#16 Cross-Chain Bridges

Lecture Notes for CS190N: Blockchain Technologies and Security November 26, 2025

This lecture examines the interoperability trilemma and the fragmented state of digital assets. We begin by explaining how specialized blockchains create isolated state machines that cannot natively observe one another, which leaves value and liquidity siloed across many ledgers. We then introduce bridges as the messaging and asset transport layer of the blockchain stack and use the interoperability trilemma to frame inherent trade offs among trustlessness, extensibility to heterogeneous chains, and support for general message passing. Building on this framework, we develop a taxonomy of bridge designs based on who verifies foreign state: externally verified trusted bridges with lock and mint models, natively verified light client bridges, and trust minimized liquidity networks. The second half of the lecture studies two concrete systems, Wormhole and Thorchain, to make these abstractions tangible. Through their architectures, major exploits, and 2025 redesigns, we highlight how different verification and security assumptions propagate to user risk, capital efficiency, and the long term path from trusted interoperability toward cryptographic trustlessness.

1 THE INTEROPERABILITY TRILEMMA AND THE FRAGMENTED STATE OF DIGITAL ASSETS

1.1 The Siloed Nature of Distributed Ledgers

The evolution of the blockchain ecosystem over the last decade has been defined by a paradox of success: the proliferation of specialized execution environments has created a deeply fragmented landscape of isolated state machines. In our previous lectures on Decentralized Finance (DeFi), we examined primitives such as Automated Market Makers (AMMs) and Lending Protocols within the context of a single synchronous environment, typically Ethereum. Within that singular state machine, composability is trivial; a transaction can atomically interact with MakerDAO, Uniswap, and Aave in a single block, creating the “Money Legos” effect that defines modern DeFi.

However, this composability shatters at the boundaries of the chain. The fundamental architecture of a blockchain is introspective; the Ethereum Virtual Machine (EVM) is aware only of its own state trie. It possesses no native opcode to query the balance of a Bitcoin wallet, the state of a Solana account, or the finality of a transaction on an Arbitrum rollup. This isolation is a feature, not a bug, it is the mechanism that ensures security and consensus stability without external dependencies. Yet, for a global internet of value to function, these distinct networks must communicate.

This necessity has given rise to the **Cross-Chain Bridge**, a cryptoeconomic primitive designed to transport arbitrary data (messages) and value (assets) between disparate consensus domains. As of late 2025, bridges facilitate billions of dollars in daily volume and secure tens of billions in Total Value Locked (TVL) [9]. They are the TCP/IP layer of the blockchain stack, yet they remain the most fragile and perilous component of the infrastructure. The history of bridges is written in the wreckage of catastrophic failures; between 2021 and 2025, over $2 billion was lost to bridge exploits, including the Ronin Bridge hack ($625 million) [7], the Poly Network hack ($611 million) [10], and the Wormhole hack ($326 million) [8].

1.2 The Interoperability Trilemma

To understand why bridges fail, we must first understand the theoretical constraints under which they operate. Just as the classic Blockchain Trilemma posits trade-offs between Decentralization, Security, and Scalability, the **Interoperability Trilemma** (Figure 1), often discussed by Vitalik Buterin [3], suggests that a bridge protocol can typically optimize for only two of the following three properties at any given time:

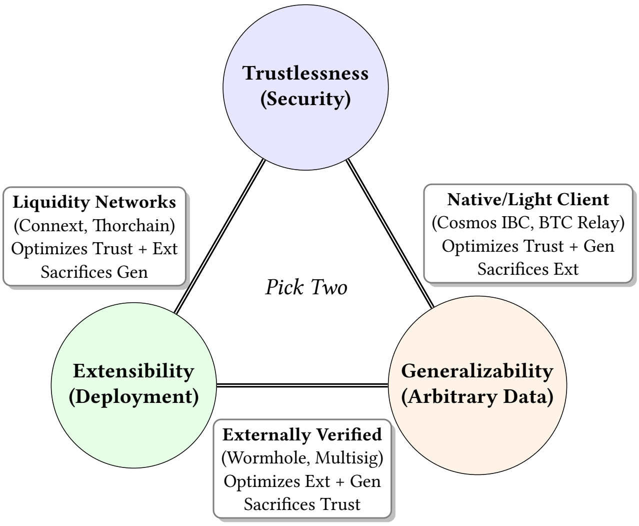

Figure 1. The Interoperability Trilemma: Constraints on Cross-Chain Design

(1) **Trustlessness:** The security of the bridge relies solely on the security of the underlying chains being connected. No additional trust assumptions (e.g., honest majority of external validators) are introduced.
(2) **Extensibility:** The bridge can easily be deployed across many heterogeneous domains (e.g., connecting EVM chains to non-EVM chains like Solana, Bitcoin, or Move-based networks).
(3) **Generalizability:** The bridge can handle arbitrary data passing (e.g., cross-chain governance votes, oracle data, contract calls) rather than just simple token transfers.

Historically, bridge designers have been forced to choose. **Natively Verified** bridges (like Cosmos IBC) optimize for Trustlessness and Generalizability but fail at Extensibility; running a light client of Ethereum on the Solana blockchain is computationally prohibitive due to gas costs and instruction limits. Conversely, **Externally Verified** bridges (like Wormhole, Ronin, and Multichain—all of which suffered massive exploits) optimize for Extensibility and Generalizability by introducing a set of trusted validators. This makes them easy to deploy anywhere but sacrifices Trustlessness, a trade-off that has led to some of the largest exploits in crypto history.

In this lecture, we will deconstruct these architectures through two primary case studies representing opposing philosophies: **Wormhole**, the archetype of the Trusted (Externally Verified) model, and **Thorchain**, the pioneer of the Trust-Minimized (Economic Security) model. We will examine their mechanisms, their failure modes, and their evolution into the landscape of 2025.

2 TAXONOMY OF CROSS-CHAIN VERIFICATION

Before dissecting specific protocols, we must establish a rigorous taxonomy for how cross-chain state is verified. When Chain B accepts a message from Chain A, who is vouching for the validity of that message?

2.1 Trusted Bridges (Externally Verified)

In this architecture, a third party, distinct from the validator sets of either the source or destination chain, is responsible for attesting to the state.

- **Mechanism:** A user deposits funds into a smart contract on the source chain. An off-chain network of validators (often called Guardians, Relayers, or Oracles) observes this event. Once a threshold of validators reaches consensus (e.g., 13 out of 19), they sign a cryptographic message. The user takes this signature to the destination chain, where a smart contract verifies the validators' signatures (not the source chain's state) and releases the funds.
- **Pros:** Extremely cheap and fast. Easy to implement on any chain that supports basic signature verification.
- **Cons:** Introduces a massive central point of failure. If the external validator set is compromised or colludes, they can sign invalid messages and steal all funds locked in the bridge.
- **Examples:** Wormhole, Ronin Bridge, Multichain.

2.2 Trustless Bridges (Natively Verified)

In this architecture, the destination chain runs a "light client" of the source chain within its own smart contract logic.

- **Mechanism:** The destination chain receives block headers and Merkle proofs from the source chain. It mathematically verifies that the source chain's consensus rules were followed. For example, an Ethereum contract would verify the Proof-of-Work (or Proof-of-Stake) finality of a Bitcoin block header.
- **Pros:** Ideally trustless; security is equal to the underlying chains.
- **Cons:** Computationally expensive and technically difficult. Verifying a ZK-SNARK or a complex consensus algorithm inside a constrained environment (like the EVM) often costs prohibitive amounts of gas.
- **Examples:** Cosmos IBC [5], Near Rainbow Bridge, various ZK-Bridge pilots.

2.3 Trust-Minimized Bridges (Locally Verified / Liquidity Networks)

This architecture abandons the idea of "moving" assets via locking and minting wrapped tokens. Instead, it relies on a peer-to-peer swap model.

- **Mechanism:** A system of liquidity pools exists on both chains. A user gives asset A to a liquidity provider on the source chain, and the liquidity provider gives asset B to the user on the destination chain. The protocol ensures atomicity using cryptographic locks (Hash Time-Locked Contracts) or economic incentives.
- **Pros:** No "wrapped" assets (IOUs) are created, eliminating de-pegging risk.
- **Cons:** Capital inefficient. Requires massive liquidity on every connected chain to function.
- **Examples:** Thorchain, Connext, Hop Protocol.

3 CASE STUDY I: WORMHOLE (TRUSTED ARCHITECTURE)

Wormhole is a generic message-passing protocol that has served as a primary artery for cross-chain liquidity, particularly between the Ethereum and Solana ecosystems. It is classified as an **Externally Verified, Lock-and-Mint** protocol (Figure 2). While it has evolved significantly by late 2025, its core architecture remains the definitive example of how trusted bridging works at scale.

3.1 The Guardian Network Topology

Wormhole does not rely on the destination chain verifying the source chain's consensus. Instead, it relies on a Proof-of-Authority (PoA) network known as the **Guardian Network**.

The Guardian Network consists of 19 members. These are not anonymous nodes but high-profile, institutional staking providers and validator companies with significant reputational capital. As of

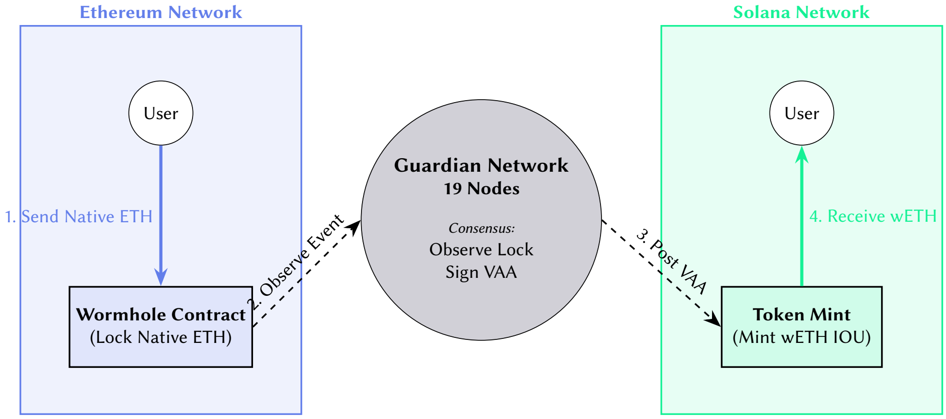

Figure 2. Wormhole Model: Lock-and-Mint (Creation of IOUs)

the 2024-2025 epoch, the set included entities such as **Jump Crypto, Everstake, Chorus One, Figment, and xLabs** [4].

The security model is a simple multisignature threshold scheme.

- **Total Guardians:** 19
- **Consensus Threshold:** 13 (Supermajority, or 2/3 + 1)
- **Assumption:** The protocol assumes that fewer than 13 of these major institutional entities will collude to forge a message. If 13 Guardians agree that an event happened on Ethereum, the Wormhole contracts on Solana (and all other chains) accept it as absolute truth.

This design decision prioritizes **liveness and extensibility**. Because the Guardians run full nodes for every connected chain off-chain, they can easily observe events on high-throughput chains like Solana or Aptos and simply sign a digest. The destination chain only needs to verify 13 ECDSA (or EdDSA) signatures, which is computationally cheap compared to verifying a full light client proof.

3.2 The Lifecycle of a Wormhole Message

The process of bridging assets involves a coordinated dance between on-chain contracts and off-chain agents (Figure 3). Let us trace a transfer of 100 USDC from Ethereum to Solana:

(1) **Emission (Source Chain):** The user calls the `transfer()` function on the Wormhole Core Contract on Ethereum. The contract locks the 100 USDC in a vault and emits an event (log) containing the transfer details (recipient, amount, etc.).
(2) **Observation (Off-Chain):** The 19 Guardians are constantly scanning the logs of all connected chains. They detect the event on Ethereum.
(3) **Verification & Signing (Off-Chain):** Each Guardian independently waits for the specified `consistency_level` (e.g., waiting for Ethereum finality to ensure the transaction cannot be reorged). Once satisfied, the Guardian signs a hash of the message using their private key.
(4) **Aggregation:** A specialized background service collects these signatures. Once it has gathered 13 valid signatures, it packages them into a **VAA (Verifiable Action Approval)**. Essentially, a VAA is a standardized signed message that acts as proof that the Guardians have reached consensus on the event.

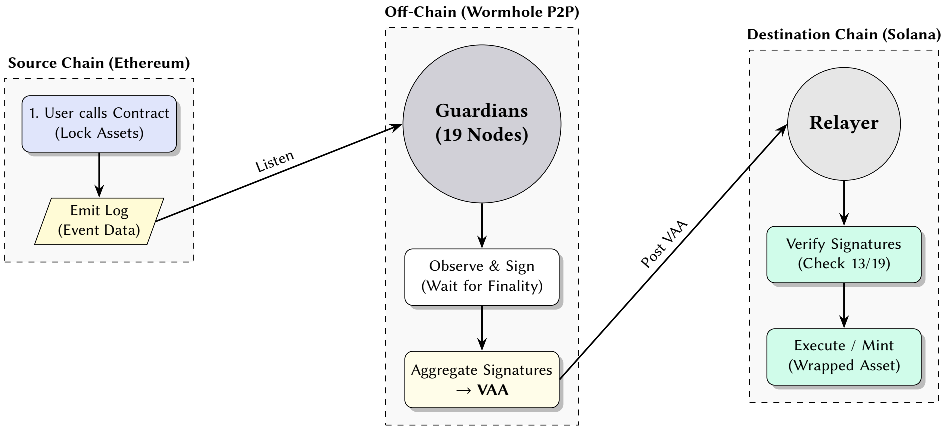
Source Chain (Ethereum)
1. User calls Contract (Lock Assets)
Emit Log (Event Data)
Listen
Off-Chain (Wormhole P2P)
Guardians (19 Nodes)
Observe & Sign (Wait for Finality)
Aggregate Signatures → VAA
Post VAA
Destination Chain (Solana)
Relayer
Verify Signatures (Check 13/19)
Execute / Mint (Wrapped Asset)

Figure 3. Wormhole Architecture: From Emission to Execution

(5) **Relaying (The "Postman"):** Wormhole is a **lazy** protocol; the Guardians do not pay gas to deliver the message. The VAA is posted to a public endpoint (often an API or a decentralized store). A "Relayer" (which can be the user's frontend or a specialized service) retrieves the VAA and submits a transaction to the Solana blockchain.

(6) **Execution (Destination Chain):** The Wormhole Core Contract on Solana receives the VAA. It performs a check: `verify_signatures(VAA)`. It retrieves the stored public keys for the active Guardian Set and validates the 13 signatures. If valid, the contract executes the instruction, in this case, minting 100 "Wormhole-Wrapped USDC" to the user's Solana wallet.

3.3 The Vulnerability of Trust: The 2022 Hack

The reliance on smart contract verification logic introduces a massive attack surface. In February 2022, Wormhole suffered a **$326 million exploit**, providing a textbook example of how subtle implementation bugs in trusted bridges can lead to catastrophic failure.

3.3.1 Technical Root Cause: The `sysvar` *Bypass.* The exploit occurred on the Solana side of the bridge. To understand the mechanics without deep knowledge of Solana's account model, consider the **"Fake Police Officer" Analogy** illustrated in Figure 4.

Imagine a high-security bank vault (the Wormhole Contract) that requires a specific government official (the `sysvar` account) to certify that a signature is valid before releasing funds.

- **Intended Design:** The user brings a signature. The bank asks the official: "Is this signature valid?" The bank *must* also check the official's badge number to ensure they are the real government representative (the real `Sysvar` address).
- **The Flaw:** The Wormhole contract checked the answer ("Yes, valid") but **failed to verify the official's badge number**.
- **The Attack:** The hacker hired a friend to dress up as a police officer (created a **Fake Account** with arbitrary data). This fake officer said "The signature is valid!" The bank, failing to verify the address of the account, trusted the fake official and minted $326M worth of funds [8].

Technical Specifics: In Solana, developers use a specialized system program called the **Instruction Sysvar** to verify cryptographic signatures cheaply. The Wormhole contract contained a deprecated function call: `load_current_index`. Crucially, this function did not enforce that the

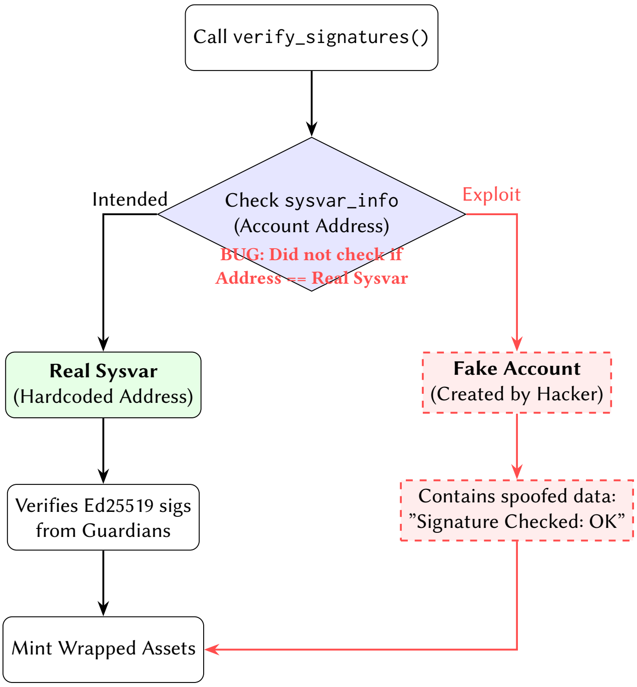

Figure 4. Anatomy of the 2022 Hack: The Sysvar Bypass

account passed to it was the official System Sysvar account. The hacker simply passed in a regular account they controlled, populated with data that said "Signature Verification Passed," and the contract accepted it.

The Exploit Flow:

(1) **Spoofing:** The attacker created a generic Solana account and populated it with data that *looked* like a Sysvar account. This fake account contained data saying "Signature Verification Passed."
(2) **Injection:** The attacker called the `verify_signatures` function on the Wormhole contract but injected their *fake* Sysvar account address instead of the real one.
(3) **Bypass:** The contract read the fake account, saw the "Success" flag, and assumed the signatures were valid. It effectively skipped the Guardian signature check entirely (Figure 5).
(4) **Minting:** The attacker used this bypass to mint 120,000 Wormhole-Wrapped ETH (WeETH) on Solana without locking any real ETH on Ethereum.
(5) **Extraction:** The attacker swapped the fake WeETH for SOL and other assets, then bridged them back to Ethereum, draining the legitimate ETH vault. The funds were subsequently laundered via Tornado Cash.

3.3.2 The Fallout and Bailout. The hack left Wormhole in a state of insolvency; the 120,000 ETH locked on Ethereum were gone, meaning the WeETH circulating on Solana was unbacked. In a centralized financial system, this would mean bankruptcy. However, in a dramatic turn of events, **Jump Crypto** (the parent company of the Guardians and a major stakeholder in the Solana ecosystem) intervened. They deposited 120,000 of their own ETH into the bridge to recapitalize it, effectively bailing out the protocol [2].

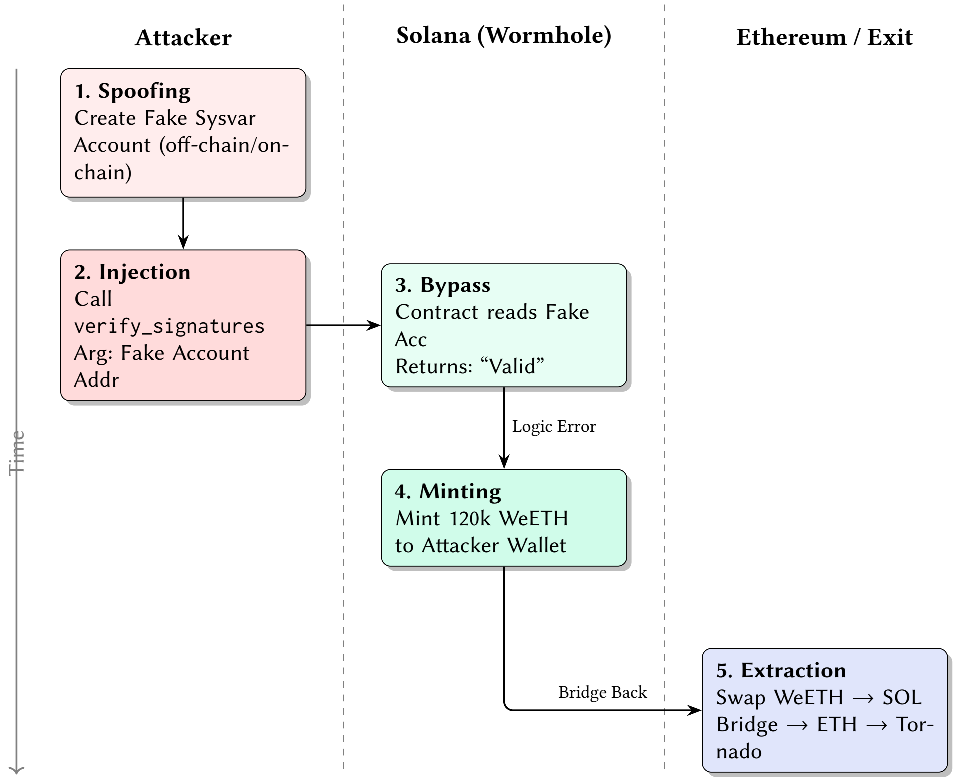

Figure 5. The 5-Step Exploit Flow of the 2022 Wormhole Hack

This event highlighted the **Centralization Risk** inherent in the Trusted model. The bridge survived not because of cryptographic resilience, but because a wealthy centralized entity decided to save it to protect their ecosystem investments.

3.4 Evolution: The ZK Roadmap (2025)

In response to the inherent risks of the Guardian model, Wormhole has aggressively pivoted toward trust-minimization in its 2025 roadmap. The protocol is transitioning from a purely Proof-of-Authority model to a **Zero-Knowledge (ZK) Verified** model [12].

- **The ZK Light Client:** Instead of 13 Guardians signing a message saying “I saw this,” the system generates a ZK-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge). This proof mathematically attests that the source chain’s consensus (e.g., the Ethereum Validator set) finalized a specific block header.
- **Trustless Verification:** The destination chain verifies the ZK proof. This reduces the trust assumption from “Trust Jump Crypto and Everstake” to “Trust the ZK Circuit logic and the underlying Ethereum consensus.”
- **Strategic Partnership:** Wormhole has partnered with hardware accelerators and ZK proving networks (like Succinct and RISC Zero) to handle the immense computational load of generating these proofs.

4 CASE STUDY II: THORCHAIN (TRUST-MINIMIZED ARCHITECTURE)

While Wormhole focused on connecting every chain through wrapped assets, Thorchain (The Open Root Chain) emerged with a radically different philosophy. It is not a bridge in the traditional sense but a **Decentralized Liquidity Network** built on the Cosmos SDK. Its goal is to facilitate **Native Asset Swaps** without ever creating a wrapped token (Figure 6).

4.1 The "No Wrapped Assets" Philosophy

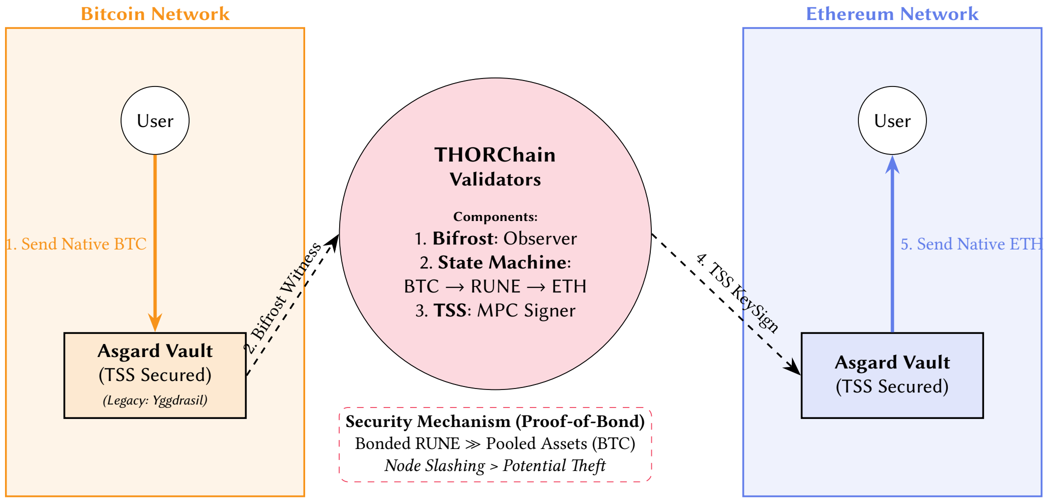

Figure 6. Thorchain Model: Native Asset Swap (Detailed Architecture)

In the Wormhole model, if you want Bitcoin on Ethereum, you get a token called `Wormhole-BTC`. In the Thorchain model, if you want to swap Bitcoin for Ethereum, you send real BTC to a vault and receive real ETH from a vault. The protocol manages wallets on all connected chains.

This architecture is fundamentally an Automated Market Maker (AMM) where every asset is paired with Thorchain's native token, **RUNE**. To swap BTC for ETH, the system executes two swaps atomically:

(1) **BTC → RUNE**
(2) **RUNE → ETH**

4.2 Infrastructure: Bifrost and TSS Vaults

Thorchain's architecture is composed of nodes (THORNodes) that run a specialized bridge module called the **Bifrost Protocol**.

4.2.1 The Bifrost.

4.2.1 The Bifrost. The Bifrost is the "eyes and hands" of the node. It connects to the external networks (Bitcoin, Ethereum, Dogecoin, etc.).

- **Observation:** The Bifrost scans the external blockchains for transactions sent to the Thorchain vaults. When a node sees a transaction, it reports it to the Thorchain consensus as a "witness" transaction.
- **Consensus:** Once a supermajority (67%) of nodes agree they saw the same transaction, the state is updated on Thorchain.

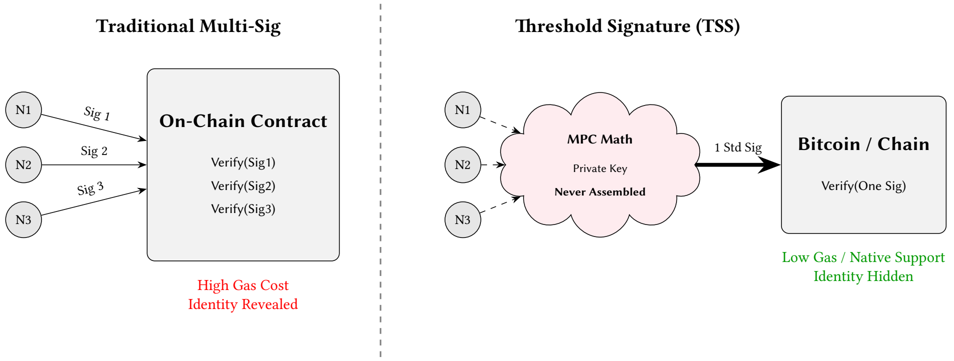

Figure 7. Comparison: On-Chain Multi-Sig vs. Off-Chain TSS

4.2.2 Threshold Signature Schemes (TSS). The most critical innovation of Thorchain is its use of **Threshold Signature Schemes (TSS)**, specifically the Gennaro-Goldfeder 2020 (GG20) MPC protocol [6], as compared to Multi-Sig in Figure 7.

Unlike a Multi-Sig wallet (used by Wormhole), where 13 distinct signatures appear on-chain, TSS uses **Multi-Party Computation (MPC)** to generate a single standard signature.

- **Key Generation (KeyGen):** When a vault is created, the nodes collaborate to generate a private key. Crucially, **the private key never exists in whole form anywhere**. Each node holds only a mathematical "share" of the key.
- **Key Signing (KeySign):** To spend funds from the vault, the nodes engage in a cryptographic ceremony. They combine their shares to produce a valid signature for the Bitcoin or Ethereum network. To the external blockchain, the transaction looks like it came from a standard single-user wallet. This allows Thorchain to support chains like Bitcoin that do not have complex smart contract capabilities for multisig verification.

4.3 Vault Architecture: Asgard vs. Yggdrasil

Historically, Thorchain utilized a two-tier vault system to balance security and speed, though the architecture has simplified over time.

(1) **Asgard Vaults (Inbound):** These are the primary, massive treasuries containing the bulk of the network's liquidity. They are controlled by a large committee of nodes (typically 20-40). Signing a transaction from an Asgard vault requires a complex, time-consuming TSS ceremony involving many participants. This is secure but slow (~20 seconds to coordinate).

(2) **Yggdrasil Vaults (Outbound - Deprecated/Legacy):** To speed up small swaps, the network formerly assigned smaller, individual vaults to single nodes. A node could sign strictly from its own Yggdrasil vault instantly. *Note: Due to security complexities and the evolution of TSS performance, modern Thorchain relies primarily on highly optimized Asgard vaults, moving away from the Yggdrasil model to reduce the attack surface of individual node compromise.*

4.4 Security Mechanisms: Economic Bonds and Churning

Thorchain does not rely on the "reputation" of its validators. It assumes validators are anonymous and profit-seeking. It secures the network through **Cryptoeconomic Guarantees**.

4.4.1 The Bond and Slashing (Proof-of-Bond). To operate a THORNode, a validator must post a bond in RUNE. The protocol enforces a strict economic relationship known as the **Bond-to-Stake Ratio**, ideally targeting 3:1 (though 2:1 is the functional minimum).

- **The Rule:** For every $1 of asset (e.g., BTC) held in the vault, the validators collectively must bond $3 worth of RUNE.
- **The Guarantee:** If a node attempts to steal $1 from the vault, the protocol can detect this unauthorized transaction. The node is slashed (penalized) 1.5x the value of the theft from their RUNE bond.
- **Rationality:** It is mathematically irrational to steal, as the attacker loses more value in RUNE than they gain in stolen BTC.

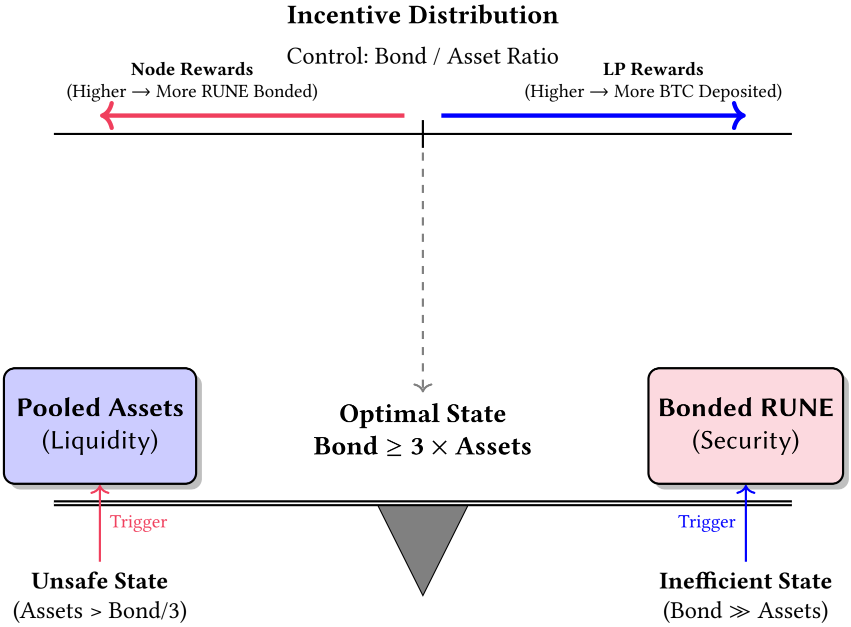

Figure 8. The Incentive Pendulum: Balancing Security and Capital Efficiency

4.4.2 The Incentive Pendulum. The system automatically balances capital efficiency and security using the **Incentive Pendulum** (Figure 8) [11].

- **State A (Under-Bonded):** If the value of assets in the liquidity pools rises (e.g., BTC pumps) such that the bond is no longer 3x the assets, the network is "unsafe." The Pendulum shifts block rewards *away* from Liquidity Providers and *to* Node Operators. This increases the yield for bonding RUNE, attracting more nodes and capital until the ratio is restored.
- **State B (Over-Bonded):** If there is too much bonded RUNE, the system is inefficient. Rewards shift to Liquidity Providers to attract more assets.

4.4.3 Churning (Moving Target Defense). Static vaults are vulnerable. **Why? Because they act as stagnant "honey pots."** If a vault address remains unchanged for years, attackers have infinite time to identify the specific nodes holding the key shares, attempt social engineering, or find zero-day exploits. By the time an attack is prepared, the target must have already moved.

To mitigate this, Thorchain implements **Churning** (Figure 9). This is a mandatory, periodic rotation of the validator set and the physical vault addresses.

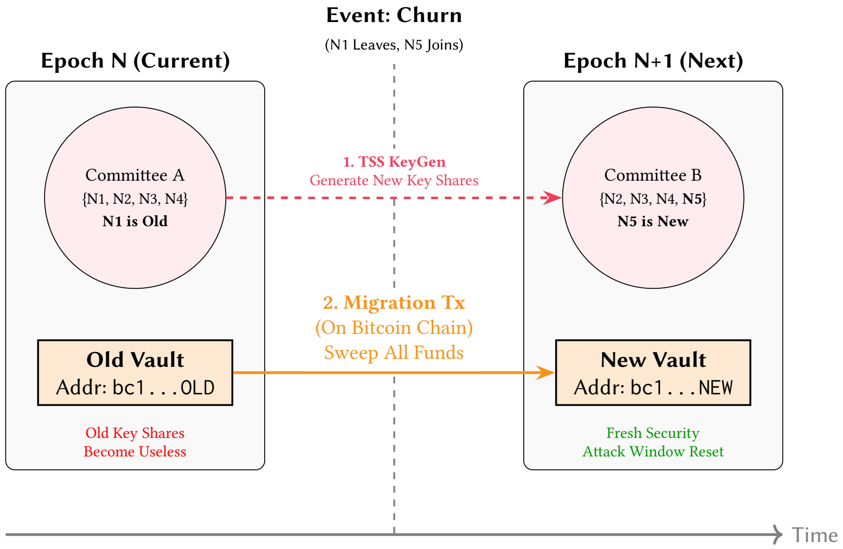

Figure 9. The Churning Process: From Retirement to Migration

Every few days (or upon node failure/entry), the network forces a "Churn":

(1) **Retirement & Selection:** Old nodes (e.g., N1) are rotated out to claim their rewards; new nodes (e.g., N5) are bonded and rotated in.
(2) **KeyGen:** The new committee (Committee B) collaborates to generate a completely new Asgard Vault key via TSS.
(3) **Migration:** The old committee signs a transaction to move all funds from the Old Vault address to the New Vault address. This is a visible on-chain transaction.

This ensures that even if an attacker manages to compromise a few nodes' key shares, those shares become useless within days when the vault moves.

5 RESILIENCE AND FAILURE: LESSONS FROM THE FIELD

Both architectures have faced existential threats. Analyzing these failures provides the deepest insight into their respective trade-offs.

5.1 Thorchain's 2021 "Chaosnet" Exploits

In July 2021, Thorchain suffered a series of hacks known as the "Chaosnet" exploits, losing approximately $8 million. Unlike Wormhole's cryptographic verification failure, these were **Logic Bugs** in the router code.

The "Fake Deposit" Attack: The attacker exploited the Ethereum router's handling of `msg.value`.

(1) The attacker sent a transaction to the router with a `msg.value` of 0 ETH.
(2) However, they wrapped this in a contract that sent a "deposit event" claiming they had deposited a massive amount of ETH.
(3) The Bifrost (observation layer) read the event log but failed to cross-verify the actual ETH value transferred in the transaction.

(4) The network credited the attacker with fake ETH, which they immediately swapped for real assets and withdrew [1].

Recovery: Thorchain demonstrated resilience by halting the network (a consensus decision by nodes), patching the bug, and covering the losses from the protocol's treasury rather than seeking a VC bailout. This reinforced the ethos of a self-sovereign decentralized network.

5.2 The "Death Spiral" Risk (Economic Security Failure)

While the 2021 hacks were software bugs, the fundamental vulnerability of the Trust-Minimized model is **Market Volatility**. Unlike Wormhole, where security is binary (signatures are either valid or invalid), Thorchain's security is linear and depends on the price of RUNE.

The Mechanics of Failure: Recall the security guarantee: *Validators must bond $3 in RUNE for every $1 of Bitcoin they secure.* This makes stealing irrational. However, crypto markets are volatile.

- **The Decoupling Event:** Imagine a "Black Swan" scenario where the price of RUNE crashes by 80% in an hour (e.g., due to panic selling), while the price of Bitcoin remains stable.
- **The Incentive Flip:**
  - **Before:** Bond Value ($300k) > Bitcoin Value ($100k). Theft is irrational.
  - **After:** Bond Value drops to $60k. Bitcoin Value stays at $100k.
- **Rational Theft:** Now, the rational economic behavior for a validator is to **steal the user's Bitcoin**. Even if the network slashes their entire bond ($60k), they still net a profit of $40k ($100k stolen - $60k penalty).

The Consequence: To prevent this, Thorchain implements strict **Halt Rails**. If the price of RUNE drops too quickly, the network automatically pauses all outbound transactions. This highlights a critical trade-off: **Safety over Liveness**. In times of extreme volatility, when users most want to bridge out, the bridge *must* stop working to prevent validators from looting the vaults. This proves that "Trust-Minimized" bridges are not secure by mathematics (like ZK bridges), but secure by market incentives.

6 COMPARATIVE ANALYSIS: TRUSTED VS. TRUST-MINIMIZED

Comparing Wormhole and Thorchain highlights the fundamental trade-offs in bridge design (Table 1).

Future Outlook: The Convergence. As we look toward late 2025 and beyond, the lines are blurring.

- **Wormhole** is shedding its trusted skin. By implementing **ZK Light Clients**, it aims to achieve the "Trustless" property of Cosmos IBC without the extensibility constraints. If successful, verify-via-math will replace verify-via-reputation.
- **Thorchain** is retrenching to simplicity. The failure of ThorFi proves that algorithmic banks are incredibly difficult to secure. The network is doubling down on its role as a decentralized exchange (DEX) and integrating with aggregators to serve as the backend settlement layer for wallets, rather than a consumer-facing bank.

7 CONCLUSION

The journey from Trusted to Trustless bridging is the defining technical challenge of this crypto cycle.

- **Trusted Bridges (Wormhole)** act as efficient "Notaries." They are fast and ubiquitous but act as custodians. The 2022 hack proved that even a single line of code can bypass the most prestigious validator set.

Table 1. Comparative Analysis: Trusted vs. Trust-Minimized Bridges

Feature	Wormhole (Trusted)	Thorchain (Trust-Minimized)
Architecture	**Lock-and-Mint:** Assets are locked on Source; Wrapped tokens (IOUs) minted on Destination.	**Native Swap:** Assets are swapped peer-to-peer via vaults. No wrapped tokens created.
Verification	**Proof-of-Authority:** 13/19 Guardians sign messages. Trust relies on the reputation of entities like Jump Crypto.	**Proof-of-Bond:** Anonymous nodes bond RUNE. Trust relies on economic disincentive (Slashing > Theft).
Attack Surface	**Key Compromise / Smart Contract Logic:** If Guardians collude or verification logic is bypassed (Sysvar bug), infinite minting is possible.	**Economic Decoupling:** If RUNE price crashes toward zero, the bond value ($3) may fall below asset value ($1), breaking security.
User Risk	**Long-Term De-peg:** Holding "WeETH" on Solana carries the perpetual risk that the bridge could be hacked years later.	**Swap-Execution Only:** Once the user receives native ETH, they have zero ongoing relationship with or risk from Thorchain.
Capital Efficiency	**High:** Assets just sit in a vault. Wrapped tokens can be minted infinitely (technically).	**Low:** Requires massive liquidity pools (RUNE + Asset) on every connected chain. High slippage for large trades.
Latency	**Fast:** Seconds to minutes (Observation + 13 Signatures).	**Slow:** Inbound Tx confirmation + TSS Ceremony (can take 10-60 mins for BTC).

- **Trust-Minimized Bridges (Thorchain)** act as "Sovereign Vaults." They offer true asset ownership but demand complex economic engineering. The 2025 insolvency showed that economic security models are vulnerable to market volatility in ways cryptographic models are not.

The takeaway is that **interoperability is not a solved problem**. It is a frontier where cryptography (ZK proofs, MPC), economics (Game Theory, Bonding), and software engineering (Smart Contract Security) collide. The secure bridge of the future will likely rely not on the reputation of Guardians nor the price of a volatile token, but on the absolute certainty of Zero-Knowledge mathematics.

REFERENCES

[1] The Block. 2021. *Thorchain suffers $5 million exploit, developers have put out a fix.* https://www.theblock.co/post/111660/thorchain-suffers-5-million-exploit-developers-have-put-out-a-fix

[2] Fox Business. 2022. *Jump Trading replaces stolen Wormhole funds after $320M crypto hack.* https://www.foxbusiness.com/markets/jump-trading-replaces-stolen-wormhole-funds-after-320m-crypto-hack

[3] Vitalik Buterin. 2021. *The Limits to Blockchain Scalability.* https://vitalik.eth.limo/general/2021/05/23/scaling.html

[4] Wormhole Docs. 2024. *Guardian Network.* https://wormhole.com/docs/protocol/infrastructure/guardians/

[5] Interchain Foundation. 2025. *The Inter-Blockchain Communication Protocol (IBC).* https://cosmos.network/ibc

[6] Rosario Gennaro and Steven Goldfeder. 2020. One Round Threshold ECDSA with Identifiable Abort. Cryptology ePrint Archive, Paper 2020/540. https://eprint.iacr.org/2020/540

[7] Halborn. 2022. *Explained: The Ronin Hack (March 2022).* https://www.halborn.com/blog/post/explained-the-ronin-hack-march-2022

[8] Halborn. 2022. *Explained: The Wormhole Hack (February 2022).* https://www.halborn.com/blog/post/explained-the-wormhole-hack-february-2022

[9] L2BEAT. 2025. *Total Value Secured by Bridges.* https://l2beat.com/scaling/tvs

[10] BBC News. 2021. *Poly Network: Hackers steal $600m in major cryptocurrency heist.* https://www.bbc.com/news/business-58163917

[11] THORChain. 2025. *Incentive Pendulum Documentation.* https://dev.thorchain.org/concepts/incentive-pendulum.html

[12] Wormhole. 2024. *Announcing Wormhole's ZK Roadmap.* https://wormhole.com/blog/announcing-wormholes-zk-roadmap