Libra: An Economic and Technical Analysis

Jan Reiser
2012-067-815

January 13, 2020

Wirtschaftswissenschaftliche Fakultät
Major in: Finance, Banking and Controlling

Prof. Dr. Fabian Schär
Center for Innovative Finance

Contents

Plagiatserklärung

Ich bezeuge mit meiner Unterschrift, dass meine Angaben über die bei der Abfassung meiner Arbeit benutzten Hilfsmittel sowie über die mir zuteil gewordene Hilfe in jeder Hinsicht der Wahrheit entsprechen und vollständig sind. Ich habe das Merkblatt zu Plagiat und Betrug vom 22. Februar 2011 gelesen und bin mir der Konsequenzen eines solchen Handelns bewusst.

Jan Reiser

Abstract

Although, media and authorities primarily focus on the presented Libra coin and the Libra Reserve, this thesis elaborates the whole Libra ecosystem, fairly weighted. By explaining the main technical parts of Libra; the programming language Move, the Logical Data Model, Libra Merkle trees and the consensus protocol Libra Byzantine Fault Tolerance, a reader develops a technical understanding of Libra. Occasionally drawn comparisons to the well-established blockchains of Bitcoin and Ethereum as state-of-the-art additionally strengthen the intuition for the Libra ecosystem. The economic section, built upon the technical insights, examines the reachability of Libra's goals, namely, the move to permissionless consensus and governance, and to provide access to financial services for people in developing countries. The economic portion on central banks and monetary policy validates the political concerns toward the disruptive effect, and influence Libra may have on the world economy. A discussion on potential outcome scenarios on the not yet determined launch of Libra will subsume this thesis and provide an outlook on the development of Libra.

1 Introduction

In June of 2019 the Libra Association announced to launch a permissioned, programmable blockchain with new application possibilities around a cryptocurrency. Parts of the new ecosystem are a development platform and the new programming language Move. Move is designed to enable businesses and developers to create solutions for a variety of demands and services[1]. The stated goal of the Libra Association is to provide access to financial services for billions of people around the world[2]. To achieve this goal the new cryptocurrency Libra coin is designed to posses stability in value as a medium of exchange for users. The stability of Libra coin is guaranteed by backing Libra coin with a selection of assets and the promise to exchange Libra coin for a basket of fiat currencies at any time, by the Libra Reserve[3].

This announcement caused a mix of interest and concerns among the people hearing about it. While the initial interest toward a mass appealing cryptocurrency, among people familiar with cryptocurrencies vanished, politicians remained concerned about the potentially disruptive impact on central banks from Libra.

To be mentioned in advance; this thesis does not aim to classify Libra coin within the spectrum of cryptocurrencies. Libra utilizes cryptography, implements a concept of chained blocks in the consensus protocol and involves multiple entities in the process of decision-making but forgoes intrinsic concepts of a blockchain[4], that are inevitably outlined in this thesis.

At the time of this writing, the coverage of Libra in academic research is low; published work focuses on specific elements of Libra, omitting technical details and restrictions. Therefore, the focus of this thesis is on the examination of Libra's ability to reach the announced goals, and, the investigation of the political concerns, from an economic and technical perspective.

At the beginning, section 2 provides an introduction to the governance structure of Libra. Initially the Libra Association consists of a committee of founding members; one representative per founding member entity will have a seat in the Association Council, the authority in the governance of Libra. The Reserve is the provider of stability for the Libra coin, that manages the assets invested by the founding members through the purchase of Libra Investment Tokens. Section 3 elaborates the technical elements of Libra. Libra is a replicated, versioned, and programmable state machine that allows members

and clients to hold a replica of the database; maintained by the founding members that act as validator nodes. First, Move, the programming language of Libra is introduced. Move provides clients and costumers with the possibility of creating individual programs. Further, Transactions, the data structure of Libra and the consensus protocol; Libra Byzantine Fault Tolerance are introduced as interrelated core elements of Libra. The insights from the technical analysis are the foundation of the economic analysis in section 4. The announced transition to a permissionless consensus-and governance structure is examined in consideration of; incentives for founding members and technical limitations. Afterwards, the reachability of the goal; to provide access to banking services for billions of people around the world is evaluated with regard to the state of infrastructure in the third world. To draw a final conclusion on Libra; the potential influence on central banks and the political concerns regarding the disruptive energy of Libra are investigated through the utilization of the IS-LM model. Subsection 4.4 briefly discusses three potential outcome scenarios in consideration of all insights acquired in this thesis. The conclusion subsumes the key insights and elaborates the difficulty of a prediction for the development of the project Libra.

2 The Governance

The Libra Association, a declared not-for-profit and independent membership organization, functions as the head of the Libra ecosystem and consists at ignition of its founding members. Becoming a founding member is strictly regulated with regard to the technical, financial and social characteristics of potential candidates [5]; these founding members act as so-called validator nodes who operate the Libra Blockchain.. The clearly defined self related goal of the Libra Association (hereafter called Association) is its transition from the initial center of governance and decision-making power into the coordinating entity of the technical open-source development. Governance and consensus finding is announced to work permissionless in the future and the Association's influence on the Libra Reserve gives way to an automated management process[6].
At the first genesis state of the blockchain, however, the governance and transaction decisions will be exercised by the Association and a permissioned consensus protocol to guarantee security and high transaction throughput in the young state of the protocol.
The main governance structure of Libra consists of the Libra Association Council (hereafter called Council), the Libra Association Board (hereafter called Board), the Social Impact Advisory Board (SIAB) and the Association's Executive Team. However, the de facto decision-making power lies solely in the hands of the Council, which consists of a maximum of one representative per founding member. The members of both boards and the Executive Team leader are appointed and removed by a majority of votes from the Council; to remove a Council member, a supermajority of votes is necessary. Officially, only the Council has authority, as the other three governance entities serve the responsibilities delegated to them by the Council.[1]. However, only investing founding member are entitled to a seat in the Council: one can become a founding member without initially investing $10mio in Libra Investment Tokens which is the necessary condition to receive a seat in the Council at the initial form of governance. Further, after the targeted launch of the project in 2020 all founding members will have the same privileges and commitments, even and especially mentioned Facebook as an initializing force of the project. Nonetheless, the total voting power of a single Council member already depends on the financial endowment of the representative's entity. An additional vote in the Council can be purchased by a $10mio payment in Investment Tokens. To prevent excessive accumu-

[1] A detailed description of responsibilities can be found in [6]

lation of voting power by one founding member, the weight a single member can achieve in the Council is capped to the higher of 1 vote or 1 % of the total votes. In contrast to the voting power, the share of interest generated by the Reserve that the Investment Tokens entitle the purchaser to is not capped and grows in the number of Tokens held. The dividends paid to the investors (founding members with Investment Tokens) serve as compensation and direct incentive to act on behalf of the system. According to the white paper, the above mentioned transition to a permissionless system will start within five years of the launch [2], and after five years, at least 20% of the Council's voting power will be assigned to validators who operate nodes based on their holding amount of Libra instead of solely the Investment Token.

Libra Reserve

The Libra white paper introduces the Libra coin by mentioning attributes of the world's best currencies, namely, stability, low inflation, wide global acceptance and fungibility. This currency will be fully backed by a Reserve and supported by a competitive network of exchanges that are buying and selling Libra, as the following pages elaborate.

The highest goal of the Reserve is to support value preservation and stability for Libra by creating intrinsic value through liquid and stable assets. This prevents value swings that are known in established blockchains and therefore guards against speculation about value appreciation. The Association specifically aims to create Libra so that consumers and users know that $Value(Libra_t) \approx Value(Libra_{t+1})$.[2] Mutual fluctuations to the value of Libra will solely be the result of foreign exchange market movements as the Reserve is managed passively. The Libra Reserve will grow only by exchanging (purchasing) Libra for fiat currencies. Initially the money will be collected from users buying Libra and investors buying the parallel Investment Token. As has been stated, founding members will pay $10mio for one vote in the Council and the prospect of dividends. Received fiat money will be partially invested in assets with low risk and over time generate interest, which will be spent to maintain and further develop the Libra ecosystem and keep transaction fees low; leftover money will flow to holders of the Investment Token. This in turn means that holders of the Libra coin will not receive a part of the yielded interest, which makes the Libra coin not an interest yielding investment. With value preservation

[2]The exact process of stabilizing the coin is not yet published, on request to Christian Catalini from the Reserve I received the following response: "The idea is for people to be able to redeem for the value of a currency basket. This is still work in progress but one approach is the ability to redeem for fixed nominal weights (like the SDR)."

and liquidity in mind, the backing assets will consist of government securities from stable central banks with low expected inflation and low volatility assets. To grant liquidity, the governmental securities will be short-term securities traded in liquid, high volume(ten- to multiple billion US Dollar daily) markets. These restrictions only allow for a certain selection of markets. According to Der Spiegel the currency composition[3] of the backing was revealed upon request by a political spokesman[7].

To prevent centralization of the Reserve, the assets are held by distributed investment-grade custodians. For users to buy Libra, there is no way to directly interact with the Association that mints the coin; rather, there will be a hierarchy with the users on one side and the Association on the other. So-called authorized resellers who, integrate with institutions and exchanges, will act as intermediaries between the Reserve, the Association and anyone else who wishes to buy Libra. One could imagine the situation as follows: a user (**1**) who wishes to buy Libra contacts an authorized reseller (**2**) who, in turn, is the interface of exchange for the Reserve where fiat money will be deposited and the Association (**3**), who mints new Libra.

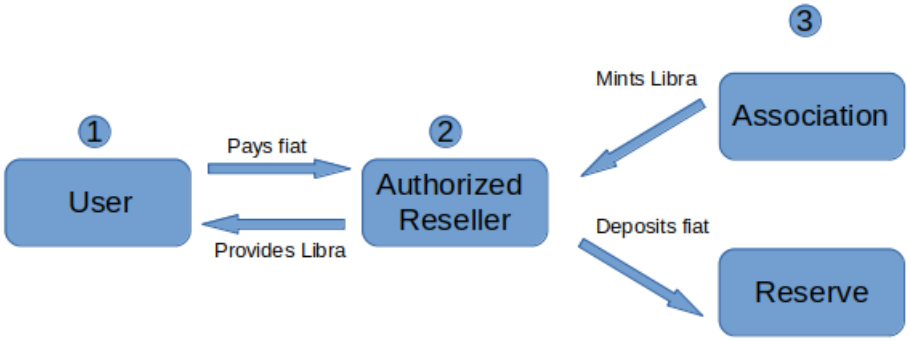

Figure 1: Indirect exchange between users and the Reserve
Users solely interact with the authorized resellers to buy or sell Libra. Step 3 is optional as long as the authorized reseller has sufficient Libra in stock.

This model may remind readers of a central bank but the concept of the minting Association and backing Reserve differs from regular central banks by separating the "mint" and the function of the buyer of last resort. The Association does not seek to create its own monetary policy; rather, it implements a currency board similar to Hong Kong's.

[3]Dollar: 50%, Euro= 18%, Yen= 14%, British Pound= 11% and Singapore Dollar= 7%. This statement confirms and specifies Catalini's answer.

The value provided in case of a reexchange from Libra to fiat may change due to economic circumstances, because the composition and value of the currency basket may change. Since the authorized resellers act on competitive exchanges, arbitrage situations will decrease; this, in turn, will lead to a smaller spread around the current value of the Libra coin. Throughout this time, the effort and costs invested in this complex project aim to provide the stable Libra coin for billions of people, enabling easier exchange and micropayments under trust and user protection. The introduction of the Reserve by the Libra Association omits key concepts, including the process of stabilization, supported currencies and a precise idea of the implementation in third-world countries. Excluding crucial information may, on the one hand, be due to the work in progress; on the other hand, it may be purposeful to not reveal early-stage developments to users, competitors and officials.

After this brief introduction to the governance structure and the Reserve, we can conclude that the power of decision-making in the Association will initially be tremendously focused on the Council. The willingness to move to permissionless governance and consensus was expressed and the official reason behind is the strengthening of competition and and lowering of entry barriers. The Libra Reserve focuses its operations on the stability of the Libra coin, and trust in Libra will come from the promise of re-exchanging Libra for fiat.

3 Technical Analysis

The Libra protocol implements the Libra blockchain as a decentralized database. This protocol is open-source and currently called by its prototype name, Libra Core. In this section, I explain the technical background of Libra and the workings of the system behind transactions, validators and clients. First there is a brief explanation of the new programming Language Move followed by an introduction to the transaction structure and requirements for transactions. The next portion of this section explains how the Authenticated Data Strucure and Storage function and points out similarities and differences to Ethereum. I then explain the consensus protocol, which is crucial to the update of the replicated database and elaborate why the consensus protocol takes high weight in transaction throughput. In the published yellow papers, there is a sometimes-confusing structure: by jumping in and, at first glance, repeating single sections, I attempt to prevent confusion as much as possible and provide insight by graphic representation; however, in some cases, there is no option other than jumping in.

3.1 The Move Programming Language

As I write this section, the Move language is not yet fully existent in the aspired high-level form. The available Move intermediate representation (IR) is a human readable form for developers and can be compiled down to bytecode to be executed by the Move Virtual Machine (MVM). This newly developed programming language enables the developers from Calibra to implement major portions of the protocol in Move. These portions are coded and stored in modules; the Libra coin, the validator management and the transaction processing are examples[1]. This way of implementation demonstrates the flexibility and expressivity of Move. Taking the validator management as an example, a change in the set of validators would be achieved by an operation within the specific module and would cause a new epoch in the ledger. Every new epoch starts with a new epoch-genesis block, and the round counter will be set to 0 [8]. This is a more convenient way of modification than needing to "update the whole blockchain"[4].

Move Goals: First-Class Resources, Flexibility, Safety, and Verifiability

Calibra stated the goal and application area of Move definitively: it aims to create a

[4]Referring to Bitcoin Cash, where an "update" on the blocksize lead to a fork of the entire blockchain.

financial infrastructure and enable developers to build individual business logic upon it; therefore, Move must satisfy certain conditions by design[1]. First-Class-Resources, Flexibility, Safety and Verifiability are fundamental factors to increase acceptance by a larger group of developers, businesses and officials.
First-Class-Resources means that Move resources can be understood as digital assets defined by a module. The module specifies the scarcity and access control of the resource and therefore enforces its value; programmers can define conditions to create, modify and destroy resources in a module. This general concept of representation is utilized to implement the Libra coin as a scarce asset in the Libra ecosystem. The underlying semantics, which were inspired by linear logic, allow the creation of custom resources. A simple real-life usage case is entrance tickets for a cinema: the coded *Module.Ticket* defines the *Resource.Ticket* and specifies the conditions to create, modify and destroy this resource. A ticket will be created if the condition of payment is met (scarcity) and will be moved to the customer who is buying it (access control). The digital asset *Resource.Ticket* then can be displayed on a smartphone and scanned to allow entrance into a predefined movie. This brief example illustrates the concept of creation and transfer of a "tokenized" asset. Furthermore, Move allows modules to call procedures that are defined within another module and therefore empowers developers enlarge this example. As flexible as modules are, they strictly enforce ownership and the right of modification on resources as well. The main procedure of every transaction is a transaction script. Flexibility means it enables users to define customized transactions which allow for more flexibility. Transaction scripts are able to summon procedures defined in external modules (on the Libra blockchain) or be utilized as one-time procedure. According to Calibra, Ethereum transactions are constrained to invoke a smart contract[5] and do not support one-off behavior for more expressive transaction constructs as transferring tokens multiple receivers.
To guarantee safety for memory, resources and types, the Move bytecode passes through an on-chain check by the bytecode verifier and if accepted, will be directly executed by the bytecode interpreter. Move's executable format allows this approach to implement source language safety properties without the need for a source compiler in the first place Strong on-chain safety comes to the cost of high computational effort and an increasing protocol complexity. The developers of Move confront this trade-off by designing the Move language to support advanced off-chain verification and increase overall veri-

[5][1]P.6,footnote

fiability. One example for Move's statically implemented safety is the ban of dynamic dispatch. Additionally, call instructions require a unique ID input for procedures, and modules' dependency on other modules follows a strict linear history of publication. The combination of these properties eases the reasoning for verification tools about the effects a procedure(-call) can have. Reentrancy attacks, therefore, are by construction, impossible.

To further illustrate the safety properties, I explain the functioning of the Bytecode Verifier (BV), which plays an important background role in the Libra protocol; afterward, I elaborate on how a module can work by explaining the previously mentioned currency module.

The Bytecode Verifier checks the binary Move format.[6] The binary format encodes tables containing structs, procedures, constants, and the like. By checking these tables for their structural correctness (e.g. wrong indices), semantics (e.g. wrong input arguments) and links (e.g. illegally invoking another procedure), the BV enforces safety. Any module or transaction procedure must, before execution, pass the BV. The semantic and linking verification follows several phases; packing coded instructions into sequentially organized blocks (not the transaction blocks), and performing stacking, type, reference checks on these block eventually leads to the acceptance of the code. In a last step before approval for execution, the self-explaining, so-called linker checks the code for correct referencing (declaration and signatures) of external structs and procedures. Further explanation on the Move BV can be found in the yellow paper [1]. On the one hand the implementation of the BV demonstrates, how serious the Libra Association is about the safety of their project. On the other hand, are these excessive safety checks an absolute necessity to build trust between users, potential investors and the still developing Libra ecosystem. In contrast, the Ethereum protocol does not provide a built-in bytecode verifier and can therefore, by design, be seen as less safe. Ethereum, however, possesses and active user base, and there exist several projects and instructions on how to verify code before actually executing it. Dan Emmons [9] provides a short article on why and how to undertake code verification utilizing Etherscan.io, and another commandline tool project can be found on GitHub[7].

[6] An example for compiled Move bytecode is provided in the Appendix 6.1

[7] `https://github.com/ConsenSys/bytecode-verifier`

Currency Module

"We use the term resource safety to describe the guarantees that Move resources can never be copied, reused, or lost. These guarantees are quite powerful because Move programmers can implement custom resources that also enjoy these protections. As we mentioned, even the Libra currency is implemented as a custom resource with no special status in the Move language."[8]

This statement is a demonstration of trust and a key feature Move. Programmers, therefore, can be confident that their resources are safe from outside attackers. An example of what a module can perform is provided by Calibra about the Currency module. In a first step, the Currency module is created.

```
module Currency {
  resource Coin { value: u64 }
  // ...
}
```

Figure 2: Creation of a module, source: Move: A Language With Programmable Resources [1], p.10
Move IR syntax, displaying the creation of a module with a resource

These lines of code declare a module named **Currency** and a resource within **Currency** named **Coin**. There is a resource Coin, but not yet a possibility to do something with Coin. Therefore, a programmer must specify a procedure that can be invoked in a transaction script; note that this procedure in a transaction script is the reusable case, meaning that more than one transaction script can utilize this procedure. For the sake of simplicity, I solely explain the **deposit** procedure.[9] The displayed procedure uses **Coin** as an input and combines it with **Coin** in the receiver's (payee) account. This happens through the destruction of **Coin** after unpacking it (the value is recorded). Unpacking is necessary to destroy a resource; without the, type system would not accept a destruction. To transfer **Coin** one must get a reference to the unique **Coin** resource in the receiver's

[8]Direct citation from: Move: A Language With Programmable Resources [1], p. 9, section 4.1, last passage

[9]A second procedure called **Withdrawal** can be found on p.11. of "Move : Language With Programmable Resources" [1]

```
public deposit(payee: address, to_deposit: Coin) {
  let to_deposit_value: u64 = Unpack<Coin>(move(to_deposit));
  let coin_ref: &mut Coin = BorrowGlobal<Coin>(move(payee));
  let coin_value_ref: &mut u64 = &mut move(coin_ref).value;
  let coin_value: u64 = *move(coin_value_ref);
  *move(coin_value_ref) = move(coin_value) + move(to_deposit_value);
}
```

Figure 3: Example of a procedure for resource **Coin**, source: Move: A Language With Programmable Resources [1], p.10
A reusable implementation of a procedure named **deposit** within the module **Currency**

account(BorrowGlobal<**Coin**>) and add the recorded value to the value of the receiver's **Coin** resource. However, worth mentioning is that the Move Virtual Machine (MVM) strictly executes unintentional programming mistakes by its creator. A typographic mistake could, for example, increase the total number of **Coin** every time the procedure runs and be exploited by looping this procedure. The Move type system would accept formally correct code, so a programmer could, by mistake, make the created resource **Coin**, de facto worthless. That clearly states that, a user still holds responsibilities in utilizing Move, as no currently existing protocol could know the programmer's intention in advance. It remains the user's responsibility to maintain safety inside modules and the Move type system's responsibility to enforce the safety outside of the module. This example demonstrates the intuition behind a module-resource-procedure relationship. It is possible (and intended) to escalate this example by adding further resources and procedures as well as implementing a graphical user interface and so create an ecosystem within an ecosystem. This would lead to the creation of so called decentralized applications (DApps) in Libra as they utilize the blockchain as back end and a module as application programming interface to present the DApp front end. The term decentralized application, in the way it was introduced in the environment of permissionless blockchains, does not suit the Libra context. DApps with regard to Libra are further called Apps.
In a nutshell, a Module contains data that defines a resource and a procedure. This procedure can be called in a transaction script and perform arbitrary (as long as it is accepted by the BV) operations on said resource.

The developers provide a brief description of further functionality.
Core Functionality: Reserve management, Libra coin, collection and distribution of fees, to name a few. These functionalities will likely be implemented in modules and further demonstrate potential usage of Move.
Parametric Polymorphism: This update will likely bring Move's flexibility to the level of Ethereum; it has been explicitly mentioned that the verifiability and safety guarantee will not be abandoned[1].[10]
Improvement for developers: A source language format for Move is planned. The exact date and how the BV will handle the source language is not further specified.
A specification language: This intends to further increase the system's safety on a level higher than the BV and, in a future step, will be applied on user-created modules. This could lead to an unofficial standard if users refuse to interact with modules that are not "stamped" by the verification tools.
Finally, Calibra states intentions about the potential size of the developer platform and therefore is directly cited.
"Support third-party Move modules. We will develop a path to third-party module publishing. Creating a good experience for both Libra users and third-party developers is a significant challenge. First, opening the door to general applications must not affect the usability of the system for core payment scenarios and associated financial applications. Second, we want to avoid the reputational risk that scams, speculation, and buggy software portray. Building an open system while encouraging high software quality is a difficult problem. Steps such as creating a marketplace for high-assurance modules and providing effective tools for verifying Move code will help." From: "Move: A Language With Programmable Resources"[1],p.20.
All these developments in progress demonstrate again the seriousness of the Libra Association with their ecosystem. Not only will there be a new stablecoin called Libra, but there will also be significant ground for developers, businesses and consumers of new, digital services that the media and especially officials do not yet seem to understand and anticipate. Currently, there is neither concrete information about the expense of gas[11] a module will consume per operation nor further plans on exactly how the transaction

[10]Whilst the implementation of parametric polymorphism is true, the flexibility gain to the level of Ethereum is a ventured guess by the author of this paper.
[11]Gas is an, in advance specified measurement, denominated in Libra coin that modules consume during computation steps.

throughput will be escalated to a number that will satisfy the potential outcome. Requests to Calibra developers remained unanswered at the time of this writing. In contrast, the more-established Ethereum blockchain provides a fee schedule in Appendix G of their yellow paper for "a number of abstract operations that a transaction may effect" [10].

As we see, does the Libra project not invent the concept of smart contracts, DApps or the tokenization of assets; these were already in existence and some have even been standardized (e.g. tokenization in ERC721). However, Move introduces a more convenient way for users to write modules (contracts) and business logic under high safety standards.

3.2 Core Elements of Libra

The Libra Blockchain does not come in actual blocks; rather, it is a sequence of transactions represented by a Merkle tree. Every transaction executed on the Blockchain will be attached as a so-called leaf; nonetheless I will still refer to Libra as a blockchain to avoid confusion. I provide brief a explanation of Libra Core, the Logical Data Model, and the Ledger State to ease the understanding of this section. After the explanation, transactions, the Libra Merkle trees and the consensus protocol are elaborated to illustrate the technical implementation of Libra as a whole.

Libra's protocol, Libra Core. Libra Core defines the blockchain as a cryptographically authenticated, replicated database. The ledger stored within this database contains the entirety of information, such as: transactions, and programmable resources. Common in the field of distributed ledger technology, clients can query validators about the latest state of the ledger or create a replica of the entire database and verify the correct state themselves. The execution of transactions (provided that there are sufficient funds for gas-fees) depends on the decision of the granted validators and therefore distinguishes itself from well-known permissionless blockchains, which draws Libra into the corner of already established payment systems. However, one must also note that the transaction history is totally visible for anyone who demands it.

Logical Data Model

In short, the decentralized database visible to clients and validators is organized within the Logical Data Model[11]. Data in the Libra Blockchain is stored in a single versioned

database. An $UInt64$ displays the version number and the number of transactions executed. Knowing this, one can say that the database is updated to a new version with every transaction executed. At version i of the database, there is a tuple (T_i, O_i, S_i), standing for transaction, output and state. Since transaction execution is deterministic it will define: (S_{i-1}, T_i,) to (O_i, S_i) in words; T_i executed against $State_{i-1}$ creates $Output_i$ and the new $State_i$.

Ledger State

The ledger contains all information about users' accounts, in any version of the database. Therefore, validators always need to know the latest state to execute, and vote on transactions. The model is account-based, which means that account addresses (keys) are mapped to account data (values), which are collections of modules, which store code, and resources, which store data values[11]. To participate in Libra, one needs an account which can be created by generating a new key pair, which consists of a verification and a signature key (vk, sk). The verification key can be understand as a public key and the signature key (in the broadest sense) as private key in Ethereum. The account's address is the cryptographic hash of the verification key H(vk)= address[12]. To sign transactions coming from the newly created address, the client utilizes the signature key, which can be changed without changing the address. It is important for any reader to understand the implication of the previous sentence. The ability to change the signature key, implies that the signature key is stored in an external database, therefore, at least one other person has potential access to the signature key. A change of the private key is impossible in any decentralized and permissionless blockchain. The possibility to change the signature key in Libra violates the nature of blockchains. Generating a new private key is the only possibility in Ethereum to "change" the private key.

3.2.1 Transactions

Transactions are the way for clients to update the ledger state. Containing a transaction script and arguments (e.g. amount of Libra coin or addresses) as well as the current state of the ledger, the transaction output (O_i) is deterministic. Executed by all validators the ledger state will be changed only after finding consensus about the output of the trans-

[12]Using SHA3-256 and EdDSA used on the edwards25519 curve. Further reading see S. Josefsson and I. Liusvaara, "Edwards-curve digital signature algorithm (EdDSA)," RFC, vol. 8032, pp. 1–60, 2017

action, and the transaction being committed. A committed transaction, in addition to a new state S_i, generates a code about the execution status, an event list and gas usage, which is documented in O_i.[13]. Events, identified with a specific, unique key to identify the event-emitting structure, contain a payload (e.g. transactions) that holds information about the event. After committing a transaction, an event is added to the history of the agreed ledger. One must understand that events serve as evidence for a successful transaction with a transaction effect, as intended. Therefore, only querying for the transaction itself may lead to a misconception of success since failed or interrupted transactions are also included in the database. This means that every transaction, successful or not, is documented and saved within storage, and only successful transactions produce events as evidence. By storing transactions and corresponding events the ledger history demonstrates how the latest ledger state was computed. In the logical data model of Libra Core, the transactions are presented in a sequence. In contrast to other blockchains the logical data model does not utilize the concept of blocks as structure of storage; transactions are batched into blocks to ease consensus finding. Block content will be elaborated upon later in the consensus protocol. Clients can query current and past ledger states in a specific way: they can ask for specific information about a specific address, such as the balance at address 0x1 at ledger version five. To answer this query, validators will utilize the ledgers history of states, transactions, outputs and events[11]. Since transactions and outputs are deterministic, a client could also recompute a sequence of transactions and compare this to the information received by the validator. This allows clients to be sure about the correctness of information received, since state and output in a manipulated environment would deviate from self calculated results. This is a central point in well-known blockchains such as the Ethereum blockchain, where manipulation will be indicated quickly through the permanent audit of the distributed ledger.

Transaction Execution

We already know that transactions update the ledger state; now we see which requirements regarding the ecosystem must be fulfilled to execute these state-changing transactions and how transactions are structured within the Libra ecosystem.

Initial State. The so-called genesis state can be understand like the Big Bang: the core components are defined within modules, including account logic, validator selection, Li-

[13]That shows whether the transaction was successful or failed, how much gas was consumed and an event list as proof of a successful transaction.

bra coin, the consensus protocol and other crucial ingredients. The genesis state, at first is empty, and a special transaction T_0 defines the modules and resources. All clients and validators must follow —they are even configured to follow— and accept ledger history only when it starts with the special transaction T_0 and can check for its existence by looking for the unique authenticator of this transaction. In this first state, there must be sufficient instantiations to pay fees for the initial transaction and a defined validator set to form a quorum to sign the authenticator (hashed transaction, authenticator and hash will hereafter be used interchangeably) to the first transaction.

Determinism. The previously mentioned deterministic transactions must be guaranteed to allow validators to agree on the same resulting state from executing the same sequence of transactions independently of each other. A different order of the same transactions would lead to a different state and a different authenticator for the state. This also enables the re-computation of the ledger history by clients.

Metered. Libra utilizes a similar transaction fee system as the Ethereum blockchain, measuring fees in gas units denominated in Libra coin. Gas costs are dynamically calculated in relation to the computational power utilized, which means that validators will execute transactions with higher gas prices as a higher priority. This approach has been implemented to reduce computation demand when the Libra system reaches its capacity limits, such as within "denial of service" attacks. These attacks "flood" the system with eventually failing messages to slow down the blockchain. The fee itself is calculated exactly as in the Ethereum blockchain, namely, $gas\ price \times gas\ cost$. As in Ethereum, transactions will be stopped if they run out of gas; these transactions do not lead to state update, but they will appear in the transaction history and already used gas will not be returned to the client.

Asset semantics. Since real-world values are encoded inside the ledger state, the MVM in the Libra protocol must ensure safety properties that correspond to the assets, such as Libra coin. With these requirements, the safety and determinism of transactions can be guaranteed, and the threat of attacks on the system can be minimized. The execution of a transaction is performed in six steps[11]. A brief description of the six steps and the intention behind each is included below.

1. Check signature. This demands that the transaction signature matches transaction data and the sender's public key.

2. Run prologue[14].
This step authenticates the sender by comparing h(vk) to the authentication key stored in the sender's account to prevent erroneous acceptance without correspondence to the account.
Then it checks for sufficient gas so that the transaction does not fail in epilogue.
Finally, it ensures that the transaction sequence number matches the sequence number under the sender's account and guarantee that this transaction is not a replay of previously executed transactions.
3.Verification of transaction script and modules.
Crucial properties are checked by the BV to guarantee safety.
4.Publish modules. Contained modules are published under the sender's account.
5. Run transaction script. The MVM binds the parameters of the transaction script to the transaction arguments and executes it. If successful, the events emitted by the transaction script are committed to global state.
6. Run epilogue. This step charges the user for gas and increments the sequence number of the transaction sender. The epilogue always runs if Step 2 is successful even if all steps after Step 2 fail; therefore, Step 6 also charges gas for unsuccessful transactions. All transactions surviving Step 2 are documented in the ledger history; transactions dying before Step 2 (e.g. insufficient Libracoin to pay for stated gas) are not recorded.

This section provided an insight on the transaction structure and the requirements for a deterministic transaction execution. The next section explains how the data is stored and how transactions and storage contribute to the whole.
A tabular illustration of the transaction structure can be found in the appendix.

[14]Prologue and epilogue are procedures defined in the LibraAccount module.

3.2.2 Libra Merkle Trees (Authenticated Data Structures and Storage)

After introducing Move and transactions in Libra, it is of further interest to know where modules written in Move and transactions which lead to state updates are handled and stored. This section explains how Libra Core stores and authenticates the integrity of data and transactions.

Transactions change the state of the ledger; the short authenticator a represents the ledger history including the new transaction[15]. In consensus finding, the validators agree on transaction ordering and the resulting execution, and the latest short authenticator is collectively signed to the new version of the database. Since a is collectively signed by the validators, a client can utilize it as a short proof to verify the validity, integrity and finality of the ledger's history even without trusting the a providing validator. Like many other blockchains, the Libra blockchain bases its data structure on Merkle trees, which are elaborated upon later, and creates the short authenticator out of these trees (tree root). However, the Libra blockchain utilizes Merkle trees with some modifications. The following pages detail the background of the Authenticated Data Structure, how the Libra Merkle tree approach works, the advantages it provides and the substructures (events and accounts) it contains.

Background on authentication structure

Historically speaking, Merkle Trees can be dated back to 1979 [12]. When he was 27, Ralph C. Merkle's book about public key systems was published. Mainstream attention to Merkle's ingenious invention was brought by the infamous Satoshi Nakamoto, who implemented it into the Bitcoin protocol to validate the authenticity of the blockchain data structure.[13].

[15]The generation of a is of course deterministic like the execution of a transaction.

How does that work in Libra? Holding the short authenticator a that commits to a larger data structure *Data*, a wary client may want to check the validity of a, for example, for the inclusion of a balance (value) under a specific path (key). Therefore, a validator computes $f(Data)$[16]and returns result r as well as proof of correctness π to the client. The client can run the command Verify(a, f, r, π), which will return true if $f(Data) = r$ and therefore proves balance (value) ϵ *Data*.

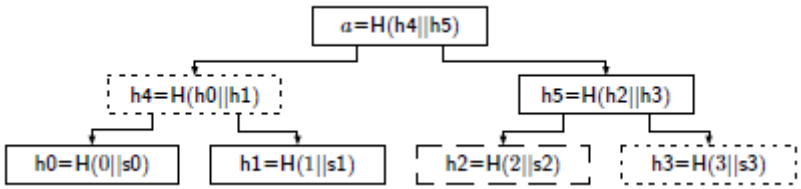

Figure 4: *Data* in tree structure to illustrate the authentication process, source: "The Libra Blockchain"[11], p.13.
In this example, a serves as the short authenticator for *Data*, $f(Data) = r$ = s2 and proof of correctness π= [h3,h4]. Note, π is not a hash function on h3 and h4.

Figure 4 depicts the example above in a slightly more technical way. To break down this easily confusing information, the inclusion of s2 is authenticated if Verify (a, f, r, π) is true and, in turn demonstrates that a = H(h4‖H(H(2‖r)‖h3))[17].
The already mentioned Merkle trees are a frequently utilized approach to store mappings between string values and integers. Mapping every integer key i to a string value si creates the structure *Data* (see Figure 4). The creation of the authenticator, which is the root of the Merkle tree, results from hashing each leaf with its so-called sibling and gradually continuing with the emerging nodes. The previously mentioned proof of correctness π comprises of the labels of the sibling of each of the ancestors of the queried node under key i. The cryptographic hash functions leading to the root of the tree are declared to be different for leaves and nodes, but are not further elaborated by the developers.

[16]Function f initiates the proof of existence for key,value pair (k,v) in *Data*
[17]H(h4‖H(H(2‖r)‖h3)) = H(h4‖ H(h2‖h3)) = H(h4‖h5) = a

Ledger history

Libra's storage protocol is another crucial core element implemented through a module. The storage module's physical engine is called RocksDB[14], which was created by Facebook[15] as an optimization of Google's LevelDB[18] to increase the performance of many central processing units in storing key-value pairs. According to Paul Dix[16] LevelDB shows better on disk usage abilities, whereas RocksDB performs better in deletes and reads; therefore, it fits the Libra ecosystem with short proofs and improved storage pruning abilities. The incompatibility of the storage module (it stores multiple data types) and RocksDB (byte arrays) is overcome by wrapping key-value pairs to serialize them as demanded in the accounts[17].

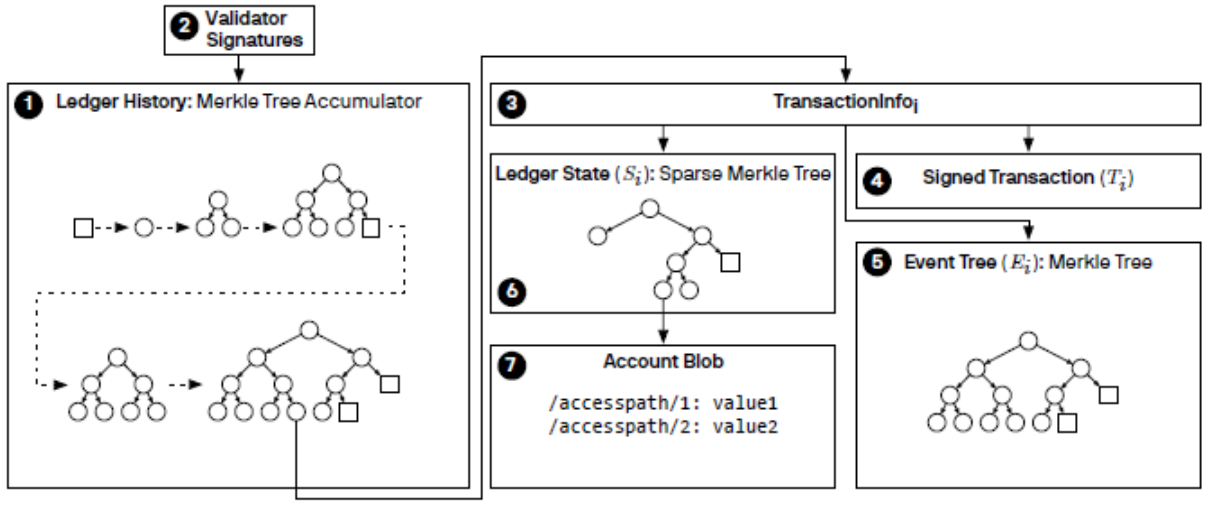

Figure 5: Complete authenticated data structure of Libra, source: "The Libra Blockchain" [11]

1 displays the ledger history; its hash authenticates the whole ecosystem; **2** Validator signatures for client queries; **3** TransactionInfo$_i$ contains hashes of **4** Signed Transaction$_i$; **5** Event Tree list for transaction$_i$; **6** new state after T_i; **7** binary large objects representing accounts.

Since Bitcoin demands no higher storage requirements, the concept of simple Merkle trees was sufficient for the system to protect its authenticity. The following pages explain how higher data amounts and states are represented. Therefore, a simple Merkle tree is no longer sufficient for storage representation. The principle of data authentication by

[18]Ethereum uses LevelDB(Go, C++ and Python implementation) and RocksDB (Rust implementation Parity)

Merkle hash trees remains.

The Libra protocol utilizes a single Merkle tree (which is hereafter referred to as main tree) to represent the entire ledger history (bullet point **1** in Figure 6). Each leaf in the main tree contains a *TransactionInfoi*(bullet point**3** in Figure 5), which, in turn, contains information about events, the current state and transactions. If clients wish to retrieve information on the state at version i or an event at version i, they perform a *Ledger Info* lookup, which returns the authenticator. The Ledger Info contains information about the versioned ledger state, its events, and transactions and is signed by the validators. By using the Merkle tree accumulator approach the Libra protocol supports appending operations, these operations enable consecutive computing of the ledger history by appending new transactions.[19] An advantage over Bitcoin is the proof of continuing ledger history commitments where two different authenticators A_i and A_j with $j > i$ present their identical history until version i. The state storage protocol allows efficient storage of all events and transactions in different versions, but since the ledger history (moreover, its root hash) serves as an authenticator for the whole system, this could lead to a shortage of storage for validators. This problem is solved by allowing validators to eliminate past states because they are no longer needed to process new transactions and generate events. The Merkle tree accumulator approach allows the elimination of historical data, since its storage demand in appending new state records is logarithmic.

The Ledger State Tree is the second tree in the Logical Data Model. The state tree is where the data lives(depicted by bullet point **6** in Figure 5). One must be aware that this sparse Merkle tree, graphically illustrated in binary form, actually looks different. The binary form is simulated to increase the performance of proofs[17][20]. The ledger state S_i at version i represents all accounts at this version as a mapping of key-value pairs. Keys are the hashes of the account addresses, and the mating value is the account authenticator. This mapping happens in a size 2^{256} Merkle tree, which becomes an inefficient (impossible) way to display. The implementation by Libra Core double crops the Merkle tree to create a more tractable representation. This is implemented through a package in the storage module called "jellyfish-merkle"[21], which is written in the language Rust.

[19]For further reading an the tree accumulation approach see [18]

[20]In fact the structure of Libra's Ledger State tree is not very different from Ethereum's Patricia Merkle trees with likewise performance. Libra's salient feature is the shorter proofs through binary form simulation, which in turn makes it more feasible for mass appealing commercial adoption.

[21]`https://github.com/libra/libra/commit/2531a2ad5b92cb4a4262a082c60477dc7e190e8f`

Rust is deeply involved in the development of Libra and serves as a role model for Move. As displayed below in Figure 6the sparsing occurs in two steps. From **1** to **2**, branch of entirely empty nodes comprising subtrees is replaced with empty placeholders. The leaves are still stored at the bottom, which means there are still 256 hashes needed to compute the leaf modifications, but the size is now tractable. A second step from **2** to **3** replaces subtrees with one leaf by a single node. The resulting sparse Merkle tree needs less hash computations when operations on the map are performed.

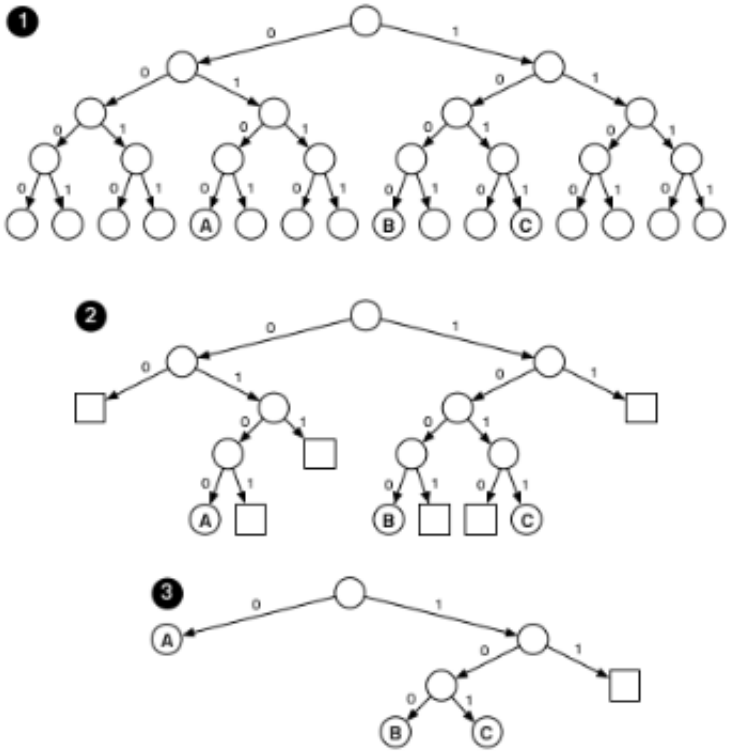

Figure 6: Full binary tree becomes a sparse tree, source: [11]
The sparse tree at version **3** in this figure represents bullet point **6** in Figure 5. This figure illustrates the simulated binary path structure (0,1) required for better proof performance.

Accounts. The state tree stores the accounts. Labeled as binary large objects (blob), data is stored here. Logically speaking, accounts are collections of modules and resources under an account address. Physically, these accounts are handled as ordered maps of keys to values similar, to paths in file systems[11]. Accounts are serialized as lists of values and access paths (here the wrapping process reveals its importance) sorted by access paths and the account authenticator is generated by hashing the serialized account (its paths and values). This results in recomputation of the authenticator after every transaction or other modification of an account. In the Libra protocol, accounts serve as ordered maps of paths to values. With this in mind, Move encourages users to hold resources under their accounts. This, on the one hand, allows for better programming abstraction, since not the whole code is displayed to the outside, but on the other hand, adds to costs for clients. Since account storage will grow and could become a problem, the sustainable usage of storage is enforced by the MVM which computes a deterministic expiration time during which the users must pay a fee based on its storage size and time. The expiration time, after which the MVM will deny access to the account, is represented in the account's authenticator. Since the developers at Calibra are not yet finished designing the rent-based system, one can only speculate about the amount of rental fees.

The Event Tree. During execution of T_i the emitted events are listed in E_i(bullet point **5** in Figure 5). We must remember the importance of events, as only an event proves the success of a transaction. Events are stored as leaves in their own tree and are serialized as a tuple (A, p, c) which represents access path (A) to the event structure emitting the event, data payload (p) and counter (c). Tuples are indexed by j, which is the order of events emitted during T_i. Inclusion proofs by validators can be constructed for clients since $TransactionInfo_i$ includes root hash E_i, which in turn proves $(j, (A, p, c)) \epsilon E_i$. The same access path A could contain more than one event, therefore the counter c event; therefore, the counter c plays its role as the listing basis of these events under A. Stored in the ledger state, the counter C is locally separated from the event tree and increments with every event executed under path A. Therefore, C displays the total list of events under A. If clients query a service they do not directly trust for an event list under path A, then clients themselves can review the response for correctness and completeness. After receiving the set of tuples corresponding to the events *(i,j,A,p,c)*, the proof of inclusion of events under A can be provided through authenticator E_i. Now the client knows the list is correct but does not know about its completeness. If the client knows

the state, obtains an authenticator for correctness of the ledger state and then queries the counter C for the event structure under A, then the client also knows the total number of events. By comparing the number of distinct events j and C the client can be sure of completeness[22] and correctness. We can therefore say that an event in the event tree constitutes the success of a transaction, and the counter C in the state tree provides us with the possibility of confirming the completeness of events.
This section outlined the database structure of Libra, how data is stored and authenticated, and the importance of Merkle trees for this process. Worth mentioning is the uncanny inspiration the developers of the Logical Data Model got from Ethereum's implementation of Merkle Patricia trees.

Trees in Libra and Ethereum

Both protocol's data authenticity strongly depends on their application of Merkle trees. To ease the understanding of the trees utilized and the purpose each tree serves in the Libra ecosystem, a brief comparison to the trees in the Ethereum blockchain is graphically illustrated on the following page. Since Libra and Ethereum are intrinsically different in many regards, one must be aware that a 1:1 comparison of trees is not possible. Nonetheless, the following pages strengthen the intuition behind the trees utilized in Libra in a conceptual way. Readers familiar with Ethereum will immediately recognize the similarities; for readers unfamiliar with Ethereum, Gavin Wood [10] provides an in depth explanation on Ethereum.
Quickly noticeable differences are the Parent Hash and Beneficiary contained in the Block Header of Ethereum; both of these items directly highlight the intrinsic differences of systems. The Parent Hash is usually a direct link the block earlier in the blockchain (anomalies, e.g. forks during attacks or revisions of the blockchain). This procedure maintains the integrity of the Ethereum blockchain and indicates for all participants where a new block will be appended to continue the chain. The Logical Data Model does not build a chain of blocks; therefore, the authenticators of earlier versions are included in the ledger history. The Beneficiary in Ethereum is the person (in a broader sense comparable to a transaction block proposer in Libra) who "mined" the new block; mining is part of the protocol in the proof of work consensus and demands, compared to the Libra Byzantine Fault Tolerance (LBFT), considerable energy, which is compensated by awarding the successful miner with Ether sent to the address in the Beneficiary field. Mining is the

[22] Completeness given if the sum of Distinct Events = C

central concept of decision making in the decentralized Ethereum blockchain. In contrast to the consensus protocol in Libra, the success in mining in Ethereum not granted.
To identify the similarities, we begin with the Event Tree in Libra, which can be regarded as Libra's counterpart to the Receipt Tree in Ethereum that contains logs about formation of the state. The root hash of the Event Tree enters the Ledger History tree, and the root hash of the Receipt Tree enters the Block Header. We can therefore say that the Ledger History Tree in Libra counterparts the Block Header in an Ethereum block. The Signed Transaction, pictured in brown in the Libra ecosystem, can be understood as corresponding to Ethereum's Transaction Tree (green); both entering their particular superordinate structure. Ethereum's Storage Tree is where data (e.g. smart contracts) for accounts in its own Storage Tree each, is stored and enters the State Tree. The State Tree matches the function of the Ledger State Tree in Libra as well. The Ledger State Tree in Libra represents all accounts' states at a specific version and enters the Ledger History Tree as the Ethereum State Tree enters the Block Header respectively. If one were to go a step further, then one could say that the Ledger Info of Libra corresponds to the block as a whole in Ethereum. The Ledger Info in Libra can be queried by clients. The whole block structure of Ethereum is intentionally omitted for the sake of clarity.
After examining the data structures, we can subsume that Libra and Ethereum are both built upon on a similar data and storage framework. Although both core protocols implement totally different consensus protocols their, relatedness in several points is uncanny and demonstrates the sophistication introduced with Ethereum.

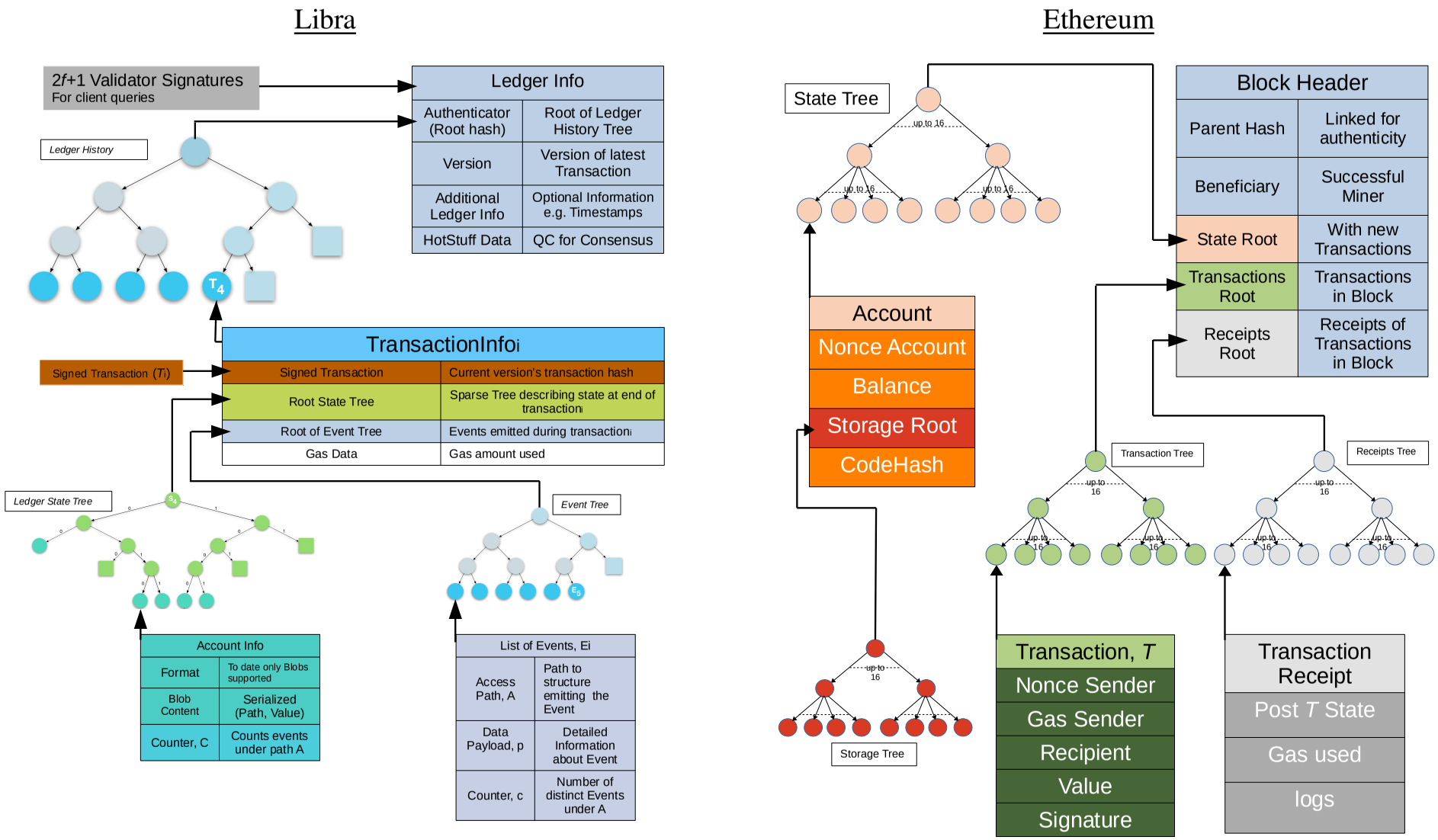

Figure 7: Trees in Libra and Ethereum, author's interpretation

Left: Graphic representation of Libra's authenticated data structure, information contained in each tree and the embedding of the authenticators into the ecosystem as a whole. This illustration provides a simplified snapshot of the current, still developing ecosystem. At the date of writing QC in the Ledger Info is still referred to HotStuff as consensus protocol.

Right: Graphic representation of the tree structure in Ethereum in a simplified form leading only to the Block Header.

3.2.3 Consensus and Permission

After the sections on transactions and data structure, it is important to understand what is happening between a signed transaction message sent by a client and the update of the state. The decision whether a transaction will be committed and the state updated is part of the consensus protocol.

At the heart of this new blockchain is a consensus protocol called LibraBFT

-State Machine Replication in the Libra Blockchain[8], p.1

This directly cited statement indicates the importance of a consensus protocol for any blockchain. Blockchain consensus protocols usually decentralize the power of decision-makers in an environment without trust and simultaneously guarantee the reliability of the system. As well known blockchains like the Bitcoin and Ethereum blockchain are considered decentralized and permissionless, the Libra Blockchain is permissioned. The term "decentralized" is true in the literal meaning of the word, but violates the principles of blockchains in their original intention. One may wonder why the Libra Blockchain is permissioned at all. This legitimate question is answered in this section. The focus of the consideration for a permissioned consensus protocol are to transaction throughput, scalability and energy consumption.

Consensus protocols in the style of the Nakamoto Consensus with the proof of work at heart are tremendously robust. This robustness even achieves fault tolerance[19] and creates incentives to cooperate in line with the protocol. But comes, the way it is implemented in Bitcoin and Ethereum, at the cost of a slow, hard-to-scale, energy consuming blockchain. Slow and energy consuming because of its predefined computational effort to reach progress and consensus by hashing the block with its content while incrementing the block's internal nonce (a simple UInt) until undershooting a defined target to commit the block. Hard to scale by its predefined block size and creation time, which limits the number of included transactions, that cannot be increased without significant changes and uproar within the community All of these challenges are handled by the Libra Association through the implementation of the LBFT protocol.

The LBFT is a modification of the HotStuff protocol, which, in turn, is an upgraded practical Byzantine Fault Tolerance (pBFT) consensus protocol. All of these consensus protocols operate in rounds of communication between validators who decide on the commit of transactions. The proposal of a block of transactions to be decided on comes

from a designated leader who is known in advance, as it is deterministically selected by the ProposerElection module. To prevent attackers from targeting a future leader (proposer), the Libra developers investigate different proposer election strategies,such as introducing pseudo-randomness [8]. To ease the understanding of the evolution of LBFT, a brief recapitulation about the history of this consensus protocol is provided. A graphic illustration of the evolution can be found in the appendix. As the word tolerance states, these consensus protocols tolerate up to a threshold number of faults within a round of consensus finding. The threshold value of faults in the LBFT is $f < \frac{N-1}{3}$ where N is the total number of validators and f depicts the number of "faulty" validators. The fault tolerance of the protocols is essential for consensus finding, without accepting a certain number of faults, for example crashes of validators, it would be nearly impossible to reach consensus at all.

Background on BFT consensus protocols

The approach of synchronous consensus-finding is intuitively sound and, on first glance, simple to implement. It allows a finite time-bound Δ for messages being delivered and received between nodes[23]. However, the strong assumption of a maximum delayed message delivery time Δ makes the synchronous model impractical for larger and distributed settings. In the Libra setting, a replicated state machine, where every validator node holds its own (identical across validators) replicated database, a synchronous setting would be inoperable, even without the presence of malicious actors. Remembering this makes the implementation of a synchronous communication network considerably more difficult and vulnerable than one may assume.

Therefore, there must be a solution in an asynchronous setting to reach agreements. Asynchronous models drop the delivery-bound Δ and assume eventual delivery in an undefined timeframe (including infinity). Now if a single node undergoes a failure, it is proven to lead to an infinite execution and impossibility of consensus-finding.[24]

Three years after the proof of impossibility in asynchronous systems, a solution was introduced. A hybrid model, named partial synchrony, was developed by Dwork, Lynch and Stockmeyer [21] that allows two phases: a phase of synchrony where an honest leader appears, and progress is achieved, and an asynchronous phase in which rounds are retired with timeouts until an honest leader emerges. This lead to a foundation for several solu-

[23]Often this bound differs between single nodes as different hardware endowment is assumed.

[24][20]

tion approaches and is utilized as a base for the pBFT setup. This approach assumes two circumstances; a delivery bound δ like the synchronous model and the Global Stabilization Time (GST)[25], which means that the desired (deterministic) state will be achieved eventually, and every message sent at t will be delivered by the time $max\{t, GST\} + \delta$. The pBFT works in a round-by-round tune. It takes more than one round to reach consensus on a proposal. One round consists of two phases; where in the first phase the validators vote on the proposal from the current leader, and a second phase where the certified proposal of the previous round is voted on again to be committed. A proposal is declared certified, if at least $2f+1$; a quorum validators vote on it, and therefore, generate a quorum certificate (QC)[8]. The pBFT model pushed the development of new consensus protocols, as pBFT enabled performance in replicated environments similar to the performance of non-replicated ones[23]. The weakness of pBFT is the growing communication overhead, increasing in the number of participants, that leads to a trade-off in advanced models between responsiveness and linearity in communication costs. Responsiveness indicates that a leader could propose a transaction and therefore a new potential state without awaiting a fixed time delay, which would speed up the system. Since the Libra Association intends to include up to 1'000 validators, the basic pBFT approach does not suffice for the current requirements[11].

This trade-off between costs and speed of progress was resolved with the HotStuff protocol, meaning that it allows linearity in communication costs with guaranteed responsiveness from leaders. The HotStuff protocol works in a three phase tune per round; the additional phase enables the simultaneous presence of responsiveness and linearity[8].

In the HotStuff protocol the proposal of a block from round $k-1$ becomes certified in round k, becomes certified again in round $k+1$ (QC-QC) and committed (finalized) in the third round, $k+2$ (QC-QC-QC). After finalizing a block, no further operations, or reversals can be executed on the transactions[24]. This way, a chain of blocks with transactions is created. It is important to mention again that, these blocks in fact reference their parent block, but this is solely for the purpose of consensus-finding and identifying an earlier, commitable block. To keep the chain (potential branching tree) of blocks in a uniform order, the LBFT protocol, deviating from HotStuff, demands the next leader to continue on the highest certified block (leaf) with a direct descendant. This procedure, leading to a uniform tree leads in turn to the possibility of gaps in the

[25]Pioneered by Edsger W. [22]

round-commit order. To guarantee determinism in transaction execution, all validators must be in synchrony of states and rounds at the point of voting and consensus finding. Without progress, regular BFT solutions just double the duration of a round until progress is observed the HotStuff protocol and LibraBFT chose another path. A round without a QC will eventually lead to an alteration and a new round by timeout. This could be caused by a crashed leader who failed to propose a block and therefore received timeout messages (summing up to a timeout certificate TC) by the other validators. A TC works as an order to skip the next round in expectation of a valid leader; said TCs can therefore act as a synchronization mechanism for the rounds of the validators, where validators broadcasting a timeout message on a round k with a certificate for round k eventually summon all validators to a common (synchronous) round. This mechanism of synchronization does not sacrifice linearity in communication overhead, described in the first version of the HotStuff research paper [25]. According to the LibraBFT team the current version of HotStuff uses a functionality called PaceMaker, without specifying its implementation, to advance rounds. The Libra BFT implements a module with exact same name to govern the collection of messages and the advancement of rounds [8]. [26] Inconsistencies within the blockchain, so called forks where two or more parallel chains exist, are prevented by construction in the Libra blockchain through the introduction of a preferred round, which acts as hydraulic brakes for the construction of several ongoing forks, as the latest committed block in a three-chain of transaction blocks leads automatically to the abandonment of the parallel chain. This means that there will never be a greater-than-two blocks parallel-chain to the main chain. In the case of malfunctions and multiple resulting versions of the blockchain, the Council has the right to resolve this branching [6]. This process is not further specified but could be conducted as a selective reversion of states.

After this brief introduction on the wide field of BFT consensus protocols, the implementation of the Libra BFT protocol is easier to understand and further demonstrates the abilities of the modules introduced previously. Since we are in the world of replicated state machines, we must be aware that any operation by a module is executed on every validators' replica in the system.

[26] Equal names might be of coincidence, but nonetheless it is interesting that a scientist involved in the development of HotStuff, is currently working (as Lead Researcher) at Calibra.

The following consensus cycle is an infinite loop performed by the Main module that handles the events and messages of the consensus protocol and between validators. At the beginning, clients send signed transaction messages to validators; these messages undergo initial checks by the admission control element of the receiving validator. This early check rejects transactions with syntactically wrong code and too small gas balances, thereby keeping them from entering the mempool (similar to the checks of the prologue explained earlier but with additional syntax screening). Accepted transactions are shared between the validators through the mempool[11], which further checks if a clients who sent multiple transactions to validators possess sufficient funds to pay for their gas all of her transactions in the mempool will consume. Then a leader batches a subset of the transactions from the mempool into a block and proposes this block to its peers. In addition to a payload, which contains transactions to vote on, a block contains crucial information for the consensus protocol. Id, round, timestampt and signature identify the block and the proposer of the block. Additionally, a block contains (if available) the QC of its parent to append to the chain of blocks and commit an earlier block. Validators receiving the proposal in round k initially process the proposal. First, the validators check for the included QC or TC of round $k-1$ to advance to the round following the one of the certificate (synchronization of all peers to k), increase their preferred round to the one of the parent of QC (round $k-2$) and commit the speculative state of the grandparent of QC (round $k-4$)[27]. A QC on k, in turn, would lead to a three chain of QCs for the proposed block of round $k-3$ and therefore to its commit in round $k+1$. The second step of processing a proposal handles the voting, which is explained later in this thesis and the third step is a speculative execution of round $k-3$ in case the round k proposal receives $2f+1$ or more votes. The speculative execution of a committable state is carried out by the Ledger module. By speculatively executing blocks, it forms a potentially branching ledger tree that is kept locally at the validators' storage. After one branch becomes committed, speculated and no longer committable parallel branches are erased. The speculative execution of blocks is of high importance for a fixed trust protocol, as a single event upset could lead a validator to diverge from its peers and vote on a different result for the same state without the LBFT protocol taking note of it. The management of the not yet committed blocks and their votes is performed by the

[27]With the proposal from leader $k-1$ the validators advanced to round k. The proposal of leader $k-1$ contained a QC-QC-QC for round $k-4$ that is committed in round k

Block-tree Module, which builds a speculative tree of blocks 1:1, mapping the position of a block in the Ledger module. Voting on the proposal of round k is performed by broadcasting a vote message to the other peers.
A signed vote message contains the following inputs:

1. VoteInfo
   - Block id
   - Execution_state_id
   - Id and rounds of parent
2. LedgerCommitInfo
   - Commit_state_id for clients Hash(Root Ledger History)
   - Hash(VoteInfo)
3. Call Safety.commit_rule

Block id and execution_state_id guarantee determinism of block execution; the id and round number of the parent enable deduction of a commit from a single block. In case of a QC where the grandparent of the QC-emitting block becomes committed, Commit_state_id in the LedgerCommitInfo certifies this new ledger state, and its hash additionally serves clients as proof of history. The Hash(VoteInfo) serves as additional authentication; therefore, a validator sending a signed vote message authenticates its vote and the speculative LedgerInfo. The invoked Safety module in a signed voting message checks whether the vote could commit an earlier block of transactions. However, before allowing a validator to vote, the Safety module ensures that the validator did not vote before in the current round; safety for the voting process as a whole can be guaranteed by controlling for the following:
$round\ of\ proposal > last\ voted\ round\ of\ validator$
$round\ parent\ of\ proposal \geq preferred\ round.$
The latter inequation is important to allow the commit of an earlier block under the assumption of safety, which means that at least $2f+1$ validators voted on the committed block. A message collecting module, called Pacemaker, advances rounds in the consensus protocol with regard to votes and time. If a validator does not receive a proposal

message from the round leader, then Pacemaker broadcasts a timeout message and simultaneously collects timeout messages from its peers; the same occurs for votes when a block is proposed. After collecting a TC or QC, Pacemaker advances its validator's round to the round of the certificate and sends the certificate to the round leader. This approach guarantees that with the emergence of an honest leader all validators synchronize their rounds within a maximum of two delays Δ. A reminder is warranted: under assumption of partial synchrony, the delay of message transmission is not known and might vary between validators. We therefore know that a validator will eventually advance to the current round at 2Δ, but the time passed in seconds remains individual; otherwise, we would operate in the world of pure synchrony. The following page provides a graphic illustration of the concepts above to visualize one cycle of consensus-finding and communication.

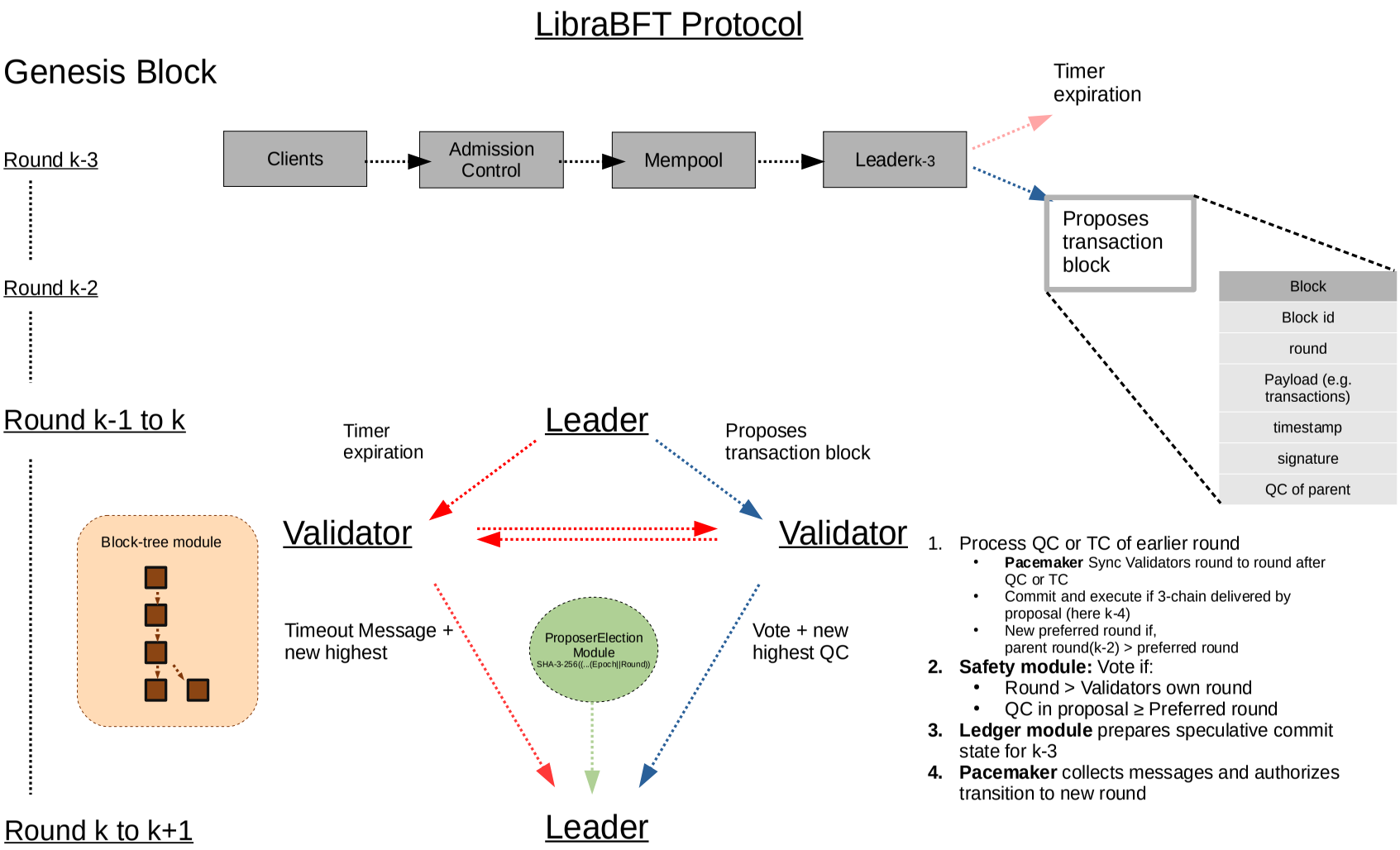

Figure 8: LibraBFT Protocol, author's interpretation
Illustration of a full consensus cycle.

After an introduction to the consensus protocol LBFT, we know that the decision-making process does not focus on a single or even a batch of transactions per se[28]. Rather, it focuses on keeping the replicated database and therefore the state machine as a whole in consistency through strong determinism; by doing so, it leverages decades of game theoretical and computer scientific research. The LBFT takes the "lion's share" in the higher transaction throughput toward permissionless consensus algorithms. However, under assumption of fault tolerance (P) and with high priority to data integrity and consistency (C) according to the CAP theorem from time to time a trade-off may occur [26]. In this situation the trade-off usually leads to the abandonment of availability (A) in pBFT. The reasoning behind ($C \succ A$) is not quantifiable and therefore deferred to the economic part of this thesis. To further minimize the occasional abandonment of A (under the assumption of C and P) the Libra Association formulated minimal requirements toward hardware. Validators must fulfill requirements regarding bandwidth, disk space and CPU performance to reach their current goal of 1'000 transactions per second (tps) [11] without abandoning C and P for the sake of throughput. Libra can, moreover must scale up to more than 1'000 tps to meet the conditions of a platform offering more than just payment transactions. To acquire an intuition about the dimensions, consider that; the well-known payment service PayPal executes approximately 400 transactions per second[29] [27]. Therefore, 1'000 tps are rather state of the art than a novelty in permissioned systems; the Hyperledger Fabric blockchain, at the time of this writing, aims toward 20'000 tps[28]. Even in the world of permissioned BFT consensus protocols exist several di- and trilemma situations. The flexible trust implementation of the Hyperledger Fabric blockchain for example forgoes the strong determinism conditions of Libra [29], and by doing so, lays a central dependency of the integrity of the blockchain on its (transaction) ordering service. We see that several decisions in their technical implementation influence, for example, the performance and can therefore say that Libra is built with focus a on the integrity of its replicated database rather than speed at any price.

[28]This also would of course still be possible by setting the mempool or a step earlier the admission control up to ignore specific addresses.

[29]Calculation based on 3'090mio transactions in the third quarter of 2019

Upon this technical introduction to Move, the transactions, the storage and the consensus protocol, we may draw an initial conclusion. We can say that compared to permissionless blockchains, the Libra replicated database's performance outmatches its competitors concerning lower energy consumption and higher transaction throughput. Written in Move, the Libra protocol is also designed to be dynamic and therefore comparatively easy scalable. Libra's data structure calls the implementation of earlier, functioning projects, to mind as the white paper announced and its database engine, developed by an Association member, already existed in a tailored manner. Finally, the consensus protocol, based on HotStuff, is a new, and particular to the field of crypto economics designed consensus protocol[25]. However, it misses a working pseudo-randomness algorithm for leader election, which in a future, permissionless version of the protocol, must be resolved. In comparison to the permissioned competitors, Libra plays the trump with Move, with its modules and high-end security standards prioritized and engraved into the protocol. Libra's currently targeted transaction performance, however, is at best mediocre and as we know, declines in the number of validators. Nonetheless- and especially because of- its involved leading scientists, strong endowment and, sophisticated research engagement, the Libra project can be a serious new player from a technical perspective.

4 Economic Analysis

Building upon the insights of the previous sections; this section investigates the potentially occurring impact the introduction of Libra takes on the world economy, and the reachability of Libra's announced goals.
The first portion focuses on the transition to permissionless consensus, a key requirement to expand the adoption of Libra. Followed by brief, economic analysis of Africa, considering the necessary infrastructure, that evaluates the chances to reach the goal of providing banking access to billions of people. The final portion, an inspection of the potential impact on monetary policy operations by Libra, utilizes the IS-LM model to build the foundation for the discussion on the potential outcome scenarios.

4.1 Transition to Permissionless Consensus: Doubts and Issues

The Libra Association plans to switch to a permissionless blockchain within five years of introducing Libra. The new form of governance and consensus finding will rely on the staking of the Libra Investment Tokens, and in its final form, additionally on the staking of Libra coins[30], and thereby distribute the power of decision-making across the whole community. Even the Libra Reserve is announced to be automated in maintaining stability for the Libra coin[30]. This announcement poses considerable disbelief within the blockchain community and even technologically uninformed people for two main reasons. From an economic perspective, a permissionless blockchain would, as already mentioned, decrease the decision power of the granted validators and distribute it over anyone in the system willing to participate in governance and consensus. This does not seem rational, since most initial members are profit maximizing entities. The second concern regards the design of the implementation. A stake-based consensus algorithm has not yet been demonstrated to work without flaws and create its own challenges. The Libra Association solely expressed its intention and charitable reasons rather than a well-defined roadmap and technical key concepts for the transition; therefore, this section contains a certain, inevitable degree of speculation.
The economic reason behind a transition into a permissionless system may be twofold.

[30]With the pointer on "Quant Funds" especially the Medallion Fund, automated Reserve management is assumed to be no ongoing issue. `https://www.bloomberg.com/news/articles/2019-11-12/the-unsolved-mystery-of-the-medallion-fund-s-success`

Members may have a headstart and accumulate large amounts of Libra (fees and dividends) to, in effect, maintain their decision power. If we assume the composition of Libra from an earlier section to be true then, the interest rate on the Reserve's assets will be $Dividend\ Yield \approx 0.4 - 0.7\ \%$[31]. This rather low dividend yield will further be utilized to keep transaction fees low [2]. The hypothesis of a headstart by accumulation of Libra can therefore be eliminated. When the system and therefore the Reserve grows, the Libra Investment Token holders will have sole claim to the leftover interest of the Reserve [3]. In a future environment of potentially higher interest rates, a permissionless blockchain will attract additional users and therefore be more profitable for the holders of the Investment Tokens. This can be understood as a reason for opening the ecosystem to everyone.

The second possible reason behind the transition to permissionless consensus and governance is slightly more speculative. As the Libra ecosystem grows in size, third-party modules become deployable and a variety of services become available, the Libra ecosystem will become increasingly intractable. A well-defined set of validators, as initially defined, can be held liable. In a more distributed environment of decision-making where adversary or even illegal behavior is possible it will be difficult to determine who to hold responsible in the first place. An intended shift of responsibilities is therefore imaginable as reason for the transition to permissionless consensus. Contradicting this thought experiment is Calibra's announcement to introduce strong customer verification standards and refund customers who lost Libra through malicious actors.[31] The economic reason for the transformation to a permissionless governance in consensus therefore is trust in the growth of the system rather than losing control over it.

Technically; the consensus algorithm based on the proof of stake (PoS), is in comparison to the LBFT, a totally decentralized consensus algorithm. Without watchmen where everyone knows each other, another incentive setup to cooperate in line with the protocol is inevitable. The PoS is based on a pseudo-random process, which depends on a combination of factors, to select a new leader for a block proposal. One factor is the stake a validator is willing to lock (through a special transaction message) and therefore forgoes the usage of this stake for a predefined time. A validator can increases its chance

[31]Source: `https://www.investing.com/` Assumption: 50% currencies and 50% bonds. The fraction of European bonds was once calculated solely in German bonds and once solely Italian(BBB) bonds resulting in a rounded discrepancy of 0.3 percentage points.

to become elected by staking a higher amount of Libra coin.
With this protocol design, two major issues arose within the development of the PoS consensus algorithm: the "Nothing-at-Stake" problem where in the event of a fork in the blockchain, the dominant strategy for validators is to stake on every existing chain because otherwise everyone else doing so would have a higher probability of being elected. This strategy is dominant to participants because it increases the chance (or at least does not decrease it) to become a leader without further costs; however, it potentially destroys the blockchain's integrity. This topic is currently researched by implementing a "slasher" algorithm that punishes a validator by removing, up to the total amount of the stake a validator locked up, in the event of wrong behavior [32]. The second issue is called a "Long-Range-Attack". As most potential block proposers do not lock their stake for an infinite amount of time, they are no longer punishable after unlocking their stake. This, and the fact that creating blocks in the PoS protocol is cheap led to the threat of a Long-Range-Attack where an adversary can build an alternative chain that better fits the adversary's preferences[33]. A new client or one who is not up to date in the latest version of the blockchain may fall for an adversary and the wrong chain by accepting it as a leader and the alternative history of the chain. The unofficial term for this situation is weak subjectivity[34], as clients depend on an external source of information they can synchronize with. The information could possibly be wrong, meaning that it may not be the longest, active chain. The term subjectivity regards the possibility of multiple chains being acknowledged by different clients (a subjective view on reality), and weak because this applies to only a few situations for new or long-time offline clients. This could lead to destructive consequences for victims of an attack and the reputation of the system as a whole. To prevent this, users must update their view on the blockchain in a sufficiently short tune, and their first information regarding the state of the blockchain must be reliable. For the sake of simplicity, further details will be forgone as they would reach beyond the scope of this thesis.
An existing approach to face the issues of the PoS consensus algorithm is called Casper, a protocol that was engineered by Vitalik Buterin and Vlad Zamfir who each introduce slightly different approaches[32]. Casper aims to resolve the issues and is a key instrument on the roadmap to PoS, which has been a topic for years in the Ethereum blockchain.

[32]A multi episode introduction to the topic is authored by Vlad Zamfir himself. `https://medium.com/@Vlad_Zamfir/the-history-of-casper-part-1-59233819c9a9`

The Casper protocol has significant potential to resolve major issues but is not eligible for the Libra blockchain. According to [32] on GitHub, Casper tends to ($PA \succ PC$) and therefore does not fit into the ($PC \succ PA$) mentality of LibraBFT. This assumption, in addition to the technical insights, incorporates the non-quantifiable possibility that users may be wary toward specific members of the Libra Association, and a data scandal where provably C was violated must be prevented at all cost, at any time starting at the first genesis state. It is therefore necessary to find another solution for BFT protocols with strong demand for consistency. An approach for BFT protocols is Tendermint. For simplicity, we will just refer to Tendermint as a consensus protocol. Tendermint is a replicated state dependent algorithm with strong determinism[35]. Engineered by Jae Kwon, it provides solution approaches beyond the permissioned versions of BFT protocols applicable to the PoS issues. The developers from Calibra are aware of the existence of Tendermint and explicitly mention Tendermint as not fitting because it does not incorporate responsiveness with a simultaneously linear communication overhead. It is, in fact true, that Tendermint incorporates one round less than HotStuff in the voting process, which is said to be the reason for HotStuff resolving this problem [35][11]. A hybrid model with Tendermint's PoS approach and HotStuff's resolved trade-offs could be in the range of upcoming new protocols. The remaining problems of weak subjectivity and Long-Range-Attacks could be mitigated in a surprisingly simple way. As Calibra stated, its wallet will be available in the application WhatsApp[31]. This, one of the most common applications in the Western world, would decrease the number of clients offline for a significant amount of time and could possibly serve as a reliable provider of current network state (oracle) for new users. While this idea, properly implemented, will most certainly work, it does come with a major drawback: a full circle back to centralization on a few or even worse, a single entity. Additionally, a protocol relying on its users instead of a proper incentive and disincentive scheme to stay consistent is not desirable.
This thesis will not, and does not aim to, revolutionize the world of game theory, computer science and blockchains. Left to say is: the PoS, while credibly promising a variety of improvements on the proof of work (energy efficiency and network safety) and the BFT (decentralization), is still an actively researched and not yet fully explored intersection of many sciences, and the Libra Association must provide insights into the development process to remain credible.

4.2 Development Ground and Banking Access

Tying to the above insights, this section investigates Libra's capability of creating the development- and financial infrastructures the Association aims to provide. Providing financial services to people who not yet have access to bank accounts presents multiple challenges. One challenge is the infrastructure itself. The Libra Association plans to "bank the unbanked" in a digital manner that needs a connection to the internet. The network expansion in for example Africa could take years if not decades to reach a continental accessibility. Whilst restricted to the development of the infrastructure in the targeted countries is there a way more immediate field of application for Libra. It will create a platform and development ground for developers and businesses.

Development Ground

The Libra blockchain and Move are designed to promote the development and implementation of business logic[1]. Modules and resources are the key factors of creating and deploying a so called DApps on the blockchain. The concept of DApps is in fact not a novelty, an ever growing number of projects is in the phase of development and many already "on-chain". *Background on DApps*

Bitcoin introduced the possibility of decentralized exchange of a valuable digital asset but lacked extensive further application. This lead Vitalik Buterin to the vision of an expanded decentralized ecosystem where people are able to create own projects and the birth of Ethereum[36]. Today is Ethereum the decentralized center of the creation of smart contracts and DApps. A DApp can be imagined as the frontend, like a homepage with a GUI. The difference to a regular homepage is that a DApp uses a smart contract instead of an application programming interface (API) and connects to a blockchain as the backend. A DApp connecting to the Ethereum blockchain is therefore decentralized by construction. This makes DApps very robust against mutations from anyone else than its creator. Several DApps running on the Ethereum blockchain reached high popularity, very well known are Decentraland and CryptoKitties. The latter even lead to jams on the blockchain through its high demand. Whilst CryptoKitties and Decentraland indeed incorporate monetary incentives and payment possibilities are there already DApps specifically designed to provide financial services.

These DApps are capable to disrupt the conventional financial sector in certain fields because their whole design is (usually) decentralized and therefore can do without banks

as crucial intermediaries. The circumstance of truly decentralized applications is in fact not the case for the Libra blockchain. At least under its permissioned protocol and for a variety of reasons. As mentioned earlier will the Libra Association introduce a strict identification process known as "know your customer" (KYC) process. Additionally will the Council and validator nodes mandate everything on the blockchain from governance to consensus.

Even though or especially because the Libra blockchain is an initially centralized system does it come with several benefits. The aforementioned higher performance might make Libra to a potentially attractive platform for users not seeking pseudo-anonymity. In a matured state with a variety of deployed Apps could Libra's development follow several possible paths. Two of which are especially interesting as each one follows the complete opposite direction. Path one would require full governmental approval and eventually the transition into a permissionless ecosystem. In this form could Libra enter any imaginable field that incorporates technology, reach out to a very high userbase and disrupt a variety of industries. Why path one is rather not probable will be explained in the next section. In path two will Libra stay permissioned and could become a platform of the range of Tencent's WeChat. WeChat is a Chinese application with more than one billion active users per month. For its variety of services WeChat is often called a "super App" and roughly spoken the very opposite of a permissionless ecosystem. The Chinese government is contractually authorized to query any arbitrary information from WeChat and strictly monitors its userbase. Whilst Libra might not be as governmental monitored as WeChat could it possibly gain a similar size userbase. Expanding the idea behind path two could Libra offer any service a developer wishes to realize and itself grow into a super App. The internal development ground of WeChat is called "Mini Programs" and allows users create smaller applications like shops and games using the internal payment mechanism WeChat Pay[37]. As we see could Libra follow WeChat, experience a very similar implementations and transform into a super App rather than a real permissionless blockchain. The mentioned paths are as different as they could be from a blockchain perspective.

However do both paths have something in common, Libra will inherit a potential userbase of billions of people and does aim even before its launch to attract more users.

Banking Access

As already mentioned is the possibility of providing banking access to the unbanked people of the world currently limited. The lack of the vital internet infrastructure poses a physical restriction on the Association's ambitious project. This analysis of banking access for unbanked people will solely focus on Africa. The reason behind the decision of focusing on Africa comes from the potential ban of Libra in India[38] and the African Economic Outlook (AEO) of 2019[39].

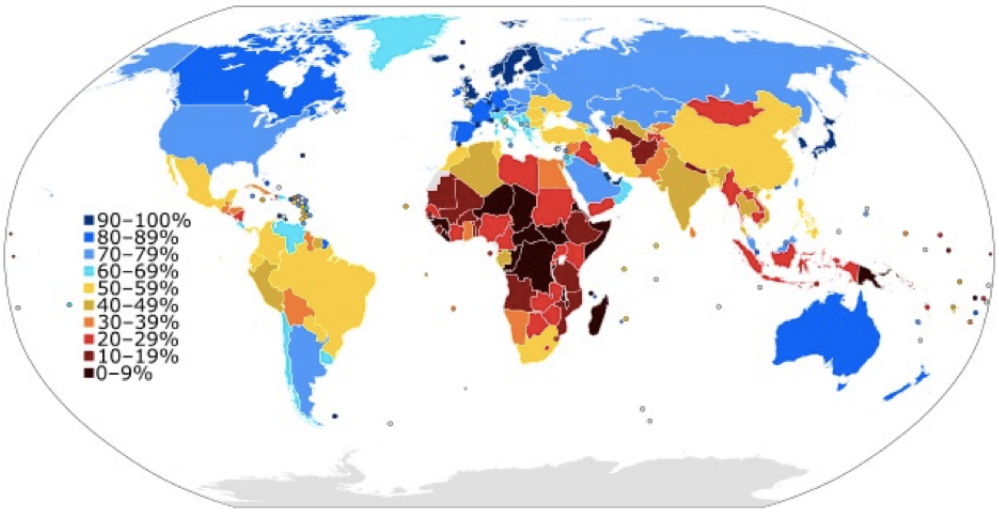

Figure 9: Percentage of internet users per country in 2015, source: Medium[40]
Developing countries show a significant deficit in internet coverage towards industrialized nations.

If one assumes the access to financial services and internet access to be approximately equally high all over the world, then the above graphic shows the main reason for the impossibility of immediate banking service for underbanked and unbanked people. This restriction is explicitly mentioned in the AEO 2019[39]. Electricity and financial services are said to be the biggest obstacles for doing business in Africa. The AEO discloses the severity of the backlog of Africa's infrastructure and overall economic condition. The backlog of Africa's overall economy in turn provides the opportunity of growth. Sub Saharan Africa is expected to grow at a rate of approximately 4%[41] and according to CNN did the number of mobile internet connections across Africa double within the two years preceding 2019[42]. It is not only Libra Association member Facebook that was launching projects to strengthen the African internet infrastructure. In fact do the

biggest and most influential technology corporation in the world invest in Africa[43]. With a growing technological infrastructure, powered by big corporations, will Africa's demand for financial services rise. The rise of demand for financial services in turn could be met by Libra. With an own programming language that allows the development of projects that could especially target financial services does the design of Libra seem to be "custom made" to bank the unbanked. Whilst this thesis aims to analyze the capability of the Libra Association to fulfil its goals does it go beyond the scope to judge whether these goals are set by pure philanthropic motivation or any other possible reason.

4.3 Central Banks and Monetary Policy

With increasing size, more services available and higher adoption, Libra will attract more users and widen its overall acceptance. Berentsen[44] elaborated on the possibility of abandoning central bank currencies in favor of digital money from a game theoretical perspective, arguing that with higher conveniences for users (lower transaction costs, faster value transfers) and under the assumption of wide acceptance (>50%), market participants gradually adopt the digital money. This scenario would lead to a situation where the Libra Reserve will eventually no longer be able to back Libra coin as announced (1:1) with a basket of currencies. Libra would, in effect become a digital fiat currency, the Libra Association a central bank and the existence as currency board abandoned. Although this scenario is possible in theory, this thesis will not further elaborate upon it, mainly because of the strong supervision of regulatory institutions and politics that already focus on the "central bank threat" of Libra[33]. Nonetheless, there is an often overlooked factor between a total ban of Libra and the transformation to a fiat currency: Libra can decrease the impact of short run monetary policy.
To create an intuition for this statement we will sonsider the simple IS-LM model. This model is often said to be outdated and not representative of the complexity of a modern economy. While this argument is primarily true, the IS-LM model still serves its purpose of demonstration.

[33]According to the Swiss minister of finance Ueli Maurer did Libra not become approved to operate. https://www.nzz.ch/schweiz/bundesrat-maurer-libra-hat-derzeit-keine-chance-\auf-bewilligung-ld.1530913

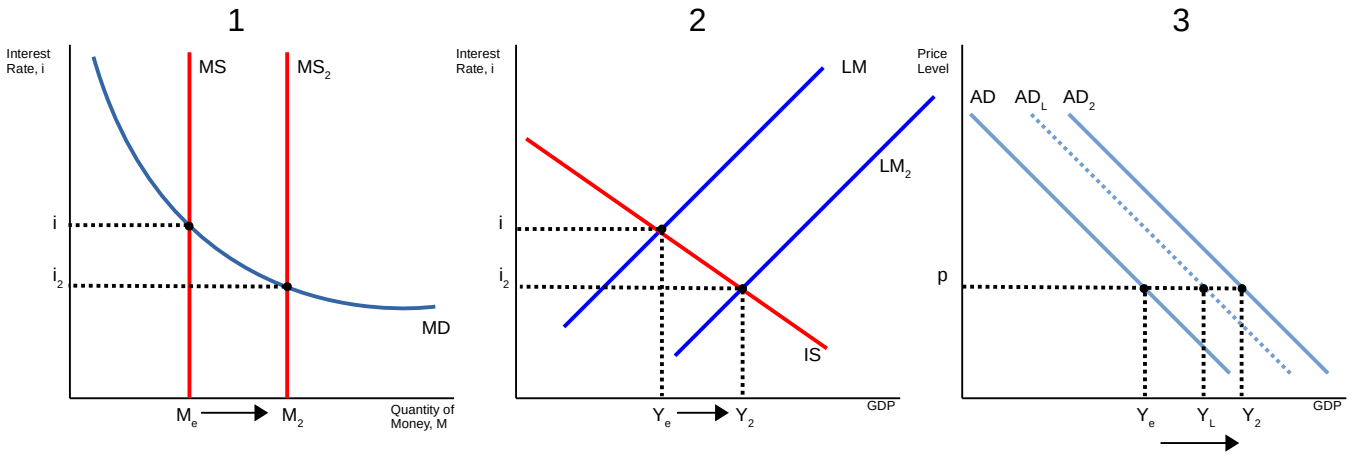

Figure 10: 1. Money Demand and Money Supply, 2. IS-LM Model, 3. Aggregate Demand.
Illustration in the style of Mankiw and Taylor[45]
Graphic illustration of the short run impact of unexpected expansionary monetary policy. 1. MS = Money Supply, MD = Money Demand, i = interest rate, 2. LM = LM-Curve, IS = IS-Curve, 3. AD = Aggregate Demand, AD_L = Aggregate Demand with Libra in economy, p = Price Level.

This graphic illustration depicts a situation where a central bank (e.g, the US-Federal Reserve) injects liquidity into a market to prevent a stagnation of the economy. In an open market operation, the central bank buys treasury bills from banks. This leads to an increase in the quantity of money (MS $\rightarrow$ MS_2) and a decrease in the interest rate i. An increase in the quantity of money leads to a downward shift of the LM-Curve (LM $\rightarrow$ LM_2). The shift in the LM-Curve, and moreover, the increased money supply and lowered interest rate lead to a shift of aggregate demand (AD $\rightarrow$ AD_2) and therefore a higher GDP = Y. Since this is a short run, unexpected monetary expansion, the price level will remain at the initial level. To understand why a decrease in the interest rate will increase the GDP in the short run, one must consider the following:

$$Y \equiv C + I + G + NX. \tag{1}$$

In equation (1) we defined the GDP as a function of consumption (C), investments (I), governmental spending (G) and net exports (NX).
Converting I into an endogenous variable by introducing i (interest rate) gives us the

following properties:

$$I(i), \frac{dI}{di} < 0 \tag{2}$$

$$I(i) = k_0 - k_1 i. \tag{3}$$

Equation (2) specifies I(i) as a decreasing function with respect to i. The intuition behind (2) is that if the overall interest rates increase in the economy, then investments will become more expensive as a higher interest rate has to be paid for the investment. Equation (3) defines investment function I(i) as a function consisting of a constant, k_0 and a variable part $k_1 i$.
Updating (1) will yield the following:

$$Y = C + (k_0 - k_1 i) + G + NX \tag{4}$$

A differentiation of (4) with respect to i will yield:

$$\frac{\partial Y}{\partial i} = \frac{dI}{di} = -k_1 < 0 \tag{5}$$

We now see that the investments I of an economy as a whole decrease (increase) with an increase (decrease) of the interest rate i, as graphically illustrated above.
If we now imagine that Libra was introduced into the US economy and that people hold a mix of US dollars and Libra (e.g. 50%), then the effect of an expansionary open market operation will decrease. People who hold large amounts or exclusively US dollars will adapt to the new situation. People holding deposits in Libra coin will be affected neither explicitly by the increased interest rate nor implicitly by the backing of the Reserve, as holders of Libra coin do not receive interest. This scenario leads us to the equilibrium GDP of $Y_L < Y_2$. In a situation of a contractionary monetary policy operation, this effect will become even more severe. Contractionary monetary policy will absorb liquidity from the economy, increase i and decrease overall investments made in US dollars. People holding Libra will again be unaffected and continue demanding and supplying services on the Libra platform, leading to a monetary policy induced advantage for Libra.

A few things are left to say. Although interest rates in the Euro area, Japan and Great Britain are low, does that not lessen the generality of the above example. The IS-LM model is a simplistic representation of the economy, and by the neutrality of money, the effects of monetary policy will last only a few months. Nonetheless, we see an possible impact on the economy as a whole after the introduction of Libra. It would be an unfavorable situation when an entity can design itself to be resistant to monetary policy. Such resistance could, in turn, lead to a significant market advantage and a serious threat to central bank money. This does not only apply to countries represented in the basket of the Libra Reserve. Especially smaller economies with an unstable monetary policy could be affected by the constrains on their monetary policy operations. We can therefore say that governmental suspicion toward the Libra Association is justified as it represents one of the most serious disruptive influences today.
Libra is able to challenge monetary policies and central banks.

4.4 Outcome Scenarios

With the above insights it is possible to formulate potential outcomes of the project Libra. It is of course impossible to determine any outcome with certainty or further describe their variety. A total ban and fail of the Libra project is unlikely, since the whole project could be scaled down and the sophisticated preparations could be integrated into a lower level project. In any possible outcome scenario is the initial governance structure of the Libra Association and the Council expected to be similar, because the governance structure of Libra does not pose a threat to authorities.

No governmental approval

In this scenario will the Libra Reserve not be approved to operate and the move to permissionless consensus fails. The Reserve will not be able to accumulate the backing for the Libra coin. The governmental structure and maintenance of the network will therefore not be financed by the yielded interests on the Reserve. This will result in overall higher transaction fees and the Libra Investment Token loses its incentive for the members to expand the ecosystem and generate more interest through higher Reserve. The Libra ecosystem will rely on the internal development platform. If any, transactions will be done via conventional channels like payment apps or credit cards. Under these conditions will Libra provide a central governed development ground and created potential

financial applications will be implemented using the fiat currency of the local government. Monetary policy's influence in operations of centrals banks will not decrease. The disruptive energy and pressure for competitors and officials is not existent to low.

Negotiations and partial governmental approval

This scenario leaves enormous space for speculation. Partial governmental approval could lead to a limitation of the Libra Reserve to one single currency. Further could the Reserve be obliged to hold only certain, native securities or full fiat backing. The Association members and their main operating businesses could be subjected to strict conditions and reporting requirements[46]. With governmental monitoring and higher reporting standards will the move to permissionless governance and consensus fail as well as in the scenario above. The Libra Investment Token will remain with a certain incentive structure and potential interest on the Reserve could keep overall fees low. The development ground will have more favourable conditions as executing Apps will be less expensive and business logic could be implemented. Libra might develop into a platform of considerable size with internal payment structures and a variety of available services. By construction might Libra be forced to split into locally hosted platforms and strictly separated Reserves leading to multiple versions of Libra coin. Central banks' operations will, depending on the negotiated design, suffer limited to no loss of influence. The disruptive energy and pressure on governments is low to at best mediocre but signals the high demand for a faster and simplified international exchange of money. However is the pressure on the private financial sector low to high. New, more convenient, innovative and cheap services could be offered with governmental approval and increase competitive pressure towards the financial sector.

Full governmental approval

By far the most interesting, unlikely and most speculative scenario was already decelerated[47][48]. For the sake of comprehensiveness and to understand the potential reason behind the political decision will the third scenario be illustrated, regardless of its current probability of occurrence. Scenario three has two possible outcomes as it could lead to a new era of decentralized economic competition or a future of centralization on a few or one association.

Full governmental approval would imply that the Libra Reserve will start its duties as proposed with the creation of the first genesis state. The Libra Reserve will accumulate initial investments and various currencies to exchange them until it meets the predefined

stability conditions. The generated interest on the Reserve's assets might cover the maintenance costs, suffice to further development costs and keep the fees as low as intended. Leftover interest distributed to Investment Token holders will serve a strong incentivizing effect on the members to expand the ecosystem as the marginal revenue on additional interest will probably be higher than the marginal costs of maintenance. Invested interest will scale the network throughput upwards and strengthen the development ground and financial infrastructure. Developers and businesses of any size could create applications with high international outreach without concerning foreign exchange rates. People all over the world could consume the services provided and benefit from fast and cheap international transfer of Libra coin. The transfer of Libra will be cheap and fast because it relies solely on one intermediary. During all these development steps will the implications of monetary and fiscal policy operations decrease as the usage and influence of the local currency decreases. At this point will the paths of scenario three split. The first path leads to a transition to a permissionless governance and consensus and the intermediary will dissolve into many single governance and consensus decision makers. This in turn will lead to a rise in competition all over the world and a decrease of governmental and financial sector influence on the economy. The second path is where the transition to permissionless governance and consensus fails. Whilst the influence of financial institutions and governments on the economy decreases will the Libra Association remains in power. The disruptive energy in both paths is enormous towards governments and the financial sector. Whilst scenario three illustrates an absolute extreme scenario does it good in displaying the impact a multi currency backed, private, financial infrastructure poses. Governmental concerns towards the Libra Association and the Reserve with regard to the free market economy and (monetary) political instruments might in fact be the reason for not approving Libra in the way it was initially proposed.

Subsuming the economic section of this thesis can be said that the transition to a permissionless governance and consensus protocol depends on two factors; the willingness of the Association and the current state of research and development. Many projects on different blockchains and different consensus styles are working towards the goal of stake based consensus algorithms. And not any arbitrary protocol fits the demands of Libra. Both mentioned factors must be fulfilled to transform the ecosystem into a permissionless platform. The development ground and banking access for the people depend on the acceptance of Move and the development of basic infrastructure in targeted regions. Move

will be used and improved in the progress of developing (business) applications but needs to meet the requirements of safety for businesses and users. The lack of infrastructure in regions of people without access to financial services poses a more immediate problem. With development projects do technology corporations worldwide aim to improve the situation. Whilst the the development of infrastructure can not be timely determined can one say that financial services will be offered to any new internet user of the future. With more convenient services and better infrastructure will the userbase of Libra and the Reserve grow. With the simplistic IS-LM model was shown that a growing Reserve will eventually lead to diminishing impact of monetary policy operations. The resistance of Libra against monetary policy led eventually to a variety of possible outcome scenarios. The disruptive energy of the Libra Association is depended on the design and governmental approval of the Libra Reserve. The third outcome scenario depicted an extreme situation is can be assumed as the reason for the latest political decisions regarding the launch of Libra.

5 Conclusion

This thesis demonstrated the potential economic impact the introduction of Libra may have on the world economy by examining the Libra ecosystem in a fairly weighted manner. The Libra Association Council is the single decision making authority of the Libra governance and allowed to overrule any former set of rules. Its restrictions toward participation in the project allow only a specific range of entities to join the project and the decision power in the Council depends on the financial endowment invested in the Libra Investment Tokens. The Libra Reserve, introduced as the passive, stability providing element revealed its potential influence throughout this thesis. The technical Analysis of the Libra ecosystem demonstrated high development effort and the inclusion of scientists that were involved in the development of underlying elements like the HotStuff protocol. Libra's technical network as a whole is designed to guarantee consistency and safety for users, guaranteed by the implementation and further development of safety enhancing elements. The programming language Move allows users to create digital assets and smart contracts called modules with special regard to safety. The examination of Libra's replicated database revealed a similar composition to the one of the permissionless blockchain Ethereum and makes also use of Merkle trees for the purposes of storing and authenticating data. The consensus protocol of Libra, the in-house LibraBFT targets fault tolerance and consistency in the first place, here, contrast to the data structure, takes the LibraBFT protocol distance from Ethereum with regard to overall decentralization of decision-making. The economic section investigated the Association's plan of the transition to a permissionless consensus protocol in consideration of technical restrictions and incentives for the Libra Association toward the decision and outlined the technical challenges to the transition plan. With regard to the design of Libra coin, the network structure and the development ground as application generating component was Libra's ability to provide banking access for billions of people demonstrated, but at the same time remains dependent on the structural development of the mentioned areas. The analysis of the potential, disruptive influence on central banks lead to the the insight of the resistance of Libra toward monetary policy operations. The last section presented a selection of potential outcome scenarios, which incorporated all technical elements of the Libra protocol.

Concluding the insights of this thesis it remains to be noted that Libra did not reinvent

the wheel with regard to permissioned replicated databases from a technical perspective. Although, Libra is designed to reach its declared goals, and, if allowed Libra can go beyond. The economic impact of Libra on the world economy is dependent on the governmental approval of the Libra Reserve.

6 Appendix

6.1 Move Intermediate Representation and Move Bytecode

Move Intermediate Representation	**Move Bytecode**

import 0x0.LibraAccount;
 	
main (payee: address, amount: u64) {
 	
LibraAccount.pay_from_sender(move(payee), move(amount));
 	
return;
 	
}
	{"code":
	[76,73,66,82,65,86,77,10,1,0,7,1,74,0,0,0,4,0,0,0
	,3,78,0,0,0,6,0,0,0,13,84,0,0,0,6,0,0,0,14,90,0,0,
	0,6,0,0,0,5,96,0,0,0,41,0,0,0,4,137,0,0,0,32,0,0,0
	,8,169,0,0,0,15,0,0,0,0,0,0,1,0,2,0,1,3,0,2,0,2,4,2
	,0,3,2,4,2,3,0,6,60,83,69,76,70,62,12,76,105,98,
	114,97,65,99,99,111,117,110,116,4,109,97,105,
	110,15,112,97,121,95,102,114,111,109,95,115,1
	01,110,100,101,114,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,2,0,4,0,12,
	0,12,1,19,1,1,2],"args":[]}

Figure 11: Peer to Peer transfer script in Move IR and Move Bytecode
This example is proof of concept for the purpose of illustration. The left side must be carefully coded by a human user, and the right hand side is what the BV actually checks and verifies.

6.2 Structure of a transaction

Libra	**Ethereum**
Sender address The MVM reads the balance, authetication key and sequence number from the account of the sender	**Receiver Address**
Sender public key The hash of the public key must match the authentication key retrieved from the senders address	**Signature**
Program Move modules and transaction scripts. Contains **Receiver Address**	**Datafield**
Gas price Number of Libra coin the sender wants to pay per gas unit for the execution	**Gas Price**
Maximum gas amount Maximum amount of gas the transaction can expend before halting	**Gas Limit**
Sequence number Incremented in Epilogue	**Nonce**

Figure 12: Tabular lineup of Libra and Ethereum transaction contents
The comparison shows a very similar transaction structure in both protocols.

6.3 Consensus

Evolution of LBFT

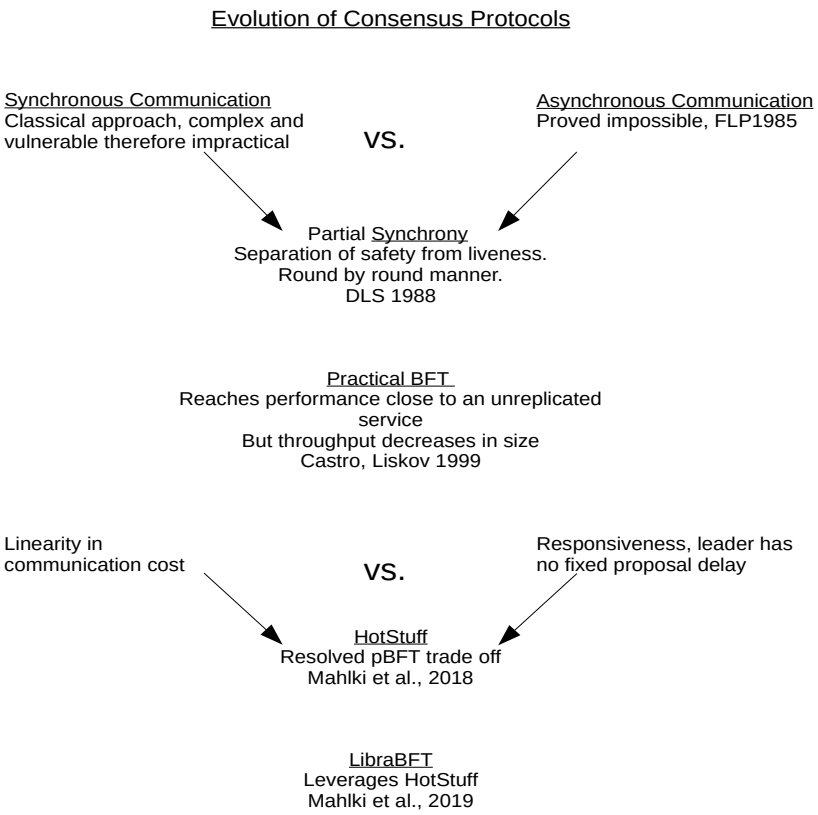

Figure 13: Evolution from Synchronous Consensus Protocols to LibraBFT, author's interpretation

Milestones in the development of BFT protocols; names refer to the main influencers.

References

[1] Authors from Calibra. *Move: A Language With Programmable Resources*. 2019. URL: https://developers.libra.org/docs/assets/papers/libra-move-a-language-with-programmable-resources.pdf.

[2] The Libra Association. "An Introduction to Libra". In: (2019). URL: https://libra.org/en-US/white-paper/#introduction.

[3] Authors from Calibra. "The Libra Reserve". In: (2019). URL: https://libra.org/en-US/about-currency-reserve/#the_reserve.

[4] Andres Bautista. "Vitalik Buterin: Libra has little in common with cryptocurrencies". In: (2019). [Online; accessed 27-December-2019]. URL: https://cryptorivista.com/ch-en/news/vitalik-buterin-libra-has-little-in-common-with-cryptocurrencies.

[5] The Libra Association. "How to Become a funding Member". In: (2019). URL: https://libra.org/en-US/becoming-founding-member/#overview.

[6] The Libra Association. "The Libra Association". In: (2019). URL: https://libra.org/en-US/wp-content/uploads/sites/23/2019/06/TheLibraAssociation_en_US-1.pdf.

[7] Tim Bartz. *Facebook verzichtet bei Libra auf chinesische Währung*. [Online; accessed 31-December-2019]. 2019. URL: https://www.spiegel.de/wirtschaft/facebook-will-kryptowaehrung-libra-nicht-an-yuan-koppeln-a-1287853.html.

[8] The LibraBFT Team. *State Machine Replication in the Libra Blockchain*. 2019. URL: https://developers.libra.org/docs/state-machine-replication-paper.

[9] Dan Emmons. *How to Verify and Publish on Etherscan*. [Online; accessed 17-December-2019]. 2018. URL: https://medium.com/coinmonks/how-to-verify-and-publish-on-etherscan-52cf25312945.

[10] Gavin Wood et al. "Ethereum: A secure decentralised generalised transaction ledger". In: *Ethereum project yellow paper* 2019 (2019).

[11] Authors from Calibra. "The Libra Blockchain". In: (2019). URL: https://developers.libra.org/docs/the-libra-blockchain-paper.

[12] Ralph Charles Merkle. "SECRECY, AUTHENTICATION, AND PUBLIC KEY SYSTEMS". In: *Information Systems Laboratory* (1979). URL: https://www.merkle.com/papers/Thesis1979.pdf.

[13] Satoshi Nakamoto et al. "Bitcoin: A peer-to-peer electronic cash system". In: (2008).

[14] The Libra Core Contributors. "This library implements a schematized DB on top of [RocksDB] (unsafe code)". In: (2019). [Online; accessed 27-December-2019]. URL: https://github.com/libra/libra/blob/f86d084141bd40b4335f7c09b9d66cb18f831116/storage/schemadb/src/lib.rs.

[15] Wikipedia contributors. *RocksDB — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-December-2019]. 2019. URL: https://en.wikipedia.org/w/index.php?title=RocksDB&oldid=931943473.

[16] Paul Dix. *Benchmarking LevelDB vs. RocksDB vs. HyperLevelDB vs. LMDB Performance for InfluxDB*. [Online; accessed 31-December-2019]. 2014. URL: https://www.influxdata.com/blog/benchmarking-leveldb-vs-rocksdb-vs-hyperleveldb-vs-lmdb-performance-for-influxdb/.

[17] Libra Developers. *Libra Github Repository*. [Online; accessed 26-December-2019]. 2019. URL: https://github.com/libra/libra/tree/master/storage.

[18] Leonid Reyzin and Sophia Yakoubov. "Efficient asynchronous accumulators for distributed PKI". In: *International Conference on Security and Cryptography for Networks*. Springer. 2016, pp. 292–309.

[19] Satoshi Nakamoto. *An E-mail of Satoshi Nakamoto*. [Online; accessed 20-December-2019]. 2008. URL: https://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html.

[20] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. *Impossibility of distributed consensus with one faulty process*. Tech. rep. Massachusetts Inst of Tech Cambridge lab for Computer Science, 1982.

[21] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. "Consensus in the presence of partial synchrony". In: *Journal of the ACM (JACM)* 35.2 (1988), pp. 288–323.

[22] Edsger W Dijkstra. "Self-stabilizing systems in spite of distributed control". In: *Communications of the ACM* 17.11 (1974), pp. 643–644.

[23] Miguel Castro, Barbara Liskov, et al. "Practical Byzantine fault tolerance". In: *OSDI*. Vol. 99. 1999. 1999, pp. 173–186.

[24] Obasi Ifegwu. *Finality*. [Online; accessed 31-December-2019]. 2019. URL: https://www.binance.vision/glossary/finality.

[25] Ittai Abraham, Guy Gueta, and Dahlia Malkhi. "Hot-stuff the linear, optimalresilience, one-message BFT devil". In: *CoRR, abs/1803.05069* (2018).

[26] Stefano De Angelis et al. "PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain". In: Jan. 2017.

[27] Statista. *PayPal's net number of payments*. [Online; accessed 31-December-2019]. 2019. URL: https://www.statista.com/statistics/218495/paypals-net-number-of-payments-per-quarter/.

[28] Christian Gorenflo et al. "Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second". In: *arXiv preprint arXiv:1901.00910* (2019).

[29] Elli Androulaki et al. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains". In: *CoRR* abs/1801.10228 (2018). URL: http://arxiv.org/abs/1801.10228.

[30] Authors from Calibra. "Moving Toward Permissionless Consensus". In: (2019). URL: https://iusletter.com/wp-content/uploads/Moving-Toward-Permissionless-Consensus_Libra-cosa-sta-accadendo-davvero.pdf.

[31] Calibra. *Coming in 2020: Calibra A New Digital Wallet for a New Global Currency*. 2019.

[32] vbuterin. *Proof of Stake FAQ*. https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ. [Online; accessed 31-December-2019]. 2019.

[33] Evangelos Deirmentzoglou. "Rewriting History: A Brief Introduction to Long Range Attacks". In: (2018). [Online; accessed 06-January-2019]. URL: https://blog.positive.com/rewriting-history-a-brief-introduction-to-long-range-attacks-54e473acdba9.

[34] Vitor Mesk. "Weak Subjectivity". In: (2019). URL: https://www.binance.vision/glossary/weak-subjectivity (visited on).

[35] Ethan Buchman, Jae Kwon, and Zarko Milosevic. "The latest gossip on BFT consensus". In: *arXiv preprint arXiv:1807.04938* (2018).

[36] Vitalik Buterin. "Decentralizing Everything with Ethereum's Vitalik Buterin". In: *Youtube* (2017). [Online; accessed 06-January-2019]. URL: https://www.youtube.com/watch?v=WSN5BaCzsbo.

[37] Wikipedia contributors. "WeChat — Wikipedia, The Free Encyclopedia". In: (2020). [Online; accessed 8-January-2020]. URL: https://en.wikipedia.org/w/index.php?title=WeChat&oldid=934653709.

[38] Anandi Chandrashekhar Raghu Krishnan. "Facebook may abort Libra launch in India". In: (2019). [Online; accessed 8-January-2020]. URL: https://economictimes.indiatimes.com/tech/internet/facebook-may-abort-libra-launch-in-india/articleshow/69867426.cms.

[39] African Development Bank Group. "African Economic Outlook 2019". In: (2019).

[40] ZOS Lending Network. *Decentralized Finance in Developing Countries: Its Potential and Constraints*. [Online; accessed 28-December-2019]. 2019. URL: https://medium.com/@ZOS/decentralized-finance-in-developing-countries-its-potential-and-constraints-5e5fc8fb8651.

[41] Anandi Chandrashekhar Raghu Krishnan. "Sub-Saharan Africa: Growth rate of real gross domestic product (GDP) from 2014 to 2024". In: (2019). [Online; accessed 8-January-2020]. URL: https://www.statista.com/statistics/805560/gross-domestic-product-gdp-growth-rate-in-sub-saharan-africa/.

[42] Aisha Salaudeen. "Why Tech giants like Google and Microsoft are investing in Africa". In: (2019). [Online; accessed 8-January-2020]. URL: https://edition.cnn.com/2019/05/17/africa/tech-giants-in-africa-intl/index.html.

[43] Anna Kuzmina. "Google, Facebook, Alibaba in Africa". In: (2019). [Online; accessed 8-January-2020]. URL: https://www.finextra.com/blogposting/16234/google-facebook-alibaba-in-africa.

[44] Aleksander Berentsen. "Monetary policy implications of digital money". In: *Kyklos* 51.1 (1998), pp. 89–118.

[45] Gregory N Mankiw and Mark P Taylor. "Economics". In: (2016).

[46] Russell Brandom. “Senators pressure Visa and Mastercard over work on Facebook’s blockchain project”. In: (2019). [Online; accessed 8-January-2020]. URL: https://www.theverge.com/2019/10/9/20906348/libra-association-visa-mastercard-stripe-blockchain-us-lawmakers-schatz-sherrod-brown.

[47] Lukas Mäder. “Finanzminister Maurer: Libra derzeit ohne Chance auf eine Bewilligung”. In: (2019). [Online; accessed 27-December-2019]. URL: https://www.nzz.ch/schweiz/bundesrat-maurer-libra-hat-derzeit-keine-chance-auf-bewilligung-ld.1530913.

[48] “Libra: Not So Fast, Swiss Government Says”. In: (2019). [Online; accessed 27-December-2019]. URL: https://www.finews.com/news/english-news/39367-switzerland-libra-swiss-government-ueli-maurer-libra-permit-facebook-finma-swiss-national-bank-snb.