

Unresolved directive in fermat-book.asciidoc - include::book-intro.asciidoc[]

Chapter 1. Chapter 1: Fermat Master Plan

As a Master Plan, Fermat has its own vision of how a collaborative project should be carried out. The main concept is to define a large master plan far into the future, and start building it phase by phase adjusting the target when needed. It is designed in this way to absorb the efforts of thousands of contributors over the years into a unified architectural vision.

After the pioneer step of Nakamoto who gave rise to **Bitcoin** and its revolutionary blockchain technology, in the last years, there has been a lot of fin tech investment money given to a myriad of startup initiating in this industry. This fostered the next evolutionary step, "**cryptocurrency 2.0**" but it has still not reached the level of adoption expected. There seems to exist an unbalance between the considerable investment of funds and effort needed to master this technology and the results they are bringing up. What is more, isolated projects trying to build something useful starting from zero point have it difficult to obtain the necessary support and the persistent endurance needed to succeed...

Precisely acknowledging this weak points in the industry, and choosing **collaboration** within a **master plan** as starting point, it is how the Fermat Community believes to succeed in solving the challenges set in their mission and purpose by building a complete new framework that solves the problem in an efficient, inclusive and attractive way.

1.1. Phase I: Foundations

The goal is to have a framework for developing financial applications in a trust-less environment. What are the services that had to be granted at the lower levels to enable this?

The basic infrastructure

the *Fermat Core Platform* is the part of the system which initializes the entire set of components, starting every plugin and putting them to run. This creates the interconnection that let all the services to be offered as being integral part of one unified system.

The basic services

- connection to the different Operating Systems that will run Fermat on a chosen device.
- interconnectivity to cryptonetworks and blockchain services providers to enable the **transport of value**.
- P2P communication of the devices running Fermat to enable the **transport of information (meta-data)**

The plug-ins environment

Fermat Plug-ins Platform deals with this task allowing systems users to be identified, low level plugins to be recognized as part of the system, communications services to link plug-ins living in different devices and in a general way,...an so on. In general terms, all these underlying construct allow various *network services* to run and start building the set of functionalities that will be taken by the applications built on top.

NOTE

At this stage of development, Fermat already offers a wide scope of REUSABLE components for third parts to build on top, encouraging further development of the plan and enrichening it with every single addition.

1.2. Phase II: First constructs

The foundations have been layed out, but to reach the end user oriented applications, it is necessary to create several functionalities. In order to make a visible structure available for further developments and also to provide certain focus to related functionalities, these are wrapped in substructures of Fermat system called **platforms** and **superlayers**. Each of them groups a subset of functionalities that offer the following services :

- a decentralized Wallet Production Line
- a Crypto Currency platform to manage the transference of value trough crypto transactions
- a Shoping platform to vinculate shop owners with brands and their customers
- a Digital Asset platform to allow issuing a vast type of digital coupons exchangeable for services and products
- a Marketing platform based on Digital Assets
- a Crypto Broker platform to easily administrate the business of exchanging crypto and fiat currency
- a Crypto Distribution Network to create local commercial environments fostered by crypto money

among others...

1.3. Phase III and beyond:

Have you grasped the *potential* of Fermat ? Can you imagine the next phases of development ? Are you interested in getting to know Phase I and II deeper?

Follow next chapters to get a more detailed description of each one.

Continue Reading ...

[Next Chapter](#)

[Previous Chapter](#)

Chapter 2: Fermat as a Platform Stack Framework

Fermat is, technically speaking, an Open Source P2P (Peer-to-Peer) Platform Stack Framework built on a Plug-ins Architecture running on end-user's devices.

Open Source

the code can be reviewed and audited by anyone.

P2P network and decentralized

censorship resistant and difficult to be attacked.

Platform Stack

functionalities are wrapped in categories easy to grasp

Plug-in Architecture

reuse of components and open to the participation of any developer.

It runs on end users' devices

users control on their money without the need to trust a third party.

What is Fermat made of ?

- **libraries:** structural components of the core of the system (low level layers)
- **desktops:** applications that run on an specific Operating System and present the GUI (graphic user interface)
- **plug-ins:** components which encapsulates a very specific sets of functions, consum services from other components and offer a public interface for the services they provide to other components. They are open to be developed by the general developers community.
- **add-ons:** plug-ins in the low level layers that provide the most sensitive functionalities to the system, and therefore are developed by trusted developers.

Fermat's architecture

Fermat as a system is built on several overlapping architectural paradigms:

1. At a higher level, it is a **Peer to Peer system with asymmetric nodes**, meaning that each node even having all the code base, it specializes itself according to the profile of the end user or the way it was configured.
2. Inside each node Fermat features a **Multi-OS architecture** meaning that the lowest level OS-dependent components are wrapped in a way that can be easily replaced when running on a

different OS without affecting the rest of the components consuming services from them. The upmost components, the ones facing the end user (GUI) are also OS-dependent. Everything in between is **not**.

3. At the same time it features a plugins architecture inserted into a **multi-layered structure**. These layers are occasionally grouped into **Superlayers**.
4. These **plug-ins** are subdivided into a hierarchy of **Platforms**. Those **Platforms** shares a common set of layers, and each one adds **Actors** and **Products** to the overall functionality of the Fermat system. Inside each Plugin, you might find a specialized structure and in many cases a database or files belonging to that plugin.
5. At a system level, Fermat uses speciallized crawlers to collect or sometimes inject information into nodes in order to recreate a system level conciousness prepeared to resist attacks or other relevant issues.

p2p architecture

Fermat is a *distributed system*, it runs on end user's devices, so it needs to stablish a connection between every actor and every device using it. This is done by creating a **P2P network** between them, based on the services provided by *Fermat P2P Network and Communication Superlayer*.

multi OS architecture

Fermat is **code-base portable**, meaning that the components are written in code that runs in different operating systems. So, through *Fermat Operating Systems API Superlayer*, the components that are specific to a given Operating System connect to the rest of Fermat that is OS independent.

multilayer architecture

Fermat needs to be adaptable, to provide highly component reutilisation, and to offer a scalability that match the complexity of the services that are running on it. Therefore the components are set in **layers** stacking one over the other. Each layer groups components that provide a similar functionality, eg. communications layer is designed to provide a way to connect one device to another building a "communications channel" using different available tecnologies (cloud servers, wifi, NFT, etc), and in performing this task, it let other components from other layers do their specific task consuming services of this communication layer in a transparent way (without any concern on how the communication is actually being stablished).

There are certain **Layers** that provide services at a system-wide scale (see **superlayers**), but other layers are defined within one **platform** and serve to provide its set of specific functionalities.

plug-ins architecture

Plug-ins have an outstanding feature: as long as their INTERFACE (i.e. the shape of the interconnection) is known and public, the plug-in can interact with others, consuming and offering services, INDEPENDENTLY of its inner structure and how it is built. As long as they conserve this INTERFACE, they can safely evolve to more efficient functioning WITHOUT altering a single service

that has been built upon it ! Inside Fermat, each plug-in is given certain specific responsibility within the whole, it lives in a certain layer, and it is allowed to consume services of components on *lower layers* and to provide services to components of *upper layers*. Plug-ins participate in high level processes and they are programmed in a way to live in an uncontrolled environment (end user devices) and to co-exist with untrusted third party plug-ins as well. Through this, Fermat is able to connect and reuse most of the infrastructure deployed by the industry.

platform architecture

A **platform** consists of a group of components living in different **layers** interconnected to offer a specific set of services for a discovered niche. For example, if we address to cryptocurrency users, we will need a *cryptocurrency wallet* for each cryptocurrency available running on *Fermat Crypto Currency Platform*. This wallet would operate on the selected cryptonetwork by means of a connection to it provided by the *Block Chain Platform*, and will interact with the user by means of an **desktop** living in the OS specific layers.

Following the Master Plan

Once we have described Fermat's components and the architecture, we will explore in the next chapters the platforms that implement the foundations (Phase I) of the master plan:

- **Fermat Core Platform**
- **Fermat Operating Systems Superlayer**
- **Fermat BlockChain Superlayer**
- **Fermat P2P Network and Communication Superlayer**
- **Fermat Plug-ins Platform**

For a cool visualization of Fermat and its constant growing number of platforms and superlayers visit <http://fermat.org> We will cover the platforms and superlayers from bottom to top and from left to right, as they are presented in the visualization.

So, let us enter the "building site" and explore the foundations.

Continue Reading

[Next Chapter](#)

[Previous Chapter](#)

Chapter 3: Fermat Core Platform

This is the **core platform** of Fermat Multi-platform System. It contains the deepest levels of layers in the layers stack. Its components encapsulate the basic definitions and functionalities that let Fermat operate as an *integrated system*.

The platform first defines a *plugin identity* for each plugin (this is done only ONCE for the whole system), which let the plugin be **recognized** as member of Fermat and then access to certain data specifically owned by it (like accessing to the file system or data base systems, etc). Then it put them to run (see [system initialization]) and builds the pattern of relationships between the isolated plug-ins, thus deploying all the system wide infrastructure.

Fermat Core Platform is written in JAVA as the most of Fermat, and that there is a JAVA implementation for each known operating system, that makes Fermat **code base portable**.

After initialization, this platform monitors the way the system is behaving as a whole.

Platform components

fermat core

Initializes all system-wide plug-ins and start a specific **core component** for each platform/superlayer running on Fermat, which in turn initializes the platform/superlayer's specific set of plugins.

android core

Initializes android's components that connect Fermat to the android user's environment.

platform specific core

There's a **core component** for each one of the platform existent in Fermat, which creates the *plugin identity* and initializes all platform/superlayer's specific plug-ins:

Continue Reading

[Next Chapter](#)

[Previous Chapter](#)

Unresolved directive in fermat-book.asciidoc - include::fbk-03b.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-04.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-05.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-06.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-07.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-08.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-09.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-10.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-11.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-12.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-13.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-14.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-15.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-16.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-17.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-18.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-19.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-20.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::fbk-21.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::conventions.asciidoc[]

Unresolved directive in fermat-book.asciidoc - include::glossary.asciidoc[]

Index

Appendix 01 - Fermat's Principles in Detail

Here is an in depth analysis of Fermat's principles.

1. Fermat must not allow censorship.

- Open Source.
- P2P architecture.

2. Fermat must not allow spying on their user base.

- Point to point encrypted communications.
- Geofenced P2P network enabling the shortest communication path between peers.
- Multi-technology communication layer (cloud, p2p, wifi, bluetooth, nfc, mesh, etc.).

3. Fermat must be secure and resistant to all kind of attacks.

- No centralized point to be attacked.
- Multiple identities and roles per user.
- Device Private Network (DPN) concept.
- Pocket Money concept and a Multi-Signature Vault owned by the DPN.
- Reviews and Reputation concepts.

4. Fermat must never loose a user's funds or assets.

- Transaction responsibility transmission protocols.
- Remote technical support by wallet and plugins developers.

5. Fermat must be usefull to each segment of the world population.

- Niche wallets concept.
- Branded wallets concept.

6. Fermat must be extensible and open to innovation with a master plan approach.

- Plugins architecture.
- Multi dimensional layer architecture.

7. Fermat must be open to any developer to participate.

- Any developer could contribute to the shared infrastructure.
- Any developer could write code for new plugins.
- Any developer could write code for new niche wallets.
- Any developer could fork an existing niche wallet to further specialize it.
- Any graphic / UX designer could provide skins to existing wallets.
- Anyone could provide language packs to existing wallets.

8. Fermat must compensate each developer by their contribution.

- Micro license concept.
- Micro license structure concept.
- FERMAT automatically enforcing license agreements.
- Plugin ownership concept.
- Niche wallet ownership concept.
- Branded wallet onwnership concept.
- Master plan design comcept, wiht no overlaping functionality but shared ownership.

9. Fermat's user base must be a shared asset.

- Any developer should be able to profit from the growing FERMAT user base by writing Plugins this user base could consume via the developers of the wallets they are using.

10. Fermat must be inclusive with crypto currency industry members.

- Any bitcoin | crypto | blockchain established startup could write their own plugins and expose their services to both FERMAT developers and the FERMAT user base.

- Any alt-coin community could write plugins to enable their crypto currency within the FERMAT wallet ecosystem.

11. Fermat must be inclusive with the legacy financial industry members.

- Banks and financial institutions could write plugins to enable FERMAT wallets hold funds in banks and other financial institutions.
- FERMAT allows it users to move money both through public value transport networks as crypto currency networks and also ththrough private or legacy financial networks.

12. Fermat must be un-banked-people friendly.

- Multiple logins per device (Device user concept)
- Crypto banking concept.

13. Fermat must be OS agnostic.

- OSA layer concept (Operative System API).
- OS dependent UI on top of FERMAT concept.

14. Fermat must learn from its user base.

- Integrated feedback functionality.
- Voting over feature wish-lists concept.

15. Fermat must be the financially most efficient way to hold, move or spend the end user's money.

- Crypto Vault for multiple wallets concept.
- Financial AI Agents concept.

16. Fermat must facilitate the regional distribution and access to crypto currencies.

- Crypto Distribution Network concept.
- Crypto Brokers concept.

Appendix 02: Fermat's History

The Story

Pythagoras had to pay his first student to convince him to study with him. He gave him three oboli (about a penny) per lesson until one day he said there was no more money. The student replied that he preferred to pay, provided that they continued with the lessons. The rumor ran through the Greek islands and so the Pythagorean School, also known as the Secret Brotherhood, a philosophical / religious monastic order counting 300 members, was formed. By then counting and calculating had been widely practiced by the Babylonians and Egyptians but the Pythagoreans went far beyond that. “All is Number” was their motto attesting to their conviction that numbers were the key to life itself. Pythagoras strongly believed that relationships between numbers could reveal, through logical demonstration, all the secrets of the universe, hence he started producing his theorems. The Brotherhood walked briskly with his theorems but did nothing to share the secrets of the universe arousing the ire of the people who one day set the school on fire.

Despite the desperate attempt of his students to save him Pythagoras himself died in the fire. The Brotherhood was dispersed until Alexander the Great founded Alexandria and in order to attract scholars to the new city he took the advice of his general, Ptolemy: “Gather the great books; great minds will follow.”

Hence the Library of Alexandria was established with the overall aim of collecting every book ever written. Each traveler had their books confiscated upon arrival to Alexandria. They would be handed over to scribes, who made a copy for the owner and sent the original to the library. The extensive collection comprising 500,000 volumes made Alexandria the intellectual capital of the ancient world and as Ptolemy had envisaged attracted the most famous scholars of the era.

Euclid was put in charge of the math section and by inventing the “reductio ad absurdum” that is the proof by contradiction he took the findings of Pythagoras one step further. Anything that would defy logic seemed abominable to Pythagoras and as such was dismissed.

Though the Pythagoreans had discovered irrational numbers (π , or the square root of two) such was their fear of the “unutterable” numbers that their study was immediately forbidden and the disciple who came up with the concept of the “square root of two” was executed. Euclid announced to everyone that Pythagoras was abominable and advocated that irrational numbers would open a new door for mathematics and as such encouraged them to think about them without fear.

The First Peer to Peer Network

Julius Caesar's attack on Alexandria marked the demise of the Royal Library as it was burnt down in a devastating fire and the invaluable collection of scrolls was reduced to ashes. In an attempt to win Cleopatra's heart Mark Antony bought the entire Library of Pergamos, which had been second only to the Great Library of Alexandria, and presented the 200,000 volumes to Cleopatra. By having done so, Alexandria continued to boast the greatest library in the world until its eradication by Islamic troops

entering the city in the 7th century under the leadership of Caliph Omar. The Caliph ordered the destruction of all books on account of either opposing the Koran, which was considered to be heresy or for being superfluous. For years the water of the public baths of Alexandria was heated using those books to feed the fire. Mathematicians learned their lesson: in order not to disappear the Brotherhood expanded to everywhere without having a center but its members continued keeping in close contact with each other. This marked the beginning of the mathematical tradition of sharing every doubt, every finding and every gossip with the near and distant colleagues (they tell feedback). Should we call it the first and oldest peer to peer network?

Soon two separate branches of mathematics i.e., applied mathematics and number theory began to evolve dividing the mathematician community. Those more intrigued by the practical application formed working groups whereas those merely interested in numbers preferred to work alone. Newton accused the representatives of the latter branch of being vulgar ego jugglers, who had been wasting their time teasing each other with riddles which lacked concrete utility. An ardent supporter of that school was a judge in Toulouse called Pierre de Fermat, whose greatest pleasure in life was to play with numbers and then to send little notes about his findings to Descartes and Pascal. When it came to mental calculations such were his skills that Fermat had no inclination to spend time putting them in writing claiming that it would halt his reasoning and hated being asked about the intermediate steps. He was solution -rather than demonstration- oriented and found it appealing to be compared to big names.

One day he took the famous theorem of Pythagoras (the square of the hypotenuse equals the sum of the squares of the legs) and concluded that the equation has no integer solutions for powers greater than two.

“It is impossible for a cube to be the sum of two cubes, a fourth power to be the sum of two fourth powers, or in general for any number that is a power greater than the second to be the sum of two like powers. I have discovered a truly marvelous demonstration of this proposition that this margin is too narrow to contain.” he scribbled in the margin of his copy of the ancient Greek text *Arithmetica* by Diophantus thereby creating one of the most notable puzzles in mathematics driving many mathematicians to insanity until proven three hundred and fifty years later.

The Prize

In the first two hundred and fifty years it was a puzzle merely for young mathematicians who immediately abandoned it as soon as they had arrived to the conclusion, previously drawn by Gauss, that solving this puzzle would not contribute to the progress of mathematics. It was toward the end of the nineteenth century that a German amateur mathematician Paul Wolfskehl struck by love sickness was on the verge of committing suicide.

While waiting for the stroke of midnight to shoot himself in the head he passed the time reading about the latest progress in solving Fermat's Last Theorem. Touched by a flash of inspiration he began working on the solution and so lost he became in it that morning found him in his library. Even though his approach failed at last the intricate beauty of number theory once again awakened his desire for life.

The young Wolfskehl turned out to have more talent for business than for pure numbers and over the years he had become a millionaire, however, he never forgot that night. In order to repay the debt he rewrote his will bequeathing 100,000 Marks to the first person who could solve the puzzle. Following his death in 1906 the Wolfskehl Prize was announced fueling public desire to prove Fermat's Last Theorem, which greatly contributed to the progress of number theory.

The Solution

A mountain of failed attempts to win the prize had been accumulating in the basement of the University of Göttingen, 621 proofs, all flawed had been sent just within the first year of Wolfskehl's death. By 1993 no serious mathematician would engage in trying to solve the longstanding riddle of Fermat's Last Theorem, only fans, mostly inmates and lunatics, persisted.

Then in 1993 it was at the Isaac Newton Institute in Cambridge, the heart of the world of mathematics (a building created with the intention of bringing together the greatest mathematicians for a week each year, a building with not a single private corner, with offices having no doors and with blackboards even in bathrooms and in the elevator) that a freckled Englishman announced to his illustrious peers that working fully alone for ten years without the aid of a computer he had solved Fermat's Theorem.

Returning from his classes at Princeton University he would sit alone at the table, sometimes even twelve hours, a piece of paper lying in front of him every so often scribbling a formula like the old Fermat did. However, unlike Fermat he diligently noted down not only each and every formula but their endless development also.

Mathematicians say their specialty is an archipelago of small certainties scattered in a sea of ignorance. The real advances in mathematics are made when the bridges connecting these islands are built. In his dissertation Andrew Wiles established so many bridges that the entire history of mathematics could be reconstructed based on that, and that is exactly what the Hindu Simon Singh did by writing his beautiful book entitled Fermat's Last Theorem.

All representatives of number theory are enumerated in that book, however, the most remarkable of all is an unnamed colleague, who when facing Wiles as he was triumphantly leaving the stage addressed him with undisguised obfuscation: "And now that you have taken away the problem, what are you going to give us in return?".

The New Problem

The secret of Fermat's Last Theorem had been deciphered at last, however, even as of today the world is abundant in puzzles, which despite the tremendous advances in technology, are still waiting to be solved. All my adult life I seem to have been attracted to problems the complexity of which would deter even the most enthusiastic experts of the field. Inspired by the difficulty of the task I have always managed to rise to the challenge and sparing no time and energy succeeded in unraveling the mystery surrounding it.

After having designed mission critical systems for banks for more than a decade, I found myself in the United Arab Emirates establishing a techno-artistic startup called Nomad Inception boasting a technology which besides having enabled the automated creation of non repetitive Islamic geometric patterns on a scale never seen before also paved the way for new original non repetitive patterns for the first time in five hundred years.

Being Argentinean I have absolutely no relation to the Arabic culture, however, my fascination with the amazing intricacy and sophistication of Islamic geometric patterns helped me grasp the secrets of a traditional art form mastered by a handful of artisans only. Solving a thousand-year-old historical puzzle and thereby potentially contributing to the preservation of our cultural heritage is a vastly rewarding experience worth all the effort invested in it.

It was in Dubai, where I was setting up my company, that I first learned about bitcoin and having spotted the excellent opportunity my new mission conceived. This era has witnessed the digitalization of music and images. By solving the “problem of double spending” i.e., preventing a digital item to be sent to multiple people more than once, the last obstacle to the digitalization of money was eliminated.

It was in 2008 that crypto-currencies were first introduced, however, while the CD player and the digital camera have already conquered the world by enabling the wide-spread use of the inventions which called them to life, the technology equivalent to these designed to facilitate the mass conversion of users of paper money to users of digital money is yet to be created. Creation of such a technology is the new problem to solve. The problem is as complex as it can be, as 6 years have passed and some hundreds of millions of dollars of venture capital has been poured into start-ups working in this space since the solution for the double spending problem was found.

We suspect the solution involves finding the islands not yet discovered and building bridges to connect them both with each other and with the entire current archipelago.

Divine Sign

I delved into studying the problem with newfound enthusiasm, however, my commitments at Nomad Inception prevented me from devoting all my attention to finding the solution until a business meeting changed my life forever.

I am anything but superstitious, however, it goes without saying that the question a potential business partner asked me upon receipt of my business card, on which the word “inception” in Arabic appeared as company logo, could only be interpreted as a divine sign compelling me to leave Nomad Inception without further delay and turn all my attention to find the solution for this modern day Fermat’s problem. You may wonder what the life-changing question was...

“Have you been aware that the word ‘inception’ in Arabic used as your logo would read ‘bitcoin’ just by adding one dot to it?”

Luis Fernando Molina

Fermat’s Project Founder Unresolved directive in fermat-book.asciidoc - include::book-appendix-03-

money.asciidoc[]