BIKECOIN TECHNICAL WHITE PAPER

Trustless Infrastructure for Smart Bikes

BIKECOIN
ECOSYSTEM

Trustless Infrastructure For Smart Bikes

Version 1.5

Authors:

- Eric Bui, Chief Blockchain Architect, EBui@VolataCycles.Com
- Wee Keong Ng, Technical Advisor, Awkng@ntu.edu.sg
- Mattia De Santis, Chief Technology Officer, MDesantis@VolataCycles.Com

1. Introduction

Ethereum[1] is a decentralized computing platform where users deploy smart contracts to perform computations. Nobody controls the Ethereum network and users completely control their own smart contracts – it is a trustless environment. This is a fantastic platform on which to build a decentralized application which uses smart contracts to process transactions. Unlike the centralized applications such as Facebook, decentralized applications offer various advantages in terms of transparency, security and (with some additional technologies) data privacy.

One of the application areas for blockchain is Internet of Things (IoT). For example, BMW, General Motors, Ford and Renault[2] have launched their own blockchain research projects for the automotive industry, seeking to leverage IoT and blockchain technologies for a smart transport network.

BikeCoin is a project to introduce decentralisation and IoT for bike sharing, especially in crowded urban centers where bicycles are an important part of the transport network. We have partnered with premium bike company Volata Cycle – a "Tesla" in the cycling industry. Together, we believe that building a blockchain ecosystem for smart bicycles is needed more than ever. We have developed BikeCoin – a trustless infrastructure with fantastic features such as asset tokenization, decentralization with completely data privacy.

BikeCoin is empowered by several critical technologies:

1. A permissioned blockchain - an off-chain layer to assist in processing transaction for Ethereum Smart Contract and along with a peer-to-peer storage layer, leveraging IPFS technology to store encrypted data.
2. ERC721 to develop a Bike-Ownership protocol for identifying and verifying an owner of the bicycle.
3. Our innovative Zero-Knowledge proof and cryptographic search algorithms on encrypted data to provide data privacy entirely on BikeCoin blockchain.
4. BikeCoin token (ERC20) with state-channel technology to automate payment progress for a decentralized bike fleet management system and a

[1] Ethereum https://www.ethereum.org/
[2] BMW, GM, Ford and Renault launch blockchain research https://techcrunch.com/2018/05/02/the-mobility-open-blockchain-initiative-bmw-gm-ford-renault/

decentralized peer-to-peer application for smart bikes. These are the first decentralized applications in BikeCoin ecosystem.

BikeCoin supports the cycling industry to make smart bikes safer, more affordable and more widely accessible to users. In the following sections we will explain our technologies in detail, beginning first with blockchain.

2. Blockchain Technologies

2.1. An Ethereum-Based System

Many public blockchain projects are introduced with a higher transaction throughput requirement than Ethereum can support. However, these projects only prove their technologies when their networks reach to Ethereum network size of 15,000 nodes[3]. Sharding and Plasma are promising scalability solutions for Ethereum. These solutions, however, have not yet been deployed on Ethereum main-net. So we will instead implement a permissioned blockchain system – an off-chain layer to meet BikeCoin's rigorous decentralized application requirements such as low latencies, immediate transaction finality, high performance and excellent scalability. In the permissioned blockchain, contributors can provide computing resources to process BikeCoin transactions and earn block rewards.

In the BikeCoin platform, all user's data will be encrypted at the client-side before being sent to the system. To maintain access to the encrypted data, BikeCoin leverages IPFS peer-to-peer storage technology and a decentralized Distributed Hash Table (DHT) that is accessible through the blockchain. This DHT layer stores references to the data but not the data themselves. DHTs have been widely used to coordinate and maintain metadata about peer-to-peer systems. Kademlia[4] is a popular peer-to-peer DHT implementation that allows efficient lookup through massive networks, low coordination overhead, and resistance to various attacks by preferring long-lived nodes.

Following is a high-level block diagram of the BikeCoin system.

[3] Vitalik Buterin 'Ethereum Next 12 months" https://www.youtube.com/watch?v=a-xHIl-G_CQ&t=812s
[4] Petar Maymounkove and David Mazieres http://www.scs.stanford.edu/~dm/home/papers/kpos.pdf

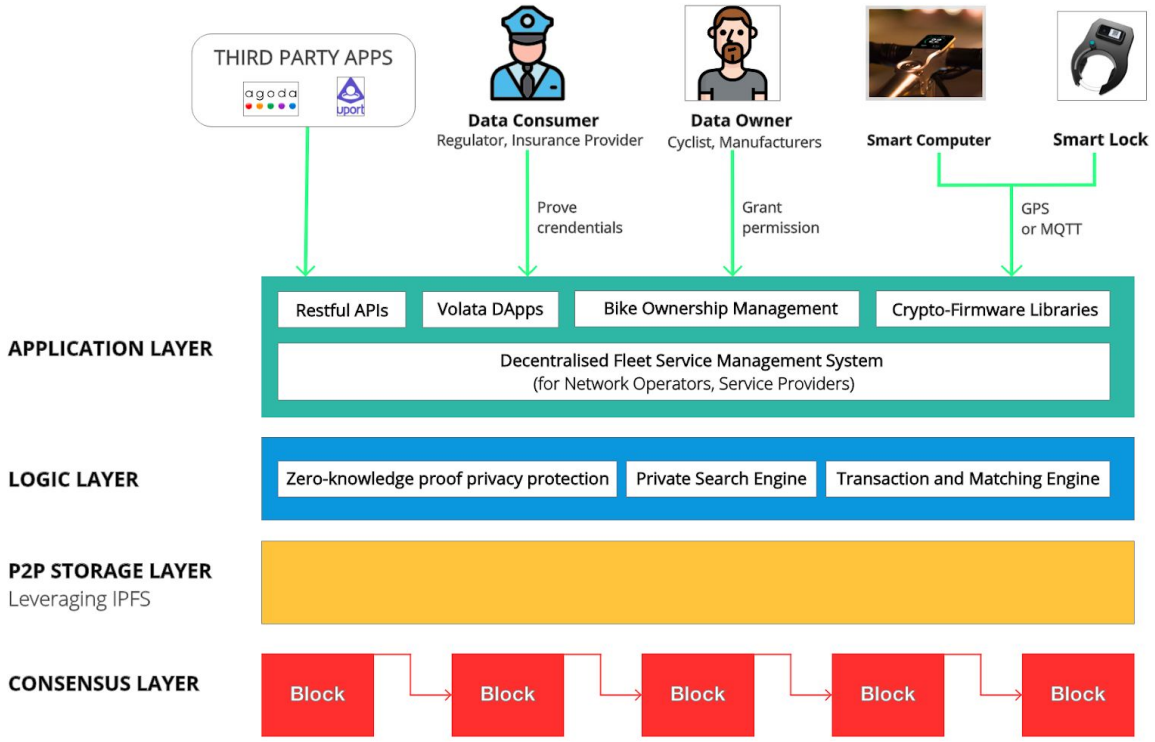

Figure 1 High-level System Architecture of BikeCoin

2.2. Asset Ownership And Tokenization

Ownership of smart bikes is rapidly increasing, but unfortunately also the theft of these expensive bikes. New smart computers or smart locks are on the market to remotely track and secure bikes. This provides opportunities for better theft prevention and new business models for bike rental. BikeCoin has developed a Bike-Ownership protocol to share ownership information. Bike owners can prove their ownership and report the location of stolen bikes by referencing immutable records on the blockchain. A smart computer or smart lock on the bike registers the position of the bike on the blockchain when they are locked. When a bike is stolen, the police are able to see the latest position to respond instantly. Potentially, insurance companies could automate claim processing for such losses through a smart contract running on the blockchain.

We have often heard how the Ethereum ERC721 smart contract is non-fungible, meaning that tokens of the same class or contract can hold a different value. The Cryptokittie[5] project is one interesting application. Cryptokitties are unique (and collectible) assets because each is tokenized with a different ERC721, having a different value.

[5] Cryptokitties https://www.cryptokitties.co/

For this reason, we developed the Bike-Ownership protocol[6] using ERC721 tokens from OpenZeppelin[7]. Each ERC721 represents a different Bike, and the value of the Bike-Ownership token is based on the metadata of token including bike technical information, picture, owner's information, supplier and manufacturer. Ownership is determined by an array of token indexes or ids that is mapped to Bike Owner address. The total supply of Bike-Ownership tokens is the length of array $allTokens$. The number of bicycles registered in the BikeCoin ecosystem is virtually unlimited, as the maximum number of any given type of ERC721 tokens, based on the unsigned integer storage type, is $2^{256} - 1$.

```
25
26  // Array with all token ids, used for enumeration
27  uint256[] internal allTokens;
28
```

Figure 2 Array allToken stores all token ids

```
109
110 /**
111  * @dev Internal function to add a token ID to the list of a given address
112  * @param _to address representing the new owner of the given token ID
113  * @param _tokenId uint256 ID of the token to be added to the tokens list of the given address
114  */
115 function addTokenTo(address _to, uint256 _tokenId) internal {
116   super.addTokenTo(_to, _tokenId);
117   uint256 length = ownedTokens[_to].length;
118   ownedTokens[_to].push(_tokenId);
119   ownedTokensIndex[_tokenId] = length;
120 }
121
```

Figure 3. Function addTokenTo in the smart contract BikeOwnershipprotocol.sol

```
248
249 /**
250  * @dev Internal function to add a token ID to the list of a given address
251  * @param _to address representing the new owner of the given token ID
252  * @param _tokenId uint256 ID of the token to be added to the tokens list of the given address
253  */
254 function addTokenTo(address _to, uint256 _tokenId) internal {
255   require(tokenOwner[_tokenId] == address(0));
256   tokenOwner[_tokenId] = _to;
257   ownedTokensCount[_to] = ownedTokensCount[_to].add(1);
258 }
259
```

Figure 4 Function addTokenTo in the smart contract ERC721BasicToken.sol

When a bicycle manufacturer or bike owner registers a new bike, the system will generate a random integer as a new token id and then call $addTokenTo$ from the BikeOwnershipProtocol.sol to issue a new Bike-Ownership token.

[6] Bike-Ownership protocol https://github.com/volatacycles/BikeOwnershipProtocol
[7] ERC721 standard OpenZeppelin https://github.com/OpenZeppelin/openzeppelin-solidity/blob/master/contracts/token/ERC721/ERC721Token.sol

```
31
32    // Optional mapping for token URIs
33    mapping(uint256 => string) internal tokenURIs;
34
```

```
58
59    /**
60     * @dev Returns an URI for a given token ID
61     * @dev Throws if the token ID does not exist. May return an empty string.
62     * @param _tokenId uint256 ID of the token to query
63     */
64    function tokenURI(uint256 _tokenId) public view returns (string) {
65      require(exists(_tokenId));
66      return tokenURIs[_tokenId];
67    }
68
```

```
98
99    /**
100    * @dev Internal function to set the token URI for a given token
101    * @dev Reverts if the token ID does not exist
102    * @param _tokenId uint256 ID of the token to set its URI
103    * @param _uri string URI to assign
104    */
105   function _setTokenURI(uint256 _tokenId, string _uri) internal {
106     require(exists(_tokenId));
107     tokenURIs[_tokenId] = _uri;
108   }
109
```

Figure 5 Each Bike-Ownership token mapped to a URI

Each Bike-Ownership token holds metadata which is the URI of the bike information record, such as manufacturer, bike frame information and smart computer lock information, held in the P2P storage layer.

Whenever users create a new Bike-Ownership token, the matching engine scans through all tokens using our searching engine (described in section 2.5.1); and if the matching engine finds any Bike-Ownership token with the same metadata, the system suggests users merge or transfer their token between users.

Consider this simple example: Alice bought a smart bicycle from Bob. This bike has a computer chip IMEI: VOLATA123456789. With the bike, Alice obtains a new Bike-Ownership token. But the matching engine detects that Alice's token has the same metadata (ie- IMEI code) as another bike-ownership token held by the bike manufacturer. The system notifies Alice and the manufacturer to merge these Bike-Ownership tokens. This could be accomplished by Alice

burning the token received from Bob and the manufacturer sending Alice the token in its possession, by a multi-signature contract.

2.3. Platform For Decentralized Applications

BikeCoin is a platform for decentralized applications (DApps). We are building two types of DApps: Fleet Services and P2P Services. Other vendors may build and launch their own DApps. This will result in a complete system as shown in the diagram below.

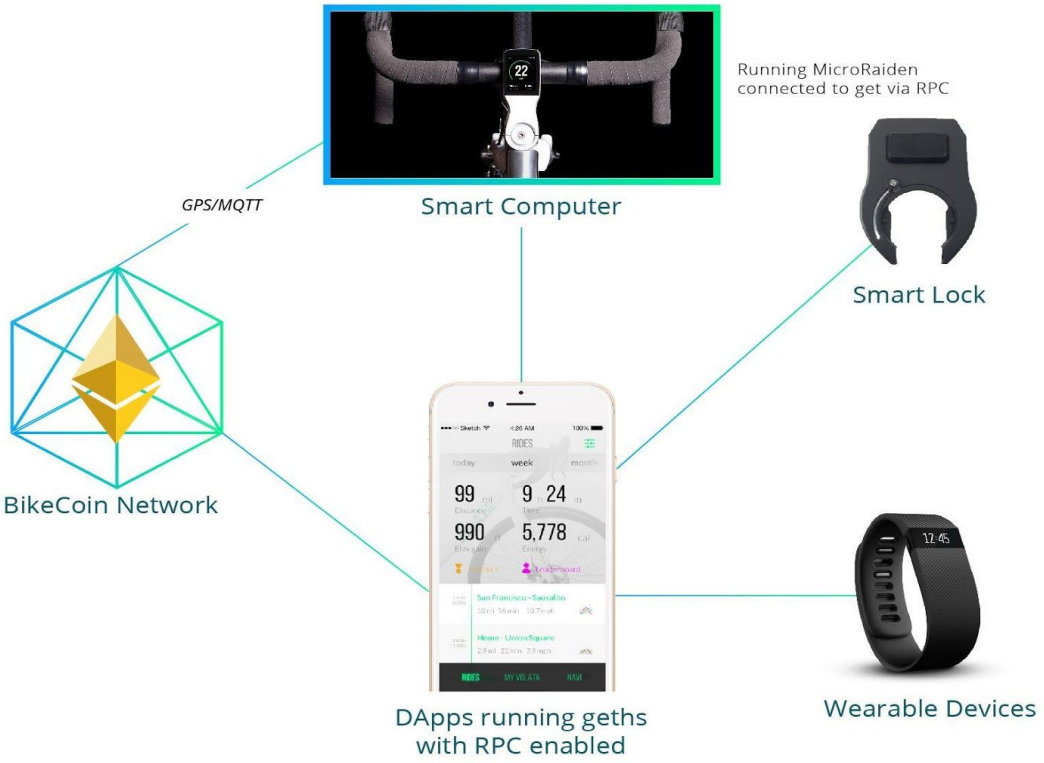

Figure 6 Networking topology with Raiden nodes in BikeCoin network

2.3.1. Fleet Services DApp

One of the unique features of BikeCoin is that multiple parties can collaborate to supply, operate and provide bicycles to the cycling public. These are described as fleet operations and the parties collaborating choose one or more of the following roles:

- Bicycle Suppliers (eg- Volata Cycles)
- Network Operators (bike retailers, bike hire companies or distributors)
- Service Providers (hotels, premium resorts, co-working spaces, local councils, police forces, universities)

The parties collaborate to supply and maintain bike rental services to end-users. The bike fleet services business model has many advantages over so-called 2nd generation bike sharing schemes (eg- Chinese dockless bike-sharing) in terms of positive user experiences and responsible ridership.

But the 2nd generation bike sharing schemes are centralized. So they rely on manual processes for bicycle maintenance service, auditing data and providing insurance coverage. In a centralized business, collaboration between bicycle suppliers, network operators, and service providers would be difficult because they would be required to store data centrally - resulting in reduced data privacy and increased data security risks.

Moreover, a membership business model - with fixed costs borne by the service provider - is not a fair way to share cost and revenue among bicycle suppliers, network operators and service providers. For example, service providers such as hotels and resorts have a seasonal business cycle. They should not bear costs when ridership is low or face bike shortage when the ridership is high. A flexible revenue-sharing mechanism is required.

BikeCoin provides a neutral and fair revenue sharing model which, once agreed by the parties, will operate in an automated manner using smart contracts.

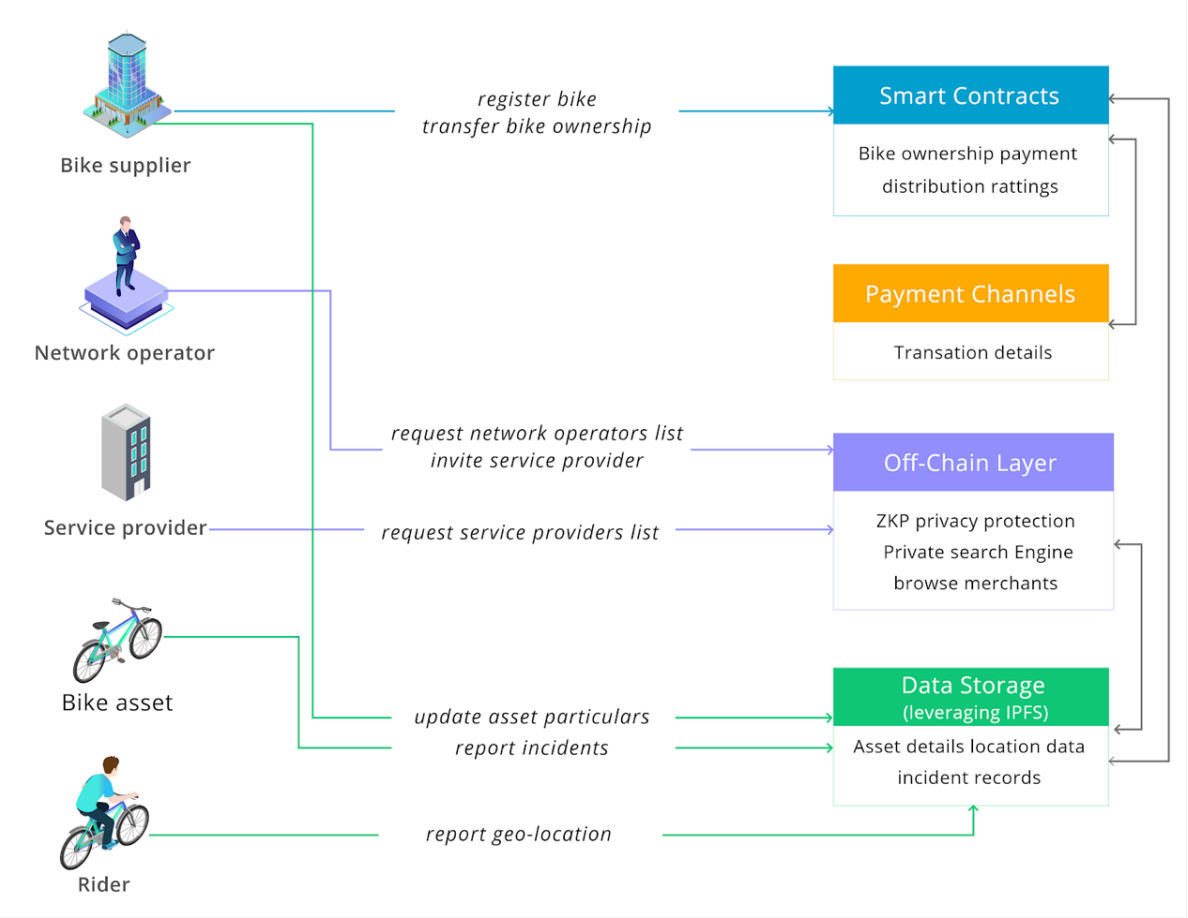

Figure 7 Asset Management & Partnering Features in Fleet Service

BikeCoin aims to build an open trustless blockchain ecosystem to decentralize bicycle fleet services where bicycle manufacturers, network operators, and service providers collaborate with each other in a pay-per-use model.

To join the platform, Bicycle suppliers only need to update a BikeCoin firmware library into the smart computer or smart locks of their bikes. Their bikes will then be able to directly interact with the BikeCoin blockchain. They can perform actions such as: closing a payment-channel, transfer BikeCoin (BKC) token to distribute revenue among stakeholders and record location, bike status, user profiles or other data into BikeCoin blockchain. In the future, BikeCoin will cooperate with IoT manufacturers to produce smart computers and smart locks which work with BikeCoin ecosystem. These steps will ensure broad adoption of BikeCoin by bicycle manufacturers.

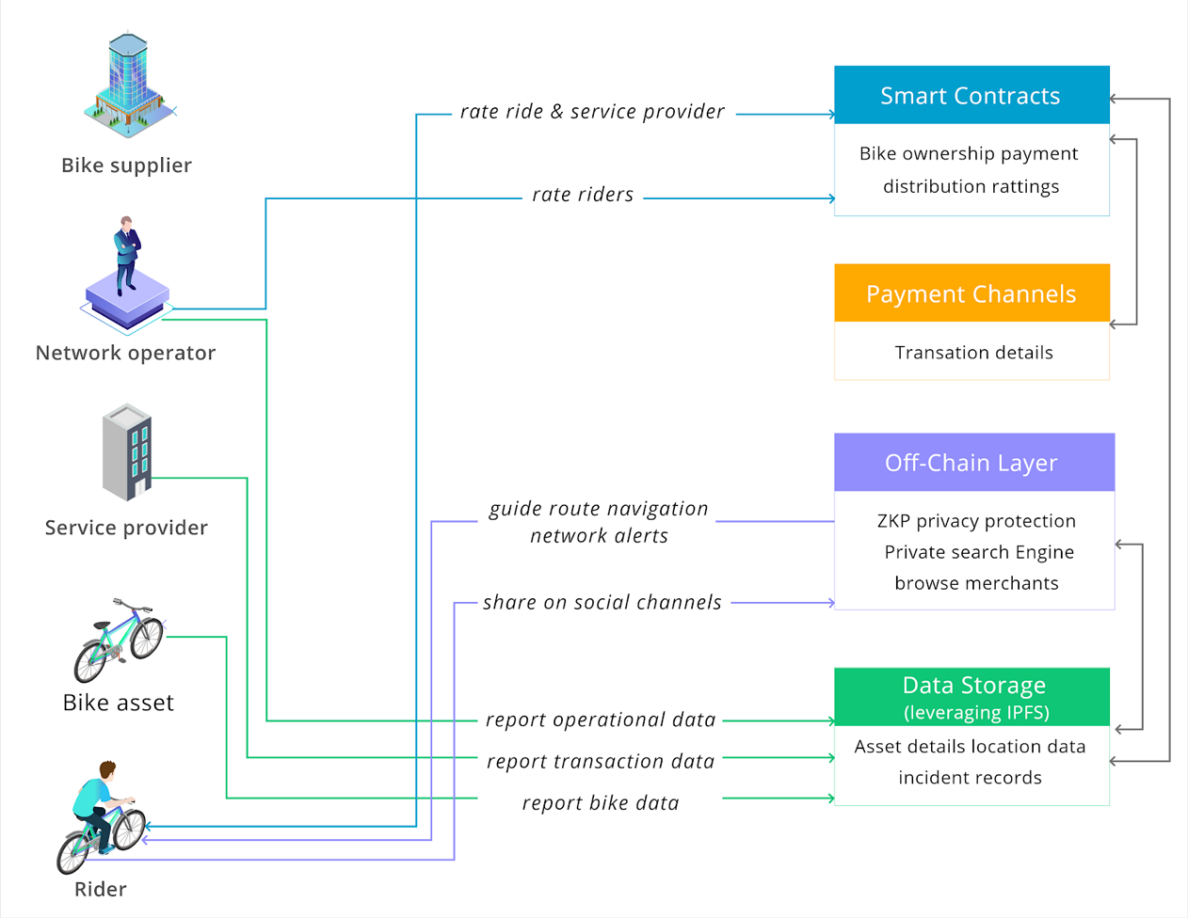

Figure 8 Data Sharing Features in Fleet Service

BikeCoin will develop a decentralized fleet management system which is flexible and adaptable to several situations, from the e-bike rental business for public use as the first decentralized app on BikeCoin ecosystem. The decentralized fleet management system helps network operators to decide whether to redistribute the bikes to areas of higher demand or to identify special infrastructure needs in locations where most bikes are ridden. Moreover, network operators and service providers are able to perform:

1. Real-time monitoring of bicycle diagnostic
2. Real-time bike geo-location
3. Reporting statistics of usage and riding behaviour
4. Scheduling maintenance services
5. Alerts for crossing restricted areas

Service providers use a decentralized app such as BikeCoin Fleet App to record performance and tracking data.

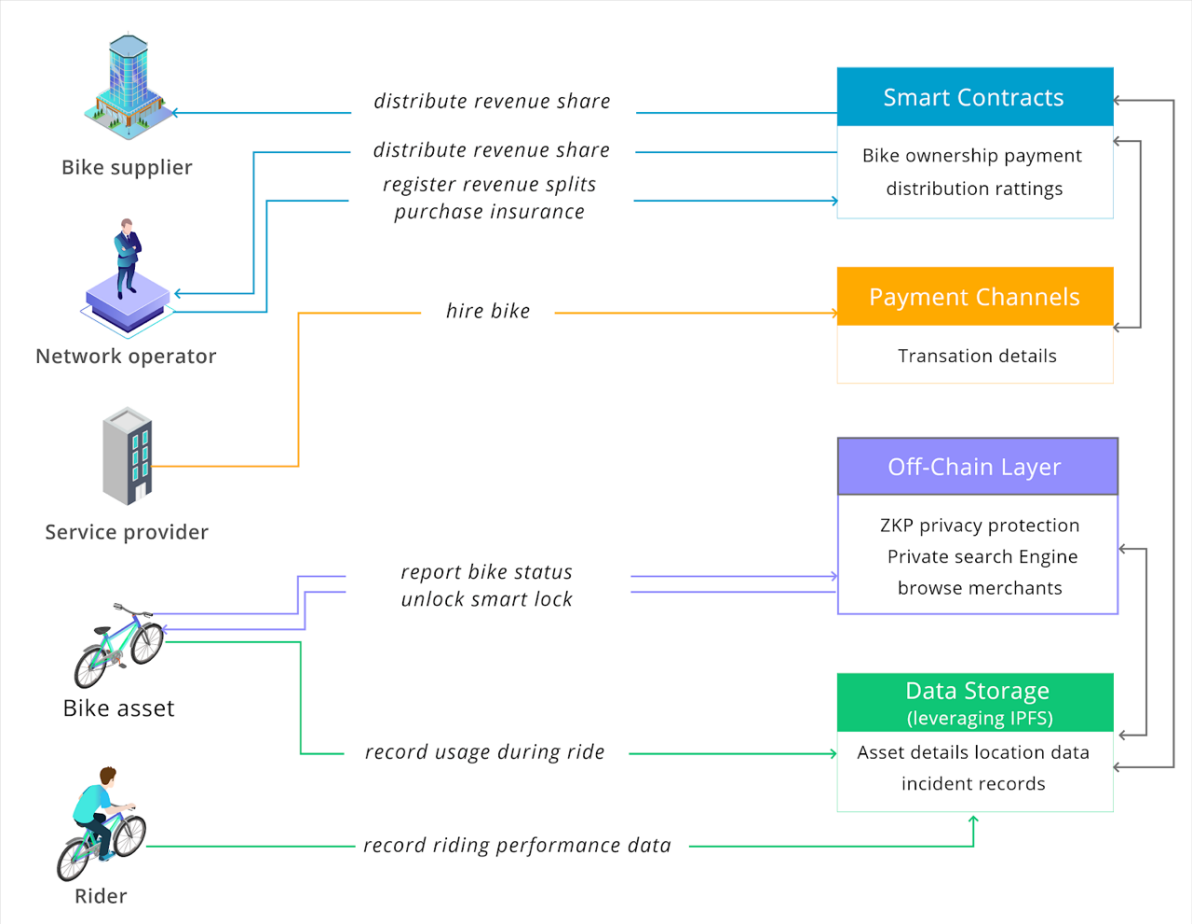

Figure 9 Transaction Support Features in Fleet Service

As an open platform, bicycle manufacturers, network operators, and service providers are able to reuse and easily customize the BikeCoin decentralized fleet management system to meet their unique business requirements. BikeCoin also incentivises community developers to develop advanced data management tools such as a bicycle diagnosis tool, smart relocation methods, etc. Potentially, insurance companies could automate their fleet insurance payment processes by using BikeCoin smart contracts, because all bike ownership and usage data are stored immutably on the BikeCoin blockchain.

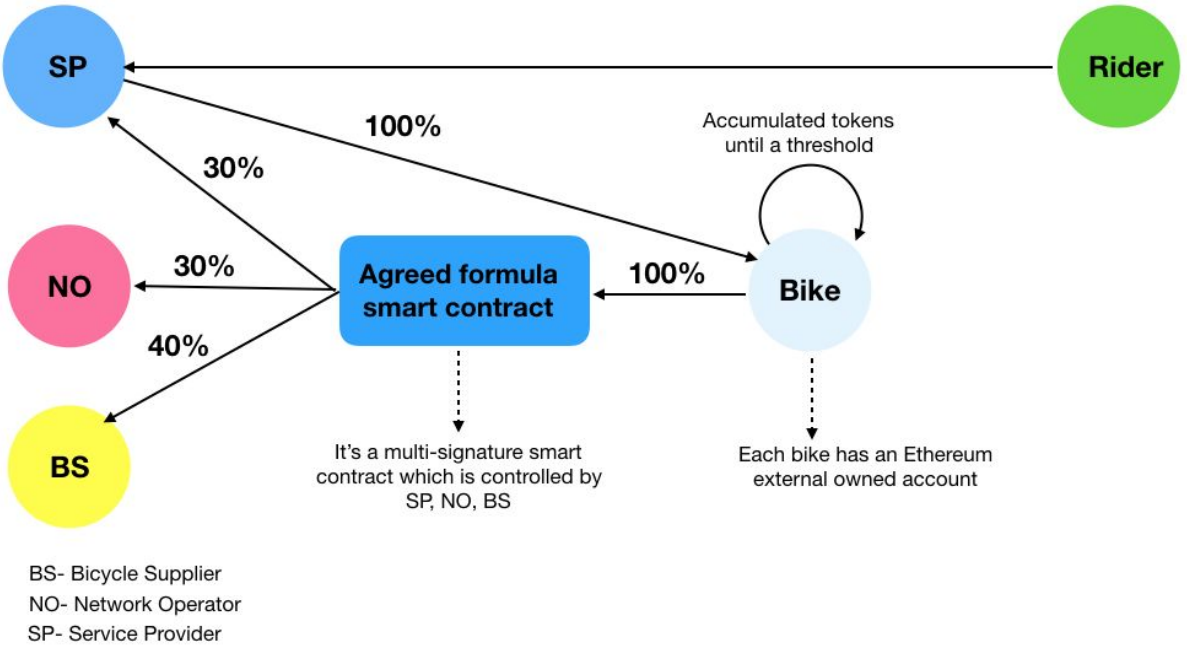

Figure 10 Revenue distributed by smart contracts.

Following is a hypothetical usage scenario. There are 4 actors in this scenario: (a) bike supplier Volata Cycles, (b) network operator BCPS, (c) service provider Marina Hotel and (d) the Rider, who happens to be a hotel guest.

1. Volata, BCPS and Marina Hotel (the merchants) make an agreement regarding deposits, revenue split, maintenance terms and costs. For example, the merchants might agree a revenue split of Volata 40%, BCPS 30% and Marina Hotel 30%. They might also agree that, regardless of how much Marina Hotel charges their guests to use the bikes, the hotel will pay BikeCoin 8 cents per riding minute, to be shared according to the split.
2. These terms are recorded immutably with an Ethereum multi-signature contract in which any change requires an endorsement of three parties.
3. Marina Hotel has to buy BKC tokens to transact. Each morning, staff of the hotel deposit 1000 BKC to open a Raiden payment channel with each Volata bike. This is an on-chain transaction; however, it is not necessary to be a real-time transaction.
4. Marina Hotel staff use the Bike Coin Fleet Services DApp to rent and unlock bikes and the Rider uses a BikeCoin DApp to record and share riding data. After each ride, the Marina Hotel account will automatically send a signed off-chain transaction directly to the Volata bike (which has its own address on the Ethereum blockchain).

5. At the end of each day, the bikes and Marina Hotel automatically sign and close the payment channel to settle their payments on-chain. This does not need to be a real-time transaction.
6. Whenever the balance on one of the Volata bikes reaches some preset threshold, the bike will automatically distributes revenue amongst the 3 merchants, according to the agreed split. BKC is transferred by the smart contract created in step #2.

In the BikeCoin ecosystem, bicycle suppliers, network operators, and service providers freely search and collaborate with each other trustlessly - without having to trust one another. All stakeholders rely on BikeCoin smart contracts. Service providers use the bike as an intermediary and the bike is only unlocked if it gets paid. In turn, the bike distributes revenue directly to all the merchants using multi-signature smart contracts.

2.3.2. P2P Services DApp

Another unique features of BikeCoin is that two parties can ride share in the same model as for example Airbnb. This is described as Peer-to-Peer (P2P) operations and the parties collaborating have either of two roles:

- Bicycle Owner
- Rider

This model enables individuals to earn an income from sharing of their smart bike. It allows riders to find a bike even if no merchants have launched fleet services in the city where they live. And it allows riders to try out a new smart bike, even though it is not displayed in showrooms, with only a small rental commitment. We believe this will promote sales of smart bikes, which can be purchased online instead of in shops. And online purchases will be less costly.

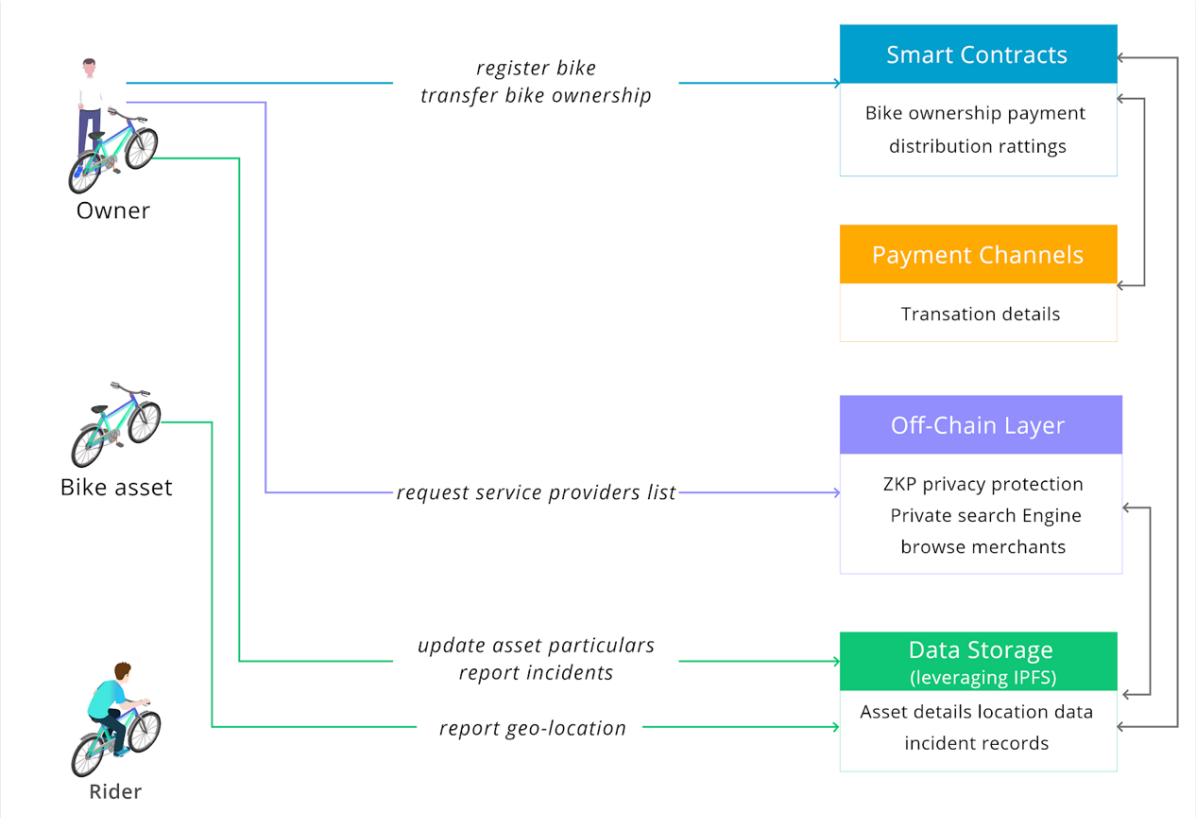

Figure 11 Asset Management & Partnering Features in the P2P service.

After registering their bike information using the BikeCoin-Ownership protocol, bike owners are able to share various information with potential renters,such as: bike information, photos, reviews and riding history. The bike owner lists bike id, location, rental fees, terms and commission (ie- fee for the Dapps) in an Ethereum smart contract. Each smart bike has an Ethereum externally owned account[8] and must participate in a payment channel of Raiden network as a receiver (learn more about how BikeCoin uses payment channels in section 2.4).

Bicycle renters register using the decentralized sharing-economy applications which are built on BikeCoin blockchain. Bike renters have to deposit BKC and open a payment channel in Raiden network as a sender. After that, they are able to search bikes with our private search engine (discussed in section 2.5.2) and then make a rental request. Each riding trip, bike renters pay directly to the bike's own account, by sending a signed off-chain transaction on Raiden Network. Whenever the bike's balance reaches a preset threshold, the bike automatically distributes revenue to bike owner(s) and rewards the BikeCoin DApps by transferring BKC to the agreed formula smart contract.

[8] Ethereum externally owned account
http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html

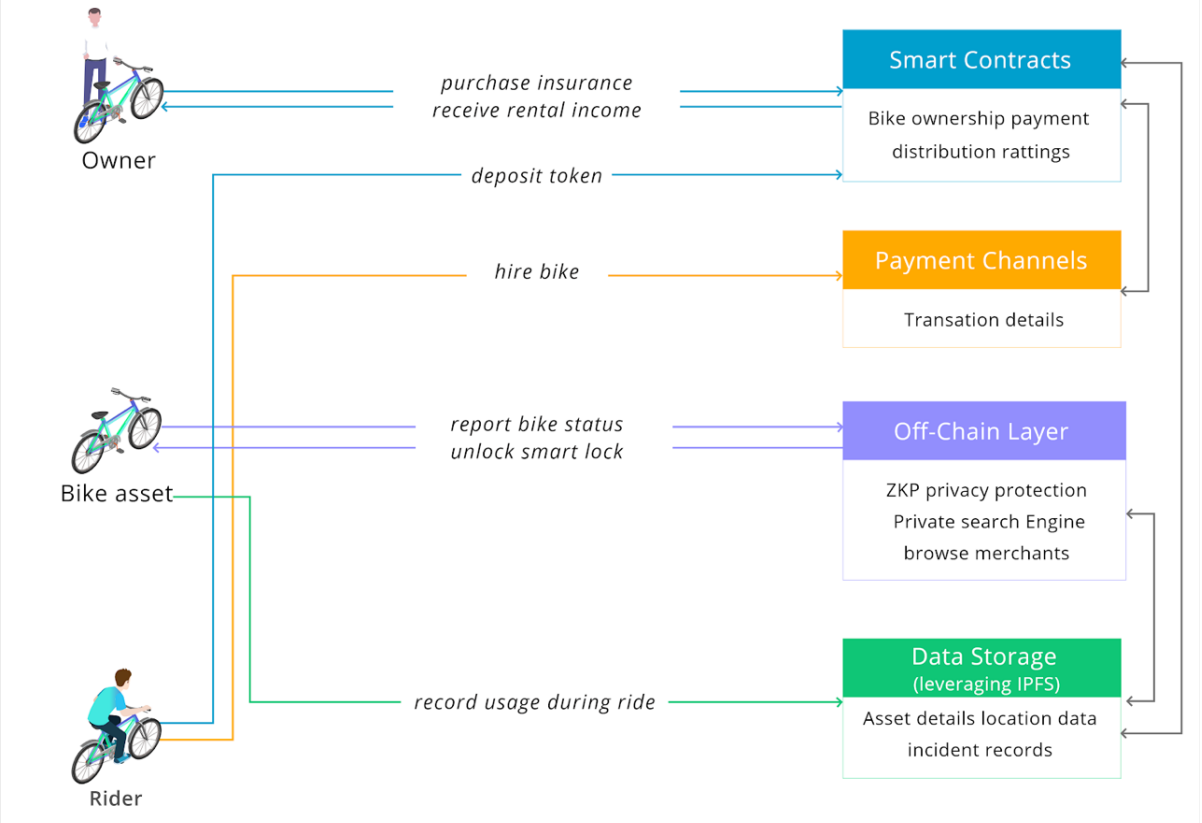

Figure 12 Transaction Support Features in the P2P service.

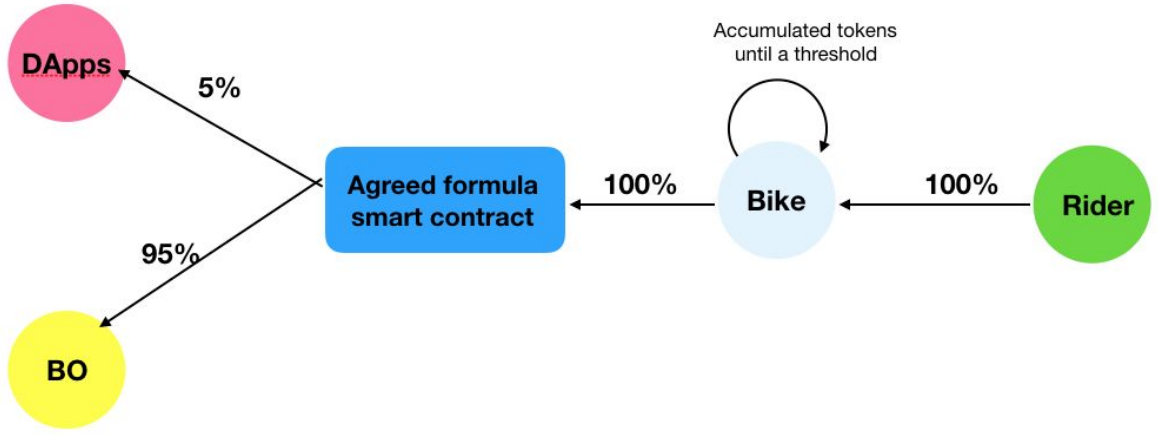

Figure 13 The payment flow in the P2P rental market

BikeCoin will release the Bike Ownership Management system for bike owners to manage all information, actions, orders and transactions; and BikeCoin sharing-economy DApps for bicycle renters to search bikes and control

their data privacy settings. BikeCoin also encourages third-party developers to develop their own sharing-economy DApps to leverage BikeCoin protocols.

2.4. Smart Contracts And Payment Channels

µRaiden (pronounced 'micro-raiden') is a payment-channel framework for frequent, fast and free ERC20 token based micropayments between two parties. Whereas its big brother the Raiden Network[9] aims to allow for multi-hop transfers via a network of bi-directional payment channels, µRaiden is for two parties only. Raiden-based networks allow us to transfer BikeCoin (BKC) near-instantly and with low-fees by using digitally signed and hash-locked[10] transfers. In the payment channel, users have to setup an on-chain deposit to open a payment channel, and then users can perform token transfers instantaneously, without limit, as long as the net sum of their transfers does not exceed the deposited tokens. Finally, users have to close the payment channel by calling functions $cooperativeClose()$ *or* $uncooperativeClose()$ in the smart contract $TransferChannels.sol$ [11] which requires on-chain transactions. BikeCoin leverages Ethereum smart contracts and payment-channel technology to automate payment processes among stakeholders.

2.5. Data Privacy

Data on a blockchain is stored in a "distributed" ledger of records and is immutable. That is, once recorded, data cannot be changed retroactively without the alteration of all subsequent blocks and a collusion of the network majority. Immutability is a powerful feature of blockchain technology, but it comes with a compromise - all data is public. Although blockchain is an open and semi-anonymous system, one can trace the route of specific coins and once they arrive at an exchange, it is often possible to tie coins to specific user accounts. This can lead to discovering the actors behind certain suspect transactions, for example to identify money launderers.

We believe that in the fleet services context, bicycle manufacturers, suppliers, network operators and service providers do not want disclosure of their commercial information. Merchants want privacy for things like supplier relationships and costs. Similarly, in the P2P context, riders and bike owners do not want to their data exposed without their permission.

---

[9] Raiden Network - https://raiden.network/
[10] Hashlock - https://en.bitcoin.it/wiki/Hashlock
[11] smart contract RaidenMicroTransferChannels.sol https://github.com/raiden-network/microraiden/blob/master/contracts/contracts/RaidenMicroTransferChannels.sol

Therefore, confidentiality for data is a fundamental requirement for the BikeCoin ecosystem. We are designing anonymous authority layers and a cryptographic search engine modeled by smart contracts to manage encrypted data with complete data privacy.

2.5.1. Anonymous Authority Layer With Zero Knowledge Proof

Anonymous Authority Layer allows a data consumer to obtain a credential from the data owner so that at some later point in time, he is able to construct a non-interactive proof of his credential. The blockchain nodes accept the request only if the attached proof is valid. The design of the component is inspired by the idea of Zerocoin[12] with Zero Knowledge Proof technology.

The following example describes how the anonymous authority layer works in practice. Let's suppose that there is a public bulletin board, one which is physical and everyone can access. There are two actors in this scenario: a cyclist who plays the role of data provider and an insurance company A which plays the role of a data consumer.

To produce a new credential for insurance company A, the cyclist firstly generates a pseudonym S for insurance company A and commits S using a secure digital commitment scheme. The resulting commitment C can be opened using a random number r known by the insurance company A. The cyclist pins C to the bulletin board, there is a set $S_C = (C_1, C_2, \ldots C_n)$ of commitments in the board. At a later point, the insurance company A is able to prove possession of such credential by producing two statements in zero-knowledge:

- He knows a commitment $C \in S_C = (C_1, C_2, \ldots, C_n)$.
- He knows the opening r for the commitment.

The BikeCoin system is used as a public bulletin board. Both the data owner and data consumer are able to access the public parts of the data stored on the blockchain. The public parts contain the commitments that we have described.

[12] Ian Miers, Christina Garman, Matthew Green, Aviel D. Rubin
http://zerocoin.org/media/pdf/ZerocoinOakland.pdf

We now present a concrete construction using cryptographic accumulator proposed by Josh Benaloh[13] , and later improved by Jan Camenisch [14]. The accumulator scheme comprises four algorithms:

- $AccumSetup(\lambda) \rightarrow params.$ Generates two primes $p, q,$ computes $N = pq$, sample $u \in QR_N$. Output $(N,\ u)$ as the parameters.
- $Accumulate(C) \rightarrow A.$ On a set of primes $C = \{c_{1,}..., c_n\},$ outputs accumulator $A = u^{c_1 \cdots c_n} \bmod N$.
- $GenWitness(\upsilon, C) \rightarrow \omega.$ Input a prime number $\upsilon \in C,$ outputs a witness $\omega = Accumulate(C - \upsilon).$
- $AccVerify(\omega, \upsilon, A) \rightarrow \{0, 1\}.$ Verifies $A = \omega^{\upsilon} \bmod N$ The security of the scheme is based on the harness of Strong RSA and Discrete Logarithm assumptions.

The description of the anonymous authority layer consists of four algorithms:

1. $Setup(1^{\lambda}) \rightarrow params.$ On the input parameter $\lambda,$ run the algorithm $AccumSetup(1^{\lambda})$ to obtain $(N,\ u).$ Generate primes p, q such that $p = 2^w q + 1$ for $\omega \geq 1.$ Let G be the subgroup of Z_q^* and select two random generator g, h such that $G = [g] = [h].$
2. $GenCred(S,\ params) \rightarrow (c,\ skc).$ Given pseudonym $S \in Z_q^*,$ select a random $r \in Z_q$ and compute $c \leftarrow g^S h^r$ such that c prime and $c \in [A,\ B],$ where $2 < A$ and $B < A^2.$ Set $skc = r$ and output $(c,\ skc),$ submit c to the blockchain.
3. $ShowCred(params,\ S,\ c,\ skc,\ Sc) \rightarrow \pi_S$ Given data consumer pseudonym S, a credential c and its secret key $skc,$ compute $A = Accumulate(params,\ S_c)$ and $\omega = GenWitness(params,\ c,\ S_c)$ and output the following proof of knowledge:

$$\Pi_s = ZKSoK\{\ (c,\ w,\ r,\ S)\ : AccVerify((N, u),\ A,\ c, \omega) = 1 \wedge c = g^S h^r\}$$

4. $VerifyCred(params,\ \pi, S_c).$ Given a proof $\Pi_S,$ and the public set of credential $S_C,$ first compute $A \rightarrow Accumulate\ (params,\ S_C),$ then verify that Π_S is the aforementioned proof of knowledge on $c, S_c.$ if the proof verifies successfully, output 1, otherwise output 0.

The zero-knowledge proof which appears in step 3 of the scheme is a non-interactive proof that only requires one round of communication. Camenisch presents an interactive zero-knowledge proof of knowledge in which an accumulator contains a committed value. The construction of the non-interactive proof in step 3 leverages a Fiat-Shamir transform on the

[13] Josh Benaloh, Michael de Mare
https://www.microsoft.com/en-us/research/wp-content/uploads/1993/01/owa.pdf
[14] Jan Camenisch, Anna Lysyanskaya http://cs.brown.edu/~anna/papers/camlys02.pdf

interactive proof. This is shown in the process flow diagram on the following page.

The $Setup$ algorithm is performed by the data owner to generate system parameters. Next, when the data consumer wishes to obtain a credential for data access, he sends a request to the data owner together with his pseudonym S. At this point, the data owner runs the $GenCred$ routine on this input S to generate a digital commitment and its secret key skc.

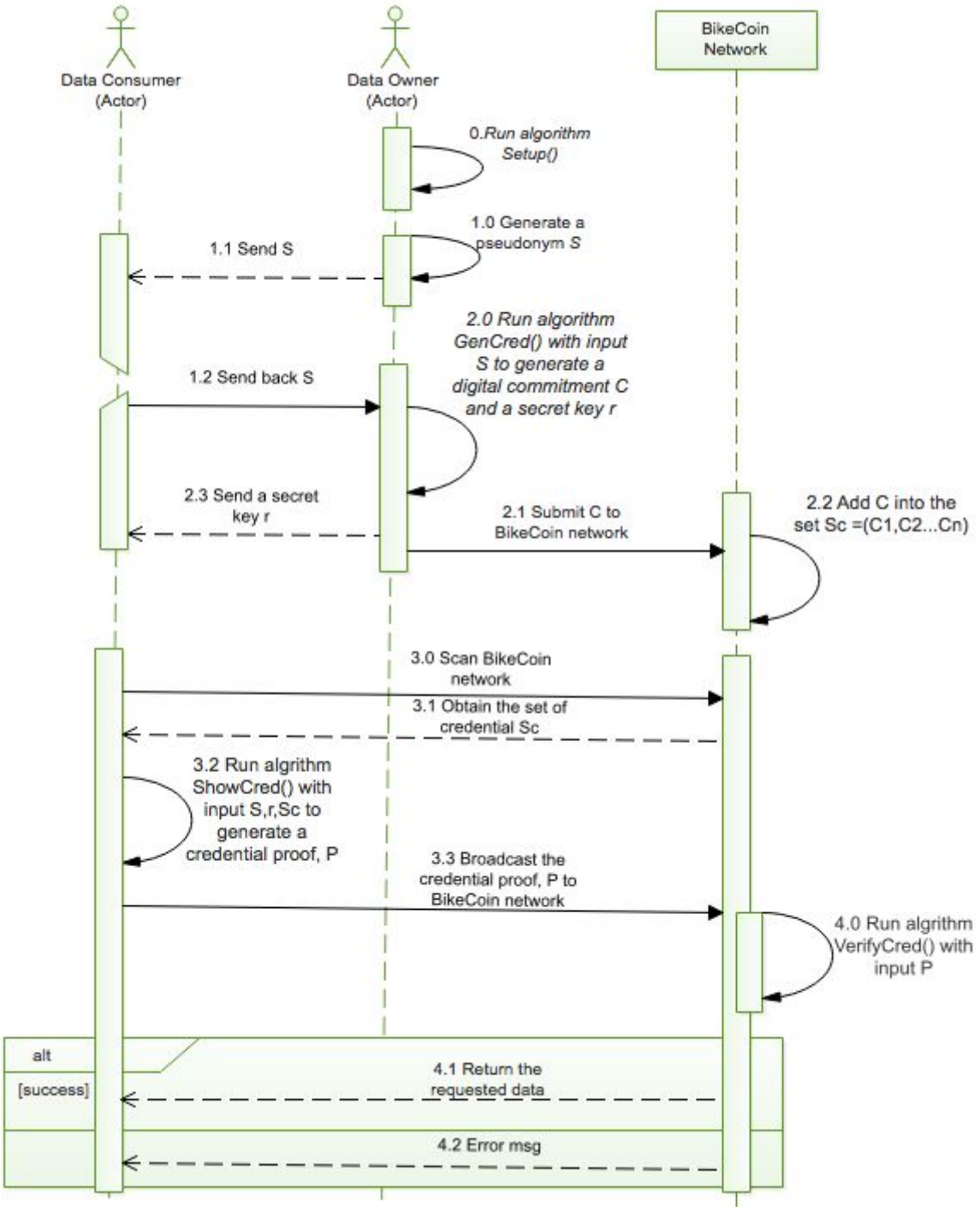

Figure 14 Sharing Data in the Anonymous Authority Layer

When the data consumer wishes to show his credential, he first scans through the blockchain (the bulletin board in our example) to obtain the set S_c consisting of all credential issued by the data owner. He then runs the $ShowCred$ routine to generate a credential proof, and broadcasts it to all blockchain nodes for verification. The blockchain nodes also collect the set of

credentials in the blockchain and validate the proof using the $VerifyCred$ algorithm. The credential certification is accepted if the last routine outputs 1.

The data consumer and the blockchain nodes are both required to compute $A = Accumulate(params, S_c)$ which requires a linear scan the blockchain data. The complexity of the protocol increases linearly with the number of registered data consumers.

With this robust anonymous authority layer, a data owner is able to freely share data with any data consumer without exposing her/his identity address. Unlikely central authentication applications like Facebook, no information about users is collected and all personally identifying data is encrypted and accessed only by smart contracts which users control. But social apps can be data consumers using BikeCoin. Riders and bike owners can freely upload riding performance data, heart rate, calories or identity documents and share them with friends on social apps such as Twitter or Facebook without exposing the data to anyone else. Bicycle manufacturers, hotels, premium resorts are able to collaborate confidently and exchange data with each other on BikeCoin network without revealing commercial information to their competitors.

2.5.2. Encrypted Data Search

But how is it possible to search this encrypted information? The crucial requirement of such a search must be fast return of results without exposing private data. Our solution is that the meta-data (ie- the fingerprint) of the encrypted data are stored in the blockchain and only authorized clients are permitted to use it to conduct searches. The authorization process is done by the Anonymous Authority Layer presented in the section 2.5.1

User's personal data is stored, shared and can be accessed via Distributed Hash Table (DHT) in a peer-to-peer storage scheme as discussed in section 2.1. An access key is computed using a hash function (ie- fingerprint of the data). The blockchain does not store the actual data content, however, it maintains the access key data so that data consumers are able to link real data to the DHT using blockchain.

We denote EKS as the encryption scheme that supports keyword search. The data owner appends a list of EKS ciphertext of each keyword to the access key and stores it in the blockchain layer. A data D with keywords $W_1, W_2, ..., W_n$ is stored in the blockchain layer under the structure: $H(D) \| EKS(W_1) \| ... \| EKS(W_n).$ An authorized data consumer is able to produce a certain trapdoor Γ_ω that

enables a smart contract to test on each data entry whether one of the keywords associated with the access key (eg- the document) is equal to the word W. Given a trapdoor and EKS ciphertext, the blockchain nodes can only test whether $W = W'$, and nothing else.

A typical keyword search cryptosystem consists of four general algorithms:

1. $KeyGen$: generates cryptosystem key.
2. $Trapdoor$: produces trapdoor T_W for a keyword W
3. $Encrypt$: produces a EKS ciphertext for keyword W
4. $Test$: tests whether keyword in the trapdoor is matched to the EKS ciphertext.

In the BikeCoin ecosystem, an additional algorithm is required for the data owner to produce a secret search key for the data consumer. We denote that algorithm $KeyDerive$. Three algorithms $KeyGen, Encrypt$ and $KeyDerive$ are performed by the data owner, while $Trapdoor$ is run by the data consumer to generate a trapdoor, and finally, the $Test$ algorithms is performed by smart contracts or the blockchain peers. We modify the protocol proposed by Raluca Ada Popa[15] to adapt to our environment.

We start the protocol description by reviewing a few concepts related to bilinear maps. We will use the following notation: G_1 and G_2 are two (multiplicative) cyclic groups of prime order p, g_1 is a generator of G_1 and g_2 is a generator of G_2. A bilinear map is a map $e : G_1 \times G_2 \rightarrow G_T$ with the two following properties: $(1)\ Bilinear : \forall u \in G_1, \upsilon \in G_2$ and $a, b \in Z$, then $e(u^a, v^b) = e(u, v)^{ab}$, and $(2)\ Non-degenerate : e(g_1, g_2) \neq 1$.

We denote $H : \{0,1\}^* \rightarrow G_1$ and $H_2 : G_T \times G_T \rightarrow \{0,1\}^*$ to be two random oracles, and g_1, g_2, g_T are respectively the generators of groups G_1, G_2, G_T. The private keyword search system consists of five algorithms as the follows:

1. $KeyGen : k \leftarrow Z_p$.
2. $KeyDerive(k, s) : k_s \leftarrow g^{\frac{k}{s}}$.
3. $Trapdoor(w, s) : T_w \leftarrow e(H(w)^s, k_s)$.
4. $Encrypted(k, w) : Random\ r \leftarrow G_T.\ Output : c = (r, H_2(r, e(H(w), g_w)^k)$.
5. $Test : Parse\ c = (r, h)$. Test whether $H_2(r, tk) = h$.

[15] Raluca Ada Popa and Nickolai Zeldovich
https://people.csail.mit.edu/nickolai/papers/popa-multikey-eprint.pdf

The data owner generates a secret key k for keyword encryption EKS, and derives keys for the data consumers. Each data consumer poses a secret $s \in Z_p$, which can be generated by a mapping from his pseudonym with the data owner. Using k and s, the data owner computes a search key k_s for the data consumer so that later he can use it for trapdoor construction.

The correctness of the protocol follows the two equations:

$$tk = e(H(w)^s, g_2^{\frac{k}{s}}) = e(H(w), g_2)^k, \text{ and } H_2(r, tk) = H_2(r, e(H(w), g_2)^k)$$

The above scheme has data hiding and token hiding properties. Data hiding (privacy) requires that the semi-honest adversary is not able to distinguish between ciphertexts of two values not matched by some token. Token hiding (privacy) requires that the adversary cannot learn the keyword that one searches for. The complexity of the protocol increases linearly with the number of data sets stored in the BikeCoin ecosystem.

2.6. Future Research

2.6.1. Scalability Solutions For Public Blockchain

It is widely accepted that a blockchain-based system is as secure and robust as its consensus model. The best consensus model will be one which best balances the following three following factors:

- Decentralization: Any node freely participates in processing transactions, and publishing a block without use of a central authority or service.
- Security: The system has to prevent double-spends, keep data in sync. There are no conflicts when data get merged. All nodes see same data at the same time.
- Scalability: The system has to provide sufficient transaction throughput to

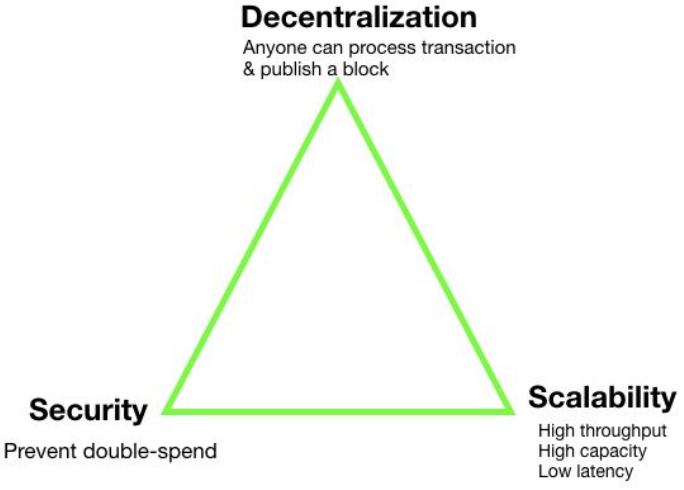

Figure 15. Three crucial factors of blockchain consensus algorithms

serve enterprise-scale or even planet-scale needs, especially when the network size increases. Following is a table listing different types of consensus algorithms, and highlighting the strengths of each.

	Types of Consensus				
Features	**PoW**	**PoS**	**PoET**	**BFT * and variants**	**Federated BFT**
Blockchain Type	Permission-less	Both	Both	Permissioned	Permission-less
Blockchain	BitCoin, Ethereum	Omisego, BitShare, PeerCoin	Intel Ledger	Hyperledger Fabric	Ripple & Stellar
Transaction finality	Probabilistic	Probabilistic	Probabilistic	Immediate	Immediate
Transaction rate	Low	High	Medium	High	High
Scalability of peer network	High	High	High	Low	High
Trust model	Untrusted	Untrusted	Untrusted	Semi-trusted	Semi-trusted
Adversary Tolerance	<= 25%	Depends on specific algorithm used	Unknown	<= 33%	<=33%

* Note: BFT is Byzantine Fault Tolerance

Figure 16. A comparison of blockchain consensus mechanisms

Sharding[16] and Plasma[17] of Ethereum blockchain are promising solutions to achieve the best balance of three factors. Sharding and Plasma divide transactions into small groups for parallel transaction processing, to increase transaction throughput of the whole blockchain ecosystem. In these small groups, an appropriate consensus algorithm is used for efficiency and performance. BikeCoin actively embraces these promising scalability solutions.

2.6.2. Security And Privacy For Data Storage And Computation

We already showed how to do storage and random queries using zero-knowledge proofs and cryptographic algorithms. However, there are other

[16] Sharding in Zilliqa blockchain https://docs.zilliqa.com/whitepaper.pdf
[17] Plasma https://plasma.io/

alternative computational solutions for security and data privacy. These include Secure Multi-party Computation and Trusted Execution Environment (TEE). Secure multi-party computation enables multiple parties to jointly compute a function over inputs without disclosing said input. The well-known Enigma[18] project focuses on this solution.

Trusted Execution Environment solutions are mostly based on Intel SGX [19] and are intended for use with the Intel SawtoothLake[20] blockchain. Based on our experience, each of the three solutions Zero Knowledge Proof, Secure Multi-Party Computation and Trusted Execution Environment have distinct advantages, and none of them can fully replace the others. Therefore, BikeCoin actively collaborates with other blockchain startups on research to develop better computational solutions for security and data privacy.

3. Other Technologies - IoT and Firmware

3.1. Emerging Trends

A bicycle is the most energy-efficient means of human transport ever invented, and the future is all about sustainability. Bikes solve multiple problems of urban mobility and sedentary life. With a rapidly increasing number of riders active in the bike-sharing market, bicycle manufacturers and tech giants are seeking to apply innovative Internet of Things (IoT) technologies to enable smart bikes which can navigate and provide a better and indeed safer user-experience.

Ofo and Huawei have worked together to build the NB-IoT-based[21] smart shared bike lock solutions providing lower power consumption, better coverage, and lower latency. Similarly, Mobike, AT&T, and Qualcomm[22] are currently collaborating on

Figure 17 the Remote Controller and Lock system

[18] Enigma https://enigma.co/
[19] Intel SGX https://en.wikipedia.org/wiki/Software_Guard_Extensions
[20] Intel SawtootLake Architecture Overview https://sawtooth.hyperledger.org/docs/core/releases/0.7/sawtooth_developers_guide/architecture_overview.html
[21] Smart Shared Bicycle Lock http://www.huawei.com/minisite/iot/en/smart-bike-sharing.html
[22] Mobike, AT&T and Qualcomm collaborate https://www.prnewswire.com/news-releases/mobike-att-and-qualcomm-collaborate-on-mobile-iot-smart-bike-share-technology-300516548.html

Mobile IoT smart bike sharing technology. Moreover, IoT-Smart bike technologies are being embraced as equipment add-ons by tech enthusiast riders. This is the case with the MAT Remote Controller and Lock system (pictured above), which won the HONOREE prize at CES 2018 Innovation Awards in the *Vehicle Intelligence and Self-Driving Technology category*.[23]

3.2. Volata Technologies

Cars have been revolutionized by software, to improve both safety and user experience. We are now witnessing the introduction of smart cars which self-drive and behave as fleets on the open highway. It's time to apply these IoT technologies to bicycles.

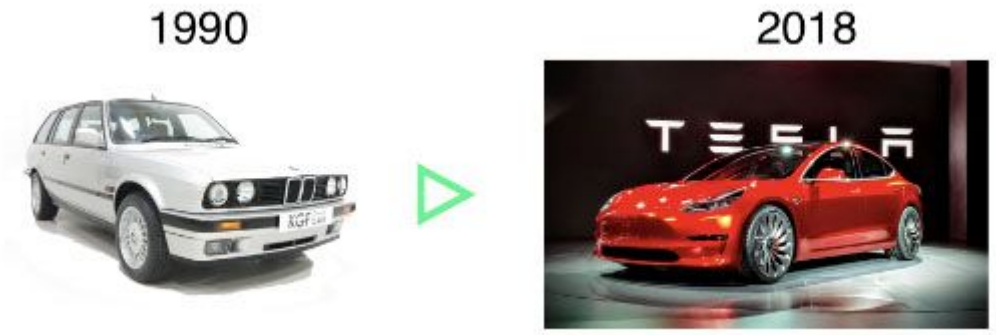

Figure 18 Car Revolution

Volata Cycles is the first bicycle company to design hardware, develop software and take care of distribution and service. As BikeCoin's first partner and supplier, we want Volata to do within the cycling industry what Tesla has done in the automotive industry. In the next sections, we discuss Volata Cycles innovations in smart bike hardware, electrical systems and software.

3.2.1. Smart Bike Hardware

Volata provides the user with all the features that he or she needs when using a bike for commuting or recreational riding, with a focus on safety, and providing them significant rewards while riding. Information is provided using a dashboard with a built-in computer which is easily controlled via thumb joystick and shows smartphone notifications (such as calls, text messages) on its display. Of particular note, the rider is able to receive navigational cues.

[23] MAT awarded at CES 2018 innovation awards
http://www.esb.bike/mat-awarded-ces-2018-innovation-awards/

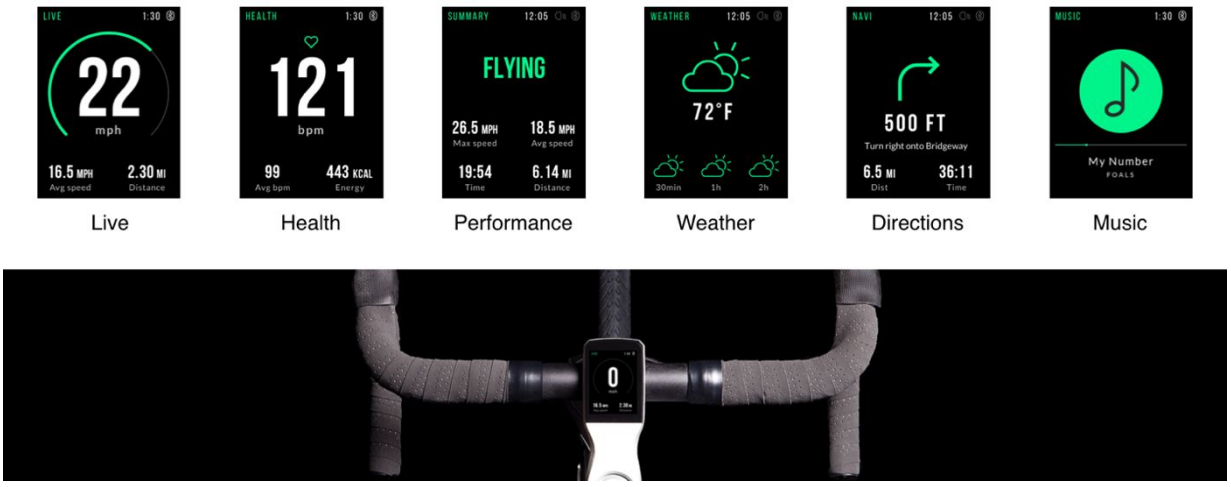

Figure 19 Built-in applications on Volata Smart Computer

Following are illustrations which show how the subsystems fit together.

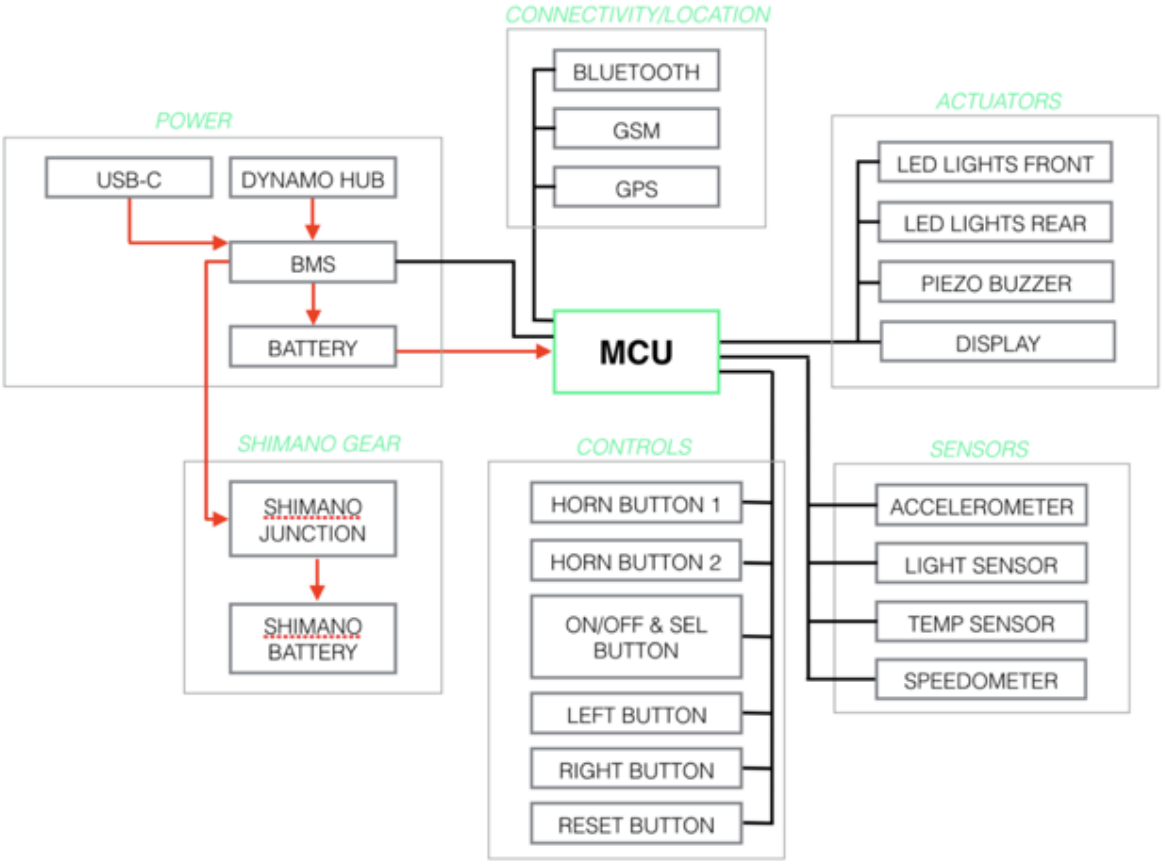

Figure 20 Overall Architecture design

The bicycle connects to the internet (via GSM) and to the rider's smartphone. Using an app, the rider can: obtain a riding data summary, record personal milestones, review a timeline of all trips, set a destination and receive navigational assistance. They can, if desired, use the app to share their performance with friends. Just like Tesla and BMW autos, a Volata bike can be located, monitored, and remotely locked or unlocked at any time from a smartphone. As part of the anti-theft system, users can receive notification if somebody tries to steal a bike, and can track the bike via GPS if it is missing.

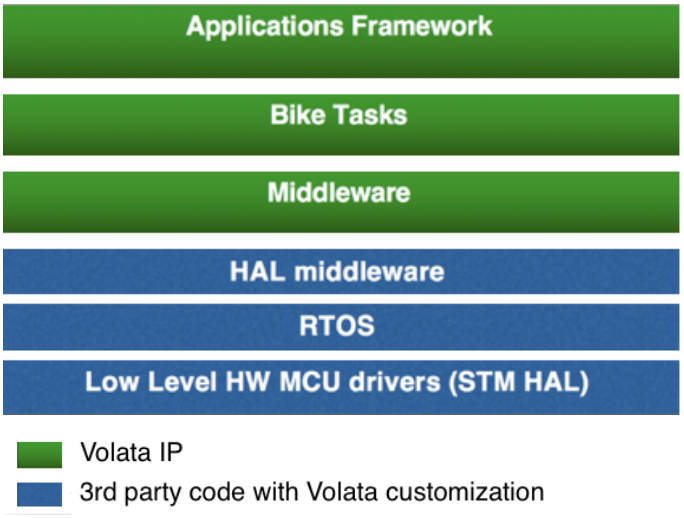

Figure 21 Software Components

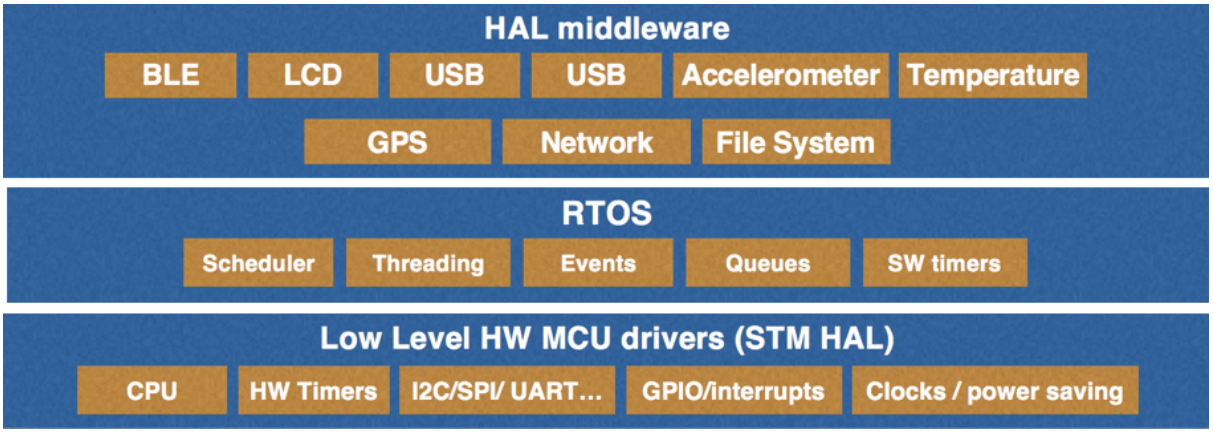

Figure 22 RTOS and Drivers

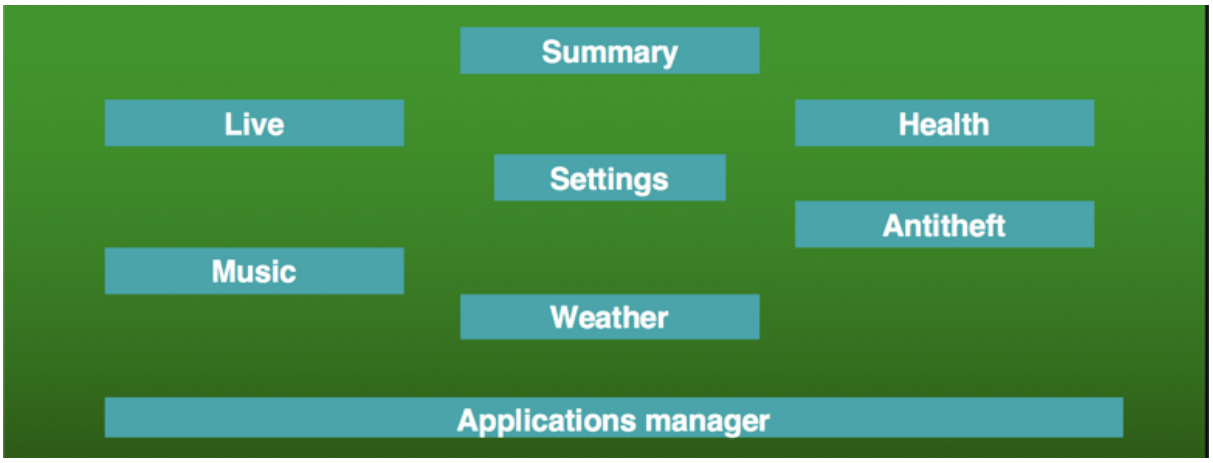

Figure 23 Application Framework

3.2.2. Electrical Systems & Software

In the front hub of each bike is a dynamo which powers the electrical system and charges a battery (build into the frame). This electrical system supports the inbuilt computer and anti-theft systems.

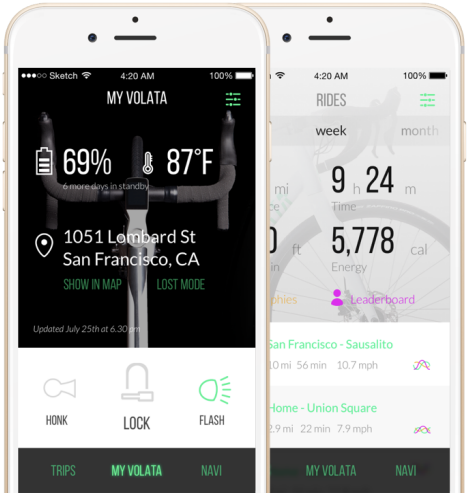

Figure 24 Volata App

The BikeCoin APIs communicate with firmware to allow suppliers and owners to track assets and allow users to contract rides and 'pay the bike'. Each bike has an Ethereum address and a corresponding account. Volata bikes used in the BikeCoin system can receive BKC tokens, open/close payment channels and automatically distribute BKC among stakeholders as revenue splits.

Electric bicycles will play a crucial role in the transition to sustainable transportation. As a next step, Volata will also expand to make electric bicycles with the same level of integration as their current premium model.

4. Discussion

In contrast with centralized systems such as cloud computing, blockchain technology allows us to build a system without a trusted authority, one in which all stakeholders collaborate with one other in a frictionless way. Blockchain

allows developers to decentralize everything[24]. However, we are still at an early stage of deployment and there are some challenges such as scalability and data confidentiality which need to be solved in order to meet rigorous requirements of enterprise applications.

In this technical whitepaper we have shown that BikeCoin combines a number of technologies to provide workable solutions without waiting for future research. We have demonstrated how we can use the public Ethereum blockchain, tokenize bikes as physical assets, execute smart contracts in a more flexible manner off-chain, create payment channels for fast settlement, establish a privacy layer allowing permissioned data sharing and bring it all together with firmware deployed on smart bikes.

BikeCoin uses the permissioned blockchain as an off-chain layer to assist Ethereum on-chain transactions. The on-chain smart contracts handle bike registration, hiring of rides and payment remuneration. The off-chain smart contracts do routing, data storage, and search transactions. This pragmatic approach helps BikeCoin deliver a fast and realistic technical roadmap. It also ensures that the project is less risky since it does not depend on further research and development.

Still, there are areas of emerging blockchain solutions that we see potential in and will continue to monitor as they make their way out of labs and onto the main public network:

- In the area of scalability, we believe that Sharding (eg- in Zilliqa[25]) and Plasma are the most promising solutions to achieve the best balance of three factors (decentralization, security, and scalability).
- In the area of security and privacy for data and computation, we are watching Zero-knowledge proof (using cryptography and local computation), fully homomorphic encryption (using pure cryptography), secure multi-party computation (sMPC using cryptography and distributed machine) and trusted execution environment (using secure hardware – Intel SGX).

As mentioned before, BikeCoin will actively collaborate with world-class blockchain research teams on such innovative solutions.

[24] Decentralized Everthing – Vitalik Talks https://cyber.capital/2017/10/11/decentralizing-everything-with-ethereums-vitalik-buterin-disrupt-sf-2017/

[25] Zilliqa blockchain - https://www.zilliqa.com/

5. Conclusion

Bicycles are gradually taking their place among the Internet of Things, and bike IoT is innovating fast through the addition of add-ons such as GPS devices and smart locks controlled by mobile apps. What BikeCoin brings is firmware that allows integration of the bikes as tokenized assets in an open and fully decentralised bike sharing platform. This trustless bike sharing system allows various types of merchants to participate in bike supply, fleet operations and retail service provision.

As described in this paper, there are several technical challenges that BikeCoin solves to make premium bike sharing a reality. Bikes are instantiated as digital assets using non-fungible ERC721 tokens. With smart contracts these bikes are able to act as intermediaries, receiving payment and distributing revenue shares according to preset splits. The entire system is made trustless by using Zero Knowledge Proofs, so that merchants using BikeCoin do not have to worry about their confidential commercial data.

A number of efficiency concerns are also addressed in this paper, namely how to ensure that transactions are fast and can be handled in the high volume that would be expected if BikeCoin had the same number of bikes as competitors such as Ofo and Mobile. This is achieved with off-chain payment channels, using Raiden technologies.

Finally it is important to conclude by reminding that BikeCoin is intended as an open and decentralised system. Merchants set their own prices. We will publish APIs, firmware and other open source components to help developers build decentralised apps on BikeCoin platform.