Apertum

Security Assessment

CertiK Assessed on Dec 9th, 2025

CertiK Assessed on Dec 9th, 2025

Apertum

The security assessment was prepared by CertiK.

Executive Summary

TYPES	ECOSYSTEM	METHODS
Layer 1	Avalanche (AVAX)	Manual Review

LANGUAGE
Other

TIMELINE
Preliminary comments published on 12/05/2025.

Apertum is implemented as a sovereign Avalanche Layer-1 (Subnet) built entirely on the unmodified Avalanche consensus and validator framework.

The audit confirms that Apertum operates on the Avalanche framework with no modifications to core consensus or protocol logic, inheriting Avalanche's security and decentralization properties. No findings impact protocol integrity.

Vulnerability Summary

0	0	0	0	0
Total Findings	Resolved	Partially Resolved	Acknowledged	Declined

0	Centralization	Centralization findings highlight privileged roles & functions and their capabilities, or instances where the project takes custody of users' assets.
0	Critical	Critical risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
0	Major	Major risks may include logical errors that, under specific circumstances, could result in fund losses or loss of project control.
0	Medium	Medium risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform.
0	Minor	Minor risks can be any of the above, but on a smaller scale. They generally do not compromise the overall integrity of the project, but they may be less efficient than other solutions.
0	Informational	Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

TABLE OF CONTENTS | APERTUM

INTRODUCTION: OBJECTIVE & SCOPE | APERTUM

The primary objective of this technical review is to document and verify the `Apertum` 's on-chain parameters and operational status within the `Avalanche ecosystem` , in response to the Apertum team's service request to validate that Apertum operates as an Avalanche Subnet.

This report goes beyond theoretical documentation by presenting Apertum's deployment data: first establishing a technical baseline via the `avalanche-cli` framework, and then performing on-chain queries through direct interactions with the Avalanche P-Chain, Glacier API, and Apertum's EVM endpoints.

To ensure data consistency, these raw findings are cross-checked with the official **Avascan** explorer and third-party registries like **Chainstack**, documenting Apertum's registration and presence within the broader Web3 ecosystem.

For this analysis, we assume that all information obtained from the endpoints provided by the Apertum team is accurate. The data sources are as follows:

1. Avalanche Subnet Explorer: https://subnets.avax.network/subnets/2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto
2. Avalanche Explorer: https://avascan.info/blockchain/apertum/info
3. Chainstack Subnet Listing: https://github.com/chainstacklabs/subnetlist/pull/28
4. Ethereum Chain Registry Commit: https://github.com/ethereum-lists/chains/commit/798b7819844b87194d9c12e6bda306aa5f7857b1
5. ChainID Network: https://chainid.network/chain/2786/
6. Chainlist.wtf: https://chainlist.wtf/chain/2786/
7. Networks.vercel.app: https://networks.vercel.app/
8. EVM Chain Info: https://evmchain.info/chain/2786/
9. EVMChainList.org: https://evmchainlist.org/?search=2786
10. Chainlist.simplr.sh: https://chainlist.simplr.sh/
11. Ethereum Chains Master List: https://github.com/ethereum-lists/chains
12. Apertum Node RPC: https://rpc.apertum.io/ext/bc/YDJ1r9RMkewATmA7B35q1bdV18aywzmdiXwd9zGBq3uQjsCnn/rpc

PART 1: BUILDING ON AVALANCHE - ARCHITECTURE | APERTUM

Avalanche Infrastructure

To understand the technical positioning of specific projects like **Apertum**, one must first grasp the underlying infrastructure of the Avalanche network. Avalanche is not merely another Layer-1 blockchain; it is architected as a Layer-0 Protocol or a Platform of Platforms.

At its core, Avalanche solves the blockchain scalability trilemma through a novel consensus mechanism based on Repeated Random Subsampling. Unlike PoW (Proof of Work) or traditional BFT (Byzantine Fault Tolerance) which can be slow or unscalable, Avalanche achieves sub-second finality and supports thousands of validators. This creates a highly decentralized environment where transactions are confirmed almost instantly, providing the bedrock for high-performance applications.

Primary Network

The foundation of the ecosystem is the Primary Network, secured by the global validator set. Instead of forcing all activity onto a single chain (which causes congestion and high fees), the Primary Network utilizes a multi-chain framework to separate concerns:

- **P-Chain (Platform Chain):** This is the governance and coordination layer. It implements the Snowman consensus protocol to manage the validator set and track active Subnets. Critically, This is where custom blockchains (like Apertum) are registered and where their security models are defined.
- **C-Chain (Contract Chain):** An implementation of the Ethereum Virtual Machine (EVM) running on the high-speed Snowman consensus engine. It serves as the general-purpose smart contract layer, allowing developers to deploy Solidity contracts with significantly higher throughput than Ethereum.
- **X-Chain (Exchange Chain):** A DAG-based (Directed Acyclic Graph) chain optimized for the parallel processing of transactions, specifically for the creation and exchange of digital assets.

Subnets and Avalanche L1s

The core innovation relevant to our review is the **Subnet** (Subnetwork). A Subnet is a dynamic set of validators working together to achieve consensus on the state of a set of blockchains.

While early Subnets were often static and permissioned, the ecosystem is evolving toward **Avalanche L1s** (formerly known as Elastic Subnets). An Avalanche L1 is a sovereign blockchain that offers:

1. **Infinite Horizontal Scalability:** Unlike Layer-2 rollups that still depend on L1 block space, an Avalanche L1 adds capacity to the network without competing for Primary Network resources.
2. **Custom Economic Models:** L1s can use their own native tokens for gas fees instead of AVAX, capturing value directly for their ecosystem.
3. **Sovereign & Permissionless Validation:** By utilizing PoS Precompiles (specialized smart contracts), L1s can manage their own staking logic, allowing anyone to become a validator without manual approval from a central authority.

Builder's Toolkit: avalanche-cli

For developers and researchers, interacting with this complex infrastructure requires specialized tooling. `avalanche-cli` is the official command-line interface designed to orchestrate the lifecycle of an Avalanche L1. It abstracts the complexity of running nodes and interacting with the P-Chain.

In this section, we break down the standard workflow for building a Subnet. Understanding this process establishes the technical baseline for analyzing live projects like Apertum.

Phase 1: Blockchain Configuration

The first step in creating an L1 is defining its specifications.

Command: `avalanche blockchain create <BlockchainName>`

Technical Analysis: This command generates the `genesis.json` file. This file is the immutable "DNA" of the blockchain and includes:

- **Virtual Machine (VM) Selection:** Most L1s, including Apertum, utilize the - **Subnet-EVM**. This is a modified version of the `go-ethereum` (geth) EVM, stripped of PoW consensus and adapted for the high-speed Snowman consensus engine.
- **Chain ID:** A unique identifier (e.g., `2786`) to prevent replay attacks across different EVM networks.
- **Initial Allocation:** Defines the distribution of native tokens at the genesis block (e.g., airdrops to admin addresses).

Phase 2: Deployment and Orchestration

Once configured, the blockchain must be deployed to a network (Local, Fuji Testnet, or Mainnet).

Command: `avalanche blockchain deploy <BlockchainName> --local`

When a developer runs this command, `avalanche-cli` performs several critical operations that leave on-chain evidence:

1. **P-Chain Interaction:** It issues a transaction to the P-Chain to create a new **Subnet ID**.
2. **Chain Initialization:** It issues a `CreateChain` transaction, associating the new Subnet ID with the `genesis.json` and the VM binary.

PART2: ON-CHAIN VERIFICATION OF APERTUM | APERTUM

Verification Strategy

In Part 1, we established how Avalanche L1s are constructed. In this section, we shift from theory to practice. Our objective is to verify that **Apertum** exists as a registered Subnet on the Avalanche Mainnet and to extract its technical configuration directly from the chain.

To ensure the integrity of our data, we bypassed third-party explorers and interacted directly with the Avalanche network.

We utilized `curl` to issue HTTP requests directly to Avalanche's API endpoints. Our methodology follows the official technical standard outlined in Ava Labs' guide on **[Issuing API Calls]**. This documentation defines the JSON-RPC 2.0 structure required to communicate with the node validators.

To establish a comprehensive chain of evidence, we will also perform a cross-verification step. We will compare our raw on-chain findings against the public data indexed by the official Avalanche Explorer on **[Avascan]**, ensuring that the node-level data aligns perfectly with the publicly recognized network state.

Basic Registration (P-Chain RPC)

Our first query targeted the P-Chain (Platform Chain). This is the "root" ledger where all Subnets must be registered. We used the `platform.getSubnets` method to show if the Apertum Subnet ID is valid.

The Query

We issued a standard POST request to the P-Chain endpoint (`/ext/bc/P`):

```
curl -X POST --data '{
    "jsonrpc":"2.0",
    "id"     :1,
    "method" :"platform.getSubnets",
    "params" :{
        "ids": ["2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto"]
    }
}' -H 'content-type:application/json;' https://api.avax.network/ext/bc/P
```

Where `2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto` is the claimed subnet ID of Apertum.

The Output

The node returned the following raw data:

```
{
  "jsonrpc": "2.0",
  "result": {
    "subnets": [
      {
        "id": "2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto",
        "controlKeys": [
          "P-avax138usvdatdk6syvjzkwmla9n39535zex8t0kr9m"
        ],
        "threshold": "1"
      }
    ]
  },
  "id": 1
}
```


Analysis of Findings

- **Existence:** The non-empty subnets array shows that ID `2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto` is present on the Avalanche Mainnet.
- **Governance & Ownership:** The threshold: `1` combined with the single address in controlKeys (`P-avax138usvdatdk6syvjzkwmla9n39535zex8t0kr9m`) indicates a Single-Sig governance model. This specific address holds the root administrative rights for the Subnet on the P-Chain, serving as the ultimate owner responsible for the network's lifecycle.

The presence of a Subnet owner key reflects administrative control at the Subnet-registration layer and does not modify or influence base Avalanche consensus logic.

This raw data aligns with the official network information indexed on **Avalanche Subnet Explorer**, showing the consistency between the node-level state and the public explorer.

Advanced Configuration (Glacier API)

While the P-Chain RPC shows the existence, it lacks architectural details. To understand what kind of Subnet Apertum is, we queried the **Glacier API**, Ava Labs' high-performance indexer service.

The Query

We executed a GET request targeting the specific Subnet ID **`2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto`**:

```
curl -X 'GET' \
  'https://glacier-
api.avax.network/v1/networks/mainnet/subnets/2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7
acqG44BnRto' \
  -H 'accept: application/json'
```

The Output

This query provided a rich dataset revealing the internal architecture of Apertum:


```
{
  "subnetId": "2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto",
  "createBlockTimestamp": 1738225810,
  "createBlockIndex": "20622860",
  "ownerAddresses": [
    "P-avax138usvdatdk6syvjzkwmla9n39535zex8t0kr9m"
  ],
  "threshold": 1,
  "locktime": 0,
  "subnetOwnershipInfo": {
    "addresses": [
      "P-avax138usvdatdk6syvjzkwmla9n39535zex8t0kr9m"
    ],
    "locktime": 0,
    "threshold": 1
  },
  "isL1": true,
  "l1ConversionTransactionHash":
"2to9mahNx825sJntcmq2VAqNDKZaTkkCiTud4dTSpCBaQePZEu",
  "l1ValidatorManagerDetails": {
    "blockchainId": "YDJ1r9RMkewATmA7B35q1bdV18aywzmdiXwd9zGBq3uQjsCnn",
    "contractAddress": "0x0feedc0de0000000000000000000000000000000"
  },
  "blockchains": [
    {
      "createBlockTimestamp": 1738225820,
      "createBlockNumber": "20622863",
      "blockchainId": "YDJ1r9RMkewATmA7B35q1bdV18aywzmdiXwd9zGBq3uQjsCnn",
      "vmId": "VpZ47y5KN9XLDsaBQWEemUCse5UyBSa5kKuN5pALtzoETEena",
      "subnetId": "2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto",
      "blockchainName": "ApertumMainnet",
      "evmChainId": 2786
    }
  ]
}
```


Analysis of Findings

This API response provides a wealth of "hidden" technical details. By decoding specific parameters, we can infer the network's consensus engine and governance evolution:

1. **Consensus Mechanism:**

- **Observation:** The `vmId` is `"VpZ47y5KN9XLDsaBQWEemUCse5UyBSa5kKuN5pALtzoETEena"`.

- **Analysis:** This specific ID is the unique fingerprint for Ava Labs' official Subnet-EVM. Usage of this VM binary mandates the use of the Snowman++ consensus protocol (a linear chain optimized implementation of Snowman). This explains why explorers like Avascan classify the consensus as Snowman++ even without an explicit API tag. Apertum's implementation uses the standard, unmodified Subnet-EVM with Snowman++ consensus, ensuring that all trust assumptions and security guarantees of Avalanche's validator set are directly inherited.

For the avoidance of doubt, Apertum does not alter AvalancheGo consensus logic, validator behavior, or underlying security primitives; it inherits these directly from the Avalanche framework. Because the Subnet-EVM VM is used without customization, Apertum relies entirely on Avalanche's native Snowman++ consensus engine with no protocol-level deviations.

2. **Architecture & Evolution:**

- **Observation:** `isL1: true` and the presence of `l1ConversionTransactionHash` (`2to9mahNx825sJntcmq2VAqNDKZaTkkCiTud4dTSpCBaQePZEu`).
- **Analysis:** Apertum operates as a sovereign **Avalanche L1&& (Elastic Subnet). The transaction hash indicates that the network likely started as a standard Subnet and executed an on-chain transformation transaction to upgrade its status. This enables it to define its own staking economics independent of the primary network.

3. **Permissionless Validation Logic:**

- **Observation:** `l1ValidatorManagerDetails` points to contract `0x0feedc0de0000000000000000000000000000000`.
- **Analysis:** This "vanity address" is the standard precompiled contract for PoS management in Avalanche L1s. It shows that validator entry/exit is handled programmatically via smart contracts on the Apertum chain itself, allowing for a truly permissionless validator set.

4. **Network Identifiers:**

- **Observation:** `blockchainId` is `"YDJ1r9RMkewATmA7B35q1bdV18aywzmdiXwd9zGBq3uQjsCnn"` and `evmChainId` is **2786**.
- **Analysis:** It is crucial to distinguish between these two IDs. The `blockchainId` is the unique handle on the Avalanche P-Chain (used to construct RPC URLs like `.../ext/bc/YDJ1.../rpc`), while the `evmChainId` is used internally by the VM for transaction signing and replay protection.

5. **Genesis & Lifecycle:**

- **Observation:** `createBlockTimestamp` is `1738225810`.
- **Analysis:** This timestamp converts to **January 30, 2025, at 08:30:10 UTC**. This precise genesis time provides a clear starting point for the network's history, allowing auditors to calculate the exact age and uptime of the blockchain.

This raw data perfectly aligns with the official network information indexed on Avascan, showing the consistency between the node-level state and the public explorer.

Apertum should not be interpreted as a Layer-2 or rollup. It is a sovereign Layer-1 blockchain within the Avalanche ecosystem. Apertum does not rely on external data availability layers, shared security models, or settlement chains. All consensus and state transitions occur natively within the Apertum Layer-1 environment. Because Apertum operates atop

Avalanche's unmodified consensus and validator framework, the trust assumptions and security guarantees of the base Avalanche protocol are directly inherited without dilution or deviation.

Liveness & On-Chain Activity

Establishing the Connection

Before diving into the chain's state, we performed two preliminary checks to ensure a secure connection. First, we verified the **Chain ID** via RPC, showing it matches the expected `2786`. Second, we briefly reviewed the **Consensus Layer** on the P-Chain, identifying a healthy set of **11 active validators** securing the network.

With the identity and security of the endpoint confirmed, we proceeded to the core of our review: interacting with the Apertum EVM to verify real-time block production and user activity.

To examine that Apertum is an active ledger and not a dormant chain, we queried the full details of the latest block. This provides more granular evidence than a simple height check, as it reveals the actual content of the ledger (transactions, miner, timestamps).

The Query

We used the standard `eth_getBlockByNumber` method with the boolean flag set to `true` to retrieve full transaction objects.

RPC Endpoint: `https://rpc.apertum.io/ext/bc/YDJ1r9RMkewATmA7B35q1bdV18aywzmdiXwd9zGBq3uQjsCnn/rpc`

```
curl -X POST --data '{
    "jsonrpc":"2.0",
    "method":"eth_getBlockByNumber",
    "params":["latest", true],
    "id":1
}' -H "Content-Type: application/json"
https://rpc.apertum.io/ext/bc/YDJ1r9RMkewATmA7B35q1bdV18aywzmdiXwd9zGBq3uQjsCnn/rpc
```

Key Output Data

The node returned the following critical data points (excerpted for clarity):

```
{
  "jsonrpc": "2.0",
  "result": {
    "number": "0x4a68ef",  // height: 4,876,527
    "timestamp": "0x692e97bf",
    "miner": "0x7a36c99f89d30ed31f3fd1d1ff302114fdffce0e",
    "gasUsed": "0xf0eb4", // Gas
    "transactions": [
      {
        "hash":
"0xdc9b3401aa50bd847f62b3bebf24b34f1dd9b264b46bbc7e9660f67ce402aa84",
        "from": "0x7e395dddac0b20813100e5cd4c5f7335024849b1",
        "to": "0x2096bd2e860b450498a7c908708757a1eb306416",
        "input":
"0x86bb8f37000000000000000000000000000000000000000000000000000000000000001441b000000000
0000000000000000000000000000000000000000000000000000032",
        "value": "0x0",
        "chainId": "0xae2" // Chain ID: 2786
      }
    ]
  }
}
```


Analysis of On-Chain Activity

The raw data retrieved from the latest block (`0x4a68ef`) offers the evidence of Apertum's operational status and consistency:

1. Block Height

- The returned block number is `0x4a68ef` .
- **Decimal Conversion:** 4,876,527.
- **Observation:** The block height continues to increment steadily (compared to previous checks), showing that the consensus engine is healthy and the network has been running stably for an extended period.

2. Transaction Analysis

- The block is **not empty**. It captured a specific transaction with the hash `0xdc9b3401aa50bd847f62b3bebf24b34f1dd9b264b46bbc7e9660f67ce402aa84` .
- **Smart Contract Interaction:** The transaction was sent to address `0x2096bd2e860b450498a7c908708757a1eb306416` with a `value` of `0` but a significant `input` payload (`0x86bb8f37000000000000000000000000000000000000000000000000000000000000001441b000000000000000000000000000000000000000000000000000000000000032`). This indicates a function call to a smart contract rather than a simple token transfer.
- **Gas Usage:** The block reports `gasUsed` of `0xf0eb4` (986,804 gas), showing legitimate computational work being performed by the EVM.

3. Identity Verification

- The transaction data includes the field `chainId: 0xae2` .
- **Decimal Conversion:** 2786.
- **Conclusion:** This shows that the RPC endpoint we connected to is indeed the correct Apertum Mainnet, matching the Chain ID registered in the genesis metadata.

This raw data aligns with the information indexed on Apertum's official explorer, showing the consistency between the node-level state and the public explorer.

PART 3: THIRD-PARTY VALIDATION | APERTUM

Beyond the Avalanche Boundary

While our previous queries verified Apertum's existence on the Avalanche P-Chain, a mature blockchain requires recognition from the broader Web3 ecosystem. In this section, we validate Apertum's status through independent, non-Avalanche third-party registries.

These listings serve two purposes: they provide social proof of the network's legitimacy, and more importantly, they offer an opportunity for **cross-verification**. By comparing the data registered on these external platforms with the raw data we queried earlier, we can show the consistency and accuracy of Apertum's public configuration.

Chain Registries

Among the various listings, two stand out for their critical importance to the EVM ecosystem. These repositories act as the trust sources for developers and infrastructure providers globally.

Ethereum Lists

- **Source:** `ethereum-lists/chains` GitHub
- **Evidence:** Github Commit

This repository is the industry-standard source of truth for EVM Chain IDs. It is maintained by the Ethereum community and underpins **EIP-155 replay protection**. Critical infrastructure, including MetaMask, WalletConnect, and hardware wallets (Ledger/Trezor), pulls metadata directly from here. A merged PR signifies that the network has claimed a unique global identifier, preventing ID collisions with other blockchains.

Consistency Check:

- **Registry Data:** Lists Chain ID as **2786**.
- **RPC Finding:** Our earlier `eth_chainId` query returned `0xae2` (Decimal: **2786**).

Chainstack

- **Source:** `chainstacklabs/subnetlist` GitHub
- **Evidence:** Pull Request #28

Chainstack is a top-tier provider of "Node-as-a-Service" infrastructure. Their subnet repository contains the technical specifications required to spin up enterprise-grade nodes. Inclusion here implies that Apertum's technical architecture has been reviewed and standardized for automated deployment. It signals that the network is structurally ready for professional node operators.

Consistency Check:

- **Registry Data:** Lists Chain ID as **2786**.

- **RPC Finding:** Our earlier `eth_chainId` query returned `0xae2` (Decimal: **2786**).

Broader Ecosystem Availability

Following the validation from authoritative sources, Apertum's data has propagated to numerous downstream aggregators. These platforms are crucial for end-user accessibility, allowing users to add the network to their wallets with a single click.

The information listed on the following platforms (RPC URLs, Ticker Symbols, Explorer Links) is consistent with our findings:

- **ChainID.network**: https://chainid.network/chain/2786/
- **Chainlist.wtf**: https://chainlist.wtf/chain/2786/
- **EVM Chain Info**: https://evmchain.info/?search=2786
- **EVMChainList.org**: https://evmchainlist.org/?search=2786
- **Networks.vercel.app**: https://networks.vercel.app/ — Search for 2786
- **Chainlist.simplr.sh**: https://chainlist.simplr.sh/ — Search for 2786

CONCLUSION | APERTUM

This review utilized the `avalanche-cli` framework as a technical reference to query the on-chain status of the Apertum network. Direct API interactions with the Avalanche Mainnet showed the registration of Apertum as an **Avalanche L1** with ID `2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto` , currently supported by a set of **11 active validators**. Internal network metrics retrieved via RPC indicate a block height exceeding **4.87 million** with recorded smart contract transactions. These on-chain parameters align with the data indexed in external registries including **Ethereum Lists** and **Chainstack**, documenting the network's operational state and its integration within the broader ecosystem. Specifically, Apertum operates as a sovereign Avalanche Layer-1 (Subnet) built on the fully standard and unmodified Avalanche Subnet-EVM, without introducing any changes to Avalanche's consensus engine, validator framework, or core security assumptions. As Apertum utilizes the unmodified Avalanche Subnet-EVM architecture and Snowman++ consensus, the network inherits Avalanche's security, decentralization, and consensus guarantees by design.

REFERENCE | APERTUM

Client-Provided Sources:

1. Avalanche Subnet Explorer: https://subnets.avax.network/subnets/2Y83BEXg7zgSCJunL2KD2i1vjMfaMnU1QEpG4M7acqG44BnRto
2. Avalanche Explorer: https://avascan.info/blockchain/apertum/info
3. Chainstack Subnet Listing: https://github.com/chainstacklabs/subnetlist/pull/28
4. Ethereum Chain Registry Commit: https://github.com/ethereum-lists/chains/commit/798b7819844b87194d9c12e6bda306aa5f7857b1
5. ChainID Network: https://chainid.network/chain/2786/
6. Chainlist.wtf: https://chainlist.wtf/chain/2786/
7. Networks.vercel.app: https://networks.vercel.app/
8. EVM Chain Info: https://evmchain.info/chain/2786/
9. EVMChainList.org: https://evmchainlist.org/?search=2786
10. Chainlist.simplr.sh: https://chainlist.simplr.sh/
11. Ethereum Chains Master List: https://github.com/ethereum-lists/chains
12. Apertum Node RPC: https://rpc.apertum.io/ext/bc/YDJ1r9RMkewATmA7B35q1bdV18aywzmdiXwd9zGBq3uQjsCnn/rpc

Additional References:

13. avalanche-cli: https://github.com/ava-labs/avalanche-cli
14. Avalanche CLI Docs: https://build.avax.network/academy/avalanche-l1/customizing-evm/04-your-evm-blockchain/01-avalanche-cli
15. AvalancheGo P-Chain RPC: https://build.avax.network/docs/rpcs/p-chain
16. Issuing API Calls: https://build.avax.network/docs/rpcs/other/guides/issuing-api-calls
17. Glacier API: https://glacier-api.avax.network/api#/

DISCLAIMER | CERTIK

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you ("Customer" or the "Company") in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

This report is not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR

UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Elevating Your Web3 Journey

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is the largest blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.