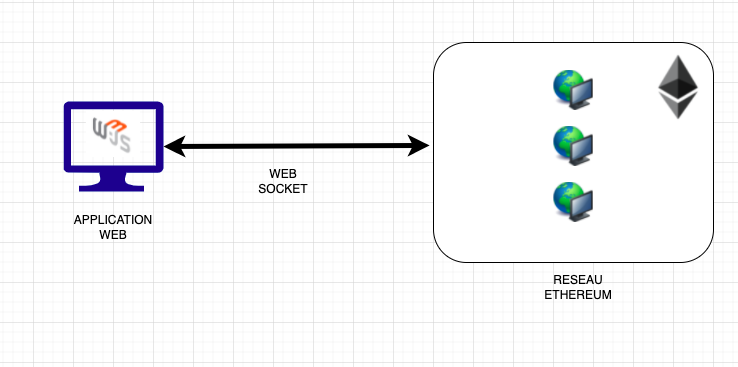
**DOCUMENTATION TECHNIQUE 5BLOC**

**1 – Architecture**



**2- Mise en place et outils techniques**

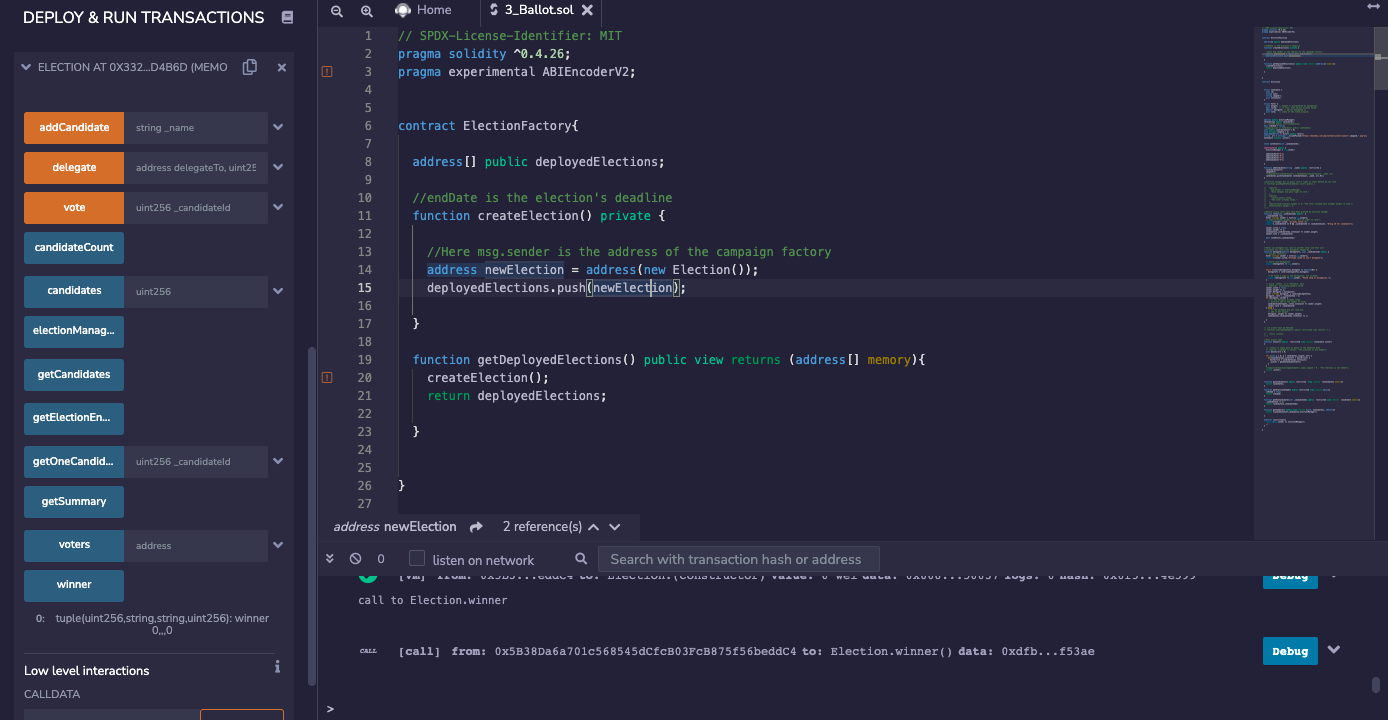
**La realisation d’une application de vote à l’aide de la technologie ethereum, avec plus precisément les outils:**

* **Go-Ethereum (Pour la creation d’un réseau local et la création des noeuds qui héberge les blocks)**
* **Web3JS/ReactJS technologies front-end pour la mise en interface des données provenant du reseau local, et aussi l’interaction (transactions) entre le client et la blockchain**
* **Solidity/Remix language de programmation et IDE servant à creer les des contrats intelligents et à ensuite les compiler et deployer dans le réseau ethereum**

**3- Mise en place du projet**

1. **Remix/Solidity**

**Tout d’abord il faut mettre en place un contrat intelligent sur la base duquel toute les opérations et montages du projets doivent se faire, la première étape est de le creer (Contrat intelligent dans l’IDE REMIX):**



**Le code est disponible dans le projet attaché à cette documentation.**

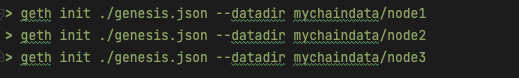
1. **Go-Ethereum**

Comme dans toute application web il faut un fournisseur de données pour échanger les données, et rendre l’application interactive raison pour laquelle il est nécéssaire d’utiliser un outils comme Go-ethereum qui nous fournit un ensemble de commandes pour la création du réseau et des noeuds qui permettront le traitement des composants comme les comptes, les adresses etc... , comme ceux-ci:

Avant toute chose il faut creer le reseau local qui abritera tout les éléments constitutifs de notre blockchain comme le montre la commande ci-dessous:



Et passer la suite.

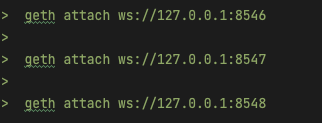


Il s’agit des commandes de création des noeuds de notre architecture nous avons trois noeuds qui sont activé grace aux commandes:



Vous ne pourrez pas voir tous le long des commandes mais il y a aussi un fichier attaché à cette documentation avec toutes les commandes.

Après avoir monté les noeuds il faut s’assurer de les attacher ou plus clairement les inscrire dans le reseau grace aux commandes:



Et ensuite s’assurer que les noeudssont liés en connection pair à pair:



Avec cette commande dans un terminal pour chaque noeud on s’assure que les noeuds sont liées en pair à pair pour les traitements et echanges, dans le fichier note.md vous avez les details de comment lier ces noeuds et les differents traitements de minage, d’aprovisionnement des comptes qu’il faut d’abord créer.

1. **Web3JS/ReactJS**

Nous avons maintenant pour la creation de l’interface client, WEB3JS et REACTJS qui permettront de creer une application front-end pour visualiser les données suite aux échanges avec le réseau ethereum.

Il faut d’abord creer le repertoire du projet et installer les depedances ci-dessous:



Notamment:

* Mocha - pour les test unitaires
* Le pack react pour les composants de l’application web
* Semantic-UI pour styliser l’interface
* Truffle pour la gestion du wallet(Nous ne l’avons pas vraiment utilisé)

1. **Conclusion**

Suite à l’execution de ces commandes nous avons materialiser le projet que vous trouverez attaché a ce document.