

Intégration de WASU dans WAF

Proposition 1 : Intégration par API (Mode Service) sans migration de l'environnement vue 2 en vue 3

Description

- **WAF** consomme directement les services de l'API de **WASU**.

Étapes

1. **Authentification** :
 - Configurer un mécanisme sécurisé (OAuth 2.0, tokens API).
2. **Appels API** :
 - Identifier les points d'intégration (gestion des utilisateurs, rôles, permissions).
 - Implémenter les requêtes entre WAF et l'API de WASU.
3. **Interface utilisateur** :
 - Intégrer les interfaces de WASU dans WAF en utilisant **vue-compact**
4. **Tests et Documentation** :
 - Réaliser des tests d'intégration.
 - Documenter les appels API.

Avantages

- WASU reste autonome.
- Faible couplage entre WAF et WASU.

Inconvénients

- Performance impactée par les appels réseau.
- Maintenance des dépendances API.
- Complexité accrue
- Incompatibilité de certaines fonctionnalités et ou plugins
- Efforts de test supplémentaires
- Risque de dette technique

Proposition 2 : Intégration via une Bibliothèque (SDK)

Description

- Transformer WASU en une bibliothèque PHP/JS (SDK) réutilisable dans WAF.

Étapes

1. **Création du SDK :**
 - Extraire les services principaux (gestion des rôles, permissions).
2. **Publication :**
 - Publier le SDK sur Composer (PHP) et/ou NPM (JS).
3. **Intégration dans WAF :**
 - Importer le SDK et l'utiliser directement.
4. **Mises à jour :**
 - Synchroniser les évolutions entre WASU et le SDK.

Avantages

- Performances élevées.
- Réduction des dépendances.

Inconvénients

- Maintenance accrue si WASU évolue rapidement.
- Moins de flexibilité.
- Conflits de versions de Vue
- Problèmes avec la réactivité

Proposition 3 : Intégration par API (Mode Service) avec migration partielle du front vers Vue 3

Description

- Migrer progressivement le front de WAF de Vue 2 vers Vue 3 pour réutiliser les composants de WASU.
- **WAF** consomme directement les services de l'API de **WASU**.

Étapes

1. **Audit** :
 - Identifier la complexité de migration de Vue 2 à Vue 3.
2. **Authentification** :
 - Configurer un mécanisme sécurisé (OAuth 2.0, tokens API).
3. **Appels API** :
 - Identifier les points d'intégration (gestion des utilisateurs, rôles, permissions).
 - Implémenter les requêtes entre WAF et l'API de WASU.
4. **Intégration des Composants WASU** :
 - Réutiliser les composants front Vue 3 de WASU directement dans WAF.
5. **Tests** :
 - Tester l'ensemble pour garantir la compatibilité.

Avantages

- Uniformité de l'interface utilisateur.
- Réutilisation du code front.

Inconvénients

- Migration potentiellement longue.
- Efforts techniques élevés.