Swarm Decentralized Compute Network: Public Architecture Overview

Version: 1.0 (Public Litepaper Edition)
Status: Public Reference Document

1. System Overview

The Cambrian-Swarm is a universal, distributed peer-to-peer infrastructure protocol designed for elastic computation. It functions as a Hybrid-Cloud Orchestrator, allowing enterprises to manage their internal hardware at maximum efficiency while seamlessly "bursting" heavily encrypted workloads to a global, permissionless spot market.

To manage the severe compute, memory, and bandwidth constraints historically associated with Fully Homomorphic Encryption (FHE), the architecture utilizes a proprietary **Decoupled Pre-Processing Paradigm**. The system strictly separates workload orchestration from mathematical execution, enabling highly parallelized computation across heterogeneous global hardware without sacrificing zero-trust security.

2. Network Topology and Node Roles

2.1. The Client Edge Gateway (The Secure Perimeter)

To interact with the network, the protocol utilizes a Client Edge Gateway deployed strictly within the user's secure perimeter (e.g., a corporate firewalled VPC).

- **Secure Pre-Processing:** The Gateway handles necessary data preparation and relational structuring natively within the client's secure perimeter prior to encryption, optimizing the data for distributed execution.
- **Automated Ciphertext Sharding:** To prevent resource exhaustion on global nodes, the Gateway automatically slices massive datasets into optimized ciphertext micro-batches, ensuring maximum mathematical efficiency and parallelization.
- **Zero-Trust Translation:** The Gateway transforms standard data science scripts into secure, static mathematical representations. Private Decryption Keys remain permanently locked inside the Gateway and are never submitted to the network.
- **Unified Output Reconstruction:** Upon workload completion, the Gateway securely fetches the encrypted output fragments, decrypts them locally, and stitches them back into a unified dataset for the end-user.

2.2. Stateless Relayers (The P2P Mempool)

The off-chain P2P Mempool is maintained by stateless, permissionless routing nodes. These relays index and gossip lightweight "Job Tickets" based on dynamic hardware-tier requirements without holding any centralized authority over assignment, financial escrows, or active hardware profiling.

2.3. The State Ledger (Coordination & Verification)

The protocol utilizes an immutable **State Ledger** (an L2 Smart Contract/AppChain) as the ultimate arbiter of truth, handling trustless routing, financial escrows, and consensus verification.

- **Bifurcated Routing Logic:** The Ledger supports two strictly segregated routing modalities to optimize network efficiency:
    1. **The Public Spot Market:** For standard burst workloads, the network utilizes a proprietary, highly gas-efficient decentralized assignment protocol. This guarantees Sybil-resistance and cryptographic randomness without suffering from the latency and gas bloat of traditional L1 on-chain matchmaking.
    2. **Dedicated Fleet Contracts:** For continuous enterprise workloads, Clients establish direct, on-chain lease contracts with specific Worker nodes. This securely optimizes cryptographic key caching and drastically reduces network bandwidth overhead.

2.4. Ephemeral Data Availability (DA) Layer

Due to ciphertext expansion, payload data cannot be stored on immutable blockchains. Encrypted micro-batches, cryptographic parameters, and compute graphs are hosted on a decoupled, **Ephemeral DA layer**.

- **State Optimization:** The DA layer implements automated, decentralized lifecycle management, ensuring intermediate execution states are securely purged once network consensus is finalized.

2.5. Universal Swarm Daemons (Workers)

Workers are background services running on anything from enterprise internal servers to permissionless public data centers.

- **The Staking Bifurcation:**
    - *Public Spot Market:* To claim permissionless bounties, Workers lock a native token stake in the State Ledger. This prevents Sybil attacks and provides collateral that can be slashed for malicious execution.
    - *Enterprise Fleets:* Enterprise nodes operating strictly on their organization's internal, private workloads are fully exempt from public Web3 staking requirements, ensuring zero-friction corporate adoption.

3. Trust-Aware Pluggable Execution Runtimes

The Universal Swarm Daemon utilizes context-aware execution routing, dynamically selecting its runtime sandbox based on the cryptographic signature and origin of the Job Ticket.

3.1. `OPEN` Compute Mode (Cleartext)

- **Public `OPEN` (Hostile Multi-Tenancy):** Unencrypted jobs signed by unknown global users default to maximum security. The payload is executed strictly inside a deterministic, highly isolated virtual sandbox. This protects host Node Operators from malicious code while enforcing strict execution determinism across diverse operating systems and CPU architectures.
- **Owned `OPEN` (The Trusted Fleet Bypass):** If an enterprise configures the Daemon on its own hardware with internal cryptographic keys, the Daemon recognizes its owner's internal workloads. It safely bypasses standard virtualization overhead, executing directly on the bare-metal host OS to achieve maximum hardware utilization and direct accelerator access.

3.2. `SEALED` Compute Mode (FHE)

- **Isolated Mathematical Execution:** FHE jobs are not distributed as executable machine binaries. They are transmitted as static mathematical representations. The Worker's native engine executes this strictly as read-only data, mathematically preventing OS compromise or Remote Code Execution (RCE) vectors.
- **Pre-Flight Verification & Bounding:** Before accepting a job, the Worker software performs a secure static analysis of the payload to verify resource requirements. During execution, operations are strictly bounded by OS-level constraints and monitoring systems to prevent intentional resource-exhaustion attacks.

---

4. Execution Guardrails, Integrity, and Determinism

4.1. Configurable Security Boundaries

By default, the Gateway SDK enforces a strict zero-trust boundary. However, advanced users may extend their Trusted Network Zone to include remote infrastructure for high-speed local processing. The protocol manages this via explicit permission architectures and auditable data lineage logs, ensuring clear data-liability boundaries.

4.2. Cross-Hardware Normalization and Consensus

Practical FHE libraries rely on hardware-accelerated math that natively produces microscopic computational variations across heterogeneous CPU architectures (e.g., Intel vs. AMD vs. ARM). Standard cryptographic consensus mechanisms fail under these conditions.

To achieve Byzantine Fault Tolerant consensus across independent Workers, the network enforces **Native Execution Determinism**:

- **Hardware Normalization:** The daemon utilizes proprietary compilation parameters and deterministic state constraints to ensure identical mathematical pathways on any machine, safely allowing multi-core execution without byte-layout divergence.
- **Multi-Stage Cryptographic Consensus:** The State Ledger utilizes a highly secure, multi-stage cryptographic commit-and-reveal protocol. This mathematically eliminates "lazy evaluation" and brute-force hash-guessing attacks.
- **Asymmetric State Finalization:** To prevent redundant network strain, the consensus layer dynamically designates workload finalization paths, cutting global bandwidth consumption by over 60% compared to standard P2P verification models.

---

5. Applied Workloads and Operational Archetypes

By shifting complex data preparation into the Client's perimeter, the Swarm acts as an infinitely scalable compute engine, transforming from a simple privacy tool into an Enterprise Orchestrator.

Archetype 1: Confidential Cloud Bursting (Hybrid Orchestration)

- **The Problem:** Enterprises over-provision expensive internal hardware to handle occasional peak computational workloads.
- **The Swarm Solution:** An enterprise utilizes the Swarm Daemon across its internal servers for free, bare-metal cleartext execution. When a workload exceeds internal capacity, the Gateway "bursts" to the Swarm. It dynamically encrypts the overflow data and broadcasts `SEALED` tickets to the public spot market.
- **Result:** Infinite, elastic scaling onto untrusted global hardware with zero legal exposure or plaintext data leakage.

Archetype 2: Massively Parallel AI Inference

- **The Problem:** Processing massive encrypted datasets simultaneously on a single FHE node is computationally unviable.
- **The Swarm Solution:** The Client's Edge Gateway structures the data internally, generating thousands of optimized micro-batches. The Gateway broadcasts these independent tasks to the global Swarm.
- **Result:** Thousands of permissionless public nodes execute the FHE inference in parallel. Massive datasets are processed in a fraction of the

traditional timeframe, permanently decoupling dataset scale from individual node hardware constraints.

Archetype 3: The Dedicated Cryptographic Fleet (Reserved Instances)

- **The Problem:** Repeatedly transferring massive FHE cryptographic parameters across a global network causes severe latency.
- **The Swarm Solution:** Enterprises issue long-term computational leases on the State Ledger to specific high-performance nodes. These nodes securely cache the Client's specific cryptographic parameters using optimized local storage management.
- **Result:** By servicing only their assigned Client, these nodes operate with near-zero network latency, converting standard cloud servers into hyper-efficient, dedicated enterprise appliances.

“‘