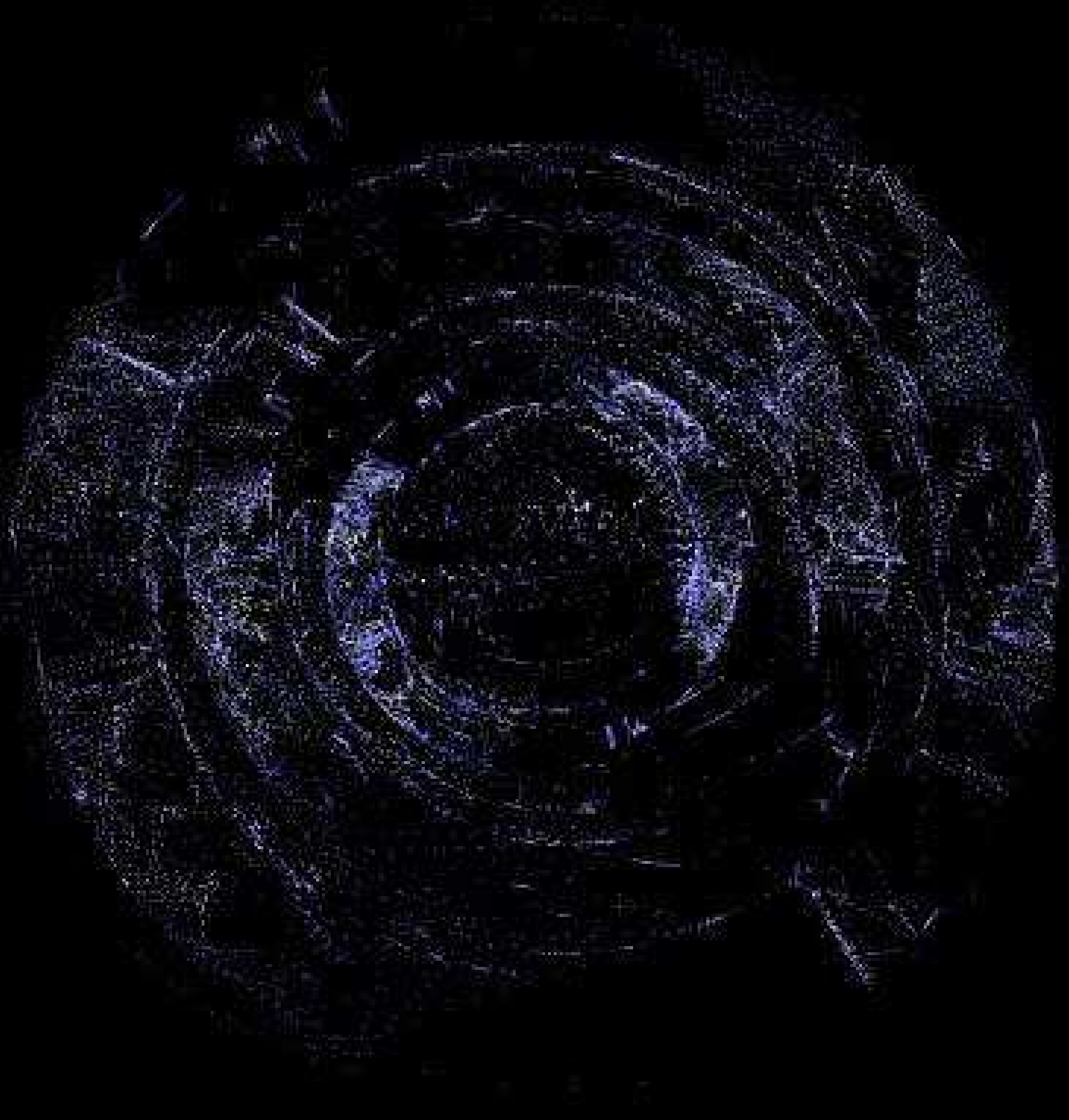
FANTOM

| BEYOND BLOCKCHAIN

Whitepaper

v1.6 | August 28, 2018



https://fantom.foundation

Contents

Disclaimer

This document is a technical white paper that presents the current status and future plans for FANTOM platform and ecosystem of Fantom Foundation Ltd (FANTOM). The sole purpose of this document is to provide information, and is not to provide a precise description on future plans. Unless explicitly stated otherwise, the products and innovative technologies organized in this document are still under development and are yet to be incorporated. FANTOM does not provide a statement of quality assurance or affidavit for the successful development or execution of any of such technologies, innovations, or activities described in this document. Also, within legally permitted scope, FANTOM rejects any liability for quality assurance that is implied by technology or any other methods. No one possesses the right to trust any contents of this document or subsequent inference, and the same applies to any of mutual interactions between FANTOM's technological interactions that are outlined in this document. Notwithstanding any mistake, default, or negligence, FANTOM does not have legal liability on losses or damages that occur because of errors, negligence, or other acts of an individual or groups in relation to this document.

Although information included in this publication were referred from data sources which were deemed to be trusted and reliable by FANTOM, FANTOM does not write any statement of quality assurance, confirmation, or affidavit regarding the accuracy, completeness, and appropriateness of such information. You may not rely on such information, grant rights, or provide solutions to yourself, your employee, creditor, mortgagee, other shareholder, or any other person. Views presented herein indicate current evaluation by the writer of this document, and are not necessarily representative of view of FANTOM. Views reflected herein may change without notice, and do not necessarily comply with the views of FANTOM. FANTOM does not have the obligation to amend, modify, and renew this document, and is not obliged to make notice to its subscribers and recipients if any views, predictions, forecasts, or assumptions in this document change, or any errors arise in the future.

FANTOM, its officers, employees, contractors, and representative do not have any responsibility or liability to any person or recipient (whether by reason of negligence, negligent misstatement or otherwise) arising from any state- ment, opinion or information, expressed or implied, arising out of, contained in or derived from or omitted from this document. Neither FANTOM nor its advisors have independently verified any of the information, including the forecasts, prospects and projections contained in this document.

Each recipient is to rely solely on its own knowledge, investigation, judgment and assessment of the matters which are the subject of this report and any information which is made available in connection with any further investigations and to satisfy him/herself as to the accuracy and completeness of such matters.

While every effort has been made to ensure that statements of facts made in this paper are accurate, and that all estimates, projections, forecasts, prospects, and expression of opinions and other subjective judgments contained in this document are based on the projection that they are reasonable at the time of writing, this document must not be construed as a representation that the matters referred to therein will occur. Any plans, projections or forecasts mentioned in this document may not be achieved due to multiple risk factors including limitation defects in technology developments, initiatives or enforcement of legal regulations, market volatility, sector volatility, corporate actions, or the unavailability of complete and accurate information.

FANTOM may provide hyperlinks to websites of entities mentioned in this paper, but the inclusion of a link does not imply that FANTOM endorses, recommends or approves any material on the linked page or accessible from it. Such linked websites are accessed entirely at your own risk. FANTOM accepts no responsibility whatsoever for any such material, or for consequences of its use.

This document is not directed to, or intended for distribution to or use by, any person or entity who is a citizen or resident of or located in any state, country or other jurisdiction where such distribution, publication, availability or use would be contrary to law or regulation.

This document is only available on www.FANTOM.foundation and may not be redistributed, reproduced or passed on to any other person or published, in part or in whole, for any purpose, without the prior, written consent of FANTOM. The manner of distributing this document may be restricted by law or regulation in certain countries. Persons into whose possession this document may come are required to inform themselves about, and to observe such restrictions. By accessing this document, a recipient hereof agrees to be bound by the foregoing limitations.

This white paper is an information paper subject to update pending final regulatory review. This paper does not constitute an offer. any such offer will be subject to final regulatory review and governed by a revised paper and conditions of sale document that will prevail in the event of any inconsistency with the paper set out below. Accordingly, any eventual decision to buy fantom tokens must only be made following receipt of the final paper, and tokens cannot be purchased until the final paper has been issued by fantom when all final regulatory requirements have been satisfied.

This paper is not a prospectus, product disclosure statement or other regulated offer document. It has not been endorsed by, or registered with, any governmental authority or regulator. The distribution and use of this paper, including any related advertisement or marketing material, and the eventual sale of tokens, may be restricted by law in certain jurisdictions and potential purchasers of tokens must inform themselves about those laws and observe any such restrictions. If you come into possession of this paper, you should seek advice on, and observe any such restrictions relevant to your jurisdic- tion, including without limitation the applicable restrictions set out in the Regulators' Statements on Initial Coin Offerings at the website of the Interna- tional Organization of Securities Commissions ("IOSCO") (https:// www.iosco. org/publications/?subsection=ico-statements). Restrictions are subject to rapid change. If you fail to comply with such restrictions, that failure may constitute a violation of applicable law. By accessing this paper, you agree to be bound by this requirement.

1.0 FANTOM

Blockchain technology has provided a way to maintain consensus across all nodes with no central authority. However the technology faces fundamental issues like a lack of real-time transaction settlement and scalability. Despite improved consensus algorithms, Some blockchain implementations such Bitcoin or Ethereum synchronize one block at a time. This results in slow confirmation times, one of the biggest factors stopping blockchain technology from being widely used across many industries. Although Smart Contract platforms such as Cardano and EOS have started to emerge, public Distributed Ledgers are still not widely used.

To address these persistent issues, a new model based on the Directed Acyclic Graph (DAG) was developed. FANTOM is a new DAG based Smart Contract platform that intends to solve the scalability issues of existing public distributed ledger technologies. The platform intends to distinguish itself from the traditional block ledger-based storage infrastructure by attempting to employ an improved version of existing DAG-based pro-tocols. The FANTOM platform adopts a new protocol known as the "Lachesis Protocol" to maintain consensus. This protocol is intended to be integrated into the Fantom OPERA Chain. The aim is to allow applications built on top of the FANTOM OPERA Chain to enjoy instant transactions and near zero transaction costs for all users.

The mission of FANTOM is to provide compatibility between all transaction bodies around the world, and create an ecosystem which allows real-time transactions and data sharing with low cost.

2.0 Why Are We Making FANTOM ?

The vision of FANTOM is to grant compatibility between all transaction bodies around the world using fast DAG technology that can be deployed at scale in the real world, and to create new infrastructure with high reliability that allows for real-time transactions and data sharing.

FANTOM has the intention of being used on a large scale in various industry verticals, such as telecommunication, finance, logistics, electric vehicle provision and others. The FANTOM Foundation intends to create the FANTOM platform along with a new Smart Contract-based ecosystem that can be used by all current and future partner companies around the world. To facilitate consistent global transactions with high accuracy and reliability, the FANTOM Foundation will lead the next generation of distributed ledger technologies.

The platform intends to be open-source: used and changed by the community, and to provide various application support tools that can be used to create decentralised applications (DApps).

2.1 Issues with blockchain

Blockchain is a software innovation for establishing digital trust between users, and facilitates the transfer of value from one entity to another over a network. Its aim is to enable the safe preservation and remittance of capital without the need for a traditional public institution or middleman. FANTOM believes that in order for blockchain technology to be used in real life with broad applicability, it needs to be easily transferable, irreversible, and the transaction fee must be small or zero. However, existing blockchain technologies have limitations due to slow confirmation times and potentially high transaction fees.

Issues of scalability

In existing blockchains, all nodes verify and store a single block at a time, leading to longer time in creating blocks and limitations in block size. Therefore, no matter how many nodes are connected, the performance will be limited by the speed of each node. The more transactions require processing, the worse the performance due to bottlenecks on the network itself. Thus, Fantom believes parallel approach is required.[1]

Fees

Various fees occur when exchanging value using blockchain. Major fees include transaction confirmation fees paid to block miners, and the block reward itself. These fees incentivise consensus participants, and secure the

[1] www.mdpi.com/2071-1050/9/12/2214/pdf

network against attacks such as DDOS and staking attacks.[2] However, Fantom believes that these fees are prohibitively expensive for a scalable and enduring blockchain with a thriving ecosystem of users and applications.

History data

Blockchain only uses information stored in blocks – it cannot obtain information from the outside world to verify prior transactions, and the information stored inside existing blockchains only can provide limited functionality.[3] In order for the blockchain technology to be embraced fully in the real world, Fantom believes that a function that manages historical information along with transactions inside the block is critical.

2.2 Solution offered by FANTOM

As a means to solve the problems of existing blockchain solutions, Fantom aims to develop a new implementation of DAG-based consensus, which intends to create a new platform that improves the scalability and versatility of existing DAGs. FANTOM's technology is intended to create potentially infinite scalability, and process hundreds of thousands of transactions per second even with large numbers of nodes participating in the network.

The FANTOM OPERA Chain is intended to solve the scalability limitations of existing blockchain with the Lachesis Protocol. This is intended to nbe achieved by adopting a method where a single event block verifies the previous transaction, and transactions are verified and processed asynchronously without being approved by the miners as in prior blockchains. Thus, increased transactional load will not lead to delayed approval or bottleneck effects. It intends to also manage historical information on its own without being assisted by external databases such as the Oracle Database. Event blocks that store information from transactions that arise include multiple data packages. A data package may include transactions, Smart Contracts, historical information, reputation management, and rewards.

The FANTOM OPERA Chain intends to make the processing infrastructure in our society more transparent and reliable. With fast and safe processing methods based on DAG and independent management of historical information through "Story Data", the Lachesis protocol is intended to be expanded into various industries along with Smart Contracts.

[2] https://bitcoin.org/bitcoin.pdf

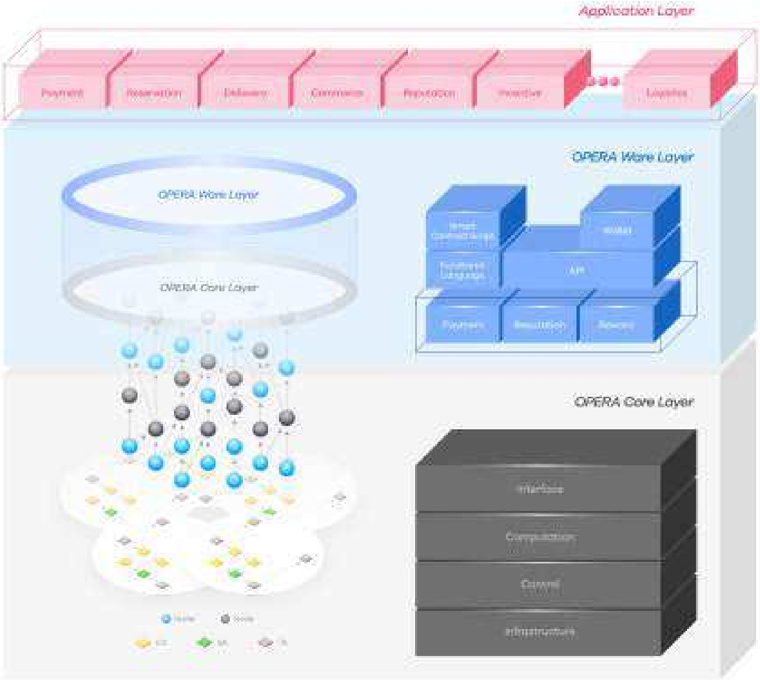
[3] https://eprint.iacr.org/2016/168.pdf

3.0 Technical Overview

3.1 Introduction

FANTOM's platform has a unique technology called OPERA. The OPERA Chain is a new type of distributed infrastructure that intends to solve the scalability issues of existing blockchains through the rapid processing of blocks on a large scale. OPERA Chain intends to process in real time not only transaction information but also Story data in a distributed environment. The "Story root" is stored in event data in order to record detailed historical information. The structure of the Story root is similar to that of a general transaction, but it has a more extended concept in including inheritance of certain properties. More explanation on how Story root operates will be presented in subsequent technical paper.

FANTOM OPERA Chain intends to use a high-level Scala-based functional programming language that compiles to smart contract bytecode on the FANTOM Network. FANTOM's OPERA Chain consists of three layers: The Core Layer processes transactions at scale, the OPERA Ware Layer which supports Smart Contracts and other functionality, and the OPERA application layer provides support for third party applications. The OPERA Core Layer is the layer whose objectives are to operate reliable transactions in FANTOM's ecosystem as well as core chain technologies for the exchange of information. OPERA's Core Layer is a chain technology that is intended to theoretically process up to 300,000 transactions per second.



<Figure 3.1> FANTOM Structure

3.2 OPERA Chain

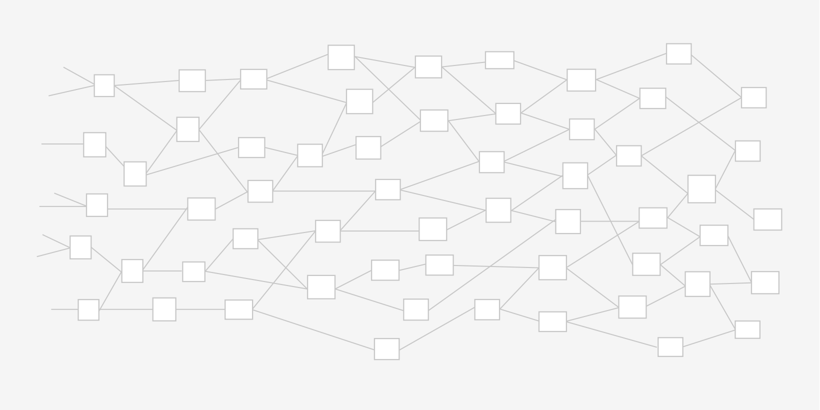
3.2.1 The Lachesis Consensus Algorithm

3.2.1.1 Introduction

The OPERA Chain will use a new consensus algorithm known as the Lachesis Consensus Algorithm (LCA), which is intended to significantly improve performance and security using DAG-based distributed ledger technologies. The LCA intends to be a Byzantine Fault Tolerant (BFT) technology that guarantees a similar level of consensus as existing blockchains. It intends to not only prevents attacks caused by a specific node failure, but also to process up to 300,000 transactions per second. OPERA Chain intends to employ cryptographic techniques to enhance security when communicating between nodes and uses a functional programming language for full smart contract support.

The LCA intends to form a "Lachesis DAG" based on the Lachesis Protocol. A set of links between event blocks form a DAG, which is a distributed system that stores arbitrary data that cannot be changed. Event blocks contain information such as transactions, smart contracts, Stories (historical information), and the values of previous events. An event is connected to the previous event block with central authority manipulating the structure. Event blocks from the previous rounds achieve more verifications as future events blocks are added. <Figure 3.2.1-1> shows the structure of a proposed Lachesis DAG.

The LCA intends to be fully asynchronous and when two identical transactions are requested (i.e. the double-spending issue), only the earliest transaction is validated. The order between transactions is arranged with aid from the Main Chain list.



<Figure 3.2.1-1> Lachesis DAG

3.2.1.2 Components

LCA operates on the Lachesis graph and it consists of Events, Clothos, Atropos, and the Main Chain.

Event Blocks

The Event Block data structure includes the following:

- **Stored Data**: An Event Block can contain multiple data packages. There are several types of data package, depending on functions such as transaction, smart contract, history information, reputation management, compensation etc.
- **Signature**: The signature of the user who created the Event Block is included and the user is identified through an account or address.
- **One or more hash values of the previous event block**: This is included to provide links between Event Blocks.

Like other Blockchain technologies, where the new event block verifies all previous event blocks (including the transactions inside them), all new Event Blocks will verify only their parent event blocks. A new event block will be connected to its parent event block through hash and all hashes will be derived from parent event blocks, so that it is impossible to modify or delete the previous event blocks. When an event block is connected, another node will build a new event block on top of that event block.

Flag Table

The Flag Table is a data structure that could save the connection data of specific event blocks (Clothos). The data structure includes the following:

- **Clotho Index:** Contains index information for each Clotho
- **Connectivity:** Contains information about the connection with other Clotho

Clotho

A Clotho is an event block that contains a Flag Table, and can see the supra-majority of blocks created in the path of previous event blocks. The event block that can connect a supra-majority among event blocks will be appointed as the Clotho and utilized for the appointment of Atropos and for the consensus of other event blocks.

Atropos

Atropos is a set of special event blocks. It is appointed based on the information- tion in Clothos and constitutes the Main Chain. It is also utilized in validation of event blocks in a specific stage.

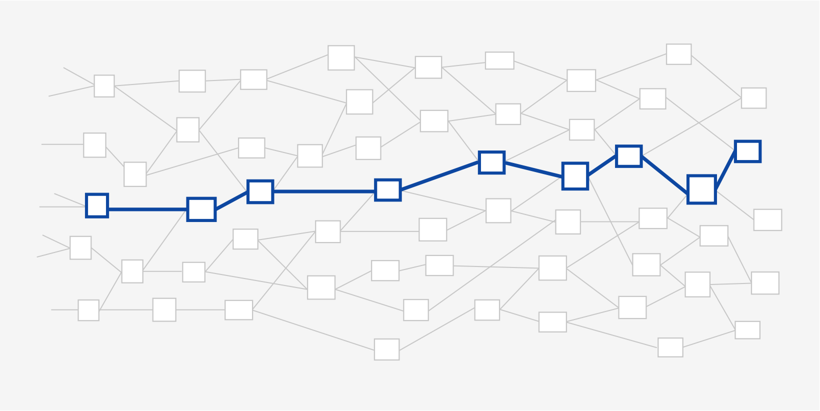
Main Chain

The Main Chain consists of Atropos and related event blocks. The Main Chain intends to be used for the validation of event blocks and to maintain the entire network structure.

3.2.1.3 How it works

The Lachesis technology intends to achieve high performance and secure data storage. All event blocks can be created asynchronously from nodes and each of these event blocks consists of a set of transactions (payment, remittance, smart contract, story, reputation, rewards). The new event block is connected to the parent event block, which is the most recent previous event block, and the node is intended to generate the block at a high speed through the Lachesis protocol.

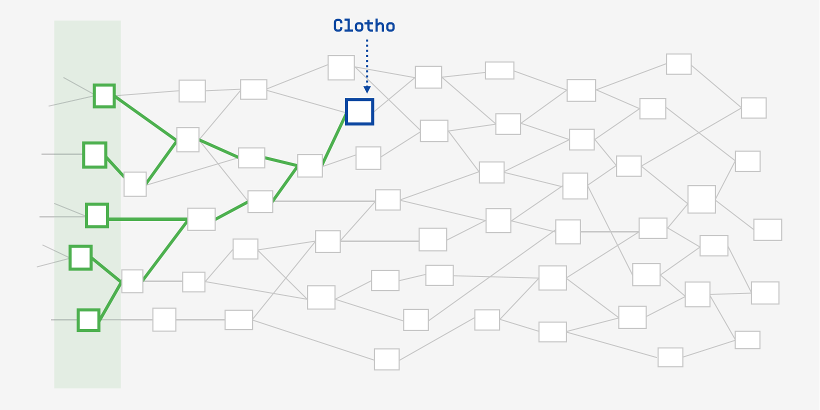
The Lachesis protocol can be represented by a graph where all event blocks are connected. There exists a chain that could be connected through set blocks and it is called the Main Chain. Figure 3.2.1-2 shows the assumption of existence of the Main Chain on any Lachesis DAG.



<Figure 3.2.1-2> Lachesis DAG Main Chain

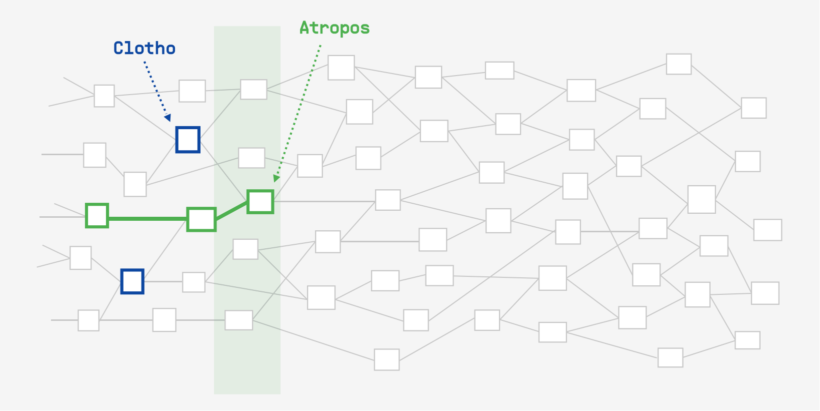
The Main Chain is a set of event blocks that can validate event blocks created over a period of time. The LCA can effectively solve various problems such as double-spending issues or malicious attacks by intentionally generating incorrect event blocks while maintaining the Main Chain. The Main Chain has an influence on the ordering between event blocks that occur asynchronously. The Main Chain helps event blocks that occurred earlier to have a priority in the sequence. At the heart of the Main Chain is Atropos.

<Figure 3.2.1-4> shows an example of the presence of a Clotho in a certain Lachesis DAG.



<Figure 3.2.1-3> The Clotho of Lachesis DAG

In the Lachesis DAG, there exists a Clotho which has a supra-majority (more than two thirds) connection with a set of event blocks. In a random distribution of event blocks the Clotho is located in a certain location at a minimum deviation from the supra-majority. Each Clotho has a Flag Table that stores its connection information with another Clotho set. By using the Flag Table which is the connecting information between Clothos, the Atropos is designated. During the process of designating an Atropos through the information of the Flag Table, consensus of the event block in between the Clotho set is met. Such consensus will implement aBFT.



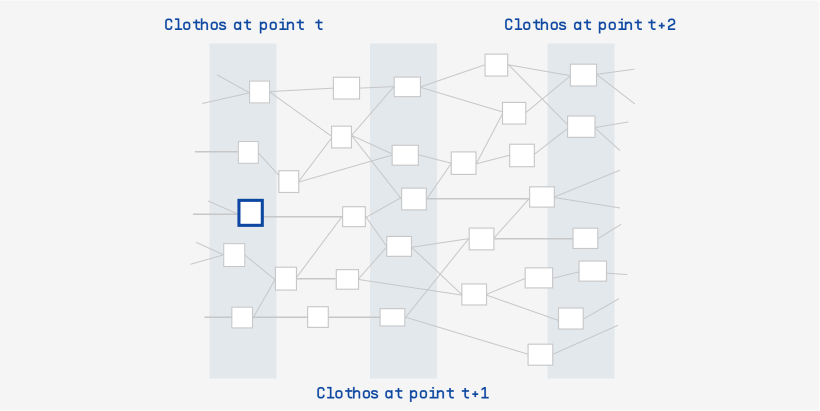
<Figure 3.2.1-4> The Atropos of Lachesis DAG

The Atropos event blocks are a subset of the Clotho event blocks that generate information for the connection of the existing Main Chain event blocks and it works to complete a new Main Chain. The Clotho works as a middle block for all event blocks and sits in an important location in the path. A single Atropos event block checks the validity of a certain round (all parent blocks in the round). The algorithm that designates the Atropos and Clothos is based on the event block itself and all its ancestor blocks. LCA completes the Main Chain asynchronously. The block that is subsidiary of the parent block cannot be linked to another parent block, meaning that generated blocks are never modified or deleted.

$$Clotho(i) = \sum_{j=0}^{n-1} \frac{(N-1)}{d_G(i,j)}, \ i \neq j$$

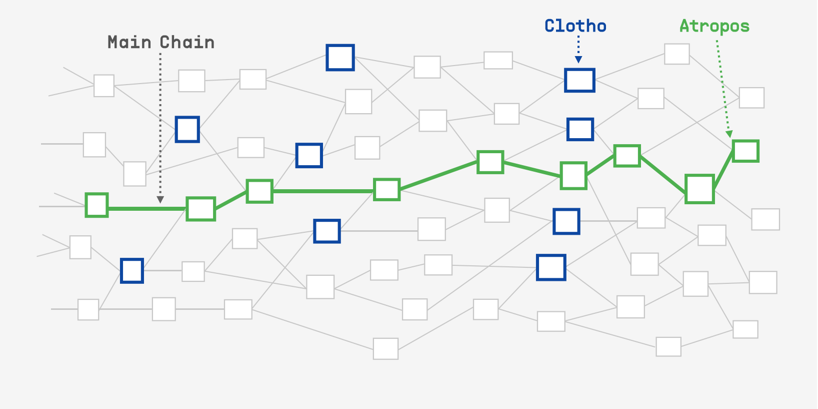
Clothos are a summed set of shortest paths from vertex i to n-1 at a specific moment.

A Clotho chooses the event block with the most centrality within the same set.



<Figure 3.2.1-5> Finding Atropos in Lachesis DAG

<Figure 3.2.1-5> is a schematic process diagram in selecting an Atropos event block in a certain Lachesis DAG. Assume that event blocks of 1, 2, 3, 4, and 5 are Clothos at point t, 13, 14, 15, 16, 17 being t+1 and 27, 28, 29, 30, 31 being t+2. All Clothos have a Flag Table and these tables have connecting information used for finding Atropos. The Lachesis DAG contains algorithms to find Atropos rapidly. In order to make this algorithm work, the Clothos need to find the Atropos and the Atropos needs to link the Main Chain.



<Figure 3.2.1-6> Main Chain and Atropos of Lachesis DAG

Once the Atropos is determined and the Main Chain is created, the order of unordered event blocks can be determined. Based on the event block connection information of the Atropos, the exact time for the consensus of each event block can be determined.

If two event blocks have the same consensus time, they will use the Atropos timestamp to determine the viewpoint. The point-in-time value of a particular event block determines the transaction order.

3.2.1.4 Procedure

```
Lachesis Consensus Algorithm
        loop
        parallel procedure 1
                  create a new block on each Node
        parallel procedure 2
                  Find_Atropos(all_block, atropos, clotho)
                  Main_Chain(MC, atropos, clotho)
        end loop
```

<Figure 3.2.1-7> Lachesis Consensus Algorithm

The main algorithm shown in <Figure 3.2.1-7> of the Lachesis Algorithm is intended to be very clear and simple. Each node can freely create new events asynchronously at the same time. This differs Lachesis from other blockchain technologies, in which the consensus requires every node to participate. The Lachesis algorithm is intended to allow each node to create blocks in parallel. Each node will make a new event block asynchronously by sending messages to each other. The information will be exchanged to make a new solid chain.

Such a simple messaging protocol is enough to implement aBFT. When each node sends and receives messages, it always keeps the order of event blocks by connecting new event blocks after the previous event blocks. While the nodes are quickly creating new event blocks, the Lachesis Algorithm uses two procedures to create the Main Chain and rapidly determine the order for event blocks. The two procedures do not require communication with each node.

```
procedure Find_Atropos (all_block, atropos, clotho)
        atropos[]
        heap clotho
        heap lookup
        for each clotho c
                traverse flagtable c
                if find atropos_path then
                        atropos[c]++
end procedure
```

<Figure 3.2.1-8> The Process of finding Atropos

<Figure 3.2.1-8> is the algorithm to find the Atropos event block to make the Main Chain. Finding the Atropos that will be included in the Main Chain is done through the 'Find_Atropos' procedure. By using the Clotho of a specific time, the Atropos event block can be found.

```
procedure Main_Chain (MC, atropos, clotho)
        heap MC
        heap clotho
        heap MC.last_block
        traverse lookup
        // optimistic MC_path between former and atropos
                if find MC_path then
                        MC append set of list
end procedure
```

<Figure 3.2.1-9> The Process of making the Main Chain

<Figure 3.2.1-9> describes the procedure for creating a Main Chain. The key factor of the Lachesis Algorithm is to maintain the Main Chain which is significant for keeping the block in order. Finding the Atropos and the Clotho in the Lachesis Algorithm is intended improve the speed of the Main Chain. The completed Main Chain should make OPERA Chain secure and play a key role in implementing aBFT.

3.2.1.5 Elliptic Curve Encryption Technology

One of the intended features FANTOM's technology is safety. FANTOM intends to use highly secure elliptic curve cryptosystem (ECC) technology to enhance security when transmitting data between signatures and nodes. Elliptic Curve Encryption has the advantage of ensuring high security with a short key size, and also allowing high-speed computations when signing. ECC makes it easy to design a secure cryptosystem by applying the most efficient algorithms to solve discrete logarithm problem. It is much more efficient since it provides equal security with shorter key lengths than other cryptographic systems, equivalent to RSA 1024-bit keys and ECC 160-bit keys. In addition, elliptic curve operation is easy to implement in hardware or software. By using ECC, FANTOM intends to add support for hardware wallets as well as software wallets, and enhance the security features for wallets. Fantom is also interested in using the secp256k1 parameter of the ECC.

3.2.1.6 Response to attacks

The Lachesis Protocol will likely be subject to attacks by malicious groups which aim to gain financial profit or to damage the system. Here we explain a few possible attack scenarios and how the Lachesis Protocol intends to take preventive measures.

Sybil attack

An attacker may make hundreds of OPERA chain nodes in a single computer. However, as the node operation method of OPERA chain will use a method similar to Delegated Proof of Stake (DPOS), the outcome through the voting system will be intended to accurately identify an incorrect node. An attacker should not obtain an additional vote to add a new node in the network. Also, since a single computer can only create a single node, a Sybil attack should not be possible in OPERA Chain.

Parasite chain attack

In a DAG-based protocol, a parasite chain can be made with a malicious pur- pose, attempting connection by making it look like a legitimate event block. When the Main Chain is created by the Atropos and the Clotho under the Lachesis protocol, verification for each event block is performed. In the verification process, any block that is not connected to the Main Chain is deemed to be invalid and is ignored, as in the case of double spending.

Transaction flooding

A malicious participant may run a large number of valid transactions from their account under their control with the purpose of overloading the network. In order to prevent such a case, OPERA chain intends to impose a minimal transaction fee. Since there is a transaction fee, the malicious user cannot continue to perform such attacks. Participants who participate in nodes are rewarded, and those who contribute to the ecosystem, such as by running transactions, are continuously rewarded. Such rewards are expected to be adequate in running transactions for appropriate purposes. However, since it would require tremendous cost to perform abnormal attacks, it would be difficult for a malicious attacker to create transaction flooding.

3.2.2 Functional Language

In order to attract vast numbers of developers, FANTOM intends to design a FANTOM virtual machine for writing contracts using existing languages. Scala, a well-known and supported functional programming language, is intended to be the initial language. The strengths of Scala are described below.

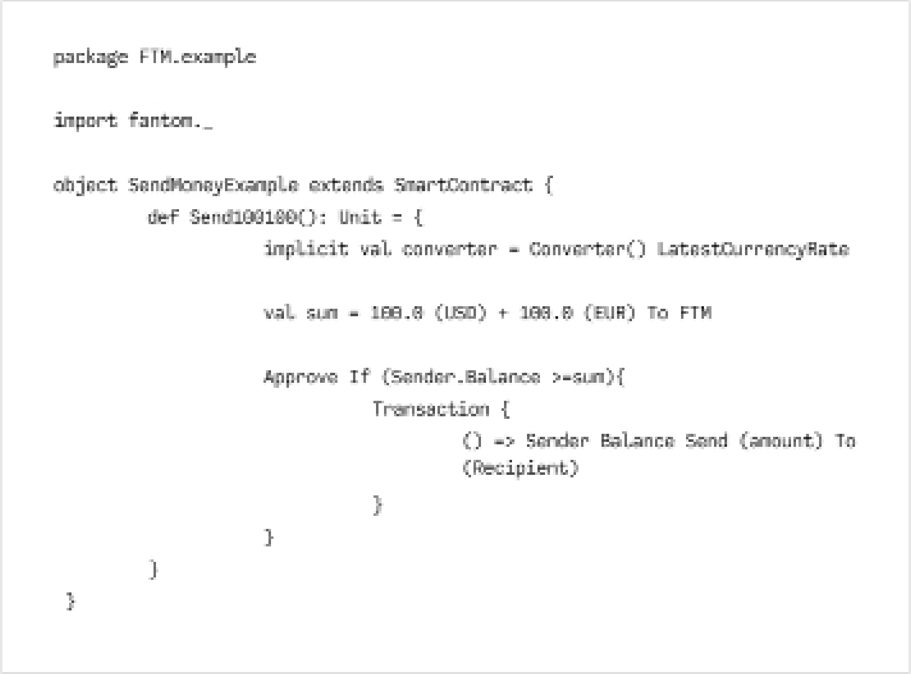
Scala, which was developed to remove the inconvenience of Java, allows developers to write organized and clean code. The strongly-typed functions of Scala can promote development and improve performance. For example, functions, macros, and tuples are only a part of the multiple advanced functions that are provided by Scala. Scala is a well-designed language which integrates functional and object-oriented programming. String pattern matching or Mixins, which include functions in class definitions, make coding enjoyable as well. A language with comprehensive documentation, FANTOM believes Scala is an excellent choice for developers with any level of experience. Closers and functions are also a part of the language. The biggest advantage of Scala is that it provides object-oriented and functional coding paradigms at the same time. Devel- opers can utilize the two methods with strength to easily write “concise” and “functional” programs.

Testing and development are convenient as well. Scala can perform the same work as Java using smaller coding lines. While Java also has several methods for reducing the code length, such methods are deviations from standard coding style, making the code less readable and leading to reduced productivity. Thanks to the properties of Scala, coding is reduced and testing and distribution become faster. Scala includes a non-expanding API library which is concise but possesses all necessary functions. Writing scalable software using Scala can make code writing, testing, debugging, and distribution easier. The language is versatile and can be used in all areas including desktop software, games, web application programs, mobile solutions, and software services. It is also a good fit for writing Smart Contracts.

Scala is a widely used functional programming language. A web App framework called Play is written in Scala, and it has successfully been established on numerous IT platforms such as Amazon and Coursera. The strengths of Scala have already been proven through practical applications in industry. Haskell, although an excellent functional programming language supported by mathematicians, does not have nearly as many users as Scala. Scala is comparatively easy to learn, and is a popular language with a large community of users. Development is facilitated since it also supports object-oriented programming. Also, it has all the strengths of a well-designed, functional programming language. By removing “Side Effects”, many coding errors as well as any changeable aspects can be identified beforehand, which also allows the easy transfer of code to a distributed environment.

Scala can introduce stringent coding techniques for compilation and formal verification. Formal verification is a mathematical methodology which is used to prove the accuracy of a computer program. This methodology has been used in protecting the software and hardware of military systems, transport infrastructure, encryption, and microprocessors. The value of formal verification in Smart Contract codes has recently been recognized as well, in particular on the Ethereum blockchain.4

As an example of Scala code, the following code can be written when transferring certain amount according to a given exchange rate.



```
package FTM.example

import fantom._

object SendMoneyExample extends SmartContract {
        def Send100100(): Unit = {
                  implicit val converter = Converter() LatestCurrencyRate

                  val sum = 100.0 (USD) + 100.0 (EUR) To FTM

                  Approve If (Sender.Balance >=sum){
                          Transaction {
                                 () => Sender Balance Send (amount) To
                                 (Recipient)
                          }
                  }
        }
 }
```

<Figure 3.2.2> Example of procedure using Scala FANTOM

4 http://antoine.delignat-lavaud.fr/doc/plas16.pdf

3.2.3 FANTOM virtual machine

3.2.3.1 Register Based VM

Virtual machines (VMs) used by existing cryptocurrency platforms are mostly stack-based, such as the Ethereum Virtual Machine (EVM) of Ethereum. Stack-based VMs can easily execute instructions using the stack data structure. However, as explained below, stack-based machines have longer code lengths and slower performance speeds in general compared to a register-based machines. As a solution to machine Storage in DAG event blocks are expensive. As code uses such storage, a large number of instructions are expensive. The FANTOM Virtual Machine (FVM) intends to extensively reduce capacity and increase processing speed. Publications5 indicate that register-based virtual machines can reduce OPCODE execution costs by over 50% and improve performance capacity by nearly double.

3.2.3.2 Stack-based model

The Stack is a basic data structure. A stack-based virtual machine uses the stack to perform operations. Assuming that we are performing a simple addition, four command lines are required to perform additional manipulation using PUSH and POP manipulation. The advantage of a stack-based model is that the operand is implicitly processed by the stack pointer: calling a stack pointer provides the next operand (POP), and there is no need to explicitly state the operand address. In a stack-based VM, all arithmetic and logical operations work by first popping and calculating the value pushed on the stack, and then by pushing the performance outcome to the stack, for example:

- **LOAD A**: Store Local Variable A to stack
- **LOAD B:** Store Local Variable B to stack
- **ADD**: Add the two values
- **STORE C**: Store operation result to Local Variable C

3.2.3.3 Register-based model

In storing the operand, a register-based virtual machine models CPU registers. While there is no PUSH or POP instruction, a command must include the name of the pointer, the operand for a command is explicitly stated. For example, when performing an addition on a register-based virtual system, the command may be expressed as follows. You can see that this code is shorter than the earlier stack-based version:

- **ADD AX, BX, CX** ; Adds AX with BX and stores to CX.

As previously mentioned, there are no POP or PUSH instructions, so there is only one line of code. However, unlike a stack, the addresses of operands,

5 Yunhe Shi, David Gregg, Andrew Beatty. Virtual Machine Showdown: Stack Versus Registers. 2005.

such AX, BX, and CX, must be explicitly stated. On the other hand, there is no overhead from pushes and pops using a stack, and accordingly, the command of register-based VM is faster.

Another strength of a register-based model is that it is possible to perform optimization which cannot be conducted in a stack-based approach. For example, if the same calculation is performed twice, the register model code can be optimized to make only a single calculation and stores the value to the register for reuse, increasing execution speed for the code.

The weakness of a register-based model is that it needs to explicitly state the operand address, increasing the average size of a command compared to a stack-based model. While a stack-based VM is very short, since it does not need to explicitly state the stack address, a register-based VM must include the location of the operand in OPCODE, increasing the size of individual commands. However, as was proven by Dalvik in comparison with the Java Virtual Machine (JVM), the size of the entire code base can be extensively reduced.6

3.2.3.4 Secure, Powerful VM with Turing-completeness

Since we cannot know what types of operations will be needed in the future, providing Turing completeness is critical for establishment of the DApp ecosystem. However, providing Turing completeness inevitably leads to the issue of decision impossibility. To address this, Ethereum introduced the concept of "GAS" in order to avoid the halting problem. However, the amount of gas consumption is hard-coded in the Ethereum code, and it is impossible to change this flexibly without a hard-fork. Also, although operations are critical since it is the executor who determines whether to perform a contract, inexpensive programs may or may not perform the operation. The DDOS attack on the EVM in September 2016 actually slowed down the network speed almost to a halt, as the attacker was able to attack the network by taking advantage of low gas prices.7 FANTOM intends to design the FVM with flex ible execution costs in mind, with the authority of the operation node being limited as well. Also, FANTOM believes that using the LCA means that there is no need to execute the same instruction set by all nodes. Even if an attack is possible, it should have limited impact on the network due to its flexibility.

The issues of security and feasibility are not limited to the EVM – they are shared by many distributed ledger projects . Some projects (such as Bitcoin) are mitigating such limitations by removing Turing completeness or, like Ethereum, by providing a large number of Smart Contract templates that enable formal verification. However, an absence of outcome functionality makes it difficult to implement a proper DApp.

The FVM looks to provide better security as well as Turing completeness. Furthermore, it looks to provide the core functionality for properly establishing a DApp ecosystem, such as External Code Linking, Library, and Import, as well as strong scalability that can be operated with a grid supercomputer. While a FANTOM-based Smart Contract can work in stand-alone, it can also

[6] David Ehringer. The Dalvik Virtuoal Machine Architecture. p5, 2010.

[7] https://blog.ethereum.org/2016/09/22/ethereum-network-currently-undergoing-dos-attack/

work jointly with other contracts, functioning as a component of the DApp Infrastructure.

3.2.4 Structure of OPERA Chain

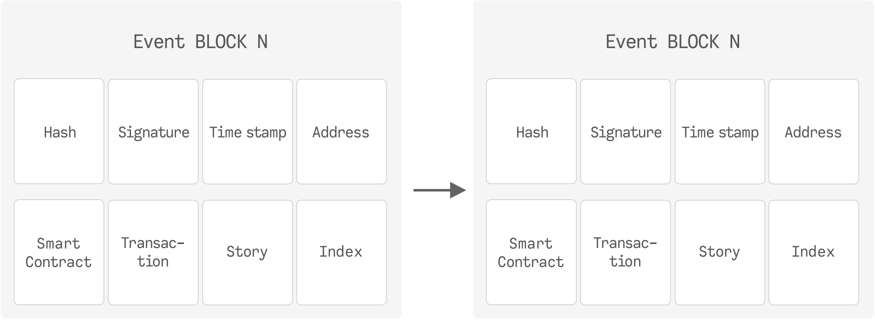
While similar to existing blockchain structures, the OPERA Chain structure of FANTOM has unique characteristics. The data structure, which is intended to include hash, signatures, time stamps, addresses, transactions, Smart Contracts, Story, and index information, is of a new type where the idea of 'Story' is added, as explained below.

In the event block data structure of OPERA chain, a hash refers to the value of the previous block.

A transaction is the value that is filled with each transaction in the list of transactions by blocks.

The Smart Contract contains the information of the Smart Contract created by the account.

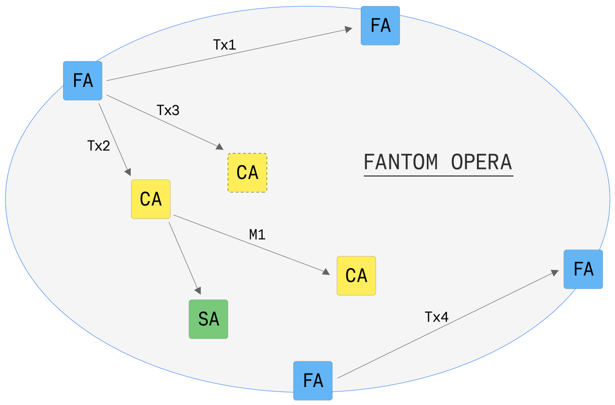
Story intends to record detailed information that is constantly changing, such as the information on distribution history. It refers to the transaction value filled with Story in transactions in the event block. While similar to transac- tions, the structure of Story has been expanded to possess information on inherited properties. Also, the algorithm for preventing any duplicate storage into event block is added.



<Figure 3.2.4-1> Data structure of event block

An OPERA chain consists of an externally owned account, which is controlled by a private key, and a contract account, which is controlled by control code. It also possesses a Story account, which is created by the contract code in order to manage Story information. An account possesses address and FANTOM tokens that are used by OPERA. Externally-owned accounts possess addresses that are controlled by a private key, and can make approved transactions for transaction-level tokens to other accounts. External accounts can create new contract accounts. Contract accounts are executed according to a pre-programmed order by sending a message once certain conditions are met. If needed, a contract account may call another contract account. However, a contract account cannot call another contract account without being called by an external account.

Smart Contract functionality in the OPERA chain should be designed to run automatically according to a pre-programmed pattern. It is run once certain conditions are met. The Story of an OPERA Chain is created once automatic execution is performed after meeting the conditions for the Smart Contract. Its role is to store transactions and certain Story data for the Smart Contract.



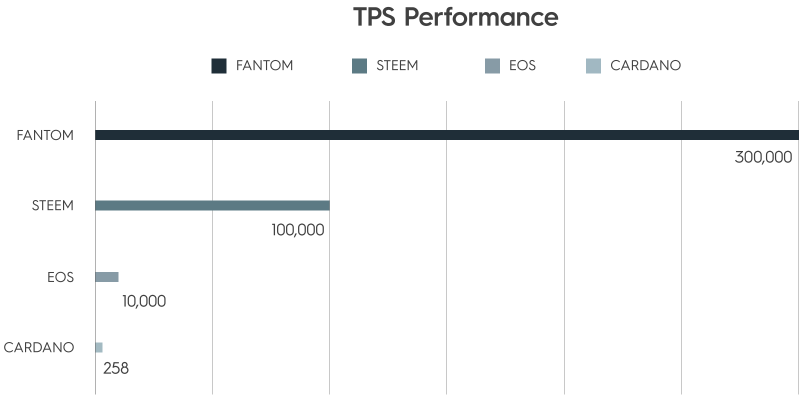
<Figure 3.2.4-2> Interactions of accounts within OPERA Chain

A distributed application is an application that operates in the distributed environment of an OPERA Chain. Using the resources of the FANTOM network, a distributed application can provide reliability and safety, and it provides functions for running contract code that is stored in FANTOM from the user's browser. By supporting a free web ecosystem, FANTOM can contribute to creating a new infrastructure, and can be effectively used in a hyper-connecting network. Also, it can trigger efficient changes in business process by working as a broker or central control authority, and can provide services that go beyond national borders.

The OPERA Chain structure stores data if a new block is created and also if the status of an account changes. To save each status value, a block head uses a hash tree structure based on a Merkle tree. In FANTOM, OPERA provides the environment for implementing a Smart Contract, conducts transactions on Story information, and saves a head note into the Story root
every time a block is created.

3.2.5 Performance of OPERA Chain

Using the unique Lachesis Protocol algorithm, OPERA Chain intends to solve the issue of scalability with the fast processing of blocks. While third-generation blockchain technology might show improved performance compared to previous impliementations of blockchain technology, the speed of creating blocks might be still very slow. OPERA Chain intends to ensure high creation and processing performance of up to 300,000 transactions per second. With a high level of reliability and scalability, FANTOM believes it is working on a strong third-generation blockchain technology which can be utilized on a large-scale across many domains and industries. OPERA chain intends to not only process large numbers of transactions at scale but also processes Story and historical data that can ensure the reliability of transactions.



<Figure 3.2.5-1> FANTOM TPS Performance

The OPERA Chain, which is based on the Lachesis Protocol algorithm of FANTOM, is intended to perform multiple verifications simultaneously, and conduct tests on the directions and validity of transactions at the same time. As each node processes all transactions that are broadcast to the FANTOM network, it should provide excellent transaction processing speed. In the past, all participants verified each block sequentially. However, the Lachesis Protocol algorithm will be designed to asynchronously verify and process event blocks in a distributed, concurrent method.

The size of each event block processed by the LCA is intended to be expanded up to 100KB, which Fantom believes will be sufficient due to faster block propagation. As an example, assuming that each transaction is 260 Bytes, a single event block can include up to 440 transactions. If the time it takes for each node to create an event block is 0.1 seconds, each node creates 7 to 10 event blocks per second. Assuming that the number of transactions requested is infinite and that 100 nodes are participating, each node would asynchronously and simultaneously create 7 to 10 event blocks per second. Every time the number of event blocks reaches 2/3 of the entire nodes participating, the Lachesis protocol adds and verifies another Main Chain. If 100 nodes are available, around 700~1000 event blocks should be created per second and are verified at the same time. Since each stage verifies and processes approximately 700 to 1000 event blocks, performance of over 300,000 TPS can be achieved. However, factors such as network latency could reduce tps.

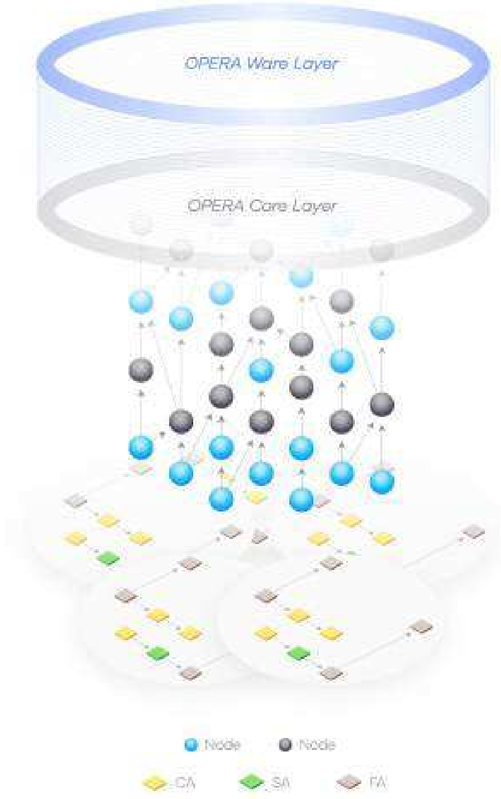
FANTOM believes the time complexity of the Lachesis algorithm means that a much faster performance speed can be achieved with O(N Log(N)).

The performance speed according to the time complexity O(N2) and O(N Log(N)) (n refers to number of nodes) is shown below.

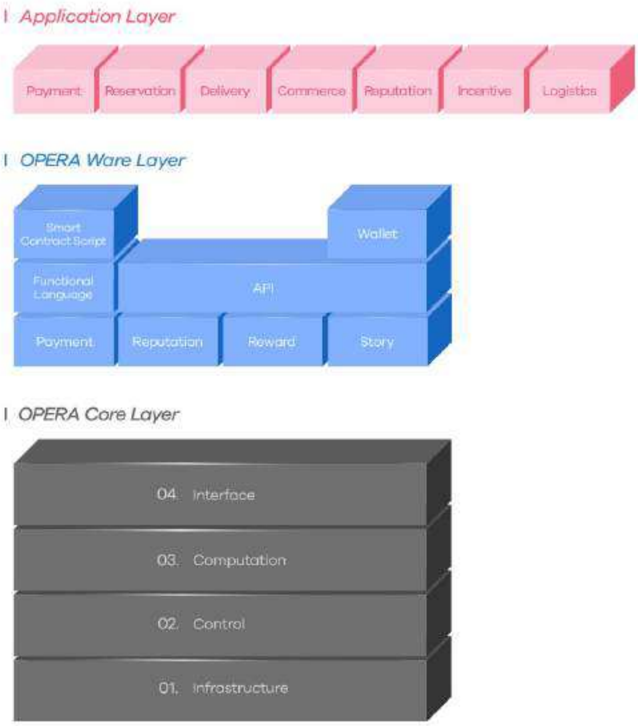
n square = n * n
n Log N = n * log(n)
n*n vs n * log(n)
n vs log(n)

If n=10, nlog(n) ~ 2.3
If n=100, nlog(n) ~ 4.6
If n=1,000, nlog(n) ~ 6.9
If n=10,000, nlog(n)   ~ 9.21
If n=100,000, nlog(n)   ~ 11.6
If n=1,000,000, nlog(n) ~ 13.8

4.0 Multiple Layers of OPERA



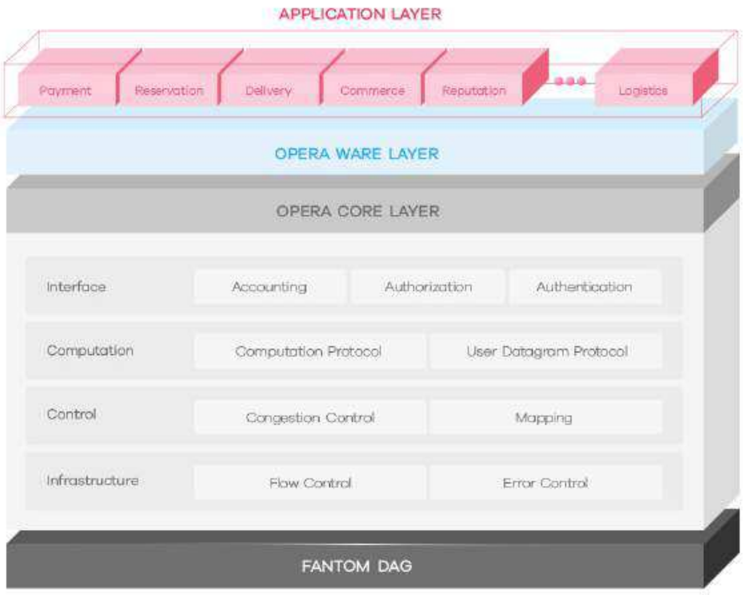
<Figure 4.0-1> Multiple layers of OPERA chain



<Figure 4.0-2> Multiple layer structure of OPERA chain

FANTOM's OPERA chain consists of a dual layer, and is composed of an OPERA Ware Layer which supports various applications and OPERA Core Layer which processes event blocks with Payment, Payment, Reservation, Delivery, Commerce, and Reputation data. Also, the OPERA core layer is the layer where core chain technologies for ensuring reliable transactions and exchange of information in FANTOM ecosystem operate.

4.1 OPERA Core layer



<Figure 4.1> Structure of FANTOM OPERA Core Layer

4.1.1 Infrastructure

OPERA Infrastructure is the core of the FANTOM ecosystem. FANTOM uses FANTOM OPERA Infrastructure (OPERA Chain™). While collecting and settling blockchain data in a distributed environment, OPERA chain should provide more functionality compared to other blockchains.

In the OPERA Infrastructure layer, information on transactions, Smart Contracts, and Story (historical information), etc. that arise from various applications in areas such as Payments, Reservations, Deliveries, Commerce, and Reputation, as well as the outcome of operating the data of the current and previous event block at the computation layer, including the data that indicates the value of the previous event block, are each mapped and stored in the Control Layer. A Main Chain of event blocks that occur in each applica tion is created, as well as the index of event blocks that are directly connect ed to the Main Chain, and the index of event blocks that are indirectly connected are stored.

The Infrastructure layer should ensure a smooth and reliable transfer of data between the entities above a physical network. It provides the functional or procedural means that are required to transfer event block data or to discov- er and fix errors. In order to verify that the data that identifies or transfers physical equipment is not falsified, it provides a function for fixing errors. It checks and modifies errors by grouping and transferring event block data into frames, and then re-calculating and comparing the checksum of each frame. Also, it performs a flow control task in order to prevent frame loss by verifying and regulating data transfer speed, as well as an error control task that processes errors by checking whether all frames have been delivered to their destination in the correct order. Furthermore, it manages possible collisions that may occur.

For certain platform services where an existing host runs the application, FANTOM should guarantees conversion to a commission-free network while maintaining the strengths of the FANTOM OPERA chain system.

4.1.2 Control

The control layer is the layer that controls the infrastructure layer. It maps the results processed in the computation layer to the infrastructure layer.

The control layer evaluates whether certain operations for a given transaction, Smart Contract, Story (historical information), reputation management, and reward have accurately been calculated and are reliable. It also records all details that are required to write a Smart Contract. Further, it verifies operations performed by the computation layer and physically maps each event block data to the infrastructure layer.

Other functions of the control layer include assigning and mediating the route for transfer of data between network systems, controlling flow, and controlling errors. It provides congestion control function for controlling congestion that arise upon data transfer, as well as the function for establishing, maintaining, and terminating network connections on data communication lines between systems to superior systems.

Also, event block data is divided into packets and are reunited after being transferred. In order to determine the optimal route for data communication, a routing algorithm is used to assign the logical address, and packets are delivered from the sender to the receiver.

4.1.3 Computation

The computation layer of FANTOM runs and operates transactions, Smart Contracts, and Story (historical) data from Applications. The computation layer analyses traffic received from the control layer and classifies the types of services. Using the TCP/UDP protocol, it communicates with the Interface Layer and the control layer to send and receive data such as transaction, Smart Contract, history data, reputation management data, and reward data in an event block.

The computation layer also checks errors to ensure accuracy in sending data, and arranges the order of event data. Providing multiple communications that distinguish trustable transfer and non-trustable transfer, it assures impartial operational processing, optimizes the overall network, and provides system scalability.

4.1.4 Interface

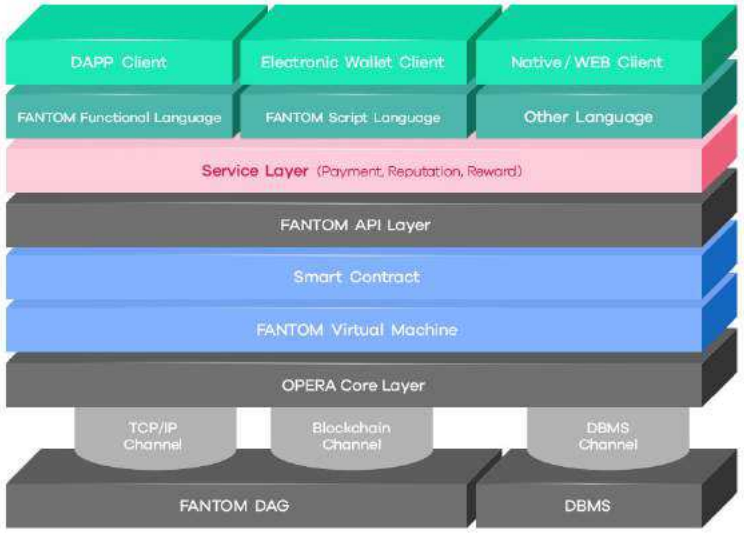
The Interface layer creates an environment where each Application accesses the FANTOM OPERA Chain. It manages and supervises accounts that participate in each network, and limits and certifies the authority of nodes. It verifies data such as FANTOM Wallet addresses and the signature of transactions in order to check whether a certain account has been properly authorized.

The OPERA Core and OPERA Ware layers in the FANTOM OPERA Chain communicate through interfaces such as the TCP/IP channel and the DBMS channel in the Linux Kernel environment, and sends and receives transaction data. The interface layer verifies and manages all the transactions for each application and maps and sends the verified data to the computation layer of the OPERA Core layer to perform the operations.

The Interface layer works on providing the control structure for each layer, and maintains and synchronizes the account settings. It manages and adjusts dialogues that are required to transfer data to the Ware layer that are being operated by different nodes. It provides the means to combine and synchronize each event block data in order and the means to establish, adjust, and terminate dialogue channels between the application layer units.

4.2 OPERA Ware layer

The OPERA Ware layer provides open source APIs, Smart Contract scripts, a functional programming language, Wallets, and Middleware (Payment, Reputation, Reward) that are to be provided to various dApps. In the OPERA Ware layer, the FANTOM token is a fundamental component of transacting. In addition to this core functionality of FANTOM tokens, OPERA Ware allows the payment of the appropriate reward associated with transactions. That payment is made using the FANTOM tokens, based on the reputation score and the transaction record for each participant (consumers, firms and producers).



<Figure 4.2> Platform architecture of FANTOM OPERA Ware

4.2.1 Middleware

The Middleware of FANTOM's OPERA chain collectively refers to protocols and APIs that link DApps. Consisting of APIs, the Scala functional language, Smart Contracts, e-Wallets, and protocols (Payment, Reputation, Reward and Story), it connects the Application and OPERA Core Layers to send and receive data.

<Figure 4.2> shows the architecture of the FANTOM Middleware platform. This architecture includes the FANTOM development environment and virtual machine. As can be seen in the architecture layers above, the FANTOM Middleware platform is made up of module layers of major services. This modular architecture allows the easy modification, expansion, and integration of services related to Smart Contracts as they are developed. Also, in order to integrate existing payment services, such as the PG service, for convenient utilization of FANTOM services, a Native/Web client SDK similar to existing types are provided in order to integrate into existing service environments.

As well as the Payment service, the FANTOM Middleware platform integrates Reward/Reputation services, providing various reward services for transaction clients using Application services that utilize the OPERA chain. Such reward services may create an environment that can provide additional integrated rewards services through an alliance with other services.

The FANTOM common framework layer manages accounts, certifications, and messaging that are required to access the platform using DApp, e-Wallet, and Native-Web clients. Account management has from the start been designed for multilingual support, ensuring easy utilization once the service expands abroad. Also, an encryption method is used for account information, including sensitive information, and adopts a strict security design so that even internal managers cannot access such information. The major flows of security services are presented in the figure below. The certification management is based on OAuth 2.0, being simple and intuitive, and allows the utilization of various encryption methods. OAuth 2.0 also supports the expansion into large-scale applications. Furthermore, messages can be processed using standard Push methods supported by IOS and Android devices with the regulatory environment in mind: message processing protocols that are legally required when providing payment services, such as text messages and emails, are available for use.

The client layer is composed of the DApp client, e-Wallet client, and Native-Web client. Regarding the DApp client, integrated development and service environment, such as the FVM and Development Language, will be introduced in the next stage of development. Also, an e-Wallet client and Native-Web client will be developed and provided at the current stage. Services provided in an e-Wallet will be designed for the exchange of FANTOM coins, and will be designed to be expandable for exchanges with other coins as necessary. Reward/Reputation services, the major characteristics of OPERA Ware, shouldl be provided as well. Finally, Native/Web client will provide payment, reward, and reputation services, which are provided by OPERA Ware service in existing service environments, in an integrated manner.

The development language layer, which consists of FANTOM's high level Functional Programming Language, FANTOM Script Language, and other languages, should allow the easy development and utilization of Smart Contract services by using the methods above. This layer is planned to be available in a subsequent development stage. Also, an integrated development environment for the development language will be provided.

The service layer consists of payment, reputation, and reward services. In the first development stage, the IOS or Android client SDK and the e-Wallet client, which are exposed externally, can be used by other services. The service layer should allow them to be expanded into Smart Contracts that are made possible by the FANTOM development language and FVM provided in the next stage.

The FANTOM API layer provides the base infrastructure where the service application developed in FANTOM service environment operates. This independent layer allows connection and expansion with other coins and facilitates expansion into other service areas. The FANTOM API layer includes modules that process Smart Contracts, blockchain, and transactions. The transaction processing offered by the FANTOM API layer is equipped with the ability to detect and forecast any suspicious recordings of FANTOM coin payments operated by domestic PG companies or card companies, allowing the systematic prevention of illegitimate transactions.

The FVM layer, which provides the common infrastructure to the FANTOM service environment and facilitates the development of various DApps, is a major component that exposes the FANTOM service ecosystem across various service environments.

The FANTOM communication layer is designed to enable communication with blockchain, as well as TCP/IP, synchronized/asynchronous messaging, and DBMS communication. Separating the abstract layer for communication from the physical layer, which makes direct communication with TCP/IP, synchronized/asynchronous messaging, and DBMS possible, allows developers to program with it without engaging in complicated communication coding.

4.2.2 Smart Contract production tools

The FANTOM OPERA chain provides a Smart Contract script editor. It facilitates writing Smart Contracts by allowing the entry of various transaction conditions that fit the characteristics of a DApp. The Smart Contract script of the OPERA chain processes transactions types that typically arise for each participant in industries such as communications, finance, logistics, and electric vehicle provision. Smart Contracts are coded in Scala and complied to bytecode through the FVM, and therefore have Turing completeness.

4.2.3 OPERA wallet

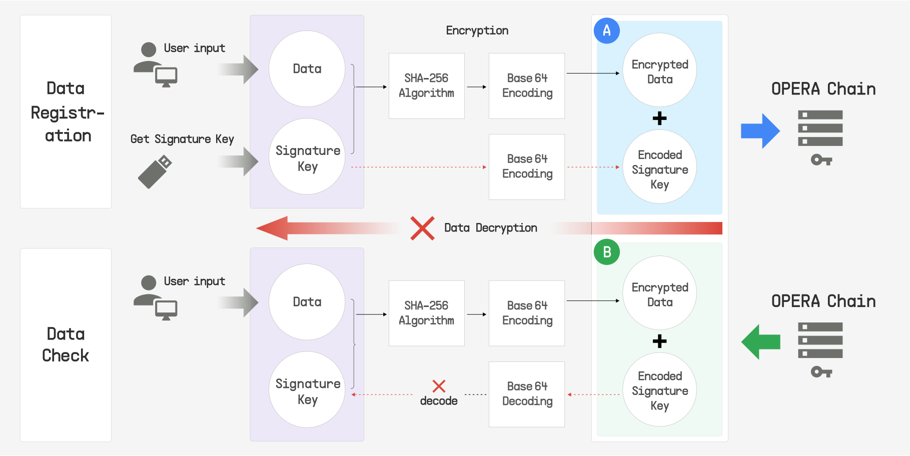
The e-Wallet and Native/Web clients used for FANTOM transactions are designed to provide distributed services under the wallet management, multi-address management (main account, withdrawal account, depos- it account), Address verification, encryption/decryption processing, and transaction currency exchange components. FANTOM uses a micro services architecture in preparation for the expansion of FANTOM coin exchange service fields. The details of each component are as follows.

The wallet management component provides services that easily recognize the e-Wallet address of service users through QR code scanning, and instantly enters into transactions once a quantity has been entered. Also, using IOS/ Android Native and Web methods, it should allow external businesses using the e-Wallet service to deploy DApps on various service environments and operating systems.

The multi-address management component manages e-Wallet addresses by dividing the accounts into the main, withdrawal, and deposit accounts. This prevents the exposure of the e-Wallet service during the use of services (e.g. upon payment, deposit, etc.), and minimizes damage arising from any possible exposure of e-Wallet addresses. For example, when a merchant is selling a product, only the address of the merchant's deposit account is provided to the customer. Even if the address of the e-Wallet using such a transaction service is exposed, it can minimize damage since the addresses only provides the function of receiving deposits.

The address verification component verifies the address of an e-Wallet by linking the information of the owners of public key, private key, and e-Wallet addresses. Such a process may make the falsification of e-Wallet address difficult, and allows the modification of verification logic according to changes in the service environment. Standard Encryption/Decryption methods provide the encryption and decryption services that are used on the e-Wallet address.

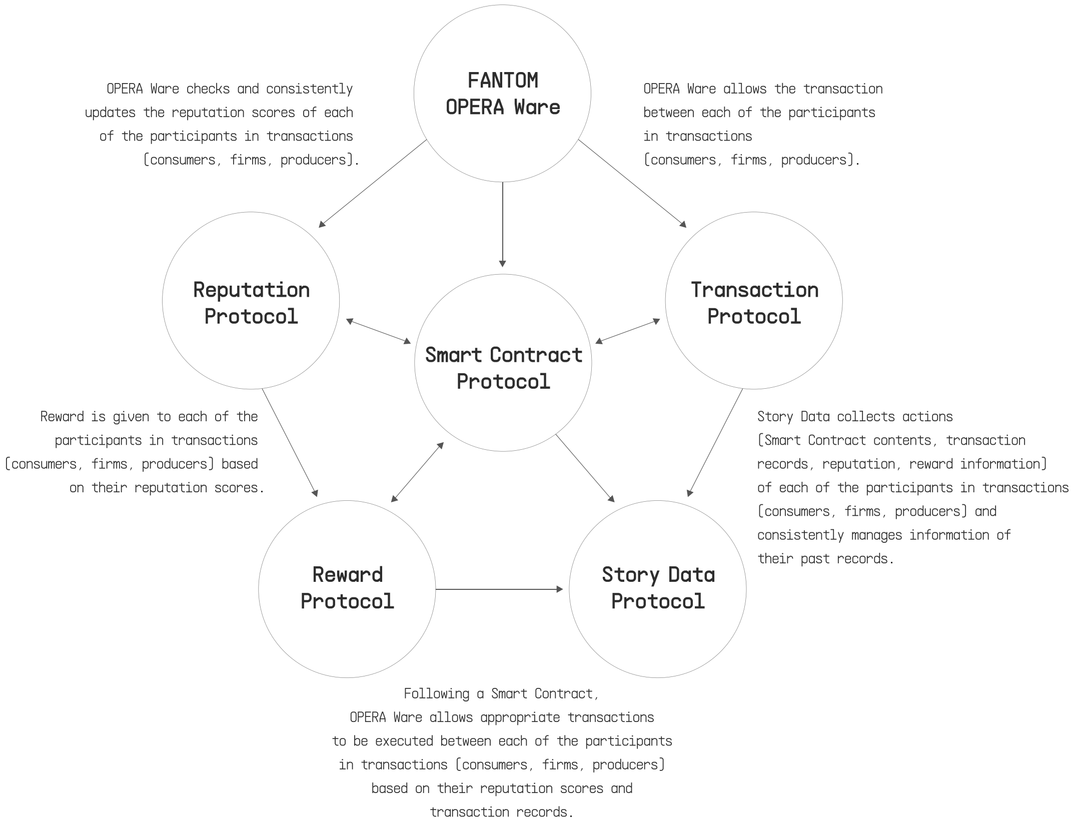
The transaction component allows users to transfer FANTOM tokens from one account to another, or other tokens that run on top of the FANTOM network.



<Figure 4.2.3> Operation platform of the FANTOM wallet

4.3 FANTOM OPERA Ware protocol

OPERA Ware uses the transactional, Smart Contract, Story, Reputation, and Reward protocols.


FANTOM OPERA Ware
Reputation Protocol
Transaction Protocol
Smart Contract Protocol
Reward Protocol
Story Data Protocol
OPERA Ware checks and consistently updates the reputation scores of each of the participants in transactions (consumers, firms, producers).
OPERA Ware allows the transaction between each of the participants in transactions (consumers, firms, producers).
Reward is given to each of the participants in transactions (consumers, firms, producers) based on their reputation scores.
Story Data collects actions (Smart Contract contents, transaction records, reputation, reward information) of each of the participants in transactions (consumers, firms, producers) and consistently manages information of their past records.
Following a Smart Contract, OPERA Ware allows appropriate transactions to be executed between each of the participants in transactions (consumers, firms, producers) based on their reputation scores and transaction records.

<Figure 4.3> FANTOM OPERA Ware protocol

4.3.1 Transaction protocol

Under the transaction protocol, participants of the OPERA ecosystem (businesses engaging in telecommunications, finance, logistics, electric vehicles, sales, restaurants, producers, etc.) provide appropriate goods or services to consumers.

For example, the transaction protocol of FANTOM's OPERA Chain can be used in the food delivery industry as outlined below.

Example Application: Food Delivery

An applications written by smart contracts running on the OPERA Chain would be intended to allow consumers to pay costs after ordering food online and receiving it (or at a predetermined time such as the time of ordering). A consumer orders food using a food delivery application host, and sends tokens. Once food is delivered, the transaction tokens paid by the consumer would be transferred to the restaurant according to the predefined Smart Contract.

If food is not delivered properly, consumers would demand the replacement of food or refund of costs. The smart contract(s) would be designed to allow the restaurant or the deliverer to make refunds to the consumer, which is performed by the OPERA payment protocol.

The restaurant or the deliverer would also approve a refund using the application host (according to the conditions of a Smart Contract). Once an approval is made, a refund is made according to the predefined Smart Contract, transferring the corresponding ratio of transaction tokens (as calculated by the infrastructure algorithm) to the consumer. Restaurants can also make payments to the application host for advertisements.

4.3.2 Smart Contract protocol

The Smart Contract protocol is a piece of code that facilitates, verifies, or executes contract requirements, online without a contract document or third party. A Smart Contract reproduces the logic of a contract's provisions.

A Smart Contract, running on a distributed ledger, allows for the exchange of money, assets, stock, or any other valuables in a transparent manner without the need for a broker or third party. A "traditional" contract requires third party intervention from brokers, government institutions, banks, attorneys, or notaries, as well as the required processing time, before receiving the goods and services. However, using Smart Contract technology, much of the above can be automated.

This Smart Contract technology could be compared to a vending machine. A vending machine is designed to run automatically according to pre-programmed rules, and its output is determined once certain conditions are met. When using a vending machine, a user deposits money in the machine and inputs a precise amount, after which the desired product can be collected. Under a Smart Contract, money will be escrowed and preserved in the OPERA chain, and once certain conditions are met, will be designed to be immediately transferred to the transaction counterpart.

The Smart Contract protocol of FANTOM's OPERA Chain processes transactions that arise between participants according to the conditions and requirements of each industry.

OPERA's STORY is created when certain conditions within a Smart Contract are met and the contract is fulfilled, and stores details in the Story data segment for transactions and Smart Contracts.

4.3.3 Story protocol

The STORY protocol is a protocol for recording and managing variable values, such as historical data from the production all the way to the distribution of each product. It uses the data captured in the STORY segment of the transaction block. While similar in structure to the transaction segment, the structure of a STORY segment has additional fields for more information and the ability to inherit properties from parent transactions. Also, an algorithm for preventing wasted space and redundant information has been applied. FANTOM's OPERA chain provides the environment for creating Smart Contracts, processes transactions with STORY data, and stores head node information to the STORY root every time a block is created.

4.3.4 Reputation protocol

The Reputation protocol is a protocol where all participants (producers, consumers, sellers, delivery agencies, deliverers, etc. - depending on industry use cases) participating in FANTOM's ecosystem can evaluate one another. Based on the data and responses from the participants, the reputation of each party will be determined; this is then fed into the criteria for appropriate rewards.

$$\sum_{i=1}^{n}(A_i\alpha_i) = \frac{A_1\alpha_1 + A_2\alpha_2 + A_3\alpha_3 + \cdots + A_n\alpha_n}{n} \quad (n \geq 1, \text{Integer})$$

<Equation 4.3.4> Equation for calculating reputation score

- A is defined as the evaluation criteria by the participants of each industry, where A is a real number between 0 and 100
- α is the weighted value of each evaluation criterion calculated by the value of A, and.
- n is the number of evaluation criteria

The sum of the product of each evaluation criterion and the weighted value, all divided by n to obtain the average value, determines the reputation score.

As shown above, the Reputation protocol of FANTOM's OPERA chain could consistently update the reputation score of each participant and utilize it by calling it whenever necessary. The reputation score is a real number between

0 and 100 (rounded to the tenth decimal point), and is applied to all parties participating in FANTOM's ecosystem.

There are 4 levels of reliability, graded from A to D, according to the reputation score. Once the evaluation criteria are defined, the evaluation score is updated based on the value calculated by using weighted values and importance. For example, an evaluation criterion that is deemed to be valuable would be calculated with a higher weight compared to the criterion whose value is deemed to be relatively lower.

4.3.5 Reward protocol

FANTOM's token economy will reward impartial evaluators with FANTOM tokens. All participants will be incentivised to engage in valuable behaviour on the platform, and a reliable ecosystem will be established. The Reward protocol will establish Smart Contracts on the Reputation protocol based on the number of transactions, and reward each transaction once the conditions are met.

For example, let's apply FANTOM's OPERA Chain Reward protocol in a food delivery use case. OPERA Chain will be designed to reward consumers, restaurants, and deliverers based on the strategy deployed by the vender and the incentive model employed.

If the conditions for the Smart Contract using the Reward protocol are met, the token will be provided to the recipient. Rewards may include experience points, or giving priority on search ranking. For example, transaction tokens or experience points would be designed to be rewarded to users in the FANTOM ecosystem who generate a high number of orders or reviews. For example, as in our first use case, the protocol would reward certain restaurants with high reputation scores by offering increased exposure and priority in search rankings. A restaurant could pay advertisement costs to the Application host using transaction tokens.

According to the Reward protocol, in this specific example it is intended that consumers would receive transaction tokens under a Smart reward contract if they write quality reviews or achieve a high number of orders or spending threshold.

5.0 Roadmap

5.1 Milestones

5.1.1 Intermezzo stage (~June 15TH ICO)

- Validate Lachesis protocol
- Validate FANTOM architecture
- FANTOM token on pre-sale
- FANTOM token crowd sale
- FANTOM Wallet development and distribution
- Open Middleware beta
- Commence OPERA Ware Layer Beta net

5.1.2 Seria stage (~3Q, 2018)

- Open main Middleware and release API
- Launch OPERA Ware Main net
- Open OPERA Core Layer beta
- Complete management of transaction processing
- Complete management of Story data
- Complete reputation management
- Complete incentive management
- Complete payment management
- Launch restaurant service Application

5.1.3 Buffa stage (~1Q, 2019)

- Begin main OPERA Core Layer
- Complete development Infrastructure Layer
- Complete development of Interface Layer
- Complete development Computation Layer
- Complete development of Control Layer
- Open Function language beta
- Open Virtual Machine beta
- Open Smart Contract production tool
- Manage Smart Contract
- Reinforce secure signature
- Reinforce client support
- Design strategies for launching Main net of Beta net
- Launch delivery service Application

5.1.4 Operetta stage (~3Q, 2019)

- Begin OPERA chain Main net
- Complete consensus model and fee model
- Complete composition of decentralized network
- Apply voting centre and voting function
- Open main Function language
- Open main Virtual Machine
- Distribute and interact with Smart Contract template
- Apply transaction processing main net
- Apply Story data main net
- Apply reputation main net
- Apply incentive main net
- Apply payment main net
- Expand into the Food Tech industry
- Expand into other industries
- Open POS application service
- Open commerce application service

5.1.6 Grand OPERA stage (~2Q, 2020)

- Support development of FANTOM open source
- Conduct global expansion
- Compose FANTOM technology development council
- Establish FANTOM research support agency
- Improve performance and reinforce security
- Enhance system model
- Reinforce reliability / assurance
- Open logistics application service
- Open financial service application

5.2 FANTOM (FTM) token distribution

There are 3.175 billion FANTOM (FTM) tokens. The FANTOM platform adopts an inflationary model to expand the ecosystem. FANTOM anticipates that there will be initially a 5% annual inflation rate that decreases as more users join the network. 20% of the total inflation is intended to be used to reward nodes and the rest is intended to be used to provide incentives for FANTOM platform users such as near zero transac- tion fees, and to reward users contributing to maintain a good flow of the ecosystem.

The FANTOM tokens will be distributed in the following manner:

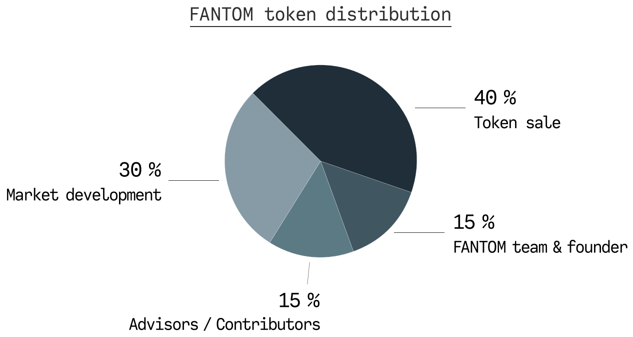
- 40% token sale
- 30% market development
- 15% advisors / contributors
- 15% FANTOM team and founders

The laws and regulations regarding tokens are evolving. Tokens will not be distributed to citizens, residents or individuals domiciled in countries or jurisdictions that from time to time restrict the purchase of tokens. These may include the People's Republic of China, the United States of America or other jurisdictions.

Similarly, the laws and regulations regarding network infrastructure operators are evolving. Licensing restrictions may apply from time to time in relation to the operation of the Fantom Foundation. Accordingly, these restrictions will be taken into account in determining the conditions and timing of any host token issuance events, for example.

Tokens for promoting the FANTOM ecosystem are intended to be used as incentives for new projects, marketing campaigns, recruitment of new em ployees, and growth for the next 5 years.

Tokens for advisors and contributors and tokens for FANTOM team and founders will be used for team members who plan the development of projects as well as partners who are involved in the FANTOM project. These FANTOM tokens are granted with 2-year disposal condition by stage.



<Figure 5.2> FANTOM TOKEN distribution

Fantom team & founder token allocation has 24 months vesting period, released monthly and Advisors / contributors token allocation has 3 months lockup.

5.3 Where FANTOM capital will be used

It is estimated that FANTOM's capital will be used in the following areas:

- Marketing expenses: 30%
- Operation expenses: 20%
- R&D / Development and application of ecosystem: 50%

Marketing expenses include the following.

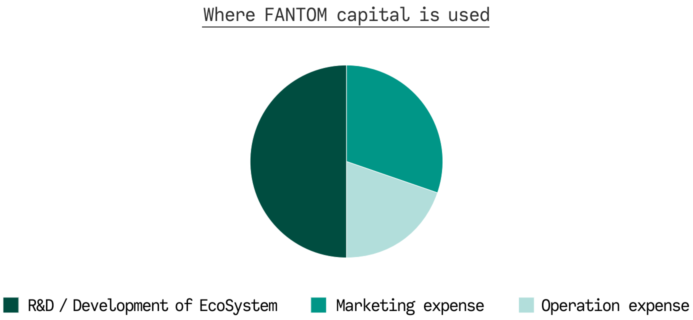
- Direct marketing for acquiring FANTOM services
- PR marketing for FANTOM
- Contents marketing of FANTOM through SNS, such as related articles and videos
- Events for growth of the community

Operation expenses include the following.

- Sales/Operation
- Legal
- Additional expansion overseas
- Prepare for contingencies

R&D / Development and application of ecosystem include the following.

- R&D for the OPERA platform
- Develop and expand OPERA Ware
- Develop and expand OPERA Wallet



<Figure 5.3> Where capital is used

6.0 TEAM & Partners

6.1 Board of Directors and team members

After going through months of research and planning, FANTOM Pty Ltd (FANTOM) was officially launched in January 2018 with the purpose of creating a global FANTOM ecosystem.

<Foundation Members>

Byung Ik Ahn
CEO
LinkedIn

Joseph Jang
COO / CSO
LinkedIn

Bob Tucker
Treasury
LinkedIn

Fred Pucci
Legal Counsel
LinkedIn

Jake Choi
CMO
LinkedIn

Michael Kong
CIO
LinkedIn

Seung Mun Lee
CFO
LinkedIn

Sung Yun
CBO
LinkedIn

Woong Jae Hyun
Eco Architect
LinkedIn

Yeweon Park
Marketing Director
LinkedIn

Gyumin Kim
Marketing Manager
LinkedIn

Yongwon Seo
Marketing Manager
LinkedIn

Sin Chul Kim
Business Development
LinkedIn

Jung Taek Lee
Business Development
LinkedIn

<Platform Development>

Yo Seob Han
Ph.D
LinkedIn
GitHub

Sang Min Choi
Ph.D
LinkedIn
GitHub

Ji Ho Park
Ph.D
LinkedIn
GitHub

Ki Young Jang
Ph.D
LinkedIn
GitHub

Hyun Joon Cheon
Ph.D Candidate
LinkedIn

Choi Hae Lim
Development
LinkedIn

Park Gyo Seok
Development
LinkedIn

Park Jong Chul
Development
LinkedIn

Kim Kang Ho
Development
Linkedin

Ye Jung Hwa
Development
Linkedin

Jo Chan Yeul
Development
Linkedin

Yang Sang Ha
Development
Linkedin

Lee Eun Jeong
Ph.D Candidate
Linkedin

Kim Sung Hwan
Development
Linkedin

Choi Im Chung
Development
Linkedin

6.2 Advisory group

<Advisory>

Steve Bellotti
Executive Advisor
Linkedin

Matthew Hur
Executive Advisor
Linkedin

Ashton Hettiarachi
Technical Advisor
Linkedin

Quan Nguyen
Technical Advisor
Linkedin

Alex Kampa
Technical Advisor
Linkedin

Eddy Travia
Advisor
Linkedin

Ran Neu Ner
Advisor
Linkedin

Francisco Jo
Advisor
Linkedin

Issac Lee
Advisor
Linkedin

Hak Kyoon Kim
Advisor
Linkedin

Min Sik Jo
Advisor
Linkedin

Hyeong Joo Kim
Advisor
Linkedin

6.3 Partner companies

Korea FoodTech Association

The Food Technology (FoodTech) industry, a new industry where traditional food industry is combined with Information & Communication Technology to make changes in all value chains including production, processing, distribution, and services, is a major area of the Fourth Industrial Revolution. Korea foresees creation of 300,000 new jobs through the FoodTech industry over the next 10 years in areas such as delivery, Smart farms, data, food safety, and education.

Korea FoodTech Industry is an Association for FoodTech startups and experts jointly developed by major FoodTech CEOs and experts, executive workers, the government, academics, and Venture Capitalists in order to promote information sharing and human interactions in the FoodTech industry, which will grow as a market valued at KRW 200 trillion. It consists of around 80 companies including businesses engaging in FoodTech platform, information services, delivery services, food-related infrastructure, online food ingredient distribution, contents, and shared franchise forums.

Oracle Corporation

Oracle Corporation, based in California, USA, is the second largest software company in terms of sales. It is a general IT company that develops and operates hardware and software for corporates, and possesses around 430,000 client companies around 175 countries. Its major products include Oracle DBMS, a database product, which brags the highest market share in the world.

Quantum Equity Partners

Quantum Equity Partners is a relatively new venture company investing in private equity. Hak-gyun Kim, who worked as the head of Venture Investment Division at Hanwha Corporation as well as Central Investment Partners is leading the company as the representative partner. Core operative personnel include Dong-woo Lee, Partner, from Neoplux and Tae-suk Choi, Partner, from Aju IB Investment. Quantum Equity Partners identifies, invests, and grows together with good businesses.

ORACLE
SB CK
SoftBank Group
coinsilium
FoodTech Association
한국푸드테크협회
SikSin
Blockchain Partners
Quantum
EQUITY PARTNERS
Coinhills
KBIPA
한국블록체인산업진흥협회
SFA
SINGAPORE FINTECH ASSOCIATION

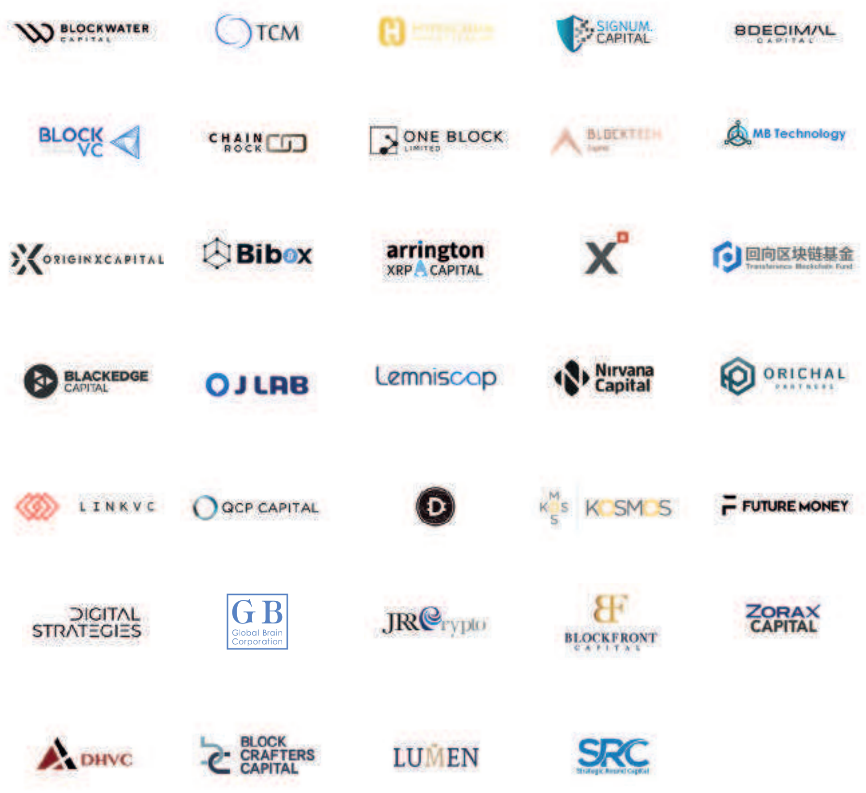
6.4 Investor companies

Blockwater Capital

Blockwater Capital is a cryptocurrency fund based in Korea that launched early 2018. Francisco Jo, Founder of Coinhills, and Issac Lee work as its GP, and Erica Kang, the CEO of KryptoSeoul and a popular influencer, and Wanlin Wang, the cofounder of Bibox, a global exchange, are working as its Partners.

TCM

TCM specializes in asset management and blockchain advisory businesses.



6.5 Legal Advisors

KING&WOOD MALLESONS

King & Wood Mallesons (KWM) is a multinational law office based in Hong Kong. KWM is the first and only global legal firm that is based in Asia, and is the largest legal firm that has its head office in regions other than USA or Europe. In 2015, it ranked in top 30 in terms of number of attorneys and sales, and is the sixth largest law firm in the world.

KING&WOOD
MALLESONS