# BeerToken WhitePaper

# INDEX

# BeerToken Presentation

**BeerToken [Beer] is a PoS utility token powered by Mchain.**

## Basic Specifications:

Max. Supply: 21.000.000.000

Premine: 1% (210Millions, 95% to early Investors – 5% airdrops)

Minimum Maturity Days: 1

Maximum Maturity Days: 7

## Staking Rewards:

Year 1: 1000%
Year 2: 100%
Year 3: 50%
Following Years: 10%

**Powered by Mchain**

# Mchain Presentation

***Mchain*** is an open platform of smart governance destined to maximize the Blockchain technology development, its relationship with cryptocurrencies, the smart contracts world and the decentralized apps inside Smart Cities.

A group of developers in love with this new technology and all around it, supported by official institutions as *Marbella´s townhall and Marbella´s Blockchain Cultural Association*, launched this Blockchain to create technological apps for Smart Tourist Destinations.

## Marbella, 5-star tourist destiny

**Marbella** is one of the most visited places in the world for its luxury and its tourist attraction. Its spectacular growth, which continued through the 20th century and the beginning of the 21st century, has made it a 5-star destination for luxury visitors. In addition, Marbella offers the largest concentration of golf courses in continental Europe.

The commercial attractiveness of Marbella gives it **a place of exception among the most distinguished capitals in the world**, occupying Puerto Banús a prominent place as synonymous with glamor and excellence. The most prestigious international companies such as Lamborghini, Versace, Gucci o Bvlgari have turned this place into an incomparable reference of luxury and distinction.

Thanks to illustrious characters such as the German-Spanish Prince Alfonso de Hohenlohe-Langenburg, the Countess Gunilla Von Bismarck (great-granddaughter of Chancellor Otto Von Bismarck), Count Rudi, Rudolf Graf von Schönburg (founder of the Marbella Club), the legendary actor Sean Connery or the new monarch of Saudi Arabia, Salman Ibn Abdelaziz, Marbella has become the epicenter of luxury throughout Europe.

## Mcoin, Mchain's fuel

***Mchain*** counts with its cryptocurrency, Mcoin, which is used to pay the computational resources needed to execute the decentralized applications and the Smart Contracts in this Blockchain.

With the introduction of Mcoin as turistic-use currency, the commercial variables within the framework of Smart City are expanded. Shops and companies are making wider the network that accepts Mcoin as payment currency.

## Smart Contracts & applications for citizenship

***Mchain*** allows shops to generate their own Smart Contracts and create their own token, which they can offer to their customers exchange it for products or services which they commercialize. Points system, event tickets, etc.

***Mchain*** applications are being developed inside ***Smart City*** environment which adapts new ways for decision making for the local government (data, analysis and knowledge) and new formulas which allows the local government cooperation with the different agents which are part from the Smart environment for the city development.

## Technology

***Mchain*** supports UTXO account systems, Smart Contract and consensus mechanism POS same as segregated witness procol SegWit and decentralized autonomous protocol. Its open source, visible to anyone in the Github´s repository code and its based in a open source platform.

***Mchain*** is compatible with Ethereum Smart Contracts, therefore it not only gives users a safe cryptocurrency experience. It also allows massive third parts contracts and DApps which allows users to experience the functionality and the charm of Blockchain technology.

It also allows **Proof-of-Stake** support, making possible create new blocks with low power devices as Raspberry Pi. *This is much more respectful with the environment than Bitcoin POW mining.*

### 1.1.1 ¿Why Mchain uses UTXO account systems?

In the **UTXO** model, transactions use as input unspent Bitcoins that are destroyed and as transaction outputs, new UTXOs are created. Unspent transaction outputs are created as change and returned to the spender [1]. In this way, a certain volume of Bitcoins are transferred among different private key owners, and new UTXOs are spent and created in the transaction chain. The **UTXO** of a Bitcoin transaction is unlocked by the private key that is used to sign a modified version of a transaction. In the Bitcoin network, miners generate Bitcoins with a process called a coinbase transaction, which does not contain any inputs. Bitcoin uses a scripting language for transactions with a limited set of operations . In the Bitcoin network, the scripting system processes data by stacks (Main Stack and Alt Stack), which is an abstract data type following the **LIFO** principle of Last-In, First-Out.

In the Bitcoin client, the developers use is Standard() function to summarize the scripting types. Bitcoin clients support: **P2PKH** (Pay to Public Key Hash), **P2PK** (Pay to Public Key), **MultiSignature** (less than 15 private key signatures), **P2SH** (Pay to Script Hash), and **OP_RETURN**. With these five standard scripting types, Bitcoin clients can process complex payment logics. Besides that, a non-standard script can be created and executed if miners agree to encapsulate such a non-standard transaction.

For example, using **P2PKH** for the process of script creation and execution, we assume paying 0.01BTC for bread in a bakery with the imaginary Bitcoin address ”Bread Address”. The output of this transaction is: **OP_DUP OP_HASH160 OP_EQUAL OP_CHECKSI**G The operation **OP_DUP** duplicates the top item in the stack. **OP_HASH160** returns a Bitcoin address as top item. To establishes ownership of a bitcoin, a Bitcoin address is required in addition with a digital key and a digital signature. OP_EQUAL yields TRUE  if the top two items are exactly equal and otherwise FALSE.

## 1.1.2 ¿Why Mchain uses UTXO account systems?

Finally, OP_CHECKSIG produces a public key and signature together with a validation for the signature pertaining to hashed data of a transaction, returning TRUE if a match occurs.

The unlock script according to the lock script is:

**<Bread Signature> <Bread Public Key>**

The combined script with the above two:

**<Bread Signature> <Bread Public Key> OP_DUP OP_HASH160**
**<Bread Public Key Hash>O  P_EQUAL OP_CHECKSIG**

Only when the unlock script and the lock script have a matching predefined condition, is the execution of the script combination true. It means, the Bread Signature must be signed by matching the private key of a valid Bread Address signature and then the result is true.

Unfortunately, the scripting language of Bitcoin is not Turing-complete, e.g., there is no loop function. The Bitcoin scripting language is not a commonly used programming language. The limitations mitigate the security risks by preventing 7 **https://en.bitcoin.it/wiki/Script Smart-Contract Information- & Value Logistics** 5 the occurrence of complex payment conditions, e.g., generating infinite loops, or other complicated logic loopholes.

In the **UTXO** model, it is possible to transparently trace back the history of each transaction through the public ledger. The **UTXO** model has parallel processing capability to initialize transactions among multiple addresses indicating the extensibility. Additionally, the **UTXO** model supports privacy in that users can use Change Address as the output of a UTXO.

The target of **Mchain** is to implement smart contracts based on the innovative design of the **UTXO** model.

## 1.1.3 ¿Why Mchain uses UTXO account systems?

Versus the **UTXO** model, Ethereum is an account based system8 . More precisely, each account experiences direct value- and information transfers with state transitions. An Ethereum account address of 20 bytes comprises a nounce as a counter for assuring one-time processing for a transaction, the balance of the main internal crypto fuel for paying transaction fees called Ether, an optional contract code and default-empty account storage.

The two types of Ether accounts are on the one hand, private-key controlled external and on the other hand, contract-code controlled. The former **code-void** account type creates and signs transactions for message transfer. The latter activates code after receiving a message for reading and writing internal storage, creating contracts, or sending other messages.

In Ethereum, balance management resembles a bank account in the real world. Every newly **generated block potentially influences the global status of other accounts**. Every account has its own balance, storage and code-space base for calling other accounts or addresses, and stores respective execution results. In the existing Ethereum account system, users perform **P2P transactions** via client remote procedure calls. Although sending messages to more accounts via smart contracts is possible, these internal transactions are only visible in the balance of each account and tracking them on the public ledger of Ethereum is a challenge. Based on the discussion above, we consider the Ethereum account model to be a scalability bottleneck and see clear advantages of the Bitcoin-network **UTXO** model.

**Since the latter enhances the network effect we wish to offer, an essential design decision for the pending Mchain release is the adoption of the UTXO model.**

# 1.2.1 ¿ Why PoS? The Consensus Management

There are ongoing discussions about consensus and which platform meets the needs of respective project requirements. The consensus topics most widely discussed are: ***PoW, PoS , Dynamic PoS9 , and Byzantine Fault Tolerance as discussed by HyperLedger***. The nature of consensus is about achieving data consistency with distributed algorithms. Available options are, e.g., the Fischer Lynch and Paterson theorem  that states consensus cannot be reached without 100% agreement amongst nodes.

In the Bitcoin network, miners participate in the verification process by hash collision through **PoW**. When the hash value of a miner is able to calculate and meet a certain condition, the miner may claim to the network that a new block is mined:

**Hash(BlockHeader) ≤ M D**

For the amount of miners M and the mining difficulty D, the Hash() represents the SHA256 power with value range [0, M], and D. The SHA256 algorithm used by Bitcoin enables every node to verify each block quickly, if the number of miners is high versus the mining difficulty.

The 80 byte BlockHeader varies with each different Nonce. The overall difficulty level of mining adjusts dynamically according to the total hash power of the blockchain network. When two or more miners solve a block at the same time, a small fork happens in the network. This is the point where the blockchain needs to make a decision as to which block it should accept, or reject. In the Bitcoin network, the chain is legitimate that has the most proven work attached.

# 1.2.2 ¿ Why PoS? The Consensus Management

Most PoS blockchains can source their heritage back to PeerCoin10 that is based on an earlier version of Bitcoin Core. There are different PoW algorithms such as Scrypt11, X1112, Groestl13, Equihash [4], etc. The purpose of launching a new algorithm is to prevent the accumulation of computing power by one entity and ensure that Application Specific Integrated Circuits (ASIC) can not be introduced into the economy. Qtum Core chooses PoS based on the latest Bitcoin source code for basic consensus formation.

In a traditional PoS transaction, the generation of a new block must meet the following condition:

ProofHash < coins × age × target

In ProofHash, the stake modifier computes together with unspent outputs and the current time. With this method, one malicious attacker can start a double-spending attack by accumulating large amounts of coin age. Another problem caused by coin age is that nodes are online intermittently after rewarding instead of being continuously online. Therefore, in the improved version of PoS agreement, coin age removal encourages more nodes to be online simultaneously. The original PoS implementation suffers from several security issues due to possible coin age attacks, and other types of attacks.

Mchain agrees with the security analysis of the Blackcoin team and adopts PoS 3.0 14 into the latest Mchain Core. PoS 3.0 theoretically rewards investors that stake their coins longer, while giving no incentive to coin holders who leave their wallets offline.

# 1.3 Mchain Contract and EVM Integration

The EVM is stack-based with a 256-bit machine word. Smart contracts that run on Ethereum use this virtual machine for their execution. The EVM is designed for the blockchain of Ethereum and thus, assumes that all value transfer use an account-based method. **Mchain** is based on the blockchain design of Bitcoin and uses the UTXO-based model. Thus, **Mchain** has an account abstraction layer that translates the UTXO-based model to an account-based interface for the EVM. Note that an abstraction layer in computing is instrumental for hiding the implementation details of particular functionality to establish a separation of concerns for facilitating interoperability and platform independence.

EVM Integration: All transactions in **Mchain** use the Bitcoin Scripting Language, just like Bitcoin. In **Mchain** however, there exist three new opcodes.
– **OP_EXEC**: This opcode triggers special processing of a transaction (explained below) and executes specific input EVM bytecode.
– **OP_EXEC_ASSIGN**: This opcode also triggers special processing such as OP_EXEC. This opcode has as input a contract address and data for the contract. Next follows the execution of contract bytecode while passing in the given data (given as CALLERDATA in EVM). This opcode optionally transfers money to a smart contract.
– **OP_TXHASH**: This opcode is used to reconcile an odd part of the accounting abstraction layer and pushes the transaction ID hash of a currently executed transaction.

Traditionally, scripts are only executed when attempting to spend an output. For example, while the script is on the blockchain, with a standard public key hash transaction, no validation or execution takes place. Execution and validation does not happen until a transaction input references the output. At this point, the transaction is only valid if the input script (ScriptSig) does provide valid data to the output script that causes the latter to return non-zero.

**Mchain** however, must accommodate smart contracts that execute immediately when merged into the blockchain. As depicted in Figure 1 , Mchain achieves this by the special processing of transaction output scripts (ScriptPubKey) that contain either **OP_EXEC**, or **OP_EXEC_ASSIGN.**

When one of these opcodes is detected in a script, it is executed by all nodes of the network after the transaction is placed into a block. In this mode, the actual Bitcoin Script Language serves less as a scripting language and instead carries data **to the EVM**. The latter changes state within its own state database, upon execution by either of the opcodes, similar to a Ethereum contract.

# 2.1 BeerToken Usecases

## Payments

As you will see in our Roadmap we are planning to make a **blockchain based Marketplace** where users will can buy our *own beer brand* for **BeerTokens** and *be rewarded* for, with cryptocurrencies.

We are planning to make **Blockchain based games** on the *Mchain blockchain*, like bet games, rpg games, collectibles games, and much more will be announced once **Mchain** team announce it officially.

So, with **BeerTokens** you will able to purchase items on those futures blockchain based games.

## Rewards

In our future **SmartMap** *powered by Mchain* you will able to be rewarded in **BeerTokens** on those affiliated breweries.

Breweries will able to join into *the Mchain SmartMap* to *reward* their clients with **BeerTokens** for its consumption in the establishment. Users only will need to load the future SmartMap APP and press "*Give me my BeerTokens*".

# 2.2 BeerToken Usecases

## Staking

As we already have mencioned in Whitepaper, **BeerToken** is a **PoS** *token utility*, you will able to Stake your BeerTokens throught the app: **app.beertoken.org**

So holders will have a huge rewards for Stake BeerTokens. We remember:

**Year 1: 1000%**
**Year 2: 100%**
**Year 3: 50%**
**Following Years: 10%**

## Trading

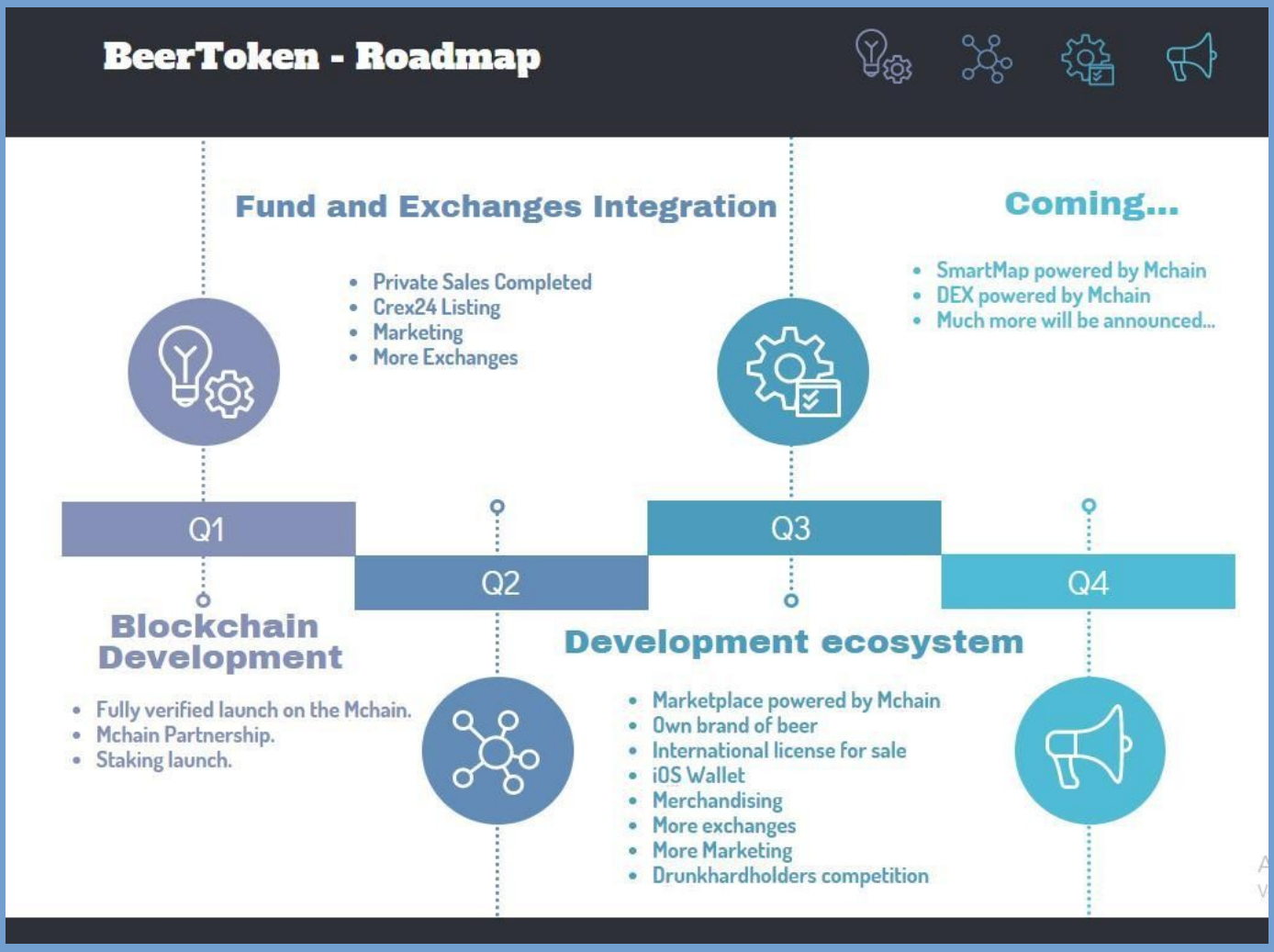
Mchain team is developing a **DEX exchange**, here BeerTokens holders will have *discounts* on trading fees.

Also, users will able to trade in the future *BeerToken pair*.

That's scheduled by Q4 2020. Take a look to our Roadmap below.

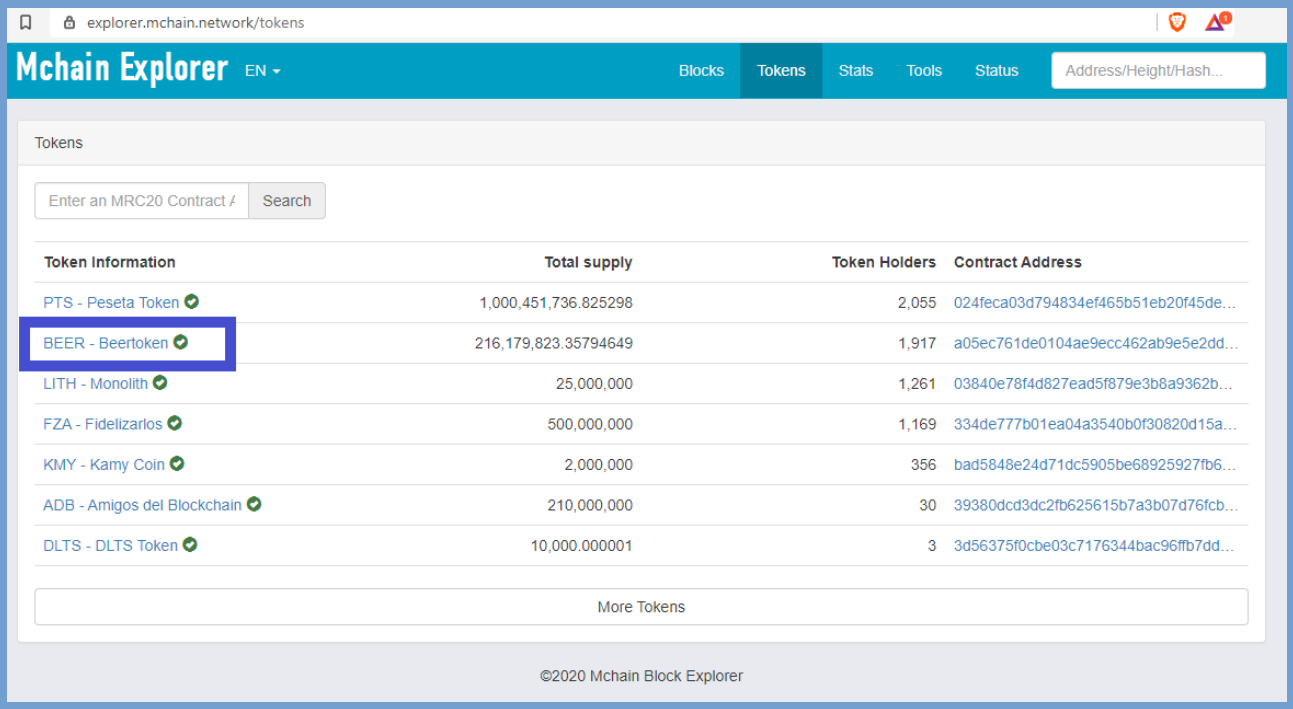
**For more information about this you can visit Mchain Roadmap:**
**https://mchain.network/roadmap**

# BeerToken Roadmap

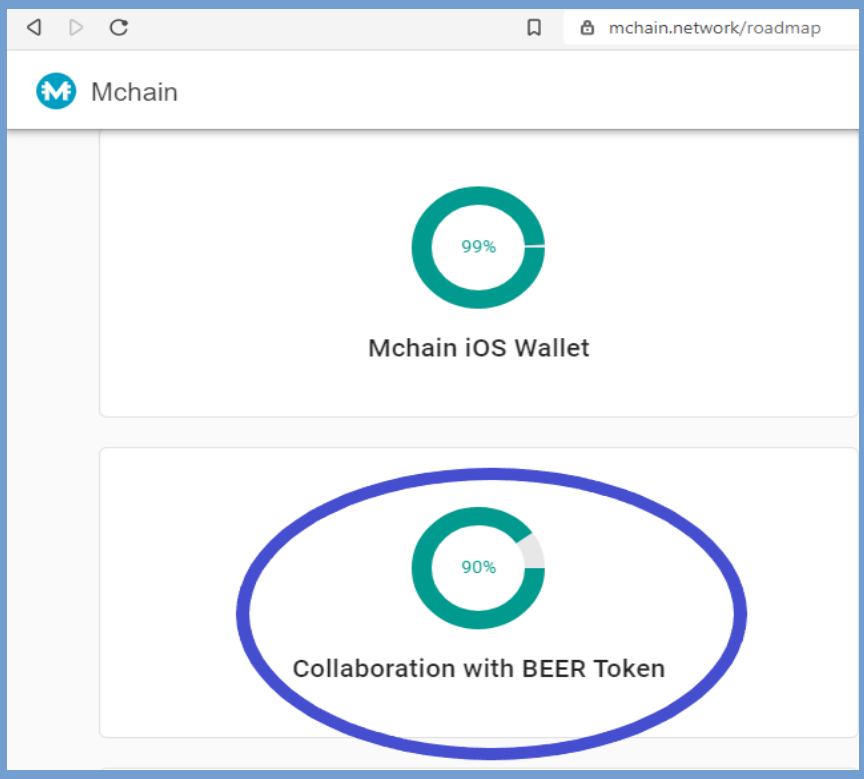


# BeerToken Roadmap – Q1

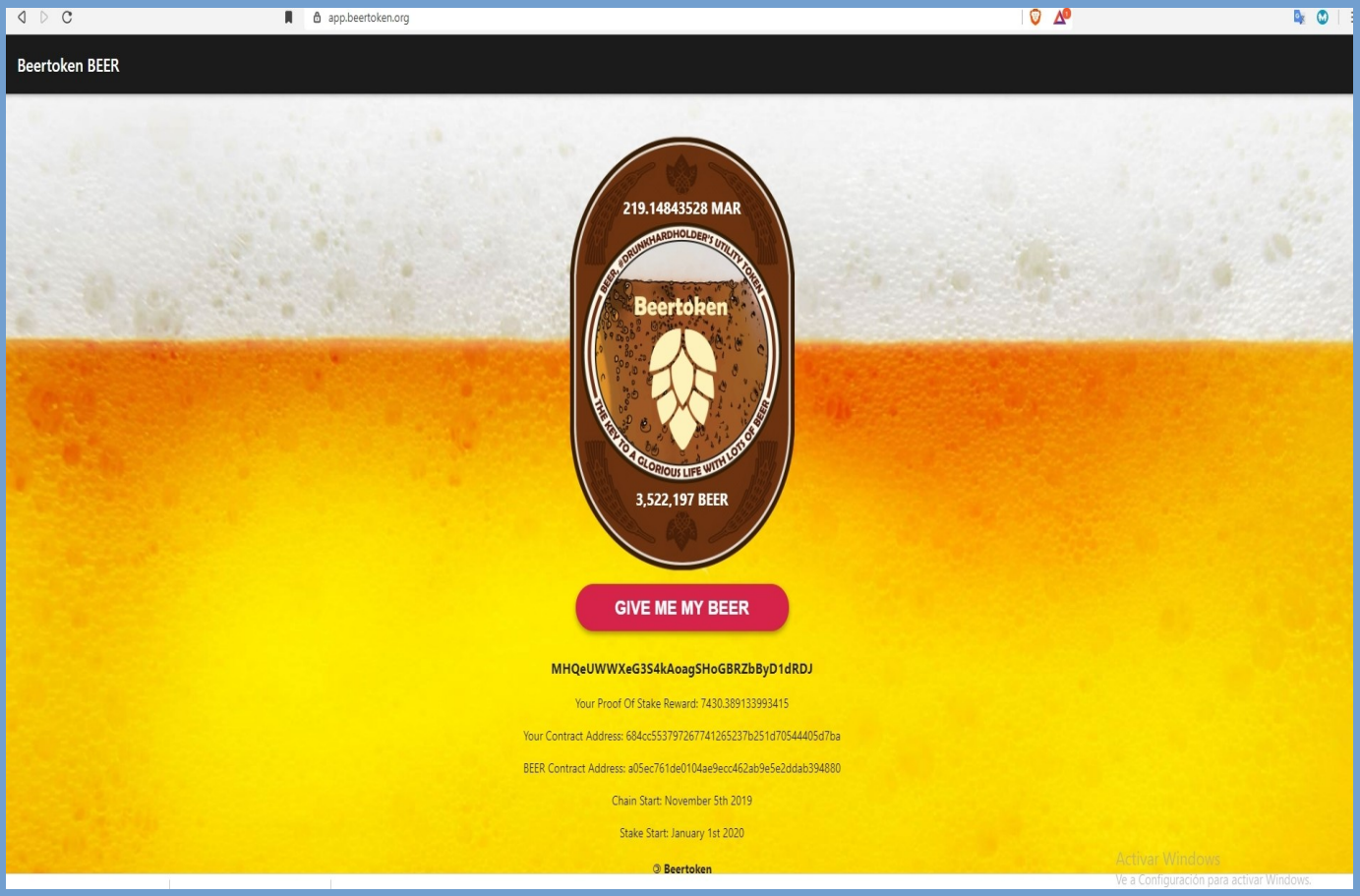
In the **First Quate**r (completed already) we developed BeerToken on the Mchain, fully verified on the ***Mchain***.




**And we Partnered with Mchain, you can see in their roadmap**
**https://mchain.network/roadmap**



**Also, We have launched the *Staking APP*:** app.beertoken.org




For **Stake BeerToken** you will have to install the Metamar google extension https://chrome.google.com/webstore/detail/metamar/efpmcpdedjpalfiidnefpphoemdplpdj

Create a wallet, and send your **BeerTokens** there, for Stake you will need to have **MAR**, *its a gas fee* like Ethereum wallets.

Then, *every 24 hours* press on **“Give me my Beer”** and you will get your Stake!

# BeerToken Roadmap – Q2

In the **Second Quater**, project going to start a **Private Sales Rounds**, based on **3 Sales.**

***Firts Sale*** with a price of 0,0000002 BTC each BeerToken (Max 20M)
***Second Sale*** with a price of 0,0000003 BTC each BeerToken (Max 20M)
***Third Sale*** with a price of 0,0000004 BTC each BeerToken (Max 20M)

After, the sales concludes, we are going to instalist on our **Partnered Exchange Crex24.com (More Exchanges will come, team is aware about and know that Its very important for the BeerToken liquidity)**

Team will contact with ***Influencers and Advisors*** for join into the *BeerToken ecosystem*.

***Marketing*** in general.

***More Exchanges….***

# BeerToken Roadmap – Q3

In the **Third Quarter** team will focus on the development and launch of:

- **Blockchain based Marketplace powered by Mchain**
- Launch of our **Own Craft Beer Brand**
- **Internacional License for Alcohol Distribution**
- i**OS Wallet** powered by *Mchain*
- More Exchanges
- Marketing
- Competitions between ***#drunkhardholders***

# BeerToken Roadmap – Q4

In the **Fourth Quarter** we will announce more information about in the future, but we can announce at the moment:

- **SmartMap** Launch powered by Mchain
- **DEX exchange** launch powered by Mchain
- More Marketing
- More Exchanges
- More will be announced in our social medias

# CONCLUSION

As you can see, BeerToken project is very ambitious, but team is very experienced and is working hard to launch everything already announced and more.

Blockchain industry is very young, in spite of, it has many applications and BeerToken team will work on to make real those application for real world.

Thanks for read this Whitepaper, maybe in the future will be upgraded

Adoption will come!

Follow the **BeerToken** project on Social Medias:

Website: https://beertoken.org/
Twitter: https://twitter.com/Beertokenorg
English telegram group: https://t.me/beertoken
Spanish telegram group: https://t.me/beertoken_es
LATAM telegram group: https://t.me/Beertoken_LATAM
Brazilian Telegram Group: https://t.me/beertoken_br

**References**


1. A.M Antonopoulos. Mastering bitcoins, 2014.
2. I. Bentov, A. Gabizon, and A. Mizrahi. Cryptocurrencies Without Proof of Work, pages 142–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
3. K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, and S. ZanellaB ´eguelin. Formal verification of smart contracts: Short paper. In Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, PLAS '16, pages 91–96, New York, NY, USA, 2016. ACM.
4. A. Biryukov and D. Khovratovich. Equihash: Asymmetric proof-of-work based on the generalized birthday problem. Proceedings of NDSSˆaA˘Z16, 21–24 February ´ 2016, San Diego, CA, USA. ISBN 1-891562-41-X, 2016.
5. B. Bisping, P.D. Brodmann, T. Jungnickel, C. Rickmann, H. Seidler, A. Stuber, ¨ A. Wilhelm-Weidner, K. Peters, and U. Nestmann. Mechanical verification of a constructive proof for flp. In International Conference on Interactive Theorem Proving, pages 107–122. Springer, 2016.
6. O. Bussmann. The Future of Finance: FinTech, Tech Disruption, and Orchestrating Innovation, pages 473–486. Springer International Publishing, Cham, 2017.
7. C. Cachin. Architecture of the hyperledger blockchain fabric. In Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.
8. K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. IEEE Access, 4:2292–2303, 2016.
9. L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos. Non-functional requirements in software engineering, volume 5. Springer Science & Business Media, 2012.
10. K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gun Sirer, D. Song, and R. Wattenhofer. ¨ On Scaling Decentralized Blockchains, pages 106–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
11. N. Emmadi and H. Narumanchi. Reinforcing immutability of permissioned blockchains with keyless signatures' infrastructure. In Proceedings of the 18th In- 26 Alex Norta ternational Conference on Distributed Computing and Networking, ICDCN '17, pages 46:1–46:6, New York, NY, USA, 2017. ACM.
12. R. Eshuis, A. Norta, O. Kopp, and E. Pitkanen. Service outsourcing with process views. IEEE Transactions on Services Computing, 99(PrePrints):1, 2013.
13. R. Eshuis, A. Norta, and R. Roulaux. Evolving process views. Information and Software Technology, 80:20 – 35, 2016. 14. D. Frey, M.X. Makkes, P.L. Roman, F. Ta¨ıani, and S. Voulgaris. Bringing secure bitcoin transactions to your smartphone. In Proceedings of the 15th International Workshop on Adaptive and Reflective Middleware, ARM 2016, pages 3:1–3:6, New York, NY, USA, 2016. ACM.

15. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7):1645 – 1660, 2013.
16. A. Kiayias, I. Konstantinou, A. Russell, B. David, and R. Oliynykov. A provably secure proof-of-stake blockchain protocol, 2016.
17. G. Kotonya and I. Sommerville. Requirements engineering: processes and techniques. Wiley Publishing, 1998.
18. L. Kutvonen, A. Norta, and S. Ruohomaa. Inter-enterprise business transaction management in open service ecosystems. In Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International, pages 31–40. IEEE, 2012.
19. L. Luu, D.H. Chu, H. Olickel, P. Saxena, and A. Hobor. Making Smart Contracts Smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, pages 254–269, 2016.
21. J. Marshall. Agent-based modelling of emotional goals in digital media design projects. International Journal of People-Oriented Programming (IJPOP), 3(1):44–59, 2014.
22. Business Process Model. Notation (bpmn) version 2.0. Object Management Group specification, 2011. http://www.bpmn.org.
23. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Consulted, 1(2012):28, 2008.
24. N.C. Narendra, A. Norta, M. Mahunnah, L. Ma, and F.M. Maggi. Sound conflict management and resolution for virtual-enterprise collaborations. Service Oriented Computing and Applications, 10(3):233–251, 2016.
25. A. Norta. Exploring Dynamic Inter-Organizational Business Process Collaboration. PhD thesis, Technology University Eindhoven, Department of Information Systems, 2007.
26. A. Norta. Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations, pages 3–17. Springer International Publishing, Cham, 2015.
27. A. Norta. Establishing Distributed Governance Infrastructures for Enacting CrossOrganization Collaborations
28. A. Norta, P. Grefen, and N.C Narendra. A reference architecture for managing dynamic inter-organizational business processes. Data & Knowledge Engineering, 91(0):52 – 89, 2014.
cryptorials.io/glossary/x11/
qtum.org
peercoin.net
blakcoin.com
groestlycoin.com
litecoin.info/scrypt
https://gihub.com/ethereum/wiki/wiki/White-Paper
https://en.bitcoin.it/wiki/Script


BEER, #DRUNKHARDHOLDER'S UTILITY TOKEN
Beertoken
THE KEY TO A GLORIOUS LIFE WITH LOTS OF BEER