Table des matières

[Zones physiques 2](#_Toc24618716)

[Considérations pour la plateforme de conteneurisation (Openshift) 2](#_Toc24618717)

[Zones de sécurité 2](#_Toc24618718)

[Tenants 2](#_Toc24618719)

[Zones logiques 3](#_Toc24618720)

[Considérations pour la plateforme de conteneurisation (Openshift) 3](#_Toc24618721)

[Zones et domaines d’intégration 4](#_Toc24618722)

[Définitions 5](#_Toc24618723)

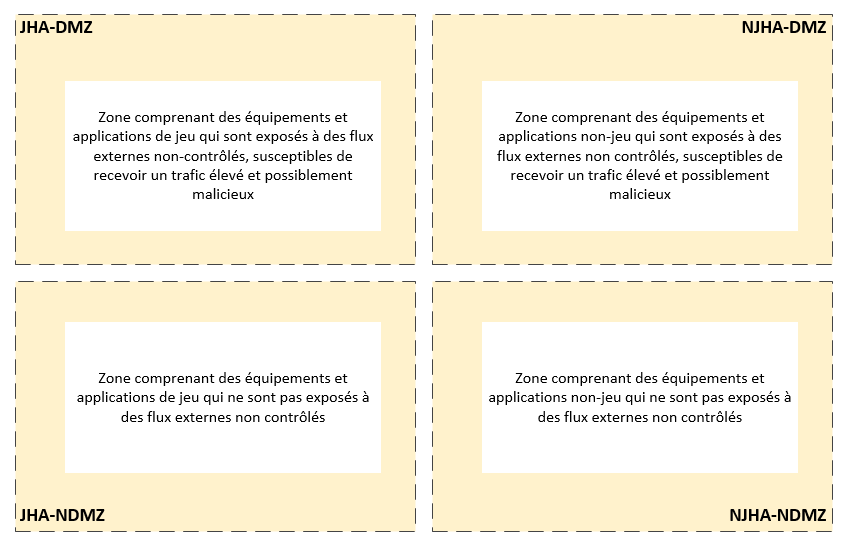
[Exemple :Domaine d’intégration Clients 6](#_Toc24618724)

[Types d’API 7](#_Toc24618725)

[Lignes directrices URL 8](#_Toc24618726)

[Exemple concret : identité d’un client LQ (aussi appelée : Partie Prenante Individu - PPI) 9](#_Toc24618727)

# Zones physiques



Références :

Architecture de sécurité 2017-2018 :

<https://portailvpcti.loto-quebec.com/prj/archec17/ar/Architecture%20de%20s%C3%A9curit%C3%A9%20Loto-Qu%C3%A9bec%20-%20Phase%20Syst%C3%A8me%20-%202%20-%20Flux%20r%C3%A9seau.docx>

## Considérations pour la plateforme de conteneurisation (Openshift)

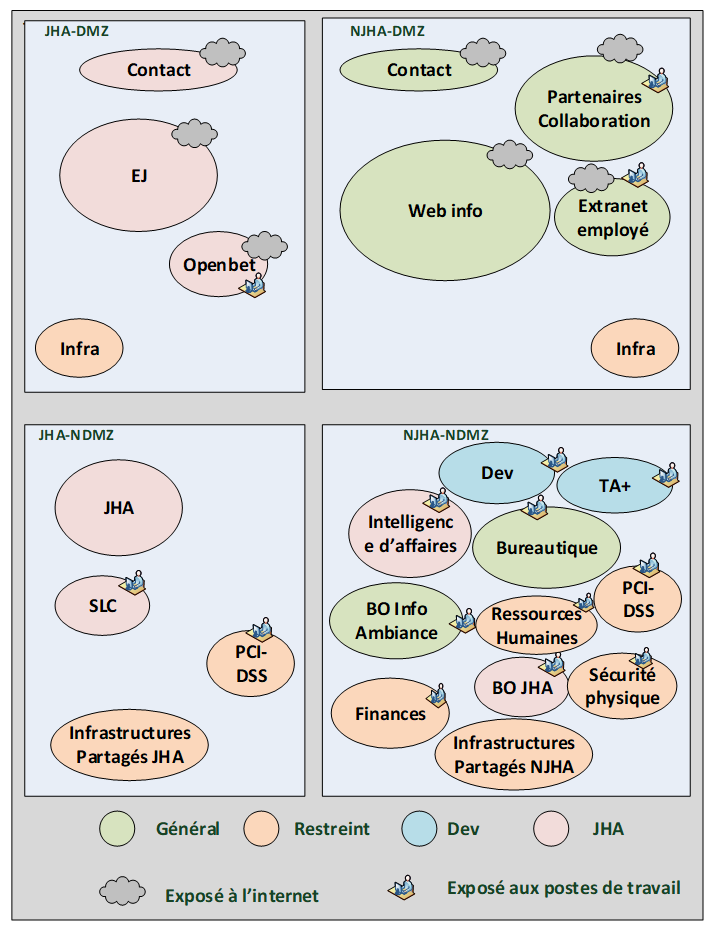
### Zones de sécurité

* Une zone de sécurité n'est pas égale à un VLAN
* Une zone de sécurité n'est pas égale à un Subnet

### Tenants

* Tenant  = autorité/responsable, scope défini.
* Tenant = **Produit ou unité organisationnelle** (ex. Service partagé infra)
* Tenant : relation 1-n avec une zone de sécurité.
* Tenant : relation 1-n namespace Kubernetes
* Un tenant est similaire à VPC (Virtual Private Connection) de AWS (Amazon Virtual Private Cloud)

# Zones logiques



Références :

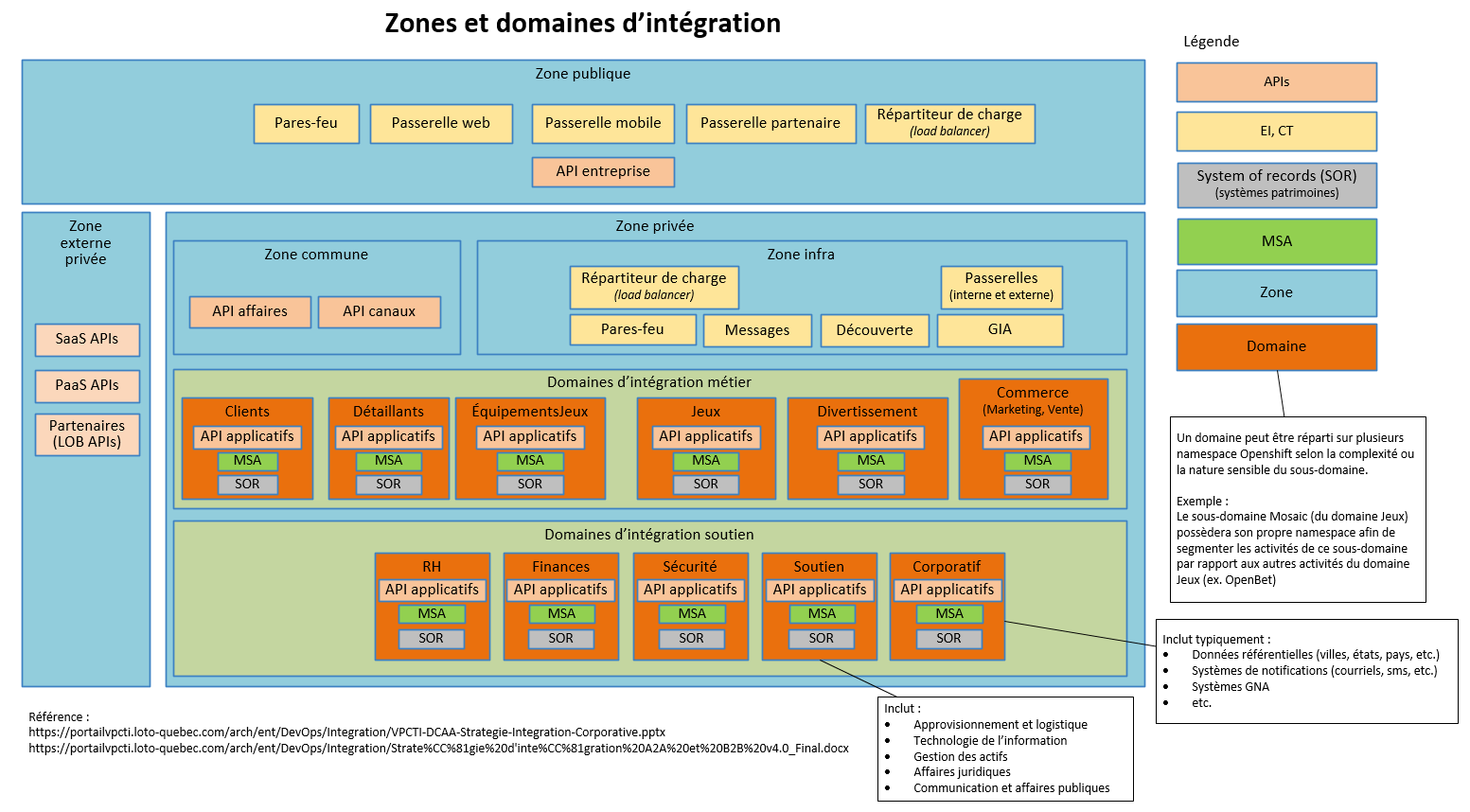
Architecture de sécurité 2017-2018 :

<https://portailvpcti.loto-quebec.com/prj/archec17/ar/Architecture%20de%20s%C3%A9curit%C3%A9%20Loto-Qu%C3%A9bec%20-%20Phase%20Syst%C3%A8me%20-%202%20-%20Flux%20r%C3%A9seau.docx>

## Considérations pour la plateforme de conteneurisation (Openshift)

* Une zone sécurité logique est égale à un tag Kubernetes (catégorie).
  + A valider si un système a une relation de 1 à 1 avec un tag, ou si c'est 1 à n. (voir page To Do)
    - 1 système : relation 1-1 avec un tag (selon PH)
    - 1 système : relation 1-n avec un tag (selon JC)

# Zones et domaines d’intégration



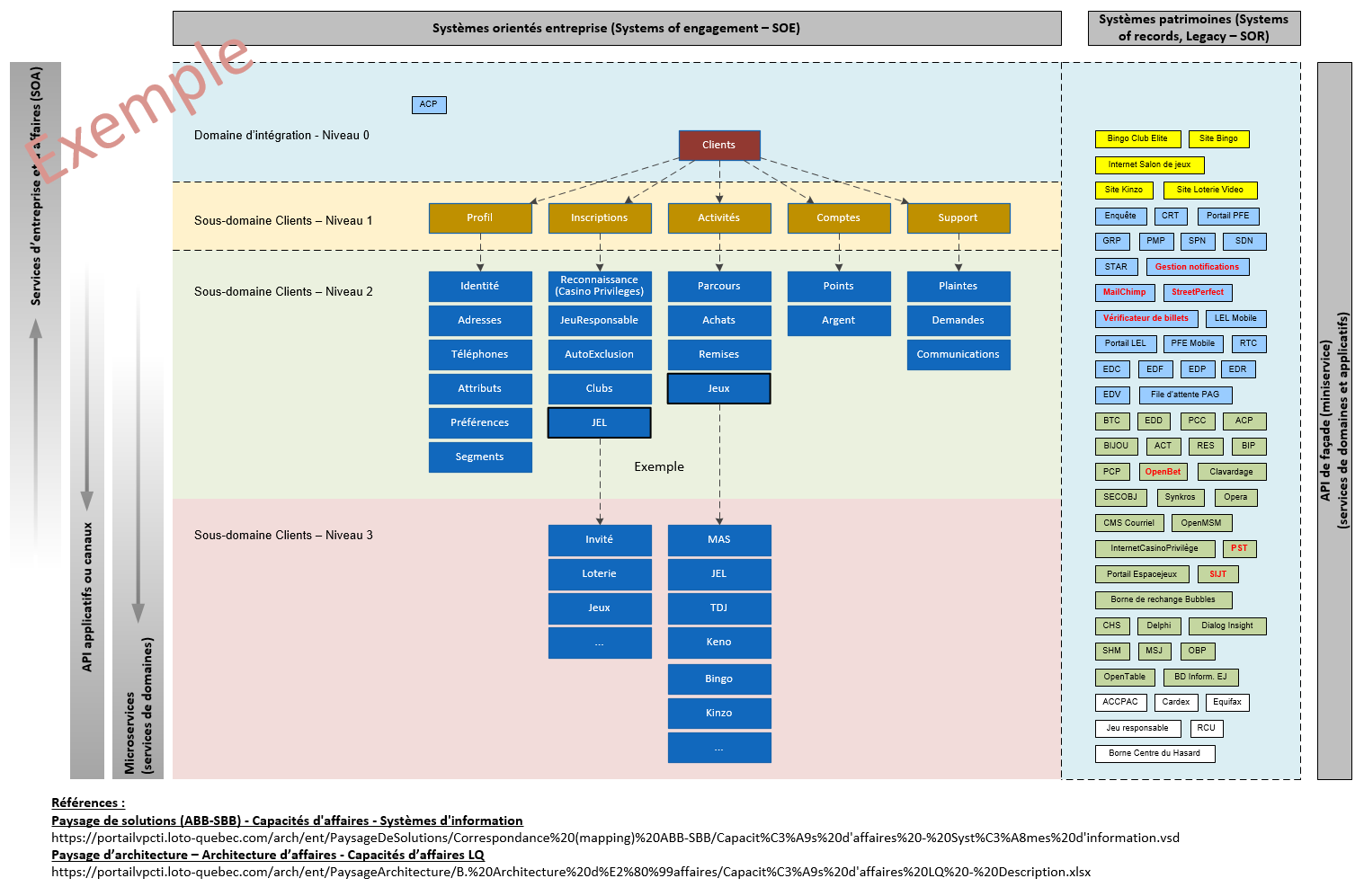
## Définitions

|  |  |
| --- | --- |
| **Zone** | **Définition** |
| publique | À compléter |
| privée | À compléter |
| Externe privée | À compléter |
| Commune | À compléter |
| Infra | À compléter |

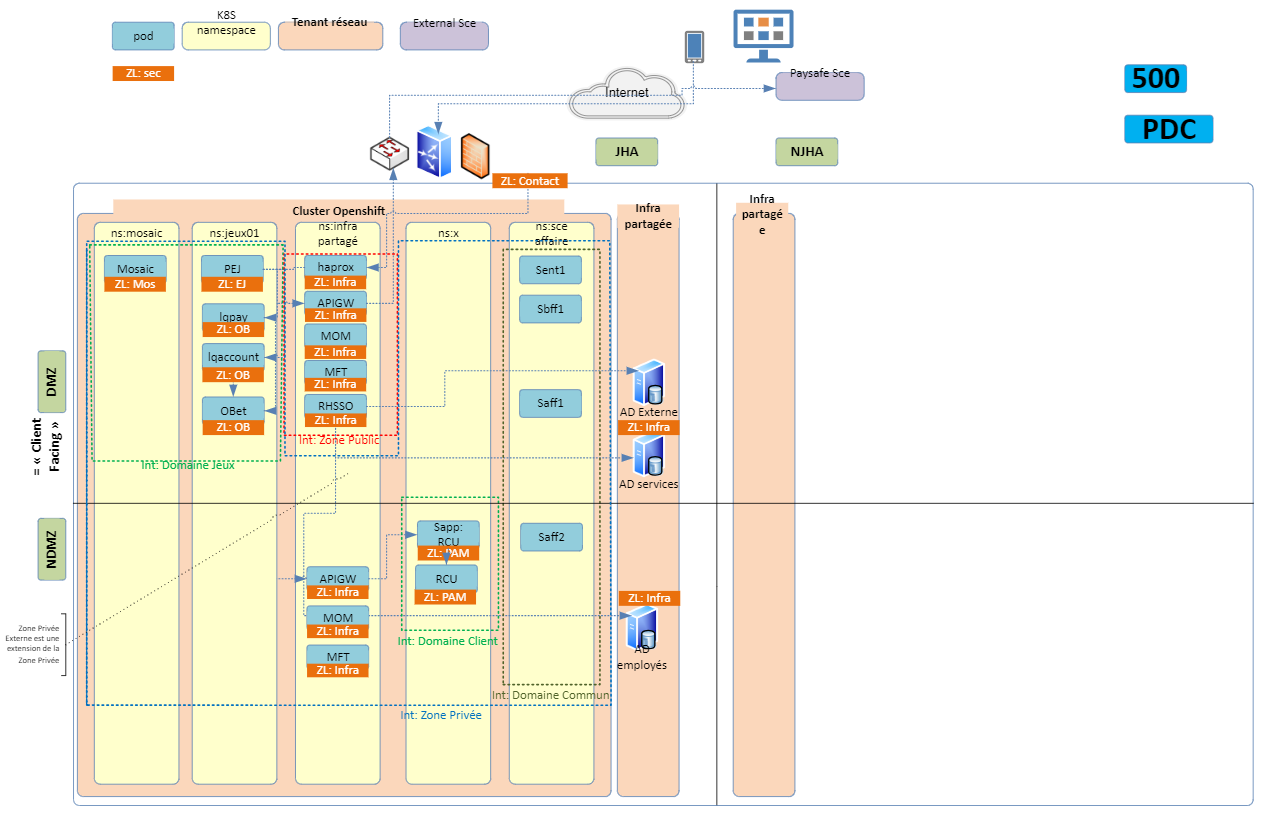
|  |  |
| --- | --- |
| **Domaine** | **Définition** |
| Clients | Domaine métier touchant les clients et les prospects |
| Détaillants | Domaine métier touchant les détaillants (ex. loteries, alv, etc.) et les OBNL. |
| Équipements | Domaine métier touchant la planification, la direction et le contrôle relatifs à l'achat, la distribution et la maintenance des équipements de jeux |
| Jeux | Domaine métier touchant la définition et l’exploitation des jeux (ex. MAS, TDJ, Keno, JEL, etc.) |
| Divertissement | Domaine métier touchant la définition et l’exploitation des divertissements (ex. hébergements, restaurants, spectacles, etc.) |
| Commerce | Domaine métier touchant les capacités d’affaires Marketing et Ventes. Soit offrir une proposition de valeur à nos parties prenantes. |
| RH | À compléter |
| Finances | À compléter |
| Sécurité | À compléter |
| Soutien | À compléter |

Les zones ou les domaines d’intégration sont définis par des libellés (Tag) au niveau de Kubernetes.

## Exemple :Domaine d’intégration Clients

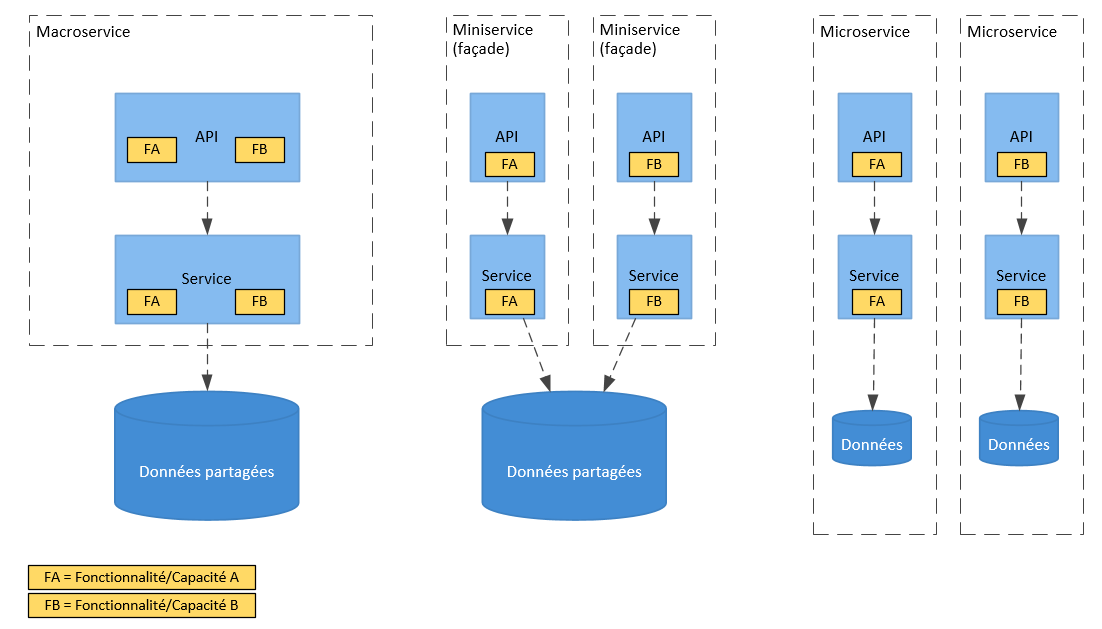


# Patrons d’intégration concrets - Openshift



# API vs \*\*Service (macro, mini, micro, nano…)

|  |  |
| --- | --- |
| **Type** | **Définition, caractéristiques et cas d’usage** |
| Service | Domicile des règles d’affaires d’un domaine ou sous-domaine d’entreprise. Hautement réutilisable (via composition). Testable. |
| API | Expose un **service** (ou plusieurs services) à un consommateur par le biais d’un contrat. |
| Macroservice | Composant API-Service ayant les caractéristiques suivantes :   * Partage ses données * Déploiement monolithique fortement couplé aux données partagées |
| Miniservice (façade) | Composant API-Service ayant les caractéristiques suivantes :   * Partage ses données * Déploiement indépendant non couplé aux données partagées |
| Microservice | Composant API-Service ayant les caractéristiques suivantes :   * Gère ses propres données * Déploiement indépendant fortement couplé à ses propres données |
| Nanoservice | Composant API-Service ayant les caractéristiques suivantes :   * Réalise une seule est unique tâche. Habituellement une portion (et non la totalité) d’une fonction d’affaires * Offre un seul et unique point d’accès (*end-point*, route) * Déploiement indépendant et totalement découplé. * Peut utiliser l’agrégation, la composition ou encore l’orchestration dans la réalisation de sa tâche.   Autres termes : *serverless functionality*, *function as a service* (FaaS)  <https://peterdaugaardrasmussen.com/2019/08/14/what-is-a-nano-service/> |



# Types d’API

|  |  |  |
| --- | --- | --- |
| **Type** | **Définition, caractéristiques et cas d’usage** | **Section de la stratégie d’intégration** |
| Domaine | Autonome, atomique, découplé, résilient.  Est une fondation à MSA ; ce sont les microservices à l’état pur d’un point de vue de la littérature.  Contient la logique d'affaires représentant une capacité d’affaires atomique ou une fonction d’affaires associée à un domaine de l’entreprise.  Possède les caractéristiques d’un microservice.  Intervient seulement comme participant au processus d'affaires. N’est pas le processus d’affaires.  Les services de domaine qui englobent une capacité d’affaires atomique et qui n’ont qu’une seule responsabilité, peuvent être considérés comme un service d’affaires. | 4.6 |
| Applicatif | Peut servir à normaliser (transformation) les entités dans la communication **inter**-domaines (médiation). Exemples :   * API domaine A appelle un API applicatif du domaine B. Patron Request-Response * API applicatif domaine A consomme un message sur un bus (en provenance d’un autre domaine). Patron Publish-Subscribe   Peut servir à intégrer, via un découplage, les APIs externes qui ne sont pas d'origine « LQ » (SaaS, PaaS, LOB interne)  Ne doivent pas composer les services de domaines.  Ne doivent pas faire d’orchestration de services.  Ne doivent pas contenir de logique d’affaires. | 4.7.2  5.2 |
| Canal | Aussi appelés « BFF : Back-end for front-end ».  Requis lorsque la logique d’affaires doit être servie sur plusieurs canaux (web, mobile, assistant, desktop, IoT, etc.).  Pas obligatoire si l’information du système est présentée sur un seul canal. Mais la bonne pratique le recommande pour évolution future.  Inclut aussi les API bidirectionnels (websockets, service workers, etc.) qui requiert un canal de communication permanent (sticky session) comme les applications de clavardage, les applications de jeux en salle, etc.  Lieu possible d’implémentation des capacités de requêtes des API (ex. oData, graphQL) | n/a |
| Affaire | Ne se conforme généralement pas aux normes MSA.  Réalise de la composition ou de l’orchestration de services de domaine et applicatif  Représente le cœur de notre organisation. Contribuent à la réalisation des capacités d’affaires qui génèrent de la valeur ajoutée. Existence en nombre limité  Modélisé selon un modèle canonique | 4.7.3 |
| Entreprise | Séparation des préoccupations entre le domaine privé et le domaine public  Toutes les APIs client ou partenaires sont normalement exposées à travers un service d’entreprise.  Étends la portée des services d’affaires et applicatifs en les exposant à l’externe. | 6.5.1 |

Références :

Stratégie d’intégration : <https://portailvpcti.loto-quebec.com/arch/ent/DevOps/Integration/Logimethods/Strate%CC%81gie%20d'inte%CC%81gration%20A2A%20et%20B2B%20v4.0_Final.docx>

BFF : (Back-end for front-end) : <https://docs.microsoft.com/en-us/azure/architecture/patterns/backends-for-frontends>

# Lignes directrices URL

À aligner avec les travaux du CSI.

Passerelles

|  |  |  |  |
| --- | --- | --- | --- |
| **Uri** | **Description** | **Zone** | **Flux entrant/sortant** |
| web.lotoquebec.com | Passerelle de la zone publique (DMZ) destinée aux applications web | Publique | Entrant |
| mobile.lotoquebec.com | Passerelle de la zone publique (DMZ) destinée aux applications mobiles natives | Publique | Entrant |
| partenaire.lotoquebec.com | Passerelle de la zone publique (DMZ) destinée aux applications partenaires | Publique | Entrant |
| externe.loto-quebec.com | Passerelle de la zone privée externe (DMZ) | Privée | Sortant |
| passerelle.lotoquebec.com | Passerelle de la zone privée interne (NDMZ) | Privée | Entrant et sortant |

APIs

|  |  |
| --- | --- |
| **Uri** | **Description** |
| api.loto-quebec.com | API de domaine (peu importe le type : MSA ou SOR) |
| apiapp.loto-quebec.com | API applicative |
| apiscq.loto-quebec.com | API d’affaires spécifiquement SCQ |
| apisejq.loto-quebec.com | API d’affaires spécifiquement SEJQ |
| apilq.loto-quebec.com | API d’affaires spécifiquement LQ |
| soa.loto-quebec.com | API d’affaires corporative Loto-Québec |
| soe.loto-quebec.com | API d’entreprise Loto-Québec |
| bff.loto-quebec.com | API de canal, incluant les API bidirectionnels (Websockets, service workers, etc.) |
| eda.loto-quebec.com | API applicative événementielle (voir si requis. Est-ce qu’on ne pourrait pas utiliser apiapp.loto-quebec.com ?) |
| webapp.loto-quebec.com | Applications web |
| mobileapp.loto-quebec.com | Applications mobiles |

Références :

Conventions de nommage et meilleures pratiques de conception d’un API

* <https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design>
* <https://cloud.google.com/apis/design/>
* <https://restfulapi.net/resource-naming/>

## Exemple concret : identité d’un client (à partir de différent contexte – *binding context*)

Le tableau suivant présente différentes routes exposant l’information touchant l’identité d’un client Loto-Québec.

Chaque route expose l’accès à l’identité du client en fonction du modèle/contexte qui est propre à son domaine ou sous-domaine.

Les exemples décrits sont fictifs mais permettre néanmoins d’illustrer l’approche proposée.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **URL (à titre indicatif seulement ; sert à illustrer le propos)** | **Détail** | **Zone**  Privée  Publique  Privée externe | **API**  Domaine  Applicatif  Affaire  Canaux  Entreprise | **Type**  Façade  (Miniservice)  Microservice  Composite | **Sous-type**  REST  oData  GraphQL |
| **apiapp**.loto-quebec.com/slc/players | Domaine Clients  Sous-domaine = Bijou(SLC)  Exemple : SLC.API | Privée | Domaine | Façade | REST |
| **apiapp**.loto-quebec.com/sijt/cli/players/ | Domaine Clients  Sous-domaine = SIJT.CLI  Exemple : SIJT.CLI.API  Note :  SIJT possède 3 sous-domaines : TGM, SJT, CLI.  SJT est un sous-domaine qui pourrait être lié au domaine Jeux. Alors que CLI est plus lié au domaine Clients. | Privée | Domaine | Façade | REST |
| **apiapp**.loto-quebec.com/openbet/clients/account | Domaine Jeux  Sous-domaine = OpenBet  Exemple : OpenBet.Client.Compte.API (LQAccount)  Du fait que l’on inscrit de la logique d’affaire propre à OpenBet dans l’API, il devient service de domaine au lieu d’être un service applicatif. | Privée | Domaine | Façade | REST |
| **apiapp**.loto-quebec.com/rcu/ppi/ | Domaine Clients  Sous-domaine = RCU  Exemple : RCU.Clients.PPI.API  Service applicatif qui se nourrit des API de facade développées par Blueway pour le RCU.  Puisque le RCU est un système fournisseur, l’API de domaine est celui développé par le fournisseur en façade au RCU.  L’API que LQ développe en façade au RCU permet d’assurer un découplage entre le RCU et les intégrations internes de LQ. Ce qui est conforme à la stratégie d’intégration (section 4.7.2, note G10). Cet API ne contient normalement pas de logique d’affaires.    Même si l’information manipulée est normalisée dans le modèle canonique d’un client, cela n’en fait pas nécessairement un service d’affaire pour autant du fait que le service ne fait pas d’orchestration avec d’autres services applicatifs ou de domaine. | Privée | Domaine | Façade | REST |
| **apiapp**.loto-quebec.com/clients/profil/identite | Domaine = Clients  Sous-domaine N1 = Profil  Sous-domaine N2 = Identité  Exemple : Clients.Profil.Identite.API  Cet exemple suppose un véritable microservice de domaine dont le seul mandat est la gestion de l’identité du profil d’un client Loto-Québec. Donc, que sa persistance n’est pas nécessairement celle du RCU (idéalement) mais une persistance exclusive au microservice.  Un microservice qui pourrait recevoir toute activité touchant l’identité du client via une architecture EDA (ex. MOM) et pourrait diffuser la mise à jour par principe de cohésion éventuelle (*eventual consistency*) vers les autres systèmes qui conservent l’identité du client (en fonction de leur sous-domaine respectif). Ces systèmes s’abonnent alors à l’événement via un service applicatif qui normalise l’information à leur sous-domaine et met ainsi à jour l’identité du client dans le système en question. Le RCU étant typiquement l’un de ces systèmes. | Privée | Domaine | Microservice | REST |
| **soa**.loto-quebec.com/clients/profil | Domaine = Clients  Sous-domaine N1 = Profil  Service d’affaires qui retourne l’information normalisée dans le modèle canonique client (ex. FournirProfilClient). Fait la composition de plusieurs informations liées au client (partie prenante individu, adresses, téléphones, segments, activités, programmes, informations de sites, etc.).  Peut faire usage des microservices du sous-domaine Client N2 ou plus (en dessous du sous-domaine Clients.Profil) ou encore faire appel au service de façade (miniservice) du RCU. | Privée | Affaires | Composite | REST |
| **partenaire**.lotoquebec.com/dialog-insight/clients/profil | Exposition externe d’une API privée. Passage par l’API Gateway public. Réécriture SSL. Redirection vers l’URL interne correspondante (ex. via service de découverte).  Authentification avec protocole autorisé (ex. SAML, OAuth ou OIDC), clé secrète et/ou filtrage IP.  Combinée à un flux OAuth-OBO ou OAuth-CCF au moment de la réécriture SSL et redirection.  Exemple fictif :  Un système comme Dialog Insight qui appellerait un service d’affaires pour mettre à jour (dans le RCU et autres systèmes par orchestration) le profil d’un client suite à un exercice de segmentation…  Redirection typique vers  **soe**.loto-quebec.com/dialog-insight/clients/profil  ou  **soa**.loto-quebec.com/clients/profil | Publique | Entreprise  Ou  Affaires | Composite | REST |
| **partenaire**.lotoquebec.com/openbet/clients/account/payment | Idem.  Exemple fictif de Paysafe qui appellerait l’API de OpenBet dans la zone publique JHA-DMZ de LQ.  Redirection typique vers  **apiapp**.loto-quebec.com/openbet/clients/account/payment  ou  **soe**.loto-quebec.com/openbet/clients/account/payment  Voir cohérence avec Louis-Julien | Publique | Entreprise  Ou  Applicatif | Façade | REST |
| **apiapp**.loto-quebec.com/clients/profil/identite/graph/ | Rend possible l’utilisation des mécanismes de requêtes « GraphQL » pour le service en question | Privée | Domaine | Microservice | Graphe |
| **apiapp**.loto-quebec.com/clients/profil/identite/odata/ | Rend possible l’utilisation des mécanismes de requêtes « oData » pour le service en question | Privée | Domaine | Microservice | oData |
| **eda**.loto-quebec.com/clients/profil/identite  ou encore  apiapp.loto-quebec.com/client/profil/identite/updateevent | Permet d’émettre un message vers une médiation (ex. MOM) lié à un événement s’étant déroulé sur l’identité du profil d’un client LQ. | Privée | Applicatif | Façade | REST |