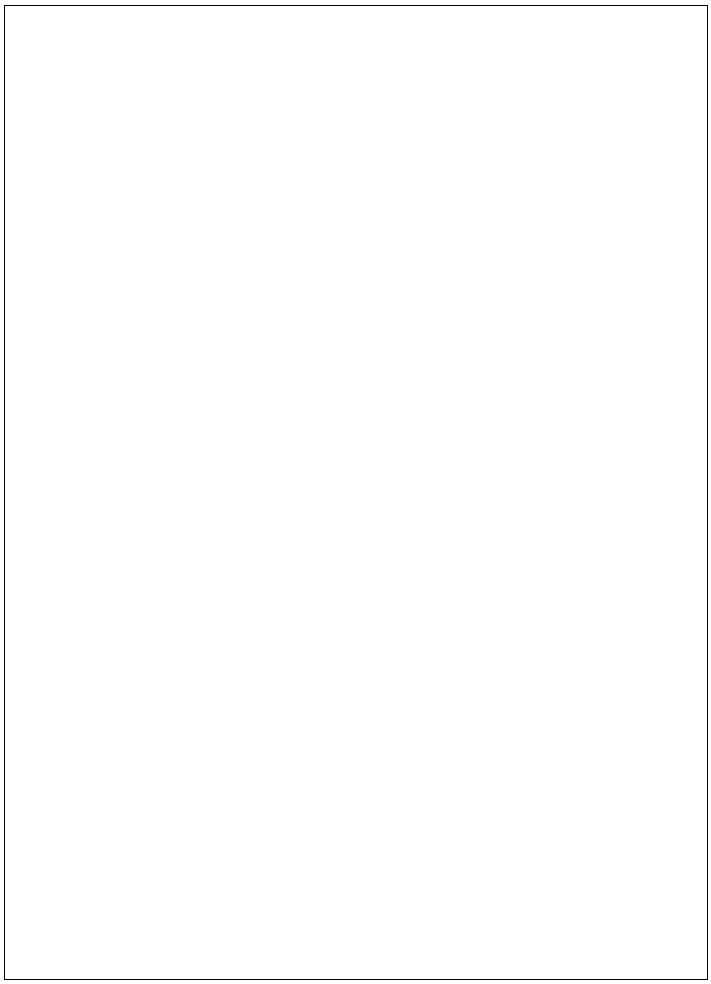


Illustration 15: Sauvegarde des forges

III.5.2.10 Mises à jour

La mise à jour des outils se fait en créant de nouvelles versions des images *Docker* dans le dépôt d'entreprise. Ceci est prévu par *Docker* et correspond aux « TAG » utilisées pour différencier les versions des différentes images. Pour mettre à jour une application, il suffit donc de la redémarrer à partir d'une image avec le « TAG » correspondant à la nouvelle version. Si la nouvelle version nécessite une mise à niveau des données (base *Postgres* par exemple), un processus de mise à niveau est inclus dans la nouvelle image. Si ce n'est pas le cas, il faut veiller à le faire manuellement.

III.5.2.11 Améliorations

La maquette de forge automatisée a été bien acceptée par les utilisateurs, il reste tout de même à étudier et au besoin, à faire évoluer certains aspects :

► *Implémenter correctement le HTTPS sur l'ensemble des outils avec la mise en place d'une PKI[49] dédiée pour les certificats.*

► *Proposer une authentification interne unifiée en intégrant un conteneur OpenLDAP[50] dans le cas des forges externes.*

► *Automatiser la création des DNS du sous-domaine (nécessite une délégation de sous-zone)*

III.6 Étude de coûts

Pour présenter les projets de maquettes, une étude de coûts concernant la mise en place d'une infrastructure de type *Cloud* au sein de l'établissement est réalisée. Dans ce tableau, issu de l'étude, on peut voir la comparaison des offres analysées avec les différentes options.

[49]**PKI**, de l'anglais « *Public key infrastructure* », outil de gestion de certificats électroniques.
[50]**OpenLDAP**, de l'anglais « Open Lightweight directory access protocol », implémentation Open source du protocole d'annuaire LDAP.

		Situation Actuelle	Cloud Privé					Cloud développement
	Solution	VMWare	Open-Source		Éditeur HP		Editeur vmWare	HP Helion Dev Platform
	Options		Libre	Supporté	HP / openstack	HP / vmWare	Vcloud (Vrealize+)	
Mise en place	Description Base **50**VM (objectif initial 200)	VMWare actuellement en production	Openstack Communauté	Openstack Support Mirantis	500*100(CSA+OO) 20000(Presta) Openstack(10000)	500*100(CSA+OO) 20000(Presta) VSphere(6*2900) VCenter(6*5000) NSX(6*4000)	prestation(20000) Vcloud-suite(6x11500) NSX(6x4000) Code Stream(6x7500)	20000(Presta) Openstack(10000) Devplatform (30000)
	Investissement Logiciel (k€)	**14000**	**0**	**22400**	**78000**	**141400**	**138000**	**60000**
	Investissement matériel (k€)	**30000**	**62000**	**62000**	**61000**	**36000**	**36000**	**40000**
	Charge RH	N/A	2 ESR 1 DEV Sur 4mois	1.5 ESR 0.5 DEV Sur 4mois	1ESR 1DEV 4mois	1ESR 1DEV 4mois	1ESR 1DEV 4mois	1ESR 2DEV 4mois
	Coût RH (k€)	0	112000	76000	72000	72000	72000	104000
	Total	**44000**	**174000**	**160400**	**211000**	**249400**	**246000**	**204000**
	Infogérance / support (k€/an)	**3300**	**0**	**9600**	**20000**	**21210**	**20700**	**14500**
	Exploitation IGN	1 ESR 10% 2 DEV 100%	1 ESR 100% 1 DEV 50%	1 ESR 75% 1 DEV 20%	1 ESR 20% 1 DEV 10%	1 ESR 20% 1 DEV 10%	1 ESR 20% 1 DEV 10%	1 ESR 20% 1 DEV 10%
	Coût d'exploitation RH (k€/an)	**187000**	**154000**	**117700**	**39600**	**39600**	**39600**	**39600**
	Formation nécessaire	*Non*	*Très importante*	*Importante*	*Importante*	*Oui*	*importante*	*importante*
Fonctions attendues	Suivi Projet *	OUI	OUI	OUI	OUI	OUI	OUI	NON
	Suivi Code *	OUI	OUI	OUI	OUI	OUI	OUI	Partiellement
	Intégration continue *	OUI	OUI	OUI	OUI	OUI	OUI	En partie
	Tests / recettes	OUI	OUI	OUI	OUI	OUI	OUI	NON
	Livraison	OUI par exploitation	EN PARTIE	EN PARTIE	EN PARTIE	OUI par exploitation	OUI par exploitation	NON
	WiKi *	OUI	OUI	OUI	OUI	OUI	OUI	LIMITEE
	VM LINUX base	OUI	OUI	OUI	OUI	OUI	OUI	NON
	VM WIN base	OUI	OUI	OUI	OUI	OUI	OUI	NON
	Déploiement Applicatif Automatisé	NON	EN PARTIE	EN PARTIE	EN PARTIE	NON	OUI	OUI
	Débordement Cloud Public	NON	NON	NON	SI COMPATIBLE	SI COMPATIBLE	SI COMPATIBLE	SI COMPATIBLE (ex Numergy)
	Portail Utilisateur	-	+	+	+++	+++	++	+++
Remarques	Type	IAAS	IAAS+	IAAS+	IAAS / PAAS	IAAS / PAAS	PAAS+/SAAS	PAAS+/SAAS
	Avantages/Inconvénients	Situation Actuelle	Les retours d'expériences des ministères ont montré que ce scénario est trop difficile (expertise poussée nécessaire)	Nécessite une implication forte de l'exploitation – portail de service moins adapté aux développeurs, plus à l'Exploitation – Liberté de création de modèle – Relative simplicité d'exploitation – capitalisation sur openstack	Portail sur mesure – nécessite une participation active et une collaboration importante de l'exploitation et des développeurs – capitalisation sur openstack	Portail sur mesure - nécessite une participation active et une collaboration importante de l'exploitation et des développeurs – compétence vmware existante	Difficulté pour obtenir des informations – Portail IT – Nécessite une participation active et une collaboration importante de l'exploitation et des développeurs – compétence vmware existante	Portail SaaS adapté aux développeurs - Portail de développement WEB uniquement (surtout JAVA), pas de Windows. Pas de « bac à sable » - pas de personnalisation – Adapté débordement sur cloud public à la demande- capitalisation sur openstack

Tableau 4 : Comparatif des solutions de Cloud

Les sommes indiquées dans ce tableau correspondent à la mise en place d'une plate-forme complète dimensionnée à 25% des besoins identifiés lors des interviews. Ce choix est fait pour obtenir des tarifs acceptables en ce qui concerne les offres commerciales. Ce sous dimensionnement laisse l'opportunité d'envisager un accroissement progressif de la plate-forme.

On constate de grandes différences avec l'estimation initiale du projet. Ces études incluent en effet la maintenance annuelle, quelle soit interne ou sous-traitée et ce poste n'était pas prévu dans les estimations du projet.

Le ROI de toutes ces solutions est très difficile à évaluer, car il se situe principalement au niveau ressources humaines. Deux améliorations sont attendues, une augmentation de la productivité des équipes de développement et une baisse de charge des équipes d'exploitation.

Théoriquement, les solutions de plate-forme PAAS doivent déplacer la charge sur les ressources humaines de l'exploitation vers le développement. Dans les faits, ce n'est pas le cas, tout du moins la première année. En effet, la mise en place de nouveau matériel va augmenter la charge des équipes d'exploitation avec un besoin de formation important. Certains développeurs voient quant à eux leur charge d'administration légèrement augmenter, même s'ils faisaient déjà de l'administration de serveurs à leurs niveaux.

Le gain en productivité des développeurs devient assez rapide, une fois les administrateurs de forges formés. En effet, le fait d'utiliser les mêmes outils sur toutes les forges facilite les échanges entre les équipes et fait disparaître les développements en doublon. La disponibilité immédiate des nouvelles forges permet aux développeurs de ne plus perdre de temps dans la mise en place de ces outils dans leurs locaux voire sur leurs machines.

> *Ce gain sera d'autant plus important lorsque la charte des projets informatiques tiendra compte des nouvelles méthodes de travail collaboratives rendues possibles par ce nouvel outillage.*

IV Déploiement de la solution de forge automatisée

Suite au report de la partie *Cloud* du projet, il est demandé la mise en production de la solution de forge automatisée.

IV.1 Recette

Pour préparer la recette de la solution, plusieurs forges sont testées par plusieurs unités de développement moyennant quelques mises au point techniques relatives au déploiement pour une utilisation en production. À la présentation de ces résultats, la recette du système de forge est demandée en COPIL. Le système de fourniture de ressources étant repoussé à 2016, il est décidé d'héberger les forges sur un des *Clusters* de virtualisation de production.

Pour effectuer cette recette, nous avons réalisé un cahier de recette, à destination des utilisateurs, indiquant la marche à suivre et les points à valider suivants :

- Architecture : tester la maîtrise des équipes par rapport aux technologies utilisées.
- Maintenabilité : tester la capacité des équipes à utiliser le système pour les différentes phases.
  - Déploiement.
  - Administration.
  - Montée en charge.
- Documentation : tester la qualité de la documentation.
- Sécurisation : tester la sécurité des données du système.
- Intégrité : valider le comportement du système lors d'une extinction brutale.
- Disponibilité : être averti d'un dysfonctionnement du système.
- Confidentialité : vérifier la cohérence des droits d'accès avec la PSSI.
- Sauvegarde : vérifier le mécanisme de sauvegarde.
- Performance : évaluer la performance selon plusieurs scénarios d'utilisation des forges.
- Intégration dans le SI : tester l'adhérence entre le système de forge et le SI.

> *À la fin de ma mission au sein de l'IDP, la recette n'était pas encore validée. Même si ma mission est terminée, je continue de suivre l'évolution de ce projet et fais toujours partie des échanges et des communications. J'ai notamment aidé un administrateur de forge dans la création d'un nouveau conteneur incluant les composants logiciels « Composer/SATIS » nécessaires au fonctionnement de leur chaîne de fabrication logicielle. Ce nouveau conteneur est, comme prévu, parfaitement intégré au système de forge et montre la capacité d'évolution de la solution proposée.*

IV.2 Accompagnement

La dernière action demandée en COPIL sur ce projet concerne la mise en place d'une formation pour les développeurs, potentiellement responsables de l'exploitation des applications présentes dans les forges. Pour être opérationnels rapidement, ces derniers ont besoin de connaître le fonctionnement intrinsèque de l'outil *Docker*. La formation se base donc sur cet outil, les autres outils utilisés dans les forges étant connus de ces derniers. La formation concerne aussi des membres de l'équipe d'exploitation qui seront amenés à maintenir l'outil *Docker* et les scripts d'automatisation des forges.

La première formation est programmée pour le dernier trimestre 2015.

> *Mon poste à la DRE m'autorise à participer à la formation des agents en tant que formateur dans mes domaines de compétences. C'est donc moi qui formerai les développeurs à l'outil Docker. Le support de formation servira également de documentation pour les administrateurs de forges.*

Conclusions

Dans son contrat d'objectifs et de performances, l'IGN doit « fournir des services pour accroître l'usage des données ». Le développement applicatif est un point critique dans la chaîne de production de ces services. Les développeurs ainsi que la gouvernance des projets sont demandeurs d'outils pour améliorer ces processus, notamment en ce qui concerne les délais de mises à disposition de ces outils.

Les solutions proposées pour favoriser le développement rapide de nouveaux logiciels montrent que les technologies liées au Cloud, qu'il soit privé ou public, sont aujourd'hui incontournables. Ce nouveau paradigme de gestion de l'informatique dans le SI de l'entreprise apporte la flexibilité nécessaire à une activité de pointe.

Le travail réalisé durant ce projet montre qu'il est possible d'envisager une mise en place de ces outils au sein du SI. Cette mise en place pourra avoir lieu au sein de l'entreprise dans un premier temps tout en envisageant une montée en charge hybride. L'utilisation de ressources *Cloud* mutualisées, fournies par les services de l'État sera alors toute indiquée. L'étude fournie concernant l'analyse d'une solution de fourniture de ressources pour le processus de développement logiciel va permettre de poursuivre ce projet quand les ressources budgétaires et humaines le permettront.

Le report de la mise en place de la partie « fourniture de ressources » de la plate-forme proposée m'a permis de me concentrer sur la partie « forge logicielle » afin de fournir un système immédiatement exploitable en production. Cette forge permet aux développeurs d'avoir un ensemble d'outils cohérent, ne nécessitant pas de configuration complexe et pouvant s'adapter à la diversité des projets grâce à sa modularité. L'automatisation de la création des forges logicielles constitue une amélioration significative du processus de développement en libérant de la charge, à la fois aux développeurs et aux équipes d'exploitations qui n'ont pas besoin de maîtriser les spécificités des logiciels utilisés dans les forges pour les déployer.

Ce système, basé sur les conteneurs *Linux* grâce à l'outil *Docker* va plus loin que la forge logicielle. Cette technologie qui n'était pas encore utilisée à l'IGN apporte en effet beaucoup de liberté et de simplicité dans la mise en place des infrastructures sous-jacentes aux déploiements de projets de développement. Ceci est particulièrement vrai pour les

technologies du *Web* qui représentent une part grandissante des projets réalisés au sein de l'Institut.

Les outils basés sur *Docker* fournis dans ce projet offrent aux développeurs une autonomie leur permettant de travailler sur leurs projets sur n'importe quelle infrastructure supportant *Docker* et ce quel que soit le support (leur propre machine, un *Cloud* public…). Cette capacité permet également de présenter le résultat du développement sous forme de démonstrateurs intégrés avec un comportement totalement similaire à la version déployée en production.

Lors de ce projet, j'ai pu travailler sur de nouvelles technologies liées au *Cloud* dont la complexité est très formatrice et particulièrement intéressante. Les échanges avec les architectes *Cloud* des ministères ont été instructifs pour comprendre les difficultés survenues lors de projets similaires et ainsi adapter les solutions proposées à nos propres problématiques.

Il est toujours passionnant de pouvoir participer à l'évolution de l'organisme dans lequel vous évoluez tous les jours. Ce projet d'infrastructure visant à améliorer un processus critique a été pour moi une excellente opportunité pour me lancer dans la conduite de projet. Selon moi, par rapport à la façon dont s'est déroulé ce projet, le point critique a été le respect des délais. Les délais ne peuvent être tenus que si les jalons du projet sont correctement définis et les acteurs parfaitement identifiés. Ainsi, les interactions continues avec les utilisateurs finaux auraient pu être incluses dans le plan d'avancement du projet, ceci afin de minimiser les écarts entre le résultat attendu et les solutions proposées. Mieux connaître la charge de travail prévisionnelle des équipes dont certaines phases dépendent aurait permis de réorganiser le calendrier et ainsi de diminuer les latences.

Du point de vue de l'organisation du travail, il n'est pas évident d'assumer à la fois le rôle de chef d'orchestre et de musicien. En effet, il m'est souvent arrivé de passer du temps à peaufiner une tâche au lieu de basculer sur la suivante. Pour gérer tous ces aléas, il est primordial d'avoir une équipe cohérente et la communication sous toutes ses formes, que ce soit au sein de l'équipe ou avec les autres acteurs du projet est la clé de voûte des éléments qui constituent la réussite d'un projet en entreprise. J'ai apprécié d'être l'interface de cette communication entre les différents intervenants, car je retrouve dans cette fonction le partage que j'ai toujours mis en avant dans mes différentes expériences professionnelles, notamment en tant qu'enseignant. Cette expérience d'un projet réel m'a aussi permis de prendre

conscience qu'il est important de ne pas négliger les échanges informels qui permettent de comprendre certains enjeux qui ne sont pas toujours écrits.

Ce mémoire marque l'aboutissement de ma formation d'ingénieur au CNAM. Celle-ci m'a permis d'être armé pour aborder le métier d'ingénieur au sein de l'entreprise. J'ai ainsi pu entreprendre avec un sentiment de légitimité un projet du point de vue du chef de projet et le mener aussi loin que possible avec le soutien et la reconnaissance de mes collaborateurs.

Soutenu par ma hiérarchie dans cette démarche d'évolution, je souhaite mettre à profit cette expérience en évoluant vers des postes exploitant mes nouvelles compétences.

Bibliographie / Webographie

Fifield T, Fleming D, Gentle A *et al.* 2014. *OpenStack Operations Guide* (Anglais). O'Reilly, 330 p.

Jackson K, Bunch C. 2013. *OpenStack Cloud Computing Cookbook* (Anglais). Seconde Edition. Packt Publishing Limited. 396p.

Syntec Numérique. Livre Blanc sur la sécurité dans le *Cloud Computing*, [en ligne], 2012. Disponible sur : http://www.syntec-numerique.fr/content/livre-blanc-cloud-computing-securite (consulté le 16 octobre 2014)

OpenStack Foundation. OpenStack documentation, (en Anglais) [en ligne]. Disponible sur : http://docs.*OpenStack*.org/ (consulté le 19 juin 2014)

Mirantis. *OpenStack* documentation. (en Anglais) [en ligne]. Disponible sur : *https://www.mirantis.com/products/mirantis-OpenStack-software/documentation/* (consulté le 09 juillet 2014)

Docker. *Docker* docs (en Anglais) [en ligne]. Disponible sur : https://docs.docker.com/ (consulté le 07 novembre 2014)

Table des illustrations

Liste des tableaux

Table des annexes

Annexe 1 : Action de développement IGN

Institut national de l'information géographique et forestière

Direction des services et du système d'information

Fiche d'action de développement

N° d'enregistrement :
La version signée est à déposer dans la GED

Titre du développement : Mise en place d'une plate-forme de développement

Responsable technique (MOE) : Cédric ESNAULT

Service maître d'œuvre : SIDT
S'il y a plusieurs services qui interviennent dans l'action de développement, on indique ici le maître d'œuvre principal

Directeur de projet (MOA) : Raphaël AURUS

Code du projet : 37PA14

Le code du projet (identifiant SAP) est constitué de 9 caractères, il est de la forme 37PAxxyyy ou 37KAxxyyy
xx : année de début de l'action de développement (14 pour 2014, 15 pour 2015 ...)
yyy : numéro d'ordre séquentiel, attribué par le maître d'œuvre
Le choix entre P et K, sans incidence majeure, est décidé par le maître d'ouvrage

Financement demandé pour 2014 : 31 k€ CCOP
Financement demandé pour 2015 : 45,3 k€ CCOP

Investissement prévu pour 2014 : 50 k€
Investissement prévu pour 2015 : 25 k€

Le financement doit être détaillé année par année s'il y a lieu

Financement total demandé : 76,3 k€ CCOP
Date de début : mai 2014
Date de fin : juin 2015

Fiche budgétaire pour l'année[51] 2014
Version du 19/06/2014

1. Frais de personnel

Catégorie	Nom	Service	Charge (préciser l'unité : mois, sem, jour)	Coût (k€ COP)	Coût (k€ CCOP)[52]
Ingénieurs					
	Cédric ESNAULT	SIDT	70j	22	22
	David CAUDRELIER	SIDT	15j	5,8	7
Ouvriers					
	Tech. IESE	SESI	5j	1,5	2
Total 1	----------	-------		29,3	31

2. Frais directs imputables au développement (hors frais de personnel)

Type de dépense	Coût k€ CCOP
Utilisation de moyens de production spécifiques, matériels ou logiciels (préciser lesquels)	
Achat de données (MNT, BDOrtho, BDTopo …)	
Achat de prestation interne (flashage SPI, vol SAA …)	
Frais de déplacements (préciser le lieu et la durée)	
Autres (préciser)	
Total 2	

3. Autres dépenses[53]

Type de dépense	Coût k€ CCOP
Sous-traitance (préciser les caractéristiques de la prestation demandée)	
Equipement prévu au PROEQ	50
Achat de formation (préciser les caractéristiques de la prestation demandée)	
Total 3	

Financement demandé sur 2014 (Total 1+2) : 31 k€ CCOP

Budget total pour l'année 2014 (Total 1+2+3) : 81 k€ CCOP

[51] Une fiche budgétaire par année civile.

[52] CCOP = COP + FDS (Frais Directs de Service) + FGV (Frais Généraux Ventilés)
COP = Charge x (Taux catégoriels + Taux matériels) :
FDS = Charge x CFH (Coût de Frais Horaire) : CFH (en €/h) dépend du service ; fourni par le SAFCG
FGV = (COP + FDS) x TFV (Taux de Frais Ventilés) : TFV (en %) dépend du service ; fourni par le SAFCG

[53] Ces dépenses ne sont pas à imputer au budget des Développements, mais à d'autres budgets (PROEQ, formation …) : elles sont demandées ici à titre indicatif en vue d'établir le budget global du développement

Fiche budgétaire pour l'année[54] 2015
Version du 19/06/14

1. Frais de personnel

Catégorie	Nom	Service	Charge (préciser l'unité : mois, sem, jour)	Coût (k€ COP)	Coût (k€ CCOP)[55]
Ingénieurs					
	Cédric ESNAULT	ENSG	70j	22	22
	Julien BACONAT	MQUSSI	5j	1,9	2,5
	Mickaël BORNE	SAI	5j	1,9	2,5
	David CAUDRELIER	SIDT	10j	3,9	4,8
Ouvriers					
	Tech. IESE	SESI	35j	10,7	13,5
Total 1	----------	-------		40,4	45,3

2. Frais directs imputables au développement (hors frais de personnel)

Type de dépense	Coût k€ CCOP
Utilisation de moyens de production spécifiques, matériels ou logiciels (préciser lesquels)	
Achat de données (MNT, BDOrtho, BDTopo …)	
Achat de prestation interne (flashage SPI, vol SAA …)	
Frais de déplacements (préciser le lieu et la durée)	
Autres (préciser)	
Total 2	

3. Autres dépenses[56]

Type de dépense	Coût k€ CCOP
Sous-traitance (préciser les caractéristiques de la prestation demandée)	
Equipement prévu au PROEQ	25
Achat de formation (préciser les caractéristiques de la prestation demandée)	
Total 3	

Financement demandé sur 2015 (Total 1+2) : 45,3 k€ CCOP

Budget total pour l'année 2015 (Total 1+2+3) : 70,3 k€ CCOP

[54] Une fiche budgétaire par année civile.

[55] CCOP = COP + FDS (Frais Directs de Service) + FGV (Frais Généraux Ventilés)
COP = Charge x (Taux catégoriels + Taux matériels) :
FDS = Charge x CFH (Coût de Frais Horaire) : CFH (en €/h) dépend du service ; fourni par le SAFCG
FGV = (COP + FDS) x TFV (Taux de Frais Ventilés) : TFV (en %) dépend du service ; fourni par le SAFCG

[56] Ces dépenses ne sont pas à imputer au budget des Développements, mais à d'autres budgets (PROEQ, formation …) : elles sont demandées ici à titre indicatif en vue d'établir le budget global du développement

Présentation de la demande[57]

1. Objectifs en production[58] :

Contexte
Le SIDT et le SAI possèdent différents environnements de travail pour leurs besoins en matière de développements et de prototypages :

- un environnement de virtualisation VMWare (2 serveurs, 1 MacPro et 1 baie de stockage iSCSI)
- un moteur d'intégration continue basé sur Jenkins
- différents outils de gestion de projets (Redmine, Mantis…)
- différents outils de gestions de codes sources (svn, git…)

Les problèmes suivants nous sont remontés aujourd'hui :

- l'environnement mis en place (cluster VMWare) correspond à une offre IAAS (Infrastructure As A Service) alors que leur besoin est plutôt au niveau SAAS (Software As A Service). La conséquence est que nous ne répondons pas au besoin de mettre en place des prototypes « clefs en main » mais seulement des machines vierges à configurer par les développeurs
- La création des environnements nécessite toujours de passer par des lourdeurs administratives même si la virtualisation simplifiée le processus de déploiement
- Le moteur Jenkins n'utilise pas le cluster de virtualisation pour lancer les batteries de test (tests unitaires, tests de non régression, ...) alors que la virtualisation serait très adaptée (provisionnement de machines à la volée, récupération des ressources après usage, ...)
- L'environnement est sous-dimensionné pour répondre aux besoins de tests de montées en charge et de pré-production

Nous notons enfin que l''environnement VMWare est très performant pour la gestion d'infrastructure (IAAS) mais son interface est très technique. Il manque une interface simplifiée de création d'environnements temporaire (bail de quelques heures à quelques jours) pour la création de prototypage par exemple. Ce n'est pas forcément le bon environnement pour une plate-forme de développement.

Objectifs

Cette Action de développement contient 3 objectifs :

- **Etudier les besoins du SAI et du SIDT** pour une plate-forme de développement commune et effectuer **un état de l'art** des solutions disponibles aujourd'hui
  - les aspects environnements de développement (gestion de projets, de codes sources, intégration continue…), qualifications (gestion des évolutions, recette, test de montée en charge…) et déploiement (mise en exploitation en hébergement interne et externe) devront être abordés

[57] Texte libre. **Aucune rubrique n'est a priori facultative.**
Si un point nécessite un plus long développement, joindre une annexe.
[58] Mentionner l'unité de production utilisatrice des développements qui sera associée à la recette.

- **Suivre les travaux au sein du MEN (Ministère de l'Education Nationale) et du MEDDE** dans une optique de rationalisation :
  - la cellule de veille technologique du MEN mène une étude sur le déploiement d'une plate-forme de développement basée sur la solution Open Source Mirantis
  - le MEDDE mène l'expérimentation G-Cloud qui met en place une solution Open Source autour d'*OpenStack*
- **Mettre en place une plate-forme de développement** dédiée aux équipes de développement du SIDT et du SAI
  - **proposer une solution technique** qui répond aux besoins du SAI et du SIDT et exploitable par le SESI. Les aspects dimensionnement et sécurité de l'infrastructure sous-jacente seront traités et des réponses techniques aux besoins fonctionnels exprimés devront être proposés. La solution devra être chiffrée avec un échéancier de mise en place.
  - **réaliser une maquette** pour valider la solution technique
  - **suivre le déploiement par le service d'exploitation et organiser la recette**

2. Travaux déjà réalisés et publications :

Néant

3. Partenaires extérieurs (déjà contactés ou potentiels) :

La cellule technologique du MEN
Le groupe de travail G-Cloud du MEDDE

4. Equipements conditionnant ce développement :

Néant

5. Prise en compte des aspects QUSSI :

Il s'agit ici d'identifier et qualifier, en collaboration avec la MQSUSI (Mission Qualité, Sécurité et Urbanisation des Systèmes d'Information), l'impact du développement sur le SI de l'IGN en matière de cohérence globale et de sécurité informatique (en particulier pour les développements Web).

Une charge de 5 jours est prévue pour la MQSUSI dans le cadre de cette action.

Présentation de la demande (suite)

6. Démarche prévue, points clés et calendrier.

Rubrique à remplir obligatoirement

Etude préalable
Etude de mai 2014 à novembre 2014
Charge de travail de 3 mois pour Cédric ESNAULT, 15j pour David CAUDRELIER
Livraison de :

- suivi des groupes de travail sur le sujet au SAI et au SIDT
- synthèse des besoins du SAI (Dév. Web, Incubateur, ...)
- synthèse des besoins du SIDT (dév. Internes, sous-traitance, Géoportail, ...)
- état des lieux des solutions en place
- état de l'art sur le sujet
- propositions techniques chiffrées (moyens humains, matériels et logiciels)

Maquette
Réalisation d'une maquette de septembre 2014 à janvier 2015
Charge de travail de 1 mois pour Cédric ESNAULT, 5j pour David CAUDRELIER, 10j pour un technicien d'exploitation, 5j pour Julien BACONAT
Livraison de :

- maquette de la solution technique retenue à l'issue de l'étude préalable
- validation de la solution par les utilisateurs

Déploiement de la solution définitive
Suivi en mode projet du déploiement du système de févier 2015 à juin 2015
Charge de travail de 3 mois pour Cédric ESNAULT, 5j pour David CAUDRELIER, 30j pour un technicien d'exploitation
Livraison de :

- cahiers des charges pour les marchés publics nécessaires (matériels, logiciels, prestations)
- planification du déploiement
- documentations (installation, exploitation)

Phase de Recette finale :

Il est impératif d'indiquer ici quels sont les éléments de recette c'est-à-dire les éléments sur lesquels le maître d'ouvrage se basera avant de prononcer l'acceptation finale des travaux.

La recette sera prononcée par le SIDT et le SAI comme service demandeur, par le SESI comme service exploitant et par la MQUSSI pour la partie urbanisation et sécurisation du SI.
En cas d'hébergement externalisé de la solution, le respect de notre Politique de Sécurité du SI sera une contrainte majeure.

Avis et décision

Avis du chef de Service maître d'œuvre :

Avis favorable
Laurent Quêne
Chef adjoint SSIDT

Avis de la MQSUSI :

Si nécessaire

Avis de la direction demandeuse :

Mention obligatoire

Avis du maître d'ouvrage (DPC ou DPDE) :

Mention obligatoire de la part du maître d'ouvrage

Décision :

Les mentions inutiles seront rayées

Demande acceptée
Pour un montant total de 76,3 k€ au budget des développements

Décision remise à une date ultérieure
Demande refusée

Annexe 2 : Reporting bimensuel

IGN INSTITUT NATIONAL DE L'INFORMATION GÉOGRAPHIQUE ET FORESTIÈRE	Compte-rendu d'avancement intermédiaire	n°10
	Projet: Plate-forme de développement	**Période :** 21/02/14 – 07/03/15
	Auteur: Cédric Esnault **(CET)**	
Diffusion : David Caudrelier (DCR)		

Vue d'ensemble de la situation du projet

Aspect	indicateur	commentaires
• Périmètre		RAS
• Planning		Nouveau décalage du COPIL 2 : 19/03/2015 => 26/03/2015
• Ressources		Manque de ressource électrique, recherche d'un autre emplacement
• Budget		Retard dans la validation du programme d'équipement 2015
• Technologique		Intégration des nouvelles offres dans l'étude

: Pas de problème identifié
: Problème identifié – Solutions trouvées pour minimiser l'impact
: Problème majeur – Nécessite une décision

Avancement des phases principales

- Étude préalable : 100 %
- **Réalisation des maquettes : 21 %**
  - **Maquette type 1 et type 2 : 15 % :** Préparation du matériel en cours
  - **Architecture hybride : 20 %** : évaluation des nouvelles offres
  - **Forge : 10 %** :
- **Planification du déploiement / livraison des outils : 40 %**
  - **Amélioration de l'intégration des outils (AGILE)**

Annexe 3 : Planning initial

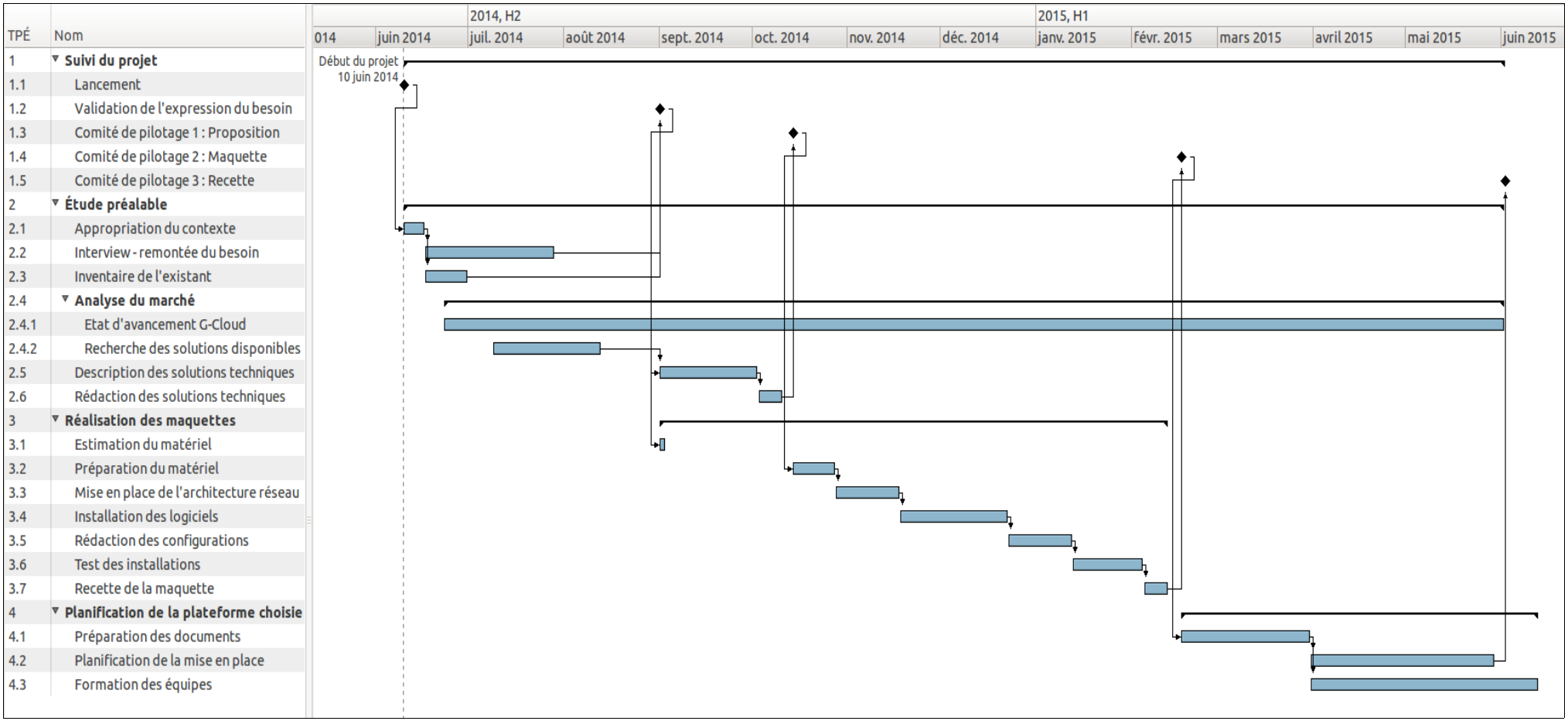


Illustration 16: Planning initial du projet

Annexe 4 : Liste des besoins exprimés par les utilisateurs

Cette synthèse présente les différents besoins émis par les services ayant à effectuer des développements au sein de l'IGN. Elle présente également les contraintes identifiées, impactant nécessairement les choix dans la mise en place des solutions retenues. La mise en place de machines virtuelles « à la demande » est la seule solution envisageable pour fournir des ressources aux équipes de développement sans faire intervenir les équipes d'exploitation. Ces machines sont indiquées « VM » par la suite.

Fonctionnalités recherchées

Les fonctionnalités recherchées par les différents services de développement dans la plate-forme sont les suivantes, classées par ordre de fréquence des demandes. À ces besoins exprimés, sont ajoutés les besoins techniques qui en découlent. Ces fonctionnalités seront regroupées par la suite en briques métiers. La charte graphique précise la criticité des besoins :

Légende

- Indispensable
  - cette fonctionnalité devra être mise en place dans la plate-forme
- Importante
  - cette fonctionnalité sera mise en place ou non selon les contraintes imposées (délai, coût, sécurité, impacts divers…)
- Souhaitée
  - cette fonctionnalité sera mise en place si elle est proposable sans surcoût

Fonctions

- Effectuer le suivi d'un projet de développement dans son ensemble en regroupant tous les aspects et outils autour d'une gestion centralisée.
  - Disposer d'un système de gestion des bugs/incidents
  - Gérer précisément les droits à la fois au niveau du suivi de projet (accès aux tickets) que de l'accès aux codes sources et aux ressources (VM)
  - Avoir des indicateurs de suivis de l'avancement des projets et de l'utilisation des ressources utilisées.

    - Sauvegarder le suivi de projet
    - Disposer d'un espace d'échange de type WIKI
- Proposer un outil de traitement de texte collaboratif.
- Archiver le code source en conservant la chronologie et l'origine des modifications effectuées sur ce dernier (l'utilisation d'un logiciel de contrôle de version (CVS) centralisé n'est pas obligatoire mais le projet doit contenir un dépôt principal sur la plate-forme)
    - Collaborer à plusieurs sur un même code source
    - Sauvegarder le code source
- Gérer les accès à la plate-forme, que les agents soient IGN ou extérieurs
- Cloisonner les projets
    - selon leur niveau d'ouverture à l'extérieur de l'IGN.
    - selon les liens avec le SI IGN
    - selon leur niveau de confidentialité
- Effectuer l'intégration continue des développements en incluant toute la diversité des environnements nécessaires
    - Client :
        - Windows XP, 7 et 8 en 32 et 64bits
        - Mac OSX (montain lion / maverick)
        - Linux (Debian 7, Ubuntu LTS)
        - Plate-forme mobile Android et IOS (ARM)
    - Serveur :
        - Windows 2008 /2012
        - Linux Debian 7, CentOS
    - accès à des ressources GPU
- Disposer de ressources matérielles pour tester les développements (hors intégration continue) sur une infrastructure disponible pouvant simuler les conditions de production, y compris l'accès aux données
    - ex : 1 serveur Web Apache et une base Postgres en réseau privé, un poste client windows 7 avec géoconcept
- Fournir un système de validation de mise à disposition des ressources.(circuit de validation)
- Administrer la plate-forme simplement coté SESI (IHM)
- Accéder à des données hébergées à l'exterieur de l'IGN, y compris Géoportail
    - préciser le type d'accès Géoportail (via LS ou via Internet)
- Accéder à des données IGN non publiques (duplication des données de production)
- Disposer d'espace de stockage (dimensionnement à venir) pour les phases de qualification, de montée en charge et de recette (Jeu de données)
- Disposer d'outils de contôle et de qualité de code
- Tester des projets sur terminaux mobiles
- Distribuer les produits finalisés en gérant les dépendances éventuelles
- Contrôler les ressources utilisées
- Capitaliser sur les développements
    - proposer un outil de recherche de code
- Mettre en place simplement des démonstrateurs
- Effectuer la maintenance des produits sur la plate-forme (reproduction de bugs).
- Proposer un système de conversion des incidents amélio du processus de production quand ceux-ci sont identifiés comme « bug »
- Tester la robustesse des développements (montée en charge)

- Accéder à des jeux tests et des environnements pour les exploiter
- Capitaliser sur les jeux tests
- Proposer des environnements de développements pré-packagés
  - Effectuer la recette des produits sur la plate-forme.
  - Exposer et référencer la documentation des développements
- Capitaliser sur les environnements mis en place (ressources déployées)
- Exposer sous forme de services WEB les développements non WEB
  - ex : générer un service web interfaçant un programme comme CIRCE
- Générer de la documentation automatiquement à partir du code source
- Proposer des environnements « ready to use » pour les sessions de formation.
- Historiser les environnements (Snapshotting des VM, points de restauration)
- Proposer la plate-forme dans différentes langues aux niveaux des interfaces

Contraintes identifiées

Les contraintes suivantes ont été identifiées, classées par thèmes avec à chaque fois le niveau de criticité de la contrainte :

- Critique
- Importante

Contraintes techniques

- L'utilisation de certains logiciels de production est soumise à licence (à prendre en compte lors de l'instanciation d'une machine virtuelle)
- Le support des machines virtuelles Mac OSX n'est assuré que par l'hyperviseur VMWare sur du matériel Apple.
- L'accès aux données du géoportail impose l'utilisation de la Liaison spécialisée ATOS
- L'utilisation d'appareils mobiles implique une infrastructure d'accès sans fil

Contraintes de sécurité

- La PSSI de l'IGN impose un cloisonnement des machines (physiques ou virtuelles) accessibles à des agents non-IGN
- La PSSI de l'IGN impose de connaître et de stocker dans un annuaire différent les intervenants extérieurs

Contraintes humaines

- Le SESI demande que soit pris en compte son expérience avec les hyperviseurs Vmware dans le choix de la solution de virtualisation
- Le SESI demande que l'administration de la plate-forme et de son architecture soit simple et ergonomique
- Une astreinte pour la plate-forme en dehors des heures d'ouverture de l'IGN devra être assurée en cas d'ouverture de la plate-forme à l'extérieur.

Contraintes de délais

- La plate-forme est attendue pour mi-2015
- Le projet pilote sera présenté début 2015
- L'architecture technique de la plate-forme sera proposé mi-octobre 2014

Annexe 5 : Les risques de la cybercriminalité

http://blog.crimenumerique.fr/2015/01/02/developpeurs-ne-partagez-pas-vos-cles-avec-nimporte-qui/

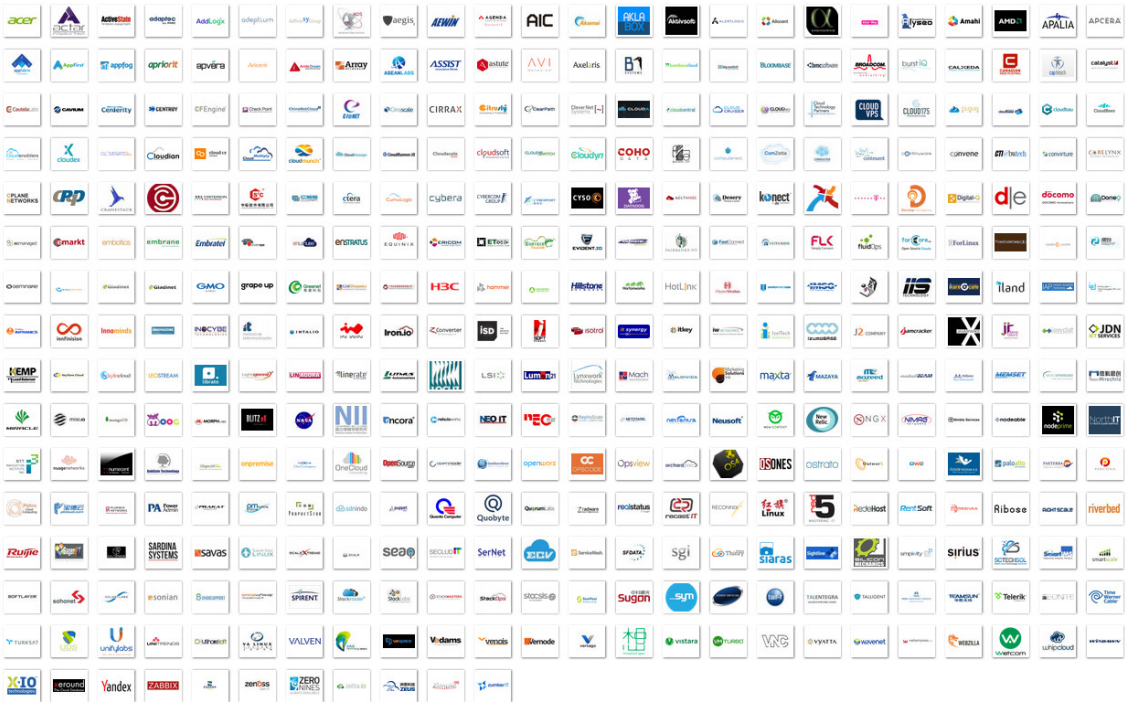
Dans cet exemple, les clés privées permettant l'accès au service de *Cloud* ont été utilisées dans un processus d'automatisation et stockées dans un gestionnaire de code public *(Github)*. Des pirates ont scanné ces gestionnaires de code à l'aide d'outils de recherches, ils y ont trouvé ces clés et les ont utilisées pour générer un grand nombre de services sur l'hébergement afin de générer des cryptomonnaies, engendrant des frais importants pour la victime.

Annexe 6 : Partenaires *d'OpenStack*

https://www.OpenStack.org/foundation/companies/

Membres

Partenaires



Annexe 7 : Architecture d'*OpenStack*

La page suivante donne l'architecture logique de base *d'OpenStack* (Illustration 17 extraite de la documentation *d'OpenStack*). On y trouve les principaux modules et les API disponibles pour communiquer avec eux.

- **Nova** : C'est le cœur *d'OpenStack*, il gère les hyperviseurs et coordonne l'ensemble des opérations.
- **Cinder** : Ce module gère le stockage en mode « block ».
- **Glance** : Ce module gère les images, c'est à dire les modèles de VM utilisables dans le C*loud.*
- **Neutron** : Ce module s'occupe de la couche réseau.
- **Keystone** : Ce module gère l'authentification et les rôles.
- **Swift** : Ce module n'est pas obligatoire, il gère le stockage en mode « Objet ».
- **Horizon** : C'est le portail *Web* par défaut *d'OpenStack*.

D'autres modules sont disponibles et permettent d'augmenter les fonctionnalités disponibles.

Illustration 17: Architecture logique d'OpenStack

Annexe 8 : Mise en place de *Mirantis OpenStack*

La marche à suivre pour l'installation d'un *Cloud OpenStack* avec la distribution *Mirantis* est la suivante et se décompose en deux phases.

La première est la préparation de l'infrastructure physique :

► Préparer les serveurs physiques, un serveur sera dédié au composant *Fuel*, les autres seront les nœuds disponibles pour la création des *Cloud OpenStack*. *Mirantis* fournit un outil en ligne pour dimensionner l'infrastructure Hardware en fonction des besoins (voir illustration 18).

► Configurer la partie physique du réseau des serveurs. Plusieurs réseaux physiques entre les nœuds sont nécessaires pour mettre en place un *Cloud OpenStack,* certains peuvent être mutualisés dans des VLAN (voir illustration 19). Un de ces réseaux est dédié exclusivement au composant *Fuel.* Il faut que les serveurs soient configurés pour démarrer sur ce réseau (boot PXE).

► Installer le composant *Fuel* sur le serveur dédié à l'aide d'une image « iso » fourni par *Mirantis.*

On voit que la partie "Hardware" est très limitée, une fois cette étape terminée, tout se passe dans l'interface *Web* de *Fuel,* il n'y a aucune intervention à faire sur les serveurs excepté la maintenance physique (pannes…).

Ensuite vient la préparation des nœuds. Voici les étapes à effectuer dans *Fuel* pour configurer un *Cloud :*

► Procéder au démarrage des serveurs qui trouveront le serveur PXE fourni par *Fuel* et seront donc détectés par ce dernier comme étant des nœuds disponibles.

► Ajouter des serveurs disponibles à notre *Cloud* et leur attribuer des rôles dans l'interface de *Fuel. Fuel* propose les rôles de base *d'OpenStack* et vérifie la compatibilité des rôles choisis pour un même serveur. Il est ainsi impossible d'installer des rôles incompatibles sur un même serveur.

► Définir les réglages réseaux et choisir parmi les options disponibles (modules supplémentaires, utilisation de composants physiques spécifiques…)

► Appliquer la configuration !

À partir de ce moment *Fuel* lance de manière totalement autonome l'installation des systèmes d'exploitation (*Ubuntu* ou *CentOS*) sur les nœuds sélectionnés puis des composants *OpenStack* choisis. Au bout d'un certain temps le portail *Horizon d'OpenStack* est disponible à l'adresse indiqué.

Il est ensuite tout a fait possible d'ajouter des nœuds au *Cloud* pour en étendre ses capacités mais aussi de déployer d'autres *Cloud* avec des réglages différents grâce au même serveur *Fuel.*

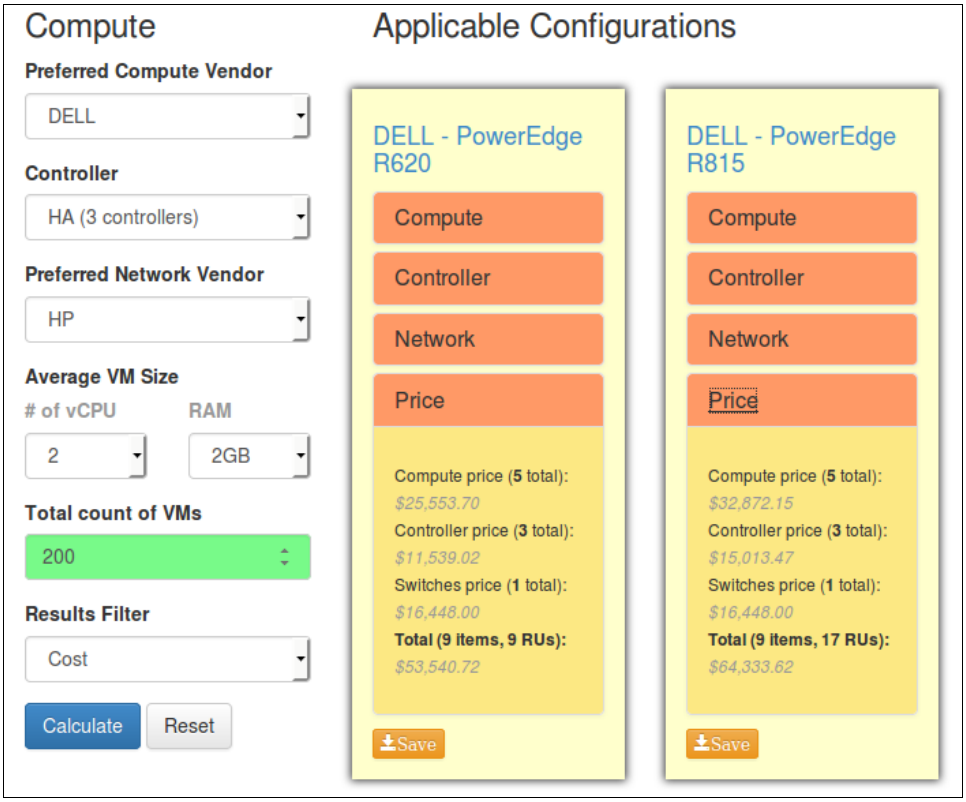


Illustration 18: Calculateur d'architecture Mirantis

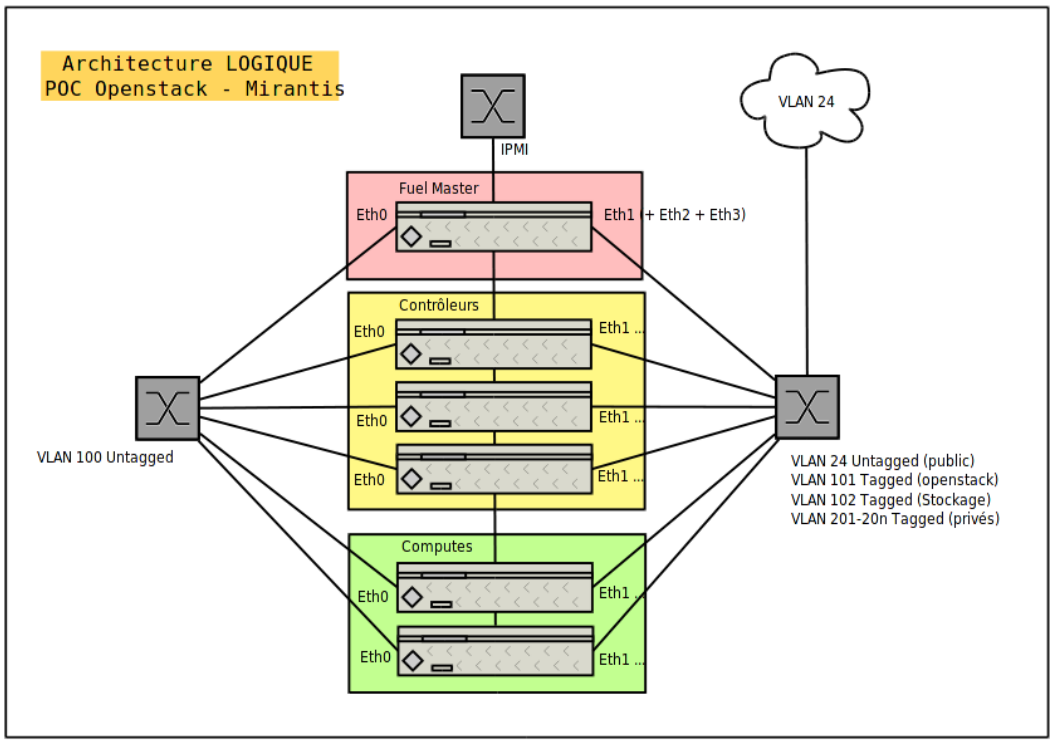


Illustration 19: Architecture logique d'une installation Mirantis OpenStack

Annexe 9 : Planning réalisé

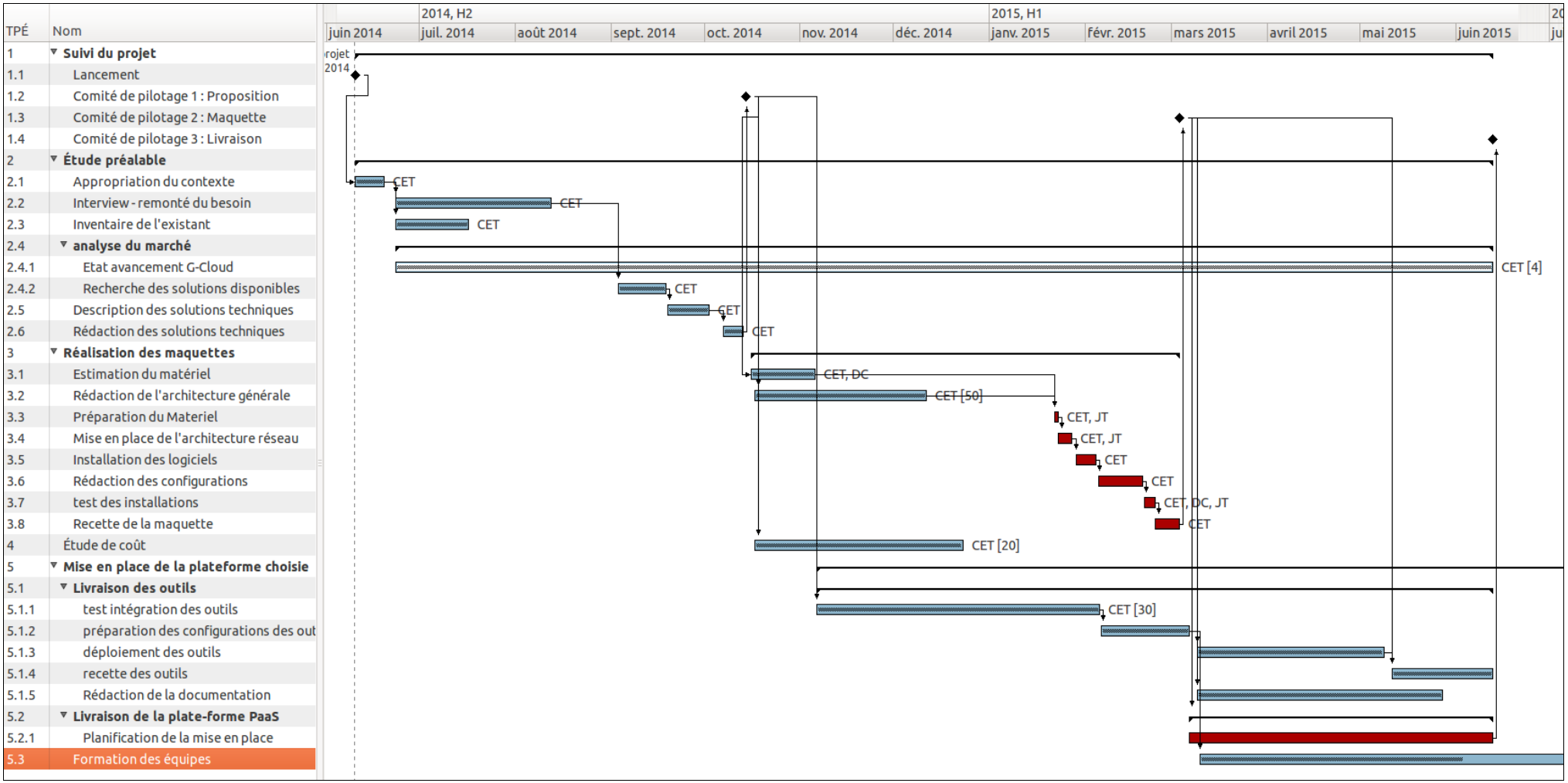


Illustration 20: Planning réalisé

Annexe 10 : DockerFile

Voici un exemple de fichier *Dockerfile* permettant la construction de l'image *GitLab*. Ce fichier est accompagné des fichiers de configurations personnalisés qui seront copiés à l'intérieur du conteneur.

```
FROM sameersbn/ubuntu:14.04.20141026
MAINTAINER cedric.esnault@ign.fr

# Définition des variables d'environnement nécessaires
ENV HTTP_PROXY http://proxy.ign.fr:3128
ENV HTTPS_PROXY http://proxy.ign.fr:3128

# Ajout des dépots et téléchargement des paquets
RUN apt-key adv --keyserver keyserver.ubuntu.com --recv E1DF1F24 \
 && echo "deb http://ppa.launchpad.net/git-core/ppa/ubuntu trusty main" >>
/etc/apt/sources.list \
 && apt-key adv --keyserver keyserver.ubuntu.com --recv C3173AA6 \
 && echo "deb http://ppa.launchpad.net/brightbox/ruby-ng/ubuntu trusty main"
>> /etc/apt/sources.list \
 && apt-key adv --keyserver keyserver.ubuntu.com --recv C300EE8C \
 && echo "deb http://ppa.launchpad.net/nginx/stable/ubuntu trusty main" >>
/etc/apt/sources.list \
 && apt-get update \
 && apt-get install -y supervisor logrotate locales \
      nginx openssh-server mysql-client postgresql-client redis-tools \
      git-core ruby2.1 python2.7 python-docutils \
      libmysqlclient18 libpq5 zlib1g libyaml-0-2 libssl1.0.0 \
      libgdbm3 libreadline6 libncurses5 libffi6 \
      libxml2 libxslt1.1 libcurl3 libicu52 \
 && update-locale LANG=C.UTF-8 LC_MESSAGES=POSIX \
 && locale-gen en_US.UTF-8 \
 && dpkg-reconfigure locales \
 && gem install --no-document bundler \
 && rm -rf /var/lib/apt/lists/*

#Copie des fichiers de conf dans le conteneur
COPY assets/setup/ /app/setup/
RUN chmod 755 /app/setup/install
RUN /app/setup/install
COPY assets/config/ /app/setup/config/
COPY assets/init /app/init
RUN chmod 755 /app/init

# définition des ports à ouvrir
EXPOSE 22
EXPOSE 80
EXPOSE 443

# définition des partages à exposer
VOLUME ["/home/git/data"]
VOLUME ["/var/log/gitlab"]

# Script au lancement du conteneur
ENTRYPOINT ["/app/init"]
CMD ["app:start"]
```

Étude de la mise en œuvre d'une plate-forme de développement applicatif et suivi de sa réalisation

Mémoire d'Ingénieur CNAM, Paris 2015

RÉSUMÉ

l'IGN souhaite rationaliser ses processus de développement pour s'adapter à l'évolution constante et rapide du marché. Les délais de mises à disposition d'outils et de ressources informatiques sont la première cause identifiée de retard. La mise en place d'une plate-forme de développement applicatif, notamment sous forme de forge logicielle apporte une amélioration significative sur ce point en offrant plus d'autonomie aux équipes de développement. Les technologies de *Cloud Computing* apportent une grande flexibilité dans l'exploitation de ces outils et permet de laisser les développeurs se concentrer sur leurs missions.

Ce rapport présente la méthodologie utilisée et les moyens mis en œuvre pour l'étude et la réalisation de cette solution.

Mots clés : IGN, Logiciel, Developpement, Cloud, PAAS, Devops, OpenStack, Docker

---

ABSTRACT

IGN wants to streamline its development processes to adapt to the rapidly-changing market. Delays in provision of tools and computing resources are the primary causes identified delays. The establishment of an application development platform, such as a software foundry provides a significant improvement in this regard by providing more autonomy to the development teams. *Cloud Computing* technology provides high flexibility in the operation of these tools and allows the developers to focus on their missions.

This report presents the methodology and resources used for the study and implementation of this solution.

Key words: IGN, Software, Development, Cloud, PAAS, Devops, OpenStack, Docker