

Gankpo Architecture Overview

1. Description Technique

Le projet **Gankpo Suite** est une solution complète, transversale et asynchrone pour la souveraineté numérique applicative en République Gabonaise et alentours. Elle englobe un gestionnaire de parapheur décentralisé (Gankpo Suite), un vérificateur mathématique universel (Gankpo Reader) et une gateway métier. L'architecture est bâtie pour survivre aux pannes de connectivité, aux cyberattaques externes, et assurer l'authentification P2P des données légales.

Gankpo ne s'appuie que sur un registre de preuves distribué dans chaque fichier `.gkp`, rendu intègre à 100% par l'implémentation de clés asymétriques Ed25519 (et X.509/PKI hybrides).

2. L'Architecture Logicielle et Composants

Le projet Gankpo adopte un pattern **Monorepo** hybride.

1. Frontend Layers (Côté Client | React / TypeScript / Vite) :

- **Gankpo Suite (Desktop & Web)** : Parapheur métier complet et Hub de messagerie GKP. Gère les licences, l'Onboarding, l'historique multisig et les Trust Anchors de domaine.
- **Gankpo Reader (Desktop & Web)** : "Visionneuse cryptographique de confiance". Isolée, dépourvue de logique complexe de création, son seul but est de valider mathématiquement qu'un objet `GKP` est intègre et scellé.

2. Core Cryptographic Rust Backend (`gkp_core` / Tauri) :

- C'est la tour de contrôle. Compile en librairie C partagée (FFI), en Wasm, ou encapsulée dans Tauri.
- S'occupe des appels asymétriques, du calcul de **Blake3**, de l'interaction native (TPM, TouchID, Keychain via `security-framework`, CNG).
- Ne retourne au Frontend que des `Multisig Validation Result` stricts sans fuite de clés ou de vecteurs d'initialisation.

```
graph TD
    UI[Frontend: React/Vite] --> |Tauri IPC API| R[Rust Backend `src-tauri`]
    R --> Core[Crate `gkp_core`]
    Core --> Crypto[Ed25519, AES-GCM, Blake3]
    Core --> PKI[Trust Mngt V3]
    Core --> OS[TPM / Keychain (CNG, SecurityFramework)]

    File[GKP File System] <--> Core
    File <--> Cloud[Relais GKP - Optionnel]
```

3. Gestion Asynchrone (Offline-First)

Gankpo a fait le choix architectural suivant : **Aucun cloud centralisé ne détient la vérité.**

- L'Archive "GKP" générée par l'enclave locale **est** la source de vérité.
- Les nœuds "Gankpo Core" lisent cette archive.

- Tout GKP partagé sur WhatsApp, Mail (SMTP), USB est auto-suffisant pour être relu et vérifié (Tant que l'application cliente Gankpo connaît la clé publique de l'identité certifiante dans son Trust Register local, ou accède au LDAP si connecté).

4. Composants Additionnels de Déploiements

4.1. Field Operations (Gankpo Mobile)

L'architecture mobile native, connectée en permanence aux bases MySQL ou SQLite synchronisées locales pour répondre aux missions (Recensement, actes décentralisés isolés).

4.2. Gankpo Web Gateway & Distribution Panel

Le réseau met en place l'environnement d'administration des licences, de l'onboarding pour Gankpo via API REST et de la configuration du Trust Engine.

5. Exemples de Résilience X-Platform

En Rust : Le `Cargo.toml` est configuré statiquement pour cibler les spécificités de confiance natives des systèmes (pour ne pas inclure un moteur PKI Mac sur Windows). L'API FFI standardise ces requêtes vers le `React [tauri-plugin]`. Même avec 50 systèmes Windows ou macOS cibles, la signature Ed25519 du GKP reste mathématiquement lisible sur n'importe lequel sans aucune recompilation.