# GKP Container Specification v1

## 1. Introduction

Le format GKP (Gankpo) est un conteneur d'archives sécurisé conçu pour encapsuler un document source ( `payload` ) ainsi que l'ensemble des preuves cryptographiques (signatures, certificats, empreintes) permettant de garantir son intégrité, son authenticité de bout en bout, et sa traçabilité de cycle de vie. Le format actuel est la version **4.0.0**.

## 2. Structure du Conteneur

Le fichier `.gkp` est une archive (basée sur ZIP) dont la structure interne est organisée comme suit :

```
document.gkp/
├── payload/
│   └── source_file.pdf       # Le fichier original haché et scellé.
├── manifest.json             # Résumé des informations en clair pour Reader/UI.
├── metadata.json             # Schema complet des attributs (Auteur, Rôle, GKP
Number).
├── signatures/
│   ├── primary.sig           # Signature cryptographique initiale (Ed25519 / PKI).
│   └── append_*.sig          # Signatures de validation (Multisig ou Seals).
├── policy/
│   └── rules.json            # Stratégie de workflow (Threshold, Ordered, etc.).
└── share/
    └── envelope.json         # (Optionnel) Configurations de partage sécurisé.
```

## 3. Schéma des Métadonnées ( `metadata.json` )

Le fichier de métadonnées est le cœur logique du GKP. Ses attributs sont liés cryptographiquement :

```
{
  "version": "4.0.0",
  "kind": "Original",
  "content_id": "UUID-X",
  "hash_hex": "blake3_hash_of_payload_file...",
  "gkp_number": "GKP-1-A1B2C3D4-202603-0001",
  "issuer": {
    "actor_type": "person",
    "name": "Jean Dupont",
    "role": "Directeur (Binding V3)",
    "public_key": "ed25519_pub_key_hex"
  },
  "flags": {
    "is_encrypted": false
```

```
  }
}
```

*Toute altération de ces valeurs invalide le sceau cryptographique du conteneur.*

## 4. Structure du Payload ( `payload/` )

Il contient le document source exact tel qu'il a été inséré. L'empreinte cryptographique de ce fichier ( `hash_hex` ) est calculée via la fonction de hachage **Blake3** avec un stream invariant.

## 5. Blocs de Signature ( `signatures/` )

Chaque document est scellé par un bloc primaire. Le système autorise l'ajout de **Records Multi-signatures** (Trailing Records) pour matérialiser :

1. **Les Countersignatures** : Acteurs humains du circuit.
2. **Les Sceaux (Seals)** : Sceaux professionnels automatisés (Certificats PKI).

Structure typique : `[Magic (4)][Version (2)][PayloadLen (4)] + CBOR_Payload`

## 6. Blocs de Politique ( `policy.json` )

S'il s'agit d'un circuit de validation, le bloc `policy` indique :

```
{
  "mode": "ordered",
  "required_steps": [
    { "role": "Financial Officer", "status": "pending" },
    { "role": "CEO", "status": "pending" }
  ]
}
```

## 7. Chain of Custody & Relais

Les événements du cycle de vie (création, ajouts de signatures, partages) sont chaînés. Chaque *signature record* ajoutée possède un pointeur `prev_record_hash` pointant vers l'événement parent (le hachage local du conteneur précédent).

## 8. Structure Share Envelope

Lorsqu'un GKP passe en statut partagé, la nature du fichier passe de `Original` à `PublicShare` ou `ProtectedShare` . Un bloc `share/envelope.json` liste les clés dérivées (via passe-mot HKDF Versioning V4) pour rendre l'archive consultable uniquement par ses destinataires valides.