



AQER

Yellow Paper

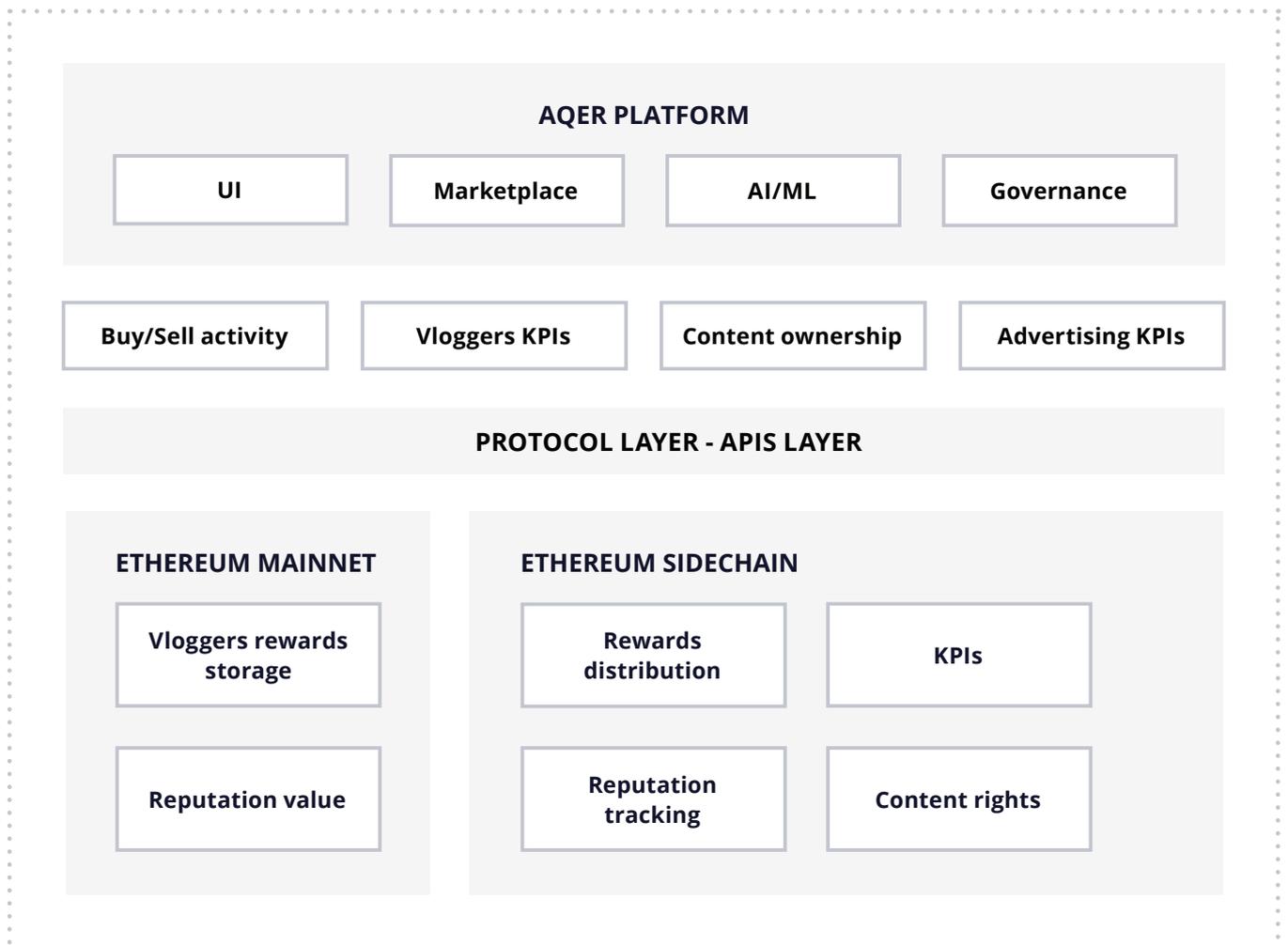
CONTENTS

AQER Technical Overview	3
AQER Ethereum Mainnet Environment	7
<i>TGE Smart Contract</i>	7
<i>Reputation Storage Smart Contract</i>	8
<i>Rewards Smart Contract</i>	9
AQER Sidechain Solution	11
<i>Protocol Rewards Tracking Smart Contract</i>	14
<i>Protocol Smart Contract for KPI's</i>	15
<i>Protocol Content Rights Smart Contract</i>	17
AQER Platform Off-Chain Elements	20
Appendix A; Purpose of Using ERC20 Tokens and the Token Sale	21
<i>AQER Token Purpose and Specifics</i>	21
<i>Token Sale</i>	21
Appendix B; Tokens Specification	22
<i>B.1 AQER Token</i>	22
<i>Sidechain AQER Token</i>	22

AQER TECHNICAL OVERVIEW

The AQER solution is a multidimensional platform based on the hybrid model of Ethereum type blockchains. The solution is based on the API connection between off-chain and on-chain elements and oraclisation of instances in the public blockchain to sidechain and vice versa.

AQER PROTOCOL STRUCTURE



Core Elements of AQER Protocol:

- Any user of AQER protocol is dealing with an off-chain platform, which provides User Interface (UI), a place for communication between each other and over governance over protocol elements.
- Through the platform, vloggers and their customers get access to all required activities described in the White Paper of the AQER solution.
- The protocol itself is a combination of API's to the AQER platform back-end to create Decentralized Applications (DApps) as it is connected to the Ethereum public blockchain and the Ethereum based sidechain, which are responsible for the provision of trust and decentralization.

The Ethereum public blockchain is used for AQER platform value storage and transfer with AQER token generated in ERC20 standard. This is a critical step to achieve business goals and market penetration for the the platform, while the system will assure the highest achievable rate of security for users value. A critical issue is that the cost for such security is equal to the gas fees required for AQER token transactions.

The core element of the AQER business model lies is in the decentralization of vloggers' activity management with the assurance of direct control over their content by vloggers themselves. Such an approach requires additional business logic execution by blockchain through smart contracts, which are not related to value transfer. Instead they are related to the management of value transfer and the transactional cost (Virtual Machine nodes fees) of such contracts are insufficient in the public blockchain. For such purposes, AQER delivers Ethereum based sidechain in the standard of consortium chain (partially permissioned blockchain). This allows AQER to avoid gas fees inside Ethereum mainnet, while assuring decentralization, transparency and direct ownership of content generated by vloggers.

A blockchain is an event-driven system and usage of it for static data operations is inefficient. To address this issue, the AQER platform has off-chain elements, which are dedicated to mutable data storage and communications between users, which are trustless as they are not related to any value transfers. Communication of off-chain elements directly with public blockchain elements is costly and leads to security holes. To avoid public blockchain value security compromisation, any off-chain element is communicating with sidechain, the role of which is to protect real value inside the public blockchain.

Ethereum based systems are relying on Virtual Machines, which leads to the ability of the system to deal with accounts, represented as public addresses and balances. To manage balances, the Virtual Machine treats any transaction as a smart contract, which contains some business logic executed as decentralized applications (DApps). Each DApp consists of blockchain at its core and some off-chain elements, including front-end for UI.

The AQER solution is based on the sequence of connected smart contracts, which are dedicated to the execution of their own business logic to provide decentralization and a trustless environment. Smart contracts run on a blockchain and allow the achievement of disintermediation from central entities, i.e. they allow the establishment of a contract, to grant immutability to it and to automatically execute a performance between parties without the need to rely on a central entity, trusted by all parties, which verifies the contract.

Below is a list of available smart contracts in the AQER ecosystem:

- Smart contract advertising (based on KPI's set by the content seeker).
- Smart contract organic advertising (based on KPI's set between content producers).
- Smart entertainment rights contract (based on KPI's set by the content provider).
- Smart contract for AQER Token Generation Event with the ability to subtract fees in the amount of 2% from AQER token transactions.
- Smart contract for vloggers ranking.
- Smart contract for vloggers rewards accounting.

Smart Contract advertising (SCa) manages every type of token in-flow to a content producer from a content seeker.

SCa has the following features:

- Involved parties pro-quota allocation.

- Agreed KPI's for token in-flow unlock.

Smart Contract organic advertising (SCoa) manages every type of token flow between content producers for services to be exchanged reciprocally.

SCa has the following features:

- Involved parties pro-quota allocation.
- Agreed KPI's for token flow unlock.

Smart entertainment right Contract (SerC) represents the right to “use” an entertainment right in a pre-determined way. It manages every type of token in-flow between a content provider or content seeker and content producers.

SCa has the following features:

- Involved parties pro-quota allocation.
- Agreed KPI's for the right of “use”.

Smart contract of AQER token generation (STGE) with the assurance of revenue stream via commission on its transactions.

STGE has the following features:

- Generates public AQER tokens, which will be used for value storage inside the public blockchain.
- In-built 2% fees on any AQER token transaction.

The Smart contract for AI/ML zk-STARKs is used to rank all vloggers based on their performance and the time they are staying with a platform based on AI/ML TEE computations.

It has the following features:

- Generates reputational tokens in the ERC721 standard to track the rank of the vlogger based on an implemented algorithm.
- Indicate multiple of vlogger rewards based on their ranking.

The Smart contract for rewards accounting (SRew) allows for an account of all vloggers activity and manages the distribution of their rewards based on their performance in a trustless and distributed manner.

SRew has the following features:

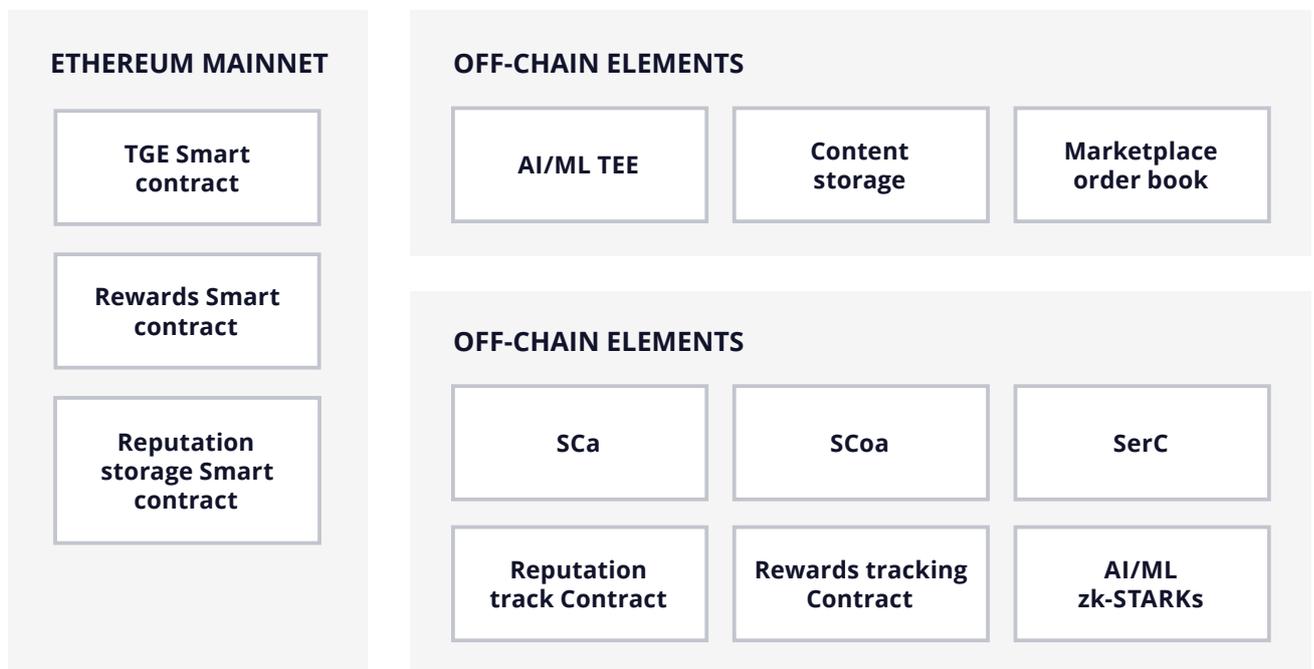
- Smart contracts fabric feature allows for the establishing of a rewarding contract per each vlogger/activity.
- Storage of all rewards distribution in the ledger.
- Indicate events for AQER token (AQER) transactions in the public Ethereum mainnet.

The smart contract natively supports and enforces the negotiation of a contract and the execution of a performance between content creators and content seekers.

The major drawback of smart contract usage in the public blockchain is their operational costs evaluated with gas fees (transaction fees in Ethereum) which are unstable in fiat currencies. As the Ethereum Virtual Machine (EVM) charges higher with the growth of smart contract business logic complexity. In order to avoid the volatility of operational costs related to gas fees, the AQER platform exports complex business logic to the sidechain based on the Proof of Stake consensus, while the real value in monetary terms is stored in public blockchain with AQER token.

Another value of the AQER platform is provided by its Artificial Intelligence off-chain element. AI itself is leaving inside containers that are providing Trusted Execution Environment (TEE) and connected to sidechain with JSON RESTfull APIs layers. Such an approach allows for influencing sidechains to balance statements based on AI computation results without the need to disclose it as Machine Learning treats it through the application of cutting-edge cryptographical technology - zk-STARKs¹.

OVERALL AQER PLATFORM ARCHITECTURE



Core Elements of AQER Architecture:

- 1 Ethereum mainnet is used for AQER tokens creation, transactions and value storage with the next public smart contract:
 - Token Generation Event (TGE) Smart Contract for AQER tokens creation and in-built commission of 2% per each transaction on the public blockchain.
 - The Rewards Smart Contract is used for tracking user rewards and payment arrangements and to store AQER tokens while users make free of charge transactions for change distribution of balances in a manner similar to the lightning network of Bitcoin².
 - The Reputational smart contract is used as storage of reputation statement right-offs to provide additional trust for vloggers.

¹ Details on zk-STARKs: <https://medium.com/coinmonks/zk-starks-create-verifiable-trust-even-against-quantum-computers-dd9c6a2bb13d>

² https://en.bitcoin.it/wiki/Lightning_Network

2

The Ethereum-based sidechain is used for actual changes in each user's balances and is free of Ethereum gas fees with sequences of smart contracts:

- Smart contracts for the setting of KPI's, to set rules of rewards distribution, which includes:
 - Smart contract advertising.
 - Smart contract organic advertising.
 - Smart entertainment right contract.
- The reputation track smart contract is used for actual changes of vloggers reputation, which is used for KPI's tracking and leads to an appropriate distribution of rewards in a trusted manner.
- The rewards tracking smart contract is used for balance equalization at the public blockchain based on the distribution of rewards after reaching KPI's in other smart contracts.
- AI/ML zk-STARKs Smart contract is used for verification of AI/ML TEE results based on ring signatures that allow ensuring ranking of vloggers are based on respectful computations and no changes in ML traits were made by any party.

3

AQER off-chain platform is required for the handling of the AI/ML TEE, marketplace order book for the arrangement of users connection and KPI's setting and the actual content storage to protect sidechain and blockchain from overflow with data.

AQER ETHEREUM MAINNET ENVIRONMENT

The AQER Ethereum mainnet environment is used to protect its sidechain from double-spending and to create AQER tokens that are a value storage for users. From that perspective, a core element of AQER solution related to mainnet is a TGE Smart contract that will create the AQER ERC20 standard token.

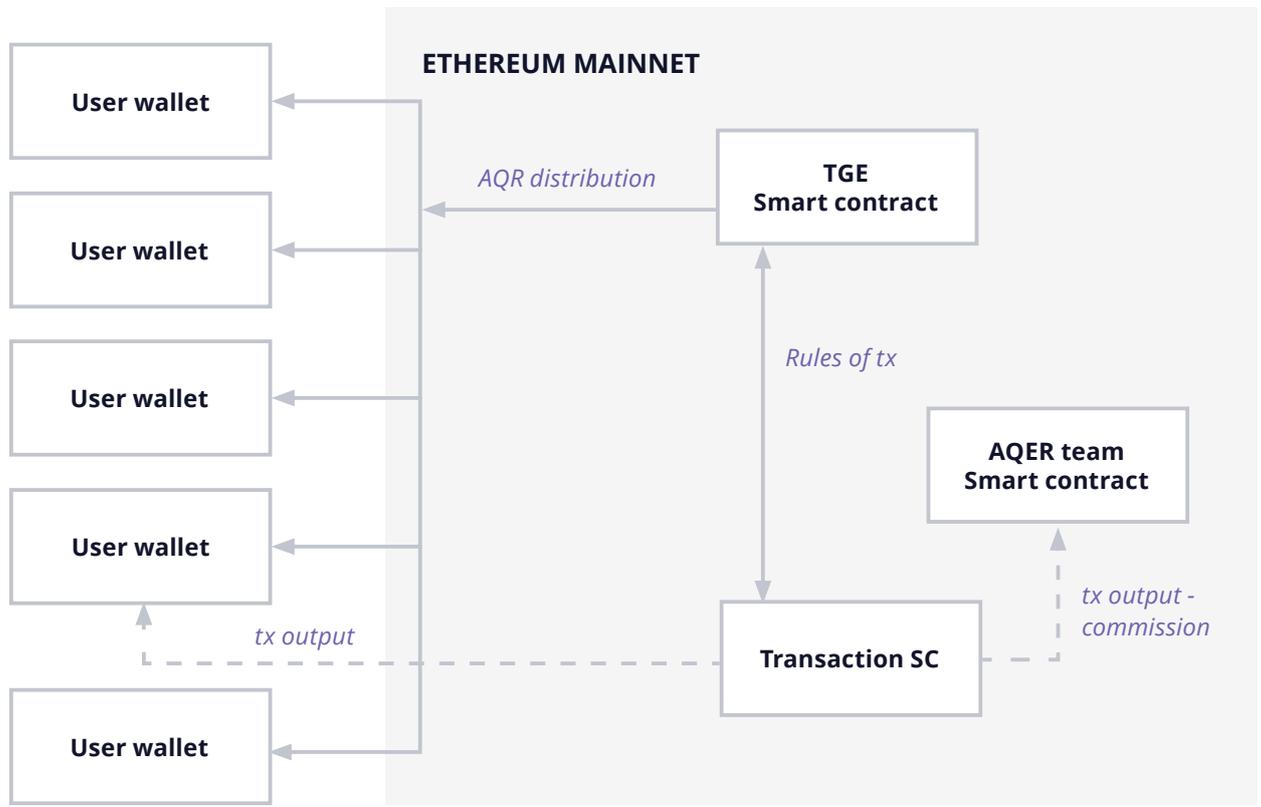
■ TGE SMART CONTRACT

AQER TGE Smart Contract Description:

- Public smart contract for the creation of pre-mined ERC20 standard tokens.
- Total supply of tokens is limited to 100,000,000 AQER with 8 decimals.
- Each transaction of AQER tokens include 2% commission fees to AQER team controllable contract address.

AQER tokens are available at the open market, which ensures public token address.

AQER TGE SMART CONTRACT ARCHITECTURE



AQER tokens exist inside the Ethereum environment, which means that each transaction of tokens inside the blockchain is arranged as a smart contract. TGE Smart contracts have the inbuilt requirement of transaction output distribution, which enforces each transaction smart contract to subtract 2% of AQER tokens to dedicate AQER team smart contract that is controllable by the team wallet. Transactions to and from public exchanges also fall under this rule, while the exchange of AQER tokens inside any centralized exchange does not include such commission due to the specifics of centralized exchanges order books operational flow.

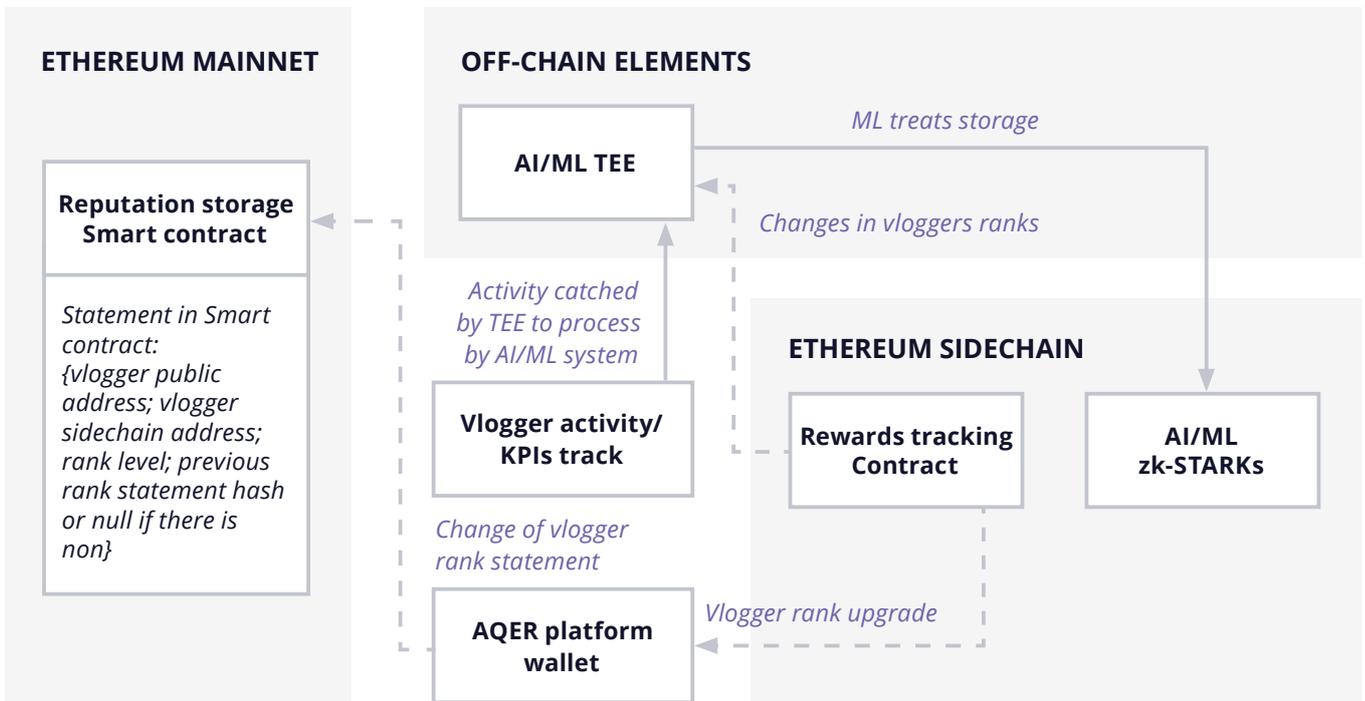
■ REPUTATION STORAGE SMART CONTRACT

The reputation storage smart contract is a utility smart contract that is used for the storage of a vlogger's reputation in the public blockchain.

Core specifics of that contract:

- The contract stores the addresses of vloggers in the public chain and sidechain and relates it to the address event of reputation level promotion.
- Contract is required ETH for storage of statements in it.
- AQER platform is responsible for making statements inside this smart contract.
- Smart contract is publicly available, but allows for the verification of the vlogger's reputation only by the registered platform user.
- Smart contract is not storing any tokens or ETH on its address.

STRUCTURE OF REPUTATION STORAGE SMART CONTRACT ARCHITECTURE



Statement of Reputation Change Includes:

- Vlogger public address to allow verification that rewards distribution in a public environment, which is related to the right addresses.
- Vlogger sidechain address that allows permitted platform users to identify a vlogger in the sidechain and to assure that rewards are directed to the appropriate person.
- New rank level to indicate that changes in ranking are really taking place on both sides - both in relation to upgrade and downgrade.
- Hash of the previous statement that allows for the creation of a connection between statements in the smart contract and its appropriate sequence.
- Null if vlogger receives its initial upgrade for the first rank.
- To avoid statements, which are not required and to reduce the number of statements in smart contract statements provided only after the first upgrade of a vlogger. This means that newly registered vloggers have no statement in a reputation storage smart contract.

■ REWARDS SMART CONTRACT

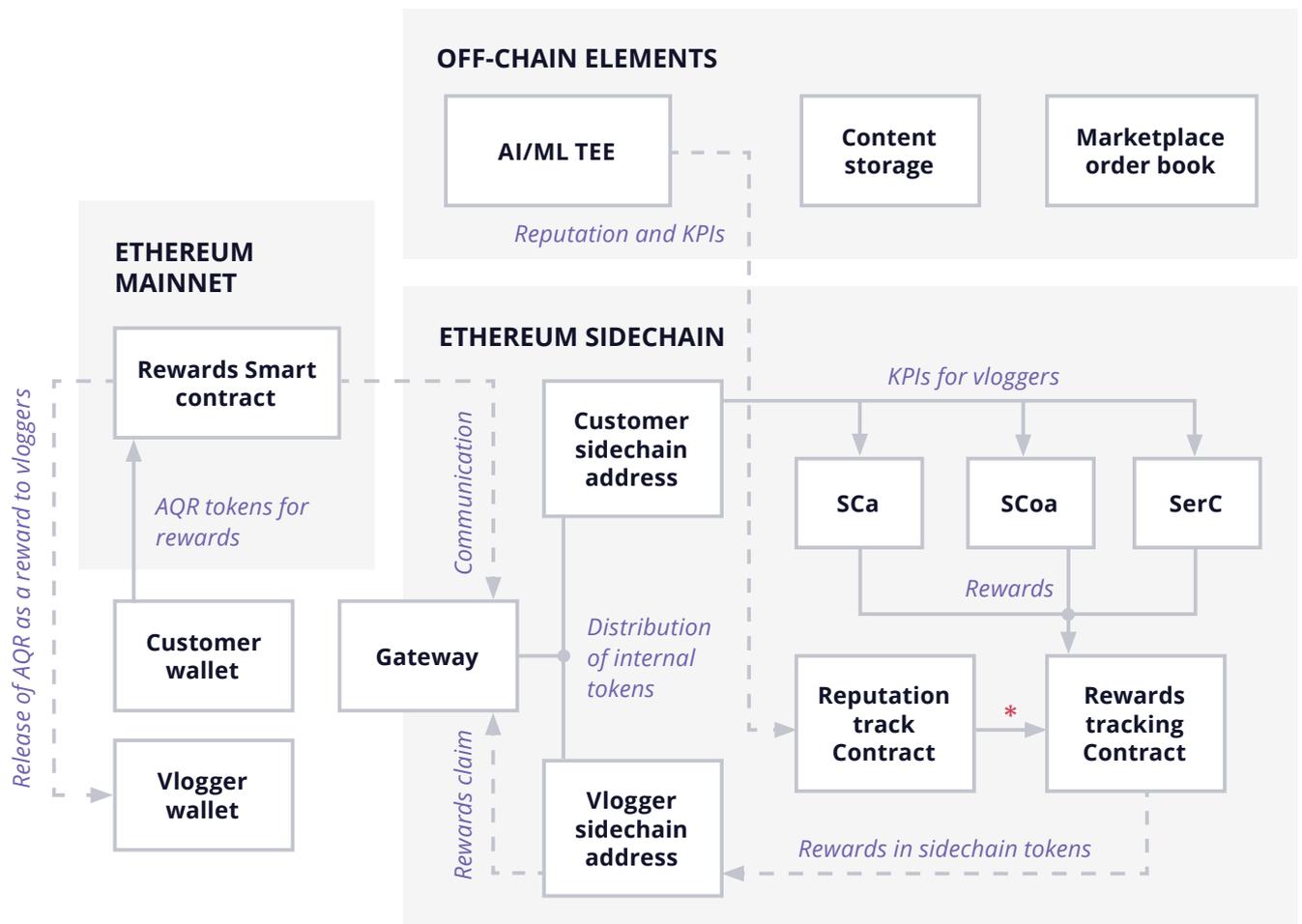
A major element of the AQER public environment and an element of the protocol kernel is a rewards smart contract. This smart contract allows for the separating of general holder of AQER tokens that hold or trade tokens from stakeholders interested in the utilization of AQER platform with its protocol layer. Instead of a controllable wallet, Rewards SC is aimed to store vloggers and their customer's tokens in a secure environment, while actual services are provided.

To deal with Rewards SC, each customer must provide a contribution of AQER tokens to its address that is reserved for rewarding of vloggers for their services.

Key features of the contract are:

- Include oraclisation function to track the distribution of vloggers rewards claims.
- Contract controllable by platform only at the amount of stored AQER tokens provided to it.
- Protect customers and vloggers from double-spending of AQER tokens due to attacks on sidechains - vloggers rewards cannot exceed Rewards SC balance.

REWARDS SMART CONTRACT ARCHITECTURE



* Reputation track SC will be described below, as it indirectly influence Rewards distribution

Core Elements of Architecture:

- Each customer who wants to obtain the services of vloggers from the AQER platform, must stake their AQER tokens to a rewards smart contract.
- Gateway between Ethereum mainnet and sidechain provide AQER sidechain tokens to customer sidechain address in an equal amount in values that allows setting rules of rewards for vloggers for their services.
- Vloggers reward distribution is arranged with tracking of KPI's in sidechain smart contracts and send internal tokens to vlogger sidechain address.
- Vlogger can claim to withdraw their rewards in AQER tokens when gateway transmits the claim to Rewards SC and receive AQER tokens to their wallets in Ethereum mainnet.

The back-end of rewards smart contract major elements are:

- JSON RESTfull API for gateway to oraclise it in rewards smart contracts.
- Infura for sidechain addresses connection to wallets.

With platform extension and the number of users growth to avoid overflow of Rewards Smart Contract, several of them will be deployed and Apache Kafka will be implemented to arrange the appropriate transfer of messages between smart contracts and gateway.

AQER SIDCHAIN SOLUTION

AQER platform protocol creates a decentralized environment for vloggers' services, which requires a combination of standard KPI's setting, blockchain decentralization and trustless connection with AI/ML technologies for vloggers' performance evaluation. This requires a significant amount of interactions between the blockchain system and off-chain elements. In the public environment of the decentralized operational systems like Ethereum mainnet itself, fee charges for computational power usage in complex computations are inefficient by operational costs. To avoid those charges, the AQER platform exports all business logic computations from the public environment to its sidechain based on Ethereum with Proof of Authority consensus behind it.

Ethereum is one of the most advanced systems, which allows for the creation of a decentralized environment with the cryptographic protection of users. With the option for the public environment, it is logical to select an Ethereum sidechain for operational purposes to assure operability between systems.

As all real value is stored and protected by public blockchain, the best consensus for sidechain is Proof of Authority. It allows avoiding "nothing in stake" problem with the high performance of computations due to the absence of Proof of Work task solving. The connection of nodes to sidechain will be managed by the general governance of the platform under an agreement between parties that hold the nodes. Blocks validation is executed by order of nodes that provide equal opportunities to govern the sidechain. There are no rewards for blocks validation for nodes and nodes holders receive benefits only under consortium agreement that governs platform stakeholders.

Metamask and Infura are used to connect to the sidechain with a user private signatures to govern sidechain transactions. To avoid actual value reduction of rewards inside the sidechain, all transactions

are created in the sidechain ERC20 token. Ether as a native token of Ethereum sidechain remains for smart contracts gas supply. Gas is used to protect the sidechain nodes from self-DDoS and is seen as an additional layer of protection for smart contracts in case of misbehaviour of any party in the sidechain.

As the AQER platform deals with a protocol that allows tokenizing user-generated content copyrights and store AI/ML performance with zk-STARKs, it requires the creation of non-fungible tokenized assets. For that purposes, AQER sidechain will contain ERC721 standard tokens and ring signatures of BLAKE-2b zk-SNARKs Smart contract as a core element of sidechain protocol, but outside the business model itself.

List of AQER Sidechain Technical Elements:

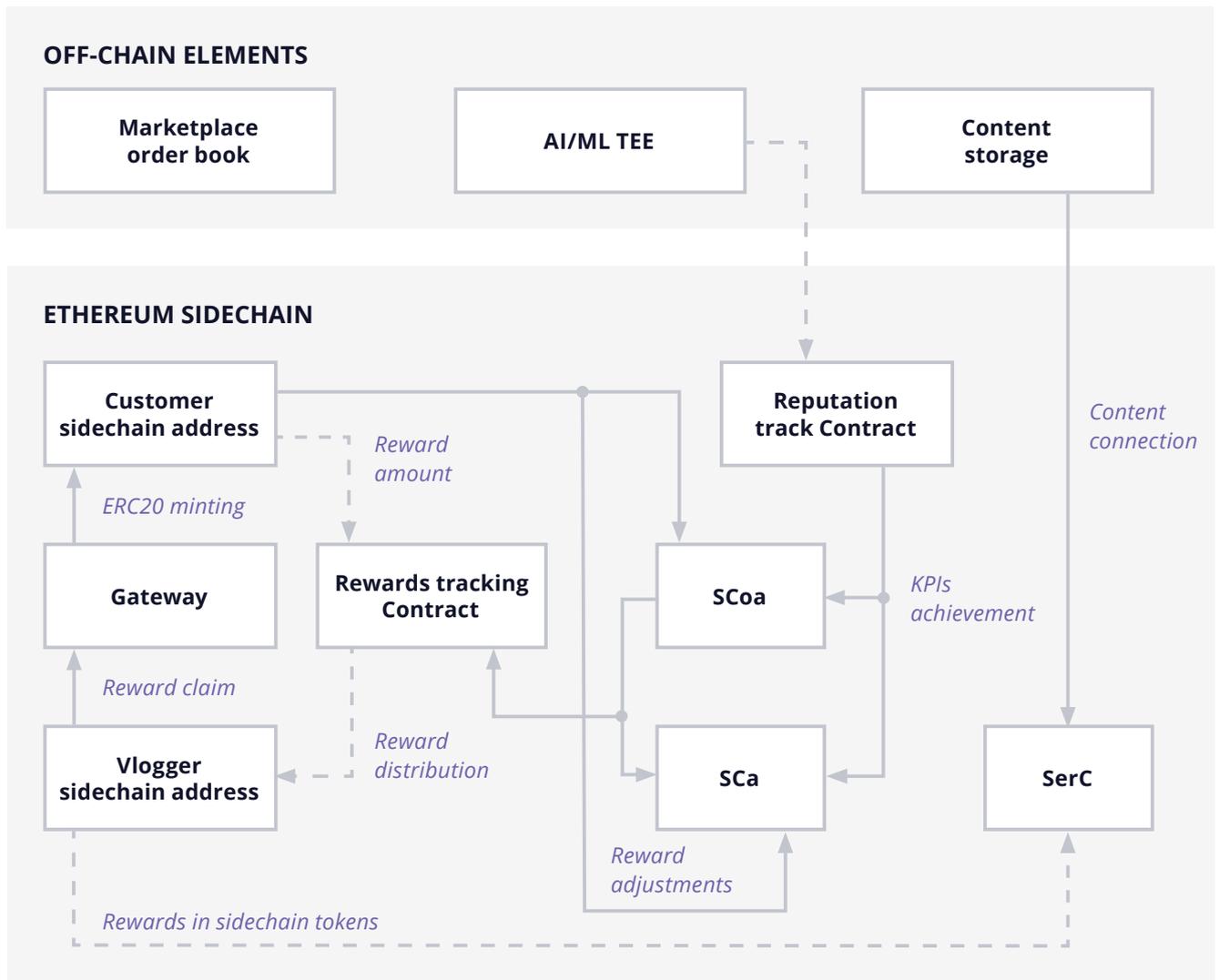
- ERC20 tokens TGE smart contract - allow for the creation of and minting of internal tokens inside the sidechain for the general transaction and operational smart contracts execution,
- ERC721 tokens TGE smart contract - allows for the creation of and minting of internal non-fungible tokens in sidechain for tokenization of content rights,
- zk-SNARKs Smart contract - allows for the introduction of BLAKE-2b cryptography and the introduction of ring signatures required for zk-STARKs proofs,
- PoA consensus - native Ethereum PoA consensus for sidechains.

Those elements are required to ensure operational flow inside the sidechain, fully controlled by sidechain consortium members and users do not need to interact with them.

As an element of protocol, the following smart contracts are used inside the sidechain:

- 1** Smart contract advertising that deals with sidechain ERC20 tokens.
- 2** Smart contract organic advertising that deals with sidechain ERC20 tokens.
- 3** Smart entertainment right contract that deals with sidechain ERC721 tokens.
- 4** Smart contract for vloggers reputation tracking that deals with sidechain ERC721 tokens.
- 5** Smart contract for vloggers rewards accounting that deals with sidechain ERC20 tokens.

OVERALL SIDECCHAIN CONTRACTS STRUCTURE



The actual operational smart contract is a reward tracking smart contract that obtains information from other smart contracts that together with off-chain TEE creates the AQER protocol. The content rights tokenization process is an additional element for rights management and is described below.

At the moment of sidechain deployment, the platform is equipped with the following technical tools:

- Front-end of HFC Platform for off-chain elements - Java (**Jdk1.8.x/Jre1.8.x** open source Java libraries - robust, secure and platform independent; **Javascript** for UI libraries to design the front-end components), **Django** (administration of front-end), **Web3.js** (communication with Ethereum blockchain/sidechain and smart contracts).
- Databases - **MongoDB/RocksDB** (databases management), Apache Kafka (databases and other systems communication).
- Agreements of off-chain storage - **MongoDB/RocksDB** (databases management), **Apache Kafka** (databases and other systems communication).

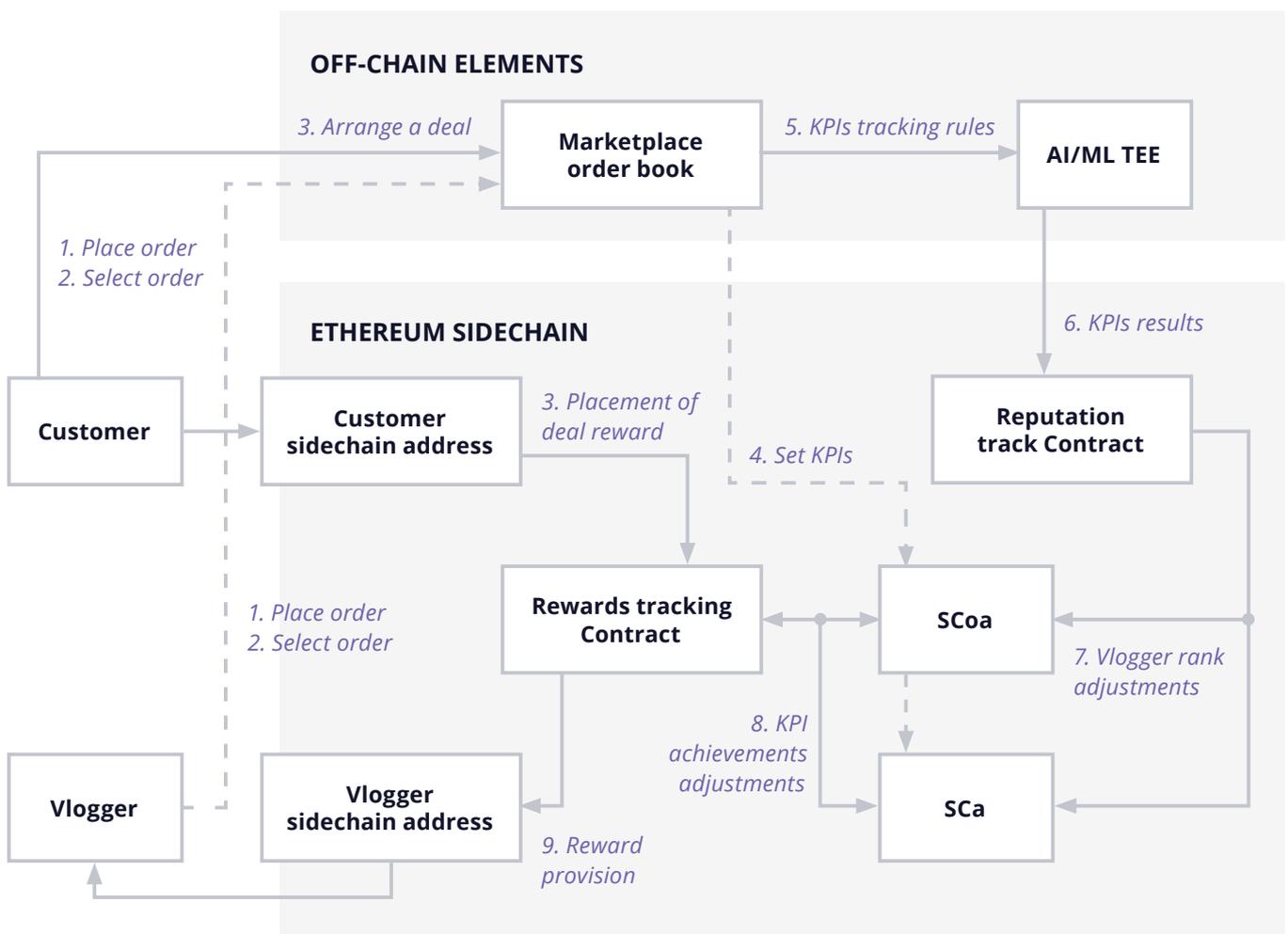
- Back-end of off-chain elements - **Spring** (support development stack at each level whether security, messaging, data handling), **Hibernate** (ORM tool to map Java classes to Database tables).
- Communication with 3rd party provider systems - **JSON RESTful API, Docker**.

Every communication channel that is used within the platform (between nodes, data storage, users dApp, etc.) is encrypted and secured by SSL/TLS certificates to establish fully encrypted channels within on/off-chain infrastructures.

■ PROTOCOL REWARDS TRACKING SMART CONTRACT

Rewards tracking smart contract is a core element of the AQER protocol kernel. This contract is represented as a template in the form of a smart contract fabric function. The core element of it is its ability to be re-deployed each time that a deal between a customer and a vlogger is started.

STRUCTURE OF REWARD TRACKING SMART CONTRACT



The Logic Flow of the Reward Tracking Smart Contract:

- 1 The customer places an order for the vlogger's services or the vlogger places their offer in the marketplace of the AQER platform.
- 2 The vlogger takes an order or the customer selects a vlogger offer based on the rewards amount and proposed KPI's. KPI's communicate at a platform level and are negotiated by both sides.
- 3 After KPI's selection, the customer provides ERC20 sidechain tokens to the reward smart contract as proof that the reward exists, which is also reflected in the marketplace.
- 4 KPI's are selected during the deal negotiations and are programmed for dedicated advertising and organic advertising smart contracts.
- 5 As KPI's are selected, marketplace back-end forwards the description of them to AI/ML TEE, which is responsible for their evaluation.
- 6 Once processed by the AI/ML TEE, the KPI's results are transmitted to Reputation Track SC and SCoa and SCa as proof of KPI's achievements.
- 7 Reputation track smart contracts collect data about the vlogger reputation rank and provide this data to reward distribution contract.
- 8 SCa and SCoa contracts provide results of KPI's achievement that indicates event for rewards release.
- 9 Rewards tracking smart contract obtains the data and releases reward based on achieved KPI's.

Levels of rewards depend on types of set KPI's while rewards tracking smart contract have timestamps, which will release funds back to the customer in case the vloggers cannot reach any of the ordered targets within the dedicated amount of time.

■ PROTOCOL SMART CONTRACT FOR KPI'S

Smart contract for advertising and smart contract for organic advertising are functional contracts for AQER protocol. Those contracts are used for KPI's setting and tracking of their achievements. Rewards tracking smart contract listens to them and distributes rewards only when certain milestones are achieved.

KPI's setting process is obtained from the marketplace at the platform, while achievement of them is related to obtaining from AI/ML TEE. Those smart contracts have no actual storage of internal ERC20 tokens and provide only a distribution of tokens.

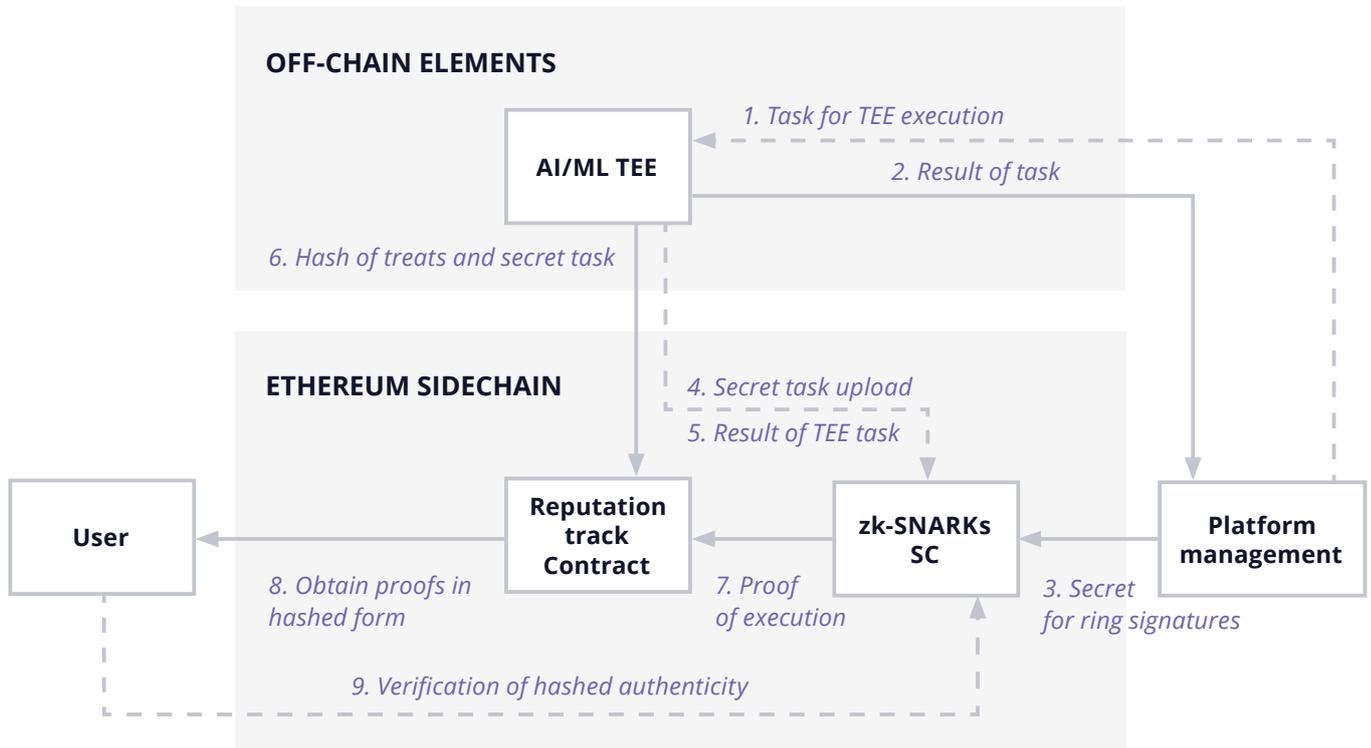
Another valuable smart contract is a reputation tracking smart contract, which executes two major functions:

- Adjust rewards for KPI's achievements based on vloggers reputational changes.
- Store proofs of unchangeable ML treats.

Adjustments of rewards based on the reputation of vloggers required in case of multiple sides rewarding, when several customers or several vloggers are parties to the contract. In that case, reputational tracking of the performance of vloggers is an exceptional value. It can occur in the case when weights of results per each party change.

As it hardly relies on AI/ML TEE computational results, this smart contract must also contain proofs that distribution and ranking are trustless. As an exact result of TEE, computations must be protected from external view, a specialized approach for proving is applied. At the moment of deal creation, a hash of ML treated is stored inside the smart contract together with the hash of the computational task for AI/ML TEE with its exact result. At this stage, the ring signature is required to use zk-SNARKs smart contract that keeps a secret for TEE to prove its immutability.

REWARD TRACKING SC



System Flow of Trust Verification:

- 1** Each day, a secret task uploaded to TEE for execution.
- 2** The result of the secret task computation is obtained by management and stored in hashed form publicly.
- 3** Task results in a mutable form stored in zk-SNARKs smart contract as a secret for ring signatures verification.
- 4** zk-SNARKs smart contract verifies secret with stored hash by uploading the secret task to TEE.
- 5** Hash of secret task computation provided to zk-SNARKs smart contract for verification of TEE respectful execution of the task.
- 6** Hash of ML treats and hash of secret task is stored in the reputation track smart contract.

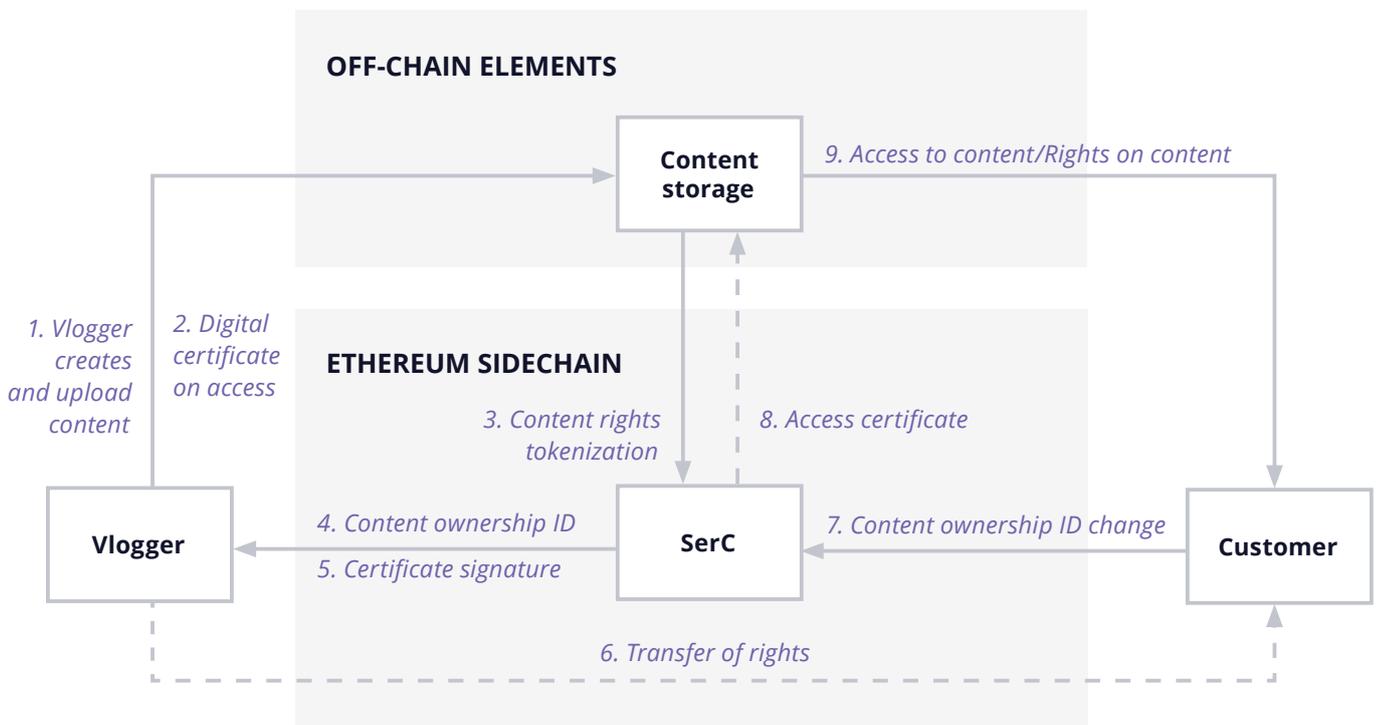
- 7** Proof of respectful execution is provided to reputation track smart contract with proofs of ML treats hash.
- 8** Any user can obtain hashed values from reputation track smart contract.
- 9** The user can request zk-SNARKs smart contracts to prove the hash secret task execution and a hash of ML treats remain unchanged.

As AI/ML TEE computational possibilities are growing with time, secret task complexity is growing, which makes this system similar to the mining process. To add more security to the system, ML treats hashes will be stored directly in sidechain blocks with dummy transactions of sidechain native Ether where a hash of previous ML treat will be concatenated with a new hash of ML treats. This allows for the creation of a sequence of connected hashed values and ensures that the new hash is not based on the newly launched AI/ML TEE.

■ PROTOCOL CONTENT RIGHTS SMART CONTRACT

As any content created by vloggers is distributed to the platform under the common licensing agreement, the AQER protocol has an element that allows the vlogger to participate in generated by them content rights management. This ability requires the tokenization of such rights that is possible only with a non-fungible standard token of the Ethereum network - ERC721 standard.

CONTENT TOKENIZATION



Key Elements of Content Rights Management:

- 1 Each vlogger uploads content to the platform storage under a license agreement.
- 2 Storage premises is using out of the box digital certificates for identity and access management.
- 3 The content is tokenized in the ERC721 token that has a unique value of content, ownership ID data and credentials to verify digital certificates on access to content (level of permissions to manage content).
- 4 The vlogger claims their exclusive rights to the content by using their sidechain wallet and providing their Owner_ID to the ERC721 token.
- 5 The vlogger's wallet is used to provide certificate credentials to verify access to content based on the validity of their Owner_ID.
- 6 The vlogger in off-channels forwards the rights on content to another party with the provision of their certificate public key on access and exclusive rights to content, SCoA and SCA as proof of KPI's achievements.
- 7 Another party receives the ERC721 token to their wallet with changes Owner_ID in it that validates the signature under the certificate on exclusive access to content and permissions to manage it.
- 8 Owner_ID is used to verify the validity of the certificate in content storage premises.
- 9 Content storage load balancer allows another party to manage content under exclusive rights in case of the valid digital certificate on access.

For such an approach, content tokenization and access validation are critical elements. As any content will fall under copyright law, it means that exclusive rights are granted to the creator at the moment of content creation and its particular form. That means that the ERC721 token must contain proofs of existence with the time and immutability of the content.

For that, the following approach for tokenization is used:

- 1 The vlogger request the sidechain to issue an ERC721 token to their wallet.
- 2 The ERC721 token is accepted by the vlogger's wallet as empty inside, but with the token ID and with indication of Owner_ID of vlogger's hash of private key (or master private key in case of HD wallets) derived from Ethereum elliptic curve (ECDSA).
- 3 The vlogger use AQER platform back-end to hash content and put it in as a metadata inside the ERC721 token. In the case of several pieces of content, their hashes are concatenated in one hash (with function of SHA-256).
- 4 With the content uploaded to the platform storage, the vlogger receives a digital certificate with credentials on permissions to manage the content selected by vloggers during content upload.
- 5 A hash of certificate concatenate with hash of content and sign with the same signature as a certificate.

- 6** In the case of content for sale, the vlogger indicates the value of the ERC721 token in internal Ethers of sidechain or as a zero if it is only a rights storage.
- 7** After creation of ERC721 token with all of the metadata inside it, the token ID is stored in platform registry and filed inside the common licence agreement to be signed with the digital signature of the vlogger.

With the token ID in the licence, the ERC721 token becomes a tokenized content right of the vlogger. Requirement to fill hash of digital certificate and signature under the hash is for the simplification of content permissions management.

Identity and access management with ERC721 token follows on from this:

- 1** Certificates have a credential that requires it to match ERC721 token Owner_ID with signature under certificate in X509 hierarchy.
- 2** If Owner_ID belongs to other signature that request access in the same X509 self-signed hierarchy.
- 3** If Owner_ID and signature of permissioned certificate are a match, that owner of ERC721 obtains access to content in storage with permissions that are allowed by those certificates.

AQER PLATFORM OFF-CHAIN ELEMENTS

AQER platform off-chain system consists of three major elements:

- Content storage.
- Marketplace.
- AI/ML TEE that is also an element of the AQER protocol.

To ensure platform flexibility and to ensure stability and latency of the system, both content storage and the marketplace are cloud-based systems with a server-less approach behind them.

Content storage in an on-demand server-less technology initially built with MongoDB. With platform growth and as the amount of content storage increase, the system will migrate to a server-based storage system with RocksDB.

For the marketplace, a cloud-based serverless technology with load balancer is applied and out-of-the-box digital certificates for identity and access Management.

The platform uses two types of certificates:

- General certificates for access to user cabinet at marketplace level
- Hierarchical self-signed X509 certificates for content rights management.

Signatures for certificates are the same, but credentials in certificate metadata are different as well as dedicated to those permissions.

As the platform grows, a stable connection with servers will be established with matches to the the marketplace front-end by Apache Kafka to manage messages between servers and increase the latency of the system.

AI/ML TEE is placed in cloud intel container, which provides TEE for computations. With growth of ML treats, Apache Spark will be used to increase performance between several containers through the Map-Reduce function for parallel computations.

APPENDIX A; PURPOSE OF USING ERC20 TOKENS AND THE TOKEN SALE

■ AQER TOKEN PURPOSE AND SPECIFICS

To satisfy the dual-token nature of the AQER solution architecture, an ERC20 standard token (AQR) will be issued for use on the Ethereum network.

This means that:

- 1** A buyer gains all the benefits of other Ethereum ERC20 tokens.
- 2** The AQER tokens are running on the Ethereum mainnet, with all of its consequent fees for mining and smart contract processing (gas price defined by the Ethereum mainnet).
- 3** The AQER tokens are used for the initial wide distribution (market penetration) and to exchange AQER token into sidechain tokens through the gateway on the AQER platform.
- 4** The sidechain tokens are based on the D-minting³ basis, where new tokens are minted for exchange out of AQER (for the exchange of AQER into sidechain tokens). This allows limiting the initial supply of token while giving flexibility. The internal ERC20 tokens amount cannot exceed AQER tokens balances.

■ TOKEN SALE

The AQER token sale will be held under the general process of a simple issue in the Ethereum mainnet without any additional smart-contracts meaning gas prices are minimized. Consequently, owners of AQER tokens will only pay fees for simple transactions without any other unnecessary gas payments.

The token sale has a fixed number of tokens since bonus tokens are awarded for buying during the sale. Total bonus tokens will depend on the results from all token buyer actions, which is variable.

Benefits of this solution are as follows:

- 1** There is a wide distribution on the market with the ability to use any wallet for ERC20 tokens that supports Ethereum.
- 2** It allows the use of the Ethereum mainnet infrastructure, services and public exchanges.

³ D-minting is a process of Token Generation Event, when part of those tokens remain undistributed and remains on ERC20 smart contract address.

APPENDIX B. TOKENS SPECIFICATION

■ AQER TOKEN

Blockchain:

- Ethereum mainnet.

Type:

- ERC20 standard token.

Usage:

- Public usage with integration to wallets that support Ether.

Supply specific:

- A fixed amount of the token to enforce community support with the growth of incentivisation.
- Tokens are unburnable.

Value:

- Defined by market.

■ AQER TOKEN

Blockchain:

- Ethereum sidechain.

Type:

- ERC721 and ERC20 standards Ethereum tokens.

Usage:

- Sidechain reward distribution arrangements.
- Content rights management.

Supply specific:

- Amount related to deposited AQER tokens in the dedicated smart contract at mainnet.

Value:

- Related to the current exchange rate of AQER tokens.