# Bridgeless Whitepaper

**Abstract.** Long-standing norms with regards to ecosystem development and transparency in the blockchain space have resulted in persistent problems related to onchain interoperability and privacy. As a solution to the persistent challenges in blockchain interoperability and privacy, Bridgeless leverages the Cosmos SDK to offer native cross-chain infrastructure and privacy-preserving features. These integrated technologies create the optimal conditions for developing solutions to the issues previously outlined. These technologies include a TSS, multi-party key management, and the integrated privacy network Zano, in addition to other features, each of which act as catalysts for ecosystem development. The ultimate vision for the Bridgeless ecosystem is the creation of nonpareil solutions to persistent problems in the blockchain space, wherein existing networks have failed to innovate due to lack of infrastructure and incentive.

## l. Introduction

Bridgeless is designed to tackle the unresolved challenges in the blockchain space, particularly with regards to cross-chain interoperability and privacy. Major blockchain networks today are largely homogeneous, collectively aiming for the growth and maintenance of onchain applications and infrastructure. Competition is staked on factors such as daily active users, transaction volume, and Total Value Locked (TVL). In optimizing for these factors, networks inevitably focus on making incremental improvements in scalability, composability, and developer tools. While this focus on optimizing internal ecosystems has driven progress, it has also left critical gaps, especially in areas like bridging solutions and privacy.

Existing cross-chain bridges often rely on centralized or semi-centralized entities, which introduce risks such as single points of failure, trust dependencies, and regulatory vulnerabilities—directly undermining blockchain's core principles of decentralization, trustlessness, and security. Furthermore, the lack of robust interoperability between privacy-focused and public blockchains has stunted the broader adoption of privacy-preserving technologies. These issues represent untapped opportunities for networks that prioritize solving fundamental technical challenges rather than simply competing on established metrics.

Bridgeless is uniquely positioned to address these gaps by offering advanced privacy features and secure, trustless cross-chain interoperability for the development of applications and infrastructure in these fields. By focusing on solving these critical issues, Bridgeless not only mitigates the limitations of existing solutions but also paves the way for a new generation of decentralized applications (DApps) that demand higher privacy and multi-chain capabilities. This approach lays the foundation for a more secure, flexible, and scalable blockchain ecosystem that can drive the future of cross-chain transactions and decentralized applications.

## ll. Fundamental Principles Underpinning Bridgeless

As outlined in the introduction section, Bridgeless is designed to address critical deficiencies in the current blockchain landscape by focusing on unsolved problems that existing solutions have failed to adequately address. Specifically, traditional networks exhibit a focus on internal ecosystem development and iteration on existing infrastructure and applications. This leaves the present onchain landscape with a number of unsolved challenges particularly with regards to interoperability and privacy. Although dedicated privacy networks and interoperability infrastructure exist, they typically operate in isolation or rely on centralized mechanisms that reintroduce trust dependencies and single points of failure.

To address these shortcomings, the Bridgeless network and its ecosystem is designed to solve critical challenges in the onchain environment, particularly those related to privacy, decentralization, and cross-chain interoperability. As alluded to in the network's title, Bridgeless is designed to bridge the unbridgeable, addressing previously unsolved problems in the interoperability space. This philosophy guides the technical and architectural decisions behind the protocol and sets Bridgeless apart from existing solutions that rely on traditional, often centralized, bridging methods. Specifically, the project aims to adhere to the following principles:

## 2.1. Interoperability

With the multitude of networks currently operating today, interoperability has become one of the most critical challenges faced by users and builders. Each blockchain typically operates largely in isolation, limiting the flow of assets, data, and functionality between networks.

This siloed structure has led to inefficiencies, such as fragmented liquidity, isolated DeFi ecosystems, and limited cross-chain asset utilization. Existing interoperability solutions, like centralized bridges, have introduced security vulnerabilities and trust dependencies that contradict the foundational principles of blockchain, such as decentralization and trustlessness.

Bridgeless enables cross-chain interoperability without compromising security, efficiency, or user experience. Current bridging solutions leave certain ecosystems underserved and impart significant risk to their users. Meanwhile, Bridgeless is designed to support multiple blockchain architectures, ranging from EVM-compatible networks like Ethereum [1] to Unspent Transaction Output (UTXO)-based chains like Bitcoin [2], ensuring a versatile and broad-reaching solution for the blockchain community. This interoperability extends beyond simple asset transfers, offering composable cross-chain operations for onchain applications.

## 2.2. Privacy

Transparency is fundamentally ingrained into numerous blockchain networks. This trend was initiated by Satoshi Nakamoto during the development of the Bitcoin network due to the expectation that transparency was a prerequisite for trustlessness. While transparency has facilitated decentralized verification and accountability, it comes with inherent privacy trade-offs, especially in contexts where confidentiality is critical. However increasingly, privacy preserving technologies are becoming available which enable trustlessness in the absence of transparency.

Bridgeless addresses these privacy concerns by integrating advanced privacy-preserving technologies that maintain trustlessness without relying on transparency. This enables the development of DApps that prioritize user confidentiality without compromising trustlessness or decentralization. This design empowers applications that require high levels of confidentiality, supporting use cases where transparency could otherwise expose users to risks or violate privacy regulations.

## 2.3. Decentralization & Trustlessness

Blockchain technology was founded on the principles of decentralization and trustlessness, aiming to eliminate intermediaries and create a system where users can interact directly with the protocol without relying on centralized entities. However, the present onchain landscape is burdened with applications and infrastructure which include centralized mechanisms for key operations like asset transfers, governance, and network validation. These centralized elements reintroduce points of failure, vulnerability to attacks, and trust dependencies, undermining the original intent of blockchain's trustless design.

Bridgeless, meanwhile, is committed to fully embracing decentralization and trustlessness at every layer of its architecture. Decentralized consensus between validator nodes as well as multi-party key management and computation, ensure that no single party holds complete control over the mechanisms required for transaction execution and block production, meaning that asset transfers and other critical operations cannot be executed even in the presence of malicious actors. Through these mechanisms, Bridgeless provides a trustless environment where users can transact and interact without the need to trust centralized intermediaries, preserving the core values of blockchain technology.

# lll. Network Architecture

Several interconnected components facilitate operations on the Bridgeless network. Fundamentally, Bridgeless is a Cosmos Software Development Kit (SDK) [3] based blockchain operating within the Cosmos ecosystem. The network state is stored and updated by validator nodes operating under Delegated Proof-of-Stake (DPoS) [4] consensus. These nodes manage data storage and message handling, maintain the blockchain ledger, and validate transactions.

Data communication and synchronization between nodes is conducted by the RabbitMQ [5] message broker and PostgreSQL [6] database. These data handlers are employed by validator nodes to ensure state consistency across the network.

Cross-chain operations are facilitated by a Threshold Signature Service (TSS) that leverages an advanced multiparty threshold Elliptic Curve Digital Signature Algorithm (ECDSA) [7]. A TSS is a cryptographic protocol that allows a group of participants to jointly produce a signature, such that only a subset (threshold) of them is required to generate it. This enables decentralized, multi-party key management, with Bridgeless's specific implementation significantly improving speed and communication efficiency. Smart contracts on compatible blockchains are responsible for asset locking and unlocking during token transfers. Interaction and integration with external blockchains is managed through Remote Procedure Call (RPC) providers. An RPC provider is a protocol that allows a program to execute a procedure on a remote server as if it were local. RPC providers allow Bridgeless to query data and submit transactions to the aforementioned external chains.

### 3.1. Cosmos SDK Core

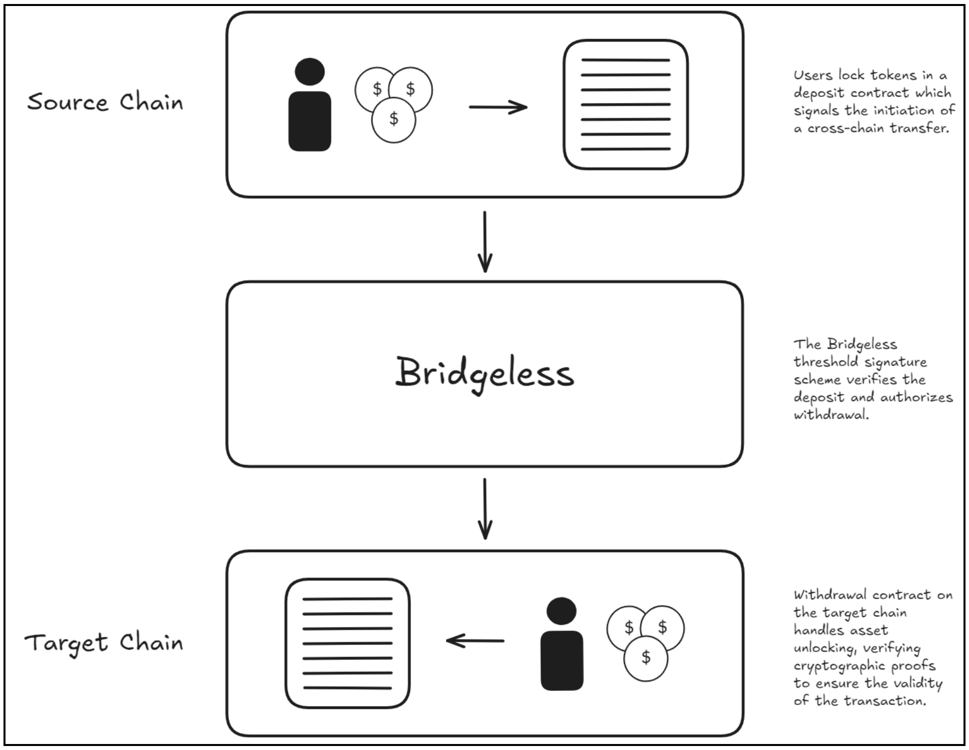
The Bridgeless network is built with the Cosmos SDK, a modular framework for the development and deployment of interoperable blockchains. Fundamentally, the Cosmos SDK is centered around the Tendermint Core [8], a Byzantine Fault Tolerant (BFT) consensus engine [9] which ensures fast finality and secure block production across a network of validator nodes. In addition to the Tendermint Core, Cosmos offers a number of prefabricated modules for governance, staking, and interoperability, allowing developers to extend functionality by creating custom modules that integrate seamlessly with existing components.

User accounts, token balances, and the issuance of fungible and non-fungible tokens (NFTs) are all handled by the SDK-based chain underpinning the Bridgeless network. With regards to token issuance, Bridgeless supports custom tokenomics as defined by the party issuing the token. These include staking mechanisms, reward distribution models, and governance token use cases. Governance for Bridgeless itself is managed by the Cosmos SDK Governance Module [10], which enables stakeholders to vote on proposals for upgrades or changes.

### 3.2. DPoS Consensus

Validators on the Bridgeless network reach consensus via DPoS to manage block production and state validation. In this model, token holders delegate their staking power to trusted validators, who are responsible for proposing and validating new blocks. This system ensures that only the most reputable validators, as determined by their delegated stake in the network, are involved in critical network functions, thereby decentralizing control while maintaining high throughput and security. DPoS offers a more scalable and efficient consensus mechanism compared to traditional Proof-of-Stake (PoS) [11] by allowing a smaller set of validators to secure the network, which results in faster block times and reduced computational overhead.

The delegation process enables token holders to indirectly participate in the consensus protocol without needing to run a node. Validators, in turn, are incentivized to act honestly and efficiently on behalf of their delegates by receiving staking rewards, which are shared with their delegators proportionally. This model creates an alignment of interests between token holders and validators, ensuring that the network remains both decentralized and secure. Additionally, DPoS enables dynamic validator selection, meaning that underperforming or malicious validators can be voted out by the community, further enhancing the network's security and trust.


Source Chain
Users lock tokens in a deposit contract which signals the initiation of a cross-chain transfer.
Bridgeless
The Bridgeless threshold signature scheme verifies the deposit and authorizes withdrawal.
Target Chain
Withdrawal contract on the target chain handles asset unlocking, verifying cryptographic proofs to ensure the validity of the transaction.

## 3.3. Validator Nodes

Validator nodes are responsible for proposing, validating, and finalizing blocks of transactions. Operating under a DPoS consensus model, validators are selected by token holders who stake their tokens and delegate their staking power to trusted validators. Therefore, only the most trusted validators are selected by the community of stakers, thereby decentralizing the process of block production. Validators store the current blockchain state via blocks, which contain a ledger of transactions as well as metadata [12] relating to cross-chain operations, asset transfers, and governance decisions.

Consensus among validators is achieved via the aforementioned Tendermint BFT algorithm, ensuring agreement on the state of the blockchain even in the presence of faulty or malicious nodes. They also participate in governance decisions, proposing and voting on protocol changes to maintain the health and evolution of the network.

Additionally, validators are responsible for state management, storing essential network state information, including mappings of cross-chain assets, public key distributions of threshold signature producers, and system configurations such as staking and governance parameters. They also maintain an audit log of all cross-chain transactions, ensuring transparency and accountability.

Several mechanisms are included to ensure resilience against faults or downtime. If the network experiences downtime or a service outage, transactions are queued and processed once the system is restored. This ensures that users face minimal disruption and all valid transactions are processed.

## 3.4. Threshold Signature Scheme

The TSS is one of the core components responsible for securely managing cross-chain asset transfers. Operating as a decentralized signing mechanism, the TSS splits signing keys across multiple parties (validators). This TSS specific protocol [7] distributes signing authority across multiple validators in a way that ensures a subgroup of $t + 1$ participants out of $n$ total can jointly sign transactions, while any subset of $t$ or fewer cannot. Unlike traditional ECDSA threshold schemes that require a costly key setup or complex zero-knowledge proofs, the scheme used in Bridgeless significantly reduces communication complexity and is robust against malicious adversaries, even under a dishonest majority model. Key aspects of the TSS include:

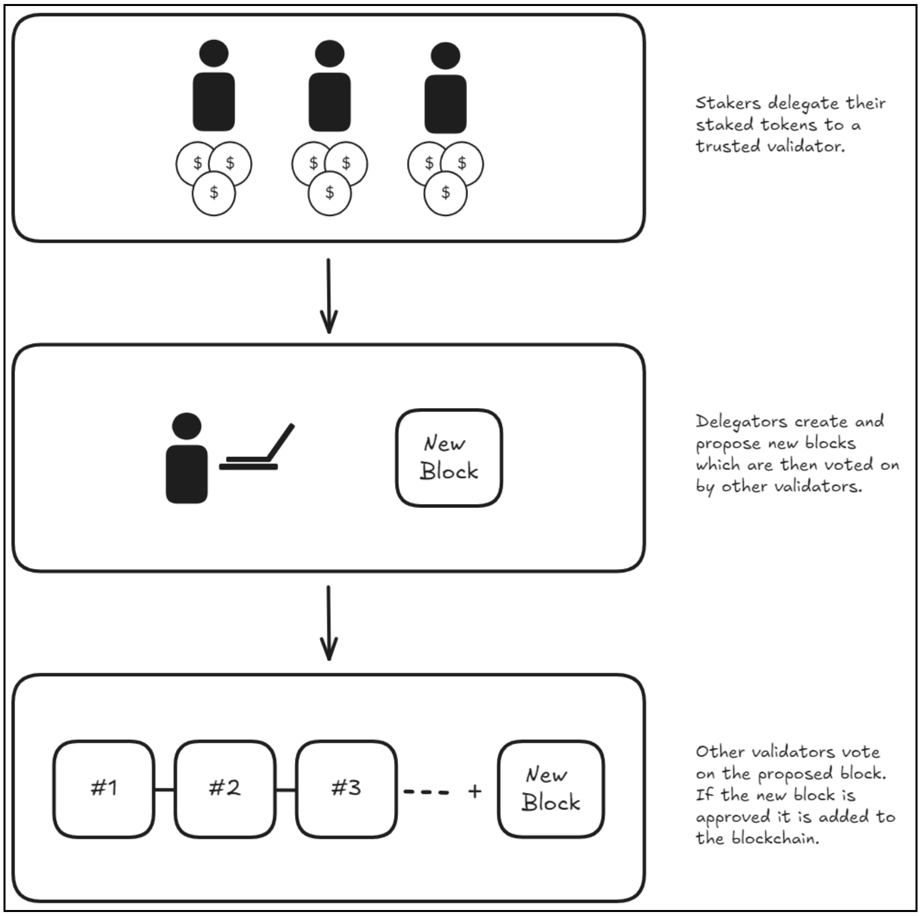
- **Decentralized Signature Generation:** The TSS operates by distributing private keys among validators using Shamir's secret sharing, ensuring no single entity controls the entire signing key. Multiple validators collaboratively compute partial values using the Gennaro-Goldfeder scheme, leveraging additive sharing and share conversion protocols to generate signatures efficiently. This process exhibits a linear communication overhead, enabling Bridgeless to scale without sacrificing security or decentralization.
- **Cross-Chain Transaction Verification:** Before any withdrawal on a target chain is authorized, the TSS verifies the corresponding deposit on the source chain. This is achieved by having validators collaboratively perform multiparty computation (MPC) [13] to verify deposits on the source chain using RPC calls (e.g., Bitcoin or Ethereum) and then sign the corresponding withdrawal transaction. This collective validation process using secure multiparty threshold ECDSA ensures resilience against attempts to tamper with transaction details.
- **Key Management and Resharing:** The TSS's dynamic key resharing mechanism ensures that validators can securely redistribute key shares when membership changes without exposing the private key, maintaining the $t + 1$ quorum for signing. This mechanism, coupled with secure MPC, prevents compromised nodes from disrupting transactions or learning the complete private key, preserving system integrity even under adverse conditions.

The integration of this TSS protocol within Bridgeless provides a highly secure, scalable, and efficient mechanism for decentralized cross-chain asset management.

## 3.5. Smart Contract Bridging

On Ethereum and other EVM-compatible chains, Bridgeless deploys a suite of smart contracts responsible for managing cross-chain deposits and withdrawals.

Users interact with deposit contracts by locking assets (ERC-20 or native tokens) on the source chain. These contracts emit events that signal the initiation of a cross-chain transfer, triggering the verification and signature processes. Upon verification of a deposit, the TSS authorizes the withdrawal of assets from the target chain via the withdrawal contract. These contracts handle signature verification and ensure that withdrawals can only occur once a valid cryptographic proof has been provided.


$ $ $
Stakers delegate their staked tokens to a trusted validator.
New Block
Delegators create and propose new blocks which are then voted on by other validators.
#1
#2
#3
+
New Block
Other validators vote on the proposed block. If the new block is approved it is added to the blockchain.

The smart contracts are designed to support onchain upgrades, allowing Bridgeless to evolve without redeploying new contracts. The upgrade process can be governed through decentralized voting, ensuring that changes to contract logic are approved by validators and stakeholders.

## lV. Bridgeless Technology Stack

The underlying technology stack of Bridgeless is designed to provide a robust, scalable, and secure foundation for general network operations as well as features related to privacy and interoperability. Optimized for high-throughput performance, the technology stack ensures the Bridgeless network can process large volumes of transactions while maintaining low latency.

Designed to handle the inherent complexity of decentralized multi-chain operations, the technology stack integrates several key components that work in unison to enable efficient cross-chain transactions. These components facilitate interface with heterogeneous blockchain networks through standardized Application Programming Interfaces (APIs), ensuring broad compatibility and seamless integration with external systems.

A modular design allows for future upgrades and continuous development, ensuring that the technology stack can evolve with the growing demands of dApps and cross-chain functionality. Bridgeless places significant emphasis on security, leveraging advanced cryptographic techniques like threshold signatures and MPC to safeguard assets and transactions across multiple blockchains.

## 4.1. Development Frameworks

The backend services and core infrastructure of Bridgeless are developed using Golang version 1.22 [14]. Cross-chain operations, such as real-time transaction tracking and proof generation, require efficient management of multiple processes, making Golang an optimal choice. Moreover, the Golang's concurrency model [15] and performance in handling parallel operations, ensures that the Bridgeless can scale to manage high transaction throughput and communication across multiple blockchains.

Smart contract development on Bridgeless is conducted using Solidity version 0.8.9 [16]. Solidity enables Bridgeless to implement the decentralized logic for asset transfers and verification processes. The network's cross-chain functionality relies on smart contracts to collect and store deposit data, generate event-based triggers, and manage withdrawals. Solidity's support for complex smart contract functionality allows for the defining of precise logic for cryptographic signature verification, secure asset locking, and withdrawal processes, while ensuring compatibility with Ethereum's established ecosystem.

The core blockchain is built using the Cosmos SDK version 0.50+ [17], which is critical for providing the modular structure required to support multi-chain operations. Cosmos's modularity allows Bridgeless to include custom network logic while maintaining compatibility with the Inter-Blockchain Communication (IBC) protocol [18]. This is essential for enabling decentralized cross-chain transactions and facilitating secure governance through a DPoS consensus mechanism.

To handle cryptographic operations, Bridgeless utilizes TSS-Lib version 2.0.2 [19], a threshold signature library. This library enables MPC where private key shares are distributed among multiple nodes, ensuring no single node holds the entire private key. The library supports threshold ECDSA and EdDSA [7] signatures, which are essential for Bridgeless's decentralized security model. By distributing cryptographic key control across multiple nodes, the TSS-Lib ensures that critical operations, such as the signing of cross-chain withdrawal transactions, require multiple participants' approval, significantly reducing the risk of compromise.

## 4.2. Validator Setup

Bridgeless is designed to be deployed in cloud environments that support dynamic scaling and high availability. The core components, including validator nodes and threshold signature producers, are run in Kubernetes clusters [20] to ensure fault tolerance and scalability.

The recommended node setup includes 6 CPU cores and 10GB of RAM per node, with 500GB of storage to support long-term data retention and higher throughput. Running ancillary services such as a blockchain explorer can require additional computational resources. Network performance is critical, as the system relies on sub-10ms latency to maintain real-time synchronization between validator nodes and threshold signature producers.

By deploying in a cloud-native environment, the system can scale horizontally by adding more nodes to meet demand, ensuring that transaction processing times remain stable under load.

## 4.3. Data Handling

Bridgeless relies on a message broker architecture to handle asynchronous communication between its components, including validator nodes, the TSS, and external services. RabbitMQ is used as the message broker, while PostgreSQL serves as the primary database for storing operational and state data.

As the message broker, RabbitMQ facilitates inter-component communication, allowing the system to handle event-driven processes such as transaction finalization, key resharing events, and signature generation requests. This event-driven model ensures that messages are processed in order and critical tasks, such as signing transactions, are distributed efficiently across the network.

The database, PostgreSQL, stores all non-volatile state information, including logs of cross-chain transactions, validator configurations, governance history, and cryptographic proofs. This database is replicated across each validator node, acting as the source of truth for tracking the network's state during failover events or restarts. The stored data includes mappings of cross-chain assets and the records of TSS participation in each transaction.

In addition to handling internal data, Bridgeless relies on RPC providers for interacting with external blockchains, enabling the network to query onchain data, verify transactions, and submit signed actions like withdrawals or deposits. These RPC interactions are crucial for real-time communication with external blockchains, supporting Bridgeless's cross-chain functionality by facilitating data retrieval and transaction execution.

For UTXO-based chains like Bitcoin, Bridgeless uses a self-hosted Bitcoin Core instance to track UTXOs, allowing it to verify deposits and manage transactions directly from the Bitcoin network. This integration ensures real-time monitoring of UTXOs, enabling decentralized control without reliance on third-party services that may impose limits or fees on transaction tracking.

For EVM-compatible chains such as Ethereum, Bridgeless leverages Infura [21], a scalable service providing access to Ethereum's network. Infura enables Bridgeless to interact with smart contracts, query transaction statuses, and submit signed transactions to EVM-based blockchains, ensuring automated and real-time execution of cross-chain deposits and withdrawals.

## 4.4. Key Management

Bridgeless's security architecture is designed to protect cryptographic keys and facilitate secure transaction execution across chains. Key management is handled using the TSS-Lib version 2.0.2, which provides the cryptographic foundation for threshold signature generation. In this model, private keys are split into shares and distributed among multiple nodes, with no single node having access to the complete key. This multi-party signature process ensures that any cross-chain transaction, such as a withdrawal from a target chain, requires multiple validators to approve the action.

In addition to using the TSS-Lib for signature generation, Bridgeless integrates with HashiCorp Vault [22] for secure key storage and management. HashiCorp Vault is used to store private key shares and ensure that all access to cryptographic materials is logged and controlled. The vault's ability to manage key rotation, backups, and access control policies adds an extra layer of security to the network's key management infrastructure. If a key compromise is detected, Bridgeless can quickly rotate keys and redistribute shares without interrupting ongoing cross-chain operations. This combination of the TSS-Lib and HashiCorp Vault ensures that the network maintains a high level of security, even in the face of potential attacks or node failures.

### 4.5. External Integrations

Cross-chain operations on Bridgeless depend on its integration with multiple blockchain protocols. The network supports Bitcoin and Bitcoin Cash through a self-hosted Bitcoin Core node which is responsible for handling UTXO-based transactions. This setup allows Bridgeless to track large volumes of UTXOs efficiently and ensures that deposits and withdrawals are processed without delays. The use of a self-hosted Bitcoin node also provides more granular control over transaction processing, which is essential for managing large cross-chain transfers.

For Ethereum and other EVM-compatible chains, Bridgeless integrates the external RPC provider Infura. This provider facilitates interaction with Ethereum smart contracts and allows the network to query transaction data, monitor blockchain events, and submit signed transactions for asset transfers. By using a reliable RPC provider, Bridgeless abstracts the complexities of blockchain-specific operations, enabling it to maintain uniform transaction flows across different blockchain ecosystems. The RPC integration ensures that transaction validation, smart contract interaction, and event logging are consistent and secure, regardless of the blockchain involved.

## V. Ecosystem Outline

Creating solutions for long-standing problems encapsulates the Bridgeless network's approach towards ecosystem development. Bridgeless aims to cultivate applications that address fundamental limitations of existing infrastructure. Although this is pertinent to any number of niches, Bridgeless aims to primarily focus on applications related to interoperability and privacy.

This focus is exhibited in Bridgeless's core architecture; features such as the TSS enable functionality for decentralized and trustless cross-chain transactions. Moreover, the platform supports native interoperability across EVM and UTXO based networks. This allows developers to expedite the process of building DApps capable of operating across multiple chains, enabling cross-chain asset transfers, liquidity sharing, and smart contract interactions without compromising security or decentralization.

Moreover, integration with Zano [23] gives applications on Bridgeless access to confidential asset transactions [24], making it one of the few platforms capable of supporting private and public blockchain interactions in a decentralized manner. By allowing applications to conduct confidential asset transactions, Zano's integration makes Bridgeless one of the few platforms capable of supporting decentralized interactions between private and public blockchains. This opens up new use cases for privacy-preserving financial systems and decentralized applications, where confidentiality is crucial. Bridgeless's ability to handle both transparent and confidential transactions within a single ecosystem bridges the gap between privacy-focused and public blockchains, introducing the potential for previously impossible applications.

At the nexus of these two focuses is Confidential Layer, Bridgeless's inaugural protocol. Confidential Layer leverages both Bridgeless's cross-chain capabilities and Zano's privacy features [25] to create a platform that supports cross-chain bridging between private blockchains, an unprecedented function. It enables developers to build applications that require secure cross-chain functionality while ensuring user privacy, making Confidential Layer a foundational component of Bridgeless's ecosystem. By providing a flexible, privacy-preserving infrastructure, Confidential Layer positions Bridgeless as a leader in decentralized, multi-chain solutions that meet the growing demand for both interoperability and privacy.

Following the launch of Confidential Layer, Bridgeless is developing a first-of-its-kind non-KYC Bitcoin-Ethereum bridge. This bridge will enable users to seamlessly transfer assets between Bitcoin's UTXO-based network and Ethereum's EVM-based environment without compromising privacy or introducing centralized intermediaries. The non-KYC feature is particularly significant, as it allows users from non-Western regions which would otherwise be unable to bridge their cryptocurrency to engage in cross-chain asset transfers without undergoing identity verification processes that are usual of existing infrastructure and would typically exclude them. This opens up new possibilities for privacy-conscious users, DeFi platforms, and developers seeking to build on a foundation that prioritizes user autonomy and security.

These two applications are indicative of the plan for the wider Bridgeless ecosystem. It is designed to foster applications that operate across both private and public networks, offering a versatile platform for projects that require the seamless movement of assets and data between different blockchain networks. This positions Bridgeless as a unique ecosystem committed to addressing some of the most pressing challenges in the present blockchain landscape.

## Vl. Future Developments

The development roadmap for Bridgeless outlines multiple phases that progressively introduce key functionalities such as cross-chain asset transfers, decentralized signature generation, and broader blockchain integration. Each phase builds upon the foundational system, adding new features to handle growing complexity, additional blockchain support, and enhanced security requirements.

### Phase 1.1 - System Launch

The initial phase is focused on deploying the core modules required for Bridgeless's basic operations. This includes the accumulator module, which aggregates data from various sources to facilitate cross-chain transactions. The NFT module is also introduced, enabling the issuance and staking of NFTs that can hold locked native tokens, adding unique tokenomics functionalities to the network.

The Staking and Distribution Module provides the core staking mechanics, allowing users to delegate and undelegate tokens, participate in governance through delegated voting, and receive staking rewards. This module also incorporates NFT-based staking with enhanced reward options. Additionally, the launch of EVM compatibility ensures that Bridgeless can operate across Ethereum and other EVM-compatible networks using smart contracts to manage true cross-chain operations. This phase includes the deployment of an explorer that allows users to track cross-chain transactions and Bridgeless metrics in real time.

### Phase 1.2 - Bridge Launch and EVM Integration

In this phase, Bridgeless upgrades its core with the integration of bridge functionality, deploying centralized signature producers (with the intention of transitioning to decentralized signature producers) to manage the cryptographic signing of cross-chain transactions on supported networks, with an initial focus on EVM-compatible blockchains. The Bridge Module is introduced, integrating the bridging logic necessary for asset transfers across these chains. This module supports the use of cryptographic signatures generated by centralized signature producers.

Additionally, Bridge Smart Contracts are deployed on EVM networks, enabling cross-chain asset locking and withdrawal processes. The contracts ensure the integrity of these operations by verifying signatures before allowing transfers. The centralized signature producer's role is extended to securely manage asset transfers between chains during this phase.

### Phase 1.3 - Bitcoin and Bitcoin Cash Support

The next phase of development extends Bridgeless's cross-chain interoperability to support Bitcoin and Bitcoin Cash networks. This involves upgrading the centralized signature producer to handle Bitcoin's UTXO-based transaction model, which is fundamentally different from the account-based model used by Ethereum. The cryptographic mechanisms within Bitcoin require specific adaptations to the system's signature generation process. Bridgeless will adapt its bridge logic and contracts to accommodate UTXO-based transactions, allowing for asset transfers between these blockchains. This phase broadens Bridgeless's compatibility, enabling interoperability with both EVM-compatible chains and UTXO-based networks.

### Phase 2 - Threshold Signature Upgrade

In the second major phase, Bridgeless will decentralize its signing process by replacing the centralized signature model with a TSS. This upgrade distributes key shares across multiple validators, ensuring that no single validator has full control over the private keys. The use of a TSS enhances security by requiring a majority of validators to collaborate in signing cross-chain transactions. The key deliverable in this phase is the deployment of decentralized threshold signature producers, which enable secure and decentralized cryptographic operations across multiple blockchains.

### Phase 3 - Zano Integration and Expansion of the TSS

In the next phase, Bridgeless will expand its cross-chain operability by integrating the Zano, a tokenization platform known for its focus on privacy and anonymity. This phase involves customizing the signature generation and bridge logic to accommodate Zano's unique blockchain properties, particularly its transaction privacy mechanisms [26]. Bridgeless will enhance the TSS to support Zano, ensuring secure cross-chain transactions between Zano and other integrated blockchains. This phase will culminate in the full decentralization of cross-chain operations and the protocol's support for a wide range of blockchains, ensuring robust and secure asset transfers in a fully decentralized manner.

By progressively introducing these enhancements, Bridgeless ensures that its cross-chain capabilities evolve while maintaining security and decentralization. Each phase of this roadmap focuses on scaling functionalities in a structured manner, with an emphasis on secure and decentralized operations using advanced cryptographic techniques.

## Vll. Conclusion

Bridgeless addresses critical deficiencies in the onchain environment by delivering dedicated solutions to persistent challenges related to privacy and interoperability. Through a specialized network architecture that integrates cross-chain interoperability mechanisms and privacy protocols, Bridgeless enables the development of applications that can maintain cross-chain functionality and confidentiality. Notably, innovations such as Confidential Layer and the non-KYC Bitcoin-Ethereum bridge exemplify its capacity to capture and address previously unserved use cases, positioning it as a key enabler of future multi-chain, privacy-centric applications. By resolving these core technical limitations, Bridgeless establishes a foundational infrastructure capable of bridging previously intractable gaps in blockchain interoperability, thereby advancing the capabilities of decentralized networks.

## Vlll. References

[1] Buterin, V. (2014). Ethereum whitepaper: A next-generation smart contract and decentralized application platform. Ethereum Foundation. https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf

[2] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

[3] Cosmos. (n.d.). Cosmos SDK documentation: Introduction and overview. Cosmos Network. https://docs.cosmos.network/main/learn/intro/overview

[4] Cosmos. (2017). Validators FAQ. GitHub. https://github.com/cosmos/cosmos/blob/master/VALIDATORS_FAQ.md

[5] RabbitMQ. (n.d.). RabbitMQ documentation. Pivotal Software, Inc. https://www.rabbitmq.com/docs

[6] PostgreSQL Global Development Group. (n.d.). PostgreSQL download. https://www.postgresql.org/download/

[7] Gennaro, R., & Goldfeder, S. (2019). Fast Multiparty Threshold ECDSA with Fast Trustless Setup. City University of New York, Cornell University. https://eprint.iacr.org/2019/114.pdf

[8] Kwon, J., & Buchman, E. (2016). Tendermint: Byzantine fault tolerance in the age of blockchains. Tendermint. https://tendermint.com/static/docs/tendermint.pdf

[9] Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. MIT. https://pmg.csail.mit.edu/papers/osdi99.pdf

[10] Cosmos. (n.d.). Governance module documentation. Cosmos Network. https://docs.cosmos.network/v0.46/modules/gov/

[11] Bitcoin POS. (n.d.). Proof of Stake Bitcoin whitepaper. https://www.bitcoinpos.net/WhitePaperBPS.pdf

[12] LCX. (n.d.). Metadata in blockchain transactions explained. LCX. https://www.lcx.com/metadata-in-blockchain-transactions-explained/

[13] Choudhury, A., Garay, J., Ostrovsky, R., & Riva, B. (2020). Multi-party computation: From theory to practice. International Association for Cryptologic Research. https://eprint.iacr.org/2020/300.pdf

[14] Go Programming Language. (n.d.). Go version 1.22 documentation. Go Programming Language. https://tip.golang.org/doc/go1.22

[15] Go Programming Language. (2012). Go concurrency patterns. Go Programming Language. https://go.dev/talks/2012/concurrency.slide#1

[16] Solidity. (2021). Solidity v0.8.9 release notes. Ethereum Foundation. https://github.com/ethereum/solidity/releases/tag/v0.8.9

[17] Cosmos SDK. (n.d.). Cosmos SDK release notes. Cosmos Network. https://github.com/cosmos/cosmos-sdk/releases

[18] Belchior, R., Pires, S., Vasconcelos, M., Correia, M., & Guerreiro, P. (2020). Inter-blockchain communication protocol. arXiv. https://arxiv.org/pdf/2006.15918

[19] Binance. (n.d.). TSS Lib version 2.0.2 release notes. Binance Chain. https://github.com/bnb-chain/tss-lib/releases/tag/v2.0.2

[20] Kubernetes. (n.d.). Kubernetes architecture documentation. Kubernetes.io. https://kubernetes.io/docs/concepts/architecture/

[21] Infura. (n.d.). Infura API documentation. Infura. https://docs.infura.io/api

[22] HashiCorp. (n.d.). HashiCorp Vault. GitHub. https://github.com/hashicorp/vault

[23] Hyle Team. (2019). Zano Whitepaper: Confidential Assets and Private Transactions. GitHub. https://github.com/hyle-team/docs/blob/master/zano/Zano_WP_latest.pdf

[24] Hyle Team. (2024). Confidential Assets Scheme for RingCT and Zarcanum. GitHub. https://github.com/hyle-team/docs/blob/master/zano/CA_paper/Zano_CA_for_RingCT_and_Zarcanum_v1.1.pdf

[25] Hyle Team. (2022). Zarcanum: Proof-of-Stake Scheme for Confidential Transactions with Hidden Amounts. GitHub.

https://github.com/hyle-team/docs/blob/master/PoS/PoS_with_HA/Zarcanum-PoS-with-hidden-amounts.pdf

[26] Hyle Team. (2024). d/v-CLSAG: Extension for Concise Linkable Spontaneous Anonymous Group Signatures. GitHub. https://github.com/hyle-team/docs/blob/master/zano/dv-CLSAG-extension/dv-CLSAG-extension.pdf