Valence Blockchain Applications v1.0

Craig MacGregor, Alex Vazquez

Abstract

Blockchain is proving it has the potential to disrupt many digital and real-world industries. Using blockchain to record transactional data has wide-reaching implications, providing businesses and customers with never before seen transparency and trust. Along with blockchain's long list of benefits comes complexity. The learning curve for developers is often very steep and mistakes can have devastating implications for applications and their users. If blockchain is to achieve mainstream acceptance developers cant be expected to have a PhD in Economics and Math just so they can build a basic application. Developers should be to be able to focus on what they do best; building great software while letting Valence handle the complexities.

1. Introduction

Valence is proposed by Craig MacGregor and Alex Vazquez, contributors to NavCoin Core and the NavTech private payment platform. NavCoin is a cryptocurrency based on Bitcoin. NavTech is a blockchain powered application which enables users to send private NavCoin payments by encrypting transaction data to a secondary blockchain known as the NavTech Subchain. Currently, this second blockchain exists for the sole purpose of storing NavTech's encrypted data.

Valence results from the need to expand the capability of this second blockchain to allow multiple applications to store a variety of data as more blockchain powered applications are built for NavCoin. Valence is designed to replace the NavTech Subchain, but in fact, it will do more than replace it - it will supersede technologically. This paper describes the Valence blockchain application platform with its enhanced privacy features, improved scalability, and simplified app deployment. All of which will be open-source and publicly available allowing other applications to be built on top of Valence and enjoy the same benefits.

1.1. The NavTech Subchain

When NavTech processes a private payment it manages the fulfillment of transactions by writing encrypted data to a secondary blockchain - the Subchain. In this way, the Subchain operates as a distributed, immutable database for authorising NavTech private payments, which can be read and written to securely by multiple parties without error.

The Subchain is only designed to store encrypted data relating to NavTech transactions. It has no capacity to handle multiple applications or complex application data.

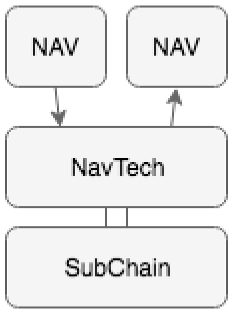

Figure 1: Existing monopurpose Subchain

1.2. Valence Platform

Valence is designed to extend the capabilities of the NavTech Subchain. It has the ability to store data relating to multiple applications, handle complex application data, perform data validation and many other systemic benefits, which will be explored in this paper.

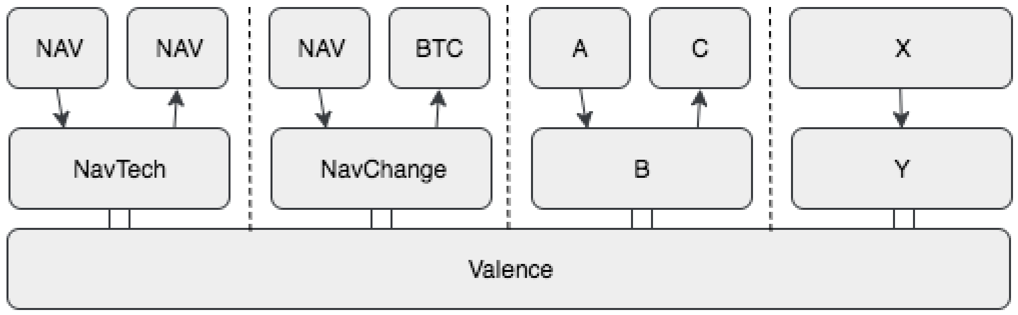

Figure 2: Proposed Valence multipurpose blockchain

2. Problem Definition

There are many problems in the current paradigm of decentralised blockchain applications, ranging from performance issues to financial fraud.

2.1. Resource Intensive

Enabling custom smart contracts requires miners to not only validate the top level transaction, but also load the smart contract into memory and evaluate additional complex logic about each transaction. Enabling Turing-complete smart contracts exacerbates the problem and can quickly turn data validation into an impossibly intensive operation at scale. In most cases, data validation is replicated by all full nodes on a blockchain, and not distributed amongst them, meaning CPU cycles are the scarcest resource on the network.

2.2. Complexity Risk

With the introduction of Turing-complete smart contracts and token issuance, contract errors have caused hundreds of millions of dollars of losses to businesses and users alike. Blockchain is an emerging industry, which means developers are required to define complex and immutable smart contracts in an often unfamiliar paradigm which has many nuances. A deep level of knowledge is required by developers and businesses before they feel comfortable engaging with blockchain and overcoming the inherent risks, especially when it can impact their revenue stream. This complexity risk is one of the major barriers to wider adoption in the business community.

2.3. Bloated Blockchains

Storing complex application data on the blockchain inherently incurs a larger data footprint than simple transactional blockchains. Managing application data in a scalable way is one of the biggest problems currently facing blockchain application platforms. Issuing tokens which often have little intrinsic value or purpose compounds a blockchains storage requirement problems by adding gigabytes of arguably unnecessary token transaction data to a blockchain, which should primarily be reserved for storing application data.

2.4. Crowdfunding Accountability

Using the Initial Coin Offering (ICO) model to fund the development of blockchain applications generates major risks for investors. In general, ICOs offer no accountability, governance, rights or assurances to investors. In a lot of cases, the generation of a token is unnecessary to the overall operation of the application and how it stores data on the blockchain. At best its an internal currency for the applications ecosystem and at worst it has no utility beyond creating an opportunity for market speculation. Due to these risks and lack of regulation token ICOs have been banned in multiple jurisdictions, and more are likely to follow suit.

2.5. Merged Concerns

At its core blockchain technology provides a trustless shared ledger of transactional data. Whether its used to record shipping information, land titles or monetary value, the main purpose of a blockchain remains the same; to store unfalsifiable data and share it seamlessly across a network. While cryptocurrencies and application blockchains share a lot of the same requirements, the architecture best suited to each remains fundamentally different. Merging these use cases into a single architecture creates a compromise, which serves neither purpose faithfully.

2.6. Privacy

By nature, blockchains are public ledgers. When using a blockchain application, people often give up their right to privacy as their interactions and application data are recorded to the public ledger. Current platforms offer no methods to give users control over how their data is recorded, aggregated, shared or cross-referenced by applications or outside entities.

3. High Level Solution

Valence is a Blockchain Application Platform which allows developers to integrate the best parts of blockchain tech into their business, websites, and apps easily.

3.1. Valence Applications (VApps)

The blockchain interface Valence provides is language agnostic making it accessible to all developers. VApps can be built and deployed in a range ways to suit the requirements of each application. VApps can be configured and initialised directly from the Valence desktop application or command line - no programming knowledge is necessary.

Valence primarily operates as a programmable de-centralised database for storing VApp data, data structures, and smart contract conditions. Blockchain architecture requires that nodes replicate rather than distribute resource load. The main requirements of the Valence network will be storage and bandwidth, which are much more scalable resources than computing power. That said, even the storage and bandwidth requirements have been minimized by Valence's design. Additional creative engineering solutions can further mitigate them, whereas increasing computing power can only be effectively achieved at a hardware level.

3.2. Smart Contracts

Script definitions will be built up by a series of primitive OpCodes which will define the application's data fields, each field's requirements, and the acceptance or rejection criteria of the data record as a whole. By design, the script definitions are non-Turing complete with no loops. This enables a wide range of safe, but innovative ways for VApps to interact and apply conditional payments based on the distributed ledger. New recipes can be proposed, developed, tested and integrated into Valence's app builder, then made available to everyone who builds on the platform.

3.3. Privacy

Valence will make a range of privacy options available to both VApp Developers and Users. These features are designed to provide autonomy over data stored on the Valence blockchain, while also protecting the contents, origin, and destination of data broadcast through the p2p the network.

Valence's encrypted communication protocol provides a high level of privacy in communications between VApps and users when using the hosted VApp architecture or broadcasting network messages. Data broadcast through the network will use asymmetric encryption to protect the contents of the transmission from everyone except the intended recipient. The protocol will only expose the data necessary for the message to reach its destination. The data will be routed through the Valenode network to mask the physical location of VApps and users, with neither needing to expose their location.

3.4. Valenodes

As well as operating as entry and exit points for Valence's encrypted communication protocol, Valenodes provide a number of important services to the Valence network:

- Secure the integrity of the blockchain by participating in Proof of Stake mining.
- Seed the Valence blockchain to other nodes on the network and serve application data to VApp users in the selective-blockchain model.
- Maintain other Infrastructure VApps and Services to help further decentralize and distribute network requirements arising from storing large amounts of data on a blockchain.

4. Solution Details

The Valence App platform comprises of a number of interconnected parts that work together to form a highly scalable, usable and secure blockchain application platform.

4.1. Valenodes

Valence Masternodes provide the distributed infrastructure backbone of the Valence Application platform.

They will act as routers for encrypted application data, entry and exit points for user - app communications, blockchain seeds and a distributed network for the utility applications deployed to the network.

A Valenode will be required to be continuously online, run the Valence daemon and store the entire blockchain. A Valenode must also provide proof of ownership of a collateral transaction, making it difficult for anyone to gain a significant number of Valenodes - creating a barrier to entry for entities trying to compromise the network's privacy functions. The collateral transaction size will be determined during network testing. It should be of relatively low value to ensure decentralisation of Valenodes and wide participation in securing the blockchain and encrypted communication protocol.

Once a Valenode is configured with proof of its collateral transaction, it can be registered on the blockchain to the Valenode Registry App. This utility application will curate the blockchain record of active Valenodes autonomously and provide a blockchain record used by the consensus algorithm to allocate the Valenode payouts from mined blocks.

Field	**Value**	**Data Type**
collateral	txid	uint256
output	vout	int
proof	signed(txid)	blob
node	public_key	blob
status	true	boolean

Table 1: Valenode Registration Data

Valenodes will receive a portion of the block rewards as payment for the services they provide. The Valenode reward is a fixed amount per block and each Valenode is rewarded sequentially until all Valenodes have been rewarded and then the reward cycle starts again. New Valenodes join the back of this reward queue. As more Valenodes come online, the rewards per Valenode will become less frequent, but the total amount of rewards per week/month/year will remain the same.

The collateral transaction can remain in a different wallet and the reward payments will be to the address which owns the collateral. This enables a Valenode to operate with a 0 balance, so the collateral can be kept offline. The collateral transaction can also be used as a staking input so the Valenode can earn both Valenode rewards and staking rewards.

Valenodes will be required to randomly and periodically record a proof of service to the status log of the Valenode registry application. When a block is mined, the block merkle root is used to derive the next random set of registered Valenodes, which must report their operational status within the next 'x' blocks. This way no extra space is taken up in the block to record which Valenodes must report yet which nodes are required to report can be calculated and agreed by consensus without being known in advance.

The requirements for proving whether a Valenode is operational is yet to be defined and will be decided during network testing. It is likely to require some type of proof of work based on a set of performance parameters.

Field	**Value**	**Data Type**
proof_of_service	result	blob

Table 2: Valenode Status Log

If a Valenode has not met its operational requirements as determined by network consensus and the blockchain record, that Valenode will not be eligible for its payment and will be pushed to the back of the Valenode rewards queue. If a Valenode is consistently failing to respond it will be considered to be abandoned and required to re-register.

4.2. Cold Staking

Valenodes require a collateral transaction to help prevent malicious node saturation. Since Valence will use the proof of stake algorithm the network needs to permit Valenodes to use their collateral transaction to stake, otherwise the network risks having only a small percentage of coins used for staking, which weakens the network's security.

Cold Staking also provides increased security to the funds held by regular stakers. They can mint blocks from a hot wallet holding 0 value, while the weighted value is stored in a separate secure wallet or completely offline.

4.3. Encrypted Communication Protocol

When sending data to an application, the data will be encrypted with the application's public key. The data is broadcast from the user to a Valenode it's connected to, then propagated through the Valenode network to the application.

The backbone of the network Valence uses to communicate between applications and users is the same that already exists in the Bitcoin source code for p2p distribution of blockchain data. The main difference is the introduction of Valenodes to act as an encrypted routing layer and gateway points for data.

Each user must be connected to and registered with at least one Valenode to act as its entry point to the Valenode network, in order to interact with private applications.

Each private application must also be connected to and registered with at least one Valenode to act as its exit point from the Valenode network.

Users and applications can register with a randomly discovered Valenode in the peer network, or a specified Valenode. Registering with a Valenode requires providing a public key used to identify communications. By registering with a trusted Valenode, it can be assured the provided public key cannot be linked to a physical location.

The encrypted data broadcast over the Valenode network contains two parts; the public key of the application and the data which has been encrypted with the applications public key.

The Application Public Key will be the public portion of a public/private ECDSA key pair, looking indistinguishable from a regular Valence Address. The destination of the request being the only publicly exposed part, it will be impossible to differentiate messages originating from a regular user from those whose source is an application.

request(app_public_key, encrypted_data)

Inside the encrypted data, there will be three parts; the public key of the sender, the data, and a signature to prove the data came from the public key expecting the response and that the data has not been modified in transit.

encrypted_data(user_public_key, data, signature)

Once the application has processed the request, it will return a response through the same mechanism as the data was received, encrypting the response with the users public key.

response(user_public_key, encrypted_data)

When a Valenode receives a message, it will forward the request to other Valenodes it is connected to. Techniques to avoid the redundant relaying of messages will be incorporated into the messaging protocol.

Additional layers of privacy could be added, e.g. performing an initial negotiation phase between user and application where they would exchange unique secret tokens, which would later be used in the place of the app/user public key, making the communication untraceable and its communicators anonymous. This unique secret token would be needed to register with the Valenode where the distributed app is running, but if the Valenode is trusted there would be no risk of compromising links between public address and secret token.

The method for connecting to other Valenodes and forwarding this information will leverage the existing p2p connectivity used to share the blockchain data between nodes.

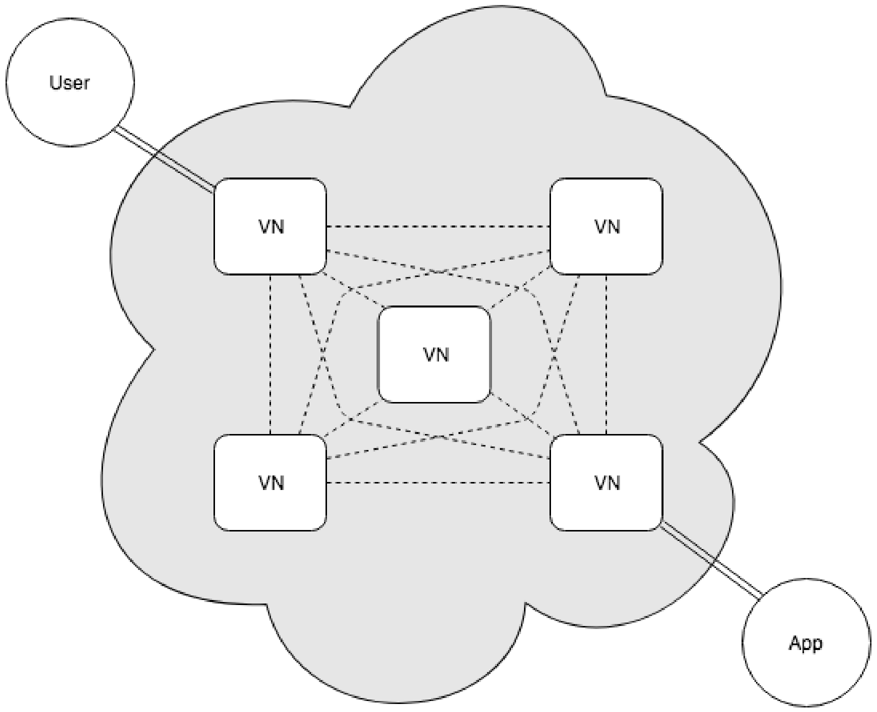

Figure 3: Encrypted Communication Protocol via Valenode Cloud

4.4. Hosted Applications and Oracles

Hosted applications and oracles can be written in any language which can read and write to an IPC socket. This allows application developers to choose the language best suited to the application they want to develop. Socket communication will be two-way between the Application and the Valence Daemon, enabling fast, reliable communication.

Making the interface language agnostic removes a lot of complexity from the daemon architecture and allows developers to integrate it seamlessly with any legacy system. Hosted apps can be interacted with via a regular web interface, or through the Valenode encrypted communication protocol.

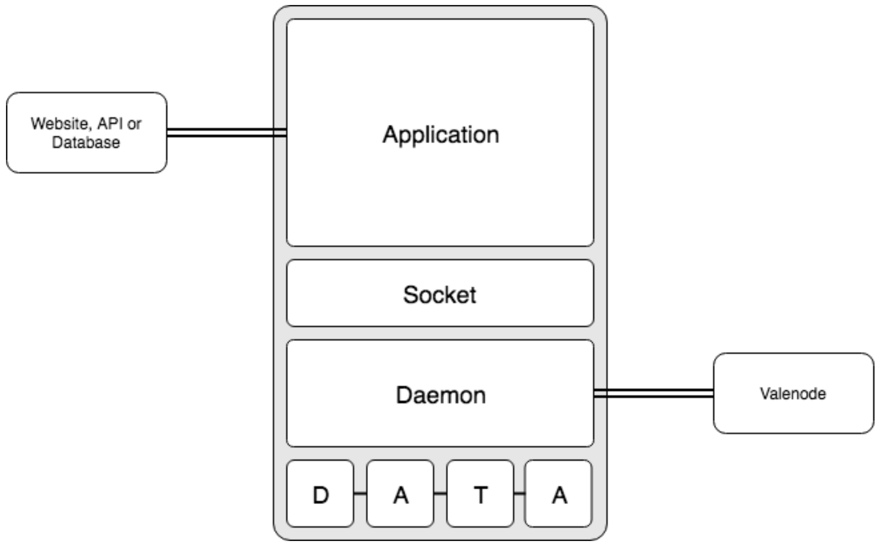

Figure 4: Hosted Application and Oracle Architecture

Communication can be transmitted in either direction or originate from the application server itself, depending on the use case of the application.

Oracles are hosted applications which are not interacted with directly. Their purpose is to find and verify real-world data and submit it to the blockchain for use in decentralised smart contracts.

4.5. Private Hosted Applications

A hosted application can take on the role of writing data to the blockchain on behalf of a user who broadcasts encrypted data to it through the Valenode network. This method disconnects the submitter from both the data on the blockchain and also the data origin on the network. The submitter will receive an encrypted receipt of the data transaction ID so they can validate that their data was written to the blockchain. Both the submitter and the application remain hidden when all communication is broadcast through the Valenode network.

4.6. Client Rendered Applications

Simple VApps, like content submissions or voting applications, which are designed to accept direct data submissions from users, can have their data interface dynamically displayed inside the Valence client. The interface would be based on the fields and conditions described in their data schema definition.

4.7. Open-source Distributed Applications

Developers can build and deploy desktop and mobile applications which act as the interface for their Valence application in common web languages. Initially, a Javascript SDK will be available with all the necessary tools to for app developers to read and write to the blockchain as well as generate desktop and mobile applications.

These application interfaces will consist of the Valence Daemon, NPM Modules required to interface with the daemon and can be wrapped into desktop or mobile applications, for example by Electron or Cordova.

The data submitted by this type of application is subject to the consensus layer as described in section 4.9 of this paper.

4.8. Thin Clients and Selective-blockchains

To remove the requirement of the end-user needing to store the entire blockchain to interact with open-source Valence Applications there will be a range of implementation options available to developers.

Thin Clients are already commonplace in blockchain software. In this model the end-user has all the software required to generate transactions, sign, and submit them to a remote server which commits them to the blockchain. This method is secure due to the signature step and gives the user autonomy over their private keys, however they are still reliant on a third party to commit the data to the network. Valenodes can easily double as thin client data endpoints, or the provider of an application using the open-source architecture described in section 4.7 could host dedicated thin client endpoints for their application users to ensure uptime.

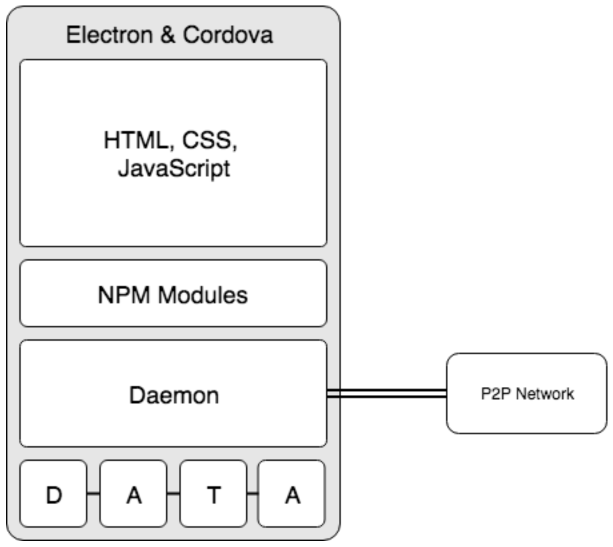

Figure 5: Open Source Application Architecture

Selective-blockchains will provide a hybrid between a thin client and a full node. They will only require users to store the block headers of the blockchain in the same way a thin client does, however they will also provide the capability to download and validate requested applications blockchain.

An advantage of this approach is that once the applications data is downloaded and validated, the selective-blockchain client has autonomy over their interaction with the application data while not requiring the client to download every applications data from the blockchain.

As with thin clients, application providers could host dedicated Valenodes to serve their application data to their selective-blockchain applications to ensure uptime.

4.9. Data Schema

The architecture of Valences database model is malleable, but is required to have defined data structures so any submitted data can be validated by peer consensus.

Each application will be able to define its own data schema and consensus rules by submitting script definitions to the blockchain. These include the available fields and their data types, acceptance criteria for the entire record such as the origin of the data, as well as conditional rules such as the state of another Valence application.

Script definitions will be built up by a series of primitive OpCodes. Functionally these scripts are simple, stack-based, and processed from left to right. They are intentionally not Turing-complete, with no loops.

The complexity of a script definition will dictate the fees necessary to pay for the data record to be validated by miners. The fees act as an incentive for miners to validate and write application data to the blockchain as well as a disincentive for anyone attempting to spam the network.

For selective-blockchains to be viable, application data needs to be isolated from the main blockchain in a way which is able to be validated as complete and accurate by the node requesting the application data from the Valenode network.

To secure isolated application data, the fees for writing application data should be transacted on the main blockchain. The fee payment should be cryptographically linked to the application data to prove its integrity. The fee redemption should be cryptographically linked to the application block containing the data to prove its integrity. These proofs combine to protect the application datas integrity with the weight of the wider Valence network.

There is no intention for applications to generate an internal token. If value transfer is required, the intention is to leverage existing cryptocurrencies as settlement layers for application outcomes.

4.10. Cross Chain Contracts

Valence data can be used in the fulfilment of cross chain conditional payments and atomic swaps. Integrations to make this possible can be implemented on top of most existing blockchain protocols without modification required to any third party protocols. Encrypt S plans to release software which utilises NavCoin as a settlement layer and serves as the primary example of cross chain conditional payments based on Valence application data.

5. Design Implications

There are some interesting implications which arise from what is described in this document.

Valences data schema encourages script definitions to be simplified into multiple smaller parts, which then interact with each other if and when necessary. This reduces the computational overheads incurred while validating application data by miners, and therefore helps achieve scalability.

The isolation of application data from the main blockchain allows for users to run nodes that can have complete autonomy over an applications data, without needing to download and validate every applications data.

The design decision to keep the script definitions Non-Turing-complete means computation cycles of the scripts execution can be accurately calculated at minimal computational cost. Non-Turing-complete scripts combined with simpler script definitions allow for extensive testing suites to be built for the most common script combinations, and safe recipes provided for the most common use cases.

Valence applications can be initialized without writing any source code. VApps could easily be configured in a simple Graphical Interface by someone with no technical knowledge of programming. This greatly reduces the barrier to entry for newcomers to blockchain - helping achieve mainstream adoption.

The implementation of a programmable database model within the application data schema allows for simple Valence apps intended for direct user interaction, like content submissions and voting applications, to be dynamically rendered directly in the Valence client with no external application required. The same dynamic rendering technique can be used in the open-source application architecture, giving developers the ability to quickly and easily generate a simple white labeled application using the SDK and available example apps.

The open-source application architecture provides intuitive ways for users to interact with non-blockchain based web applications, or the real-world in real-time, and to easily commit outcomes to the blockchain for contract fulfillment. The hosted application and oracle architecture provides the same benefits to businesses and websites.

The use of collateral to secure the Valenode network and therefore the encrypted communication protocol has advantages over previously designed private communication networks - primarily because it is financially difficult for someone to acquire enough nodes to compromise communications.

Using isolated application chains, selective-blockchain clients, non Turing-complete smart contracts and eliminating blockchain bloat caused by unnecessary token transactions leaves Valence powerful enough to cater for most real world use cases yet lightweight enough to scale.

Abstracting data away from cryptocurrency blockchains and into a highly specialised and secure platform creates the opportunity for Valence to become the ubiquitous decentralised data source used to fulfil smart contracts for the thousands of cryptocurrencies which already exist.

`https://valenceplatform.org`