Landblock Whitepaper v0.4.3

Landblock Core Team

April 2026

Introduction

Elevator Pitch

"Landblock is a blockchain federation protocol that enables global land registry interoperability without compromising sovereignty."

Landblock is an open, neutral blockchain protocol that creates a shared trust layer for land registries worldwide. It enables registries to verify each other's records, exchange proofs, and cooperate across jurisdictions while maintaining complete autonomy over their data and processes.

Declaration of Neutrality

"Landblock does not replace courts or governments. Courts decide land ownership. Governments decide policy. Landblock provides the neutral infrastructure that makes those decisions globally verifiable."

Built on the Polygon PoS blockchain, Landblock provides timestamped, cryptographically chained records of land assertions, ownership transfers, supporting evidence, and authority attestations. No land data is stored on chain — only hashes, timestamps, signatures, and minimal metadata. This is not a concession; it is the design. It preserves privacy, minimizes operational cost, and makes political adoption feasible.

Current Status (April 2026)

- **Phase 1-3 Complete**: Federation architecture designed, core contracts implemented and tested
- **Production-Ready Components**: DisputeRecord.sol, PrivacyVerifier.sol, EvidenceStore.sol, LDBKToken.sol
- **Active Development**: ZK circuits, web applications, Registry Template, indexer service
- **Testnet Deployment**: Core contracts ready for Amoy testnet deployment
- **Feature Documentation**: Comprehensive documentation for all major capabilities

Problem Space

1.1 The Global Land Registry Crisis

More than 70% of Africa's land is informally held, depriving owners of credit and legal protection. Fraud affects 30% of land transactions in emerging markets.

Specific Failure Modes

Failure Mode	Example	How Landblock Helps
Corruption / Silent Rewrites	Kenya: 5,000+ families evicted after fraudulent titles issued (2021)	Detects silent record modifications; immutable audit trail
Disaster (fire, flood, war)	Paper registries destroyed in conflict zones	Independent off-site cryptographic proof survives physical destruction
Insider Abuse	Officials altering records without oversight	Every change is attributed, timestamped, and permanently visible
Conflict / Competing Claims	60% of India's court cases are land disputes	Neutral timeline of competing claims with timestamped evidence
IT Failure / System Gaps	Ghana: 40+ years of paper records not digitized	Cryptographic restore point; audit baseline for digitization

1.2 What Landblock Proves — and Does Not Prove

"Proof that this version existed at this time, unchanged since." This is the difference between **verification** and **authority**.

Category	Landblock DOES	Landblock DOES NOT
Records	Prove a record existed at a specific point in time	Declare that the record is legally valid or true
Integrity	Prove a record has not been altered since recording	Automatically correct or heal corrupted data
Authority	Record attestations from recognized authorities	Replace courts, governments, or authoritative registries
Ownership	Timestamp a claim made by an identified party	Award, adjudicate, or enforce ownership

1.3 Reframing for Adoption

Most land registry projects fail because they try to replace trust. Landblock succeeds because it preserves trust under stress. Instead of asking "should we trust this new system?", Landblock asks "how can we verify what we already trust?".

1. **Sovereignty anxiety**: Landblock does not claim authority over land outcomes.
2. **Political resistance**: Landblock is complementary to existing registries, not competitive.
3. **"Blockchain takeover" fear**: Landblock is a neutral verification layer, not a replacement system.

Design Principles & Goals

1.1 Core Design Principles

1. **Clarity over opacity**: Every claim, every change, every attestation is attributed and visible.
2. **Record, don't rule**: The protocol records assertions; truth is determined by courts and community processes outside the protocol.
3. **No land data on chain**: Only hashes, timestamps, signatures, and minimal metadata appear on the blockchain to preserve privacy and reduce cost.
4. **Resilience over fragility**: Prioritizes designs that adapt to change and remain useful under stress.
5. **Composability over monoliths**: Separate modules (parcels, identity, evidence, disputes, tokens) with clear interfaces.
6. **Stewardship over short-term optimization**: Designs must outlast any single transaction or institutional cycle.
7. **Explicit tradeoffs over hidden assumptions**: Surface policy choices rather than concealing them.

1.2 Functional Design Goals

- **Verifiable ownership**: Every parcel state transition is tamper-evident and auditable.
- **Fast settlement**: Ownership transfer completes in minutes, not months.
- **Institutional interoperability**: Government workflows integrate without full institutional replacement.
- **Privacy-preserving identity**: Participants prove eligibility via ZKP selective disclosure.
- **Geospatial integrity**: Physical parcel boundaries represented as GeoJSON and cryptographically linked to evidence.

1.3 Non-Functional Goals

- **High throughput**: Network-level throughput via Polygon PoS (~7,000 TPS; Polygon zkEVM path available for future phase upgrades).

- **Predictable costs**: Near-zero transaction costs for users via the Paymaster model.
- **Fault tolerance**: Resilience across node, service, and data-layer failures.
- **Legal defensibility**: Outputs capable of surviving institutional audit and court scrutiny.

System Overview

1.1 The Fundamental Design Rule

CRITICAL: NO LAND DATA ON CHAIN Only cryptographic hashes, timestamps, digital signatures, and minimal metadata are written to the Polygon PoS blockchain. Documents, images, GeoJSON boundaries, and personal data are stored off-chain (IPFS + Filecoin) and referenced by content-addressed hashes.

1.2 Core Components

1. **Solidity Protocol Contracts (L1)**: 16+ core modules (directory, federation, registry template, identity, evidence, disputes, privacy verification, tokens, gas station, spatial units, property tokens) running on Polygon PoS.
2. **Off-Chain Storage (L2)**: Content-addressed storage for heavy artifacts using IPFS and long-term persistence on Filecoin.
3. **Oracle Layer (L2b)**: Pyth Network provides external values for fee pricing and compliance checks.
4. **Derived Read-Models (L3)**: PostgreSQL read-models synchronized via Polygon event log indexers for high-performance querying.
5. **Applications (L4)**: Web applications (Explorer, Investor, Privacy Verifier) and mobile registration (Register).

1.3 Technical Stack Snapshot

Layer	Technology	Rationale
Blockchain	Polygon PoS	EVM-compatible; near-zero tx costs; massive ecosystem; enterprise track record
Smart Contracts	Solidity (OpenZeppelin v5)	Industry standard; battle-tested security primitives; broad auditor pool
Multi-Sig	Gnosis Safe	Gold-standard EVM multi-sig; billions in TVL; full Polygon PoS support
Identity	SpruceID (DIDs + ZKP)	W3C standard; selective disclosure; self-sovereign
Oracles	Pyth Network	Signed multi-publisher feeds; low-latency compliance

Layer	Technology	Rationale
		checks
Storage	IPFS + Filecoin	Content-addressing for integrity; economic persistence
Indexing	Polygon Event Indexer + PostgreSQL	High-performance read-layer; GraphQL/REST APIs
ZK Proofs	Circom + SnarkJS	Privacy-preserving verification without data exposure
Federation	Custom Protocol	Cross-registry interoperability without sovereignty loss

Core Models

All data models described here represent what is stored **ON CHAIN**. All heavy content (GeoJSON, documents, personal ID) is stored **OFF CHAIN** (IPFS/Filecoin) and referenced via cryptographic hashes (CIDs).

1.1 Parcel Record

Field	Description
parcel-id	Globally unique string derived from chain ID + centroid hash
assertion-timestamp	Block time at registration (immutable)
asserting-party	SpruceID DID of the asserting party
boundary-hash	IPFS CID of the GeoJSON boundary payload
evidence-links	Array of IPFS CIDs for supporting evidence
status	Enum: active
version-history	Immutable change log referencing prior boundary-hashes
witnesses	Array of DID references for community attestors
transfer-locked	Boolean — true if dispute is open or validator hold active

1.2 Identity Record

Field	Description
identity-id	SpruceID DID (W3C standard)
identity-tier	self-asserted

Field	Description
attestations	Verification chain (attester, timestamp, tier)
document-hashes	IPFS CIDs of verification docs (never the docs themselves)
zkp-credentials	ZKP-enabled selective disclosure credentials
account	Ethereum address (EOA or Gnosis Safe) linked to DID

1.3 Evidence Record

Field	Description
evidence-id	Unique identifier
content-hash	IPFS CID verifying file integrity
document-type	geojson-survey
submitted-by	DID reference
parcel-refs	Array of Parcel IDs associated with this evidence

1.4 Dispute Record

Field	Description
dispute-id	Unique identifier
parcel-refs	Parcels with overlapping or contested claims
claimants	Array of DIDs with competing claims
status	open
transfer-lock	Always true when status is open or under-review

1.5 Property Token Record

Field	Description
token-id	NFT identifier
parcel-id	Backing land record reference
total-supply	1.0 for whole ownership; fractions sum to 1.0
ownership-table	{account: fraction} map
evidence-threshold	Evidence score met at mint time
restrictions	Jurisdiction-specific Solidity access controls (OpenZeppelin AccessControl; KYC gate, transfer lock-up)

Architecture

1.1 Layer Architecture

Landblock follows a strictly decoupled five-layer federation architecture:

Layer	Name	Responsibility	Technology
L0	Consensus	Transaction ordering, finality, security	Polygon PoS (Proof of Stake, EVM-compatible)
L1	Federation Protocol	Core registry federation, identity, evidence, disputes, privacy verification	Solidity Smart Contracts (OpenZeppelin v5)
L2	Off-Chain	Documents, images, GeoJSON (hashes on L1)	IPFS + Filecoin
L2b	Oracle Layer	External values: fee pricing, compliance checks	Pyth Network
L3	Indexing	Queryable views, GraphQL/REST, event streaming	Indexer + PostgreSQL
L4	Applications	User-facing registration, exploration, investment, privacy verification	Web + Mobile (Next.js, React Native)

1.2 Core Protocol Modules (Solidity)

Sixteen+ Solidity contracts constitute the protocol core, organized by federation layer:

Global Directory Layer

1. **RegistryDirectory.sol**: Hierarchical index of participating registries, jurisdictions, and geographic coverage
2. **FederationProtocol.sol**: Cross-registry interoperability standards and proof verification

Registry Layer

3. **IdentityRegistry.sol**: SpruceID DIDs with tiered attestations and cross-registry resolution
4. **SpatialUnitRegistry.sol**: LADM-compliant spatial units (parcels, BAUnits) with coordinate validation

5. **RRRRegistry.sol**: Rights, Restrictions, and Responsibilities (RRR) management
6. **IBAUnitRegistry.sol**: Basic Administrative Units (BAUnits) lifecycle management

Federation Services Layer

7. **EvidenceStore.sol**: Tamper-evident content storage with cryptographic integrity guarantees
8. **DisputeRecord.sol**: Role-gated injunctions and evidence-based arbitration framework
9. **PrivacyVerifier.sol**: Zero-knowledge proof verification for cross-registry validation without data exposure

Economic Layer

10. **LDBKToken.sol**: Fixed-supply utility token (21M LDBK) for platform operations
11. **LGTToken.sol**: Dynamic-supply governance token for protocol decision-making
12. **LandblockPaymaster.sol**: ERC-2771 meta-transaction support for gasless user experience

Supporting Infrastructure

13. **CoordsValidator.sol**: OGC-compliant coordinate validation and spatial integrity
14. **PropertyToken.sol**: Optional registry-permissioned collateral instruments
15. **LandblockGovernance.sol**: Aragon DAO integration for protocol governance
16. **ParcelRegistry.sol**: Legacy parcel assertions (transitioning to LADM-compliant registries)

1.3 System Context Diagram

```
graph TD
    subgraph L4 [Application Layer]
        A[Landblock Register - Mobile]
        B[Landblock Explorer - Web]
        C[Landblock Investor - Web]
        D[Privacy Verifier - Web]
    end

    subgraph L3 [Indexing Layer]
        E[Polygon Event Indexer]
        F[(PostgreSQL Read-Models)]
    end

    subgraph L1 [Federation Protocol Layer - Polygon PoS]
        subgraph Global_Directory [Global Directory]
            G[RegistryDirectory.sol]
            H[FederationProtocol.sol]
        end

        subgraph Registry_Services [Registry Services]
            I[IdentityRegistry.sol]
            J[SpatialUnitRegistry.sol]
```

```
            K[RRRRegistry.sol]
            L[IBAUnitRegistry.sol]
        end

        subgraph Federation_Services [Federation Services]
            M[EvidenceStore.sol]
            N[DisputeRecord.sol]
            O[PrivacyVerifier.sol]
        end

        subgraph Economic_Layer [Economic Layer]
            P[LDBKToken.sol]
            Q[LGTToken.sol]
            R[LandblockPaymaster.sol]
        end
    end

    subgraph L2/L2b [External Service Layer]
        S[(IPFS + Filecoin - Documents)]
        T[Pyth Network - Oracles]
        U[Circom Circuits - ZK Proofs]
    end

    A --> E
    B --> E
    C --> E
    D --> E
    E --> F
    F --> Global_Directory
    F --> Registry_Services
    F --> Federation_Services
    F --> Economic_Layer
    Global_Directory --> S
    Registry_Services --> S
    Federation_Services --> S
    Federation_Services --> U
    Economic_Layer --> T

end

A & B & C --> D
D --> E
F & G & H & I & J & K -.-> D
F & G & H & I & J & K <--> L & M

```

> [!NOTE] For a detailed technical breakdown of the monorepo architecture and repository structure, see the Architecture Overview system of record.

Key User Flows

1.1 Land Registration Flow (LADM-Compliant)

1. **DID Registration**: User creates a SpruceID DID with tiered attestations. Protocol Paymaster covers transaction fees.
2. **Spatial Unit Definition**: User defines parcel boundary using Coords URIs (OGC-compliant coordinate system). Register app validates spatial integrity.
3. **BAUnit Creation**: System creates Basic Administrative Unit (BAUnit) in IBAUnitRegistry.sol with proper LADM semantics.
4. **RRR Assignment**: Rights, Restrictions, and Responsibilities are established in RRRRegistry.sol.
5. **Evidence Attachment**: Supporting documents are stored in EvidenceStore.sol with cryptographic integrity guarantees.
6. **Witness Attestation**: Community witnesses attest via their DIDs, creating multi-party verification.
7. **Registry Recording**: SpatialUnitRegistry.sol records the complete LADM-compliant parcel structure.

1.2 Cross-Registry Verification Flow

1. **Proof Request**: Registry A requests verification of a record from Registry B.
2. **Privacy-Preserving Query**: PrivacyVerifier.sol enables verification without exposing underlying data.
3. **ZK Proof Generation**: Registry B generates zero-knowledge proof of record validity.
4. **Federation Verification**: FederationProtocol.sol validates proof against global trust standards.
5. **Trust Context Return**: Verification result includes conformance tier, proof age, and validation basis.

1.3 Dispute Resolution Flow

1. **Dispute Initiation**: Automatic (spatial overlap) or manual filing creates DisputeRecord.sol entry.
2. **Action Freezing**: Role-gated injunctions freeze specific actions (RRR mutations, BAUnit transfers).
3. **Evidence Submission**: Parties submit tamper-evident evidence packages to EvidenceStore.sol.
4. **Arbitration**: Multi-party arbitration with evidence-based decision making.
5. **Resolution Recording**: Final ruling updates dispute status; automatic injunction expiry.
6. **Appeal Support**: Dispute lifecycle supports appealed and superseded states.

1.4 Property Tokenization Flow

1. **Eligibility Check**: System verifies LADM compliance, dispute-free status, and seasoning period.
2. **Permissioned Minting**: Registry-approved PropertyToken.sol minting creates collateral instrument.
3. **Lien Recording**: Token creation automatically records liens in RRRRegistry.sol.
4. **Federation Integration**: Optional cross-registry collateral verification for lending.
5. **Compliance Enforcement**: Transfer restrictions enforced by Solidity access controls.

Security and Trust

1.1 Formal Verification (Foundry + Slither)

Critical contract invariants are verified using **Foundry invariant tests** and **Slither** static analysis before deployment. Key verified properties include: - DisputeRecord.sol: Role-gated injunctions cannot be issued without proper authorization; automatic expiry prevents permanent freezes - PrivacyVerifier.sol: Zero-knowledge proofs validate without exposing underlying data; proof verification is cryptographically sound - EvidenceStore.sol: Content-addressed storage provides tamper-evident integrity; cryptographic hashing prevents content alteration - LDBKToken.sol: Fixed supply mechanics prevent inflation; controlled burns maintain scarcity - Federation Protocol: Cross-registry verification maintains sovereignty; trust contexts prevent false validation

All state-mutating contract functions use OpenZeppelin `ReentrancyGuard` and `AccessControl` to prevent unauthorized access and reentrancy attacks.

1.2 Privacy-Preserving Verification

Zero-Knowledge Proof Security Model: - **Data Sovereignty**: Registries prove record validity without revealing underlying data - **Cryptographic Integrity**: Circom circuits ensure mathematical proof soundness - **Federation Trust**: Multi-tier conformance system prevents false validations - **Temporal Security**: Proof age and staleness checks prevent replay attacks

1.3 Content-Addressed Document Integrity

Documents are stored on IPFS/Filecoin with dual persistence. Their content identifiers (CIDs) are cryptographic hashes recorded on-chain. This provides mathematical proof that retrieved documents are identical to those submitted. Authenticity verification requires only hash comparison against blockchain records.

1.4 Federation Security Model

Cross-Registry Trust Framework: - **Conformance Tiers**: Mirror, Verified, and Full Federation tiers with escalating trust requirements - **Bilateral Agreements**: Explicit trust relationships between participating registries - **Contested Zones**: Neutral handling of disputed territories without political adjudication - **Legal Reference Library**: DAO-governed index of national property law frameworks

1.5 Sybil Resistance & Identity Gating

- **Tiered Trust**: Self-asserted, Community-verified, and Authority-verified DID attestations
- **Cross-Registry Resolution**: Identity provenance passing between federated registries
- **Operation Gating**: High-trust operations require appropriate identity verification levels
- **Multi-Signature Authorization**: Critical operations require multi-party approval via Gnosis Safe

1.6 Oracle Security (Pyth Network)

Fee pricing and compliance data uses Pyth's signed multi-publisher feeds. Oracle fallback strategies include DAO-configured static rates to prevent stale data exploits. Staleness thresholds are governance parameters to balance security with usability.

Tradeoffs and Alternatives

1.1 Rationale for Polygon PoS & Solidity

Criterion	Polygon PoS (Solidity) Advantage	Alternative Risk
Transaction Cost	Near-zero (~$0.001/tx); ERC-2771 meta-transactions eliminate user gas costs	Ethereum L1 fees prohibitive for mass registration
Ecosystem & Tooling	Full EVM ecosystem: Hardhat, Foundry, OpenZeppelin, Tenderly	Niche chains require bespoke tooling and smaller developer pools
Contract Safety	OpenZeppelin proven primitives; Foundry invariant tests; Slither; Certora available	Unproven or limited audit ecosystems
Multi-Sig	Gnosis Safe — gold standard; billions in TVL; full Polygon support	Bespoke key management solutions lack battle-tested track records
Governance	Aragon natively on Polygon — no cross-chain hop needed	Cross-chain governance adds complexity and bridge risk
Auditor Pool	Broad — all major security firms audit Solidity	Narrow for niche contract languages
Human-Readable	Solidity auditable by any EVM-	Proprietary contract languages limit external

Criterion	Polygon PoS (Solidity) Advantage	Alternative Risk
	trained developer or lawyer	review

Prior rationale for Polygon PoS/Solidity is recorded in ADR-0008 for historical reference.

1.2 Governance: Dual-Token DAO Model

Landblock implements a sophisticated dual-token governance model to separate platform operations from protocol governance:

1. **Proto-DAO Phase**: Written Constitution + Founding Steward Gnosis Safe. Ensures explicit alignment before on-chain encoding.
2. **Federation Phase**: Multi-steward Gnosis Safe with bilateral registry agreements. Protocol governance through Safe proposals.
3. **DAO Phase**: Aragon on Polygon with LGT governance token. LGT holders vote on protocol decisions while LDBK powers platform operations.
4. **Mature Federation**: Global Directory governance with registry-weighted voting and Legal Reference Library oversight.

1.3 Decision: Content-Addressed Off-Chain Storage

- **Choice**: Store heavy files (GeoJSON, PDFs) on IPFS/Filecoin; hashes on Polygon PoS.
- **Benefit**: Minimum L1 cost, high privacy (no PII on chain), and mathematical integrity proof for every document.
- **Cost**: Additional infrastructure layer for storage pinning and retrieval.

1.4 Decision: Federation Over Centralization

- **Choice**: Global Directory + Federation Protocol enables cross-registry interoperability without sovereignty loss.
- **Benefit**: Political adoption feasible; existing registries can participate incrementally; neutral handling of contested territories.
- **Cost**: Complex trust relationships; bilateral agreement management; hierarchical directory maintenance.

1.5 Decision: Privacy-Preserving Verification

- **Choice**: Zero-knowledge proofs for cross-registry validation without data exposure.
- **Benefit**: Enables verification while maintaining data sovereignty; cryptographic integrity without revealing sensitive information.
- **Cost**: ZK circuit complexity; proof generation computational overhead; Circom development requirements.

Open Questions & Research Spikes

1.1 Technical Questions (Resolved)

- ✅ **Validator quorum policy**: Retired - federation model uses registry sovereignty, not quorum
- ✅ **Oracle fallback strategy**: Resolved - DAO-configured static rates with staleness thresholds
- ✅ **DID interoperability**: Resolved - SpruceID with tiered attestations and cross-registry resolution
- ✅ **ZK circuit complexity**: Active development - Circom circuits for privacy verification

1.2 Current Technical Priorities

- **Indexer determinism**: How to ensure PostgreSQL read-models remain consistent with Polygon event logs?
- **ZK proof performance**: What computational overhead is acceptable for proof generation/verification?
- **Cross-chain evidence**: When should off-chain API verification be supplemented with on-chain methods?
- **LADM Rights ≠ Parcel separation**: How to handle RRR that span multiple spatial units?

1.3 Product & Operations Questions

- **Registry Template deployment**: Self-hosted vs. hosted options for government adoption?
- **Tenure Conversion Workflow**: How to parameterize for different customary/communal tenure systems?
- **Mirror Mode incentives**: What operational benefits justify registry participation costs?
- **Application performance**: What are acceptable load times for mobile registration in low-connectivity areas?

1.4 Governance Questions

- **Federation trust tiers**: How should conformance tiers evolve as the network matures?
- **Legal Reference Library**: What governance process manages national law framework curation?
- **Emergency dispute resolution**: What triggers DAO intervention in cross-registry disputes?
- **LGT distribution model**: How to balance community rewards with protocol sustainability?

1.5 Research Spikes (Active)

- **Lending integration model**: Verification-as-a-service vs. tokenized collateral approaches

- **Geospatial standards compliance**: Full OGC/INSPIRE alignment requirements and tradeoffs
- **Contested territory handling**: Neutral escalation paths for geopolitical disputes
- **Multi-party arbitration**: Scalable dispute resolution for complex stakeholder scenarios

Roadmap

Phase 1-3: Foundation Complete (Months 1-12)

- ✅ **Governance Constitution**: Ratified proto-DAO with founding multisig on Polygon Amoy
- ✅ **Core Contracts**: 16+ Solidity modules implemented and tested (DisputeRecord, PrivacyVerifier, EvidenceStore, etc.)
- ✅ **Federation Architecture**: Global Directory and Federation Protocol designed with LADM compliance
- ✅ **Dual Token Model**: LDBK (21M fixed supply) and LGT (governance) tokens fully specified
- ✅ **Feature Documentation**: Comprehensive documentation for all major capabilities
- 🔄 **Testnet Deployment**: Core contracts ready for Amoy deployment; ZK circuits in development

Phase 4: Federation Launch (Months 13-24)

- **Registry Template**: Complete LADM-compliant registry software for governments without digital infrastructure
- **ZK Circuit Implementation**: Privacy-preserving verification circuits for cross-registry validation
- **Web Applications**: Explorer dashboard, Investor portal, and Privacy Verifier interface completion
- **Indexer Service**: PostgreSQL read-models with Coords URI resolution and event determinism
- **Government Outreach**: Mirror Mode pilots with Peru (SUNARP/COFOPRI) reference implementation

Phase 5: Global Adoption (Months 25-36)

- **Federation Network**: Multi-registry federation with bilateral trust agreements and conformance tiers
- **Aragon DAO**: Full transition to LGT-powered governance on Polygon
- **Lending Integration**: Verification-as-a-service and optional tokenized collateral for land-backed finance
- **Contested Territories**: Neutral handling framework for geopolitical disputes

- **Legal Reference Library**: DAO-governed curation of national property law frameworks

Phase 6: Ecosystem Maturity (Months 37-48)

- **Open Protocol**: Third-party implementations and registry integrations
- **Advanced Features**: Cross-chain evidence portability, multi-party arbitration, geospatial intelligence
- **Global Standards**: Recognition as authoritative land registry federation protocol
- **Sustainable Governance**: Self-funding through LDBK fees and LGT inflation model

Appendix: Glossary

Term	Definition
ADR	Architecture Decision Record — captures the rationale behind a significant decision.
Asserting Party	The individual or organization making a land claim. Does not imply legal ownership.
CID	Content Identifier — a cryptographic fingerprint of a file (IPFS). Used to verify integrity.
DID	Decentralized Identifier — a W3C-standard self-sovereign identity anchor.
Ejido	Mexican communal land tenure system — land held collectively by farming communities.
Paymaster	An ERC-2771 meta-transaction relayer allowing the protocol to pay transaction fees on behalf of users. No gas required from end users.
Gnosis Safe	The gold-standard multi-signature smart contract wallet on EVM chains. Used by Landblock for founding steward governance.
Landblock	The immutable verification protocol and platform described in this whitepaper.
Mirror Mode	Adoption Phase 1: Landblock acts as an immutable backup for an existing registry.
Polygon PoS	The EVM-compatible Proof of Stake blockchain used as Landblock's L0 consensus layer. Near-zero transaction costs and broad ecosystem support.
Solidity	The EVM smart contract language used to implement Landblock protocol contracts,

Term	Definition
	using OpenZeppelin security primitives.
ZKP	Zero-Knowledge Proof — allows proving a condition without revealing the underlying data.

Appendix: References

Internal References

- docs/01-overview/README.md
- docs/01-overview/CONSTITUTION.md
- docs/architecture/overview.md (Architecture System of Record)
- docs/05-decisions/ (Architecture Decision Records)
- docs/05-decisions/ADR-0008-Blockchain-Migration-Polygon PoS-to-Polygon.md (Migration Rationale)
- docs/phases/phases.md

External References

- **Polygon PoS**: polygon.technology (EVM-compatible Proof of Stake chain)
- **OpenZeppelin**: openzeppelin.com (Solidity security primitives and contract standards)
- **Gnosis Safe**: safe.global (Multi-signature smart contract wallet)
- **Foundry**: getfoundry.sh (Solidity testing framework including invariant tests)
- **SpruceID**: spruceid.com (Decentralized Identity & ZKP Proofs)
- **Pyth Network**: pyth.network (Low-latency Oracle Feeds)
- **Aragon**: aragon.org (Modular DAO Governance)
- **IPFS**: ipfs.tech (Content-addressed Storage)
- **Filecoin**: filecoin.io (Decentralized Storage Persistence)