BETHEL Blockchain Development Platform
(B_2 Platform)

[Whitepaper – Version 1]
August 21 , 2020

By Future CX Team
http://www.futurecx.com.au

Abstract

The invention of the blockchain and the distributed ledger technology has offered the world of abilities to decentralize all aspects of blockchain eco system in a more effective and a secure way thus, allowing innovation and development to thrive within. Even though the positive aspects of the blockchain and distributed ledger technology is such, integrating it and utilizing to a working model has always been a strenuous and a challenging task.

The paper presented herein proposes ***B_2****, a* ***Multi-Language Containerized blockchain development platform*** *coupled with a ledger ecosystem, also known as '****Smart Container'****. Such a ledger ecosystem would have useful applications in Finance, Trade, Healthcare, Government enterprise, AI, IOT and many more. A unique feature* ***(Multi Language)*** *in* **B_2** *allows even an amateur blockchain developer to interact with blockchain ecosystem easily with short learning and development curve. Any developer looking forward to utilize blockchain or distributed ledger technology for their software development can do so with minimal effort.*

B_2 *ecosystem is made up of clusters of trustless nodes consisting standard and guardian nodes in-par with notary standards, forming an integrated universal database which facilitate data retrieval. Fundamentally, this system enables and provides any individual or entity with the ability to develop their data collection, retrieval and storage inclusive of data state facilities incorporated in their operation.* ***B_2*** *Platform embedded with all benefits that a blockchain environment provides. Enhancing their data storage and retrieval efficiently with improved data security, reduced cost and uptime due to decentralized nature of the proposed ecosystem. The general introduction presented in this paper envisions all fundamental aspects of* ***B_2*** *Platform and the technical paper that follows will go into depths of its functionality.*

Contents

Contents

1. Introduction

B_2 as a platform, functions up to its mark in decentralized space for all the blockchain development needs to take place with its scalability across the board. It out-performs any current data storage system [1], which goes beyond merely saving but smooth and secure transaction of data. Furthermore, reducing cost with higher levels of privacy [2]. In **B_2** we believe privacy to be the paramount factor in the fore-front of data transmission within a decentralized space. Usage of **B_2** ledger ecosystem is limitless for any individual or entity looking forward to make use of the capabilities that the **B_2** platform offers. **B_2** Platform portrays the future of data storage, with zero duplication [3], ensuring automated and error free smooth transmission of data. Work within the **B_2** Development Kit (**B_2**DK) and related modules, transaction of data and retrieval adhered to the current data transmission standards [4] in a better improved way.

B_2 Platform shares **B_2** Development Kit (**B_2**DK) to transform the future of data representation, storage, retrieval, privacy and using the ability of data state logging mechanism [5]. Most importantly, the **B_2** ecosystem working independently in their own operations and together with each other to make up the trustless data transacting environment that we envision and aspire to bring forth.

B_2 mainly focuses and caters to the developer community and extends to all industries looking forward to utilize blockchain technology for their development needs. **B_2** extends its capability beyond simple data storage or information transaction covered by many players. **B_2** focus towards helping the developers who keep the systems running and innovate to utilize decentralized ledger ecosystems and platforms to achieve their development goals within the blockchain space. Containerized Smart Contracts within the **B_2** ecosystem is end-user customizable with unique ability to integrate instructions set forth by the developer. Business logic can be set forth by the end user or the developer using **B_2** Development Kit. Governance being a part of the blockchain technology, **B_2** offers a complete customizable aspect catering developers' need and requirements. Proof of Distribution [POD] Consensus [Annexure I] being a unique part of the governance function of **B_2** Platform where a developer will be able to customize through the consensus-governance mechanism provided in the **B_2** Development Kit. Consensus can be customized to any and all requirements of the end-user/developer in any programming language for the utilization and implementation of the business logic. Smart Containers could be written with Software Development Kits Starting with JAVA, Rust, Node.js, Python, up to seven languages named C++, Go and .net.

2. Context

The design of **B_2** Platform was a result of a long-term research carried out in the blockchain eco system. The requirements of a secure transaction with the ability of multi-language in blockchain platform ecosystem was the impetus for the idea of **B_2** Platform. After a comprehensive Research and Development work and studying the issues in the current system, **B_2** multi-language Platform is designed by the **Future CX** architects which became a success. The motivating issues, which **B_2** seeks to solve is, non-availability of a multi-language platform for the developers, with the ease of the extensive training and development requirement. **B_2** provide unique facility for developers with Containerized Smart Contracts as privacy, be the paramount factor in the fore-front of data transmission within a decentralized space. Regardless of numerous blockchain platform which provide a certain degree of data security we see an inefficient pattern in today's business environment. The developers who depend on current open source and enterprise platforms limit their capabilities due to restrictions in user languages in the system. Learning different languages cost time and money and also the learning curve cause significant drawbacks in their efficiency. Applications restricted to a particular platform by an individual with limited literacy in the multi-language blockchain environment.

With the current global data, duplication can lead to inconsistencies, and it drives a need for costly matching of data reconciliation and fixing errors by individual and among various parties in a transaction.

To the extent that transaction between two firms could be viewed by a third firm as the unwanted transaction, this is also a source of risk, and some of it potentially systemic.

Centralized market infrastructure utilities have moved way towards increasing the amount of data and business-logic shared between firms. But this development creates unfavorable trade-offs of its own, such as the possibility of manipulation of data by anyone who has the access to the system. This has been evident how the world of financial and data transactions still lags far behind which has been achieved in the realm of consumer and web-based software, since the advent of the blockchain ecosystem.

The evolution of data transaction protocols through blockchain technology [6] has provided an excellent opportunity to upscale the efforts into developing solid data storage systems with beneficial capabilities such as data state embedding. Stepping into the future with stable data recording systems decentralized in nature while providing an unbroken privacy. Setting new standards of data transaction protocol avenues, by providing a smooth and workable environment with innovative architectures be built on suiting the actual needs which enable complex development and setting new standards in the decentralized data transaction space.

We believe that the maturation of cryptographic techniques, exemplified in part by what is commonly referred as "**Smart Container**", provides a new opportunity with a possibility of authoritative system of records that are securely shared between targeted audience, and which enable a large subset of each firm's transactions to be managed in a common but need to know basis.

Systems that provide a guarantee to be the best in blockchain environment, lacks developer skills, non-interoperability and also isolated systems, limit abilities, cost time, duplication and error.

In essence, we foresee an era of Blockchain industry that is developed and optimized at a higher level of privacy, flexibility, scalability, performance and security.

The authors of this white-paper can be reached via email: (info@futurecx.com.au)

3. Vision

We ,**Future CX** (**F**uture **C** =Blockchain **X** =for Multiple Industries) aspire to improve our world and everyone's lives with the adoption and use of blockchain technologies.

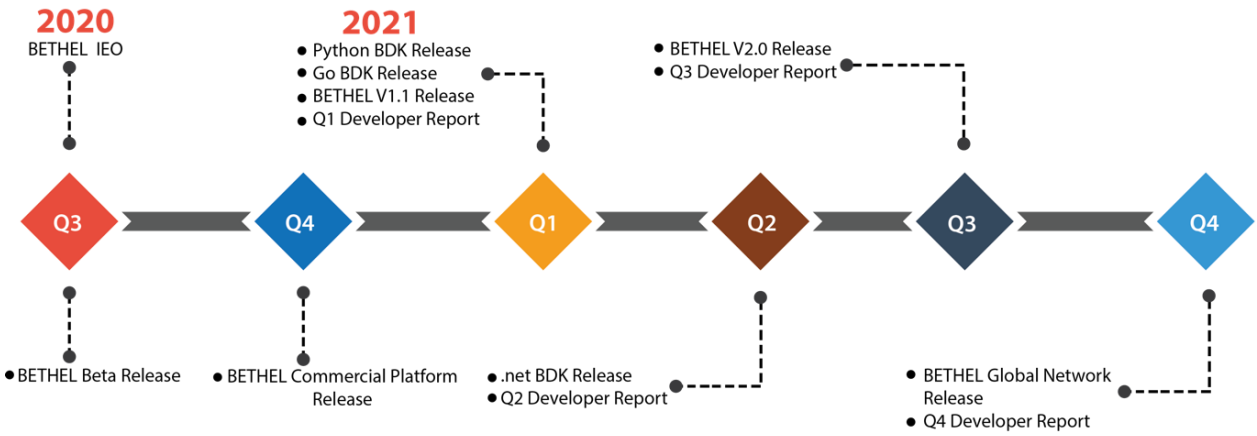
We aspire that governments and businesses will be more efficient, effective and secure, that the individual members of society will be the primary beneficiaries.

No single company can unlock the potential and full business value of blockchain on its own. Productive ecosystems establish a true platform and drive the value of any real blockchain network. **Future CX** provide **B_2** Platform with a range of opportunities to blockchain developers, ease of mind and innovative technology to unleash and accelerate their potential of growth.

B_2 Platform is designed to allow blockchain developers the freedom of using multi-language design capability with minimal learning and development curve. This stands in contrast with other enterprise-focused blockchain platforms, which are designed to be used a single language and in isolated instances of two languages in each supported application. A unique feature of Containerized Smart Contracts (**Smart Container***)* in **B_2** Platform will close the gaps in security of privacy, be the paramount factor in the fore-front of data transmission within a decentralized space. **B_2** Platform offers ultra-fast data transmission(12) within and outside of the ecosystem which is unmatched in the current industry.

Our mission is to become the most trusted and innovative blockchain provider in the industry.

TIME LINE



4 B_2 Platform

B_2 Platform consists of a network of trustless nodes working independently with each other transacting information within and outside of the network and as required with the use of an Oracle. The nodes consist of standard nodes as well as guardian nodes inclusive of the notary service, where transactions are verified and protected against double spends and integrity. All transactions that occurs in the **B_2** network is saved and treated as Documents within the network. The network also has the ability to save such documents containing information in two states, one being the original information embedded and if such information was changed, the previously original information would be marked as Archived and the updated information treated as the original. **B_2** Platform consists of separate channels cater to respective transmissions of data within the network and clearly managing transmission traffic through ultra-speed within the network and outside if required.

The storage of information within the network is facilitated by transaction persistence protocol where the business logic can be implemented by whichever database protocol that is needed by the end user giving them the flexibility to their own ways of development aspects.

B_2 offers a complete Development Kit (**B_2**DK) that caters to the open source as well as the enterprise level usage. It comes with a complete setup with all the capabilities and abilities of functioning a blockchain.

5. B_2 Principles

5.1, Privacy

B_2 envisions complete privacy for data transmissions, within and outside of the **B_2** ecosystem. Taking into account exploitation of the current data security loopholes in the blockchain industry as well as data transmission related industries, we, **Future CX** have come up with a unique privacy principle through the **B_2** Platform.

B_2 Platform offers the end user with the ability to integrate and use the unique **B_2** Channel Module which provides designated channels to cater nominated entities and departments within a designated model respective to their data transmission needs.

B_2 channel module utilize four types of channels:

- Nodes channels
  - Standard Nodes
  - Guardian Nodes
  - Micro Nodes
- Private channel (Includes Message Routing)

5.2 Flexibility

B_2 Development Kit (**B_2**DK) provides a number of languages to support the developers to work with. It is well known fact that developers looking to utilize blockchain technology in their blockchain applications would need to be aware of the functionality and the implementation of the blockchain related jargon. However, setting up a blockchain and customizing it for individual requirements is a time consuming and a tedious task. The time consumed for such a task and learning the blockchain functionalities across the board and setting up the same would consume time that they would otherwise utilize to enhance their core project development aspects. The flexibility of the **B_2** Platform is such that **B_2** offers the end user, the ability to utilize blockchain functions in their development needs with complete ease of efforts.

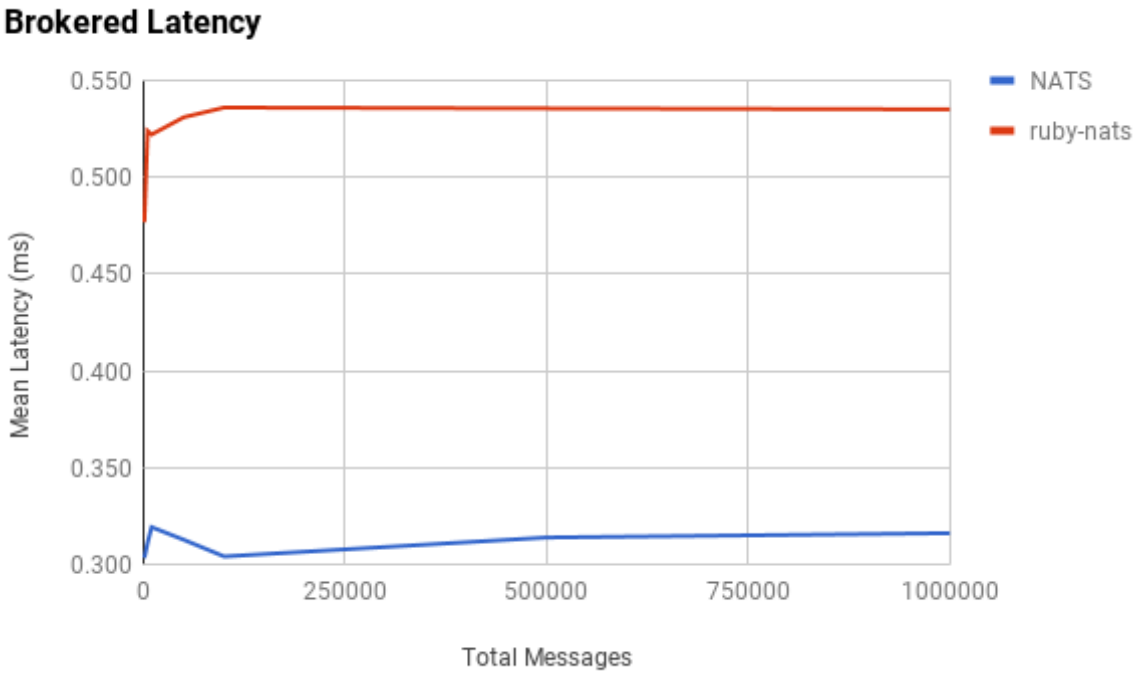
5.3 Scalability

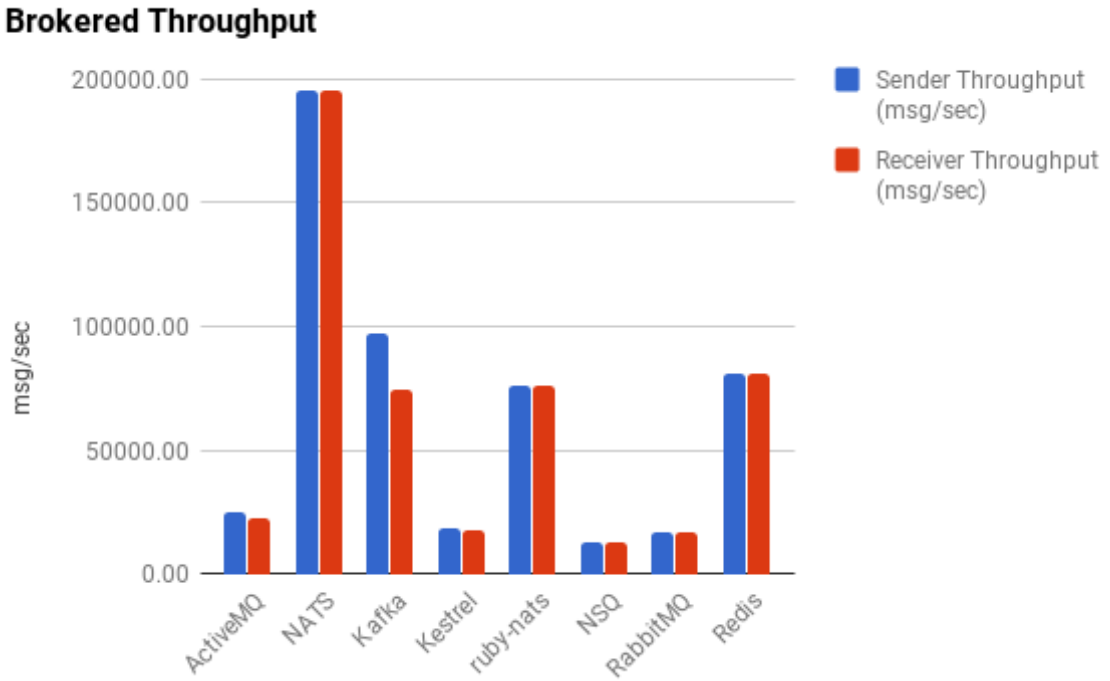
Understanding the unique needs of developers and their project requirements, it is a known fact that there are questions and limitations when it comes to use development platforms and development environments. Therefore, **B_2** offers enhanced scalability and increased engagement to developer to scale and allocate resources with complete flexibility within the entire **B_2** Ecosystem and to be able to customize the entire blockchain function to cater to their development needs. **B_2** Platform makes it so simple that these

customizations and scaling can be achieved through the use of a configuration file or through a tick of a check box.

5.4 Performance

In the present world of information super high way, speed is a key factor in data transmission [12], Even though there can be other bottlenecks preventing these achievements due to respective architectures and implementation, **B_2** Platform will be in the forefront of achievements which enable transmission, transaction, verify data and massages in capacities amounting to millions per second. **B_2** Platform offers ultra-fast data transmission within and outside the ecosystem through the use of NATs P2P protocol [13]. Thereby having the ability to transmit, transact, verify, data and messages up to 1 million transmissions per second [14].





5.5 Security

In today's digital age of data security and privacy, attackers with various ulterior motives utilize vulnerabilities and loopholes within the system to steal or to obtain unauthorized access and gather sensitive information undermining privacy of an operation or an individual. **B_2 Development Kit** offers the unmatched advantage of the Double Encryption Technology [15] within its ecosystem. Data stored in the ledger ecosystem and all data passing through all communications, transactions and transmission channels within the **B_2** Platform is double encrypted ending any fears of data breaches or sensitive information falling into the wrong hands.

6. True Decentralization

The term True Decentralization focuses on having an architecture in **B_2** Platform which is capable of operating on its own without the dependence of a third party. Therefore, having the advantage of less failures and factors affecting such situations.

The entire operations within the B2 Platform from the point of deployment to verification of the deployed containers through the validator nodes are handled by the **B_2** container manager (Annexure I) and therefore, no third-party involvement in the said operations, which is native to the B2 Platform.

7. Smart Containers

A virtual container (unique to **B_2** Platform) having smart contract [7] or business logic within and the ability to execute instructions set forth by the end user. Smart container is able to operate within a single instance or in a context of multiple instances of decentralized environment without the involvement of a third-party. The

main role of the smart container is to hold the entire set of instructions/business logic and act on it as set forth by the developer.

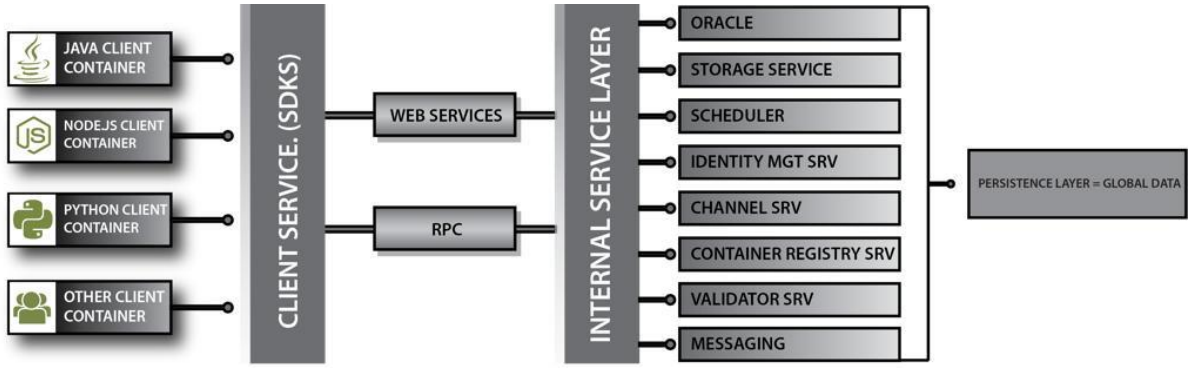
8. Global Data

B_2 Platform a global database having all the transactions occurring within Distributed Ledger to ensure transaction integrity, versioning and data state verifications. (**Diagram B)**

9. B_2 Platform Architecture

B_2 model consists of cluster nodes. Each cluster consisting of guardian, standard and micro nodes. The guardian nodes holding the ability to initiate containerizing operations and assist directly with the dockerization loading aspects of the entire **B_2** Platform. The standard nodes being the holding node of the entire network platform together. Micro nodes cater to IoT use cases directly engaging with interrelated computing devices. Each cluster is a part of the distributed ledger aspect of the **B_2** Platform, which acts as the main embeds within its transactions. All deployments of dockers, container changes and their state, verified against container registry for any changes done and authorized by respective user when initiated the respective container using **B_2** Platform. (Diagram A)

Diagram A



- User utilizing the **B_2** Development Kit has the liberty to choose their docker package catering to their needs and choice. And start putting together their business logic as a smart contract on the provided docker container, respective to their preferred programming language.

- Once the docker container has been updated with the business logic, and the smart contract then containerized into a docker, ready for deployment.

- The docker is then prepared to be deployed, thus the request is received by a validator node obtaining the docker signature for the next stage of verification prior to full deployment.

- The obtained Docker signature is therefore, embedded into the ledger as a (UXTO) having its output Docker ID Signature and Public Key.

- The transaction generated and embedded into the ledger is then added to **B_2** global network, a database backed by MongoDB within the persistence layer of the entire **B_2** Platform. This particular database acts as container registry, having the container information, its Transaction ID, Signature and Public Key. The first entry to the record set is named as Version One.

- The deployment stage is assisted by **B_2**'s very own Proof-Of-Distribution concept, (Annexure I) where a docker is signed by the user and the unique public key of the container is looked up by the container registry against the existing public key, matching the deployed container. If a match is found a transaction is generated and embedded to the blockchain with container information and the container granting access to be synced into the guardian nodes for deployment and implementation.

10.Principle Components

10.1. Proof-of-Concept (POC)

Term 'Proof of Concept'[17] brings a significant terminology to the blockchain space. **B_2** terms its use of Business logic within the **B_2** ecosystem as Proof-of-Concept.

10.2 B2 Proof-of-Distribution (POD) (Refer Annexure I)

B2 Proof-of-Distribution clearly states the nature of its function. The main function of the proof-of-existence is being, verification of information existing within a container and against the original state of data in such container during operations, such as retrieval and/or updating of data. Such data distributed within the **B_2** ecosystem is then verified against the registry for its existence sequentially within the respective containers. Therefore, performing verification checks against existing data. Unique POD consensus in **B_2** Platform is developed by **Future CX** architects.

10.3 Proof-of-Value (POV)

B_2 Proof-of-Value serves the purpose of verification of containerized data and its content against the business logic or smart contract functions. It was initially written against the Distributed Ledger. The Proof-of-Value concept mainly focuses on verifying the existing data for integrity on the global ledger against the blockchain as well as updating of such information and semantic versioning of updated information. (Diagram B)

10.4 Storage Service

Storing of data within the blockchain operational aspect occurs as transactions embedded into the blockchain and the storage of supportive information

inclusive of business logic is stored on the NOSQL backend through transaction persistence where, transmission of data, verification and messaging is facilitated between the blockchain, backend through the ultra-fast NATS P2P transmitting protocol. Same is being used in order to transmit information to and from the blockchain ecosystem, documents and the backend.

10.5 Container Registry Service

The role of the container registry is to have its database backed by MongoDB to have all deployed container information as a Registry. It will have the container Signature, its Transaction ID and Public Key and the first entry to a record-set named as Version One.

10.6 Validator Service

The validator service plays the important role of verifying the container registration, container transaction and the transactions in Distributed Ledger and also its task includes to maintain integrity of possible double-spends and other data mis-matches.

10.7 Identity Management Service (IMS)

IMS is the implementation of a wallet based or blockchain based authentication with the use of a Public and a Private key in order to use the **B_2** Development Kit. In contrast to otherwise use of a “username” and a “password”.

10.8 Scheduler Service

The scheduler service allows scheduling of the transaction occurrences and signing of the same on a future date as per schedule to occur without any involvement of a third-party and within the decentralized space of the platform.

10.9 Messaging Service

Messaging transaction protocol within the platform is unique to **B_2** Peer to Peer (P2P), as the transactions are not broadcasted publicly and are transmitted privately to the respective P2P endpoints in full duplex.

10.10 Channel Service

B_2 channel module utilize four types of channels:

- Nodes channels
  - Standard Nodes
  - Guardian Nodes
  - Micro Nodes
- Private channel (Includes Message Routing)

The above channels are used by the respective smart containers as per the smart contract or the business logic stated by the end user and instructions set forth therefore.

10.11 Oracle Service

B_2 Platform offers an Optional Oracle Module, where operations can be handled from a higher level of authority and major changes can be authenticated and implemented via the **B_2** Oracle Module. It uses for external data access and work as a cross service with the other APIs.

11. B_2 Global Network

The global network is backed by the MongoDB, which makes a part of the backend of the entire **B_2** Platform, the global network within its operation respectively handles container registration, verification and deployment authorizations.

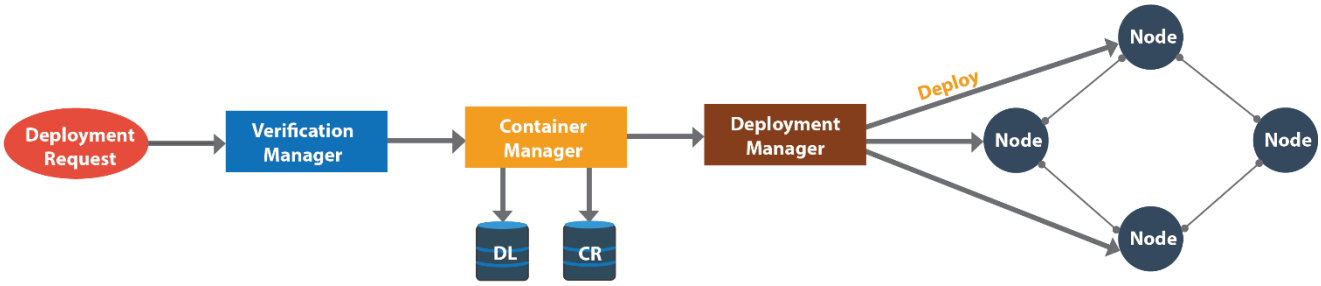
12. Smart Container - Proof-Of-Distribution (POD)

A user is given the ability to use a supported language of their choice in order to write the smart contract or the business logic or the set of executable instructions into a container [Annexure I] with the ability to be dockerized. The docker created is then requested from the node network to be added into the validator node. In this instance docker signature obtains and verifies against the container registry for a presence of a docker with the same signature and further verifies against the distributed ledger for further confirmation. If no match was identified for the docker signature presented, it is added to the container registry after creation of a transaction containing its information on the distributed ledger. The information embedded on the transaction in the distributed ledger would then be the Public Key of the user, Docker Signature and the Docker ID. The same information is also added to the container registry inclusive of the Transaction ID and the docker version becoming One and its state becoming Active. The docker therefore is then allowed by the Node Gateway to be pushed to the respective Nodes as per the smart contract or the business logic on the dockerized container.

If an existing user needs to update an existing dockerized container and once such a request is made as above, the same route of verification follows where the user Public Key and the dockerized signature is verified against the container registry as well as the distributed ledger for its existence and authenticity. If no user was found with the Public Key match, the request is then and therefore rejected. If the user was verified and the request was found to be authentic, then the docker signature is updated for the new one on the container registry after a creation of a transaction on the distributed ledger and therefore the same is updated on the container registry inclusive of the Transaction ID. The current docker version becoming Two and its previous state becoming Archive and the current state becoming Active, and the docker is therefore, authenticated for deployment throughout the respective nodes and the previous docker undeployed. **(Diagram 1, 2 and 3 of Annexure I**)

(Annexure I Diagram 1)

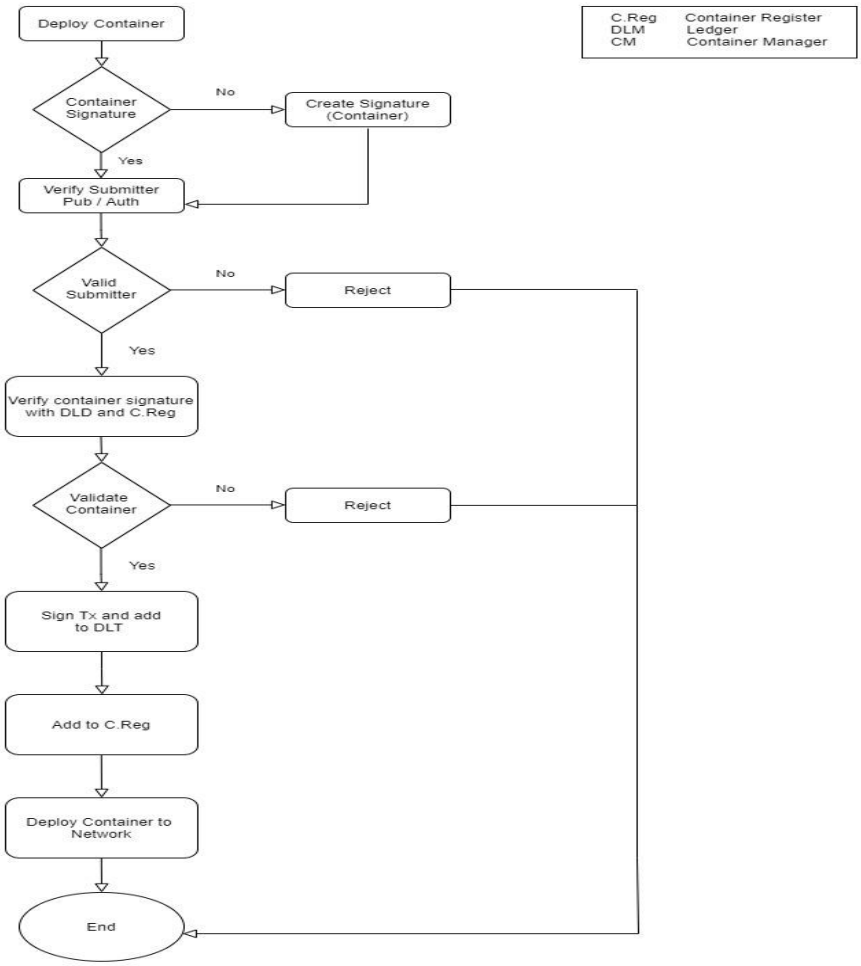
Proof of distribution Architecture



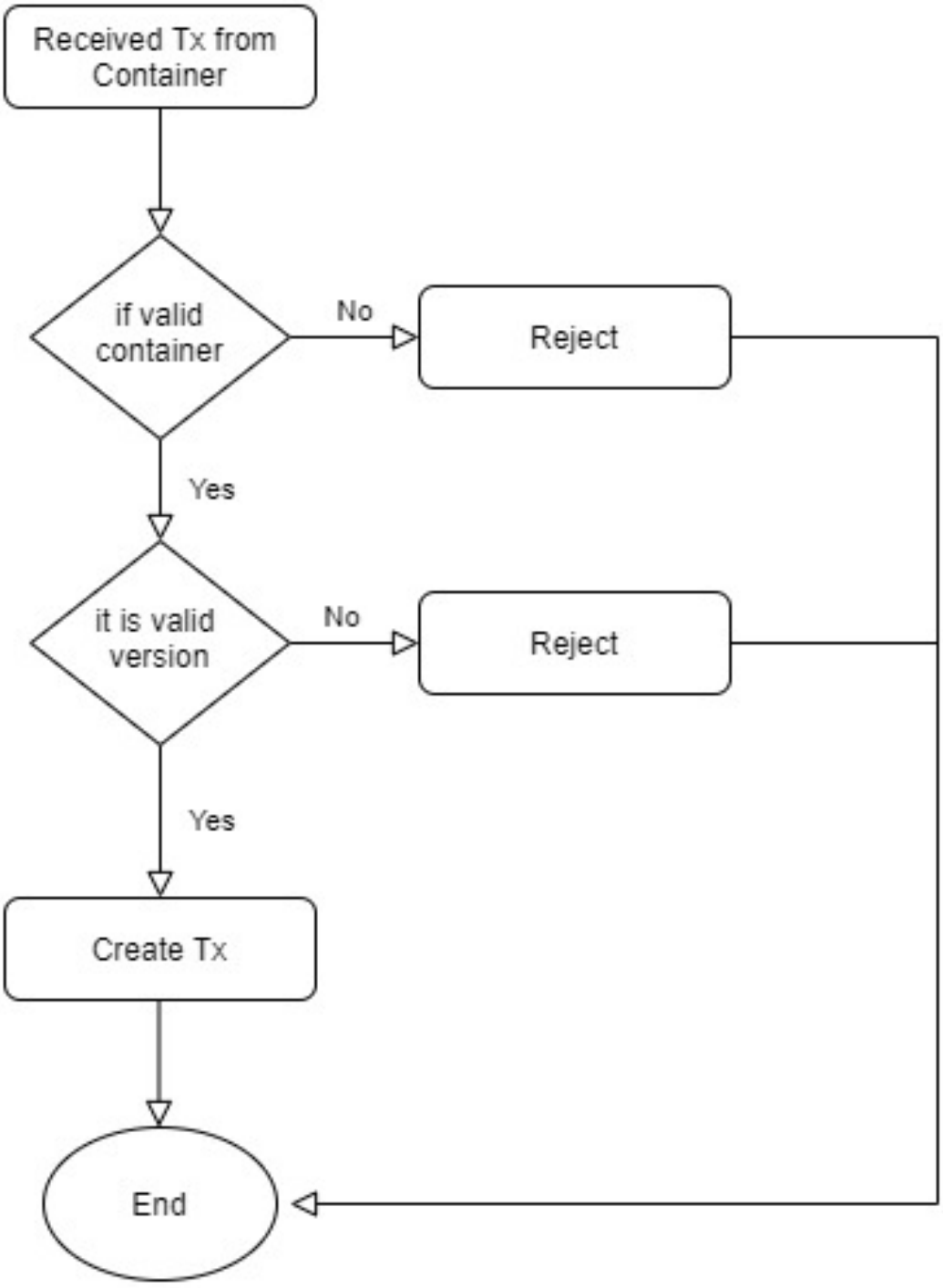
DL - Distributed Ledger

CR - Container Register

(Annexure I: Diagram 2): Container Deployment and Verification



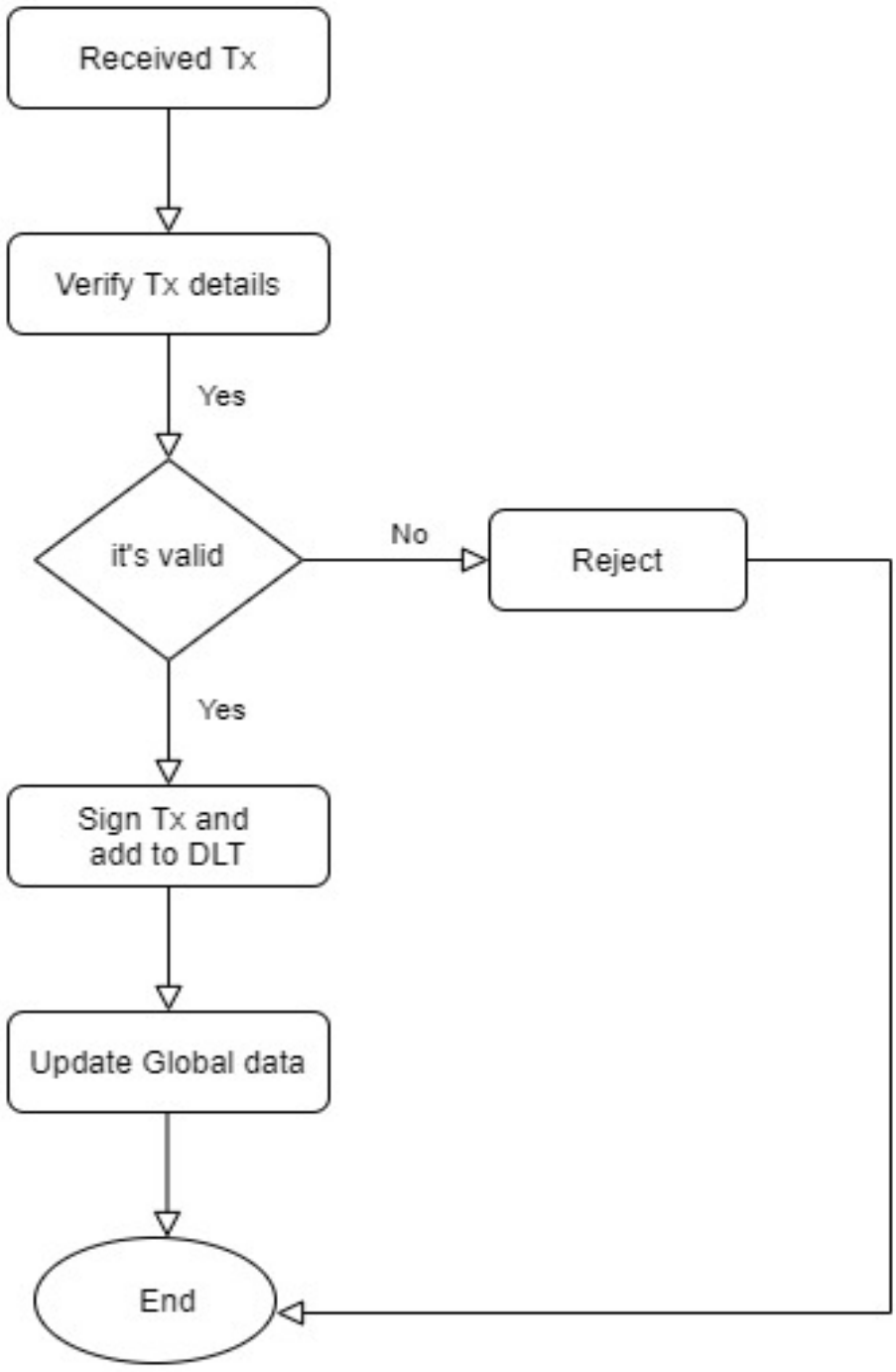
(Annexure 1 Diagram 3) :Container Transaction Verification



13. Global Data - Proof-Of-Value (POV)

The Proof-Of-Value is the verification of all data within the subsystems of the **B_2** Platform and the verification of data and the data synchronicity between the distributed ledger and the container registry and all values within it to be verified against each other for inconsistencies and mismatches. (Diagram B)

Diagram B:



14. Cryptographic Layer (Signature Analysis)

14.1 Transaction Signing

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

Fig 1. Public-key algorithms for different security levels

Transaction signing [8] occurs at the first stages of deploying a dockerized container, where as soon as the validator node verifies the origin of the Public Key of the dockerized container and a transaction is created with its information on the distributed ledger.**(Fig 1)**

14.2 Block Signing (Multi-Signature)

Benefits of the Multi-signature brings about immense security paybacks in block creation and signing, the same applies when it comes to the **B_2** decentralized operations,

14.3 Container Verification

Container signature verification is entirely handled by the Validator Node and portrays a 1024-bit key using asymmetric RSA.

15 Leader Election

Leader election is decided as per the availability of nodes to carry out a certain operation within the **B_2** network, as experiences gained through the Byzantine leader elections problems in the dynamic networks, thus depicting the election of a leader on a Synchronous ring [9]. (Fig 2).

Algorithm 8 Clockwise

1: **Each node** v executes the following code:
2: v sends a message with its identifier (for simplicity also v) to its clockwise neighbor. {If node v already received a message w with $w > v$, then node v can skip this step; if node v receives its first message w with $w < v$, then node v will immediately send v.}
3: **if** v receives a message w with $w > v$ **then**
4: v forwards w to its clockwise neighbor
5: v decides not to be the leader, if it has not done so already.
6: **else if** v receives its own identifier v **then**
7: v decides to be the leader
8: **end if**

Fig 2. Algorithm Clockwise

16 Smart Contract

16.1 Multi-Language

Smart contracts or the Business logic can be stated in all supported languages on the **B_2** Platform.

16.2 Container Infrastructure

The container infrastructure mainly contains the smart contract or the business logic or any instruction that the developer states therein (Annexure I) that will assist the dockerized [16] container to carry out its functions and operations with the node network as instructed by the developer.

16.3 Smart Contract Languages

B_2 Supported smart contract languages include, Java, Rust, Node.js, Python, C++, Go and .net

17 Business Logic

B_2 smart contracts are the blood that keeps it alive, where all the fundamentals of **B_2** Platform come together, the operations such as Proof-of-Concept (POC) [17], Proof-of-Distribution (POD) and Proof-of-Value(POV) makes what **B_2** Platform is for everyone.

18 Governance

B_2 Platform offers complete customization of governance needs of the blockchain user decides to set. The user would have complete control over the needed governance mechanism for their application to perform.

19 Network Parameters

B_2 offers complete customization of network parameters as per user requirements and as needed by them for their deployments. This is achieved right through the implementation of business logic /smart contract system. The user being given the ability to decide on the network parameters of the blockchain that is being setup for them. This is completely achieved through the business logic smart contract system.

20 Comparison with other Platforms

B_2 Platform stands out when it comes to the following areas in the industry and usage of blockchain technology towards ledger ecosystem.

- When it comes to industry focus **B_2** Platform mainly focuses and caters to the developer community and extends to all industries looking forward to utilizing blockchain technology for their development needs.

- The usage of **B_2** Platform extends beyond simple data storage or information transacting where these aspects are covered by many players out there. As **B_2** our focus is towards helping the ones who keep the systems running, the developers who innovate and look forward to utilizing decentralized ledger ecosystems and platforms to achieve their development goals within the blockchain space.
- Smart Contracts within the **B_2** ecosystem is end-user customizable by **B_2**'s very own ability to integrate instructions set forth by the developer, business logic is a term we use to identify the same and this logic can be set forth by the end user or the developer by the use of the **B_2** Development Kit.
- Governance being a part of the blockchain technology is offered by **B_2** Platform being a complete customizable aspect catering to the developers' requirements and needs, which thereby can customize simply the business logic implementation.
- Consensus being a part of the governance function of the entire blockchain. it is also able to be customized through the consensus governance mechanism provided through the **B_2** Development Kit which is customizable to any and all requirements of the end-user/developer, whatever the programming language is utilized for the implementation of the business logic.
- **B_2**'s mode of operation is as customizable as **B_2** itself to user needs. It completely customizable to the development needs of the end user as **B_2** offers three modes of operations and channels, Public, Private and Hybrid abilities, which the end user can benefit from for their development needs.

21. Conclusion

As discussed in the entire white-paper, **B_2** Platform stands out the rest and provides a dedicated blockchain development platform for any developer to utilize on their innovations. Providing a platform with unmatched range of tools for anyone to work on their very own projects that shape a blockchain module or would like to look forward to use blockchain technology on their existing implementations. Either way **B_2** is ready to provide and function on all those aspects limitlessly.

B_2 employs a Multi-language and containerize block chain platform with true decentralized nature which is accessible by both private and enterprise, ensuring a high level of integrity and security for all the transactions happening on the blockchain network [3]. **B_2** is containerized, which can hold any form of information such as transactions, records, and events with defined rules for information updates[10].

Some of the unique features of **B_2** are the proof of distribution (POD) used to ensure verification of containerized smart contracts (smart container), Proof of

value (POV) which used to verify data transactions in distributed ledger before adding to global data. **B_2** development kit which allows room for innovation, save time on learning and development curve. Public, Private and hybrid networks. ensuring the highest level of cyber-security and privacy[11].

Deploying multi language blockchain platform across the developer network will improve advance digital diversity in all enterprises, by making the processes trustworthy with guaranteed, Privacy, Flexibility, Scalability, performance and double encrypted transmission and transaction security.

B_2 will look forward broadening its presence in the Decentralized Finance space, as well as focus into the usage of **B_2** Platform and the Oracle module jointly in pursuing future goals.

22. B_2 Architecture – Terminologies

- **Network**

A network consists of the entire blockchain and sub systems working independently and together to carry out the end operation of a decentralized information transaction.

- **Nodes**

Nodes are points of the network which are working independently and with other nodes, holding the entire blockchain together. The internal processing of information, signing, verification and transaction messaging originates within a node and is transmitted to the entire network.

- **Transaction**

A transaction consists of inputs and outputs of data and generates a signed hash verifying the contents of the transaction and its consistency, this information is then distributed to the entire network through nodes and embedded to the blockchain.

- **State**

The state of a transaction is the state of the outputs generated against a transaction and later becomes an input for the same transaction. **B_2** refers state as changes to the information that is saved within the blockchain, if the information has been updated or left as it is.

- **Channels**

The channel as referred here is the number of modes of connectivity that is offered within the blockchain, outside or both. These channels offer transmission of information within the blockchain ecosystem, transmission of information outside, vice versa and both aspects together.

- **Contract**

Contracts within the **B_2** ecosystem is treated as logic and the set of instructions that the end user would like the system to store and perform. It

is also termed as business logic and include a number of steps or instructions with involved protocols that the end user would like the system to carry out on behalf of the entire operation.

- **B_2 Development Kit - Open Source**

The **B_2** Development Kit -Open Source consists of all aspects of the entire **B_2** ledger ecosystem with working examples of implementations and uses respective to the programming language needs of the end user.

- **B_2 Development Kit - Enterprise**

The **B_2** Development Kit -Enterprise version consists of all aspects of the entire **B_2** ledger ecosystem with further additional functions such as private and hybrid channel capabilities as well as with a UI Studio Suite with all the customizable functioning and enterprise functional support.

23. References

[1] G. Abinaya, Preksha Kothari, Alex Pavithran KP, Manasi Biswas, Farheen Khan, "International Journal of Engineering and Advanced Technology (IJEAT)", ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019. [Online]. Available:https://www.ijeat.org/wp-content/uploads/papers/v8i4/D6206048419.pdf

[2] Rui Zhang, Rui Xue, and Ling Liu. 2019. Security and Privacy on Blockchain. ACM Comput. Surv. 1, 1, Article 1 (January 2019), 35 pages. https://doi.org/10.1145/3316481

[3] Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https:// bitcoin.org/bitcoin.pdf, 2008.

[4] GS1, 2019. GS1 Blockchain Standards. [Online]. Available at: https://www.gs1.org/standards/blockchain

[5] Neisse, Ricardo & Steri, Gary & Nai Fovino, Igor. (2017). A Blockchain-based Approach for Data Accountability and Provenance Tracking. 1-10. 10.1145/3098954.3098958. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3098954.3098958

[6] Paik, Hye-young & Xu, Xiwei & Bandara, Dilum & Lee, Sung & Lo, Sin Kuang. (2019). Analysis of Data Management in Blockchain-based Systems: From Architecture to Governance. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2961404. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8938787

[7] Buterin et al. A Next-Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/wiki/%5BEnglish% 5D-White-Paper, 2014.

[8] C. Paar and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners. Berlin Heidelberg: Springer-Verlag, 2010. [Online]. Available: http://swarm.cs.pub.ro/~mbarbulescu/cripto/Understanding%20Cryptography%20by%20Christof%20Paar%20.pdf

[9] H. Attiya and J. Welch, Distributed Computing: Fundamentals, Simulations and Advance Topics, John Wiley & Sons inc., 2004, chap. 3 [Online]. Available: https://doi.org/10.1002/0471478210.ch3

[10] (Sikorski, Haughton, & Kraft, 2017).(2017) Architecture of BCT-based supply chain system supply chain ... in privaten Blockchain-Anwendungen (**Sikorski**, **Haughton, & Kraft**, **2017**).

[11] (Kshetri, 2017) the impact of the blockchain on supply chain security (**Kshetri**, **2017b**)

[12] Ford, Bryan & Srisuresh, Pyda & Kegel, Dan. (2006). Peer-to-Peer Communication Across Network Address Translators. [Online]. Available: https://arxiv.org/pdf/cs/0603074.pdf

[13] T, Sharvari & K, Sowmya. (2019). A study on Modern Messaging Systems- Kafka, RabbitMQ and NATS Streaming. [Online]. Available: https://arxiv.org/pdf/1912.03715.pdf

[14] T, Tyler. (2014). Dissecting Message Queues. [Online]. Available: https://bravenewgeek.com/dissecting-message-queues/

[15] Nav Team. NavCoin: NavTech Decentralization. https://cryptorating.eu/whitepapers/NavCoin/NAV-Coin-Whitepaper.pdf, 2016.

[16] Willis. Docker: Docker and the Three Ways of DevOps. https://goto.docker.com/rs/929-FJL-178/images/20150731-wp_docker-3-ways-devops.pdf, 2015 Docker. Introduction to Container Security Understanding the isolation properties of Docker, https://www.docker.com/sites/default/files/WP_IntrotoContainerSecurity_08.19.2016.pdf, 2016

[17] https://en.wikipedia.org/wiki/Proof_of_concept

Annexure I

Proof-Of-Distribution Consensus Mechanism (POD)

[Whitepaper - Version 1]
June 30, 2020

P S Sanjeewa Silva
https;//www.linkdin.com/in/sanjeewa-silva

Abstract

The invention of data verification and distribution protocols, has got its own advantages as well as disadvantages of which a single point of failure being the main disadvantage of such existing protocols .

The paper presented herein proposes Proof-Of-Distribution, a blockchain based docker management system coupled with the features of a distributed ledger made up of clusters of trust-less nodes acting as a truly decentralized system. Therefore, curbing the issue of data distribution protocols having a single point of failure.

Proof-Of-Distribution

A user is given the ability to use a supported language of their choice in order to write the smart contract or the business logic or the set of executable instructions into a container with the ability to be dockerized. The docker created is then requested from the node network to be added into the validator node. In this instance docker signature obtains and verifies against the container registry for a presence of a docker with the same signature and further verifies against the distributed ledger for further confirmation. If no match was identified for the docker signature presented, it is added to the container registry after creation of a transaction containing its information on the distributed ledger. The information embedded on the transaction in the distributed ledger would then be the Public Key of the user, Docker Signature and the Docker ID. The same information is also added to the container registry inclusive of the Transaction ID and the docker version becoming **One** and its state becoming Active. The docker is then allowed by the Node Gateway to be pushed to the respective Nodes as per the smart contract or the business logic on the dockerized container.

If an existing user needs to update an existing dockerized container and once such a request is made as above, the same route of verification follows where the user Public Key and the dockerized signature is verified against the container registry as well as the distributed ledger for its existence and authenticity. If no user was found

with the Public Key match, the request is then and therefore rejected. If the user was verified and the request was found to be authentic, then the docker signature is updated for the new one on the container registry after a creation of a transaction on the distributed ledger and therefore, the same is updated on the container registry inclusive of the Transaction ID. The current docker version becoming **Two** and its previous state becoming Archive and the current state becoming Active, and the docker is therefore, authenticated for deployment throughout the respective nodes and the previous docker un-deployed.

Diagram 1

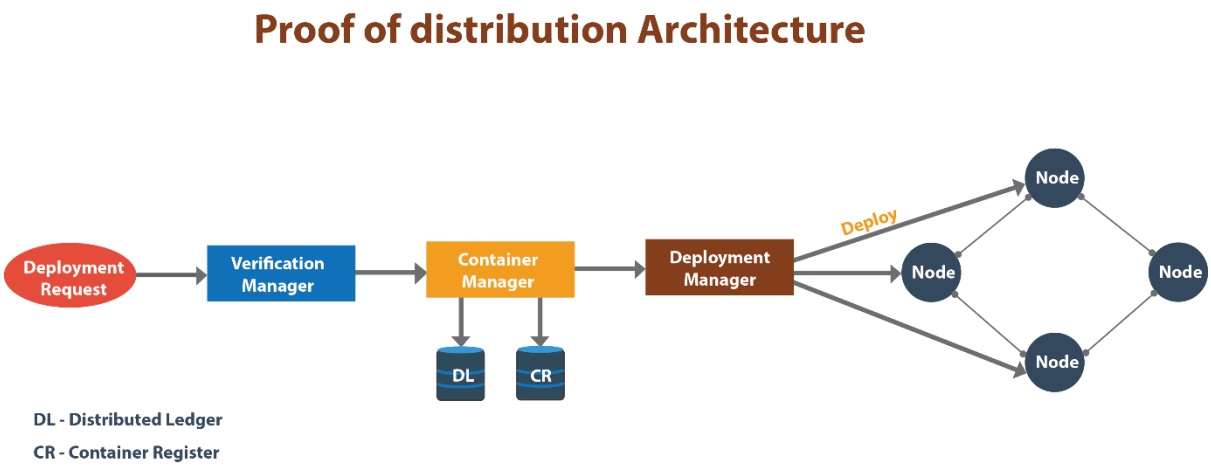


Diagram 2: Container Deployment and Verification

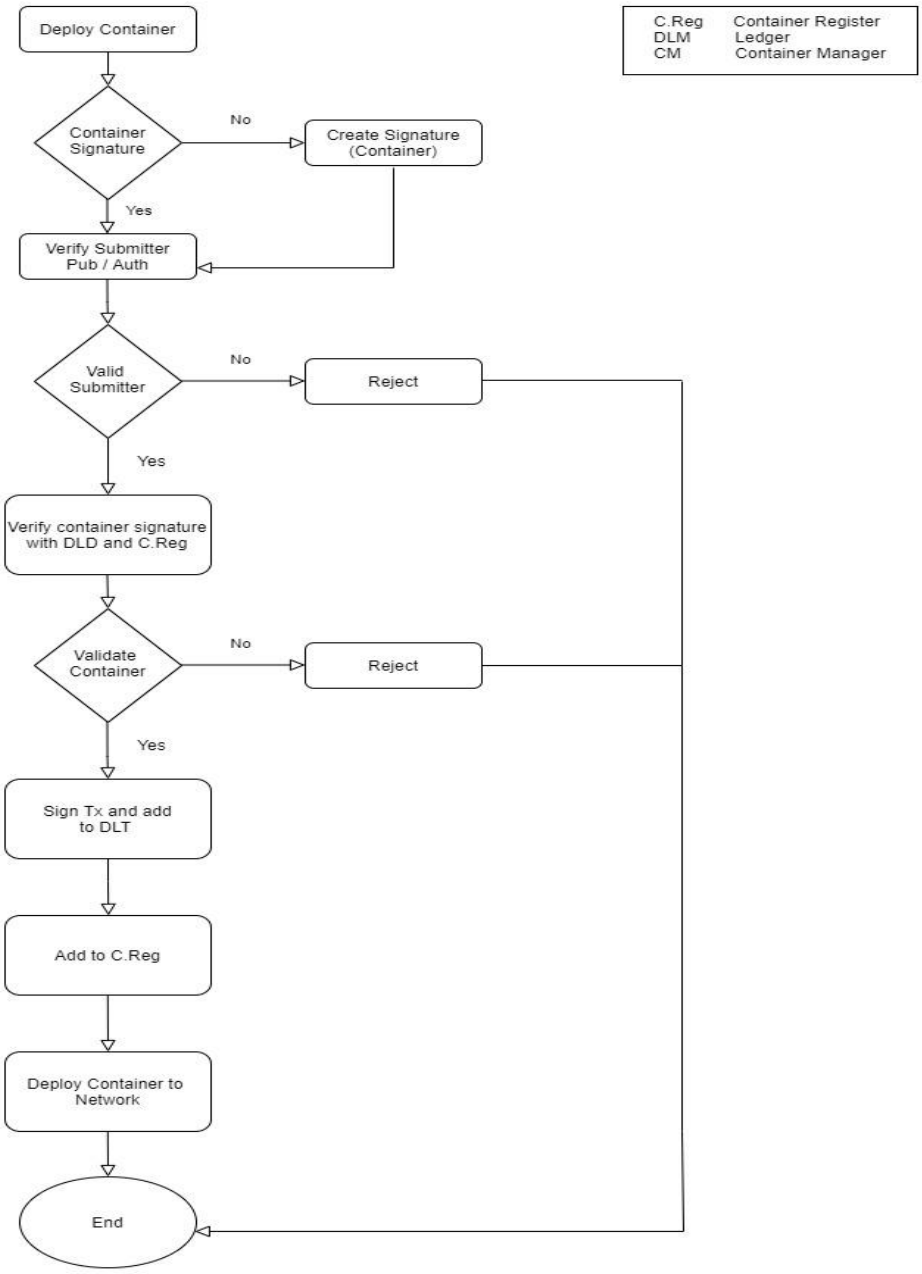


Diagram 3 :Container Transaction Verification