SnapScale

# TECHNICAL WHITEPAPER

## Abstract

SnapScale is a DLT (Distributed Ledger Technology)-solution running on edge computing servers and powering the Internet of Things (IoT). SnapScale leverages mobile operators' 5G and Multi-Access Edge Computing (MEC) infrastructure to craft a consortium chain and enable optimized MEC resource management, secure data sharing, interconnection and interoperability of all types of IoT devices.

SnapScale aims to build an intelligent and distributed IoT ecosystem at the network edge through partnering with Mobile Network Operators (MNO), enterprises and OTT (Over The Top) service providers.

# 1 Project background

## 1.1 Challenges

The last few years have seen the Internet of Things (IoT) grow exponentially, from a concept to a major priority for a number of organizations. Yet, some major challenges remain with respect to achieving higher security, privacy, scalability and and interoperability.

**Security**: IoT devices have been proliferating across multiple industries. There is, however, no unified security standard, which leaves massive potential for hackers to take over devices, use them for cyberattacks.

**Privacy**: in recent years, the world has experienced scandals due to data breaches and identity thefts. This has led IoT manufacturers & developers to rethink the problems of data ownership and monetization, while striving to preserve security and business interests.

**Scalability**: Cloud computing is still the foundation of the IoT. In parallel to the dramatic rise of IoT sensors, operation and maintenance costs of network infrastructure are skyrocketing; traditional cloud computing networks cannot meet the Ultra Reliable Low Latency Communications (URLLC). requirements of applications in fields such as Autonomous Vehicles, Industrial Internet of Things (IIoT) and Augmented Reality / Virtual Reality.

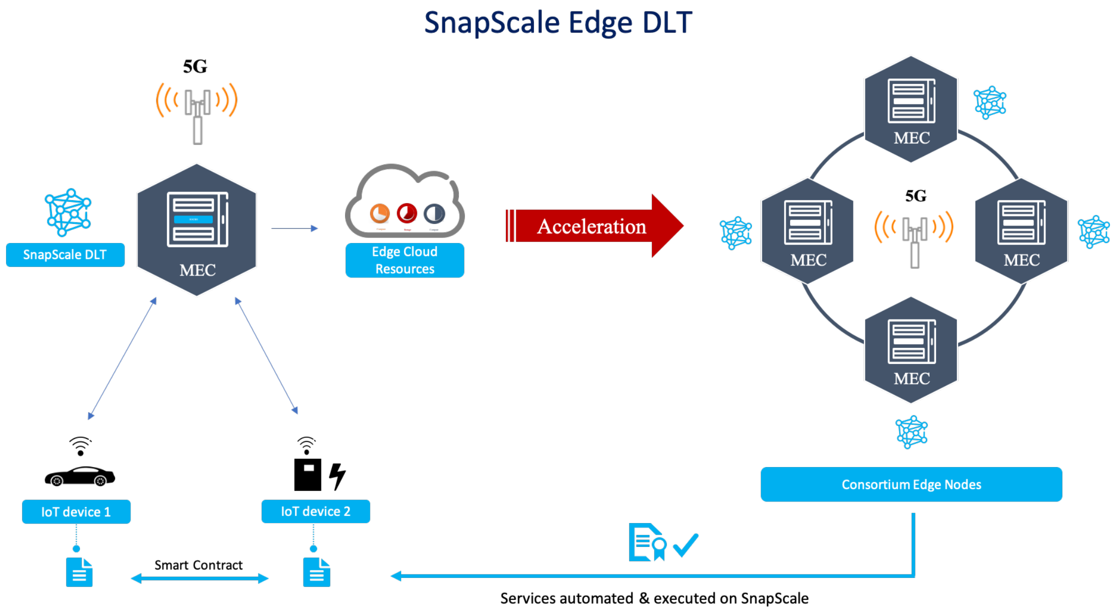
**Interoperability**: A wide array of IoT applications now require collaboration between different parties or stakeholders. The cost of dealing with middlemen to guarantee trust is high and incurs risk, while reducing the efficiency of communication and interoperability between various agents.

## 1.2 SnapScale-based Edge Computing solution for IoT

5G is the new global wireless standard for telecommunications. As such, it is meant to deliver a set of new capabilities, which includes: Enhanced Mobile Broadband (eMBB), Massive Machine Type Communications (mMTC), Ultra Reliable Low Latency Communications (URLLC). With the help of 5G technology, Autonomous Vehicles, Virtual Reality (VR), Augmented Reality (AR), IIoT, Smart Healthcare and other applications that require ultra-low latency are set to become reality.

SnapScale leverages edge computing technology to reduce service latency and improve Quality of Service (QoS), which eventually results in improved user-experience. Traditional cloud computing networks are highly centralized, with data being gathered at the outermost edge and sent to the centralized main servers for processing before being transmitted back to the terminal. Edge computing, however, provides cloud-like computing, storage and communication facilities closer to where data is generated. Since the information need not travel as far as it would under a traditional cloud architecture, network latency is significantly reduced, which is crucial in the context of 5G applications.

As shown in the following diagram: SnapScale DLT resides in the MEC server. Nearby IoT devices use smart contracts to execute automated operations such as identity checks, trusted data management (TDM) and Machines to Machines (M2M) payments. All the data is transferred to other MEC servers through consortium edge nodes. In the near future, mobile network operators and third-party certification agencies will be able to join the consortium as edge nodes to provide services such as identity authentication and cross-operator communication.



A blockchain is a distributed digital ledger that is used to record events and transactions in an immutable manner. SnapScale's blockchain (Distributed Ledger Technology) allows IoT devices and edge computing nodes to perform peer-to-peer (P2P) transactions and avoid data falsification or tampering from centralized organizations while reducing the cost of trust. More importantly, it permits trusted data sharing between IoT devices across multiple platforms, bridging the gaps between traditional IoT networks and breaking down silos.

IoT is distributed by nature: billions of devices are geographically spread over multiple networks. Blockchain and edge computing architectures are also both distributed. SnapScale's solution represents the perfect fit to combine and leverage IoT, blockchain and edge computing distributed characteristics. The blockchain platform is deployed through the MEC server as a core service and can be integrated with various edge applications, giving full play to data security, CA (Certificate Authority) certification and privacy protection. Meanwhile, edge computing provides low-latency capabilities and lightning-fast cloud environment at the edge for blockchain nodes to improve performance.

The SnapScale solution boasts the following characteristics:

### Distributed account system

According to GSMA Intelligence, the world's number of IoT connections (cellular and non-cellular) will exceed 25.2 billion in 2025. Present day network infrastructure is inadequate in processing large chunks of data flows relative to registrations, authentications and transactions between these devices. SnapScale will build a global distributed account system to record and report such events.

### Automated transactions between trusted devices

Smart contracts can be easily deployed through SDK calls, enabling a large number of autonomous IoT devices to perform trusted transactions within distributed applications (DApps), allowing for a significant drop in the cost of trust.

### Safety

Centralized databases cannot fundamentally guarantee security, data integrity and lawful usage of data. The decentralized and P2P nature of the blockchain network essentially makes data tampering impossible.

### Data traceability and auditability

The blockchain properties such as tamper-proofness and availability ensure that on-chain data can be traced through timestamps, cryptographic hashes and mathematical consensus algorithms, thereby increasing the difficulty of counterfeiting and alleviating the need of arduous audit processes.

### High performance and scalability

SnapScale adopts a hybrid dual-chain governance structure and coexists with a dedicated FPGA hardware acceleration device which permits to improve overall blockchain performance. Hybrid governance structure is instrumental in reducing congestion on the main chain, balancing SnapScale's workload and improving the efficiency of transaction processing between IoT devices. SnapScale's associated acceleration device supports the offloading of intensive computational tasks away from the MEC CPU, dynamically mobilizing the task load and executing application acceleration when needed.

# 2 Design

## 2.1 Design Concepts

SnapScale is an integrated blockchain and edge computing solution for IoT. It was designed to address recurring pain-points, among which data security, data circulation and data sharing. SnapScale is dedicated to establishing a consortium platform for MEC resource orchestration and management, aiming to connect together IoT devices of any types.

SnapScale uses cloud-native edge architecture to replicate cloud-native microservice management capabilities and ecosystem to the edge. Developers can implement one-button smart contract development on SnapScale consortium. At the same time, edge computing provides reliable, low-latency service by bringing computing closer to the data source.

In the future, the solution will feature a hardware acceleration device combining software and hardware to break performance bottlenecks and achieve an easy-to-use and high-performance blockchain platform.

## 2.2 Design principles

### Modularity

SnapScale adopts a modular design principle in order to reduce software complexity and to simplify the operation of program design, debugging and maintenance.

### Decoupling

The core objective of the Blockchain 3.0 era is "decoupling". Interactions between main chain and sub-chains aim to decouple functions and increase performances. SnapScale strives to achieve decoupling through different functional layers.

### Compatibility

SnapScale’s principle of compatibility enables different application developers to integrate quickly and easily. For example, the use of common data communication and network standards and the account system design can meet different needs in most scenarios.

### Pluggability

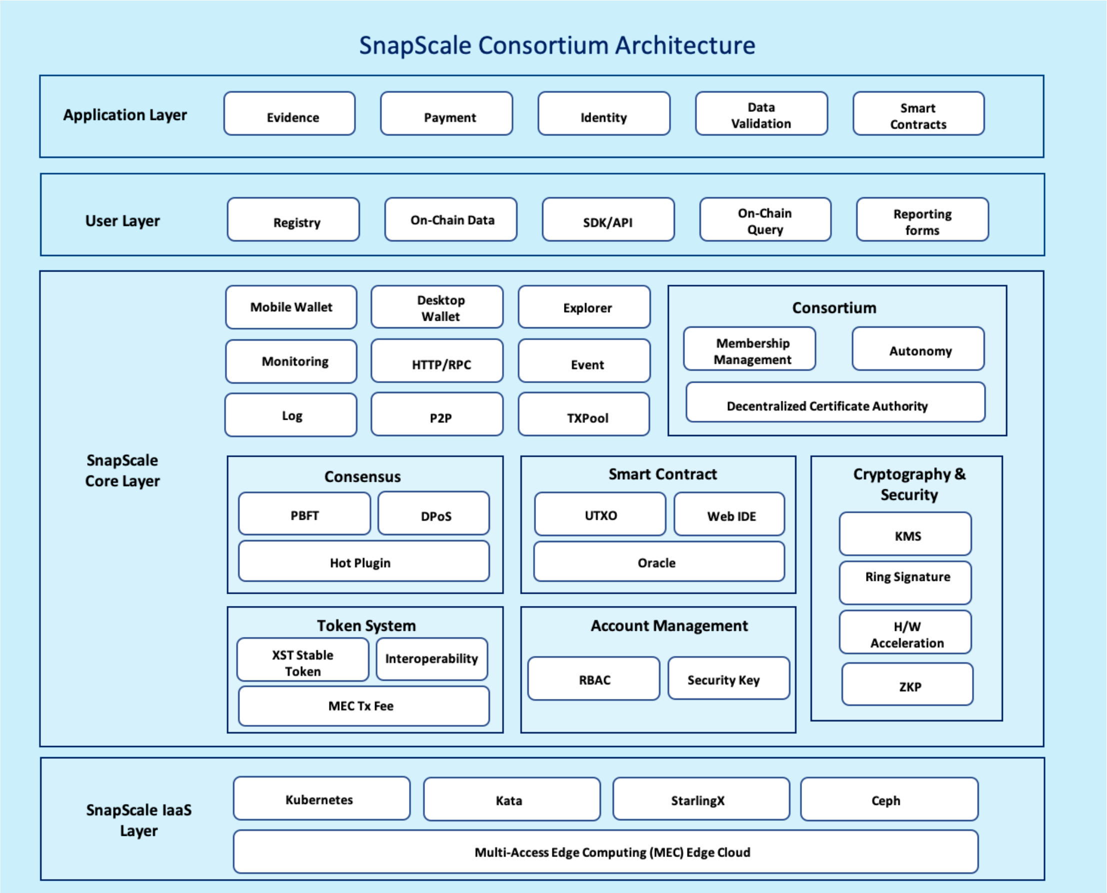
A plug-in is a program module that can be independently developed, dynamically embedded and freely deleted and replaced. Therefore, it can improve the parallelism to speed up software development efficiency, reduce the difficulty of design and development, shorten the development cycle, and further enhance application programs’ operability, testability and maintainability.

### Safety and scalability

Safety and scalability should be amongst the first design principles when building any blockchain system. SnapScale effectively improves scalability to meet the requirements of billions of IoT devices while giving utmost attention to security constraints.

# 3 SnapScale Consortium Architecture

The SnapScale Consortium architecture consists of the application layer, user layer, blockchain core layer and blockchain infrastructure layer. As shown below:



## 3.1 DPOS+Pipelined BFT Consensus Mechanism

SnapScale is currently operating on a DPoS + Pipelined BFT consensus mechanism

**DPoS – How to be block producer**

DPoS（Delegated Proof of Stake）introduces a voting system that requires token holders to vote for “delegates”: these delegates are essentially "supernodes", among which 21 are eventually selected to be Block Producer (BP), ensuring continuous maintenance and update within the SnapScale network.

### Block production mechanism

Blocks are produced every 0.5 second. In each round of blocks, every BP is responsible for producing 12 blocks with a 0.5 second block time. Thus, each round contains 252 blocks (21 BPs * 12 blocks each). BPs are selected prior to the beginning of each round of blocks. In order to avoid missing new blocks due to excessive blockout speed or commutation latency, SnapScale adopts a predefined block production order, which means BP that needs lowest communication latency could be successive block producers.

### Pipelined BFT - block confirmation

Under the traditional PBFT (Practical Byzantine Fault Tolerance) mechanism, new blocks are generated at regular intervals and the following blocks can only be produced once the "ancestor" has been confirmed by all validators. The process can be relatively lengthy and meeting the 0.5s-blocktime requirement may be arduous.

SnapScale uses a pipelined BFT (Pipelined Byzantine Fault Tolerance) algorithm so block generation and consensus can be processed in parallel. Once a block is generated, it does not need to wait for network confirmation. Instead, the BP continues to produce blocks while participating in the PBFT-confirmation of the previous block. Following confirmation, a new block is deemed irreversible.

### Punishment mechanism

Only one block producer is authorized to generate blocks at a given time. If a block is not successfully generated within a certain time, it must be skipped. Any block producer that misses a block while having failed to generate any block in the last 24 hours will be automatically removed from the BP list. This contributes to ensuring the smooth operation of the network.

### Fork or not

Under normal circumstances, the DPoS blockchain will not be forked because block producers cooperate in producing blocks rather than they compete. Should a fork attempt occur, consensus will automatically migrate to the longest chain. The more block producers, the faster the chain length growth. In addition, delegates are not allowed to concurrently produce blocks on two different forks. If such behavior block producer has been founded done so, it may be voted out.

## 3.2 Smart Contracts

Smart Contracts allow trusted transactions without third parties. Such transactions are traceable and irreversible. Smart Contracts are written in C++ and the Smart Contract SDK integrates the following commands and capabilities:

- Account API: queries account data

- Chain API: queries the internal state of the chain
- Database API: Stores and retrieve blockchain data
- Math API: Defines commonly used mathematical functions
- Action API: Defines the API for querying operation attributes
- Memory API: Defines commonly used memory functions
- Console API: Enables applications to record / print text messages
- System API: Defines the API used to interact with system-level internal functions
- Token API: Defines the ABI for interacting with standard-compliant token messages and database tables
- Transaction API: Defines the API for sending transactions and internal messages

## 3.3 UTXO Model

UTXO (Unspent Transaction Output) model was first introduced by Satoshi Nakamoto as part of the Bitcoin original whitepaper. This model suggests that the total of all individual transaction outputs can be spend by users. When a user sends a certain amount to a different address, the system will generate a new UTXO to the receiver’s wallet. If the receiver wants to “switch” to being payer, his or her address will be automatically displayed under the payer's account. When initiating a transaction, one must ensure that all of the UTXO outputs are used.

SnapScale supports both the account balance and UTXO models. Transactions can be made within and between these two models, which ultimately serves as the basis for SnapScale’s anonymous transaction model.

## 3.4 Anonymous Transaction Model

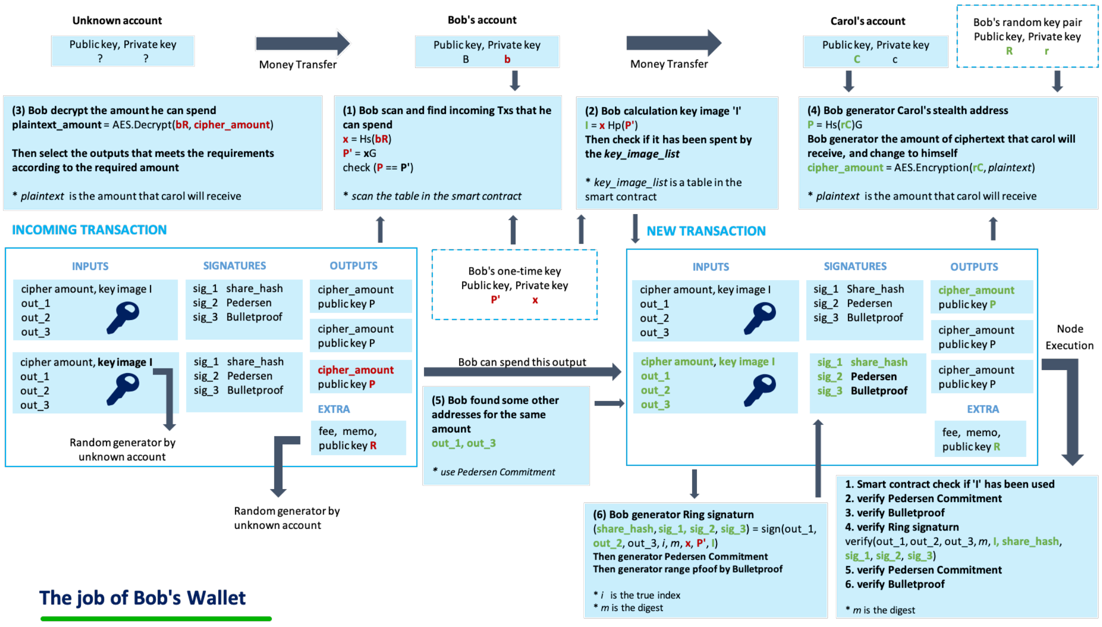
SnapScale combines UTXO model with Ring Signature to achieve anonymous transactions, completely hide the sender, receiver and amount of the transaction, which cannot be tracked. SnapScale supports both anonymous and transparent transactions and the four transaction types are as follows:

| Sender | Receiver | Amount | Anonymity |
|---|---|---|---|
| Account | Account | Visible | Transparent |
| Account | Address | Visible | Receiver Invisible |
| Address | Address | Invisible | Completely anonymous |
| Address | Account | Tracable | Sender Invisible |

The problem that the ring signature solves is that if Person A says a word to Person B, other people can know that someone within a group of people said this sentence to Person B, but they cannot detect who that person is. It applies to the signing party whose computing power is weak.

SnapScale also uses Key Image to avoid double-spending and zero-knowledge proof (ZKP) to solve the problem of proof of range after the transaction amount is hidden.

The following diagram is the process in which the sender Bob initiates a transaction to the receiver Carol, and Bob uses cryptographic tools to generate the data required for the transaction:



**(3) Bob decrypt the amount he can spend**
**plaintext_amount** = AES.Decrypt(**bR**, **cipher_amount**)

**Then select the outputs that meets the requirements according to the required amount**

* *plaintext* is the amount that carol will receive

**(1) Bob scan and find incoming Txs that he can spend**
**x** = Hs(**bR**)
**P'** = **x**G
check (**P** == **P'**)

* *scan the table in the smart contract*

**(2) Bob calculation key image 'I'**
I = **x** Hp(**P'**)
**Then check if it has been spent by the *key_image_list***

* *key_image_list* is a table in the smart contract

**(4) Bob generator Carol's stealth address**
P = Hs(rC)G
**Bob generator the amount of ciphertext that carol will receive, and change to himself**
cipher_amount = AES.Encryption(rC, *plaintext*)

* *plaintext* is the amount that carol will receive

**(5) Bob found some other addresses for the same amount**
**out_1, out_3**

* *use Pedersen Commitment*

**(6) Bob generator Ring signaturn**
(share_hash, sig_1, sig_2, sig_3) = sign(out_1, out_2, out_3, *i*, *m*, **x**, **P'**, I)
**Then generator Pedersen Commitment**
**Then generator range pfoof by Bulletproof**

* *i* is the true index
* *m* is the digest

**1. Smart contract check if 'I' has been used**
**2. verify Pedersen Commitment**
**3. verify Bulletproof**
**4. verify Ring signaturn**
verify(out_1, out_2, out_3, *m*, I, share_hash, sig_1, sig_2, sig_3)
**5. verify Pedersen Commitment**
**6. verify Bulletproof**

* *m* is the digest

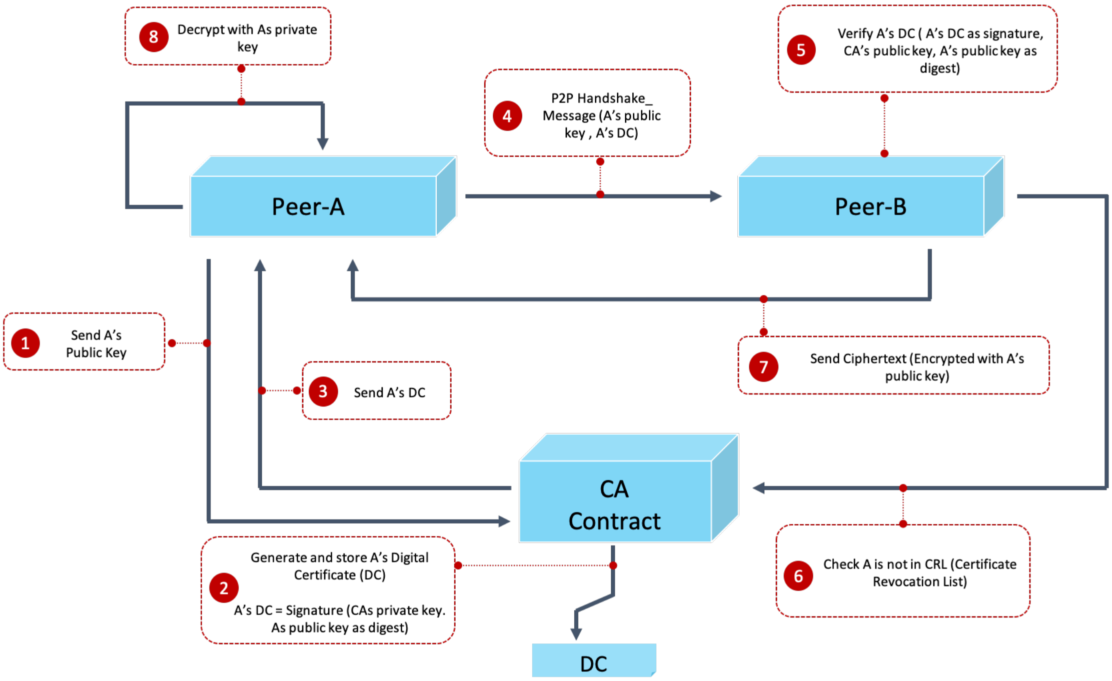
Cryptographic algorithms related to anonymous transactions:

- Hide the originator of the transaction: Borromean Ring Signature, Key Image
- Hide the recipient of the transaction: ECDH, Stealth Address
- Hide the transaction amount: ECDH, Pedersen Commitment, Bulletproofs (ZKP)

## 3.5 Consortium Model

MEC nodes and individual users who join the SnapScale consortium must register and obtain a certificate issued by a Certificate Authority (CA).

### 3.5.1 How does a node access the consortium



8 Decrypt with As private key
4 P2P Handshake_ Message (A's public key , A's DC)
5 Verify A's DC ( A's DC as signature, CA's public key, A's public key as digest)
Peer-A
Peer-B
1 Send A's Public Key
3 Send A's DC
7 Send Ciphertext (Encrypted with A's public key)
CA Contract
2 Generate and store A's Digital Certificate (DC)
A's DC = Signature (CAs private key. As public key as digest)
6 Check A is not in CRL (Certificate Revocation List)
DC

Process description:

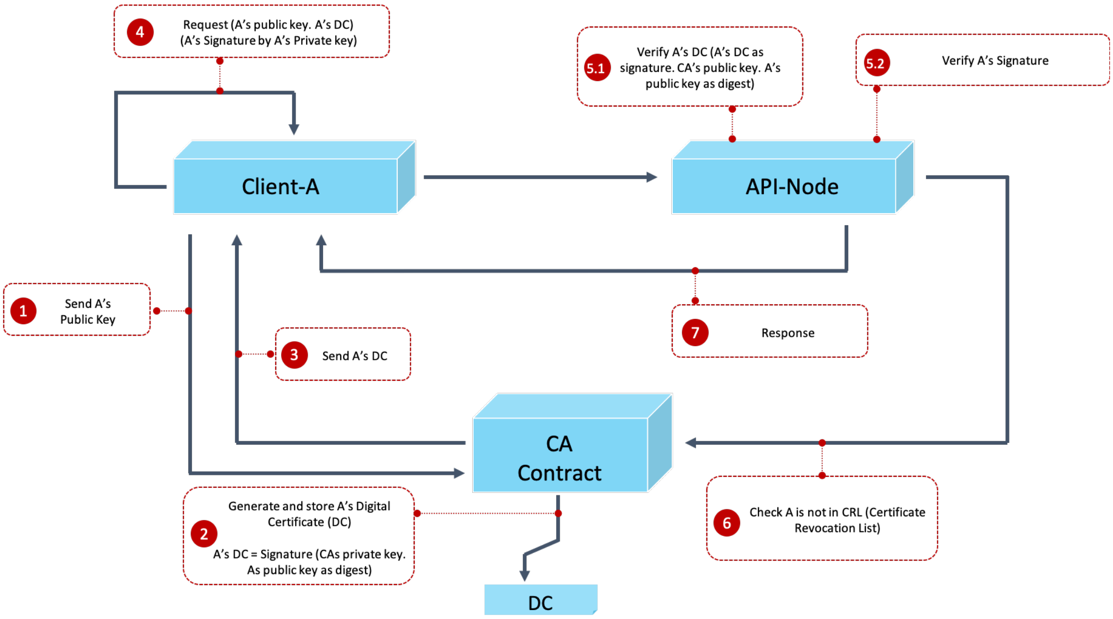
Organization A wants to set up a new node

- If A does not have an account, it must register first
- Organization A needs to prepare a set of key pairs (public key and private key)
  - key pairs are simultaneously generated when registering an account
  - or A imports an existing set of key pairs

1. A files an application for joining and submits it to CA (main application component: A's public key within SnapScale's consortium)
2. CA processes A's application request, and, upon agreement to the request, will automatically generate a Digital Certificate (DC) for A's public key

3. CA sends A’s DC
4. A configures CA’s public key, A’s public key, A’s private key, A's DC and IP (one or more nodes, maybe PeerB’s ) in the consortium, and initiates his or her own node PeerA to connect to PeerB
5. PeerB verifies PeerA’s DC
6. PeerB checks whether PeerA is in Certificate Revocation List (CRL) or not
7. After PeerA and PeerB are successfully connected, PeerA uses encrypted channels to exchange data with Peer B

### 3.5.2 How does the client access the consortium


4 Request (A’s public key. A’s DC) (A’s Signature by A’s Private key)
5.1 Verify A’s DC (A’s DC as signature. CA’s public key. A’s public key as digest)
5.2 Verify A’s Signature
Client-A
API-Node
1 Send A’s Public Key
3 Send A’s DC
7 Response
CA Contract
2 Generate and store A’s Digital Certificate (DC)
A’s DC = Signature (CAs private key. As public key as digest)
6 Check A is not in CRL (Certificate Revocation List)
DC

Process description:

Client A uses the wallet to generate key pairs (public key and private key)

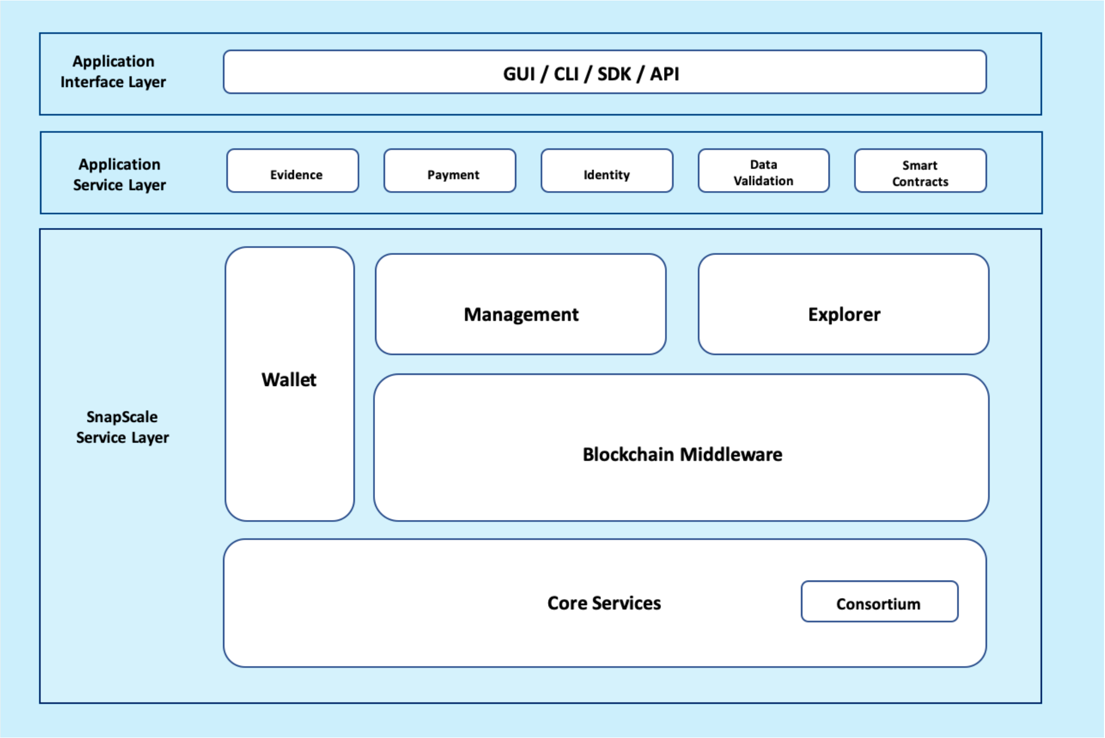
1. Client A sends the generated public key to the CA certificate management department to submit application to join consortium;
2. CA certificate management department conducts a preliminary review of the information, and a DC is generated for A’s public key if the review is passed;
3. CA sends A’s DC
4. A uses his own public key, private key, his own DC and the public key of the API node in the consortium to generate an Http header token and initiate a request;
5. API node verification;
    - API node verifies A’s DC
    - API node verifies A’s signature
6. The API node checks whether A is in the CRL or not
7. The API node responds to A’s request;

## 3.6 Reward model

SnapScale's reward model is based on configuration setting and transaction fees, such as CPU, memory and net resources consumption. The charged fee will be allocated to edge nodes as well as blockchain nodes for block generation. The first version of the detailed reward model will be shown in the upcoming white paper.

# 4 Product architecture

The SnapScale product architecture consists of SnapScale DLT Service Layer, Application Service Layer, and Application Interface Layer. The SnapScale DLT Service Layer includes: core services (distributed ledger, consensus mechanism, encryption and authorization technology, smart contracts), wallets, explorer, blockchain middleware and blockchain management platform. The Application Layer includes evidence services, payment services and authentication services, etc. The Interface Layer provides various interfaces for different users, such as GUI, CLI, SDK and API.



# 4.1 Service Layer

## 4.1.1 Core services

**Distributed ledger**

Blockchain is essentially a distributed ledger and is jointly maintained by the nodes of the consortium. Each node replicates and saves an identical copy of the ledger which allows for mutual supervision between nodes and guarantees data immutability.

**Consensus mechanism**

The consensus mechanism is an algorithm used to reach agreement between different validators on the data to be recorded on the chain and propagated through the network. Consensus ensures that the latest blocks are rightfully added to the blockchain and that different nodes keep their instance of the ledger consistent with each other. Furthermore, it helps in preventing several types of malicious attacks.

**Encryption and authorization technology**

Part of the transaction information stored on the blockchain is public, but the account ident information is encrypted. Data can only be called through API and checked after obtaining authorization from the data owner, thereby safeguarding data security and personal privacy.

**Smart contracts**

Based on these credible and non-tamperable data, agreements are automatically enforced when the pre-defined rules are met.

## 4.1.2 Wallet

The wallet is a tool for users to manage keys and digital assets (tokens), and users will use tokens to obtain services in the SnapScale ecosystem. The wallet contains a collection of key pairs, each consisting of a private key and a public key. Private keys are used to sign and verify transactions to prove that users in fact have ownership of those funds. While, the public key is used to verify the signature using a corresponding private key.

### 4.1.3 Explorer

Individual users, enterprises, MON, OTT service providers and other stakeholders can query the transaction information recorded on SnapScale through the explorer. This information includes: chain information, block information, transaction information and contact information, etc. Every transaction occurring on the IoT that has already authorized can be queried in the explorer, enabling transaction data to be transparent and traceable.

### 4.1.4 Blockchain middleware

Blockchain middleware consists the following functions:

- Making interaction within blockchain core services more convenient;
- Providing more secure access to blockchain core services;
- Provide blockchain statistics information in different dimensions;

### 4.1.5 Blockchain management platform

SnapScale presents a one-button blockchain management and deployment platform where developers don't need to worry about complex configuration and could easily perform development, testing and source code debugging of smart contracts by just a few clicks of the mouse, greatly improving development efficiency.

SnapScale supports easy development and deployment of consortium blockchain. MNO and OTT service providers can join the consortium as chain nodes. The entire network is jointly maintained by consortium members, and other nodes (such as enterprises) can only join the consortium after the authorization. Detailed consortium architecture and rules will be released in the next version.

## 4.2 Application Layer

**Evidence Service**

Relying on the blockchain's timestamp and hash encryption technology, SnapScale provides evidence services for data and digital files. Once the data or file is uploaded tp SnapScale, a specific "digital fingerprint" will be generated and recorded it, which will never be tampered with. SnapScale evidence system can monitor the entire life cycle of electronic data and is a suitable deployment platform for application such as copyright protection and food traceability.

**Payment service**

A reliable, secure and interoperable payment system for data transactions and sharing between smart devices will be provided on top of SnapScale. The payment system adopts friendly and flexible charging standards. At the same time, SnapScale's high-performance, high-reliability, and high-security can support millions of concurrent IoT devices.

**Authentication service**

Smart IoT devices connected to SnapScale will achieve decentralized identity management through distributed identification and verifiable claims. Cryptographic technologies such as digital signatures and zero-knowledge proofs can further prevent privacy infringement and enable users to truly control data ownership.

## 4.3 Interface layer

All functions within SnapScale ecosystem can be achieved through the interface layer. Developers or application components only need to call relevant interfaces to build and execute blockchain applications, regardless of the complexity of the underlying technology.

# Conclusion

SnapScale is committed to exploring the infinite potentials of “5G + DLT” at the edge and crafting a new IoT ecosystem with operators and OTT service partners.

SnapScale is a mobile operator-focused consortium that utilizes Distributed Ledger Technology (DLT) and a variety of protocols articulated for performance and scale within a Multi-access Edge Computing (MEC) environment. SnapScale's standardized architecture and set of interfaces empowers edge infrastructure owners with the capacity to process any interactions, transactions or data flows between IoT devices at the edge, while combining Security, Scalability with an optimal level of decentralization. This newborn platform aims to rejuvenate IoT communities, accelerating the on-boarding of decentralized applications (DApps) and unlocking extended monetization capabilities.

Further details will soon be available on SnapScale Foundation website:
https://snapscale.org/