

XION: The Generalized Abstraction Layer

Burnt Research

Draft-Release, December 7, 2023

Abstract

We propose XION, the Generalized Abstraction layer that enshrines user experience (UX) at the protocol level. This enables secure, intuitive, and seamless user experiences through protocol-level implementations related to abstracted accounts, signatures, fees, interoperability, and more. Combining the modular design of XION’s meta accounts, their signature agnostic implementation, and XION’s parameterized fee layer offers a composable, future-proof infrastructure layer. XION simplifies the developer experience and enables a user-centric environment that retains the decentralized ethos of blockchain technology, while significantly lowering the entry threshold for mainstream users. Through abstracted interoperability, we additionally propose a solution to extend the seamless user experience of XION to all connected chains, and explore several novel use-cases it enables.

1 Introduction

Technical challenges impede the widespread adoption of blockchain technology. New users encounter confusing complexity involving wallet setup, management, cross-device usage, purchases, gas fees, multi-chain interactions, and more. While blockchains hold immense potential, their widespread adoption is precluded by these complexities. Such intricacies contribute to high user drop-off rates and low user interest, suggesting the need for a user-friendly blockchain infrastructure. Proposed solutions, such as the introduction of account abstraction¹, have attempted to alleviate these issues. However, implementations which avoid consensus-layer changes have resulted in fragmentation issues²³, significant deployment and execution costs, centralization risk, miner-extractable-value (MEV) capture⁴, and censorship¹. There currently doesn’t exist a holistic solution to bridge the gap between the industry’s technical complexities and mainstream appeal.

This paper explores the novel architecture of XION. The Generalized Abstraction layer’s founda-

tion is built upon the seamless integration of complex blockchain functionalities – such as accounts, signatures, fee management, and interoperability – directly at the protocol level. XION eliminates significant barriers to entry for new users, while circumventing fragmentation challenges for developers. Its signature-agnostic infrastructure supports a wide array of existing cryptographic curves and is readily adaptable to future developments, which not only broadens its market reach but also assures long-term viability and interoperability across different blockchain protocols. Through abstracted interoperability, XION is able to offer a seamless experience not only for native applications, but also those built on connected chains.

The remainder of the paper follows this structure: Section 2 presents a background of the technical concepts XION relies on; Section 3 presents Generalized Abstraction which includes meta accounts, signature abstraction, device abstraction, a parameterized fee layer, and enhanced token minting dynamics; Section 4 presents abstracted interoperability; Section 5 presents novel use-case examples; and Section 6 concludes.

2 Background

2.1 Externally Owned Accounts

In general, blockchains have two types of accounts: 1) externally owned accounts, and 2) smart contract accounts⁵. Traditionally users interact with blockchains through externally owned accounts (EOAs) which involve the use of asymmetric cryptography. These accounts have a public/private key pair, whereby the public key is stored on the blockchain, and the private key is stored off-chain by the user. The private key, known only to the user, is used to sign transactions, while the public key is used to verify the signature's authenticity.

EOAs, however, have many drawbacks: they lack the ability to implement additional authentication mechanisms, they aren't able to perform autonomous operations or smart contract executions, and a private/public keypair can not be changed. As a result, EOAs are a central point of failure for users. If a user loses access to their private key, they lose access to their account entirely. Similarly, if a user's private key is compromised, their account is irreversibly compromised.

2.2 Smart Contract Accounts

The second type of account, smart contract accounts (SCAs), are governed by code on the blockchain. Their creation involves being deployed to the blockchain through a transaction initiated by an EOA. Once deployed, these smart contracts reside at a specific address on the blockchain and their code dictates the rules and conditions under which they operate. These rules are executed autonomously when certain conditions are met or when they are triggered by transactions or other smart contracts. Traditionally, they are unable to initiate transactions due to an absence of private keys.

As a solution to EOA drawbacks mentioned above, account abstraction has been proposed as a way to enable SCAs to initiate transactions¹. Rather than having transaction authentication determined

by a predefined set of rules at the state machine level, account abstraction delegates this task to SCAs. SCAs can then implement customized authentication logic as needed, such as accommodating key rotation, performing autonomous tasks, integrating multiple-factor authentication, and much more.

2.3 Signatures

Cryptographic signatures are created using a digital signature algorithm, which involves a set of mathematical operations. The most commonly used signature schemes in Web3 are Elliptic Curve Digital Signature Algorithm (ECDSA)⁶ and Edwards-curve Digital Signature Algorithm (EdDSA)⁷. Ethereum and Bitcoin use ECDSA, specifically leveraging the Secp256k1 curve⁵⁸, while Solana uses EdDSA, specifically leveraging the Ed25519 curve⁹. Another widely accepted cryptographic curve, Secp256r, is used by most popular consumer devices such as Apple's Secure Enclave¹⁰ and Android devices¹¹.

2.4 Transactions

Within the standard Cosmos SDK¹², each transaction is composed of messages. Once a transaction is set to be included in a block, it goes through a unique component called the AnteHandler. The AnteHandler is responsible for executing specific checks and operations through a series of functions, known as decorators, which are executed prior to each transaction. These ensure the validity of the transaction, signatures, fees, nonce verification, and more prior to the transaction execution. Once the messages have been executed, a PostHandler is invoked with its own set of decorators. These decorators are designed to execute logic such as events, logs, post-transaction logic, and more after a transaction has been processed and prior to committing it to the block.

2.5 Transaction Fees

Transactions fees, often referred to as "gas", are used to measure the amount of computational work required to execute a transaction. Traditionally, each

transaction consumes a certain amount of gas depending on its complexity and the computational resources it requires. Validators, who are responsible for confirming transactions and adding them to the blockchain, set a minimum gas price. Users who want their transactions processed must pay at least this minimum gas price. The total fee a user pays for a transaction is calculated by multiplying the amount of gas used by the gas price. Users include the gas fee in their transactions and it is paid in the native token of the blockchain they are using. When a transaction is included in a block, the gas fee is deducted from the user's account and awarded to the validator who included the transaction in the block. As covered in Section 3.4, XION's parameterized fee layer implements significant changes in order to abstract away complexities for end-users, provide frictionless gasless experiences, and enable token agnostic payments & denominations.

3 Generalized Abstraction

Generalized Abstraction is a unique and all-encompassing infrastructure solution aimed at removing inherent crypto complexities for all users. The crux of this innovation is XION's implementation directly at the protocol level, which seamlessly abstracts accounts, signatures, gas, interoperability, pricing, devices, payments, and more. In doing so, through a holistic approach, XION lays a robust foundation for the next-generation of projects to bring the promises of Web3 to mainstream audiences worldwide. The sections below detail the different facets of Generalized Abstraction.

3.1 Protocol-Level Account Abstraction: Meta Accounts

XION introduces *meta accounts*, through a protocol-level implementation involving smart contract accounts (SCAs) and state machine changes¹³. These *meta accounts* streamline user interactions by decoupling the conventional private-public key model and enabling the creation of more intuitive user interfaces that align with traditional Web2

login systems. Users interact with their accounts in familiar ways such as through email or biometric authentication methods, eliminating the need for direct private key management, while still remaining fully non-custodial.

XION's modular *meta accounts* framework introduces a highly adaptable and secure permission management system, with the ability to support advanced features such as key weights, key rotation, rule sets, and a variety of authentication methods. This system offers significant versatility and enhanced security for account management:

- **Key Rotation:** A parameter integral to maintaining security by proactively reducing the risks associated with potential key compromise by enabling the changing of account keys, thus limiting the duration of their exposure.
- **Rule Sets:** A parameter enabling account holders to set any number of custom rules that govern the account, ranging from transaction limits to recurring payments.
- **Key Weights:** A parameter enabling different levels of importance or authority to each key, allowing for a nuanced access control system within the account, where certain actions require keys with higher weights.
- **Diverse Authentication Methods:** By supporting a range of authentication methods, *meta accounts* achieve both interoperability across different devices & platforms, resilience against evolving cryptographic threats, seamless support for cryptographic innovations, and ensures robust protection for user accounts.
- **Multi-Factor Authentication Framework:** A multi-factor authentication framework enables a flexible and robust security structure, ensuring that access and control within the account adhere to specific, customizable parameters defined by the user or organization when executing transactions.

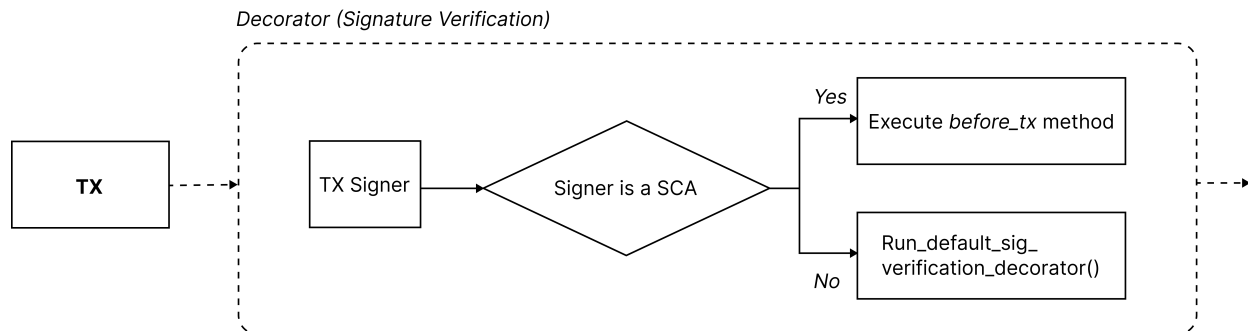


Figure 1: SCA Signature Verification Decorator

3.1.1 Smart Contract Account Implementation

In order to enable smart contract accounts to sign, the responsibility of transaction authentication must shift from the state machine to the smart contract accounts (SCAs). This is achieved by integrating two critical methods into SCAs: *before_tx* and *after_tx*. Prior to a transaction’s execution, the state machine calls the *before_tx* method, which provides the SCA with detailed transaction information and its signing credentials, allowing for signature verification and other programmed actions. Following the transaction’s execution, provided the *before_tx* method and all transaction messages have been successfully executed, the *after_tx* method is activated, enabling the SCA to perform additional programmed actions.

3.1.2 State Machine Implementation

The state machine updates involve two decorator changes in order to trigger the SCA’s *before_tx* and *after_tx*. The `SigVerificationDecorator` is updated with a new decorator which triggers the *before_tx* method when SCAs sign, or simply proceeds with the default `SigVerificationDecorator` logic. This is demonstrated in Figure 1. An additional decorator is then added to the `PostHandler` to trigger the *after_tx* method.

3.2 Signature Abstraction

XION’s curve-agnostic implementation extends its SCA capabilities, offering significant advantages

over existing SCA solutions. Up to 256 different authenticators can be added to a user’s *meta account*. When a user creates an account or logs in within the decentralized application, rather than being hard coded into the protocol, signature verification is implemented through a dynamic request transmitted to a user’s account. This enables transaction verification with arbitrary logic and state, establishing XION as a future-proof, curve-agnostic protocol. By utilizing arbitrary logic for transaction verification, XION is not limited to any specific verification schema. As a result, XION seamlessly supports various cryptographic curves without requiring specific adaptation or modification. This stems from XION’s unique approach to abstraction, which separates the underlying cryptographic mechanisms from the user-facing interfaces, thereby enabling the system’s seamless adoption of different cryptographic curves.

XION’s Generalized Abstraction layer supports a wide spectrum of curves, including well-established ones like Ethereum’s `Secp256K1` and Solana’s `Ed25519`, alongside such widely adopted standards as `Secp256R1` and `RS256` used by JSON Web Tokens (JWTs). In addition, it is able to support any new future cryptographic curves as well, ensuring it can adapt to emerging trends and developments. Developers building on XION can trust that their smart contract accounts will remain compatible and adaptable as cryptography continues to evolve quickly.

3.3 Device Abstraction

Combining the implementations above, XION abolishes the need for a user to store and manage private keys. This approach eliminates the safety risk, complexity, and friction traditionally present when users attempt to use their accounts across multiple devices. Consequently, XION’s unique approach renders it device-agnostic as users can interact with their accounts seamlessly across various devices, including computers, smartphones, or tablets. This universally accessible architecture significantly streamlines the user experience, fostering widespread adoption by reducing entry barriers and enhancing ease-of-use of all applications accessible from XION.

When interacting with apps through XION, a user is presented with multiple login methods including email, social accounts, FaceId, or for advanced users the option to log in with Web3 credentials such as Keplr or MetaMask wallets. XION is therefore able to cater to all audiences, while retaining seamless user experiences for non-crypto native users. In addition, its *meta accounts* enable additional security, flexibility, and ease-of-use such as multiple-factor authentication, session keys, key rotation, and more. In short, XION’s protocol level account abstraction combined with its signature agnostic implementation enables users to seamlessly access their accounts across multiple devices in a safe and frictionless manner.

3.4 Parameterized Fee Layer

To enable many of the functionalities necessary to create a seamless user experience, XION takes a new approach to the handling of fees on the network which includes 1) implementing global fee abstraction, 2) instituting PlatformSend, a new type of fee, and 3) combining them with the use of FeeGrant.

3.4.1 Fee Abstraction

XION allows the use of any token for transaction fees through its global fee abstraction. This is achieved by accumulating the transaction fees paid by users in

the fee collector, swapping these non-native tokens for the native XION token, and distributing the resulting native tokens back to the fee collector. As covered in the subsequent Section 3.5, the resulting native tokens in the fee collector are then utilized to determine the chain’s inflation.

There are a variety of ways for the collected fees to be swapped for the native XION token, opening up interesting fee market possibilities. One implementation is such that the exchange rate is determined by using a time-weighted average price (TWAP) by periodically retrieving data from the desired decentralized exchange (DEX). In Figure 2, it is demonstrated using the Asynchronous Interchain Query Module (Async-ICQ)¹⁴.

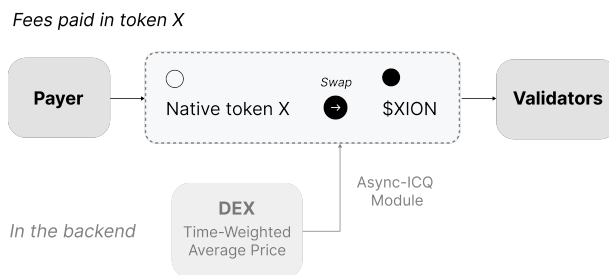


Figure 2: Fee Abstraction

3.4.2 PlatformSend & FeeGrant

Transactions on the network can be grouped into two major categories, 1) value is exchanged, 2) no value is exchanged. PlatformSend is enacted when transactions from the first category occur on the network. Since value is exchanged between participants, a portion of that value is taken as a fee to support the network and prevent sybil attacks. Synonymous to the way gas fees mitigate network attacks, the PlatformSend fee is a cost barrier that requires an attacker to pay proportionally to the value exchanged. This elegant design, demonstrated in Figure 3 allows users to transact directly in any currency of choice, without requiring gas tokens. Then, utilizing the global fee abstraction outlined in

Section 3.4.1, the fees collected are swapped with the native token in order to determine network inflation and compensate network participants.

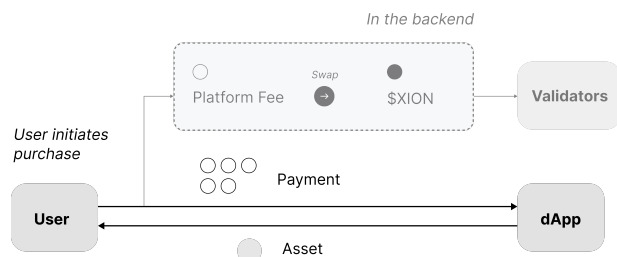


Figure 3: PlatformSend

When transactions occur from the second category, regular gas fees are charged in order to prevent network spam, and for these types of transactions XION implements the FeeGrant. As demonstrated in Figure 4, this allows developers to seamlessly sponsor transactions on behalf of users and provide a gasless experience. In addition, sponsoring can be done in a variety of ways such as utilizing staking rewards, employing a minimum transaction threshold, or other configurable parameters.

User initiates transaction

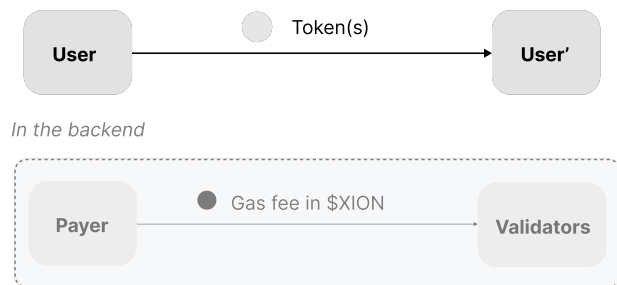


Figure 4: Fee Sponsoring

3.4.3 Result: Pricing & Payment Abstraction

Through Generalized Abstraction’s parameterized fee layer mentioned above, XION is the first

blockchain able to use USDC, a fully-reserved digital dollar, as its primary transaction currency¹⁵. Figure 5 shows the dynamics of USDC fees in the fee collector being auto-swapped to the native token, enabling all products built on XION to be denominated and paid for by users simply using USDC. This eliminates the traditional friction of acquiring gas tokens and allows users to seamlessly onboard with familiar pricing, while simultaneously reducing unwanted volatility and speculation. In addition, through its parameterized fee layer, XION is natively interoperable with every connected ecosystem, as users have the ability to pay in any token of choice. Combined with the end-users’ gasless experiences, XION’s Generalized Abstraction stands out in its ability to reduce friction and drive mainstream adoption of Web3.

In the backend of a TX

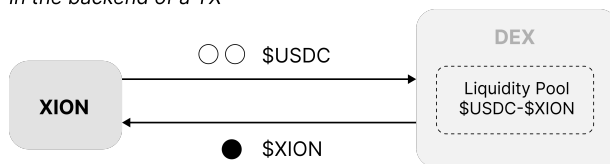


Figure 5: USDC Fee Swap

3.5 Token Minting Dynamics

The macroeconomic dynamics of a network are shaped by the nuances of its token minting, and these minting parameters are of utmost importance for the long term viability of the system. Closely tied to Generalized Abstraction’s parameterized fee layer, XION’s design takes a first-principles approach to token minting dynamics in order to balance network participant incentives and long term sustainability. XION makes two substantial changes to the Mint Module:

- Accumulated fees are used to offset inflation, when possible.
- Token inflation is calculated against staked tokens only.

3.5.1 Mint Module Overview

The mint module's goal is to maintain the pre-determined ratio of staked vs liquid tokens in an ecosystem. This is achieved by setting three values: the upper bound for inflation, the lower bound for inflation, and the target staking ratio. Once the inflation rate is determined, the module will then calculate how many tokens need to be generated in a block in order to meet the inflation percentage. It is an important balance to strike between market liquidity and staked supply, as it impacts the long-term security and growth of the network, and the formulas below demonstrate this balance.

If $P_{bonded} < P_{goal}$: $IR = \min(IR + \Delta IR, IR_{max})$
In this scenario, ΔIR is the rate at which the inflation increases when P_{bonded} is below P_{goal} .

If $P_{bonded} = P_{goal}$: $IR = \text{constant}$
In this scenario, when the bonded percentage is at the goal, the inflation rate doesn't change.

If $P_{bonded} > P_{goal}$: $IR = \max(IR - \Delta IR, IR_{min})$
In this scenario, ΔIR is the rate at which the inflation decreases when P_{bonded} exceeds P_{goal} .

Above, IR is the current inflation rate. P_{bonded} is the current percentage of staked tokens. P_{goal} is the target percentage for staked tokens. IR_{max} is the maximum allowable inflation rate. IR_{min} is the minimum allowable inflation rate.

These formulas effectively capture the dynamic adjustment of the inflation rate based on the relationship between the current percentage of bonded tokens and the target bonded percentage, along with the constraints of maximum and minimum inflation rates.

3.5.2 Offsetting Inflation With Fees

The default mint module currently takes fees accumulated in a block and passes them to validators, on top of the tokens that get minted every block. This

is represented as

$$T_{net} = T_{minted} + F_{collected}$$

where T_{minted} is $T_{total} \times IR$, and T_{total} is the current total supply of tokens.

The first change XION has implemented is to use the fees that have been collected in a block, and offset how many tokens need to be minted in the block by these accumulated fees. This is demonstrated as

$$T_{net} = T_{minted} - F_{collected}$$

If $F_{collected}$ is significant enough, it can reduce T_{net} , thus effectively decreasing the inflationary impact by using the fees collected to offset the newly minted tokens.

As chain usage increases, this will result in inflation decreasing as the accumulation of fees continue to offset the minting of new tokens. When fee accrual exceeds the number of tokens per block necessary to distribute to stakers, then the extra tokens are burnt and the token supply is thereby reduced, naturally becoming deflationary.

3.5.3 Token Inflation on Staked Tokens

Regarding token inflation, the default mint module calculates inflation against all existing tokens. This can be represented as:

$$T_{minted} = IR \times T_{total}$$

where T_{minted} is the total tokens to be minted, IR is the inflation rate, and T_{total} is the total token supply.

As a result, annual percentage yield (APY) for stakers can be represented as

$$APY = IR \div P_{bonded}$$

where $APY \geq IR$ as a function of P_{bonded} . This often leads to a higher realized staking annual percentage yield (APY) than the inflation rate, since not all tokens are staked.

XION’s mint module implements a significant change. Instead of calculating against all tokens, it calculates against staked tokens.

$$T_{minted_x} = IR \times T_{bonded}$$

Since T_{bonded} is less than T_{total} , the total inflationary impact T_{minted_x} will be less than or equal to T_{minted} for the same IR .

The result is that inflationary impacts will be reduced and stakers will always see $APY = IR$. This change creates a much more stable environment, ensuring the safety of the network by helping to attract long-term participants.

4 Abstracted Interoperability

XION’s abstracted interoperability, a key aspect of its Generalized Abstraction, enhances the cross-chain user experience significantly. This is achieved through the use of package forwarding middleware¹⁶¹⁸ that enables users to perform actions on any host chain from a controller chain, such as XION. By integrating this middleware with the existing Generalized Abstraction framework, a streamlined protocol-level interface is created. As demonstrated in Figure 6, it allows users to perform actions on any chain that establishes a channel with XION, all while enjoying XION’s seamless user experiences.

XION’s approach to interoperability addresses the common issue of account fragmentation in multi-chain environments. A user can link accounts that they own across multiple chains to their XION *meta account*, offering them a seamless way to manage their assets all from one central account. Each account has a unique identifier, combining the source chain’s identifier with the original account’s address, ensuring a distinctive mapping across connected chains. Users thereby maintain control over their multi-chain accounts via a single XION interface.

XION achieves this by leveraging symmetrical communication through several implementation

User operates action A and action B on XION

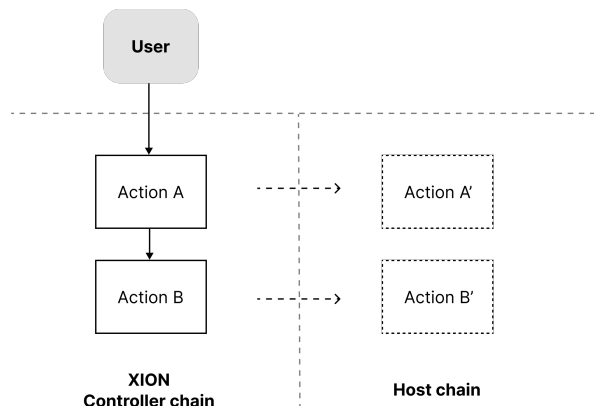


Figure 6: Multi-Interaction to Host Chain

methods¹⁷¹⁸. It establishes reliable, ordered, and authenticated communication channels between separate blockchains. A package forwarding middleware component is then instrumental in providing practical user account control and management on different chains through the abstraction of account interfaces, enabling seamless interaction with any chain connected to XION as seen in Figure 7.

Combined with XION’s Generalized Abstraction layer, *abstracted interoperability* extends all of its protocol-level abstractions, facilitating new cross-chain possibilities and seamless user experiences previously unattainable.

5 Generalized Abstraction Applications

XION’s Generalized Abstraction layer enables both vastly enhanced user experiences for existing applications, as well as enabling a range of novel ones. Below, we start by providing a few non-exhaustive examples of use-cases made possible by Generalized Abstraction on XION. We then provide additional examples of applications able to leverage abstracted interoperability to enable innovative cross-chain use-cases.

User operates action A on XION

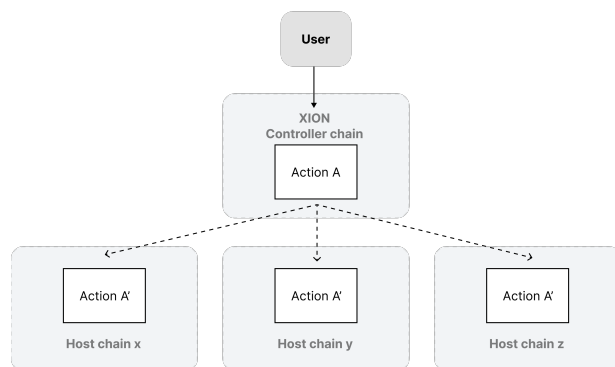


Figure 7: Multi-Chain Interactions from XION

5.1 XION Enabled Use-Cases

5.1.1 Digital Banking

A digital banking application leverages Generalized Abstraction to allow customers to set temporary session keys for limited-time access, ensuring higher security for transactions while allowing the users to also define their own transaction limits and conditions. These users can also set up multiple authentication method requirements for large transactions, as well as the ability to recover account information should they lose access to certain authentication methods.

5.1.2 Global Venmo Messenger

A decentralized messaging service leverages Generalized Abstraction to enable users to safely access their messaging chats seamlessly with the same account, whether they're using a smartphone, tablet, or desktop. Users are able to transact globally, sending assets cross-border through seamless gasless transactions directly within the messaging app using familiar fiat denominations.

5.1.3 Creator Economy Applications

A decentralized content streaming service leverages Generalized Abstraction to create subscription-based accounts, where users are automatically charged monthly through smart contract triggers without

needing manual renewals. Creators and their fans of all ages are able to frictionlessly create these accounts, and access the content from all their devices.

5.1.4 Enhanced Gaming Experiences

An on-chain game leverages Generalized Abstraction to enable the seamless use of session keys, batching of transactions, and gasless transactions to enable smooth and secure gameplay without endangering the user's assets, all while reducing latency.

5.1.5 User-friendly DAOs

An online collaborative platform leverages Generalized Abstraction to set up decentralized organizations where members have different permission levels, enabling non-technical users to participate in governance or decision-making processes through intuitive, familiar Web2 interfaces.

5.1.6 Cloud Storage Services

A decentralized cloud storage platform leverages Generalized Abstraction to enable a family to seamlessly access their content across multiple devices. Through the use of account permissions, family members have different levels of access and editing privileges.

5.2 Abstracted Interoperability Extensions

5.2.1 Globally Connected Applications

Through abstracted interoperability, XION is able to tap into all existing Web3 applications. For example, a user holds funds on the Base network from their usage of the FriendTech app. This user now wants to use StepN, a fitness app on Solana, and fund their purchase of digital shoes using their FriendTech funds. Using XION, they can seamlessly interact and use StepN through their XION *meta account* which controls accounts on both Base and Solana. In the background, the user's Base funds are transferred, converted, and used to purchase the digital shoes on a Solana marketplace. For the end user,

it's a straightforward, seamless experience, showcasing XION's ability to make complex cross-chain interactions user-friendly.

5.2.2 Global Decentralized Marketplaces

Users can participate in decentralized marketplaces spanning multiple blockchains, buying and selling goods or services with various cryptocurrencies, all managed through a single account with unified security policies and cross-chain payment capabilities by leveraging Generalized Abstraction's abstracted interoperability.

5.2.3 Cross-Chain Gaming

Through abstracted interoperability, gamers can engage in cross-chain gaming experiences where assets and achievements in one blockchain game can be used or traded in games on other chains, all under one unified gaming account. This gaming account can be easily accessed from all devices, given different levels of access controls, and safety measures can be implemented to eliminate the possibility of account loss.

5.3 DAOs

A decentralized autonomous organization (DAO) can leverage Generalized Abstraction and abstracted interoperability to seamlessly operate across multiple blockchains. This allows it to effectively tap into different applications, a wider user-base, more flexible and efficient governance, all made possible through common Web2 methods for mainstream user engagement.

5.4 Global Profiles

A cross-chain identity verification system can be established, whereby users are able to interact across multiple blockchains using the same credentials and reputations. This enables a myriad of possibilities as recognizable reputations allow users to access a wide range of services through one verifiable identity, bringing all ecosystems closer together.

6 Conclusion

XION addresses key challenges that have impeded mainstream adoption through its Generalized Abstraction layer. XION simplifies the user experience, making blockchain technology more accessible to a broader audience. It removes the complexities associated with account creation, interaction, transaction fees, and interoperability, replacing them with familiar interactions. In addition, it further extends this seamless user experience to encompass cross-chain interactions. Generalized Abstraction not only enhances the XION ecosystem's capabilities but also contributes to the overall growth of the blockchain industry by combining seamless connectivity with widespread adoption. Through a multifaceted Generalized Abstraction layer, XION offers a sustainable, flexible, and user-centric blockchain infrastructure paradigm, capable of propelling the industry towards a new era of innovation and widespread acceptance.

References

1. V. Buterin, and Y. Weiss. “ERC-4337: Account Abstraction Using Alt Mempool [Draft].” 29 Sept. 2021. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-4337>
2. N. Mudge, “EIP-2535: Diamonds, Multi-Facet Proxy.” Ethereum Improvement Proposals, Ethereum Foundation. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-2535>.
3. F. Giordano, et al., “EIP-1271: Standard Signature Validation Method for Contracts.” Ethereum Improvement Proposals, Ethereum Foundation. [Online]. Available: <https://eips.ethereum.org/EIPS/eip-1271>.
4. P. Daian, S. Goldfeder, et al., “Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges,” in 2020 IEEE Symposium on Security and Privacy (SP), 2020, pp. 914-931.
5. V. Buterin, “A next-generation smart contract and decentralized application platform,” Whitepaper, 2014. [Online]. Available: <https://ethereum.org/en/whitepaper/>.
6. D. Johnson, A. Menezes, & S. Vanstone, “The Elliptic Curve Digital Signature Algorithm (ECDSA).” *IJIS* 1, 36–63 (2001). [Online]. Available: <https://doi.org/10.1007/s102070100002>
7. I. Liusvaara, and S. Josefsson. “RFC 8032 Edwards-Curve Digital Signature Algorithm (EdDSA).” Internet Engineering Task Force, Jan. 2017. [Online]. Available: <https://rfc-editor.org/rfc/rfc8032>.
8. S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System.” Bitcoin. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>.
9. A. Yakovenko, “Solana: A new architecture for a high-performance blockchain v0.8.13.” Solana Foundation. [Online]. Available: <https://solana.com/solana-whitepaper.pdf>.
10. Apple Inc., “Secure Enclave,” Apple Support Guide. [Online]. Available: <https://support.apple.com/guide/security/secure-enclave-sec59b0b31ff/web>
11. A. Hamlin, “Secure Multiparty Computation.” Comp 116, Tufts University. [Online]. Available: www.cs.tufts.edu/comp/116/archive/ahamlin.pdf.
12. J. Kwon, E. Buchman, “A Network of Distributed Ledgers.” Tendermint. [Online]. Available: <https://v1.cosmos.network/resources/whitepaper>.
13. L. Lyu, “abstract-account,” GitHub repository. [Online]. Available: <https://github.com/larry0x/abstract-account>
14. A. Raja, E. Saradar, “Asynchronous Interchain Query Module (Async-ICQ),” GitHub repository, August 30, 2022. [Online]. Available: <https://github.com/strangelove-ventures/async-icq>
15. J. Allaire, S. Neville, “Introducing USD Coin (USDC) a Fully Reserved Stablecoin,” Circle Blog, Sept. 26, 2018. [Online]. Available: <https://www.circle.com/blog/introducing-usd-coin>
16. T. Yun, S. King, J. Lee, “Cosmos/ibc ICS-027: Interchain Accounts Specification,” GitHub repository, [Online]. Available: <https://github.com/cosmos/ibc/tree/main/spec/app/ics-027-interchain-accounts>
17. C. Goes, “The Interblockchain Communication Protocol: An Overview,” Interchain GmbH. [Online]. Available: <https://arxiv.org/pdf/2006.15918.pdf>
18. Abacus Works, “Hyperlane v3 Docs,” Abacus Works, 2023. [Online]. Available: <https://docs.hyperlane.xyz/>