

Évaluer et intégrer des nouvelles technologies

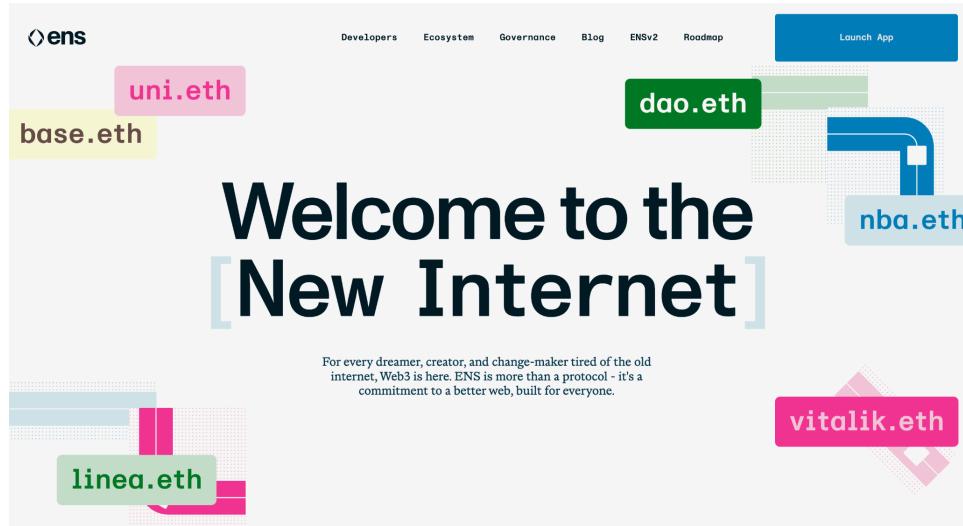
4 ensembles de technologies découverts

2 changements de technologie suite à des évaluations et tests

2 technologies extrêmement récentes intégrées & 1 autre en cours d'évaluation

Technologies découvertes & utilisées durant le développement

- [Ethereum Name Service](#) (communément appelé ENS), un système similaire au DNS qui permet d'utiliser des noms simples à mémoriser pour reconnaître des utilisateurs, plutôt qu'une adresse complexe (exemple: 0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045). Nous avons décidé d'utiliser ENS afin de simplifier les interactions de nos utilisateurs dans l'application.



Page d'accueil d'ENS

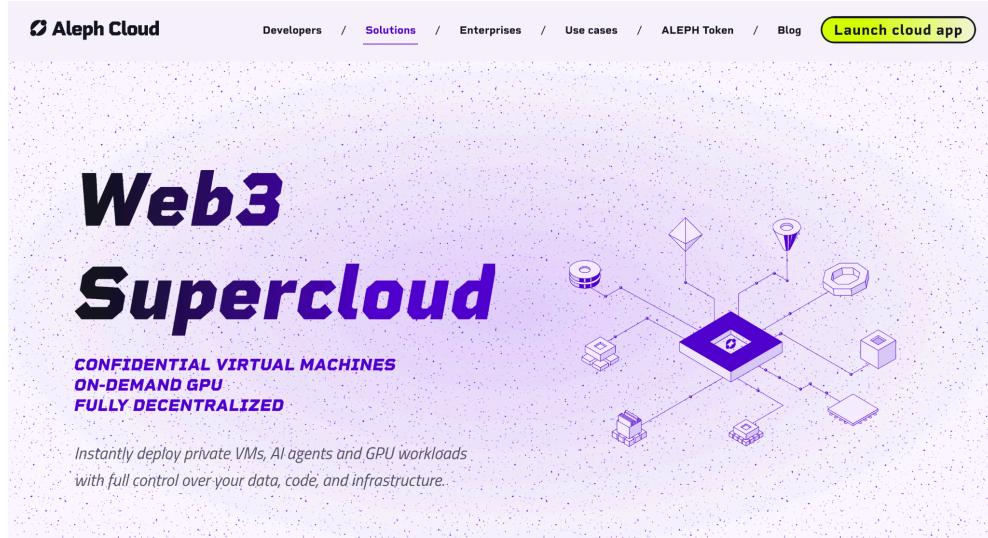
- [Solidity](#), langage de création de “smart contracts”, des programmes immutables sur la blockchain Ethereum et ses couches secondaires, que nous avons utilisé avec ENS pour modifier les smart contracts par défaut afin de les adapter à notre cas d’usage
- [Aleph Cloud](#), le cloud décentralisé sur lequel repose notre projet et qui se veut être une alternative aux géants comme [AWS](#), [GCP](#) ou [Azure](#). Nous avons utilisés diverses parties (pour certaines encore en beta et parfois instables) de ce cloud, y compris:
 - les [messages](#) qui permettent de faire transiter de la donnée entre les différents services et de stocker des informations sous forme de bases de

données clé-valeur (que nous utilisons pour stocker les données chiffrées des utilisateurs)

- les [fonctions](#), similaires aux célèbres [AWS Lambda](#) (sans pour autant en suivre le standard), qui permettent de faire tourner du code backend de façon décentralisée sur les plus de 600 nœuds de compute du réseau, évitant ainsi un contrôle centralisé ou une interruption de service suite à un incident isolé.

Nous utilisons 2 fonctions pour faire tourner des backend permettant la résolution et création de noms d'utilisateurs simplement (sans faire payer de frais à l'utilisateur), et pour gérer l'achat de crédits (ajout / retrait de crédits dans la balance, puisque ces opérations ne peuvent pas être réalisées en frontend de façon sécurisée).

- le [stockage de fichiers](#), permettant via de simples appels HTTP de stocker et récupérer des fichiers sur les plus de 100 nœuds contrôleur du réseau, utilisant [IPFS](#) pour gérer la communication pair à pair et la redondance. La composante centrale de Bedrock étant le stockage de fichiers, leur contenu (toujours chiffré évidemment) est ainsi stocké et récupéré sur le réseau Aleph.
- le [web hosting](#) qui permet de déployer des sites web de façon décentralisée avec le contenu stocké sur IPFS et la résolution DNS gérée par le réseau Aleph. Notre application (ainsi que des previews lors du développement et tests du projets) est déployée sur Aleph avec cette fonctionnalité (notamment grâce à un outil que nous avons créé, voir l'explication détaillée dans le KPI "Développer les contributions communautaires").

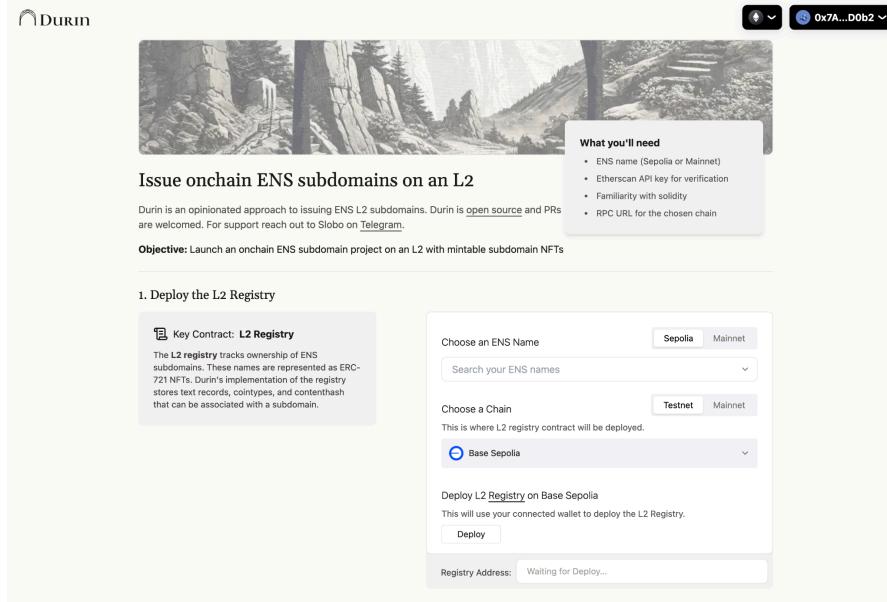


Page d'accueil d'Aleph Cloud

- [Privy](#) et [Thirdweb](#) pour la connexion de wallets sur notre application: nous avons découvert différentes librairies et kits de développement pour connecter des portefeuilles blockchain à des applications web ([viem](#), [wagmi](#), [ethers...](#)), mais pour simplifier l'expérience et autoriser la connection “social login” (connexion avec Google ou par email par exemple) nous avons exploré différentes solutions. Notre premier plan d'action avant de débuter le projet prévoyait d'utiliser [Web3Auth](#), cependant après plusieurs comparaisons et tests [Privy](#) s'est avéré plus adapté notamment sur la richesse du kit de développement afin de pouvoir l'intégrer avec celui d'Aleph le plus simplement possible. Cependant nous avons changé de solution en cours de route en passant à [Thirdweb](#) notamment suite à un test à [un hackathon](#) montrant sa supériorité sur les produits autour de la connexion du portefeuille, comme l'implémentation d'un système de paiement pour l'achat de crédit sur notre application sans utiliser une autre solution supplémentaire, grâce à leur produit [Universal Bridge](#).

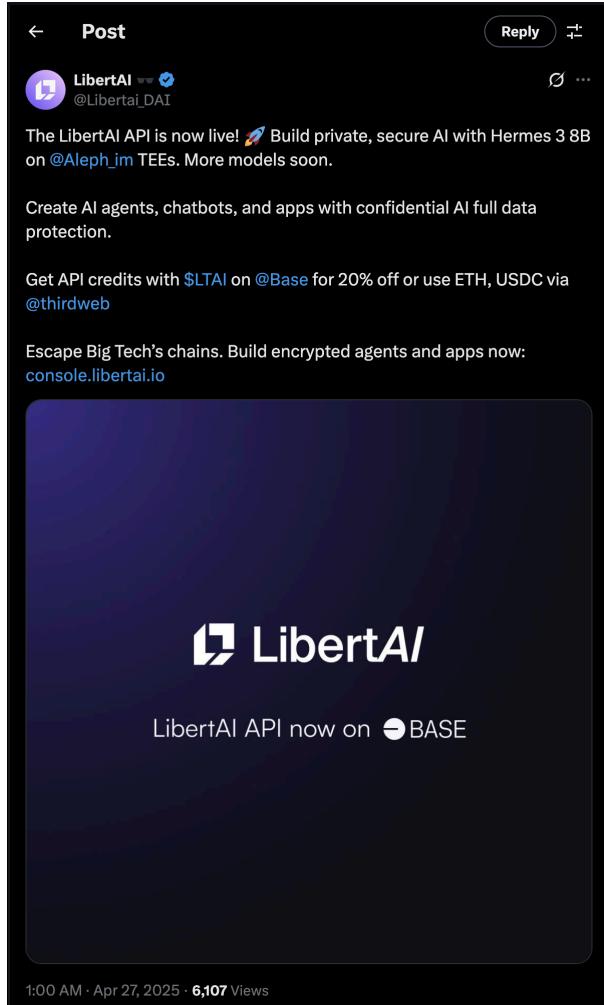
Technologies extrêmement récentes (< 1 an) & émergentes utilisées

- Intégration des “subnames” d’ENS (Ethereum Name Service, présenté plus haut), qui ont un fonctionnement similaire à des sous domaines et qui nous servent à stocker les noms d’utilisateurs sur la blockchain, directement liés avec leur adresse, et permettant aux utilisateurs d’interagir entre eux (se partager des fichiers par exemple) uniquement avec le nom d’utilisateur au lieu d’informations complexe (adresse, clé publique...). Les smart contracts permettant de gérer ces utilisateurs ont été créés avec <https://durin.dev>, un outil pour créer des subnames sur des couches secondaires d’Ethereum (également appelés “L2” ou “layer 2”), [introduit fin 2024](#). ENS étant leader de la simplification de la résolution d’adresses à l’aide de nom d’utilisateur, intégrer une de leur nouvelle solution fait parfaitement sens dans le cadre de notre projet et de cette track.



La plateforme durin.dev

- Pour notre intégration IA permettant aux utilisateurs de discuter avec un chatbot avec le contexte de leurs documents, nous avons utilisé [l'API de LibertAI](#), lancée fin avril 2025. Cette API nous permet d'offrir des fonctionnalités d'IA sans compromettre les données de nos utilisateurs et en restant décentralisé, puisque LibertAI est open source et que leur API tourne sur des instances Aleph de type "TEE" (Trusted Execution Environment), c'est à dire avec un chiffrement matériel de la RAM, du CPU et du disque, permettant une totale confidentialité sans que personne, y compris l'opérateur du noeud Aleph sur lequel tourne l'instance ne puisse accéder aux données.
Ceci est possible grâce à des CPU particuliers permettant la création de machines virtuelles avec ce niveau de sécurité, en l'occurrence ici les CPU EPYC de 9ème génération d'AMD qui intègrent la technologie [SEV](#).
Une alternative existe également du côté d'Intel appelée [TDX](#), mais uniquement celle d'AMD est actuellement supportée par Aleph et donc par LibertAI par extension.



[Post sur X annonçant le lancement de l'API LibertAI](#)

- Nous étudions également des changements suite à [la mise à jour Pectra d'Ethereum](#) qui a eu lieu en mai 2025, qui contient notamment l'[EIP \(Ethereum Improvement Proposal\) 7702](#), simplifiant énormément les smart accounts et le gas sponsoring, ce qui pourrait nous permettre de supprimer notre backend de gestion des noms d'utilisateurs pour tout gérer nativement et depuis le frontend, rendant l'expérience plus transparente.
Cette mise à jour étant extrêmement récente nous n'avons pas encore exploré en détail le sujet, mais c'est une piste intéressante

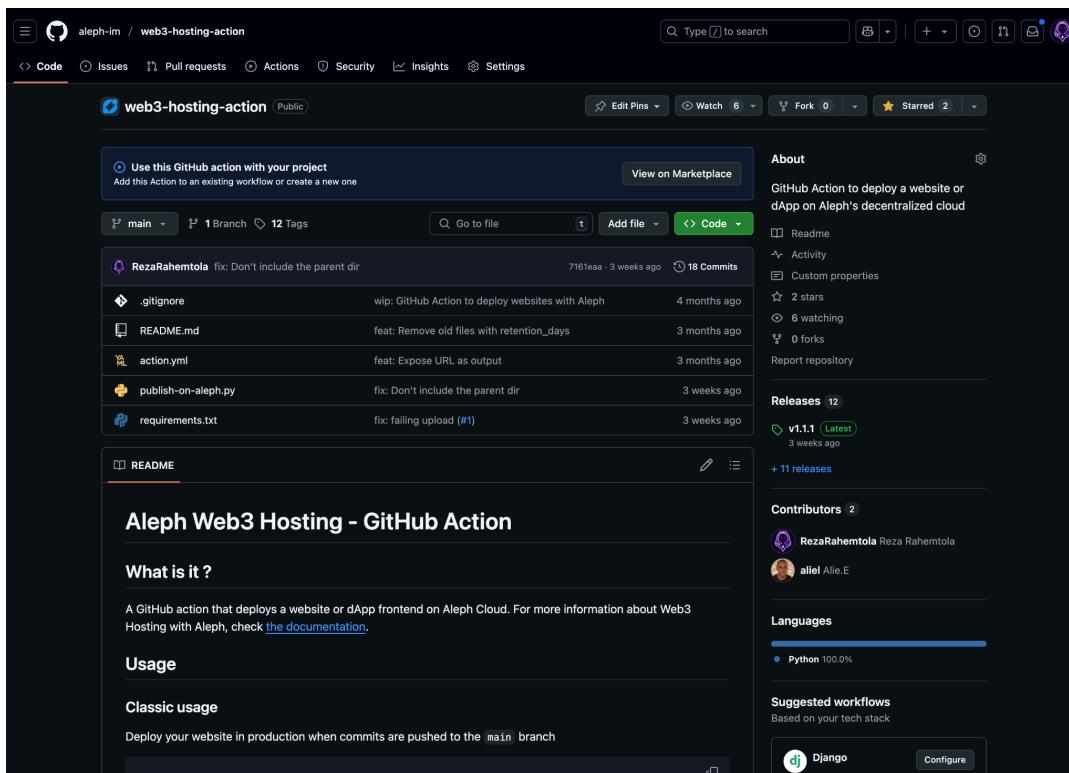
Développer les contributions communautaires

8 contributions aux repositories GitHub open-source d'Aleph Cloud (7 Pull Requests et 1 Issue)

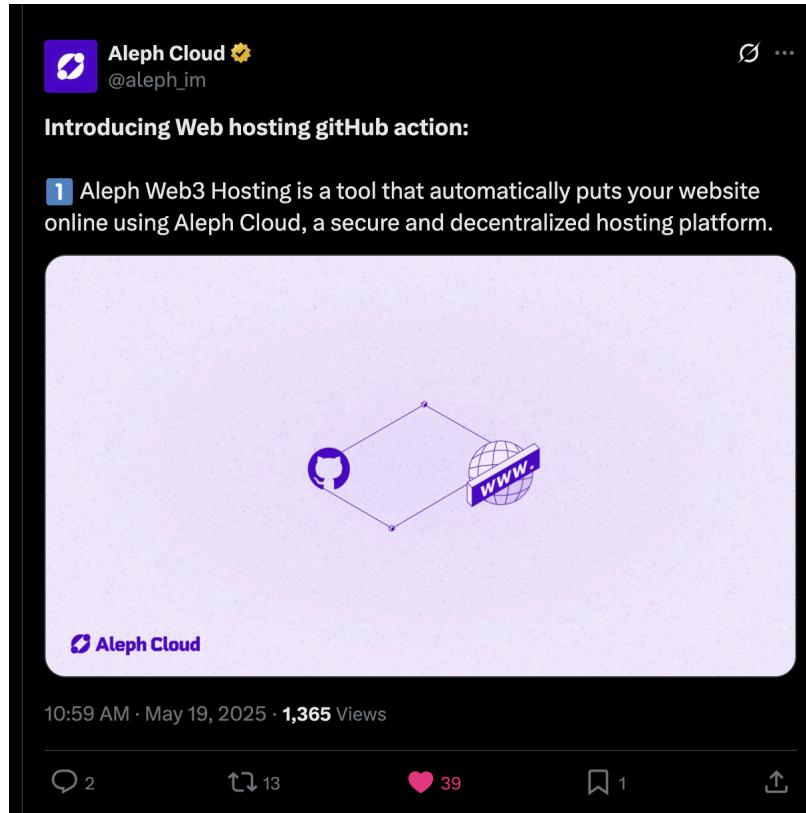
1 projet améliorant l'expérience des développeurs de site sur Aleph créé

Durant le développement de Bedrock, nous avons rencontré différents bugs avec notre utilisation des technologies mentionnées précédemment (principalement Aleph Cloud puisque c'est le pilier majeur de notre projet). Nous avons ainsi pu remonter différents soucis et suggestions d'amélioration, mais également contribuer directement à la résolution de ces bugs ou même créer des nouveaux outils pour également simplifier l'usage d'Aleph par le reste de la communauté.

La principale contribution que nous avons réalisée est la création d'une [GitHub Action permettant de déployer des sites web](#) facilement sur Aleph Cloud. L'outil est utilisé par notre propre projet et a également été partagé à la communauté Aleph pour qu'il puisse bénéficier au plus grand nombre.



Page de la GitHub Action



[Post sur X d'Aleph Cloud](#)

Nous avons également réalisé d'autres petites contributions pour fixer divers bugs rencontrés au cours du développement de Bedrock.

D'autres bugs ont également été remontés en privé par message Telegram ou via des issues GitHub.

- CLI
 - Fix d'un bug sur MacOS lors de la création d'une archive pour l'upload d'une fonction
<https://github.com/aleph-im/aleph-client/pull/370>
 - Fix de la commande SSH donnée pour se connecter à une instance Aleph <https://github.com/aleph-im/aleph-client/pull/307>
- SDKs
 - Fix critique suite à une migration d'une librairie n'autorisant plus par défaut les variables d'environnement externes
<https://github.com/aleph-im/aleph-sdk-python/pull/216>
 - Fix d'une docstring (et donc des informations affichées dans les IDE) ne reflétant pas les vraies valeurs
<https://github.com/aleph-im/aleph-sdk-python/pull/192>

- Amélioration des messages FORGET pour autoriser l'envoi à l'aide du système de permissions d'Aleph utilisé dans Bedrock
<https://github.com/aleph-im/aleph-sdk-ts/pull/179>
 - Issue ouverte pour remonter un soucis de dépendances dans les packages du SDK Typescript
<https://github.com/aleph-im/aleph-sdk-ts/issues/186>
 - Documentation
 - Multiple typos et petites erreurs dans la documentation
<https://github.com/aleph-im/aleph-docs/pull/110>
 - Mauvais affichage du formatage markdown d'une partie de la documentation <https://github.com/aleph-im/aleph-docs/pull/106>

fix: Pydantic v2 errors on unknown env vars #216

Merged

olethanh merged 1 commit into aleph-im:main from RezaRahemtola:patch-1 on Apr 22

Conversation 0

Commits 1

Checks 22

Files changed 1

RezaRahemtola commented on Apr 18 · edited

Contributor ...

Without the fix, any variable present in a `.env` file where the SDK is used (including the CLI, and products using the CLI, like the [Web3 Hosting GitHub Action](#)) throws an error that make it completely unusable:

```
(venv) reza@mac ~[D/w/f/1/aleph-client (master)> aleph about version
Aleph CLI Version: 1.5.1
(venv) reza@mac ~[D/w/f/1/aleph-client (master) [0|1]> cat .env
[bat error]: '.env': No such file or directory (os error 2)
(venv) reza@mac ~[D/w/f/1/aleph-client (master) [0|1]> nano .env
(venv) reza@mac ~[D/w/f/1/aleph-client (master)> cat .env

TEST_ENV_VAR="hey"

(venv) reza@mac ~[D/w/f/1/aleph-client (master)> aleph about version
Traceback (most recent call last):
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/venv/bin/aleph", line 5, in <module>
    from aleph_client._main import app
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/src/aleph_client/_main_.py", line 5, in <module>
    from aleph_client.commands import (
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/src/aleph_client/commands/about.py", line 7, in <module>
    from aleph_client.utils import AlephType
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/src/aleph_client/utils.py", line 21, in <module>
    from aleph.sdk.conf import MainConfiguration, load_main_configuration, settings
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/venv/lib/python3.12/site-packages/aleph/sdk/_init__.py", line 3, in <module>
    from aleph.sdk.client import AlephHttpClient
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/venv/lib/python3.12/site-packages/aleph/sdk/client/_init__.py", line 1, in <module>
    from .abstract import AlephClient, AuthenticatedAlephClient
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/venv/lib/python3.12/site-packages/aleph/sdk/client/abstract.py", line 39, in <module>
    from aleph.sdk.conf import settings
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/venv/lib/python3.12/site-packages/aleph/sdk/conf.py", line 247, in <module>
    settings = Settings()
        ^^^^^^^^
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/venv/lib/python3.12/site-packages/pydantic_settings/main.py", line 176, in __init__
    super().__init__(
  File "/Users/reza/Documents/work/freelance/libertai/aleph-client/venv/lib/python3.12/site-packages/pydantic/main.py", line 253, in __init__
    validated_self = self._pydantic_validator._validate_python(data, self_instances=self)
        ^^^^^^^^^^^^^^
pydantic_core._pydantic_core.ValidationError: 1 validation error for Settings
test_env_var
  Extra inputs are not permitted [type=extra_forbidden, input_value='hey', input_type=str]
    For further information visit https://errors.pydantic.dev/2.1/v/extra_forbidden
(venv) reza@mac ~[D/w/f/1/aleph-client (master) [0|1]> ]
```

With the fix, it ignores those variables:

```
(venv) reza@mac ~[D/w/f/1/aleph-client (master)> cat .env
TEST_ENV_VAR="hey"

(venv) reza@mac ~[D/w/f/1/aleph-client (master)> aleph about version
Aleph CLI Version: 1.5.1
(venv) reza@mac ~[D/w/f/1/aleph-client (master) [0|1]> ]
```

This is due to Pydantic v2 being more strict when unexpected fields are given, we have to explicitly ignore them.

Exemple de Pull Request de contribution à Aleph

Collaborer avec des experts

3 entreprises différentes consultées

7 appels avec des experts techniques et/ou hauts représentants de ces projets

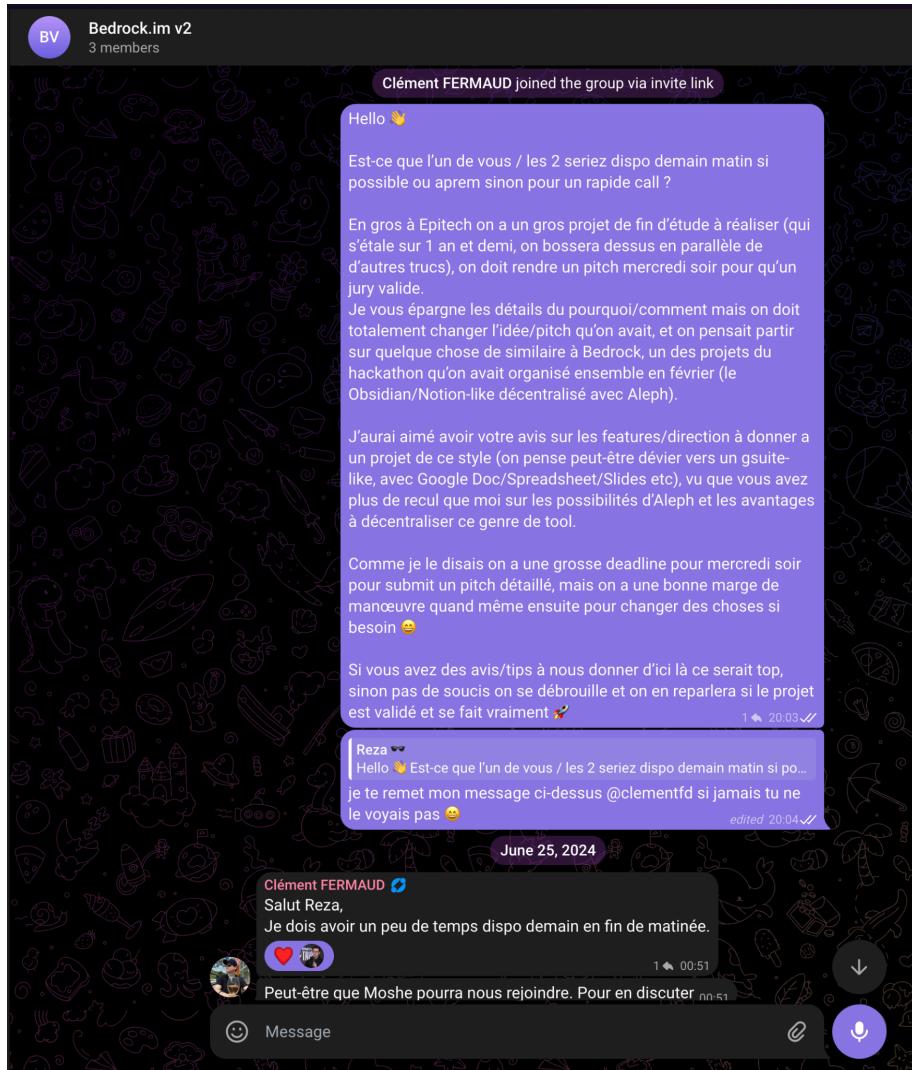
1 suivi long terme interne (LibertAI)

Tout au long du développement de Bedrock, nous avons eu l'occasion d'interagir avec les projets et technologies que nous utilisons afin d'avoir des retours, avis et pistes d'améliorations.

Cela a eu lieu **principalement avec Aleph**, qui est au centre de notre solution.

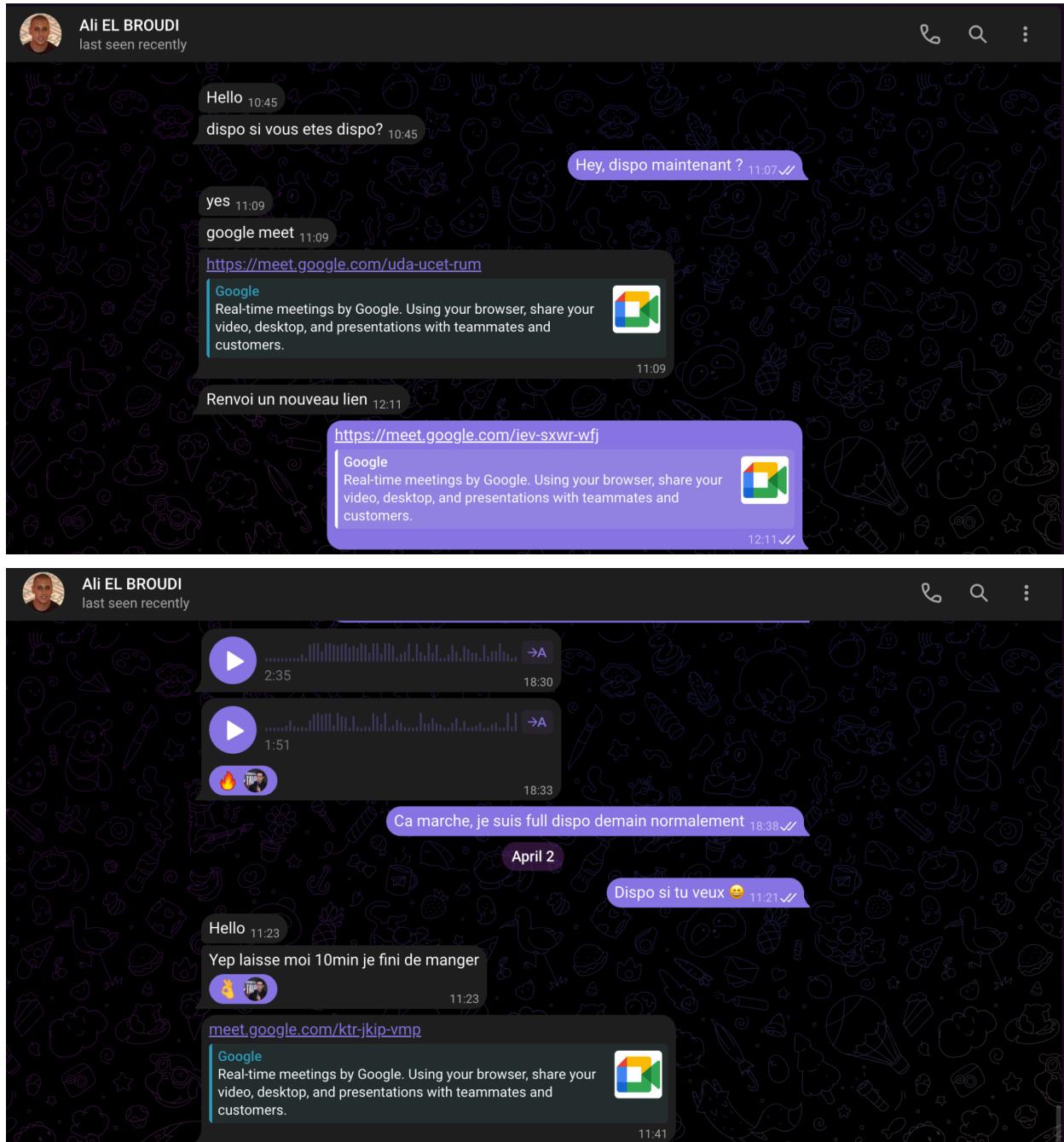
Bedrock ayant vu le jour lors d'un hackathon organisé par 1 de nos membres avec eux, et où 2 autres de nos membres ont créé la base de projet (avec un angle différent à l'époque), il était logique de continuer en explorant les possibilités avec eux.

Nous avons donc commencé par un appel avec le CEO [Jonathan Schemoul](#) (qui est un développeur également) et le CMO [Clément Fermaud](#) lors de la piscine EIP afin de recueillir leurs suggestions qui nous ont permis de définir la vision de ce projet.



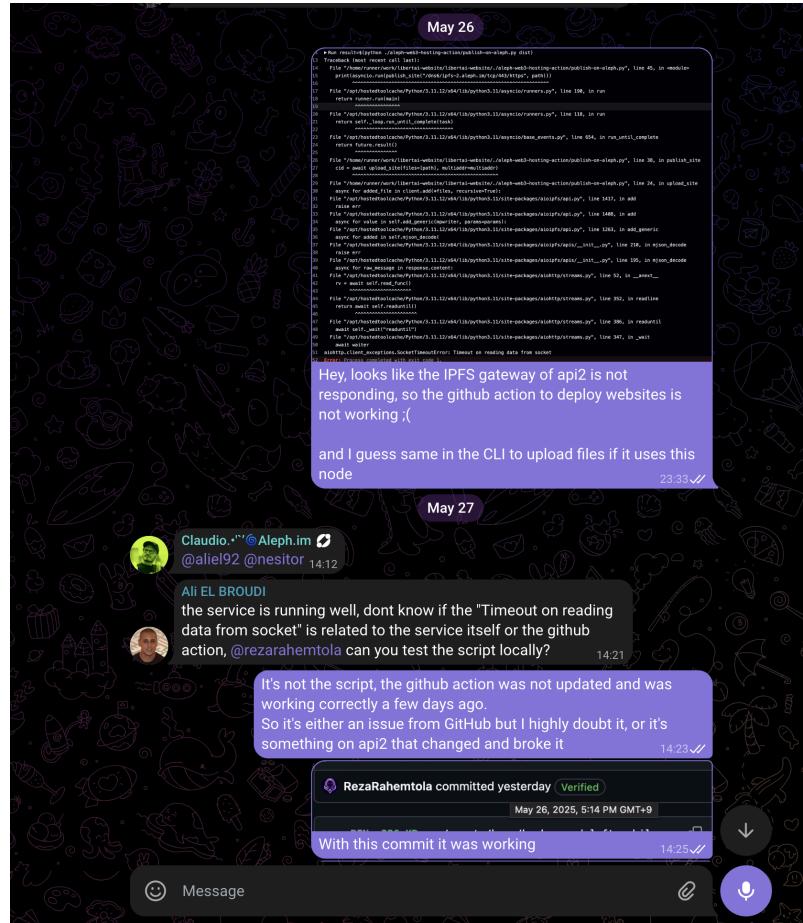
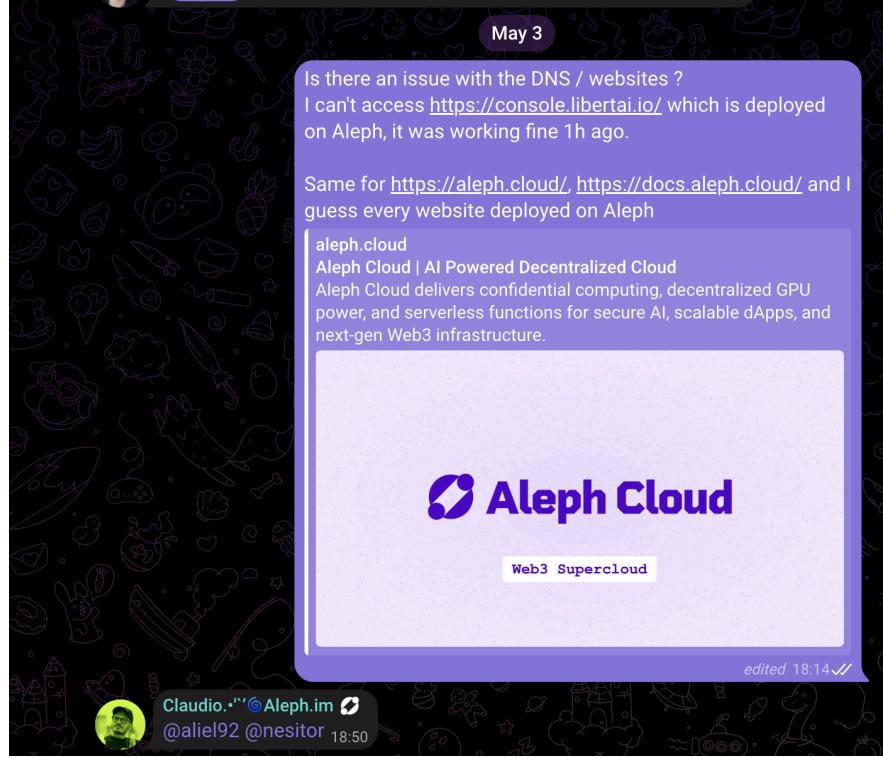
Conversation sur Telegram pour prévoir un appel à la création du projet

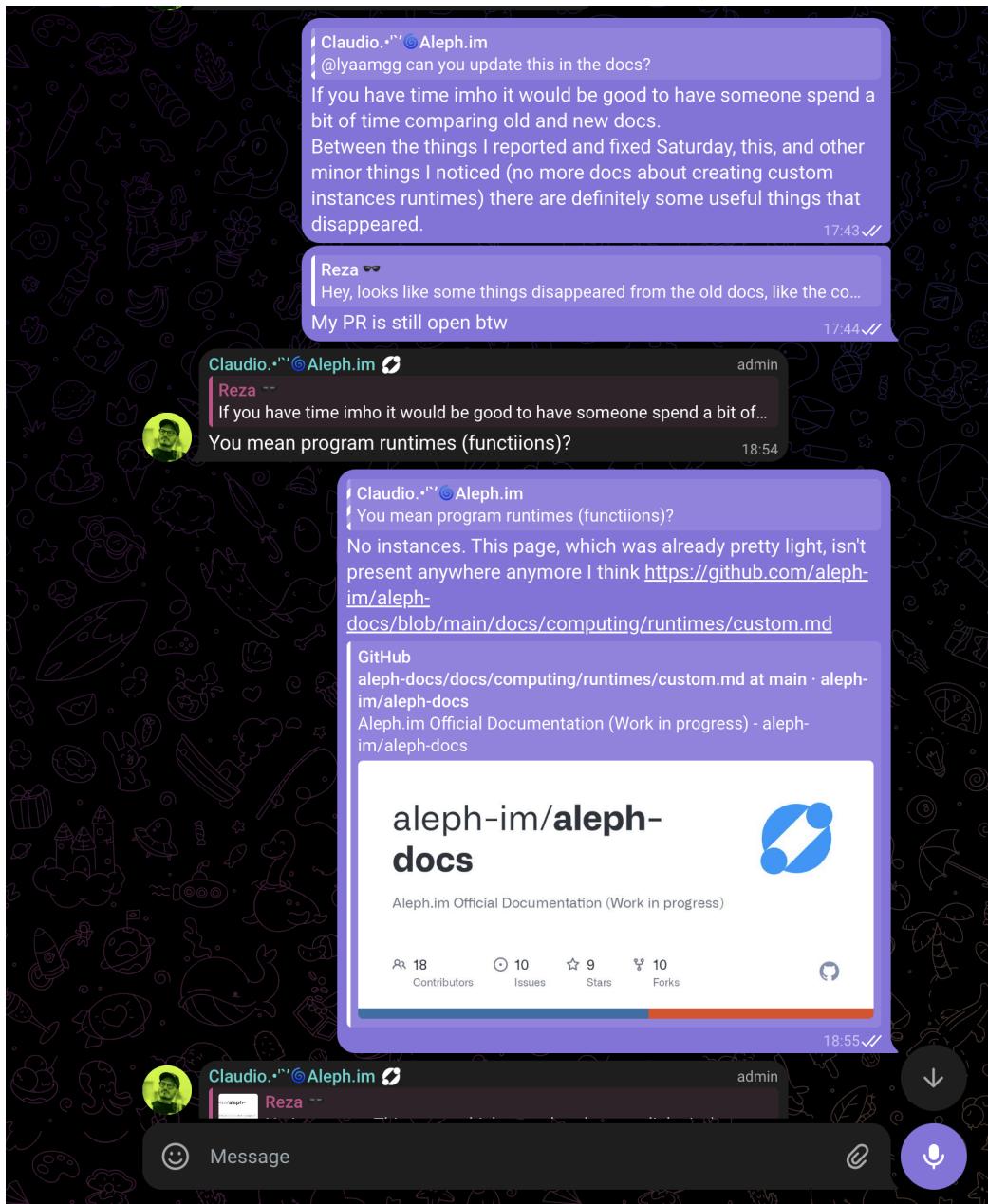
Suite à cela, nous avons fait 5 appels (~ tous les 2 mois) durant le développement du projet afin de leur présenter nos avancées, recueillir leur retours et se tenir à jour des changements et améliorations à venir sur leurs produits qui pourraient nous impacter. Ces appels ont principalement été réalisés avec [Alie El Broudi](#), DevOps Engineer & Blockchain Developer chez Aleph pendant plusieurs années et désormais en transition vers LibertAI en temps que CTO, qui est donc la personne parfaite pour nous aiguiller techniquement sur Aleph et également sur notre intégration avec LibertAI.



Messages sur Telegram pour prévoir un appel et discussions sur le projet

Nous avons également pu interagir avec d'autres personnes de l'équipe d'Aleph lors de la remontée de bugs et feedbacks, comme [Claudio Pascariello](#) (cofondateur et CPO), ou [Andrés D. Molins](#) (Tech Lead).





Diverses interactions sur Telegram avec les membres de l'équipe Aleph

Au-delà des appels et messages sur Telegram, nous avons également eu l'occasion d'échanger avec les équipes d'Aleph et de LibertAI en physique lors de divers événements, comme [Serverless Brussels](#) en juillet 2024 où notre équipe était présente.



Monaco, juin 2025

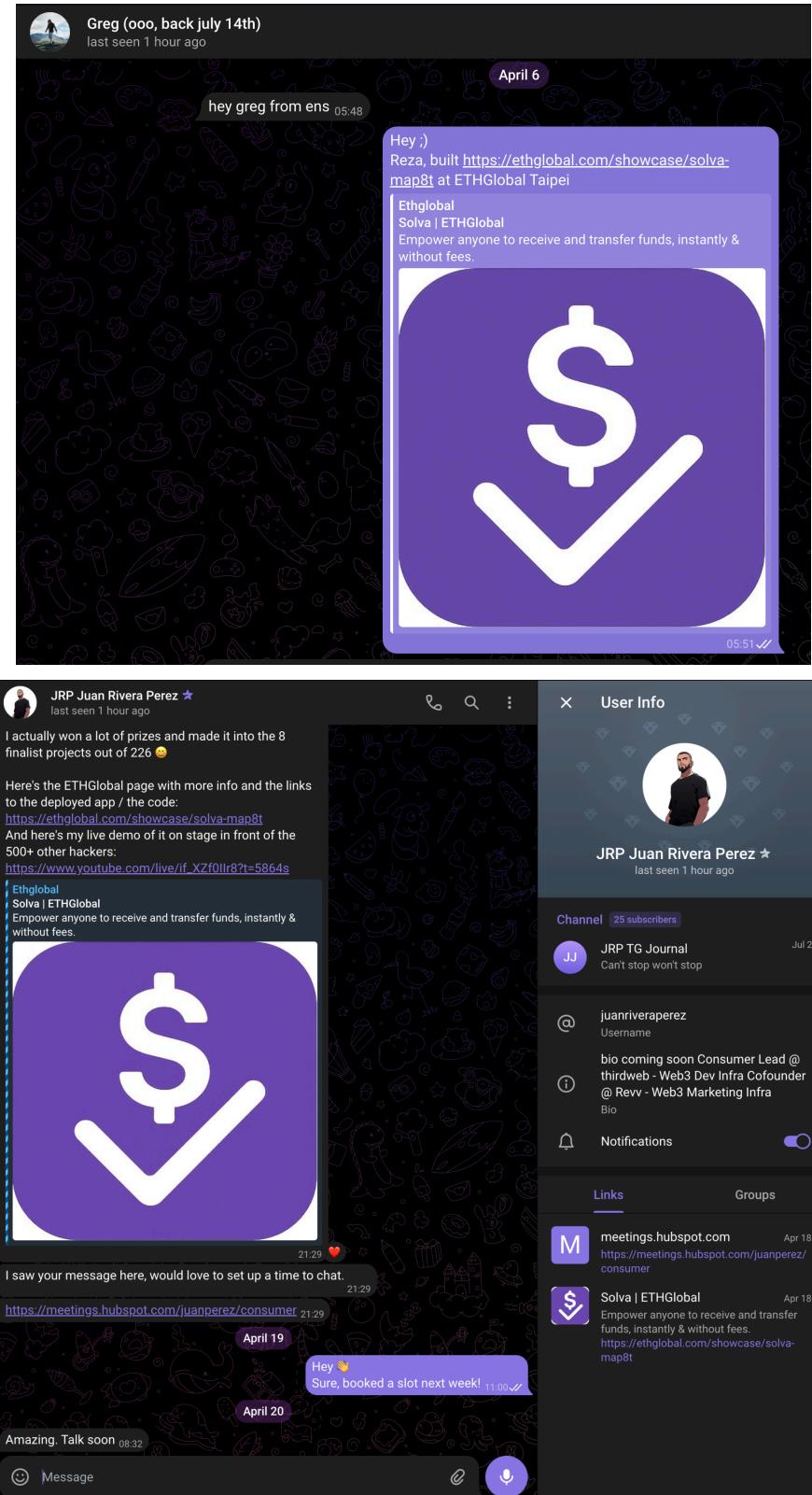
De gauche à droite: James de [INTU.xyz](#) (aucun lien avec le projet, rencontré sur place), [Joanna Caraballo Serrano](#) (Head of Marketing de LibertAI) et Reza Rahemtola (membre du projet Bedrock)



Cannes, juillet 2025

De gauche à droite: Reza Rahemtola (membre du projet Bedrock), Andrés D. Molins (Tech Lead d'Aleph) et Jonathan Schemoul (CEO et fondateur d'Aleph) lors d'un événement pétanque & networking

Enfin, en plus d'Aleph et de LibertAI, nous avons également pu interagir succinctement avec des personnes travaillant chez ENS et Thirdweb dans le contexte d'un hackathon utilisant leur technologies, où nous avons rapidement mentionné notre travail sur Bedrock et réalisé une petite démonstration



Échanges sur Telegram avec des membres d'ENS et de Thirdweb