

Mon Compte Mobilité

Standardisation des MaaS - Gateway

Document d'Architecture Technique
Version 1.2

Sommaire

1.	Introduction	5
1.1.	Définitions et termes	5
1.2.	Documents de référence.....	6
1.3.	Documents applicables	6
1.4.	Table des révisions.....	6
2.	Contexte et motivation du projet	7
2.1.	Ecosystème	7
2.2.	Positionnement	8
2.3.	Objectifs.....	10
2.4.	Périmètre fonctionnel du PMV	11
2.5.	Contraintes	11
2.6.	Périmètre du DAT.....	13
3.	Architecture conceptuelle	14
3.1.	Hypothèses fonctionnelles	14
3.2.	Acteurs.....	14
3.3.	Processus métier.....	15
3.4.	Cas d'utilisation.....	16
3.5.	Cartographie fonctionnelle	20
3.6.	Principaux concepts métier	20
3.7.	Exigences fonctionnelles	25
3.8.	Exigences non fonctionnelles	26
4.	Architecture logique.....	29
4.1.	Architecture Hub	29
4.2.	Architecture Gateway	30
4.3.	Scénarios d'intégration	31
4.4.	Domaines et composants logiques.....	33
4.5.	Stockage et utilisation des données.....	34
4.6.	Flux et cinématique	34
5.	Principes directeurs.....	40
6.	Architecture technique	42
6.1.	Schéma de principe.....	42
6.2.	Justification des choix d'architecture.....	43
6.3.	Architecture technique du PMV	44
6.4.	Licences open sources autorisées.....	45
6.5.	Composants logiciels	46
6.6.	Provisionnement des ressources dans Azure	48
6.7.	Schéma réseau	49
6.8.	Plateformes et environnements	50
7.	Sécurité	53
7.1.	Gestion des identités, identification & authentification	53

7.2.	Certificats serveurs et nom de domaine	53
7.3.	Autorisation et contrôle d'accès.....	54
7.4.	Intégrité	54
7.5.	Confidentialité.....	54
7.6.	Traçabilité / Journalisation	54
7.7.	Imputabilité et non répudiation.....	55
7.8.	Anonymisation & Pseudonymisation des données.....	55
7.9.	GDPR	55
7.10.	Sauvegarde / restauration	55
7.11.	Purge	55
7.12.	Archivage	55
7.13.	Tests d'intrusion / Revues de Code Source	56
7.14.	Limitation volumétrie.....	56
7.15.	Anti-DDoS.....	56
7.16.	Cloisonnement	56
7.17.	Système de détection d'intrusion.....	56
7.18.	Montée de version des systèmes d'exploitation et patchs sécurité	56
8.	Modèle de dimensionnement théorique	57
8.1.	Entrants écosystème	57
8.2.	Hypothèses de Capgemini.....	57
8.3.	Modèle de dimensionnement théorique	58
8.4.	Performances.....	58
9.	Processus de livraison	60
10.	Approche DevOps	61
10.1.	Introduction à GitOps	61
10.2.	Implémentation	62
10.3.	Pipelines de build et déploiement	64

Tables des illustrations

Figure 1 - Rôles dans l'écosystème des MaaS	7
Figure 2 - Enjeux de la Standardisation des MaaS	8
Figure 3 - Apports de la Standardisation des MaaS.....	9
Figure 4 - La standardisation des MaaS concrétisée par la mise en place d'un Hub, qui préfigure la mise en place de standards, et d'API standardisées	10
Figure 5 - Périmètre fonctionnel Hub MCM	11
Figure 6 - Orientations sur les standards (vue fin 2021)	12
Figure 7 - Processus et parcours MaaS	15
Figure 8 - Processus d'intégration MSP	16
Figure 9 - Cas d'utilisation Citoyen dans le MaaS supportés par la Gateway	18
Figure 10 - Point de vue MaaS	19
Figure 11 - Point de vue de l'administrateur AOM	19
Figure 12 - Point de vue de l'administrateur technique	20
Figure 13 - Cartographie fonctionnelle Gateway	20
Figure 14 - Modèle de données Gateway (domaines couverts).....	21
Figure 15 - Configuration système : modèle de données GW	22
Figure 16 - MSP : exemple de Modèle de données	23
Figure 17 - MCD tables msp calls paramétrage	24
Figure 20 - Architecture générale Hub MCM.....	29
Figure 21 - Architecture détaillée Hub MCM	30
Figure 22 - Architecture générale Gateway MCM Std MaaS	30
Figure 23 - Scénario d'intégration territorial	31
Figure 24 – Domaines et composants logiques	34
Figure 25 - Cinématique des flux	35
Figure 26 - Architecture globale de l'APIM Gravitee	37
Figure 27 - Flux entrants Gateway MCM Std MaaS	38
Figure 26 – Microservices internes	38
Figure 28 – Microservice des flux sortants Gateway MCM Std MaaS	39
Figure 29 – Principes directeurs d'architecture	40
Figure 30 - Architecture technique – schéma de principe.....	42
Figure 31 - Architecture technique retenue pour la phase d'expérimentation	45
Figure 32 - Licences opensource	46
Figure 33 - Options de segmentation du trafic réseau dans Azure	49
Figure 35 - Schéma réseau dans Azure	50
Figure 34 - Organisation des ressources dans Azure.....	51
Figure 36 - Mesure des performances	59
Figure 37 - Processus de livraison	60
Figure 38 - Principes de GitOps.....	62
Figure 39 - Approche DevOps : orientation GitOps	63
Figure 40 – Pipeline et livraison	64

1. Introduction

Ce document présente l'architecture du système MCM « Standardisation des MaaS » Gateway dans son écosystème, il sera complété et enrichi de manière itérative.

1.1. Définitions et termes

Terme	Description
MSP	“Mobility Service Provider”, fournisseur d'un service de mobilité.
MaaS	“Mobility As A Service”, plateforme réunissant l'information, la réservation et le paiement de l'ensemble de l'offre de mobilité disponible.
MCM	Mon Compte Mobilité
moB	Produit du projet Mon Compte Mobilité
Std MaaS	Standardisation des MaaS
Gateway (GW)	Passerelle
CMS	Compte Mobilité Standardisé
HUB	Plateforme regroupant les services MCM de Gateway et de Compte Mobilité (MOB / CMS)
Mobilité durable	Mobilité décarbonée.
RGPD	Règlement Général sur la Protection des Données.
RGAA	Référentiel Général d'Accessibilité pour les Administrations.
CNIL	Commission Nationale de l'Informatique et des Libertés
CEE	Certificats d'Économie d'Energie
SIS	Système d'Information et de Services pour la billettique
SIV	Système d'Information Voyageur
TC	Transports en Commun
IV	Information Voyageur
RI	Recherche d'Itinéraires
IVTR	Information Voyageur Temps Réel
REF	Référentiel
IDFM	Île-de-France Mobilités
GART	Groupement des Autorités Responsables de Transport
AOM	Autorité Organisatrice de Mobilité
PRIM	Plateforme Régionale d'Information pour la Mobilité

VE	Véhicule Électrique
VLS	Vélo en Libre-Service
VTC	Véhicule de Tourisme avec Chauffeur
PMV	Produit Minimum Viable (MVP Minimum Viable Product)

1.2. Documents de référence

N°	Document	Version	Date
[R01]	DAT Mon Compte Mobilité MOB	1.1	10/2022

1.3. Documents applicables

N°	Document	Version	Date
[A01]	fabmob/standard-covoiturage: Standard covoiturage dans le MaaS - MaaS standard for ridesharing (github.com)	1.0.0.alpha	29/07/2022
[A02]	TOMP-WG/TOMP-API: Transport Operator to Mobility-as-a-Service Provider-API development for Mobility as a Service (github.com)	1.3.0	17/01/2022
[A03]	le.taxi - api.gouv.fr	1.0.0	

1.4. Table des révisions

Version	Description	Auteur	Date
0.1	Initialisation du document	Capgemini	17/12/2021
1.0	Version en fin de phase de cadrage	Capgemini	19/01/2022
1.1	Version phase de Build	Capgemini	04/11/2022
1.2	Version fin d'expérimentation	Capgemini	03/03/2023

2. Contexte et motivation du projet

2.1. Ecosystème

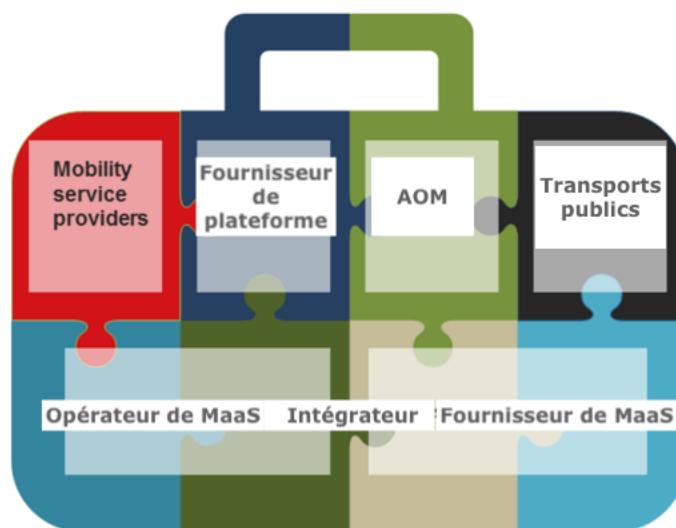


Figure 1 - Rôles dans l'écosystème des MaaS

Source : <https://www.transport20.no/wp-content/uploads/2016/06/maas.pdf>

Le projet Mon Compte Mobilité « Standardisation des MaaS » s'inscrit dans l'écosystème de la mobilité et des MaaS. Ce dernier est l'objet de nombreuses publications qui introduisent une nomenclature des rôles joués par les différents acteurs :

- Les fournisseurs de MaaS qui éditent des frameworks ou des solutions logicielles clé en main permettant de mettre en œuvre des MaaS ;
- Les intégrateurs qui prennent en charge le déploiement, la configuration, l'interconnexion et l'extension éventuelle des systèmes de MaaS ;
- Les opérateurs sont responsables du fonctionnement, de la sécurisation, de la surveillance, du maintien en condition opérationnelle et du support aux usagers ;
- Les Mobility Service Providers (MSP) fournissent des services de mobilité qui sont agrégés par les MaaS ;
- Ils s'appuient sur des services d'infrastructure ou de plateforme fournis par des tiers (fournisseurs de plateforme, Cloud ou OnPremise) ;
- Les Autorités Organisatrices de la Mobilité (AOM) assurent l'organisation du réseau de transport urbain sur leurs territoires. Elles en délèguent le plus souvent l'exploitation à des tiers.

Certains acteurs peuvent porter plusieurs de ces rôles simultanément.

Selon ces définitions, MCM Std MaaS Gateway serait :

- Une plateforme ;
- Au service des MaaS et des AOM ;
- S'appuyant sur des MSP ;
- S'appuyant sur d'autres plateformes techniques comme Azure ;
- Développée conjointement par Capgemini et la Fabrique des Mobilités ;
- Opérée dans un premier temps par Capgemini puis transmis à un tiers non identifié à ce stade.

2.2. Positionnement

2.2.1. Extension du projet Mon Compte Mobilité

Ce projet élargit grandement la visée du projet Mon Compte Mobilité.

Les travaux du projet initial MCM donnent lieu au développement et au déploiement d'un produit appelé « MOB » innovant mais spécifique à un besoin, celui de la gestion des aides à la mobilité durable et douce.

Ce besoin s'inscrit dans un contexte d'écosystème très riche et complexe, entre des collectivités (AOMs), des entreprises et des acteurs de la mobilité tels que les MaaS.

Au sein d'un tel écosystème, de nombreux enjeux de standardisation existent depuis plusieurs années et prennent de plus en plus d'importance. Parmi ces enjeux, beaucoup sont liées aux MaaS et MOB est donc directement concerné.

C'est pourquoi les travaux sur le programme MCM ont été étendus avec cet avenant au projet MCM appelé « Standardisation des MaaS ».

2.2.2. Enjeux de standardisation

Cette extension de projet vise donc à répondre à certains enjeux de standardisation, illustrés et définis ci-après.



Figure 2 - Enjeux de la Standardisation des MaaS

Pour chacun des acteurs principaux de l'écosystème présenté en amont, le projet permettra de répondre aux à leurs différents enjeux :

- **MaaS**
 - o Réduire les coûts de développement et de maintien des interfaces spécifiques avec les MSP
 - o Faciliter leur construction
 - o Faciliter l'intégration des MSP & comptes mobilité dans les MaaS pour limiter les coûts

- **MSP**
 - o Réduire la redondance et la complexité d'intégration à des plateformes de mobilité
 - o Réduire les coûts d'interfaces pour les MSP
 - o Augmenter les perspectives de croissance des MSP, notamment les petits acteurs locaux, en leur facilitant l'accès aux citoyens
- **AOM**
 - o Développer l'offre de mobilité du territoire et renforcer son attractivité
 - o Changer les comportements grâce à une mobilité plus fluide
 - o Permettre l'émergence de nouveaux acteurs engagés pour la transition énergétique
 - o Abaisser la barrière financière et technique à l'entrée pour développer des MaaS
 - o Favoriser l'harmonisation de la mobilité et faciliter la synchronisation des travaux entre les AOMs
- **Citoyen**
 - o Réduire la dispersion des offres de mobilité sur les territoires en augmentant la multimodalité et la portée géographique
 - o Faciliter l'usage de la mobilité servicielle en simplifiant et en unifiant les parcours : expérience utilisateur sans couture

La standardisation permettra de relier plusieurs MSP à un ou plusieurs MaaS via des interfaces standards ainsi que de faciliter l'intégration des MSP et de MOB / CMS dans les MaaS

moB

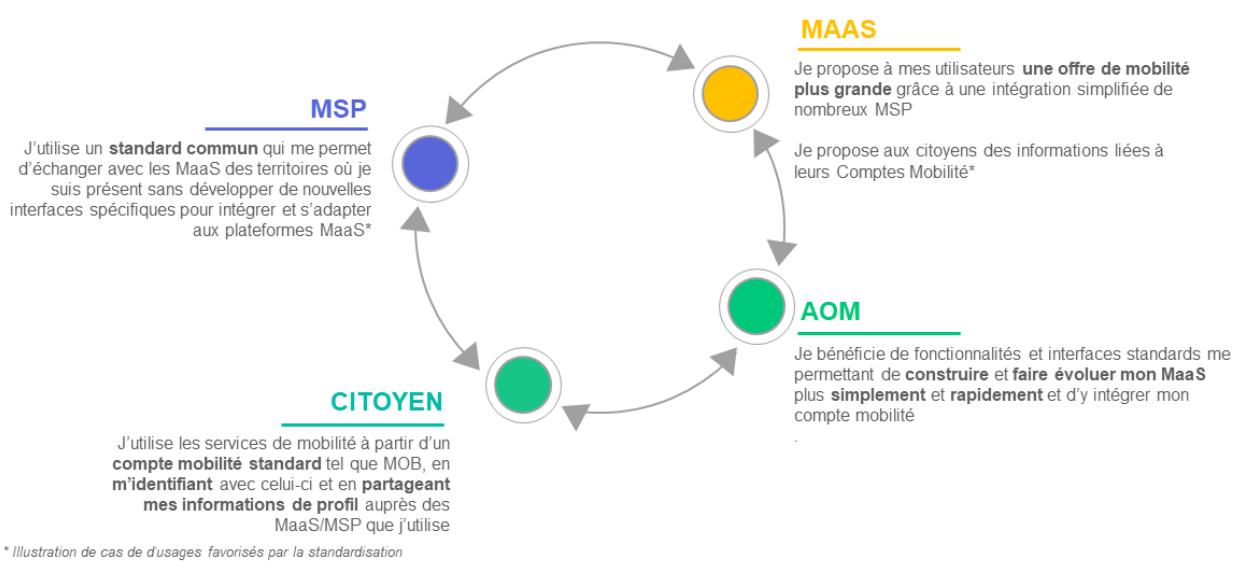


Figure 3 - Apports de la Standardisation des MaaS

2.2.3. Hub Mon Compte Mobilité

Le projet Mon Compte Mobilité devient alors un Hub concentrant les acteurs de la mobilité et plus particulièrement les MaaS et les MSP autour des grands éléments suivants :

- La **GATEWAY**, passerelle d'accès standards aux services des MSPs
- Le **COMpte MOBILITE STANDARDISE**, spécification et construction d'un compte mobilité partagé et utilisable par tous
 - o **MOB**, compte mobilité unique de gestion et d'accès aux aides à la mobilité, sera un démonstrateur de ce CMS

Le positionnement du Hub Mon Compte Mobilité consiste à promouvoir l'usage de moyens de mobilité douce en rendant plus accessibles aux citoyens les offres de mobilité et les dispositifs incitatifs correspondants proposés par

les organismes publics et les entreprises. Le Hub à l'aide de sa Gateway, doit fournir un point d'accès à ces mêmes acteurs afin de les aider à piloter leurs offres, leurs financements et optimiser la mobilité sur leurs territoires.

Le Hub ne doit pas entrer en compétition ou empiéter sur les domaines des autres acteurs privés. Il ne doit pas par exemple devenir un super-MaaS ou remplacer les supports de paiement.

Il y a un unique Hub pour le PMV.

Ce Hub pourrait être décliné selon différentes instances à l'issue de la phase d'expérimentation, selon les stratégies de déploiement et les objectifs territoriaux qui seront poursuivis.

2.3. Objectifs

L'objectif suivi est d'apporter une solution à la problématique d'universalité en définissant les principes techniques de standardisation et leurs conditions de mise en œuvre.

Derrière l'objectif global de standardisation des interfaces, nous pouvons identifier les objectifs suivants :

- Préparer la mise en œuvre d'interfaces standardisées ouverts et accessibles entre MSP et MaaS
- Fournir aux citoyens un accès agrégé et unifié à de multiples services de mobilité
- Contribuer aux développements et l'harmonisation de la mobilité
- Inciter les citoyens à utiliser des moyens de mobilités durables
- Donner plus de visibilité sur les offres de mobilités existantes

Afin de répondre à ces objectifs et de supporter l'écosystème dans les transitions vers les standards, un « optimum » Hub et sa Gateway, supportant des API standards pour les différentes verticales de mobilité, est proposé.

La Gateway propose des interfaces reliant plusieurs MSP à un ou plusieurs MaaS, et permettant aux MSP et MaaS de ne pas redévelopper des interfaces spécifiques.

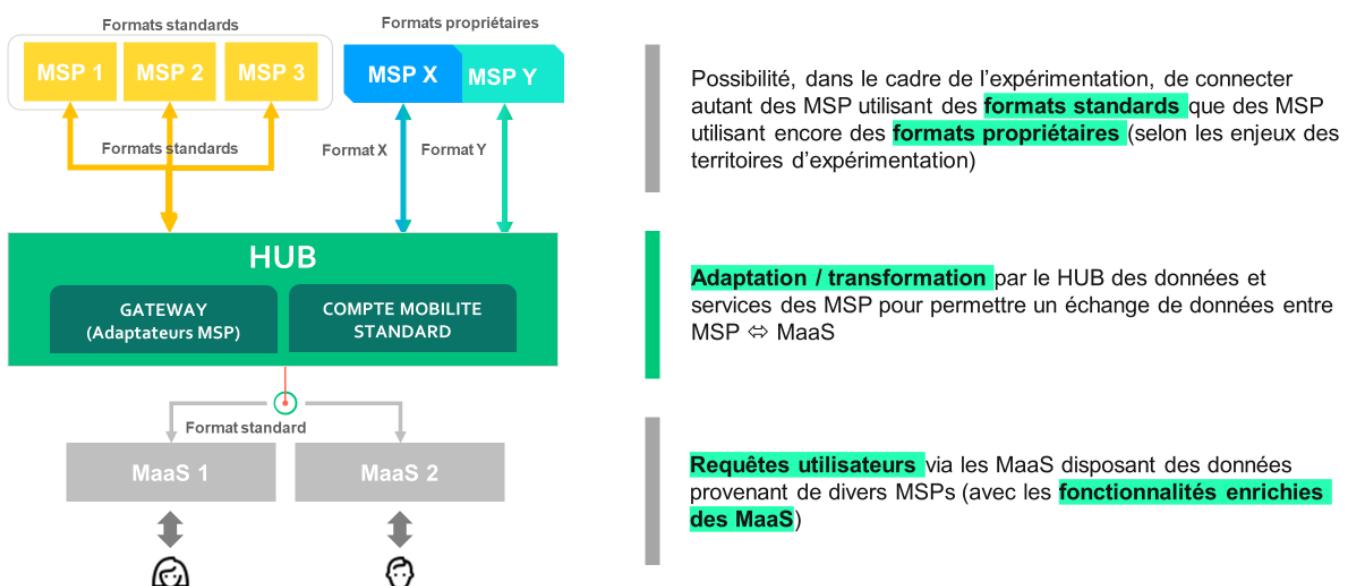


Figure 4 - La standardisation des MaaS concrétisée par la mise en place d'un Hub, qui préfigure la mise en place de standards, et d'API standardisées

2.4. Périmètre fonctionnel du PMV

Les fonctionnalités du HUB ont été définies et priorisées selon les besoins de l'écosystème et les enjeux spécifiques des partenaires pressentis pour l'expérimentation. Elles s'appuient sur des blocs fonctionnels communs :

- API de point d'entrée unique pour les MaaS
- Aiguillage des requêtes MaaS vers l'adaptateur pour le format du MSP concerné
- Envoi des requêtes aux MSP
- Recensement/découverte des services des MSPs
- Données en cache pour optimisation des réponses

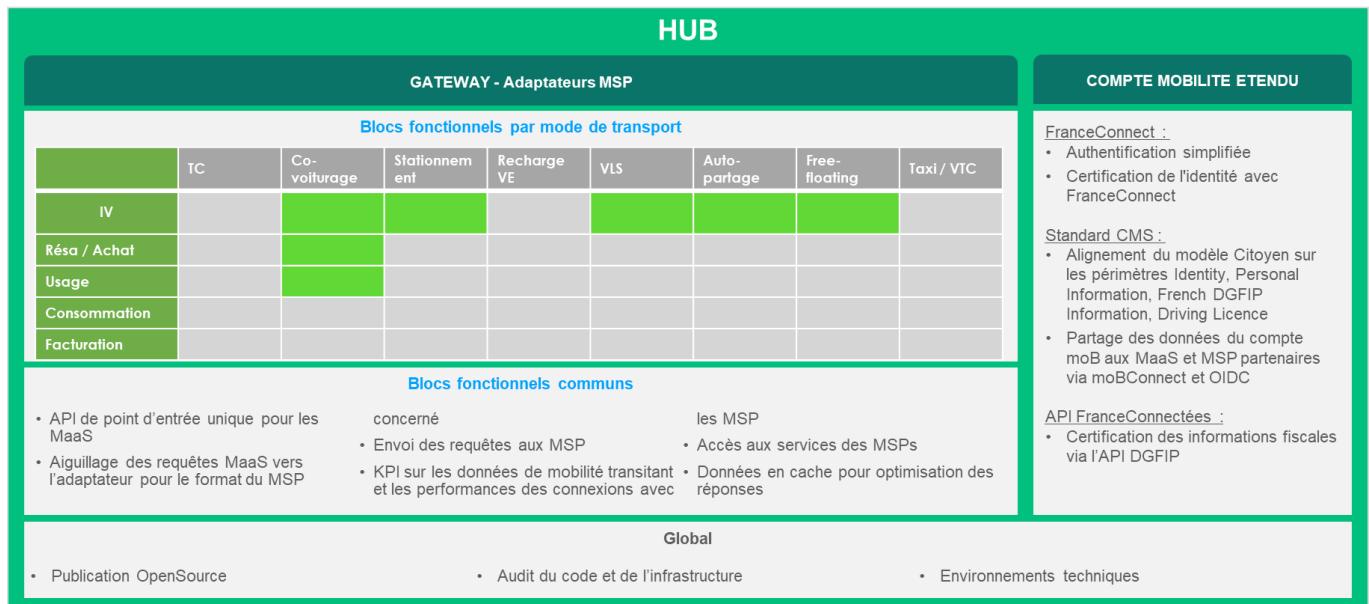


Figure 5 - Périmètre fonctionnel Hub MCM

Les cas d'usage et fonctionnalités suivants ne seront PAS couverts par le PMV :

- Interopérabilité entre les MaaS ; le HUB n'est pas un intermédiaire possible entre 2 solutions MaaS
- Services de paiement ;
- Services de SAV ; pas d'enjeux de normalisation transverse identifiés sur l'échange de tickets de SAV
- Services liés aux blocs fonctionnels en gris sur le schéma ci-dessus
 - o Transport en Commun
 - o Covoiturage - Consommation / Facturation
 - o Stationnement - Réservation / Achat / Usage / Consommation / Facturation
 - o Recharge VE
 - o VLS / Autopartage / Free Floating - Réservation / Achat / Usage / Consommation / Facturation
 - o Taxi / VTC

2.5. Contraintes

2.5.1. Existant des territoires

Le projet MCM Standardisation des MaaS a pour ambition à terme une portée nationale. Dans un premier temps, les territoires pilotes seront en nombre limité. Toutefois, il est probable que leurs maturités et leurs organisations diffèrent.

Dans tous les cas, MCM Standardisation des MaaS devra être en mesure de prendre en compte les choix déjà effectués par les régions et l'existant de la mobilité.

Une instantiation des composants sera possible selon la stratégie adoptée par les territoires.

2.5.2. Existant des standards

La construction de la Gateway s'appuiera sur les résultats des différents travaux de standardisation, notamment les spécifications des différents standards qui sortiront prochainement.

Elle s'inscrit en tant que support à la standardisation et est une plateforme permettant d'accélérer le développement et l'usage de ces standards au travers de l'écosystème de mobilité.

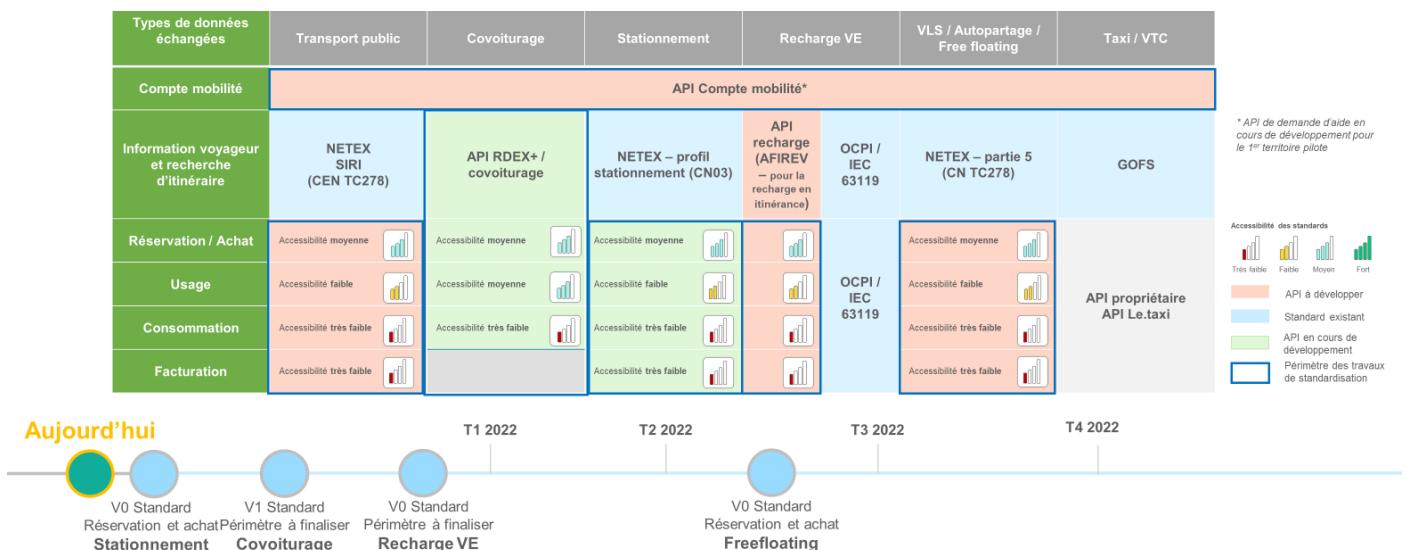


Figure 6 - Orientations sur les standards (vue fin 2021)

La standardisation va s'appuyer sur les orientations issues du GT MaaS en réunissant AOM, éditeurs de MaaS et MSP, avec des niveaux de maturité différents selon les types de mobilité.

La matrice ci-dessus définit 2 niveaux de couverture :

- **Horizontale** ; ce sont les différentes profondeurs d'intégration fonctionnelle des services d'un MSP, suivant le parcours d'un citoyen utilisant un service de mobilité
- **Verticale** ; ce sont les différents types / famille de MSPs

Dans un premier temps, peu de standards seront disponibles, il conviendra d'être en capacité de s'adapter et de proposer les meilleures alternatives si nécessaire, au regard :

- La disponibilité de standards publiés
- Le périmètre et la couverture fonctionnelle de ces standards
- Le niveau d'engagement des acteurs et d'adoption par l'écosystème

Livrables associés

Un certain nombre de livrables est attendu à propos d'un standard à intégrer.

- Document de présentation du standard

Nom et versions disponibles, cycle de vie, historique des changements

Périmètre et liste des acteurs qui le supportent

Points d'accès techniques des plateformes de test des acteurs

Cas d'usage et fonctionnalités

Points de contact et de support

- Spécifications techniques

Documentation APIs (swagger)

Jeu de données d'exemples

- Accès au dépôt du code source pour proposer des idées/contributions
- Kit de développement (SDK)

2.5.3. Protection des données

Le règlement n°2016/679, dit Règlement Général sur la Protection des Données (RGPD) doit être appliqué sur les données à caractère personnel.

2.6. Périmètre du DAT

L'architecture décrite dans ce document est relative à la phase d'expérimentation (PMV), sur sa composante GATEWAY.

La composante « Compte Mobilité Etendu » est adressée dans le document [R01].

3. Architecture conceptuelle

3.1. Hypothèses fonctionnelles

- Le périmètre de la Gateway s'appuie sur les **pans fonctionnels déterminés par le GT Architecture MaaS Gart**.
- Sur l'horizontale Informations Voyageurs (IV)/Recherche d'Itinéraires (RI), **la Gateway MCM Std MaaS couvrira l'IV, très peu la RI** car celle-ci est portée par les solutions MaaS (pas de calculateur, ni d'aggrégateur, seulement un accès à la demande de trajet Covoiturage ou Taxi/VTC).
- Un **standard Français**, s'il existe, sera pris en charge **en priorité**. Le standard TOMP-API sera pris pour référence sinon.
- L'API MaaS de la Gateway MCM Std MaaS sera construite à partir des **spécifications des standards** et de la **TOMP-API en priorité**. Elle pourra se baser également sur la spécification de l'[API MaaS](#) proposé par la Gateway MaaX d>IDFM.
- Au cours de la phase de build, un **standard français de covoiturage** a été publié, par conséquent, il a été pris comme référence pour tous les MSPs de covoiturage

3.2. Acteurs

La Gateway MCM Std MaaS est une plateforme de médiation entre les MaaS et les MSP. Elle expose des services de façon standard à des MaaS, les seules applications clientes de la plateforme.

Les MSP n'ont pas vocation à devenir client de la Gateway. En revanche, leurs services seront relayés par la GW auprès des MaaS.

Côté humain, des administrateurs techniques et fonctionnels auront accès aux outils de gestion et/ou de monitoring de la plateforme.

Acteurs et rôles	Auth.	Description (PMV)
MaaS	Oui	<p>Les MaaS sont les clients principaux de la GW.</p> <p>Ils s'authentifient auprès de la GW et accèdent à des services via un unique point d'entrée, selon des interfaces API standardisées.</p>
MSP	Non	<p>Les MSP sont intégrés dans la Gateway.</p> <p>Elle est cliente des MSP et se connectent à leurs interfaces API, via des interfaces standardisées ou propriétaires.</p> <p>Ces API donnent accès aux services proposés par les MSP.</p>
Administrateur AOM/MaaS connecté	Oui	Ils suivent les flux fonctionnels des MSP, listent les trajets les plus ou moins fréquentés et configurent les MaaS/MSP de leur territoire.
Exploitant plateforme HUB	Oui	Ils configurent les MaaS/MSP et montrent les flux techniques de la GW

Plateforme Data	Oui	<p>Une plateforme Data d'une collectivité par exemple peut aussi être cliente de la Gateway.</p> <p>Elle s'authentifie auprès de la GW et accède à des données/services via le même unique point d'entrée que pour les MaaS, selon des interfaces API standardisées.</p>
-----------------	-----	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.2.1. Liste des MSP supportés

Dans cette section, les acteurs MaaS/MSP supportés par la Gateway seront listés par verticale. Le niveau d'intégration des services et les formats d'échanges utilisés (standards ou non) seront précisés.

Acteur	Mode	Niveau d'intégration	Formats d'échange
Mobicoop	Covoiturage	IV / Réservation / Usage	Standard Covoiturage
Coopgo	Covoiturage	IV / Réservation / Usage	Standard Covoiturage
Karos	Covoiturage	IV / Réservation / Usage	Standard Covoiturage
BlablaCar	Covoiturage	IV / Réservation / Usage	Standard Covoiturage
Klaxit	Covoiturage	IV / Réservation / Usage	Standard Covoiturage

3.3. Processus métier

3.3.1. Parcours MaaS

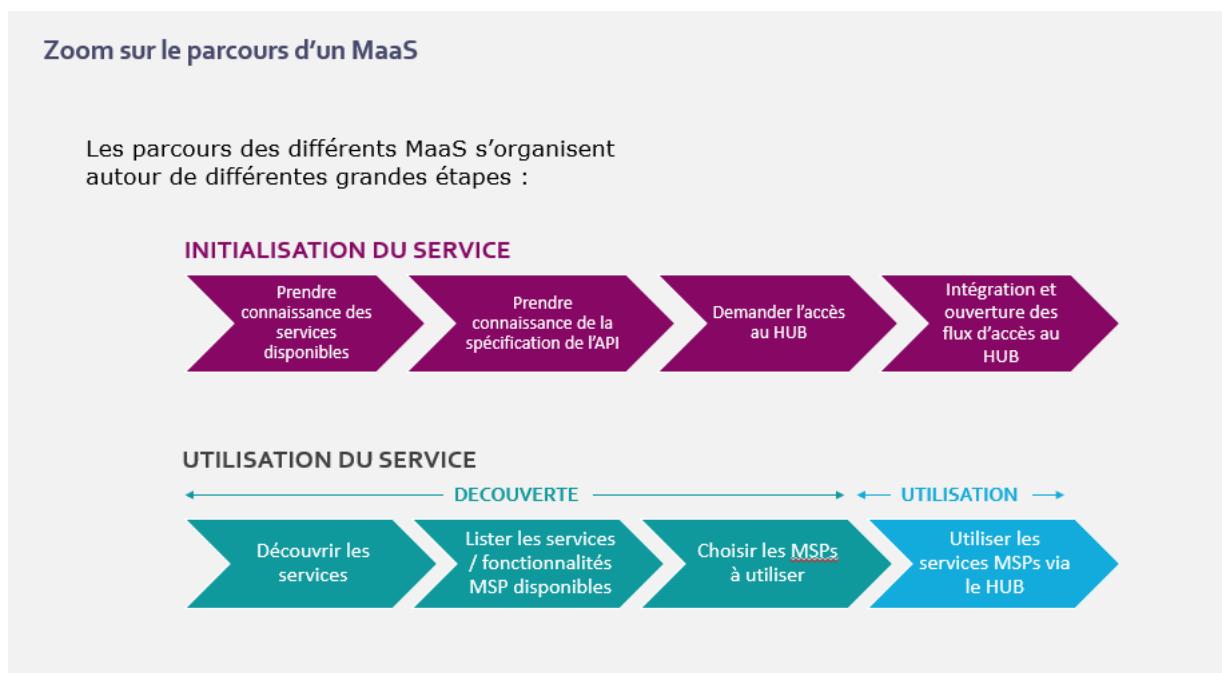


Figure 7 - Processus et parcours MaaS

3.3.2. Parcours d'intégration MSP

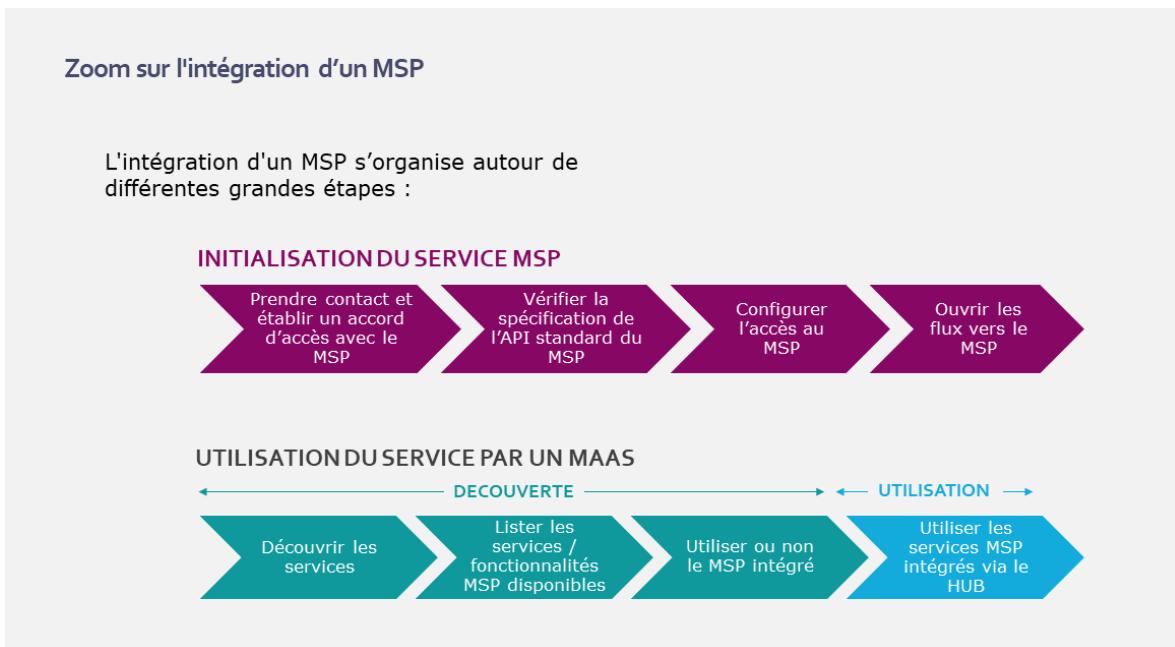


Figure 8 - Processus d'intégration MSP

3.4. Cas d'utilisation

3.4.1. Périmètre du PMV

Les cas d'usage suivants seront couverts par le PMV :

1. Définition d'une API en tant que point d'entrée unique et standardisé pour les MaaS
2. Inscription des MaaS, MSP
3. Définition du catalogue d'offres de mobilité, information des usagers
Authentification des MaaS dans le GW, et Authentification de la GW auprès des MSP
4. L'Information Voyageur (IV) pour les verticales Covoiturage, Stationnement en infra, Recharge VE, VLS, Autopartage, Free Floating et Taxi/VTC
5. La Réservation/Achat pour les verticales Covoiturage, Stationnement en infra, Recharge VE, VLS, Autopartage et Free Floating
6. L'Usage pour les verticales Stationnement en infra, Recharge VE, VLS, Free Floating et Taxi/VTC
7. La Consommation pour les verticales Stationnement en infra, Recharge VE, VLS, Autopartage et Free Floating
8. La Facturation pour les verticales Stationnement en infra, VLS et Free Floating
9. Supervision des services MSPs les plus utilisés

3.4.2. Vue d'ensemble des cas d'utilisation

Le diagramme de cas d'utilisation qui suit constitue une tentative de capturer l'ensemble du périmètre fonctionnel sous une forme synthétique et standard (UML). Il se lit de la façon suivante :

- La frontière des systèmes est représentée par un cadre.
- Un acteur est représenté par un bonhomme allumette.

- Deux acteurs peuvent être liés entre eux par une relation de généralisation/specialisation aussi nommée relation d'héritage représentée par une flèche dont l'extrémité est creuse. Elle se lit « *est un* » dans le sens de la flèche. Par exemple : « l'acteur Administrateur AOM *est un* Utilisateur Authentifié ».
- Chaque cas d'utilisation est représenté par une ellipse. Un cas d'usage constitue la raison pour laquelle l'utilisateur souhaite interagir avec le système. Par exemple, « Gérer le contenu éditorial ».
- Un acteur peut être relié à un cas d'utilisation primaire par une ligne pleine non orientée. Dans ce cas, le cas d'usage est primaire et la relation représente l'intention/la finalité de l'utilisateur. Par exemple, « en tant qu'Administrateur Fonctionnel MCM, je souhaite Gérer le contenu rédactionnel ».
- Un cas d'usage primaire peut être lié à un ou plusieurs cas d'usage secondaires à l'aide de flèches orientées, en pointillés. Cette relation est assortie d'un stéréotype « *include* » ou « *extend* » signifiant, respectivement, que le cas cible est obligatoire inclus/requis ou optionnellement étendu par le cas d'utilisation source. Par exemple, « Gérer le profil requiert une authentification préalable ». Autre exemple : « Afin moment de rejoindre une communauté, je peux soumettre un justificatif ».

Par ailleurs, nous avons opté pour le code couleur suivant :

- Apparaissent en **vert** tous les cas d'utilisation accessibles sans authentification préalable par le Citoyen. Les fonctionnalités correspondantes sont par conséquent accessibles aux visiteurs anonymes, c'est-à-dire n'étant pas inscrit ou ne s'étant pas connectés avec leurs identifiants/mots de passe sur le MaaS.
- Même si le Citoyen n'est pas identifié/connecté au MaaS pour certaines actions (en vert), le MaaS est lui obligatoirement authentifié et connecté auprès de la Gateway pour accéder aux services.
- Nous avons représenté en **rouge** les cas d'utilisation disponibles uniquement après une authentification réussie du Citoyen auprès du MaaS.

Le schéma ci-dessous décrit les cas d'usage couverts par le PMV du point de vue d'un Citoyen dans le MaaS. La Gateway supporte les cas d'usage du MaaS déduits de ces cas d'usage Citoyen.

Cas d'usage Citoyen dans le MaaS supportés par la Gateway

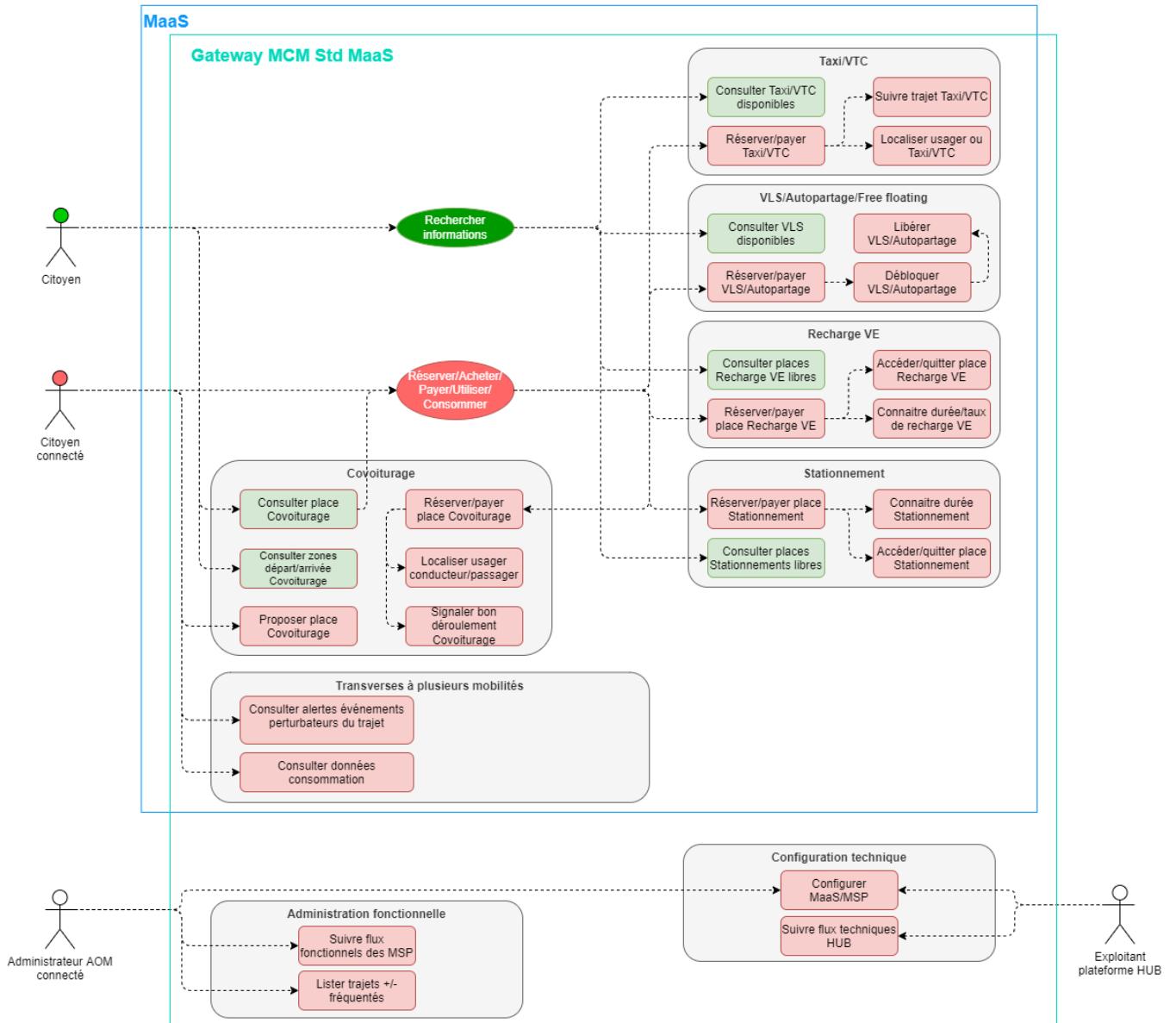


Figure 9 - Cas d'utilisation Citoyen dans le MaaS supportés par la Gateway

3.4.3. Point de vue MaaS

Le schéma ci-dessous décrit les cas d'usage couverts par le PMV du point de vue du système MaaS (pour les fonctionnalités disponibles à un citoyen via le MaaS), ainsi que du point de vue d'un administrateur du MaaS. Dans les deux cas, une connexion authentifiée du MaaS ou de son administrateur est nécessaire.

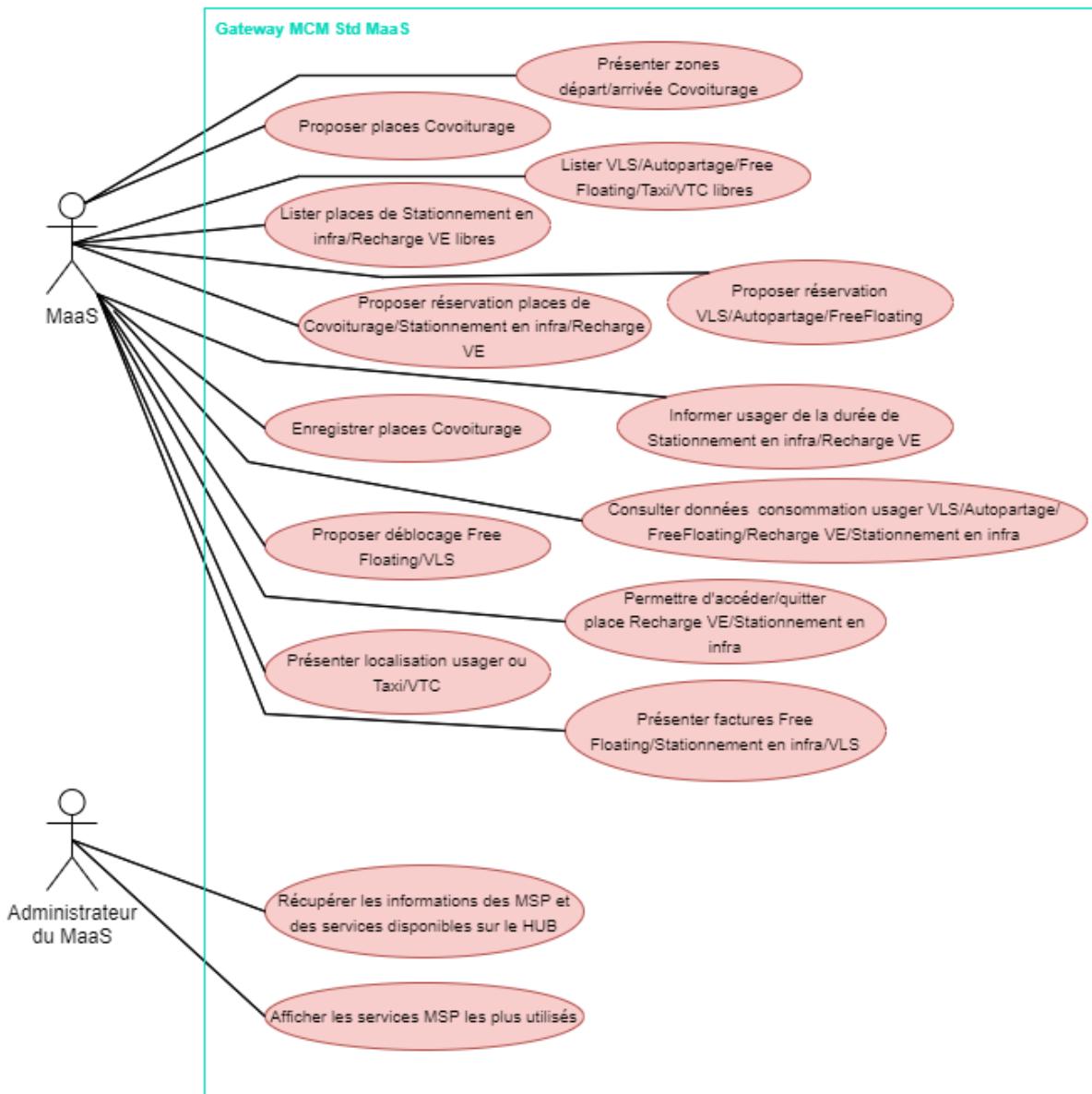


Figure 10 - Point de vue MaaS

3.4.4. Point de vue de l'administrateur AOM

Le schéma ci-dessous décrit les cas d'usage couverts par le PMV du point de vue d'un administrateur AOM. Celui-ci sera authentifié lors de la connexion à la Gateway.

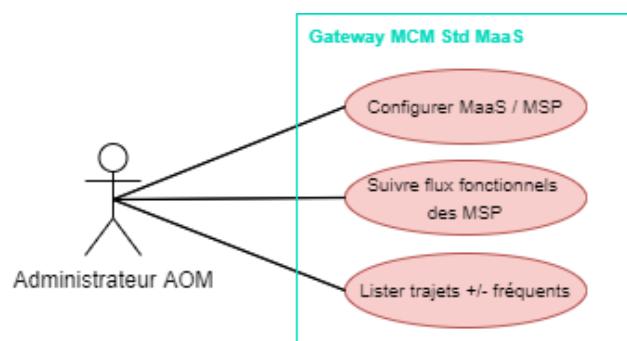


Figure 11 - Point de vue de l'administrateur AOM

3.4.5. Point de vue de l'exploitant plateforme HUB (administrateur technique)

Le schéma ci-dessous décrit les cas d'usage couverts par le PMV du point de vue d'un exploitant de la plateforme HUB. Celui-ci sera authentifié par une connexion à la Gateway.

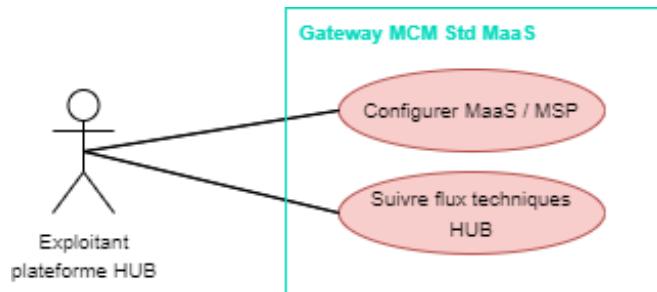


Figure 12 - Point de vue de l'administrateur technique

3.5. Cartographie fonctionnelle

Ce tableau de cartographie fonctionnelle présente les cas d'usage couverts par le PMV du point de vue d'un citoyen qui y accède via un MaaS connecté à la Gateway.

Les cas d'usage sont classés par **type de mobilité** (communément appelé « verticale ») et par **domaine fonctionnel** (également appelé « horizontale »).

	TC	Covoiturage	Stationnement (en infra)	Recharge VE	VLS	Auto-partage	Free-floating	Taxi / VTC
IV		Connaitre places disponibles	Connaitre places disponibles	Connaitre places disponibles	Connaitre véhicules disponibles	Connaitre véhicules disponibles	Connaitre véhicules disponibles	Connaitre véhicules disponibles
Résa / Achat		Réserver place	Réserver/payer place Connaitre durée stationnement	Réserver/payer place Connaitre durée/taux recharge	Réserver/payer véhicule	Réserver/payer véhicule	Réserver/payer véhicule	
Usage		Envoyer message Changer statut de réservation	Accéder/quitter place	Accéder/quitter place	Débloquer véhicule		Débloquer véhicule	Connaitre localisation usager ou véhicule
Consommation			Connaitre ses données de consommation	Connaitre ses données de consommation	Connaitre ses données de consommation	Connaitre ses données de consommation	Connaitre ses données de consommation	
Facturation			Accéder à ses factures		Accéder à ses factures		Accéder à ses factures	

Figure 13 - Cartographie fonctionnelle Gateway

3.6. Principaux concepts métier

3.6.1. Diagramme

Le modèle de la Gateway MCM Std MaaS couvre 4 principaux domaines métier : les MSP, l'information Voyageur (IV) et partiellement la recherche d'itinéraire (RI), la réservation et l'usage, la consommation et la facturation.

Ci-dessous une vue générale du modèle de données :

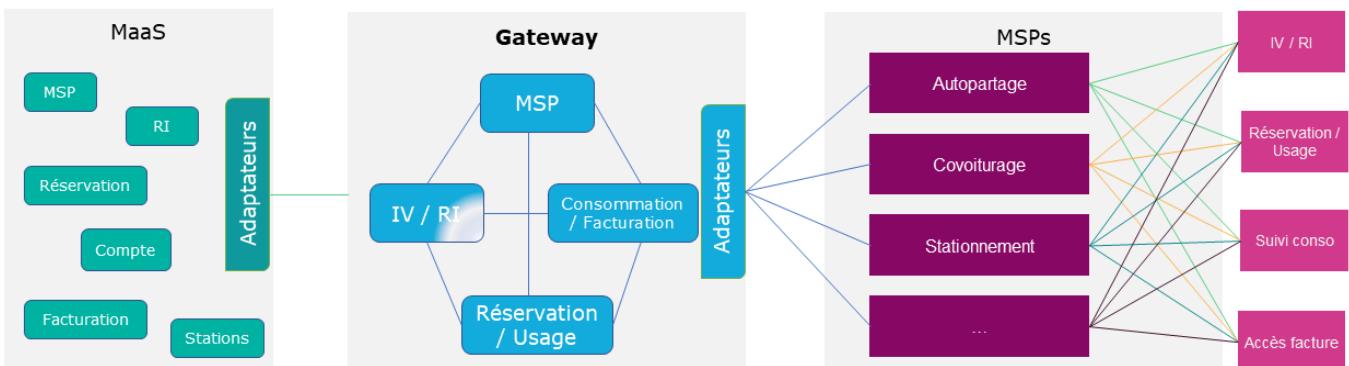


Figure 14 - Modèle de données Gateway (domaines couverts)

Les diagrammes ci-dessous constituent un exemple de définition des entités manipulées par le système.

Le modèle de données de la GW MCM Std MaaS sera construit dans un second temps.

3.6.2. Modèle de données

La base de données GW contient deux principaux schémas : MSP et configuration

Le schéma **MSP** permet de stocker toutes les données relatives aux partenaires nécessaires pour effectuer des appels, traduire les requêtes / réponses ainsi que les adapter aux formats standards.

Le second schéma **Configuration** autorise quant à lui l'activation ou la désactivation de la fonction de mise en cache sur l'ensemble des microservices de la GW. Lorsqu'un appel est effectué depuis le MaaS, la Gateway va chercher la réponse dans le cache au lieu de faire un appel aux MSPs pour récupérer les réponses. Cette fonctionnalité offre un gain de temps et de performance.

Le modèle de données de la Gateway est représenté par le schéma ci-dessous.

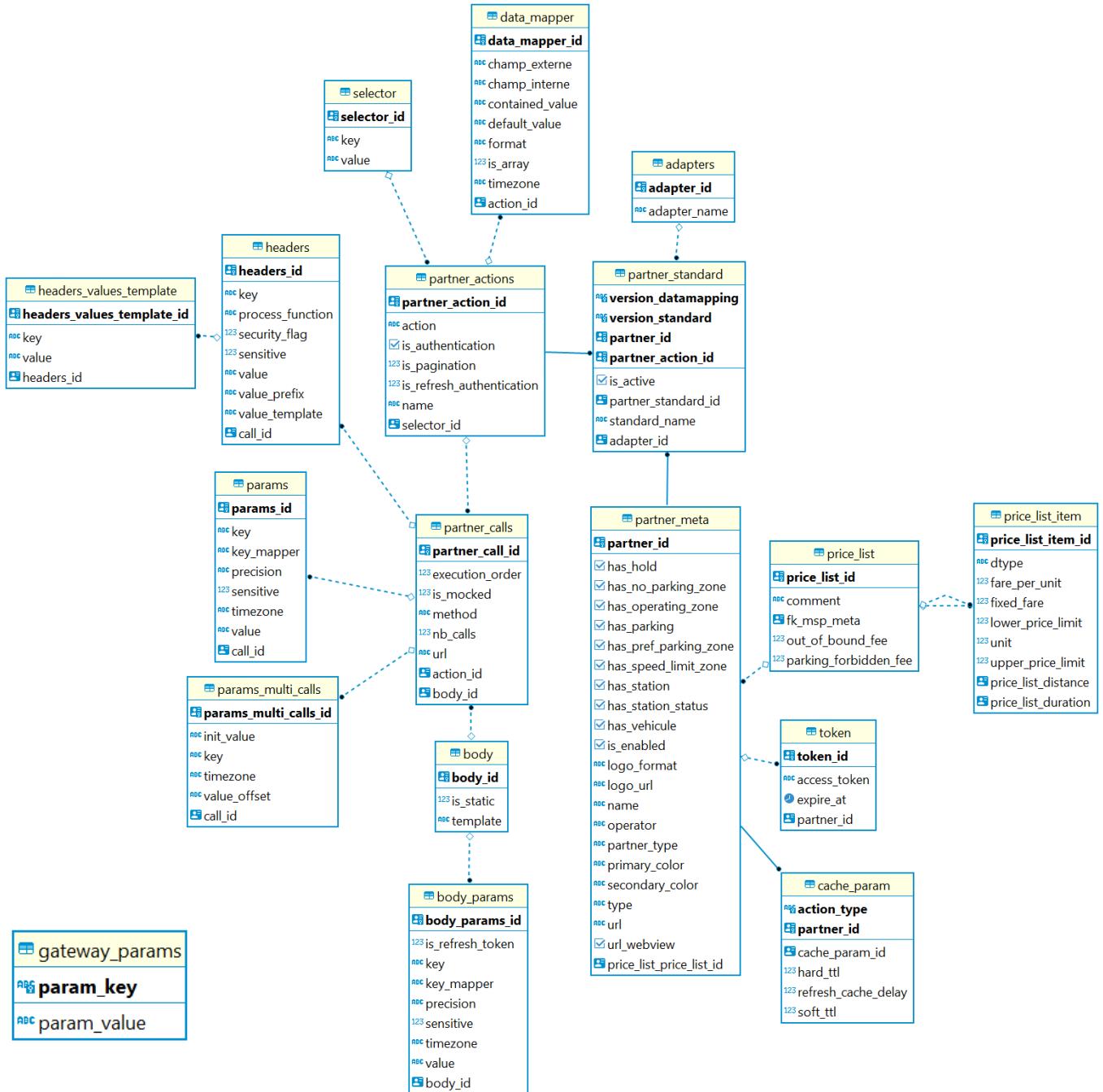


Figure 15 - Configuration système : modèle de données GW

3.6.3. Entités

Les concepts clés sont décrits de façon plus détaillée dans les paragraphes qui suivent.

Partner Meta

Un **Partner** est un objet central du système. C'est un fournisseur de services de mobilité, il peut être un MSP ou un MaaS. Le modèle de données d'un partenaire définit ses caractéristiques principales et ses données statiques tel que le nom, le code, le logo ou encore le type d'opérateur (avec ou sans véhicule, station, zone d'opération, ...).

Les tables **msp_meta**, **price_list** et **price_list_item** regroupent ces données.

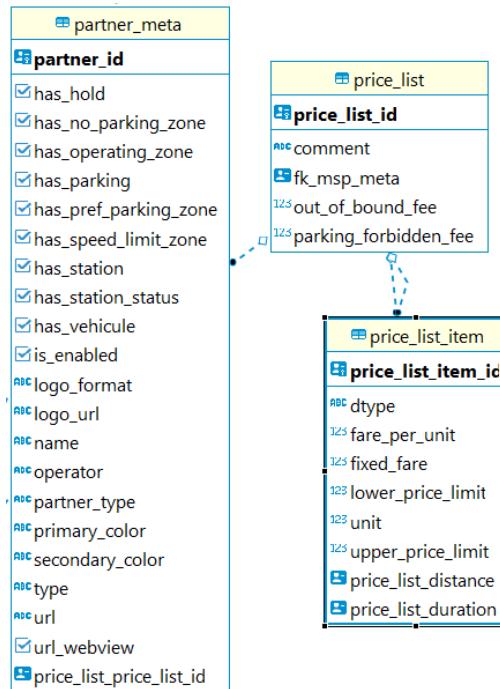


Figure 16 - MSP : exemple de Modèle de données

Partner Standard

Pour un partenaire / MSP en base, il est nécessaire de savoir quelles actions il expose (recherche des véhicules, réservation d'un véhicule ...) et quel standard est suivi pour construire les requêtes et réponses associées à chacune.

La table **partner_standard**, permet de faire cette association partner/action/versions (du standard et du datamapping).

Partner actions

Les tables **partner_actions** et **selector**, décrivent une action (SEARCH_VEHICULES, BOOK_VEHICLE, SEARCH_STATIONS ...) et permettent d'aiguiller une requête en fonction de l'action correspondante.

A chaque action sont associés le ou les calls à effectuer afin d'obtenir une réponse auprès du MSP / Partner.

La table selector permet de retrouver le sous-objet attendu par l'action dans la réponse du MSP / Partner.

Data Mapper

La persistance en base du DataMapping permet de stocker toutes les informations nécessaires pour la configuration des requêtes vers des partenaires et les traduire au format exposé par la Gateway. Cette traduction est réalisée en convertissant un champ utilisé par un partenaire en champ utilisé par la Gateway, en prenant en compte la date et le fuseau horaire en fonction de la position du partenaire.

Cette persistance est représentée par la table **data_mapper**.

Partner calls params

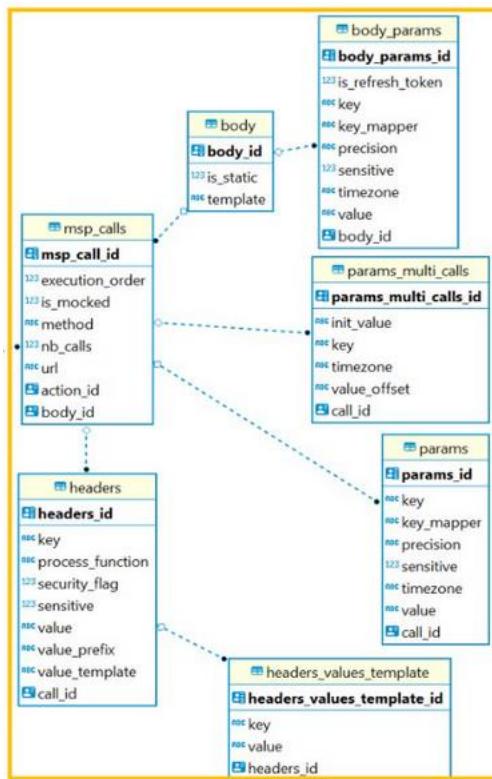


Figure 17 - MCD tables msp calls paramétrage

Ce bloc de tables décrit tous les paramètres nécessaires aux appels http (url, méthode, body, paramètres, headers).

Une fois une réponse obtenue auprès du MSP, la table data_mapper, expose la correspondance entre les champs externe (du MSP) et les champs internes (de la gateway).

Ces informations permettront de convertir la réponse du MSP au format Gateway.

Les tables contenues dans ce bloc sont décrites ci-dessous :

- La table **partner_calls** permet de décrire les appels associés à une action avec l'ordre d'exécution de chacun.
- La table **headers** donne les headers à appliquer à un call. Elle prend en compte la sensibilité des champs censurés.
- La table **params** permet de trouver les différents paramètres de l'appel (call) http et leurs valeurs.
- Dans le cas d'un appel de type "multi-call" qui nécessite donc plusieurs appels consécutifs, la table **param_multi_calls** donne les clé-valeur des paramètres associés.
- Certaines requêtes peuvent contenir un body. La table **body** donne le template/schéma de celui attendu par le MSP pour un call donné (la table msp_calls est associé à son body via le body_id)
- La table **body_params** permet de récupérer les valeurs (soit directement en base soit via leurs clé dans le body de la requête originelle) des différents champs du body.template

Configuration système

En complément du modèle de données de l'API MaaS, la Gateway MCM Std MaaS dispose d'un modèle de données DDL pour la configuration du système. Il détient l'intelligence principale de la GW car sa souplesse permet d'ajouter rapidement un Partner par simple configuration.

Ce modèle permet de référencer les informations concernant un MSP et ses paramètres techniques d'intégration :

- Nom, code, logo
- Caractéristiques de mobilité
- Liste des actions/opérations supportées
- URL et paramètres par action
- Mapping des attributs

Toute la partie configuration des MSPs est à paramétrier en BDD, il n'y a pas d'IHM ou de console d'administration à ce stade. Les opérations de rafraîchissement des données sont accessibles par l'API admin.

3.7. Exigences fonctionnelles

3.7.1. Authentification

La valeur de la Gateway « Standardisation des MaaS » réside dans le fait de simplifier l'accès aux services MSP pour les MaaS.

Le MaaS peut découvrir les services MSP / Partners disponibles et les utilise.

Une fois intéressé, le MaaS demande à s'inscrire dans la GW.

Une fois sa demande approuvée, il peut consommer les services offerts par la GW.

3.7.2. Propagation d'identité

Pour l'ensemble des fonctionnalités qui nécessitent une connexion/authentification du citoyen, la propagation du jeton d'authentification du citoyen doit être assurée jusqu'au MSP destinataire de la requête.

Autrement dit, le MaaS appelle le service MSP via la Gateway pour le compte du citoyen.

3.7.3. Conformité RGPD

Identification des données personnelles

Il n'y a pas de données personnelles identifiées dans le cadre du PMV.

Aucune donnée personnelle ne sera stockée.

Ce sont des données techniques (identifiants techniques) qui transitent.

Sous-traitance : identifiée et contractualisée

Les données ne seront pas confiées à un prestataire.

Transferts : respect des obligations en matière de transfert de données en dehors de l'Union européenne

Les données ne seront pas transférées en dehors de l'Union européenne.

3.8. Exigences non fonctionnelles

3.8.1. Volumétrie cible

La volumétrie dynamique sur la Gateway MCM Std MaaS dépend du nombre d'interactions entre le backend des MaaS (clients) et la Gateway pour honorer les sollicitations : IV autour de moi, RI, réservation, usage, consommation, facturation.

Il n'y a pas de volumétries connues à ce stade.

Cependant, l'idée est que la Gateway MCM Std MaaS puisse supporter les volumétries équivalentes à celles rencontrées par les MaaS qui déportent une partie de leurs flux MSP vers la GW.

3.8.2. Rétention des données

Les seules données qui vont être retenues dans la Gateway :

- Les données statiques stockées dans le cache (liste des parkings, liste des stations de recharge VE, ...)

3.8.3. Portabilité

À l'issue de la phase d'expérimentation, le système MCM devra être en capacité d'être transmis à un opérateur tiers restant à définir. De même l'infrastructure post phase d'expérimentation reste à définir.

- Nécessité de pouvoir transporter le système à moindre frais sur une infrastructure différente
- Nécessité de maîtriser (limiter ou identifier clairement) les adhérences au IaaS/PaaS

3.8.4. Performance

Non défini à ce stade.

La performance de la GW sera mesurée en tenant compte des performances proposées par les services des partenaires.

3.8.5. Disponibilité

La plage d'ouverture des services de la Gateway est de **24h24, 7j/7** hors interruptions programmées.

RPO : la Gateway MCM Std MaaS ne stockant pas de données critiques, l'expression d'un RPO faible est approprié (24h de production)

RTO : en cas de dysfonctionnement d'un composant d'infrastructure le délai de remise en fonctionnement est de 2h

Sauvegardes : la fréquence des sauvegardes est alignée sur le RPO

3.8.6. Confidentialité

Les données d'utilisation par les MaaS sont anonymisées et leur accès est cloisonné par MaaS/AOM.

3.8.7. Données à caractère personnel

Stockage

La Gateway MCM Std MaaS ne stocke pas de données confidentielles hormis les éléments de connexions aux MSP : il n'y a pas de gestion de données personnelles ni de données bancaires.

Dans le cas des opérations de réservation, la Gateway MCM Std MaaS manipule un élément d'authentification de l'utilisateur avec son compte MSP. Cet élément est tracé dans les logs de l'application pour permettre audit, analyse et diagnostic en cas d'erreur.

Chiffrement

Les données sensibles doivent être chiffrées :

- Au repos
- En transit

3.8.8. Intégrité des données

3 cas de figure à prendre en compte :

- S'assurer que les données ne sont pas altérées au repos
- Ni lors d'un transfert
- Au cours des transactions applicatives, les modifications doivent être atomiques

3.8.9. Traçabilité

Toute modification de l'état du système doit être journalisée.

Toute connexion doit être tracée.

Toute lecture d'information sensible.

Toute modification d'information sensible donne lieu à une notification.

Toute opération d'administration.

Format : quel identifiant a utilisé tel service, et à quelle date ?

3.8.10. Non répudiation

Rendue possible grâce aux traces et à l'auditabilité. Aucune exigence connue sur ce point.

3.8.11. Extensibilité

Le système MCM Std MaaS GW doit être extensible et doit permettre :

- D'intégrer de nouveaux MaaS
- D'intégrer de nouveaux MSP
- De se mettre à jour sans indisponibilité de service, à chaud, par simple paramétrage

3.8.12. Résilience

Le système étant par nature fortement connecté à des tiers, il doit résister à une défaillance de l'un des composants. Une panne ou une indisponibilité d'un tiers ne doit pas compromettre la stabilité de la GW MCM Std MaaS.

3.8.13. Interopérabilité

Aucune exigence d'interopérabilité entre deux instances de GW n'est retenue.

3.8.14. Démarche open source

La Gateway est un objet de bien commun, dont le code source sera en libre accès, sous une licence appropriée à la réutilisation et facilitant la contribution.

Les schémas d'architecture, ce DAT, le code source seront documentés et publiés selon des règles de validation et d'approbation à définir.

4. Architecture logique

4.1. Architecture Hub

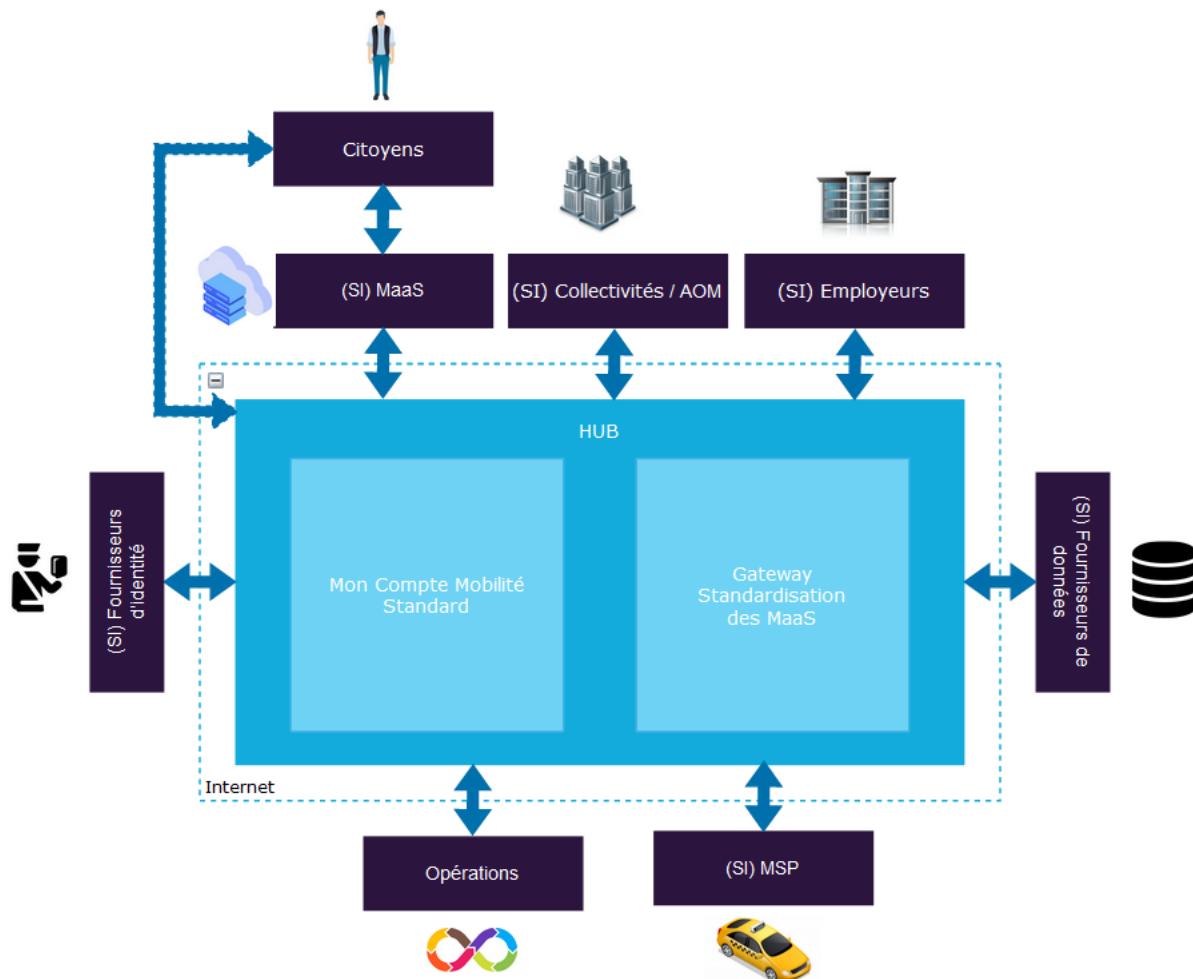


Figure 18 - Architecture générale Hub MCM

Le HUB Mon Compte Mobilité comporte à la fois Mon Compte Mobilité Standard (extension standardisée de MOB) et la Gateway MCM Std MaaS.

Une caractéristique importante de ce HUB réside dans le fait qu'il est susceptible d'initier de nombreuses interactions avec de multiples systèmes externes.

Nous avons présenté les principaux acteurs pouvant solliciter le HUB Mon Compte Mobilité : les Citoyens, les opérateurs humains intervenant au sein des Collectivités et des Employeurs affiliés. Les citoyens peuvent se connecter au HUB soit directement dans le cas de l'utilisation de leur compte moB standard, soit par le biais d'un système externe (SI MaaS).

Le HUB Mon Compte Mobilité devra également servir et s'appuyer sur des SI externes : fournisseurs d'identité, fournisseurs de données certifiées, SI des Collectivités / des Entreprises, SI des MaaS et SI des MSP. SI des Collectivités et des Entreprises.

Enfin, des points d'accès sont provisionnés (bloc opérations), ils sont destinés aux équipes de développement et d'exploitation de la plateforme.

Architecture générale détaillée

Légende

- Données théoriques et temps réel
- Données d'usage
- Données du compte mobilité

ACTEUR
FONCTIONNALITÉ
MSP
ADAPTATEUR

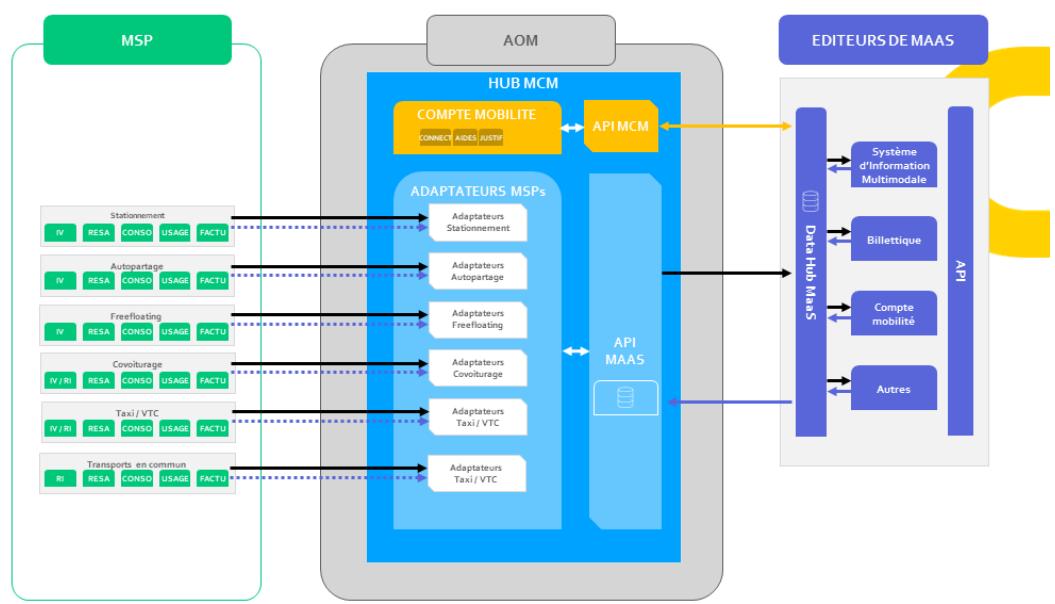


Figure 19 - Architecture détaillée Hub MCM

4.2. Architecture Gateway

L'un des objectifs majeurs de la Gateway est de proposer des interfaces standardisées ouvertes et accessibles entre MSP et MaaS. Ainsi, ses principales interactions sont initiées avec ces acteurs externes : les SI MaaS et les SI MSP.

L'autre acteur qui pourrait solliciter la Gateway MCM Std Maas est le SI AOM (Autorités Organisatrices de la Mobilité) qui assurent l'organisation du réseau de transports sur leurs territoires.

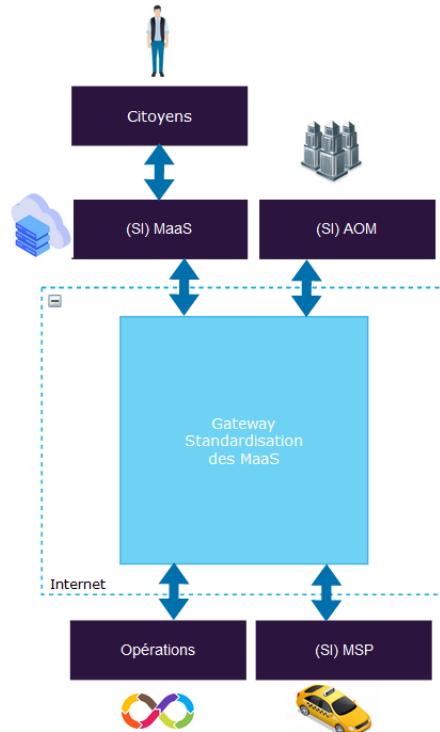


Figure 20 - Architecture générale Gateway MCM Std MaaS

4.3. Scénarios d'intégration

La Gateway est un objet qui se veut ouvert et instanciable à différents étages.

Plusieurs scénarios ont été identifiés et étudiés pour l'intégration et l'articulation du HUB MCM avec l'écosystème :

4.3.1. Hub national

- Le Hub MCM peut s'orienter vers un objet unique national permettant d'associer plusieurs solutions MaaS à plusieurs MSP, sur une plateforme partagée.
- Le Hub national serait l'objet central de tous MaaS et MSP impliqués dans la démarche.
- Cet objet faciliterait une interopérabilité globale entre les acteurs, selon des formats standards.

4.3.2. Hub régional

- Le Hub MCM peut s'orienter vers un standard réplicable à l'échelle régionale
- Chaque région garde la maîtrise en termes de financement, de gestion du Hub, d'infrastructure. Un MaaS aura la possibilité de déployer le Hub dans son SI et bénéficiera des MSPs disponibles dans sa région.

4.3.3. Hub territorial (local)

- Au sein d'un Hub national, le Hub standardisé peut également s'orienter vers un standard territorial (local) permettant à un MaaS ou une AOM de pouvoir déployer le Hub dans son SI avec ses préférences de configuration (version d'API supportées, fonctionnalités disponibles, paramètres spécifiques) et bénéficier d'un certain niveau d'isolation, des configurations et des extensions distinctes.
- Un MaaS et/ou une AOM doit pouvoir déployer le Hub local dans son SI, donc disposer de l'ensemble des composantes, de la documentation, et d'une version allégée en termes d'architecture technique, plus adaptée à un contexte « mono-MaaS », il peut être instancié différemment d'un territoire à un autre selon différentes stratégies.

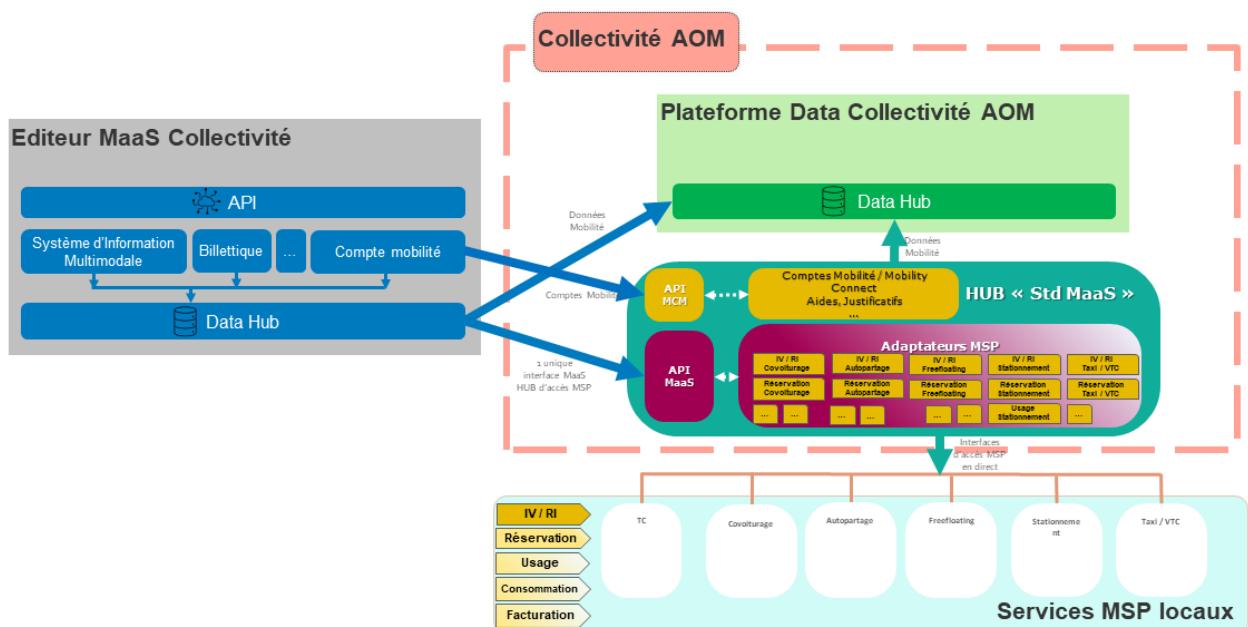


Figure 21 - Scénario d'intégration territorial

4.3.4. Hub fédéral

- Le Hub standardisé peut aussi être dédié à une fédération d'acteurs et offrir des fonctionnalités d'administration et de branding propres à cette fédération, c'est à dire un ensemble de règles et de fonctionnalités.
- Tout comme le Hub local, le Hub fédéral est un Hub national, mutualisé ou être hébergé, ou simplement exploité de manière autonome.
- Au sein d'un Hub fédéral, comme pour tout Hub, chaque acteur doit disposer de possibilités d'administration et de configuration, dans le respect des règles collectives du Hub.
- Une fédération doit pouvoir déployer le Hub en propre, donc disposer de l'ensemble des composantes, de la documentation. Une version allégée en termes d'architecture technique, a aussi son utilité, car il est possible d'être dans un contexte qui ne requièrent pas un haut niveau de scalabilité et un haut niveau de service.

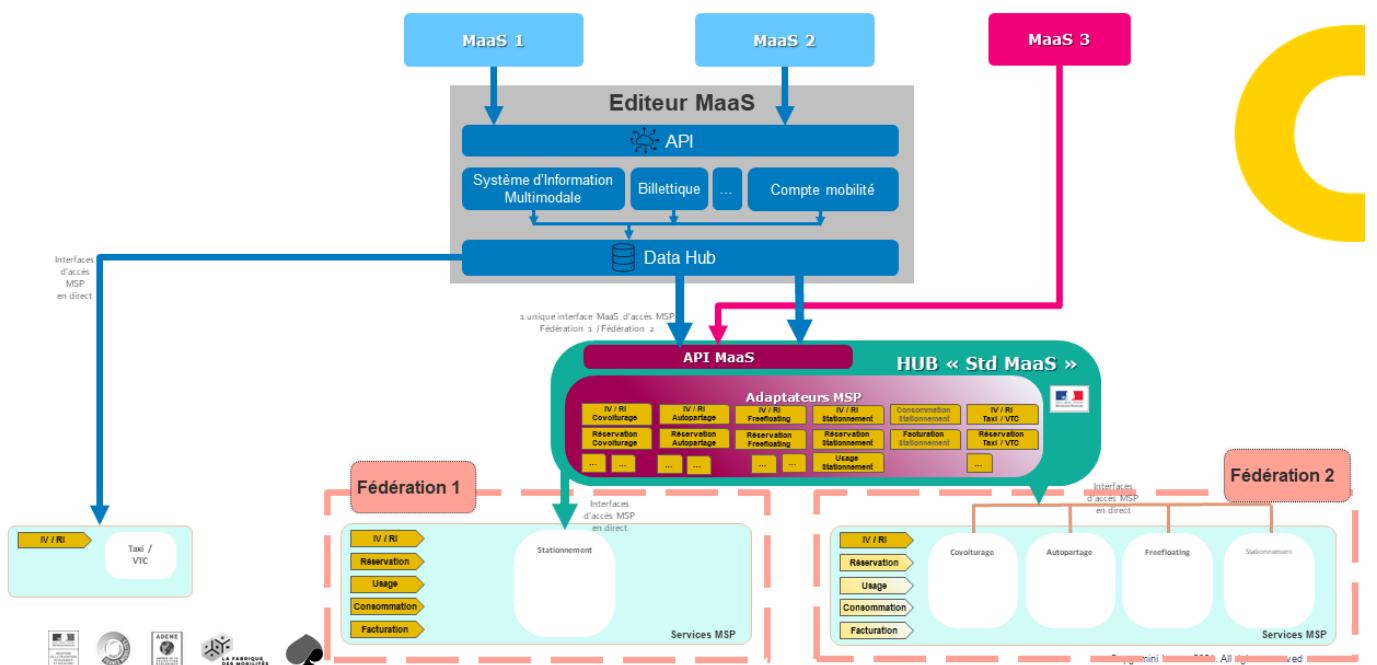


Figure 23b - Scénario d'intégration fédéral

4.3.5. Rôle et fonction du Hub

Selon l'avancée de la démarche de standardisation des MaaS et des MSP, le Hub est une plateforme pouvant assumer successivement plusieurs rôles à différents horizons : court, moyen et long terme.

Code	Etat de standardisation MSP	Rôles et fonctions du Hub	Apports du Hub pour les MaaS
R1	Pas de standard publié	Adaptation de formats propriétaires MSP vers un format unique pour les MaaS Accélérateur de la construction de nouveaux MaaS	Point d'entrée unique suivant un format partagé, public, rétro compatible autant que possible. Gère la relation avec le SI MSP Gère les montées de version et les changements de formats propriétaire.

R2	Standard publié récent, en cours d'appropriation par les MSP	R1 + Adaptation des formats standards MSP vers le format unique pour les MaaS. Accompagnement à la transition selon la vélocité du MSP à migrer vers le standard. Support à l'adoption des formats standards MSP (bac à sable)	R1 + Gère les montées de version et les changements des formats standard.
R3	Standard publié et utilisé massivement par les MSP	R2 - Décommissionnement de l'adaptation de formats propriétaires .	Pas d'apport pour les MaaS ayant intégré les standards . Apports R2 pour les MaaS n'ayant pas intégré certains standards et les nouveaux MaaS .

4.4. Domaines et composants logiques

Les communications depuis et vers les systèmes externes transiteront via des interfaces d'entrées/sorties définies, à l'aide de protocoles de transport standards et suivant des schémas publics, maintenus et versionnés (APIs).

L'interfaçage avec les MSP constitue un cas particulier car il n'existe pas de standard permettant de normaliser les flux. Nous proposons de résoudre ce problème à l'aide d'adaptateurs : la logique métier interne de la Gateway MCM Std MaaS s'appuiera sur un protocole et un format pivot d'API MaaS que les adaptateurs seront en charge de convertir pour les rendre compatibles avec les contraintes imposées par l'existant des MSP.

La Gateway MCM Std MaaS fournit au backend MaaS un ensemble de composants et de services « standardisés » d'accès aux différents MSP.

Ces services sont regroupés par domaine : IV / RI, la réservation et l'usage, la consommation et la facturation, la configuration du référentiel MSP et l'administration.

Le domaine d'administration concerne l'API d'administration ainsi que le monitoring technique et applicatif.

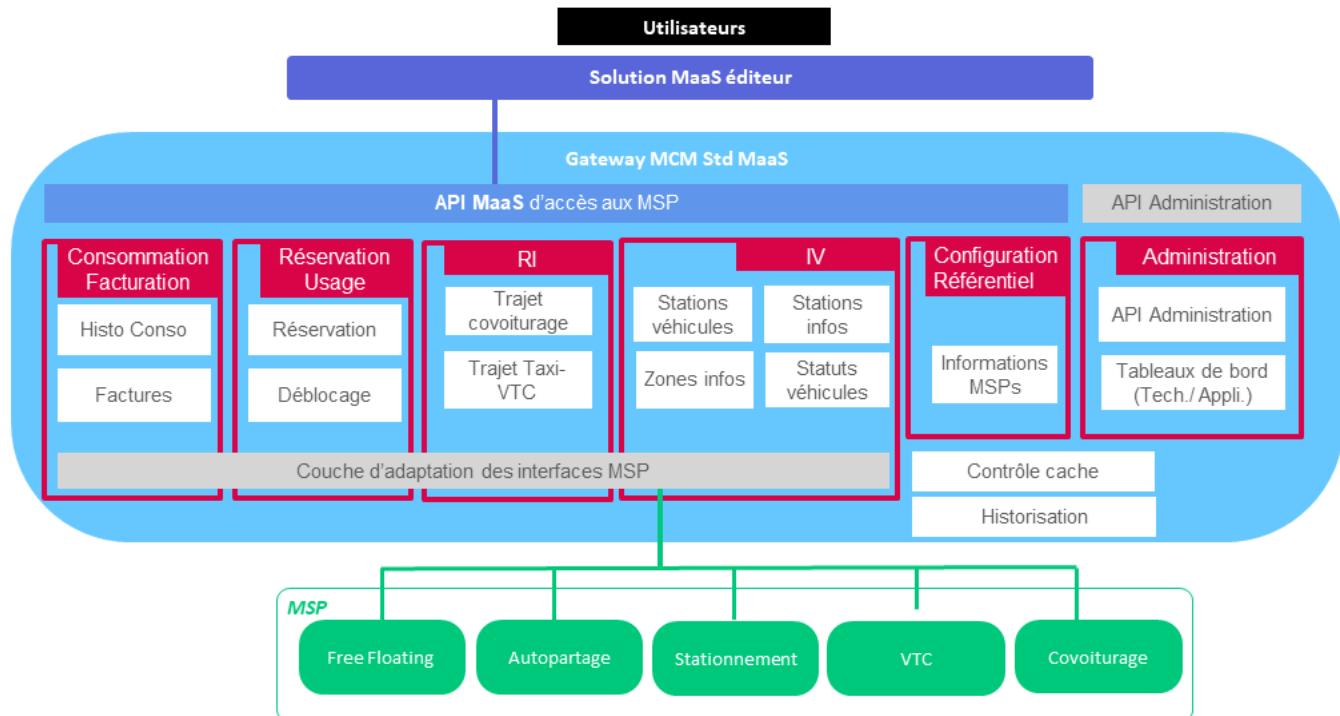


Figure 22 – Domaines et composants logiques

4.5. Stockage et utilisation des données

Chaque domaine logique est responsable des données qui lui sont propres. Les entités ne sont définies que dans un seul domaine référentiel. Les autres domaines peuvent faire référence à une entité qu'ils ne possèdent pas par son identifiant technique, les attributs essentiels pouvant être répliqués à condition que la cohérence soit garantie.

Voici un tableau listant les différents types de stockages envisagés :

Type de stockage	Nature des données à persister	Commentaire
Cache	Données à rafraîchir périodiquement	<input type="checkbox"/> Les données d'initialisation de certains MSP (liste des adresses de parkings, des différentes zones ...)
Relationnel	Configuration référentiel MSP	<input type="checkbox"/> URL d'accès, labels, noms des MSP, actions réalisables, paramètres d'appels

4.6. Flux et cinématique

4.6.1. Vue d'ensemble

Le schéma qui suit illustre les flux entrants et sortants de MCM. Les pastilles rouges indiquent la séquence des messages échangés. Afin d'être complets sémantiquement et de permettre une compréhension de bout en bout du processus, nous avons également capturé en gris les échanges intervenant en dehors de la Gateway MCM Std MaaS.

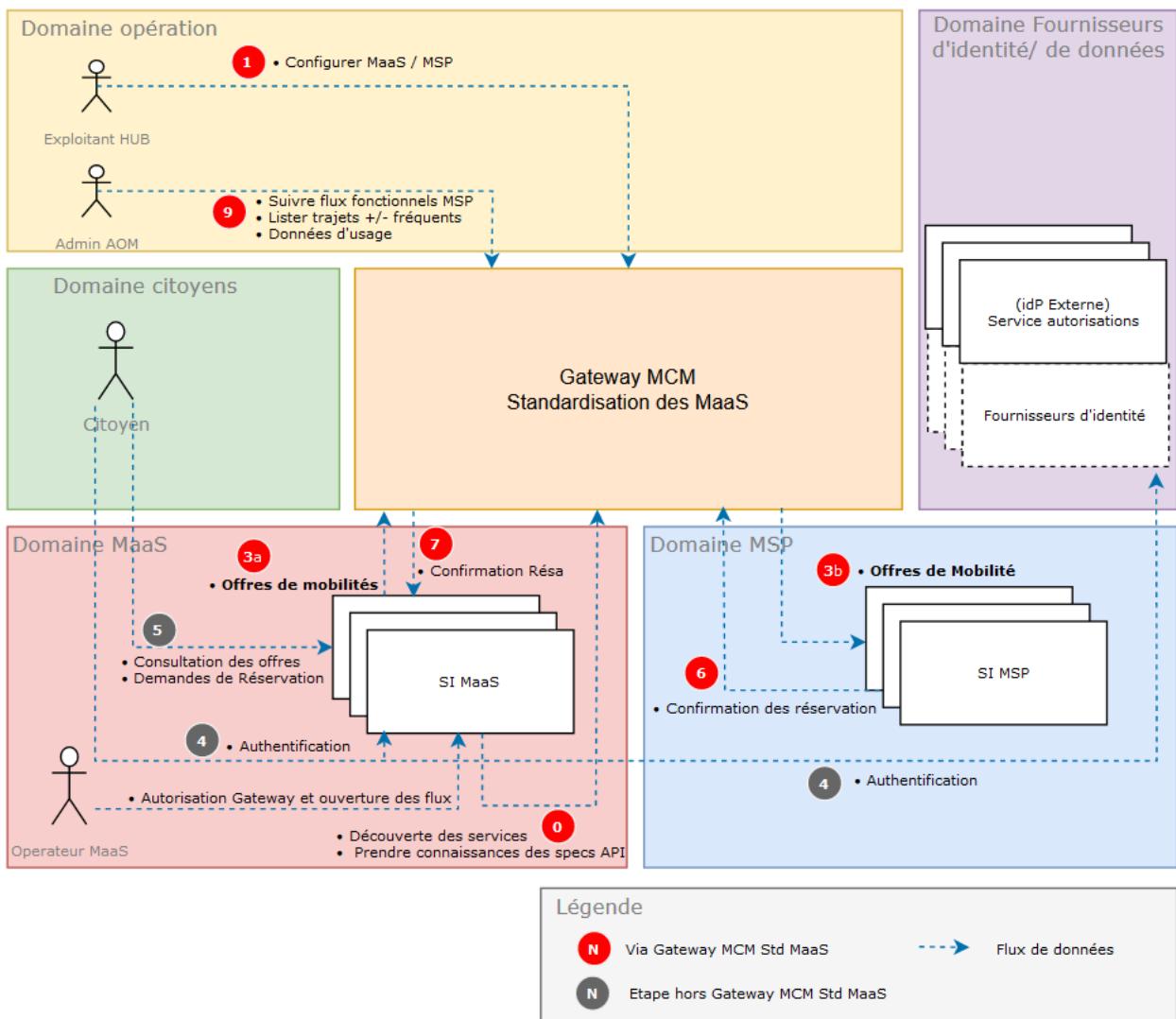


Figure 23 - Cinématique des flux

Le diagramme peut être interprété de la façon suivante :

0. Le SI d'un MAAS prend connaissance des services disponibles proposés par la Gateway MCM Std MaaS ainsi que la spécification de l'API'API, Si les services l'intéressent, le MaaS demande l'autorisation d'accès auprès de la Gateway pour utiliser ses services
1. Un administrateur technique de la Gateway MCM crée les comptes des MaaS et des MSP en renseignant les caractéristiques du point de terminaison technique de chacun.
2. Afin que le SI d'un MaaS/MSP puisse être contacté par la Gateway MCM, un lien de confiance doit être créé. Un administrateur doit autoriser MCM à contacter le SI. Il s'agit d'enregistrer l'URL et éventuellement l'adresse IP de la Gateway MCM auprès du service et ouvrir les flux.
3. (3a, 3b) Une fois les MaaS et les MSP dûment enregistrés, les offres de mobilité peuvent être transmises à la Gateway MCM.
4. Le citoyen se connecte MaaS et s'authentifie auprès d'un fournisseur d'identité (celui du MaaS ou d'un de ses comptes liés).
5. Le citoyen parcourt le site du MaaS à la découverte des offres de mobilité et des dispositifs incitatifs. Il procède à une demande de réservation. Celle-ci est transmise à la Gateway avec le jeton d'authentification du citoyen.

6. La Gateway soumet la demande de réservation au MSP. Celui-ci lui retourne la confirmation de la réservation. L'usager peut consommer le service de mobilité.
7. Une fois la demande de réservation validée, la confirmation est retournée au MaaS via la Gateway.
8. Les AOM peuvent consulter les statistiques d'usage et/ou exporter des données en vue d'une intégration et d'un traitement automatisé dans son propre SI.

4.6.2. API Management

Tous les flux à destination de la GW passent impérativement par la solution API Management.

L'API Management est un outil de management qui consiste à piloter au mieux les API en permettant de gérer la publication, la promotion, et la supervision des échanges de flux entre la Gateway et les partenaires qui effectuent les appels.

Afin de répondre à ces besoins de management, nous avons choisi la solution open source **Gravitee.io**, référencé dans le [socle interministériel des logiciels libres](#).

Gravitee.io est une suite complète open source d'outils liées à la gestion des API répondant à différents besoins (gestion SSL, load-balancing, failover, resources filtering, monitoring, ...). Gravitee.io APIM permet également de contrôler qui, quand et comment les API sont utilisées.

Gravitee.io APIM est une plateforme qui se compose de :

- De **Gateways** qui ont vocation de diriger le trafic des API. La Gateway est le point d'entrée pour appeler les API de la Gateway Std MaaS.
- Une console d'administration et de supervision « **API Console** » qui a le rôle d'un outil de publication permettant aux admins de la Gateway de définir les API, leurs politiques d'accès et d'utilisation, la gestion de cycle de vie des API.
- Un portail « **API Portal** » donnant accès aux partenaires leur permettant de voir les API exposés ainsi qu'à leurs documentations à partir de Swagger. Ce portail permet aux partenaires MaaS ou MSP de s'abonner aux API exposés par les administrateurs depuis la console d'administration.
- Une RESTful API interne « **APIM API** » qui expose des services pour gérer et configurer les interfaces utilisateur Web de la APIM console et du APIM portal. Tous les services exposés sont restreints par des règles d'authentification et d'autorisation.

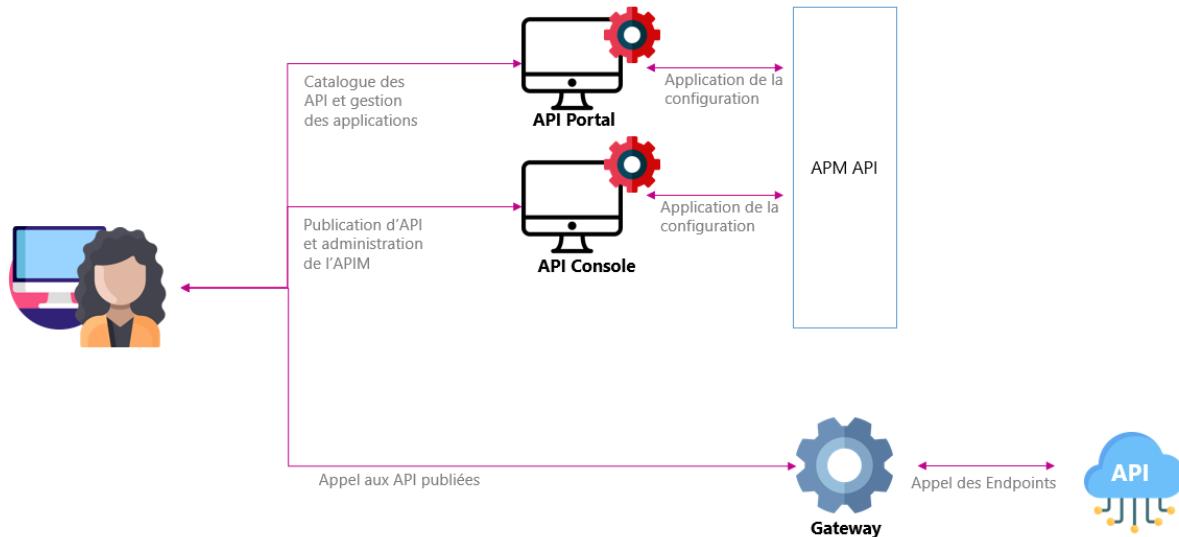


Figure 24 - Architecture globale de l'APIM Gravitee

4.6.3. Flux entrants

Cette section décrit les flux à destination de la GW. Cela comprend les points d'entrée des APIs proposées aux MaaS.

Flux entrants API Gateway

Bloc	Source	Destination	Protocole	Standard	URL	Cible	URL Cible	Description
Informations MSP	MaaS / MSP	APIM Gravitee	HTTPS		https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners	Récupérer les métadonnées de tous les MSPs / MaaS
	MaaS	APIM Gravitee	HTTPS		https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}	Récupérer les métadonnées d'un MSP
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/global-view	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/global-view	Récupérer une vue globale autour de moi
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/areas/{areaType}	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}/areas/{areaType}	Pour un MSP donné, obtenir des informations sur une zone spécifique
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/aroundMe	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/aroundMe	Retrouver les informations des véhicules des MSPs autour de moi
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/vehicles-types	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}/vehicles-types	Retrouver tous les types de véhicules d'un MSP
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/available-assets	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}/available-assets	Retrouver les véhicules disponibles d'un MSP
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/stations	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}/stations	Retrouver toutes les stations d'un MSP
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/stations-status	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}/stations-status	Récupérer l'état des stations d'un MSP
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/assets	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}/assets	Récupérer tous les véhicules d'un MSP
	MaaS	APIM Gravitee	HTTPS	TOMP v1.3.0	https://apim-gateway.[env]-gw.moncomptemobilite.fr/api/partners/{partnerId}/system-pricing-plan	MCM API Gateway	https://api.[env]-gw.moncomptemobilite.fr/v1/partners/{partnerId}/system-pricing-plan	Récupérer tous les parkings d'un MSP au format CSV

On peut remarquer 3 grands formats de flux supportés par la GW à ce stade :

- TOMP-API sur l'offre IV
- Standard Covoiturage V1 sur le covoitnage
- Internes, sur la présentation des partenaires et des services proposés par la GW

Bloc	Source	Destination	Protocole	Standard	URL	Cible	URL Cible	Description
Information voyageur	MaaS	API M Gravée	HTTPS	COUVERTURE	https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/passenger_regular_trips	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/passenger_regular_trips	Récupérer une collection de trajets réguliers de passagers
	MaaS	API M Gravée	HTTPS		https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/passenger_journeys	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/passenger_journeys	Récupérer une collection de trajets passagers planifiés ponctuels
	MaaS	API M Gravée	HTTPS		https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/driver_regular_trips	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/driver_regular_trips	Récupérer une collection de trajets réguliers de conducteurs
	MaaS	API M Gravée	HTTPS		https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/driver_journeys	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/driver_journeys	Récupérer une collection de trajets conducteurs planifiés ponctuels
covoiturage	MaaS	API M Gravée	HTTPS	COUVERTURE	https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/bookings	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/bookings	Synchroniser une demande de réservation initiée par une plateforme vers une deuxième plateforme impliquée dans le trajet ponctuel aller partagé
	MaaS	API M Gravée	HTTPS		POST https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/bookings/{bookingId}	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/bookings/{bookingId}	Effectuer une réservation
	MaaS	API M Gravée	HTTPS		GET https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/bookings/{bookingId}	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/bookings/{bookingId}	Retrouver les détails d'une réservation
Message	MaaS	API M Gravée	HTTPS	COUVERTURE	PATCH https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/bookings/{bookingId}	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/bookings/{bookingId}	Mettre à jour le statut d'une demande de réservation existante.
	MaaS / MSP	API M Gravée	HTTPS		https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/messages	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/messages	Permettre à un utilisateur d'écrire un message texte au propriétaire d'un trajet récupéré
Status	MaaS / MSP	API M Gravée	HTTPS		https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/status	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/status	Indiquer l'état de santé du service Web.
Webhook	MSP	API M Gravée	HTTPS	COUVERTURE	https://apim-gateway.[env]-gw.moncomtemobilite.fr/api/partners/{partnerId}/carpooling/booking_events	MCM API Gateway	https://api.[env]-gw.moncomtemobilite.fr/v1/partners/{partnerId}/carpooling/booking_events	Permettre à un opérateur de covoiturage d'envoyer des informations de réservation à un fournisseur tiers.

Figure 25 - Flux entrants Gateway MCM Std MaaS

4.6.4. Flux internes

Cette section montre les flux internes entre microservices. Cela comprend tous les flux définis dans la brique jaune.

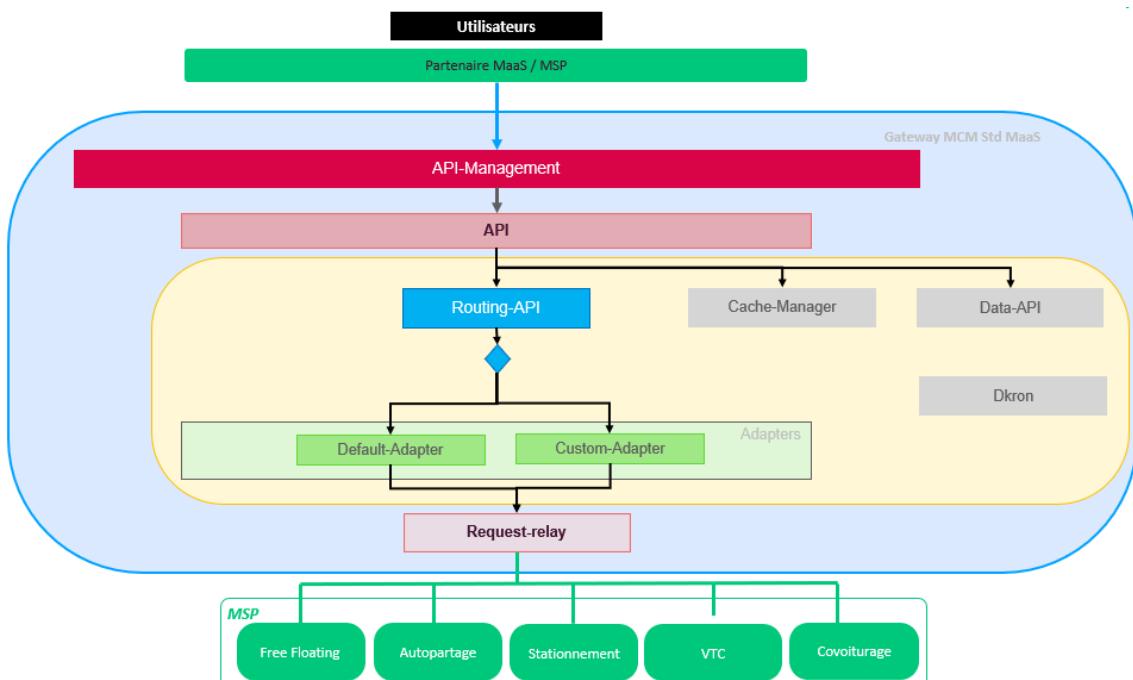


Figure 26 – Microservices internes

Ordonnanceur

Le module 'déclencheur dkron des refresh cache' est un service déployé sur la solution dkron et sous les termes de licence LGPLv3 <https://dkron.io/license>. Cette solution permet de programmer des tâches cycliques et est utilisée afin de déclencher les appels REST vers le cache manager qui lui-même, se charge de lancer les requêtes à l'origine de la collecte d'informations auprès des MSP et leur mise en cache dans redis.

Ce microservice est déployé à travers son image docker invoquée dans un template Helm chart. Une interface UI est disponible mais non exposée en externe de la Gateway, afin de visualiser les jobs paramétrés, les modifier, afficher l'historique des exécutions et les administrer. Une api est aussi disponible afin de permettre la manipulation des configurations <https://dkron.io/api/> et notamment celle des jobs sur /v1/jobs.

Les jobs instanciés dans dkron devront communiquer avec les url internes des différents microservices. Dans le cas d'un déploiement sur kubernetes, les url utilisées des services seraient sous la forme : <service.namespace.svc.cluster.local:port>

Relations avec les autres modules :

Le microservice **cache-manager** est interrogé via des requêtes http par le composant dkron. Il devra contenir les informations nécessaires et suffisantes afin que les requêtes d'alimentation couvrent le scope de ce qui est candidat au cache. La fréquence de paramétrage des sondes devra être en lien avec le soft TTL de la table cache_param et il est conseillé qu'il soit au moins 2 fois inférieur à ce dernier si le hard TTL n'est pas au moins 2 fois plus grand (afin de palier à une perte de donnée en cas d'indisponibilité du MSP lors de l'alimentation).

4.6.5. Flux sortants

Cette section décrit les flux en sortie de la GW. Toutes les requêtes en entrée de la GW transitent dans les services internes définis dans la section précédente, pour passer ensuite par le microservice **request-relay** qui est le seul point d'accès en relation directe avec les services des partenaires.

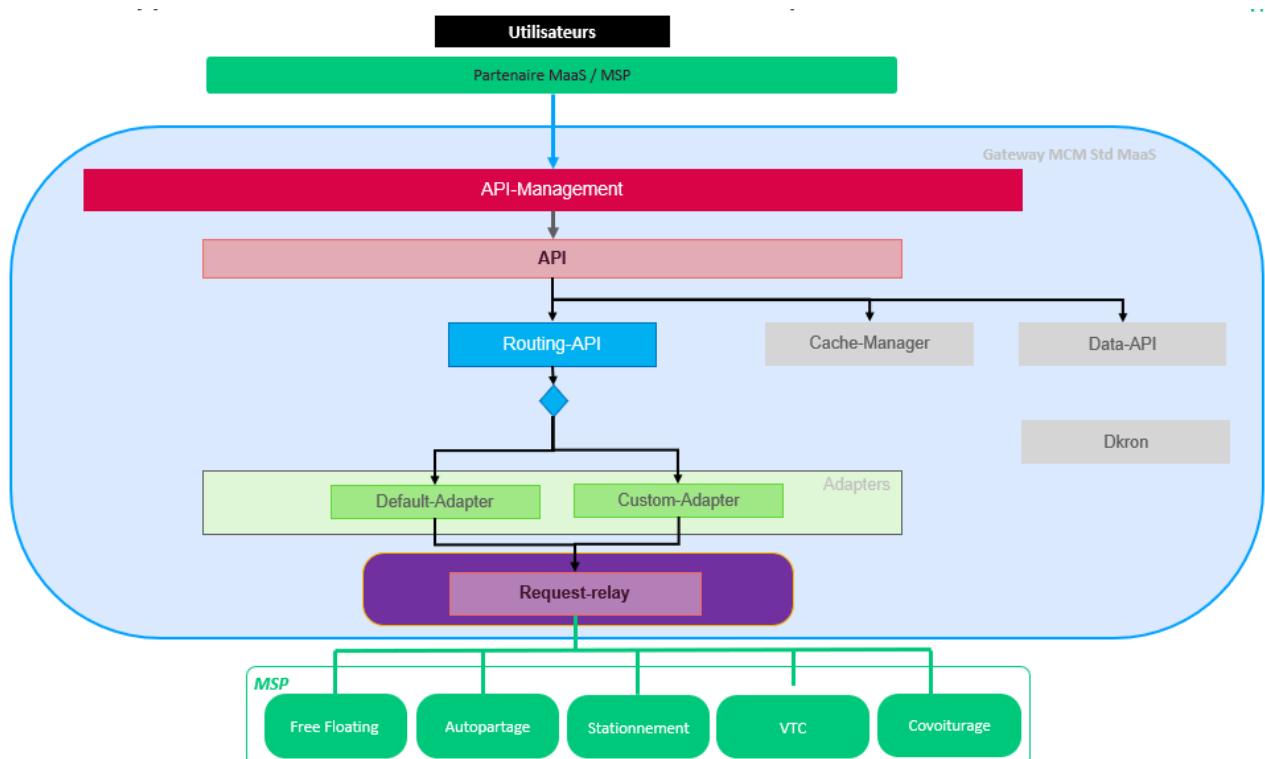


Figure 27 – Microservice des flux sortants Gateway MCM Std MaaS

5. Principes directeurs

L'architecture applicative de la Gateway MCM Std MaaS est conçue avec le souci d'extensibilité pour l'ajout progressif de MSP et de nouvelles fonctions sur des MSP existants à travers une plateforme d'API sur lesquels nous viendrons connecter les différentes fonctions MSP et éditeur.

Les technologies retenues pour l'expérimentation permettront d'accompagner le passage à l'échelle de la solution sur un périmètre plus large ou sur plusieurs déclinaisons, en bénéficiant de la souplesse et de l'évolutivité horizontale apportées par l'état de l'art d'orchestration de conteneurs.

Nous proposons donc une architecture technique s'appuyant sur des accélérateurs technologiques tels que la containerisation, l'utilisation de services Cloud sur étagère (PaaS) afin de nous affranchir des travaux d'installation et de configuration.

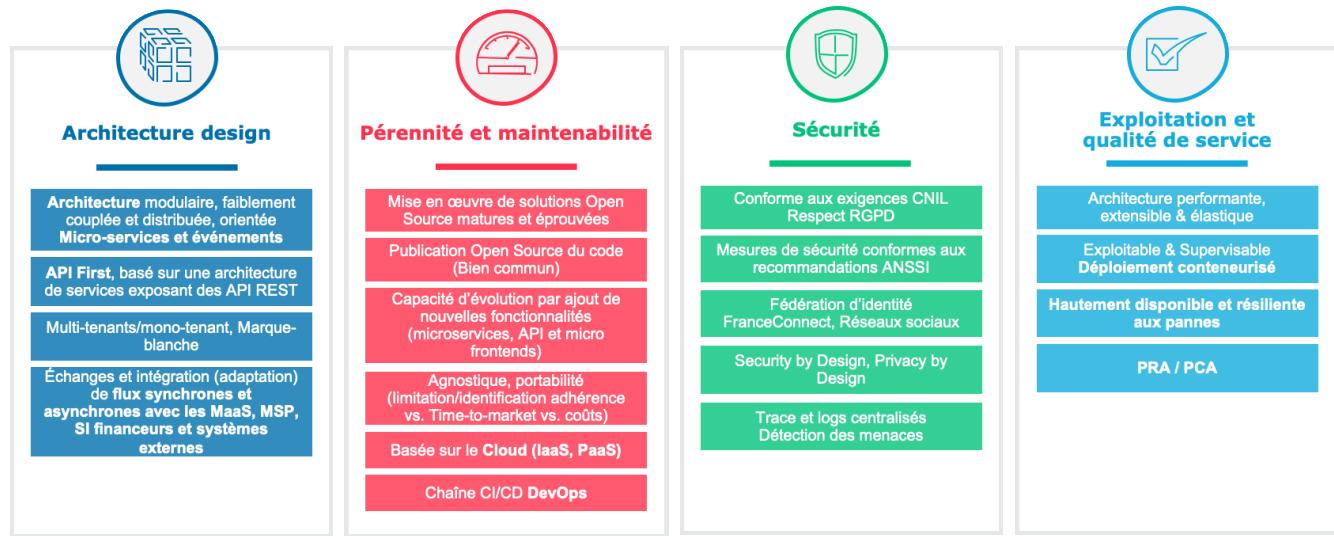


Figure 28 – Principes directeurs d'architecture

Architecture design

L'architecture MCM Std MaaS GW est urbanisée, modulaire, faiblement couplée et distribuée.

Les modules implémentent des services métiers (micro-services) exposés sous la forme d'APIs REST proposant des contrats d'interface.

L'architecture MCM Std MaaS GW est conçue pour supporter les modes multi-tenants, mono-tenant et marque blanche. Elle est suffisamment souple pour être déployée selon différentes complexités d'infrastructure. Elle est instanciable pour un acteur ou une fédération d'acteurs.

Le sous-système gérant les échanges permet d'intégrer des sources et flux de données de manière évolutive et modulaire. Il supporte les flux synchrones et asynchrones et réalise les échanges avec les MaaS et MSP.

Pérennité & Maintenabilité

L'architecture MCM Std MaaS GW utilise des technologies et solutions Open Source matures et éprouvées.

L'architecture proposée, du fait de sa construction et sa modularité, offre des capacités d'évolution par ajout de nouvelles fonctionnalités dans les modules et par ajout / remplacement de modules.

Afin de maîtriser l'évolution du système dans le temps et d'assurer une gouvernance des APIs et des données (cycle de vie, catalogue, versioning, analyse impacts, ...), une gouvernance outillée est mise en œuvre.

L'architecture MCM Std MaaS GW est hébergée dans le Cloud Azure et met en œuvre des ressources IaaS et PaaS.

Les choix de composants et de solutions s'attachent à être le plus agnostique possible et à faciliter la portabilité entre environnements d'hébergement.

L'architecture MCM Std MaaS GW met en œuvre une chaîne CI/CD, en approche DevOps, permettant de réaliser l'intégration continue des composants, de mesurer la qualité, et d'automatiser la non-régression, les tests et le déploiement.

Sécurité

L'architecture MCM Std MaaS GW met en œuvre les recommandations et exigences de la CNIL dans le respect de la GDPR (Privacy by Design).

Le système MCM intègre les exigences et contraintes de sécurité (Security by Design) et met en œuvre les principaux mécanismes et mesures de sécurité suivants :

- Chiffrement (HTTPS, SFTP, SSL/TLS) des flux échangés
- Cloisonnement et sécurisation des réseaux
- Mise en œuvre de groupes de sécurité
- Accès APIs protégé par la Gateway MCM Std MaaS / Reverse proxy
- Contrôle d'intégrité (checksum) lors des échanges de fichiers
- Traces et logs centralisées
- Mise en œuvre et vérification des bonnes pratiques de la sécurité dans le code
- Vérification automatique (chaîne CI/CD) des vulnérabilités Top 10 OWASP
- Vérification automatique des vulnérabilités divulguées publiquement et contenues dans les dépendances des librairies des projets
- Gestion de l'identité et des accès des administrateurs techniques et les administrateurs fonctionnels par la mise en œuvre d'une solution d'IAM (Identity Access Manager) permettant la fédération des identités et le Web SSO
- Permissions d'accès basées sur RBAC (Role-Based Access Control)
- Sauvegarde / snapshot

Exploitation & Qualité de service

L'architecture du système MCM met en œuvre des solutions d'administration technique et fonctionnelle et des outils de supervision & monitoring permettant d'exploiter le système.

Les mécanismes de haute-disponibilité (composants redondés, multi-instances, répartition de charge, tolérance aux pannes), l'extensibilité et les performances des solutions proposées permettent de se conformer au niveau de qualité de service demandé et de garantir la résilience du système en conformité avec le PRA/PCA.

L'hébergement du système MCM dans le Cloud Azure (IaaS et PaaS) couplé à une automatisation de la création des instances et des environnements (Infrastructure as Code), et à l'utilisation d'un orchestrateur de conteneurs permet d'offrir une extensibilité et une élasticité du système.

6. Architecture technique

6.1. Schéma de principe

Dans cette section, nous nous intéressons aux composants de la plateforme MCM Std MaaS GW. Dans un premier temps, nous ferons abstraction des choix technologiques, des implémentations et optimisations éventuelles en nous concentrant uniquement sur la fonction des sous-systèmes.

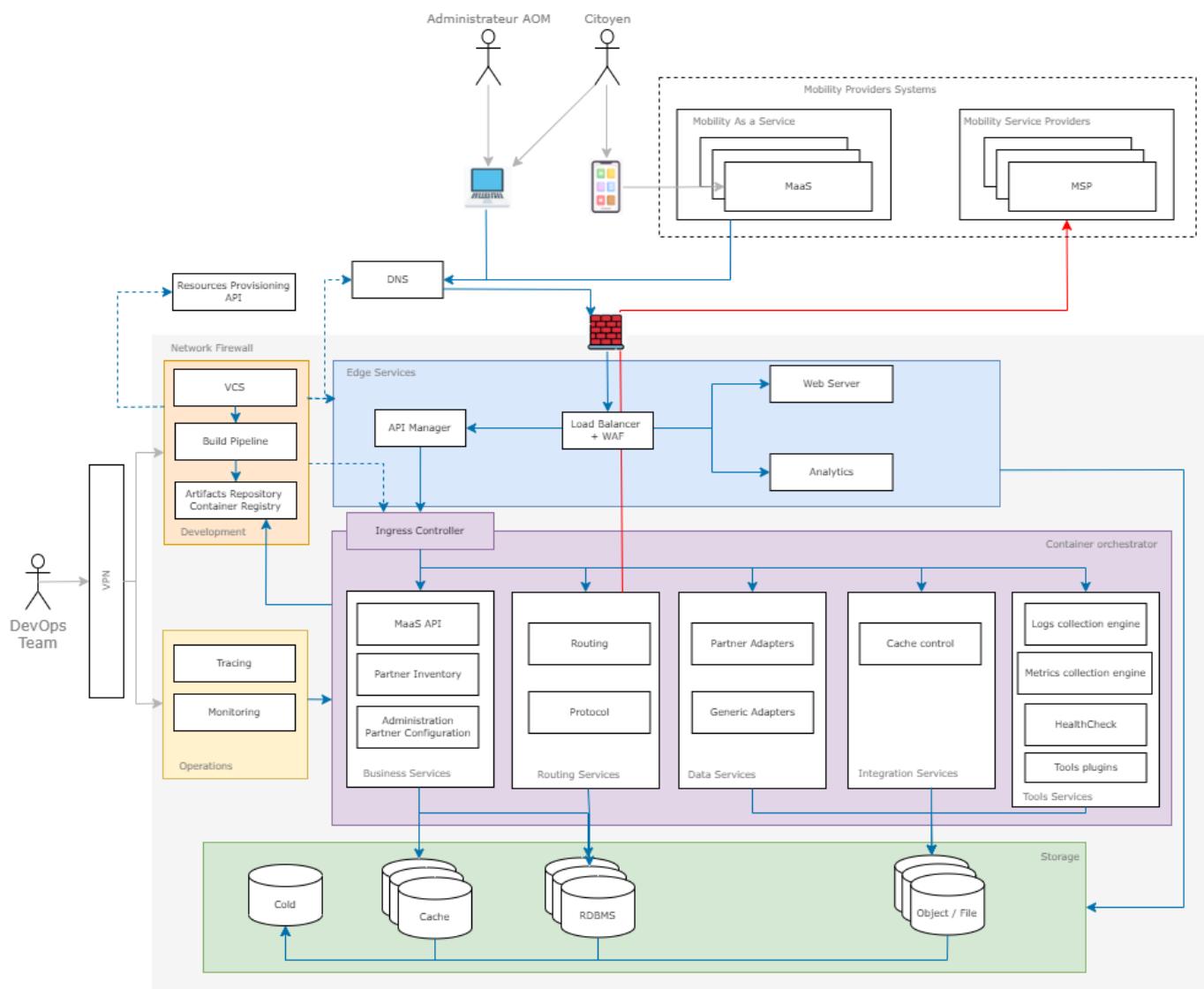


Figure 29 - Architecture technique – schéma de principe

Pour décrire le comportement du système, nous pouvons considérer 2 phases de son cycle de vie : la construction par l'équipe de développement et l'utilisation par les bénéficiaires du service.

L'ensemble des ressources de calcul est hébergé dans une infrastructure dont l'accès est contrôlé par un pare-feu. L'unique prérequis est que l'infrastructure supporte le provisionnement de ressources de façon programmatique via une API. Il peut aussi bien s'agir d'un cloud public que d'un cloud privé.

Les capacités de stockage sont également fournies sous la forme de services supportant divers types de bases de données : relationnelles ou non relationnelles, cache ou topic.

La forge logicielle est hébergée au sein de cette infrastructure et s'exécute dans des machines virtuelles ou des conteneurs. Elle est composée d'un système de gestion de versions, d'un outil d'intégration et de déploiement continu, d'un dépôt de binaires et d'images.

La chaîne d'intégration/déploiement continu est en mesure d'invoquer un outil d'« infrastructure as code » qui automatise la construction des ressources telles que les réseaux, machines virtuelles, groupes de sécurité, les bases de données. L'outil d'IaC utilise pour cela l'API exposée par le gestionnaire de l'infrastructure.

La chaîne de déploiement continu est par conséquent en mesure de configurer aussi bien l'infrastructure que les environnements applicatifs.

Le firewall est configuré pour permettre les connexions à la forge uniquement aux employés de Capgemini affectés au projet MCM Standardisation des MaaS. Les développeurs devront s'authentifier à l'aide des mécanismes de sécurités fournis par Capgemini (authentification multi-facteurs).

En revanche, le pare-feu autorise les connexions entrantes sur les ports réservés à http (80) et/ou https (443) quelle que soit leur origine. C'est par ce canal que les requêtes provenant des MaaS parviendront à la Gateway.

Le MaaS émettra des appels d'APIs au Backend de la Gateway MCM Std MaaS, en fournissant une clé d'API. Ces requêtes seront interceptées, sécurisées, auditées et routées par l'API Manager.

La logique métier côté backend est implémentée sous la forme de multiples conteneurs, selon le style d'architecture microservices. Ces conteneurs sont lancés, dimensionnés, surveillés par un orchestrateur qui s'assure de leur état de santé, les redémarre si nécessaire.

L'orchestrateur constitue également une plateforme facilitant le développement. En proposant une approche déclarative plutôt qu'impérative, il se substitue au développeur et détermine automatiquement les transitions nécessaires pour atteindre l'état cible configuré. Par ailleurs, il comporte un annuaire auprès duquel les services exposés par les conteneurs peuvent s'enregistrer afin qu'ils puissent être découverts par leurs clients. Enfin, le contrôleur d'entrée (ingress controller) exploite les informations de l'annuaire de services pour assurer la répartition de charge et permettre la mise en œuvre de stratégies de déploiement avancées comme le blue/green deployment ou les rolling updates.

Le trafic sortant vers les systèmes externes est contrôlé par le firewall.

Un point d'accès spécifique permet aux administrateurs de réaliser les opérations de surveillance et de maintenance du système.

6.2. Justification des choix d'architecture

6.2.1. Stratégie globale

Plusieurs facteurs ont été pris en compte afin de sélectionner les composants permettant d'implémenter cette architecture cible :

- La rapidité de mise en œuvre : le souhait de l'équipe projet est de valider la valeur de la proposition auprès des utilisateurs participant à l'expérimentation. Ce but peut être plus facilement atteint si l'effort de développement est concentré sur le fonctionnel plutôt que l'infrastructure technique. Par ailleurs, les coûts de réalisations sont inversement proportionnels à la vitesse.
- La maîtrise des coûts de licence : ces derniers constituent une incitation forte à favoriser les composants open source.
- Les coûts : le modèle de responsabilités partagées proposé par les fournisseurs de cloud permet de libérer l'équipe projet de certaines tâches récurrentes telles que les mises à jour. Inversement, en l'absence de

contrat de maintenance, le choix de briques open source est susceptible d'augmenter l'effort de maintien en condition opérationnelle.

- Le vendor lock-in et l'impact sur la réversibilité : le choix d'un service managé doit être entouré de précautions car il implique un couplage à la plateforme cloud. Si l'interface dudit service respecte un standard supporté par plusieurs implémentations tierces, alors une substitution un-pour-un est possible. En revanche, dans le cas d'une interface propriétaire un rework plus ou moins doit être provisionné afin de permettre la réversibilité. A ceci s'ajoute la difficulté potentielle à migrer les données. Ce risque existe même en présence d'interfaces normalisées : c'est le cas par exemple des services/composants gérant des secrets lorsque ceux-ci ne peuvent être exportés à des fins de migration.

En définitive, 3 stratégies sont possibles pour choisir les technologies :

1. Le tout managé qui offre l'avantage de la rapidité au détriment des coûts de licence et de réversibilité ;
2. Le tout open source qui contourne le problème des licences, facilite la réversibilité — pourvu que les modules choisis soient approuvés par le repreneur — mais augmente les coûts de construction et d'opérations ;
3. Une combinaison des deux stratégies précédentes : opter pour des services managés lorsqu'une ou plusieurs options open source interchangeables existent ; démarrer avec une brique open source lorsque la migration n'est pas triviale.

La stratégie 3 est celle retenue pour notre solution.

6.2.2. Disponibilité

Pour atteindre les exigences de disponibilité dans le contexte de cette expérimentation, Capgemini a fait les choix suivants :

- Hébergement Azure : facilité de « provisioning » et haute disponibilité des services d'infrastructure réseau, serveurs, stockage, supervision, et des services de plus haut niveau comme Kubernetes, cache, base de données
- L'applicatif Gateway est conçue pour être déployée sous forme d'une ferme de nœud « sans état » et un cache
 - Plusieurs noeuds actifs pour assurer la redondance et la tenue des performances
 - S'appuyant sur des services de la plateforme Azure à fort taux de disponibilité (cache 99,9%, base de données 99,99%, Kubernetes 99,5% - non garantie)
- Le cluster Kubernetes est dédié à la Gateway

6.3. Architecture technique du PMV

Nous préconisons de recourir aux services managés d'Azure pour la plupart des aspects à l'exception de :

- La forge logicielle : migrer d'Azure DevOps vers toute autre solution impliquerait une réécriture des scripts d'intégration et de déploiement. Par ailleurs, une partie des données d'historique serait perdue. Enfin, l'expérience développeur est supérieure dans le cas de la solution concurrente. Il est prévu de capitaliser sur la licence GitLab premium acquise dans le cadre de moB.
- La solution d'API Management Azure n'est pas celle retenue car elle ne permettrait pas d'exporter les configurations vers une autre solution, et n'est pas satisfaisante d'après différents retours d'expérience Capgemini.
- Le coffre-fort est accessible à travers des interfaces non standards, induisent un coût initial de mise en œuvre significatif — et par conséquent un risque de rework qu'il est souhaitable d'atténuer — du fait de sa complexité, et embarquent des données difficiles à reprendre.
- Enfin, les briques d'intégration permettant l'interfaçage avec les MaaS et MSP. Elles seront fortement sollicitées par le code des adaptateurs. Par conséquent, remplacer les composants d'intégration engendrerait

des modifications significatives aux adaptateurs. Cette situation n'est pas désirable du fait du nombre d'adaptateurs susceptibles d'être développés.

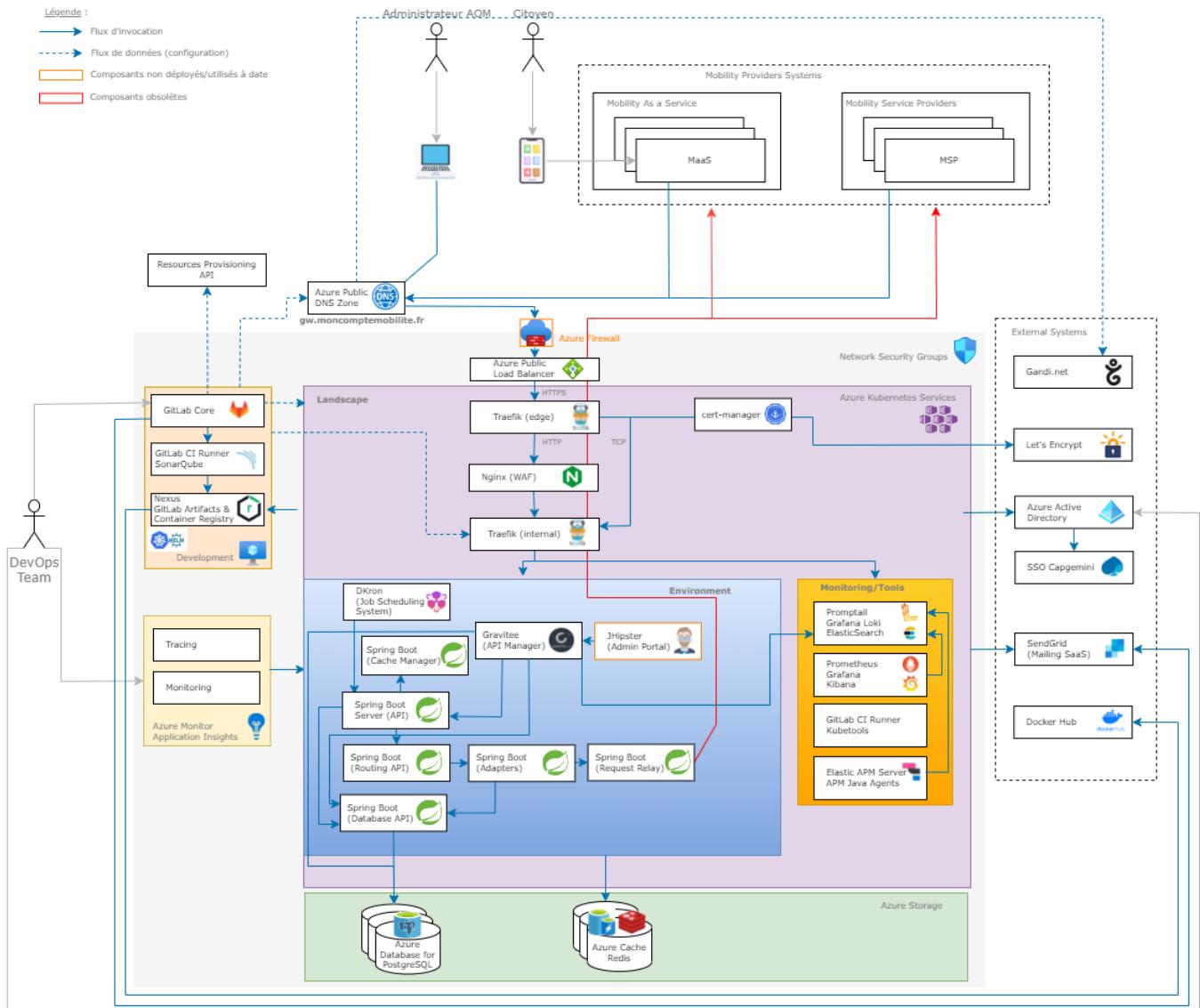


Figure 30 - Architecture technique retenue pour la phase d'expérimentation

6.4. Licences open sources autorisées

Selon les objectifs de protection recherchés, il existe plusieurs classes de licences largement répandues et conduisant à des obligations diverses telles que les licences permissives MIT/BSD, les licences « copyleft » comme GPL ou encore celles du type « network protection ».

Il existe également d'autres licences sur le marché qui ont une portée internationale, parmi celles qui sont connues, on retrouve Apache 2.0 qui est utilisée par le standard TOMP-API notamment.

Par ailleurs, l'Etat français définit les licences qui sont applicables aux codes sources des logiciels publiés par l'administration sur le site data.gouv.fr : <https://www.data.gouv.fr/fr/licences>

Licences permissives	identifiant SPDX
Apache License 2.0	Apache-2.0
BSD 2-Clause "Simplified" License	BSD-2-Clause
BSD 3-Clause "New" or "Revised" License	BSD-3-Clause
CeCILL-B Free Software License Agreement	CECILL-B
MIT License	MIT

Licences avec obligation de réciprocité	identifiant SPDX
CeCILL Free Software License Agreement v2.1	CECILL-2.1
CeCILL-C Free Software License Agreement	CECILL-C
GNU General Public License v3.0 or later	GPL-3.0-or-later
GNU Lesser General Public License v3.0 or later	LGPL-3.0-or-later
GNU Affero General Public License v3.0 or later	AGPL-3.0-or-later
Mozilla Public License 2.0	MPL-2.0

Figure 31 - Licences opensource

La licence [CeCILL-B](#) utilisée pour le projet MCM est très proche de la licence Apache 2.0 mais appliquée à la législation française, elle est plus récente et donc beaucoup moins répandue.

La principale différence entre ces deux licences est dans l'obligation forte de citation qui se trouve dans CeCILL-B (article 5.3.4).

- ☞ La variante **CeCILL-B** utilisée en mai pour la publication de MCM Historique permet d'ajouter **l'obligation de citer MCM** qui reste importante, notamment pour **conserver la légitimité du projet dans les éventuelles reprises du projet** (par ex. cela permet d'éviter la création d'un 2^{ème} MCM sans évoquer le 1^{er}).
- ☞ La variante **CeCILL-C** ajoute une **obligation de réciprocité** et va donc plus loin, car elle **oblige les contributeurs à partager leurs modifications à la communauté**. On pourrait souhaiter cela pour les travaux de standardisation des MaaS, mais cela peut également être un frein.

Ainsi, toute personne (physique/morale) réutilisant le logiciel MCM doit mentionner ce dernier et s'il le redistribue à un tiers, il doit faire en sorte que ce tiers mentionne également le logiciel MCM.

Cette traçabilité importante pour le projet est absente avec la licence Apache.

Tout comme le projet Mon Compte Mobilité, le projet MCM Standardisation des MaaS possède une visée nationale, il est donc préférable de publier le code source sous l'une des licences CeCILL-B ou CeCILL-2.1/CeCILL-C qui sont des transpositions en droit Français des licences BSD et GPL/LGPL respectivement.

6.5. Composants logiciels

Le tableau ci-dessous précise les composants et versions logicielles utilisés dans le cadre du projet MCM Std MaaS GW.

Notez bien que les composants listés ci-dessous se répartissent en 2 catégories :

- Ceux qui sont déployés avec l'application ;

- Les composants technologiques qui sont mis en œuvre uniquement pendant la phase de construction ou de test par l'équipe de développement.

Composant	Solution	Version	Editeur	Licence / Souscription	Lien
Système d'exploitation serveur	Ubuntu	18.04LT S	Canonical	Multiples, principalement GPL	https://www.centos.org
Conteneurs	Docker	19.03.8	Docker	Apache License 2.0	https://www.docker.com
Orchestrateur de conteneurs	Azure Kubernetes Services	1.21.7	Azure	Apache License 2.0	https://docs.microsoft.com/fr-fr/azure/aks/ https://kubernetes.io/
Load Balancer	Azure Load Balancer		Azure		https://learn.microsoft.com/en-us/azure/load-balancer/
API Manager / API Gateway	Gravitee	3.12	Gravitee	Apache License 2.0	https://www.gravitee.io/
Proxy Inverse / Ingress Controller	Traefik	2.6.x	Traefik Labs	MIT License	https://doc.traefik.io/traefik/
Gestion des certificats	Cert-Manager	1.4.0	Cert Manager	Apache License 2.0	https://cert-manager.io/
Autorité de certification	Let's Encrypt		Internet Security Research Group (ISRG)		https://letsencrypt.org/
Proxy Inverse / Web Application Firewall	Nginx	1.21.6	NGINX, Inc. Sysoev	BSD 2-clauses (d)	https://www.nginx.com/
Générateur de sites statiques	Jhipster	7.x	Jhipster	Apache License 2.0	https://www.jhipster.tech/
JVM / JDK	AdoptOpenJDK	11	AdoptOpenJDK	GNU General Public License, version 2	https://adoptopenjdk.net/
Serveur d'applications Java	Spring Boot	2.6.6	VMWare	Apache License 2.0	https://spring.io/projects/spring-boot
Administration serveur d'applications Java	Spring Boot Admin	2.6.1	Codecentric	Apache License 2.0	https://github.com/codecentric/spring-boot-admin
Système de gestion de bases de données	Azure Database for PostgreSQL	13.6	PostgreSQL	PostgreSQL License	https://www.postgresql.org https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/
Administration système de gestion de bases de données	pgAdmin	4.20	PostgreSQL	PostgreSQL License	https://www.pgadmin.org

Administration système de gestion de bases de données	Dbeaver	22.2.3	DBeaver Community Edition	Apache 2.0	License	https://dbeaver.io/
Supervision & Métriques	Prometheus	2.33.0	Prometheus	Apache 2.0	License	https://prometheus.io
Dashboard Supervision & Monitoring	Grafana	8.0.3	Grafana Labs	Apache 2.0	License	https://grafana.com
Centralisation des logs	Grafana Loki	8.0.3	Grafana Labs	Apache 2.0	License	https://grafana.com/oss/loki/
Agent de gestion des logs	Promtail	0.17.2	Grafana Labs	Apache 2.0	License	https://grafana.com/docs/loki/latest/clients/promtail/
Base de données analytics	ElasticSearch	7.16.3	Elastic	Apache 2.0	License	https://www.elastic.co/fr/ https://github.com/elastic/elasticsearch
HealthCheck & Monitoring	Uptime-kuma	1.11.3	Louislam	MIT License		https://github.com/louislam/uptime-kuma
Stockage mémoire	Azure Cache Redis	6.0	Redis Ltd.	BSD		https://redis.io/ https://docs.microsoft.com/en-us/azure/azure-cache-for-redis
Versionnning code	GitLab Premium	13.x	GitLab	Commercial License		https://gitlab.com/gitlab-org/gitlab
Gestion projet (code)	Maven	3.6.3	Apache Software Foundation	Apache 2.0	License	https://maven.apache.org
Gestionnaire dépôts objets binaires	Nexus Repository OSS	3.x	Sonatype	Eclipse Public License 1.0		https://github.com/sonatype-nexus-community
Qualimétrie	Sonarqube	8.5.x	SonarSource	LGPL		https://www.sonarqube.org/
Gestionnaire de paquets Kubernetes	Helm	3.9.0	-	Apache 2.0	License	https://helm.sh/
Scheduling job	Dkron	3.x		LGPLv3		https://dkron.io/
Infrastructure as Code	Terraform	1.1.3	HashiCorp	Mozilla Public License v2.0		https://www.hashicorp.com/products/terraform
Envoi de mails	SendGrid	Essential s 40K	Isaac Saldana, Jose Lopez, Tim Jenkins			https://sendgrid.com/

6.6. Provisionnement des ressources dans Azure

Par plateforme, nous entendons un groupe de ressources machine, réseau et stockage isolé pouvant accueillir des piles logicielles. Des conditions d'accès distinctes peuvent s'appliquer aux plateformes.

Un environnement s'exécute au sein d'une plateforme. Il est en compétition avec les autres environnements de la même plateforme pour l'accès aux ressources. Toutefois, les environnements n'interfèrent pas entre eux.

Au sein de chaque environnement, des tiers peuvent être distingués : tiers web, logique métier et données. Le trafic réseau interne à l'environnement devrait idéalement faire l'objet de restrictions afin de forcer l'application des bonnes pratiques et limiter les risques de vulnérabilité.

Chaque environnement peut donner lieu à des déploiements successifs, chaque déploiement contenant une combinaison différente de versions de ses composants. La conservation de l'historique des déploiements et des binaires correspondants, la transparence sur leurs contenus/versions et la traçabilité jusqu'aux commits sont des qualités opérationnelles déterminantes pendant les phases de mise au point et de production.

Afin de répondre aux objectifs de sécurité et favoriser la lisibilité des rapports d'utilisation/factures des services Azure, il est souhaitable de ségréguer les ressources et le trafic réseau :

- Vis-à-vis des autres projets ;
- Entre les plateformes de développement et celles en production ;
- Entre environnements d'une même plateforme ;
- Entre les couches d'un même environnement.

Ces séparations peuvent être mises en œuvre grâce aux concepts supportés par Azure : les souscriptions, réseaux virtuels, groupes de sécurité et firewalls.

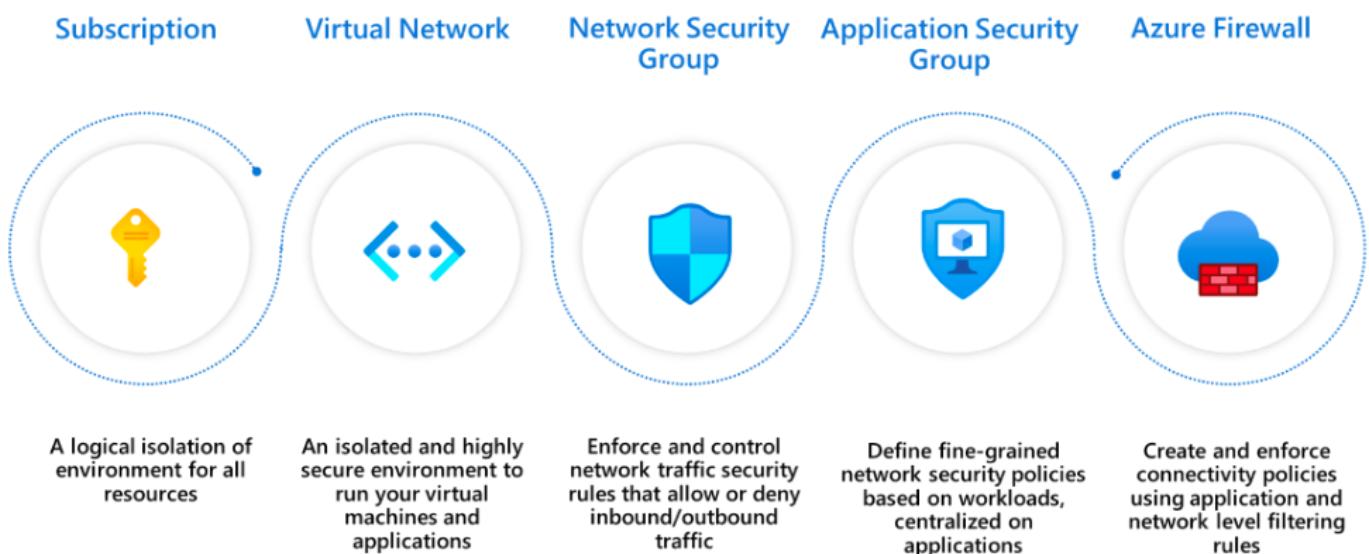


Figure 32 - Options de segmentation du trafic réseau dans Azure

Source : <https://docs.microsoft.com/fr-fr/azure/architecture/.../hybrid-networking/network-level-segmentation>

6.7. Schéma réseau

En termes de topologie réseau :

- Chaque plateforme disposerait de son propre VNet.
- Le VNet associé à la plateforme des services partagés (CI/CD) serait appairé à celui de toutes les autres plateformes.
- Au sein de la plateforme, et pour chaque environnement, des subnets seraient réservés pour les conteneurs de chaque tiers.

L'application des principes exposés à travers les sections précédentes conduirait à l'implémentation ci-dessous :

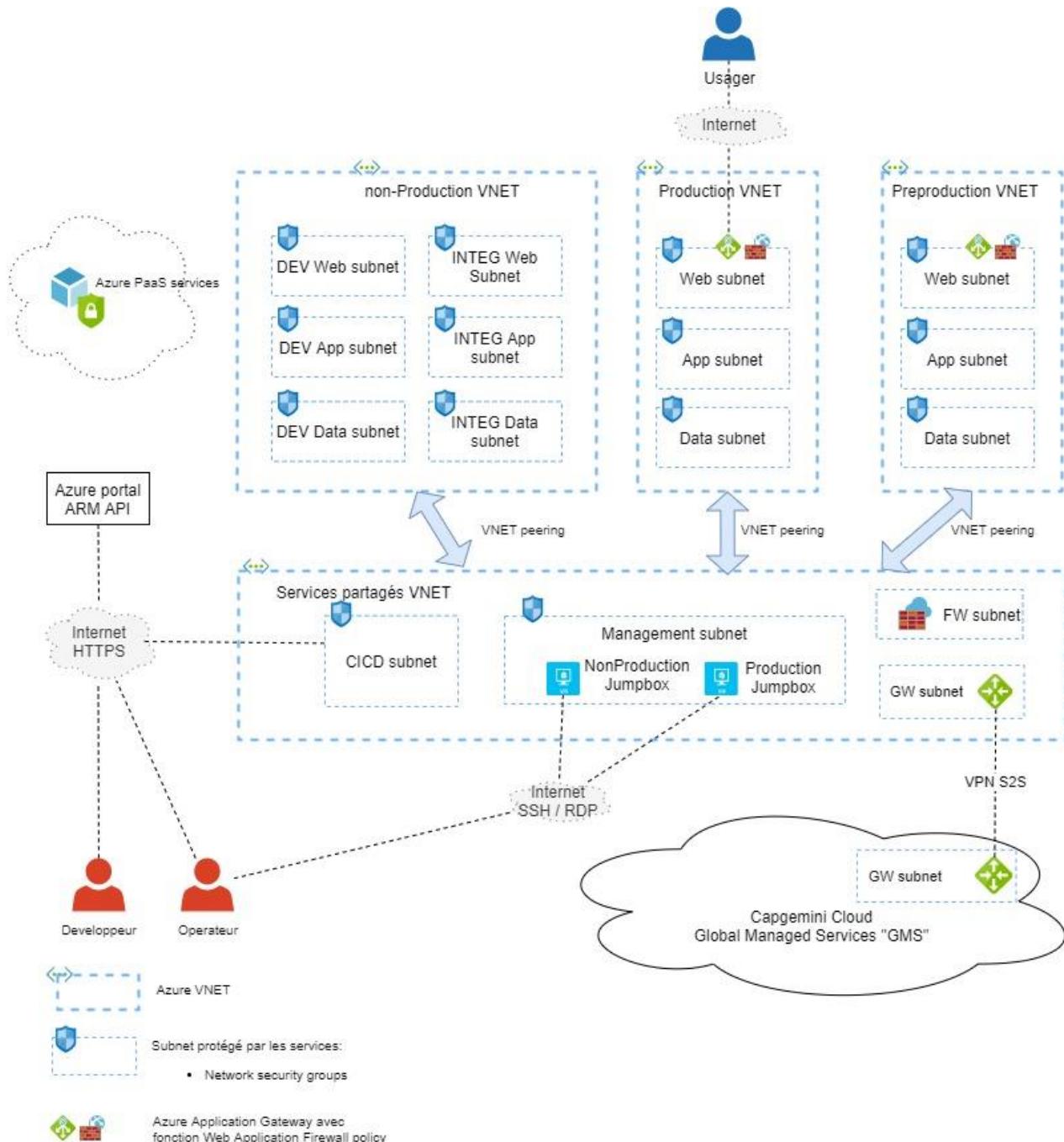


Figure 33 - Schéma réseau dans Azure

6.8. Plateformes et environnements

Nous préconisons de distinguer les plateformes de non-production de celles dédiées à la production car elles devraient être opérées par des équipes distinctes.

Nous recommandons par conséquent les plateformes suivantes :

- 1 **infrastructure Cloud** pour héberger les ressources informatiques nécessaires au projet, en provisionner rapidement de nouvelles et bénéficier de services avancés :
- 1 plateforme de **développement** opérée par l'équipe de développement : Cette plateforme contiendrait deux environnements develop et testing, elle permettrait de développer le produit et tester les évolutions en interne avec des services partenaires bouchonnés.

- 1 plateforme de **préproduction** opérée par l'équipe infrastructure : elle permettrait de valider les évolutions des interfaces avec les plateformes de tests de nos partenaires pilote.
- 1 plateforme de **production** opérée par l'équipe infrastructure : elle permettrait de déployer l'expérimentation pour nos partenaires pilotes sur une plateforme stable.
- 1 plateforme pour le **build et l'intégration continue CI/CD** permettant d'optimiser la chaîne de construction et de déploiement de l'infrastructure et des développements logiciels sur les différents environnements.

La répartition des plateformes dans les souscriptions Azure serait la suivante ; chaque plateforme bénéficierait de sa propre souscription :

- La **plateforme de développement** serait affectée à une **première souscription**.
- La **plateforme de préproduction** sera affectée à une **seconde souscription**.
- La **plateforme de production** sera attribuée à une **troisième souscription**.
- La plateforme de **build et d'intégration continue CI/CD** serait isolée dans une **4ème souscription**. Cette dernière sera conjointe avec MOB / CMS durant l'expérimentation.

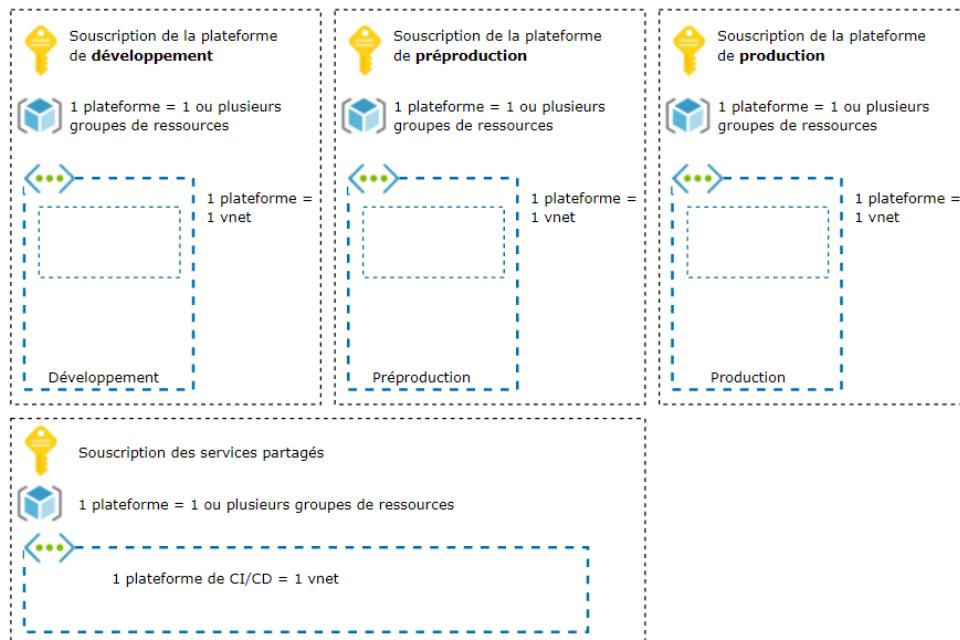


Figure 34 - Organisation des ressources dans Azure

Le tableau ci-dessous détaille les environnements techniques ainsi que les souscriptions dans le cloud Azure :

Landscape	Souscription Azure	Suffixe DNS	Résumé / Commentaires / Usage
CI / CD	SS- MCM-shared	*.cicd.moncomptemobilite.fr	<p>Ressources « Forge » partagées sur tout le programme MCM.</p> <p>Les pipelines de chaque projet GitLab permettent de déployer sur les landscapes DEV/ PREPROD/ PROD au moyen de ces ressources.</p> <p>1 cluster Docker Swarm constitué de :</p> <ul style="list-style-type: none"> • 1 VM GitLab Premium + Nexus • 1 VM Sonarqube

DEV	SS-MCM-StdMaaS-developement	*-develop.preview-gw.moncomptemobilite.fr *.testing-gw.moncomptemobilite.fr	1 cluster AKS de DEV global permettant d'héberger : <ul style="list-style-type: none"> 1 instance Gateway develop.preview-gw basée sur la branche develop du dépôt Gateway. Cette instance est redéployée à chaque merge/commit, les données sont potentiellement rechargées à chaque fois. 1 instance [uservice]-[feature-nroBranche-description]-develop.preview-gw par branche US. Ces instances ont une durée de vie limitée. 1 instance Gateway [uservice].testing-gw basée sur une Release Candidate GitLab. Cette instance est stable et s'appuie sur des BDD managées Azure. Les données ne sont jamais écrasées, seul le delta de version RC est pris en charge. C'est le dernier environnement avant livraison aux équipes OPS.
PREPROD	SS-MCM-StdMaaS-Preproduction	*.preprod-gw.moncomptemobilite.fr	1 cluster AKS de préprod permettant d'héberger 1 instance Gateway preprod-gw <ul style="list-style-type: none"> Version « ISO-prod » à destination de l'exploitant afin de reproduire et corriger les anomalies constatées en production ainsi qu'au partenaires de la Gateway afin de valider l'intégration des évolutions du produit
PROD	SS-MCM-StdMaaS-production	*.gw.moncomptemobilite.fr	1 cluster AKS de prod permettant d'héberger 1 instance de la Gateway

7. Sécurité

7.1. Gestion des identités, identification & authentification

7.1.1. Acteurs humains

Catégorie utilisateur	Authentification
Administrateur technique	Accès VPN SSL/TLS + Bastion Gestion comptes administrateur via console d'administration Super utilisateur : accès via user/password Administrateur : accès via AK (Access Key) / SK (Secret Key) Gestion authentification et autorisation Web SSO (IAM) sur les ressources : <input type="checkbox"/> Logging
Superviseur	Accès VPN SSL/TLS + Bastion Gestion authentification et autorisation Web SSO (IAM) sur les ressources : <input type="checkbox"/> Logging <input type="checkbox"/> Monitoring & Supervision
Développeur	Accès internet SSL/TLS

7.1.2. Applications clientes

Catégorie application	Authentification
MaaS	Accès internet SSL/TLS Gestion comptes via la solution API Management (apiKey) Gestion authentification et autorisation Web SSO (IAM) sur les ressources : <input type="checkbox"/> Logging <input type="checkbox"/> Monitoring & Supervision

7.2. Certificats serveurs et nom de domaine

Les certificats serveur utilisés sont obtenus auprès de l'autorité Let's Encrypt : crt.sh/gw.moncomptemobilite.fr

7.3. Autorisation et contrôle d'accès

Les permissions d'accès sont basées sur RBAC (Role-Based Access Control) et ACL (Access Control List).

7.4. Intégrité

L'intégrité des échanges est assurée par les protocoles HTTPS, SSL/TLS.

La Gateway MCM Std MaaS ne gère aucune donnée métier en persistance, l'intégrité des données concerne uniquement :

- Les données de configuration des partenaires (token, urls, mappings de champs entre modèle interne / modèle externe)
- Les données d'initialisation de certains partenaires (liste des adresses de parkings, des différentes zones, ...)
- L'intégrité des données concerne donc en premier lieu les échanges pour lesquels la Gateway assure la médiation entre la plateforme et les partenaires

Les éléments de l'architecture qui permettent d'assurer l'intégrité sont en particulier l'utilisation du chiffrement des flux entre les différents partenaires de la Gateway.

Concernant les données persistées, les accès à la base de données font l'objet d'un contrôle d'accès, et sont injectées depuis une source de données de référence gérées en configuration (GitLab).

Concernant les échanges asynchrones, mettant en jeu des données conservées en tampon par la Gateway, l'intégrité des données sera assurée par la recette fonctionnelle de l'application.

7.5. Confidentialité

Tous les échanges externes sont chiffrés en utilisant SSL/TLS, HTTPS ;

L'accès aux APIs HTTPS est sécurisé par une clé d'API, révocable à tout moment.

7.6. Traçabilité / Journalisation

La suite Grafana est utilisée pour collecter et centraliser les logs.

7.6.1. Logs techniques et applicatives

Les données concernant les logs techniques et applicatives seront journalisées. La durée de rétention reste à définir (durée courte).

7.6.2. Logs administrateurs techniques et fonctionnels

Tous les accès et actions (CRUD) réalisés par les administrateurs techniques et fonctionnels dans le système seront loguées. La durée de rétention reste à définir (durée longue correspondant à la période légale).

7.7. Imputabilité et non répudiation

L'imputabilité se base sur les mécanismes de journalisation mis en œuvre.

Le besoin de gérer la non-répudiation n'a pas été identifié comme exigence ; aucune mesure n'est mise en place.

7.8. Anonymisation & Pseudonymisation des données

Les données personnelles transitent par la plateforme mais ne sont pas stockées, seules les données de profil et d'usage le sont, associées à des identifiants techniques non signifiants fonctionnellement.

Notamment, les données utilisées pour les statistiques se basent sur les identifiants techniques et ne nécessitent pas de processus d'anonymisation et/ou de pseudonymisation.

7.9. GDPR

Il n'y a pas de stockage de données personnelles liées au citoyen.

7.10. Sauvegarde / restauration

Les logs techniques et applicatives sont sauvegardés. La durée de rétention des sauvegardes est paramétrable et reste à préciser.

La stratégie de sauvegarde est adressée au niveau applicatif :

- dump des bases de données avec les outils natifs des solutions (PostgreSQL)
- fonctionnalité de snapshots disponible nativement et instantanée
- utilisation du service de stockage objet pour archiver des données

Les snapshots permettent au CSP de pouvoir récupérer les données en cas de désastre (le cas échéant).

NB : Les sauvegardes sont stockées chiffrées.

7.11. Purge

Les données de logging et d'usage de l'API Management sont purgées selon une périodicité configurable. Elle reste à définir.

7.12. Archivage

Pour répondre aux exigences juridiques et légales, les logs des accès des administrateurs (techniques et fonctionnels) sont sauvegardés et archivés. La durée de rétention des archives est à définir.

7.13. Tests d'intrusion / Revues de Code Source

Une campagne de tests d'intrusion a été menée par une équipe spécialisée en cybersécurité sur une instance de la plateforme. Les vulnérabilités remontées ont été traitées.

Une analyse statique de sécurité du code (SAST) est exécutée périodiquement sur le code source du produit. Les vulnérabilités détectées sont traitées au fur et à mesure.

7.14. Limitation volumétrie

Le gestionnaire d'API et le contrôleur Ingress permet de mettre en œuvre du rate-limiting / throttling permettant de limiter le taux de sollicitation des APIs.

7.15. Anti-DDoS

L'outil NGINX ModSecurity WAF est mis en œuvre pour protéger l'application contre les attaques DDoS en traitant les requêtes dans un seul thread de manière asynchrone, offrant ainsi une faible utilisation de la mémoire.

7.16. Cloisonnement

7.16.1. Cloisonnement zones

Le système est découpé en zone cloisonnées (zone front-end, zone applicative, zone données, zone administration fonctionnelle, zone logging, supervision & monitoring).

7.16.2. Cloisonnement réseau

Les réseaux sont cloisonnés en VPC, VLAN / subnet.

7.17. Système de détection d'intrusion

La détection d'intrusion est basée sur l'analyse des logs de journalisation.

7.18. Montée de version des systèmes d'exploitation et patchs sécurité

L'ensemble des actions de montée de version des systèmes d'exploitation et de mise à jour des patchs sécurité est réalisé par l'équipe d'exploitation.

8. Modèle de dimensionnement théorique

Le **dimensionnement théorique de l'environnement de production** du système est basé sur :

- les entrants fournis par l'écosystème,
- les hypothèses proposées par Capgemini,
- les retours d'expérience de Capgemini dans le cadre de projets similaires,
- des métriques et des abaques.

L'ensemble de ces éléments doit permettre de réaliser un premier modèle de dimensionnement théorique du système MCM Std MaaS GW, qui reste à établir.

8.1. Entrants écosystème

Entrants	PMV	Cible basse	Cible haute	Commentaires
Nombre utilisateurs MaaS 1 (citoyens / usagers)				
Nombre utilisateurs MaaS 2 (citoyens / usagers)				

8.2. Hypothèses de Capgemini

Hypothèses	PMV	Cible basse	Cible haute	Commentaires
Nombre MaaS (en moyenne)	1			
Nombre MSP (en moyenne)	5			

8.2.1. Interrogations « IV »

Un MaaS constitue un cache / une image du positionnement des mobilités. Les sollicitations « Autour de moi » sur l'application MaaS n'implique pas d'interaction équivalente directe sur la Gateway MCM Std MaaS.

La Gateway MCM Std MaaS répond aux sollicitations du backend d'un MaaS pour l'entretien de son cache :

- 1 groupe d'appel toutes les 2 minutes**
 - o Ce groupe d'appel représente **une dizaine appels véhicules et stations retournant 5Mo de données** (nombre exact dépend du nombre de MSP)

La Gateway MCM Std MaaS constitue elle-même un cache de données :

- 1 groupe d'appel toutes les 2 minutes / 5 minutes / 24h**
 - o Ce groupe d'appel représente **une vingtaine appels véhicules et stations retournant 5Mo de données** (nombre exact dépend du nombre de MSP)

8.2.2. Interrogations « RI »

Un appel RI sur l'application nécessite 3 à 5 appels à la Gateway selon le nombre de combinaisons possibles (nous retenons **5 appels**).

Entrants	Utilisation	Nombre d'appel à la Gateway	Pic
Perturbations dans les transports à une heure d'entrée ou de sortie des bureaux	80% des utilisateurs font appels à la RI par 2 fois sur une durée de 1 heure	$(2000 \times 0,8) \times 2 \times 5 = 16\ 000$ Soit : 266 appels/min	5 appels/s
Plage « Journée »	50% des utilisateurs font appels à la RI par 1 fois en 600 minutes	$(2000 \times 0,5) \times 5 = 5\ 000$ Soit 8,3 appels/min	1 appels/s
Plage Heure de pointe	50% des utilisateurs font appels à la RI par 1fois en 120 minutes	$(2000 \times 0,5) \times 5 = 5\ 000$	1 appels/s

8.2.3. Interrogations « réservations / usage » covoiturage

Chaque demande de réservation sur l'application mobile se traduit par une interaction équivalente sur la Gateway MCM Std MaaS

- On retient **1 appel/s**

8.3. Modèle de dimensionnement théorique

Dimensionnement	PMV	Cible basse	Cible haute	Commentaires
Nombre connexions concurrentes				
Nombre calls APIs par seconde				
Nombre calls SQLs par seconde				
Taille offres (moyenne) (en Mo)				
Taille données utilisateurs (en Go)				

8.4. Performances

Pour la phase d'expérimentation, les services GW seront sollicités par les MaaS/MSP des territoires pilote.

En pic le pilote MaaS devra supporter une activité correspondant à :

- 80% des utilisateurs exécutant 1 « Autour de moi » + une RI sur une durée de 1h (Simulation d'une activité liée à un incident de transport ou heure de pointe)

Pour gérer et améliorer la performance :

- La Gateway MCM Std MaaS utilise une solution de cache partagé pour fournir les informations relatives aux MSP. Il s'agit des données statiques ou à mise à jour de faible fréquence
- La Gateway MCM Std MaaS est déployée sur plusieurs nœuds, une répartition de charge permet d'alimenter ces nœuds.
 - o Il est possible de « gonfler » chaque nœud, ou d'ajouter un nœud sans modification de l'application, uniquement de la configuration Kubernetes
- Mise en place de tests de performances permettront de mesurer le comportement
 - o La Gateway met en place des métriques permettant d'identifier les performances de chaque élément de la chaîne de liaison

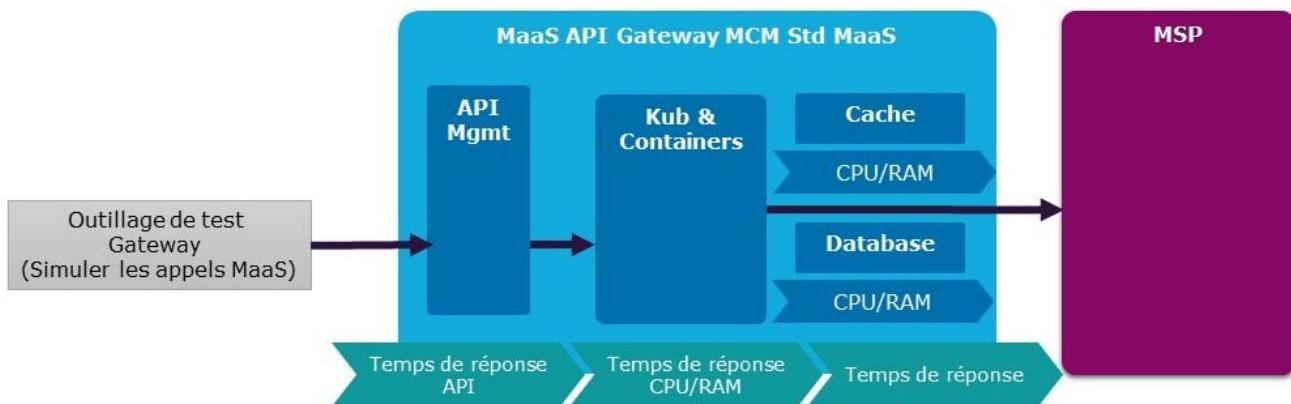


Figure 35 - Mesure des performances

9. Processus de livraison

Trois repositories sont concernés par le process de déploiement.

- Un premier gère les pipelines de build, de déploiement sur un environnement par branche avec du docker kompose, gère les releases et la publication des artefacts et helm charts.
- Deux autres repositories existent avec des comportements similaires mais pour des déploiements sur des environnements distincts.
 - o D'un côté preview et testing dans les mains des DEVs
 - o D'autre part les déploiements en preproduction et production dans les mains des Ops.

Les pipeline helm chart et infra (delivery-ops) sont poussées par la pipeline Gateway (celle de build). Ces pipelines qui sont poussées permettent le déploiement des ressources Kubernetes par utilisation des helm charts. La publication d'une branche dans laquelle est suivi d'une demande de merge request sur la branche principale.

Les ops valident la merge request et peuvent lancer le déploiement du ou des services de la gateway.

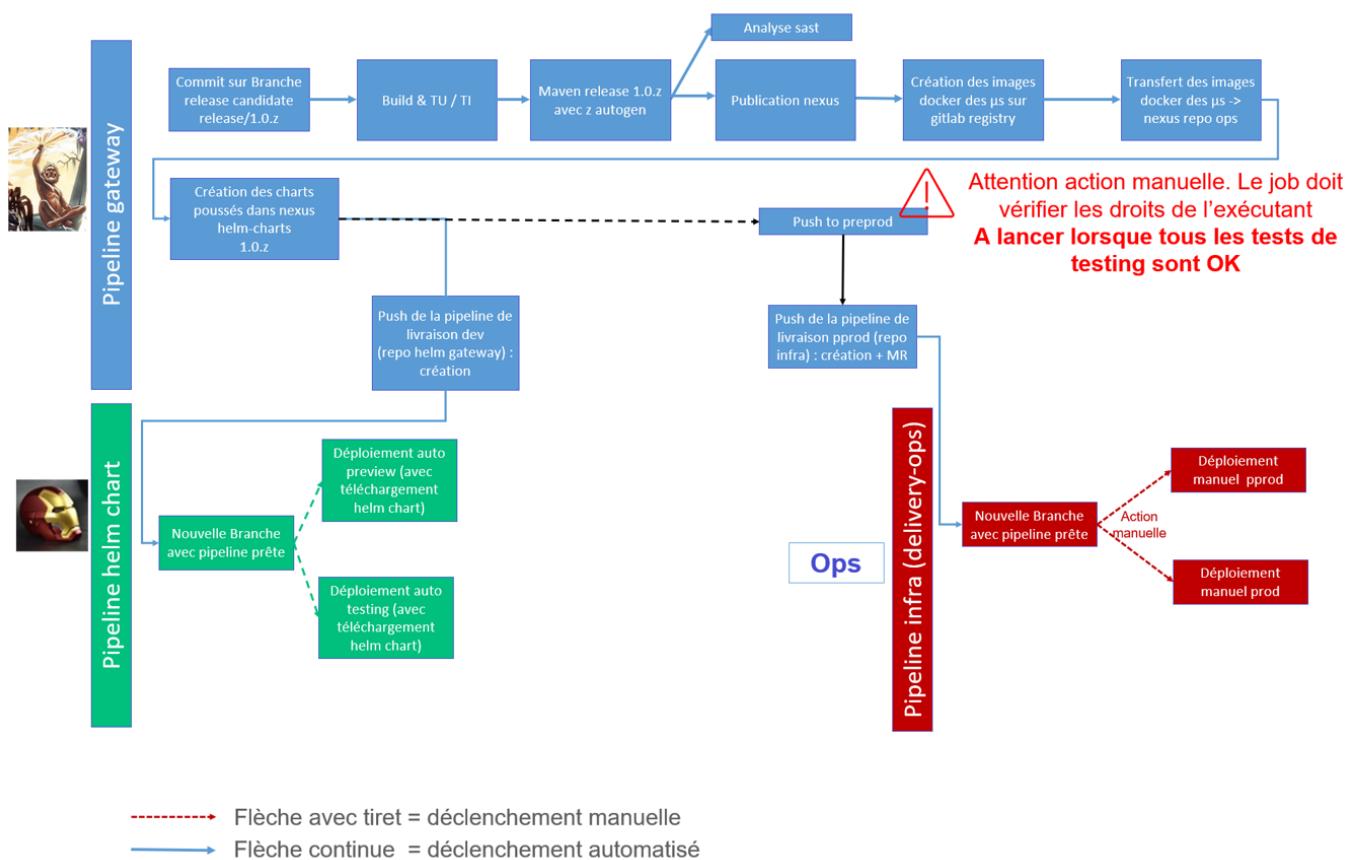


Figure 36 - Processus de livraison

10. Approche DevOps

Le style d'architecture choisi — orienté microservices — conduit à un nombre important d'unités de déploiement. En outre, le passage à l'échelle implique une multiplication des instances/replicas. Par ailleurs, par rapport à une approche monolithique, le caractère distribué du système introduit de la complexité supplémentaire : les appels inter-services qui jusqu'ici étaient in-proc sont maintenant distants, visibles de l'extérieur. Ces appels sont par conséquent sensibles aux aléas du réseau et sont susceptibles d'échouer plus fréquemment.

Toutes ces raisons font qu'il est hautement désirable d'introduire certaines contre-mesures :

- Une stratégie de test appropriée devrait être mise en œuvre ; celle-ci requiert la capacité de créer de multiples environnements, à la demande ;
- Des patterns de stabilité tels que les timeouts, les circuit breakers, etc. devraient être implementés ;
- Un monitoring de bout en bout devrait être implanté afin d'identifier rapidement tous incident pendant l'exploitation ;
- L'environnement de production devrait utiliser les mêmes binaires que ceux de validation ; la structure de ces environnements devrait être identique ; la configuration devrait être réalisée au moyen de variables d'environnement ;
- Toutes les opérations de déploiement devraient être automatisées de façon à éviter les erreurs et pour pouvoir récupérer rapidement.

Des scripts sont communément utilisés pour automatiser les opérations de build, test et déploiement. Le principal inconvénient est qu'il s'agit d'une approche impérative qui impose à l'auteur des scripts de connaître l'état initial du système ainsi que toutes les opérations nécessaires pour atteindre l'état cible. Une attention particulière est alors requise pour s'assurer que ces scripts soient idempotents.

Dans le but de rendre le système le plus déterministe possible, nous préconisons d'éviter tout script impératif et de favoriser une approche déclarative. C'est la raison pour laquelle nous envisageons de recourir à Terraform et/ou Ansible pour toutes les tâches bas niveau de déploiement et sur des manifestes Kubernetes pour le déploiement d'applications, sachant que ces technologies encouragent nativement un style déclaratif.

Une dernière qualité opérationnelle souhaitable est la capacité de conserver un historique de tous les déploiements et d'avoir la possibilité de revenir à un état antérieur si nécessaire. Malheureusement, ceci n'est pas proposé par Kubernetes. Une fois qu'un manifeste est soumis, l'orchestrateur ne se souvient plus de l'état précédent du cluster. De plus, les déploiements ne sont pas transactionnels et peuvent laisser le système dans un état imprévisible en cas de problème.

10.1. Introduction à GitOps

Le problème exposé ci-dessus peut être résolu en adoptant une approche à 2 niveaux appelée « GitOps ». Ce concept peut être défini par la formule suivante :

*Tout en tant que code + Configuration déclarative + Gestion de versions + Pull/Merge requests
 + Opérateur de réconciliation*

=

Déploiement automatique dans Kubernetes

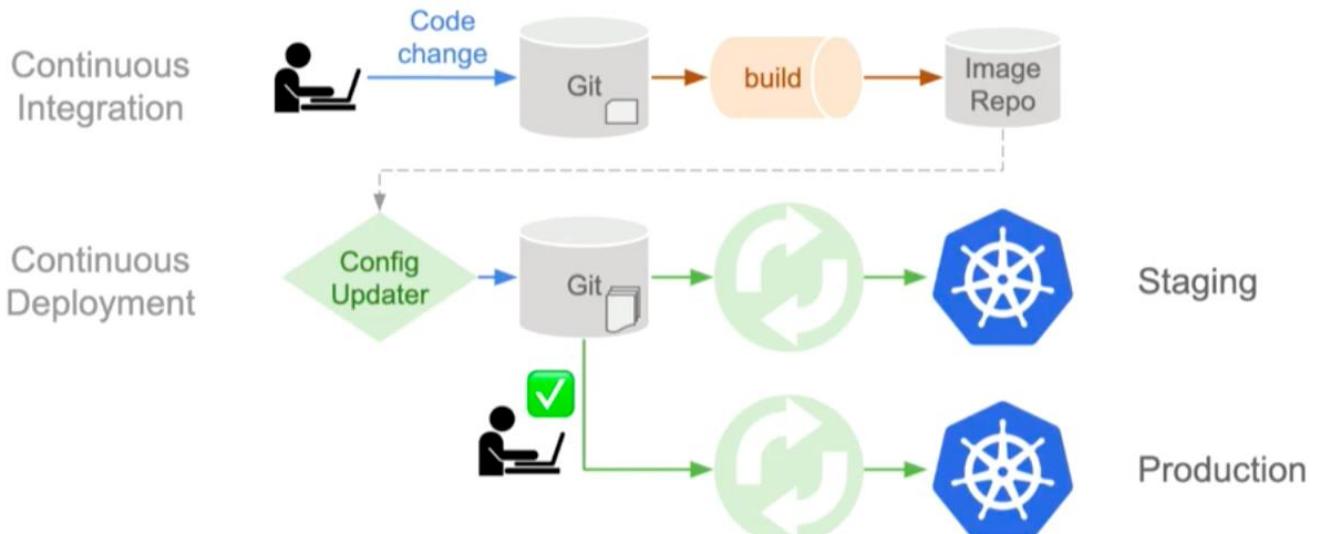


Figure 37 - Principes de GitOps

Source : <https://www.weave.works/blog/automate-kubernetes-with-gitops>

La première partie du flux de travail est une intégration continue classique : lorsque les développeurs soumettent des modifications de code dans le dépôt git, un pipeline de construction et d'intégration est automatiquement déclenché. Puisque nous construisons un système basé sur des microservices, le pipeline devrait produire des images de conteneurs stockées dans un référentiel d'images.

Le déploiement continu est géré par un outil spécial appelé Réconciliateur (Config Updater) : il est chargé de récupérer la dernière version des manifestes de déploiement, de les mettre à jour avec les versions d'image appropriées et de soumettre le résultat final dans un second dépôt git dédié aux opérations. Enfin, au lieu de pousser la configuration vers le paysage cible, un service dédié s'exécutant à l'intérieur du cluster Kubernetes est chargé d'extraire les manifestes de git et de les appliquer de manière transactionnelle.

10.2. Implémentation

Nous avons exposé au cours des paragraphes précédents l'approche que nous préconisons pour la chaîne de déploiement continu. Elle devrait présenter les caractéristiques suivantes :

- Recourir à l'Infrastructure as Code pour automatiser intégralement le déploiement de l'infrastructure aux couches applicatives.
- Faire de Git le point d'entrée unique pour la gestion de l'infrastructure et les déploiements applicatifs. Promouvoir les builds à l'aide des mécanismes de Pull/Merge Requests.
- Adopter un style purement déclaratif.
- Réserver un dépôt pour le code applicatif et un autre référentiel distinct pour l'infrastructure et la mise en production.
- Gérer l'état de Kubernetes à l'aide d'un opérateur déployé dans chaque cluster. L'opérateur surveille le dépôt Git approprié et répond aux événements pertinents (commits, pushes, tags).

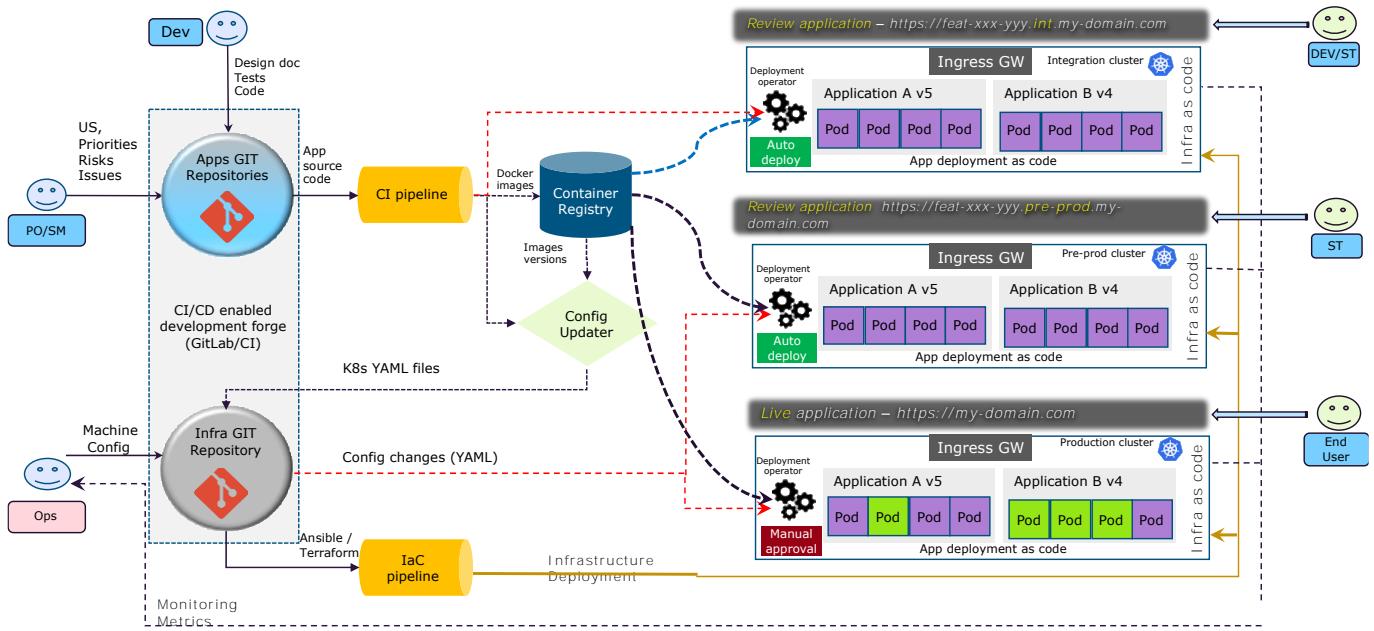


Figure 38 - Approche DevOps : orientation GitOps

Le réconciliateur est déclenché par l'équipe de développement lorsqu'elle considère le seuil de qualité atteint. Cela peut être fait à partir d'une exécution du pipeline de build existante et réussie en lançant manuellement une tâche supplémentaire dédiée. Le réconciliateur ne pousse que les manifestes compilés et la configuration vers le dépôt Git réservé à l'infrastructure. Cela déclenche un déploiement immédiat dans le cluster de préproduction sans rien reconstruire. L'opérateur du cluster de préproduction extrait simplement les bonnes images du registre de conteneurs et crée les services décrits dans les manifestes.

Enfin, une opération manuelle est nécessaire pour provoquer le déploiement final dans l'environnement réel de production.

Nous préconisons l'emploi de GitLab qui fournit en un seul outil une expérience intégrée couvrant l'ensemble du cycle de développement.

Par ailleurs, nous recommandons également de mettre en place :

- 1 dépôt unique avec une seule branche pour la gestion des infrastructures.
 - 1 dépôt unique pour l'ensemble du code applicatif. Organiser les modules dans une hiérarchie de dossier, selon une approche monorepo pour simplifier la gestion des dépendances.
- Le pipeline doit être conçu de sorte à construire et déployer uniquement les modules modifiés.

10.3. Pipelines de build et déploiement

10.3.1. Inclusion dans le processus de livraison

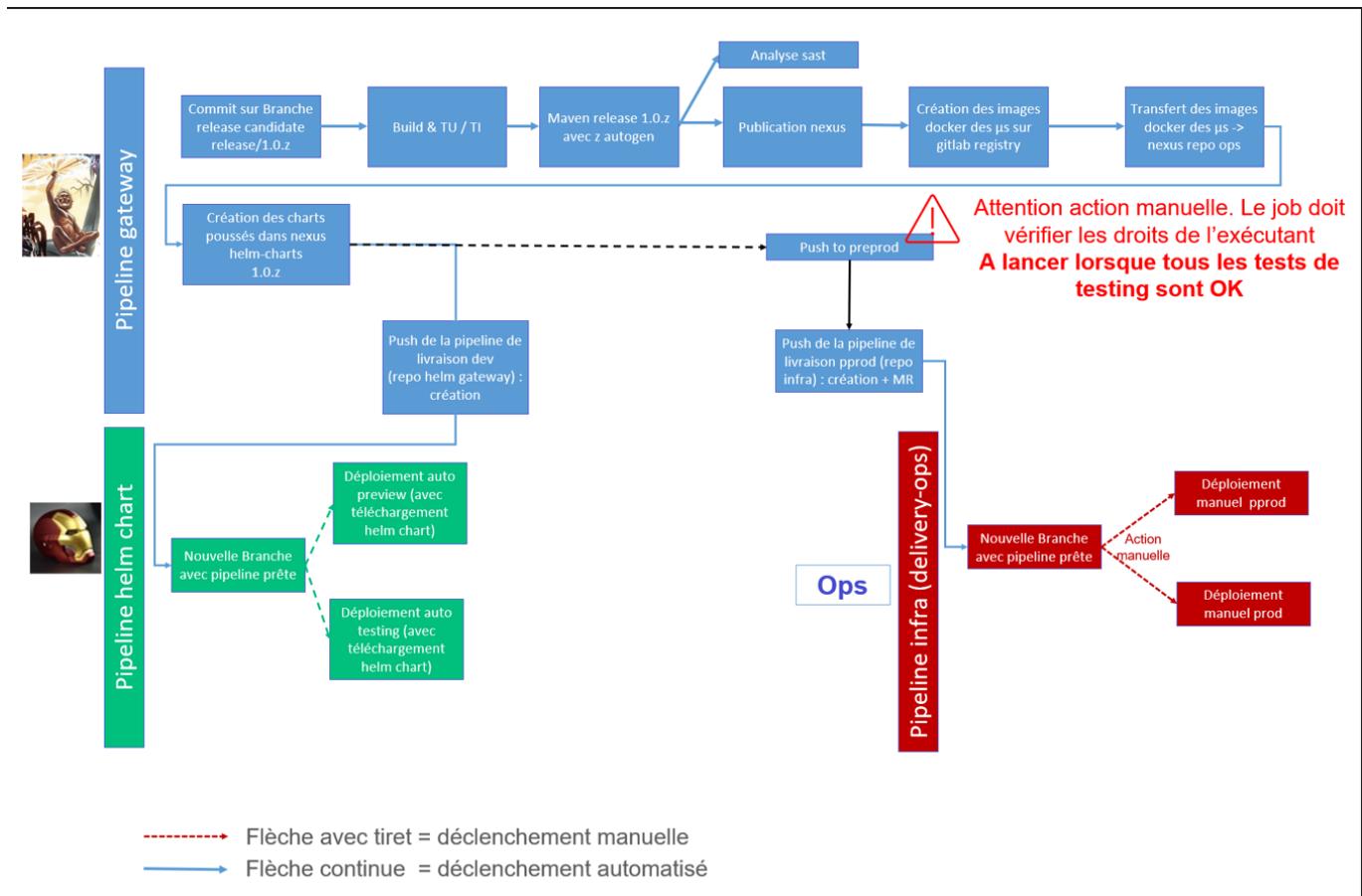


Figure 39 – Pipeline et livraison

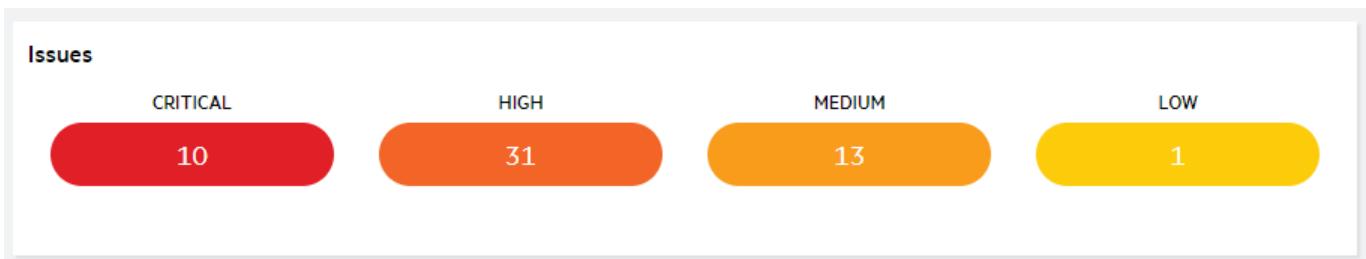
10.3.2. Publication du code

Plusieurs étapes ont été définies pour mener à bien la publication du code en Open source :

Analyse SAST

Mise en place d'une analyse SAST (Static Application Security Testing) du code permettant de détecter les anomalies critiques/majeurs potentielles à corriger avant la mise en production du PMV et la publication du code. L'analyse a été effectuée en utilisant l'outil « [Fortify On Demand](#) », paramétré pour analyser la branche principale du projet.

L'analyse permet de remonter les vulnérabilités potentielles classées en 4 criticités : critiques, hautes, moyennes et faibles, comme le montre un tableau résultat exemple ci-dessous :



Ces vulnérabilités sont traitées au fur et à mesure en amont de chaque publication. Certaines restent en l'état car non applicables dans le contexte d'utilisation du projet.

Nettoyage de code

Le nettoyage du code est essentiel avant la publication de celui-ci. Plusieurs actions ont été identifiées :

- Déterminer le périmètre et les parties de code qu'il faudra publier.
- Nettoyer le code du mot de passe et liens référencés en dur dans le code en utilisant des variables
- Vérifier les noms des variables, des fonctions et commentaires qu'il faudra traduire en anglais
- Vérifier le contenu des TU et des mocks.

Documentation

La documentation est l'un des points nécessaires à mettre en place avant de prévoir une publication.

Les documents tel que le DAT, les cas d'utilisation, le guide de configuration et d'exploitation de la Gateway sont essentiels. Un répertoire doc a été créé afin de référencier tous ces documents importants.

D'autres documents sont également prévus à l'avenir :

- Le guide d'administration de la gateway
- Le document de livraison

Pipeline de publication

Lorsqu'une branche est livrée aux ops, un tag est posé. La pipeline de build peut permettre la publication de code sur un repository github dont les informations sont paramétrées dans les variables de CI/CD de gitlab.

Afin de publier le code il est impératif de sélectionner un tag, et de passer en variable de lancement de la pipeline la variable PUBLISH_CODE avec comme valeur « true ».

Ainsi au lieu de lancer la pipeline de build c'est une pipeline de publication du code qui se déclenche. Une branche est créée sur le repository github basé sur le nom du tag et une pull request est ouverte.

