codein.
Digital addict

Business case

D'UNE ARCHITECTURE MONOLITHIQUE À UNE ARCHITECTURE MICROSERVICE

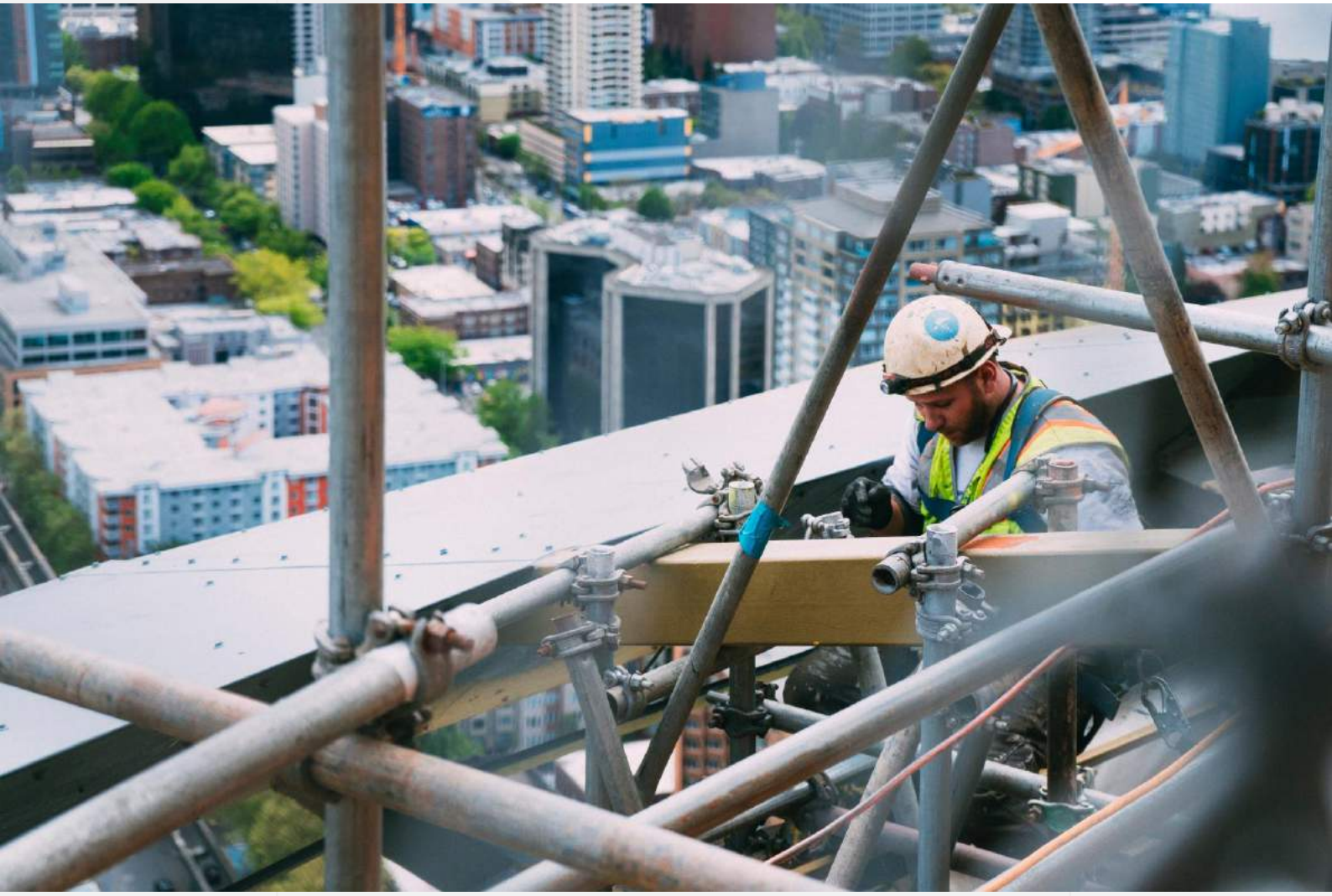
OPPBTP

OPPBTP
La prévention BTP

#1

Le contexte

L'OPPBTP



L'Organisme professionnel de prévention du bâtiment et des travaux publics (OPPBTP) est un organisme français administré paritairement par des représentants des salariés et des employeurs des entreprises du BTP.

Il a pour mission de sensibiliser les professionnels du bâtiment et des travaux publics pour prévenir les accidents du travail et les maladies à caractère professionnel, améliorer leurs conditions de travail tout en leur apportant des services en ligne personnalisés.

OPPBTP
La prévention BTP

Le contexte

L'OPPBTP propose une plateforme Web d'information diffusant du contenu personnalisé en fonction du métier du professionnel : actualités, conseils pratiques, formations dédiées, espace documentaire. Des services personnalisés sont également proposés au sein d'un espace connecté propre à chaque entreprise (SAAS). De 2012 à 2018, une dizaine d'outils ont été créés au sein de cet espace connecté. Riches techniquement comme fonctionnellement, ces outils ont représenté une charge de travail d'environ 2500 jours hommes entre leur conception et leur réalisation. L'ensemble de cette plateforme est géré grâce au CMS eZ Publish.

Les outils s'additionnant, la complexité globale du projet s'amplifiait de manière croissante, emmenant avec elle sa dette technique et ses lenteurs. Le coût ainsi que l'effort humain que les migrations laissaient suggérer n'ont fait que repousser leurs échéances. Les versions logicielles utilisées sont progressivement devenues dépréciées puis obsolètes. Par ailleurs, les usages du Web évoluant, l'OPPBTP s'est retrouvé bloqué face aux besoins de ses utilisateurs afin de ne pas alourdir une dette technique déjà importante.

Codéin a ainsi été contacté afin d'établir la stratégie globale de migration de l'ensemble de la plateforme. Les enjeux étaient les suivants :

- Proposer une architecture cible où chaque outil est indépendant.
- Éviter l'effet tunnel en réalisant une migration étape par étape.
- Ne pas reproduire dans 5 ans ce même constat.

#2

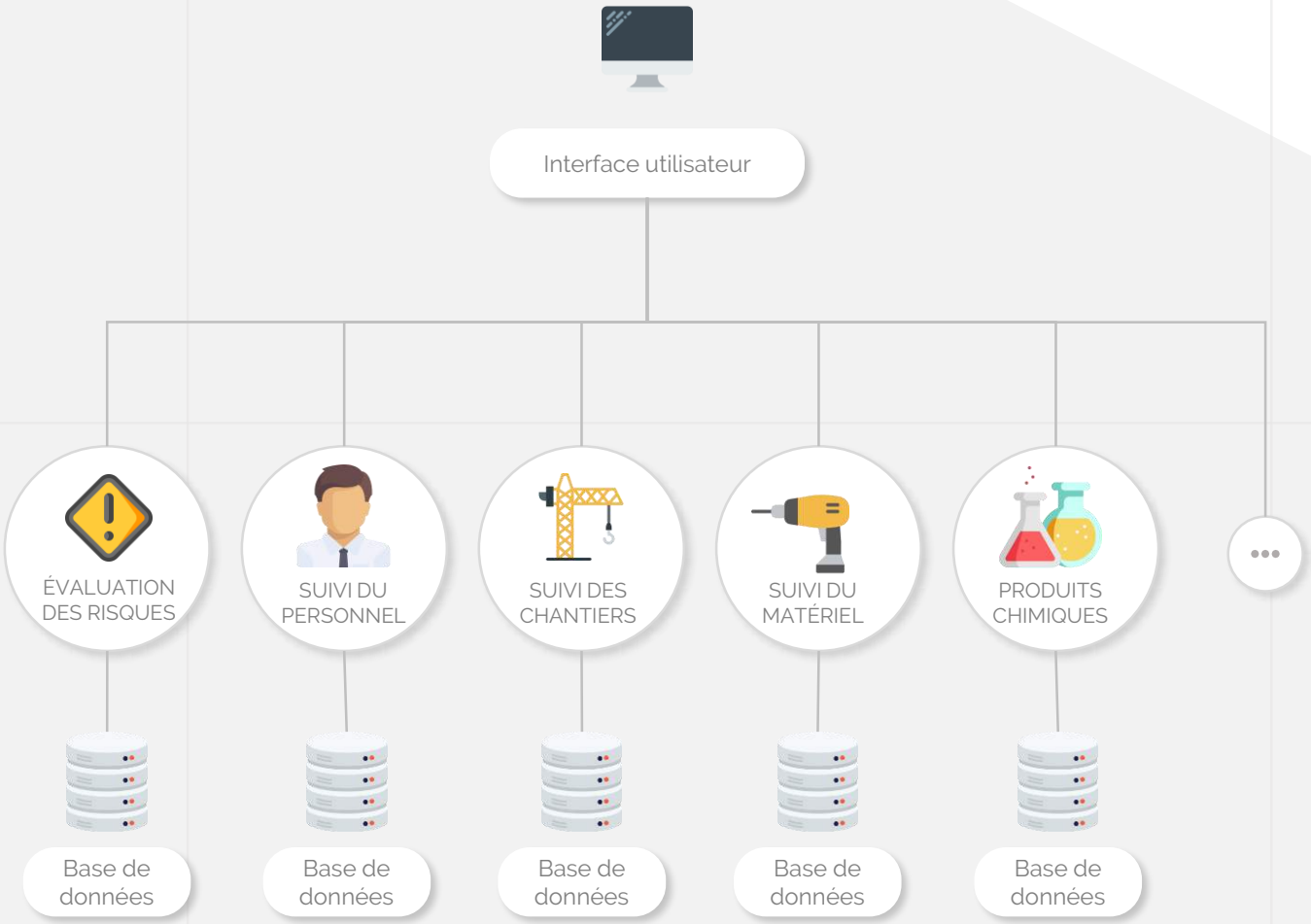
Notre préconisation

Notre préconisation

D'UNE ARCHITECTURE MONOLITHIQUE...

Notre préconisation

...À UNE ARCHITECTURE MICROSERVICES



Notre préconisation

UNE ARCHITECTURE MICROSERVICES

Face à la variété fonctionnelle de chaque brique composant la plateforme Web de l'OPPBTP, de leur articulation et de leur complexité, la mise en place d'une architecture en microservice a été préconisée et adoptée. C'est-à-dire la décomposition d'une application « massive » en **un ensemble d'applications spécialisées et indépendantes** communicant entre elles par un protocole d'échange.

Afin de rendre chaque processus du microservice les plus indépendants possible, les couches MVC doivent également être physiquement cloisonnées. Ceci sera réalisé en séparant la couche de présentation (l'interface utilisateur) grâce à un framework Javascript tandis que la couche métier et d'accès aux données sera réalisée grâce à un moteur d'API. Le tout réuni forme une application Web.

Une refonte complète de chaque outil plutôt qu'une migration au sens propre est ainsi préconisée afin de répondre conjointement aux besoins techniques d'une telle architecture ainsi qu'aux besoins des utilisateurs qu'ils soient ergonomiques, métiers ou issus d'une simple envie de modernité.

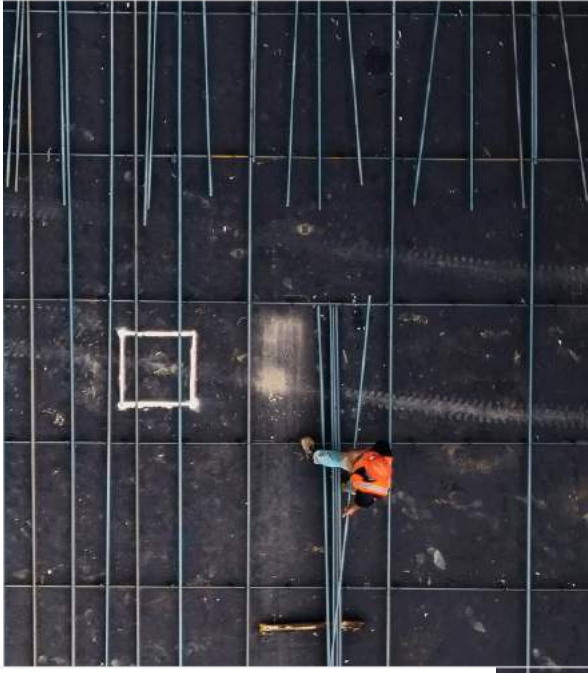
AVANTAGES

- **Indépendance technologique** : grâce aux échanges de données uniquement basées via les API exposées.
- **Réutilisabilité** : la séparation du front et du back permet d'envisager la mise en place de futurs besoins (application mobile, site partenaire…).
- **Time-to-market réduit** : les développements d'outils peuvent être parallélisés et confiés à plusieurs prestataires de manière simultanée.
- **Dette technique maitrisée** : les migrations peuvent être envisagées à une fréquence plus élevée grâce à la réduction des dépendances et à la spécialisation des services. Les régressions apportées par la mise en place d'évolutions applicatives sont contrôlées.
- **Maitrise des coûts** : le dimensionnement serveur est ajusté aux besoins de chaque microservice, les montées en compétences sont plus rapides et les délais de migration sont réduits.
- **Répondre aux besoins des utilisateurs** : par une mise à disposition progressive sans coupure de service.

#3

Mise en oeuvre

Mise en oeuvre



L'architecture microservice doit être validée par un cas concret de migration. L'outil d'évaluation des risques (EVR), cœur de l'offre de service de l'OPPBTP a été choisi pour être la première application à être réécrite dans ce contexte. Ce choix a été motivé pour plusieurs raisons :

- Son faible taux d'indépendance avec les autres services de la plateforme actuelle.
- Le fort besoin de modernisation remonté par les utilisateurs, couplé à des problèmes de performance et de stabilité.
- Sa place stratégique au sein du dispositif de services de l'OPPBTP. La satisfaction des utilisateurs de l'OPPBTP est fortement conditionnée par l'efficacité de cet outil.

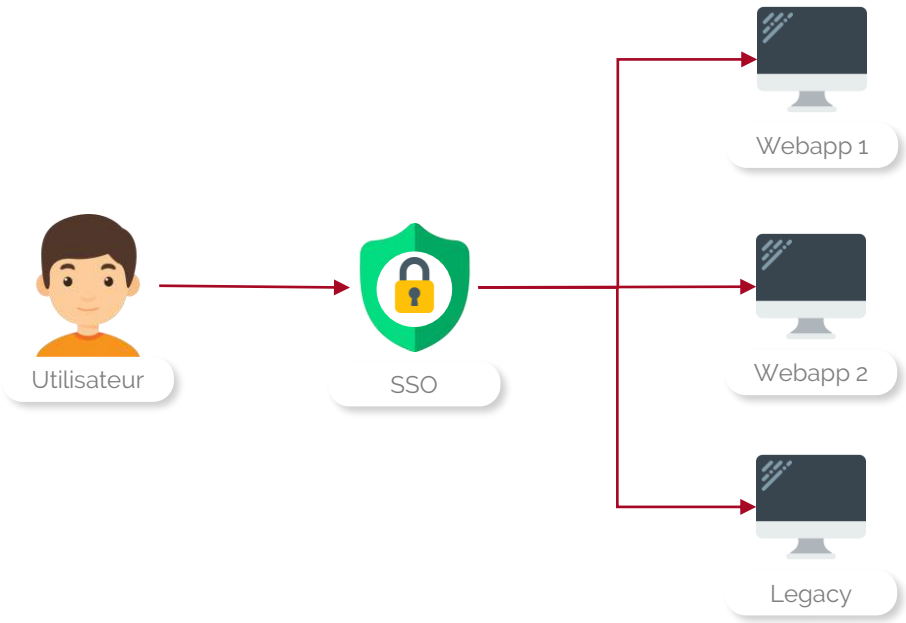
L'outil d'évaluation des risques propose l'élaboration du Document Unique (DU) ou Document Unique d'Evaluation Des Risques Professionnels (DUERP) dont la réalisation, parfois complexe, est une obligation réglementaire en application du Code du Travail.

Mise en oeuvre

L'AUTHENTIFICATION, UN POINT CRITIQUE

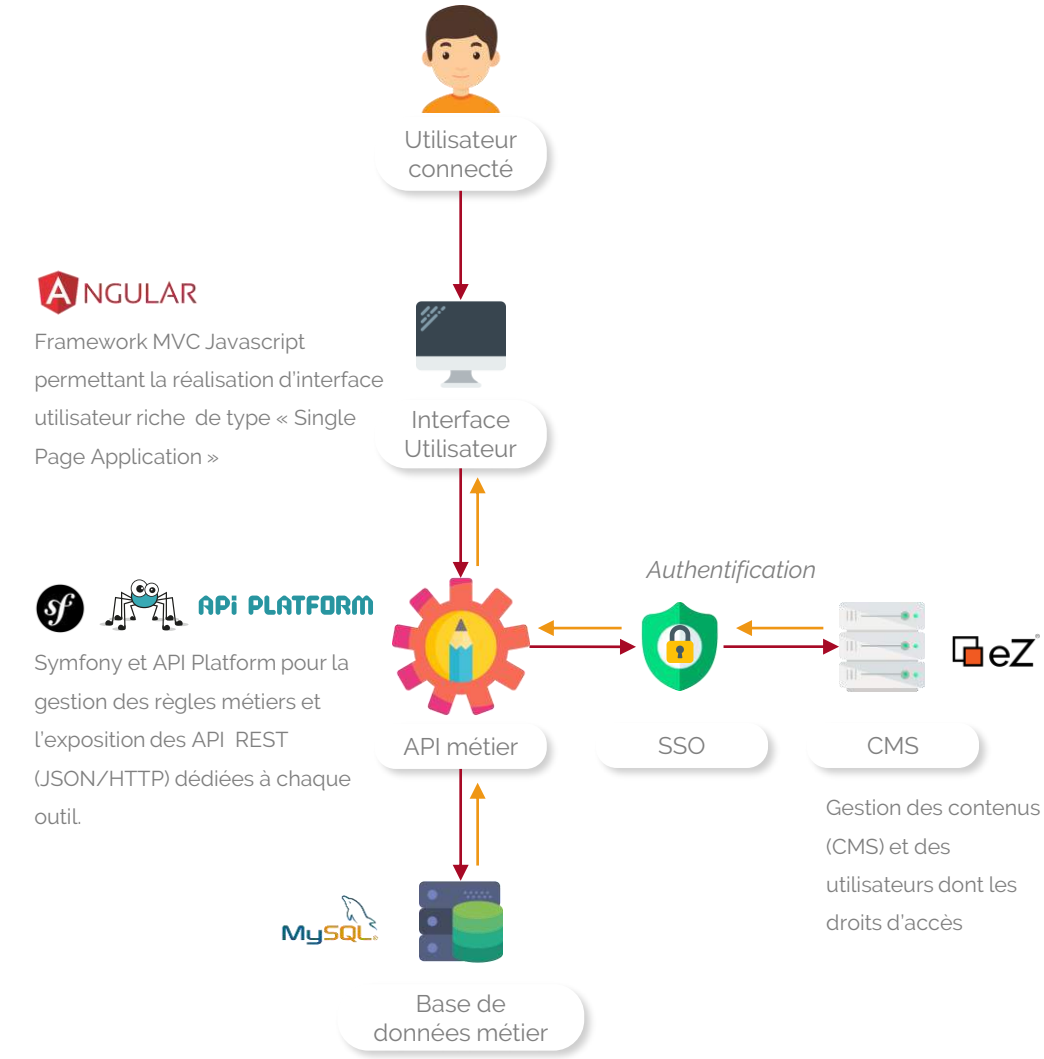
La mise en place d'une architecture microservice soulève des problématiques d'authentification. Les refontes de chaque outil allant s'opérer dans le temps, **l'architecture sera ainsi mixte entre les services disponibles sur l'ancienne plateforme Web et les nouveaux microservices créés**. L'expérience utilisateur pourrait être impactée négativement s'il s'avérait nécessaire de se connecter à chaque outil utilisé.

La mise en place d'un SSO (Single Sign-On) a été réalisée à cet effet. Il permet ainsi à un utilisateur de ne se connecter qu'une seule fois au sein de son espace, puis d'utiliser l'ensemble des services, qu'ils soient disponibles au sein de l'ancienne plateforme (nommée Legacy) ou bien sur un microservice migré.



Mise en oeuvre

L'ARCHITECTURE DÉTAILLÉE



Focus sur la migration des données de l'EVR

EXPERTISE ET PLAN DE MIGRATION DES DONNÉES

La migration des données de l'ancienne vers la nouvelle architecture est un point clé de la réussite du projet. Des millions d'enregistrements doivent être migrés d'une base de données unique vers les bases de données de chaque outil. Cette opération s'est déroulée en plusieurs temps :

01 **Définition/qualification des données.** Une première expertise a été réalisée afin de définir précisément le périmètre des données à migrer : localisation des données, identification des différentes relations, détermination de la volumétrie… Cette expertise a également permis de définir les stratégies de conversion des données de l'ancienne vers la nouvelle architecture de données.

02 **Extraction des données brutes.** Dans un second temps, toutes les données identifiées et qualifiées à l'étape 1 sont extraites telles quelles sans transformation ou redressement.

03 **Validation, consolidation et chargement des données.** Les données extraites à l'étape 2 ont ensuite fait l'objet de contrôles et de validation avant d'être chargées dans le nouveau modèle de données

04 **Contrôle de la reprise de données.** Un rapport de traitement a été automatiquement réalisé tout au long du processus d'extraction et de chargement des données. Ce rapport a permis d'identifier précisément les données conformes ainsi que les rejets de données. C'est-à-dire les données incohérentes ou non compatibles avec le nouveau modèle de données.

Compte tenu de la volumétrie et de la complexité des données l'utilisation de Talend pour ces opérations s'est avérée indispensable.

talend

Focus sur la refonte fonctionnelle de l'EVR

Le document d'évaluation des risques est complexe à rédiger et ce travail devait être facilité par l'application développée qui s'appuie sur plusieurs référentiels de données. Il devait ainsi être possible d'identifier automatiquement la liste des risques et des actions à mettre en place en fonction du ou des métiers de l'entreprise. La saisie en ligne se fait en 3 étapes :

01

Sélection du ou des métiers de l'entreprise, avec possibilité d'ajouter des éléments personnalisés au référentiel.

02

L'application propose ensuite une liste de risques associée aux métiers sélectionnés à l'étape 1. À nouveau l'entreprise a la possibilité de rajouter des risques personnalisés dans le référentiel.

03

Pour chaque risque identifié à l'étape 2, la plateforme propose à l'entreprise de mettre en place des actions pour prévenir ce risque. L'entreprise a également la possibilité de planifier ces actions (rappels et notifications mails).

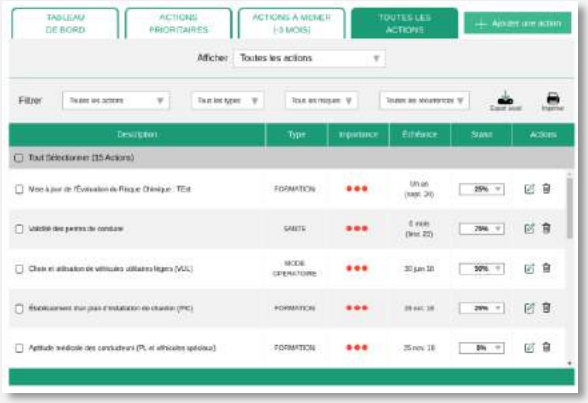
Retour d'expérience

Les architectures microservices apportent de nombreux avantages, néanmoins elles ne sont pas exemptes de contraintes et de points de vigilances.

L'AUGMENTATION DU NOMBRE D'APPELS AU SERVEUR

Dans une application monolithique (MVC classique) le Modèle, la Vue et le Controller échangent des informations côté serveur. Ces échanges permettent de construire la réponse demandée par un utilisateur. Généralement, un seul appel est nécessaire pour construire tous les éléments d'une page de l'application (affichage d'un tableau d'éléments, des modalités de sélection - zone de liste…).

Dans une application microservice où les couches MVC sont physiquement séparées, la couche de présentation est assurée par le framework Javascript tandis que la couche métier et d'accès aux données est réalisée grâce à un moteur d'API. Dans ce contexte, plusieurs appels seront nécessaires pour charger tous les éléments de la même page.



ARCHITECTURE MONOLITHIQUE

Un seul appel serveur construit l'ensemble de la page.

- http://web.app/toutes-les-actions

ARCHITECTURE MICROSERVICE

7 appels indépendants au serveur construisent chaque élément de la page

- GET http://api/liste-zone-de-liste- 1
- GET http://api/liste-zone-de-liste-2
- GET http://api/liste-zone-de-liste- 3
- GET http://api/liste-zone-de-liste-4
- GET http://api/liste-zone-de-liste-5
- GET http://api/liste-zone-de-liste-6
- GET http://api/listes-toutes-actions

Cette contrainte ne doit pas être négligée au moment de la définition de l'architecture serveur (augmentation du risque de congestion du réseau).

Retour d'expérience

LA STABILITÉ DE LA CONNEXION INTERNET DES CLIENTS

Les échanges de données entre le poste client utilisateur et le serveur sont grandement multipliés. L'expérience utilisateur sera fortement pénalisée par l'instabilité d'une connexion Internet. Celle-ci pourrait notamment entrainer :

- Des **chargements de données infinis** (perte de la connexion au moment d'un appel au serveur).
- De nombreuse**s déconnexions** de l'utilisateur de l'application.
- Une **altération de l'intégrité des données** en cas d'enregistrement partiel des données.

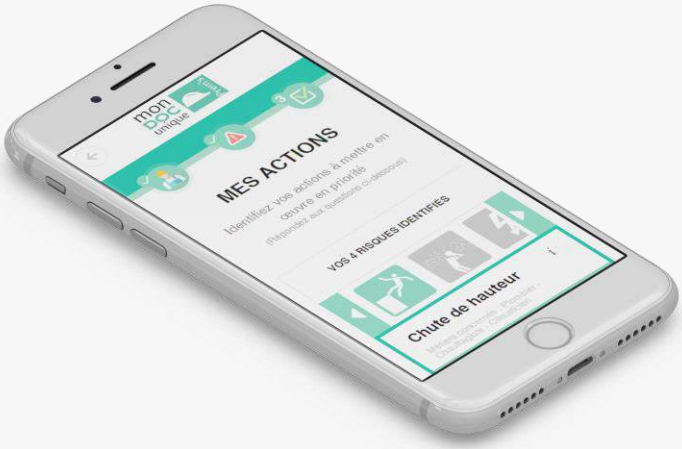
Des **bonnes pratiques et solutions** existent pour réduire ces risques :

- Gestion des codes d'erreur HTTP pour éviter les chargements infinies.
- Persistance de la connexion au sein d'un système SSO afin de ne pas déconnecter l'utilisateur.
- Enregistrement hors connexion.
- …

Ce risque doit être pris en considération lors de la définition d'une solution technique compatible avec l'utilisateur cible.

Résultats obtenus

- 1 000 nouvelles entreprises sont créées tous les mois pour 60 000 entreprises utilisatrices
- 500 connexions par jour
- 70 000 évaluations des risques enregistrées dans l'application.
- 500 000 actions de prévention créées.
- Un référentiel de données contenant plus de 2 millions d'entrées entre métiers, situations de travail et risques associés.
- Plus de 100 de millions d'enregistrements en base de données.
- Une application accessible sur tous les devices.



À propos de Codéin

Située à Montpellier, Codéin est une agence conseil en ingénierie digitale open source. Spécialisés dans l'architecture et la structuration de vos projets digitaux et applications métiers (WebApp), nos experts vous accompagnent dans le traitement de vos problématiques techniques complexes (multisites, multilingues) Mais pas que.

Codéin, c'est aussi des conseils en marketing et UX Design, pour une vision à 360°. Une agence web qui se tient à vos côtés pour réussir votre transformation digitale et activer votre e-business !

codein.

Digital **addict**

Espace French Tech, 1 place Francis Ponge, 34000 Montpellier | 09 72 42 26 03 | marketing@codein.fr