🌳 The Garden of Eden (Eden)

Whitepaper v1.0 – Indexer-First, Intelligence-Native Marketplace

Author: Bill Draper (CTO)
Date: 2026

Abstract

Eden is an **indexer-first economic and intelligence system** where the traditional blockchain is no longer the parent, but the *child* of the indexer. Eden replaces gas fees, smart-contract rigidity, and token-centric governance with **LLM-governed intelligence fees**, **federated indexers**, and a **ROOT Certificate Authority (ROOT CA)** that certifies trust, services, and replication.

Eden is designed to be: - Gas-free - Indexer-driven - LLM-native - Service-oriented - Self-policing, self-governing, and self-replicating

---

1. Core Philosophy

> *Blockchain records history. Eden understands it.*

Eden assumes: - History is immutable - Meaning is contextual - Trust is certified, not mined - Intelligence is the new gas

Indexers are **priests**, not miners. ROOT CA is **law**, not power. Users are **free actors**, not wallet addresses.

---

2. System Actors

2.1 ROOT CA (Law / Moses)

- Global certification authority
- Certifies indexers and services
- Collects minimal intelligence fee ($\approx$0.001%)
- Guarantees fallback, insurance, and dispute resolution

2.2 Indexers (Knowledge Trees)

- Federated, Docker-deployed nodes
- In-memory Redis-like databases
- Run identical LLM versions (DeepSeek-class)
- Hold certificates + private keys

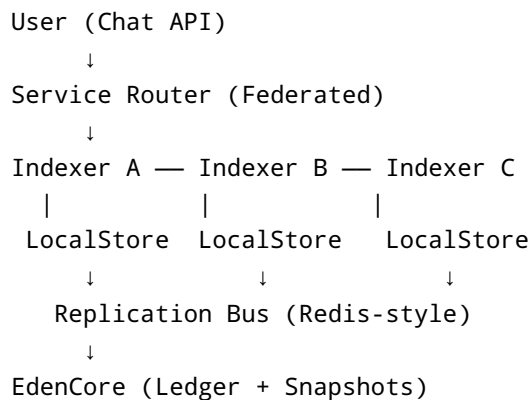• Provide intelligence, routing, pricing, and policing

2.3 Users (Humans)

• Google-certified identity (email only)
• No wallets required
• Multiple identities allowed
• Pay via intelligence usage (iGas)

2.4 Service Providers (Apples on Trees)

• Movies, DEX pools, content, goods, APIs, marketplaces
• Register via ServiceRegistry
• Bonded and reputation-scored
• Can be legacy platforms (AMC, MovieCom, etc.)

---

3. Indexer-First Architecture

```
User (Chat API)
     ↓
Service Router (Federated)
       ↓
Indexer A ── Indexer B ── Indexer C
   |           |            |
 LocalStore  LocalStore   LocalStore
     ↓           ↓            ↓
   Replication Bus (Redis-style)
      ↓
EdenCore (Ledger + Snapshots)
```

Key rule:

If an indexer can answer it, the chain does not need to exist.

---

4. Event-Driven Replication Bus

Eden uses a **database-level replication model** instead of consensus mining.

Event Types

• `SNAPSHOT` – transaction / action
• `SERVICE_UPDATE` – price, availability
• `BOND_UPDATE` – trust & insurance changes

Properties

- Deterministic
- Replayable
- Stateless consensus
- Redis Streams compatible

Indexers replicate state, not blocks.

---

5. Intelligence Gas (iGas)

- No blockchain gas
- No native token
- iGas = LLM + routing + reasoning cost

iGas Redistribution

Recipient	Share
ROOT CA	Governance & insurance
Indexers	Compute & routing
Service Providers	Quality incentive
Users	Usage credit

This creates a **positive-sum economy**.

---

6. Service Registry & Routing

- All services must be registered
- Each service includes:
- Price
- Location
- Bond
- Reputation

Example User Query

"I have 10 USDC. Where can I watch *Catch Me If You Can* tonight at best price?"

Flow: 1. Query broadcast to indexers 2. Indexers evaluate local + federated services 3. LLM aggregates best result 4. EdenCore executes transaction

---

7. Dynamic Bonds & Pricing

• Every successful transaction:
• Increases service bond
• Improves reputation

• Can lower fees

• Poor service:

• Bond reduced
• Routing deprioritized

This replaces ratings with **economic truth**.

---

8. No-Rug DEX Model (Optional Layer)

• Pools must be ROOT-certified
• Creator cannot rug without losing bond
• Trades increase pool value slightly
• High-frequency traders rewarded, not penalized

Eden supports DEX/CEX without native tokens.

---

9. SaaS & Legacy Integration

Eden integrates via **API plugins**: - AMC - MovieCom - Banks - Wallets - Payment processors

Legacy systems keep control; Eden handles intelligence, trust, and settlement.

---

10. Security & Identity

• Google identity only
• Email-based trust
• Abuse detection via LLM policing
• No private key management for users

---

11. Deployment Model

• Docker-only
• Low hardware requirements

- Horizontal scaling
- Stateless indexers

Each indexer = 1 Knowledge Tree

---

12. Eden-Sim (Reference Implementation)

- TypeScript
- <1500 LOC
- No Web3 dependencies
- Mock LLM
- Redis-style replication

Purpose: economic + architectural validation

---

13. Why Eden Wins

Problem	Eden Solution
Gas fees	Intelligence fees
Smart contracts	Dynamic reasoning
Rug pulls	Bonded services
MEV	Certified transparency
Fragmentation	Federated indexers

---

14. Genesis Statement

Eden is not a protocol.
It is a living system.

ROOT CA gives law.
Indexers give wisdom.
Humans give meaning.

The Garden grows.

End of Whitepaper.