Coinweb

Names for blockchain,
built on a powerful new platform.

White Paper

Table of contents

Summary

Millions use blockchain today. But billions don't. If blockchain is to become mainstream, it has to become **more powerful**, **easier-to-use** and **scalable**. And **that's our mission.**

We've built a powerful parallel processing system to improve all of blockchain

Our innovative new platform connects blockchains in parallel, making blockchain **more powerful** so that for the first time applications can use more than one blockchain at a time for **more features** and global **scale** using our **dsLayer**. And our dsLogic framework allows a new class of blockchain applications – **dsApps** – to take advantage of parallel processing with greater efficiency and security.

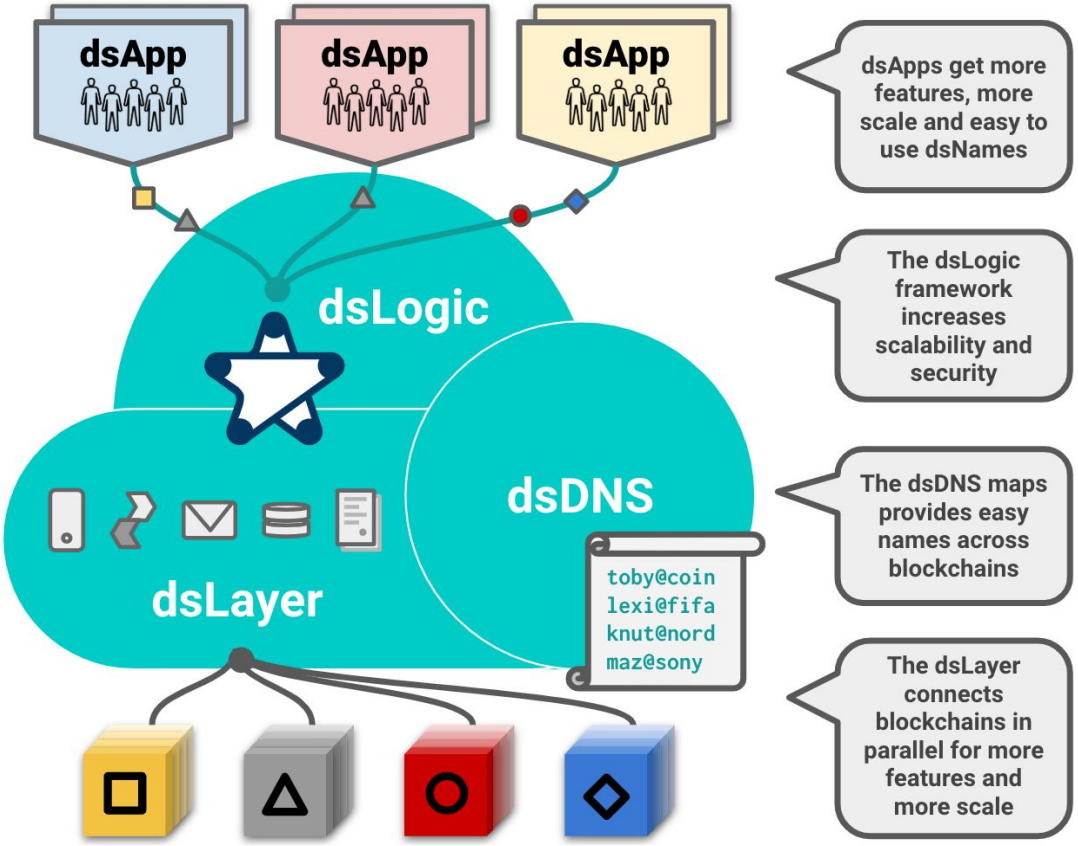
We provide easy to use names for blockchain

The DNS mapped the IP addresses used by the internet into the web addresses we use today. We use our platform to power our dsDNS to do the same for blockchain, replacing confusing hash addresses with familiar names. And we're building it with Paul Mockapetris, inventor of the DNS.

We have a clear path to market adoption

With our powerful platform, businesses can bring the benefits of blockchain to their customers. We've already signed existing companies with strong track records and millions of customers to use our names and our platform, jumpstarting adoption.

Fig. The Coinweb Platform



Evolving blockchain

Bitcoin was the first blockchain and remains the most valuable. But its limitations – slow performance, limited scripting, low throughput, limited privacy – have spawned many competitors, such as Ethereum, EOS, SV and Libra. Others have built Bitcoin add-ons, like Lightning, to improve it. In all, there are thousands of blockchains and tokens vying for success in the market, but instead of driving crypto into the mainstream, they divide the community into rival factions and help keep blockchain from broad adoption.

We connect blockchains, in parallel

Blockchain needs more power and flexibility. No one blockchain has everything: some are secure, but slow. And some are fast, but limited. Building a dApp means picking the blockchain with the least-worst set of trade-offs. Even then, your dApp can be left isolated from other blockchains, and dependent on the success and scale of just one.

Coinweb's innovative platform connects multiple existing blockchains **in parallel**, enabling much greater power and scalability, and includes:

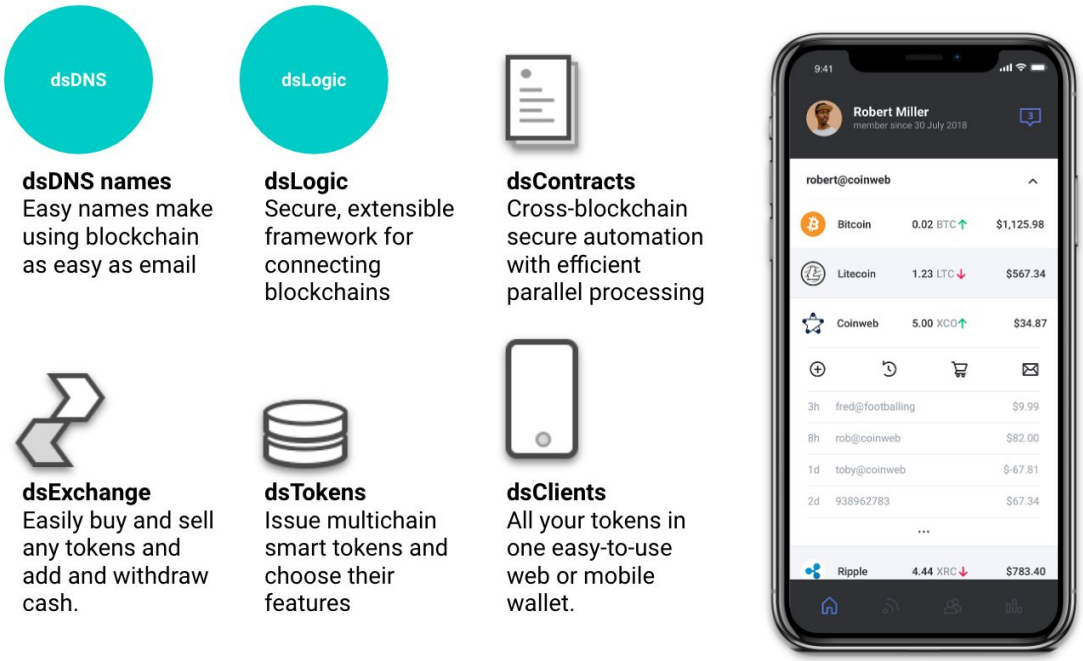
- **Parallel processing across blockchains**, so that dApps can use the best features of different blockchains and can be combined can scale up to handle vastly more users, transactions, and data (chain-based sharding)
- **Parallel processing within blocks**, so that transactions within a block are executed independently of other transactions.
- **Parallel processing within transactions**, so that complex transactions can be processed in parallel by a cluster, removing the limits of current blockchains and without requiring expensive, high end nodes.
- A **fully extensible system**, so that the community can safely and securely build on the platform for even greater gains for blockchain
- A **fully compatible system**, which requires no changes to existing blockchains.

Key platform components

- **dsLayer** connects blockchains so that the best features of each can be used (ds for **d**istributed **s**ecure, meaning it can scale up while remaining secure).
- **dsLogic** provide a logical framework to establish relationships between blockchain entities, making it possible to safely delegate and process actions in parallel for more powerful, more secure, and more efficient dApps.
- **dsDNS** provides easy names across blockchain, making transactions as easy as email

- **dsContracts** provide smart contract-like features, but beyond a single blockchain, so that dApps can leverage as many blockchains and features as they require, and scale up as their needs do.
- **dsTransactions** allow blockchain transactions to span more than one block, making more sophisticated dApps possible, making multi-chain smart contracts possible, and making parallel processing of transactions possible, vastly increasing the efficiency of coinweb nodes.
- **dsClients** provide users all the benefits of the Coinweb platform even on lightweight endpoints like mobile phones and browsers.
- **dsExchange**, built on the platform, makes it easy to exchange tokens

Fig. Coinweb's Platform features



"A POWERFUL, SCALABLE AND SECURE PLATFORM"

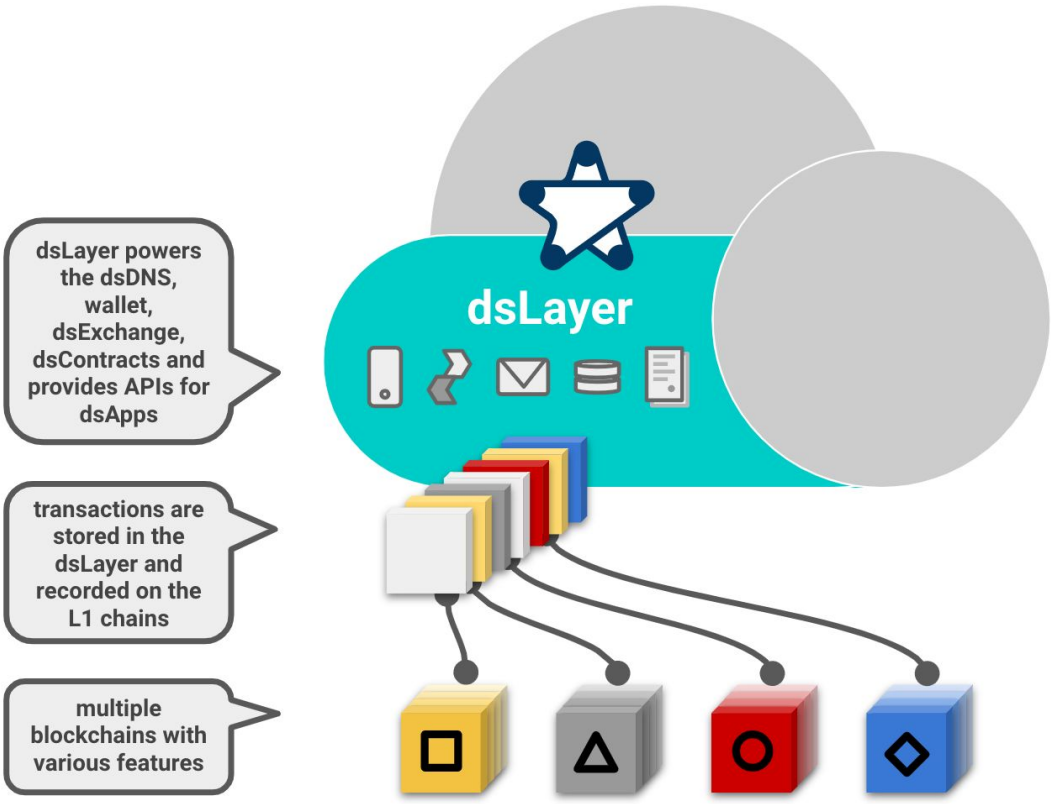
dsLayer

The dsLayer is a layer 2 protocol (L2), which sits above existing blockchains like a hyper-layer and connects (L1) blockchains in parallel. The dsLayer maintains state built up from data embedded in the L1 blockchains, taking full advantage of their security and scale.

- Embedded data on the L1 blockchains serves as an L2 blockchain within an L1 blockchain. (In the same manner, there are L1 and L2 transactions, blocks, block-headers, coins, and so on.)
- To record L2 transactions, parts of an L1 transaction serve as containers for L2 data. The dsLayer reads this data from the L1 chains to add them to the L2 mempool. As with an L1 mempool, invalid transactions are filtered out.
- Each time a new L1 block is mined, an implicit L2 block is created to contain L2 transactions, prioritized according to their size, complexity, type and fees.
- As with an L1 block, an L2 block includes an L2 block-id and an L2 block-header which references the previous L2 and L1 blocks.
- When an L2 transaction has been executed, it moves from the L2 mempool into an L2 block.
- There are no L2 miners, but L2 fees are charged in order to limit spam and set priorities. L2 fees are paid with the Coinweb XCO token.
- The dsLayer state is present in each Coinweb node. Because each node will share the same view of the L1 chains, thanks to L1 consensus, each will hold the same state, the same mempool, and produce the same L2 block.

“CONNECTING BLOCKCHAINS IN PARALLEL FOR MORE POWER, EFFICIENCY AND SCALE”

Fig. dsLayer reading from the underlying chains



Transaction ordering

On any blockchain, there is an implicit order of transactions. i.e whether transaction A came after (is newer) or before a transaction B, and this is shared among its nodes.

In a similar way, to allow the transactions of different blockchains to interact with each other, we define a transaction order *between* blockchains. As block's timestamp is not reliable and each node might receive blocks in a different order, we use the following mechanism to define order:

- We create a special L2 anchor-transaction, or **anchor**, containing as part of its payload an L1 block-id. Because hashes are unique, and a block id is not available before the block is mined, referencing block-ids within blocks proves the block was mined prior to the containing block. This is sufficient to define the partial order required.
- If we can move from a current block to another through a chain of block-id hashes (either from L1 block headers, L2 block headers or anchors), then it is ahead of those that follow.
- There is an implicit L2 genesis block common to every ancestor of every L2 block, meaning every L2 block is greater than (newer than) the common L2 genesis block.

- If there's no chain of block-ids between blocks, we consider them to be non-comparable, and transactions between are not linked.

This *greater than* relation over the subset of all L2 blocks forms a lower-semilattice, and therefore, a partial order over the set of L2 transactions.

Connecting transactions across blockchains

Using this ordering mechanism, we can connect L2 transactions between L2 blockchains:

1. A special type of L2 transaction, the **hop-transaction**, contains as payload a tuple with a blockchain identifier (i.e **btc**, **eth**, **zcash**), and another L2 transaction.
2. When we want to link a transaction **T** from blockchain **A** to blockchain **B**, we issue into blockchain **A** a hop-transaction **T-prime**, containing the tuple ("B",T) as payload.
3. To be accepted into **A's** mempool, nodes check transaction if **T-prime** is valid, meaning:
    a. **T-prime** contains enough funds to fund **T**.
    b. **T** cannot be proven invalid.
    c. Identifier "**B**" refers to a known blockchain.

When transaction **T-prime** is executed it is moved from the L2 mempool into an L2 block. Balances are subtracted to fund **T**, and the transaction **T** is marked as ready to be moved to blockchain **B**.

Once there is an L2 block from blockchain **B** containing an anchor to a descendent of the block when **T-prime** was executed, we copy transaction **T** from blockchain **A** into blockchain **B's** mempool.

- Note that in order to link transactions from blockchain **A** to blockchain **B**, anchors have to point to valid L1 blocks. However, a reorganization of the underlying blockchains can cause valid blocks to become invalid.
- When this happens, L1 reorganizations will be cascaded into L2 reorganizations, potentially spanning several L2 blockchains. Any L2 data linked to a reorganized point will be re-evaluated and a new set of L2 blocks will be computed.
- In order to minimize the impact of reorganizations, anchors pointing to unstable L1 blockchains may be constrained to reference not the latest block, but one of its ancestors. This increases stability at some cost of latency. This parameter is set as a with mid-level FFV consensus.

Blockchain neighborhoods

To reduce communication overhead between blockchains, we limit the pairs of blockchains which communicate with each other. If direct communication is allowed, we consider L2 blockchains **neighbors**.

The neighborhood of an L2 blockchain **A** is the set of L2 blockchains that are neighbors of **A**. By constraining the size of any neighborhood, we ensure the computational resources required to evaluate a single blockchain remain reasonable, regardless of the total number of blockchains supported.

As long as blockchains are connected through a chain of neighbors, a transaction can be connected between them, even if they are not direct neighbors, by recursively including hop-transactions inside hop-transactions.

Coinweb nodes

There are two different Coinweb nodes: full nodes and edge clients (**dsClients**). Full nodes consists of **clusters** of computers running the Coinweb layer 2 stack. Coinweb full nodes act as dsBrokers between dsClients and the rest of the network, trustlessly providing them with computational resources to keep up to date with the L2 state or to compute complex queries.

They also provide the liquidity needed to pay for the native fees of each L1-blockchain.

Full nodes also **broadcast** transactions to the underlying blockchains, charging a fee in XCO

dsClients

Coinweb dsClients are light clients which use the Coinweb system while running on low-end devices or near end users; for example, wallets running in web browsers or on mobile phones, oracles or anything making use of the system's features.

dsClients have two types of connections: to **blockchain** and to **dsBroker**.

Blockchain connections

dsClients connect to the underlying L1 blockchains through light SPV (Simple Payment Verification) to retrieve the last L1 block-header.

These are push-like connections, and dsClients receive information every time a new block is mined.

dsBroker node connections

dsClients also connect to a dsBroker, through a REST-like stateless connection. dsClients communicate with the dsBroker to ask a query, or to submit a transaction. This enables a single dsBroker to serve a large number of clients.

dsClients can freely change to a different dsBroker at any time for better performance or lower fees, using the discovery protocol.

Broker queries

Because dsClients do not have a complete view of all blockchain data, nor compute the MIDS (merkelized implicit data structure) which verify them, they rely on dsBrokers, which are provided by Coinweb nodes.

dsClients use a dsBroker to get information such as account balances, claims or naming information.

This relation is trustless, and dsClients can verify if a dsBroker is authentic.

When a dsClient has a query, or when information changes on the L1 chain, it sends a query to a dsBroker along with an L2 header, and an XCO fee.

The dsBroker will either respond or fail. The reasons a dsBroker node might fail are:

1. **Query too complex.** The dsBroker node can reject a query if it is too complex to answer with the fees provided. The dsClient can retry with a larger fee. This serves as a spam protection mechanism and also incentives the creation and maintenance of dsBrokers.
2. **Broker Out of sync**. The dsBroker has not seen the L2 header the client is querying about. This can happen when a recently mined block has been observed by the client, but not yet by the dsBroker. As dsBrokers are usually better connected with the network than dsClients, this is rare. The dsClient can wait and retry again, and if the problem continues it would mean the dsBroker was disconnected from the network and the dsClient should choose a different one.
3. **Unexpected/timeout-failure**. If a dsBroker (or the network itself) fails for any reason, the dsClient will choose a new dsBroker.

When a dsBroker responds, the dsClient verifies the answer using the MIDS, as described below. Because query responses can be verified, dsBrokers can not lie by giving the wrong information or by omitting the right information.

If a dsBroker fails to answer, the dsClient can retry, or change to a different dsBroker.

If the dsClient is not able to verify a dsBroker response, it will reject it and retry.

Block reorganization

dsClients monitor block reorganization through their L1 connections. If there is a L1 block reorganization, the dsClient will mark its information as to be verified, and the next time this information is needed, (for example, when displayed to the user) the client will update it with a dsBroker.

Evaluating and verifying queries

A dsClient can produce queries with a range of complexity from something as simple as checking the balance of an account, to something as complex as a series of operations over the blockchain with arbitrary turing-complete code. dsBrokers evaluate these queries and return the result to the dsClients, who can verify it.

Query verification:

1. A dsClient makes a request to its current dsBroker node. This request contains a reference to the L2 block header to be used and the query itself.
2. The dsBroker starts running the query and will collect every part of the MIDS it visits while running the query.
3. Once the query has finished, the subset of MIDS that has been visited will be packed and sent as a response to the dsClient as MIDS-prime.
4. The dsClient checks that MIDS-prime hashes are valid, and that its root matches the L2-block header sent with the request.
5. The dsClient will then run the query, using the smaller MIDS-prime instead of the full MIDS. Because these queries are deterministic, the computation will either fail due to a missing part of the MIDS-prime (proving the dsBroker node was not honest) or it will return the same value as the dsBroker, proving it is valid and can be used.

RDoC verification

dsClients can prove anything a dsBroker can compute as long as they have a correct L2 block header. They use a versum-like protocol to minimize the chances they will use an invalid L2 block header. A previous block header is ok, as dsClients can prove a block header is up to date with respect to most recent L1 block header.

The dsClient will query as many dsBrokers as possible for the L2 block headers. If the dsClient receives the same L2 block header from all the dsBrokers, the L2 block header will be assumed to be true. If some of the received L2 block headers differ, the dsClient will use the protocol to find out which of the L2 block headers is invalid.As long as at least one of the L2 block headers is valid, the dsClient can find out which one it is.

Note that in a sybil attack, millions of colluded attackers might flood the network, meaning that a light client might not connect to a single honest dsBroker, and incorrectly believe one of them is valid. To prevent this:

- Each dsBroker is associated with a stake (a proof-of-burn). If it is too low, it is ignored as spam. If it is repeated, then also ignore them as we want responses from different accounts/person/organizations, not just from different physical machines that might be controlled by the same bad actor. This means creating a sybil attack is

extremely expensive. (As a side effect, it will increase the XCO value). Note the stake is always burned regardless of the node being honest or not, which also makes any attack expensive.

- Each transaction from the dsClient is associated to a previous L2 block header. So even if the attacker is successful, the tricked dsClient will become invalid and it won't be able to send any funds. This means an attacker might be able to DDoS a dsClient (by rendering it invalid and/or providing inaccurate responses), but will not be able to steal its money nor trick it to send money.
- dsBrokers are required to sign each reply: "I, broker associated to user/account xxxxx, given the L1 block header xxxxxx and xxxx and xxxx, do confirm that the L2 block header is xxxxxxx." If the dsBroker is a liar, we will have an audit trail.

Our version of the versum protocol, plus these counter-measures, constitutes our **RDoC** (Refereed Delegation of Computation)

"EFFICIENT SECURITY MEASURES FOR GREATER SCALE"

dsLogic framework

dsLogic is an extendable and reusable set of logic propositions compiled into client-side (off-chain) code and on-chain rules from which infrastructure and API definitions can be created, such as indexes and materialized views.

Claims and rules

The dsLayer is an extensible and reusable logic framework. Its key elements include:

- **On-chain claims**, which anyone can read or write, bound to their creator. A claim might be, for example, “Kim owns the domain kardash.com.”
- **Off-chain claims**, stored outside of the blockchain, meaning only the entity that stored it is able to access it. For example, “Sharon trusts the dsDomain registry about the claim that *Kim owns the domain kardash.com*” could be a claim stored in Sharon’s wallet.
- **On-chain rules** create logic infrastructure, as above. An on-chain rule could be “We define that entity E owns a domain D if the dsDomain authority reports that E owns the domain, the billing authority reports that E’s account is in good standing, and there are no competing claims about the domain D in the Trademark Dispute authority.”

Authorities

Each authority could be an on-chain entity (like a dsContract) or an external authority (like an oracle) or something in between (perhaps a dsContract fed by an oracle).

Default authorities could be run by Coinweb, for example to have a human search for trademark violations before granting domain rights, and might require fees. Since everything is transparent, no authority could cheat without being exposed.

Note that the example above outlines a simplified model of the dsDNS system which we are building to provide easy to use names for the blockchain. This shows that using the dsLayer, powerful systems can be built to improve blockchain in fundamental ways.

Key benefits

We can also use the dsLayer to implement key innovations to power our parallel architecture and help unleash the massive potential of a new blockchain platform.

They include:

1. **Safe cross-blockchain scripting: dsContracts**, based on logical programming to eliminate risks such as reentrancy semantics and simplify multi-smart-contract

interactions. Logical relations are compiled into js-client side code, on-chain data entries, on-chain data queries, and self-triggered data rules.

2. **Efficient cross-blockchain transactions: dsTransactions**. Special infrastructure, such as tables, indexes, materialized views and map-reduce-op, are shared between dsContracts. This allows significant scaling compared to most platforms where data structures are not shared.

3. **dsTransactions which span multiple blocks.** Unlike current L1 smart-contracts where transactions must start and finish on the same block, dsTransactions can span several blocks, for example, to implement timeout recovery. This, along with **clustered nodes**, also allows for **massively parallel computation** for much greater scale.

4. **Secure light clients.** Thanks to delegated security, dsClients maintain full security with minimum overhead, without sacrificing privacy.

5. **Extensibility.** Any developer can extend a dsContract and dsLogic. For example if developer A creates the dApp crypto-kitties, and developer B creates the dApp crypto-puppies, then it is possible for developer C to combine the existing state from both dsApps, into a new dsApp, crypto-pets.

6. **Automated load management**. L2 fee policies prevent abuse by malicious actors, and rules are lazily evaluated, meaning the actor attempting to read the output from a dsApp pays for it, preventing spam.

7. **Autonomy.** dsContracts and oracles can awaken on their own, even if nothing has called them, in order to monitor processes or perform background tasks.

8. **Backwards compatibility**. The dsLogic, dsDNS and dsApps are compatible with existing blockchains, so new systems can be built on top of the investments and successes of blockchain today.

"A SECURE, EXTENSIBLE LOGICAL FRAMEWORK FOR BLOCKCHAIN"

dsDNS: Name System

Providing real names for the blockchain.

The dsDNS is a key pillar of what we are building to make the blockchain more mainstream. We all use names to send email, find websites and do business online. The blockchain needs names.

No more hash addresses.

A Bitcoin (BTC) hash address looks like this:

```
1XPTgDRhN8RFnzniWCddobD9iKZatrvH4
```

An Ethereum (ETH) address is no clearer:

```
0xb794f5ea0ba39494ce839613fffba7427579268
```

Hash addresses are hard to type and impossible to remember. Users often save them in a file or on a USB stick, which creates a security risk. Or they write them down and lose them. Even copying and pasting hash addresses is error-prone.

Hash addresses don't prevent fraud. If a fraudster slightly changes a hash address for receiving tokens, it is hard to detect.

dsNames: easy to use names for blockchain

dsNames are easy to remember, like email addresses, and are safer.

- Similar to an email address, a dsName is an address that can receive coins and tokens.
- Every dsName is registered on a blockchain.
- A dsName can be used as an alias for any supported cryptocurrency and to register multiple addresses.
- A dsName can hold tokens and can be used to sign all popular crypto address formats.

Organizations can have their own dsDomain (dsDomain is the part of the dsName that follows the @), such as **@coinweb**. Members of that organization can in turn have dsUsernames (which precede the @) within that domain.

Only dsDomain owners can issue these dsUsernames, which helps keeps transactions secure.

Key benefits of dsNames

Easy	dsNames are as easy to use as email addresses and web addresses, and easier to register.
Secure	dsNames are more than just a mapping of text to a hash address or public key. Repeated use of an address can weaken privacy and security. Our names use best practice cryptographic techniques to keep them safe.
Universal	dsNames are not built on a blockchain for a blockchain — they are built to work across all blockchains, thanks to their integration with the Coinweb **dsLayer** and **dsLogic**.
Compatible	dsNames integrate with the DNS system the internet has relied on since its inception, so current email clients and web browsers already understand them, without plug-ins.

dsDomains

dsDomains are similar to Internet domain names, but don't have a TLD suffix like .com or .net. In a dsName, the dsDomain follows the @ symbol.

To discourage squatting, name registrations incur a fee, adjusted dynamically based on the rate of registrations, and paid in XCO. Anyone can register a new dsDomain, or can choose to buy one on a resale market.

dsUsernames

dsUsernames are unique with a dsDomain and follow established naming conventions. The **dsUsername** is the first part of a dsName, up to its @ symbol. The owner of a domain has the right to register or sell dsUsernames within it.

dsMail

The dsMail service allows users to easily send and receive cryptocurrency, as well as messages. dsMail works across all blockchains, thanks to its integration with dsContracts, dsTransactions, dsLogic and dsLayer.

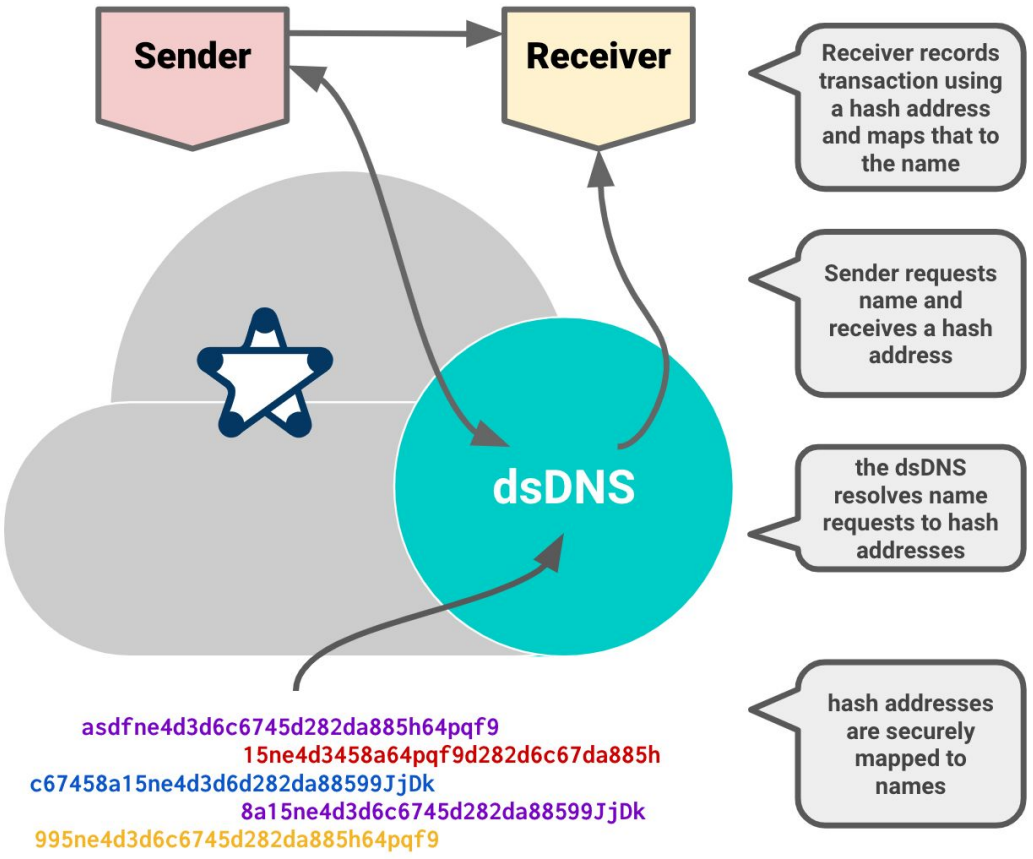
dsDNS and the DNS

Coinweb will maintain an internet domain for users, for example **.xco**. Owners of dsDomains will have use of subdomains of that top-level Coinweb domain.

> **Example 1:** As the owner of the dsDomain **kardash**, Kim also has use of the Internet domain **kardash.xco**.
>
> **Example 2:** As the owner of the dsName **kim@kardash**, Kim has use of the web address **http://kim.kardash.xco**.

The Coinweb dsDNS is developed by experienced **professionals** – such as Paul Mockapetris, who invented the naming system used by the internet – and funded for **long term support** and **continuous evolution**. It's nodes can be run and improved by the community. As such, the dsDNS is also an ideal mixture of centralized leadership from Coinweb and decentralized operation from participating community nodes.

fig. dsDNS name resolution



dsTokens

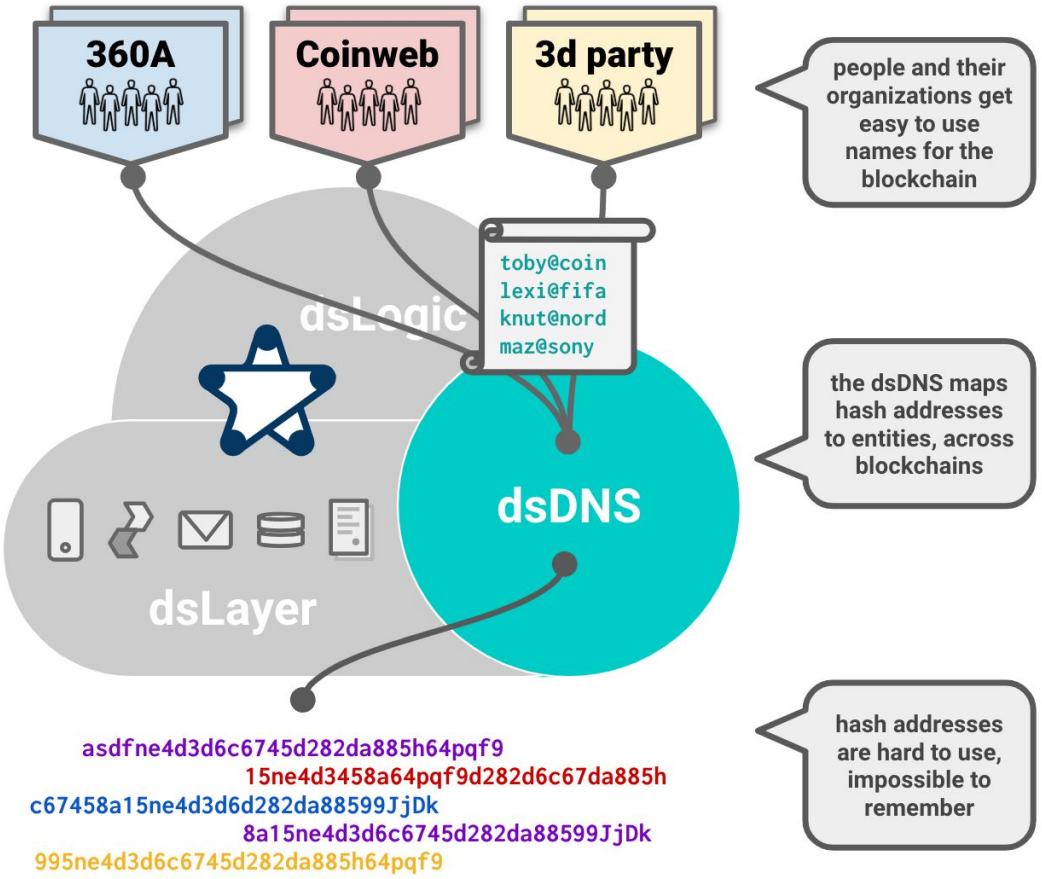
dsTokens are Coinweb tokens that the owner of any dsDomain or dsName can issue. These tokens can be traded and escrowed on any Coinweb-accessible blockchain.

The token name format is *[dsUsername]–[dsDomain]*.

Tokens might be used as coins, assets, trading cards, coupons, shares or something else depending on their use case. They can be utility, security or stable tokens. Tokens can also have a limited supply or be time-locked.

Tokens may be issued in the name of a dsDomain. For example the owner of **@kardash** can issue their own currency, which will be referred to as **kardash**. They may also issue other kinds of domain tokens, such as **kim-kardash** and **khloe-kardash**.

fig. dsDNS (Coinweb Name System)



dsContracts

There are significant scalability limitations in today's smart contracts because they require completion in a single block and interact with the global state of the blockchain.

Distributed smart contracts (**dsContracts**) overcome this limitation. A dsContract is a set of collaborating smart contracts deployed across multiple blockchains, communicating through the dsLayer.

By design, a dsContract does not require the global state. Instead, it works by linking transactions to and from blocks and blockchains in an actor-model fashion.

Similar to Ethereum, dsContract execution requires **gas**, or a fee. Gas is priced in XCO, and if not enough gas is present, the contract stops. Coinweb has a lower and higher gas usage limit for each hour. If less than the lower bound is used, the gas price decreases. If more than the higher bound, the price increases.

Fig. dsContracts

Multi-block transactions

Most blockchains today enforce atomic scripting, meaning that a transaction cannot span more than one block. This keeps the model simple, but also means that blockchain nodes can suffer from high execution load (and high expense), limiting the power of smart contracts.

dsContracts and dsLogic enable transactions which span blocks and blockchains, allowing both parallel processing within a block, thanks to clustered nodes, and between blockchains. In this way dsContracts can be much more powerful, and their execution much more efficient.

When an inter-blockchain transaction is created by a dsContract, it is stored in the MIDS. These transactions can span blockchains and behave like any other external transaction.

When a block is created, it applies the same rules for insertion to all transactions. When a dsTransaction is picked for insertion, it is marked as mined.

Even though a crypto signature can not be made to prove the transaction ownership, knowing that it has been mined indicates its ownership is known.

Efficiences

Multiple transactions can be pooled. This adds efficiency on several levels.

A dsClient can sign several transactions with one signature, saving bytes by including the broadcaster, fee, deadline and signature just once. They also control of the order of grouped transactions, which can be useful in instances where a buy has to be cancelled before a new buy can be placed.

A single node can group together several senders. As there's a fixed overhead cost of broadcasting messages, the marginal cost of adding data is quite low. This reduces both the average cost per message and per sender.

Relayed dsTransactions

dsTransactions from a contract in one blockchain can be relayed into another blockchain in subsequent blocks. If the fees to do so are insufficient, or there is a network disruption, these transactions can stall. Any dsContract transaction can add fees to another transaction in order to restart a stalled dsContract.

Because transaction time-outs are handled in the dsLogic framework, recovery for network failures can be written into dsContracts, and cross-chain transactional behavior, with rollback, can be implemented.

"AUTOMATION ACROSS BLOCKS AND BLOCKCHAINS"

dsExchange

The dsExchange functions like a stock market, where a newly placed order to trade tokens either matches instantly with an existing order, or is placed in the order book awaiting a matching bid or ask.

The protocol matches orders, updates balances, and escrows funds. No middle man is needed, and no accounts need be set up.

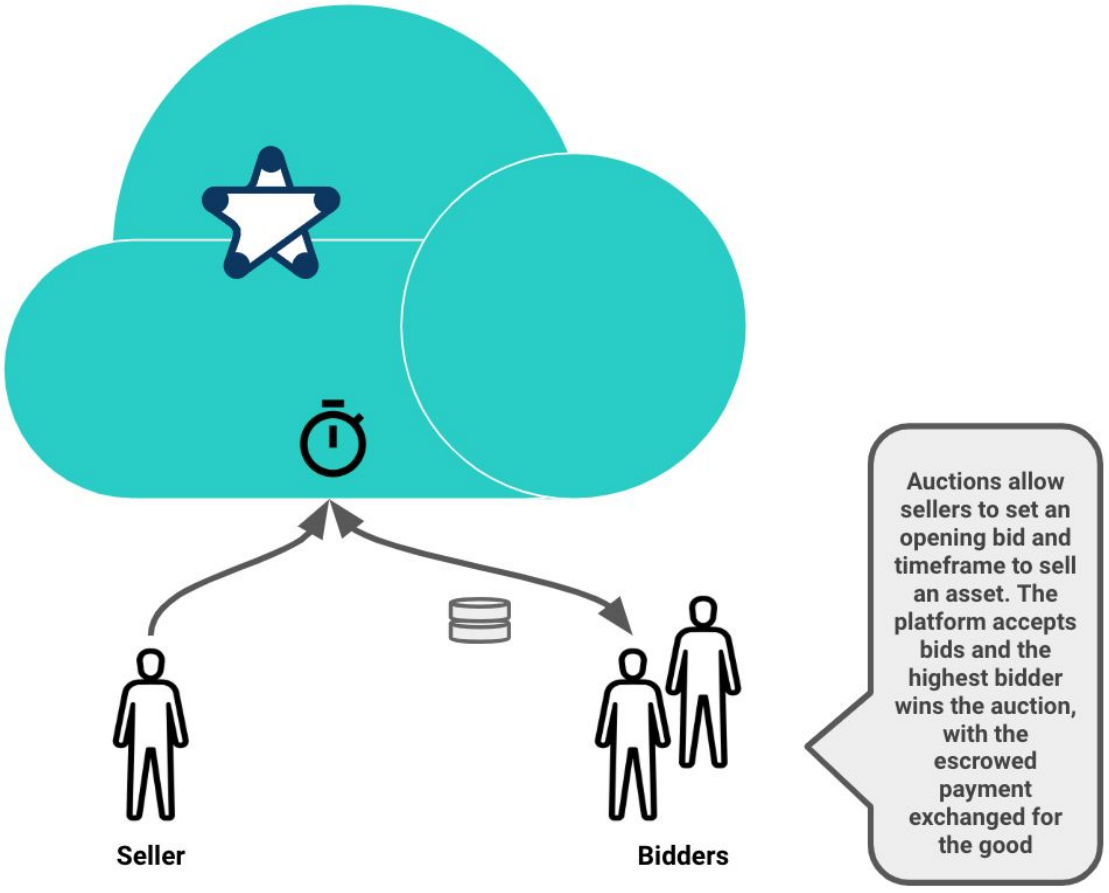
The dsExchange is powered by dsContracts and dsLogic and is built on the Coinweb platform.

Auction contracts

As a DEX is not suitable for all types of trades, sellers can offer auction contracts. For example, if you own a token representing a rare trading card; you can place an **ask** offer on the dsExchange. Since it's rarely traded, you might have little information with which to set the price. Using an auction you can list it for sale and receive the highest price the market is willing to pay.

The auction contract supports several different models, including crowdsales.

Fig. Auction Contracts



XCO: Coinweb's native token

XCO symbol

Coinweb's native coin (token) is named XCO. The X symbolizes the token's cross-chain properties, and CO are the first two letters of Coinweb. (The initial X is also in conformance with the ISO 4217 currency designation standard.)

XCO utility

Coinweb usage fees, such as those incurred when smart contracts are executed, are paid in XCO. The fee protects against spam attacks and ensures the burden on nodes remains manageable. Since XCO fees can be added to each transaction, this eliminates the need to pre-fund the broadcasters.

XCO is the sole token used to pay node servers, and users must pay registration fees for tokens, addresses and names in XCO.

In addition to gas and L2 mining fees, xco pays the dsBroker nodes to fund embedding transactions into the L1 blockchains. It can also be used for:

- As stake for dsBroker-nodes.
- For PoB voting for mid- and long-term consensus.
- For query computation.

"XCO IS A UTILITY TOKEN FOR FUNDING TRANSACTIONS AND REGISTERING NAMES"

Consensus and governance

While projects rely only on the principle that the code is the law, code alone has been proven insufficient for healthy governance. Rather, protocol rules, governance, and community work together to best guide the evolution of a project.

Coinweb governance functions on three levels: short-term (hours), mid-term (months), and long-term (years). The combination of these consensus mechanisms we call **FFV**: Family, then Function, then Verification.

- **Short-term** is how the network reaches a consistent common state. Because the system state is calculated from consensus-consistent L1 blockchains, and computed L2 state is deterministic, nodes will reach a consensus consistent view of the L2 state. (Nodes not able to make this computation can discover it using RDoC).

- **Mid-term** is how the network adjusts system parameters such as the minimal fee for each transaction, the maximum number of transactions per block, the maximum gas and so on. These are agreed upon by the nodes through a proof-of-burn L2 voting. If there is insufficient consensus, then the current default is not changed. (Note that as the system is deterministic, the issues commonly related to proof of stake and proof of burn do not apply.)

- **Long-term** is about the project governance and input from all stakeholders, including Coinweb, so that contentious forks are avoided. Market forces will help guide the project toward succeed and ensure resources are used for continued innovation. In cases of a significant protocol change, PoB proportional voting will be used.

Strong, proven team

Our team has deep experience in crypto and commercial-scale platforms, with decades of successful leadership at Microsoft, Google, Barclays, Cisco, Oracle, and Bitcoin.
We've designed and delivered products which are used by hundreds of millions of people all over the world every day.

Toby Gilbert
CEO

Founder and Investor in technology and Telco, Toby is an experienced and successful entrepreneur. After attending UCL, he founded, built and sold a portfolio of successful businesses including Bellingham Telecommunications and HTR Telecoms Hong Kong. Toby has extensive experience operating in Asia and Africa.

Knut Arne Vinger
Chief Scientist

Tech pioneer and early adopter of digital currencies, and payment systems, Knut attended University of Oslo. His thesis on evolutionary hardware was published by NASA and the US Department of Defense. Knut developed one of the first online mobile slot machines and has worked extensively in telecommunications as CTO of Nouvel Air Group.

Mike Conte
CTO

Experienced and accomplished tech executive, Mike has led the Excel, Office, Entertainment, Shopping groups for Microsoft. He's the founder of several startups and former CEO of Musiwave, the largest distributor of Digital Entertainment in Europe, which was acquired by Microsoft in 2008. Mike loves building new products and businesses.

Alejandro Duran-Pallares
Lead Developer

Alejandro has worked on multiple crypto projects, created the haskell bitcoin library and the TxOcean bitcoin mixer, and also has experience with AI and design of distributed systems. He is an experienced Haskell developer.

Paul Davis
Strategy Director

After studying Computer Science at MIT, Paul became the first Windows Evangelist for Microsoft, reporting to Steve Ballmer. He later led the Mac Word team and has served as a leader and CTO for many startups in social media, entertainment and commerce.

Alexander Kjeldaas
Architecture Director

Experienced technical architect focused on AI, blockchain and security, Alexander has led teams at Google and FAST and founded astor.ai, a security system using blockchain. He developed the crypto library for the linux kernel and also contributed to the Bitcoin core and Haskell.

"GREAT TEAMS LOVE HARD PROBLEMS"

Active Advisory Board

Coinweb's team includes an active group of external experts.

Paul Mockapetris
dsDNS Director

Inventor of the DNS, Paul is one of the fathers of the Internet. Paul has held key positions at Salesforce.com, Oracle, and DARPA, and has served as chair of the IETF and been inducted into the Internet Hall of Fame. He remains a sought after conference speaker.

John Hunter Maxwell
Business Advisor

An accomplished leader, John has held executive and board positions at Xerox, Prudential Banking, Royal Sun Alliance, HomeServe Diageo, Institute of Advanced Motorists, the Board at London Finance and The Royal Automobile Club Motor Sports Association.

Christopher Darnell
Financial Advisor

Chris served as a member of senior leadership teams for strategic technology businesses including Microsoft Office 365, and Xbox. Chris worked in businesses across industries including technology, energy, media and entertainment.

Tom Yoritaka
Advisor

A venture capital investor, a director/interim executive/advisor for startups, an entrepreneur and Web software product management executive at Cisco and Yahoo!, Tom has expertise in IoT, AI, big data, cloud, cybersecurity & blockchain, enterprise collaboration software & online commerce.

Chris Blackhurst
Press Advisor

Formerly the Editor of *The Independent* and City Editor of the *London Evening Standard*, Chris is an award-winning journalist, national newspaper columnist, and TV presenter who has contributed at the *Independent*, *Daily Express*, *Sunday Times* and *London Live TV*. Chris is a Cambridge University Law graduate.

Ting Peng
Marketing and Community Manager

Ting is a growth hacker, marketer, business development professional and strategist. She's helped blockchain and tech startups in the EU and China achieve rapid growth and success. She's taught at a private university in South Korea and hosted a radio show in English.

Glossary

B2B. Business to Business. Like IBM.

B2B2C. Business to Business to Consumer.

B2C. Business to Consumer. Like Starbucks.

Bitcoin Pizza Day. On 17 May 2010 a pizza was sold for 10,000 Bitcoins. With Bitcoin at around $7,000, the price of that pizza is $70 million today.

Block. The building block of the Blockchain. Bitcoin's original block size was one megabyte.

Blockchain. A linked series of blocks, where each new block includes the fingerprint of all the previous blocks, rendering it immutable without detection.

Coin. A unit of cryptocurrency, either granted to miners who create blocks, or created at the launch of a blockchain.

Consensus. In a blockchain, the individual nodes, or Miners check the work of others. When the majority of Miners agree on the validity of the next block, it is accepted. This process is called Security by Consensus.

dApps/dsApps/Distributed Apps. Applications which use servers, like nodes, distributed across the Internet.

DEX/Distributed Exchange. DEX's allow cryptocurrency trading on a distributed ledger, without ever taking control of either user's currency, eliminating a security risk.

DNS. The Domain Name System, which converts the readable names we know, like www.facebook.com, into the IP addresses which are used by the underlying protocol.

Dust. Small cryptocurrency values, sort of like pennies, left over after transactions, whose value is too low for it to make economic sense to spend them.

ECDSA sep256k1 DER-encoded. Methods of cryptographically encoding a value.

Encryption. The process of encoding information in such a way that only authorized parties can access it

ERC20. The technical standard used for Ethereum smart contracts for implementing tokens.

FFV. Short for Family, then Function, then Verification. A model to reach consensus combining long-term and medium-term consensus with RDoC verification.

Fork. To make a copy of the source code from one software project and use it to start another, occasionally resulting in a separate and distinct project and supporting community.

Gas. The execution fee for blockchain operations such as executing smart contracts.

Hash. A function that can map data of arbitrary size to data of a fixed size.

dsLayer. Coinweb's master network which connects to multiple blockchains.

Hyperserver. A server which connects the dsLayer to a particular blockchain to offer compatibility or additional functionality.

ICO/IEO. An Initial Coin Offering/Initial Exchange Offering, where crypto tokens are sold in order to fund a project or company.

Metalayer. A platform which connects with and writes its information to an underlying blockchain.

MIDS. Merkelized Implicit Data Structure, a highly secure and efficient data storage method.

Miner. A node participating in a blockchain, miners compete to record the next block, and earn fees and coins.

Multisig Encoding. A digital signature scheme which allows a group of users to sign a single document.

Node. A participant in a distributed app, for example, a miner.

OP_RETURN Encoding. A Bitcoin script opcode used to mark a transaction output as invalid, and can be used to store arbitrary information on the blockchain.

Oracle. In the context of blockchains, an agent that finds and verifies real world occurrences and submits this information to a blockchain for use in smart contracts.

Proof of Burn/Proof of Stake. An economic measure to deter attacks on a network by requiring investment of stake from the requester.

Proof of Work. An economic measure to deter denial of service attacks on a network by requiring investment of work from the requester. In a Blockchain this is processing time by a miner.

Protocol. In the context of crypto, the underlying communication architecture and features of a distributed app or blockchain.

RDoC. Short for Refereed Delegation of Computation. Protocols that allow client to delegate the computations to *several* servers, where the client is guaranteed to determine the correct answer as long as even a *single* server is honest

ScriptSig Encoding. An encoding method used in blockchain transactions.

Seed Funding. Funds used to start up a new company.

SegWit. Short for Segregated Witness (Consensus layer), SegWit is a backward compatible evolution of the Bitcoin protocol which allows, for example, larger block sizes.

Sha256. A hashing algorithm.

Time-locked Tokens. Tokens which cannot be spent immediately, and are instead unlocked over time as controlled by a smart contract. This is similar to stock vesting.

Token. A digital asset, for example, a unit of cryptocurrency, or **Coin**.

Turing Complete. Named after English mathematician and computer scientist Alan Turing, a programming language is said to be Turing complete if it can be used to simulate any Turing machine. Used to describe fully capable programming languages.

Utility/Security/Stable Tokens. Utility tokens can be used to purchase goods or services, much like a prepaid card. Security tokens hold value in an asset, much like an equity. Stable token have their value pegged to another currency.

Disclaimers

This white paper is for information purposes only and may be subject to change. Coinweb cannot guarantee the accuracy of the statements made or conclusions reached in this white paper. Coinweb does not make and expressly disclaims all representations and warranties (whether expressed or implied by statute or otherwise) whatsoever, including but not limited to:

- *any representations or warranties relating to merchantability, fitness for a particular purpose, description, suitability or non- infringement;*
- *that the contents of this document are accurate and free from any errors; and*
- *that such contents do not infringe any third party rights. Coinweb shall have no liability for damages of any kind arising out of the use, reference to or reliance on the contents of this white paper.*

This white paper may contain references to third-party data and industry publications. As far as Coinweb is aware, the information reproduced in this white paper is accurate and that its estimates and assumptions are reasonable. However, there are no assurances as to the accuracy or completeness of this information. Although information and data reproduced in this white paper are believed to have been obtained from reliable sources, we have not independently verified any of the information or data from third party sources referred to in this white paper or ascertained the underlying assumptions relied upon by such sources.

As of the date of publication of this white paper, XCO Tokens have no known potential uses outside of the Coinweb platform ecosystem and are not permitted to be sold or otherwise traded on third-party exchanges. This white paper does not constitute advice nor a recommendation by Coinweb, its founders, officers, directors, managers, employees, agents, advisors or consultants, or any other person to any recipient of this paper on the merits of participation in the Coinweb Token Sale. Participation in the Coinweb Token Sale carries substantial risk that could lead to a loss of all or a substantial portion of funds contributed.

No promises of future performance or value are or will be made with respect to XCO Tokens, including no promise of inherent value, no promise of continuing payments, and no guarantee that XCO Tokens will hold any particular value. Unless prospective participants fully understand and accept the nature of Coinweb's proposed business and the potential risks inherent in XCO Tokens, they should not participate in the Coinweb Token Sale. XCO Tokens are not being structured or sold as securities. XCO Tokens are not a participation in Coinweb and XCO Tokens hold no rights in Coinweb. XCO Tokens are sold with an intended future functionality on the platform to be developed by Coinweb and all proceeds received during the Token Sale may be spent freely by Coinweb on the development of its business and platform.

This white paper does not constitute a prospectus or offering document and is not an offer to sell, nor the solicitation of an offer to buy any investment or financial instrument in any jurisdiction. XCO Tokens should not be acquired for speculative or investment purposes with the expectation of making an investment return.

Legal Disclaimer

No regulatory authority has examined or approved any of the information set out in this white paper. No such action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this white paper does not imply that applicable laws, regulatory requirements or rules have been complied with.

XCO Tokens could be impacted by regulatory action, including potential restrictions on the ownership, use, or possession of such tokens. Regulators or other competent authorities may demand that Coinweb revises the mechanics and functionality of XCO Tokens to comply with regulatory requirements or other governmental or business obligations. Nevertheless, Coinweb believe they have taken commercially reasonable steps to ensure that its planned mechanics are proper and in compliance with currently considered regulations. Coinweb is in the process of undertaking further legal and regulatory analysis of the intended functionality and mechanics of XCO Tokens. Following the conclusion of this analysis, we may be required to amend the intended functionality of XCO Tokens in order to ensure compliance with any legal or regulatory obligations that apply to us. We shall update this white paper and publish a notice on our website in the event that any changes are made to the XCO Token functionality.

Forward-Looking Statements

This white paper contains forward-looking statements or information (collectively "forward-looking statements") that relate to Coinweb's current expectations and views of future events. In some cases, these forward-looking statements can be identified by words or phrases such as "may", "will", "expect", "anticipate", "aim", "estimate", "intend", "plan", "seek", "believe", "potential", "continue", "is/are likely to" or the negative of these terms, or other similar expressions intended to identify forward-looking statements. Coinweb has based these forward-looking statements on its current expectations and projections about future events and financial trends that it believes may affect its financial condition, results of operations, business strategy, financial needs, or the results of the token sale or the value or price stability of XCO Tokens.

In addition to statements relating to the matters set out here, this white paper contains forward-looking statements related to Coinweb's proposed operating model. The model speaks to its objectives only, and is not a forecast, projection or prediction of future results of operations.

Forward-looking statements are based on certain assumptions and analysis made by Coinweb in light of its experience and perception of historical trends, current conditions and expected future developments and other factors it believes are appropriate, and are subject to risks and uncertainties. Although the forward-looking statements contained in this white paper are based upon what Coinweb believes are reasonable assumptions, these risks, uncertainties, assumptions, and other factors could cause Coinweb's actual results, performance, achievements, and experience to differ materially from its expectations which are expressed, implied, or perceived in forward-looking statements. Given such risks, prospective participants in this token sale should not place undue reliance on these forward-looking statements.

coinweb.io