



# TECHNICAL WHITEPAPER

*THE EQUITY MARKET IS BROKEN  
LET'S START AGAIN*

24/07/2018  
v3.0

# Table of Contents

1	Introduction .....	4
2	Component Overview .....	5
2.1	Public Application Services	6
2.2	APIs & External 3rd Party Services	7
2.3	Artificial Intelligence Module	8
2.4	Blockchain Abstraction Layer	10
2.5	Public Ethereum Blockchain	11
2.6	Oracle Services	11
2.7	Storage Systems	11
2.8	Queuing Mechanism	13
3	The Token Environment.....	14
3.1	The CHX Token	14
3.2	Offer Tokens	14
4	Our Technology Stack .....	16
5	Summary.....	17

## Table of Figures

Figure 1: Overall System Components.....	5
Figure 2: Our AI Approach.....	8
Figure 3: Training & Test Modules for Machine Learning.....	9
Figure 4: Our Abstraction Layer .....	10
Figure 5: Our Oracles .....	11
Figure 6: Entity Relationships Explained.....	12
Figure 7: Token Relationship Diagram.....	14
Figure 8: Our current technology stack.....	16

# 1 Introduction

At Own, we have a simple mission - to disrupt the global equity market.

We stand for any business that wants to raise capital to grow. We stand for any investor who wants to buy shares in businesses they believe in. That's why we built Own. To connect businesses directly to investors, using blockchain. To make buying and selling shares easy.

This document explains the architecture and technology layers that allow us to create the Own ecosystem.

- **Our core data and trust model** is built around the Public Ethereum Blockchain and the solidity programming language for smart contracts. This allows us to create a direct, tamper-proof and immutable record between a business owner and investor. It enables a true trust model, between business owner and investor, which was not achievable with previous technology and relational database models.
- **Our own Blockchain Abstraction Layer** reduces complexity of the trading and smart contract logic. This means our products are independent from the underlying frameworks, more flexible and future-proof. It also simplifies development on our platform, as developers are able to call specific equity transaction routines without having to worry about the underlying smart contract logic.
- **Off-chain Storage Systems** - in addition to the benefits of blockchain we are also aware of its architectural limitations. So we are only using the blockchain to store transactional data. All other data is stored in off-chain storage systems allowing us to leverage the benefits of traditional database architecture such as transaction speed and privacy.
- **Our Oracle Services and APIs** link the on-chain and off-chain solutions and interface with the wider world. We are creating an open and flexible technology ecosystem where others can contribute, add and interact. The community can access our core libraries and core system components, and build additional value-add services.
- **Our unique Artificial Intelligence Module** actively supports business owner and investor decisions through the use of prediction modelling, sentiment analysis and machine learning. This allows both parties to make decisions with the best possible information.
- **Our development team** has significant experience in delivering large scale global financial services platforms, and has a proven track record managing some of the world's largest share registers, trading gateways and financial applications.

## 2 Component Overview

Own's core technology components include:

1. **Public Application Services:**
  - o The fully featured business owner user interface.
  - o A mobile only iOS and Android app for investors.
2. **Private APIs and 3<sup>rd</sup> Party Services APIs** allow external providers and developers to access Own to build their own value-add apps and services. These include proxy voting, AGM services, reward and incentive scheme apps.
3. **Artificial Intelligence (AI) Module** forms the machine learning interfaces, driving our reporting, analytics and prediction models.
4. A **Blockchain Abstraction Layer** allows mapping services between any underlying blockchain framework and our own modules.
5. The underlying **Public Ethereum Blockchain** and ERC 20 smart contracts.
6. **Oracle Services** allow querying of internal and external sources of real-time data to feed into smart contracts on the blockchain.
7. **Storage Systems** are used to store all off-chain data and keep blockchain transaction data to a minimum.
8. A **Queuing Mechanism** to manage and batch transactions in a smarter way.

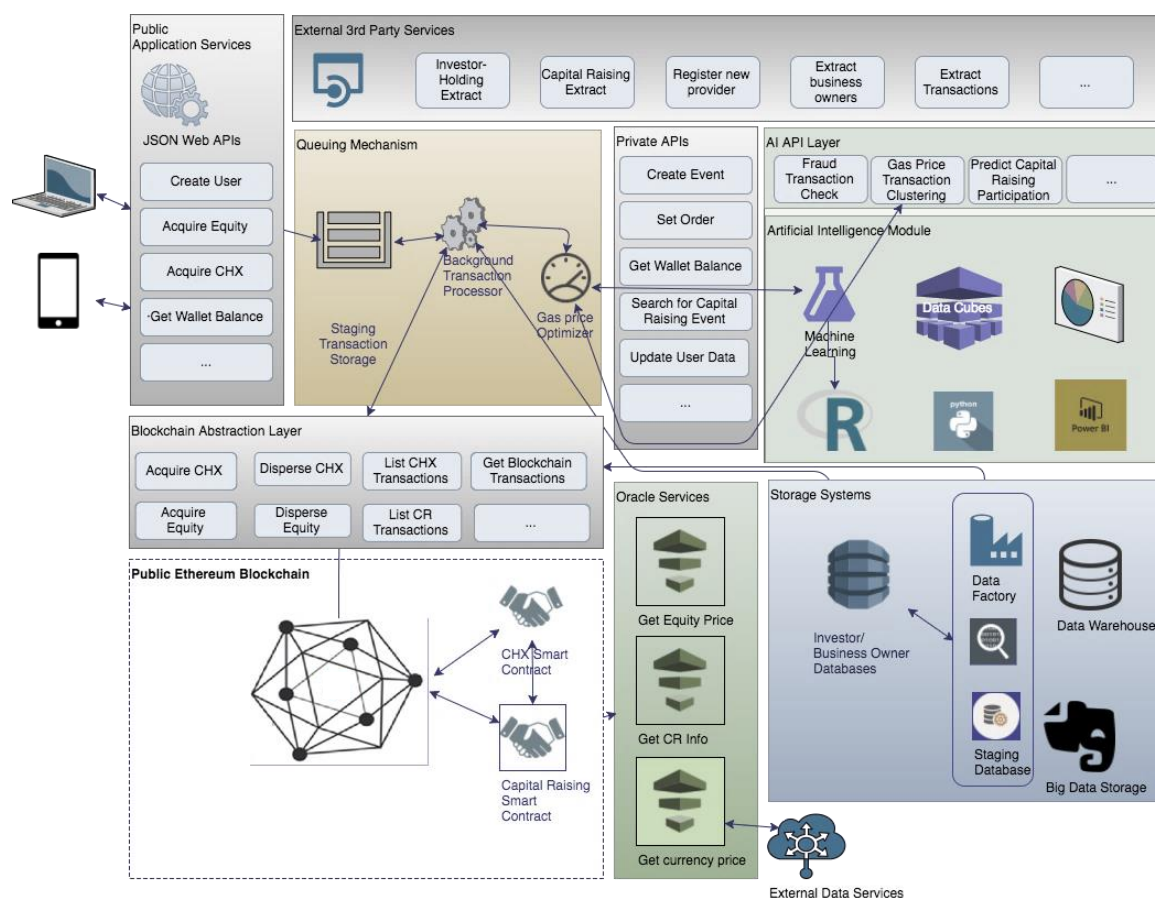


Figure 1: Overall System Components

## 2.1 Public Application Services

Our front-end provides the interface for users to interact with our system. We have two applications:

1. **Web app** - The fully featured, responsive web app that allows business owners to administer equity offers and manage their investor base
2. **Mobile app** - The simple mobile app that allows investors to purchase equity and trade in secondary markets

### 2.1.1 Web app

The web app is implemented as a responsive Single Page Application (SPA) with Angular. The SPA development approach enables us to treat both the web app and mobile app in the same way, from an API perspective. Angular 2+ is written in TypeScript, which is the most popular JavaScript alternative. Being a superset of JavaScript, it is easy to learn, but the tooling and resources provided by the community ensures a much better experience than working with plain JavaScript.

### 2.1.2 Mobile app

The mobile app supports Android and iOS. To avoid having two completely different codebases for the same app (Objective-C/Swift for iOS; Java for Android), we have developed our mobile apps using Xamarin. Xamarin enables us to build a native app, while sharing significant parts of the code between two platforms.

### 2.1.3 Behind the scenes...

Our backend system is composed of multiple small RESTful services, with narrow responsibilities. Focusing services on a small set of specific tasks enables faster development and release cycles, as well as minimising coupling.

We use .NET, with C# as our primary language, and F# for more complex business logic and data processing activities. Three main drivers motivated our decision:

- We have extensive previous experience building systems with .NET, especially in the financial service space, and we have access to a team of great developers.
- High quality tooling is available for .NET, making development easier and quicker and giving us a better chance to build scalable solutions for financial transactions.
- The ability to reuse code and skills between our mobile apps (Xamarin) and back-end services.

Our services are stateless whenever possible, to enable easy scale-out scenarios. We also host our services in a cloud environment, which further enables us to dynamically scale depending on load.

We generally automate the release process by applying automation and integration tools such as Jenkins, Ansible and Docker.

## 2.2 APIs & External 3rd Party Services

Our decentralised application environment allows the creation of multiple third-party solutions within the Own ecosystem. Applications might include, voting, communication and know-your-customer checks but are not limited.

Whilst we protect and manage the core platform and libraries, we allow third party companies and developers to utilise a number of APIs to interface with the platform. We have a set of functions and procedures that allow the creation of applications which can access features and libraries across multiple components.

To support third parties, we manage a number of dedicated industry grade interfaces in order to provide a best in class service:

- SWIFT gateways
- Trading interfaces
- Banking partners
- Stock Exchange and market data feeds

Our APIs are implemented as a set of stateless RESTful web services, using standard HTTP methods. The APIs are split according to the system module they are communicating with:

- **Web APIs** to communicate between our system and web, android and iOS apps
- **Third Party API Services** used by third party providers, building their solutions on our ecosystem
- **External System Feeds** for data from trusted third parties
- **Artificial intelligence APIs** communicating with the prediction models deployed as a web service
- **Blockchain APIs** communicating with the blockchain smart contracts
- **Database APIs** retrieving and storing data in our storage systems

Our primary data format is JSON, but we plan to support other data formats (e.g. XML) in future. The APIs are secured using the JSON Web Token (JWT) standard, which fits well with the stateless services architecture and enables high request throughput and better scale out.

API requests are handled asynchronously, which improves the responsiveness of the whole system.

All long running jobs are queued and processed by our backend services, preventing stalled API requests and enabling granular control over the scheduling and execution of jobs.

## 2.3 Artificial Intelligence Module

Data on the Own ecosystem represents one of our biggest assets. When analysed securely and anonymously, this data can help our customers make better decisions. *Please see the Security Whitepaper for information about how we secure customer data.*

Our Artificial Intelligence (AI) Module is based on historical data that is stored on our storage systems and transactional data held on blockchain. The data is moved from the various data sources into our data warehouse and Hadoop storage. Prediction models are then built using R-scripts and Python analytical libraries. The trained model is deployed as a web service and can be called by our system components.

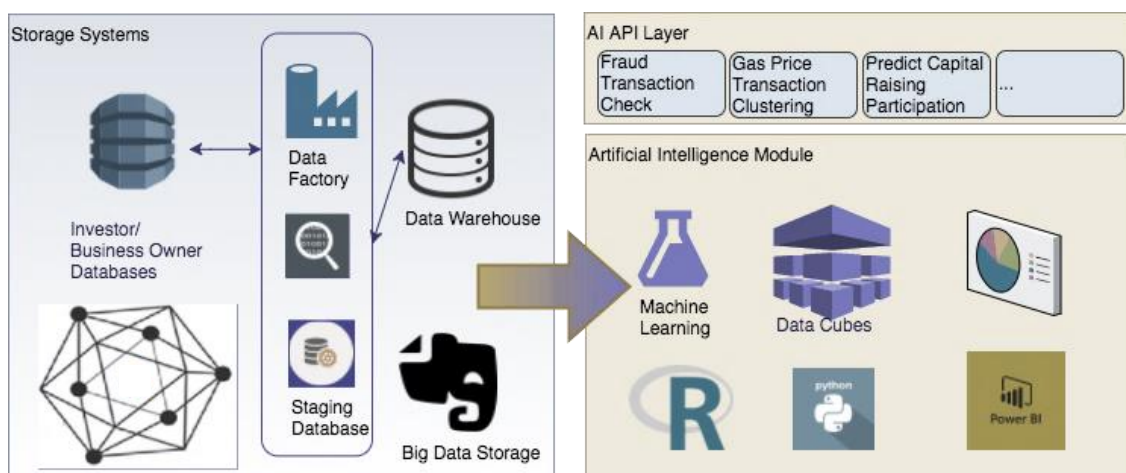


Figure 2: Our AI Approach

We have built a set of prediction models, using both supervised and unsupervised learning, that use our data to help customers make better informed decisions.

Data visualization is an important aspect of our approach. We have significant previous experience in delivering data visualization dashboards with Power BI, and we are using Power BI as our main data visualization tool.

### 2.3.1 Supervised Learning

We use Supervised Learning to create prediction models based on historical behaviours and event outcomes. Supervised learning allows us to create models that help to:

- Predict investor participation in a share offer
- Predict investment size in a share offer
- Predict price development after the completion of a share offer



A typical high-level flow diagram for supervised learning is shown below. In this process, the data is fetched from our internal and external service channels and from blockchain. The data is then enriched, merged and transferred to fit the machine-learning algorithm. These steps are common to all machine learning models. However, in a supervised learning process, the data is split into two buckets. A bigger date bucket called training data, which is being used to train our models, and a smaller data set called testing data, which is used to test the accuracy of the model produced by the training data set. Once we are satisfied with the model accuracy, the model is deployed as a service and can be used on the platform.

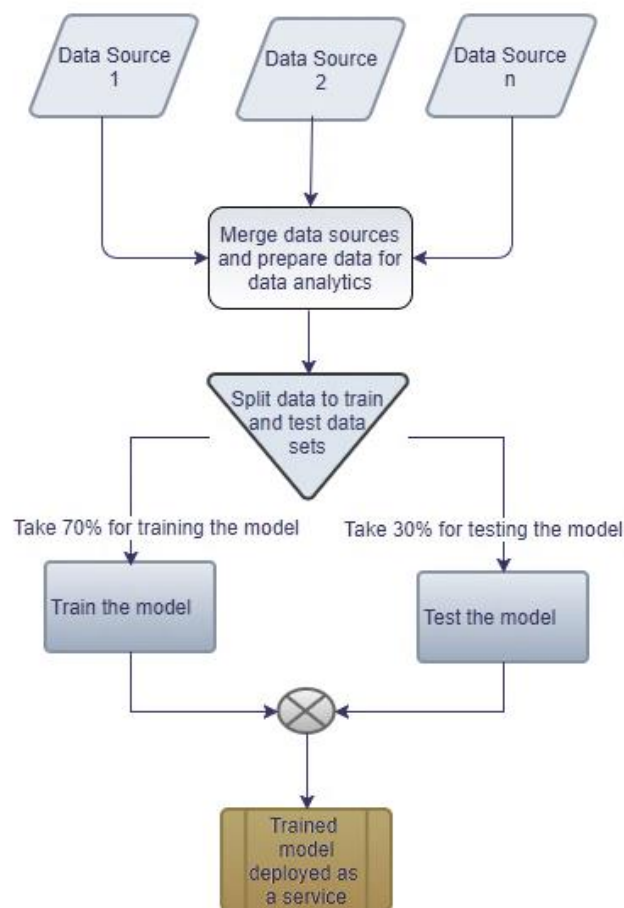


Figure 3: Training & Test Modules for Machine Learning

### 2.3.2 Unsupervised learning

Unsupervised learning is used to recognise patterns and classify customers and events and create clusters of similar objects. The goal of this type of data modelling is to organize data according to behavioural patterns or to describe an object structure based on object properties. We use unsupervised learning to:

- Recognise potentially fraudulent transactions
- Cluster high value investors
- Identify potential customers (for a retail business) among investors

With every new transaction, our AI models are "re-trained". Through this process, our AI module keeps learning and delivers more accurate predictions. By utilising the power of our community, we hope to develop more features and functions in future.

## 2.4 Blockchain Abstraction Layer

Our abstraction layer exposes the logic of the underlying blockchain and standardises the way our system works. It abstracts the system communication with blockchain, making development faster and more modular. The abstraction layer is a set of predefined functions that execute one specific business process or rule on the blockchain. The function has a predefined name, several parameters and the output wrapper that always stay the same, independent of the blockchain system being used.

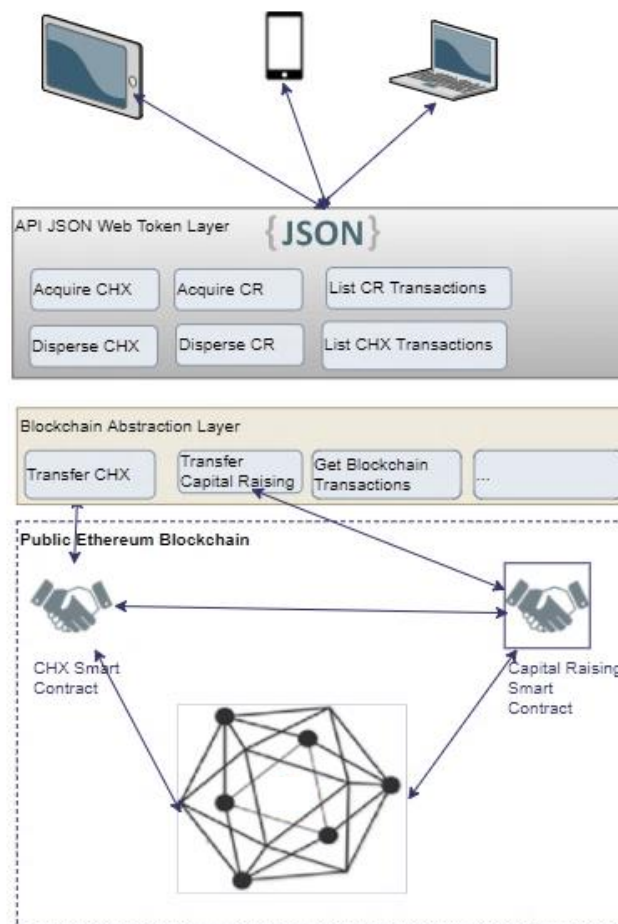


Figure 4: Our Abstraction Layer

Our Blockchain Abstraction Layer allows us to future-proof Own and minimise the risks associated with being completely dependent upon a single public blockchain.

## 2.5 Public Ethereum Blockchain

As our smart contracts are implemented on the Ethereum blockchain, we are using Solidity as the programming language for smart contracts. Development on a blockchain introduces many complexities in the development workflow. However, these complexities can be abstracted away allowing for a simplified software development process, by employing a blockchain development framework. One of the most popular blockchain development frameworks for Ethereum is the Truffle Framework, which we are using as our main blockchain development tool.

## 2.6 Oracle Services

We use Oracle services for querying sources of real-time data to feed into smart contracts on the blockchain. Our blockchain solution supports smart contracts which run or execute depending on a particular condition from an external source. An example of such a function is a "vesting or options contract", which puts shares in either a "vested" or "granted" condition based upon a specified price or date. Oracle services monitor and query the smart contract and inform the smart contract if a condition is met.

Oracle examples include:

- Querying fiat currency prices from external data feeds
- Accessing an equity price from an internal data feeds
- Accessing the setup information for an equity offer

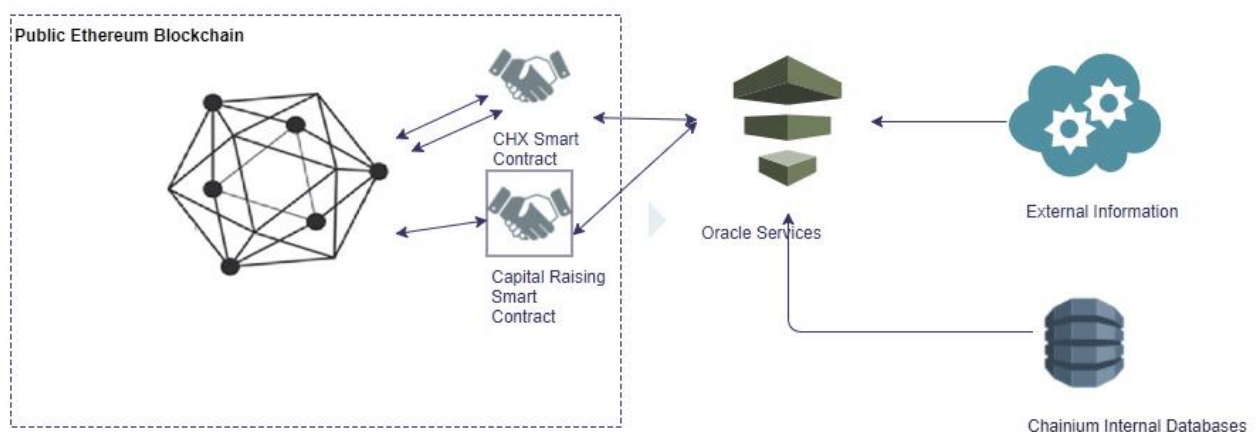


Figure 5: Our Oracles

## 2.7 Storage Systems

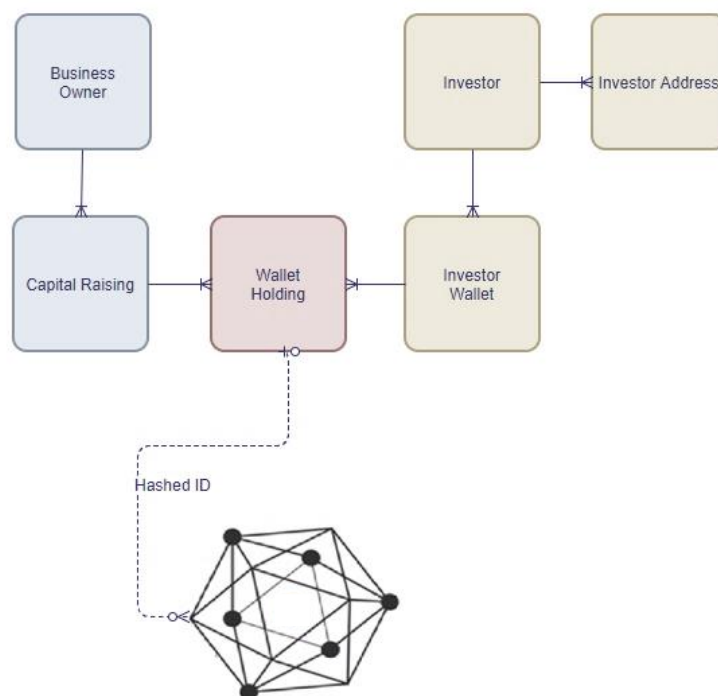
In order to keep the usage of blockchain technology as efficient and low cost as possible, we store only transactional data on blockchain. This increases the

transaction processing throughput significantly. Any other data required, is stored off-chain. Data stores include common relational databases, data streams, NOSQL archives, and cloud storage solutions.

We use PostgreSQL as our primary relational data storage system based on the team's past experience. The traditional off-chain relational database holds the entity definitions and relationships. It stores information such as:

- Business Owner personal and company details
- Equity offer inputs and requirements
- Investor details and personal information
- Balances and portfolio details

The entity relationship is described below:



**Figure 6: Entity Relationships Explained**

The benefit of a hybrid on- and off-chain model is that for non-transactional data we can leverage the benefits of the off-chain traditional solutions:

- Faster processing than blockchain and ability to process data heavy structures
- Querying and interrogation of mass data is easier and simpler
- Reporting can be faster and simpler
- Scalability is increased, given the size of the data stores

Our database systems link to the blockchain transactions using hashed ID, stored on both relational databases (off-chain) and blockchain (on-chain). NoSQL database stores are used for archiving documents and for storing data stream information.

The underlying infrastructure for our off-chain services is Infrastructure as a Service (IaaS), in both a public and hybrid cloud model.

## 2.8 Queuing Mechanism

We have built a new transaction queuing and batch process mechanism to manage the potentially billions of individual transactions. This approach optimises gas price and other transactional fees associated with public blockchain frameworks.

### 2.8.1 Transaction Price

All transactions on Own use the CHX token. The transaction price is usually determined with the following:

$$\text{Transaction Price} = [\text{Acquisition Price}] + [\text{Blockchain Gas Price}] + [\text{Marketplace Fee}]$$

Three parts make up the transaction price:

- **Acquisition Price** – The cost to acquire the equity as set by the business owner
- **Acquisition Price** = (Offer Share Price \* Number of Shares Bought)
- **Gas Price** – The Ethereum transaction fee
- **Marketplace Fee** – For standard investors, we charge no fee - in other circumstances, certain fees may apply

### 2.8.2 Optimizing Gas Price

In order to optimize Ethereum gas costs when processing transactions, we extend the smart contract implementations with batch processing functions. The role of these functions is to enable cost-effective processing of multiple balance updates at once, instead of submitting separate transactions for each balance change.

### 3 The Token Environment

#### 3.1 The CHX Token

Buying shares on Own is undertaken on a peer-to-peer basis using either fiat currency, ETH or BTC. Own uses the CHX utility token to build trust in equity offers, as well as to pay for the fees of the ecosystem. CHX tokens can be acquired directly through the business owner or investor apps, through a secondary exchange or during our token sale. The CHX tokens are created through an ERC20 smart contract on Ethereum.

#### 3.2 Offer Tokens

Each business equity offer has its own token, created through a dedicated Offer Smart Contract. The Offer Smart Contract is bound to the CHX contract, in the sense that any Offer Contract requires a minimum amount of CHX tokens to be held in reserve for the life of the shares. These reserve CHX tokens remain the property of the business owner

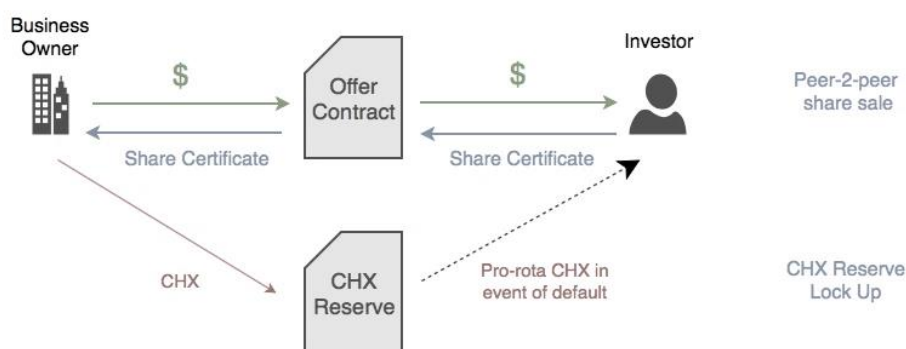


Figure 7: Token Relationship Diagram

##### 3.2.1 Business Owner View: Creating an Offer & Selling Shares

Once a business owner has decided to raise capital by selling shares, they can advertise their Offer on Own by following the simple walkthrough process on the web application. Behind the scenes, each Offer requires a new instance of an Offer Smart Contract, which maintains the balances of token holders for that offering. Offer contracts also contain a reference (Ethereum address) for the business owner, as well as the token offer price and currency and the amount of CHX reserve held.

For each equity offer, completion parameters such as duration of sale and maximum number of shares for sale, are set by the Business Owner upfront. In addition the minimum number of CHX tokens must be purchased by the business owner and used as a reserve for the Offer Contract. CHX tokens can be purchased through the app, on a third-party exchange or during our token sale.

As the Offer progresses, the business owner can see the funds transferred to them by investors for every share sold. Once the completion parameters are met, then the business owner can access the funds. The CHX tokens that the business owner provided as the reserve remain locked up in the Offer Contract for the life of the equity instrument.

### **3.2.2 Investor View: Buying Shares**

Investors use the investor app to identify offers they want to participate in. Once an investor has decided that they want to participate in an offer by purchasing shares, they simply click through the purchase equity option on the app using fiat or crypto from their linked account.

## 4 Our Technology Stack

We carefully select technologies that best meet our requirements, in particular scalability and regulatory compliance. As we are technology agnostic, we will always choose the best product for the job.

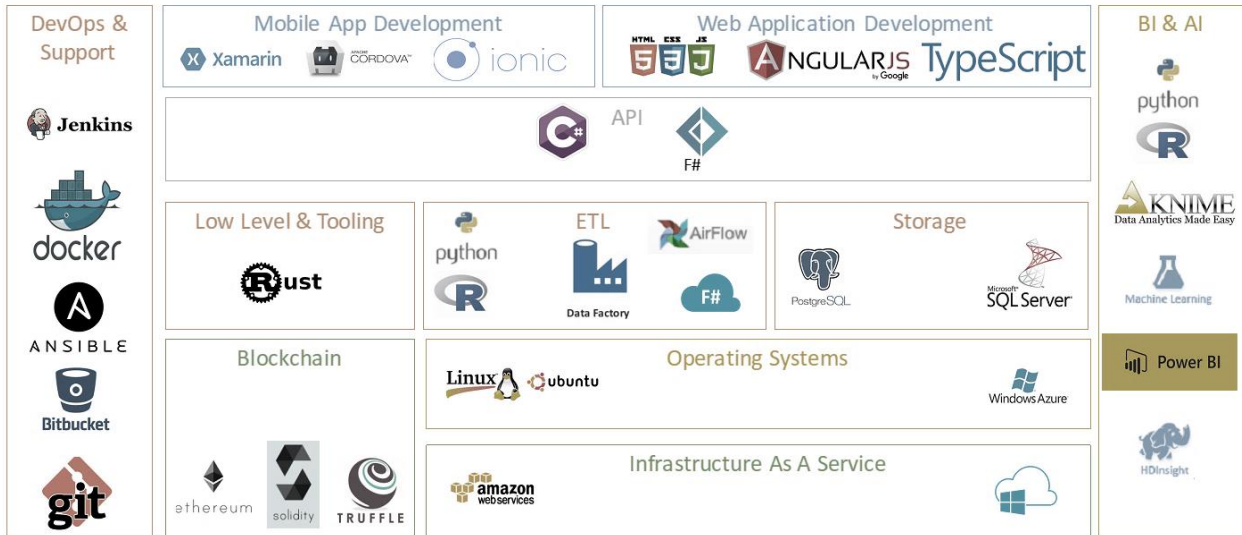


Figure 8: Our current technology stack



## 5 Summary

Own has an experienced technical and development team that have worked together on multiple large financial services platforms over many years.

For our customers, we have created two dedicated apps that make buying and selling shares easier and simpler than ever before. We are using our CHX utility token as the integral mechanism to build trust and to pay for services.

Behind the scenes, our architectural approach is to utilize both the advantages of the public blockchain and traditional off-chain storage solutions. We achieve this through the use of a blockchain abstraction layer, oracles and APIs.

Our upcoming token sale will allow us to bootstrap the community, create a token economy and raise the necessary funds to complete development of the full production Own ecosystem.