Camp: Protocol Deep Dive

Transforming IP Ownership for the AI-Native Era

The Camp Specification

Camp Foundation • Abundance • Gelato

Contents

Abstract

Intellectual property anchors more than $80 trillion in global intangible value and sits at the center of how digital systems create and exchange content. From every meme shared on WhatsApp to every beat streamed on SoundCloud, from AI models training on creative works to virtual worlds built on user-generated content, IP is essential to the promise of AI. Yet despite blockchain's promise of programmable ownership, onchain IP remains trapped in a primitive state: NFTs that point to static JPEGs, royalties that can be circumvented, and no composable protocol layer for the derivatives, remixes, and AI training that define how IP actually creates value in the digital age.

The absence of a sovereign IP protocol has left creators dependent on centralized platforms that extract 30-50% of value while providing zero guarantees of permanence, provenance, or programmable rights. Camp Network changes this equation entirely by establishing the first L1 purpose-built for intellectual property operations, where every creation, derivation, and monetization event becomes an immutable, composable primitive that can be built upon in many types of applications.

1 Introduction

Intellectual property mostly lives online, but it's still governed by antiquated systems built for the offline world. Ownership, attribution, and licensing all rely on centralized entities, with value ultimately accruing to those intermediaries rather than to rights holders or licensees. IP transactions remain limited and siloed from everyday users.

AI has made that gap even more obvious. Ideas can be created, copied, and distributed instantly, but original creators rarely get credit or paid. On the flip side, AI companies can't realistically manage millions of individual licenses or royalty splits. Existing IP licensing frameworks aren't dynamic enough to keep up with the speed at which content can be copied or remixed.

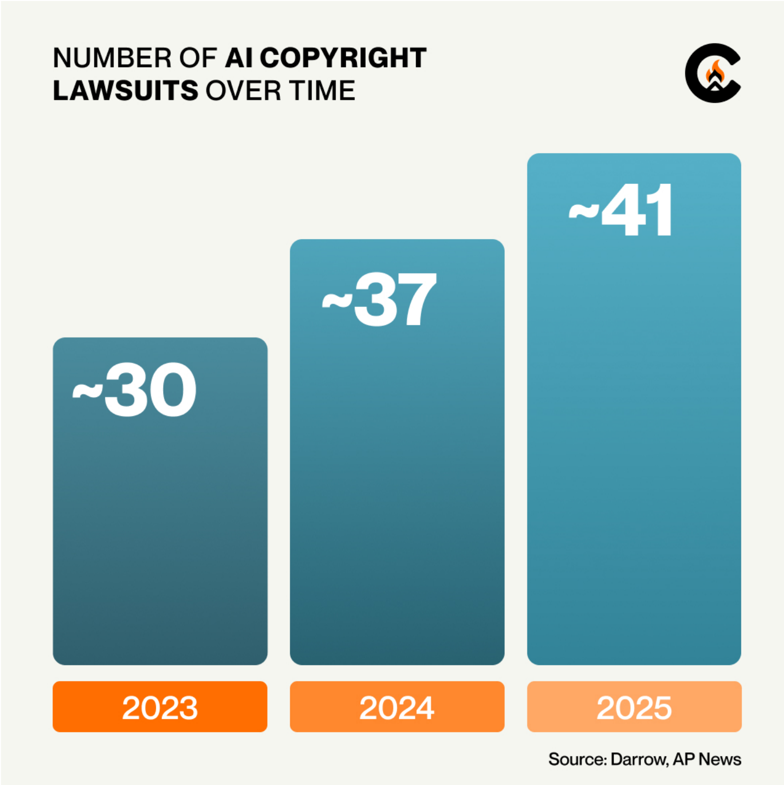

Figure 1: Rising trend of AI copyright litigation

As a result, we have a digital economy where IP moves freely but value doesn't. The core building block, **ownership**, still lacks both liquidity and legibility.

Today, going from co-creation to licensing to revenue can take six months and an enormous amount of manual work. Meanwhile, uploading an image to a platform like Instagram takes minutes, and the platform earns instantly through a royalty-free license from the creator.

Camp Network addresses this challenge by introducing a blockchain protocol purpose-built for programmable intellectual property and autonomous agents. Traditional blockchain architectures lack the specialized infrastructure and throughput necessary to support the scale of IP registration, licensing, and agent-driven transactions that the emerging digital economy demands. Camp delivers both the performance characteristics for gigagas-level throughput, sub-second finality, and dedicated IP blockspace; and the specialized primitives required to support massive-scale IP management and agentic interactions that will define the next generation of on-chain activity.

2 Architecture

Camp is purpose built for IP and agentic coordination. As a sovereign rollup leveraging the Abundance Stack architecture from Gelato, Camp achieves the raw performance needed for IP workloads: gigagas throughput which provides upwards of 100x performance boost compared to traditional rollups without the overhead of traditional L2 bridges or settlement constraints present by Ethereum. By using Celestia for consensus and data availability, Camp operates as an independent settlement layer optimized purely for execution speed.

Camp extends this foundation with a critical innovation: dedicated blockspace allocation for IP operations. This ensures IP transactions and agent coordination receive guaranteed computational resources regardless of network congestion.

Together, these architectural choices enable:

1. **50,000 TPS+**

2. **Giga throughput**
3. **Sub second blocks**
4. **Sovereign settlement**
5. **Dedicated Blockspace for IP** (Camp enhancement)

This architecture - sovereign rollup performance combined with IP-optimized blockspace allocation - is described in detail below.

2.1 Sovereign Rollups

Unlike Layer 2 solutions that require an enshrined bridge to Ethereum for settlement, sovereign rollups operate as independent networks that leverage external blockchains solely for data availability and ordering but maintain their own canonical state like L1s. Camp implements this architecture to achieve complete sovereignty over its execution environment while inheriting Celestia's consensus and data availability guarantees.

The distinction is fundamental: L2 rollups defer their canonical state and transaction validity to a settlement layer through enshrined bridge contracts. Every state transition must be verified by a smart contract on Ethereum, creating dependencies on Ethereum's gas costs, throughput limitations, and governance decisions. While the introduction of EIP-4844's blob mechanism reduces data availability costs by allowing rollups to post data in temporary blob storage rather than permanent calldata, this optimization introduces a trade-off. Blobs are pruned from Ethereum nodes after approximately 18 days, meaning historical data availability relies on third-party archival services or the rollup's own infrastructure rather than being permanently guaranteed by Ethereum validators.

Camp determines its canonical state through its own node network, functioning as a settlement layer in its own right. There is no external smart contract on Ethereum or any other chain that defines what constitutes a valid Camp block and the finalised state and blocks are posted on celestia. This architectural choice eliminates dependence on external settlement layers while ensuring data remains accessible through Celestia's consensus guarantees, avoiding the pruning limitations inherent in Ethereum's blob storage model.

This sovereignty enables Camp's specialized IP infrastructure. Without bridge overhead consuming block space and computational resources, Camp dedicates its full gigagas throughput to execution. Without settlement constraints, Camp introduces architecture changes dedicated to IP blockspace allocation, custom sequencing rules for IP transactions, and the ability to serve as a settlement layer for application-specific SideCAMPs. The removal of bridge dependencies alone recovers 20-30% of computational capacity typically consumed by settlement proof generation and verification.

For intellectual property operations, sovereignty translates directly to performance: IP registration that settles in 1 second rather than waiting for Ethereum finality, licensing transactions at sub-cent costs rather than competing with MEV bots for block inclusion, and protocol upgrades that can respond to emerging IP use cases without requiring approval from a generalized L1 governance process.

2.2 Node Architecture

The ABC Stack is architecturally split into two distinct node types:

1. **Celestia Nodes (for DA and Consensus):**
    - **Full nodes:** Store entire chain history and participate in Celestia's decentralized PoS consensus

- **Light nodes:** Perform data availability sampling (DAS) to verify data availability without storing full blocks, ensuring decentralized verification

2. **Camp Execution Nodes:**

- **BaseCAMP nodes:** Primary EVM execution client handling transaction execution, state management, and exposing standard JSON-RPC/WS interfaces.
- **SideCAMP nodes:** Specialized execution environments for domain-specific optimizations (e.g., IP blockspace) while maintaining compatibility with unified Camp state

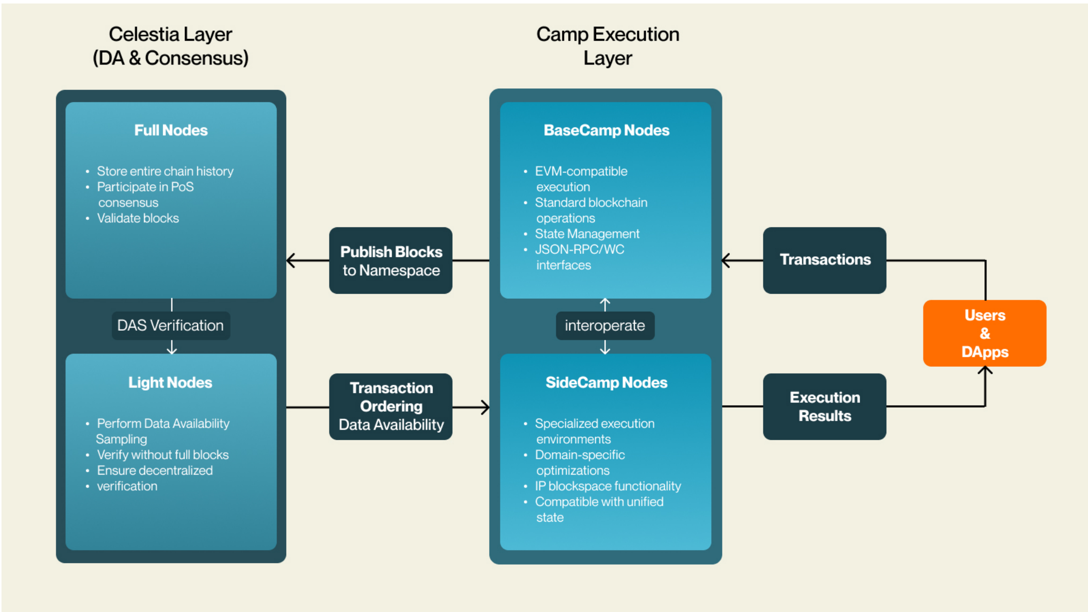

Figure 2: Camp Network node architecture

2.3 Execution Layer: ABC Stack

Execution refers to the computational layer that processes transactions, runs smart contracts, updates account balances, and computes state transitions. The execution layer takes a sequence of transactions and produces a new state from the proposed state transitions.

The ABC Stack execution layer is built on **Reth**. This provides full EVM compatibility while delivering superior performance through Rust's memory safety and zero-cost abstractions compared to traditional Go-based clients. The Rust implementation provides measurable performance advantages in transaction processing, state management, and memory efficiency directly contributing to Camp's gigagas-level throughput and ability to support specialized features.

2.4 Consensus and Data Availability Layer: Celestia

Celestia uses a CometBFT proof-of-stake consensus mechanism with a decentralized validator set. Validators achieve Byzantine Fault Tolerant (BFT) consensus on the ordering of transactions, producing blocks that contain data from multiple rollups simultaneously. Each rollup's data occupies a distinct namespace within Celestia blocks, allowing different chains to share infrastructure while maintaining logical separation. Traditional blockchains require full nodes to download entire blocks to verify data availability, creating a fundamental scalability bottleneck blocks can only be as large as what a single full node can process. Celestia solves this through probabilistic verification.

Light clients on Celestia perform random sampling of small portions of each block, requesting specific data chunks and verifying the corresponding cryptographic proofs. Each light client samples different random portions of the block. As more light clients join the network and perform independent random sampling, the collective confidence that all data in the block is available increases exponentially. If any portion of a block is withheld, the probability that at least one light client will detect the missing data approaches certainty as the number of samplers grows.

2.5 State Management

Separation of State and Execution enables Camp Network to achieve exceptional performance characteristics that monolithic architectures cannot match. By delegating consensus and data availability to Celestia while focusing exclusively on transaction execution, Camp processes upwards of 50,000+ transactions per second.

This architectural separation also eliminates state bloat, a critical problem plaguing traditional blockchains. In monolithic chains, every node must store the complete history of all transactions and state changes, creating an ever-growing burden that makes running nodes increasingly expensive and excludes participants with limited resources. Camp's nodes only need to maintain current state for execution purposes historical data lives on Celestia, where it remains permanently accessible and verifiable by anyone who needs it. The state separation model comes directly from the ABC Stack, which stores historical data on Celestia while execution nodes maintain only live state.

The result is a blockchain that scales without compromise. Camp maintains zero state bloat on execution nodes while preserving complete verifiability through Celestia's permissionless data availability layer. Anyone can verify Camp's entire history by retrieving block data from Celestia and re-executing transactions, but they're not forced to store that data indefinitely.

2.6 Gigagas throughput

Gas is the fundamental unit of computational work on EVM-compatible blockchains, measuring the effort required to execute any operation, from simple token transfers ∼(21,000 gas) to complex smart contract interactions (which can consume hundreds of thousands of gas units). Rather than using transactions per second (TPS), which fails to capture the varying complexity of different operations, the community increasingly measures performance in gas per second, metric that accurately reflects a network's true computational capacity as the virtual Machine it is.

Camp Network achieves **1 gigagas per second** (1,000 megagas/s), representing a leap in throughput. With 1-second block times and a 1 gigagas block limit, Camp processes approximately 1 billion gas units every second. To contextualize this performance: Ethereum's Layer 1 processes roughly 1-1.5 megagas per second with its ∼15 million gas target per block and 12-second block times. This means Camp delivers approximately **700-1000x the computational throughput of Ethereum mainnet**, or the equivalent of 25 Ethereum blocks worth of execution capacity in a single Camp block.

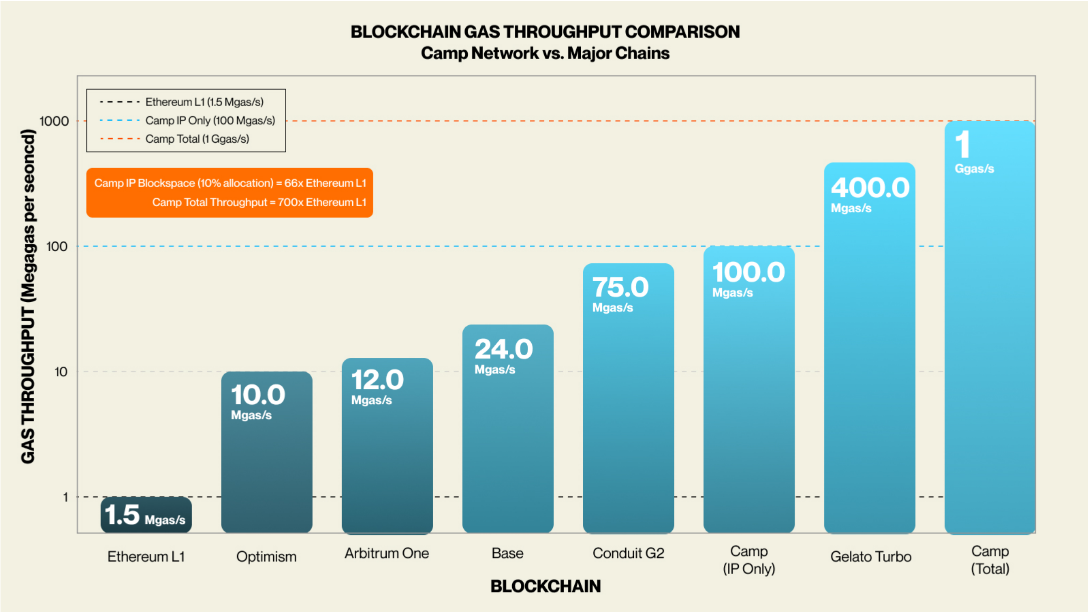

Figure 3: Blockchain gas throughput comparison - Camp Network vs. major chains

This throughput advantage translates directly into real-world capabilities. Where Ethereum processes 15-30 transactions per second depending on gas complexity, Camp can handle upwards of 50,000+ transactions per second while maintaining sub-cent transaction costs even during peak demand. A single Camp block can process approximately 47,000 simple ERC-20 transfers or 10,000 Uniswap-style swaps, operations that would require 25 Ethereum blocks and 5 minutes to complete on Ethereum L1, but execute in just 1 second on Camp. This gigagas-level performance positions Camp among an emerging class of next-generation blockchains purpose-built for Web3 applications that demand Web2-like responsiveness: decentralized social networks, onchain gaming, high-frequency DeFi, and the massive-scale intellectual property registration and licensing operations that Camp is specifically designed to support.

2.7 BaseCAMP And SideCAMPs

Camp Network has onboarded a strong native ecosystem of Web 2.5 teams, which are dApps built onchain with the goal of onboarding the next generation of users. A major UX challenge for most dApps today is high latency - users often face delays between clicking an action and seeing results, largely due to congested blockspace on general-purpose L1s.

To solve this, Camp's architecture introduces **BaseCAMP** as the primary Layer 1 optimized for IP management, while **SideCAMPs** are application-specific chains that settle back to BaseCAMP. This architecture creates a scalable and developer-friendly environment by providing dedicated blockspace for each application while preserving security and interoperability.

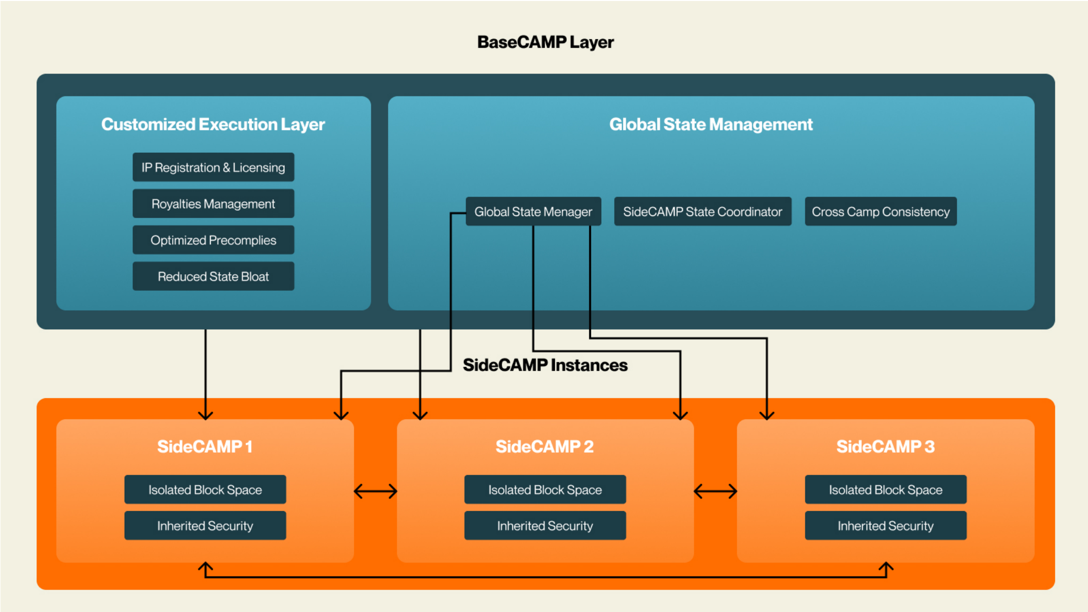

Figure 4: BaseCAMP and SideCAMP architecture

This introduces two elements:

Parallelization → The BaseCAMP and SideCAMPs architecture enables parallel execution across specialized environments, preventing application-specific congestion from impacting core network functionality. This isolates application load from BaseCAMP, which maintains focus on processing fundamental IP transactions including copyright registration, ownership transfers, and licensing agreements. By distributing computational workload across multiple execution environments, Camp avoids the congestion bottlenecks that plague single-execution-layer blockchains.

Scalability → Camp's modular architecture supports deployment of multiple SideCAMPs, each optimized for specific application domains or use case requirements. The network can accommodate n number of SideCAMPs operating simultaneously, with each inheriting the security guarantees of Camp's unified consensus model and Celestia's data availability layer.

2.8 Dedicated IP blockspace and sequencing

Camp implements specialized infrastructure for intellectual property transactions through a dedicated blockspace allocation mechanism that operates at the sequencer level. Before transactions enter the sequencing queue, Camp's execution layer screens incoming transactions to identify IP-related operations, including IP registration, ownership transfers, licensing agreements, and royalty distributions. These IP transactions receive priority execution status and are guaranteed access to a reserved portion of each block's computational capacity.

Specifically, Camp allocates **10% of its total gas capacity**, approximately 100 megagas per block, exclusively for IP transactions. This reservation ensures that content creators, innovators, and IP-intensive applications always have guaranteed throughput regardless of network congestion from DeFi trading, NFT minting, or other high-volume activities.

This dedicated IP allocation alone, representing just one-tenth of Camp's total capacity, exceeds Ethereum's entire mainnet throughput by 66-100x. While Ethereum processes 1-1.5 megagas per second across all transaction types, Camp provides 100 megagas per second dedicated exclusively to intellectual property operations, with an additional 900 megagas per second available for general-purpose execution. This architectural design prevents the scenario common on congested blockchains where critical IP transactions must compete for blockspace against speculative trading activity, ensuring that the foundational infrastructure for digital creativity and innovation remains consistently accessible and affordable.

3 Transaction Flow

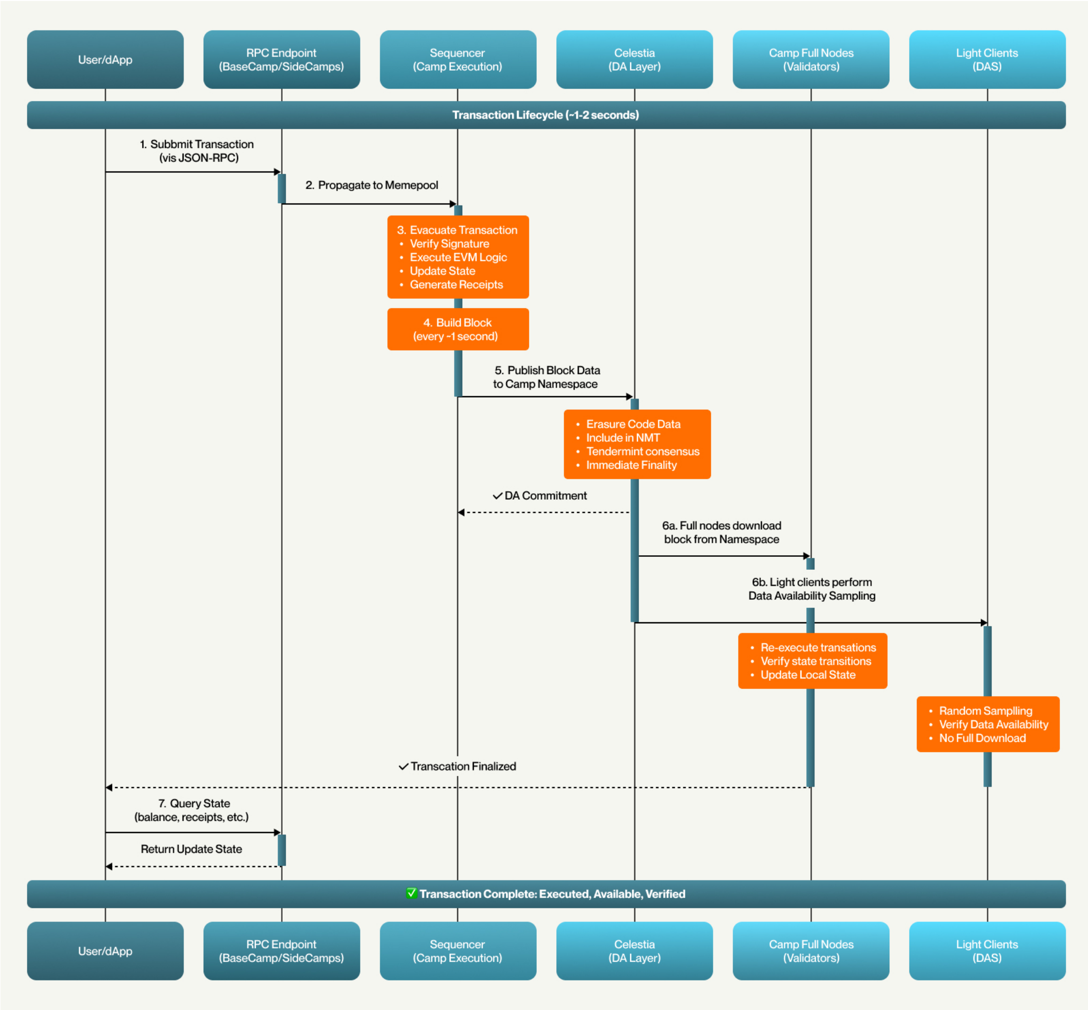

Figure 5: Complete transaction lifecycle in Camp Network

3.1 1. Transaction Submission

Users submit transactions to Camp Network through standard JSON-RPC endpoints exposed by BaseCamp or SideCamps nodes. This interface is identical to Ethereum's RPC specification, allowing existing wallets, dApps, and developer tools to interact with Camp without modification. The transaction contains standard EVM fields: sender address, recipient, value, gas limit, gas price (or EIP-1559 fee parameters), and calldata for smart contract interactions.

3.2 2. Transaction Propagation to Sequencer

The transaction is propagated to Camp's sequencer—node responsible for ordering transactions and producing blocks. The sequencer maintains a mempool of pending transactions:

- **IP Transaction Reserved Capacity:** Up to 300 megagas per block (30% of Camp's 1 gigagas total block capacity)
- **Standard Transaction Space:** Remaining 700 Million gas for general-purpose EVM operations

- **Dynamic Allocation:** If IP blockspace is underutilized, unused capacity becomes available for standard transactions

3.3 3. Execution and State Transition

The sequencer executes transactions using Camp's EVM execution environment. Sequencer:

- Verifies the signature and checks that the sender has sufficient balance for gas fees
- Executes the state transitions
- Computes the new state root
- Generates execution receipts containing logs, events, and transaction outcomes

This execution happens on Camp's execution layer (BaseCamp or SideCamps), completely independent of Celestia. The sequencer determines transaction validity and state transitions autonomously.

3.4 4. Block Production and Data Publishing

Once the sequencer has executed a batch of transactions (typically every 1 second based on Camp's block time), it packages them into a block containing:

Field	**Description**
slot	the slot the block belongs to
proposer_index	the ID of the validator proposing the block
parent_root	the hash of the preceding block
state_root	the root hash of the state object
body	an object containing several fields, as defined below

Table 1: Block header structure

Complete transaction data for all included transactions.

3.5 5. Data Availability Commitment to Celestia

The sequencer publishes the complete block data to Celestia's data availability layer within Camp's dedicated namespace. This involves:

- Encoding the block data with erasure coding for redundancy
- Receiving a commitment from Celestia that the data has been included in a Celestia block
- The data becomes part of Celestia's Namespaced Merkle Tree, provably available through DAS (Data Availability Sampling)

3.6 6. Finality and Verification

Once Celestia commits to the block data:

- The block achieves finality through Celestia's Tendermint consensus which provides Single Slot finality.
- Independent nodes can download and verify the camp state by re-executing every transaction.
- Users receive confirmation that their transaction is finalized and irreversible

4 Future Roadmap

4.1 10-Second Target

Camp's current architecture achieves finality through Celestia's Tendermint consensus, which provides immediate Byzantine Fault Tolerant finality once blocks are committed to Celestia's chain. However, the time between transaction submission and final Celestia commitment can be optimized. Through enhanced coordination between Camp's sequencer and Celestia's block production, combined with optimistic confirmation mechanisms, Camp targets reducing perceived finality time to 10 seconds or less. This improvement maintains the security guarantees of Celestia's consensus while providing users with faster confirmation of irreversible state transitions. Applications requiring rapid finality, such as high-frequency trading or real-time content licensing, benefit significantly from this optimization.

4.2 Solana Virtual Machine (SVM) SideCamp

The introduction of SVM-compatible SideCamps represents a major architectural expansion, enabling developers to deploy Solana programs on Camp Network while benefiting from Celestia's data availability and Camp's unified settlement layer. This multi-VM approach allows applications originally built for Solana's execution environment to operate within Camp's ecosystem without code rewrites. The SVM SideCamp maintains compatibility with Solana's accounts model, rust rbpf, and runtime while settling to Camp's shared state layer. Cross-VM composability enables novel applications: a DeFi protocol might execute high-frequency operations on the SVM SideCamp while interfacing with IP licensing contracts on EVM BaseCamp.

4.3 WebAssembly (WASM) Runtime Support

WASM-compatible SideCamps unlock polyglot development, allowing smart contracts written in Rust, C++, Go, or other languages to execute on Camp. This expansion dramatically increases the developer base that can build on Camp without learning Solidity. WASM's performance characteristics enable compute-intensive applications like on-chain machine learning inference or complex cryptographic operations to execute efficiently.

4.4 Homomorphic Encryption for Private Computation

Fully homomorphic encryption enables computation on encrypted data, allowing IP-intensive applications to process confidential information without exposing it. Use cases include private recommendation systems that learn from user behavior without seeing raw data, confidential licensing auctions where bid amounts remain hidden, and collaborative filtering without revealing individual preferences.

5 Conclusion

Camp Network establishes the first blockchain infrastructure specifically engineered for intellectual property operations. The protocol's sovereign rollup architecture removes the computational and economic constraints that have limited previous attempts at onchain IP management.

The performance metrics speak to practical requirements, not theoretical maximums. Gigagas throughput enables millions of IP registrations daily. Sub-second finality makes real-time licensing viable. Dedicated IP blockspace ensures creators always have guaranteed access regardless of network congestion from other activities. These are not optimizations of existing systems, but necessary capabilities for IP infrastructure at internet scale.

These architectural decisions reflect the unique demands of intellectual property workloads. IP transactions require different guarantees than DeFi swaps or NFT mints. They need persistent availability for registration, complex state management for licensing relationships, and computational resources for royalty calculations across derivative chains. Camp's specialized execution environment, implemented through BaseCamp and SideCamps, provides these capabilities without forcing IP operations to compete for resources with unrelated network activity.

The integration of AI agents as first-class participants acknowledges where the IP economy is headed. When autonomous systems can create, license, and monetize intellectual property, the infrastructure must support programmatic interaction at scale. Camp's architecture anticipates this reality, providing the throughput and cost structure necessary for millions of agent-driven transactions.

Future protocol expansions to support multiple virtual machines and advanced cryptographic primitives will extend Camp's capabilities. But the core achievement stands: intellectual property now has dedicated blockchain infrastructure that matches its economic importance. The $80 trillion digital economy built on IP no longer depends on general-purpose chains that treat intellectual property as an afterthought.

6 References

1. Al-Bassam, M. (2019). *LazyLedger: A Distributed Data Availability Ledger With Client-Side Smart Contracts.* arXiv:1905.09274. Retrieved from https://arxiv.org/abs/1905.09274

2. Haas, A., Rossberg, A., Schuff, D., Titzer, B., Gohman, D., Wagner, L., Zakai, A., & Bastien, J. (2017). *Bringing the Web up to Speed with WebAssembly.* Retrieved from https://doi.org/10.1145/3062341.3062363

3. Reth Developers. (n.d.). *Reth: Fast, modular Ethereum client.* Retrieved from https://reth.rs/overview/

4. Squads. (2023). *Solana SVM: Sealevel virtual machine overview.* Retrieved from https://squads.xyz/blog/solana-svm-sealevel-virtual-machine

5. Abundance. (n.d.). *ABC Stack: High-performance EVM Sovereign Rollups.* Retrieved from https://www.abundance.xyz/

6. Buterin, V., et al. (2023). *EIP-4844: Shard Blob Transactions.* Ethereum Improvement Proposals. Retrieved from https://eips.ethereum.org/EIPS/eip-4844

7. Ethereum Foundation. (n.d.). *Ethereum.org: The complete guide to Ethereum.* Retrieved from https://ethereum.org/

8. Celestia Labs. (n.d.). *Celestia: The Modular Blockchain Network.* Retrieved from https://celestia.org/

9. Gelato Network. (n.d.). *Gelato: Smart Wallets, Gasless UX & Rollups on One Platform.* Retrieved from https://gelato.cloud/