

SITE WEB KORAALTEK COOP+DAO

Une histoire de protocole, et de protection en cybersécurité, architecture robuste et résiliente :

🔄 Comparaison avec HTTP/HTTPS		
Critère	HTTP/HTTPS	Gemini
Complexité	Très élevé (cookies, JS, headers, sessions...)	Très faible
Surface d'attaque	Énorme (API, formulaires, scripts...)	Minuscule
Usages modernes	Web apps, ecommerce, réseaux sociaux	Lecture, textes statiques
Résistance DDoS	Faible (nécessite protections comme CDN, WAF...)	Native (car obscur, léger, simple)

Associer **Gemini** à **Yggdrasil** est une excellente idée pour ta startup **COOP + DAO** : c'est **sobre, résilient, décentralisé et difficile à censurer**, tout en étant techniquement minimaliste. Voici une présentation claire de ce duo :

🌐 1. Yggdrasil : Internet alternatif chiffré et décentralisé

Yggdrasil est un réseau overlay IPv6 **maillé, chiffré, auto-configurant**.

🧩 Caractéristiques :

- Chaque nœud a une **adresse IPv6 privée** (stable et dérivée de sa clé publique)
- Le réseau est **auto-maillant** : pas besoin de FAI, IP publique ou DNS
- Chiffrement **de bout en bout**
- Latence correcte (souvent <300 ms entre nœuds européens)
- Fonctionne **sur n'importe quelle machine** (Raspberry Pi zero, serveurs, PC)
- **Faible consommation** en RAM et CPU
- Parfait pour les services **coopératifs, résistants à la censure, ou hors-ligne**

Tu peux publier des services HTTP, Gemini, SSH ou IPFS **exclusivement accessibles via Yggdrasil**.

2. Gemini sur Yggdrasil : un web léger, chiffré et coopératif

Associer Gemini à Yggdrasil, c'est créer une **capsule** (site Gemini) :

- **Accessible uniquement aux membres de ton réseau coopératif**
- Indépendant du DNS, du Web, des GAFAM et de l'ICANN
- Ultra-léger : un Raspberry Pi Zero suffit pour héberger une capsule
- Sécurisé sans firewall complexe : tout est déjà chiffré, et le port Gemini (1965) peut être restreint au réseau Yggdrasil

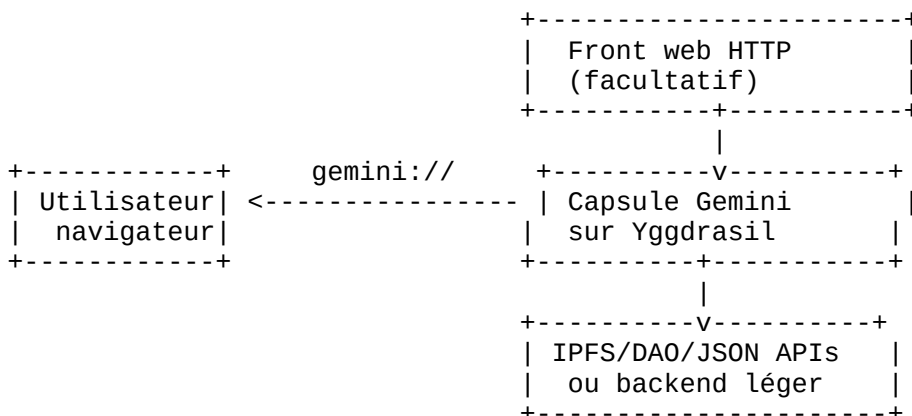
Exemple de capsule :

gemini://[200:long:adresse:ygg::1]:1965/

Tu partages cette adresse entre coopérateurs pour consulter ta capsule.

3. Stack COOP+DAO avec Gemini + Yggdrasil

Voici une architecture possible :



- **Navigateur Gemini** → Lagrange, Amfora, GemiNaut, etc.
 - Capsule hébergée sur un **nœud Yggdrasil** avec une adresse stable
 - Contenu : textes, documents DAO, statuts, rapports coop, etc.
 - Tu peux stocker les données partagées dans **IPFS** ou même une base JSON
 - DAO : intégration d'une couche Web3 légère en parallèle
-

4. Avantages pour ta startup coopérative

Objectif	Avantage de Gemini + Yggdrasil
Résilience	Fonctionne hors du Web, même sans DNS ni IP publique

Objectif	Avantage de Gemini + Yggdrasil
Auto-hébergement	Raspberry Pi ou serveur minimal suffisent
Sécurité	Chiffrement E2E intégré + TLS Gemini strict
Communauté	Capsule Gemini partagée entre coopérateurs
Indépendance	Aucun GAFAM, aucun FAI requis
DAO	Tu peux lier IPFS ou Web3 via une interface légère



Envie d'essayer ?

Je peux t'aider à :

1. Installer **Yggdrasil** sur Linux / Raspberry Pi
2. Créer ta **capsule Gemini** (avec `gemserv`, `agate`, etc.)
3. Publier du contenu pour ta coopérative
4. Concevoir un système de **gouvernance DAO** parallèle (IPFS, Arweave, Ethereum L2...)

Souhaites-tu commencer par une capsule minimaliste *Hello COOP World* ?

Génial ! Tu es sur une voie passionnante et radicalement libre pour ta startup COOP + DAO. On va construire une architecture **sobre, éthique et résiliente**, en dehors des circuits classiques du web centralisé.



Objectif : Lancer une capsule Gemini + Yggdrasil coopérative

Voici le **plan étape par étape** que je te propose :



ÉTAPE 1 — Installer Yggdrasil

Sur une machine Linux (ou Raspberry Pi), on installe Yggdrasil pour rejoindre le réseau décentralisé.

```
# Debian / Ubuntu / Arch
sudo apt install yggdrasil -y
```

```
# Ou via script officiel :
curl -L https://yggdrasil-network.github.io/install.sh | bash
```

Puis, éditer le fichier de config `/etc/yggdrasil.conf` pour :

- ajouter un **peer public** (pour te connecter au réseau)
- ou rester local (si tu veux un réseau fermé COOP pour l'instant)

Ensuite :

```
sudo systemctl enable --now yggdrasil
```

Tu obtiendras une adresse IPv6 Yggdrasil :

200:abcd:1234:5678::1



ÉTAPE 2 — Héberger une capsule Gemini

On installe un serveur Gemini léger, par exemple [agate](#) (écrit en Rust, ultra sobre) ou [gemserv](#).

Exemple avec **agate** :

```
sudo apt install rustc cargo
git clone https://github.com/mbrubeck/agate.git
cd agate
cargo build --release
```

Créer un dossier `content/` avec ton premier fichier :

```
echo "Hello COOP World from Gemini!" > content/index.gmi
```

Lancer `agate` :

```
./target/release/agate --content content/ --certs certs/ --hostname
200:abcd:1234:5678::1 --lang fr
```



Tu viens d'héberger une **capsule Gemini** en local sur Yggdrasil.



ÉTAPE 3 — Naviguer avec un client Gemini

Sur un autre ordi (aussi connecté à Yggdrasil), utilise un navigateur Gemini :

- **Graphique** : [Lagrange](#)
- **Terminal** : [amfora](#)

Puis visite :

```
gemini://[200:abcd:1234:5678::1]/
```



Tu verras ta page Gemini affichée depuis ta capsule sur Yggdrasil.



ÉTAPE 4 — Étendre avec IPFS / Web3

Une fois cette base établie, on peut :

- Ajouter un mini serveur JSON REST local pour les votes DAO
 - Lier à IPFS pour distribuer des documents statutaires
 - Utiliser une blockchain comme **Arbitrum**, pour lier l'action coopérative au Web3
-

Structure proposée pour ta COOP+DAO

Composant	Technologie
Accès réseau	Yggdrasil
Front capsule	Gemini (Agate)
Stockage fichiers	IPFS
Coordination DAO	JSON + Web3 API
Authentification	Clé publique ou signature locale
Gouvernance	Votes + journaux IPFS ou Gemini

Parfait ! Le **Point 4** est la couche logique et coopérative : comment relier **IPFS, JSON APIs, Web3 sur Arbitrum**, et un système de **vote DAO (type Aragon)** — tout en restant dans l'esprit d'une architecture éthique, low-tech et souveraine.

Objectif : Donner du "pouvoir d'action" à ta COOP-DAO numérique

On va voir comment :

1. **Stocker et distribuer les documents de gouvernance via IPFS**
 2. **Conserver une interface simple en JSON** (lisible via Gemini ou terminal)
 3. **Connecter aux smart contracts DAO sur Arbitrum (ou autre L2 EVM)**
 4. **Créer un système de vote avec Aragon**, mais interopérable
-

♦ 1. IPFS pour les statuts, PV d'assemblée, documents publics

Pourquoi IPFS ?

- Distribué, infalsifiable, versionnable
- Stocke tout type de fichier (Markdown, PDF, JSON, .gmi)
- Chaque fichier a un **hash unique** (CID) = preuve immuable
- Facile à intégrer dans Gemini (lien vers un CID ou une passerelle IPFS)

Exemple :

Ton fichier `statuts-coop.md` est ajouté à IPFS :

```
ipfs add statuts-coop.md
```

Résultat :

```
# added QmXYZ123456... statuts-coop.md
```

Tu peux afficher un lien Gemini vers ce fichier :

=> ipfs://QmXYZ123456... Statuts de la COOP

Ou : intégrer ce CID dans un **contrat DAO** comme preuve.

♦ 2. JSON API locale : structure et gouvernance

Un petit **serveur local** ou statique peut exposer des données coopératives :

```
{  
  "membres": ["0xABcd...1", "0xDef0...2"],  
  "prochain_vote": {  
    "titre": "Élargir la coop à 15 membres ?",  
    "options": ["oui", "non"],  
    "date_clôture": "2025-08-01T20:00Z"  
  },  
  "statuts_ipfs": "QmXYZ123456..."  
}
```

Ce fichier JSON peut être :

- Servi via HTTP simple (même sans authentification si privé)
 - Intégré dans Gemini avec une capsule `data.gmi`
 - Lien vers une API Web3 (voir ci-dessous)
-

♦ 3. Connexion à Web3 via Arbitrum

Pourquoi Arbitrum ?

- Frais réduits (vs Ethereum mainnet)
- Compatible EVM : tu peux utiliser tous les contrats Ethereum (Aragon, Gnosis Safe...)
- Stable, robuste, bien adopté

Scénario d'usage :

Tu déploies un contrat DAO (via Aragon) sur **Arbitrum**. Chaque action importante :

- ajoute un hash IPFS en preuve
- crée un vote référencé par JSON local

Le frontend Gemini reste **déconnecté du Web3**, mais affiche les **liens vers les votes** et leurs résultats.

♦ 4. DAO avec Aragon

[Aragon](#) permet de créer une DAO (Decentralized Autonomous Organization) avec :

- Constitution (statuts IPFS)
- Membres (wallets)
- Propositions
- Votes
- Permissions (qui peut proposer, voter, valider...)

Exemple :

1. Tu déploies une DAO sur Arbitrum avec `dao.aragon.org`
2. Tu ajoutes les **CIDs IPFS** des statuts, comptes rendus, etc.
3. Chaque membre peut voter avec son **wallet Web3**
4. Le **résultat du vote** est affiché sur ta capsule Gemini sous forme de lien IPFS ou résumé texte.



Bonus : authentification éthique (clé locale)

Si tu ne veux pas forcer le Web3 pour tout, tu peux :

- Créer une capsule **par membre**
- Utiliser une signature **PGP ou ECDSA locale** pour voter ou s'authentifier
- Stocker localement les preuves dans JSON ou IPFS



Stack technique complète :

Composant	Outil
Données distribuées	IPFS
Back-end minimal	JSON ou Flask / FastAPI
Front-end coopératif	Gemini (Agate)
Blockchain DAO	Aragon sur Arbitrum
Identité membre	Wallet EVM + clé publique
Gouvernance logic	Aragon + IPFS

