The Force Protocol

Decentralized Cryptoassets Loan Protocol

The Force Protocol

Decentralized Cryptoassets Loan Protocol

Whitepaper V2.0

Jan, 2019

This article is used for reference only, it cannot be treated as an offer to trade securities in any administrative region.

Abstract

By January of 2018, the overall global value of cryptoassets has reached to $800 billion. In addition to cryptocurrency exchange, cryptotokens mortgage loan is another best way to get liquidity for cryptoassets holders. However, as different platforms of cryptoasset loan are based on different rules, rare liquidity among cryptoassets loan services exists, user experience improvement is in urgent need. Based on EOS Blockchain platform, The Force Protocol team will develop cryptocurrency loan protocols and standards in different scenarios. By converting cryptocurrency borrowing request into tokens exchange-like trading, order format standardization is realized. Stablecoins can cooperate with loan services. To make full use of stablecoins, Dual Token Model is proposed here, which can overcome shortcomings of existing stablecoins economic models. With this model, policy risk can be removed by introduction of supervision from third party custodian bank and auditing institution. In addition, sustainable full currency reserve mechanism is proposed for the first time. The Force Protocol will open source, and its instructions are encouraged to be called through API by third-party applications (supernode), with the purpose of providing protocol layer services for different cryptocurrency loan applications around the world. Moreover, liquidity and depth of loan platform are also expanded.

Reading Tips：

In this whitepaper, our initial plan is described, which is to develop a new underlying protocol based on blockchain, original token EFOR will be used on the platform of The Force Protocol. Any comment in this article can neither be considered as commitment or promise of ways to develop business, platform and token, nor effective or value of them. The plan outlined in this white paper can be appropriately adjusted according to actual project development needs and external factors encountered. The project will inevitably be affected by varies of objective factors, such as market turbulence, policy risk and cryptocurrency industry factors, etc. Any description of future events is based on the analysis of problems by The Force Protocol described in this document.

This white paper cannot be treated as an offer or sale agreement for token or any other token purchase mechanism, such as but not limited to simple agreement for The Force Protocol token. This white paper only serves as document for community fans to understand the content and problems related to The Force Protocol, The Force Protocol Token, and The Force Protocol ecology. When comes to investment decisions, investors should read this white paper combined with other established quotation documents. The swap or donation of the original token is affected by many potential risks, contributors of The Force Protocol should be aware that risk of losing all or part of their value exists.

Content

1 Summary

1.1 Introduction

Blockchain technology such as cryptocurrency defines a new production relationship for the world. In the new relationship, benefits of production activities in all aspects of digital production are shared, without the need for any centralized organization to formulate distribution rules and intercept most of the revenue. The development of blockchain technology represented by Ethereum, EOS, NEO, and other public chains makes the generation and application of cryptotokens easier and more convenient, which accelerates the tokenization process of physical assets or cryptoassets. In the past few years, the total market value of global cryptoassets continues to increase. By January 8, 2018, the global marketable cryptoassets (including cryptocurrency and Token) generated by blockchain technology reached $830 billion. It means that a large economy, which connecting the world and all mankind is on the way. Cryptoasset loan is the most basic financial demand in the cryptoeconomy.

Although there has been many DAPPs for loan at the time, such as ETHLend, SALT, CRED, etc. Most demands in the loan market still are not fully met. Several factors leading to the situation: First, decentralized loan platforms couldn't solve the problem of cross-chain token value exchange, while centralized ones have risk of security and trust. Second, loan demands are divided by several independent Apps with no information shared among each other, which becomes obstacle to development of cryptoeconomy. Third, although stablecoin/stabletoken like USDT are widely used to offset the effects caused by price fluctuations in cryptocurrencies, Tether's opaque financial and the fact of not being endorsed by the US government makes USDT under the risk of uncertainty, which further affects the development of cryptocurrencies market.

Based on these considerations, **The Force Protocol** is developed, which supports decentralized loan platform construction, orderbook sharing among decentralized

platforms around the word, and distributed stablecoin system endorsed by governments. Moreover, strategy of anti-money laundering and measures for bursting lost reduction is also designed.

1.2 Targets of Our Work

1.2.1 Global Network of Token Loan

We develop something important to break the process-heavy and unfair loan market we all know. We want to make loan available and reachable on global scale, without information islands among DAPPs. Instead of relying on traditional loan institute, we believe borrower should have the possibility to obtain loan anytime and anywhere in the world. With our protocol, Institutes can set up their own loan platform/supernode, which connects to a shared orderbook maintained by other supernodes. Anyone can participate in this system easily and securely.

1.2.2 Simplified User Experience

The Force Protocol is devoted developing a user-friendly demonstration interface for the above mentioned DAPPs, which make user experience simpler. The loan process is simplified from tens of steps into 2-3 steps process. Additionally, instead of requiring users sending gas before trading to finance the loans, “Approve/Lock”-functions will be used to collect these assets from user’s wallet eliminating unnecessary steps. Which means, less resource will be required.

1.2.3 Decentralized Loan and Supervised Dual Token Model

The Force Protocol provides a global decentralized loan platform based on a shared orderbook. Dual token model is introduced to remove obstacles among different token and fait currency. In addition, trading across different chains is realized using a centralization-like strategy.

1.2.4 Lender Side Offers and Risk Shared Strategy

The Force Protocol will develop loan offers that users can place on the decentralized

application. Loan offers are placed by stating liquidity amount and on what price the lender is willing to provide the liquidity. In the updated version, lender also indicates which tokens would be accepted as collateral or which credit rating must borrower met. In the future, The Force Protocol will introduce option for crowd loan where the risk is shared amongst multiple lenders.

1.2.5 Insurance Measures for Market Volatility

The Force Protocol will use price feeds for popular tokens to adopt sophisticated collateral management where in case the collateral value (value of the tokens) drop below a certain threshold, borrower must fulfil collateral or lender can call the collateral and sell it on market. Alternatively, insurance against market fluctuations is introduced to compensation for extreme situations.

1.2.6 Artificial Intelligence and Big Data for Prediction

Credit history is born once there is data available on paid loans, the interest rates, possible collateral, and details of lenders and transactional data of certain address. Based on this data, artificial intelligence (AI) and EOS distributed ledger data technology can be used to prediction, which could enhance credit loans in the future.

1.3 Contributions

1.3.1 APIs for Loan Platform or Individual

The Force Protocol will provide APIs for institute to set up loan platforms/supernode easily. The only thing, which supernode must do is reduced to build up a website with database to call the APIs. In addition, individual users can complete a sole trading easily with APIs of our protocol, which can provide guarantee for both lender and borrower.

1.3.2 Stablecoin Endorsed with Fiat Currency Reserve

In order to improve shortcomings of stablecoin economic model such as USDT, GUSD and PAX, we introduce a new stablecoin generation mechanism, which is issued by supernodes (DAPPs). As the provider of fund management service, the supernode signs

investment agreement with investor, and receives investor's fiat funds into third-party custodian bank. At the same time, supernode issues corresponding quantity of stablecoins in the Ethereum network, all these processes will be supervised by custodian bank and auditing institution. Stablecoin in The Force Protocol has fully fiat currency reserve. The use of stablecoin in our protocol will bring in huge amount of incremental funding, which will promote development of cryptoeconomy.

1.3.3 Shared Orderbook

All the supernodes (DAPPs) based on our protocol around the world shares one orderbook, where great aggregation effect can be significantly released. Based on the shared orderbook, more works can be devoted to leverage service quality, rather than distracting attention to construction of loan platforms. Therefore, better service will be provided by supernodes after our protocol is online.

1.3.4 Insurance Product for Emergency Fluctuation

To avoid enormous lost from mandatory liquidation, insurance product will be designed for lenders. When mandatory liquidation occurs, lenders who have bought the insurance service will get compensation from the insurance pool.

1.3.5 Example Instance of Our Protocol

Finally, an example DAPP platform (Supernode) instance will be developed, with which several goals will be achieved: First, as the first DAPP based on our protocol, users can lend/borrow tokens on the platform. Second, the DAPP can be used to test the performance of our protocol in a long time, which is very important for the improvement and update process. Finally, the organizations (relayers/ supernode) will get an intuitive impression on the usage of our protocol.

1.4 Benefits of The Force Protocol

1.4.1 Decentralization

Decentralization provides more security and trust by design, where the trust is

eliminated by executing code that does not require central management or servers. Decentralization change dramatically the architecture of loan. Decentralization also means that borrowers and lenders do not need to trust anyone once the smart contract is deployed. These smart contracts provide a trustless and transparent loan environment, which is not available on fiat loan market.

1.4.2 Trustless

Trust between trading counterparties are based on codes. When the lender/borrower places the lend/borrower request on DAPPs (supernode) based on our protocol, the counterparty, the DAPP (supernode), the protocol or any other party cannot manipulate, stop or prevent the loan request once the corresponding smart contract is deployed. Instead of the need to trust the counterparty, decentralization removes the necessity to trust your counterparty. Removing risk from the counterparty or third party is vital to avoid any unfair and unwanted behavior. By trustless environment, we are able to avoid risk that are associated with third parties. With the protocol, we do not have to consider whether service provider is under cyberattack, incurs fraud or the service provider would end up into insolvency proceedings.

1.4.3 Transparency

Blockchain provides a ledger, which is open for inspection from transaction to transaction. It means that every transaction is recorded and any transaction which is deployed on Ethereum blockchain could be explored though block explorers. Transparent ledger removes the trust that normally one would need to have when making a transaction between two banking institutions. Transparency brings more power to finance industry. When it comes to loan, timing is vital. Loan capital must move between borrowers and lenders as fast as possible in a global scale. Today, current banking system does not provide such tools for the loan market, while cryptocurrency and blockchain technology fulfil this need. Moreover, our protocol provides decentralized loan possible in the crypto-world without being exposed to loss of loan capital. The lender is always able to check whether borrower received the loan and vice versa. No trust needed.

2 Stablecoin Endorsed with Fiat Currency Reserve

In recent years, with development of cryptocurrency technology, purchase of goods or services using cryptocurrencies such as BTC, EOS and ETH has gradually been accepted by the public. Due to price fluctuations in cryptocurrencies such as BTC, exchange rate conversion is required at the time of payment, which brings certain inconvenience to users. In addition, holders of top cryptocurrencies such as BTC, EOS and ETH tend to hold them for price increase more than use them as currency. This idea inhibits use of top cryptocurrency such as BTC in people's daily life.

Stablecoin represented by USDT are widely accepted by investors and exchanges of cryptocurrency during 2017 to 2018. This phenomenon shows demand for stablecoin exists in cryptocurrency investors. However, due to Tether's opaque financial and the fact that it has not been endorsed by the US government, position of USDT is relatively embarrassing. On the one hand, people have to widely use USDT on the demand for stablecoin. On the other hand, once there is a stablecoin which technology and policy background are better than USDT, there is no doubt that a large-scale replacement will happen. Since then, coins such as DAI, TUSD, and USDC have been launched, which shows the industry's exploration of improving mechanism of stablecoins.

After years of development, whether investment products represented by Bitcoin and ETF can be approved by governments mainstream countries, determines whether huge amount of funds in traditional financial systems have legal channels to enter cryptocurrency investment activities. The regulatory trends of major countries affect the trends of various interest groups such as investors, exchanges, and project groups. In September 2018, GUSD and PAX, two kinds of stablecoin which regulated by the US financial system, were widely interpreted as a major breakthrough in the field of cryptocurrency and stablecoin. With stimulation of this breakthrough, other stablecoin projects are rapidly induced to the industry, which indicated the potential best development orientation of stablecoin.

We believe that since the stablecoin needs to be linked to fiat and maintain a 1:1

exchange rate, its issuance must be under the supervision of governments. Completely decentralized issuance of stablecoins is hardly to be recognized by governments around the world. There are also shortcomings in the mechanism for seeking stable prices by algorithm, which makes widely applied of algorithm backed stablecoins at a large scale impossible. From the perspective of the issuance mechanism of stablecoins such as GUSD and PAX, as the demand increases, its issuers must constantly invest and lock fiat to keep the stability of coin price. The total amount of fiat which single stablecoin issuer such as Gemini exchange could invest in are limited. Once the demand side exceeds its release capability, de-anchor to fiat like the USDT will most likely occur. In addition, in order to maintain its own business operations, the issuer has a strong motivation to misappropriate fiat which should be locked. Therefore, under the existing issuance mechanism, in the long run, as the demand side continues to increase, stablecoins like GUSD and PAX may also have a system collapse.

What kind of stablecoin issuance mechanism is more reasonable? Take issuing mechanism of fiat as referral, the main tools such as Open Market Operations and Rediscount are used by central bank to control currency offering. In the United States, for example, the Federal Reserve offers US dollars to government through purchase of US government bonds. The US government then offers liquidity to the economy through various direct Investment and procurement channel, these operations boost economic growth. Commercial banks obtain USD from Federal Reserve through channels such as rediscount, enterprises and individuals obtain fiat through corporate bonds and personal loans, thereby promoting circulation of fiat within the economic system. The issuance of US dollar is backed by credit of the US government, while ensuring global circulation of US dollar with its national strength and credit, which constitute main mechanism for issuance and circulation of US dollar. It’s very clear that credit has a decisive role in the creation of money in modern society. As a stablecoin in the cryptocurrency system, its issuance mechanism can also base on the obligatory right to fiat.

Therefore, the loan business is naturally linked to stablecoin business. In order to

improve shortcomings of stablecoin economic model such as USDT, GUSD and PAX, we introduce a new stablecoin issuance mechanism here.

In the system of The Force Protocol Stablecoin, there are three parties involved, including supernode, investor and borrower, their roles are:

2.1 Supernode of The Force Protocol

The supernode is a financial service provider based on The Force Protocol's global shared loan order book in various countries and regions. Under the premise of compliance, it conducts business such as fiat or cryptocurrency loan, payment, transaction, and clearing. The shared loan order book between supernodes can be understood as a consortium blockchain, joining and exiting of supernode needs permission through community governance process. Supernode may be a DAPP, a cryptocurrency wallet, a centralized exchange, a financial license holder, etc. All supernodes must obtain the cryptocurrency to fiat exchange license issued by their local government and accept government's compliance supervision. At business level, Supernodes need to interface with The Force Protocol global loan protocols, third-party custodian of fiat and cryptoassets, investors, borrowers, regulators, and more.

Stablecoin within The Force Protocol system is issued by supernodes. As the provider of wealth management service, the supernode signs investment agreement with investor, and receives investor's fiat funds into third-party custodian bank. At the same time, the supernode issues corresponding type and quantity of stablecoins in the Ethereum network. Fiat received by supernode is supervised by custodian bank and auditing institution. The amount of stablecoins issued by supernode is transparent within the blockchain network, mechanism of both fiat and crypto channel fundamentally guarantees that each stablecoin in The Force Protocol system has corresponding fiat asset backed. In addition, due to the long-term stable income from the fund management services provided by supernodes, fiat funds in the traditional financial system will flood into the loan system of The Force Protocol. In theory, the amount of stablecoins in The

Force Protocol system can be infinitely close to the amount of global fiat currency issued. Such a mechanism will also bring huge incremental fluidity to the field of cryptocurrency.

When investment agreement expires, if investor chooses to redeem wealth management product and withdraw the fiat currency, supernode will recycle stablecoin of the corresponding amount and burn them, thereby maintaining the stability of coin price. Besides the wealth management service, supernode will also provide a small retail exchange service to support the two-way exchange of fiat and stablecoin. Similarly, the fiat currency that supernode obtains from retail customers will be deposited in a third-party custodian bank to ensure compliance and security. This is another mechanism of stablecoin issuing and buring within The Force Protocol system.

Stablecoins in The Force Protocol system will be used for borrower's mortgage, them can also be used for circulation within the crypto-economy. After borrowers obtain stablecoin, they can exchange it into fiat from supernode, borrowers will also directly use stablecoins in scenarios such as exchanges and encrypted payment, thereby promoting growth of the stablecoin ecosystem.

Supernode's profit model includes:

- Order Management Fee: In a typical loan order, the borrower's cost includes loan interest, smart contract execution resorce and order management fee, where the order management fee is the supernode's income.
- Currency exchange fee: If the borrower needs to convert borrowed stablecoin into fiat currency, it needs to operate through the supernode compliance process. In theory, the supernode’s compliance exchange service will be the only fiat currency conversion channel in The Force Protocol system.
- Investor Financial Plan Management Fee: Investors aim to obtain a stable return by lending their fiat or cryptocurrency. Supernodes can provide investors with 6 months to long-term wealth management plan products about fiat currency

investment. Fiat currency obtained by these products will be converted into stable currency and invested in the loan order within The Force Protocol system. These investment orders enjoy the depth and speed of transaction, also risk protection provided by The Force Protocol. The wealth management plan products replace investors to manage their funds and effectively invest them in the loan market to earn a sound return. Within The Force Protocol, all collateral assets are determined and endorsed by the blockchain, which is safer and more efficient than traditional P2P financing.

- Compulsory Liquidation insurance income: Due to high liquidity and volatility of the cryptocurrency market, the borrower's pledge may face a sharp decline in value in a short period of time, although The Force Protocol has set up a pledge replenishment mechanism and an emergency liquidation mechanism. In extreme cases, the system may still be too late to carry out risk control operations, resulting in compulsory liquidation of the pledge. If the value of pledge is not enough to cover investor's input, the investor will suffer losses. In order to cope with extreme situation, The Force Protocol system will use the platform token FOR to design a platform-level solution for the function of pledge blow up insurance. In-depth optimization about parameters of the compulsory liquidation insurance will be carried out with the increasing number of loan orders to better protect investors. Benefits of insurance product will be important source of income for supernode.

2.2 Investor

The investors in The Force Protocol ecosystem include, but are not limited to, high net worth individuals, institutions, long-term funds, family funds, etc. Those investors hold large amounts of fiat or cryptocurrency in various countries and regions, they hope to obtain stable and value-added wealth growth. Investors are willing to accept medium to long-term wealth management plans, with low risk appetite, and hope to avoid losses as much as possible. These requirements are just met by the loan business supported by

blockchain. In The Force Protocol system, the collateral is guaranteed by smart contracts and has very strong risk control capabilities.

Investors can benefit from investment through two channels: the fiat currency channel and the cryptocurrency channel. For the fund side, only fiat currency is allowed to be withdrawn from the fiat currency channel, and the cryptocurrency withdrawal only from crypto-channel. After the fiat currency be saved in the third-party custodian bank, supernode will use smart contract which The Force Protocol offers to generate corresponding amount of stablecoin for loan orders within The Force Protocol system. Investors cannot directly obtain stablecoins after they purchase wealth management products. For investors, the products provided by supernodes are regular fund management plans, which are designed to obtain interest and principal appreciation.

2.3 Borrower

Borrowers are the source of credit for The Force Protocol system. They are distributed globally and holding cryptocurrency. They hope to obtain fiat or stablecoin in the case of temporary transfer their ownership of cryptocurrency. Borrower can be a natural person or a business entity, or even AI in smart contract. Through mortgages of cryptoassets, borrowers will receive funds for real economy consumption or investment, or for cryptocurrency market speculation. The group characteristics of borrower can be summarized as: recognition of the cryptocurrency ecology, willingness to hold cryptocurrency for a long time; short-term capital turnover is difficult, optimistic about rise of cryptocurrency market and unwilling to sell coins; seeking reasonable capital borrowing costs.

2.4 Arbitrator

Individual or team elected by the community of The Force Protocol, has a certain period of time and will be re-elected upon expiration. The arbitrator's election and employment mechanism will be initially submitted to the community by The Force Protocol

development team at appropriate time, after amendments are made, these mechanisms will be officially implemented through community voting. When supernode's EFOR loan rate is insufficient, the monitoring smart contract will inform the supernode to supplement the pledged EFOR. When the supernode fails to replenish EFOR within specified time, the smart contract will feed back relevant information to arbitrator, arbitrator will judge the supernode's ability to continue operation according to the rules. If arbitrator judges that the supernode cannot provide loan service normally, the judgment result will be submitted to The Force Protocol Community Governing Body to apply for the removal of the corresponding supernode from The Force Protocol system and start any investor assistance program needed.

2.5 Other Participants

Parties involved in the issue of stablecoin in The Force Protocol system also include third-party custodian banks, government regulatory agencies, audit institutions and other off-chain entities. Their existence can guarantee parties within The Force Protocol system are conducting business under compliance and maintain system stability.

3 Technical Specifications

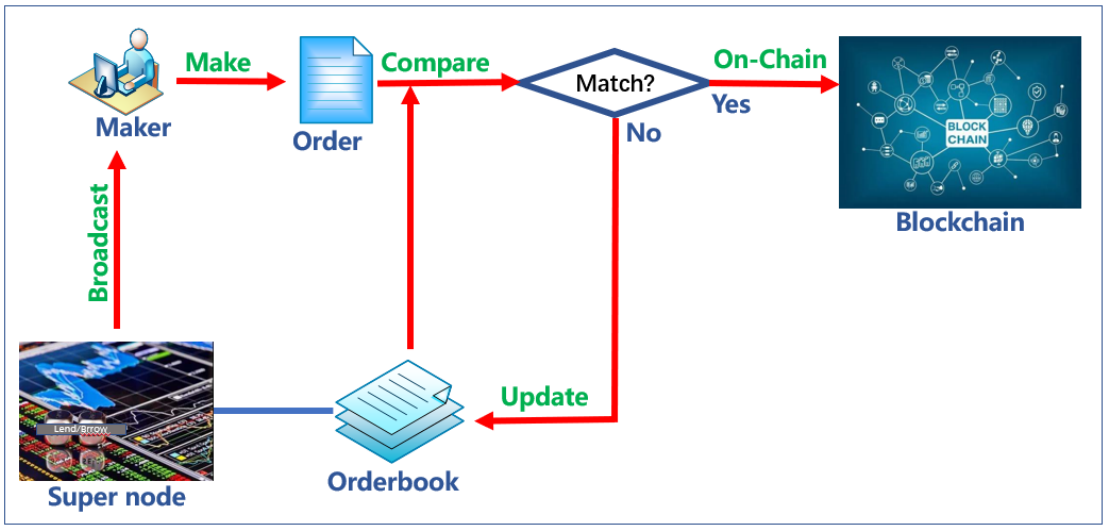
3.1 Public Trading Process

For liquid markets to emerge, there must be public locations where users can post orders that are subsequently aggregated into Orderbooks. The protocol we have described allows any institute who has qualification to be a loan platform, maintain a Shared Orderbook and charge transaction fees on all resulting liquidity. We refer to entities which host and maintain an Orderbook as ***Supernodes*** rather than platforms, while the orders of this kind are named pubilc orders. ***Supernode*** merely facilitates signaling between market participants by a Shared Orderbook that consists of generic messages. ***Supernode*** does not execute trades on behalf of any market participant as the process would require market participants to trust them. Instead, takers execute their own trades.

The message format for broadcast orders do not specify a taker address, allowing a broadcast order can be filled by anyone manually or automatically. Moreover, a pubilc order contains parameters including *feeLender*, *feeBorrower*, *addrSupernode* and *addrFeeRecipient*, which specify transaction fee values and address used by the specified Supernode to collect transaction fees. These fees will be transferred to *addrFeeRecipient* when an order is filled. Here, the initiator of a trading is named ***Maker*** while the one who fills the order manually is named ***Taker***. A trading can be initiated/filled by any one, no matter they are lenders or borrowers. Here, both cryptocurrencies such as BTC, EOS, ETH et al. and tokens are referred to as ***token*** with no distinguish for convenience.

Previous work: Supernode publishes loan rules which contains parameters like collateral rate et al., and fee schedule, address *addrFeeRecipient* used for collecting transaction fees are also broadcasted.

3.1.1 Order Making and Automatic Matching



Step1: Maker approves the decentralized smart contract the right to transfer specified amount loan tokens or collateral tokens.

Step2: Maker creates an order, which specifies type and number of loan tokens, type and number of collateral tokens, loan period, APR, expiration time of the order. In addition, *feeLender or feeBorrower* together with *addrFeeRecipient* are set to receive

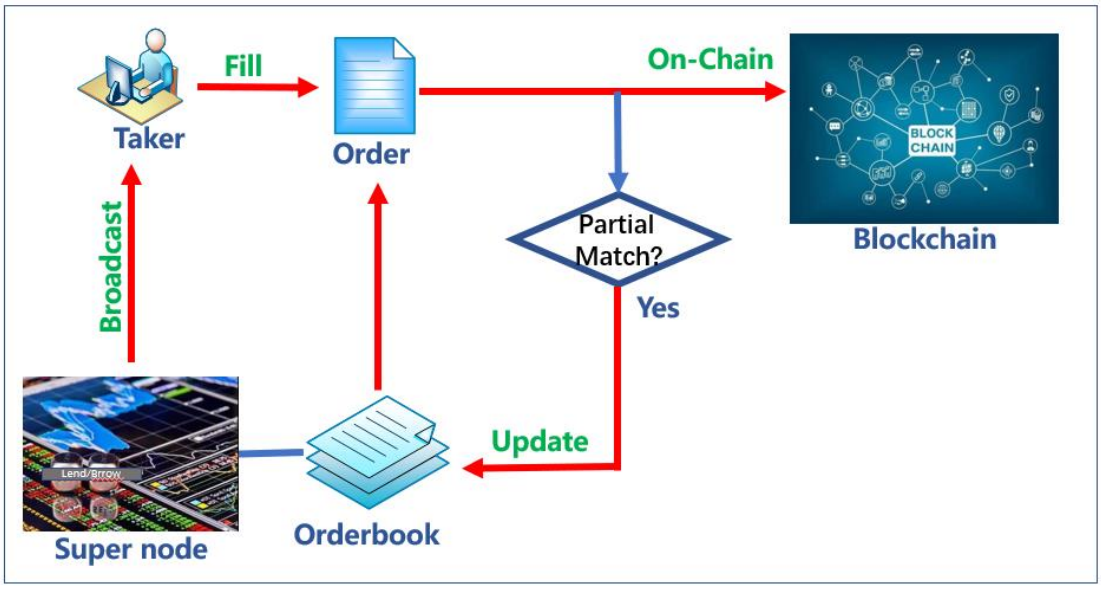
by Supernode. Then, maker signs the order with its private key.

Step3: Supernode receives the signed order, whose validation is then checked. If the order meet Supernode's requirements, the order is accepted as **orderNew**, otherwise it will be rejected.

Step4: The Supernode traverses orders in Shared orderbook to compare with **orderNew,** while validation of each order is also checked. (1) If there was no order satisfies conditions, orderNew will be updated to the Shared orderbook. (2) If multiple orders match orderNew, the earlies one will be locked. (3) For partially match cases, the compatible orders are locked in sequence of compatibility until no orders matches. The locked order which named **orderMatch** is removed from the Shared orderbook. The unmatched portion of any party will be updated to the Shared orderbook. Here, match means borrowing rate greater or equal to lending rate.

Step5: OrderNew and orderMatch are sent to smart contract for trading process.

3.1.2 Manual Matching



Step1: Takers receive the newest block of the Shared Orderbook from any Supernode.

Step2: Taker fills one order in the orderbook named **orderMatch**. A new order begins to construct automatically, which specifies number of tokens needed, loan period,

annual interest rate, expiration time of order, etc. Taker approves the decentralized smart contract the right to transfer specified amount tokens as required, and signs the order, which is then named orderNew.

Step3: Supernode receives orderNew and orderMatch, whose validation are been checked. If both the two orders meet Supernode's requirements, OrderMatch is removed from the Shared orderbook, otherwise both of orderNew and orderMatch will be rejected.

Step4: OrderNew and orderMatch are sent to smart contract for trading process.

3.1.3 On-Chain Process

Step1: A decentralized smart contract instance is constructed, with parameters contained in OrderNew and OrderMatch.

Step2: The decentralized smart contract locks collateral tokens.

Step3: The decentralized smart contract transfers tokens from lender to borrower.

Step4: The contract begins to take effect

3.1.4 Repayment Process

When expiration time of the loan contract is reached, the repayment process will be triggered. The process goes as follows:

Step1: The Supernode based on our protocol notifies its users that contract expires through SMS, email, message in APP, phone call, etc.

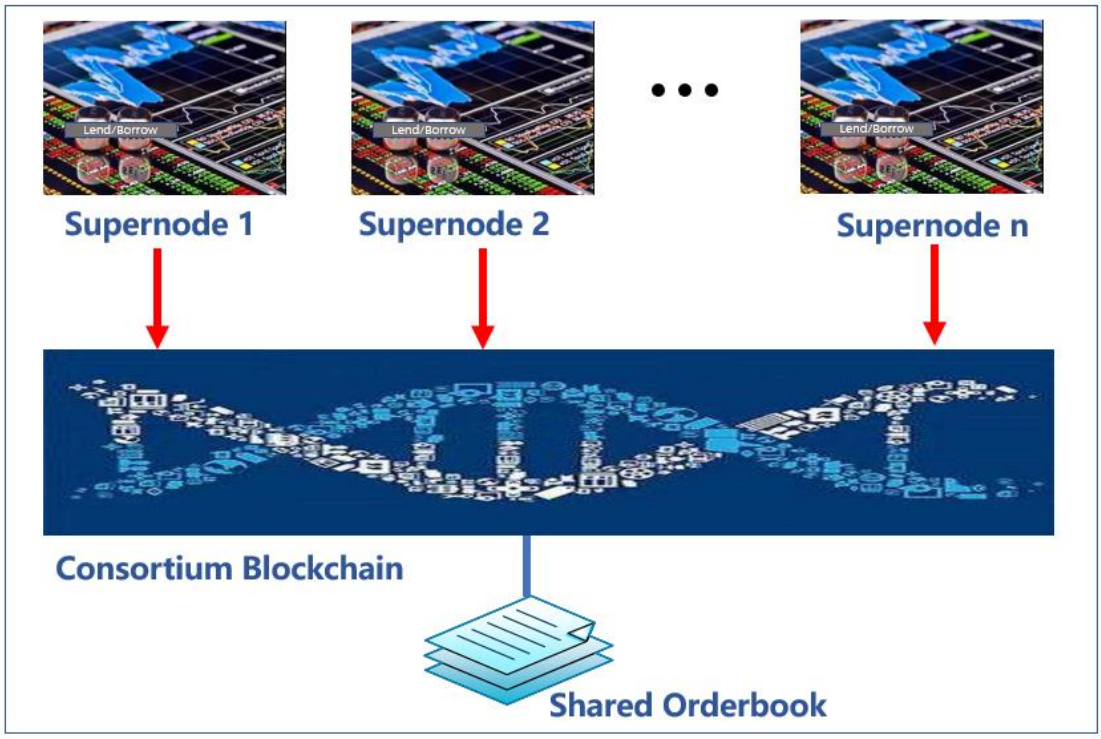
Step2: Borrower transfers specified number of borrowed tokens and interest to the decentralized smart contract.

Step3: The decentralized smart contract transfers obtained tokens and interest to lender.

Step4: The decentralized smart contract transfers collateral tokens to the borrower.

3.1.5 Shared Orderbook

In our protocol, Shared orderbook is achieved by consortium blockchain, which are maintained by all the Supernodes. Specifically, parity with Aura technology is used to construct the consortium blockchain.



In our protocol, each Supernode has a single account named validator. Whenever a Supernode join in the protocol, a corresponding validator is added into the Shared validator list. The smart contract can set up strategies for determining the validators' qualification. Timestamp is configured in genesis block, and one legal validator is selected every 3 seconds to build, validate, sign and broadcast a block. Legal validator is determined by the following formula.

index= (UNIX_TIMESTAMP/BLOCK_TIME) % NUMBER_OF_ VALIDATORS;

When orderbook updating happens, local validator will check validation of the instruction. If the related order has been locked before, the instruction will be rejected. Otherwise, local validator will broadcast the order to the legal validator for package. If more than one updating instruction related to one order, only the earliest one is accepted. All the accepted instructions and the resulting orderbook status are packaged in the block, which is then be broadcasted to all other validators. If the update is accepted by

51% of validators, the order is confirmed. When a block produced by a legal validator is received, it will be accepted by all validators. Otherwise, it will be rejected. If there was no orderbook updating in a period, no blockchain will be produced.

3.2 Peer-to-Peer Trading Process

3.2.1 Loan Process

Over-the-Counter (OTC) order allow two parties to match trading using any communication medium. Then the FOR protocol is opted to approving the process for both lender and borrower. The process goes as follows:

Previous work: A deal is made by lender and borrower using any communication medium, internet for example. An order is signed by the maker, who will fill all the parameters required according to standard format defined by The Force Protocol.

Step1: Borrower(A) approves the decentralized smart contract to lock specified balance of Token(A) for collateral.

Step2: Lender(B) transfers specified amount of Token(B) to the decentralized smart contract.

Step3: The decentralized smart contract transfers Token(B) obtained in Step2 to borrower(A).

Step4: The decentralized smart contract approves the validation of the order.

3.2.2 Repayment Process

When an order's expiration time is reached, the repayment process will be triggered. The process goes as follows:

Step1: The lender(B) notifies Borrower(A) to repay Token(B).

Step2: Borrower(A) transfers specified amount of Token(B) to the decentralized smart contract.

Step3: The decentralized smart contract transfers Token(B) obtained in Step2 to lender(B).

Step4: The decentralized smart contract unlocks collatered Token(A).

3.3 Loan Order Rules Configuration

3.3.1 Interest Rate Discovery Mechanism

The loan interest rate in the system will be determined by loan participants together with market supply and demand, and each additional loan/loan order here will also affect market supply and demand. That is, reference rated will be provided by supernode, and the final rate will be decided by participants. The market interest rates here are annualized interest rates, and the interest rates in other cycles will also be converted into annualized interest rate.

In order to avoid interest instability caused by malicious pending order, the system will check the validity of order. User that repeatedly places and cancels order will be frozen for certain period. In addition, the system will limit the range of interest rate according to market fluctuations, such as the change in interest rate cannot exceed 50% within 10 minutes. More calculation rules and conditions of use will be investigated and confirmed during the business development process.

3.3.2 Loan Cycle

The Loan cycle of a trading is decided by both Supernode and traders. That is, the Supernode will provide a list of loan cycle options for makers to select. When the order initiated by the maker is updated in orderbook, the filler will select the most suitable one to fill in.

3.3.3 Collateral Rate and Compulsory Liquidation Line

To control the risk of collateral loans, the types of collateral tokens, collateral rates, and

compulsory liquidation line are all set by the protocol development team. All these parameters are stored in a smart contract, where modification of each parameter should be agreed by the key develop members. When the community matures in the future, the modification proposal will be submitted to DAO for voting.

In The Force Protocol (Version1.0), market value of cryptoasset token and the 24-hour turnover are selected as criterion to judge select mortgage token. Later, we will continuously update and expand the types of collateral tokens based on operation data. At the same time, we will also launch community voting to encourage users to elect mortgage tokens they want.

The collateral rate is defined as the ratio of token available to the present value of mortgage. Generally, the collateral rate must not exceed 70% in traditional loan institution. In the field of cryptoasset loan, there are no clear regulations or historical experience to reference. Considering dramatic changes of cryptoasset in recent years, the collateral rate of most loan platforms is set to 50%-80%. Due to slight difference in price change of the specific token and turnover of 24-hours crypto-exchange, the mortgage rate according to specific market conditions by our platform, which are:

A_n= Amplitude of the n^{th} cycle forward(%);

$A_{predict}=A_1+Max（0，A_2-A_1）/2^{(2-1)}+Max（0，A_3-Max（A_1，A_2））/2^{(3-1)}+...+Max（0，A_n-Max（A_1，A_2，...，A_{n-1}））/2^{(n-1)}$; where n is usually set to 3.

Mortgage Rate=$Min（70\%，1-A_{predict}）$.

3.4 Oracles: Security Mechanism

During effective period of the contract, web crawler is used to get real-time price information of tokens related to the contract. An off-chain computation is executed for safe collateral rate. Whenever the precaution level is reached, oracle is used to provide feeds from social media for creating a complete proof profile. Then warning message is sent to borrower for more collateral. The message will be transferred by SMS

platform or phone call and details of message will be sent to decentralized smart contract for backup. If compulsory liquidation line is reached and no collateral are received, the compulsory liquidation process is activated, which goes as follows:

Step1: Oracle function is called to provide feeds from social media for creating a complete proof profile.

Step2: Supernode calculate the value of principal tokens and interest, equal value of collateral tokens will be deducted remained smart contract. The remains are unlocked and sent to the borrower's address.

Step3: The deducted tokens are sent to the lender.

3.5 Insurance: Emergency Management

(1) Emergency Management

Great volatility is very common in cryptocurrency market, thus extreme situation must be considered previously: when the price of a mortgage token falls rapidly in a short period of time, and the smart contract has no time to enforce compulsory liquidation, and the collateral price has fallen below the liquidation line. In this case, to prevent the risk exposure from expanding, smart contract will automatically perform the mandatory liquidation immediately.

(2) Insurance Option

Whenever mandatory liquidation occurs, the lender will suffer from certain amount of lost. Sometimes, the situation will exceed tolerance limitation of lenders, which will be the biggest obstacle in loan market. Based on this consideration, insurance option will be designed based on our protocol. When mandatory liquidation occurs, lenders who have bought the insurance will get compensation from the insurance pool. More details about insurance product will be demonstrated in the following versions.

(3) Collateralized Tokens Selection

If market price of one token fluctuates significantly, the Force Protocol team (and finally the DAO) has the right to decide to move certain types of cryptocurrencies out of the collateralized tokens and even temporarily close the entire network of mortgage loans. Since our protocol's main business is built entirely on Ethereum (which may be extended to EOS, NEO, and other top public chains in the future), the security and usability of our protocol depend on performance of Ethereum system on a large extent. In the long run, the team will continuously pay close attention to development of the blockchain underlying technology, the whole loan ecology will benefit from prosperity of entire cryptocurrency ecosystem to achieve sustained and healthy development.

4 Message Format

There is a package with standard format for each order, which contains all the parameters required and related signatures. All the parameters are hashed to 32 Byte by Keccak SHA3, and then ECDSA signature is produced. The final format of an order in the smart contract will be stored in a struct variable.

For broadcast orders, Supernode is used to host and broadcast orderbooks, a small amount of fee will be charged. As a result, the message format at user client is a little different.

4.1 Broadcast Order

The order created originally is different from the order on blockchain, as the tasks of orders are different. Here is the order format stored in Supernode. The format of order on-chain will be illustrated in Smart Contract section.

Name	Data Type	Description
addrVersion	address	Address of the smart contract, changes whenever the protocol is updated.

Name	Data Type	Description
nFeeLender	uint256	Fees provided by Lender.
nFeeBorrower	uint256	Fees provided by Borrower.
addrFeeRecipient	address	Address to receive trading fees for Supernode.
addrBorrower	address	Borrower address.
addrLender	address	Lender address.
addrTokenA	address	Address of Token A for collateral.
addrTokenB	address	Address of Token B for lending.
nTokenA	uint256	Total number of Token A for collateral.
nTokenB	uint256	Total number of Token B for lending.
nExpiration	uint256	Time at which the order expires (seconds since unix epoch).
nLendingCycle	uint256	Lending Cycle.
ufMortageRate	ufixed0x256	Mortgage rate.
ufInterestRate	ufixed0x256	Earnings per Token B for lenders (ETH or FOR)
vLender	uint8	ECDSA signature of the above arguments by Lender.
rLender	bytes32	
sLender	bytes32	
vBorrower	uint8	ECDSA signature of the above arguments by Borrower.
rBorrower	bytes32	
sBorrower	bytes32	

4.2 Peer-to-Peer Order

Name	Data Type	Description
addrVersion	address	Address of the smart contract, changes whenever the protocol is updated.
addrBorrower	address	Borrower address.
addrLender	address	Lender address.
addrTokenA	address	Address of Token A for collateral.
addrTokenB	address	Address of Token B for lending.
nNumTokenA	uint256	Total number of Token A for collateral.
nNumTokenB	uint256	Total number of Token B for lending.
nExpiration	uint256	Time at which the order expires (seconds since unix epoch).
nLendingCycle	uint256	Lending Cycle.
ufRate	ufixed0x256	Mortgage rate.
ufInterestRate	ufixed0x256	Earnings per Token B for lenders (ETH or FOR)
vLender	uint8	ECDSA signature of the above arguments by Lender.
rLender	bytes32	
sLender	bytes32	
vBorrower	uint8	ECDSA signature of the above arguments by Borrower.
rBorrower	bytes32	
sBorrower	bytes32	

5 Smart Contract

For our protocol, a cluster of smart contracts are developed to cover the functions needed to provide by our protocol. Several libraries with solidity language will be included to reduce the amount of resource cost in the future.

5.1 Related Library

Generally speaking, most computations related will be computed off-chain, while oraclize is mainly used for providing proof for the result. However, Math library is still included to deal with some simple computation at present or in the future. For space limitation consideration, only some sample functions are listed for demonstration purpose.

```
#include <utility>
#include <vector>
#include <string>
#include <eosiolib/eosio.hpp>
#include <eosiolib/time.hpp>
#include <eosiolib/asset.hpp>
#include <eosiolib/contract.hpp>
#include <eosiolib/crypto.h>
#include <eosiolib/symbol.hpp>
#include <vector>
#include <string>
#include <eosio.token/eosio.token.hpp>
#include <eosiolib/eosio.hpp>

using namespace eosio;

using eosio::indexed_by;
using eosio::const_mem_fun;
using eosio::asset;
using eosio::permission_level;
using eosio::action;
using eosio::print;
using eosio::name;
```

5.2 TheForceProtocol

TheForceProtocol contract represents the main contract responsible for combining needed functionality for trading and risk management. Whenever a trade is submitted to the smart contract, an instance of the contract will be introduced and stored on the public blockchain. There are dozens of structs, functions, libraries, and smart contracts to define for the main contract to achieve its targets, such as structs like tokena_info, asset_type and order, and functions like inittokens(), addtokena(), addtokenb(), borrow(), lend(), closepstion(), cancelorder(), callmargin(), repay(), forcerepay(). Here, require_auth() is used to control access permission. Parts of contracts are displayed as

follows.

```
void inittokens();

void addtokena(const account_name &contract, const string &sn);

void addtokenb(const account_name &contract, const string &sn);

void borrow(const account_name &borrower, const int32_t &lending_cycle,
    const asset_type &token_a, const asset_type &token_b, const int32_t &pledge_rate,
    const int32_t &interest_rate, const int32_t &fee_rate, const int64_t &order_id);

void lend(const int64_t &order_id, const account_name &lender, const asset_type &token_b);

void closepstion(const int64_t &order_id, const account_name &by);

void cancelorder(const int64_t &order_id, const account_name &by);

void callmargin(const int64_t &order_id, const account_name &borrower, const asset_type &token_a);

void repay(const int64_t &order_id, const account_name &borrower, const asset_type &token_b);

void forcerepay(const int64_t &order_id, const asset_type &token_a);
```

Contract code of token increment:

```
//@abi action
void loan::addtokena(const account_name &contract, const string &sn) {

    require_auth(_self);

    auto token_itr = tokena.find(contract);
    if (token_itr == tokena.end()) {
        tokena.emplace(_self, [&](auto &token) {
            token.contract = contract;
            token.tokens = std::vector<string>{sn};
        });
    } else {
        std::vector<const string>::iterator iter = std::find(token_itr->tokens.begin(), token_itr->tokens.end(), sn);

        eosio_assert(iter == token_itr->tokens.end(), "token already exist");

        tokena.modify(token_itr, _self, [&](auto &token) {
            token.tokens.push_back(sn);
        });
    }
}
```

Borrowing contract code:

```
//@abi action
void loan::borrow(const account_name &borrower, const int32_t &lending_cycle,
                  const asset_type &token_a, const asset_type &token_b,
                  const int32_t &pledge_rate, const int32_t &interest_rate,
                  const int32_t &fee_rate,const int64_t &order_id) {

    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr == orders.end(), "order already exists");

    eosio_assert(token_a.asset_.is_valid(), "invalid token-a");
    eosio_assert(token_a.asset_.amount > 0, "token-a must be positive quantity");

    ...//omitted code

    eosio_assert(lending_cycle >= 7, "lending-cycle must be greater than 7 days");

    require_auth(borrower);

    action(
            permission_level{borrower, N(active)},
            token_a.contract, N(transfer),
            std::make_tuple(borrower, _self, token_a.asset_, std::string(""))
    ).send();

    // Store new order
    auto new_order_itr = orders.emplace(borrower, [&](auto &order) {
        order.id = order_id;
        order.borrower = borrower;

        ...//omitted code

        order.deadline = eosio::time_point_sec(0);
        order.status = ORDER_STATUS_PENDING;
    });
}
```

Lending contract code:

```
//@abi action
void loan::lend(const int64_t &order_id, const account_name &lender, const asset_type &token_b) {
    auto order_itr = orders.find(order_id);

    eosio_assert(order_itr != orders.end(), "order not found");

    eosio_assert(lender != order_itr->borrower, "cannot lent to self");

    ...//omitted code

    auto eos_token = eosio::token(token_b.contract);
    auto eos_token_balance = eos_token.get_balance(lender, order_itr->token_b.asset_.symbol.name());

    eosio_assert(eos_token_balance >= order_itr->token_b.asset_, "overdraw balance");
    require_auth(lender);

    asset token_b_fee = order_itr->token_b.asset_ * order_itr->fee_rate * 2 / 10000;
    asset token_b_to_borrower = order_itr->token_b.asset_ - token_b_fee / 2;

    action(
            permission_level{lender, N(active)},
            order_itr->token_b.contract, N(transfer),
            std::make_tuple(lender, platform, token_b_fee, std::string(""))
    ).send();

    ...//omitted code

    orders.modify(order_itr, _self, [&](auto &order) {
        auto deadline = now() + order.lending_cycle * 24 * 60 * 60;//in seconds
        ...//omitted code
        order.status = ORDER_STATUS_ACCEPTED;
    });
}
```

Repay contract code:

```
//@abi action
void loan::repay(const int64_t &order_id, const account_name &borrower, const asset_type &token_b) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");

    ...//ommited code

    asset total_to_pay = order_itr->token_b.asset_ + order_itr->token_b.asset_
                         * order_itr->interest_rate * order_itr->lending_cycle / 3650000;
    print("pay", (order_itr->token_b.asset_ * order_itr->interest_rate
                         * order_itr->lending_cycle / 3650000));
    print(" pay", total_to_pay);

    eosio_assert(token_b.asset_ >= total_to_pay, "insufficient balance");

    //repay
    action(
            permission_level{borrower, N(active)},
            order_itr->token_b.contract, N(transfer),
            std::make_tuple(borrower, order_itr->lender, total_to_pay, std::string(""))
    ).send();

    ...//ommited code

    orders.modify(order_itr, _self, [&](auto &order) {
        order.status = ORDER_STATUS_COMPLETED;
    });
}
```

Force repay contract code:

```
//@abi action
void loan::forcerepay(const int64_t &order_id, const asset_type &token_a) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");

    eosio_assert(order_itr->status == ORDER_STATUS_ACCEPTED, "invalid operation");

    auto time_now = time_point_sec(now()/*current time*/);

    eosio_assert(order_itr->deadline <= time_now,
                 "operation not allowed before deadline");

    ...//ommited code

    require_auth(_self);

    action(
            permission_level{_self, N(active)},
            order_itr->token_a.contract, N(transfer),
            std::make_tuple(_self, order_itr->lender, token_a.asset_, std::string(""))
    ).send();

    action(
            permission_level{_self, N(active)},
            order_itr->token_a.contract, N(transfer),
            std::make_tuple(_self, order_itr->borrower, order_itr->token_a.asset_- token_a.asset_,
                std::string(""))
    ).send();

    orders.modify(order_itr, 0, [&](auto &order) {
        order.status = ORDER_STATUS_FORCEREPAYED;
    });
}
```

Close position contract code:

```
//@abi action
void loan::closepstion(const int64_t &order_id, const account_name &by) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");

    eosio_assert(order_itr->status == ORDER_STATUS_ACCEPTED, "invalid operation");

    auto time_now = time_point_sec(now()/*current time*/);

    if (order_itr->deadline > time_now) {
        eosio_assert(by == _self, "only owner of this contract can do this operation before deadline");
    } else {
        eosio_assert(by == order_itr->lender | by == _self,
                     "only lender or owner of this contract can do this operation");
    }

    require_auth(by);

    action(
            permission_level{_self, N(active)},
            order_itr->token_a.contract, N(transfer),
            std::make_tuple(_self, order_itr->lender, order_itr->token_a.asset_, std::string(""))
    ).send();

    orders.modify(order_itr, 0, [&](auto &order) {
        order.status = ORDER_STATUS_LIQUIDATED;
    });
}
```

Cancel order contract code:

```
//@abi action
void loan::cancelorder(const int64_t &order_id, const account_name &by) {
    auto order_itr = orders.find(order_id);
    eosio_assert(order_itr != orders.end(), "order not found");
    eosio_assert(by == order_itr->borrower || by == _self, "only borrower or contract can operate");
    eosio_assert(order_itr->deadline == eosio::time_point_sec(0) && order_itr->lender == N(none) &&
                 order_itr->status == ORDER_STATUS_PENDING, "order is already accepted");
    require_auth(by);
    action(
            permission_level{_self, N(active)},
            order_itr->token_a.contract, N(transfer),
            std::make_tuple(_self, order_itr->borrower, order_itr->token_a.asset_, std::string(""))
    ).send();
    orders.erase(order_itr);
}
```

5.3 System Security

(1) Dust Attack

When an order is submitted, situation in which large number of dust orders are broadcasting may exist. Thus, during trade matching process, the system will abandon such orders according trading rules, reducing the pressure from Orderbook, so that the dust order will have no impact on the system.

(2) Order Data Tampering

In the process of Orderbook generation, network security problems are involved. Thus, encryption algorithm is performed during data transmission process, which can avoid network data interception and data tampering.

(3) Network Congestion

If information is not limited, excessive network traffic will slow down device response, and network delay occurs. In the process of smart contract submission, the message queue is used to calculate bandwidth speed and concurrent quantity to avoid

unnecessary congestion of the network.

5.4 Contract Risk Control

The global risk control parameters of the loan contract are set as follows:

Parameters	Rules
Mortgage token selection	Historical price fluctuation variance of the tokens, 24-hour turnover, number of listed mainstream exchanges
Mortgage rate	Range of price fluctuation during the loan period, current token price, confidence level
Borrowing Token	BTC, EOS, ETH, USDT, etc.
Borrowing Cycle	7 days, 14 days, 1 month, 2 months, 3 months, 6 months, 9 months, 12 months
Clearance Warning Notice	Total value of mortgage token decreased to 120% of the principal and interest payable, and the mortgaged assets need be replenished in 24 hours after the notice was issued.
Close Position	Total value of the mortgage token fell to 105% of the principal and interest payable
Repayment Notice	Notify 24 hours prior to the time when the contract should be repaid
Overdue	The platform pays the lender's receivable principal and interest, and sells the mortgage token
Penalty for Overdue	5% penalty for received principal and interest

6 APIs

TheForceProtocol.js is the JavaScript API service for our protocol on Ethereum. In specific, TheForceProtocol.js is a collection of libraries which allow you to interact with smart contract when setting up a SuperNode or making a peer-to-peer trade. With the maturity and stability of our protocol, more and more universal APIs will be developed to cover all the functions our protocol provided.

Limited to the space, most of the APIs cannot be described in details. Part of the interfaces will be listed for demonstration purpose. Detailed description and content of

source code will be uploaded to open source community after rounds of rigorous test and audition.

6.1 APIs for Supernode

There are several types of APIs for institutional organization to construct and maintain a Supernode. The functions range from consortium blockchain construction, Supernode construction, Orderbooks updating, to order submission, fiat tokenization and across-chain trading (BTC, EOS, etc.). For space consideration, only interface name with several parameters are listed for demonstration purpose.

```
/*************************************************************/
/*The MakeLendOrder() interface can be used for an lender to make an order on smart contract through a Supernode dApp*/
/*************************************************************/
TheForceProtocol.ETH.MakeLendOrder(
    address      addrVersion,
    address      addrLender,
    address      addrSuperNode,
    bytes32      strTokenLend,
    uint256      nTokenLend,
    uint256      nTokenCollateral,
    uint256      nExpiration,
    uint256      nLendingCycle,
    ufixed0x256  ufRate,
    ufixed0x256  ufInterestRate,
    uint8            vLender,
    bytes32      rLender,
    bytes32      sLender
)
```

```
/*****************************************************************/
/*The MakeBorrowOrder() interface can be used for anBorrower to make an
order on smart contract through a Supernode dApp*/
/*****************************************************************/
TheForceProtocol.ETH.MakeBorrowOrder (
    address       addrVersion,
    address       addrBorrower,
    address       addrSuperNode,
    bytes32       strTokenBorrow,
    uint256       nTokenBorrow,
    uint256       nTokenCollateral,
    uint256       nExpiration,
    uint256       nLendingCycle,
    ufixed0x256   ufRate,
    ufixed0x256   ufInterestRate,
    uint8              vBorrower   ,
    bytes32       rBorrower,
    bytes32       sBorrower
)
```

```
/*****************************************************************/
/*The MakeSuperNode() interface can be used for an lender to make an
order on smart contract through a Supernode dApp*/
/*****************************************************************/
TheForceProtocol.ETH.MakeSuperNode (
    address                 addrVersion,
    address                 addrSuperNode,
    map(address=>uint)      addrReceipt,
    bytes32[]                strReceipt,
    …                       //plenty materials are needed for authorization.
)
```

6.2 APIs for Individual

The following two APIs are provided by The Force Protocol for Peer-to-Peer users. These two APIs can be wrapped by DAPPs, which should be expressed in the forms of command lines or web page. Only two interfaces are listed for demonstration purpose.

```
TheForceProtocol.P2P.lendToken(
    address      addrVersion,
    address      addrLender,
    address      addrBorrower,
    address      addrLender,
    bytes32      strTokenLend,
    bytes32      strTokenCollateral,
    uint256      nTokenLend,
    uint256      nNumTokenCollateral,
    uint256      nExpiration,
    uint256      nLendingCycle,
    ufixed0x256  ufRate,
    ufixed0x256  ufInterestRate,
    uint8        vLender,
    bytes32      rLender,
    bytes32      sLender
)
```

```
TheForceProtocol.P2P.BorrowToken(
    address      addrVersion,
    address      addrBorrower,
    address      addrLender,
    bytes32      strTokenLend,
    bytes32      strTokenCollateral,
    uint256      nTokenLend,
    uint256      nNumTokenCollateral,
    uint256      nExpiration,
    uint256      nLendingCycle,
    ufixed0x256  ufRate,
    ufixed0x256  ufInterestRate,
    uint8            vBorrower,
    bytes32      rBorrower,
    bytes32      sBorrower
)
```

7 Bi Bi Dai 2.0: Demonstration DAPP

In this section, we will demonstrate how DAPP based on our protocol works from the user experience perspective. The platform we developed is named Bi Bi Dai 2.0, which is an upgrade version of a centralized loan platform Bi Bi Dai 1.0. We will go step-by-

step from how to request a loan to fund a loan. In this demonstration instance, you will need to have Google Chrome and the Scatter plugin installed. Scatter is Google Chrome plugin to authorize decentralized application, such as, Bi Bi Dai 2.0, to be run in the web browser without downloading the full Ethereum node.

It should be noted that, the content of interface, operation flow process described later is used for demonstration purpose. It doesn't represent the final form of supernode. The Force Protocol team will develop the platform according to latest research results.

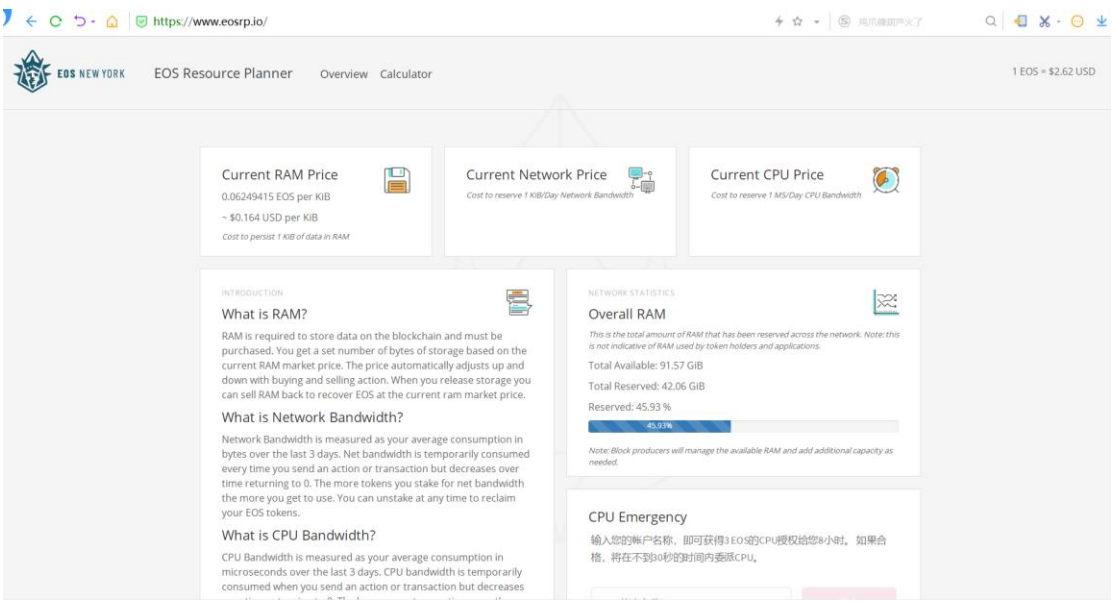
7.1 Preparatory Work

Scatter plugin of Google Chrome browser is required, more details can be found in the official website.

7.2 Demonstration of the DAPP

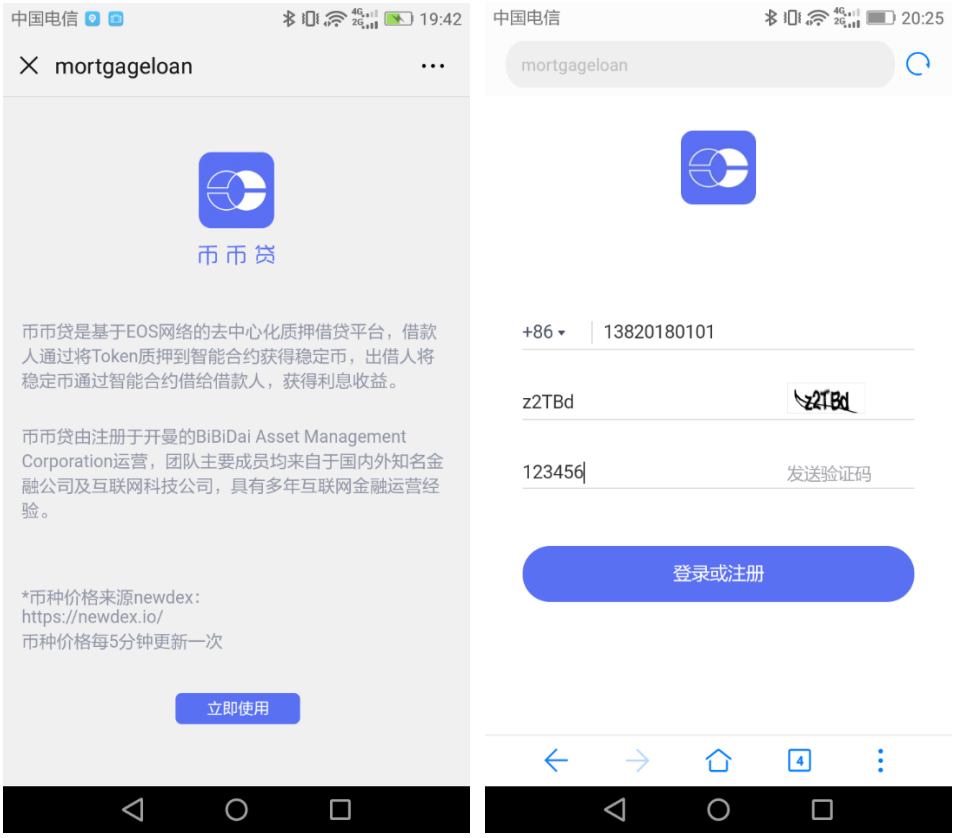
This part is to tell readers about how the decentralized loan services work, as the development of blockchain technology, any part of this content may be different from what is demonstrated here.

Before the borrowing and lending trip begins, users are encouraged to look up the real time price of EOS mortgage and ransom from https://www.eosrp.io/, where the users can get CPU, memory and network bandwidth.



7.2.1 login/ register

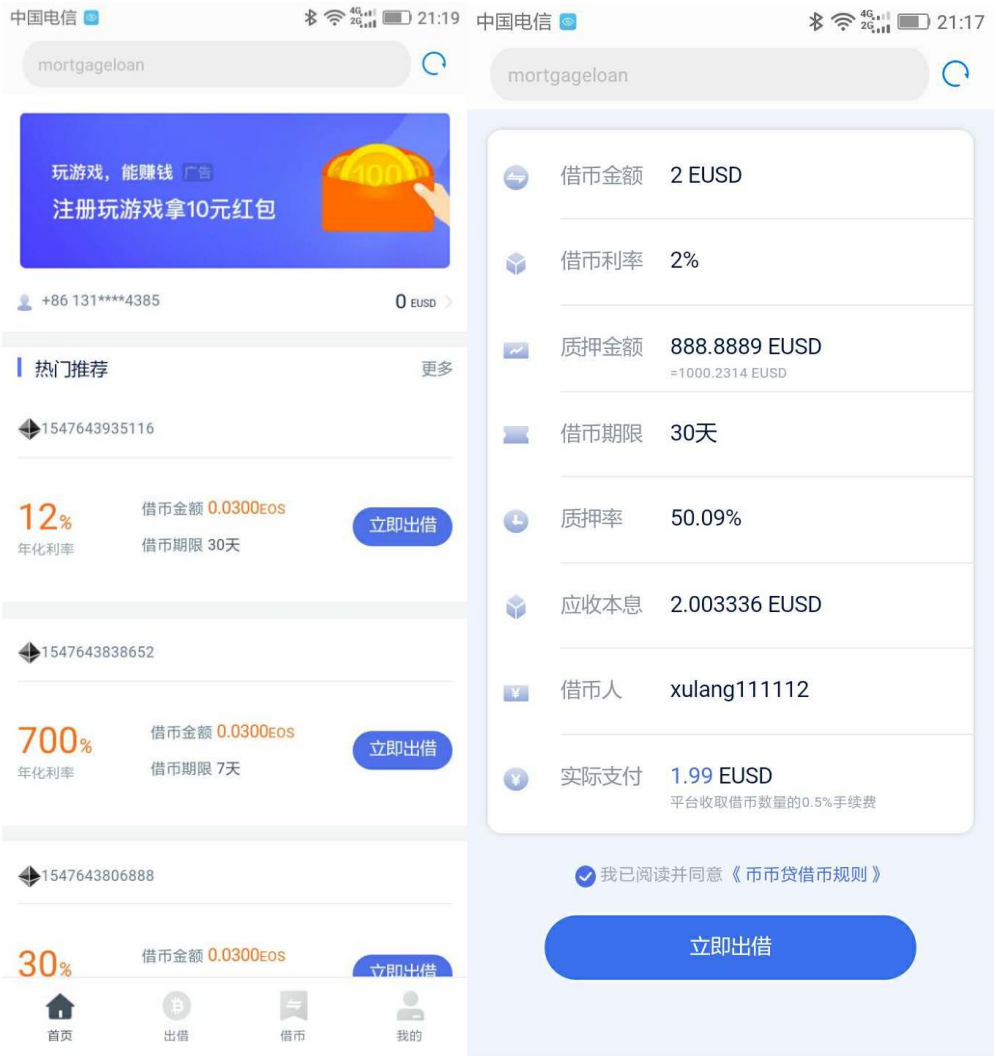
Login the website of http://www.bibidai.com, platform introduction is displayed which is shown in left part of the following figure. Click “Apply It” button, login/ register page is popped out, which is shown in right of the following figure.



Enter phone number and verification code in the figure, then click “send verification code” button, home page will appear.

7.2.2 Home Page

In the home page shown in the figure(left) bellow, users can browse the recommended orders and select proper one directly by clicking “lend immediately”. Lending page will be popped up like right figure below.



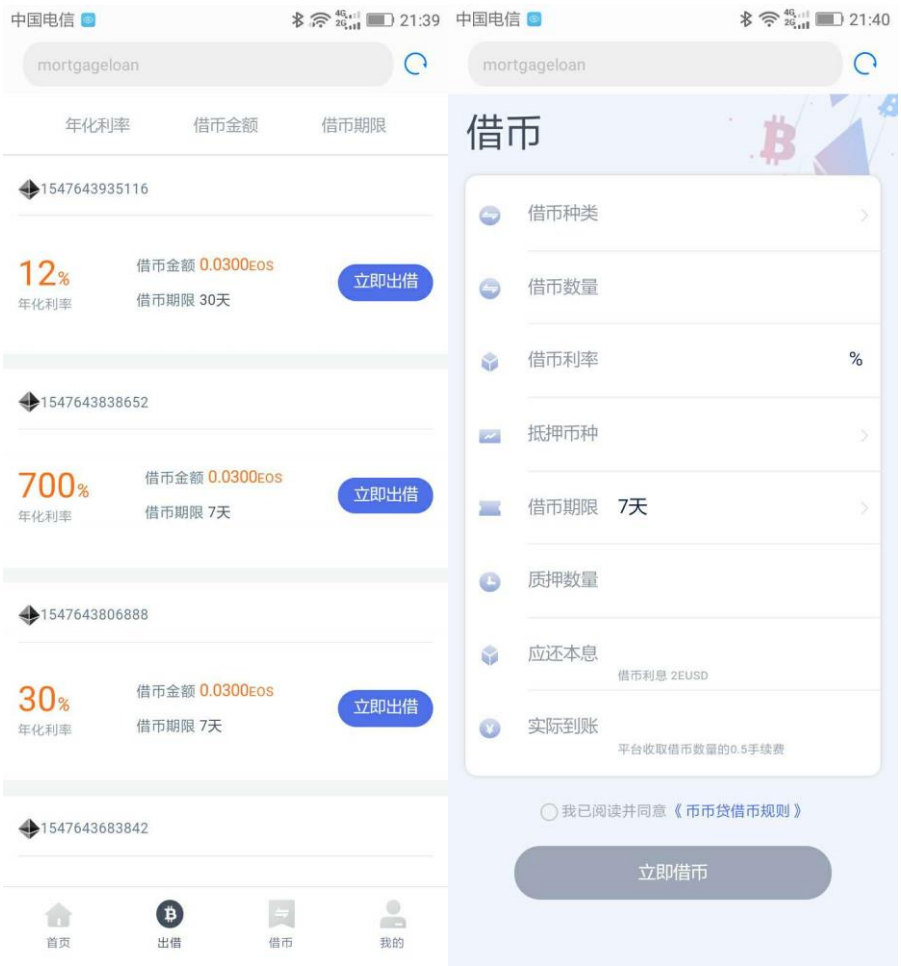
7.2.3 Lending

Lenders can also choose to click “lend” menu at bottom of the home page to browse more lending orders, which is shown in the following figure(left). A lending page will

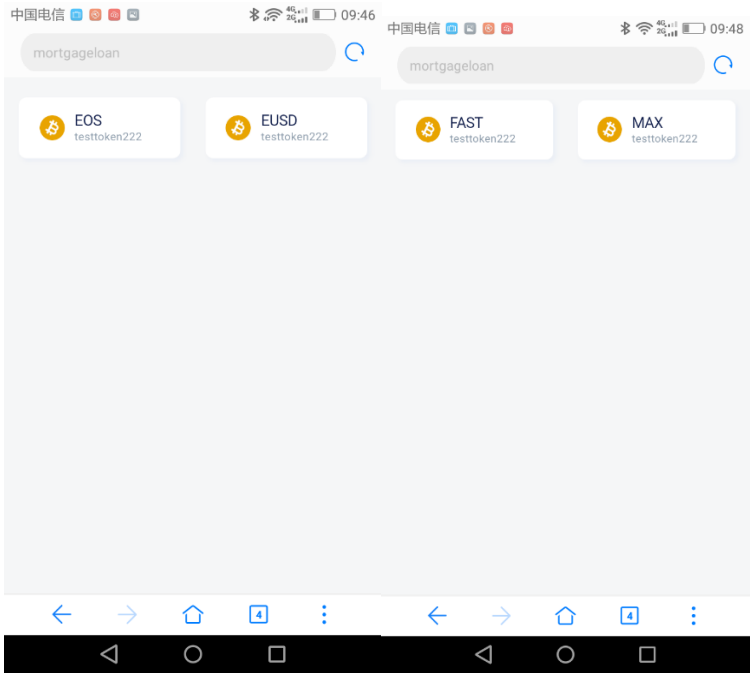
appear when “lend immediately” button was clicked by the lenders, which is the same with the situation in home page part.

7.2.4 Borrowing

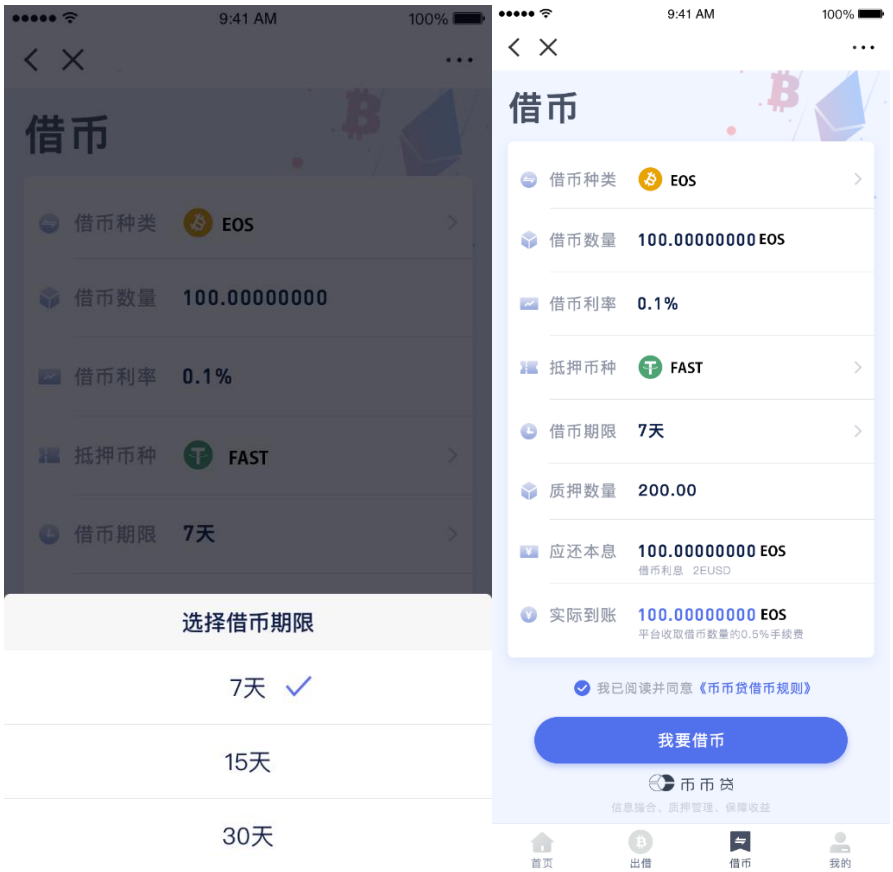
Click “borrowing” menu at the bottom, borrowing page will appear which is shown in the right part of the following figure.



Borrower can select token type for borrowing and mortgage, which is shown in the following figure.

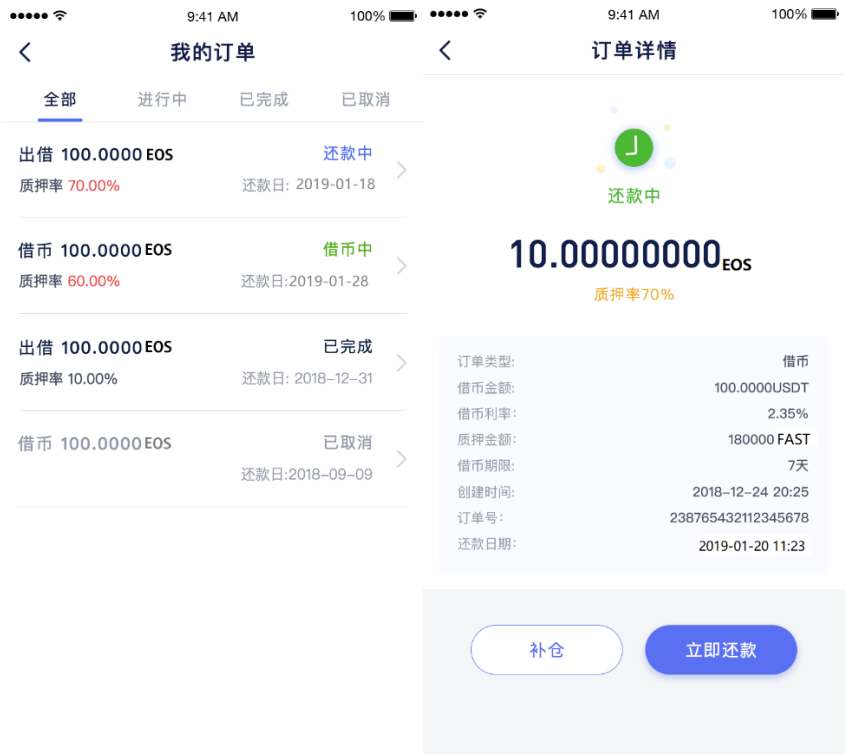


Select borrowing cycle, fill in token amount, interest rate, the platform will feed back information about mortgage amount, values to repay and asset arrival.



7.2.5 Repay and Call Margin

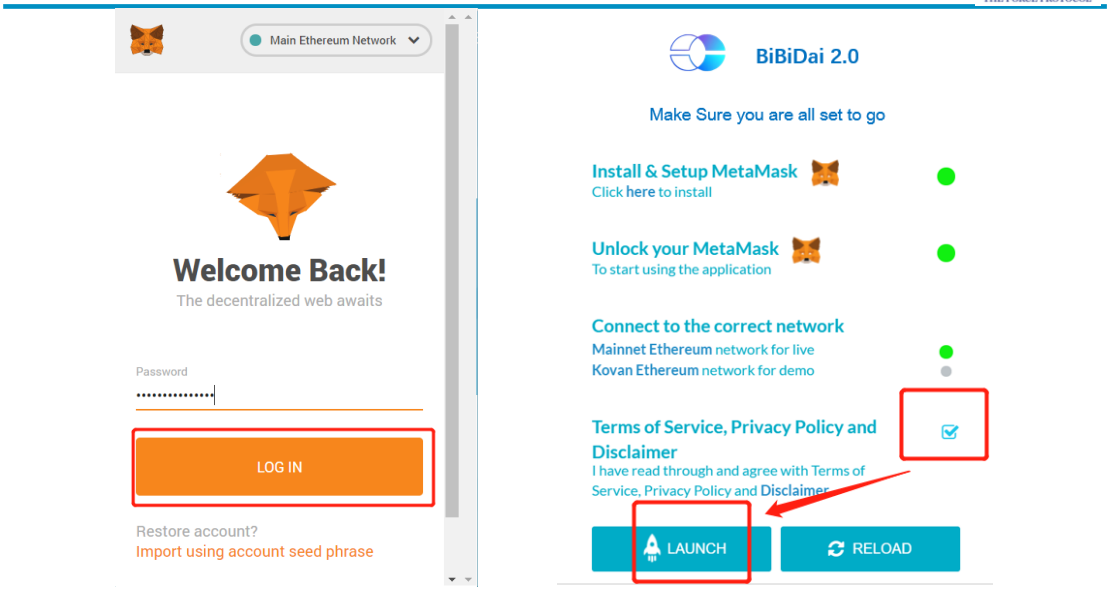
Users can query “my order”, where all the orders related to the users will be listed in the following figure(left). Users can repay or call margin any order with the “in borrowing” status, which is shown in the following figure.



7.2.6 Exit

When all the actions are completed, user can choose to exit.


9:41 AM
100%
18712345678
我的资产（USDT）
100.0000
您确定退出当前账号吗?
确定
再想一想
关于我们
退出登录
首页
出借
借币
我的



Then, another page will appear. Select the checkbox which is tagged by red rectangular in the above image, click "**LUNCH**" to login our application.

8 EFOR Token

We will issue token with name EFOR, which will play key role in The Force Protocol ecosystem.

8.1 Utility of EFOR Token

The token will not only promote operation of the ecosystem, but also serve as carrier for autonomy of decentralized organizations. The EFOR will play following roles in The Force Protocol ecosystem:

8.1.1 Transaction Fee Deduction

In The Force Protocol system, when the loan orders match, the smart contract will deduct a small amount of tokens from both parties, and send them to the supernodes submitting the orders of both parties as the service fee income. Under normal circumstances, the handling fee is 0.5%, which is charged in both directions. When user holds the EFOR token, the smart contract will calculate the commission amount base on user's EFOR holdings, and then deduct the calculated payment fee. In order to

prevent the supernode from selling to the market immediately after the collection of EFOR fee and leading to deduction of total EFOR value, The Force Protocol system will set a freeze period for each EFOR obtained in the form of service fee. After freeze period is over, the supernode will get released EFOR fee. This mechanism is in order to maintain the price of EFOR token from being centralized sell-off by supernodes, and stabilize the ecology of The Force Protocol.

8.1.2 Supernode Pledge Lock

Within The Force Protocol system, each supernode needs to pledge a certain amount of EFOR tokens when it joins in the system. This part of the token will be hosted by a special smart contract. The main function is shown in the following section. The smart contract will also periodically scan the pledge level of supernode EFOR token. If the pledge is lower than minimum requirement of the system, supernode will receive a pledge notification. If supernode does not replenish the pledge EFOR within specified time, the system will submit information to the arbitrator according to the preset conditions to determine whether supernode can perform the function normally. If judgment result is negative, arbitrator will submit the deletion proposal for this supernode to community governance system.

8.1.3 Mining by Borrowing

In order to promote user's action on mortgage loan, we reserve strategy of mining by borrowing of EFOR token. The Force Protocol development team will submit a detailed plan for Mining by borrowing to community governance mechanism at appropriate time after the platform is online and running. Content of the plan will include a series of key factors such as the total amount of trading, mining rules, mining time and so on. After community discussion, decision-making and voting, the plan will be implemented as planned.

8.1.4 Collateral

We will promote EFOR to be listed on top global exchanges as soon as we finish our token sale. It is expected that EFOR will become the main collateral in The Force Protocol's ecology and will be favored by borrowers and investors. EFOR will enjoy the preferential loan rate and exchange fee deduction when it is used as mortgage collateral. This part is shown in "**Transaction Fee Deduction**" and "**Increased LTV**" sections described above respectively.

8.1.5 Increased LTV

If borrower uses EFOR as the collateral for loan, it can enjoy a certain uprising ratio above the basic LTV (loan-to-value), borrower could obtain more loans. This setting can promote the use of EFOR and increase the user's stickiness to The Force Protocol. The specific LTV will have a preset ratio at the beginning of the system's launch. As the loan orders continue to accumulate, we may modify the EFOR LTV to a more reasonable point and submit it to the community for voting.

8.1.6 Community Governance

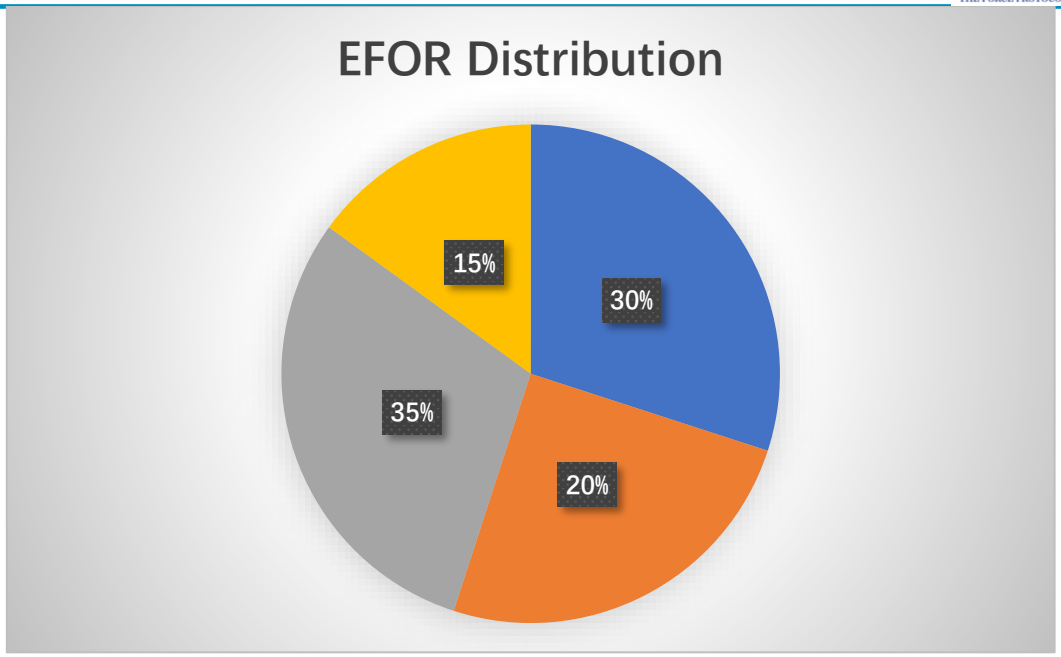
EFOR is the only tool for members of The Force Protocol community to participate in community voting. First, when there are any important issues that needs to be submitted to the Community Governance Committee for discussion, the proponent must hold EFOR and submit proposal to a dedicated smart contract, mortgage a certain number of EFOR before submitting the proposal to the community discussion board. Community holders can submit suggestions for changes to the content of proposal within a certain period of time. All changes will form an iterative version and be recorded by the blockchain. After the specified time limit is over, EFOR's holder will vote on the content of the proposal. All locking EFOR token will not be counted in the ticket, different proposals need to meet specific number of votes to be approved. All EFORs for voting will be locked by a smart contract address for a certain period, during which these EFORs are out of circulation system.

8.1.7 Cryptoassets Offering and Pledge

As the decentralized mortgage loan platform, The Force Protocol welcomes all tokens that meet the interests of community users for conducting loan transactions. However, in order to prevent some of the “air tokens” and “scam tokens” listed on our platform and occupy the public computing resources of our community, we will also set up a mechanism for all the mortgage collaterals on The Force Protocol, except BTC, ETH, XRP, BCH, EOS, XLM, LTC, ADA, XMR, TRX, DASH, BNB, NEO, ONT, ETC, XEM, ZEC, USDT, USDC, TUSD, GUSD, PAX and other top cryptocurrencies. The owner of specific token should hold and mortgage certain amount of EFOR token, which can be used as a collateral after being voted by the community. We must ensure that all tokens on The Force Protocol platform follow the interests of our community. All tokens that may harm the interests of our community will be banned from going online as collateral. Of course, we also welcome all excellent cryptocurrency projects to be listed on our platform.

8.2 EFOR Distribution Plan

The total amount of FOR tokens is 1 billion with no additional issuance forever. Directed by The Force Protocol initiative team, 85% of Tokens will be used for community building and community donation programs, of which community ecological construction accounts for 30%, The Force Protocol Foundation accounts for 20%, and strategic investors and community donations account for 35%. The remaining 15% will be reserved by the founding team, which can be used as reward for their early contribution to the project, and for new team members. The tokens allocated to founding team will be locked up for 3 years from first exchange listing, of which 30% 12 months, 30% 24 months, and last 40% 36 months. The pie chart of allocation for EFOR token is shown in the following figure.



8.2.1 Community Ecological Construction

Construction of community ecological includes but not limited to: ecological incubation and incentives of the Force Protocol community decentralized application (DAPP), developer community construction, commercial and industrial cooperation, marketing promotion, academic research, education investment, laws and regulations, etc.

8.2.2 The Force Protocol Foundation

We have registered a non-profit Foundation in Singapore. The main tasks of the Foundation are: construction and operation of The Force Protocol Ecology, development of business direction, issuance and management of EFOR tokens, and management of funds tokens obtained by donation.

8.2.3 Strategic Investors and Community Donations

Based on initiation and operational requirement of the project, 35% of the tokens will be reversed to reward strategic investors and community members.

9 The Force Protocol Project

9.1 The Team

Allen—Co-founder & CEO

Former district manager of Enactus China covering Tianjin, Shandong and Hebei Province. Managing Director of Qianshenghui Capital. He is focusing on research and investment of Internet finance industry and has invested in many high-quality projects in the sub-segment of cash loan and car loan et al. He is early member of Bitcoin Talk community, early member of the XDAG-Chinese community, and partner of AlphaCoin Fund. He has invested in tens of projects like Rsk, Celer, Box, Dcc in blockchain field, with the idea of code being law, privacy being freedom, computing being right. He laid out a lot of ecological projects related to blockchain technology in advance, such as the mines, mine pool, wallets, exchanges, and media. He is responsible for financing and external cooperation in The Force Protocol Team.

Yu Hongxue—Co-founder & CTO

Master of Beihang University in computer science. Former core development member of Big Data Platform in Sogou Ltd. Who was responsible for ETL, core indicator calculation, task monitoring, task scheduling, task optimization and so on. He is good at development of DAPP and security technology. Recently, he is focus on Anti-cheating and recommendation algorithm research. As early follower of blockchain technology, he is familiar to source code of Bitcoin, Ethereum, EOS, etc. He is responsible for development of supernode and blockchain in The Force Protocol Team.

David Lei—Co-founder & COO

Master of Tsinghua University. He participated in bitcoin mining from early 2012, in early 2017, his systematically study about blockchain technology and cryptocurrency lead to a deep understanding of blockchain industry’s potential development orientation.

He then invested in several high-quality projects such as EOS, Filecoin and Cybermiles, participated in community construction of multiple cryptocurrency projects. With his extensive experience in community evolving, he participated in the development and operation of the XDAG Chinese community, founded self-media “BW investment” in cryptocurrency research field. He is responsible for operations, media campaigns and strategic research in The Force Protocol Team.

Xu Chao—Co-founder & Product Director

Master of Harbin Engineering University, former Cloud Computing Architect of ZTE, head of domestic SaaS cloud service innovation project. He participated in creating SaaS product deployment architecture and business operation system, which achieved 100 million revenue scale from 0. He is also good at network structure and protocol development. He participated in the planning and design of ZTE's electronic license scheme based on blockchain technology, planning and design of the blockchain technology route of China Unicom Research Institute. He is a Blockchain technology enthusiast and early community project participant, who has participated in crowdfunding projects such as NEO and EOS. He is responsible for products, business model design and external cooperation in The Force Protocol Team.

Zhang Linbo—Co-founder, Chief scientist & Blockchain Development Engineer

Ph.D. of Artificial Intelligence, Chinese Academy of Sciences, Senior Engineer, Senior Network Planning Designer granted by Ministry of Industry and Information Technology. His research interests include artificial intelligence and data mining. He once worked in Chinese Academy of Transportation, during which he independently completed one national-level project, in charge of over more than 10 provincial-level information projects and participated in more than 30 provincial-level information

projects. He awarded title of senior engineer in 2013, which make him the youngest member at the time. He ever took charge of policy research on domestic application of calling taxi online and autonomous driving technology, as former member of national think tank in the field of Intelligent Transportation. He is good at C, C++, C#, JAVA, Python programming languages, and has researched distributed accounting, cryptocurrency, cross-chain information interaction for several years. He is responsible for cryptography, artificial intelligence and blockchain underlying technology in The Force Protocol Team.

Liu Gang

Bachelor of Wuhan University with major of software engineering. He used to be the product director of several P2P and consumer finance companies. He has participated in the establishment and operation of the companies from 0 to 100 and has extensive experience in online lending. His products have a transaction volume of more than 100 billion RMB and have accumulated tens of millions of users. He is good at product design, user growth and financial risk management.

Zheng Yajun—Blockchain Development Engineer

Bachelor of Harbin Institute of Technology with major of computer science, former senior research and development engineer at Baidu. At present, he is director of risk control and data center of Shenzhen Frontline Financial Services Ltd., who is in charge of designing company's overall business risk control system and structure, making big data analysis, credit rating, loan Management and development of wind control model. He has extensive work experience in the Internet and finance fields. He is a technology geek and has participated in multiple GitHub open source project code submissions.

Ivan Wang—Blockchain Development Engineer

Master of Peking University in software information. He engaged in data analysis and mining research after graduation, and has entrepreneurial experience in the field of artificial intelligence. He has conducted in-depth research on the underlying source code of blockchain projects such as distributed Ledger technology, Bitcoin and Ethereum.

Dai Shaopeng—Blockchain Development Engineer

Ph.D. student in Chinese Academy of Sciences, his research covers data mining, Machine learning, Benchmark, and so on. Several of his paper has been indexed by SCI.

Wang Jie - Project Operations and Finance

Ph.D. of Department of Mathematics, Zhejiang University, former Sponsor of China Galaxy Securities co., LTD. He has participated in IPO, mergers and acquisitions of several companies, and has extensive experience in the capital market.

9.2 The Advisory Board

Frozen Xie

Blockchain and DApp developer, core developer & maintainer of XDAG project, contributor of several blockchain and opensource projects. Co-founder of Consensus Approach Ltd., founder of TeamTaoist Studio. Producer of mobile game and crypto game. Former technical staff and project manager of Lucent Technologies Bell Labs. Expert in IBM DB2, Network Communication and Blockchain Technologies. Senior software engineer in iOS/Android/Html5.

Tian Hongfei

Master's degree of Massachusetts Institute of Technology. He is partner of Green Pine

Capital Partners Co. LTD, and former partner of SIG Asia Venture Capital Fund. He has more than 15 years of experience in e-commerce and network security, covers high-tech industries and investment banks in Silicon Valley, Germany and China.

Andrew Yi

He has worked in Oracle and Google, both of which are Fortune Global 500 Companies. He ever served as Executive Dean of Shenwan Hongyuan Research Institute, which one of the largest securities companies in China. He is the first promoter and advocate of Internet finance, big data and cloud computing industries in China. He is one of the best analysts who ranked first, and one of Persons of the Year voted by Tencent, Sina in Chinese Financial Technology, who has invested nearly 100 related companies.

Li Zhongnan

Product Partner of Hetrone Financial, who was in charge of products like ufenqi and JiandanDai. As the first batch of phased shopping malls for college students, ufenqi is ranked third in the comprehensive industry in 2016 years, with a maximum daily transaction volume over 100 million. He was the former product design experts of Meituan Ltd., who was responsible for design work of all mobile platform products and developed mobile product design systems and standards. The work experience in QQ products of Tencent and Tmall mobile products of Ali, provided him rich experience in product design and R&D management.

Simon Liu

Master of Computer Science and Technology, Tsinghua University. He has worked as search technology development engineer in Baidu as an early staff, core engineer of Chinese search in Google, and chief engineer of Bing search in Microsoft.

9.3 Strategic Partners

TokenInsight

TokenInsight is a global token data rating agency, which is focus on certification risk ratings to help investors avoid risks and increase revenue. The Force Protocol team works closely with Tokeninsight on selection of encrypted digital asset collateral, mortgage rate setting and risk control level, which can improve mortgage lending risk control level.

BabelBank

BabelBank is a trustworthy blockchain bank, who dedicated to build an open and shared blockchain asset financial services system. Its business covers digital currency deposits and loans, blockchain asset financing, and digital currency financing. In the future, it will launch more business and related financial derivatives. As a pioneer in the field of digital currency mortgage lending, BabelBank will jointly maintain the global loan order pool with role of super node of the force protocol, which can help optimize resource allocation, jointly develop decentralized digital currency lending industry specifications for users, provide fast and convenient lending services.

InVault

As the first licensed virtual asset custody platform in the Asia-Pacific region, InVault holds a Trust and Corporate Services license in Hong Kong and has launched virtual asset custody solutions for digital lending platforms and quantitative trading funds, both of which are vertical businesses. Apart from centralized enterprise wallet, InVault also provides collaborative custody and special account custody and other virtual asset

custody solutions applicable to various business scenarios, including primary market fund custody, secondary market fund custody, exchange wallet custody, FOF fund custody, collateral custody, OTC fund supervision and so on. The force Protocol super node can use InVault's enterprise-level virtual asset hosting technology and scheme to launch services such as mortgage loan, multi-currency portfolio mortgage loan and large-amount multi-account mortgage loan for users. Its multi-tiered risk control system ensures that users' mortgage assets can minimize losses in case of large price fluctuations.

He-Legal Chain Joint Library

He-Legal Chain Joint Library is founded by The Force Protocol Team and Renmin University's legal chain laboratory, which is focus on blockchain legal compliance, regulatory policy research, and achieve supervision and governance of blockchain industry. As academic guidance organization of Chinese Internet Finance Supervision Center, the Joint library will actively participate in research of legal supervision in the fields of financial technology and blockchain finance.

BIRISE

Birise is a digital asset wallet with function of market quantitative analysis. It provides comprehensive services, quantitative analysis, news announcements and other derivative services, which can help overcome shortcomings of digital asset investment and realize value-adding of digital assets. The Force Protocol will cooperate with Birise in the field of digital asset lending.

ThinkBit

ThinkBit is the next generation digital asset trading platform, which serves to trading

and depositing of digital assets conveniently. It is the first exchange with 100% cold wallet technology in the world, which is extremely safe. The Force Protocol and ThinkBit will collaborating on blockchain technology, digital asset management and currency stabilization.

eArk-Green Blockchain

The green chain provides ultimate solution for human environmental protection by efficient consensus, credible circulation, and global transactions of environmental data assets. All of these are realized by a global green value network, which integrates satellite communication network, intelligent Internet of Things, and information internet technologies.

Hetrone Financial Company

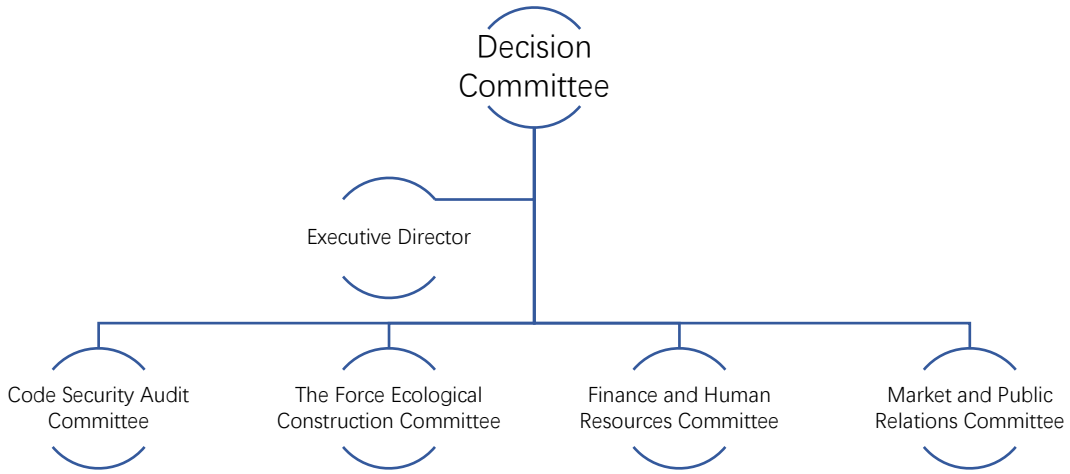
Hetrone Financial Company is a technology company, which is focus on personal and microfinance innovation. It is devoted to forward-looking research and in-depth application of data engineering, artificial intelligence, cloud computing and other technologies in the field of inclusive finance. It adheres to the vision of “Making Finances Free of Borders” and continues striving to achieve “Providing equal and effective financial services to everyone”. It provides equal and convenient Internet financial information services for individual users and small enterprises, and makes technological innovation play leading role in the economic and social development.

9.4 Governance

The Force Protocol team firmly accepts the concept of decentralized community autonomous management model, and each participant in the community has the right to participate in project management and rulemaking. The Force Protocol Foundation (hereinafter referred to as The Force Foundation) has been established in Singapore as

the main body of project management, which is responsible for fair, open, transparent, and nonprofit operation of the Force Protocol project. In addition, The Force Foundation will be in charge of the security of all cryptoassets raised, and support The Force Protocol development and operations teams building. If profit was generated, it will be retained as fund for community activities, rather than being distributed among members. The Force Foundation is approved by the Accounting and Corporate Regulatory Authority (ACRA) and is regulated by the Singapore Companies Act, which operates independently from the government.

In order to help The Force Foundation making rational use of the funds and resources under premise of fairness, openness and transparency, Decision committee is set up, with subordinate committees of Code Security Audit Committee, The Force Ecological Construction Committee, Finance and Human Resources Committee, and Market and Public Relations Committee. In this way, development of The Force Ecology can be promoted, application scenarios of The Force Protocol will be expanded, and more institutions, companies, projects and organizations will be absorbed into the Ecology.



Decision Committee:

The Decision Committee is the highest decision-making body of The Force Foundation. It undertakes the final decision-making function and is responsible for reviewing major issues, such as the strategic planning, annual plan, budget, etc., and voting on major issues of The Force Ecology on behalf of the Foundation.

Executive Director:

The executive Director is elected by The Force Decision Committee, who is responsible for the day-to-day operation of the foundation, coordination of the subordinate committees, presiding decision-making committee meetings, and reporting to the decision-making committee in regular.

Code Security Audit Committee:

The Code Security Audit Committee is responsible for code security audits of the Force Protocol, decision-making of technology research and development, open data interfaces, and development of technology patents. In addition, the committee will also maintain communication with community members and ecological participants in the community, via hold technical exchanges irregularly.

The Force Ecological Construction Committee:

The Force Ecological Construction Committee is responsible for the development of The Force Ecology and partners. The committee will use the funds raised to carry out activities for ecological construction and business cooperation, encouraging more developers to build applications based on the Force Protocol, and abstracting more potential partners into the original ecosystem.

Finance and Human Resources Committee:

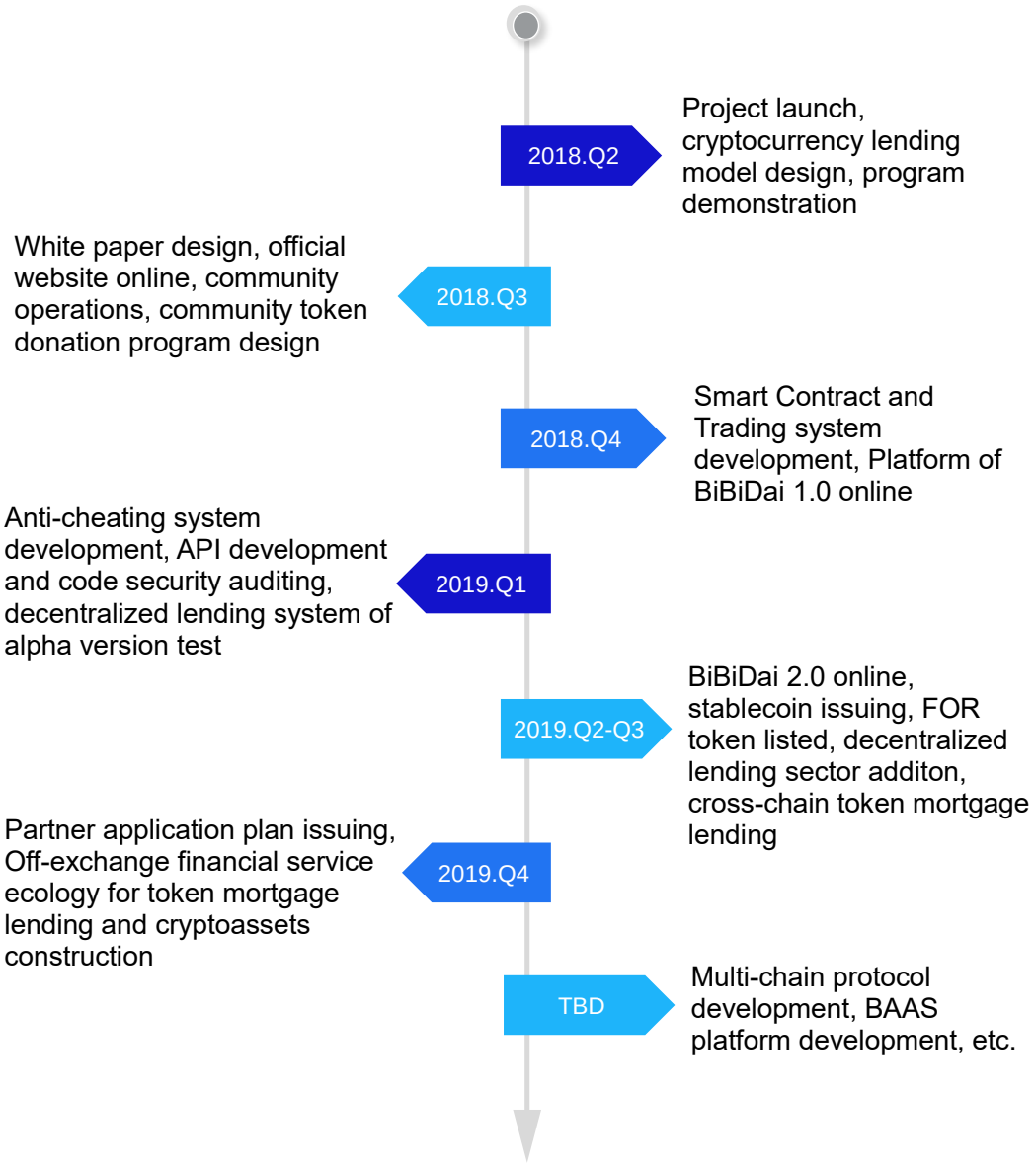
The Finance and Human Resources Committee is responsible for the use and review of foundation funds, personnel recruitment and compensation management, together with daily operating expenses management.

Market and Public Relations Committee:

Market and Public Relations Committee is responsible for marketing and promotion of the Force Protocol and ecological projects, holding community communication

meetings, participating in blockchain field exhibitions and academic seminars. Meanwhile, maintaining public relations and communication with industry associations and government regulatory organizations is as important as the function stated above for the Committee.

9.5 Roadmap


2018.Q2
Project launch, cryptocurrency lending model design, program demonstration
White paper design, official website online, community operations, community token donation program design
2018.Q3
2018.Q4
Smart Contract and Trading system development, Platform of BiBiDai 1.0 online
Anti-cheating system development, API development and code security auditing, decentralized lending system of alpha version test
2019.Q1
2019.Q2-Q3
BiBiDai 2.0 online, stablecoin issuing, FOR token listed, decentralized lending sector additon, cross-chain token mortgage lending
Partner application plan issuing, Off-exchange financial service ecology for token mortgage lending and cryptoassets construction
2019.Q4
TBD
Multi-chain protocol development, BAAS platform development, etc.

10 Legal Evaluation

The Force Protocol serves loan in period of Blockchain in the form of “loan without intermediary”. Traditionally loan involves due diligence by financial institutions before loan is issued, which requires access to local credit score or rating system maintained by third parties. Moreover, as the legislation for loan varies from country to country, the main question is to comply with legislation while ensure that lenders are protected.

Cryptocurrency is relatively new to governments, and governments have not been good at regulating or defining the legal status on cryptocurrencies. Moreover, it is acknowledged that the definition given in other fields of law, such as tax law does not finally settle or give analogy to use the definition in other fields of law. Therefore, in the most sensible definition, cryptocurrencies may be managed through contractual agreements.

10.1 Contract Relationship

As cryptocurrencies are not regulated, they are contractual agreements between different parties. This would mean that the agreement between the borrower and the lender would be conducted on the basis on contract law. The question is to decide which contract law would apply since there are jurisdictional differences. The basic principle in any contract law despite the jurisdictional differences is the freedom to contract, which means anyone can agree to lend or borrow.

Even thought, the loan is on Smart Contract and on a trustless environment, both the lenders and borrowers will be living under governed jurisdictions. This means that the loan agreement should be valid, even though one might not have the ability to enforce it. To provide a binding loan agreement, smart contracts based on our protocol should implement all legal functions of the loan in simple terms and conditions. These terms include the loan period, premium, the collateral, the lender, the borrower, governing law and dispute resolution. Therefore, the parties do not have to deal with the juridical questions to participate in the loan market.

10.2 Collateral

New form of collaterals. In most jurisdiction collaterals, also known as pledges are governed to establish the legal standing of a pledge against third parties. This has been important through time because enforcing collateral is seen as an exception on the right to ownership. Moreover, most of the rules on collateral from jurisdiction to jurisdiction apply on real property, and not much jurisprudential literacy on pledging tokens.

Traditionally, collateral is required to comply with certain actions to establish valid collateral and thus in the case of default enable the transfer of ownership. These rules do apply in most jurisdictions where would be governed. In that case, the collateral must be transferred to the lender. However, the situation is different, where the property could be held by a third party as well. By using Smart Contract, lender does not hold the collateral. Instead, the collateral is held on the blockchain which is controlled by the Smart Contract. The purpose behind possession is to inform third parties that the item is pledged. This aim is achieved by design of Ethereum blockchain where all transactions can be inspected. Therefore, when ERC20 token is pledged on Supernode based on our protocol, the collateral is moved to the Smart Contract and locked until the loan is repaid. This information is accessible to anyone, anywhere with an internet connection.

10.3 Know Your Customer (KYC)

Supernodes (supernodes served as financial product manager excluded) based on our protocol do not lend or hold assets, which is a decentralized application running on Ethereum blockchain network. Even the supernode issuing stablecoin will have no privilege other than a common lender who holds plenty number of stablecoins. Supernodes does not control any assets between the lenders and the borrowers. This means that when one party places a loan request, he creates the Smart Contract. When the Smart Contract is created, the data is broadcasted to Ethereum network and is not stored on any servers locally. Supernode could be seen as a tool set loans and broadcast

loans on Ethereum network.

KYC regulation applies on money loan between borrowers and lenders. The thresholds do vary from jurisdiction to jurisdiction but remain similar in principle. However, uncertainty lies within loan ETH since ETH is not a currency by definition of governments. Therefore, whether loan ETH is subject to KYC is an unsettled question.

In the view point of our protocol, it endorses the KYC policies due to the fact that regulation might follow sooner or later because decentralized environments do not occur in a vacuum. For this reason, The Force protocol will implement integrations for assisting the parties to comply with KYC regulations. The aim is to provide KYC when deemed to be needed and enable it in the best possible manner. Currently, KYC only can be solved through messaging and interaction between the lender and borrower.

In the future, The Force Protocol team will work to provide a KYC solution that would not spill over to off-chain. The easiest way to provide KYC without the need to spill the application to off-chain would require the borrower to insert a link to the material that would comply with KYC. This material would include identification, proof of address and origin of funds. Decentralized storage could also be used to achieve the storage of KYC data. The Force Protocol team will focus on the progress of the KYC solution, and introduce appropriate on-chain KYC mechanism at appropriate time.

11 Important Reminder and Risk Management

11.1 Important Reminder

This white paper is intended as a conceptual document describing the technical direction development plans of The Force Protocol and EFOR tokens. It does not constitute a prospectus, offer document, securities offer, investment tender, or sale offer of any product or asset. We cannot guarantee the accuracy and completeness of the Whitepaper information, and you should consult your legal, financial, tax or other professional advisors before participating in any of the activities described in this whitepaper.

All supporters should read the whitepaper and the relevant instructions on the official website carefully, which is important to fully understand blockchain technology and risks of our project. Once the investor participates in the project, he is considered to understand and accept risks of the project. It should also be pointed out that purchase EFOR token is essentially a donation, it stands for the willing to take risks and grow together with The Force Protocol community. They will not receive any direct or indirect benefits or dividends by holding EFOR tokens.

As a license for the ecology of The Force Protocol, EFOR does not represent the benefits promise of dividends, value-added, equity, securities and their derivatives. No channel will be provided for resale, and the holders have the right to decide usage of it. This whitepaper is available in multiple languages. If there is any disagreement, the English version prevails. You will be acknowledged that you have read and understood the English version of this whitepaper yourself.

11.2 Risk Tips

1. At present, the attitudes and policies to cryptocurrency financing based on blockchain projects are not clear for the major countries in the world. There is possibility of losses for investors due to policy factors;

2. For the EFOR tokens, the risk of dramatic fluctuation in price and being manipulated by dealers exists. This is due to the fact that high uncertainty exists in cryptoasset market, while mature supervision is lacked.

3. There will be fierce competitions, as many projects exist in current blockchain field. Based on the rich experience and industry resources, our team will try our best to ensure the continuous development of the project. However, even we fully have confidence to ensure success of our project, uncontrollable events may lead to a different result, and no promise can be made.

4. The Force Protocol team will spare no effort to achieve the goals set forth in the white paper, and explore the longer-term development space of the project actively. However, due to uncertainty of the external environment and internal resources, we will retain the right to adjust content of the whitepaper. There was no voluntary obligation for us to inform all changes to the whitepaper, and participants are requested to keep up with changes through relevant channels.

5. The Force Protocol is based on blockchain technology and cryptography algorithms. As blockchain technology and cryptography are still in the process of rapid development, The Force Protocol team cannot completely ensure the realization of all technologies. Moreover, all technical projects have the potential risk of user losses, which are caused by hacker attack or code vulnerability.

6. As cryptocurrency investment is still a new field, there may be various risks which we have not mentioned or expected.

12 Updates and Opensource

12.1 Update

Once our smart contract is deployed to the blockchain, its internal logic will not be changed. Therefore, to update a protocol on the blockchain, one must deploy a completely new smart contract that either forks the network or disrupts users and

processes that depend on the protocol until they suit to the latest version. a disruptive protocol update could invalidate all open orders and require each market participant to approve a new smart contract to access their trading balances. Alternatively, the protocol could fork into two versions that operate in parallel, neutralizing network effects created by DAPP interoperability. At the time, smart contract abstraction will be used to continuously integrate updates into a protocol without disrupting higher-level processes.

In the context of loan, one contract instance is constructed each trading, which is specified in the order. As the orders made before the update point to the older version, the action followed will proceed according to the old version. However, such an update mechanism may also create security risks for end users, e.g. collateral locked by the decentralized smart contracts may be gained by attackers in the worst case. The EFOR tokens may be used to drive a decentralized update mechanism that allows for continuous integration of updates into the protocol while also protecting the protocol's users and stakeholders.

12.2 Open Sourced Code Community

When development of the whole project is completed, tens rounds of verification will executed for security purpose. After that, all essential source codes will be upload to open sourced code community such as Github for sharing, which can be found at:

https://github.com/TheForceProtocol/TheForceProtocol.git

Here, all smart contracts in our project will shared in “\contracts” directory, and libraries used will be stored in “\lib” sub-directory, which will contain solidity files like *MathLib*, *StringLib* and so on. APIs will be provided in “\APIs” directory, where the source code may be written in JavaScript Language. In addition, source codes for migration are also updated.

At present, the source code is underdevelopment, and no source code is updated. Subsequently, all code for our smart contracts will be published here. If you have any

issues to report please open an issue on GitHub to discuss with the team.

Fianally, if there is any incident on the Github website that may affect normal open source activity or communication in the community, The Force Protocol team reserves the right to change code publishing ways and inform community members through relevant channels.