ARTEFACT

ArtefHack

BlockChain et Marketing éthique

Réseau WIFI



Connectez-vous à : Artefact

Mot de passe : WeHaveWifi

Rejoignez le discord si ce n'est pas encore fait : https://discord.gg/N2nc4J9

Agenda Artef'Hack!

```
17h30 - 18h10 : finalisation des équipes
18h10 - 18h15 : mot de bienvenue de notre CTO
18h15 - 18h30 : présentation de l'épreuve
18h30 : publication du sujet sur le salon #sujet
18h30 - 19h : petit tutoriel solidity pour ceux qui le souhaitent
19h30 - 20h : arrivée des pizzas
22h30 : fin des soumissions
22h30 - 22h40 : résultats temporaires et mot de la fin
```

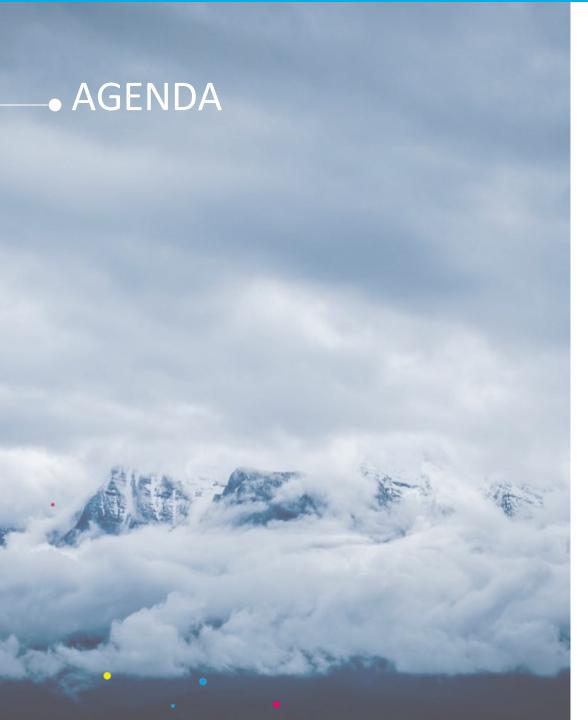
Présentation du sujet

Artefact explore l'avenir du digital marketing à travers la blockchain! La blockchain permet de rendre le web plus transparent sûr, et offre des opportunités d'application au digital marketing.

Le projet ArtefHack vise à établir un équilibre entre les différents acteurs du marché :

- l'Advertiser, dont le but est de partager son message à un public le plus large possible
- le Publisher, dont le but est de diffuser les contenus qu'il produit et d'être rémunéré pour
- le User, dont le but est de consommer des contenus de qualité

Commencez le projet en téléchargeant le .zip disponible sur discord



- 1. Structure globale du sujet
- 2. Zoom sur les fonctions à implémenter
- 3. Zoom sur les paiements
- 4. Test de votre implem et scores
- 5. Soumission de votre implem

1. Structure globale du projet

User

visite la plateforme et reçoit un message/contenu qu'il évalue

Légende :

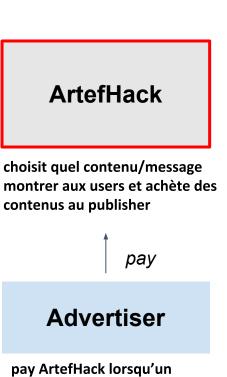
Contracts

Addresses

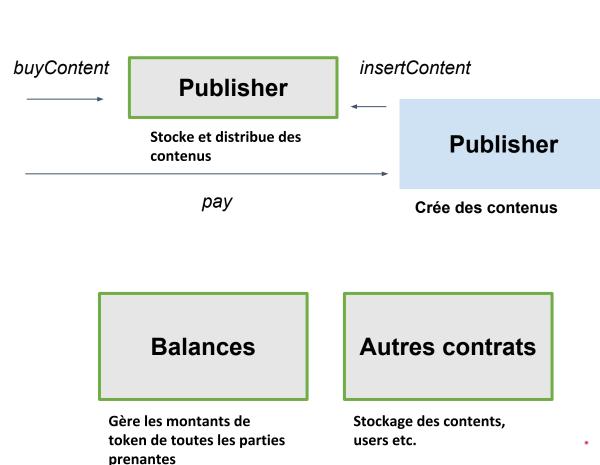
Implémentation par les participants

Donné au début de l'épreuve





message est montré



2. Zoom sur les fonctions à coder : visit et eval

ArtefHack

Content

0 <= preference < 100

qualification du type de content renvoyé

Message

bool message

impression d'un message publicitaire ou non visit : renvoie un content et un message

eval : le user renvoie sa réaction au content/message

if : user a déjà vu le content then : eval negative

if : user voit un message alors que ça le dérange then : eval negative

else: eval positive si |pref(content)
- pref(user)| < 10, négative sinon</pre>

User

0 < preference < 100 bool message



• 3. Zoom sur les paiements : échanges

Advertiser

L'advertiser dispose d'une balance en token élevée

L'advertiser paye ArtefHack un prix fixe pour chaque impression d'un message publicitaire



paiement

ArtefHack

Achète des contenus au Publisher

Reçoit des paiements de l'Advertiser



paiement

Publisher

mapping(bytes32 identifier, uint preference) buyContent(bytes32 identifier)

Stocke les contenus crées par le publisher

ArtefHack peut acheter des nouveaux contenus à montrer aux users

ArtefHack paye un prix fixe au Publisher lorsqu'il achète des contenus

Balances

mapping(user address => uint amount)
pay(user from, user to, uint amount)

Stocke les montants détenus par tous les participants

Gère les transferts entre participants

Libre à vous de mettre en place de nouveaux flux!

Légende :

Contracts

Addresses



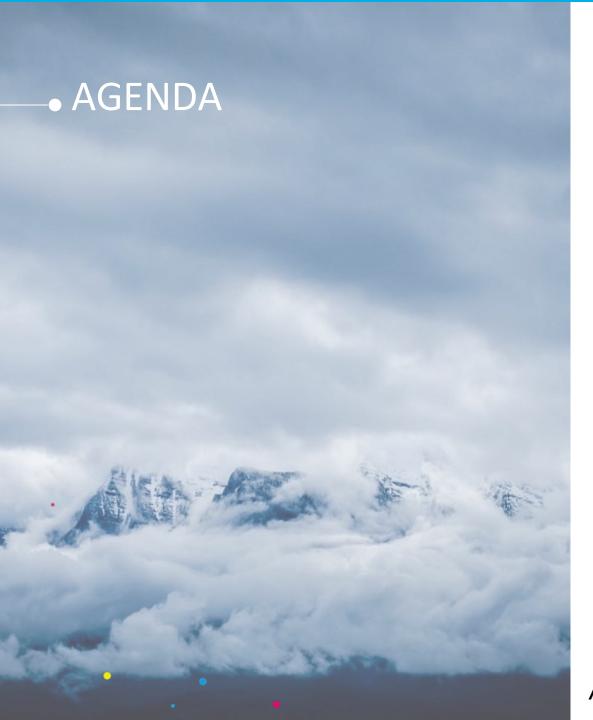
4. Simulation : test de votre implémentation et obtention de votre score

- Opérations pour tester la validité de votre implémentation:
 - lancer testrpc : testprc -a 504 -l 5000000
 - Déployer les contrats :
 - déploiement sur testrpc déjà lancé : truffle migrate
 - ne pas oublier de rm -rf build/contracts/ si vous avez déjà déployé auparavant et de relancer testrpc
 - Lancer la simulation :
 - depuis le dossier test/
 - activer le virtualenv python3 : source ~/.venv-py3/bin/activate
 - commande: python3 simulate.py --setup True --starting-users-count s --max-users-count m --contents-count c --days-count d
 - attention à mettre des valeurs faibles dans un premier temps pour que la simulation ne dure pas trop longtemps

Vous pouvez poser des questions sur le chat Discord, nous serons ravis d'y répondre!!

5. Soumissions des résultats

- Ne modifier que le fichier ArtefHack.sol
- Envoyer le fichier en message privé sur discord à Diplomate (Ulysse)
- Votre implémentation sera évaluée :
 - avec un jeu de données différent de celui que vous aurez eu (les paramètres seront identiques)
 - avec les autres contrats qui vous auront été donnés
- Les paramètres d'évaluation (on sommera les scores des 3 scénarios):
 - contents-count : 50 / 100 / 400
 - starting-users-count : 10 / 20 / 50
 - max-users-count: 10 / 50 / 300
 - days: 5 / 20 / 50
 - timeout: 1 minute / 10 minutes / 30 minutes



TIPS

Tips - Compilation

- restart testrpc && rm -rf build && truffle migrate
- pensez à modifier les migrations truffle le cas échéant
- impossible de retourner des variables de taille dynamique
- ne laissez pas de warning
- ne vous embêtez pas avec la visibilité de vos fonctions

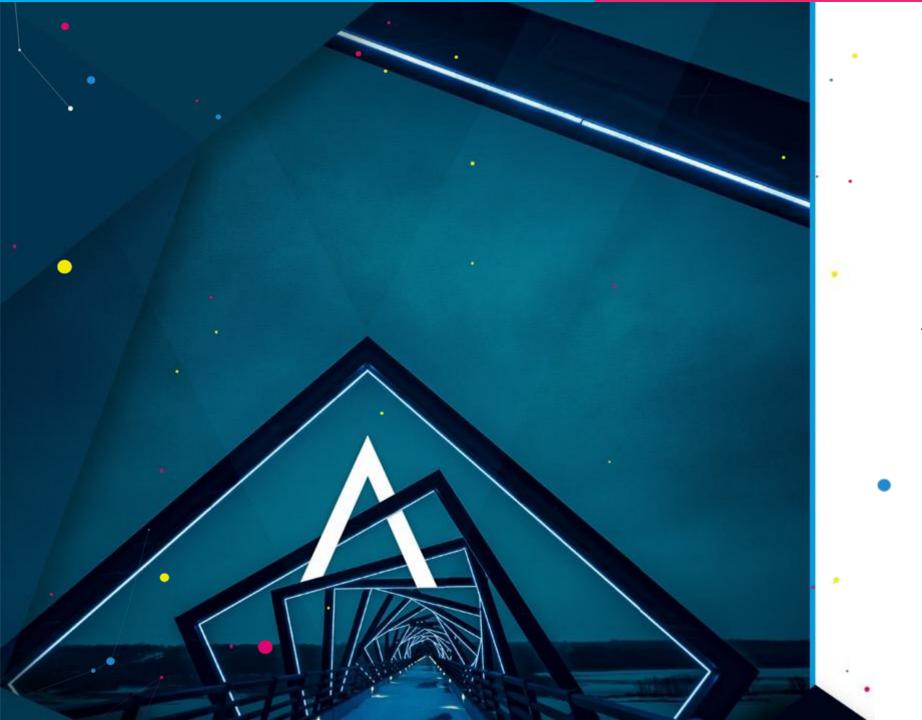
Tips - Structures de données

- N'hésitez pas à dédier des contrats entiers à vos structures de données
- Eviter les boucles for sur des index au maximum, préférez multiplier les mappings
- Les mappings: https://medium.com/@robhitchens/solidity-crud-part-1-824ffa69509a

Tips - Erreurs courantes

invalid opcode: tester l'appel de sa fonction "à la main":
 https://web3py.readthedocs.io/en/latest/

- not enough gas: où sont les boucles?!



ARTEFACT

Bon Courage :)