Experimental protocol (EN)

Tool: http://onto-tool.blade-blockchain.eu/

Introduction

Context: we are working on a project to develop an ontology of blockchain software patterns, based on previous results obtained during a systematic literature review. Through this ontology, we wish to formalize the knowledge obtained but also to build an ontology-based software pattern recommendation tool to assist architects in the selection of blockchain patterns. We are in the validation phase of the tool.

H_1 : A practitioner can leverage the tool to navigate from the solution space (blockchain-based patterns), to the problem space (requirements).
H_2 : A practitioner can leverage the tool to navigate from the problem space (requirements), to the solution space (relevant blockchain-based patterns).
H_3 : A practitioner can leverage the tool to design blockchain applications.

Process: This experimentation will take place in four steps:
1. The objective will be, for a given case study, to see if it is easy for the people trying the tool to tell or not to what degree the pattern is useful in the design of the solution answering the need of the case study.
2. The objective will be, for a given case study, to let the participants select 5 patterns that they think are appropriate for the design of the solution meeting the need of the case study.
3. This part concludes the vocal interview by allowing the participants to exchange with the research team (feedback, questions, ...).
4. A form will be sent to the participants after this study to evaluate the participants' opinion towards the tool itself (usability, relevance, ...).

Attention: when carrying out part 2, you will have to save the result obtained after recommendation as well as the 5 selected patterns by using the Export option, respectively available on the Recommendation and Patterns tab.

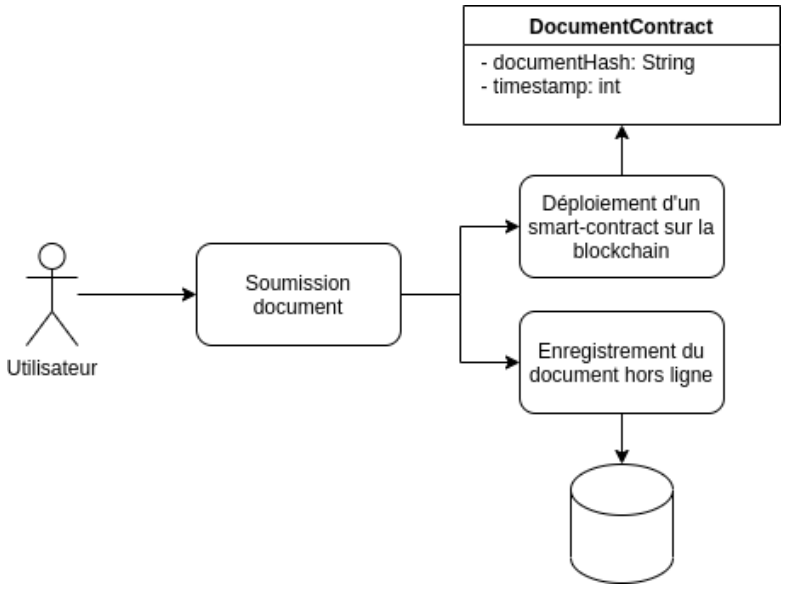
First part

Context

In this part, we consider a "Soleau envelope" type application, allowing to prove the existence of a document at a given date without revealing its content.

The operating mode of the solution is as follows: participants can register on the web platform by filling in their personal information and their public address linked to their blockchain wallet (example: Metamask). A participant can then upload a document on the platform. Two operations follow: a smart-contract is created on the blockchain containing a hash of the document and its metadata (size, timestamp), and the document is saved on an off-chain server for permanent storage. It is the responsibility of the participant to sign the transaction allowing the deployment of the smart-contract via his blockchain wallet.

Following this, it is possible to verify the proof of existence of the document later in time, by resubmitting it to the platform.



Questions

Below are four software patterns. For each of them, give a score from 1 (not applicable) to 5 (extremely applicable) on the relevance of applying them to the given study case, using the Explore tab of the tool to judge this.

- ➔ Contract registry
- ➔ Off-chain data storage pattern
- ➔ Oracle pattern
- ➔ Factory contract pattern

Second part

Context

The second application studied in this survey is a decentralized lending application based on the Ethereum blockchain. (note: the operation described here is very simplified for this survey)

For this, a smart-contract is deployed in the blockchain and contains cryptocurrencies. It is possible for a user to borrow cryptocurrency by depositing a larger amount of cryptocurrency as collateral. In order to get their collateral back, the user must then repay the loan. If the value of the collateral ever drops below a certain threshold, it is sold to cover the value of the loan. In order to provide the cryptocurrencies needed for such loans, other users can deposit their cryptocurrency on the smart-contract. In exchange, they will receive a portion of the fees collected by the smart-contract during loans.

Everything is done via a web platform, connected to a blockchain node to track users' transactions. The latter will use a blockchain wallet such as Metamask to sign the transactions.

As the application is deployed on the Ethereum mainnet, it must be efficient to avoid high transaction costs. Also, since the application handles large amounts of cryptocurrencies, the smart-contract must be secure enough to prevent the exploitation of flaws and vulnerabilities. Finally, in order for the smart-contract to know the value of the stored assets, it will need to receive data from off-blockchain services (e.g. cryptocurrency exchange API)

Objective

Use the Recommendation tab on the tool to select 5 patterns that you feel are the most relevant for designing an application that satisfies the requirements and context provided.

Protocole d'expérimentation (FR)

Outil : http://onto-tool.blade-blockchain.eu/

Introduction

Contexte : nous travaillons sur un projet visant à développer une ontologie de software patterns blockchain, à partir de précédents résultats obtenus lors d'une revue de littérature systématique. Par cette ontologie, nous souhaitons formaliser ces connaissances obtenues mais aussi construire un outil de recommandation de patterns logiciels basé sur l'ontologie pour assister les architectes dans la sélection de patterns blockchain. Nous en sommes à la phase de validation de l'outil.

H_1 : Un utilisateur peut utiliser l'outil pour naviguer de l'espace des solutions (patterns basés sur la blockchain) à l'espace des problèmes (exigences).

H_2 : Un utilisateur peut tirer parti de l'outil pour naviguer de l'espace du problème (exigences) à l'espace de la solution (patterns pertinents basés sur la blockchain).

H_3 : Un utilisateur peut tirer parti de l'outil pour concevoir des applications blockchain.

Déroulement : Cette expérimentation se déroulera en quatre étapes :

- 1. L'objectif sera, pour un cas d'étude donné, de voir si il est facile pour les personnes essayant l'outil de dire ou non à quel degré le pattern est utile dans le design de la solution répondant au besoin du cas d'étude.
- 2. L'objectif sera, pour un cas d'étude donné, de laisser les participants sélectionner 5 patterns qui leur semblent convenir au design de la solution répondant au besoin du cas d'étude.
- 3. Cette partie conclut l'entretien en vocal en permettant aux participants d'échanger avec l'équipe de recherche (feedback, questions, …).
- 4. Un formulaire sera envoyé après cette étude aux participants pour évaluer l'opinion des participants envers l'outil en lui-même (utilisabilité, pertinence, …).

Attention : lors de la réalisation de la partie 2, il faudra enregistrer le résultat obtenu après recommandation ainsi que les 5 patterns sélectionnés en se servant de l'option Export, respectivement disponible sur l'onglet *Recommendation* et *Patterns*.

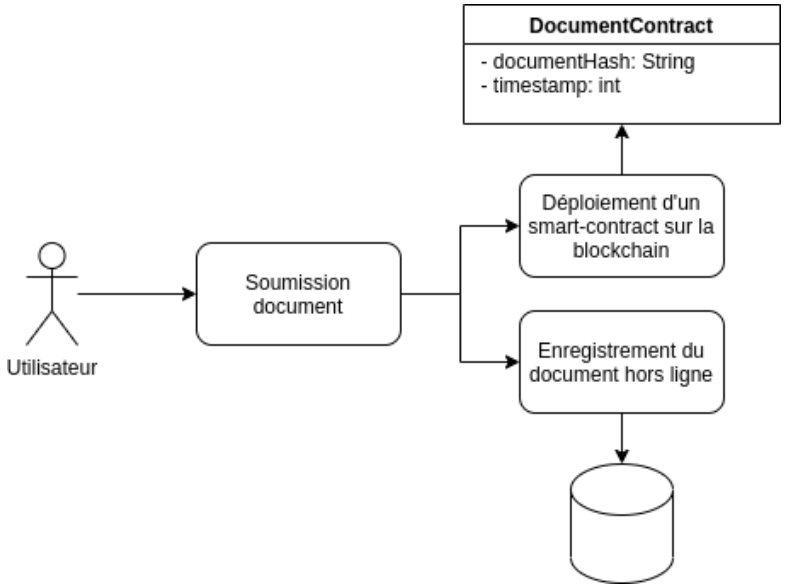
Première partie

Contexte

Dans cette partie, nous considérons une application type “enveloppe Soleau”, permettant de prouver l’existence d’un document à une date donnée sans pour autant en révéler son contenu.

Le mode opératoire de la solution est le suivant : les participants peuvent s’inscrire sur la plateforme web en renseignant leurs informations personnelles ainsi que leur adresse publique liée à leur portefeuille blockchain (exemple: Metamask). Un participant peut ensuite déposer un document sur la plateforme. S’en suit deux opérations: un smart-contract est créé sur la blockchain contenant un hash du document et ses métadonnées (taille, timestamp), et le document est sauvegardé sur un serveur hors-chaine pour stockage permanent. Il est de la responsabilité du participant de signer la transaction permettant le déploiement du smart-contract via son portefeuille blockchain.

Suite à cela, il est possible de vérifier la preuve d’existence du document plus tard dans le temps, en le soumettant à nouveau sur la plateforme.



Questions

Vous trouverez ci-dessous quatre patterns logiciels. Pour chacun d’entre eux, donner une note allant de 1 (non applicable) à 5 (extrêmement applicable) sur la pertinence de les appliquer au cas d’étude donné, en utilisant l’onglet *Explore* de l’outil pour juger de cela.

- ➔ Contract registry
- ➔ Off-chain data storage pattern
- ➔ Oracle pattern
- ➔ Factory contract pattern

Seconde partie

Contexte

La seconde application étudiée dans cette enquête est une application de prêt décentralisée basée sur la blockchain Ethereum. *(note : le fonctionnement décrit ici est très simplifié pour cette enquête)*

Pour cela, un smart-contract est déployé dans la blockchain et contient des cryptomonnaies. Il est possible pour un utilisateur d'emprunter de la cryptomonnaie en déposant une montant supérieur en cryptomonnaie en collatéral. Pour récupérer son collatéral, l'utilisateur doit donc rembourser le prêt. Si jamais la valeur du collatéral baisse en dessous d'un certain seuil, il est vendu pour couvrir la valeur du prêt. Afin de fournir les cryptomonnaies nécessaires à de tels prêts, d'autres utilisateurs peuvent déposer leur cryptomonnaie sur le smart-contract. En échange, ils recevront une partie des frais collectés par le smart-contract lors de prêts.

Tout se fait via une plateforme web, connectée à un nœud blockchain pour faire suivre les transactions des utilisateurs. Ces derniers utiliseront un portefeuille blockchain tel que Metamask pour signer les transactions.

L'application étant déployée sur le mainnet Ethereum, elle doit être efficiente pour éviter de trop grands coûts de transactions. Aussi, l'application manipulant de larges montants de cryptomonnaies, le smart-contract doit être suffisamment sécurisé pour prévenir de l'exploitation de failles et vulnérabilités. Enfin, afin que le smart-contract puisse savoir la valeur des actifs stockés, il lui faudra recevoir des données provenant de services hors-blockchain (eg. API des cours cryptomonnaies)

Objectif

Utiliser l'onglet *Recommendation* sur l'outil pour sélectionner 5 patterns vous semblant être les plus pertinents pour le design d'une application satisfaisant les exigences et le contexte fourni.