TAC - Litepaper

June 2024

Anton Bryantsev
anton@tac.build

Sergey Lepesev
sl@tac.build

Sergey Polyansky
sp@tac.build

Vitaly Bakatov
vb@tac.build

Pavel Altukhov
pa@tac.build

Marco Monaco
marco@tac.build

Abstract

TAC is the first-of-its-kind EVM Network Extension for TON. It's an advanced Layer 1 Proof-of-Stake (POS) blockchain built using CosmosSDK and Ethermint [1] [2] that is natively integrated with TON [3] using a Native Integration Bridge. Designed as a compatibility layer that connects the TON blockchain with Ethereum Virtual Machine (EVM) [4], TAC facilitates direct access to EVM-compatible applications in a secure, efficient, and user-friendly manner. It introduces innovative features like Proxy Apps and an additional decentralised message sequencer network to manage cross-chain transactions seamlessly. TAC is not parasitic to TON, but extends TON features with EVM equivalence: TAC transaction lifecycle starts and ends on TON and fees are paid with TON tokens, contributing to the TON network growth. TAC does not fragment TON users, liquidity or developers.

What is TAC?

TAC is the combination of a bridge and a fully-fledged Layer 1 blockchain that allows Ethereum-compatible dApps to deploy Solidity code on the TAC EVM chain that can be triggered natively with transactions happening on the TON Network. This approach ensures that these dApps can expose themselves to TON's massive user base without requiring modifications to TON's architecture. TAC, on the TON side, leverages its own set of smart contracts and sequencer-based consensus mechanisms to manage and validate cross-chain transactions between the two ecosystems.

How TAC Works

TAC employs the concept of TON Proxy Apps, which are proxy smart contracts deployed on both TON and EVM. These proxies enable TON users (or other TON dApps) to interact with EVM-based applications directly from their native TON wallets. Here's a typical transaction scenario:

1. A user initiates a transaction using their TON wallet, connecting to a specific Proxy App.
2. The Proxy App constructs a cross-chain transaction and transfers assets to the bridge EVM wallet smart contract (part of the execution process).

3. The bridge EVM wallet communicates with the messaging layer, where sequencers collect messages and form Merkle trees to manage transaction sequencing and validation.
4. Once the Merkle tree is verified, the transaction is executed on the EVM side, and assets are transferred between chains, completing the operation.

Key Components of TAC

TAC EVM Layer

TAC's architecture includes a Layer 1 EVM chain secured by dPoS and built with CosmosSDK and Ethermint where EVM-compatible decentralised applications are deployed. This layer relies on existing battle-tested technologies like Tendermint Core [5], Geth Client and IBC [7]. The execution environment provides EVM equivalence, including non-Layer2 related OP CODES introduced up to Shanghai Fork, like PUSH0 and EIP-1559 for gasfees management. The TAC EVM Layer relies on an inflationary native token that is used for both gasfees and proof-of-stake security mechanism. This layer will be secured with Babylon Bitcoin Staking [6] once this will be publicly available.

TAC Native Integration Bridge

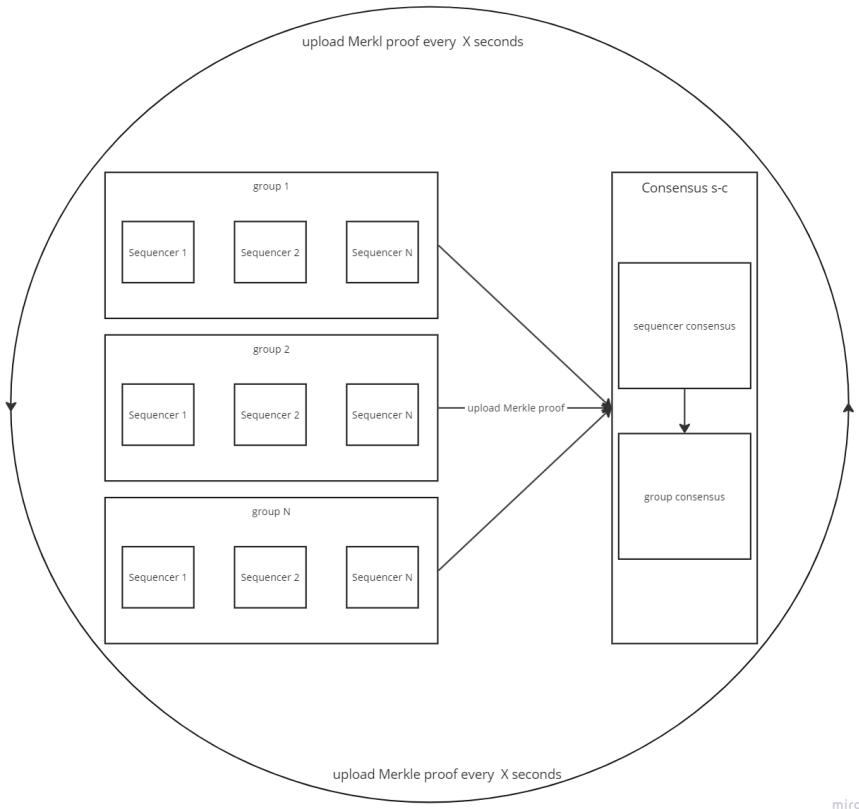
TAC operates using a set of smart contracts on both the TON and EVM sides to facilitate operations such as minting, unlocking, and consensus processes. These layers manage data availability, elections, and cross-chain message execution.

TAC Bridge Sequencers

The sequencer network is a decentralized, off-chain component that uses POS consensus to manage transaction sequencing on the cross-chain layer. Sequencers form transaction trees using Merkle proofs, ensuring accurate and efficient cross-chain messaging. The sequencers' responsibilities include:

- **Indexing events**: Collecting and processing log messages from the TON blockchain and events from the EVM side.
- **Cross-chain execution**: batched or parallel processing of events
- **Consensus formation**: Achieving consensus on transaction trees before execution.

Sequencers are grouped into Sequencer Groups, which enhance security and facilitate efficient transaction validation.



Proxy Apps

Proxy Apps are lightweight proxy smart contracts deployed on both the TON and EVM sides. They serve as intermediaries, handling calls between the bridge layer and dApps on TAC, as well as between the bridge layer and TON wallets.

TAC Transaction Lifecycle

The transaction lifecycle in TAC involves multiple stages to ensure security, efficiency, and reliability. Here's an overview:

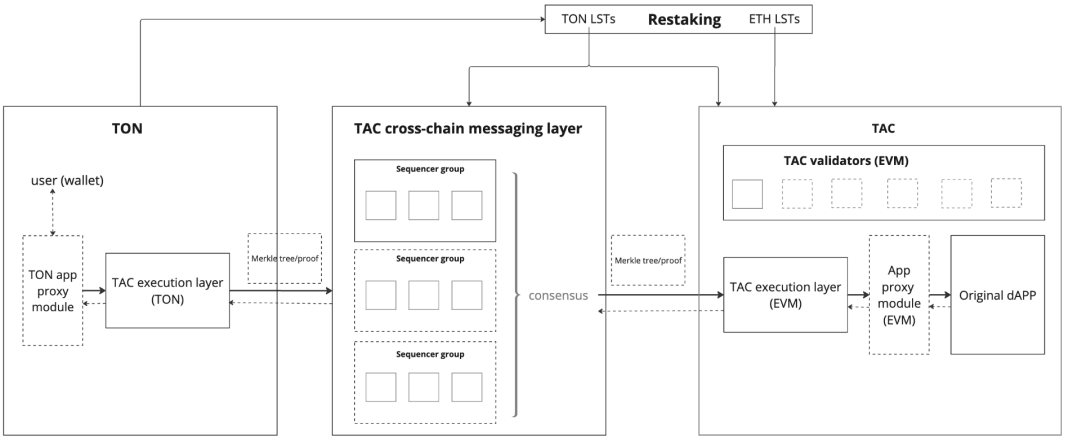
1. **Origination:** A dApp FrontEnd, using the TAC SDK, prepares and builds a TON transaction targeting a specific Proxy App on TON.
2. **Initialization**: Proxy App sends assets with parameters for cross-chain operations. A log message is generated on the TON side, while an Action event is triggered on the EVM side.
3. **Event Detection**: Sequencers detect and index new events, storing them in a local database.
4. **Root Hash Formation**: Periodically, sequencers form Merkle trees, resulting in a root hash used for consensus.
5. **Consensus**: Sequencer groups must achieve consensus on the transaction tree. If 66% of the sequencers within a group agree, the tree is submitted for further validation.
6. **Execution**: Once consensus is reached, transactions are executed, assets are transferred, and smart contract methods are called. Transaction fees are paid to the EVM Layer PoS validators involved in the validation process.
7. **Transaction Rollback**: In the event of failure (e.g., insufficient gas), the system allows for

transaction rollbacks, ensuring no funds are lost.

8. **Commission Distribution**: Transaction fees for Cross-Chain operations are distributed among sequencers based on their performance and collateral contributions (TON/TON LSTs).

High-Level Architecture

1. **TAC EVM Layer:** Self contained EVM chain built with CosmosSDK + Ethermint and secured by delegated Proof of Stake. Used to store the global state of dApps and assets deployed on TAC.
2. **Cross-Chain Layers**: Smart contracts on both TON and EVM sides manage core operations such as asset minting and unlocking, consensus, and elections.
3. **TAC Bridge Sequencers and Consensus Mechanism**: Sequencers form a decentralized network that constructs Merkle trees from transaction logs. Sequencer groups must reach a consensus on these trees, ensuring the security and efficiency of cross-chain transactions.
4. **Proxy Apps**: These proxies facilitate seamless communication between TON wallets and EVM-based applications, enabling users to interact with Solidity code deployed on TAC EVM Layer.
5. **Execution and Validation Layers**: The execution layer manages transaction processing, while the validation layer ensures that transactions meet all security and consensus requirements.



Security and Validation

To ensure robust security, TAC implements a multi-layered validation and consensus mechanism:

- **EVM Validation:** TAC EVM Layer requires specialised operators to form consensus around EVM state updates. TAC manages this relying on Tendermint Core, with delegated Proof of Stake.
- **Bitcoin Staking**: To enhance the security of the TAC EVM Layer, TAC will rely on Babylon Bitcoin Staking solution once available. This will provide a high degree of security on top of the faster consensus derived from Tendermint Core. Specialised

nodes in the Babylon Network will sign TAC blocks and stake Bitcoin to ensure that no double-signing happens (chain reorgs, double spends, long-range attacks).

- **Bridge Sequencer Groups**: These are structured to enhance security by forming transaction trees that are verified through Merkle proofs. Only when consensus is reached among sequencer groups is the transaction tree uploaded for execution on the EVM Layer.
- **Bridge Collateral and Slashing**: Bridge Sequencers must lock collateral to participate in transaction validation. If a sequencer provides incorrect data, slashing is imposed, and collateral may be forfeited to maintain network integrity.

Core Benefits of TAC

1. **For Users**: TAC provides TON users with access to EVM-compatible dApps without leaving the Telegram ecosystem. Users benefit from seamless, low-cost transactions and a wide array of battle-tested decentralised applications, ranging from DeFi primitives to GameFi and SocialFi.
2. **For Developers**: TAC offers a high-throughput environment for deploying EVM-based applications and exposing them directly to the TON userbase. Developers can use familiar Ethereum tools to build on TAC, expanding their reach to TON's extensive user base.
3. **Enhanced Security**: The POS consensus model, combined with sequencer-based transaction validation, ensures secure and efficient cross-chain operations. The EVM Layer is coming from battle-tested and proven technology, enhanced by the use of Bitcoin Staking.

Conclusion

TAC is the first-of-its-kind EVM Network Extension for TON: a fully-fledged Layer 1 blockchain that connects TON userbase and EVM dApps, empowering TON users and developers with seamless access to battle-tested decentralised applications and giving developers the tools they need to build secure, scalable solutions. By integrating CosmosSDK and Ethermint, TAC delivers a high-throughput, low-cost blockchain infrastructure that extends TON without any parasitic approach, paving the way for a rapid growth of the TON ecosystem.

References

[1] Buchman, Ethan. "Cosmos: A Network of Distributed Ledgers." 2016.
[2] Kwon, Jae, and Ethan Buchman. "Cosmos and Ethermint: A Brief Overview." 2017.
[3] Durov, Nikolai. "Telegram Open Network." 2019.
[4] Wood, Gavin. (Yellow Paper) "Ethereum: A Secure Decentralised Generalised Transaction Ledger." 2014
[5] Kwon, Jae. "Tendermint: Consensus without Mining." 2014.
[6] Wang, Haifeng, et al. "Babylon: Bitcoin-Empowered Interoperability for Proof-of-Stake Blockchains." 2023.
[7] Christopher Goes. "The Interblockchain Communication Protocol: An Overview" 2020.

Disclaimer