Agentic Consensus over ZAP

Multi-Agent Intelligence with Blockchain-Grade Finality

Lux Industries, Inc.

April 2026 — v1.0

Abstract

We present a system for multi-agent AI consensus using the ZAP (Zero-copy Application Protocol) wire protocol and Lux Quasar consensus engine. Heterogeneous AI agents—running different models, architectures, and specializations—propose, vote, and converge on collective decisions with the same finality guarantees used by blockchain validators. The system achieves sub-second agent consensus with 17× lower latency than protobuf-based alternatives, supports 30+ programming languages natively, and provides cryptographic non-repudiation of agent decisions via BLS aggregate signatures. Agents communicate over ZAP natively; MCP (Model Context Protocol) serves as a compatibility shim for clients that require it.

The Problem

Current multi-agent AI systems lack:

1. **Convergence guarantees**: No formal mechanism to ensure agents reach agreement. Most systems use "last writer wins" or simple majority voting without Byzantine fault tolerance.
2. **Heterogeneous participation**: Agents running different models (Claude, GPT, Qwen, Gemini, local models) cannot interoperate at the protocol level.
3. **Cryptographic accountability**: No way to prove which agent proposed what, or that a decision was genuinely collective.
4. **Performance**: JSON-RPC and protobuf add 10–100ms per message. Agent swarms need sub-millisecond messaging.

Architecture

Protocol Stack

```
Application Layer:  Agent Logic (propose, vote, synthesize)
                    |
Consensus Layer:    Lux Quasar (Focus -> Prism -> Horizon)
                    |
Transport Layer:    ZAP (zero-copy, 17x faster than protobuf)
                    |
Network Layer:      TCP/UDP, mDNS discovery, TLS mutual auth
```

ZAP: The Wire Protocol

ZAP is a zero-copy binary protocol with implementations in 30+ languages:

Metric	ZAP	Protobuf
Relative decode throughput	17×	baseline
Memory per msg	0 allocs	11–121 allocs
GC pressure	14× lower	baseline
Languages	30+	12

ZAP achieves zero-copy deserialization: the message is read directly from the network buffer without intermediate copies or allocations. This makes agent-to-agent communication as fast as in-process function calls.

MCP Compatibility

MCP (Model Context Protocol) is a JSON-RPC based protocol used by Anthropic Claude and other AI clients. For clients that only speak MCP:

```
Claude Code -> hanzo-mcp (JSON-RPC) -> ZAP bridge -> Consensus
```

The bridge is stateless and adds <1ms overhead. Native ZAP agents bypass MCP entirely.

Consensus Protocol

Agent consensus follows the Quasar three-phase protocol adapted for AI decision-making:

Phase 1: Propose (Focus)

Each agent independently generates a response to the prompt. Proposals include:

- Response content
- Confidence score (0–1)
- Reasoning chain (optional, for auditing)
- Agent capabilities and specialization

Proposals are broadcast to all agents via ZAP multicast.

Phase 2: Vote (Prism)

Each agent evaluates all proposals and submits ranked preferences:

- Quality score per proposal (0–1)
- Ranked preference ordering
- Specific critiques or improvements

Votes are weighted by agent confidence and historical accuracy:

$$\text{weight}_i = \text{confidence}_i \times \text{accuracy}_i \times \text{specialization\_match}_i$$

Phase 3: Synthesize (Horizon)

The system computes consensus scores:

$$\text{score}(p) = \frac{\sum_{i=1}^{N} w_i \cdot v_i(p)}{\sum_{i=1}^{N} w_i}$$

where w_i is agent i's weight and $v_i(p)$ is their score for proposal p.

Convergence criterion:

$$\text{score}(p^*) > \theta \quad \text{and} \quad \text{score}(p^*) - \text{score}(p^{(2)}) > \delta$$

where $\theta = 0.67$ (supermajority) and $\delta = 0.1$ (clear winner margin).

If convergence is not reached, top-K proposals are merged and a new round begins.

Finality

The final decision is signed with BLS aggregate signature by all participating agents, providing:

- **Non-repudiation**: Every agent's vote is cryptographically bound
- **Auditability**: The full decision trail is verifiable
- **Tamper resistance**: No agent can retroactively change their vote

Agent Mesh Topology

Discovery

Agents discover each other via:

1. **mDNS**: Local network agents (sub-millisecond discovery)
2. **ZAP handshake**: Capability exchange (model type, specialization, compute budget)
3. **Registry**: Global agent registry for cross-network participation

Heterogeneous Agents

The system supports any AI model:

Agent	Role	Transport
Claude (Opus)	Deep reasoning, code review	ZAP via hanzo-mcp
GPT-4	Broad knowledge, creative	ZAP native
Qwen-3	Multilingual, math	ZAP native
Gemini	Multimodal, search	ZAP via bridge
Local LLM	Privacy-sensitive tasks	ZAP native
Copilot	Code generation	ZAP via bridge

Specialization Weighting

Not all agents are equal for all tasks. The system matches agent specializations to task requirements:

- Code review task → higher weight for code-specialized agents
- Security audit → higher weight for security-focused agents

- Creative writing → higher weight for creative models
- Math proof → higher weight for reasoning models

Implementation

The agentic consensus system is implemented in `lux/consensus/ai/`:

File	Purpose
`agent.go`	Agent lifecycle, proposal, voting (13 functions)
`engine.go`	Module orchestration, learning, coordination (16 functions)
`interfaces.go`	Type definitions, consensus data interface
`specialized.go`	Domain-specific agent configurations
`models.go`	Model registry and capability matching
`modules.go`	Pluggable inference/decision/learning modules

The agentic ZAP example (`examples/agentic-zap/`, 925 lines) demonstrates a full mesh of heterogeneous agents reaching consensus on a shared task.

Performance

Metric	ZAP Agents	JSON-RPC Agents
Message latency (p50)	0.1ms	12ms
Message latency (p99)	0.3ms	85ms
Consensus (5 agents)	50ms	800ms
Consensus (20 agents)	200ms	3.2s
Messages/sec/agent	100K	2K
Memory per agent	4MB	45MB

Comparison to Existing Systems

Property	Ours	AutoGen	CrewAI	LangGraph
Formal consensus	✓	×	×	×
Byzantine tolerance	✓	×	×	×
Crypto accountability	✓	×	×	×
Heterogeneous models	✓	✓	✓	✓
Sub-ms messaging	✓	×	×	×
30+ language support	✓	×	×	×
On-chain finality	✓	×	×	×

No existing multi-agent framework provides formal consensus guarantees, cryptographic accountability, or sub-millisecond messaging. These properties come from building on blockchain consensus infrastructure rather than ad-hoc coordination.

Conclusion

Agentic consensus over ZAP unifies AI agent coordination with blockchain consensus primitives. By using the same Quasar protocol that finalizes blocks, agent decisions inherit the same finality, accountability, and Byzantine fault tolerance properties. ZAP provides the performance (17× faster than protobuf, 30+ languages), while Quasar provides the guarantees (convergence, non-repudiation, auditability).

MCP compatibility ensures existing AI clients can participate, but the native path is ZAP—zero-copy, zero-allocation, sub-millisecond agent communication at any scale.

References

[1] Lux Industries, "ZAP: Zero-Copy Application Protocol," 2023.

[2] Lux Industries, "Triple-Proof Quantum Finality," 2026.

[3] Anthropic, "Model Context Protocol Specification," 2024.