Intitulé de l'épreuve : Informatique

Nombre de copies : 6

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

EXERCICE 1

1.1) 3 façons de communiquer avec le NOS :

⇒ Interface Web (HTTP)
Cas général d'application pour les administrateurs qui permet d'accéder aux fonctionnalités de manière conviviale.

⇒ Console en SSH
Cas plus spécifiques lorsqu'un navigateur n'est pas accessible (par ex. lors de rebond sur des machines intermédiaire).
Il est nécessaire de connaître les commandes qui peuvent être spécifiques au NOS pour cela et reste réservé aux utilisateurs avancés.

⇒ SNMP
Cas plus spécifique qui permet de monitorer le statut du routeur (ou assimilé). Un certain nombre de commandes peuvent aussi être opérées à distance sur le routeur.

N° A.R.1

1.2)

Unicast : mode de transmission pair-à-pair. La source et le destinataire sont clairement identifiés. ex : 192.168.0.1 -> 192.168.0.2

Multicast : des groupes de transmission sont établis par enregistrement des candidats. Les trames ne sont transmises qu'aux candidats inscrits. ex. 224.0.0.1

Broadcast : les trames sont transmises à l'ensemble des équipements sur le même domaine de diffusion (ethernet), ex. 192.168.0.255/24

1.3) Common Vulnerabilities and Exposures (CVE) fait référérence à l'ensemble des failles et vulnérabilités connues et répertoriées par différents acteurs.
Ces CVE sont répertoriées dans un registre numérique disponible la plupart du temps publiquement (il peut exister des registres plus privés à but commercial).
A l'aide d'outils appropriés tels les SCA (Software Component Analysis), il est possible de corréler l'utilisation des librairies d'un produit avec ces CVEs permettant ainsi de lever les alertes en amont sur la sécurité de l'application produite.

1.4) Une Autorité de Certification (AC ou CA en anglais) est un garant de l'authenticité d'un certificat (x509 en général) produit.

Prenons le cas de Bob et Alice qui ne se connaissent pas. Alice veut échanger avec Bob en toute confiance. et réciproquement. D'un autre côté, Alice et Bob connaissent Anne-Claudie (AC) qui peut certifier de la réelle identité de ces personnes. Alice et Bob vont donc reposer sur cette personne seule pour assurer l'identité de chacun.

Appliqué aux navigateurs, les CA permettent de garantir la véracité des sites visités selon un tiers de confiance.

1.5) Une approche NoCode est une approche où l'utilisateur n'a pas besoin de produire de code pour pouvoir produire une application habituellement écrite par un développeur.
Wordpress peut par exemple être considéré comme une approche NoCode.

Une approche LowCode vise à être utilisé par des profils intermédiaires. Une grosse partie de l'application peut être produite par le logiciel utilisé, mais il est souvent nécessaire de produire une partie sous forme de code afin d'apporter les comportements plus spécifiques.
On pourra notamment citer Bonita, un logiciel qui permet de modéliser du BPMN(2) et qui dans une certaine mesure nécessite d'utiliser du code (ex Groovy) pour implémenter le comportement des tâches métier.

N° 3 R1

l'une et l'autre des approches visent à diminuer la nécessité de passer exclusivement par des développeurs notamment dans les phases amont de la production d'un prototype.

1.6) Types de stockage.

Fichier : Cas d'utilisation du système de fichier dans un Système d'Exploitation donné. Par ex, sous Linux, la partition racine (dite "/") au format ext4 héberge l'ensemble des fichiers du système d'exploitation. NFS est un autre exemple de stockage fichier distant.

Bloc : C'est le format de stockage utilisé au niveau des disques, ou dans un SAN. Des blocs stockés ne le sont pas nécessairement de manière contigues (fragmentation), du système de fichiers utilisé au dessus de ce stockage Bloc détermine la politique appliquée d'exploitation de ces ressources.

Objet : C'est un format de stockage plus récent adressé principalement par des APIs HTTP. On retrouve notamment les protocoles Swift (Openstack) et S3 (AWS).

Intitulé de l'épreuve : Informatique

Nombre de copies : 6

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

1.7) D'un point de vue développeur, la conteneurisation peut être appréhendée dès le début du cycle de développement. Dans les démarches Devops on peut ainsi intégrer les concepts inhérents aux Ops (exploitation) bien en amont en définissant des architectures de déploiement les plus proches de la cible (en production).

On s'oppose en général à la virtualisation matérielle car celle-ci est considérée comme plus coûteuse en termes de ressources si l'on considère un cas d'usage consistant à la reproduction de l'environnement cible. C'est particulièrement vrai dans la mesure où la conteneurisation s'appuie sur une couche légère d'isolation au niveau du noyau (Linux). On notera cependant que de nouvelles approches visent à intégrer la micro virtualisation au sein des processus habituellement dédiés aux conteneurs (orchestrateurs type K8S) afin d'apporter une isolation en terme de sécurité.

Point important à la conteneurisation, c'est qu'il est mal adapté à l'utilisation de ressources en dehors du monde Linux (ce qui rend l'intégration d'un OS Windows impossible).

18) Iaas : Infrastructure As A Service

sauvegardes.

Le prestataire cloud fournit un ensemble de ressources sous la forme de machines (le plus souvent virtuelles) et d'infrastructure réseau. Le client est alors responsable de gérer tous les autres aspects (OS, interconnexion, sécurité). La responsabilité du prestataire s'arrête donc au fonctionnement desdites infrastructures.

Paas : Platform As A Service

Le prestataire fournit alors, en général un cadre plus managé où le système d'exploitation peut alors être imposé. Certains services peuvent être proposés pour faciliter l'exploitation (sauvegarde); ils sont alors souvent facturés en volume ou à l'utilisation.

Une plateforme proposée par plusieurs acteurs pourrait être un kubernetes managé (OVH, AWS, Openshift).

Saas : Software As A Service

Le prestataire fournit cette fois-ci un logiciel en utilisation distante. Il est alors entièrement géré par le prestataire selon les meilleures pratiques (mises à jour, sauvegardes, ...). Certaines parties restent à la main du client la plupart du temps (identification des utilisateurs par ex.).

N° 121

1.9) Node.js est un framework basé sur le langage Javascript.
Javascript est un langage de programmation (ECMAScript).
Node.js exprime souvent la notion de backend Javascript, là où Javascript a souvent été considéré comme une technologie client (navigateur), et où la partie backend a souvent été développée avec des langages historiques (type Java).
En tant que framework, Node.js vient avec un ensemble de patrons de développement pré-intégrés qui facilitent doréenavant le développement d'applications même en dehors du navigateur.

1.10) Un WAF est un équipement de sécurité de niveau 7 (OSI 4.7 : application).
Il permet ainsi d'apporter des règles d'analyse et de filtrage sur de nombreuses caractéristiques opérés à ce niveau notamment sur HTTP (il ne faut pas que le flux soit chiffré).
Un exemple classique consiste à opérer un filtrage sur le chemin de la requête HTTP, en interdisant à tous l'accès, ou en adoptant une politique plus spécifique autorisant un accès à une IP source sélectionnée.
Par ex : http://myservice/admin
chemin filtré.

Ceci n'est pas réalisable au niveau 3/4, couches où se situent traditionnellement un firewall IP.

N° 7.121

1.1.1) Spring-Boot est un facilitateur à l'utilisation du framework Spring.

Les technologies nombreuses et largement utilisées du framework Spring en font un outil puissant mais d'une complexité grandissante (e.g Spring Security, ...).

Spring-Boot apporte les fonctionnalités du framework dans l'esprit "Configuration by default" qui permet l'intégration d'une fonctionnalité par simple ajout d'un module par exemple (Spring-Init).
La configuration est alors paramétrable très simplement par définition dans un fichier (properties ou yaml). Il est possible de reprendre la main sur le comportement complet en ajoutant le code correspondant.

1.1.2) Une API RESTful permet d'accéder à des ressources selon un modèle CRUD à travers le protocole HTTP.
JSON est souvent utilisé dans le corps des requêtes / réponses (même si cela n'est pas obligatoire).
L'Accès à des ressources veut aussi bien dire, accès
GET → en lecture à des informations potentiellement sensibles, mais aussi la capacité à modifier des
POST, PUT → ressources, éventuellement dans le but de nuire.
Il est donc nécessaire de protéger l'accès à ces ressources.

La sécurisation s'opère en général à l'aide d'une authentification plus ou moins sécurisé (Basique, jeton, ...)
Cette authentification peut être portée à différents niveaux de la chaîne d'accès (application, serveur web, API Gateway, Reverse proxy, voire WAF).
Afin d'éviter toute possibilité d'interception, il faut bien évidemment communiquer sur un canal sécurisé (HTTPS).

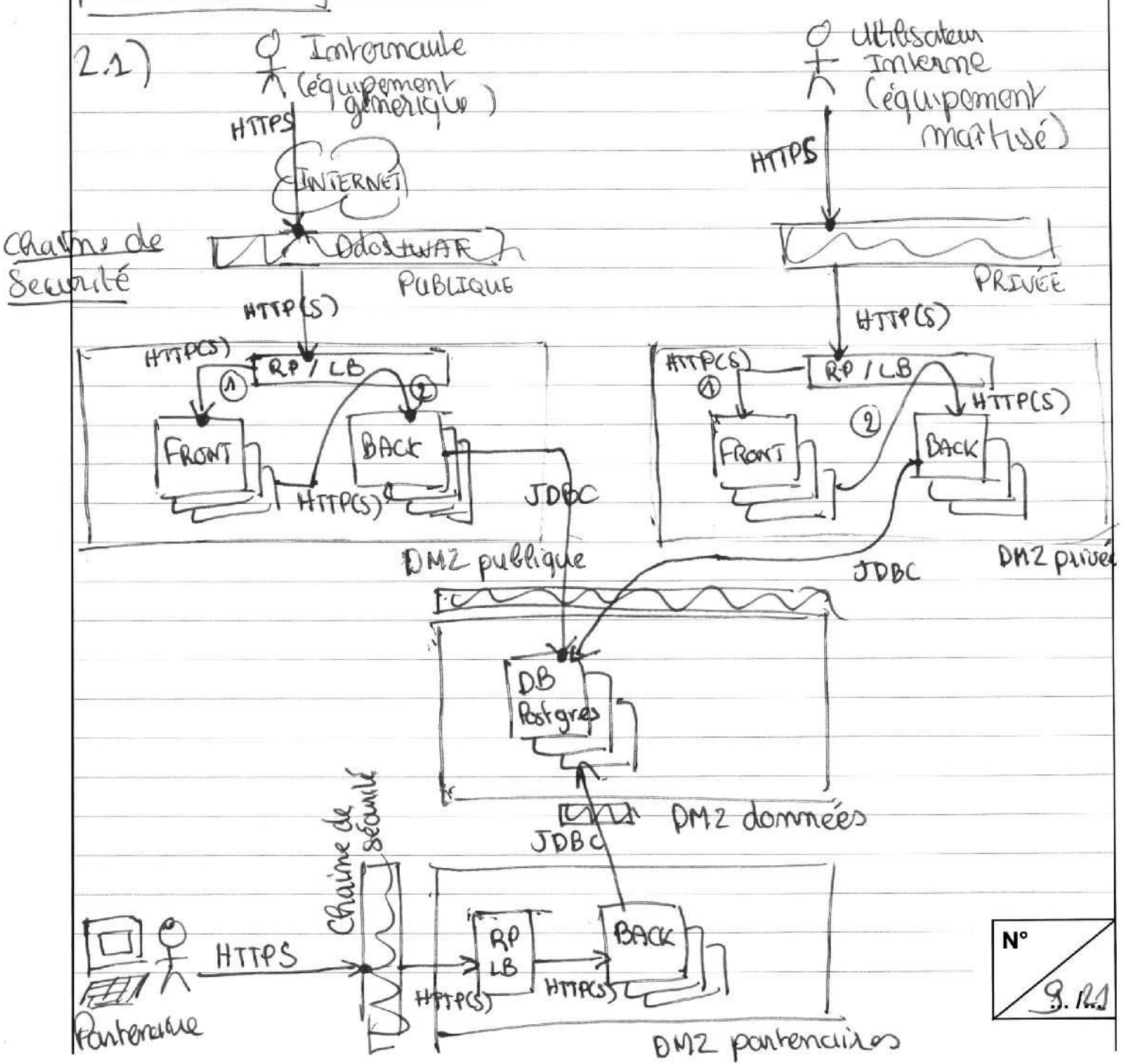
Intitulé de l'épreuve : Informatique

Nombre de copies : 6

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

EXERCICE 2

2.1)



N° 9.2.1

Pour note HTTP(S) désigne l'utilisation au choix selon le niveau de sécurité retenu. On notera qu'en Zero Trust, notamment sur la zone d'accès internet cela ne devrait pas être un choix, les défenses périmétriques étant dorénavant largement insuffisantes.

On pourra choisir dans la même optique un protocole JDBC over TLS pour chiffrer les flux d'accès à la base de données.

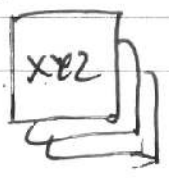
Le protocole HTTP est retenu (API Rest).

Les ports par défaut de ces protocoles:
HTTP: 80
HTTPS: 443
JDBC: 5432

RP/LB : Reverse Proxy / Load Balancer (Filtrage, adaptation ; Répartition de charge, Résilience)
on retiendra habituellement Apache / HAProxy

Une chaîne de sécurité est implémentée par divers composantes, notamment Firewall L3, L7 (WAF) mais aussi de potentiels éléments plus spécifiques délégués à l'infrastructure (anti-Ddos).

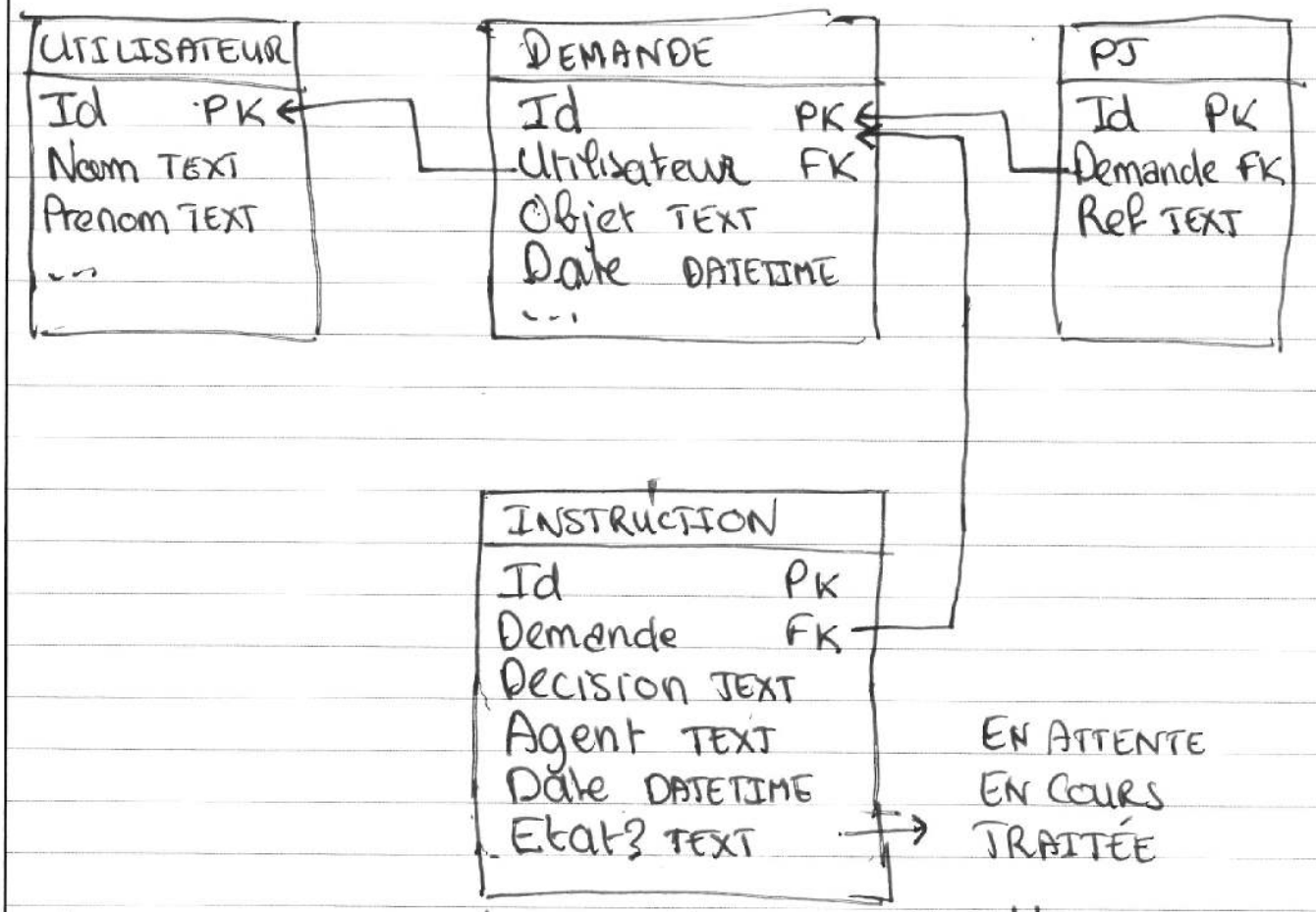
La base de données placée dans une DMZ de données (qui garantit en général qu'aucun flux sortant n'est possible pour éviter les extractions de données frauduleuses). C'est le maillon central pour l'échange des données entre les trois pans applicatifs : utilisateurs internes, agents et partenaires.



désigne un cluster de noeuds applicatifs pour assurer la résilience du système, conformément à la politique LB.

N° 10/21

2.2) Schéma de base de données.



Un UTILISATEUR dispose d'un ensemble d'informations minimales pour l'identifier. On veillera particulièrement à ne pas stocker plus d'informations que nécessaire conformément à la RGPD.

Une DEMANDE est associée à un utilisateur (il peut donc en faire plusieurs). Elle dispose au moins d'un objet qui devra faire l'objet d'une instruction. On se garde le droit d'y ajouter d'autres attributs qualifiant si nécessaire.

Une PJ désigne la référence à une Pièce Jointe. Cette pièce jointe est associée à une demande (qui peut donc en avoir plusieurs). Je privilégie habituellement une référence (pointeur) à la

ressource réelle afin d'éviter de stocker en base des éléments potentiellement volumineux qui rendent l'exploitation plus complexe.
Attention cependant alors à la gestion transactionnelle.
La Référence peut être de différente forme, par exemple une ressource S3.

Une INSTRUCTION désigne le choix opéré par un instructeur pour une demande donnée. Elle référence donc une demande et rend une décision (texte ici).
On ne fait que donner une référence à l'agent (supposée externe: on cherche en effet à éviter
1/ une duplication d'information qui existerait par ailleurs (annuaire)
2/ de découvrir de potentielles informations sensibles sur un instructeur/agent qui pourrait s'exposer à des risques en cas de divulgations de données personnelles.

23) Des droits d'accès à l'application doivent être gérés traditionnellement par :
1/ L'authentification
2/ L'autorisation

On évitera tant que possible de gérer ces aspects au niveau applicatifs en le déléguant à un IDM/IDP. (Identity Management).
Pour les utilisateurs externes, il peut s'agir de l'identité Numérique de La Poste par exemple.
Pour les utilisateurs internes MEAE, il s'agirait d'Arobas (Midpoint).
Le protocole utilisé pour l'autorisation en sous-jacent sera alors classiquement SAML / OpenIDC.
En se basant sur ces briques, la gestion des droits d'accès se fait uniquement à l'établissement de la session utilisateur (sans stockage persistant).

N° 12.1.1

Intitulé de l'épreuve : Informatique.
Nombre de copies : 6

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

La gestion de l'authentification sera préférablement déléguée à l'API Gateway en cas d'API REST.

Pour les partenaires, s'agissant de communication M2M, il sera préférable de s'appuyer sur des droits portés par un token JWT.

2.4) Pour la protection des données :

- Isolation de la base de données dans une DMZ dédiée (et SIEM adapté)
- Conservation minimale des données personnelles en base, en accord avec le RGPD.
- Chiffrement, si caractère sensible, des bases de données (FS, database, table, colonne, ...) et des P.J. fournies par l'utilisateur.
- Analyse automatique des contenu des PJs si jamais les données sensibles doivent être refusées.

N° 13/21

2.5)

Dans sa version initiale, l'utilisateur n'a pas accès aux informations d'instruction.
Cela permet d'imaginer une version plus sécurisée de l'application avec 2 bases, l'une pour les utilisateurs (sans données d'instruction), et l'autre pour l'instruction. Cette dernière peut agir par synchronisation régulière des données utilisateur (pour éviter l'utilisation de transaction distribuées habituellement inefficaces).

De sorte ramené à l'utilisateur peut être fait sur une base unique (présentée ci-avant). Ainsi l'utilisateur pourra accéder à son interface dédiée pour vérifier l'avancement de l'instruction.
On pourra préférer user deux bases séparées pour des raisons spécifiques de sécurité en établissant une base utilisateur simplifiée porteuse d'informations d'instruction élaguées (sans l'instructeur notamment).

Une autre option qui m'impacte en rien la sécurité est de passer par l'envoi de messages à l'utilisateur. (mail, SMS, ...). Ceci est pris en charge par le code applicatif. Il sera alors probablement nécessaire d'ajouter les drapeaux (flags) correspondant pour stocker sur l'envoi effectif desdites confirmation, ces envois étant sujets à erreur.

N°

2.6) Pour un dépôt fluide des PJs utilisateur, on privilégiera des envois et stockages asynchrones.

On évitera donc un envoi et stockage transactionnel de la demande.

On visera probablement un processus scindé en deux parties :
- Stockage de la demande
- Traitement des pièces jointes avant stockage.

En effet, s'agissant de données externes, les PJ. peuvent présenter des vecteurs d'attaques qu'il convient de jugula avant stockage et rattachement.
On pense notamment aux fichiers PDF voire même aux images (attaque par stéganographie).
Une bonne pratique consiste à transformer (ou retransformer) ces PJ. en image avant stockage pour en avoir une version maîtrisée (attention aux aspects légaux de la transformation de PJs).
L'utilisation d'un antivirus reste complémentaire.
Ce processus est bien évidemment long et nécessite un traitement disjoint du simple stockage de la demande.

On pourra privilégier une notification asynchrone de la bonne prise en compte des PJs soit directement dans l'application (e.g. toaster asynchrone) soit par l'envoi d'une notification mail lorsque l'analyse peut s'avérer longue de manière pragmatique.

En backend, l'utilisation de files de traitement, de MoM et d'une architecture micro-services bien pensée permet de garantir une gestion efficace de ces traitements. On pourra par exemple appliquer

des principes d'élasticité sur un micro-service conteneurisé et appliquer dans le cadre d'une orchestration K8S.

2.7) S'agissant d'une application déjà orientée RESTful, le passage à une application mobile devrait être simple.

On aura privilégié en amont l'utilisation d'une technologie front versatile comme React. Le passage au mobile pourra donc se faire en utilisant le framework associé React Mobile.

On n'oubliera pas, comme dans le cas de l'application Web, d'appliquer le System Design de l'État (obligation pour toute nouvelle application exposée sur le WEB), une qualification pour le SIG pourra être nécessaire en ce sens.

De la même manière, on appliquera au mieux le RGAA pour satisfaire le plus grand nombre d'utilisateurs.

Intitulé de l'épreuve : Informatique.

Nombre de copies : 6

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

EXERCICE 3

Phases de vie

① . Naissance du projet / Exigences
② . Sélection du prestataire externe
③ . Initialisation
④ . Développement du projet
⑤ . Maintient en Condition Opérationnelles / de Sécurité (MCO/MCS)
⑥ . Décommissionnement

N° 12 12.1

[redacted]

① Naissance du projet / Exigences

- Intervenants : . Métier / AMOA
  . Chef de projet / AMOE DNUM
- Livrables : . cahier des exigences fonctionnelles
- Comitologie : . informelle, à la demande métier, à adapter selon le cadrage voire conseil technologique nécessaire
- Moyens : . Salles de réunions
  . Outils de collaboration numériques

② Sélection du prestataire

- Intervenants : . Bureau des marchés
  . AMOA
  . AMOE DNUM
  . Prestataires externes
- Livrables : . Proposition de marché (à publier)
  . Réponses aux marché
- Comitologie : Au besoin pour clarifier les points attendus.
- Moyens : Téléphone, mails, salles de réunions, outils collaboratifs

N° 18/21

③ Initialisation

- **Intervenants** : . AMOE DNUM
  . Prestataire retenu
  . Métier ? AMOA ?
- **Livrables** : . Méthodologie de mise en œuvre de la méthode de développement retenue (e.g agile / SCRUM)
- **Comitologie** : . 1 ou 2 réunions d'échange.
- **Moyens** : Salles de réunions, outils collaboratifs.

④ Développement du projet (supposé agile)

- **Intervenants** : . AMOA / PO
  . AMOE / Scrum Master (SM)
  . Développeurs
- **Comitologie** :
  - . Daily (Dev + SM) 1/j
  - **SCRUM** 1/sprint :
    - . Sprint planning (Dev + SM + PO)
    - . Sprint review (DEV + SM + PO)
    - . Sprint retrospective (Dev + SM + PO)
    - . Sprint grooming (Dev + SM)

Pour note, selon l'envergure du projet, on devra envisager des personnes intermédiaires (dites proxy) pour faciliter l'intermédiation.

On travaille en SCRUM généralement sur des sprints de 2 semaines que l'on pourra allonger

de par la disponibilité des PO (méthéa).

Des review, rétrospectives et autres réunions nécessitant la participation des agents sera préférablement opéré dans les locaux du ministère, mais la présence régulière dans les locaux du prestataire reste nécessaire.

, Moyens :

; de messagerie instantannée, le téléphone et plus généralement les outils collaboratifs sont nécessaires à la bonne mise en oeuvre des fonctionnalités attendues.

- La méthodologie agile contribue à une analyse rapide des divergences qu'il pourrait existes en ce sens.

- On privilegiera une plateforme de gestion de code et plus généralement de Dev (Scc) opé fournie par le ministère lorsque cela est possible (Forge Cloud par exemple si le projet ne possède pas de qualification sensible).

- Dans le cas contraire, on demandera au prestataire de nous fournir un accès sur sa plateforme équivalente respectant les principes de sécurité établis par le NEAF en collaboration avec le RSSI.

- Enfin, en cas d'impossibilité on veillera à récuperer les rapports fournis par les outils classiques de gestion SAST et/ou SCA, en refusant éventuellement une livraison qui n'attendrait pas les critères d'acceptation énoncés en amont.

N° / 2021

Intitulé de l'épreuve : Informatique

Nombre de copies : 6

Numérotez chaque page (dans le cadre en bas de la page) et placez les feuilles dans le bon sens.

(5) MCO / MCS

. Moyens :

On s'appuiera nécessairement sur des outils de SCA pour qualifier les nouvelles vulnérabilités qui sont publiées tous les jours.

Le cas échéant, on veillera à apporter une mitigation soit logicielle lorsque la Chaîne d'Intégration (CI) et Chaîne de Déploiement (CD) est encore maîtrisée.

Sinon on s'appuira sur d'éventuelles protections au niveau du système (WAF).

En dernier recours, il pourra être décidé de décommissionner une application devenue trop risquée d'un point de vue de la sécurité.

N° 27.12.1

N°

... /...

N°

... /...

N°

... /...