Entangle Protocol: A Decentralised Cross-Chain Message Relay Subnet on Bittensor

Entangle Protocol Research Group

research@entangleprotocol.com https://entangleprotocol.com

Protocol Version 2.0

ABSTRACT.

The Web3 ecosystem suffers from deep structural fragmentation: sovereign blockchains cannot interoperate natively, trusted bridge architectures have lost over $2.8 billion to exploits, and existing cross-chain messaging protocols concentrate relay authority within small, permissioned operator sets. This paper introduces **Entangle Protocol**, a decentralised cross-chain message relay subnet deployed on Bittensor, with initial support spanning **13 blockchain networks** across five distinct ecosystems: EVM (Ethereum, Arbitrum, Optimism, Base, Polygon, BNB Chain, Avalanche), Solana, SUI, Stellar, and Cosmos (Hub, Osmosis, Neutron), with the network designed to expand continuously as new chains are onboarded.

Entangle rejects the centralised relay paradigm entirely. Instead, it instantiates a **dual-miner architecture**-*Scanner Miners* that detect source-chain events via competitive timestamp-race scoring, and *Relay Miners* that execute destination-chain delivery through sealed-bid auctions-both governed by Bittensor's Yuma Consensus and rewarded with TAO emissions split across discovery and execution roles in a configurable ratio (default 30/70). Entry is **permissionless**: any operator registers by paying the dynamic TAO burn with no guardian whitelists, no DVN gatekeeping, and no KYC requirements.

The security model is grounded in threshold attestation requiring signatures from validators controlling a configurable percentage of registered stake (default $>$60%), anchored to secp256k1 key schemes uniformly across all supported ecosystems including EVM, Solana, SUI, Stellar, and Cosmos. Cross-chain replay is eliminated via a 4-tuple deduplication key enforced uniformly across all five smart contract ecosystems. A dynamic gas oracle, updated at a configurable interval (default $\sim$2 minutes) via stake-weighted validator consensus with a configurable safety buffer (default 1.25$\times$), ensures relay miners are never systematically underfunded by gas price volatility.
Economically, Entangle implements a **dual-flywheel tokenomics model**: TAO emissions drive initial liquidity while protocol fee buybacks-funded by a configurable share of every message fee (default 100%)-provide a real-revenue demand floor independent of TAO market conditions. Score weighting per chain is adjustable to reflect each chain's relative gas costs, ensuring fair and representative miner incentivisation across heterogeneous networks.

1. INTRODUCTION

Blockchain technology has fulfilled the foundational promise of trustless value exchange *within* individual networks. It has not solved value and data exchange *between* networks. Today, over 13 major blockchains operate as sovereign islands-Ethereum cannot natively communicate with Solana; Cosmos chains cannot compose with EVM L2s; Stellar assets have no mechanism to participate in on-chain protocols on other networks without trusted intermediaries. The network is architected to onboard new chains continuously, with no upper bound on chain coverage.

Solutions deployed to bridge this gap-centralised relay architectures, validator multisig relays, and permissioned DVN networks-have collectively failed along two axes. Architecturally, they create honeypots: concentrated custodied value that rational adversaries target. Economically, they create oligopolies: the top three relayer protocols control 78% of cross-chain volume through operator sets accessible only to approved parties.

Entangle Protocol proposes a third model, drawing from the design philosophy of foundational internet protocols: **establish an open standard and let permissionless competition do the rest.** TCP/IP did not require router operators to apply for permission; it specified the packet format and let the market build the infrastructure. Entangle specifies the relay protocol and lets any operator compete for rewards by providing the service.

The enabling technology is Bittensor-a decentralised network designed specifically for incentivising competitive digital intelligence. Bittensor's Yuma Consensus aggregates validator assessments of miner quality and distributes TAO proportionally to performance. Entangle instantiates this as a two-sided relay market: a discovery layer that races to detect on-chain events, and an execution layer that competes to deliver them.

1.1 Key Innovation

- A **permissionless dual-miner architecture** splitting event detection (Mechanism 1, configurable emission share) from message delivery (Mechanism 2, configurable emission share)
- **Continuously expanding multi-ecosystem smart contract coverage** via a unified `ChainAdapter` interface requiring no new contracts for same-ecosystem chain additions
- A **sealed-bid auction relay market** with five-dimensional

execution scoring that aligns miner incentives precisely with dApp quality requirements, with per-chain score weights configurable to reflect each chain's gas cost profile
- A **dual-flywheel tokenomics model** decoupling relay miner economics from TAO price volatility through real-revenue fee buybacks
- A **threshold attestation security model** requiring a configurable validator stake consensus threshold with anti-gaming protections at every scoring pipeline layer

2. PROBLEM SPACE

2.1 The Fragmented Web3 Stack

The blockchain ecosystem has converged on a pattern of vertical integration: each chain optimises internally while cross-chain communication is delegated to auxiliary infrastructure that has proven to be the weakest link in the security chain.

$2.8B lost to relay and bridge exploits (2022–2024)
67% of crypto hacks target cross-chain contracts
78% of cross-chain volume controlled by top 3 protocols

2.2 Structural Failures of Existing Solutions

Centralised Relay Architectures. Many cross-chain messaging implementations delegate critical relay decisions to small trusted operator committees. This concentrates risk: a single validator collusion event or key compromise can affect the entire relay pipeline. The mechanism is architecturally fragile regardless of operator quality.

Permissioned Relayer Sets. Wormhole's Guardian network, LayerZero's DVN model, and Axelar's validator set all require aspiring operators to satisfy off-chain approval criteria. This is not decentralisation-it is federated trust with extra steps, reducing censorship resistance and creating durable oligopolies.

Single-Chain Specialisation. Most relay protocols support a limited chain set through bespoke per-chain deployments. Adding a new chain requires governance approval and custom contract development, creating perpetual lag behind chain ecosystem growth.

2.3 The Entangle Thesis

Cross-chain message relay is a **digital commodity**: fungible, measurable, and improvable through competition. Like packet routing on the internet, it does not require operator identity-only operator performance. Entangle's thesis is that instantiating this commodity market on Bittensor's incentive infrastructure creates a relay network that is simultaneously more secure (no custodied assets), more decentralised (permissionless entry), and higher quality (performance-based rewards) than any existing alternative.

3. SYSTEM ARCHITECTURE

3.1 Three-Phase Message Lifecycle

Every cross-chain message processed by Entangle passes through three deterministic phases illustrated in Figure 1.

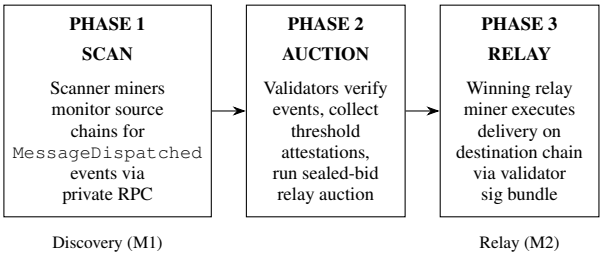

Figure 1: Entangle three-phase message lifecycle with TAO emission allocation.

3.2 Network Topology

Figure 2 shows the full network topology from the Bittensor metagraph through validators and both miner types to source and destination chains.

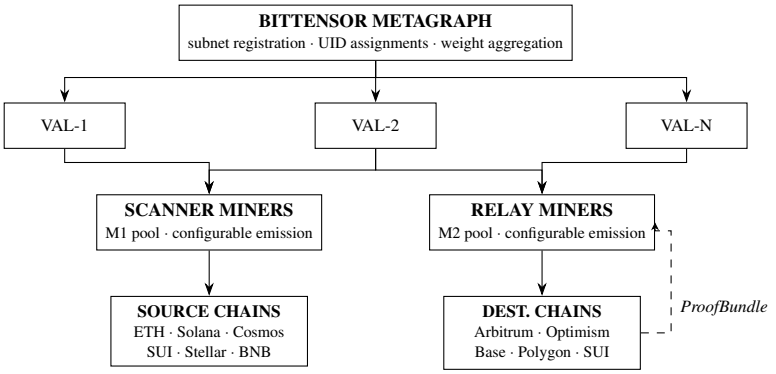

Figure 2: Network topology: Bittensor metagraph orchestrates validators, which score both miner types connected to source and destination chains.

3.3 Design Principles

Permissionless Entry. Any operator registers by paying the dynamic TAO burn. No approval process, no guardian whitelist, no DVN gatekeeping.

Objective Scoring. Every score dimension uses on-chain or cryptographically verifiable data. Validator receipt timestamps prevent self-reporting manipulation entirely.

One Contract Per Chain. A single Entangle deployment handles both `sendMessage` (source) and `verifyMessage` (destination) roles on every supported blockchain.

ChainAdapter Interface. New chains are added without deploying new contracts. Same-ecosystem chains are zero-config additions; new ecosystems need one Python class implementing five abstract methods.

Deterministic Task IDs. Task identifiers are computed identically by all validators, enabling cross-validator score consistency without out-of-band coordination:

```
task_id = sha256(
  sha256(src_chain + ":" + protocol_addr
         + ":" + seq + ":" + src_tx)
  + ":" + validator_hotkey[:8]
)
```

4. THE DUAL-MINER MODEL

Entangle uses Bittensor's multi-mechanism capability to run two independent scoring systems on a single subnet, each with its own task type, miner population, and emission pool.

Table 1: Dual-Miner Mechanism Summary

Attribute	Mechanism 1 (Discovery)	Mechanism 2 (Relay)
Emission share	Configurable (default 30%)	Configurable (default 70%)
Miner role	Scanner	Relay
Core task	Detect `MessageDispatched`	Execute delivery
Capital needed	Low (read-only RPCs)	High (funded wallets)
Scoring basis	Detection speed rank	5-dim execution score
Winner selection	Rank-based	Sealed-bid auction

4.1 Scanner Miners (Mechanism 1)

Scanner miners monitor source chain contracts for `MessageDispatched` events and report them to validators. The validator records its own receipt timestamp (miners never submit timing information) and assigns rank-based scores shown in Table 2.

Table 2: Scanner Miner Detection Scoring

Detection Rank	Score
1st (fastest verified response)	1.00
2nd	0.70
3rd	0.50
4th	0.20
Non-response or incorrect data	0.00

The period discovery score is the mean of per-event scores across one Bittensor epoch (100 blocks default). Miners covering more chains score on more events, creating a natural incentive for broad chain coverage. Infrastructure requirements are deliberately lightweight: private low-latency RPC endpoints per monitored chain and a Bittensor hotkey/coldkey (no funded wallets required).

4.2 Relay Miners (Mechanism 2)

Relay miners execute actual message delivery to destination chains. When validators produce a threshold-attested `MessageDispatched` event, they broadcast a `BidRequest` to all registered relay miners. The auction flow is shown in Figure 3.

Relay execution follows a deterministic 8-step flow: dedup check, pre-flight verification, in-flight lock, `PendingMessage` construction, `adapter.relay_message()` execution under a 12-parallel concurrency semaphore, `RelayStatus` mapping, `ProofBundle` assembly, and dedup cache insertion.

4.3 Combined Scanner+Relay Miners

A single Bittensor UID may declare `roles: ["scanner","relay"]`, participating in both mechanisms independently. The strategic advantage is pre-positioning: a combined miner that detects an event via its scanner function can pre-construct its relay bid before the `BidRequest` arrives, gaining a measurable latency advantage in competitive rounds.

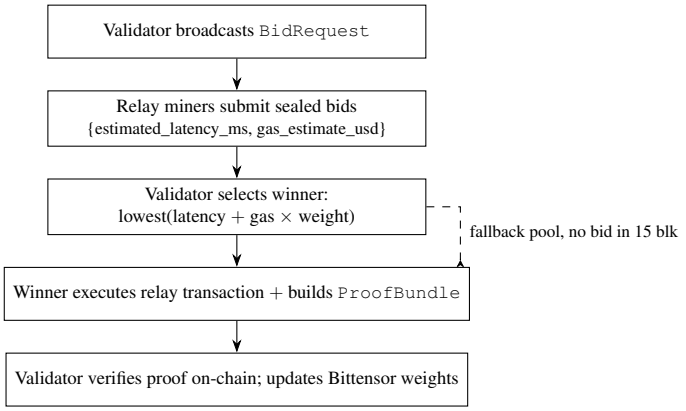

Figure 3: Sealed-bid relay auction flow with fallback relay pool activation.

5. VALIDATOR ARCHITECTURE

Validators are the orchestration layer. They do not scan chains themselves - they delegate to scanner miners and verify results, maintaining security through cryptographic attestation.

5.1 Validator Responsibilities

1. Broadcast `ChainScanSynapse` to scanner miners; record receipt timestamps server-side
2. Spot-verify events (∼20% sampled; all via multi-reporter consensus)
3. Maintain consensus gas oracle (updated at a configurable interval, stake-weighted median)
4. Sign verified messages with per-ecosystem attestation keys
5. Run sealed-bid auctions; select and signal relay miner winners
6. Independently verify relay execution via multi-RPC quorum
7. Call `set_weights()` on the Bittensor metagraph every 100 blocks

5.2 Attestation Key Architecture

Validators maintain separate signing keys per destination ecosystem, loaded into memory at startup and never written to disk (HSM or encrypted secrets manager required). All supported ecosystems use the secp256k1 signature scheme for uniformity and auditability:

Table 3: Validator Attestation Key Schemes by Ecosystem

Env. Variable	Ecosystem	Sig Scheme
`ATTEST_KEY_EVM`	All EVM chains	secp256k1
`ATTEST_KEY_SOLANA`	Solana	secp256k1
`ATTEST_KEY_SUI`	SUI	secp256k1
`ATTEST_KEY_COSMOS`	Cosmos chains	secp256k1
`ATTEST_KEY_STELLAR`	Stellar	secp256k1

5.3 Validator Scoring Loop

Figure 4 illustrates the five-step loop executed each epoch.

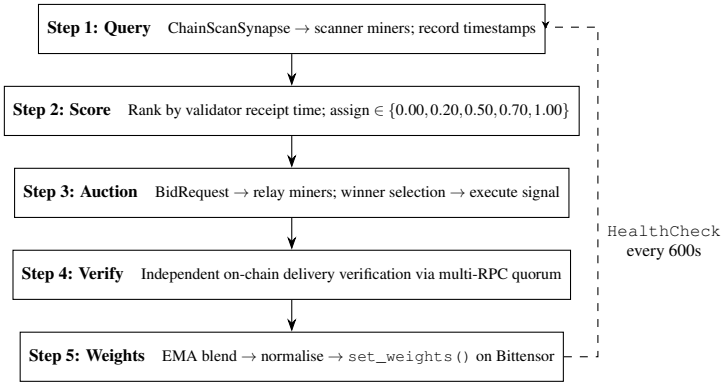

Figure 4: Validator scoring loop per epoch (100 Bittensor blocks) with background liveness probes.

6. SCORING & INCENTIVE MODEL

6.1 Discovery Score (Scanner Miners)

The period discovery score is the mean of per-event scores across all unique `(chain_id, seq_no)` events in the scoring window:

$$S_i^{\text{disc}} = \frac{1}{|E|} \sum_{e \in E} s_i^e, \quad s_i^e \in \{0.00, 0.20, 0.50, 0.70, 1.00\}$$

6.2 Execution Score (Relay Miners)

Relay miners are evaluated across five independent dimensions. This multi-dimensional structure prevents gaming through single-axis optimisation. Per-chain score weights are configurable to reflect each chain's relative gas cost, ensuring that higher-cost chains yield proportionally higher scoring weight and thus fairer miner compensation:

Table 4: Five-Dimensional Relay Execution Scoring

Dimension	Default Wt.	Scoring Logic
SUCCESS	50%	1.0 confirmed; 0.4 already relayed; 0.15 pending; 0.0 failed
SPEED	25%	relay_block_ts − source_block_ts (from on-chain headers)
CORRECTNESS	15%	Multi-RPC quorum; asymmetric fallback for validator infra failure
FEE_EFF.	5%	Derived from on-chain receipt via oracle (never miner-reported)
RELIABILITY	5%	50-task rolling window; 0.50 neutral baseline for new miners

The final blended score per relay round:

$$S^{\text{win}} = 0.80 \cdot S^{\text{exec}} + 0.20 \cdot S^{\text{bid}}$$

$$S^{\text{non-win}} = S^{\text{bid}}, \qquad S^{\text{no-bid}} = 0.00$$

6.3 New Miner Reliability Bootstrap

To prevent impossible first-mover economics, reliability scoring uses a graduated baseline:

Table 5: New Relay Miner Reliability Baseline Schedule

Tasks Completed	Reliability Score Applied
0–5	0.50 (neutral; no penalty, no bonus)
6–15	Rolling average of completed tasks only
16–49	Rolling average, floor of 0.20
50+	Full 50-task rolling average, no floor

6.4 Anti-Gaming Protections

Self-reporting manipulation prevented. Scanner miner speed rankings are computed from the validator's own receipt timestamp. Miners never submit a timing field.

Three-layer correctness verification: (1) Attestation consistency - did the miner return the validator attestation unmodified? (2) Destination chain event query via multi-RPC quorum. (3) Asymmetric fallback: validator infra failure yields 0.5 neutral; miner fabrication yields 0.0.

ALREADY_RELAYED farming prevented. Code 208 partial credit requires raw signed transaction bytes proving genuine attempt; without these: 0.0.

Fee self-reporting eliminated. `fee_paid_usd` is derived from on-chain receipts via Chainlink/Pyth price oracles; miner-reported values are completely ignored.

7. SMART CONTRACT ARCHITECTURE

7.1 One Contract Per Chain

Entangle deploys a single contract per supported blockchain, serving both source and destination roles, minimising audit surface and operational complexity:

- **Source role:** `sendMessage()` collects fees, enforces the fee invariant, emits `MessageDispatched`
- **Destination role:** `verifyMessage()` validates the threshold signature bundle against the registered validator set

7.2 Fee Invariant

Every `sendMessage()` call enforces:

```
require(msg.value >=
  platform_base_fee[dst_chain]
  + gas_reserve[dst_chain])
// gas_reserve = oracle_median
//               * gas_buffer_multiplier
// Default buffer: 1.25x (25% safety margin)
// Buffer is configurable via governance
```

7.3 Cross-Chain Replay Protection

A 4-tuple deduplication key is enforced uniformly across all five smart contract ecosystems, preventing a message intended for one destination from executing on another:

Table 6: Cross-Chain 4-Tuple Deduplication Keys

Ecosystem	Dedup Key Construction
EVM	`keccak256(src_chain, DST_CHAIN, seq_no)`
Solana	PDA: `["executed", src_hash[:8], dst_hash[:8], seq_le]`
SUI Move	`ExecutionKey{src_chain, dst_chain, seq_no: u64}`
Cosmos	Map key: `(src_chain, dst_chain, seq_no)`
Stellar	`DataKey::Executed(src_hash, dst_hash, seq_no)`

Table 7: Core Smart Contract Function Reference

Function	Description & Access Control
`sendMessage()`	Accepts fee, emits `MessageDispatched`; any dApp
`verifyMessage()`	Validates threshold sig bundle; destination dApp
`updateGasOracle()`	Submits gas estimate; registered validator only
`updateValidatorSet()`	Replaces validator keys/weights; multisig + 48h timelock
`setSignatureThreshold()`	Sets minimum sig threshold; multisig + 48h timelock
`freezeOracle()`	Emergency manual gas freeze; protocol multisig only
`getTotalFee()`	Returns current total fee for dApp UIs; anyone

7.4 Core Contract Functions

7.5 Protocol Data Structures

The **MessageEnvelope** is the chain-agnostic canonical description of a pending cross-chain message. Key fields include: `message_id` (sha256 of src context, globally unique), `src_chain`, `src_ecosystem`, `src_tx`, `src_seq`, `dst_chain`, `dst_recipient`, `payload_hex` (max 8 KiB), and `requires_ack`.

The **ProofBundle** returned by relay miners includes: `proof_type` (one of `evm_tx_receipt`, `solana_signature`, `sui_digest`, `stellar_hash`, `cosmos_tx_hash`), `relay_tx`, `dst_height`, `dst_timestamp`, and raw ecosystem-specific data.

8. SECURITY MODEL

8.1 Threshold Attestation

All message delivery is gated by a threshold signature bundle. Before any destination-chain state change, `executeMessage()` verifies that validators controlling a configurable percentage of registered stake (default >60%) have each independently signed the message. This threshold is adjustable via governance to allow the network to tune its security-liveness tradeoff as the validator set evolves:

```
for (uint i = 0; i < bundle.validator_uids.length; i++) {
  ValidatorRecord memory v =
      validators[bundle.validator_uids[i]];
  require(v.active, 'InactiveValidator');
  require(bundle.payload_hashes[i] == payload_hash,
          'PayloadMismatch');
  bytes32 digest = _attestDigest(
      src_chain, seq_no, DST_CHAIN_NAME,
      payload_hash, dst_recipient);
  require(_recoverSigner(digest,
      bundle.signatures[i]) == v.signingKey,
          'InvalidSig');
  attestedStake += v.stakeWeight;
}
uint128 required =
    totalRegisteredStake * THRESHOLD_PERCENT / 100;
// THRESHOLD_PERCENT is governance-configurable
require(attestedStake >= required,
        'BelowStakeThreshold');
```

Key security properties:

- **Payload integrity:** contract hashes submitted payload; validator signatures commit to that hash-any miner substitution invalidates all signatures
- **Recipient integrity:** `dst_recipient` is explicitly in the attestation digest; substituting it invalidates every validator signature simultaneously
- **Pre-execution enforcement:** stake threshold check is the first gating operation; nothing executes until it passes
- **No new trust set:** Bittensor validators are already economically staked; colluding validators face TAO slashing and reputational loss

8.2 Source Chain Reorg Protection

Validators advance their block cursor only to a computed "safe height", respecting per-chain finality depths before attesting:

```
safe_height = current_height
            - FINALITY_BLOCKS[chain]
# ETH: 12 blocks; EVM L2: 1; Polygon/BNB: 64
# Solana: 1 (confirmed); SUI: 1; Cosmos: 1
msgs = await adapter.get_pending_messages(
    from_height=cursors[chain],
    to_height=safe_height)
cursors[chain] = safe_height
```

8.3 Cross-Validator Collusion Detection

Validators gossip score vectors via `ScoreGossip` synapse after each epoch. Because most scores are deterministic from on-chain data, outliers are provably wrong:

```
deviation = mean_absolute_error(
    vec, stake_weighted_median(scores))
if deviation > 0.15:  # >15% deviation
    self._flag_validator(vuid, deviation)
    # flagged validators' weight contributions
    # are discounted in final aggregation
```

8.4 Oracle Collusion Circuit Breaker

The gas oracle contract rejects any update deviating more than a configurable percentage (default 30%) from the prior epoch's accepted value unless countersigned by the protocol multisig, preventing coordinated oracle suppression attacks that would systematically underfund relay miners.

9. COMPETITIVE DIFFERENTIATION

Table 8 provides a direct comparison against the three most significant incumbents across six key dimensions.

Entangle's primary structural advantage is converting relay quality into a **measurable, objective performance metric** that Bittensor's consensus mechanism rewards directly. No other protocol uses a machine intelligence incentive layer to continuously improve relay quality through competition.

Table 8: Cross-Chain Protocol Comparison Matrix

Dimension	Entangle	Wormhole	LayerZero	Axelar
Miner entry	Permissionless TAO burn	Permissioned Guardians	Permissioned DVNs	Perm. validators
Incentive	Competitive TAO + fees	Protocol revenue share	DVN fees	Staking rewards
Architecture	Open subnet	19-Guardian multisig	DVN operator set	PoS set
Latency	<60s / <90s L1	1–5 min	<1 min	2–5 min
Fee model	Oracle-driven per-dst	Fixed/variable	Pay-as-you-go	Fixed
Chain ext.	ChainAdapter interface	Per-chain deploys	DVN per chain	Per-chain voting

10. SUPPORTED CHAINS & ECOSYSTEM ADAPTERS

10.1 Launch Chain Coverage

At launch, Entangle supports 13 blockchain networks across five ecosystems. The protocol is designed for continuous expansion; same-ecosystem chains can be onboarded with zero code changes, and new ecosystems require only a new `ChainAdapter` implementation.

Table 9: Initial 13 Supported Chains at Launch

Chain	Ecosystem	Finality Model	Sig
Ethereum	EVM	PoS probabilistic (2 blocks)	secp256k1
Arbitrum One	EVM L2	Sequencer + L1 (1 block)	secp256k1
Base	EVM L2	Sequencer + L1 (1 block)	secp256k1
Optimism	EVM L2	Sequencer + L1 (1 block)	secp256k1
Polygon	EVM PoA	Checkpoint (64 blocks)	secp256k1
BNB Chain	EVM PoA	Checkpoint (64 blocks)	secp256k1
Avalanche	EVM Snow	Instant after acceptance (1)	secp256k1
Solana	Solana	Tower BFT (32 slots, ∼13s)	secp256k1
SUI	SUI	Narwhal/Bullshark (1 checkpoint)	secp256k1
Cosmos Hub	Cosmos	Tendermint BFT (1 block)	secp256k1
Osmosis	Cosmos	Tendermint BFT (1 block)	secp256k1
Neutron	Cosmos	Tendermint BFT (1 block)	secp256k1
Stellar	Stellar	SCP federated (1 ledger)	secp256k1

10.2 ChainAdapter Interface

All chain-specific logic is encapsulated in a single Python abstract class. New ecosystems require only this interface to be implemented-a deliberate architectural choice that reduces the cost of expansion to days rather than months:

```
class ChainAdapter(ABC):
  def get_pending_messages(
      self, from_height, to_height
  ) -> list[PendingMessage]: ...
  def is_message_delivered(
      self, seq_no, src_chain) -> bool: ...
  def relay_message(
      self, msg, private_key,
      max_fee_usd) -> RelayResult: ...
  def verify_relay(
      self, relay_tx, expected_seq
  ) -> tuple[bool, dict]: ...
  def get_current_height(self) -> int: ...
```

Existing implementations: **EVMAdapter** (7 chains via web3.py, PoA middleware auto-injected); **SolanaAdapter** (solana-py, Anchor IDL event parsing); **SUIAdapter** (SUI JSON-RPC, programmable transaction blocks); **StellarAdapter** (Horizon REST + Soroban RPC split); **CosmosAdapter** (Stargate REST, CosmWasm, per-chain denom handling).

11. FEE MODEL & GAS ORACLE

11.1 Fee Architecture

Every `sendMessage()` call directs the full fee (100%) to the **protocol reserve**. Rather than splitting fees at the point of execution, the system aggregates all fees into a unified reserve, which is deployed through **subnet alpha buybacks**.

The proceeds from these buybacks are distributed across network participants - **miners, validators, and protocol builders** according to a fixed ratio enforced by **Yuma Consensus**.

11.2 Gas Oracle Mechanics

The gas oracle adjusts `gas_reserve[dst_chain]` dynamically at a configurable interval (default ∼2 minutes): (1) each validator samples current gas price on the destination chain via RPC, (2) estimates gas cost of a typical `receiveEntangleMessage()` call, (3) submits `updateGasOracle(dst_chain, estimate)` on-chain, (4) contract computes the **stake-weighted median** of recent submissions, (5) updates `gas_reserve` = median × configurable safety buffer (default 1.25×).

The configurable deviation circuit breaker (default 30%) prevents colluding validators from suppressing the gas reserve below operationally sound levels.

11.3 Fee-to-Quality Flywheel

The fee model creates a self-reinforcing quality loop: higher relay volume → more fees collected → relay reward attracts better miners → lower latency → more dApp integrations → higher relay volume.

12. TOKENOMICS & ECONOMICS

12.1 The Alpha Token

Alpha is Entangle's subnet token on Bittensor-a **work token** (utility) with two demand drivers: (1) TAO emissions converting via the subnet's internal AMM in bootstrap mode, and (2) programmatic fee-backed buybacks every 7 days via TWAP.

12.2 Alpha Buyback Mechanism

Figure 5 shows the per-message fee flow. Monthly Alpha buyback formula:

$$B_{\text{USD,month}} = M_{\text{msgs}} \times F_{\text{native}} \times P_{\text{native/USD}}$$

where:

- $B_{\text{USD,month}}$ = total monthly buyback volume (in USD),
- D_{msgs} = average monthly message volume,
- F_{native} = fee per message denominated in the **native token** (e.g., ETH),
- $P_{\text{native/USD}}$ = price of the native token in USD,

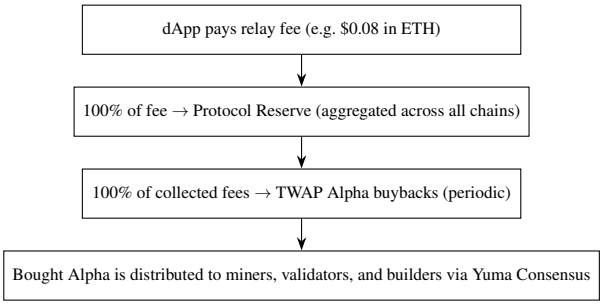

Figure 5: Protocol-level fee aggregation and buyback flow.

Year 1 baseline (50K msgs/day, \$0.08/msg): $B = \$120{,}000$/month.
Year 3 scale (3M msgs/day, \$0.05/msg): $B = \$4{,}500{,}000$/month.

12.3 Supply Dynamics & Deflationary Inflection

Table 10: Alpha Token Supply Dynamics Over Time

Daily Msgs	Mo. Emission	Mo. Buyback	Net Supply
5,000	216,000α	10,667α	+205K (inflationary)
30,000	216,000α	64,000α	+152K (moderate)
50,000	216,000α	106,667α	+109K (controlled)
500,000	216,000α	1,066,667α	**−851K (deflationary)**
3,000,000	216,000α	6,400,000α	−6.18M (strongly deflation.)

The **deflationary inflection point** occurs at ∼90,900 messages/-day, when buyback alone exceeds new Alpha emission. Combined with the emission burn rate, the protocol becomes structurally deflationary before Year 2 at target growth.

12.4 The Dual Flywheel

Most Bittensor subnets are structurally dependent on TAO emissions to sustain miner and validator incentives. As a result, their security and participation levels are tightly coupled to the market price of TAO. In bear markets, declining token value directly reduces effective rewards, leading to degradation in network performance.

Entangle introduces a second, orthogonal demand driver by sourcing value from **external execution environments** (e.g., Ethereum, BNB Chain, Polygon). Instead of relying solely on endogenous inflation, the protocol captures **real economic activity** in the form of message fees paid in native gas tokens (ETH, BNB, MATIC, etc.).

These fees are:

1. Collected across all supported chains,
2. Aggregated into a unified protocol reserve,
3. Converted via TWAP execution into Alpha through open-market buybacks.

This creates a **cross-domain value bridge**, where economic activity originating outside Bittensor is continuously funneled back into the subnet economy.

As a result, the system operates two reinforcing flywheels:

- **Emission Flywheel (Endogenous):** TAO emissions bootstrap and sustain early-stage participation.
- **Revenue Flywheel (Exogenous):** Cross-chain usage generates fee revenue, which is converted into persistent buy pressure on Alpha.

13. GOVERNANCE

13.1 Protocol Multisig

All Entangle governance uses a **3-of-5 Gnosis Safe** (EVM) or equivalent multisig per ecosystem: Squads multisig (Solana); package upgrade cap (SUI); CosmWasm admin (Cosmos); Stellar multisig admin. Signer sets are kept synchronised across all chains within 24 hours of any rotation. All signer addresses and transactions are publicly on-chain verifiable.

13.2 Timelock Schedule

Table 11: Governance Operations & Timelock Schedule

Operation	Timelock	Trigger
Update Chain Config Registry	None	New chain / rotation
`setBaseFee(), setGasBuffer()`	None	Operational
`freezeOracle()`	None	Emergency
Change emission split	48 hours	Governance proposal
`updateValidatorSet()`	48 hours	Validator rotation
`setSignatureThreshold()`	48 hours	Security parameter
Emergency key rotation	None	Key compromise
Freeze subnet registrations	None	Emergency only

13.3 Major Protocol Upgrades

New ecosystem support and architectural changes follow a structured five-step process: (1) public RFC posted to project GitHub; (2) minimum 7-day community comment period; (3) multisig review and response to substantive comments; (4) 48-hour timelock on execution where applicable; (5) public announcement of execution transaction hash.

13.4 Bootstrapping Phase

Phase 1 (Month 1–3): A treasury-controlled test dApp emits 10–50 synthetic `MessageDispatched` events per day per chain. Scanner miners that correctly report receive a flat 0.50 participation score-no speed-race dynamic during the bootstrapping window.

Phase 2 (Live scoring): Triggered automatically when real events reach ≥20/day per chain. Full rank-based scoring resumes.

Emergency Fallback: If no scanner miner responds for 3 consecutive cycles (∼90 seconds), validators activate a thin RPC poller built into the validator binary. Events are flagged as validator-originated; no scanner miner receives a score. This exists solely to prevent relay pipeline stalling.

14. DAPP INTEGRATION

Entangle's integration model places application logic firmly in the dApp contract. The receiving dApp implements `receiveEntangleMessage()`:

```
function receiveEntangleMessage(
    bytes calldata payload,
    uint64 seq_no,
    string calldata src_chain,
    bytes calldata src_addr,
    bytes calldata sig_bundle
) external {
  require(!executed[seq_no], 'ALREADY_EXECUTED');
  bytes32 msg_hash = keccak256(abi.encode(
      src_chain, block.chainid, seq_no,
      src_addr, address(this), payload));
  require(
    entangle.verifyMessage(msg_hash, sig_bundle),
    'BAD_SIGS');
  executed[seq_no] = true;
  _handleMessage(payload);
}
```

Table 12: dApp Integration Summary by Ecosystem

Ecosystem	Relay Miner Calls	Sig Scheme
EVM chains	`dApp.receiveEntangleMessage()`	secp256k1
Solana	`receiveEntangleMessage` instr.	secp256k1
SUI	Move function receive	secp256k1
Cosmos	CosmWasm receive function	secp256k1
Stellar	Soroban contract function	secp256k1

New same-ecosystem chains (e.g., a new EVM L2) require only contract deployment and a Chain Configuration Registry entry-no code changes. New ecosystem chains require a new `ChainAdapter` implementation and a governance vote.

15. RISK FACTORS & MITIGATIONS

Table 13: Risk Register with Impact Ratings and Mitigations

Risk	Impact	Mitigation
All scanner miners offline	Critical: relay stalls	Validator fallback at ∼90s; zero-score economics disfavour downtime
Scanner gaming speed rank	Medium: unfair emissions	Validator-side timestamps only; miners cannot self-report
Oracle collusion	High: miners underfunded	Configurable deviation circuit breaker; multisig freeze; public audit trail
Gas price spike	Medium: temp. underfunding	Configurable safety buffer; configurable oracle interval; manual multisig override
Fabricated event reports	Medium: false pipeline	20% spot-verification; fabrication scores 0.00; 3 flags → 24h suspend
Attestation key compromise	High: arbitrary sig forgery	Keys in HSM; single key cannot meet threshold; emergency rotation
Smart contract vuln.	High: reserve loss	Formal audit; threshold sig limits blast radius; $500K bug bounty; auto-pause
TAO price collapse	High: scanner income collapse	50% max burn; Insurance Fund; relay miners paid in native gas tokens
Relay miner cartel	Medium: price-gouging	Max bid cap; fallback relay pool; slashing for bid collusion

16. CONCLUSION

Entangle Protocol demonstrates that the fragmentation problem in Web3 infrastructure does not require a permissioned validator set or a centralised coordination layer to solve. It requires the right incentive structure.

By instantiating cross-chain message relay as a competitive digital commodity on Bittensor's incentive infrastructure, Entangle delivers **fully permissionless entry**, **objective performance-based rewards**, **threshold-attested security without new trust assumptions**, and **economically resilient tokenomics** that function independently of TAO market conditions. All critical protocol parameters-including the attestation threshold, emission split, gas oracle update interval, safety buffer, and per-chain score weights-are governance-configurable, allowing the network to adapt as it grows.

The dual-miner architecture-scored across objectively verifiable on-chain dimensions, protected at every layer against gaming and manipulation, governed by a publicly-auditable multisig with appropriate timelocks-represents a materially different approach to decentralised infrastructure than anything deployed in production today.

Every additional miner that joins the Entangle network makes it more secure, cheaper, and faster. That is the correct design property for infrastructure that aspires to be the universal messaging layer for Web3.

17. ROADMAP

Table 14: Entangle Protocol Development Roadmap

Phase	Milestones
Phase 0 *Q2 2025*	Integrate 5 ecosystems across 12 chains; complete internal audits and security patches; deploy core contracts (testnet); implement validator & miner logic; launch protocol dashboard; establish messaging pipeline (target ∼50 msgs/day on testnet)
Phase 1 *Q2 2026*	Mainnet contract deployment; public audit: migrate subnet to mainnet: open miner/validator participation; initial dApp integrations and cross-chain activity
Phase 2 *Q3–Q4 2026*	Introduce governance logic; multi-signature voting system; DAO structure formation; open-source contribution framework; establish Entangle Foundation; expand ecosystem participation
Phase 3 *Q1–Q2 2027*	Expand to 28 chains; enterprise B2B SDK (exchanges, wallets); scale to 500K msgs/day; introduce relay miner insurance pool; optimize routing (cheapest-path execution); ∼$900K/month revenue
Phase 4 *2027+*	60+ chains; 3M+ msgs/day; fully decentralized governance; protocol-owned Alpha liquidity; zk-proof-based scanner verification for trustless operation; ∼$4.5M+/month revenue

GLOSSARY

Term	Definition
Alpha	Entangle's subnet token on Bittensor; work/utility token
Bittensor	Decentralised AI network providing incentive and consensus layer
ChainAdapter	Python class implementing chain-specific logic for one ecosystem
ChainScanSynapse	Synapse type sent by validators to request event detection
Discovery Mechanism	Mechanism 1-configurable emission share; runs scanner miner scoring
HealthCheckSynapse	Liveness probe sent by validators every $\sim$10 min to all miners
Metagraph	Bittensor's on-chain registry of subnet participants and weights
MessageDispatched	On-chain event emitted by Entangle when a dApp sends a message
MessageEnvelope	Chain-agnostic canonical description of a pending cross-chain message
ProofBundle	Cryptographic evidence of delivery returned by a relay miner
Relay Mechanism	Mechanism 2-configurable emission share; runs relay miner scoring
RelayTask	Primary synapse carrying message context from validator to miner
Scanner Miner	Miner monitoring source chains for events (Mechanism 1)
Relay Miner	Miner executing message delivery to destination chains (Mechanism 2)
Sealed-Bid Auction	Validator-run process for selecting which relay miner executes
TAO	Bittensor's native token; emitted proportional to performance
Threshold Attestation	Configurable validator stake consensus required before message execution
Yuma Consensus	Bittensor mechanism aggregating validator weights to distribute TAO