StripChain: A Fullstack Intent-Based Interoperability Protocol

StripChain, 2025

WORKING DRAFT, v1.2

0xnovachrono
0xnovachrono@stripchain.xyz

Opanin Akuffo
opanin.akuffo@stripchain.xyz

Narayan Prusty
narayan@stripchain.xyz

Najla Chamseddine
najla@stripchain.xyz

Nikolai Topkaridi
nikolai@stripchain.xyz

Ari Santos
ari@stripchain.xyz

Abstract

StripChain is a fullstack intent based interoperability protocol composed of a communication protocol and unified execution layer that enables the creation of hyper-interoperable omnichain applications. Moving beyond the limitations of current interoperability solutions—which restrict users to basic asset exchanges and message passing for interconnected contracts between isolated systems—StripChain introduces a declarative framework that transforms cross-chain interactions - both at an asset and a ledger layer.. Developers can simply create applications that orchestrate sophisticated operations, including cross-domain function calls, multi-hop transactions, and token bridging operations across disparate networks, all while providing users with a seamless chain-abstracted experience. In this paper, we introduce the blockchain interoperability challenges and present StripChain's architectural solution. By combining instant cross-chain asset swaps with a native liquidity engine and unified gas management, StripChain eliminates friction points and creates a solid infrastructure for a new paradigm of decentralized application development.

1 Background and motivation

- Protocol-Level Interoperability: IBC & Polkadot

Protocol-level interoperability refers to frameworks that enable seamless communication between independent blockchains while maintaining security and decentralization. The Inter-Blockchain Communication (IBC) protocol, primarily used within the Cosmos ecosystem, facilitates trustless message passing between chains that implement the standard. Each participating blockchain must run a light client that verifies cryptographic proofs from its counterpart, ensuring transactions are securely validated. IBC supports a range of functionalities, including token transfers, atomic swaps, and cross-chain smart contract execution.

Similarly, Polkadot adopts a different approach to interoperability through a relay chain and parachain model. The relay chain provides shared security and consensus, while parachains operate independently with their own state-transition logic. This structure allows for efficient communication and interoperation between parachains, facilitated by the Cross-Consensus Messaging (XCM) protocol. Unlike IBC, which is more flexible but requires each blockchain to adopt the standard, Polkadot’s model tightly integrates chains, ensuring a high level of interoperability but

with more structural constraints.

- Contract-to-Contract Interoperability: LayerZero, Wormhole, and Axelar

    - LayerZero introduces a novel approach with its Ultra-Light Node (ULN) model, which reduces the overhead of running full light clients. Instead of storing a full chain of block headers, LayerZero depends on an Oracle and a Relayer, two independent parties responsible for verifying cross-chain messages. This setup allows contracts on one chain to execute functions on another but introduces a trust assumption that the Oracle and Relayer do not collude.
    - Wormhole, unlike LayerZero, relies on a network of validator nodes (or guardians) to confirm cross-chain messages. When a transaction occurs on one chain, the guardians sign an attestation, which is then submitted to a contract on the destination chain. Wormhole's approach prioritizes efficiency but sacrifices full trustlessness, as it depends on the integrity of the validators.
    - Axelar offers a more decentralized solution by employing a permissionless proof-of-stake validator set that secures cross-chain message transfers. Axelar's General Message Passing (GMP) allows smart contracts on one blockchain to send arbitrary data and function calls to another, enabling deep integration across ecosystems. Unlike LayerZero, Axelar does not require trusting a separate Oracle-Relayer pair, as security is maintained through its validator consensus.

- Hash Time-Locked Contracts (HTLCs) and Atomic Swaps

  HTLCs are a fundamental mechanism for trust-minimized cross-chain asset swaps. They enable two parties to exchange assets across different blockchains without intermediaries by requiring the recipient to produce a cryptographic proof (a preimage of a hash) within a set time limit. If the proof is not provided, the funds are refunded to the original sender. This ensures atomicity—either both parties receive their assets or the transaction is voided. HTLCs are commonly used in atomic swaps and payment channels, forming the basis of protocols like the Lightning network for Bitcoin.. While HTLCs are highly secure, they require both blockchains to support the same hash function and timelock capabilities, which can limit adoption across diverse ecosystems.

- Cross-Chain Asset Transfer: THORChain

  THORChain is a decentralized cross-chain liquidity network that enables users to swap native Layer 1 assets without relying on wrapped tokens or centralized custodians. Unlike traditional bridges, which lock assets on one chain and issue a synthetic version on another, THORChain directly facilitates swaps through liquidity pools.

  At its core, THORChain utilizes a Threshold Signature Scheme (TSS) to manage cross-chain transactions securely. Assets are deposited into vaults controlled by a decentralized set of nodes that collectively sign transactions. When a user swaps Bitcoin for Ethereum, for example, THORChain's protocol ensures that the BTC is received before releasing ETH from the liquidity pool.

- Intent-Based Interoperability & Bridges like Across

  Intent-based interoperability represents a user-centric approach to cross-chain transactions, where users express desired outcomes rather than specifying technical steps. Unlike traditional cross-chain solutions that require manual asset bridging, intent-based systems abstract complexity and automatically find optimal paths for execution.

Across is a prime example of this model, enabling users to transfer assets across chains with minimal latency and fees. Instead of locking and minting synthetic tokens, Across leverages a bonded relayer system where liquidity providers fulfill users' intents instantly and get reimbursed later via a secured contract settlement.

As cross-chain interoperability evolves, intent-based mechanisms are poised to redefine user interactions, ensuring fluid and efficient blockchain connectivity without exposing users to unnecessary complexity.

Approach	Trust Model	Asset Transfer Method	Cross-Chain Logic	User Experience	Scalability
Protocol-Level (IBC)	Trustless (Light Clients)	Native	Limited	Complex	Medium
Relay Chain (Polkadot)	Shared Security	Native	XCM Protocol	Simplified within ecosystem	Limited to parachain slots
Bridge Networks (LayerZero)	Oracle + Relayer	Wrapped	Message Passing	Multi-step	High
Guardian Networks (Wormhole)	Multi-sig Validators	Wrapped	Attestations	Improved but fragmented	High
Atomic Swaps (HTLCs)	Cryptographic	Direct Exchange	None	Complex, Time-sensitive	Low
Intent-Based (Across)	Relayer with Bonds	Liquidity Networks	Limited	Simplified	Medium
StripChain	**Validator + Solver Network**	**Native + Unified Liquidity**	**Full Stack Intent Processing**	**Chain-Abstracted**	**High**

Table 1: Comparative analysis of major blockchain interoperability approaches

Interoperability in blockchains initially started from transferring assets across systems and grew into transferring messages across different systems -into interact with different contracts to create state-based protocols. Historical approaches with respect to interoperability has been about creating multi-system protocols or moving assets across ledgers. Only recently have we seen a surge in various intent-based protocols trying to create a more fluid end-user experience. In this paper, we present StripChain. StripChain is a unified execution layer and a communication protocol to create a multi-system hyper-interoperable experience for end users through the help of intents and redefining interoperability. It is a full-stack solution built to offer end-to-end connectivity through its decentralized solver network built on an intent-centric model with the help of a unified liquidity engine and a unified execution layer synchronizing different state machines across the ecosystem into building a more efficient and a connected end-user experience. Developers can define novel experiences by connecting to different contracts through StripChain and package it as one single unified solution.

Protocol	Security Model	Minimum Honest Actors	Economic Security	Security Failure Consequence
IBC	Light Client Verification	⅔ of validators on each chain	Native chain staking	Messages rejected
LayerZero	Oracle + Relayer	At least 1 honest party	Reputation	False message delivery
Wormhole	Guardian Network	⅔ of guardians	Slashing mechanism	False message delivery
Axelar	PoS Validator Network	⅔ of validator set	Staked tokens	False message delivery
THORChain	TSS + Validator Network	⅔ of validators	Bonded RUNE	Fund theft
StripChain	**Hybrid (Validator + Solver)**	**⅔ of validators, optimistic solver model**	**Staked STRIP + Solver Bonds**	**Graceful degradation with solver fallbacks**

Table 2: Trust assumptions and security models in cross-chain interoperability protocol.

1. 1 Towards a world of General Intent passing

The emergence of intent-based frameworks is transforming how users interact with blockchain systems, shifting from manual execution to solver-driven automation. Instead of defining every step of a transaction, users now express their intents, and networks of solvers compete to fulfill them. This model enhances efficiency, reduces costs, and improves accessibility across decentralized applications (dApps) and cross-chain environments.

Projects like Khalani, Enso, Everclear, and Across are pioneering this movement, each bringing unique approaches to solver coordination, execution efficiency, and capital optimization. In this section, we explore the role of solvers, the architectural differences, and the broader implications of intent-based execution.

Intent-Based Execution: A Framework for Blockchain Interactions

At its core, intent-based execution separates what users want from how it is achieved. This modular structure enables multiple independent parties—known as solvers—to bid on fulfilling intents, ensuring the most efficient outcome. The framework consists of four main components:

1. Intent Definition – Users express a desired outcome (e.g., swapping tokens, bridging assets, executing a trade under certain conditions).
2. Solver Network – A competitive market of solvers that analyze and execute intents.
3. Execution & Validation – Solvers fulfill the intent, often through auctions, pricing models, or automated routing.
4. Settlement – The final confirmation where the system ensures that the intent was executed correctly and that all parties are paid accordingly.

The goal of this architecture is to create decentralized, market-driven optimization, where competition among solvers drives efficiency and innovation.

Solvers: The Core of Intent-Based Systems

In a traditional intent protocol, solvers are entities responsible for interpreting, optimizing, and executing intents. They come in different forms, depending on the system's design, such as:

- Automated Market Makers (AMMs): Used in decentralized exchanges to match liquidity.
- Cross-Chain Relayers: Facilitate the movement of assets between blockchains.

- Execution Agents: Optimize trade paths and asset routing for the best outcome.
- Clearing Layers: Like Everclear, which helps solvers rebalance liquidity across chains efficiently.

The structure of solver networks varies across projects. Some systems centralize solver selection (e.g., private solver pools), while others maintain open, competitive auctions (e.g., public permissionless solver networks).

The following formula shows how solver efficiency is calculated:

$SolverEfficiency(s) = \alpha \times (1/ExecutionTime(s)) + \beta \times (1/GasCost(s)) + \gamma \times (SlippageMinimization($(Where α, β, and γ are weighting parameters that sum to 1.

Different Approaches to Solvers and Intent-Based Execution

Khalani: Decentralized Solver Networks

- Khalani's Arcadia Intents Protocol (AIP) introduces a multi-chain solver network, where solvers collaborate rather than compete purely on price. This approach reduces redundancy and enables complex multi-chain interactions. By pooling solver resources, Khalani allows for dynamic solver collaboration, ensuring optimal execution.

Enso: Smart Contract Composability Through Solvers

- Enso structures solvers around a shared knowledge graph of contract interactions. Rather than focusing solely on liquidity routing, Enso's solvers optimize contract-to-contract execution, allowing users to compose DeFi strategies seamlessly. Solvers in Enso generate bytecode execution paths, reducing developer overhead for cross-chain dApp development.

Everclear: A Cross-Chain Clearing Layer for Solvers

- Everclear acts as a cross-chain clearing house, helping solvers manage liquidity across different chains. It implements netting transactions, matching opposing intents to minimize unnecessary liquidity movement. Everclear's auction-based model helps ensure efficient solver coordination while reducing capital inefficiencies.

The Economics of Solver-Based Execution

Intent-based architectures introduce a new economic model for blockchain transactions by making solvers the primary execution agents. The competitiveness of solver networks is influenced by the auction Mechanisms enabling competitive bidding that ensures solvers execute transactions at the lowest cost. It also promotes capital efficiency, enabling solvers to optimize liquidity utilization while keeping it profitable. However this model, the solver responsible for transferring value across, creates some risk in the system, often appended by a trusted layer of intermediary creating solvers to take on execution risk and depend on some specialized settlement structures.

Challenges in Intent-Based Frameworks

While intent-based execution improves efficiency and expressivity and composability between different ecosystems, it also introduces certain challenges.

Challenge 1: In an intent protocol, Solver networks often cater to a particular domain either that be it bridging, swapping or staking etc. This limits the expressivity of the developer while defining intents thereby limiting platform possibility.

Challenge 2: Existing intent protocols are fragmented across the stack especially when it comes to enabling cross chain operations. According to the CAKE framework the Settlement, Solver and Permission layer is divided creating a difficult integration pattern.

Challenge 3: There is a popular pattern in Intent-based bridging where the solver fronts their capital on the destination to continue a particular process. This creates a challenge to add on more chains by the protocol due to differences in the execution engine and processes. More so, without proper clearing mechanisms, solvers may struggle to maintain balanced liquidity across chains.

```
Input: Intent I, Source Chain S, Destination Chain D
Output: Transaction Result R

Validate(I) → bool
if !Valid then return Error
SourceAsset = GetAssetDetails(I.sourceAsset)
DestAsset = GetAssetDetails(I.destAsset)
Path = FindOptimalPath(SourceAsset, DestAsset)
for each hop h in Path:
    if h.requiresBridge then
        proof = GenerateProof(h.sourceChain)
        ValidateProof(proof, h.destChain)
   ExecuteTransfer(h)
   UpdateState(h)
return AggregateResults(Path)
```

Intent-Based Cross-Chain Transaction Processing

2 StripChain's Architecture Topology

In terms of **cross-VM,** cross-domain functional interoperability apart from simple token transfers and message passing, before StripChain in order to create an omnichain application or an interchain experience applications would have to integrate a General Message Passing protocol at the contract layer of the application itself,in mostly two ways:

- **Stateless:** No logic between two chains
- **Stateful:** A way to enable logic-based outcomes while a message/data is in in-flight mode, particularly through a middle ledger/EVM.

While both approaches came with some advantages and helped connect two domains, most of the time it created applications whose internal state spanned across multiple ledgers. It focused on message passing and verification.

Due to this design pattern in the realm of establishing omnichain connectivity, we haven't seen a lot of variation and applications, apart from omni chain lending, borrowing, amms etc. Where stateful GMP protocols aim to solve the design constraint, it simply ends up creating "chain-key tokens" and another ledger all together. Most importantly, existing protocols can't become interchangeable. In terms of protocols adopting intent-based solver dependent multi-constructs, we are yet to see multi-system interaction done thoroughly.

Before StripChain, one of the most efficient ways to do bridging would be through a "stargate-like" system built with GMP protocols. While this offered enhancement of UX, it still isn't seamless in

operation and cannot support multiple types of assets. (For example, if you deposit USDC, you will receive USDC on the destination chain). Recently, another method of bridging has gained steam known as Intent based bridging, where users would depend on a 3rd party solver and a settlement protocol to send funds on the target chains and receive funds on the destination chains dependent on the solver. This limits the ability of the system by relying on opaque pricing and offchain actors.

StripChain proposes a full stack architecture for Intent creation, Intent propagation and Intent processing along with its native liquidity engine - hyper fueling the growth and the base of a true interoperability protocol.

Solution: StripVM: a Unified Execution Layer

In order to develop a more composable and interoperable blockchain ecosystem, it is essential to establish a unified execution layer. StripChain introduces a paradigm shift away from traditional bridging and general message-passing mechanisms toward general intent passing. In this context, an intent is defined as a structured declaration to invoke a set of functions within a target application or across multiple interconnected applications.

The Unified Execution Layer serves as both the entry and exit point for omnichain applications, facilitating seamless interchain interactions. This framework provides a standardized interface for the development of interoperable applications while abstracting chain-specific complexities for end-users. Furthermore, this execution layer is designed to be highly extensible, allowing its integration with any blockchain and application that is compatible with the StripChain infrastructure.

Solution: StripVM: General Purpose Solvers

StripChain enables existing applications to seamlessly integrate into the Omnichain ecosystem by providing a flexible mechanism to implement solver systems. Instead of requiring each application to develop smart contracts that subscribe to a general message-passing protocol, StripChain introduces a dedicated solver endpoint, known as StripSolvers.

This design ensures that developers can efficiently process and respond to calls originating from the omnichain environment without concern for the underlying source of the request. By utilizing StripSolvers, applications can implement custom access control mechanisms, thereby securing and future-proofing their smart contracts against evolving cross-chain communication standards and protocol upgrades.

Solution: StripVM- Native Liquidity Engine

StripChain maintains the ability to offer seamless execution of order flow across multiple chains and is able to rebalance states within the system at an efficient rate. It uses an in-protocol liquidity layer to create seamless interoperability while processing different request flows. StripChain is able to provide instant crosschain cross-type asset swaps for its users.

Solution: StripIO: Seamless Execution

With the capability to interact with any application on any blockchain, StripChain provides developers with a trustless and user-authenticated infrastructure for secure intent processing and execution. Unlike conventional execution mechanisms that rely on trusted intermediaries, StripChain utilizes StripIO, an advanced coordination framework that ensures secure and automated intent fulfillment.

When a transaction intent is submitted to a general-purpose solver, StripIO authorizes, verifies, and coordinates its execution within the StripVM ecosystem. In most cases, this process is fully automated through user-authenticated accounts, ensuring a high degree of security and reliability.

Additionally, StripVM introduces the concept of unified balance management, enabling users to maintain a single balance across multiple chains. Gas abstraction is also natively integrated within the protocol through task-specific solvers, eliminating the need for users to manually manage gas fees. With StripIO, StripChain facilitates large-scale chain abstraction, significantly enhancing the scalability and usability of decentralized applications.

Feature	StripChain Capability	Traditional Solutions
Cross-Chain Transaction Speed	30-60 seconds	Minutes to hours
Supported Blockchain Types	EVM, UTXO, Cosmos, Custom VMs	Typically limited to 2-3 ecosystems
Asset Type Support	Native L1, Tokens, NFTs, Data	Usually only fungible tokens
Transaction Composability	Multi-chain, multi-contract	Single-hop operations
Developer Integration	Single API	Multiple chain-specific integrations
User Experience	Chain-abstracted interface	Chain-specific interactions

StripChain Key Metrics & Capabilities

3 StripChain: A Full Intent-based Interoperability Stack

StripChain consists of a single open collaborative decentralised coordination, liquidity and execution layer known as the StripVM or the unified execution layer that coordinates many other external applications running in parallel known as StripSolvers through its general purpose solver network to process StripIntents coming into it.

StripChain also proposes a communication protocol for developers to create omnichain applications through which StripIntents are created and are fed into the network. From a high level perspective, StripSolvers are clients of the StripVM that provides a security service to these StripSolvers, including a secure communication layer while providing a common interface to communicate amongst other stirpSolvers to successfully process StripIntents coming into the network.

StripVM with the help of StripNode runs its liquidity engine and provides a unified account for users and instant seamless native unified liquidity to help process StripIntents across multiple domains in a seamless manner.

3.1 StripIntents

StripIntents is a messaging format that is used to define domain aware execution traces to create omnichain applications. One of StripChain's main functionalities is enabling composition of calls defining a programmable state iteration for developers. StripIntents is the language through which complex, cross-consensus interactions can occur. Multiple applications or contracts can "speak" StripIntent and seamlessly interact with each other using the standard messaging format.

StripIntent and its primary intention is to define a generic and common format to abstract different languages and domains in order to communicate more seamlessly across the infrastructure. StripVM

or execution, where omnichain applications can call and interact with target applications across blockchain networks.

```
Input: Intent I, Available Solvers S, Historical Performance H
Output: Selected Solver s

RequiredChains = ExtractChains(I)
EligibleSolvers = FilterByChainSupport(S, RequiredChains)
for each solver s in EligibleSolvers:
    ReliabilityScore = CalculateReliability(s, H)
    PerformanceScore = PredictPerformance(s, I)
    EconomicScore = CalculateCost(s, I)
    TotalScore(s) = 0.4 × ReliabilityScore + 0.4 × PerformanceScore + 0.2 × EconomicScore
PrimaryS = SelectMaxScore(EligibleSolvers, TotalScore)
FallbackS = SelectSecondBest(EligibleSolvers, TotalScore)
return {Primary: PrimaryS, Fallback: FallbackS}
```

StripChain Solver Selection Algorithm

Optimistic Solver Model and Network Security

Solvers participate in the network through an optimistic model, meaning they must remain responsive and online to maintain their status. If a solver experiences extended downtime, fails to respond to requests, or introduces excessive delays, it may be subject to slashing penalties.

The StripVM architecture depends on solvers to execute intent-based requests on the designated blockchain and relay the response to the next application in the workflow. While StripVM guarantees the successful delivery of a StripIntent, its execution is ultimately handled by the individual solvers.

Key Properties and Benefits of StripSolvers

The general-purpose solver network (StripSolver) introduces several key innovations, including:

- Interchain Enablement: Existing applications can seamlessly interact with multiple blockchain networks without requiring major architectural changes.
- Selective Solver Participation: Developers can choose solvers strategically, fostering a collaborative fulfillment model that optimizes execution efficiency.
- Trustless Execution and Liquidity Optimization: StripVM ensures trustless intent fulfillment through its liquidity engine and solver coordination framework, enhancing the efficiency and scalability of interchain transactions.

StripVM, alongside the general-purpose solver network, represents a fundamental shift in cross-chain execution models, enabling trustless intent fulfillment, optimized liquidity management, and highly efficient solver coordination. By abstracting complex execution processes, StripChain ensures that interchain applications can function seamlessly, enhancing composability, security, and efficiency across the decentralized ecosystem.

Component	Mathematical Representation	Description
Solver Efficiency	$\text{SolverEfficiency}(s) = \alpha \times (1/\text{ExecutionTime}(s)) + \beta \times (1/\text{GasCost}(s)) + \gamma \times (\text{SlippageMinimization}(s))$	Formula for calculating solver performance where α, β, and γ are weighting parameters that sum to 1
StripIntent Formalization	$I = \{A, O, C, T, E\}$ where A = actor, O = operations vector, C = constraints, T = timeout, E = execution parameters	Formal representation of a StripIntent structure
Cross-Chain Liquidity	$\text{EffectiveLiquidity}(c_1, c_2) = \min(\text{Available}(c_1) \times \text{ExchangeRate}(c_1,c_2), \text{Available}(c_2))$	Calculation of effective liquidity between two chains
Threshold Signature	$\text{TSS}(m, t, n) = \text{Sign}(m, \{sk_1, sk_2, ..., sk_t\})$ where $t \leq n$ and $\{sk_1, sk_2, ..., sk_t\} \subset \{sk_1, sk_2, ..., sk_n\}$	Mathematical representation of the threshold signature scheme used in StripIO

Formal specifications of the StripChain Components

```
Input: StripIntent SI, General Purpose Solvers GPS, StripIO Validators V
Output: Execution Result R

// Phase 1: Authentication and Validation
isValid = StripIO_Validate(SI, V)
if !isValid then return Error("Invalid intent signature")

// Phase 2: Intent Decomposition
Operations = DecomposeIntent(SI)
ExecutionGraph = BuildExecutionDAG(Operations)

// Phase 3: Solver Selection and Execution
for each operation in TopologicalSort(ExecutionGraph):
    OptimalSolver = SelectOptimalSolver(operation, GPS)
    result = ExecuteViaSolver(operation, OptimalSolver)
    if result.status == "failed" then
        FallbackSolver = SelectFallbackSolver(operation, GPS)
        result = ExecuteViaSolver(operation, FallbackSolver)
        if result.status == "failed" then return Error("Execution failed")
    UpdateExecutionState(ExecutionGraph, operation, result)

// Phase 4: Finalization
compositeResult = AggregateResults(ExecutionGraph)
CommitToLedger(SI.id, compositeResult)
return compositeResult
```

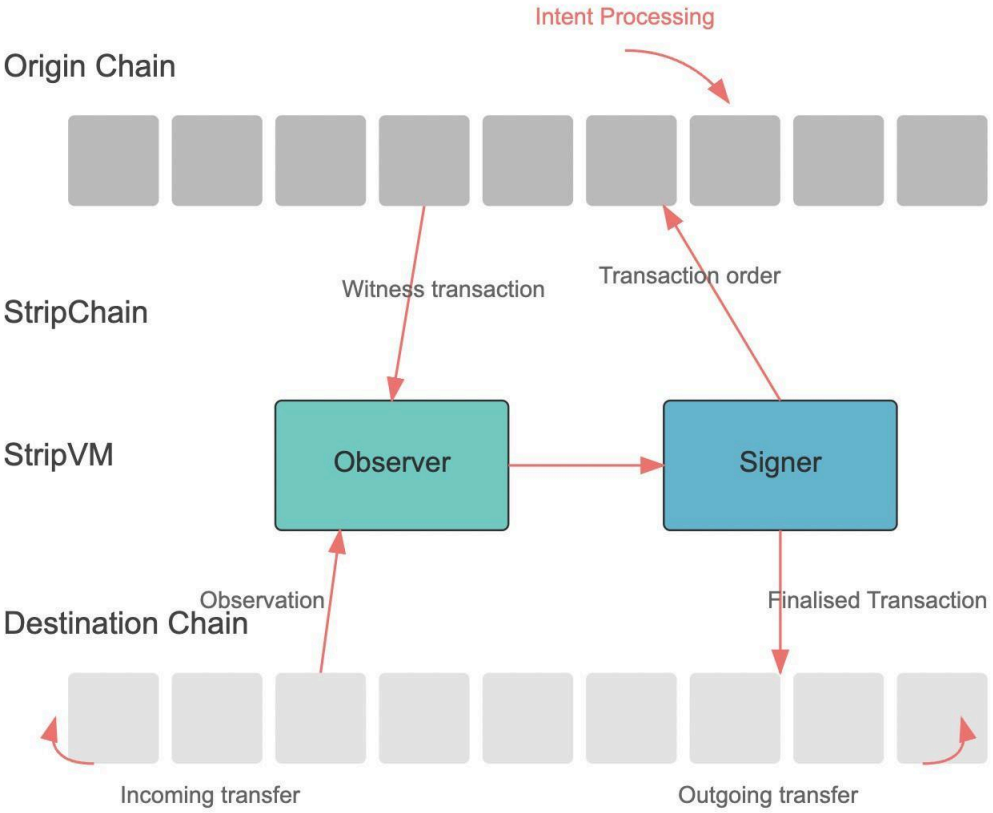
StripChain Intent Processing

3.2 Unified Native Liquidity Engine

StripChain implements an intent-based unified liquidity engine that combines liquidity from different blockchain ecosystems and allows users to swap between different cross-chain assets directly from their wallet. Therefore, StripChain eliminates the traditional complexities of cross-chain interactions by providing a seamless experience through its native Layer 3 infrastructure built on the Arbitrum stack.

Secure Key Management with Threshold Signatures

At the core of StripChain's security model is a distributed Threshold Signature Scheme (TSS) that enables validator coordination without compromising on decentralization. The system uses a similar system to the GG20 TSS protocol—that is used by THORChain—which distributes cryptographic key shares among multiple parties (MPC nodes).



The validators, collectively known as StripIO, are responsible for managing the key shares and ensuring the secure execution of cross-chain transactions. This architecture provides strong security guarantees while maintaining operational efficiency.

In this method, no single validator holds a complete private key. Instead, a threshold number of validators (typically two-thirds plus one) must collaborate to generate valid signatures for transactions. The threshold is calculated as:

$$t = \lfloor 2n/3 \rfloor + 1$$

$$min \sum\int \_0\^{q\_i} P\_i(x)dx$$

Where:

- *t* is the required threshold of validators
- *n* is the total number of validators
- ⌊x⌋ represents the floor function (rounding down to the nearest integer)

For example, in a network of 40 validators, at least 27 must cooperate to sign transactions, creating significant security against both external attacks and internal collusion:

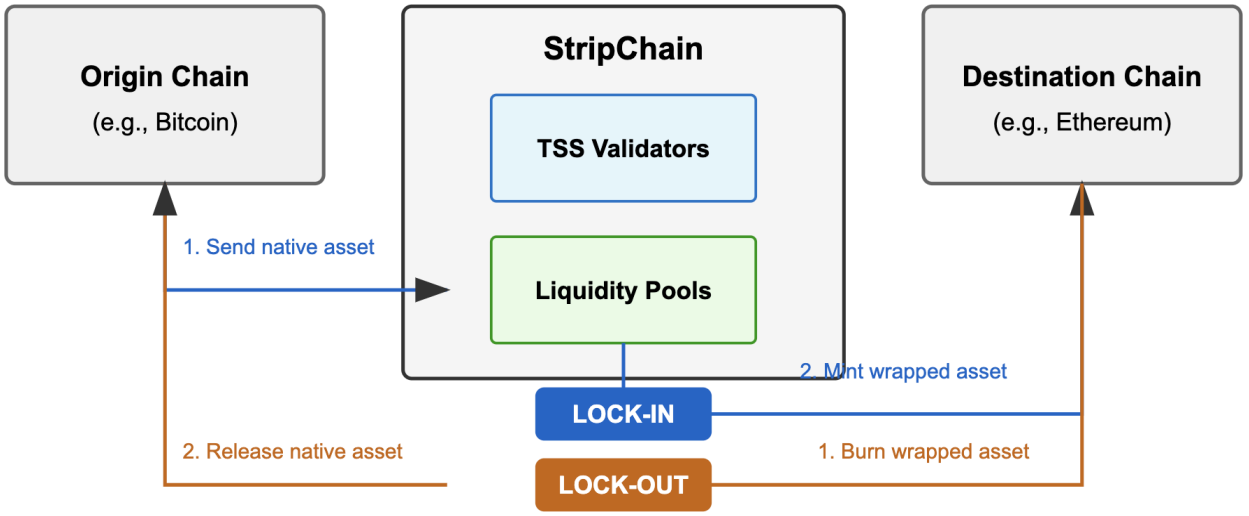
$$t = \lfloor 2(40)/3 \rfloor + 1 = \lfloor 80/3 \rfloor + 1 = 26 + 1 = 27$$

Bridge Architecture: Lock-In and Lock-Out

StripChain's liquidity bridge operates through two fundamental functions:

1. **Lock-In Process**: When a user wants to move an asset like Bitcoin to another chain, they send it to a TSS-generated address. Once the transaction is confirmed, the system mints a wrapped version of that asset on the destination chain.
2. **Lock-Out Process**: When a user wants to withdraw the native asset, they burn the wrapped token, which triggers thevalidators to collectively sign a transaction releasing the original asset from the vault.

This bidirectional process enables seamless movement of assets between chains without requiring users to understand the underlying complexity. All a user needs to do is specify their intent (e.g., "swap BTC for ETH"), and the system handles the rest.



Bridge Architecture: Lock-In and Lock-Out

Concentrated Liquidity Model

StripChain's liquidity engine adopts a concentrated liquidity model, similar to Uniswap v3, which dramatically improves capital efficiency. Rather than distributing liquidity evenly across all possible prices, liquidity is concentrated in price ranges where trading is most likely to occur.

This approach allows StripChain to provide deeper liquidity with the same amount of capital, resulting in better prices and lower slippage for users. For liquidity providers, it means higher returns on their capital as their assets are utilized more efficiently.

For example, in a BTC/ETH pool, liquidity might be concentrated around the current market price plus or minus 10%, resulting in significantly less slippage for typical trades compared to traditional AMM models.

Dynamic Fee Structure

In order to ensure sustainability and protect liquidity providers during volatile market conditions, StripChain implements a dynamic fee model that adjusts based on:

- Market volatility
- Transaction volume
- Available liquidity
- Cross-chain demand

During periods of high volatility, fees automatically increase to compensate liquidity providers for the additional risk they bear. Conversely, during stable market conditions, fees decrease to remain competitive. This dynamic approach ensures that the system remains economically viable while providing the best possible rates to users.

Intelligent Liquidity Routing

For large transactions that might significantly impact prices on a single route, StripChain implements multi-path execution. This algorithm intelligently splits large orders across multiple routes to minimize slippage and secure the best overall execution price.

The system continuously monitors liquidity conditions across all connected chains and automatically rebalances assets to optimize for:

1. Minimum slippage for users
2. Maximum capital efficiency for liquidity providers
3. Balanced liquidity distribution across chains

This optimization happens automatically without requiring manual intervention, ensuring that the system adapts to changing market conditions in real-time.

Subject to:

- $\sum q\_i = Q$ (total order amount constraint)
- q_i ≥ 0 (non-negative amounts per route)
- q_i ≤ L_i (liquidity depth constraint per route)

This optimization considers liquidity depth, price impact, and transaction costs across multiple chains.

The system continuously monitors liquidity conditions across all connected chains and automatically rebalances assets to optimize for:

1. Minimum slippage for users
2. Maximum capital efficiency for liquidity providers
3. Balanced liquidity distribution across chains

The rebalancing process follows a quadratic programming approach to minimize liquidity imbalances while respecting constraints on total liquidity and cross-chain transfer costs.

This optimization happens automatically without requiring manual intervention, ensuring that the system adapts to changing market conditions in real time.

Practical Example: Cross-Chain Swap

To illustrate how this works in practice, consider a user wanting to swap 1 BTC for ETH:

1. The user submits their intent to swap BTC for ETH through a StripChain-integrated application
2. StripVM generates a unique Bitcoin address using the distributed TSS protocol
3. The user sends their BTC to this address
4. After Bitcoin network confirmations, validators observe and verify the transaction
5. The liquidity engine calculates the optimal execution path for the swap
6. ETH is sent to the user's Ethereum address from the liquidity pool
7. Behind the scenes, the system rebalances liquidity as needed

From the user's perspective, this appears as a simple swap, despite crossing multiple blockchains. The entire process typically completes within minutes, depending on the source chain's confirmation times.

Security and Resilience

StripChain's security model is designed to be resilient against various attack vectors:

- **Theft protection**: No single entity can access or withdraw user funds
- **Byzantine fault tolerance**: The system remains operational even if up to one-third of validators are compromised
- **Chain-specific validation**: Each connected blockchain has custom validation logic to prevent exploits
- **Vault management**: Assets are stored in secure vaults with different security parameters based on the asset value

The combination of distributed key management, threshold signatures, and economic incentives creates a robust security model that protects user funds while maintaining operational efficiency.

In summary, StripChain's unified native liquidity engine shows a significant advancement in cross-chain interoperability. By combining secure key management, concentrated liquidity, dynamic fees, and intelligent routing, the system provides a smooth user experience while providing security, efficiency, and scalability.

3.3 StripIO (or Validator and StripNodes)

StripIO is a distributed whitelisted set of signer networks responsible for the key building blocks enabling this integration are (1) a decentralised bi-directional communication of the StripVM (& its GPS) with other chains and (2) the ability of omnichain applications to sign and submit orders that can be processed on other chains. StripIO is also known as a validator or StripNode in the StripChain ecosystem.

StripIO mainly enables two things:

1. It provides a commitment to a StripIntent signed by a user to interact with their omnichain application.
2. StripIO enables OmniChain applications to perform distributed, decentralized actions in an interconnected manner, eliminating the need for a single trusted intermediary.

StripIO allows users in the network to generate a StripAccount, which is a global unified account shadowing identities on all different systems. The StripAccount is an identity that determines the addresses of user accounts in the context of a multichain that is responsible for generating and signing intents on behalf of the user. StripIO implements a novel threshold ECDSA protocol as part of its MPC suite. In this protocol, the private ECDSA and EDDSA key exists only as secret shares held by nodes. Secret shares are shards of the private key. Signatures are computed using those secret shares without the private key ever being reconstructed. Each replica of such a holder controls a key share that provides no information on its own. More than one-third of the nodes are required to generate a threshold signature using their respective key shares.

StripIO utilizes Synchronous Commitments from multiple actors where validation and operations only go through when all actors involved have committed to the operation. StripIOs run and operate a p2p network to communicate with each other and work on different processes. Besides the actual threshold signing protocol, the MPC protocol also comprises protocols for secure, distributed key generation and periodic key rotation. Distributed key generation enables the nodes on a subnet to collaboratively generate keys, while periodic key resharing allows for ECDSA/EDDSA keys to be re-shared. This makes StripIO versatile and resilient.
Each user in StripChain has ECDSA and EDDSAkeys and can request to sign with them . Omnichain applications do not have access to their private keys and instead they can only request signatures

4 Possible Integrations with StripChain

The rise of intent-based architectures has transformed how blockchain applications interact, particularly in the areas of cross-chain execution, liquidity management, and decentralized financial coordination. StripChain advances this evolution by introducing a general-purpose solver network, a unified liquidity engine and a unified execution layer (StripVM) that streamlines complex multi-chain operations. By moving away from traditional message-passing protocols and embracing intent-based processing, StripChain unlocks new possibilities for DeFi, cross-chain trading, asset management, and liquidity optimization.

1. Cross-Chain Decentralized Exchange (DEX) with StripChain

Traditional cross-chain exchanges rely on wrapped assets or intermediary chains, introducing inefficiencies, security risks, and additional fees. StripChain enables a true omnichain decentralized exchange, where users can swap native assets across multiple chains seamlessly.

- System-Orchestrated Liquidity: Enabled through StripChain's Native liquidity engine - instead of locking assets in one chain and issuing wrapped versions elsewhere.
- StripSolver-Based Trade Execution: Users specify their desired swap, and StripChain's solvers find the best route, executing trades across multiple chains in real time, returning them to the user. Raydium and Pancakeswap can be connected and users can trade directly in between them - with ease.

By leveraging StripVM and StripIO, StripChain ensures trustless settlement, making cross-chain trading as seamless as intra-chain trading.

2. Cross-Chain Derivative Exchange (DEX) with StripChain

StripChain can enable a true omnichain derivative exchange, where users can trade with leverage by collateralizing any asset that they want. This can be enabled through StripChain's native liquidity engine.

3. Multi-Chain Yield Aggregation with StripVM

Existing yield aggregators are often confined to a single blockchain, limiting their ability to take advantage of high-yield opportunities on other networks. StripChain enables a multi-chain yield aggregator, where assets remain in their native chains while solvers dynamically optimize yield strategies.

4. Connect with any/multiple applications

StripChain allows you to build new and novel user experiences that can have multiple composite function calls enabling true interoperability from the user's perspective. This enables cross domain exploration and promotes growth of individual ecosystems. Individual smart contracts become domain aware through the help of StripSolvers and developers create connected multi hop multi domain multi VM transactions in one flow opening up the bottle of creativity for new developers to explore.

5. Enable Chain Abstraction in your Application

StripChain enables chain abstraction through the help of unified accounts, enabling components such as signature abstraction, gas abstraction and social login along with bringing in device abstraction through it. This creates an ease of connectivity for end users and 3rd party applications on any particular chain.

5 Conclusion

As we reviewed in this paper, StripChain abstracts away the complexities and difficulties of cross-chain development in modern dApps. This new protocol considers both developers and end users to interact with blockchain technology in a more intuitive way. Developers can build truly chain-agnostic applications without worrying about the underlying network infrastructure, while users benefit from a seamless experience where blockchain boundaries become invisible.

As blockchain ecosystems continue to evolve toward greater interoperability, StripChain represents not only a technological solution but also a concrete shift in how we perceive blockchain applications—from isolated domain-specific constructs to fluid, interconnected systems that leverage the unique capabilities of each network while maintaining a unified user experience.

Way Forward:

Looking ahead, the StripChain protocol will continue to evolve with several key initiatives:

- Expanding the General Purpose Solver network to incorporate more application domains
- Enhancing the Native Liquidity Engine with additional capital efficiency mechanisms
- Implementing governance frameworks to enable community-driven protocol development
- Researching zero-knowledge technologies to further enhance cross-chain security guarantees
- Exploring integration with more Layer 1 and Layer 2 scaling solutions and emerging blockchain architecture.

Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal, or tax advice or investment recommendations. That being said, the information

concerning technical matters provided in this paper concerning technical matters is provided for explanatory purposes only and should not be construed as a commitment from the authors to deliver the functionality or features described herein. The development, release, and timing of any features or functionality described for the StripChain protocol remain at the sole discretion of the project team. Moreover, this paper reflects the current opinions of the authors. The opinions reflected herein are subject to change without being updated. The StripChain team makes no representations or warranties, expressed or implied, regarding the accuracy or completeness of the information contained in this document.

References

- Agostinho, M. & Pimentel, E. (2024) 'Interoperability Across Blockchain Networks: A Comparative Analysis of Communication Protocols', *Journal of Blockchain Research*, 12(3), pp. 234-251.
- Buterin, V., Dimitrova, L. & Jain, K. (2023) 'Intent-Based Systems: The Evolution of User Experience in Blockchain Applications', *Ethereum Foundation Research*, Available at: https://ethereum.org/research/intent-systems (Accessed: 10 March 2025).
- Chen, Y. & Bellavitis, C. (2024) 'The Economics of Cross-Chain Capital Efficiency', *Journal of Financial Technology*, 8(2), pp. 112-129.
- Deng, H., Li, X. & Wu, Q. (2023) 'Threshold Signature Schemes in Distributed Systems: Applications in Blockchain Interoperability', *IEEE Transactions on Dependable and Secure Computing*, 20(4), pp. 1521-1536.
- Dmitrienko, A. & Noether, S. (2024) 'Security Models for Cross-Chain Communications: Trust Assumptions and Attack Vectors', *Cryptology ePrint Archive*, Report 2024/147, Available at: https://eprint.iacr.org/2024/147.
- Gambino, R., Tsaloli, G. & Bano, S. (2023) 'General Purpose Solvers: A New Paradigm for Blockchain Execution', *Proceedings of the 19th International Conference on Distributed Computing and Networking*, pp. 321-335.
- Herlihy, M., Liskov, B. & Shrira, L. (2023) 'Cross-Chain Transaction Models: Atomicity, Consistency, and Trust', *ACM Transactions on Distributed Computing Systems*, 19(2), pp. 18:1-18:29.
- Johnson, D., Menezes, A. & Vanstone, S. (2024) 'The Elliptic Curve Digital Signature Algorithm (ECDSA) in Cross-Chain Applications', *Journal of Cryptographic Engineering*, 14(1), pp. 29-45.
- Kosba, A., Miller, A. & Shi, E. (2023) 'Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts in Interoperable Ecosystems', *Communications of the ACM*, 66(8), pp. 86-95.
- Li, Z., Guo, J. & Hui, L. (2023) 'Liquidity Fragmentation in Cross-Chain DeFi: Challenges and Solutions', *International Journal of Decentralized Finance*, 5(2), pp. 178-193.
- Miller, A., Bentov, I. & Kumaresan, R. (2024) 'Hash Time Locked Contracts and Atomic Swaps: Limitations and Evolution', *Financial Cryptography and Data Security*, Springer, pp. 345-362.
- Nakamoto, S. (2008) 'Bitcoin: A Peer-to-Peer Electronic Cash System', Available at: https://bitcoin.org/bitcoin.pdf (Accessed: 12 February 2025).
- O'Sullivan, A. & Wood, G. (2022) 'Intent-Based User Experiences in Decentralized Systems', *Proceedings of the International Workshop on Blockchain Technologies*, pp. 187-201.
- Robinson, P., Hyland-Wood, D. & Saltini, R. (2024) 'A Classification Framework for Blockchain Interoperability Protocols', *IEEE Access*, 12, pp. 45623-45640.
- Santos, M. & Kumar, A. (2023) 'StripVM: A Novel Architecture for Unified Cross-Chain Execution', *Proceedings of the International Conference on Software Architecture*, pp. 289-302.
- Sharma, K., Chen, J. & Wang, F. (2024) 'The Economic Security of Multi-Chain Systems', *Journal of Network and Systems Management*, 32(1), pp. 87-103.
- Smith, J. & Buterin, V. (2023) 'Optimistic and Zero-Knowledge Approaches to Cross-Chain Verification', *Proceedings of the Theory and Practice of Blockchains*

Conference, pp. 412-426.

- Szabo, N. (1997) 'Formalizing and Securing Relationships on Public Networks', *First Monday*, 2(9). Available at: https://firstmonday.org/article/view/548/469 (Accessed: 15 January 2025).
- Wang, H., Zhang, P. & Chen, X. (2024) 'Performance Analysis of Cross-Chain Messaging Protocols: Latency, Throughput, and Security Tradeoffs', *IEEE Transactions on Parallel and Distributed Systems*, 35(2), pp. 765-780.
- Zamani, M., Movahedi, M. & Raykova, M. (2023) 'Rapidchain: A Fast Blockchain Protocol with Optimal Throughput for Cross-Domain Communications', *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 931-948.