



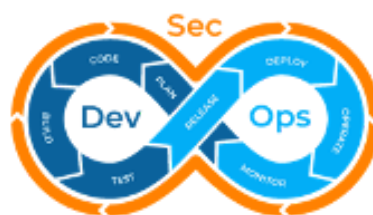
MINISTÈRE
DE L'INTÉRIEUR
ET DES OUTRE-MER

*Liberté
Égalité
Fraternité*

Cadre de Cohérence Technique (CCT)

Volet : *Cloud (IT)* Native

à portée interministérielle



Version : alpha.v4

Date : 27/03/2023

Auteur : Direction du numérique MIOM

A propos de l'appel à commentaire de ce document.

Ce même document est disponible sur le repository GitHub de la direction du numérique du MI :

<https://github.com/dnum-mi/CCT-Cloud-Native/>

Vous pouvez remonter vos commentaires et suggestions sur ce document de plusieurs manières:

- 1/ utiliser le fichier de relecture proposé [Fichier pour commentaires:](#)
 - <https://github.com/dnum-mi/CCT-Cloud-Native/blob/main/gabarit-pour-commentaires.ods>
 - et l'envoyer à :
dnum-architecture-entreprise@interieur.gouv.fr
- 2/ déclencher des 'Pulls request' sur le repository Github
- 3/ enregistrer des issues dans github

TABLE DES MATIÈRES

1 - Guide d'utilisation rapide	5
2 - Le contexte, les enjeux, la vision	6
3 - Principes généraux cadre Cloud Native	9
Les configurations d'hébergement prises en compte	10
Gestion des non-conformités, dérogations et contribution	11
Le modèle organisationnel, de responsabilité et de collaboration Cloud Native	11
Préconisations générales d'architecture et technique	16
Des spécificités à prendre en compte sur la création des conteneurs	17
Des spécificités à prendre en compte sur la topologie réseau et les ouvertures de flux	17
Des spécificités à prendre en compte autour de la qualité et de la sécurité des applications	18
Modèle d'intégration d'une application dans le cadre Cloud Native	18
4 - Présentation de l'offre interMinistérielle Cloud Pi Native et de ses évolutions pressenties	20
Les magasins de composants kubernetes et d'image de base	22
5 - Référentiel d'exigences et modalités d'usage	22
6 - Annexes	24
Les normes industrielles, institutionnelles applicables	24
Liens vers autres contenus utiles(informatif)	25
Glossaire	26
Référentiel d'exigences applicables au CCT Cloud Pi Native	34

version préliminaire pour appel à commentaire

--- page vide ---

1 - Guide d'utilisation rapide

Le présent volet, du cadre de cohérence technique Cloud Pi Native, s'adresse aux développeurs, architectes et en général aux acteurs gérant l'élaboration et le maintien de systèmes numériques. Ce cadre décline la doctrine cloud au centre de l'Etat promouvant le cloud en premier "cloud first", le *mode produit* avec un fonctionnement en équipe intégrée et l'agilité. Le respect de ce cadre permet à la direction d'application l'usage du socle de déploiement et de sécurité Cloud Pi Native.

Ce volet est en lien avec la nouvelle offre proposée par le ministère de l'intérieur Cloud Pi Native. Elle inclut des services mis en cohérence : un orchestrateur DevSecOps à 2 étapes permettant d'accélérer la production et l'homologation continue, une offre d'hébergement kubernetes sécurisée (sécable) et des charts / operators kubernetes communautaires.

Le lecteur pressé peut aller directement à la revue des exigences qui sont de 2 natures : **Primordiales**, obligatoires dont le non-respect peut entraîner une exclusion administrative dans le cadre d'un marché public et **Importantes** qui sont recommandées pour maximiser la performance de la conception et du contexte d'organisation avec l'offre. Les applications peuvent contribuer via un échange préalable, aux fonctionnalités de l'offre directement via un processus d'intégration en continu.

Ce cadre, ainsi que l'offre de services mis à disposition, permet de :

- guider les concepteurs d'applications afin d'optimiser les architecture produites selon les normes industrielles, les meilleures pratiques DevSecOps et Cloud tout en maintenant une capacité d'innovation des équipes ;
- diminuer la quantité de code produite réduisant ainsi les coûts et les délais ;
- réaliser et maintenir des applications avec une efficience des ressources consommées (financière, RH, énergétique) ;
- soutenir, en continu, la qualité et la sécurité des solutions ;
- normaliser et optimiser l'usage nominal de l'offre ;
- autonomiser les clients dans leur usage des services de l'offre ;
- maintenir un niveau de compatibilité minimal avec d'autres offres cloud ;
- soutenir l'accès à des ressources d'infrastructures utilisables en quelques jours ;
- fluidifier et sécuriser le déploiement en continu, soutenant l'agilité et permettant de réduire la distance entre le fonctionnel proposé et les attentes des usagers;
- mettre en place un modèle de responsabilité et de collaboration étendu ;
- disposer d'une trajectoire soutenable pour ceux en charge de maintenir l'offre.

Ce cadre se décline sur les configurations suivantes :

- l'hébergement d'applications sur les infrastructures internes infogérées par le ministère de l'intérieur ;
- l'hébergement d'applications sur des infrastructures cloud externes ;
- l'hébergement d'applications sur des infrastructures Ministérielles dédiées et gérées par l'application.

Le cadre de cohérence technique est découpé en 2 parties applicables

- ce document qui présente et explique le présent cadre et les spécificités de l'offre ;
- le référentiel d'exigences techniques, organisationnelles et administratives.

La direction d'application doit valider les conditions générales d'usage de l'offre (CGU) Cloud Pi Native lors de la souscription. Les CGUs reprennent les exigences qui ont été déclinées au sein de l'offre. Le lecteur est invité à vérifier qu'il dispose de la dernière version de ce document d'explication ainsi que de la liste d'exigences.

2 - Le contexte, les enjeux, la vision

Audience : ce paragraphe s'adresse à tout acteur considérant l'usage de l'offre Cloud Pi Native du ministère de l'intérieur, il présente les principes fondateurs. Le cloud : des nouvelles possibilités techniques, une collaboration étendue des acteurs pour répondre aux enjeux d'un contexte exigeant, incertain et accéléré.

À travers sa doctrine « Cloud au centre », l'État encourage l'ensemble des acteurs publics à se saisir de son potentiel afin de développer une nouvelle génération de services numériques de qualité et qui répond au besoin dans un cadre de normes juridiques adapté à l'usage. Cela répond notamment au besoin accru d'agilité et d'efficacité de l'administration.

Le Cloud est une approche d'accès à l'infrastructure d'hébergement à travers une interface standardisée et rendant abstraite et normalisée les technologies sous-jacentes. Cette abstraction permet une automatisation poussée, supprimer les frictions organisationnelles, les non-qualités et lenteurs induites par des opérations manuelles. L'infrastructure est pilotable par du logiciel et donc automatisable avec des processus reproductibles.

La technologie Cloud Native fait référence à l'usage de Kubernetes. Kubernetes est une technologie issue des travaux des grands acteurs de l'Internet il y a plus de 15 ans pour rendre encore plus efficace et sécurisée l'usage des infrastructures techniques, la résilience des hébergements et apporter une souplesse organisationnelle accrue. Les grands services de l'internet s'appuient sur cette technologie, elle permettent une résilience extrême et permet d'absorber un trafic extrêmement important.

Les architectures des applications se simplifient avec une abstraction de plus en plus grande de l'infrastructure avec notamment la montée en puissance de services managés, fonctions-as-a-service dans l'objectif recherché de diminuer la quantité de code produite et l'objectif est de réduire charge des équipes intégrées de développement, déchargé de nombreuse problématique de gestion de l'infrastructure de leur application.

L'ensemble des organisations ayant mis en œuvre cette technologie telle que EDF, Orange, des services de vente en ligne, des Banques, Airbus, Urssaf, etc... ont vu également leurs efficacités de l'usage du numérique augmenter, il y a un *avant et un après*.

Le ministère de l'intérieur, l'un des premiers acteurs étatiques à avoir proposé une offre Cloud il y a plus de 5 ans, étend son offre de service, en proposant l'offre Cloud Pi Native combinant une offre d'hébergement kubernetes sécurisée jusqu'au niveau DR. Cette offre est accompagnée d'un modèle DevSecOps outillé permettant une fluidité organisationnelle accrue et un renforcement de la qualité des solutions numériques.

Les approches cloud, devops et l'agilité ont progressivement permis de concilier des postures antagonistes : les développeurs ayant besoin de pouvoir déployer fréquemment, et l'exploitation ayant au contraire besoin de stabilité et de diminuer les risques liés au changement. La clé réside dans une collaboration étendue de tous les acteurs en prenant compte de la sécurité à toutes les étapes : le devops.

Une évolution des pratiques pour un numérique efficient et éco-responsable et réactif

Les contraintes s'accroissent sur la production de services numériques, le standard de qualité général a augmenté massivement avec les acteurs du net et industrielle qui produisent des solutions ergonomiques, sécurisées qui montent à l'échelle facilement. Un fossé important s'est creusé entre l'efficacité du numérique 'legacy' et ce monde moderne.

L'environnement change rapidement, il devient incertain, il faut réagir de plus en plus rapidement, la pression augmente sur la production de solutions numériques ergonomique, nécessitant des cycles raccourcis, la prise en compte de l'éco responsabilité, de l'accessibilité et le maintien d'un haut standard de qualité et de sécurisation.

Seul un changement majeur de pratiques s'appuyant sur l'opportunité du Cloud Native (kubernetes) et du DevSecOps permettent de satisfaire ce nouveau standard d'exigences.

Le mode produit et l'agilité sont indispensables en complément de l'utilisation du cloud, extrait de la doctrine cloud au centre:

“ L'adoption du cloud doit s'accompagner de celle des pratiques associées à l'excellence dans la production de services numériques (proximité entre métiers et équipes informatiques, scalabilité, agilité, « devops », « continuous delivery » qui sont les garants de l'adaptation des produits à leurs utilisateurs) ; “

Les principales caractéristiques du modèle opérationnel Cloud Native:

Le fonctionnement évolue vers la suppression de la fragmentation des responsabilités dans la chaîne de production et l'automatisation de celui-ci. (cf mode2 proposé par le Gartner)

L'équipe projet intégrée (équipe produit) voit ses prérogatives étendues:

- Elle est organisée autour du produit numérique livré. Elle fonctionne en modalité intégrée et de manière autonome en lien avec la vision et les contraintes fixées.
- Elle est composée de développeurs, architectes, ergonomes (ux-design), juriste, gestionnaire du changement, etc... orchestrée par le product owner et généralement facilitée par un coach / scrum master agile. Elle est focalisée sur l'ergonomie, la qualité et la performance de la solution mise à disposition des usagers. C'est le modèle «You build it, you run it, you support it ». (vous l'avez construit vous l'opérez)
- Une équipe d'aide appelée “service team” qui peut être temporaire ou permanente selon la taille de produit est chargée de mettre en place l'automatisation et les environnements de travail et de production. Cette dernière doit être aguerrie à ces technologies et l'offre cloud native.

L'hébergeur assure quant à lui, la mise à disposition d'une offre de service hautement disponible et sécurisée. L'usage de l'offre est réalisé via une console, une interface technique normée (API), une documentation et des exemples accélèrent la prise en main.

Il assure également la fourniture de services d'abstraction tels que des services managés ou fonctions as a service, un pipeline devsecops etc...

L'équipe intégrée est autonome et travaille sans échange avec l'hébergeur. La console mise à disposition assure ce lien avec une documentation associée. L'ensemble des opérations réalisées manuellement auparavant lors des étapes d'élaboration sont complètement automatisées. Le code logiciel est testé en permanence et automatiquement par un outillage : l'orchestrateur DevSecOps.

Pour assurer la qualité du code, plusieurs principes sont mis en œuvre, une couverture de test (100% sur le back-end), l'analyse statique du code, l'analyse réursive de vulnérabilité des composants importés (bibliothèque, images), des tests de sécurité et de performances, tests spécifiques liés à des besoins de vérifications particulières.

L'équipe de développement est prévenue au plus tôt par l'orchestrateur DevSecOps en cas de non-qualité. L'orchestrateur offre des options techniques pour intégrer les retours dans le flux de travail du développeur afin de procéder à la correction au plus tôt des anomalies, cette modalité s'appelle "shift-left". Infine, seul un logiciel ayant atteint le niveau de qualité fixé peut être déployé.

Les développeurs assurent continuellement la qualité et la sécurité du code produit, la non-régression et l'absence de vulnérabilité soutenus par l'outillage mis en place, la chaîne devsecops, les environnement de codage (IDE) et gestionnaire de code deviennent de plus en plus performant et identifient dès l'écriture ou l'intégration du code les erreurs d'algorithmies ou si un secret est laissé dans le code.

Les architectures sont modulaires et les composants internes ainsi que les interfaces entre applications sont découplées, c'est à dire rendus indépendants techniquement et organisationnellement via des interfaces normées (API). Cela permet la maintenance et les évolutions indépendantes entre composants. Ce découplage contribue fortement à réduire le coût des changements et faciliter les transitions technologiques en cas d'obsolescence.

La conception de l'architecture, le choix des langages sont faits pour une efficience de l'usage des ressources d'infrastructures et également sur l'impact sur le poste de travail. La conteneurisation, l'élasticité dynamique de la solution d'orchestration de conteneurs kubernetes, la mutualisation des infrastructures physiques soutiennent fortement cette efficience. Le non-respect de ce principe de modularité a été identifié par la direction interministérielle du numérique comme l'une des causes d'échec des grands programmes de l'État. La modularité, l'indépendance technique et organisationnelle des modules lors du déploiement, permettent de réduire la taille des déploiements ce qui contribue à réduire les risques et fluidifier les mises en production. Les déploiements peuvent ainsi être rendus transparents pour des usagers et les retours arrière éventuels sont facilités. Il est ainsi possible de déployer en confiance, plusieurs évolutions par jour si nécessaire.

L'automatisation permet de mieux contrôler et rendre les actions prédictibles, faciliter la reprise en cas d'incident et minimiser les coûts de maintien en conditions opérationnelles et de sécurité des applications. Plus aucune intervention manuelle n'est effectuée sur les environnements ce qui permet de réduire la variance, les non-qualités, et aussi de (re)construire très rapidement des plateformes.

In fine, la conception doit s'inscrire dans une démarche d'éco-conception et de sobriété numérique des conceptions (green IT) permettant un usage plus efficient des ressources qu'elles soient RH, financières. L'État devant être exemplaire. cf guide d'éco-conception.

3 - Principes généraux cadre Cloud Native

Audience : ce paragraphe s'adresse à la communauté des concepteurs et architectes solutions, le lecteur est réputé compétent et formé sur les sujets abordés .

Le cadre de cohérence technique régule et normalise les différents domaines associés à l'élaboration et au maintien des ressources partagées nécessaires à la mise à disposition de solutions numériques de qualité répondant au besoin. Il s'assure que l'ensemble peut-être mis en œuvre de manière cohérente avec une consommation minimisée des ressources : financière, RH et écologique. Il recommande ou fixe les mesures permettant d'atteindre l'objectif, tout en favorisant l'innovation, la prise en compte de l'obsolescence régulière des technologies et la manœuvre RH nécessaire (formation continue, recrutement ...)

Le volet Cloud Native du ministère de l'intérieur, hérite de normes industrielles, interMinistérielles, européennes. La portée est interMinistérielle, ce document a fait l'objet d'échanges avec la direction interMinistérielle du numérique et des ministères primo-accédants.

Pour le ministère de l'intérieur, il encadre la conception et l'hébergement d'applications qu'elles soient hébergées dans les datacenters du ministère ou bien à l'externe sur des clouds publics. Ce volet s'applique notamment lors de la conception d'une nouvelle application ou une évolution significative d'une application existante. (cf Doctrine Cloud au centre)

Pour les autres entités étatiques, ce volet sert de présentation de bonnes pratiques et présente les conditions générales d'utilisation de l'offre Cloud Pi Native.

Ce document ainsi que le référentiel d'exigences sont annexés aux dossiers de consultation des entreprises. La portée administrative étant précisée par le règlement de consultation (RC) du marché lui-même, typiquement le RC et le Cadre des clauses administratives particulières (CCAP) peuvent exclure administrativement un candidat en cas de non-respect des exigences P primordiales. (hors dérogation soumise, instruite et accordée).

D'autres référentiels d'exigences ou des guides peuvent être applicables ou conseillés. voir plus loin le chapitre sur les cadres de normes supérieures.

Offre de service Cloud (π) Native :

Concerne la description de l'offre de service managé d'infrastructure Cloud π et d'une chaîne DevSecOps assurant l'homologation en continu et le déploiement en production. Cf. présentation de l'offre plus loin dans ce document. Cette offre évoluera selon les demandes et financement disponible vers services d'abstraction de l'infrastructure, tel que service managés, fonction, etc...

Poste de travail agent :

Dans le cadre d'une application rendue accessible sur le poste de travail de l'agent, le lecteur est invité à se conformer également au volet *Environnement Numérique de Travail*, notamment sur les aspects d'intégration au SSO et la politique des navigateurs.

Ouverture des données :

Sur la thématique de l'ouverture et de la circulation de la donnée, le projet est invité à se mettre en conformité avec le volet idoine. Cela concerne notamment le référencement des objets métiers dans le référentiel de cartographie des données et la mise à disposition d'une facilité technique d'accès à la donnée basée sur un standard d'échange de type API. L'ensemble des acteurs de l'État est invité à faire circuler la donnée au profit d'une simplification du fonctionnement des administrations et d'un service public ergonomique et proactif . (cf rapport Bothorel, lois CRPA et 3DS, ...)

Périmètres du CCT et configurations prises en compte

Ce CCT concerne l'usage de l'offre Cloud Pi Native dans les configurations précisées ci-dessous.

Le ministère de l'intérieur dispose de plusieurs capacités d'hébergement d'application. Ces offres peuvent être historiques et liées à une entité (ex: Sgami, ANTS) ou centrales.

Les offres centrales sont découpées en plusieurs catégories :

- **Physique ou virtualisées** généralement de type VMWare tel qu'Isocèle (DNUM), STIG (STSI2). Cette offre est accessible qu'au MIOM et seul l'exploitant gère la plateforme et les actes d'intervention techniques via Ticketing ITMS
- **Offre Cloud Pi¹ 'legacy'** : offre IaaS basée sur OpenSack, actuellement en bascule vers Gen2. La gestion interne des 'tenants' est à la main du développeur, le reste via ticketing. Le développeur gère son outillage en autonomie
- **Offre Cloud Pi Native** (objet de ce volet de CCT) : nouvelle offre associant une homologation continue, un socle d'homologation et un hébergement étatique sur kubernetes avec une ouverture 'accéléré' des flux réseau.

Les configurations suivantes sont prises en compte par ce volet Cloud (Pi) Native du CCT.

- Hébergement sur les clusters kubernetes managés par le ministère de l'Intérieur, jusqu'au niveau « donnée restreinte » ;
- Hébergement sur un cluster kubernetes externe au ministère, compatible avec la sensibilité des données manipulées ;
- Hébergement sur un cluster kubernetes dédié et géré par l'application;
- Une approche hybride multi-clusters (plusieurs environnements de production on prem ou cloud public).

Pour l'ensemble de ces configurations l'usage de la chaîne DevSecOps managée par le Ministère de l'Intérieur est impératif. (hors cadre dérogatoire accordée)

¹ PI : produit de l'Intérieur

Gestion des non-conformités, dérogations et contribution

L'évolution rapide des technologies cloud peut conduire à ce que le cadre CCT restreigne l'innovation. Il est également souhaité, pour éprouver le modèle, de notifier le département architecture d'entreprise du Ministère de l'intérieur au plus tôt des éventuelles impossibilités ou limitations remarquées. Les directions d'applications ou les organisations utilisatrices peuvent contribuer, via un échange préalable, à enrichir les fonctionnalités de l'offre ou du cadre lui-même. Sur l'offre la contribution est effectuée directement sur le repository open source de la solution via un pull request.

En cas de non-conformité au CCT ou absence de contribution à l'offre, une demande de dérogation dûment motivée sera formulée à l'avance par la direction d'application. Seule la notification de la décision permet d'amender le besoin de conformité au cadre, temporairement ou de manière pérenne dans le cadre d'une dérogation. Dans le cadre d'une dérogation, la direction d'application prend à sa charge le surcoût complet de possession. (formation, homologation, personnel assurant la tme, etc...)

Lors de l'utilisation du cadre et de l'offre Cloud PI Native, toute organisation souhaitant décliner ce cadre dans un document de norme inférieur pour un besoin propre est invitée à référencer la dernière version de ce document en l'état. Dans la hiérarchie des normes, une instruction de niveau inférieur ne peut entrer en conflit ou contredire ce présent document.

Le modèle organisationnel, de responsabilité et de collaboration Cloud Native

L'architecture, le modèle de responsabilité et d'organisation à mettre en place est orienté pour maximiser la qualité, la sécurité, la fluidité opérationnelle et l'évolutivité du produit en tirant parti au maximum des possibilités offertes par la technologie kubernetes, un flux de production DevSecOps et une collaboration étendue entre les acteurs.

L'élargissement de la responsabilité du développeur

La responsabilité du développeur est élargie dans le cadre Cloud Native. Il élabore et exploite une solution qui répond au besoin métier généralement une automatisation d'un ou plusieurs processus métiers . Le développeur s'assure de la qualité et de la disponibilité du service rendu à l'utilisateur selon le précepte : « You build it, you run it ». Il s'organise en équipe intégrée, si nécessaire avec de l'externalisation.

Le développeur met à disposition d'un point de vérité du code sous la forme d'un ou plusieurs dépôts de code logiciel fonctionnel et d'infrastructure. Il met en place un flux intégré et continu de production en s'appuyant sur un orchestrateur primaire DevSecOps qu'il construit et opère.

Le développeur initialise et supervise le pipeline de l'orchestrateur secondaire opéré par le ministère de l'Intérieur (cf plus loin sur la configuration de l'offre Cloud Pi Native). Il intègre les étapes de vérification de sécurité génériques imposées par le ministère et spécifiques issus de la démarche d'homologation.

L'ensemble combiné des orchestrateurs primaire et secondaire assure la fonction d'homologation et de déploiement en continu du produit numérique.

Dans le cas de la détection d'une non-qualité, la progression du déploiement est stoppée par la/les chaînes, le développeur est prévenu en temps réel et doit corriger au plus tôt les défauts remontés. Cette approche permet de garantir un niveau de qualité, évite des régressions et maintient la dette technique au niveau le plus bas.

Sur le plan organisationnel le développeur met généralement en place :

- l'agilité avec des itérations courtes de constructions et de vérification des usagers ;
- le découpage des livraisons en lot de taille de réduite ;
- la mise en place d'une culture de collaboration étendue et des pratiques intégrant la sécurité à toutes les étapes.

La répartition des responsabilités s'établit de la manière suivante :

L'équipe produit intégrée :

- est responsable de l'application, de la qualité du code et du bon fonctionnement de l'application pendant l'ensemble du cycle de vie de l'application.
- est responsable de définir et d'ajuster l'infrastructure en s'appuyant sur l'élasticité du cloud.(sur la base de l'offre Cloud adaptée selon la sensibilité des données)
- s'appuie sur les patterns applicatives et services managés mis à disposition, les magasins de charts helms et des *operators* disponibles. Il est déconseillé, par exemple, de repackager une base de données alors qu'un *operator*, un chart ou un service managé est disponible. (simplification et systématisation du mcs)
- fourni un code de qualité exempt de défaut d'algorithmes, de qualité et de vulnérabilité ;
- met en œuvre le flux de production, logiciel local permettant d'assurer la production et la démonstration d'un code de qualité exempt d'anomalie fonctionnelle, de non-qualité et de vulnérabilité, notamment dans les librairies importées ;
- Il pose les normes de développement des langages utilisés ;
- Il met en place des pratiques DevSecOps visant un maintien de la qualité dans le temps avec les composantes suivantes (cf outillage DevSecOps) :
 - test driven development ;
 - une couverture de test unitaire à 100% du back-end ;
 - une couverture significative des tests du front de l'application ;
 - analyse statique de qualité du code ;
 - analyse récursive des vulnérabilités des librairies importées ;
 - utilisation exclusivement d'images sources maintenues en condition de sécurité et certifiées (distribution LTS) ;
 - la conception des tests d'intégration en sandbox ;
 - la fourniture des outils nécessaires à la remontée de l'état de santé des briques applicatives destinées à fonctionner en production (healthcheck) ;
 - la fourniture des indicateurs nécessaires au suivi en temps réel de la qualité en condition opérationnelle de sa solution (exports prometheus) ;
 - l'exploitation des logs remontés.
- Il met en place un hébergement sur une plateforme kubernetes afin d'assurer la démonstration du bon fonctionnement de l'application avec la solution qu'il préfère soit internalisée (avec un moyen de mener des démonstrations) ou sur cloud public.
- Il met en œuvre l'intégration technique et organisationnelle avec la chaîne DevSecOps de l'offre Cloud Pi Native et initialise le flux logiciel global (cf plus bas).

- Il maintient un point de vérité du code logiciel ainsi que celui du code d'infrastructure. Celui-ci est accédé par la chaîne DevSecOps étatique, la sécurisation d'accès issus par token.
- Il est responsable de la surveillance de l'ensemble des pipelines, y compris pour celui géré côté ministère.
- Il met en place une intégration du flux de retour d'anomalie "shift-left" des orchestrateurs afin de permettre une correction au plus tôt des anomalies.
- Il effectue l'apprentissage comportemental du firewall applicatif Web (WAF) vis-à-vis de l'application dans le cadre fixé par le ministère. (anticipation avant la mise en production)
- Il est invité à mettre en œuvre ce pipeline au plus tôt dans le processus de réalisation.

Notes : (cf exigences CCT)

L'équipe de développement respecte les règles suivantes permettant une qualité de code en progression et un maintien de la sécurité :

- minimise la portion spécifique de code développés en s'appuyant sur le catalogue des services proposés. (revoir régulièrement)
- met en place une couverture de test unitaire complète du back-end (et fourni les moyens de vérification automatisé à la chaîne secondaire)
- mener une analyse de code systématique le plus tôt possible (les langage et IDE modernes fournissent des fonctions de ce type)
- mener une analyse de CVE des dépendants importées

L'équipe projet met en œuvre une activité continue de refactoring du code produit. La qualité du code ne peut être décroissante.

Elle fournit les preuves que des tests de sécurité, de qualité, de robustesse des algorithmes ont été mis en œuvre, et qu'ils n'ont pas remonté de vulnérabilités ou d'erreurs majeures. En s'appuyant notamment sur les logs des analyses des outils de la chaîne primaire. Elle fournit la preuve (ex: le document) des normes de développement et pratiques permettant de maîtriser la qualité du code produit. (refactoring, peer review, etc..)

L'exploitant Ministériel de l'orchestrateur DevSecOps :

- Il assure de la disponibilité et de la qualité de fonctionnement de la chaîne DevSecOps et maintient à jour la documentation sur le fonctionnement des interfaces et assure les évolutions fonctionnelles ;
- Il assure les retours d'anomalies "shift-left" lors des opérations de déploiement continu. L'intégration et leurs traitements sont à la charge du Concepteur / développeur ;
- Il contribue à la mise en place et l'évolution du catalogue d'operator Kubernetes, de charts helm et du référentiel de pattern de référence;
- Il est également en lien avec les autorités d'homologation afin de s'assurer que l'ensemble est en condition d'assurer la protection d'ensemble;
- Il intègre les propositions d'évolution "pull request" proposée en fonction de son plan de charge et d'une négociation préalable.
- Il fournit la documentation dont les éléments d'orientation ou d'aide à la migration permettant au développeur de s'orienter vers l'opérateur cloud qui hébergera son produit. (cela dépend également des décisions projets ou d'homologation)

L'opérateur Cloud :

Il assure le maintien en condition de disponibilité et de sécurité de l'offre d'hébergement, l'interface API de management, la console, la gestion capacité et les offres de services managées.

Le Cloud PI est constitué de plusieurs régions (elle même constituées de plusieurs centres de calculs), zones de sensibilité usuel et DR.

Chacune des régions peuvent faire l'objet d'une affinité ou restriction de services, sur la nature des données, des typologies de projets, ministériels et interministériel, chaîne de SecOps secondaire; etc...

La gouvernance du produit 'Cloud Pi Native' à portée interministérielle et ministérielle définit la politique d'ensemble, cette dernière traite d'aspects tel que : restriction ou affinité d'accès aux régions, projets, chaîne secondaire ;

- politique de données définie dans l'extension de la définition de "données restreintes" ;
- la capacité en matière de ressources nécessaires à héberger de nouveaux projets ;
- la localisation géographique et le nombre de centres de calculs constituant sa région ;
- les services mis à disposition ;
- les possibilités d'interconnexion réseau avec d'autres systèmes;

La version courante de l'offre et la politique d'usage est mise à disposition sur portail et notamment vers l'utilisateur lors de la connexion à la console. La version courante est présentée dans les grandes lignes au paragraphe 4.

Des pratiques complémentaires sont introduites dans la configuration Cloud Native :

Le "GitOps", contraction de git et opération, est indispensable à la gestion des applications Cloud Native avec Kubernetes. Ce mode d'organisation du code d'infrastructure permet de maîtriser la description de l'infrastructure de production avec les mêmes pratiques de revue collaborative que celle du logiciel. Il est par exemple strictement interdit de faire des modifications «à la main » sur l'environnement de production, toute variation est supprimée, l'infrastructure réelle est strictement celle décrite par les fichiers d'infrastructure.

Le **"shift-left"** (vers la gauche, du processus) fait référence à la remontée le plus tôt possible vers le développeur des anomalies identifiées par la chaîne de déploiement et de vérification DevSecOps. Ce flux est notamment mis en œuvre depuis la chaîne secondaire.

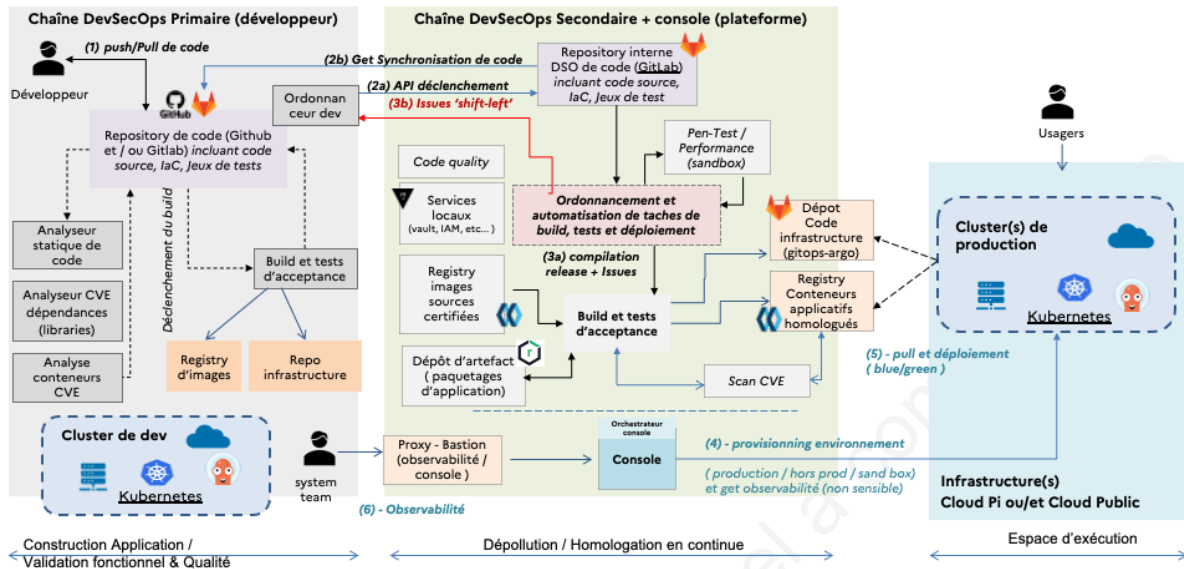
Présentation du cycle d'usage de l'offre pour les directions d'applications:

Phase d'initialisation du projet

- Le développeur initialise l'environnement de développement, il est autonome pour les choix techniques, il respecte les exigences organisationnels et de processus automatisé permettant de maintenir une qualité constante ;
- Le développeur décide de l'infrastructure d'hébergement en fonction des contraintes sur les données et la liste des options autorisées et maintenue par la Dinum et l'ANSSI en lien avec la doctrine Cloud au centre, notamment : Cloud Pi, cloud externe de confiance ou plateforme dédiée (si besoin spécifique)
- Le développeur commande, (signature de convention), initialise l'espace projet au ministère et configure selon son choix d'infrastructure les environnements désirés. Il récupère les clés techniques nécessaires à l'intégration des pipelines.

- Le développeur effectue l'intégration des pipelines, cf labels (2) , et (4) si l'infrastructure est externe.
- Il vérifie que l'ensemble du pipeline est opérationnel à partir d'un code d'exemple fourni de type "hello word".

Principe de fonctionnement du pipeline d'ensemble (chaînes primaire et secondaire)



- **[1]** Le code logiciel ainsi que celui de description des infrastructures sont produits au sein de l'espace du développeur/concepteur, généralement en externe au Ministère de l'Intérieur.
- **[2a] [2b]** Une interface bi-directionnelle entre l'espace du concepteur / développeur et celui de DSO permet en push-pull à la chaîne secondaire de récupérer automatiquement l'ensemble du code et des dépendances nécessaires.
- Le développeur déclenche par appel API les services de synchro / build / deploy. (il n'y accède pas directement sauf via un Bastion)
- **[3a][3b]** La chaîne d'orchestration DevSecOps effectue la récupération du code, des tests de qualité du code, scan de vulnérabilité des dépendances, la reconstruction, les tests de nos régressions, des tests d'homologation, vérification des manifests / charts etc... au regard des politiques de sécurité et dépose les images certifiées sur la registry de la chaîne ainsi que le code d'infrastructure.
- Le concepteur/développeur accède à un retour d'information détaillée sur le succès ou sur les éventuels défauts, lors du build, deploy de l'application par DSO. (via message ou webhook)
- **[4]** : La console provisionne si nécessaires les environnements en 'poussant' les ressources nécessaires une fois (secrets, application.yaml (argo), certificats, etc...)
- **[5]** : L'infrastructure vérifie régulièrement les changements sur le dépôt d'infrastructure (ou déclenchement forcé par API) et synchronise l'environnement à la cible visé et opère une bascule blue-green transparente, cf ArgoCD (si échec l'environnement de prod reste inchangé)
- **[6]** : Le développeur accède à un proxy d'observation du fonctionnement de l'application
- Note : Les développeurs n'accèdent pas directement à la production. Seuls les administrateurs habilités peuvent y avoir accès via bastion.

Préconisations générales d'architecture et technique

Ce chapitre précise les aspects importants liés à l'usage de kubernetes dans le cadre du ministère de l'intérieur. Il est attendu que les acteurs soient correctement formés à la solution kubernetes et se maintiennent à jour. La technologie évoluant rapidement.

Le fondement des normes techniques est issu du cadre "Cloud Native", largement accepté et appliqué au sein de l'État et le secteur privé, tel que les "15 factors".

C'est le respect de ces normes qui permet à la fois d'adresser les enjeux de performance en termes de vitesse de livraison et de qualité de service, mais aussi de normaliser les applicatifs pour une meilleure évolutivité et maîtrise de la dette technique. Enfin, elles assurent une intégration fluide au sein des systèmes d'information Ministériels.

Un des principes cœurs est de laisser un certain degré de liberté au concepteur/développeur sur le fonctionnement interne de son application. Au contraire, les interactions avec les autres applications et services seront particulièrement contraintes.

Il est à noter qu'uniquement la plateforme d'orchestration de conteneurs Kubernetes est considérée dans le cadre de ce cadre de cohérence, celle-ci étant considérée comme l'état de l'art, et open-source de surcroît.

À propos de la solution mutualisée d'hébergement Cloud Native et de l'ergonomie pour le développeur

La philosophie générale est de balancer la réduction de la surface d'attaque cyber et l'ergonomie indispensable pour le développeur. La réponse définitive ne peut être que celle éprouvée en run, nous livrons ici une intention qui doit être rodée opérationnellement.

- Sur les **environnements côté "primaires"**, le développeur est (doit être) maître 'root,' il les maîtrise et les utilise pour son besoin de déboguer tester des choses, etc... l'administration n'intervient pas ici.
- Sur les éventuels **d'environnement hors production côté secondaire au MI**, l'environnement est dédié au développeur il dispose d'un accès root à sa plateforme selon les règles de la PSSI applicables. (voir plus loin)
- Sur **l'environnement de production côté MI**, les développeurs n'accèdent pas directement, cet accès s'effectue via une console dédiée et un flux "gitops". (via l'usage d'ArgoCD). L'accès à la production nécessite une accréditation renforcée et adaptée et un poste 'rouge' (niveau habilitation SD).
- Sur **l'environnement de production chez un hyperscaler public**, le développeur gère sa plateforme. Généralement, il met en place un modèle d'opération réduisant les accès à la plateforme de production aux seuls administrateurs habilités.

Notes :

A propos de l'accès à des ressources d'environnements d'exécution situé côté administration, la règle fixée par la P-SSI (politique SSI de l'état) s'applique, notamment :

L'accès hors production nécessite une accréditation / enquête et un poste de travail adapté et maîtrisé par l'administration que l'on puisse raccorder au réseau du MI. Pour l'accès distant, un poste "bastion" adapté, que l'on appellera ici "poste jaune", sera nécessaire. (cela s'appelle SPAN aujourd'hui).

Sur l'environnement de production l'approche est différente et plus contrainte, l'accès à la production est restreint au strict minimum et nécessitera une accréditation renforcée et adaptée et un poste 'rouge' (niveau habilitation SD).

Il est recommandé d'anticiper au plus les contraintes en activant rapidement l'environnement de production et de tester le plus tôt le déploiement en production pour que cette partie sujets soit vite maîtrisée par les équipes.

Nous réfléchissons à la mise en place d'un "proxy" en lecture seule (au sens une représentation distante) des éléments de vie du système tel que l'état des pipelines et l'observabilité au développeur situé côté primaire ainsi un flux d'échange et de collaboration automatisé, dans la limite où cela n'augmente pas la surface d'attaque.

Des tests de compatibilité avec d'autres solutions d'hébergement d'acteurs du cloud public sont menés, les premiers retours sont concluants.

Des spécificités à prendre en compte sur la création des conteneurs

Kubernetes impose une rigueur un peu plus élevée à l'initialisation que d'autres solutions.

Les pods (conteneurs) sont **obligatoirement rootless**, c'est à dire que le compte root n'est jamais utilisé pour faire fonctionner le service et ils utilisent uniquement des ports > 1024.

Important : le lancement de pod en 'root' est interdit et bloqué au lancement. Ce point n'est pas modifiable. Ceci est un point d'attention majeur, la quasi-totalité des conteneurs à disposition sur les plateformes de partage de conteneurs ne sont pas rootless.

Les pods doivent démarrer dans leur configuration cible sans état, ou de nécessiter un passage de paramètres de démarrage ou d'environnement.

Les pods doivent démarrer rapidement afin de permettre au mécanisme d'orchestration de fonctionner rapidement.

Les pods sont responsables de vérifier au lancement, si l'application est dans la condition initiale de 1er lancement, ou bien s'il faut initialiser ou modifier d'autres ressources telles qu'une base de données.

L'architecture de l'application, hors persistance de données, est conçue pour être complètement stateless, c'est-à-dire, sans aucune persistance de sessions, états et liens, les pods peuvent être basculés à la volée d'un nœud à l'autre sans préavis.

Des spécificités à prendre en compte sur la topologie réseau et les ouvertures de flux

L'organisation de réseau est segmenté par type de service porté par le flux. (flux usagers, interdatacenters, interapplicatifs). Les flux réseau sont ouverts en lecture des manifests kubernetes / helms notamment Ingress et egress.

Des spécificités à prendre en compte autour de la qualité et de la sécurité des applications

L'objectif d'ensemble est de s'assurer que le code produit est de qualité constante ou accrue, exempt de vulnérabilités algorithmiques ou importées néfastes.

Pour atteindre ces objectifs plusieurs mécanismes doivent être mis en place par l'équipe de développement intégrée :

- minimiser la portion spécifique de code développés en s'appuyant sur le catalogue de service proposé.
- mettre en place une couverture de test unitaire complète du back-end (et fournir les moyens de vérification automatisé à la chaîne secondaire)
- mener une analyse de code systématique le plus tôt possible (les langage et IDE modernes fournissent des fonctions de ce type)
- mener une analyse de CVE des dépendants importées

La chaîne secondaire reconstruit les images à partir des codes sources et procède aux mêmes tests avec des outils complémentaires. L'orchestration du pipeline secondaire est gérée par l'équipe et intègre les tests de vérification issue de la démarche d'homologation de l'application qui fixe les seuils de blocage de déploiement.

Les tests typiques consistent à vérifier la qualité du code (et la bonne couverture des tests), et le bon fonctionnement de l'application (non regression) et le scan de vulnérabilité.

L'équipe de développement reçoit via l'interface "shift left" une notification des rapports qui doit être intégrés au flux de travail pour correction.

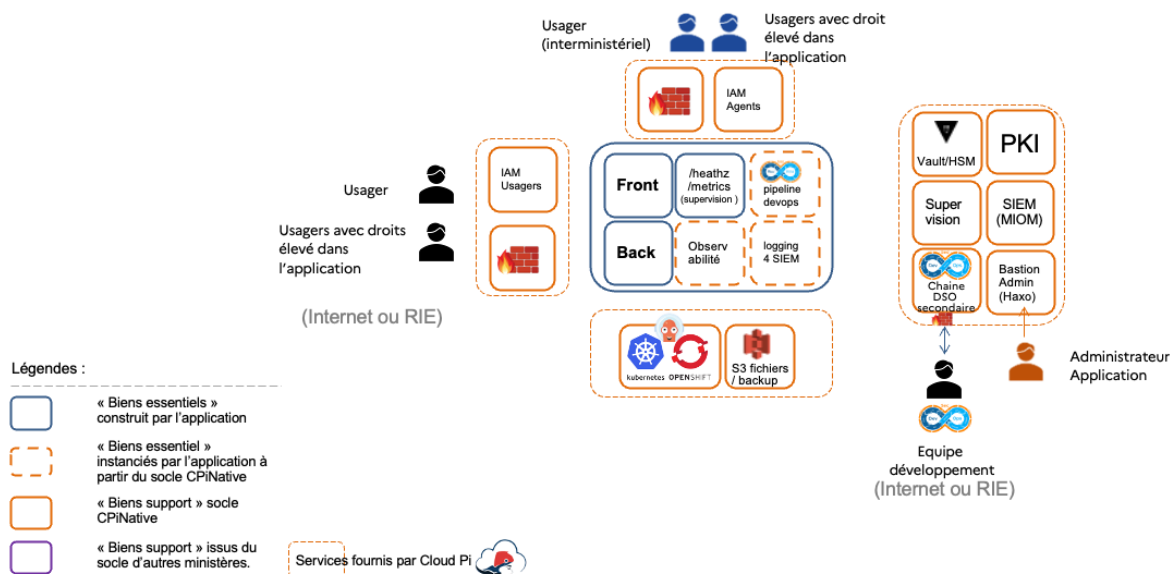
La chaîne secondaire bloque les déploiements si la qualité d'ensemble du code est en baisse ou si l'ensemble porte des vulnérabilités critiques.

L'équipe projet est invitée à mener une activité constante de refactoring du code produit.

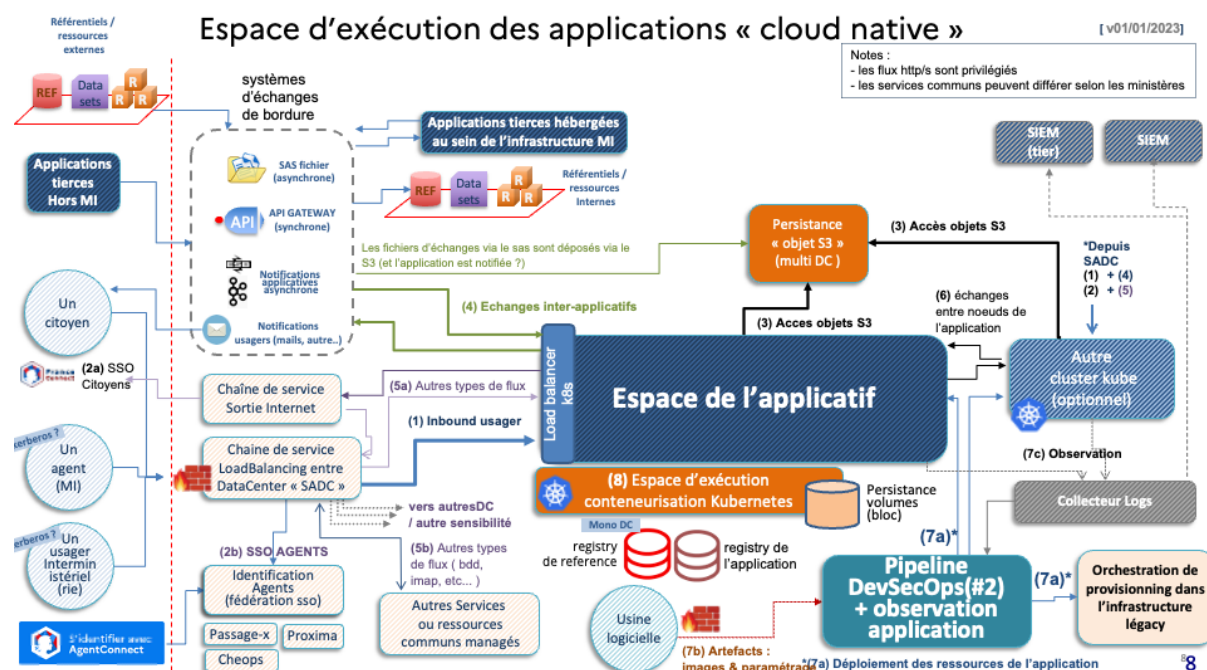
Modèle d'intégration d'une application dans le cadre Cloud Native

Le schéma ci-dessous précise le cadre général d'intégration d'une application. Des variantes sont possibles entre les ministères, elles sont précisées directement auprès des équipes concernées. Le respect de cadre permet à la direction d'application d'accéder à un socle de sécurité accélérant les homologations, l'ouverture automatique des segments réseau et l'homologation en continu.

Vision logique d'ensemble et services de socles



Architecture d'intégration réseau et flux typiques



- (1) Inbound usager : accès à l'application des usagers https / websockets (depuis RIE ou Internet)
- (2a) SSO Citoyens + (2b) SSO AGENT : authentification des usagers (OIDC / SAML V2)
- (3) Acces objets S3 : accès à la persistance objets de l'application
- (4) Echanges inter-applicatifs (bordure externe de l'application) : permet d'échange entre des applications de porteurs différentes, selon plusieurs modalités possibles : API restful synchrone, Asynchrone , fichiers
- (5a) Autres types de flux : autres types d'échange, sortie vers internet, vers d'autres zone d'hébergement, ou entre des zones de sensibilité différentes
- (5b) Flux d'accès à des services communs ou ressources managées (de protocoles plus variés)

- (6) échanges entre noeuds de l'application : permet la réplication de l'application entre 2 data centers au même niveau sensibilité de données
- (7a) Déploiement des ressources de l'application : gestionnaire & console DEVSECOPS / le pipeline interagit avec le/les clusters kubernetes et les gestionnaires d'infrastructures utilisés (ouverture de flux réseaux, etc...)
- (7b) Artefacts images & paramétrage : ensemble des ressources liées à une application ou communes (ex : sources d'images de référence)
- (7c) Observation : permet de collecter les données liées à l'usage pour la mise au point de l'application ou données de vie.
- (8) Kubernetes, sous la forme d'un ou plusieurs namespace(s) isolés ou couplés : fournis l'espace d'exécution de l'application et la gestion des volumes pour le stockage bloc.

4 - Présentation de l'offre interMinistérielle Cloud Pi Native et de ses évolutions pressenties

L'offre Cloud PI native répond aux exigences du CCT à travers un ensemble organisationnel et technique. Elle propose une offre Cloud régaliennne, souveraineté, sécurisée et isolée de toute problématique juridique extra-européenne.

La composition de l'offre est amenée à évoluer en termes de catalogue de service selon la demande et les financements disponibles, ces évolutions permettront la diminution de la quantité de code produit par les équipes de développement et l'accélération des performances, typiquement : fonctions as services, services managés, gpu;

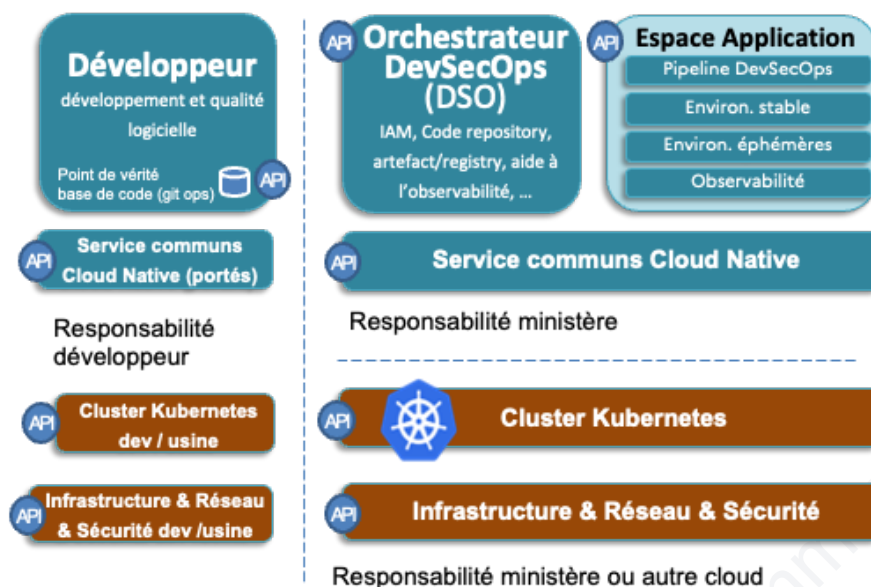
Les offres d'hébergement compatibles avec les applications « Cloud Native » du ministère de l'intérieur sont :

- Hébergement de l'application sur les infrastructures internes infogérées ;
- Hébergement de l'application sur des infrastructures cloud externes ;
- Hébergement de l'application sur des infrastructures gérées par l'application.

L'ensemble de l'administration technique de la plateforme et des infrastructures est automatisée, pilotée par le développeur/concepteur via l'orchestration DevSecOps avec mise en œuvre d'un principe dit *gitops*: la plateforme de production "tire" le déploiement.

Pour rappel, le développeur n'accède pas directement à l'environnement de production. Toute correction ou évolution suit le processus de déploiement automatisé passe via le principe "gitops" évoqué ci-dessus. Le développeur dispose d'un accès libre à ses environnements (via un poste bastion si l'environnement est situé côté ministériel) il dispose également d'un accès proxifié aux indicateurs de la production. La service team qui l'accompagne, avec les accréditations nécessaires, dispose quant à elle d'un poste dédié lui permettant d'accéder directement via un bastion aux services d'observabilité et faciliter le débogage en production. (note modalité en cours d'évaluation incrémentale)

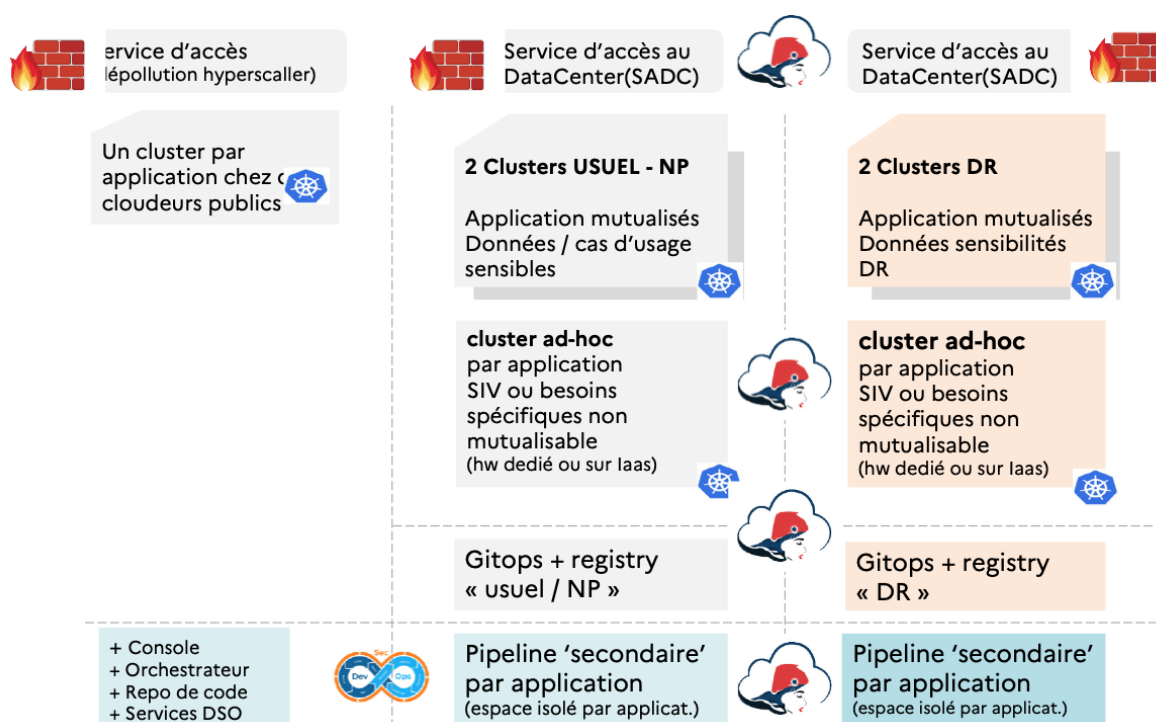
Le modèle de responsabilité est présenté ci-dessous:



L'ensemble du code source de l'offre Cloud PI Native et sa documentation sont disponibles en open-source sous la licence Apache2. Toute contribution est bienvenue.

La politique de segmentation d'hébergement est présentée ci-dessous :

Focus : Politique de segmentation de l'hébergement



Note : chaque région est autonome dans son fonctionnement. Seul le service de stockage objet de type S3 est accessible sur l'ensemble des régions ministérielles. (réplication en proximité dans le datacenter).

Les magasins de composants kubernetes et d'image de base

Associée avec l'offre Cloud pi Native, des magasins de composants kubernetes sont mis à disposition incrémentalement selon les besoins des applications cela inclut celui de l'éditeur RedHat. Le développeur peut dès aujourd'hui s'appuyer sur un catalogue porté par l'Insee autour de son produit Onyxia : <https://github.com/InseeFrLab/helm-charts>

L'équipe produit est fortement invitée à l'utiliser ces composants courants et ne pas refabriquer une version dédiée dont le cycle ne sera pas en adéquation avec les besoins de réactivité en mcs.

A propos des images de base nues, il est recommandé d'utiliser les versions dites "LTS" , certifiées et maintenues selon un processus qualité au sein des communautés ou éditeurs pour la construction des pods/conteneurs. Debian, Redhat, Ubuntu font partie des communautés les plus attentives.

Sur les besoins "classiques" de persistance : postgres, redis, mariadb, mongodb, elastic, etc... l'approche recommandée de s'appuyer sur les opérateurs kubernetes disponibles et les objets statefulSet. Sur le cluster de production les operators sont déployés par l'hébergeur car généralement ils requièrent les droits globaux. L'équipe projet doit vérifier les versions disponibles lors de la conception de son projet.

5 - Référentiel d'exigences et modalités d'usage

Les exigences du CCT sont classées en 2 niveaux d'exigence (périmètre du Ministère de l'Intérieur) :

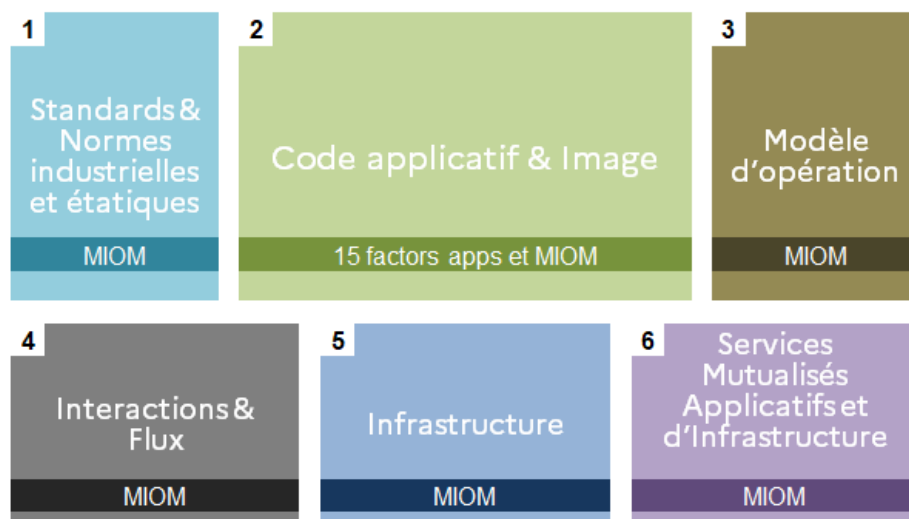
- Primordial : L'exigence est impérative et traitée administrativement.
- I – Important : Exigence prise en compte pour la notation technique de la solution

Précisions sur le cas de l'exclusion administrative (périmètre du Ministère de l'Intérieur) :

- La non-conformité au cadre de norme entraîne l'exclusion administrative lors du dépouillement et la mise en œuvre des actions de remédiation du marché lors de l'exécution du marché.
- La non-conformité aux exigences d'architecture entraîne l'impossibilité d'utilisation du socle de sécurité associé à l'offre Cloud Native

Par défaut les règles du CCT s'imposent. Elles peuvent être précisées dans le cas d'un appel d'offres dans le règlement de consultation pour le dépouillement des offres et dans le CCAP pour l'exécution du marché. Une demande de dérogation est possible. (cf paragraphe ad hoc)

Pour information les exigences sont organisées telles que décrites ci-dessous.



1. **Standards & Normes industrielles et étatiques** : ensemble des exigences relatives aux normes de niveau supérieur à respecter pour toute application étatique
2. **Code applicatif & Image** : exigences issues des “15 factors” pour garantir la conception d’une application “Cloud Native”, associées aux exigences minimales permettant de s’intégrer au contexte du Ministère de l’Intérieur
3. **Modèle d’opération** : voir le chapitre précédent
4. **Interactions & Flux** : exigences d’intégration et d’interaction inter-applicatives dans le contexte étatique et du Ministère de l’Intérieur
5. **Infrastructure** : exigences et prérequis concernant l’infrastructure sous-jacente (notamment Kubernetes)
6. **Services mutualisés Applicatifs et d’Infrastructure** : exigences d’intégration aux services centralisés du Ministère de l’Intérieur, permettant une homogénéisation de la production, un meilleur contrôle et une maîtrise de la dette technique

6 - Annexes

Les normes industrielles, institutionnelles applicables

La conception de système d'information dans le cadre de l'État est encadrée par un ensemble de recommandations ou règles à mettre en œuvre. Ces normes sont citées ci-dessous. Le lecteur est invité à vérifier qu'il dispose des versions les plus à jour.

Norme industrielle	Kubernetes : https://kubernetes.io/fr/ ArgoCD : https://argo-cd.readthedocs.io/en/stable/
Guides & outils pour la conception	DSFR : Design System FR. La charte internet de l'État (qui intègre le RGAA) https://www.systeme-de-design.gouv.fr/ Guide d'éco conception : https://ecoresponsable.numerique.gouv.fr/publications/referentiel-general-ecoconception/ Divers guides de conceptions logiciels: https://guides.etalab.gouv.fr https://catalogue.numerique.gouv.fr https://schema.gouv.fr https://code.gouv.fr
Cadres de pratiques de conception et de conduite de projet agile	https://www.numerique.gouv.fr/actualites/guide-pour-allier-agilite-et-securite-numeriques/
Logiciel libre	Socle InterMinistériel des Logiciels Libres (SILL) de par sa fonction de source pour le référentiel de produits du CCT Ministériel : https://sill.etalab.gouv.fr/fr/software
Normes interMinistérielles de conception de solutions	Doctrine cloud de l'état : https://www.legifrance.gouv.fr/circulaire/id/45205 Référentiel Général d'Accessibilité pour les Administrations : https://accessibilite.numerique.gouv.fr/
Référentiel Général de Sécurité, en association avec le règlement européen et l'EIDAS.	https://www.ssi.gouv.fr/entreprise/reglementation/confiance-numerique/liste-des-documents-constitutifs-du-rgs-v-2-0/

Référentiel Général de Gestion des Archives	https://www.ssi.gouv.fr/entreprise/reglementation/confiance-numerique/le-reglement-eidas/
Référentiel Général de Gestion des Archives	https://francearchives.fr/fr/circulaire/R2GA_2013_10
règlement européen sur la protection des données personnelles	https://www.cnil.fr/fr/reglement-europeen-protection-donnees

Liens vers autres contenus utiles(informatif)

<https://kubernetes.io/fr/>

<https://www.rancher.com/products/k3s>

<https://www.redhat.com/en/technologies/cloud-computing/openshift>

<https://argo-cd.readthedocs.io/en/stable/>

<https://www.redhat.com/en/topics/microservices/what-is-a-service-mesh>

<https://www.redhat.com/en/topics/devops/what-is-gitops>

<https://www.cloudcomputingpatterns.org/>

<https://12factor.net/fr/>

<https://tanzu.vmware.com/content/blog/beyond-the-twelve-factor-app>

<https://www.techworld-with-nana.com/devops-bootcamp>

<https://ecoresponsable.numerique.gouv.fr/publications/bonnes-pratiques/bonnes-pratiques/#bonnes-pratiques-services-numeriques>

La documentation sur le CloudPI (RIE) : <https://pi.minint.fr/reseau-cas-dusage/>

Glossaire

Terme	Description / définition
Agile	Une méthode agile est une méthode de développement informatique permettant de concevoir des logiciels en impliquant au maximum le demandeur (client), ce qui permet une grande réactivité à ses demandes. Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles visent la satisfaction réelle du besoin du client, et non d'un contrat établi préalablement. La notion de méthode agile est née à travers un manifeste signé par 17 personnalités (parmi lesquelles Ward Cunningham, l'inventeur du Wiki), créateurs de méthodes ou dirigeants de sociétés. (Source : https://www.techno-science.net/definition/743.html)
ADR	Enregistrement des décisions d'architecture suivant le modèle MADR
API	Une API est le moyen « standard » désormais, par lequel est exposée une ressource, afin d'en permettre l'accès. Le qualificatif « RESTFULL » renvoie à la conformité de l'API au modèle « REST » qui est un modèle de représentation de l'URL de l'API. Une API est assortie d'un contrat de service qui décrit son fonctionnement. Ce contrat doit être conforme au standard « OPEN API V3 » et accessible aux développeurs.
API Management	Processus de gestion de la totalité du cycle de vie d'une API, de son idée jusqu'à son retrait de service. Décrit dans la Stratégie d'API. Une plateforme d'exposition d'API existe à la DNUM : api.minint.fr , ainsi qu'une autre, de niveau interMinistérielle : api.gouv.fr
BATN	Bureau Appui à la Transformation Numérique
CaaS	Les CaaS ou Containers en tant que Service (Containers as a Service en anglais) sont une catégorie de services Cloud permettant aux développeurs de logiciels de télécharger, d'organiser, d'exécuter, de gérer, de mettre à l'échelle et d'arrêter des containers en utilisant l'interface web ou l'API d'un fournisseur. Source : www.lebigdata.fr
CCAP	Cahier des Clauses Administratives Particulières
CCT	Cadre de Cohérence Technique du MIOM.
CCTP	Cahier des Clauses Techniques Particulières
CCU	Cadre Commun d'Urbanisation

CERFA	Centre d'Enregistrement et de Révision des Formulaires Administratifs)
CHAP	Challenge Hash Authentication Protocol
CI/CD	<p>CI/CD signifie « Continuous Integration/Continuous Delivery », ou intégration continue/livraison continue ; c'est une méthode de mise en œuvre logicielle utilisée par les équipes de développement pour apporter des modifications de code plus fréquentes et plus fiables. Le CI/CD englobe deux ensembles de pratiques complémentaires, chacune reposant fortement sur l'automatisation.</p> <p>(Source : https://www.splunk.com/fr_fr/data-insider/what-is-ci-cd-pipeline.html#:~:text=CI%2FCD%20signifie%20%C2%AB%20Continuous%20Integration,plus%20fr%C3%A9quentes%20et%20plus%20fiables.)</p>
Client	Dans une architecture client-serveur, le client est celui qui est à l'initiative des requêtes faites au serveur
Cluster	Cluster (grappe) : plusieurs systèmes sont interconnectés soit pour augmenter la puissance de calcul (on parle alors de cluster de performance), soit pour offrir une tolérance de pannes accrue par la redondance des composants unitaires (on parle alors de cluster de haute disponibilité). Dans les deux cas, pour bénéficier de l'architecture en grappe, il faut que les applications aient été conçues en conséquence ou que le système d'exploitation, le compilateur et les logiciels sous-jacents (bases de données, middlewares, etc.) prennent en charge les fonctions adéquates de parallélisation des traitements ou de reprise sur incident.
Conteneur	<p>Les conteneurs sont des unités exécutables de logiciel dans lesquelles le code d'application est empaqueté, avec ses bibliothèques et ses dépendances, de manière commune, afin qu'il puisse être exécuté n'importe où, que ce soit sur un ordinateur de bureau, dans un système informatique traditionnel ou dans le cloud.</p> <p>(Source : https://www.ibm.com/fr-fr/cloud/learn/containers#:~:text=Les%20conteneurs%20sont%20des%20unit%C3%A9s,traditionnel%20ou%20dans%20le%20cloud.)</p>
DevSecOps	Le DevSecOps inclut la composante sécurité (security) dans l'approche DevOps, qui lie elle-même le développement (développement) et l'exploitation (opérations).
DINUM	Direction Interministérielle du Numérique

DITP	Direction InterMinistérielle de la Transformation Publique (placée sous l'autorité du ministre de l'Action et des Comptes publics, chargée de la réforme de l'État).
DNUM	Direction du Numérique
DSFR	Design System FR. La charte internet de l'État.
ENTA	Environnement Numérique de Travail de l'Agent
FIP	Factory Instrumental Protocol (Flux d'Information Processus). Actuellement FIP est une norme française (NF C46 601 à NF C46 607) et une norme européenne (EN 50170-3). La promotion et une part d'assistance technique de ce réseau sont effectuées par l'organisation WorldFIP dont le siège se trouve en France. La cible privilégiée de WorldFIP est l'interconnexion de capteurs, actionneurs et automates dans les systèmes automatisés. Comme la quasi-totalité des réseaux de terrain WorldFIP a une structure en trois couches. (https://www.i3s.unice.fr/~map/Cours/LPIREEL/COURS3FIP.pdf)
Gitops	L'approche GitOps repose sur l'utilisation de référentiels Git comme unique source de vérité pour distribuer l'infrastructure en tant que code. Le code envoyé vérifie le processus d'intégration continue, tandis que le processus de distribution continue vérifie et applique les exigences relatives à certains aspects, comme la sécurité, l'infrastructure en tant que code (IaC), ou toute autre limite fixée pour le framework d'application. Toutes les modifications apportées au code font l'objet d'un suivi, ce qui facilite les mises à jour et le contrôle de versions en cas de restauration. (Source : https://www.redhat.com/fr/topics/devops/what-is-gitops#:~:text=Le%20GitOps%20peut%20%C3%AAtre%20consid%C3%A9r%C3%A9,les%20configurations%20de%20l'infrastructure.)
Hébergement	L'hébergement, dans son sens le plus générique, est un service par lequel des ressources de stockage et de calcul sont fournies à une personne ou à une organisation pour l'hébergement et la maintenance d'un ou plusieurs sites Web et services connexes. (Source : https://definir-tech.com/hebergement/)
IAM	Gestion des identités et des accès (Identity and Access Management)
Java	Java est un langage de programmation et une plate-forme de calcul lancé par Sun Microsystems en 1995. (Source : https://www.java.com/fr/download/help/whatis_java.html)

Kubernetes	Kubernetes est une plate-forme open-source extensible et portable pour la gestion de charges de travail (workloads) et de services conteneurisés. Elle favorise à la fois l'écriture de configuration déclarative (declarative configuration) et l'automatisation. C'est un large écosystème en rapide expansion. (Source : https://kubernetes.io/fr/docs/concepts/overview/what-is-kubernetes/#:~:text=Kubernetes%20est%20une%20plate%2Dforme,large%20%C3%A9cosyst%C3%A8me%20en%20rapide%20expansion.)
Logiciel Libre	Un logiciel libre est un logiciel dont la licence dite « libre » donne à chacun le droit d'utiliser, d'étudier, de modifier, de dupliquer, de donner et de vendre ledit logiciel sans contrepartie. La notion de logiciel libre ne doit se confondre ni avec celle de logiciel gratuit (freeware ou graticiels) ni avec celle de shareware (partagiciels). De même, les libertés octroyées par la licence d'un logiciel libre sont plus étendues que le simple accès au code source, ce qu'on appelle parfois logiciel « à sources ouvertes ».
MCO	Maintien en Condition Opérationnelle
MI	Ministère de l'intérieur
MIOM	Ministère de l'intérieur et des Outre-Mer
Node	NodeJS est une plateforme construite sur le moteur JavaScript V8 de Chrome qui permet de développer des applications en utilisant du JavaScript. Il se distingue des autres plateformes grâce à une approche non bloquante permettant d'effectuer des entrées/sorties (I/O) de manière asynchrone. (Source : https://grafikart.fr/tutoriels/nodejs-intro-792)
Open API	Une API ouverte, parfois appelée API publique, est une interface de programmation d'application (Application Programming Interface) qui permet au développeur d'accéder à une application logicielle propriétaire par voie de programmation. (https://www.lemagit.fr/definition/API-ouverte#:~:text=Une%20API%20ouverte%2C%20parfois%20appel%C3%A9e,propri%C3%A9taire%20par%20voie%20de%20programmation.)
Openshift	OpenShift est un service de plate-forme en tant que service de la société Red Hat qui permet de déployer des projets dans des containers. Pour ce faire, OpenShift utilise les technologies Docker et Kubernetes. (https://fr.wikipedia.org/wiki/OpenShift)

Open Source	La désignation Open Source (« source ouverte » en français) s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire la possibilité de libre redistribution, d'accès au code source, et de travaux dérivés. Les logiciels Open Source et les logiciels libres désignent les mêmes logiciels, ceux dont la licence est reconnue libre par l'Open Source Initiative ou la Free Software Foundation. Le terme « Open Source » est en concurrence avec le terme « logiciel libre » (Free Software) recommandé par la FSF. Le terme Freeware (graticiel) désigne des logiciels gratuits qui ne sont ni nécessairement ouverts, ni libres. (Source : https://fr.wikipedia.org/wiki/Open_source)
Pods	Un pod représente une collection de conteneurs d'applications et de volumes fonctionnant dans le même environnement d'exécution. Les pods, et non les conteneurs, sont les plus petits artefacts déployables dans un cluster kubernetes. Les applications s'exécutant dans le même pod partagent la même adresse IP et le même espace de nom réseau. Source : Kubernetes maitrisez l'orchestrateur des infrastructures du futur
PP	
Proxy	Un serveur proxy est une sorte de pont qui vous relie au reste d'Internet. Normalement, lorsque vous naviguez sur Internet, vous vous connectez directement au site Web qui vous intéresse. Un proxy établit à votre place la communication avec le site Web. (Source : https://www.avast.com/fr-fr/c-what-is-a-proxy-server#:~:text=Un%20serveur%20proxy%20est%20une,communication%20avec%20le%20site%20Web.)
Python	Le langage Python est un langage de programmation open source multi-plateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures. (Source : https://www.futura-sciences.com/tech/definitions/informatique-python-19349/)
RACI	L'acronyme RACI (responsible, accountable, consulted et informed) ou RAM (responsibility assignment matrix) désigne dans le domaine du management une matrice des responsabilités . Elle indique les rôles et les responsabilités des intervenants au sein de chaque processus et activité. La matrice RACI donne une vision simple et claire de qui fait quoi dans le projet, en permettant d'éviter une redondance de rôles ou une dilution des responsabilités. (Source : https://fr.wikipedia.org/wiki/RACI)

R2GA	Référentiel Général de Gestion des Archives sur le portail national des archives
Restful API	<p>Une API REST (également appelée API RESTful) est une interface de programmation d'application (API ou API web) qui respecte les contraintes du style d'architecture REST et permet d'interagir avec les services web RESTful. REST (Representational State Transfer).</p> <p>(Source : https://www.redhat.com/fr/topics/api/what-is-a-rest-api#:~:text=Une%20API%20REST%20(%C3%A9galement%20appel%C3%A9e,avec%20les%20services%20web%20RESTful.))</p>
RGAA	Référentiel Général d'Accessibilité pour les Administrations
RGI	Référentiel Général d'Interopérabilité
RGPD	Le sigle RGPD signifie « Règlement Général sur la Protection des Données ». Le RGPD est le règlement européen sur la protection des données, articles de lois sur l'accessibilité des données à des fins de recherche
RGS	Référentiel Général de Sécurité, en association avec le règlement européen eIDAS
Rootless	<p>Les conteneurs rootless font référence à la possibilité pour un utilisateur non privilégié de créer, d'exécuter et de gérer des conteneurs. Ce terme inclut également la variété d'outils autour des conteneurs qui peuvent également être exécutés en tant qu'utilisateur non privilégié.</p> <p>"Utilisateur non privilégié" dans ce contexte fait référence à un utilisateur qui n'a aucun droit d'administrateur et qui n'est "pas dans les bonnes grâces de l'administrateur" (en d'autres termes, il n'a pas la possibilité de demander que plus de privilèges lui soient accordés à eux, ou pour les progiciels à installer).</p> <p>(Source : https://rootlesscontaine.rs/)</p>
SDITN	Sous-Direction à l'Innovation et Transformation Numérique
Shift left	<p>Le Shift Left décrit un principe qui consiste à rendre les flux de travail des entreprises plus efficaces, grâce à des tests et avec des suivis précoces. Cette méthode vous permet de transmettre les connaissances de votre service d'assistance rapidement et facilement à tous les employés de votre entreprise. (Source : https://freshservice.com/fr/shift-left-blog/#:~:text=Le%20Shift%20Left%20d%C3%A9crit%20un,les%20employ%C3%A9s%20de%20votre%20entreprise.))</p>

SI	Selon la définition restreinte donnée par Joël de Rosnay, « Un système est un ensemble d'éléments en interaction dynamique, organisés en fonction d'un but ». Le système d'information n'échappe pas à cette définition. Il est un ensemble dont les éléments sont les constituantes de toute organisation (entreprise, administration, association, groupement, ...). Ces éléments sont de plusieurs natures : organisationnelle, informationnelle, métier, technique, technologique. Tous ces éléments forment un tout (plus ou moins cohérent) et participent à la réussite de l'organisation dans son objectif.
SIC	Systèmes d'Information et de Communication
SILL	Socle interministériel de logiciels libres. Il regroupe l'ensemble des logiciels libres préconisés au sein des ministères. Il est alimenté par des agents publics volontaires Ministériels, sous le contrôle de la DINUM
SPOC	Single Point Of Contention : littéralement « point individuel de contention ». Consiste dans un système à identifier, pour les différents composants (matériels et/ou logiciels), l'existence de points constituant un goulet d'étranglement. Ce composant est alors considéré comme un SPOC pour le système.
SPOF	Single Point Of Failure : littéralement « point individuel de défaillance ». Consiste dans un système à identifier, pour les différents composants (matériels et/ou logiciels), l'existence de points de défaillance pouvant générer un dysfonctionnement du système de par l'impossibilité de redonder ce composant ou de par le choix de ne pas le redonder. Ce composant est alors considéré comme un SPOF pour le système.
ST(SI) ²	Service des Technologies et des Systèmes d'Information de la Sécurité intérieure. Le ST(SI) ² est chargé de concevoir, de piloter et de conduire les projets liés aux systèmes d'information, de communication et de commandement pour l'ensemble des policiers et des gendarmes. Il contribue à la définition de l'action, de la stratégie et de la politique de sécurité du ministère de l'intérieur en matière de système d'information et de communication. Il coordonne les services SIC de proximité de la police et de la gendarmerie. Il anime la politique d'innovation technologique du ministère dans ce domaine. (Source : https://fr.linkedin.com/company/stsisi)
Swagger	<p>Une définition Swagger spécifie un ensemble de métadonnées qui décrivent une API REST.</p> <p>Si vous avez un fichier Swagger définissant une API REST, vous pouvez l'ajouter à votre projet en tant que source de synchronisation externe. Cette source peut être synchronisée avec le projet.</p>

	(Source : https://www.ibm.com/docs/fr/rtw/9.0.1?topic=testing-swagger-definitions)
TCP	TCP (Transmission Control Program) est un protocole permettant l'ouverture de circuits virtuels entre applications
VM	Une machine virtuelle, ou « virtual machine », est « le client » créé dans un environnement informatique, « l'hôte ». Plusieurs machines virtuelles peuvent coexister sur un seul hôte. Les principaux fichiers qui constituent une machine virtuelle sont un fichier journal, un fichier de paramètres de RAM non volatile, un fichier de disque virtuel et un fichier de configuration. (Source : https://www.vmware.com/fr/topics/glossary/content/virtual-machine.html)
VPN	Virtual Private Network, réseau privé virtuel (RPV) : Le principe du RPV consiste à créer un réseau privé au sein d'un réseau public. Cette démarche existe depuis longtemps : les opérateurs s'en servent pour gérer les lignes privées de leurs clients au sein des mêmes « tuyaux ». Aujourd'hui, on parle surtout de réseaux privés virtuels sur Internet. Les RPV mettent en œuvre des mécanismes de contrôle d'accès (authentification des utilisateurs) et assurent la confidentialité des données (cryptographie). Le terme de réseau privé virtuel s'applique aussi au réseau téléphonique : les opérateurs font ainsi transiter sur le réseau public des services évolués de téléphonie jusque-là cantonnés au réseau privé de l'entreprise appel en numérotant uniquement l'extension, renvoi d'appel, conversation à plusieurs, etc. Cette technologie s'étend aussi aux mobiles.
Windows Server	Windows Server est un système multi-tâches, multi-utilisateurs qui dans ses fonctionnalités peut se comparer au système UNIX/Linux. Il présente l'avantage que certains logiciels soient moins chers que leur équivalent fonctionnant sous UNIX, et plus rarement, Linux. Par ailleurs, la quasi-totalité des éditeurs propose des versions de leurs produits pouvant tourner sur serveur Windows.

--- fin du document ---

Référentiel d'exigences applicables au CCT Cloud Pi Native

Note: le terme développeur est générique et fait référence à l'individu ou l'organisation pluridisciplinaire qui est chargée de produire et maintenir : la base de code, le corpus de tests et les fichiers de description d'infrastructure et les documentation technique et usager.

Il est responsable de l'adéquation et de la qualité de la solution au besoin des usagers en collaborant de manières étendues avec les autres acteurs impliqués.

ID	Type	Exigence	Nature	Catégorisation
EX1	I	Respect des standards et des normes applicables industrielles, européennes et étatiques, pour la conception de solutions numériques hébergées dans le cloud native (kubernetes), Design Système de l'État, RGAA, RGS, RGI, doctrine Cloud au centre.	Ministériel	Autres normes et standards
EX2	P	Principe de non déviance et de non fragmentation des normes. Les éventuelles instructions de niveaux inférieurs ne peuvent prévaloir sur les présentes exigences.	Ministériel	Autres normes et standards
EX3	P	Dans le cadre d'un appel d'offre en cas d'incohérence entre les documents, le cct et la liste des exigences sont supérieurs. Le lecteur est invité à vérifier s'il dispose de la version la plus à jour.	Ministériel	Autres normes et standards
EX4	P	Conformité au modèle de responsabilité Cloud Pi Native. Les développeurs/concepteurs sont responsables de la partie développement, du maintien en continu d'une qualité constante de la solution, l'absence de défaut de sécurité et de correction avant toute mise en production.	Ministériel	Modèle d'Opération
EX5	P	Intégration technique de l'usine logicielle à la chaîne De SecOps, maintien par le développeur en condition de disponibilité et de sécurité du point de vérité du logiciel et code d'infrastructure. Exigence limitée à la durée du marché pour un opérateur économique, mais permanent pour la direction d'application.	Ministériel	Modèle d'Opération

EX6	I	Conformité au modèle de responsabilité “you built it - you run it” Les développeurs/concepteurs sont responsables du bon fonctionnement de l’application en production. (il collabore avec l’hébergeur le cas échéant)	Ministériel	Modèle d'Opération
EX7	P	L’orchestration de la sauvegarde est de la responsabilité de l’application	Ministériel	Modèle d'Opération
EX8	I	L’équipe projet teste régulièrement sa capacité à restaurer l’ensemble de l’architecture et des données de l’application avec un scénario de test connu à l’avance	Ministériel	Modèle d'Opération
EX9	I	Couverture fonctionnelle des tests du front end permet de s’assurer de non-régression majeure. (exigence à affiner)	Ministériel	Modèle d'Opération
EX10	I	La couverture fonctionnelle des tests unitaires du back end est de 100%	Ministériel	Modèle d'Opération
EX11	I	Le développeur, usagers de la chaîne de livraison logicielle, signale les éventuels défauts ou besoin d’évolution, il soumet le cas échéant une rectification ou une évolution sur le dépôt de code après l’avoir testé.	Ministériel	Modèle d'Opération
EX12	P	<p>Mise en place organisationnelle et technique de modalité de collaboration étendue continue intégrant (« shift-left ») le flux d’erreur de la chaîne de DevSecOps secondaire.</p> <p>La chaîne secondaire reconstruit les images à partir des codes sources et procède aux mêmes tests avec des outils complémentaires.</p> <p>L’orchestration du pipeline de la chaîne secondaire est supervisée par l’équipe et intègre obligatoirement les tests de vérification issue de la démarche d’homologation de l’application, test de qualité du code (et de couverture de test) et de vulnérabilité.</p> <p>note : La chaîne secondaire bloque les déploiements si la qualité d’ensemble du code est en décroissance ou si l’ensemble porte des vulnérabilités critiques.</p> <p>L’équipe de développement reçoit via l’interface “shift left” une notification des rapports qui doit être intégrée au flux de travail pour correction.</p>	Ministériel	Modèle d'Opération

		Exigence limitée à la durée du marché pour un opérateur économique, mais permanente pour la direction d'application.		
EX13	P	Conformité au modèle de responsabilité Cloud Pi Native L'opérateur de la chaîne CI/CD est responsable du build, de l'exécution des tests fournis et du déploiement de l'application	Ministériel	Modèle d'Opération
EX14	P	Le développeur prend en compte qu'aucune modification n'est effectuée directement en production. Il n'accède pas à l'API kubectl. Toute modification en production doit être effectuée via un versionning du point de vérité GitOps	Ministériel	Modèle d'Opération
EX15	P	L'ensemble des éléments permettant de compiler / builder, tester et exécuter l'application ainsi que déployer l'infrastructure et ouvrir les flux réseaux doivent être fournis par le Développeur / Concepteur à l'opérateur de la chaîne CI/CD L'ensemble de ces éléments sera soumis à un contrôle strict de la part de la chaîne CI/CD du Ministère, notamment pour vérifier que l'infrastructure et les flux réseaux sont en cohérence avec la politique de sécurité du Ministère.	Ministériel	Modèle d'Opération
EX16	I	Intégration technique de l'usine logicielle à la chaîne DevSecOps, maintien par le développeur en condition de disponibilité et de sécurité du point de vérité du logiciel et code d'infrastructure et du WebHook "shift-left" par l'équipe de développement tout au long du contrat.	Ministériel	Modèle d'Opération
EX17	I	L'agilité (à l'échelle) en tant que cadre de travail privilégié et mise en place d'un principe de livraison continue par lot de taille réduite.	Ministériel	Modèle d'Opération
EX18	P	Mise en place d'un principe de collaboration étendue, continue et intégré des processus de livraisons, adéquation de la solution au besoin via l'observabilité, maintien de la qualité, de la disponibilité et de sécurité depuis le développement jusqu'à la production	Ministériel	Modèle d'Opération
EX19	P	La disponibilité cible de l'application est de la responsabilité de la direction d'application qui définit l'architecture technique et organisationnelle en s'appuyant sur les services de l'opérateur et la disponibilité visée	Ministériel	Infrastructure

EX20	I	Formation et de maintien réguliers des compétences des personnels à la technologie Cloud Native (kubernetes), l'agilité et au devsecops, intégration et contribution à la communauté de pratiques Cloud (pi) Native.	Ministériel	Modèle d'Opération
EX21	I	Remontée des défauts, contribution à l'enrichissement de la chaîne DevSecOps et l'offre Cloud Native.	Ministériel	Modèle d'Opération
EX22	P	Le code applicatif doit être testé avant d'être soumis à la chaîne CI/CD du ministère(fonctionnellement, techniquement et sécurité)	Ministériel	Code Applicatif & Image
EX23	P	L'application déploie un collecteur de log (à partir des patterns/service disponibles) et l'intègre avec le collecteur "SIEM" du SOC. Une convention de service est établie.	Ministériel	Code Applicatif & Image
EX24	P	Le concepteur/développeur est libre d'utiliser le langage de programmation de son choix issu de la liste des langages autorisés. (principaux langages)	Ministériel	Code Applicatif & Image
EX25	P	L'application doit pouvoir être monitorable techniquement et fonctionnellement (dont healthcheck) au travers du service de télémétrie mis à disposition Mise à disposition obligatoire au sein de la solution des API de supervision auto descriptive /health (json+problem) et /metrics (prométheus) et Intégration obligatoire de la solution dans la (ou les) chaîne(s) de supervision	Ministériel	Code Applicatif & Image
EX26	I	Respect du guide d'éco-conception. La conception frugale vis à vis des ressources d'infrastructures consommées et l'impact vis-à-vis du terminal de la solution.	Ministériel	Code Applicatif & Image
EX27	I	Couverture et capitalisation des tests, respect des métiers (non régression): La couverture des tests unitaires cible visée pour le back-end doit être de 100% code et significative sur le front end tel que par exemple vérifier des séquences d'usage principale, (login, actes métiers fréquents, etc...) Les bugs détectés donnent lieu d'abord à l'implémentation d'un test automatique avant l'écriture du correctif. Cela permet d'augmenter la base des cas de tests et de renforcer la performance	CloudNative	Code Applicatif & Image

		des tests de non régression.		
EX28	P	Les pods “stateless” doivent pouvoir être redémarrés à tout instant, très rapidement et sans perte de sessions, données ou de transactions.	Ministériel	Code Applicatif & Image
EX29	P	Les sources d'OS servant à la constitution de images de l'application proviennent exclusivement de souche os ayant un support long terme (LTS) d'une communauté active qui maîtrise les packages par un processus organisationnel à révision contrôlé. (typiquement éviter Alpine). note : La liste est maintenue en interministérielle (Dinum et Anssi)	Ministériel	Code Applicatif & Image
EX30	I	Utilisation privilégiée de composants open-source avec une communauté active, suivi des versions maintenues. (Politique & roadmap des versions, fréquence des commits, nb de contributeurs ou garantie)	Ministériel	Code Applicatif & Image
EX31	I	L'architecture de la solution est modulaire, le couplage organisationnel et technique est lâche entre les composants (ex API), les services de haut niveau ne dépendent pas de l'interface offerte de composants de niveau inférieur (dépendance inversion principe)	Ministériel	Code Applicatif & Image
EX32	P	Les conteneurs s'exécutent sans requérir à l'utilisateur root et les ports réseaux internes sont > 1024. (l'exécution de conteneur non rootless est bloquée en production)	Ministériel	Code Applicatif & Image
EX33	I	Le point de présence et de vérité de la sauvegarde est assuré par un service de type S3 indépendant et résilient de l'architecte de production dont le Concepteur/Développeur s'assure régulièrement de sa capacité à restaurer.	Ministériel	Code Applicatif & Image
EX34	P	La compilation et la certification des images est effectuée au sein de l'orchestrateur DevSecOps ministériel. Le développeur s'organise autour de cette contrainte.	Ministériel	Code Applicatif & Image
EX35	P	Choix d'un hébergement adapté à la nature de la donnée manipulée et selon le cadre légal adapté. (Cloud public, Cloud Régalien référencé par la doctrine Cloud au centre, Dédié)	Ministériel	Infrastructure
EX36	I	Les services applicatifs d'observabilité (logs et télémétrie), registre d'image et gestion des secrets mis à disposition doivent être utilisés.	Ministériel	Services Mutualisés

EX37	I	Consommation systématique des données de référence ministérielles	Ministériel	Services Mutualisés
EX38	I	Référencement des objets Métiers dans le catalogue des données Métier du MI	Ministériel	Services Mutualisés
EX39	P	Les services applicatifs mis à disposition ne doivent pas être remplacés par un équivalent Consommation systématique des services socles à enregistrement automatisés : IAM, service d'échange (ident/authent, preuve probante/non-répudiation, exposition de services, référentiels, pfs SIG, ...)	Ministériel	Services Mutualisés
EX40	I	Le service de stockage objet doit systématiquement être utilisés pour les sauvegardes	Ministériel	Services Mutualisés
EX41	P	Le bastion doit systématiquement être utilisé pour les activités d'administration	Ministériel	Services Mutualisés
EX42	I	L'application minimise le besoin d'échange synchrone, les échanges asynchrones sont réalisés via un bus de message.	Ministériel	Interactions & Flux
EX43	I	L'application prend en compte dans sa conception : la séparation des flux usagers, de services et d'administration, l'utilisation privilégiée du protocole https, l'authentification mutuelle des composants.	Ministériel	Interactions & Flux
EX44	I	Toute API exposée doit l'être via le service API management mis à disposition qui identifie + authentifie les consommateurs. L'application met en place la traçabilité en s'appuyant sur la solution d'observabilité disponible.	Ministériel	Interactions & Flux
EX45	I	Les objets et données métiers doivent systématiquement être exposés via le service d'API management fourni. Enregistrement automatisé des objets métiers via API (restful, GraphQL,gRPC) sur les passerelles API Cloud Native. La direction de programme fournit une description de son api et précise les conditions d'accès: La direction de programme fournit un kit de proxification de l'API de son service avec une donnée libre d'usage (Ex: Données anonymisées) permettant de dérouler des cas de tests significatifs. Le kit livré est déployable sous la forme d'un micro-service déployable dans	Ministériel	Interactions & Flux

		Kubernetes via chart Helm qui est inscrit au catalogue du mi.. (Voir les exemples de service d'accélération)		
EX46	P	L'application est responsable de fournir un fichier de description (ex: open api / swagger) qui respecte les standards, préalablement à l'enrôlement en production l'application une vérification est effectuée et l'application doit corriger les défauts.	CloudNative	Code Applicatif & Image
EX47	P	Utilisation de la variabilisation des configuration d'environnement, des services de support (bdd, cache distribué,...), service externe, connectivités.. la configuration de l'application ne doit jamais être codé en dur	CloudNative	Code Applicatif & Image
EX48	I	Maximiser l'accès aux Services tiers et locaux au travers de chaînes de connexion standard (non spécifique au fournisseur de solutions pour favoriser les changements de Service sans impact utilisateur (Principe de d'architecture BackEnd)	CloudNative	Code Applicatif & Image
EX49	P	L'application s'exécute sans état (Stateless). Ses processus/microservices sont indépendants les uns des autres. Les états sont stockés dans un magasin de données centralisé. Cela pour garantir un mise à l'échelle et continuité de service de l'application (Principe Stateless)	CloudNative	Code Applicatif & Image
EX50	P	Garantir l'idempotence des transactions en cas d'arrêt progressif des services (SIGTERM).	CloudNative	Code Applicatif & Image
EX51	I	S'efforcer de minimiser le temps de démarrage du processus/service en limitant l'environnement au juste nécessaire. (principe Jetable)	CloudNative	
EX52	I	L'application est prévue pour un déploiement en continu, la conception du modèle de données permet à la version future et l'actuelle de coexister sans impact fonctionnel.	CloudNative	Code Applicatif & Image
EX53	I	L'ensemble des services applicatifs/processus de l'application, et ses services de supports génèrent leurs propres flux d'événement bruts pour apporter une visibilité sur son comportement. La norme de nommage des flux d'événement est, sans condition, respectée. (Principe d'observabilité)	CloudNative	Code Applicatif & Image

EX54	P	Toutes applications mise en ligne doit fournir une API clairement définie et documentée, pour faciliter sa consommation future des contrats d'interface (Principe de Api first)	CloudNative	Code Applicatif & Image
EX55	P	La performance de toutes les transactions réalisées par l'application doit être mesurable en temps réel pour contrôler la qualité de services et soutenir les investigations en cas de dysfonctionnement. Il est préconisé d'assurer la surveillance de l'application à minima au travers de : Flux d'événement de performance applicative, flux d'événement et donnée pour analyse et reporting métier, journaux d'intégrité et du système (Principe de Télémétrie)	CloudNative	Code Applicatif & Image
EX56	P	Identification utilisateur : L'application doit obligatoirement utiliser le SSO Agent disponible et/ou France connect pour les citoyens	CloudNative	Code Applicatif & Image
EX57	I	Pour la persistance de données personnelles soumises au RGPD, le modèle de données intègre dès la conception, un tag RGPD, des champs dupliqués dédiés à l'anonymisation et des règles et processus d'anonymisation ainsi qu'une politique de droits associés.	Ministériel	Autres normes et standards
EX58	P	<p>L'équipe de développement respecte les règles suivantes permettant une qualité de code en progression et un maintien de la sécurité :</p> <ul style="list-style-type: none"> • minimise la portion spécifique de code développés en s'appuyant sur le catalogue des services proposés. (revoir régulièrement) • met en place une couverture de test unitaire complète du back-end (et fourni les moyens de vérification automatisé à la chaîne secondaire) • mener une analyse de code systématique le plus tôt possible (les langage et IDE modernes fournissent des fonctions de ce type) • mener une analyse de CVE des dépendants importées <p>L'équipe projet met en œuvre une activité continue de refactoring du code produit. La qualité du code ne peut être décroissante.</p> <p>Elle fournit les preuves que des tests de sécurité, de qualité, de robustesse des algorithmes ont été mis en œuvre, et qu'ils n'ont pas remonté de vulnérabilités ou d'erreurs majeures. En s'appuyant notamment sur les logs des analyses des outils de la</p>	Ministériel	Modèle d'Opération

		chaîne primaire. Elle fournit la preuve (ex: le document) des normes de développement et pratiques permettant de maîtriser la qualité du code produit. (refactoring, peer review, etc..)		
--	--	--	--	--

version préliminaire pour appel à commentaire