INSTITUT DE L'ÉTAT-CONTRAT

GUIDE TECHNIQUE — INFRASTRUCTURE BLOCKCHAIN

Ethereum + Ethereum Linea

Version 5 — Architecture complète 5 modules

Frédérick Gérald
contact@institutetatcontrat.fr · www.institutetatcontrat.fr · Mars 2026

Introduction — Vue d'ensemble de l'architecture

Ce guide décrit l'architecture blockchain de l'État-Contrat, telle que définie dans le livre et le Manifeste de Frédérick Gérald. Il couvre les cinq modules dans leur ordre logique : du plus central (le budget) au plus systémique (le vote citoyen).

PRINCIPE FONDATEUR : « La confiance ne peut plus reposer uniquement sur la morale des individus. Elle doit s'appuyer sur des mécanismes techniques rendant les décisions traçables et vérifiables. » — Le Manifeste de l'État-Contrat

L'architecture repose sur deux réseaux complémentaires :

- Ethereum L1 — ancrage permanent des preuves critiques, infalsifiable, résistant à toute censure
- Ethereum Linea L2 — exécution quotidienne des flux, ~100x moins coûteux que L1, sécurisé par L1 via ZK-proofs

Les 5 modules dans leur ordre logique

Module	Nom	Poids	Objet	Priorité
A	Le Trésor Programmable	50%	Chaque euro public traçable de l'engagement à la réalisation	1 — Capital
B	Le Permis de Gouverner	15%	Certification psychologique et cognitive de ceux qui gouvernent	2 — Fondamental
C	Les Contrats de Mandat	15%	Engagements chiffrés, jalons, sanctions automatiques	3 — Structurant
D	Les Registres d'Intégrité	20%	Anti-corruption, lobbying, patrimoine, conflits d'intérêts	4 — Systémique
E	Le Vote Citoyen	5%	Vote anonyme vérifiable via ZK-SNARKs sur les sujets structurants	5 — Complémentaire

LOGIQUE NARRATIVE : On commence par ce qui est central (le budget), puis on vérifie qui le gère (le Permis), puis on encadre ses promesses (le Mandat), puis on surveille ses réseaux (l'Intégrité).

MODULE A — Le Trésor Programmable (50%)

Le budget est le seul endroit où le réel résiste. On peut promettre n'importe quoi, voter n'importe quelle loi — mais on ne peut pas dépenser indéfiniment ce qu'on ne produit pas. C'est pourquoi le Trésor Programmable est le cœur de toute l'architecture.

Principe

Chaque euro public devient traçable de la loi de finances jusqu'au bénéficiaire final. L'attribution, le montant, la règle qui l'autorise, la preuve d'exécution — tout est enregistré sur la blockchain de manière immuable et publiquement vérifiable.

Flux technique

```
LOI DE FINANCES
      ↓
SMART CONTRACT DE BUDGET
  (règles d'attribution encodées)
      ↓
PAIEMENT CONDITIONNEL
  (déclenché automatiquement si critères remplis)
      ↓
ENREGISTREMENT SUR LINEA L2
  (hash, montant, bénéficiaire pseudonymisé, timestamp)
      ↓
ANCRAGE SUR ETHEREUM L1
  (preuve permanente, infalsifiable)
      ↓
VÉRIFICATION PUBLIQUE
  (tout citoyen peut contrôler via Lineascan / Etherscan)
```

Données enregistrées sur la blockchain

- Hash du document budgétaire source (loi de finances, décret)
- Montant alloué par mission (Santé, Éducation, Défense, etc.)
- Critères d'attribution encodés dans le contrat
- Identifiant pseudonymisé du bénéficiaire
- Horodatage de chaque étape du flux
- Preuve d'exécution et justificatif IPFS

Mesures concrètes

Périmètre	Objectif	Délai d'audit
80–95% des paiements publics	Comptes budgétaires programmables	0–7 jours (vs 12–24 mois aujourd'hui)
100% des marchés publics	Critères, scoring, attribution, paiement traçables	Temps réel

Périmètre	Objectif	Délai d'audit
100% des subventions	De l'émission au bénéficiaire final	Temps réel
Exceptions strictes	Défense, urgences (cadre légal renforcé)	Audit différé sécurisé

Pourquoi la blockchain est indispensable ici

- Registre append-only : aucune retouche comptable discrète possible
- Preuve opposable : hash, transaction et règle du contrat vérifiables publiquement
- L'entité contrôlée ne maîtrise plus les données sur lesquelles porte le contrôle
- Passage du contrôle a posteriori (après coup) au contrôle ex ante (en temps réel)

Exemple de Smart Contract — Trésor Programmable

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract TresorEtatContrat {

    struct Paiement {
        uint256 id;
        string  mission;           // ex: 'Sante', 'Education'
        uint256 montant;           // en centimes
        bytes32 beneficiaireHash;  // pseudonymise
        string  ipfsDocHash;       // justificatif IPFS
        uint256 timestamp;
        address validateur;
    }

    mapping(uint256 => Paiement) public paiements;
    uint256 public compteur;

    event PaiementEnregistre(uint256 indexed id,
        string mission, uint256 montant, uint256 timestamp);

    // Lecture publique — tout citoyen peut appeler sans restriction
    function getPaiement(uint256 _id)
        external view returns (Paiement memory) {
        return paiements[_id];
    }
}
```

CE QUE CE MODULE CHANGE : La fraude prospère sur 4 facteurs — opacité des flux, complexité procédurale, possibilité de réécriture, absence de sanction automatique. Le Trésor Programmable supprime les 4 simultanément.

MODULE B — Le Permis de Gouverner (15%)

C'est la contribution la plus originale et la plus disruptive du modèle. Il n'existe nulle part dans les démocraties occidentales actuelles. Principe : on exige un permis pour conduire une voiture, une certification pour exercer la médecine — mais aucune vérification n'est exigée pour gérer le budget des retraites ou appuyer sur le bouton nucléaire.

« La stabilité d'un système repose sur l'équilibre de ceux qui le dirigent. »

Principe fondateur

Gérer le destin de millions de personnes est une fonction critique. Elle nécessite une intégrité mentale vérifiée. Le Permis de Gouverner est une condition préalable à toute prise de fonction à haute responsabilité — non pas un jugement idéologique, mais une condition de fonctionnement.

Les 3 composantes du Permis de Gouverner

Composante 1 — Tests Psychopathologiques

Objectif : écarter les profils présentant des traits incompatibles avec l'exercice responsable du pouvoir.

- Détection des profils narcissiques pathologiques — recherche du pouvoir pour le pouvoir
- Détection des profils paranoïaques — suspicion systématique, décisions imprévisibles
- Détection des profils prédateurs — exploitation d'autrui, absence d'empathie
- Évaluation de la tolérance au stress et à la pression décisionnelle
- Capacité à intégrer des avis contradictoires sans déstabilisation

Ces tests sont réalisés par des psychologues cliniciens indépendants, sous protocole certifié, sans que l'évaluateur connaisse l'identité politique du candidat.

Composante 2 — Intégrité Cognitive

Objectif : s'assurer que le jugement n'est pas altéré par des dépendances ou des états cliniques.

- Tests toxicologiques — absence de dépendances altérant le jugement
- Bilan neuropsychologique de base — capacités décisionnelles, mémoire de travail
- Absence de pathologies non traitées affectant le raisonnement
- Renouvellement périodique obligatoire en cours de mandat

Composante 3 — Certification et Renouvellement

- Obligatoire avant toute prise de fonction à haute responsabilité
- Gradué selon le niveau : élu local → parlementaire → ministre → chef d'État
- Renouvelé à mi-mandat et en cas de suspicion clinique signalée
- Résultat archivé de manière immuable sur blockchain via ZK-proof (pas le détail médical)

Le défi technique : confidentialité médicale + preuve publique

C'est ici qu'interviennent les Zero-Knowledge Proofs (ZKP). Ce mécanisme cryptographique permet de prouver publiquement qu'un test a eu lieu et que le seuil d'aptitude est atteint, SANS jamais révéler les données médicales brutes.

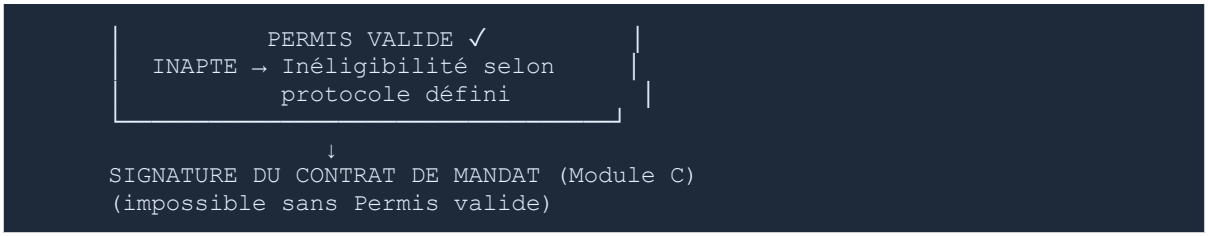
```
PROCESSUS TECHNIQUE DU PERMIS DE GOUVERNER

ÉTAPE 1 — TEST réalisé par psychologue indépendant accrédité
          (anonymat du candidat garanti côté évaluateur)
        ↓
ÉTAPE 2 — RÉSULTAT BRUT stocké chiffré hors-chaîne
          (secret médical — jamais sur la blockchain)
        ↓
ÉTAPE 3 — ZK-PROOF généré
          Preuve cryptographique que :
          • le test a eu lieu
          • le seuil d'aptitude est atteint
          SANS révéler le contenu du résultat
        ↓
ÉTAPE 4 — HASH + ZK-PROOF enregistrés sur LINEA L2
          Horodatage immuable, identité pseudonymisée
        ↓
ÉTAPE 5 — ANCRAGE sur ETHEREUM L1
          Preuve permanente, infalsifiable
        ↓
ÉTAPE 6 — VÉRIFICATION PUBLIQUE
          Tout citoyen peut contrôler :
          « Ce candidat possède un Permis valide »
          SANS accéder aux données médicales
```

Condition préalable au Module C — Contrat de Mandat

Dans l'architecture de l'État-Contrat, le Permis de Gouverner est une condition sine qua non. Un élu ne peut pas signer de Contrat de Mandat (Module C) tant que son Permis n'est pas valide et enregistré sur la chaîne.

```
CANDIDAT À UNE FONCTION DE HAUTE RESPONSABILITÉ
                    ↓
     PASSAGE DES TESTS (Module B)
     • Psychopathologie
     • Intégrité cognitive
     • Toxicologie
                    ↓
     ┌──────────────────────────────────┐
     │  RÉSULTAT ?                      │
     │  APTE  → ZK-Proof généré         │
     │          Enregistré sur Linea L2 │
     │          Ancré sur Ethereum L1   │
```



Exemple de Smart Contract — Permis de Gouverner

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract PermisDeGouvernerRegistry {

    enum Niveau { LOCAL, PARLEMENTAIRE, MINISTRE, CHEF_ETAT }
    enum Statut { INEXISTANT, VALIDE, EXPIRE, SUSPENDU }

    struct Permis {
        bytes32 candidatHash;     // identité pseudonymisée
        Niveau  niveau;           // niveau de responsabilité
        Statut  statut;
        bytes32 zkProofHash;      // ZK-proof (pas le résultat médical)
        bytes32 evaluateurHash;   // organisme évaluateur certifié
        uint256 dateEmission;
        uint256 dateExpiration;   // renouvellement obligatoire
    }

    mapping(bytes32 => Permis) public permis;

    // Vérification publique — tout citoyen peut appeler
    function estApte(bytes32 _candidatHash, Niveau _niveau)
        external view returns (bool) {
        Permis memory p = permis[_candidatHash];
        return (p.statut == Statut.VALIDE
             && p.niveau >= _niveau
             && block.timestamp < p.dateExpiration);
    }

    // Appelé par le Module C avant toute signature de mandat
    function validerAvantMandat(bytes32 _candidatHash, Niveau _niveau)
        external view returns (bool) {
        require(estApte(_candidatHash, _niveau),
            'Permis de Gouverner invalide ou expire');
        return true;
    }
}
```

Gouvernance du Module B

Acteur	Rôle	Indépendance garantie par
Collège de psychologues accrédités	Réalise les évaluations	Rotation aléatoire, anonymat du candidat

Acteur	Rôle	Indépendance garantie par
Organisme de certification	Valide et génère le ZK-proof	Séparation stricte de l'État
Smart contract blockchain	Enregistre et vérifie la validité	Code open source, audité publiquement
Tout citoyen	Vérifie qu'un élu a un permis valide	Interface publique sans identification
Tribunal constitutionnel	Tranche les contestations	Procédure définie dans le socle contractuel

POINT CLÉ : Le Permis de Gouverner ne définit pas quelle politique mener. Il garantit simplement que celui qui gouverne est mentalement apte à le faire de manière responsable. C'est une condition de fonctionnement, pas un jugement idéologique.

MODULE C — Les Contrats de Mandat (15%)

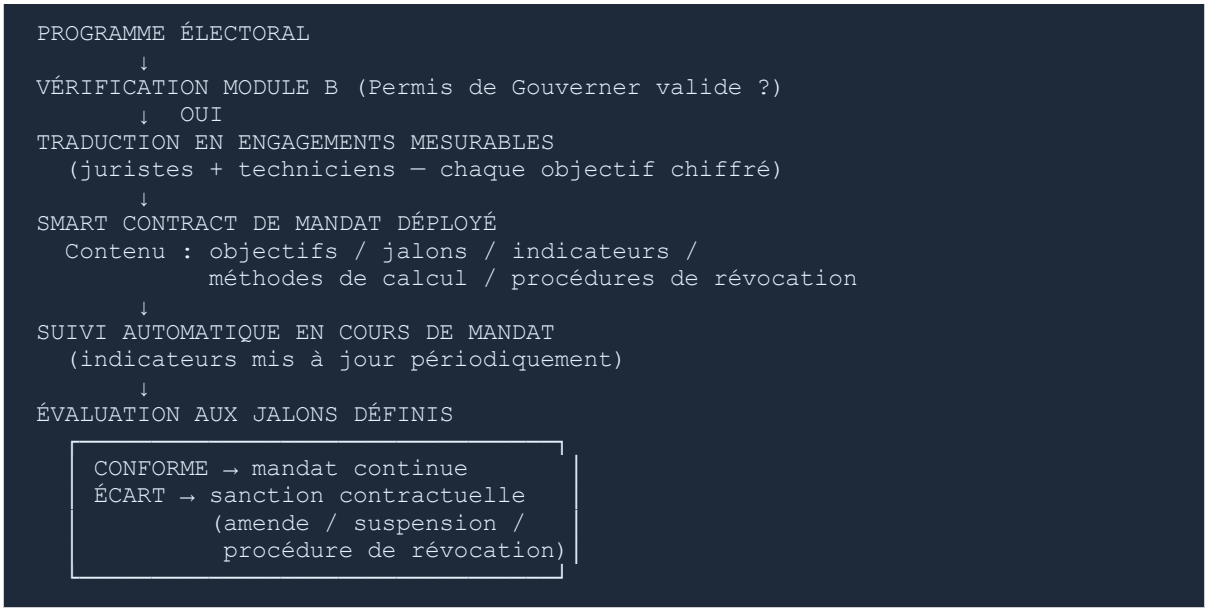
CONDITION PRÉALABLE : Le Permis de Gouverner (Module B) doit être valide et enregistré sur la chaîne AVANT toute signature d'un Contrat de Mandat. Le contrat ne peut pas être déployé sans cette vérification automatique.

Principe

Les engagements d'un élu cessent d'être des promesses. Ils deviennent des contrats exécutables — avec objectifs chiffrés, jalons mesurables, et sanctions automatiques en cas de non-respect.

L'élu ne reçoit plus un chèque en blanc : il signe un cahier des charges. Ce qui est décidé engage juridiquement et techniquement dans le temps.

Flux technique



Contenu du Contrat de Mandat

Élément	Description	Exemple concret
Objectifs chiffrés	Résultats mesurables attendus	Réduire le chômage de X% en 3 ans
Jalons temporels	Points de contrôle intermédiaires	Rapport à 6 mois, 1 an, 2 ans
Indicateurs	Méthode de calcul définie à l'avance	Source INSEE, méthode publiée
Caution de mandat	6 à 24 mois de rémunération selon niveau	Exposée contractuellement dès la prise de poste
Procédure de révocation	Conditions et seuils déclenchant la révocation	Écart > X% pendant Y mois consécutifs

Ce que ce module change

- **Le débat politique ne porte plus sur qui a raison — mais sur qui respecte ses engagements**
- L'alternance cesse d'être destructrice : la trajectoire contractuelle survit aux changements d'équipe
- La révocation cesse d'être exceptionnelle — elle devient un outil normal de régulation
- Le charisme et la rhétorique ne suffisent plus à couvrir l'échec

MODULE D — Les Registres d'Intégrité (20%)

Ce module crée un registre public et immuable de tout ce qui, aujourd'hui, reste opaque : les conflits d'intérêts, le lobbying, le patrimoine des élus, leurs agendas décisionnels et les nominations sensibles.

Domaines couverts

Domaine	Obligation	Délai
Lobbying	100% des rendez-vous d'influence déclarés	Sous 48 heures
Conflits d'intérêts	Participations et intérêts déclarés	Avant prise de fonction + mises à jour
Patrimoine	Déclaration en début et fin de mandat	J+30 de la prise de poste
Agendas décisionnels	Rendez-vous liés à des décisions publiques	Dans les 7 jours
Nominations sensibles	Critères et processus de sélection	Avant annonce officielle

Pourquoi la blockchain est indispensable ici

La fraude réelle ne consiste pas en l'absence de déclaration — mais en sa modification ultérieure. Horodatage + immutabilité = disparition de la manipulation silencieuse.

- Une déclaration déposée ne peut plus être rétroactivement modifiée
- Toute tentative de modification est visible et datée
- Les données de confidentialité partielle sont protégées via ZK-proofs

Sanctions automatiques

Manquement	Sanction contractuelle automatique
Non-déclaration d'un rendez-vous d'influence	Amende paramétrée + mention publique
Déclaration tardive (> délai contractuel)	Pénalité progressive selon retard
Conflit d'intérêts non déclaré	Suspension + procédure de révocation
Fausse déclaration prouvée	Inéligibilité et procédure pénale

POINT CLÉ : La sanction n'est pas politique, elle est contractuelle et automatique. Elle ne dépend pas d'une volonté politique de punir — elle s'applique mécaniquement dès que le manquement est constaté sur la chaîne.

MODULE E — Le Vote Citoyen via ZK-SNARKs (Zone 2 — Utile)

Dans l'État-Contrat, le vote n'est pas le cœur du système — le cœur c'est le budget, l'engagement et la preuve. Mais le vote citoyen sur les sujets structurants est l'outil de validation démocratique finale. Il doit être à la fois totalement anonyme et totalement auditable. ZK-SNARKs résout cette contradiction apparente.

Principe

Le citoyen prouve qu'il a le droit de voter — son éligibilité — sans révéler qui il est. Sa voix est comptabilisée sur Linea L2, et la preuve globale du scrutin est ancrée sur Ethereum L1. L'auditabilité est totale, mais l'anonymat est mathématiquement garanti.

C'est l'inverse exact du vote actuel : aujourd'hui l'anonymat est déclaratif (on vous fait confiance). Avec ZK-SNARKs, l'anonymat est mathématiquement prouvé — personne, même l'État, ne peut relier un vote à un votant.

Pourquoi ZK-SNARKs et pas une simple blockchain ?

- Une blockchain classique enregistre les transactions publiquement — ce qui permettrait de tracer qui a voté quoi
- ZK-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) prouve la validité d'un vote SANS révéler son contenu
- Le résultat : un registre de votes vérifiable par tous, sans qu'aucun vote individuel ne soit identifiable

Flux technique complet

```
LE CITOYEN SOUHAITE VOTER
                    ↓
ÉTAPE 1 — PREUVE D'ÉLIGIBILITÉ (hors-chaîne)
  Le citoyen prouve cryptographiquement :
  • qu'il est inscrit sur les listes électorales
  • qu'il n'a pas encore voté sur ce sujet
  • qu'il répond aux critères requis
  SANS révéler son identité
                    ↓
ÉTAPE 2 — GÉNÉRATION DU ZK-SNARK (côté client)
  Un circuit cryptographique génère une preuve :
  'Je suis éligible et je n'ai pas encore voté'
  La preuve ne contient aucune donnée personnelle
                    ↓
ÉTAPE 3 — VOTE CHIFFRÉ + ZK-PROOF envoyés sur LINEA L2
  • Le vote est chiffré (personne ne peut le lire en transit)
  • Le ZK-proof garantit sa validité sans l'identifier
  • Un nullifier unique empêche le double vote
    (sans révéler l'identité du votant)
                    ↓
ÉTAPE 4 — COMPTABILISATION SUR LINEA L2
```

```
   • Les votes valides sont agrégés
   • Chaque vote est vérifié mathématiquement
   • Le dépouillement est automatique et public
                        ↓
ÉTAPE 5 — RÉSULTAT + PREUVE GLOBALE ancrés sur ETHEREUM L1
   • Résultat définitif, infalsifiable
   • Audit complet possible par n'importe qui
   • Aucun individu ne peut être identifié
```

Les 3 garanties mathématiques

Garantie	Mécanisme	Ce que cela signifie concrètement
Anonymat total	ZK-SNARK	Personne — pas même l'État — ne peut relier un vote à un votant
Unicité du vote	Nullifier	Chaque citoyen éligible ne peut voter qu'une seule fois par scrutin
Intégrité du résultat	Ancrage L1	Le résultat final est immuable et vérifiable par tout citoyen

Le Nullifier — empêcher le double vote sans identifier le votant

C'est l'un des mécanismes les plus élégants de l'architecture. Un nullifier est une empreinte unique dérivée de l'identité du votant et de l'identifiant du scrutin — mais qui ne révèle ni l'un ni l'autre.

```
COMMENT FONCTIONNE LE NULLIFIER

nullifier = Hash(identité_secrète_votant + id_scrutin)

→ Avant le vote : le contrat vérifie que ce nullifier
  n'a jamais été utilisé pour ce scrutin
→ Après le vote : le nullifier est enregistré
→ Une tentative de revoter produit le même nullifier
  → rejeté automatiquement

Résultat : double vote impossible, identité invisible.
```

Usages dans l'État-Contrat

Le vote citoyen n'est pas consulté en permanence — il intervient sur des sujets structurants, après cadrage institutionnel (chapitre 10.7 du livre).

- Validation des grandes orientations de civilisation
- Choix des priorités économiques et sociales structurantes
- Arbitrages budgétaires majeurs rendus lisibles et chiffrés
- Politiques environnementales et territoriales importantes
- Révision du socle contractuel de l'État-Contrat

RÈGLE FONDAMENTALE : Le vote ne porte jamais sur des slogans ou des récits. Chaque sujet soumis au vote est d'abord qualifié par un organe constitutionnel indépendant, chiffré, assorti de ses conséquences mesurables, et présenté en options claires. Le citoyen arbitre entre des réalités, pas entre des promesses.

Exemple de Smart Contract — Vote ZK-SNARK

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

interface IVerificateur {
    function verifier(uint[2] memory a, uint[2][2] memory b,
        uint[2] memory c, uint[2] memory input)
        external view returns (bool);
}

contract VoteCitoyenZK {

    struct Scrutin {
        string  sujet;             // ex: 'Priorité énergétique 2030'
        uint256 ouverture;
        uint256 fermeture;
        uint256[] resultats;       // votes par option
        bool    clos;
    }

    mapping(uint256 => Scrutin)  public scrutins;
    // nullifier => utilisé ? (empêche le double vote)
    mapping(bytes32 => bool)     public nullifiers;

    IVerificateur public verificateur; // circuit ZK-SNARK

    event VoteEnregistre(uint256 indexed scrutinId,
                         uint256 option);

    function voter(
        uint256    _scrutinId,
        uint256    _option,
        bytes32    _nullifier,   // empêche double vote
        uint[2] memory a,        // ZK-proof partie 1
        uint[2][2] memory b,     // ZK-proof partie 2
        uint[2] memory c         // ZK-proof partie 3
    ) external {
        Scrutin storage s = scrutins[_scrutinId];
        require(block.timestamp >= s.ouverture, 'Pas encore ouvert');
        require(block.timestamp <= s.fermeture, 'Scrutin clos');
        require(!nullifiers[_nullifier], 'Deja vote');

        // Vérification ZK : le votant est éligible
        // SANS révéler son identité
        uint[2] memory input = [uint256(_nullifier),
                                _scrutinId];
        require(verificateur.verifier(a, b, c, input),
            'Preuve ZK invalide — non éligible');
```

```
        nullifiers[_nullifier] = true;
        s.resultats[_option]++;
        emit VoteEnregistre(_scrutinId, _option);
    }

    // Résultat public — consultable par tout citoyen
    function getResultat(uint256 _scrutinId)
        external view returns (uint256[] memory) {
        return scrutins[_scrutinId].resultats;
    }
}
```

Outils spécifiques au Module E

Outil	Usage	Lien
SnarkJS	Génération et vérification ZK-SNARKs	github.com/iden3/snarkjs
Circom	Création des circuits de vote	docs.circom.io
Tornado Cash (ref)	Référence d'implémentation nullifier on-chain	github.com/tornadocash
MACI (Ethereum)	Anti-collusion pour votes sensibles	privacy-scaling-explorations.github.io/maci
Semaphore	Protocole ZK d'identité anonyme pour le vote	semaphore.appliedzkp.org

SEMAPHORE est particulièrement recommandé : c'est un protocole open source développé par l'équipe Ethereum Foundation, spécialement conçu pour le vote anonyme vérifiable. Il implémente nativement les nullifiers et les ZK-proofs d'appartenance à un groupe.

Architecture Système Complète — Les 4 Modules

Outils Techniques — Liste Complète

A. Environnement de développement

Outil	Usage	Lien
Node.js v20+	Runtime JavaScript	nodejs.org
VS Code	Éditeur de code	code.visualstudio.com
Git	Versioning	git-scm.com
npm / pnpm	Gestionnaire de paquets	Inclus avec Node.js

B. Smart Contracts

Outil	Usage	Lien
Hardhat	Framework : développement, tests, déploiement	hardhat.org
Foundry	Alternative rapide (tests)	getfoundry.sh
Solidity v0.8.x	Langage des smart contracts	docs.soliditylang.org
OpenZeppelin	Bibliothèque de contrats audités	openzeppelin.com

Installation Hardhat :

```
mkdir etat-contrat && cd etat-contrat
npm init -y && npm install --save-dev hardhat
npx hardhat init
```

C. ZK-Proofs — Spécifiques au Module B (Permis de Gouverner)

Outil	Usage	Lien
Circom	Création de circuits ZK	docs.circom.io
SnarkJS	Génération et vérification ZK-proofs	github.com/iden3/snarkjs
Noir (Aztec)	Langage ZK plus accessible	noir-lang.org
Polygon ID	Identité auto-souveraine avec ZK-proofs	polygon.technology/polygon-id

Les ZK-proofs sont la technologie clé du Module B : ils permettent de prouver publiquement l'aptitude d'un élu SANS jamais révéler ses données médicales. C'est la seule solution compatible avec le secret médical.

D. Connexion aux réseaux

Outil	Usage	Lien
Infura	Nœud Ethereum/Linea en API	infura.io
Alchemy	Alternative à Infura	alchemy.com
Linea RPC officiel	Accès direct réseau Linea	rpc.linea.build

Configuration Linea dans Hardhat :

```
linea_mainnet: { url: 'https://rpc.linea.build', chainId: 59144 }
linea_testnet: { url: 'https://rpc.sepolia.linea.build', chainId: 59141 }
```

E. Wallets et sécurité des clés

Outil	Usage	Lien
MetaMask	Wallet de développement et tests	metamask.io
Ledger Hardware Wallet	Sécurisation des clés en production	ledger.com
dotenv	Variables d'environnement sécurisées	npm install dotenv

RÈGLE ABSOLUE : Ne jamais mettre une clé privée dans le code source. Toujours utiliser .env + .gitignore.

F. Tests et audit de sécurité

Outil	Usage	Lien
Hardhat Test + Mocha/Chai	Tests unitaires	hardhat.org
Slither	Analyse statique de sécurité	github.com/crytic/slither
Mythril	Détection de vulnérabilités	github.com/ConsenSys/mythril
OpenZeppelin Defender	Surveillance en production	openzeppelin.com/defender

G. Interface publique — Tableau de bord citoyen

Outil	Usage	Lien
Next.js	Framework React	nextjs.org
ethers.js / viem	Connexion JavaScript ↔ blockchain	ethers.org
wagmi	Hooks React pour blockchain	wagmi.sh
The Graph	Indexation des données blockchain	thegraph.com
Vercel	Hébergement web	vercel.com

H. Explorateurs publics

Outil	Usage	Lien
Etherscan	Explorateur Ethereum L1 — audit citoyen	etherscan.io
Lineascan	Explorateur Linea L2 — audit citoyen	lineascan.build

I. Stockage décentralisé des documents

Outil	Usage	Lien
IPFS	Stockage décentralisé (PDF, protocoles)	ipfs.tech
Pinata	Pinning IPFS — garantit la disponibilité	pinata.cloud
Filecoin	Stockage avec garantie de durée	filecoin.io

Environnement de Test et Planning

Testnets gratuits — commencer sans risque

Réseau	Usage	Faucet ETH gratuit
Sepolia (Ethereum testnet)	Test Layer 1	sepoliafaucet.com
Linea Sepolia (Linea testnet)	Test Layer 2	faucet.goerli.linea.build

Planning de mise en œuvre

Phase	Durée	Contenu	Coût
0 — Préparation	1-2 sem.	Installation, testnets, Hardhat, MetaMask	0 €
1 — Module A prototype	4-6 sem.	Trésor Programmable + interface de lecture basique	0 € (testnet)
2 — Modules B, C, D	8-10 sem.	Permis + ZK-proofs + Contrats + Registres	0 € (testnet)
3 — Audit sécurité	4-6 sem.	Audit externe + corrections	5 000 – 30 000 €
4 — Déploiement	2-3 sem.	Linea Mainnet + ancrage Ethereum L1	100 – 500 €
5 — Exploitation	Continu	API publique, The Graph, documentation	20 – 100 €/mois

Équipe et Ressources

Compétences nécessaires

Profil	Rôle	Module(s)
Développeur Solidity	Smart contracts principaux	A, B, C, D
Expert ZK-proofs (Circom/Noir)	Architecture du Permis de Gouverner	B
Développeur Full-Stack Next.js	Interface publique et tableau de bord citoyen	Tous
Expert sécurité blockchain	Audit des contrats avant déploiement	Tous
Juriste constitutionnaliste	Traduction juridique des engagements	C, D
Psychologue clinicien	Définition des protocoles d'évaluation	B
Expert ZK vote (Semaphore/MACI)	Architecture du vote anonyme vérifiable	E
DevOps	Infrastructure, déploiement, monitoring	Tous

Ressources documentaires

Ressource	URL
Ethereum Developer Docs	ethereum.org/developers
Linea Developer Docs	docs.linea.build
Hardhat Documentation	hardhat.org/docs
Solidity Documentation	docs.soliditylang.org
OpenZeppelin	docs.openzeppelin.com
Circom (ZK-proofs)	docs.circom.io
SnarkJS	github.com/iden3/snarkjs
Noir Language	noir-lang.org/docs
The Graph	thegraph.com/docs
IPFS	docs.ipfs.tech
Semaphore (vote ZK)	semaphore.appliedzkp.org
MACI (anti-collusion)	privacy-scaling-explorations.github.io/maci

Institut de l'État-Contrat

contact@institutetatcontrat.fr · www.institutetatcontrat.fr

« La paix n'est pas un idéal : c'est un effet secondaire d'un cadre clair. »