getsirius.io

1 SUMMARY

2 HISTORY

3 CONSENSUS

4 USER EXPERIENCE

5 TEAM

6 PLANNING

SUMMARY

Sirius is a next-generation smart blockchain based on Qtum with a consensus algorithm from the Proof-of-Stake family.

To encourage developer participation, Sirius is distributed for free among developers and blockchain enthusiasts. Its consensus-building features include decentralized governance, enabling fast adaptation of blockchain parameters to changing conditions, the Mutualized Proof-of-Stake algorithm and smart contracts to facilitate any kind of decentralized application, from simple lotteries to the building of whole ecosystems, all running on the Sirius smart blockchain.

To encourage the adoption of the smart contract platform, Sirius will feature a smart contract wizard. This feature will enable bets, lotteries and token crowdsales to be easily organised. This can be done even with no programming experience, finally putting the power of smart contracts in the hands of users, not just coders. Finally, Sirius has a low 1% inflation and dropping, making it the smart contract platform with the lowest inflation on the market.

1 SUMMARY

2 HISTORY

3 CONSENSUS

4 USER EXPERIENCE

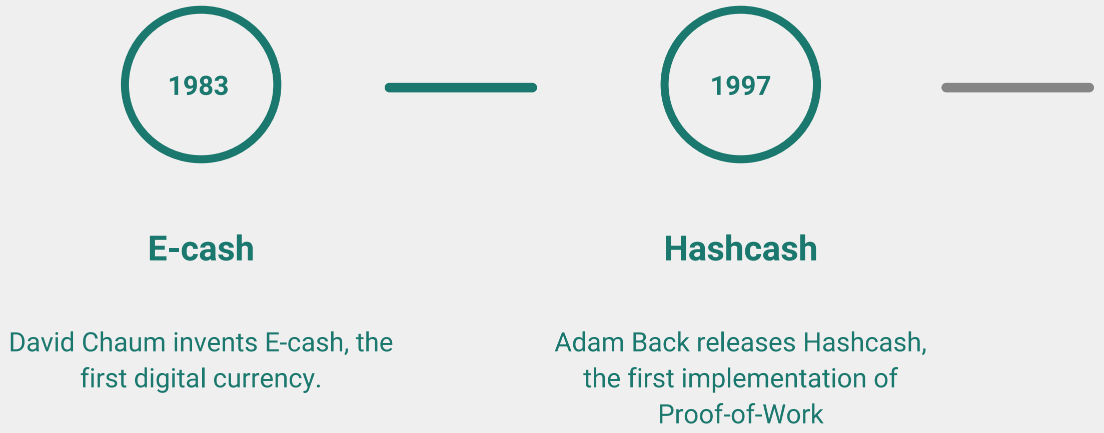
5 TEAM

6 PLANNING

HISTORY

Sirius hails from a long line of digital currencies, its most direct ancestors being Qtum, Ethereum and Bitcoin.

But the idea of digital money goes all the way back to 1983 when a first attempt was developed in the form of e-cash, which was written by David Chaum. In this scheme a central issuer signed a message containing a client-generated random serial number. Even though this was a centralized scheme, it offered a high degree of privacy, and put some more control in the hands of the user. More than a decade later, in 1997, Adam Back invented the Hashcash Proof-of-Work algorithm, which was implemented in a wide array of cryptocurrencies, first and most notably Bitcoin.



1983

E-cash

David Chaum invents E-cash, the first digital currency.

1997

Hashcash

Adam Back releases Hashcash, the first implementation of Proof-of-Work

HISTORY

In 1998, Wei Dai introduced the idea of a decentralized consensus mechanism which incorporated the hashcash proof-of-work function in his proposal of B-money. Money was transferred by broadcasting the transaction to all participants in the network, who then kept record of the transactions in the network. Only a subset of the network participants, called servers, kept track of the transactions. Thus other participants had to verify their balance with the servers before being able to send a transaction. In 2005, Hal Finney incorporated this idea together with computational puzzles into reusable proofs of work or RPoW. A client in the RPoW network would create a token via proof-of-work of a certain difficulty. A central server would then register that token and assign it to the private key used to sign the token creation. This fell short because in the end it still relied on a central point of trust to store the network state.

1998

B-Money

Wei Dai introduces B-Money, featuring the first decentralized consensus mechanism.

2005

RPoW

Hal Finney invents RPoW.

2008

Bitcoin

Satoshi Nakamoto releases Bitcoin.

1 SUMMARY

2 HISTORY

3 CONSENSUS

4 USER EXPERIENCE

5 TEAM

6 PLANNING

HISTORY

Hal Finney went on to work on Bitcoin, which was introduced in 2008 by Satoshi Nakamoto. Bitcoin is administered through a decentralized peer-to-peer network. Transactions are broadcasted to miners, who by solving a Proof-of-Work puzzle can gain the right to add the next block of transactions to the chain. With this, Bitcoin became the first fully decentralized currency.

The use of proof-of-work as the sole consensus mechanism has major drawbacks. The transaction speed is slow and the network uses large amounts of energy. In 2012 Scott Nadal and Sunny King proposed Peercoin. They intended to solve the energy problem by using Proof of Stake alongside PoW. A year later, in 2013, Vitalik Buterin described the first turing-complete scripting language for a cryptocurrency. With this, Ethereum introduced smart contracts and the decentralized application platform ideas that have also been implemented in Sirius. Not much later, Qtum, the first Proof-of-Stake smart contract platform, and Sirius's direct ancestor was released.

2012

2013

2017

Peercoin

Sunny King implements the Proof-of-Stake algorithm for the first time.

Ethereum

Vitalik Buterin introduces smart contracts with the release of the Ethereum white paper.

Qtum

Qtum introduce their smart contract platform, featuring the Mutualized Proof-of-Stake algorithm.

CONSENSUS

The Sirius network powers a decentralized smart contract platform geared towards accessibility and user experience.

In this chapter, we will talk a bit about the technical choices made for Sirius, and how this has resulted in a state-of-the-art platform, ready to compete with Ethereum and EOS, accessible to everyone, not just programmers. To illustrate these, an introduction to consensus algorithms is useful.

PROOF OF WORK

The Proof-of-Work algorithm allows anyone to produce a block of transactions and receive tokens as a reward by doing a calculation with a partially free to choose input, and a certain target answer. Everyone knows the desired answer, but nobody knows the required input.

In this way, Proof-of-Work resembles a game of Jeopardy. However, unlike on Jeopardy, the contestants, or miners, are free to try as many inputs to the calculation as they want. Because of this, the participants want to try as many calculations as possible in the shortest amount of time. This results in a race to acquire more hardware and electricity to power it, using up more and more of both resources. To solve this problem, another algorithm was invented where only participants who are holding some tokens are allowed to participate. This algorithm is called Proof-of-Stake, and a modified version of it powers the consensus logic for the Sirius network.

CONSENSUS

PROOF OF STAKE

Proof-of-Stake differs in a few ways from Proof-of-Work. The idea works as follows. Participants are only allowed to try a calculation if they hold some tokens. They then use the proof that they hold these tokens as part of the input. The difficulty of the calculation becomes lower if more tokens are used, the 'stake'. Additionally, they can only try one input every so often, so the resource use problem is resolved. Unlike Ethereum, which currently has a hybrid Proof-of-Work / Proof-of-Stake version in the test phase, Sirius is fully Proof-of-Stake right now.

However, to ameliorate some theoretical attacks, Sirius uses a modified version of this algorithm called Mutualized Proof-of-Stake, where half of the reward is paid immediately, and the other half about a day later.

DAG BLOCKCHAINS

Recently some newer digital currencies such as IOTA and Dagcoin have seen an upsurge in popularity due to their novel use of directed acyclic graphs as the underlying chain structure. We are purposefully steering clear of this technology because of its inability to offer consistency, meaning that the full list of transactions will differ from node to node, creating a data structure unsuited for cryptocurrencies. Finally, current implementations are centralized and so offer no real benefits compared to other centralized forms of digital money.

CONSENSUS

DECENTRALIZED GOVERNANCE

Despite both PoS and PoW algorithms enabling a great deal of self-regulation for the network, past events in the Bitcoin and Ethereum communities have shown that it is necessary to provide additional decentralized means of controlling the network. Bitcoin and Ethereum have suffered from constant forks and community divisions over blockchain parameters due to a lack of such means. Sirius includes decentralized governance features from release, which enables the community to democratically decide on changes to important parameters such as block size, and the minimum price of gas, which is a unit denoted in hundredths of a mSIRX. This will allow any desired changes to the blockchain to be implemented in a stable and controlled way.

TRANSACTION MODELS

Bitcoin-based currencies and Ethereum differ in a number of ways. One of these is the way they keep track of transactions. Where as Ethereum uses an account-based model, where only the account state is modified, Bitcoin uses the UTXO model, where only the outputs of transactions are spent. Sirius uses the UTXO model because it offers a higher degree of privacy than Ethereum's account-based model.

SMART CONTRACTS

Putting all these pieces together, we can start to talk about smart contracts. Smart contracts are really programmable money. For example, betting can be done automatically by sending some SIRX to a smart contract which is programmed to send the money to the winner under the right conditions. It's not hard to imagine how the same mechanism could be used to construct a fully

CONSENSUS

automatic living will. The myriad of other use cases include lotteries, fundraisers, fraud prevention and much more. The current situation reminds many of the early internet. A revolutionary new network is now operational, but the user experience is still far from optimal, and because of this, mass adoption has not occured, despite increasing popularity of buying and holding digital currencies. Only the increased use of smart contract platforms, through an improved user experience, will signal true mass adoption of programmable money.

ACKNOWLEDGEMENTS

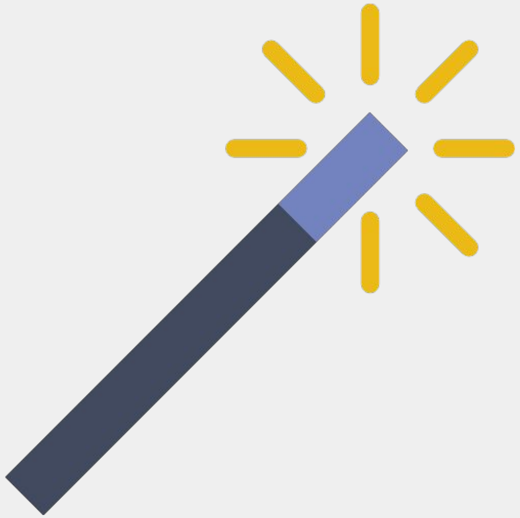
Finishing up, we would like to end this chapter with a word of thanks to the Qtum developers, for developing most of the codebase which Sirius is currently based on, and for doing their part in making cryptocurrencies mainstream.

1 SUMMARY
2 HISTORY
3 CONSENSUS
4 USER EXPERIENCE
5 TEAM
6 PLANNING

USER EXPERIENCE

‘You’ve got to start with the customer experience and work back toward the technology - not the other way around.’

Steve Jobs



ACCESSIBILITY

In order for programmable money to break through to the masses, it must be easy to use for novices as well as programmers. Sirius features the EVM as its engine for smart contracts, the code for which can, among others, be written in the popular Solidity language. More importantly though, the Sirius Core wallet software will include a smart contract *wizard* where a choice can be made between various template contracts, such as bets, lotteries or fundraisers. Contracts will be selected from a pool of already audited contracts, to minimize any security risks. Implementing this wizard will have a high priority on our roadmap.

PLATFORMS

With platform-independency being essential in the multi-device age, next to our desktop-based wallet software, a mobile wallet is being developed and will feature on the roadmap, and we are interested in possible web-based interfaces.

1 SUMMARY

2 HISTORY

3 CONSENSUS

4 USER EXPERIENCE

5 TEAM

6 PLANNING

TEAM

The Sirius team includes people with a wide array of skills, from design, marketing and economics, to physics and mathematics.

To foster further developer inclusion, an extended developer team has been started, which currently contains around 30 developers.

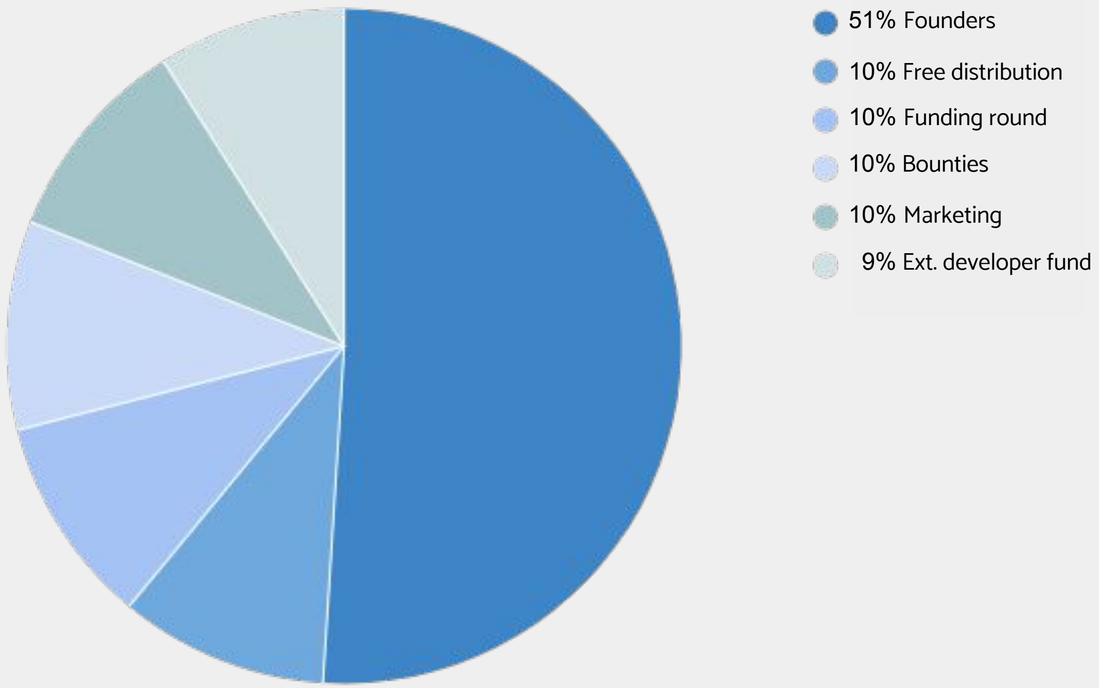
The main team currently includes Andreas Lepidis, Iolar Demartini Junior, Mischa Wiedouw and Axel Karlsson. Extended team members include Ryan Loomba, Mara McLean, David Clutter, and Nicholas Xie. The marketing team is headed by Paul Gruber, with Joseph Stuhlman as our community lead.

1 SUMMARY
2 HISTORY
3 CONSENSUS
4 USER EXPERIENCE
5 TEAM
6 PLANNING

PLANNING

FOUNDATION

We are currently in the process of setting up the Sirius foundation, which will guard the continued development of the software and developer cooperation, as well as serve as the legal entity with which we interact with other companies and foundations.



DISTRIBUTION

A total of 100 million tokens have been created before launch. The current distribution plan is as follows: 51 million SIRX is reserved for the founders. 10 million will be freely distributed through several free distribution rounds, the first of which has already taken place. A further 10 million will go to a donation-based funding round. The next 10 million will go to contribution bounties.

Another 10 million is reserved for marketing.
Finally, 9 million will go to the extended developer team fund.