CyberMiles: A Next Generation Blockchain Protocol for Business Transactions

By 5xlab

A technical white paper

v1.5

[Commercial scenarios and token sale terms are discussed in a separate document]

Disclaimer

This is a conceptual document ("**Technical White Paper**") describing our proposed CyberMiles blockchain protocol and direction for its network development. It may be amended or replaced at any time. However, there is no obligation to update the Technical White Paper or to provide the recipient with access to any additional information.

Readers are notified as follows:

Not available to all persons: the CyberMiles platform and CyberMiles tokens are not available to all persons. Participation may be subject to a range of steps, including the need to provide certain information and documents.

No offer of regulated products in any jurisdiction: CyberMiles tokens (as described in this Technical White Paper) are not intended to constitute securities or any other regulated product in any jurisdiction. This Technical White Paper does not constitute a prospectus nor offer document of any sort and is not intended to constitute an offer or solicitation of securities or any regulated product in any jurisdiction. This Technical White Paper has not been reviewed by any regulatory authority in any jurisdiction.

No advice: this Technical White Paper does not constitute advice in relation to whether you should participate in the CyberMiles platform or buy any CyberMiles tokens, nor should it be relied upon in connection with, any contract or purchasing decision.

No representations or warranties: No representations or warranties are made as to the accuracy or completeness of the information, statements, opinions or other matters described in this document or otherwise communicated in connection with the project. Without limitation, no representation or warranty is given as to the achievement or reasonableness of any forward-looking or conceptual statements. Nothing in this document is or should be relied upon as a promise or representation as to the future. To the fullest extent permitted under applicable law, all liability for any loss or damage whatsoever (whether foreseeable or not) arising from or in connection with any person acting on this Technical White Paper, or any aspect of it, notwithstanding any negligence, default or lack of care, is disclaimed. To the extent liability may be restricted but not fully disclaimed, it is restricted to the maximum extent permitted by applicable law.

Other companies: other than the CyberMiles Foundation Limited ("**Foundation**") and 5miles LLC ("**5miles**"), the use of any company and/or platform names and trademarks does not imply any affiliation with, or endorsement by, any of those parties. References in this Technical White Paper to specific companies and platforms are for illustrative purposes only.

You must take all necessary professional advice, including in relation to tax and accounting treatment. We hope the CyberMiles project will be highly successful. However, success is not guaranteed and digital assets and platforms involve risk. You must assess the risks and your ability to bear them.

Executive Summary

The blockchain technology holds great promise in business applications. However, current generation blockchains suffer from low execution efficiency and low developer productivity. As a result, they are not widely adopted for common business transactions. In this paper, we present a new blockchain network protocol, called CyberMiles blockchain, that is specifically optimized for business contract transactions.

Our proposed solution is a protocol innovation that makes a stack of proven business middleware technologies accessible to a distributed virtual machine on the blockchain. The new blockchain is highly performant and scalable supporting over 10,000 transactions per second. It enables businesses to write Smart Business Contracts, which are distributed middleware applications that codify business rules and processes. The network's native crypto currency, the CyberMiles Token (CMT), may be used to settle transactions, reward network validators (who execute Smart Business Contracts), and incentivize community members to provide services to each other.

A unique advantage of the CyberMiles blockchain is that it will be deployed to support 5miles' existing e-commerce network of over 10 million US-based registered users, and over $3 billion USD in estimated annual transactions. This would immediately create the largest blockchain-based commerce network in the world. The network could provide services such as decentralized user identity and credit management, decentralized settlement clearing house, peer-based voting and conflicting resolution. Example applications on the network platform include decentralized personal information "wallets", peer-to-peer small business loans, and peer dispute arbitration.

Table of Contents

1. INTRODUCTION

1.1 Bitcoin and Ethereum

The Bitcoin is the first killer application of blockchain technology. The Bitcoin network, known as blockchain 1.0, is primarily a distributed ledger system with a built in decentralized consensus mechanism. Through the UTXO technology, although it is possible to write programs to run on the Bitcoin network, the low-level UTXO programs are very limited in capacity. It is a Tuning incomplete programming environment, and is very hard to use. As a result, the Bitcoin network is mostly a distributed ledger system to record bitcoin transactions with very little community-developed applications.

The Ethereum project aimed to build Blockchain 2.0. By adding a Turing complete virtual machine (called the Ethereum Virtual Machine, or EVM), the Ethereum blockchain aspires to be the "world's computer" by supporting 3rd party scripts known as Smart Contracts to move tokens / cryptocurrencies between accounts when certain conditions are met (e.g. one of the use cases is for the Smart Contract to act as an escrow account). Those Smart Contracts are executed by Ethereum nodes at real time. Their results are validated and saved into the blockchain by miners (or validators).

Furthermore, Ethereum also supports the concept of Decentralized Apps (aka DApps), which run outside of the blockchain but can make calls to Smart Contract methods in the blockchain. In a typical setup, a DApp could be a web application that provides an UI for the corresponding Smart Contract.

1.2 Major Problems and Related Work

However, it is also widely accepted that blockchain technologies today suffer from the twin related problems of low efficiency and low developer productivity.

As a decentralized system, a blockchain network requires many independent and uncooperative nodes to perform the same computing tasks over and over, and then reach consensus on what is "true". This makes the system very inefficient and hard to scale, as the computing effort increases geometrically with the size of the network. Because of the scalability / performance problem, 3rd party computational tasks allowed on the blockchain network must also be very limited. That, in turn, causes very poor developer experience and low productivity. As a result, Ethereum Smart Contract DApps are not widely used today.

There are several proposed solutions on the horizon to address the performance and scalability issues of blockchain technology.

- New consensus mechanisms. Both the current Bitcoin and Ethereum blockchains use a highly inefficient consensus mechanism called Proof-of-Work (PoW) in order to secure the network from untrusted participants. A lot of work has been done to replace PoW with a much more efficient mechanism called Proof-of-Stake (PoS). Leading contenders in this space include the Tendermint's Byzantine fault tolerance (BFT) consensus engine, as well as Ethereum's own CASPER solution.

- Sharding of the network. A commonly used approach to scale a network is by sharding the network into several sub-networks. Then the entire network can scale horizontally by adding more sub-networks. In a decentralized blockchain network, however, the sub-

networks need to communicate with each other and reach consensus on their states. That is a much harder problem than regular database sharding. Leading solutions in this space include the Cosmos Internet of Blockchains and the Polkadot network.

- Off-chain computations. An even more direct solution to the performance problem is to move much of the heavyweight computational tasks off the blockchain itself, and use the blockchain consensus mechanism to record computational results only. There are also many experimentations in this space, ranging from Lightning Network's off-chain state channels, to Plasma's fraud proof side chains, to TrueBit's off-chain Ethereum Smart Contract transaction framework.

In this paper, we will not attempt to solve fundamental issues of blockchain scalability. We believe that, in time, a good solution will emerge by the community consensus. Future blockchain networks will incorporate all three approaches to become highly performant and scalable.

However, after these problems are resolved, the blockchain network still needs to attract and support enterprise application developers in order to be commercially useful. In this project, our goal is to propose an architectural solution to make 3rd party enterprise applications on blockchain networks (the Smart Contracts) much more powerful and much easier to develop at the same time.

1.3 A Better Smart Contract

As a first-generation technology, and for the scalability / performance reasons we discussed above, the Ethereum EVM and DApp are hard to use. We aim to drastically improve the EVM and its associated software stack to make it more developer-friendly and enterprise-ready.

- The Smart Contract often needs to be triggered by events external to the blockchain. In Ethereum, that requires an “oracle” to provide authoritative and deterministic state information of the external world. The oracle is a fragile solution, as it is not standardized and could change without the Smart Contract’s knowledge.

- The Smart Contract can only be loosely coupled with the DApp middleware. Without "knowing" what's available in the DApp, the Smart Contract cannot make calls to any software component in the DApp. Due to the difficulty of programming complex rules using Turing complete procedural programming languages, most Smart Contracts only implement simple business transaction rules.

- The DApp middleware cannot be encapsulated and reused. DApp developers must make architectural decisions and write one-off applications.

- The DApp middleware is not integrated with the blockchain cryptocurrency incentive system. DApp nodes need to contribute significant amounts of computing resources, but cannot receive cryptocurrency as reward. That has resulted in DApps being run in a centralized manner by corporations.

2. PROPOSED SOLUTION

To address the shortcomings of Ethereum and create a blockchain based "virtual machine" that is suitable for business developers to create distributed applications, we propose a new blockchain protocol to support what we term as "Smart Business Contract". The protocol not only would include a virtual machine, but would also define the middleware software stack outside of the blockchain (which is today handled by DApps in a non-standard way). Every node in the blockchain would not only run the blockchain ledger but also supports standardized middleware.

Borrowing a page from past successful enterprise software's playbook, the key is not to create an all-powerful virtual machine or programming language, but to build an extensive library of reusable software components, and then standardize the whole stack of software. A good example is the Linux operating system. It is widely adopted by enterprise users only after the community has extended the core OS with thousands of business-friendly software packages, and after Fedora / Red Hat came along to standardize the stack. Other past examples include the Java 2 Enterprise Edition platform, or the LAMP stack, or the Ruby on Rails platform. The strengths of those enterprise platforms lie in their standardized libraries and frameworks.

Software encapsulation and reuse are the most important best practices in enterprise software. It is time for us to apply this best practice to blockchain platform as well.

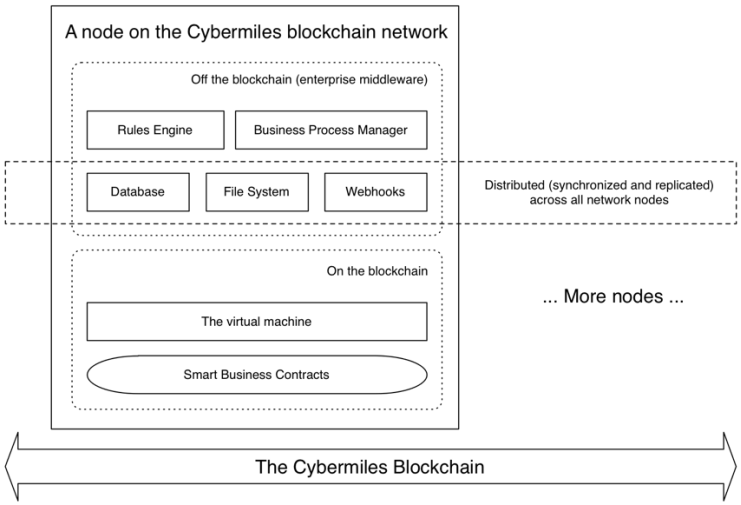

Figure 1. shows the overall architecture of the CyberMiles blockchain. As you can see, a significant amount of reusable software components resides outside of the blockchain.

2.1 Smart Business Contract

A Smart Business Contract on the CyberMiles blockchain is analogous to the Smart Contract on the Ethereum blockchain. It is executed by the blockchain node, and validated by the miner when a new block is created. The results from the Smart Business Contract is saved in the new block.

However, the main difference between CyberMiles Smart Business Contract and Ethereum Smart Contract is that, instead of writing every application from scratch, a Smart Business Contract would have access to an integrated stack of powerful business software middleware. Therefore, a Smart Business Contract can be easy to develop and highly reusable in itself.

Since the Smart Business Contract is part of the blockchain, the computing power required to execute it, including the efforts to run the whole stack of external enterprise middleware, can be accounted for using the CyberMiles system's cryptocurrency, the CyberMiles Token (CMT).

Network users who conduct transactions pay network validators small transaction fees in CMT to compensate their work in ensuring the data integrity.

2.2 Middleware Stack

The Smart Business Contract can access business software frameworks outside of the blockchain itself. Those software frameworks are embedded in each of the nodes running the blockchain. Those frameworks will run every time a Smart Business Contract is executed, and when blockchain miners validate the results. The stack of enterprise middleware frameworks included in the CyberMiles system include the following.

- A rules engine. Most business contracts must follow certain rules. Compared with a general purpose procedural programming language, a dedicated rules engine is proven to be both easy to use and efficient. It is already used by many businesses.

- A business process manager (BPM). A BPM system is a state machine that mimics the execution state of a multi-step contract. It is driven by external actions by contracting parties, and the BMP typically would use the rules engine to determine next steps.

- A distributed database. A distributed database is needed to support complex application frameworks and store application data. This database is replicated and synchronized across nodes on the blockchain. It does not store transaction results, which would be stored in the blockchain itself.

- A distributed file and data storage service. The Smart Business Contract and its related middleware services will need access to file services to manage larger data files required for decision making.

- A distributed webhook service. As a business system needs to interact with external entities that completes contract obligations (e.g. FedEx delivery notification for e-commence applications), we will build in a distributed webhook system that can receive external events related to the Smart Business Contracts.

A Smart Business Contract would incorporate complex rules, processes, data, and webhooks. But there is still need for a program to glue all the components together and orchestrate their work. This requires a general and Turing complete programming language. We could support this with the CyberMiles virtual machine would be shipped with the blockchain software to each network node.

2.3 Business Ready Contract Templates

A key aspect of Smart Business Contract is that the contracts are not only built on top of reusable software components, but also reusable themselves. Since most business transaction scenarios are well-defined (both from legal and commercial points of view), it is possible to create Smart Business Contract templates that can be reused by only changing key terms as parameters (e.g., contract party names, dates, amounts etc.). This library of templates would reduce the cost of building and deploying business applications and increase the value of the network itself.

2.4 Decentralized Apps with Smart Business Contracts

In the CyberMiles blockchain system, there is still a concept of decentralized applications (DApps). A CyberMiles DApp would manage all data and logic that should not be stored on the blockchain for privacy or performance reasons. The business logic related to the business transaction could be completely offloaded to the Smart Business Contract.

3. TECHNOLOGY

The technology solutions for the CyberMiles blockchain's middleware stack is relatively straightforward, as they are all proven technologies already widely used in the world of business middleware. We are building an engineering solution to incorporate those technologies into the blockchain framework and design proper economic incentives for the system to function.

The specific technology frameworks discussed in this paper are for illustration purposes only. The community may hold a discussion and voting event on a later date to elect a technology governing committee, which would then collectively determine the exact technology choices for the CyberMiles middleware stack.

3.1 The Rules Engine

The CyberMiles system will incorporate a forward chaining inference rules engine in its blockchain validator software. The rules engine resides outside of the blockchain itself but is used by the validator to execute Smart Business Contracts.

The rules engine implements the Rete algorithm to match patterns (facts from business actions) to rules, and resolve potential conflicts. The Rete algorithm is complex and beyond the

scope of this paper. We want to emphasize that, however, there are mature and successful Rete-based forward chaining rules engines widely used in enterprises today. Examples of such rules engines include Drools and Jess.

At the conceptual level, the forward chaining rules are a complex set of IF and THEN statements. The rules engine provides a special "programming language" that allows business analysts, as opposed to software developers, to express the business rules. The example below shows a set of rules written in a pseudo rules language. It shows how to determine the pricing of a product based on the buyer's profile. In this case, if the buyer's FICO score is above 740, he will be offered an 80% discount on the price.

```
Rule "Pricing"
  dialect "mvel"
  when
    m : Message(status==Message.GET_PRICE)
  then
    when
      m.fico_score > 740
    then
      m.price = m.listed_price * 0.8
End
```

The Smart Business Contract can now execute the rules when it is invoked by DApps. Notice that the data required for the "buyer's profile" and product pricing are retrieved from a distributed database on the CyberMiles platform, which we will discuss further in Section 3.3.

```
engine = load_rules("pricing.rl");

m = new Message ();
m.status = Message.GET_PRICE;
m.fico_score = 741; // Get from profile DB
m.listed_price = 100; // Get from product DB
```

```
engine.send(m);
return m.price; // Returns 80 to Dapp caller
```

While this example is simple, it is illustrative. It is easy to see how the Smart Business Contract can handle complex rules and encapsulate much of the business logic in the system.

3.2 The Business Process Manager (BPM)

In most business systems, the rules are only reactively applied when certain conditions are met. For example, the product pricing rule is only applied when a potential buyer asks for the price (e.g., by loading the product details web page). In this sense, the rules engine is invoked "on demand", and the system spends much of its time in a "waiting" state. That behavior can be modeled by a finite state machine (FSM).

A widely-used class of enterprise software product that implements the FSM is known as the Business Process Manager (BPM). The BPM also provides its own declarative language for the business analyst to specify the process. Each state may correspond to a business rule to determine how to trigger the next state. The BPM language is often XML-based. The example below shows a subset of states a BPM might handle in a typical e-commerce scenario.

```
<process-definition name="purchase process">

  <start-state name="request a purchase">
    <transition to="evaluate"/>
  </start-state>

  <state name="evaluate">
    <!--...-->
    <transition name="approve" to="approved"/>
    <transition name="disapprove" to="done"/>
  </state>
```

```
  <fork name="approved">
    <transition to="decrement inventory" />
    <transition to="credit seller" />
    <transition to="deduct from buyer" />
  </fork>

  <state name="decrement inventory">
    <!--...-->
    <transition to="done" />
  </state>

  <state name="credit seller">
    <!--...-->
    <transition to="done" />
  </state>

  <state name="deduct from buyer">
    <!--...-->
    <transition to="done" />
  </state>

  <end-state name="done" />
</process-definition>
```

The BPM script could manipulate variables, start or end parametrized tasks, or even make references to the external rules engine.

The programming in the Smart Business Contract script can now be simplified to a series of declarative statements to check the FSM state.

```
// pid is the ID of a process
// associated with a shopping session
// It is stored in the distributed DB
if (pid) {
  process = load_process (pid);
} else {
  process = start_process("purchase.bpm");
  pid = process.id;
  // Save pid to the DB
}
```

```
while (process.next()) {
  if (process.state == “credit seller”) {
// Do the transaction …
  }
  if (process.state==”deduct from buyer”) {
// Do the transaction …
  }
  … …
}
```

Widely used enterprise middleware BPM solutions include the jBPM, Enhydra Shark, and OpenSymphony OSWorkflow.

3.3 The Distributed Database

Both the rules engine and the BPM need to store its internal data in a database to function efficiently. As the business application grows in complexity, the application itself also needs to manage data outside of the transaction records in the blockchain. That requires a database embedded in all the blockchain nodes. Due to the distributed natural of blockchain applications, this node must also be replicated and synchronized across all blockchain nodes.

Fortunately, distributed database technologies have seen great progress in recent years. It is now possible to build Internet scale distributed databases using off-the-shelf open source software. However, the tradeoff is that those databases are typically NoSQL, and cannot guarantee system consistency at any given time. Instead, they aim for “eventual consistency” as the system gradually resolve potential conflicts. They adopt a very different yet useful conflict resolution strategy than the blockchain itself.

We have preliminarily decided to use the popular Apache Cassandra as the default distributed database for CyberMiles.

3.4 The Distributed File System

The Smart Business Contract often needs to manage files or blobs of data besides blockchain and database records. Those files need to be replicated and accessible across the nodes on the blockchain. So, a decentralized file and data storage system is needed.

The CyberMiles system will use blockchain-friendly distributed file system technologies such as the Ethereum Swarm and IPFS as its standard file storage facility.

3.5 The Distributed Webhooks

As noted above, the business system reacts to external events. The BPM waits for input from contracting parties (known as an "oracle" for the external world state). Then invokes the rules to determine what to do next. After it reaches the next state, it waits for input again. As the ecommerce infrastructure resides on the Internet, the CyberMiles system must have an interface to receive events from the Internet.

To accomplish this, each CyberMiles blockchain node will also embed a web server that can receive external messages and trigger BPM events. Each of the Smart Business Contract application can publish one or more webhook URLs to receive external events. The active nodes on the blockchain registers themselves on the DNS system and can all receive incoming HTTP requests.

4. BLOCKCHAIN

A Smart Business Contract ties together all the business middleware components, and connects them with the transaction ledger maintained by the blockchain. Following the lead of Ethereum, CyberMiles is building a Turing complete virtual machine attached to the blockchain. The virtual machine can be programmed via a JavaScript-like scripting language (similar to Ethereum's Solidify programming language). It can complete tasks such as connecting webhook events to BPM processes, loading business rules, and accessing shared databases (see Figure 1).

For a Smart Business Contract, the developer would need to bundle together the application code, BPM configuration files, webhook configuration, and business rules file, into a single archive and then submit the archive file to the blockchain for processing and deployment. Once the Smart Business Contract is deployed, external systems can access it via blockchain addresses. For example, DApps can build a UI for the application, and it utilizes Smart Business Contract to process all business logic, and have the resulting token transactions recorded in the blockchain.

4.1 Blockchain and Consensus

For the blockchain layer of the CyberMiles system, we aim NOT to reinvent the wheel, but to build on an existing blockchain framework instead. Our main criteria for the underlying technology include the following.

- It must be a community driven and actively developed open source project. That would allow CyberMiles to make changes to the infrastructure software, influence future directions of the software, and contribute back to the community.

- The software architecture must support a clean separation between the core blockchain logic (i.e., the consensus logic) and the application logic to validate transactions. The consensus engine takes care of the process to propose and commit new blocks on the chain; The custom applications validate the transactions, including execution of Smart Business Contracts, and determine which transactions should be recorded in the blockchain. The CyberMiles virtual machine and entire software stack for Smart Business Contracts would be written as custom applications on the blockchain.

- The blockchain consensus engine must have proven performance to scale to consumer-grade applications with millions of users. Ideally, it should be one of the recognized top candidate solutions for Ethereum scalability. In other words, it must be market leading in terms of engineering maturity.

The CyberMiles team has conducted extensive research comparing available blockchain infrastructure solutions. We have tentatively concluded that we will build the first iteration of the CyberMiles blockchain on the Tendermint / Cosmos platform. Figure 2 shows the proposed software architecture on each of the validator nodes on the CyberMiles blockchain. In fact, CyberMiles is already making technical contributions to the Tendermint / Cosmos platform.

- The Tendermint project creates a Byzantine fault tolerant (BFT) consensus engine. It is a very active and well-funded open source effort (after a successful ICO itself). The blockchain itself can withstand up to 1/3 of node failure (crashed or subverted). In a

DPoS (Delegated Proof of Stake) setup, individual validators are strongly incentivized against subverting the network, making Byzantine failures exceedingly rare.

- Tendermint has a modern and modular architecture. The consensus engine can be independently plugged into other types of blockchains. For example, the Ethermint project utilizes the Tendermint consensus engine to scale Ethereum. The ABCI (Applicaion BlockChain Interface) is a simple and clean application logic interface that enables CyberMiles to develop its virtual machine and application stack. As a new transaction comes in, the blockchain would pass it to the CyberMiles application via ABCI; Once the relevant Smart Business Contracts are executed and the transaction is validated by the CyberMiles application, it would be passed back to the blockchain consensus engine for record keeping.

- Tendermint is a high performance blockchain implementation based on the DPoS (Delegated Proof of Stake) consensus mechanism. It is officially endorsed by Ethereum as an Ethereum scalability solution. During tests, it can reliably support 10,000 transactions per second, making it one of the leading engineering solutions.

Due to the underlying Tendermint DPoS mechanism, the CyberMiles blockchain would have a block generation time of under 10s, and transactions in a block are instantly confirmed once the block is committed.

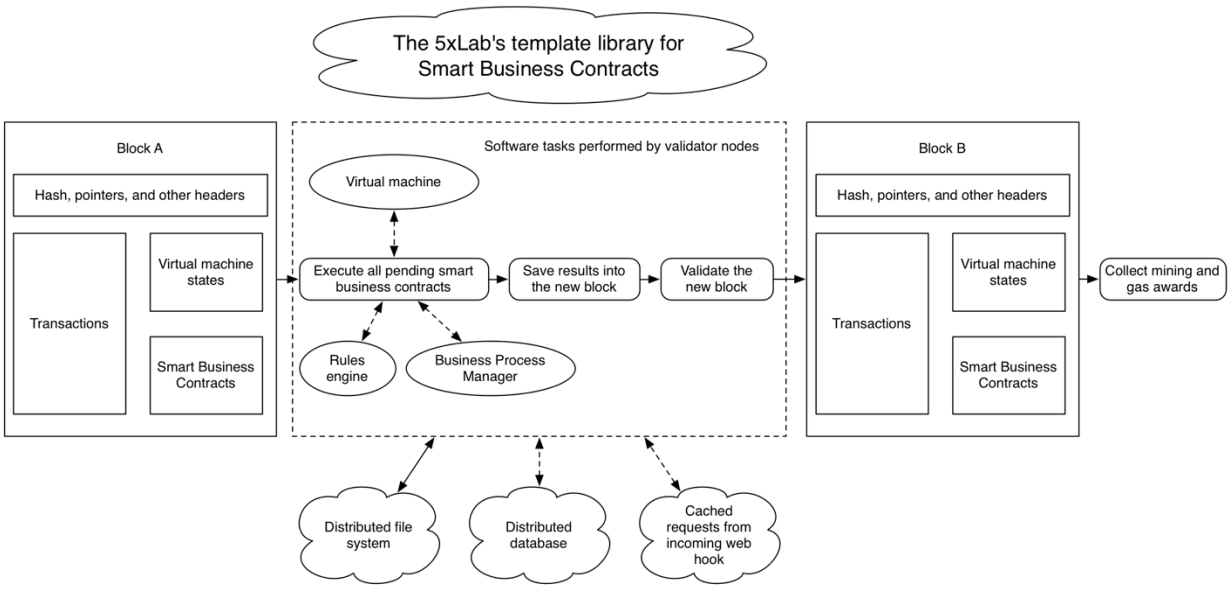

Figure 2. illustrates the work CyberMiles blockchain validators perform.

4.2 The Crypto Token

The CyberMiles blockchain will create and record a native crypto token called the CyberMiles Token (CMT). There are two uses of the CMT: to compensate community members for the services they provide, and to facilitate financial transactions on the network. Those two use cases are also inter-related as the "fee" collected from each transaction settlement is used to pay validators who provide services to facilitate the transaction. Let's now consider the two use cases in detail. First, network participants can earn CMTs by providing services.

- They could become validators to service the network. Specifically, participants would execute Smart Business Contracts for DApps (such as the 5miles app) to earn CMTs (gas paid by the DAapps, see below). Or, they could validate and record new transactions on the blockchain and earn CMTs from the DPoS consensus protocol.

- They could provide services to their peers on the network. Consumers and businesses on the network could use CMT to pay each other for services, such as arbiter services for dispute resolution and even development services for Smart Business Contracts

Note: The CMT would be converted to gas at a dynamic exchange rate – so that the converted value of a unit of gas remains stable. The gas is then used to pay for CyberMiles nodes that execute Smart Business Contracts. The gas price of the Smart Business Contract will be estimated by the system at the time when the Smart Business Contract is submitted to the blockchain.

Second, the CMT could be used as the internal settlement currency on CyberMiles network applications. For instance, a small business loan application (see section 5) could use CMT to settle loans and repayments without a centralized clearing house to ensure privacy, transparency, and fund safety; A supply chain management application could settle intermediate transactions in CMT, and only allow for conversion to fiat currencies for balances at the end of a day. That reduces fiction and transaction costs. In both cases, the network extracts a small fee from each settlement transaction to pay validators that execute the Smart Business Contracts related to the transaction.

Both use cases of the CMT are well-accepted in the blockchain technology community. The CMT is necessary because we are building a new CyberMiles blockchain infrastructure, and hence cannot simply use ETH or BTC for functions native to the new blockchain. The CMT can be compared with some popular tokens that exist today.

Compare with XRP

Like the Ripple network, the CyberMiles network would use its native crypto token to facilitate decentralized settlement of transactions.

However, Ripple “burns” a small amount of XRP for each transaction while CyberMiles would collect the transaction fee to pay validators who execute the Smart Business Contracts associated with the transaction. In addition, the Ripple network is a permission-based blockchain, and all nodes are big financial institutions. The CyberMiles network would be a public blockchain serving small businesses.

Compare with ETH

Like the Ethereum network, the CyberMiles network would reward validators for both creating new blocks and executing Smart Contracts in the block (via gas fees).

However, the current generation of Ethereum is very slow and hence prohibitively expensive to run complex smart contracts. CyberMiles is designed to be highly performant and scalable for running complex Smart Business Contracts. In addition, Ethereum aims to be a general purpose computing network, while the CyberMiles Smart Business Contracts are specifically optimized for e-commerce transactions.

4.3 Jumpstart the Network Effect

For the CyberMiles network to take off, it is essential that reaches the network effect and provides enough value for businesses and miners to join the network. One of the purposes of an ICO is to provide resources to jumpstart the network. We aim to accomplish the following through the CyberMiles ICO.

First, the development team will leverage 5miles’ extensive experience from running one of the largest e-commerce web sites in the USA to build Smart Business Contract templates. There are thousands of contract templates in 5miles divided into 20 major categories. They are tested for real world applications and are readily available for reuse. Furthermore, the system operates a

"store", which is in itself a DApp on blockchain, to sell Smart Business Contract templates developed by third party users. The templates can be priced in CMT or in gas units.

Second, 5miles will create a new application to support small business loans between its 10 million US-based users and small businesses. The application will support decentralized personal identity and credit management, as well as decentralized loan / repayment settlement (without a central clearing house). This effort could potentially move 10 million American user identities and credit histories to the CyberMiles blockchain. This application could provide a blueprint for other developers to take advantage of the user identify and credit history on CyberMiles, and build their own consumer facing applications. By using the CMT as a settlement currency, the CyberMiles network can extract a small fee to pay validators for the execution of Smart Business Contracts related to the loans terms. As more applications are built on CyberMiles, the consumption of CMT gas will increase.

Finally, 5miles will migrate its flagship C2C (Consumer to Consumer) e-commerce application to the CyberMiles blockchain. It will be a DApp managed by 5miles, and supported by Smart Business Contracts on CyberMiles. That could potentially move $3 billion worth of transactions to the CyberMiles platform. As a result, 5miles itself will purchase and consume significant amount of CMT in order to pay the gas cost for running Smart Business Contracts.

5. APPLICATIONS

The CyberMiles blockchain platform would primarily support middleware operations for business transactional applications. As such, it would be mostly beneath the UI of the user-facing application. However, due to the unique characteristics of the decentralized blockchain, its Smart Business Contracts could enable potential new features and applications that were not possible in the world of centralized e-commerce operations.

5.1 A Decentralized Identify Management Platform

As the Equifax hack demonstrated (personal identify and credit history of over 100 million Americans were stolen in 2017), centralized personal identity management creates high risk for consumers and high liability for companies that hold such data. To solve this problem, one must rethink the whole paradigm of identity management. One obvious solution is to let the user have full control of her personal information. The user should be able to decide, on a case by case basis, who have access to her data. The access timing, duration, and accepted use of the data should all be approved by the user. In this case, there will be no central repository of personal information to attack. However, without blockchain-based Smart Business Contracts, such systems are also very hard to implement.

Blockchain networks manage identities through cryptographic keys. The user's "wallets" on bitcoin or Ethereum blockchains are decentralized, and entirely controlled by the user through her private key. Using Smart Business Contracts, we can extend the concept of "wallets" to include a secure deposit of not only crypto tokens, but also arbitrary personal information. Like crypto currency wallets, there could be many "personal identity wallets" on the network. Upon user's request (a transaction signed by the user's private key), the wallet can authorize 3rd party applications to access the data temporarily via the OAUTH protocol. A user can use different wallets for different purposes, just like how crypto token wallets are used today.

The workflow below and Figure 3 illustrate how an "online wallet" for personal information could work. This particular "wallet" stores the user's personal banking information. Hence the user can authorize financial applications on the CyberMiles network to utilize it. An example is the peer to peer small business lending application illustrated in Section 5.2.

1. The user selects a "wallet" app she trusts.
2. The user registers personal information and banking information with wallet.
3. The wallet does AML / KYC validations for government mandated anti-money laundry check.
4. The wallet generates a public / private key pair and then broadcasts the public key to the blockchain for record.
5. The wallet authorizes and tests the banking link.

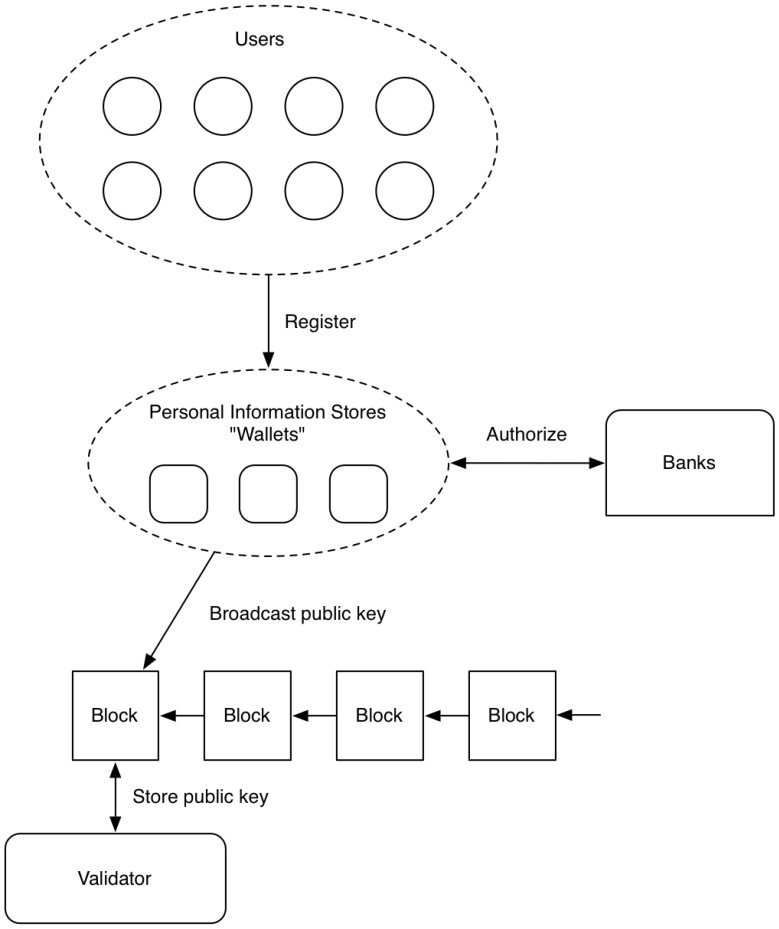

Figure 3. A decentralized identity management platform on CyberMiles

5.2 A Peer to Peer Small Business Loan Marketplace

A potential application built on the CyberMiles' blockchain would be a peer-to-peer small business loan marketplace. As described in section 5.1, we will build a decentralized identity management platform on CyberMiles. The blockchain can then record the credit history for each user identified by her public key.

With the identity and credit history, we can build a loan matching engine (the loan "exchange") on the blockchain. And once loan terms are matched, the Smart Business Contracts would automatically settle the loan directly from each party's bank account using CMT (authorized via their "personal information wallets") without a central clearing house. The workflow below and Figure 4 describes how to match and settle a loan.

1. The user logs into the exchange via OAUTH from her wallet. The exchange caches but not stores personal information.
2. The user submits her desired loan terms (borrow or lend, term, interest rate).
3. The exchange suggests matches.
4. The exchange provides detailed credit scores and histories for matched candidates.
5. If the user selects a candidate. Both parties will need to agree.
6. The loan contract is recorded by the exchange and on the blockchain.
7. The exchange requests the wallets to settle both parties via their bank accounts.

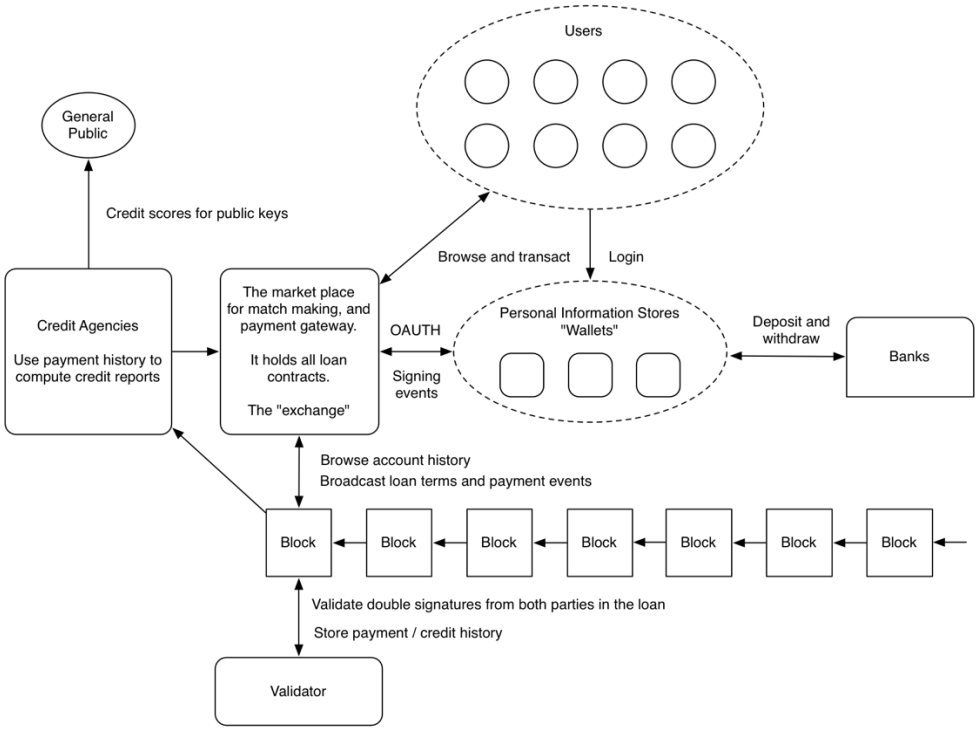

Figure 4. Match and settle a decentralized loan.

Throughout the term of the loan, when a payment is due, the Smart Business Contract would automatically execute the following.

1. The exchange requests both party's wallets to settle payments via their bank accounts.
2. The transaction result is broadcasted to the blockchain, and become part of the credit history.

5.3 Supply Chain Cash Flow

The virtual token in the CyberMiles blockchain system (the CMT) is primarily used to compensate for accessing the network (i.e., the businesses pays to execute their Smart Business

Contracts and validate the blockchain network). However, it could also be used as an in-network medium to settle accounts for parties, including end consumers and sellers, on the supply chain.

Since CMT is a digital token, its settlement would be instant, free, and secure. The CMT allows for highly efficient supply chain management, as the "transaction flow" could happen at the same time as the products move. The parties would only need to convert their CMT balance to other assets periodically through exchanges on the network.

5.4 Certified Products

One of the key features of the blockchain is its ability to keep immutable and secure digital records. This feature helps to address one of the most difficult issues in global e-commerce: counterfeit products.

Smart Business Contracts could be set up for product makers / producers to create authenticity certificates for each of the product items they make (e.g., through an API connection between the factory's production system and the CyberMiles Smart Business Contract). This certificate could then be transparently tracked as the product moves through the supply chain from sellers to buyers.

5.5 Community-based Dispute Resolution

A centralized e-commerce company needs to hire customer services to resolve disputes between buyers and sellers. An e-commerce company building a DApp on top of CyberMiles blockchain could obviously do the same. However, as a decentralized platform, the CyberMiles would offer another compelling solution.

CyberMiles community users could volunteer to become arbiters in exchange of CMTs. Since key steps of the transaction is recorded on the blockchain (including the authenticity certificates of products, and delivery receipts), a Smart Business Contract could develop a mechanism for the arbiter to locate those records at the consent of both the seller and buyer. The Smart Business Contract could hold an escrow pledge in CMT from the seller and buyer pending the conflict resolution. Once the arbiter resolves the conflict and both parties are satisfied, the escrow will be released to the "winning" party and the arbiter will receive a percentage allocation.

GLOSSARY

CyberMiles blockchain: A new decentralized blockchain protocol optimized for business transactions.

Smart Business Contract: A business application that can be executed on the CyberMiles blockchain.

CyberMiles Token (CMT): Crypto currency / token used to award people who host CyberMiles blockchain nodes to maintain the blockchain and execute smart business contracts. Business and parties who submit Smart Business Contracts to be executed on the network must pay CMTs depending on the complexity of the contract.

CyberMiles Validator: A person or entity who contributes computing power to maintain the CyberMiles blockchain infrastructure, including executing Smart Business Contracts. This person

can be anywhere in the world, and s/he may be unaffiliated with 5miles. S/he is incentivized by awards (CMTs s/he can receive as a by-product of maintaining the network).

CyberMiles Application: Any business can build and deploy applications on the CyberMiles blockchain. The business will submit a set of Smart Business Contracts to be executed on the network. The business must buy CMTs to pay the network for access, and could use CMTs to settle or facilitate internal financial transactions.

End User: Buyers and sellers on the 5miles application do not have to be aware of CyberMiles at all. Smart Business Contracts can exchange their USD to / from CMTs immediately before and after the transaction.

5miles: It is a C2C (Consumer to Consumer) e-commerce marketplace application developed by the 5Miles LLC. 5miles has over 10 million US customers with an estimated $3 billion in annual run-rate transaction value.

ACKNOWLEDGMENTS

The 5xlab would like to acknowledge Dr. Michael Yuan and Dr. Lucas Lu for their contributions to this paper.

REFERENCES

[1] Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf 2008.

[2] The Ethereum Team. A Next-Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/White-Paper 2014

[3] Kwon, J. Tendermint: Consensus without Mining. https://tendermint.com/static/docs/tendermint.pdf 2014.

[4] Popov, S. IOTA: The tangle. https://iota.org/IOTA_Whitepaper.pdf 2016.

[5] Zamfir, V. Introducing Casper “the Friendly Ghost”. https://blog.ethereum.org/2015/08/01/introducing-casper-friendly-ghost/ 2015.

[6] Kwon, J and Buchman, E. Cosmos: A Network of Distributed Ledgers. https://cosmos.network/whitepaper 2016.

[7] Wood, G. Polkadot: Vision for a heterogeneous multi-chain framework. https://github.com/polkadot-io/polkadot-white-paper 2016.

[8] Poon, J and Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. https://lightning.network/lightning-network-paper.pdf 2016.

[9] Poon, J and Buterin, V. Plasma: Scalable Autonomous Smart Contracts. http://plasma.io/plasma.pdf 2017.

[10] Teutsch, J and Reitwiebner, C. A scalable verification solution for blockchains. http://bit.ly/2vIConl 2017.

[11] Forgy, C. Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem. Artificial Intelligence. 19: 17–37. 1982.

[12] Oracle. The Java Enterprise Edition Platform. https://www.oracle.com/java/technologies/java-ee.html

[13] Redhat. The Drools Business Rules Management System. http://drools.org/

[14] Sandia National Laboratories. Jess, the Rule Engine for the Java Platform. http://www.jessrules.com/

[15] Redhat. jBPM, a flexible Business Process Management Suite. http://www.jbpm.org/

[16] Lazo, D. OSWorkflow. http://shop.oreilly.com/product/9781847191526.do 2007

[17] DataStax. The Apache Cassandra database. http://cassandra.apache.org/

[18] The Ethereum Team. Swarm, serverless hosting incentivised peer-to-peer storage and content distribution. http://swarm-gateways.net/bzz:/theswarm.eth

[19] Benet, J. IPFS - Content Addressed, Versioned, P2P File System.

[20] Protocol Labs. Filecoin: A Decentralized Storage Network. http://filecoin.io/filecoin.pdf 2017.

[21] The Civic Team. Civic Whitepaper. https://tokensale.civic.com/CivicTokenSaleWhitePaper.pdf 2017

[22] Thomas, S. & Schwartz, E. A Protocol for Interledger Payments. https://interledger.org/interledger.pdf 2015