

FICHE : PoC & CHOIX TECHNOLOGIQUES

=> Après avoir rempli cette fiche :convertir ce document en PDF et le téléverser dans votre dépôt Github dans un répertoire nommé 'doc'.

COMPTE RENDU DE LA VEILLE TECHNOLOGIQUE PoC

QUESTIONS DE RECHERCHE SUR LA PoC

L'opération la plus risquée du projet est de rendre l'application disponible sur plusieurs OS différents : il pourrait y avoir des incompatibilités.

Je me demande si des problèmes de compatibilité entre les différentes plateformes arrivent souvent. En tout cas, avec mes recherches je n'en ai pas trouvé d'important.

POC = PREUVE DE CONCEPT

Pour vérifier qu'il n'y a pas de problème, il faudrait tester chaque fonctionnalité de l'application sur les différentes plateformes cibles (Windows en priorité, puis Linux et enfin macOS). Si tout va bien, alors à priori il n'y a pas de soucis de compatibilité. Sinon il faut se documenter sur les potentiels correctifs à effectuer.

LES MARQUE-PAGES IDENTIFIÉS LORS DE VOS RECHERCHES

<https://raindrop.io/ClementDrn/net-22860221>

LES PREUVES DE CONCEPT

Pour chaque preuve de concept réalisée : identifier le but de la preuve de concept (ce qu'elle vérifie), le lien vers le sous-répertoire de votre dépôt GitHub qui contient le code de la preuve de concept ainsi que les résultats de votre expérimentation, puis, finalement, vos conclusions.

Au moins une preuve de concept doit être documentée et réalisée.

PREMIÈRE POC RÉALISÉE

Preuve : (Que prouve la Poc)

URL Github :

EXPLIQUEZ VOTRE Poc

Décrivez la Poc en détails.

Que PROUVE la Poc ?

Que reste-t-il à prouver ?

Quels sont vos résultats de la Poc ?

PREMIÈRE TECHNOLOGIE SÉLECTIONNÉE (LA NOUVELLE)

Technologie : Avalonia

URL : <https://avaloniaui.net/>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Avalonia est certes très récent, ce framework est très prometteur. Il est tenu à jour. Il permet de créer des applications pour plein de plateformes différentes : Windows, macOS, Linux, Android et iOS (en cours).

De plus, il permet d'utiliser le langage C# (avec XAML) et aimant le C et le C++, j'ai favorisé ce framework

Grâce à cette solution j'ai la possibilité de créer une application de bureau moderne (contrairement à Swing) pour plusieurs plateformes sans avoir à créer une appli web (contrairement à Electron par exemple).

[voir tableau comparatif Avalonia/Swing]

C'est pourquoi j'ai choisi la sérialisation en fichiers JSON.

DEUXIÈME TECHNOLOGIE SÉLECTIONNÉE (LA CONNUE)

Technologie : JSON

URL : <https://www.json.org/>

JUSTIFIER VOTRE CHOIX TECHNOLOGIQUE POUR CETTE TECHNOLOGIE

Cette technologie est supportée par C# concernant la sérialisation et est conseillée par Microsoft. Je n'ai pas besoin d'une base de données : il n'y a pas de relations entre plusieurs objets mais plus des compositions et appartenances.

De plus, la sérialisation en JSON est plus sécurisée que la sérialisation en binaire lorsqu'on utilise les librairies par défaut de C#.

[voir tableau comparatif JSON/binaire]

C'est pourquoi j'ai choisi la sérialisation en fichiers JSON.