CommerceBlock

CommerceBlock Whitepaper

An open network for building decentralised commerce

Abstract

Approximately $350 trillion in global assets are held and traded within the confines of the fiat banking system. Limited by a lack of liquidity, transparency, and accountability between centrally managed financial institutions and their customers, capital markets do not operate as freely and efficiently as possible.[**?**]

We believe that the inefficiencies of traditionally styled financial infrastructure limits global trade and economic growth. By moving assets and trade to a public blockchain, we can eliminate the need for trusted intermediaries whose externalities have weighed negatively on global trade flows. CommerceBlock has engineered the first platform for enabling global economic trade in a secure, private, and efficient manner on public blockchains.

All of our current and future product offerings will be based on peer-reviewed research authored by experts from the open source and academic cryptocurrency communities. CommerceBlock has active integrations with clients in the financial and commercial real estate industries.

Contents

Overview

CommerceBlock is a public blockchain infrastructure company that is architecting a platform that allows anyone to build and use financial products and services historically reserved for commercial banking customers. The CommerceBlock network will be the first technology platform that provides a combination of trust minimal trade, decentralised contract execution, on-chain derivatives, and asset-backed token issuance to public blockchains.

Problem

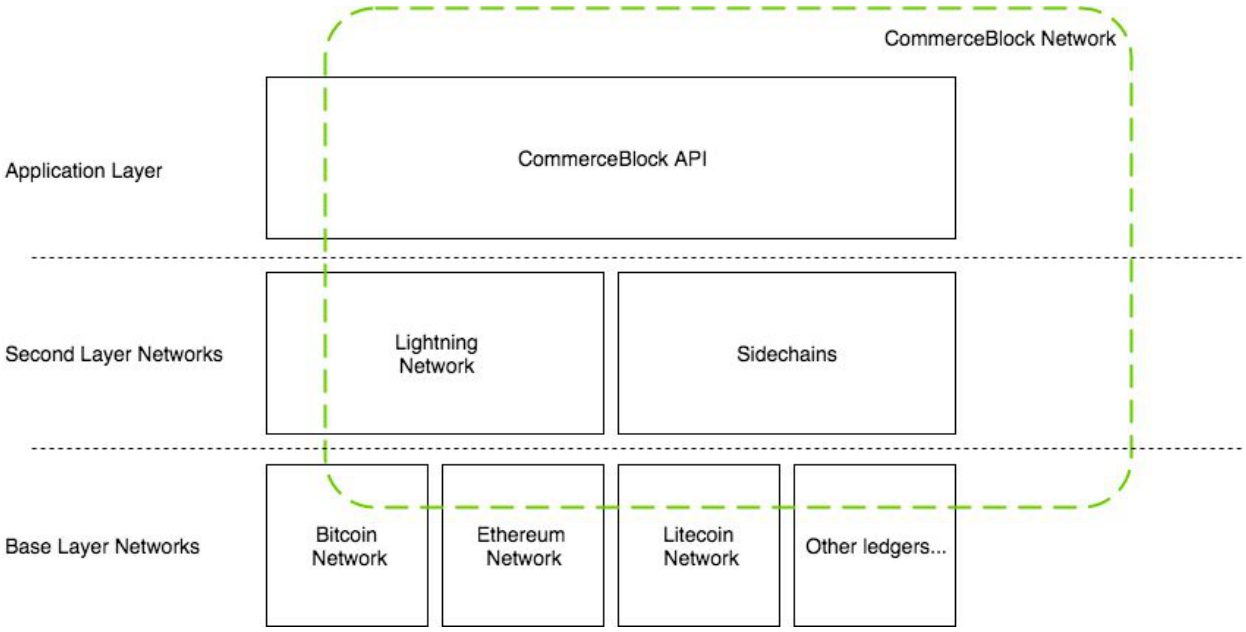
The large majority of commerce still relies on traditionally architected financial products and services. These permissioned systems route and account for value in centrally managed databases. These custodians and their services are subject to censorship, seizure, failure, hacking risk, monopolistic pricing, and other externalities.

Solution

The advent of the Bitcoin protocol [**?**] has enabled permission-less financial innovation. CommerceBlocks product offerings provide a suite of tools that enables anyone to build and use services that construct contracts, manage trade flows, engage in multiparty dispute management, issue assets, and hedge currency risk. Developers and end users will be able to manage all stages of a business interaction and fulfil their contractual obligations by utilising the CommerceBlock platform.

These services will be integrated into infrastructure we have already implemented. [?] We have released the first open-source implementation of the pay-to-contract and homomorphic address protocol outlined by Timo Hanke and Ilja Gerhardt. [?] The protocol has been designed in such a way that all business logic, customer funds, and trade details are managed on the client side meaning at no point does CommerceBlock have access to customer funds or private information.

Architecture overview



Building applications that leverage public blockchains must be done with the utmost care. There is a long history of poorly formatted transactions, improperly seeded signatures, overpaid fees and other such mistakes that have caused irreversible loss of funds. The CommerceBlock platform has been designed to abstract away these intricacies to end users, while still allowing them to build complex applications.

The CommerceBlock APIs and SDKs will provide a wide array of functionality to users in the form of well-tested libraries and highly available (HA) services. The exposed libraries can interact directly with base layer public blockchains, layered protocols such as the Lightning Network [**?**] and Sidechains [**?**] as well as embedded consensus systems. [**?**]

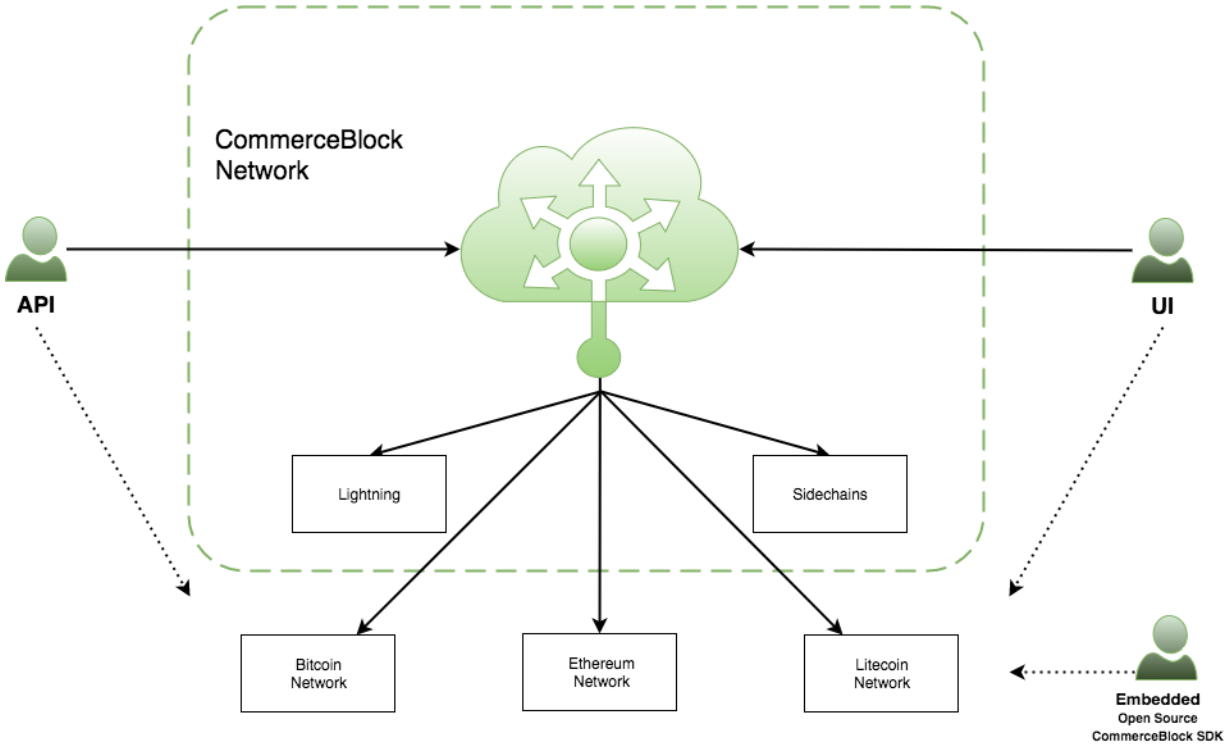
Specifically, the CommerceBlock API will allow users to construct zero-knowledge invoices, conduct multiparty escrow [**?**], engage in templated in-channel contracts on the lightning network [**?**] issue assets [**?**], [**?**], and utilise other carefully selected advanced smart contracting techniques found in peer-reviewed research.

BIP proposal

CommerceBlocks first product offering is based on our implementation of the pay-to-contract protocol described by Gerhardt and Hanke. This protocol is designed to mimic real-world payment interactions between merchants and customers. The protocol results in only the merchant and customer having cryptographic proof of who is being paid and for what.

We have specified and implemented their multiparty key derivation scheme by extending BIP-0032 (Hierarchical Deterministic Wallets). The process has been generalised and made BIP-0043 compliant. We have submitted a draft BIP for approval. [**?**]

CommerceBlock network actors



Much of CommerceBlocks tooling will be available for public use on the CommerceBlock site. Developers who wish to run our tools on their own hardware can download our open source libraries and SDKs. Those who do not want to run their own infrastructure can use our paid-access APIs.

Product offerings

- **Cryptocurrency agnostic APIs, SDKs, and web portal:**
    - The CommerceBlock web portal will provide both prepaid and subscription models for users who wish to access our tools for building smart contracts and issuing/distributing assets.
    - API users will pay per API call.
    - Our SDKs will be available for free download.
- **Trade flow templates:** CommerceBlock provides pre-packaged, paid-access trade flow templates to its clients. These templates abstract away the intricacies and pitfalls of trade management.
- **Enterprise integrations:** CommerceBlock does bespoke implementations of its toolset for enterprise customers. Enterprise customers can use both our APIs and SDKs. Pricing is determined on a case-by-case basis, and revenue shares/partnerships are considered.
- **Third-party integrations:** CommerceBlock has made inroads integrating third-party standards in identity, storage, and security. For example, we will make it easy for developers to integrate Ledger wallets into their product offerings.
- **Commitment to open standards:** CommerceBlock is committed to open standards. All our code will be open-source and available for peer review.

CommerceBlock clients

All of our clients want to trade something. In some cases they want to tokenize the value they are trading, in others they require only contractual agreements and escrowed funds. We provide the tooling to make trust minimal trading possible in a variety of contexts. Below we will describe some of our current clients as well as some use cases we see as strong candidates for integration with CommerceBlock.

Our first client is building an OTC style marketplace where buyers and sellers of Bitcoin can find each other and engage in a B2B exchange of Bitcoin for direct deposit without trusting a third party or being exposed to currency risk. To achieve this goal, the client requires two primary tools that we will provide: homomorphic multisig pay-to-contract addresses and in-channel Lightning Network derivative contract templates.

The flow is very straightforward: after engaging in a trade, the counterparties must agree to the terms of a contract. For example, the contract would specify that the seller is willing to part with $10,000USD worth of Bitcoin after proof has been made of direct deposit. In this hypothetical scenario, at the time of trade this is worth 2 Bitcoin. A homomorphic pay-to-contract multisignature "channeled" address is then derived. This channel will have a counterparty taking an opposing bet in a Contract for Difference, meaning there are 4 Bitcoin in total. The client will provide market makers for these bets to occur.

The funds are then deposited into the address by the seller. Once the funds have been confirmed, the buyer will proceed to fulfill their contractual obligations. Once proof of payment has been made, the in-channel contract will be closed. If the price of Bitcoin has fallen to $9,500 the buyer will receive 2.1053 Bitcoin to cover the drop in price.

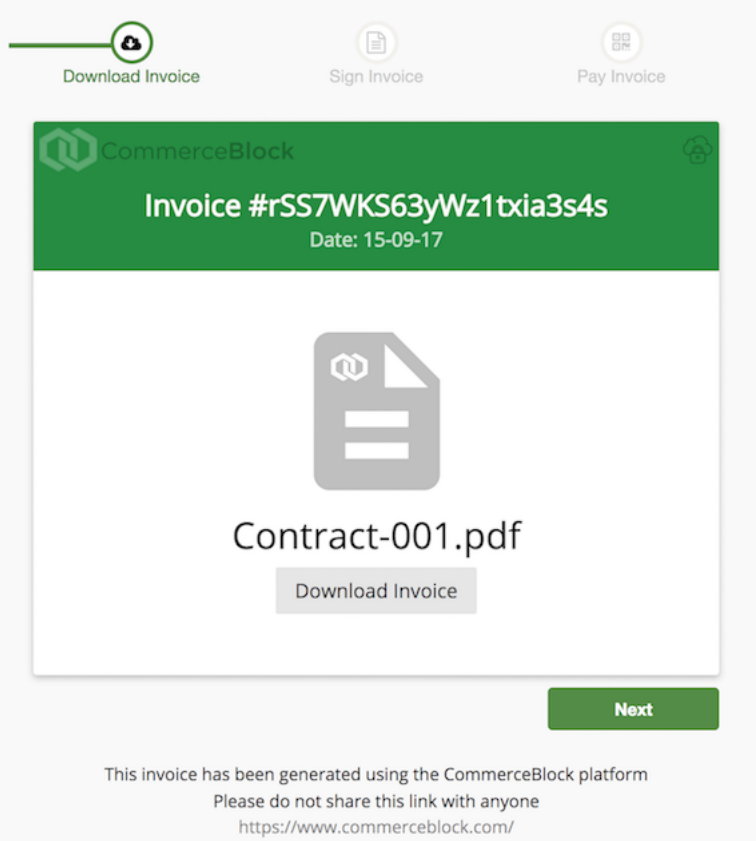
The advantages this client sees from using our tooling are three fold. First and foremost, they have no custodial risk as the counterparties in the trade manage their own keys. This has implications from both regulatory and operational security standpoints. They are no longer considered a custodian and they have no hot wallet risk. This saves them an effectively unbounded amount of legal and operational costs. They also can provide their customers with trust minimal on chain derivative contracts which is otherwise completely unavailable in the market. And finally, the terms of the agreed upon contract are unequivocal. No party can claim they did not agree to the terms, making dispute mediation more manageable.

CommerceBlock will provide API endpoints and SDKs to make all of the above possible. This is just one extremely simple manifestation of our tooling in production. Arguably some of the strongest use cases for our infrastructure will involve Lightning Network based exchanges that allow customers to trade derivatives, cryptocurrencies and tokenized assets without trusting the exchange to hold their keys. The exchange would manage the order book, matching engine and routing without custodial risk.

Our second active Enterprise Integration partner provides a platform for property owners and investors to come together to issue, buy, sell and trade tokenized real estate equity and debt. This partner requires our asset issuance functionality. These asset specific marketplaces can be used as a template for other potential clients. We'd ultimately like to see all sorts of trading platforms built on top of the CommerceBlock platform. For

example, that allow for the exchange of digital assets (in-game items, gift cards, reward points, etc...) and other natural trading pairs to digital bearer assets.

Finally, our Web UI is perfect for payment redirects and one-off trading between peers. Both e-commerce and Craigslist style websites could implement our pay-to-contract invoicing directly into their site or they could redirect users to CommerceBlock for invoicing and settlement. This would be an alternative to using sites like BitPay.



Business risks

A product such as ours poses certain business risks. For example, a traditionally styled infrastructure company is a central point of failure; if it is hacked or falters, sensitive customer data and business logic can be compromised and critical services can go down. Unfortunately, security vulnerabilities and bugs are a reality of software engineering; dealing with peoples funds leaves no room for error, so all of our code will be available for public peer review, and we will also contract third-party firms to audit it. These efforts, combined with the CommerceBlock architecture of client-side fund and data management, have mitigated at least some of our infrastructure risk.

Privacy is also an issue. While payments on most public blockchains lack privacy, we are able to mask any business logic or trade details from public view by using our pay-to-contract implementation. In the future, we will integrate privacy-enhancing transaction protocols into our product offerings.

Scaling is another common concern when building applications on public blockchains. Network costs and wait times for customers using our platform can be variable, depending on the available capacity of the public blockchain and the protocols they are using. With proper fee estimates, replace by fee functionality, transaction cut-through, Schnorr aggregation across inputs, lightning networks, and sidechains, we can sufficiently manage this reality. We understand that the security of a decentralised system requires certain usability trade-offs; transaction backlogs and a healthy free market are required to maintain the probabilistic security guarantees of the system. [**?**] We embrace this reality and will engineer tools that smoothen the rough edges of these security-first distributed systems.

Public blockchains can also experience controversial hard forks, an event wherein a consensus code is modified in a backwards-incompatible way. This results in the cryptocurrency fracturing into multiple coins. In the event of an irreconcilable hard fork, CommerceBlock reserves the right to select which networks its infrastructure and development efforts will support.

In comparison, CommerceBlock has a very unique approach. It is well known that proof-of-work-based blockchain consensus systems have base-layer scaling, security and privacy issues. Attempts to enable advanced smart contracting techniques on chains have further degraded these already weak areas. There are other strong arguments to be made that requiring every full node to execute and validate arbitrarily complex smart contracts is a poor design decision. [**?**] In addition, transactions cannot be made dependent on real-world input without having a middleman (oracle) produce the data. This implies that the real-world contract logic should exist off-chain, while the actual execution of the funds occurs on-chain.

Our pay-to-contract implementation is the first step in this direction. Merchants and customers can embed their contract into an address, so that the payment itself proves who is being paid and for what. The wider network does not know the details of the contract, but it is provable in zero knowledge. This is done in a way that looks like a completely normal transaction. There is no data bloating the blockchain through OP_RETURN or bare multisig pubkey stuffing; if a dispute arises, there will be no question as to what the contractual obligations are. While blockchains cannot solve all disagreements that occur in meat space, they can certainly make them less ambiguous and easier to mediate. We think the distributed ledger and blockchain-based systems that attempt to include all this information in the smart contract itself are misguided.

Our second step towards this end is advanced templating for in-channel lightning network smart contracts using discrete log oracles. For example, two parties want to enter a bet on the price of Bitcoin (as a hedge); the nature of that bet does not concern the general public (including competition). This requirement for privacy would make working on platforms that encourage programmatic expressiveness and complexity a competitive risk. These parties do not want to associate their funds with a publicly visible smart contract that represents a specific business function, not to mention the fact that the talent pool of those who are competent enough to write smart contracts with complex conditional logic, game theoretic implications, and non-deterministic code execution pathways is completely illiquid. Pursuing such a strategy is not an efficient use of time or money.

These bettors could instead leverage our contract templates to create a structured derivatives bet completely P2P in multisig output scripts. With Schnorr signatures, these transactions will be indistinguishable from other transactions with Schnorr signatures and MAST would allow for elaborate conditional pathway scripting. The bettors would open channels with each other by depositing funds into a multisig address. The channel equilibrium would shift when the contract was up and the discreet log oracle blindly signed off on the final price. It would also be possible for one party to novate and atomically swap their coins with someone who wished to take over their position without prematurely ending the bet. With the modest script tooling available on blockchain systems focused on security and code simplicity, you can execute a plethora of excitingly innovative smart contracts.

Our infrastructure will have advantages for asset issuance as well. Currently, issuing assets on public blockhains results in a total loss of privacy for asset holders and issuers. Whether it is color coins, ERC-20 or a similarly structured asset issuance protocol, tokens can easily be tracked by anyone monitoring the blockchain. These protocols also do not

scale well and create incentive incompatibilities. To address these issues we will utilize a privacy preserving solution, most likely to be implemented with closed sealed sets directly on the main chain. This protocol will allow assets to be hidden from public view while still retaining strong resistance to double spends and attempts at counterfeit.

CommerceBlock Tokens

CommerceBlock plans to issue CommerceBlock Tokens (CBTs), a utility token that will be tracked on a public blockchain. To use services in the CommerceBlock ecosystem, customers will have to pay in CBTs. The token will be initially tracked on the Ethereum blockchain using an ERC-20 smart contract. When a sufficiently viable sidechain or colour coin scheme is available on a more secure public blockchain, we will transfer the value there. We imagine a future in which customers using our infrastructure will also require payment in CBT, creating an ecosystem of applications revolving around CommerceBlock.

CommerceBlock ecosystem

Developers building infrastructure on the CommerceBlock platform will have access to API endpoints in our ecosystem that provide the following functionality:

- **Privacy-preserving invoicing and payment portal:**
    - Merchants and customers receive cryptographic proof that a Bitcoin transaction is associated with a real-world contract in zero knowledge.
- **Smart contract templates and escrow wizard:**
    - Escrow and smart contract templates will be available for trade flow management. These templates will make multisignature dispute mediation easier for B2B, B2C, and P2P trading applications.
    - Templates for in-channel lightning network smart contracts will be available as well. API users in multisignature escrow can engage their funds into swaps, CFDs, and other financial instruments. Extended with atomic swaps, a Bitcoin-backed derivatives market could be constructed on top of the lightning network.

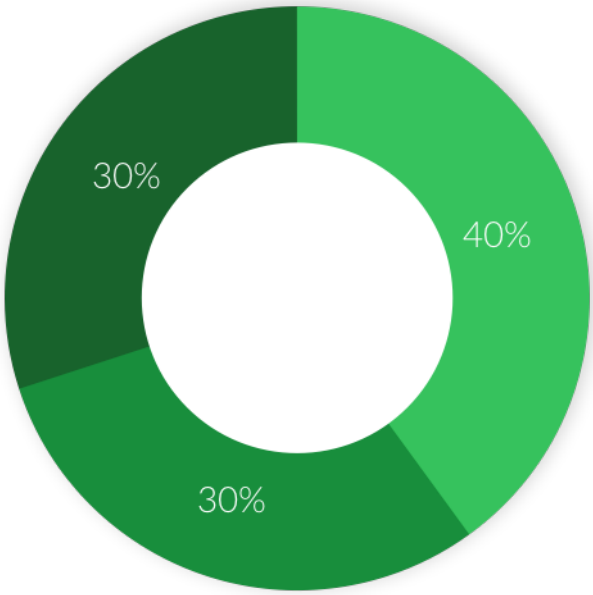
- **Token management and distribution:**
    - Asset issuers will be able to construct and distribute assets on public blockchains in a privacy-preserving manner. The token purchase agreement and other legal information can be directly associated with the token.
- **Data Analysis**
    - Our suite of analytical tools will allow customers to easily analyse their trade history and search for inefficiencies in trade flows.

Token economics

- **Token type** — ERC-20
- **Ticker** — CBT
- **Total supply** — 1,000,000,000 CBT
- **Token sale** — 40% Token Generating Event
- **Payment method** — BTC or ETH
- **Maximum raise** — $25m

Token allocation

- Token sale — 40%
- Partners — 30%
- Company — 30%



The token sale will take place on the CommerceBlock token issuance platform. On the site, users can make payments in Bitcoin or Ethereum. Once the purchasers payment has been confirmed, they will be put in an allocation queue based on the block height that their transaction is confirmed in. Each user will be given a unique deposit address to ensure miners cannot game transaction-ordering and temporarily censor transactions. Tokens will be allocated and distributed asynchronously after being processed through the queue.

A maximum of 40% of tokens will be sold to the public; of that 40%, 20% will be available to pre-sale purchasers, which is 8% of the total token sale allocation. If the pre-sale allocation is not fully consumed, the extra tokens will roll into the wider token sale. If there remain unpurchased tokens from this allocation after the sale has ended, the extra tokens will be provably burned. Any overflow payments will be refunded.

A total of 30% of the tokens will be reserved and used as incentives for our Enterprise Integration partners. It will be easier to sell our services to potential clients if their usage of CommerceBlock is heavily discounted. The tokens allocated for incentives will not be released until the CommerceBlock platform is prepared to take CBTs as payment. Another 30% will be locked up for six months from the closing date of token sales.

Revenue usage

- Software development — 50%
- Operations — 20%
- Marketing — 15%
- Open-source software development — 15%

A large majority of our expenses will be used to fund CommerceBlock engineers tasked with building the platform. Operational costs will include stipends for consultations with legal and regulatory experts, customer relationship management, human resources, and related expenditures. Marketing funds will be reserved for large-scale advertising campaigns managed by experts.

A portion of our revenue will be reserved for incentivising engineers to work on the core protocols of public blockchains. We will pay developers to contribute to some of the most impactful open-source software packages in the space.

Roadmap

We will be ramping up very quickly over the coming months. As stated, we have already completed an open-source implementation and BIP specification of the pay-to-contract protocol. We will be using our own Ethereum smart contracts and token distribution web interface for our token sale planned for November of this year. Once completed, we will be able to provide this infrastructure to our Enterprise Integration partners for use in their ICOs.

Following a successful token sale in November, our immediate priorities will be to hire a larger team, research our token strategy, and complete our first Enterprise Integration. This requires us to extend our pay-to-contract implementation to use BIP 148s P2WSH. More details about this integration will be announced in the coming months. We will have a working private beta for this client by early Q2 2018.

Our dedicated research teams will establish our token strategy and begin drafting a specification for Q1 2018. We will see this infrastructure in beta by Q2 2018. Another team will research and implement an alpha design of our templated in-channel Lightning Network contracts by Q3 2018.

Our philosophy is to build tools and first test them internally and with our Enterprise Integration partners. CBTs will be the first production token on our asset issuance platform, and our pay-to-contract APIs and asset issuance platform will be first tested by our Enterprise Integration partners. We will roll out our products to the public when they have been thoroughly tested.

Team

Having been involved in Bitcoin as early as 2012, the founding team has deep experience in the cryptocurrency space. With engineering backgrounds and complementary experience in investment and traditional banking as well as technology start-ups, the founders are well suited to build out a blockchain-based infrastructure company. The skills of the founding team are greatly enhanced by the hand-picked business development and engineering teams, many of whom have experience in the cryptocurrency ecosystem.

Nicholas Gregory, CEO

Background in technology and finance. Senior roles at Merrill Lynch and JPMorgan. Involved in bitcoin since 2012. Working in crypto space since 2015.

Omar Shibli, CTO

Technology start-up veteran, having had leadership roles at ZocDoc and Eyeview; open-source contributor to crypto projects on GitHub.

Dan Eve, Operations

Seasoned business analyst in a FTSE 50 financial company. Evolved into a cryptocurrency advisor, trader, and miner.

Shachaf Rodberg, Design

UX/UI designer with years of experience and a passion for cryptocurrencies.

Omri Shaiko, Engineering

Open source developer. Started career at EyeView. Passionate for cryptocurrency.

Valerio Leo, Engineering

JavaScript expert and smart contract developer.

Conclusion

CommerceBlock has a compelling story. Our team has domain expertise, and we have clients prepared to pay for our Enterprise Integration services, have released production-ready code, our BIP first is available for peer review, and our roadmap is clearly defined. Combine this with a well-funded marketing campaign, and our company is poised for success.

While success is never guaranteed, it is clear that the disintermediating capabilities of public blockchains present a massive opportunity for financial engineers interested in all areas of the global economy. New and more powerful financial products and services will be designed and implemented over the coming years in all sorts of unexpected developments. While CommerceBlock cannot predict all the ways in which public blockchains will be used to improve global trade, we can, however, position ourselves to be the primary infrastructure provider in a future with permission-less financial innovation.

References

[1] https://www.imf.org/external/pubs/ft/wp/2013/wp1328.pdf

[2] https://bitcoin.org/bitcoin.pdf

[3] https://github.com/commerceblock/pay-to-contract-ui

[4] https://arxiv.org/pdf/1212.3257.pdf

[5] https://github.com/lightningnetwork/lightning-rfc

[6] http://www.jbonneau.com/doc/GBGN17-FC-physical_escrow.pdf

[7] https://adiabat.github.io/dlc.pdf

[8] https://blockstream.com/sidechains.pdf

[9] https://blockstream.com/bitcoin17-final41.pdf

[10] https://petertodd.org/2016/commitments-and-single-use-seals

[11] https://github.com/commerceblock/pay-to-contract-protocol-specification/blob/master/bip-draft.mediawiki

[12] http://www.columbia.edu/~jl4130/BTC.pdf

[13] https://cyber.stanford.edu/sites/default/files/russelloconnor.pdf

Important Notice

THIS DOCUMENT AND ANY OTHER DOCUMENTS PUBLISHED IN ASSOCIATION WITH THIS WHITE PAPER RELATE TO A POTENTIAL TOKEN OFFERING TO PERSONS (CONTRIBUTORS) IN RESPECT OF THE INTENDED DEVELOPMENT AND USE OF THE NETWORK BY VARIOUS PARTICIPANTS. THIS DOCUMENT DOES NOT CONSTITUTE AN OFFER OF SECURITIES OR A PROMOTION, INVITATION OR SOLICITATION FOR INVESTMENT PURPOSES. THE TERMS OF THE CONTRIBUTION ARE NOT THEREFORE INTENDED TO BE A FINANCIAL SERVICES OFFERING DOCUMENT OR A PROSPECTUS. THE TOKEN OFFERING INVOLVES AND RELATES TO THE DEVELOPMENT AND USE OF EXPERIMENTAL SOFTWARE AND TECHNOLOGIES THAT MAY NOT COME TO FRUITION OR ACHIEVE THE OBJECTIVES SPECIFIED IN THE WHITE PAPER. THE PURCHASE OF TOKENS REPRESENTS A HIGH RISK TO ANY CONTRIBUTORS. TOKENS DO NOT REPRESENT EQUITY, SHARES, UNITS, ROYALTIES OR RIGHTS TO CAPITAL, PROFIT OR INCOME IN THE NETWORK OR SOFTWARE OR IN THE ENTITY THAT ISSUES TOKENS OR ANY OTHER COMPANY OR INTELLECTUAL PROPERTY ASSOCIATED WITH THE NETWORK OR ANY OTHER PUBLIC OR PRIVATE ENTERPRISE, CORPORATION, FOUNDATION OR OTHER ENTITY IN ANY JURISDICTION. THE TOKEN IS NOT THEREFORE INTENDED TO REPRESENT A SECURITY INTEREST.