Execution Machine Explained

Written by Rohit Pathare

What is Execution Machine (EXM)?

Execution Machine (EXM) is a developer platform that provides the ability to create and leverage **blockchain-based (permanent) serverless functions** without the need for knowledge of or access to blockchain technologies like wallets and tokens.

This further enables us to
create **composable**, **immutable** and **trustless** applications in a frictionless manner.

This might seem like a mouthful so let us understand what it all means.

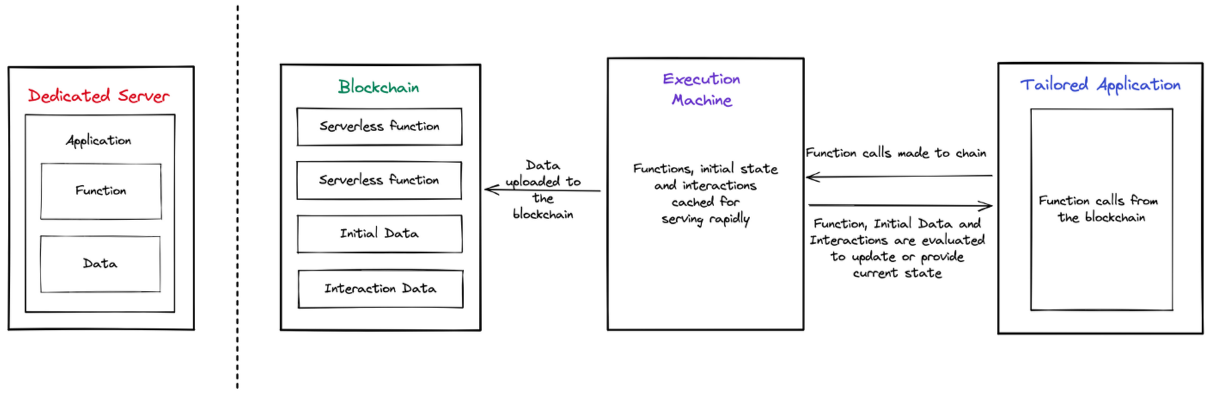
Serverless Functions

Serverless functions are functions hosted and maintained on an infrastructure like cloud services. In our case, however, they are stored on a blockchain chain (Arweave) through an intermediary (EXM) that also stores a copy as cache to rapidly serve applications at any time. Additionally, EXM covers the cost for uploading the data to Arweave and making the process crypto agnostic for devs.

The infrastructure handles the storage and execution. This eliminates the need for maintaining a dedicated server, reduces costs and adds a layer of modularity.

The modularity also brings in **composability** to select and assemble functions in various combinations to create custom applications suited to our requirements. These functions, and interactions with them, are ***permanently***

stored on chain, they cannot be tampered with and are available for anyone to view, making them **immutable** and **trustless**.


Dedicated Server
Application
Function
Data
Blockchain
Serverless function
Serverless function
Initial Data
Interaction Data
Data uploaded to the blockchain
Execution Machine
Functions, initial state and interactions cached for serving rapidly
Function calls made to chain
Function, Initial Data and Interactions are evaluated to update or provide current state
Tailored Application
Function calls from the blockchain
Functions on dedicated servers vs serverless functions on blockchains

Now that we understand what Execution Machine provides, let us look at how it does so.

What are the supporting technologies that make this possible?

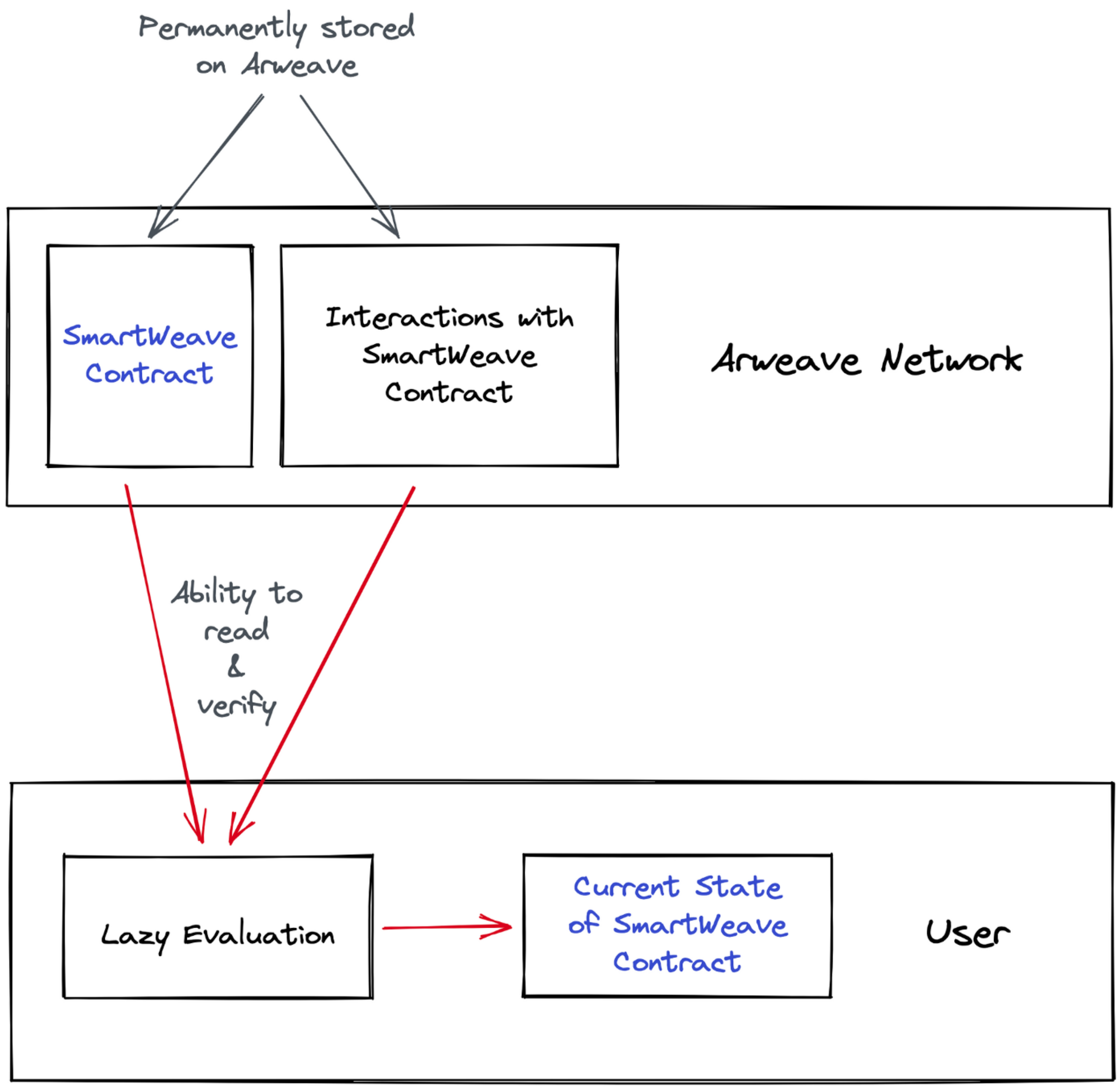
Smart Contracts

Smart contracts are programs intended to automatically execute according to the terms specified (programmed) within them. As the outcome for a given set of terms is always the same and all the information is immutable and publicly accessible on chain, all the involved parties can place their trust in this system to impartially execute its functions.

However, the traditional method for verifying contracts through a consensus phase that all nodes must go through reduces performance and is expensive. Every node must store the entire copy of the state of the blockchain which is expensive and every node must execute each transaction locally to verify and arrive at consensus which is time consuming and reduces performance.

Lazy Evaluation is a method that aims to solve these obstacles.

Lazy Evaluation


Permanently stored on Arweave
SmartWeave Contract
Interactions with SmartWeave Contract
Arweave Network
Ability to read & verify
Lazy Evaluation
Current State of SmartWeave Contract
User

Lazy Evaluation Explained

Lazy evaluation, as the name suggests, is a method for lazily evaluating smart contracts and their current state on the blockchain. The smart contract itself and any interactions (write operations) with them are stored on chain and can be accessed by any user.

It aims to shift the burden of processing from the nodes to the users. The user can opt to evaluate and interpret the smart contract code and interactions with it locally to verify the current state of the contract.

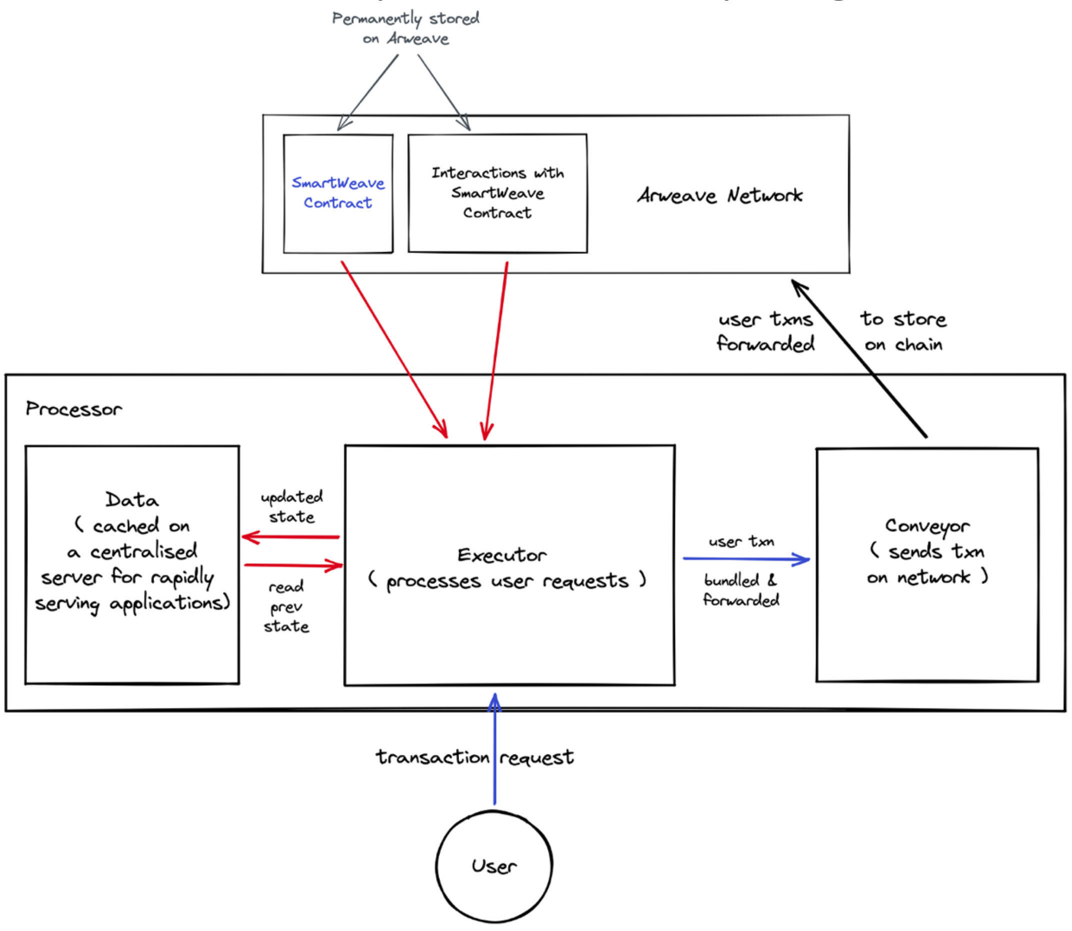
This eliminates the need for nodes to store the full copy of the current state of a chain and arrive at a consensus on it. Thus, reducing the cost and improving performance, respectively.

As everyone has access to the same data, everyone will interpret the it in the same way ensuring everyone has access to the same current state of information.

Verifiable Computing

Verifiable computing is a form of computing that takes advantage of the benefits of centralised system while still guaranteeing a decentralised result.

Every serverless function either has the ability to read or update the state of some information. Using verifiable computing, this state is cached in a centralised server which allows for greater performance as consensus is not needed at the time of processing, but the information is always available for verification by the users. This allows users to "lazily evaluate" even when it is stored on the cache layer before eventually being moved on chain.



Permanently stored on Arweave
SmartWeave Contract
Interactions with SmartWeave Contract
Arweave Network
user txns forwarded
to store on chain
Processor
Data (cached on a centralised server for rapidly serving applications)
updated state
read prev state
Executor (processes user requests)
user txn
bundled & forwarded
Conveyor (sends txn on network)
transaction request
User

Verifiable Computing Explained

For verifiable computing to work seamlessly, some core parts must be implemented.

- Executor: A software that processes user transaction requests and caches them.
- Processor: A centralised pipeline (system) responsible for receiving transactions by a single or multiple users. After receiving the different bulks of transactions sent, processor must re-evaluate the smart contract with the new data. As transactions are received, the latest state of the smart contract must be upgraded and saved with accessibility to the user. The processor is responsible for ordering the transactions, usually by timestamp.
- Conveyor: A centralised system that establishes a bridge between a data-based blockchain. All the transactions received by the processor must be sent to the conveyor, the conveyor will guarantee the success of storing these operations in a data-based blockchain like Arweave.

Bringing it all together…

Now that we understand the technologies Execution Machine relies on, we can see how it all works together.

EXM is an abstraction built on top of a supercharged execution engine (3EM) that enables Lazy Evaluation and Verifiable Computing.

A user sends a transaction request to a dedicated EXM server. With the help of Verifiable Computing, Execution Machine is able to process user requests in a quick and performant manner, eliminating the need for blockchain technology like tokens and wallets, while still maintaining a decentralised result. EXM then updates its cache layer with the updated state while also uploading the data to Arweave. The cache layer is used as an aid to rapidly serve applications at any time.

Additionally, EXM is able to maintain a **trust minimised** environment as users can verify the transactions and current state of the contract/ functions using Lazy Evaluation through 3EM.

Compared to serverless functions on regular servers, EXM's serverless functions provide permanency, accountability and transparency. They can be used for applications that require certain level of trust no matter who the authority is. Examples of such applications include a legal registry, healthcare records, supply chains, decentralized social media, or just a way to bridge data into Arweave. This is where EXM comes in.

That's all folks!

If you're interested in learning more about Execution Machine, check out https://exm.dev.