epsi®
l'école d'ingénierie informatique

Pro Alterna®

CERTIFICATION PROFESSIONNELLE EXPERT EN INFORMATIQUE ET SYSTEME D'INFORMATION RNCP - RNCP N°35584

BLOC 4 – Concevoir et développer des solutions applicatives métier et spécifiques (mobiles, embarquées et ERP)

Cahier des Charges de la MSPR « Conception d'une solution applicative en adéquation avec l'environnement technique étudié »

COMPÉTENCES ÉVALUÉES :

- Collecter les besoins métiers des utilisateurs en menant des interviews auprès d'eux pour comprendre leurs activités et leurs contraintes métier afin d'étudier les opportunités et la faisabilité technologique d'une solution applicative spécifique ou métier.
- Concevoir une architecture applicative selon la complexité du système d'information existant de type architecture distribuée, ou micro-service évolutive et tolérante aux pannes.
- Développer une application adéquate selon la stratégie applicative de l'environnement en utilisant un langage de programmation approprié dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.
- -Développer une solution applicative intégrée en utilisant le paramétrage et le langage de programmation spécifique de l'éditeur dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.
- Effectuer les tests de la solution applicative paramétrée ou développée pour identifier les erreurs et dysfonctionnements et établir les plans de correction/d'amélioration avant sa mise en production.
- Appliquer l'intégration continue dans le cadre du développement d'une application en utilisant un outil d'intégration continue afin de vérifier la conformité de la solution et les besoins utilisateurs.
- -Vérifier la conformité entre la solution développée ou paramétrée et les fonctionnalités attendues à partir des retours des directions métiers afin de rédiger la documentation et les référentiels orientés utilisateurs.
- Conduire le changement auprès des métiers lors du déploiement d'une solution applicative ou intégrée en mettant en place une démarche de participation, de communication et de formation pour accompagner les utilisateurs à l'intégration du nouvel outil dans leurs habitudes de travail.

PHASE 1 : PRÉPARATION DE CETTE MISE EN SITUATION PROFESSIONNELLE RECONSTITUÉE

- Durée de préparation :
  - *25 heures*
- Mise en oeuvre :
  - *Travail d'équipe constituée de 4 apprenants-candidats (5 maximum si groupe impair)*
- Résultat attendu :
  - Réalisation d'une solution applicative fonctionnelle sous la forme d'API REST.
  - Travailler selon les méthodes Agiles et pouvoir l'expliciter

PHASE 2 : PRÉSENTATION ORALE COLLECTIVE + ENTRETIEN COLLECTIF

- **Durée totale par groupe** : 50 mn se décomposant comme suit :
  - 20 mn de soutenance orale par l'équipe.
  - 30 mn d'entretien collectif avec le jury (questionnement complémentaire).
  - Objectif : mettre en avant et démontrer que les compétences visées par ce bloc sont bien acquises.

- **Jury d'évaluation** : 2 personnes (binôme d'évaluateurs) par jury – Ces évaluateurs ne sont pas intervenus durant la période de formation et ne connaissent pas les apprenants à évaluer.

I - PRÉSENTATION DE L'ENTREPRISE ET DE SON ACTIVITÉ

Préambule : L'entreprise choisiepour cette MSPR est fictive,les prénoms sont fictifs, toute ressemblance à un cas réel serait purement fortuite.

La société PayeTonKawa est spécialisée dans l'import de café en France.
Depuis 2018, la société vend du café provenant du monde entier aux particuliers via son site internet public payetonkawa.fr.
Cette année elle souhaite moderniser son SI et se lancer dans la vente de café aux professionnels de la restauration via un réseau de distributeurs dédiés.

II - DESCRIPTION DU SYSTEME D'INFORMATION CLIENT

La société dispose actuellement d'un ERP contenant les informations sur les commandes, les produits et stocks.
Elle dispose aussi d'un outil de CMR contenant la liste de ses clients.
Ces 2 outils sont actuellement utilisés par un site web qui fournit une interface web de type e-commerce

La société dispose actuellement d'une équipe de 2 développeurs et 1 informaticien SysOps qui déploient et maintiennent la solution existante sans usine logicielle

III - CONTEXTEDU BESOIN – CAHIER DES CHARGES

3.1Développements et conduite du changement:

Evolution de l'infrastructure :

Le but de cette demande est de faire évoluer l'infrastructure existante en migrant vers une solution développée de zéro basée sur une architecture micro-services.
L'architecture micro-services doit permettre :

- Une réorganisation des équipes (1 équipe dédiée par micro-service)
- Accélérer les rythmes des déploiements en déployant les services indépendamment.
- Être capable d'utiliser différentes technologies suivant les services en fonction des besoins.

La nouvelle infrastructure devra supporter 3 services :

- Gestion des clients
- Gestion des produits
- Gestion des commandes

Chaque service devra fournir, via une API REST, des opérations de création/modification/suppression et recherche des objets liés (cf détails des endpoints à développer).

Cela devra donc passer par la mise en place d'une plateforme qui permettra de mettre à disposition des APIs REST exposant les données aux consommateurs (Webshop / Revendeurs / ...) en limitant leurs accès aux seuls données utiles à leur besoin.

Réorganisation des équipes
La mise en place d'une plateforme d'intégration continue et de déploiement continue permettra d'automatiser les tests et les déploiements de la solution afin que permettre la mise en place d'une organisation d'équipe séparant les rôles de développement et de déploiement.

Afin de pouvoir grandir et pouvoir délivrer de nouveaux services, la société PayeTonKawa recrute 10 à 15 développeurs et 2 chefs de projets. L'objectif est de pouvoir délivrer rapidement de nouvelles APIs en fonction des besoins qui seraient remontés par les équipes commerciales et les distributeurs partenaires.
Actuellement les équipes de développements sont sollicitées directement par les équipes commerciales ou le management dès qu'un nouveau besoin est exprimé.

Un process Agile permettra de gérer, via un backlog, l'avancement des développements, bugs, et suivi projet.

Un plan de conduite du changement devra être construit afin de permettre une transition vers cette nouvelle organisation et processus. En lien avec ce plan, la société PayeTonKawa s'attend aussi à des propositions d'organisation des équipes de développements.

Tests et qualité
Les APIs étant critiques et une indisponibilité des APIs impactant le chiffre d'affaires, la société PayeTonKawa souhaite s'assurer de la qualité des livrables pour s'assurer que ses revendeurs ainsi que le Webshop puissent commander à tout instant.

Il faudra donc mettre en place des outils afin de pouvoir s'assurer de la qualité des livrables, ainsi qu'un outil permettant de mesurer la qualité du code source et la dette technique des applications ainsi que les alertes de sécurité potentielles (OWASP TOP 10 sera le référentiel que la société souhaite utiliser).

Monitoring
Le monitoring de la plateforme sera important afin de pouvoir suivre, via des logs, les erreurs potentielles et faire le support de la plateforme en cas d'appels en échec.

Les administrateurs Payetonkawa doivent pouvoir visualiser à tout moment les statistiques sur :
- Le nombre d'appels http par API
- Les code http de retours
- Les temps moyens d'exécution des appels http
- Le nombre de messages échangés sur le message broker par file d'attente

3.2 Objectifs

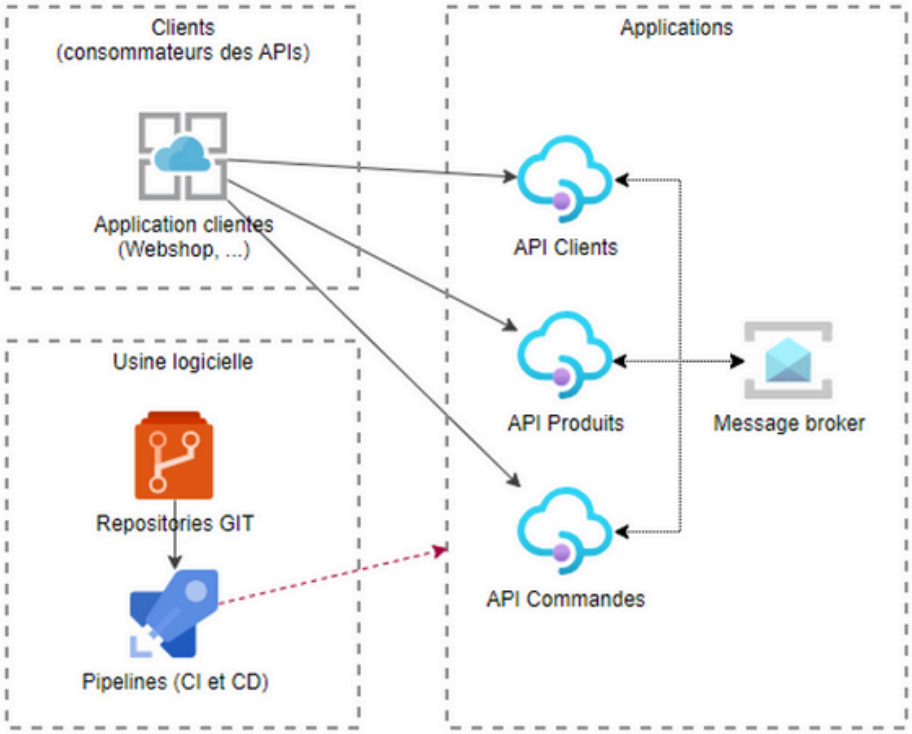
Le travail demandé sur ce projet consiste à :
- Créer 3 applications exposant des API REST. Le langage à utiliser est ouvert et au choix des équipes. Chaque application devra avoir son repository git.
- Sécuriser l'accès aux APIs.
- Monitorer les APIs
- Configurer un usine logicielle capable d'exécuter des pipelines de CI (intégration continue) et des pipelines deCD (Déploiement continue)

L'architecture doit respecter le design pattern micro-service, chaque API doit donc être autonome et avoir sa propre base de données. Les applications doivent être packagées sous la forme d'images Docker afin de pouvoir être déployées facilement sur n'importe quelle infrastructure supportant ce type de service.

Un mécanisme de synchronisation des données entre les services doit donc être mis en place afin de garantir que chaque service utilise des données à jour (ex de modification d'un produit ou d'une fiche cliente qui nécessiterait de mettre à jour les données de commandes). Un mécanisme de message broker devra être mis en place pour permettre cette synchronisation.

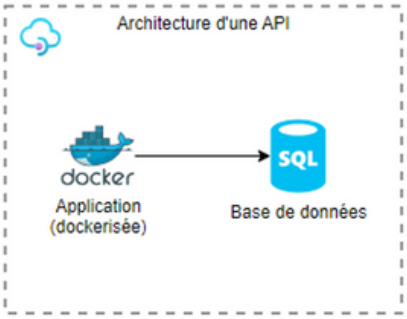
Chaque API devra s'abonner ou publier des messages sur ce bus de messages afin de notifier ou d'être notifié des changements.

L'architecture doit évoluer pour être équivalente à cette cible :



L'usine logicielle permet de gérer les repositories GIT et le pipeline d'intégration continue (CI) et de déploiement (CD) afin de déployer les applications dans l'environnement cible.

Chaque API devra être composée de l'application conteneurisée ainsi qu'une base de données :



Chaque API devra supporter les appels REST standard permettant les opérations CRUDs sur les objets.
Ex pour l'API clients :

Method	Url	Description
GET	/customers	Récupère une liste de client.
GET	/customers/{id}	Récupère un client par son identifiant
POST	/customers	Créé un nouveau client
PUT (ou PATCH)	/customers/{id}	Met à jour une fiche client
DELETE	/customers/{id}	Supprime une fiche client

3.3 Livrables :

Pour chacune des APIs livrées, la société PayeTonKawa souhaite que soient livrés et accessibles à tout instant:

- Le code source, disponible sur un repository git
- Une collection POSTMANet un fichier environnement permettant de tester manuellement l'API
- La documentation technique décrivant les points suivants (argumentés) :
  - Schéma d'architecture (infrastructure et dépendances)
  - Sécurité de l'API
  - Langage de programmation
  - Règles d'hébergement (scaling de la solution)
- Gestion du code source
- Gestion du déploiement (CI / CD)
- La documentation technique décrivant les points suivants (argumentés) :
  - Langage de programmation
  - Gestion du code source
- Gestion du déploiement (CI / CD)

Dans le cadre de la mission d'accompagnement au changement des équipes, sera attendu une description de la démarche proposée afin d'accompagner les équipes de PayeTonKawa dans ces changements et réorganisation sur les différents points :

- Changements techniques liés à la nouvelle architecture (monolithique vers micro-services). L'équipe actuelle n'étant pas nécessairement ouverte aux nouvelles technologies.
- Changements organisationnels liés à la mise en place de l'agilité. Les équipes de développement actuelles fonctionnant en cycle en V. Les équipes management et commerciales qui expriment les besoins et sollicitent les développeurs pour ajouter des fonctionnalités.

3.4 Contraintes techniques :

APIs
Le langage de programmation est libre. Python, C# Java. Il devra être justifié.

Bases de données
Le choix du type de base de données est libre. SQL Server, MySQL, … relationnel ou non relationnel. Il devra être justifié.

Message broker
Le choix du message broker est libre. RabbitMQ, solutions PaaS, … Devra être justifié le choix des files d'attentes et types de messages qui transiteront.

Conteneurisation
Chaque application doit être conteneurisable via Docker. La conteneurisation doit permettre de gérer la montée en charge en montant plusieurs instances d'une même image au cas où un pic de commandes par exemple génère une surcharge de la plateforme. Ex : 2 ou 3 instances de l'API Produits doivent pouvoir fonctionner en parallèle.

Monitoring
Le choix de l'outil de monitoring est libre ainsi que la manière de stocker les logs. Payetonkawa souhaite obtenir des informations sur le nombre d'appels aux APIs (nombre d'appels http, code http de retours et exceptions dans le code).

Intégration continue
Le choix de l'outil de CI/CD est libre et devra être justifié (ex : Jenkins,Github Actions, …)

La société utilise actuellement GitFlow pour sa gestion de branches et souhaite garder ce principe pour tous ses repositories. Les commits devront être organisés pour être lisible et refléter l'évolution du produit sur la branche de développement et la branche principale. L'objectif étant de pouvoir facilement retracer les évolutions en cas d'erreur en production pour retrouver le code et l'évolution ayant générée un bug. Tout changement de flow (gestion des branches) devra être justifié.

-Dans le cadre de la MSPR, il est accepté que le déploiement continue puisse ne pas être fonctionnel à 100%, mais il faudra à minima que les images cocker puissent être récupérable depuis les pipelines pour être déployées manuellement.

Hébergement des APIs
Le choix de l'hébergement devra être justifié.Dans le cadre de la démonstration un déploiement local est accepté mais une solution hébergée chez un hébergeur devra être étudié et argumentée.

Tests
Des tests unitaires devront être mis en place et une couverture du code d'au moins 95% est attendue.
Des tests d'intégration automatisés devront être mis en place afin de pouvoir tester de manière automatisée les APIs déployées.

La plateforme d'intégration continue devra être utilisée pour automatiser ces exécutions.

IV - EXPRESSION DE LA DEMANDE– RÉALISATION ATTENDUEDE L'APPRENANT.E

La réalisation attendue de l'apprenant est de se mettre à la place du prestataire réalisant les applications. Il devra être décisionnaire tout au long du projet et proposer une solution technique en cohérence avec le cahier des charges du client. Cela nécessite qu'il montre son savoir-faire sur les sujets suivants :

- Choix de l'outil de développement et du langage de programmation à adopter
- Appel d'API Rest
- Sécurisation d'une API Rest
- Développement et tests unitaires
- Réalisation de documentation technique
- Mise en place de pipelines de CI / CD sur un repository git

Un plan de conduite du changement liés à la réorganisation technique des équipes et de leur mode de fonctionnement afin de faciliter la transition vers la nouvelle organisation.

V - SPÉCIFICATION(S) TECHNIQUE(S), DOCUMENTATION, ANNEXES,SCHEMAS, BASES

5.1 : Sources de données

Des données du CRM et de l'ERP sont mockées et mises à votre disposition via des APIs REST. Vous pouvez donc effectuer vos tests d'intégration sur la base de ces APIs. Les données renvoyées étant mockées, le contenu renvoyé peut ne pas avoir de lien avec le contexte(cafés).

Le mock des APIs ERP et CRM est accessible via l'url : https://615f5fb4f7254d0017068109.mockapi.io/api/v1.

Cette API expose une liste d'objets qui devront être reproduits dans les nouvelles APIs. Le nouveau découpage en 3 APIs aura nécessairement un impact sur les modèles exposés. Ex : Le champ orders des clients ne devra plus être exposés dans l'API clients.

La liste des client sissue est accessible via l'API REST : /customers.
La liste des commandes d'un client est accessible via l'API REST /customers/<customer id>/orders.
La liste de produits d'une commande est accessible via l'API REST /customers/<customer id>/orders/<orderid>/products.

Ex d'url pour accéder aux produits de la commande3 du client 11 :
615f5fb4f7254d0017068109.mockapi.io/api/v1/customers/11/orders/3/products

La liste des produits est accessible via l'API REST : /products.
Le détail d'un produit peut être accessible via l'API REST : /products/<product id>.
La liste des produits peut varier des informations produites liées aux commandes.

Cette différence vient de la plateforme de Mock. Cela ne doit pas vous bloquer pour vos développements et tests.

Cette plateforme n'a pas vocation à être appelée directement mais doit servir uniquement à des fins de requêtage (GET) afin de regarder les modèles de données.
Si vous souhaitez cloner le mock dans votre propre espace, vous pouvez le faire via l'url suivante : https://mockapi.io/clone/615f5fb4f7254d001706810a

Les compétences évaluées durant cette MSPR :

- Collecter les besoins métiers des utilisateurs en menant des interviews auprès d'eux pour comprendre leurs activités et leurs contraintes métier afin d'étudier les opportunités et la faisabilité technologique d'une solution applicative spécifique ou métier.
- Concevoir une architecture applicative selon la complexité du système d'information existant de type architecture distribuée, ou micro-service évolutive et tolérante aux pannes.
- Développer une application adéquate selon la stratégie applicative de l'environnement en utilisant un langage de programmation approprié dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.
- -Développer une solution applicative intégrée en utilisant le paramétrage et le langage de programmation spécifique de l'éditeur dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.
- Effectuer les tests de la solution applicative paramétrée ou développée pour identifier les erreurs et dysfonctionnements et établir les plans de correction/d'amélioration avant sa mise en production.
- Appliquer l'intégration continue dans le cadre du développement d'une application en utilisant un outil d'intégration continue afin de vérifier la conformité de la solution et les besoins utilisateurs.
- -Vérifier la conformité entre la solution développée ou paramétrée et les fonctionnalités attendues à partir des retours des directions métiers afin de rédiger la documentation et les référentiels orientés utilisateurs.
- Conduire le changement auprès des métiers lors du déploiement d'une solution applicative ou intégrée en mettant en place une démarche de participation, de communication et de formation pour accompagner les utilisateurs à l'intégration du nouvel outil dans leurs habitudes de travail.