EB-SPEC-001 · PROTOCOL SERIES

Technical Protocol Specification

The normative engineering reference for the Earth Blockchain Layer-1 protocol: cryptographic primitives, network model, data structures, Proof of Staked Authority consensus, execution semantics, staking and slashing, system contracts, governance mechanics, and the canonical parameter registry.

NORMATIVE | POSA CONSENSUS | EVM EXECUTION | SYSTEM CONTRACTS | GREEN VALIDATOR NETWORK

Version 1.0 · **Status** Public Release · **Date** July 2026

Canonical source Earth Blockchain Whitepaper v2.0

Audience Protocol engineers · node implementers · security auditors · researchers

EB-SPEC-001 · PROTOCOL SERIES · VERSION 1.0 · JULY 2026

Technical Protocol Specification

NORMATIVE ENGINEERING REFERENCE · EARTH BLOCKCHAIN LAYER-1

Status of This Document

This document is the normative engineering specification of the Earth Blockchain protocol. It expands the Earth Blockchain Whitepaper v2.0 — the canonical design reference — with implementation-level definitions: cryptographic primitives, wire and state data structures, the Proof of Staked Authority (PoSA) consensus state machine, execution semantics, staking and slashing rules, system contracts, governance mechanics, the canonical bridge protocol, and a consolidated parameter registry. Where this document and the Whitepaper describe the same mechanism, the Whitepaper defines intent and this document defines behavior; any conflict is a defect in this document.

All genesis parameter values stated here are normative at network genesis and remain subject to modification exclusively through EarthDAO governance (§9), as provided in Whitepaper §10 and §16.

Conventions and Normative Language

The key words MUST, MUST NOT, SHALL, SHOULD, SHOULD NOT, and MAY in this document are to be interpreted as described in RFC 2119. Byte sequences are big-endian unless stated otherwise. `||` denotes byte concatenation, `H(·)` denotes Keccak-256, and `RLP(·)` denotes Recursive Length Prefix encoding. Quantities of EARTH are expressed in the base unit *gr* (1 EARTH = 10^{18} gr), mirroring EVM conventions so existing tooling operates unmodified.

> **Reading map.** Implementers building a node: §2–§8. Auditors reviewing economic security: §5, §7, §12–§13. Bridge and infrastructure operators: §10–§11. All readers should retain Whitepaper v2.0 alongside this specification.

Table of Contents

1. Introduction and Scope

WHAT THIS SPECIFICATION DEFINES

Earth Blockchain is an EVM-compatible Layer-1 network secured by Proof of Staked Authority, operated on the Green Validator Network, and denominated in the EARTH native utility currency (Whitepaper §4–§6, §11). This specification defines the protocol precisely enough that an independent team could implement an interoperable node: the objects that appear on the wire and in state, the rules by which validators produce and finalize blocks, the conditions under which stake is slashed, the semantics of execution and fees, and the privileged system contracts through which staking, reputation, governance, and slashing are administered.

1.1 In Scope

- **Consensus-critical behavior.** Anything on which two honest nodes must agree: encodings, state transitions, validity rules, finality, and slashing conditions.
- **System-contract interfaces.** The addresses and externally observable behavior of protocol predeploys.
- **Genesis parameters.** The values in the Canonical Parameter Registry (§13).

1.2 Out of Scope

Client architecture, database layouts, peer-scoring heuristics, mempool policy beyond validity, and BaaS service design are implementation-defined and covered by the Developer Documentation (EB-DEV-002) and Validator Handbook (EB-VAL-004). Application-level standards (EBS-20 through EBS-6551) are protocol-registered contract standards, specified operationally in the Smart Contract Cookbook (EB-SCC-005); this document defines only their registry (§8.5).

2. Cryptographic Primitives and Identifiers

HASHING · SIGNATURES · ADDRESSES · AGGREGATION

Primitive	Definition
Hash function	Keccak-256 for all protocol hashing: transaction hashes, block hashes, state trie nodes, and log topics — preserving EVM-ecosystem compatibility.
Transaction signatures	ECDSA over secp256k1 with EIP-155 replay protection. Signature recovery yields the sender address; nodes MUST reject signatures with non-canonical *s* values (s > n/2).
Addresses	20 bytes: `addr = H(pubkey)[12:32]`, hex-encoded with EIP-55 checksum casing at presentation layers.
Consensus attestations	BLS signatures over BLS12-381 with proof-of-possession registration, enabling constant-size aggregation of the quorum certificate (§5.5). Each validator registers a BLS attestation key bound to its secp256k1 operator identity in the Staking Contract (§8.1).
Randomness for rotation	Epoch seed `seed(e) = H(finalized_hash(last block of e−1) || e)`, used only for tie-breaking within the deterministic proposer schedule (§5.4); no external randomness beacon is required for safety.
Chain identifier	A fixed EIP-155 chain ID assigned at genesis and published with the genesis artifacts; wallets and tooling MUST include it in every signature.

Key separation (normative). The operator key (secp256k1, holds stake authority), the attestation key (BLS, signs consensus messages), and any withdrawal address MUST be distinct key materials. Compromise of the attestation key alone MUST NOT permit stake withdrawal.

3. Network Model and Node Roles

TOPOLOGY · GOSSIP · SYNCHRONY ASSUMPTIONS

Nodes communicate over an authenticated, encrypted peer-to-peer overlay (devp2p-compatible transport with protocol extensions for attestation gossip). The network is partially synchronous: safety of finalized blocks holds under arbitrary network delay, while liveness assumes messages between honest validators are delivered within a known bound Δ after unknown global stabilization time — the standard partial-synchrony model under which the f < n/3 fault bound (Whitepaper §6.1) is meaningful.

3.1 Node Roles

Role	Duties	Typical Tier (Whitepaper §5)
Active validator	Proposes blocks per schedule, attests every block, serves state; bonded and slashable.	All tiers; enterprise tier for sustained throughput.
Standby validator	Registered and bonded, outside the active set; follows the chain, accrues uptime reputation as observer-attester.	Community / solar tiers.
Full node	Validates all blocks and executes all transactions without consensus duties; serves RPC.	Any.
Archive node	Full node retaining all historical state for indexing, explorers, and audit queries.	Enterprise.
Light client	Verifies quorum certificates and header chains only; trusts finality proofs, not peers.	Embedded / mobile / bridge verifiers.

3.2 Gossip Channels

Three gossip channels are consensus-relevant and MUST be implemented: `txs` (signed transactions), `blocks` (proposals with body), and `attest` (BLS attestation shares and aggregated quorum certificates). Nodes SHOULD apply per-peer rate limits and MUST drop messages failing signature or basic validity checks before propagation.

4. Data Structures

BLOCKS · TRANSACTIONS · STATE · RECEIPTS

4.1 Block Header

The header follows the Ethereum layout with PoSA-specific fields replacing proof-of-work artifacts. Fields marked ◆ are PoSA-specific.

Field	Type	Semantics
parentHash	B32	Keccak-256 of the parent header.
stateRoot	B32	Root of the global Merkle-Patricia state trie after executing this block.
transactionsRoot / receiptsRoot	B32	Trie roots over the block's transactions and receipts.
logsBloom	B256	Standard 2048-bit log bloom filter.
number / timestamp	uint	Height; Unix seconds. `timestamp` MUST exceed the parent's and SHOULD track the 3 s slot cadence.
gasLimit / gasUsed	uint	Block gas ceiling and consumption.
baseFeePerGas	uint	Protocol base fee for this block (§6.2).
◆ proposer	B20	Operator address of the scheduled proposer for this slot.
◆ epoch	uint	Epoch index `e = ⌊number / 28800⌋`.
◆ validatorSetHash	B32	Commitment to the active set (operator, attestation key, stake) in force for this epoch; light clients verify set transitions against it.
◆ quorumCert	bytes	Aggregated BLS signature + participation bitmap over the *parent* block's finality payload (§5.5). Genesis' certificate is empty.

Fig. 1 — Block Anatomy and Finality Linkage

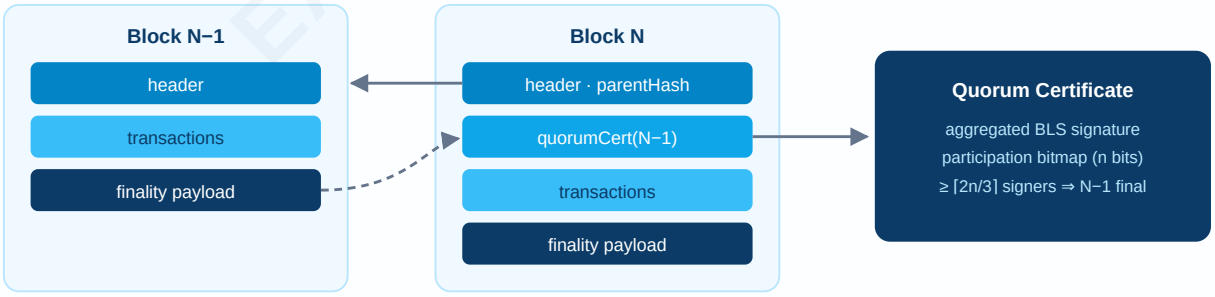

Fig. 1. Block anatomy. The finality payload of block N−1 is what validators sign; block N carries the resulting certificate.

4.2 Transactions

Nodes MUST accept legacy (type 0x0), EIP-2930 access-list (0x1), and EIP-1559 dynamic-fee (0x2) transaction envelopes. Type 0x2 is RECOMMENDED; its fields `maxFeePerGas` and `maxPriorityFeePerGas` interact with the base-fee rule of

§6.2. Intrinsic validity — signature, nonce ordering, balance ≥ maximum cost, gas ≥ intrinsic gas — matches Ethereum semantics exactly, preserving wallet and tooling compatibility (Whitepaper §7.1).

4.3 World State and Receipts

World state is the standard account model: `(nonce, balance, storageRoot, codeHash)` per account in a Merkle-Patricia trie. Receipts record status, cumulative gas, logs, and bloom. EARTH balances are protocol-native state — EARTH is not an EBS-20 contract (Whitepaper §11); the canonical wrapped form wEARTH is a system predeploy (§8.5) for contract composability.

5. Consensus: PoSA Formal Specification

EPOCHS · SELECTION · PROPOSAL · FINALITY · FORK CHOICE · SLASHING

5.1 Time Structure

Time is divided into **slots** of 3 s, one block per slot, and **epochs** of 28,800 slots (≈24 h). Validator-set changes, reward settlement, and reputation updates occur only at epoch boundaries; within an epoch the active set is immutable, which keeps quorum verification O(1) against the header's `validatorSetHash`.

Fig. 2 — Epoch Timeline and Set Rotation

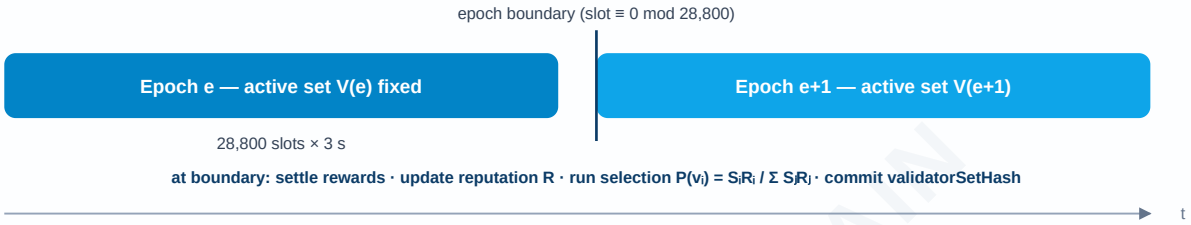

Fig. 2. Slots, epochs, and the boundary procedure. All economic settlement is epoch-atomic.

5.2 Validator Set Selection

At each boundary the protocol admits n = 21 validators (genesis size) from the bonded candidate pool by sampling without replacement with probability proportional to stake-weighted reputation, exactly as defined in Whitepaper §6.2:

$$P(v_i) = (S_i \cdot R_i) \,/\, \Sigma_j \, (S_j \cdot R_j)$$

where S_i is bonded stake (self + delegated, in gr) and $R_i \in [0,1]$ the on-chain reputation score. Sampling uses `seed(e)` (§2) so every node derives the identical set deterministically. Candidates MUST satisfy the minimum self-stake and registration requirements of §8.1 to enter the pool.

5.3 Reputation Function

$$R_i = \text{clamp}(\ 0.4 \cdot U_i + 0.4 \cdot A_i - 0.2 \cdot F_i,\ \ 0,\ 1\ )$$

U_i: trailing 30-epoch uptime ratio. A_i: signing accuracy — valid, timely attestations divided by expected duties over the same window. F_i: fault index, incremented per recorded infraction and decayed as `F ← F·2^(−1/14)` per epoch (14-epoch half-life). New registrants enter at R = 0.5. All three inputs derive from on-chain evidence recorded in the Reputation Registry (§8.2); no discretionary input exists (Whitepaper §6.3).

5.4 Proposer Schedule

Within an epoch, slots are assigned by a reputation-weighted round-robin: the active set is ordered by `H(seed(e) || operator)`, and each validator receives a contiguous quota of slots proportional to R_i (minimum one slot per rotation cycle). Each slot also deterministically designates two **backup proposers** (the next validators in the ordering). If no valid proposal from the primary is observed within 1.5 s of slot start, backups MAY propose in order; attesters prefer the highest-priority valid proposal received. This preserves the 3 s cadence under individual proposer failure (Whitepaper §6.4).

5.5 Attestation and Deterministic Finality

Upon receiving a valid proposal for slot t, each active validator MUST verify it (§5.6) and broadcast a BLS attestation share over the block's **finality payload** `FP = H(blockHash || number || epoch || validatorSetHash)`. Any node aggregating shares from at least **Q = ⌈2n/3⌉** distinct active validators (Q = 14 at n = 21) forms the quorum certificate. A block is **final** when a valid certificate over its FP exists; the certificate is embedded in the child block's header (Fig. 1). Under f < n/3 Byzantine validators, two conflicting blocks at the same height cannot both obtain certificates — each would require ≥ Q signers, forcing ≥ 2Q − n > n/3 validators to equivocate, all of whom are slashable under condition S2 (§5.7).

Fig. 3 — PoSA Slot State Machine (per validator)

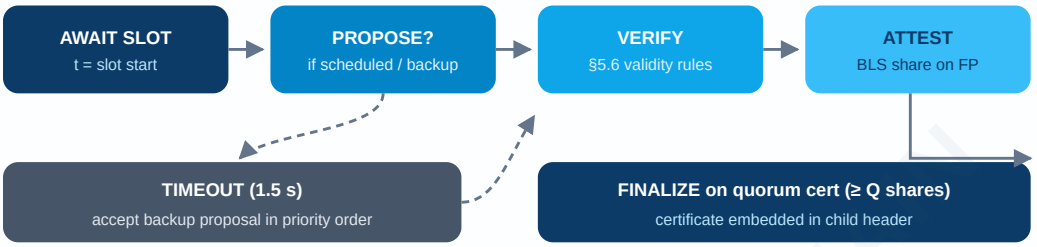

Fig. 3. Per-slot behavior of an active validator: propose when scheduled, verify, attest, finalize; timeouts route to backup proposals without breaking cadence.

5.6 Block Validity Rules

A proposal is valid iff all hold: (a) header fields well-formed and parent known and final or pending-final; (b) `proposer` matches the schedule for the slot (primary or eligible backup after timeout); (c) embedded `quorumCert` verifies against the parent's FP and the epoch's `validatorSetHash`; (d) every transaction is intrinsically valid and execution yields the claimed `stateRoot`, `receiptsRoot`, and `gasUsed`; (e) `baseFeePerGas` equals the value prescribed by §6.2. Attesting to an invalid block is itself a slashable fault (S3).

5.7 Fork Choice and Slashing Conditions

Fork choice. The canonical chain is the chain of certified (final) blocks; among children of the last final block, nodes follow the valid proposal of the highest-priority proposer for the current slot. Because finality is per-block and deterministic, reorganizations of final blocks are impossible under the fault bound — there is no probabilistic head to contest (Whitepaper §6.4, §6.7).

Slashing conditions. Evidence is submitted by any party to the Slashing Contract (§8.3); penalties apply `Penalty = α·S` per Whitepaper §6.6 with slashed stake routed to the Treasury:

ID	Condition (evidence)	Class	α range
S1	Missed duties beyond grace thresholds (uptime records)	Liveness	0.01 – 0.05
S2	Two signed attestations for conflicting FPs at the same height (both shares)	Equivocation	0.50 – 1.00
S3	Attestation over a block failing §5.6 (block + share)	Invalid-state	0.05 – 0.20
S4	Participation in a certificate for a fork conflicting with a final block (certificate)	Finality attack	1.00

Bonded stake remains slashable through the entire 21-day unbonding period, so evidence discovered after exit initiation still binds (Whitepaper §5.2, §6.7).

6. Execution Layer

EVM EQUIVALENCE · GAS · BASE FEE · BURN

6.1 EVM Equivalence

The execution layer implements the Ethereum Virtual Machine instruction set, gas schedule, and precompiles of the protocol's pinned EVM baseline (published with genesis artifacts and upgraded only via EarthDAO hard forks, per Whitepaper §7.1). Contracts, tooling, and audits valid on the baseline EVM are valid here without modification. Deviations from the baseline are limited to: PoSA header fields (§4.1), system predeploys (§8), and the fee-burn split (§6.3).

6.2 Base-Fee Update Rule

Each block carries a protocol-computed `baseFeePerGas`. With target gas `T = gasLimit/2` and elasticity denominator 8:

$$\text{BaseFee}_{t+1} = \text{BaseFee}_t \times \left(1 + (\text{GasUsed}_t - T) / (T \times 8) \right)$$

bounded below by a 1 gr floor. A transaction of type 0x2 pays `min(maxFeePerGas, BaseFee + maxPriorityFeePerGas)` per gas; the priority component is the tip. This damping rule yields the fee predictability required for enterprise workloads (Whitepaper §7.2) — at 3 s slots, even sustained full blocks move the base fee ≤ 12.5% per block.

6.3 Fee Split and Burn

$$\text{Burned}_t = \beta \times \text{BaseFee}_t \times \text{GasUsed}_t, \quad \beta = 0.30 \text{ (genesis)}$$

The burned share is destroyed at block execution; the remainder of the base fee and all tips accrue to the epoch reward pool for pro-rata settlement (§7). β is EarthDAO-governable within [0,1] and implements the deflationary counterbalance `NetEmission = GrossEmission − β·TotalGasFees` of Whitepaper §15.6.

Invariant. No execution path mints EARTH except the epoch reward settlement drawing on the 27.0B emission pool (§7.2). Total supply **MUST NEVER** exceed 90,000,000,000 EARTH (Whitepaper §15.1).

7. Staking, Delegation, and Reward Settlement

BONDING · EMISSION DRAW · PRO-RATA SETTLEMENT · UNBONDING

7.1 Bonding and Delegation

Operators register with the Staking Contract (§8.1), binding operator address, BLS attestation key, endpoints, hardware/energy declaration, and minimum self-stake. Delegators bond EARTH to a chosen validator, sharing rewards and slashing risk pro-rata; validator commission $c_i \in [0,1]$ is declared on-chain and changes take effect only at epoch boundaries with a one-epoch notice, preventing retroactive commission changes on earned rewards.

7.2 Emission Draw

The epoch's gross emission is the discrete draw on the decaying curve of Whitepaper §15.4 with k = ln 3 / 5 per year:

$$\text{Emit}(e) = E_{\text{pool}} \times \left(e^{-k \cdot t(e)} - e^{-k \cdot t(e+1)} \right), \quad E_{\text{pool}} = 27.0\text{B EARTH}$$

where t(e) is the epoch start in years since the Genesis Distribution Event. Emission is computed in gr with round-down; residue remains in the pool. The epoch reward pool equals `Emit(e) + (1−β)·baseFees(e) + tips(e)`.

7.3 Settlement

$$\text{Reward}_i = \text{Pool}(e) \times (S_i / S_{\text{total,active}})$$

$$\text{Reward}_d = (1 - c_i) \times \text{Reward}_i \times (S_d / S_i)$$

identical to Whitepaper §6.5, executed atomically in the epoch-boundary system transaction. Rewards accrue as claimable balances in the Staking Contract; claiming is a normal transaction and does not affect consensus.

7.4 Unbonding and Exit

Unbond requests enter a 21-day queue during which the stake earns no rewards, retains no selection weight, and remains fully slashable. Governance-initiated removal (Whitepaper §10.3, validator framework) follows the same queue. Withdrawal executes automatically at queue maturity to the registered withdrawal address.

8. System Contracts

PROTOCOL PREDEPLOYS · PRIVILEGED INTERFACES

Protocol logic that must be state-visible and governable is implemented as **system contracts** predeployed at reserved addresses in the range `0x…1000–0x…10FF` (full addresses published with genesis artifacts). System contracts are upgradeable exclusively through EarthDAO-executed proposals (§9); no external account holds administrative keys after the decentralization milestones of Whitepaper §16.

§	Contract	Responsibility
8.1	StakingContract	Registration, bonding, delegation, commission, unbonding queue, claimable rewards, withdrawal addresses, BLS key registry.
8.2	ReputationRegistry	Per-epoch U, A, F inputs and computed R; append-only evidence log for auditability.
8.3	SlashingContract	Evidence intake for S1–S4, verification, penalty computation α·S, Treasury routing, double-jeopardy guards per evidence hash.
8.4	GovernanceCore + Timelock + Treasury	Proposal lifecycle, voting-power accounting with lock multiplier, 48 h timelock execution, DAO Reserve custody (receives slashed stake and forfeited deposits).
8.5	EBSRegistry + wEARTH	Canonical registry of audited EBS reference implementations (EBS-20…EBS-6551) and the canonical wrapped-EARTH predeploy.
8.6	BridgeGateway	Lock/mint and burn/release execution, attestation-set key registry, per-asset rate limits, guardian pause (sunset per roadmap).

8.1 Illustrative Interface (StakingContract, excerpt)

```
interface IStaking {
  // operator lifecycle
  function register(bytes blsKey, bytes32 endpointsHash, bytes32 energyAttn) payable;
  function delegate(address validator) payable;
  function undelegate(address validator, uint256 amountGr);   // enters 21-day queue
  function setCommission(uint16 bps);                          // effective e+2
  function claimRewards(address account) returns (uint256 gr);
  // views consumed by consensus
  function activeSet(uint64 epoch) view returns (address[] memory, uint256[] memory);
  function bondedStake(address validator) view returns (uint256 gr);
}
```

Interfaces are illustrative excerpts; canonical ABIs ship with genesis artifacts and the SDK & API Documentation (EB-API-003).

9. Governance Mechanics

PROPOSAL OBJECT · THRESHOLDS · EXECUTION PATH

EarthDAO mechanics implement Whitepaper §10 exactly. Voting power for participant i is:

$$VP_i = G_i \times (1 + \lambda \cdot L_i / L_{max}), \quad \lambda = 1, L_{max} = 24 \text{ months}$$

A proposal is an on-chain object `(targets[], values[], calldatas[], descriptionHash)` submitted with a 1,000,000 EARTH deposit. Lifecycle states and transitions:

State	Transition Rule
Pending → Active	One-epoch review delay after submission.
Active → Succeeded	After 7 days: participation ≥ 20% of governance-staked EARTH, and approval > 50% (standard) or ≥ 66.7% (protocol-upgrade class, i.e., any proposal touching system contracts or registry parameters §13).
Active → Defeated	Otherwise; 50% of the deposit is forfeited to the Treasury, the remainder refunded.
Succeeded → Queued → Executed	Enqueued in the 48 h Timelock; after expiry, anyone MAY trigger execution; calls run atomically from the Timelock, the only address privileged to modify system contracts.

Integrator guarantee. Because every consensus-visible parameter change passes the Timelock, node operators and enterprises always have ≥ 48 h notice of protocol change — the reaction window promised in Whitepaper §10.2.

10. Canonical Bridge Protocol

MESSAGE FORMAT · ATTESTATION · RATE LIMITS

The canonical bridge implements the lock-and-mint / burn-and-release design of Whitepaper §9.1 with an attestation set of k keyholders (validator-operated, keys distinct from consensus keys) of which m MUST co-sign; genesis m-of-k is 7-of-10, EarthDAO-governable.

10.1 Transfer Message

```
TransferMsg = RLP(
  srcChainId, dstChainId,      // EIP-155 identifiers
  srcTxHash,  logIndex,        // provenance of the lock/burn event
  asset, amountGr,             // canonical asset id + quantity
  recipient, nonce )           // destination account + per-route nonce
digest = H(TransferMsg)
```

Attesters sign `digest` only after the source event is *final*: for Earth Blockchain sources, inclusion under a quorum certificate (§5.5); for external sources, the chain-specific finality policy published in the bridge registry. Nonces are strictly sequential per (route, asset), making replay and reordering detectable on-chain.

10.2 Enforcement

- **Rate limits.** Per-asset rolling-window mint/release ceilings; transfers above the ceiling queue rather than fail, bounding worst-case loss from attester compromise.
- **Pause.** A guardian process MAY pause routes pending EarthDAO review; guardian powers sunset per the decentralization schedule (Whitepaper §16).
- **Representation.** Inbound assets mint as wrapped EBS-20 with canonical metadata; outbound EARTH is represented externally per Whitepaper §15.1.

Fig. 4 — Bridge Transfer Sequence (inbound)

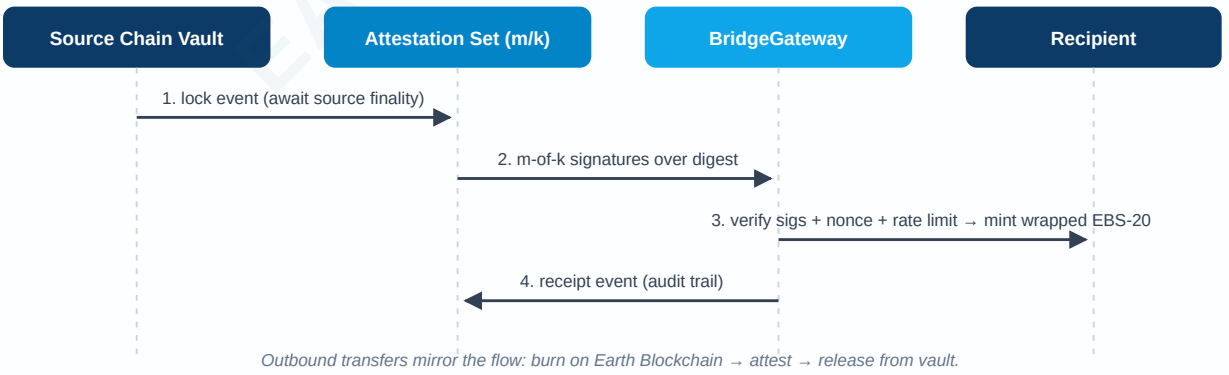

Fig. 4. Inbound canonical-bridge sequence with finality-gated attestation and on-chain enforcement.

11. Node Interfaces (JSON-RPC)

STANDARD NAMESPACE · EARTH EXTENSIONS

Nodes expose the standard Ethereum JSON-RPC namespace (`eth_`, `net_`, `web3_`) with unchanged semantics — existing SDKs connect without modification (Whitepaper §7.1) — plus an `earth_` namespace for PoSA-specific data:

Method	Returns
earth_getValidatorSet(epoch)	Active set: operator addresses, BLS keys, bonded stake, reputation R.
earth_getQuorumCert(blockHash)	Aggregated signature, participation bitmap, signer count — the finality proof consumed by light clients and bridges.
earth_getFinalizedHead()	Highest certified block (hash, number, epoch).
earth_getEpochInfo(epoch)	Boundary block, seed, emission drawn, fee burn, reward pool.
earth_getReputation(validator)	Current R with U, A, F components and evidence references.
earth_getSlashingEvents(range)	Recorded S1–S4 events with evidence hashes and applied α.

Full request/response schemas, error codes, and streaming (WebSocket) semantics are specified in the SDK & API Documentation (EB-API-003).

12. Security Considerations

THREATS, MITIGATIONS, AND RESIDUAL ASSUMPTIONS

- **Safety bound.** All finality guarantees assume f < n/3 Byzantine weight in the active set. The economic analysis of Whitepaper §6.7 (cost of corruption = slashable stake + reputation destruction) is the operative deterrent; implementers MUST NOT weaken slashing evidence checks, which would convert an economic bound into an honor system.
- **Long-range and key-leak attacks.** Deterministic finality plus 21-day slashable unbonding closes long-range rewrites; light clients MUST refuse validator-set transitions not certified by the previous set (`validatorSetHash` chain).
- **Proposer denial / censorship.** Backup proposers bound censorship to seconds; sustained censorship requires ≥ Q collusion, which is publicly visible in participation bitmaps and governance-actionable.
- **Time manipulation.** Timestamp rules (§4.1) prevent drift-based fee or schedule gaming; nodes SHOULD use disciplined clocks (NTP or better) and MUST reject blocks from the future beyond 1.5 s.
- **Bridge risk.** The attestation set is a distinct trust domain; rate limits cap worst-case loss, and finality-gating removes reorg-based double-mints. Operators of external-chain verifiers MUST respect each source's published finality policy.
- **System-contract upgrade risk.** The Timelock is the single privileged writer; monitoring its queue is the primary operational control for exchanges, custodians, and enterprises.
- **Implementation hygiene.** BLS libraries MUST enforce subgroup checks and proof-of-possession; RLP decoders MUST be strict; state execution MUST be deterministic across platforms (no floating point in consensus code).

13. Canonical Parameter Registry

GENESIS VALUES · ALL EARTHDAO-GOVERNABLE UNLESS MARKED FIXED

Parameter	Genesis Value	Reference
Active set size n	21	§5.2; WP §6.1
Slot time	3 s	§5.1; WP §6.1
Epoch length	28,800 slots	§5.1; WP §6.1
Quorum Q	⌈2n/3⌉ = 14	§5.5; WP §6.4
Fault tolerance	f < n/3	§5.5; WP §6.1
Proposer timeout	1.5 s	§5.4
Reputation weights (w_u, w_a, w_f)	0.4 / 0.4 / 0.2	§5.3; WP §6.3
Reputation windows	30-epoch inputs; F half-life 14 epochs; entry R = 0.5	§5.3
Slashing α by class	S1 0.01–0.05 · S3 0.05–0.20 · S2 0.50–1.00 · S4 1.00	§5.7; WP §6.6
Unbonding period	21 days	§7.4; WP §6.1
Base-fee elasticity	target = gasLimit/2; denominator 8	§6.2
Fee-burn share β	0.30	§6.3; WP §15.6
Emission pool / constant	27.0B EARTH; k = ln 3 / 5 yr^{-1}	§7.2; WP §15.4
Maximum supply	90,000,000,000 EARTH — **fixed**	§6.3; WP §15.1
Governance deposit	1,000,000 EARTH	§9; WP §10.2
Voting period / timelock	7 days / 48 hours	§9; WP §10.2
Quorum / thresholds	20% participation; >50% standard; ≥66.7% upgrade class	§9; WP §10.2
Lock multiplier (λ, L_{max})	1; 24 months	§9; WP §10.1
Bridge attestation m-of-k	7 of 10	§10
Base unit	1 EARTH = 10^{18} gr	Conventions

References

NORMATIVE AND INFORMATIVE

[1] Earth Blockchain Whitepaper v2.0 (July 2026) — canonical design reference for architecture, consensus intent, products, EBS standards, governance, and tokenomics. earth.ooo.

[2] Earth Blockchain public platform and explorer — earth.ooo · scan.earth.ooo, retrieved July 2026.

[3] RFC 2119 — Key words for use in RFCs to Indicate Requirement Levels.

[4] Ethereum ecosystem specifications informing compatibility surfaces: RLP encoding; EIP-155 (replay protection); EIP-55 (checksums); EIP-1559 (fee market); EIP-2930 (access lists); JSON-RPC namespace conventions (eips.ethereum.org).

[5] BLS signature schemes over BLS12-381 with proof-of-possession, as standardized in the IETF CFRG draft series — informative basis for §2 and §5.5.

EB-SPEC-001 v1.0 (July 2026). This specification describes the Earth Blockchain protocol design as finalized for public launch. Genesis parameter values are normative at genesis and modifiable only through EarthDAO governance. Interface excerpts are illustrative; canonical ABIs and genesis artifacts are published at earth.ooo. This document does not constitute financial, investment, tax, or legal advice.