Fogo: A High-Performance SVM Layer 1

Version 1.0

Abstract

This paper introduces Fogo, a novel layer 1 blockchain protocol delivering breakthrough performance in throughput, latency, and congestion management. As an extension of the Solana protocol, Fogo maintains full compatibility at the SVM execution layer, allowing existing Solana programs, tooling, and infrastructure to migrate seamlessly while achieving significantly higher performance and lower latency.

Fogo contributes three novel innovations:

- A unified client implementation based on pure Firedancer, unlocking performance levels unattainable by networks with slower clients—including Solana itself.
- Multi-local consensus with dynamic colocation, achieving block times and latencies far below those of any major blockchain.
- A curated validator set that incentivizes high performance and deters predatory behavior at the validator level.

These innovations deliver substantial performance gains while preserving the decentralization and robustness essential to a layer 1 blockchain.

1. Introduction

Blockchain networks face an ongoing challenge in balancing performance with decentralization and security. Today's blockchains suffer severe throughput limitations that make them unsuitable for global financial activity. Ethereum processes fewer than 50 transactions per second (TPS) on its base layer. Even the most centralized layer 2s handle less than 1,000 TPS. While Solana was designed for higher performance, limitations from client diversity currently cause congestion at 5,000 TPS. In contrast, traditional financial systems like NASDAQ, CME, and Eurex regularly process over 100,000 operations per second.

Latency presents another critical limitation for decentralized blockchain protocols. In financial markets—especially for price discovery on volatile assets—low latency is essential for market quality and liquidity. Traditional market participants operate with end-to-end latencies at millisecond or sub-millisecond scales. These speeds are only

achievable when market participants can co-locate with the execution environment due to the speed of light constraints.

Traditional blockchain architectures use globally distributed validator sets that operate without geographic awareness, creating fundamental performance limitations. Light itself takes over 130 milliseconds to circumnavigate the globe at the equator, even traveling in a perfect circle—and real-world network paths involve additional distance and infrastructure delays. These physical limitations compound when consensus requires multiple communication rounds between validators. These inter-regional latencies compound when consensus requires multiple communication rounds between validators. As a result, networks must implement conservative block times and finality delays to maintain stability. Even under optimal conditions, a globally distributed consensus mechanism cannot overcome these basic networking delays.

As blockchains integrate further with the global financial system, users will demand performance comparable to today's centralized systems. Without careful design, meeting these demands could significantly compromise blockchain networks' decentralization and resilience. To address this challenge, we propose the *Fogo* layer one blockchain. Fogo's core philosophy is to maximize throughput and minimize latency through two key approaches: first, using the most performant client software on an optimally decentralized validator set; and second, embracing co-located consensus while preserving most of the decentralization benefits of global consensus.

2. Outline

The paper is broken down into sections covering the major design decisions around Fogo. Section 3 covers the relationship of Fogo to the Solana blockchain protocol and its strategy with regards to client optimization and diversity. Section 4 covers multi-local consensus, its practical implementation, and the tradeoffs it makes relative to global or local consensus. Section 5 covers Fogo's approach to initializing and maintaining the validator set. Section 6 covers prospective extensions that may be introduced after genesis.

3. Protocol and Clients

At a base layer Fogo starts by building on top of the most performant widely used blockchain protocol to date, Solana. The Solana network already comes with numerous optimization solutions, both in terms of protocol design and client implementations. Fogo targets maximum possible backwards compatibility with Solana, including full compatibility at the SVM execution layer and close compatibility with TowerBFT

consensus, Turbine block propagation, Solana leader rotation and all other major components of the networking and consensus layers. This compatibility allows Fogo to easily integrate and deploy existing programs, tooling and infrastructure from the Solana ecosystem; as well as benefit from continuous upstream improvements in Solana.

However unlike Solana, Fogo will run with a single canonical client. This canonical client will be the highest performance major client running on Solana. This allows Fogo to achieve significantly higher performance because the network will always run at the speed of the fastest client. Whereas Solana, limited by client diversity will always be bottlenecked by the speed of the slowest client. For now and the foreseeable future this canonical client will be based on the Firedancer stack.

3.1 Firedancer

Firedancer is Jump Crypto's high-performance Solana-compatible client implementation, showing substantially higher transaction processing throughput than current validator clients through optimized parallel processing, memory management, and SIMD instructions.

Two versions exist: "Frankendancer," a hybrid using Firedancer's processing engine with the rust validator's networking stack, and the full Firedancer implementation with a complete C networking stack rewrite, currently in late-stage development.

Both versions maintain Solana protocol compatibility while maximizing performance. Once completed, the pure Firedancer implementation is expected to set new performance benchmarks, making it ideal for Fogo's high-throughput requirements. Fogo will start with a Frankendancer based network then eventually transition to pure Firedancer.

3.2 Canonical Clients vs. Client Diversity

Blockchain protocols operate through client software that implements their rules and specifications. While protocols define the rules of network operation, clients translate these specifications into executable software. The relationship between protocols and clients has historically followed different models, with some networks actively promoting client diversity while others naturally converge on canonical implementations.

Client diversity traditionally serves multiple purposes: it provides implementation redundancy, enables independent verification of protocol rules, and theoretically reduces the risk of network-wide software vulnerabilities. The Bitcoin network demonstrates an interesting precedent - while multiple client implementations exist, Bitcoin Core serves as the de facto canonical client, providing the reference implementation that defines practical network behavior.

However, in high-performance blockchain networks, the relationship between protocol and client implementation becomes more constrained. When a protocol approaches the physical limits of computing and networking hardware, the space for implementation diversity naturally contracts. At these performance boundaries, optimal implementations must converge on similar solutions as they confront the same physical limitations and performance requirements. Any significant deviation from optimal implementation patterns would result in degraded performance that makes the client non-viable for validator operation.

This dynamic is particularly visible in networks targeting minimum possible block times and maximum transaction throughput. In such systems, the theoretical benefits of client diversity become less relevant, as the overhead of maintaining compatibility between different client implementations can itself become a performance bottleneck. When pushing blockchain performance to physical limits, client implementations will necessarily share core architectural decisions, making the security benefits of implementation diversity largely theoretical.

3.3 Protocol Incentives for Performant Clients

While Fogo allows any conforming client implementation, its architecture naturally incentivizes using the highest-performing client available, driven by the practical demands of high-performance co-located operations.

Unlike traditional networks where geographic distance creates the main bottlenecks, Fogo's co-located design means client implementation efficiency directly determines validator performance. In this environment, network latency is minimal, making client speed the critical factor.

The network's dynamic block time and size parameters create economic pressure to maximize throughput. Validators must choose between using the fastest client or risking penalties and reduced revenue. Those running slower clients either risk missing blocks by voting for aggressive parameters or lose revenue by voting for conservative ones.

This creates natural selection for the most efficient client implementation. In Fogo's co-located environment, even small performance differences become significant - a slightly slower client will consistently underperform, leading to missed blocks and penalties. This optimization happens through validator self-interest, not protocol rules. While client choice cannot be directly enforced by protocol, economic pressures naturally drive the network toward the most efficient implementation while maintaining competitive client development.

4. Multi-Local Consensus

Multi-local consensus represents a novel approach to blockchain consensus that dynamically balances the performance benefits of validator co-location with the security advantages of geographic distribution. The system allows validators to coordinate their physical locations across epochs while maintaining distinct cryptographic identities for different zones, enabling the network to achieve ultra-low latency consensus during normal operation while preserving the ability to fall back to global consensus when needed.

Fogo's multi-local consensus model draws inspiration from established practices in traditional financial markets, particularly the "follow the sun" trading model used in foreign exchange and other global markets. In traditional finance, market making and liquidity provision naturally migrate between major financial centers as the trading day progresses – from Asia to Europe to North America – allowing for continuous market operation while maintaining concentrated liquidity in specific geographic regions. This model has proven effective in traditional finance because it recognizes that while markets are global, the physical limitations of networking and human reaction times make some degree of geographic concentration necessary for optimal price discovery and market efficiency.

4.1 Zones and Zone Rotation

A zone represents a geographical area where validators co-locate to achieve optimal consensus performance. Ideally, a zone is a single data center where network latency between validators approaches hardware limits. However, zones can expand to encompass larger regions when necessary, trading some performance for practical considerations. The exact definition of a zone emerges through social consensus among validators rather than being strictly defined in the protocol. This flexibility allows the network to adapt to real-world infrastructure constraints while maintaining performance objectives.

The network's ability to rotate between zones serves multiple critical purposes:

1. **Jurisdictional Decentralization**: Regular zone rotation prevents the capture of consensus by any single jurisdiction. This maintains the network's resistance to regulatory pressure and ensures no single government or authority can exert long-term control over network operation.
2. **Infrastructure Resilience**: Data centers and regional infrastructure can fail for numerous reasons - natural disasters, power outages, networking issues, hardware failures, or maintenance requirements. Zone rotation ensures the network isn't permanently dependent on any single point of failure. Historical examples of major

data center outages, such as those caused by severe weather events or power grid failures, demonstrate the importance of this flexibility.

3. **Strategic Performance Optimization**: Zones can be selected to optimize for specific network activities. For example, during epochs containing significant financial events (such as Federal Reserve announcements, major economic reports, or market opens), validators might choose to locate consensus near the source of this price-sensitive information. This capability allows the network to minimize latency for critical operations while maintaining flexibility for different use cases across epochs.

4.2 Key Management

The protocol implements a two-tier key management system that separates long-term validator identity from zone-specific consensus participation. Each validator maintains a global key pair that serves as their root identity in the network. This global key is used for high-level operations such as stake delegation, zone registration, and participation in global consensus. The global key should be secured with the highest possible security measures, as it represents the validator's ultimate authority in the network.

Validators can then delegate authority to zone-specific sub-keys through an on-chain registry program. These sub-keys are specifically authorized for consensus participation within designated co-location zones. This separation serves multiple security purposes: it allows validators to maintain different security models for different key types, it minimizes the exposure of global keys by keeping them offline during normal operation, and it reduces the risk of key compromise during physical infrastructure transitions between zones.

The delegation of zone-specific keys is managed through an on-chain program that maintains a registry of authorized zone keys for each validator. While validators can register new zone keys at any time using their global key, these registrations only take effect at epoch boundaries. This delay ensures that all network participants have time to verify and record new key delegations before they become active in consensus.

4.3 Zone Proposal and Activation

New zones can be proposed through an on-chain governance mechanism using global keys. However, to ensure network stability and give validators adequate time to prepare secure infrastructure, proposed zones have a mandatory delay period before they become eligible for selection. This delay, set as a protocol parameter, must be sufficiently long to allow validators to:

- Secure appropriate physical infrastructure in the new zone

- Establish secure key management systems for the new location
- Set up and test networking infrastructure
- Perform necessary security audits of the new facility
- Establish backup and recovery procedures

The delay period also serves as a security measure against potential attacks where a malicious actor might attempt to force consensus into a zone where they have infrastructural advantages. By requiring advance notice for new zones, the protocol ensures that all validators have a fair opportunity to establish presence in any zone that might be selected for consensus.

Only after a zone has completed this waiting period can it be selected through the regular zone voting process for future epochs. This careful approach to zone activation helps maintain network security and stability while still allowing for the addition of new strategic locations as network requirements evolve.

4.4 Zone Selection Voting Process

The selection of consensus zones occurs through an on-chain voting mechanism that balances the need for coordinated validator movement with network security. Validators must achieve quorum on each future epoch's co-location zone within a configurable quorum time before the epoch transition. In practice, the epoch schedule may be determined with some lead time, such that voting during epoch n selects the zone for epoch n + k. Votes are cast through an on-chain registry program using validators' global keys, with voting power weighted by stake. This process uses global keys rather than zone keys since it is not latency-sensitive and requires maximum security.

The voting process requires a supermajority of stake weight to establish quorum, ensuring that a small group of validators cannot unilaterally force a zone change. If validators fail to achieve quorum within the designated timeframe, the network automatically defaults to global consensus mode for the next epoch. This fallback mechanism ensures network continuity even when validators cannot agree on a co-location zone.

During the voting period, validators signal both their preferred zone for the next epoch and their target block time for that zone. This joint selection of location and performance parameters allows the network to optimize for both physical constraints and performance capabilities of each zone. Importantly, the voting period provides time for validators to prepare infrastructure in the selected zone, including warming up zone-specific keys and testing network connectivity. This preparation period is crucial for maintaining network stability during zone transitions.

4.5 Global Consensus Mode

Global consensus mode serves as both a fallback mechanism and a foundational safety feature of the protocol. While Fogo achieves its highest performance through zone-based consensus, the ability to fall back to global consensus ensures the network's continued operation under adverse conditions. In global consensus mode, the network operates with conservative parameters optimized for globally distributed validation: a fixed 400ms block time and reduced block size to accommodate higher network latencies between geographically dispersed validators.

The protocol enters global consensus mode through two primary paths:

- **Failed Zone Selection**: If validators fail to achieve quorum on the next epoch's consensus zone within the designated voting period, the network automatically defaults to global consensus for that epoch.
- **Runtime Consensus Failure**: If the current zone fails to achieve block finality within its designated timeout period during an epoch, the protocol immediately switches to global consensus mode for the remainder of that epoch. This fallback is "sticky" – once triggered mid-epoch, the network remains in global consensus until the next epoch transition, prioritizing stability over performance recovery.

In global consensus mode, validators participate using a designated key for global operation, which may or may not be one of their zone-specific keys, and the network maintains the same fork choice rules as zone-based consensus. While this mode sacrifices the ultra-low latency achievable in co-located zones, it provides a robust foundation for network continuity and demonstrates how Fogo maintains safety without sacrificing liveness under degraded conditions.

5. Validator Set

To achieve high performance and mitigate abusive MEV practices, Fogo will utilize a *curated validator set*. This is necessary because even a small fraction of under-provisioned validating nodes can prevent the network from reaching its physical performance limits. Initially, curation will operate through proof-of-authority before transitioning to direct permissioning by the validator set. By placing curation authority with the validator set, Fogo can enforce social layer punishment of abusive behavior like a traditional proof-of-authority system, but in a way that's no more centralized than the fork power that 2/3 of stake already holds in traditional PoS networks like Solana.

5.1 Size and Initial Configuration

Fogo maintains a permissioned validator set with a protocol-enforced minimum and maximum number of validators to ensure sufficient decentralization while optimizing for

network performance. The initial target size will be approximately 20-50 validators, though this cap is implemented as a protocol parameter that can be adjusted as the network matures. At genesis, the initial validator set will be selected by a genesis authority, which will retain temporary permissions to manage validator set composition during the network's early stages.

5.2 Governance and Transitions

The genesis authority's control over validator set membership is designed to be temporary. After an initial period of network stabilization, this authority will transition to the validator set itself. Following this transition, changes to validator set membership will require a two-thirds supermajority of staked tokens, matching the same threshold required for protocol-level changes in proof-of-stake networks.

To prevent sudden changes that could destabilize the network, protocol parameters limit validator turnover rates. No more than a fixed percentage of the validator set can be replaced or ejected within a given time period, where this percentage is a tunable protocol parameter. This ensures gradual evolution of the validator set while maintaining network stability.

5.3 Participation Requirements

Validators must meet minimum delegated stake requirements to be eligible for the validator set, maintaining compatibility with Solana's economic model while adding the permissioned component. This dual requirement – sufficient stake and set approval – ensures that validators have both economic skin in the game and the operational capabilities to maintain network performance.

5.4 Rationale and Network Governance

The permissioned validator set does not materially impact network decentralization, as in any proof-of-stake network, a two-thirds supermajority of stake can already effect arbitrary changes to the protocol through forking. Instead, this mechanism provides a formal framework for the validator set to enforce beneficial network behaviors that might otherwise be difficult to encode in protocol rules.

For example, the ability to eject validators enables the network to respond to:

- Persistent performance issues that degrade network capabilities
- Abusive MEV extraction that damages network usability
- Network destabilizing behavior that can’t be enforced directly in protocol, such as leaching but not forwarding Turbine blocks

- Other behaviors that, while potentially profitable for individual validators, harm the network's long-term value

This governance mechanism recognizes that while certain behaviors may be profitable in the short term, they can damage the network's long-term viability. By enabling the stake-weighted validator set to police such behaviors through membership control, Fogo aligns validator incentives with the network's long-term health without compromising the fundamental decentralization properties inherent to proof-of-stake systems.

6. Prospective Extensions

While Fogo's core innovations focus on multi-local consensus, client performance, and validator set management, several additional protocol extensions are under consideration for either genesis or post-launch implementation. These features would further enhance network functionality while maintaining backwards compatibility with the Solana ecosystem.

6.1 SPL Token Fee Payment

To enable broader network access and improve user experience, Fogo will potentially introduce a *fee_payer_unsigned* transaction type that allows transactions to be executed without SOL in the originating account. This feature, combined with an on-chain fee payment program, enables users to pay transaction fees using SPL tokens while maintaining protocol security and validator compensation.

The system works through an out of protocol permissionless relayer marketplace. Users construct transactions that include both their intended operations and an SPL token payment to compensate the eventual fee payer. These transactions can be validly signed without specifying a fee payer, allowing any party to complete them by adding their signature and paying the SOL fees. This mechanism effectively separates transaction authorization from fee payment, enabling accounts with zero SOL balance to interact with the network as long as they possess other valuable assets.

This feature is implemented through minimal protocol modifications, requiring only the addition of the new transaction type and an on-chain program to handle relayer compensation. The system creates an efficient market for transaction relay services while maintaining the security properties of the underlying protocol. Unlike more complex fee abstraction systems, this approach requires no changes to validator payment mechanisms or consensus rules.

7. Conclusion

Fogo represents a novel approach to blockchain architecture that challenges traditional assumptions about the relationship between performance, decentralization, and security. By combining high-performance client implementation with dynamic multi-local consensus and curated validator sets, the protocol achieves unprecedented performance without compromising the fundamental security properties of proof-of-stake systems. The ability to dynamically relocate consensus while maintaining geographic diversity provides both performance optimization and systemic resilience, while the protocol's fallback mechanisms ensure continuous operation under adverse conditions.

Through careful economic design, these mechanisms emerge naturally from validator incentives rather than through protocol enforcement, creating a robust and adaptable system. As blockchain technology continues to evolve, Fogo's innovations demonstrate how thoughtful protocol design can push the boundaries of performance while maintaining the security and decentralization properties that make blockchain networks valuable.