MultiChain

Multichain

Multichain is a free and open source blockchain platform to create private/permissioned blockchain networks. Multichain is an extended version of the bitcoin core software. The bitcoin engine provides security and control over peer to peer communications to Multichain.

Language support

The thing which makes Multichain more powerful is its support to 5 high-level languages, they are Python, JavaScript, and Ruby, Php, and C #. Multichain provides a simple API and a command line interface for the application development. The developer can download Multichain packages for all those languages from Github repository and start development. Multichain doesn't use cryptocurrencies and smart contracts. So that financial transactions are not possible with it.

Compared to other blockchain platforms average block time is very low in Multichain and it is about 2 seconds. But the speed increases the chance of hash collisions.

Security

The main feature of a Multichain is that the visibility of blockchain activities are kept private within the chosen participants. Only those selected participants can see the activities in the blockchain. It uses a set of collective admins, i.e.; a set of identifiable entities are defined as miners and all the validation/mining task will be done by them only. But the mining process doesn't involve the proof-of-work (PoW) scheme. Many blockchain networks including bitcoin use PoW scheme, in which a is a piece of data is created to verify the transaction which is difficult to produce but easy for others to verify

In Multichain, all the transactions between two participants are secured with a handshaking mechanism. In which a handshaking message and acknowledgment message is used to make sure that the exact participants are available on the communication channel.

Following steps will take place between two participants before starting a transaction

- Each node submits its identity as a public address in a permitted list.
- Each node will verify that the other nodes address is there in its permitted list
- Each node will send a challenge message to the other node.
- After receiving the challenge message each node will send back a signature of the challenge message providing their ownership of the private key corresponding to the public key.

If the node who sent the challenge message received a signature message then the transaction between those two participants will start. And if a satisfying signature is not received then the peer to peer connection between those two nodes will disconnect.

Mining

Multichain introduces a new parameter called ‘mining diversity’ for defining the mining process. The mining diversity is used for defining the participation of minors in the mining process. In Multichain, mining is done in a “Round-robin schedule”. The minors can create valid blocks in a round robin fashion so that all the minors can participate in the mining process equally.

The mining diversity is defined as

“0<= Mining Diversity <=1”

The ‘1’ represents that every permitted miner will participate in the round robin rotation and ‘zero’ represents no restrictions in mining. All miners can equally participate in the mining process and the validity of the block can be verified through different steps.

In a Multichain platform, there is no transaction cost or block rewards by default. But we can define those parameters in the params.dat file. The params.dat file contains several parameters for defining the blockchain behavior. There is an agricultural supply chain application already available on the Multichain platform. The app can control the entire supply chain system starting from the farmer to the customer. Each stage can be tracked through blockchain and the approach will help to increase the quality, reach and profitability of the product.

HydraChain

HydraChain is an open source blockchain platform, which is developed by Brainbot technologies and Ethereum project. HydraChain is an extension of the Ethereum Blockchain platform which provides support to create private/permissioned Blockchain networks. The supporting language for HydraChain is python. As an extension of Ethereum, HydraChain is fully compatible with all the API level and contract level protocols in Ethereum. There are several well-defined tools in Ethereum for creating smart contracts and DApps, (Decentralized Apps). You can reuse all those tools in HydraChain also. So it will be easy for those who know Ethereum to move on to HydraChain.

Smart contracts and HydraChain

Solidity/Serpent based smart contracts can co-exist in the same chain with the Python based smart contracts. Yes...! They are interoperable. Smart contracts created using HydraChain is independent of EVM (Ethereum virtual machine) as it is developed in Python programming language. The EVM provides a runtime environment for the contracts. EVM will execute all the untrusted codes and can provide security by restricting the accessing of each other's state. But Python based smart contracts will bypass the EVM so that the contract execution is fast. And we know, python is an easy to use language, less time consuming and easier to debug too. In fact, you don't need to go for a new language like solidity for developmental purpose.

How blocks are added?

Basically, HydraChain is providing the permissioned network creation services. So the validation is a great concern here. There will be a registered accountable validators in the network who is responsible for the validation of the blocks and transactions. In a HydraChain network, all the blocks are not allowed to enter the network without validation. That means a block will be added to the network only when the validators sign that the block is required. So once a block is entered into a network it is persistent. There are no reverts.

The HydraChain keeps a limitation in creating the blocks. HydraChain will create a new block if and only if there is a pending transaction. Whenever the Blockchain is unable to hold a transaction then it will create new block and validator has to sign it as a required block. The block will be added to the network after validation. As the validators are registered, a KYC is used for the participants, to make sure that the transaction are take place between registered participants only.The HydraChain platform provides a customizable nature in different components of Blockchain like transaction cost, gas limits, genesis allocation and block time. All these components have an inevitable role in a blockchain.

Transaction cost

The transaction cost in a HydraChain is the cost required for executing the computational steps in a transaction. In the HydraChain, you have the provision to configure the cost as per your requirement. The transaction cost can be calculated using an equation i.e.;

Transaction cost = Gas unit * Gas price.

Gas limits

The gas units are the basic units for an executed transaction in a Blockchain. A transaction can be divided into several ‘opcode’ and each opcode will have a specific number of gas units based on the type of the opcode.A ‘zerostep’ opcode has ‘0’ gas units, ’quickstep’ opcode has 2 gas units and a ‘faststep’ opcode has 5 gas units and so on. When a transaction is executing in a blockchain first it will extract the opcodes from the transaction, then the number of gas units will be identified from it, now the gas price will multiply with the total gas units, and that will be the transaction cost for that particular transaction. In HydraChain you can customize the gas limit also. Where the gas limits are the maximum number of computational steps in a transaction. For example: If a blockchain has a gas limit 50 then each transaction should have maximum 50 gas units.

Genesis Allocation in HydraChain

The genesis allocation is related to the hashing and mining capacity of a blockchain. The HydraChain has the power to customize its genesis allocation.

Usually, a Blockchain provides two types of mining methods they are direct mining and indirect mining. In Ethereum you will be using the cryptocurrency ‘ether’ for financial transactions. While creating a HydraChain you have the provision to decide whether to perform direct mining or indirect mining. Cryptocurrencies like ETH and ETC can be mined directly. And for other currencies like bitcoin, zeta coin etc. you can define an indirect mining with the help of sha256 hashing algorithm. In indirect mining, first you have to mine the Ethereum anyway, at payout time the ETH will exchange with other currencies of your choice at the latest exchange rate. This customization feature in mining will provide a facility to do financial transaction with all currencies.

Block time

One more customization is possible in HydraChain that is block time. Block time is the time delay between the validations of two blocks. An average block time in a Bitcoin Blockchain is 10 minutes that means it will take 10 minutes for the addition of a new block to a Blockchain.

Installation

HydraChain can be downloaded and installed from GitHub. It also provides easy deployment of networks. For the network deployment, several docker file templates are already available in HydraChain. These templates can be used while we creating new networks.

IOTA

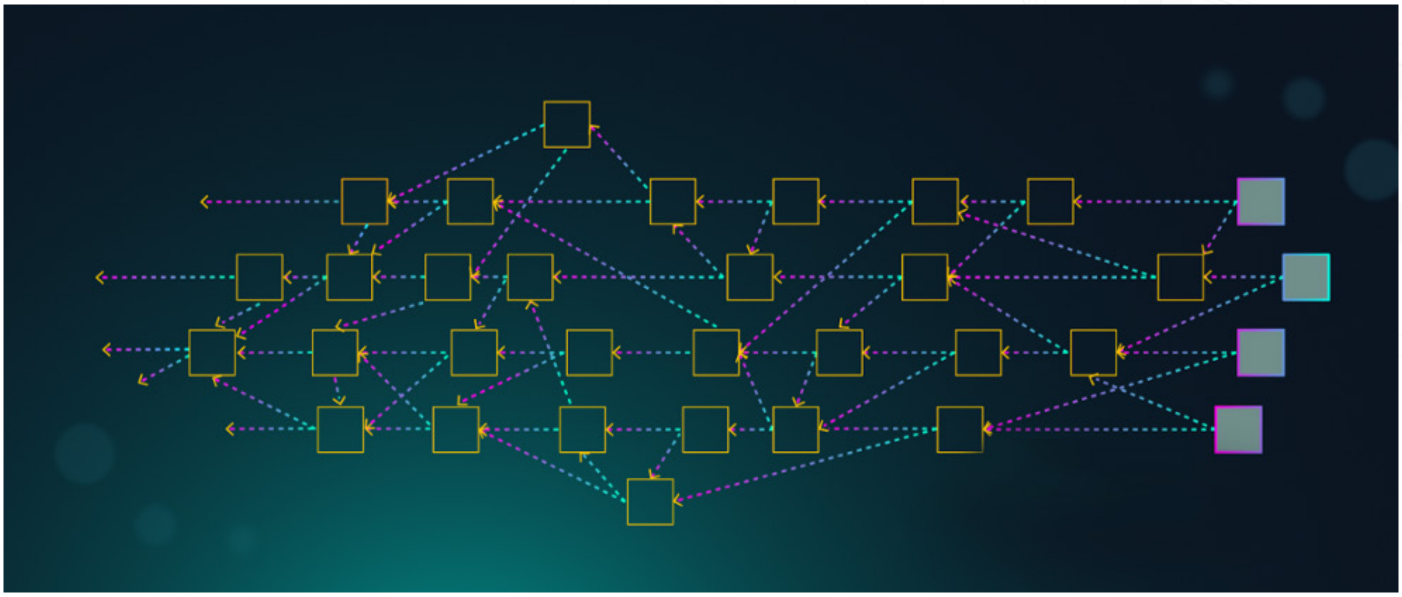
IoT (Internet of Things) has made immense progress from conceptual to deliverable aspect in the last couple of years. We started our digital era with sharing of files only, now it has turned into a stage where we can share anything as a digital product. From wearable gadgets to vehicles to home appliances, the objects that are connected to the internet are increasing exponentially. In the beginning, we were unable to share these physical entities through the internet now we are in the 4th industrial revolution where we can transfer anything in the world through internet.We know IoT is the network of physical devices like gadgets and home appliances, vehicles etc., But what is IOTA?

In simple words, IOTA is blockchain technology which enables the digital transaction of IoT products across the network. The Blockchain technology is known as the future internet where IOTA (Internet of Things tAngle) is known as future Blockchain. Yes!!! These things are making the real technological

revolution in the industry.

IOTA- a 'Block' less Blockchain

IOTA provides a Blockchain for the Internet of Things. But the interesting thing is that there are no blocks in IOTA. Then how Blockchain is created? Then answer lays in another concept known as DAG (Directed Acyclic Graph) which is a directed graph without cycles. In IOTA there is no ledger as in normal blockchains instead they use a DAG called Tangle for the transaction management.



In Tangle, the vertices of the graph represent the nodes/physical devices and the directed edges represent the transaction from one device to another. In short, IOTA network is a lightweight tangle which is scalable to any extent for adding any number of transactions and DAG is the backbone of it.

No Transaction Fees?

In most Blockchain networks the transaction cost is often a matter of concern. But IOTA is not charging any fee for the transaction. The IOTA is mainly designed to perform Nano transactions and these Nano transactions will be executed without any transaction fee. The IOTA can create both private and permissioned networks of IoT, and can manage the transactions with Tangle.

No miners?

In case of a Blockchain mining is a vital element, but IOTA is an exception to it. There is no miners or mining process in IOTA network. So how the transactions are verified in IOTA? The tangle just needs a verification only. And the verification is done by the node who generated the transaction, with the help of a validation algorithm. But the node will be able to proceed this transaction only after verifying two other random transactions in the network using the same validation algorithm.Instead of the Bitcoin protocol, IOTA uses the GHOST(Greedy Heaviest Observed Subtree) protocol, which is a modified version of bitcoin protocol itself. GHOST just modify the bitcoin protocol by creating a tree instead of a blockchain.

Weighted graphs

In IOTA, the priority of the transaction is measured with the help of 'weight' associated with each transaction. DAG is a weighted graph and the weight of each transaction is proportional to the amount of work that the issuing node invested into it. And obviously, the transactions with higher weight will get higher priority than the transactions with lower weight

Developing IOTA

The supporting languages for IOTA are Python and JavaScript. The library packages for the languages are available on the IOTA website itself. The developer may use an IOTA sandbox environment or install IOTA core client for developing the IOTA network. In IOTA network, every object is considered as a service. Or in other words, the physical existence of a 'thing' will be converted as a 'service' in the tangle. This conversion is done with the help of IoT sensors. An IoT sensor is nothing but a simple hardware device attached to the physical entity, which will detect and digitize all the movements of that particular entity. The digitized data will be used in our IOTA network for the management of these entities. All machine to machine communication are controlled using this IoT sensors in IOTA network.

Security

IOTA offers a high level of security for both transactions and assets. The data transfer through the tangle will be in encrypted form and fully protected from external attacks. IOTA uses the masked massaging technique ensure the security of data transfer. In Masked messaging service, the data is encrypted with quantum proof security which makes the data broadcasting also easy. Starting from the weight calculation to restricting an external attack, IOTA employs several mathematical equations which are capable of detecting any small changes in the graph. This highly mathematical approach ensures the protection of data from any kind of external attacks.

The combined advantage of blockchain and IoT has already brought many application areas to IOTA. As the IoT and blockchain is expanding rapidly, more existing services may come under this technology in near future.

Corda

Corda is a distributed ledger platform specially designed for the financial sector. It is an open source platform that can be used to build apps for financial institutions on top of it. It is a permissioned private network designed to record, manage and synchronize contracts and other shared data between partners. Corda is governed by R3 consortium which is a collaboration of 70+ finance institutions. According to R3, Corda is a distributed ledger technology and isn't a blockchain. In fact, R3 provides a platform for developing and deploying distributed apps for different financial use cases. The distributed apps created with Corda is known as CorDapps. DemoBench is a standalone desktop application provided by Corda to configure and launch local Corda nodes. It is a useful tool for training sessions and development of CorDapps.

Corda has many similarities as well as differences with many existing blockchain/distributed ledger technologies. Corda allows the creation of immutable records for financial events. But unlike other blockchains, the transactions are done privately in Corda. Corda smart contracts can be written in Java or any other JVM language like kotlin (a java derived language).

And most importantly, Corda is not tied to any particular consensus algorithm and it doesn't have its own cryptocurrency. It uses the "Notary" infrastructure for 'sequencing of transactions' and validating the transactions. And it does not broadcast a transaction globally for validation purpose. A Corda network may have multiple 'Notaries' and they validate the transactions using different algorithms. The ultimate objective of Corda is to remove costly friction in business transactions by avoiding businesses intermediaries. Since it is only focusing on finance domain, its architecture is simple than that of Ethereum or Fabric. This approach gives performance and security advantage for Corda over other enterprise-level blockchain frameworks. Just like many other distributed technologies, Corda is also in its infant stage and it is hard to make a conclusion on its prospects.

Elements Project

The Elements Project is a protocol level technology which is used to extend the functionalities of bitcoin blockchain. It is an open source community driven project which intends to create new extensions to the Bitcoin and build more bitcoin-based applications.Elements project uses sidechain technology to easily integrate the new application to bitcoin blockchain. Side chains are the separate blockchains having all the components of a normal blockchain including the smart-contract. It exists along with the main blockchain with a back and forth transaction compatibility.

Some of the deployed Elements.

The community members has developed and deployed many Elements already and some of them are on the way of deployment. The well-known side chains of bitcoin including Alpha, Gem are the product of elements project. Some other promising elements are listed here,

1. Asset Issuance

Physical assets are mostly being converted to digital assets nowadays. Since asset has high value than any other documents it needs more protection and security. Asset Issuance helps to issue assets asset including deposits vouchers, shares, currencies, deposits, coupons bonds, etc. with better security.

2. Confidential Transactions

It is a highly sought extension of the bitcoin blockchain. Confidential Transactions hide the amount of bitcoin transferred between two parties from a third party.

3. Segregated Witness

Using Segregated Witness we can reduce the space that used by every transaction in the block. The application separates the properties of a transaction that reflects on the ledger from the data that needed for the validation. This reduces the memory required for a block to store a transaction detail.

4. Relative Lock Time

It allows setting a timeout for the transactions. You can set a particular time period to complete the transaction if it is not completed within the time the transaction will be canceled. It can augment the transaction security of bitcoin blockchain.

5. Schnorr Signature Validation

This extension provides a new way for making signatures for validation and providing new methodologies for multi-signatures

6. New Opcodes

It introduces new opcodes to bitcoin includes DETERMINISTICRANDOM and CHECKSIGFORMSTACK. Furthermore, it re-enabled several scripts in side-chain that are previously available in Bitcoin blockchain.

7. Signature Covers Value

The value can be used for validating the transaction fast. Since the signature on a transaction will be invalidated if the inputs have been spent. This will help you to validate the transaction very easily.

8. Deterministic Pegs

This extension offers a cross-chain facility which simplifies the token transfer between two blockchains.

9. Signed Blocks

The Signed Block is a useful extension to bitcoin. This will allow the user to sign the blocks cryptographically. It is helpful to verify the creator of a block in future.

Other than these listed Elements there are many other extensions available under elements project. More elements are being created by community participant. The project has helped the bitcoin blockchain to work with more customized features and augmented its application domains.

Chain

Chain Core

Chain Core is a blockchain management software developed by Chain Inc. in 2014. The software is designed to manage the permissioned blockchain networks. The chain core can manage any number of independent blockchains or it acts as a blockchain client for different permissioned blockchains. Chain core keeps the copy of the ledgers of multiple blockchains and updates these ledgers during the validation of transactions. The validation and consistency in Chain core are ensured by a Federation of block signers. Here any digital assets including digital currencies, securities, bonds etc. are issued in a common format and represented using any units of value guaranteed by the trusted issuer

There are two editions of Chain core available. A Free Open Source Developer edition and an Enterprise Edition. The Developer edition can be used to test and make prototypes. The Enterprise Edition is essential to develop and deploy the original product based on this prototype.

The leading financial service firms like Visa, Citi group etc. are working with Chain core to develop their blockchain infrastructure.

There are basically three operations available in the software.

1. Create a blockchain.

This option is for creating a new blockchain. The chain core act as a block generator as well as a block signer in the created network. The core provides a Url and a blockchain id for the created network. The id and Url are useful when another core is going to join this network.

2. Connect to an existing Blockchain network

This option enables a core to connect to an existing network. A user must have a blockchain url, a blockchain id, and an active access token for managing the transactions and digital assets.

3. Connect to the test blockchain network

This option is basically oriented to beginners. They may join the blockchain network of chain core and test the blockchain network by making basic operations like account creation, transaction, digital asset management etc.

Development & Security

The chain core application can be developed with Java, node.js, or Ruby. The respective packages and APIs are available in respective repositories. Chain core uses HSM (Hardware Security Module) for a production environment. Compared to other platforms this approach provides a better security standard for the digital assets.

Chain core uses private & public key pairs for the locking and unlocking of assets. Assets are always loaded with a control program. The transactions are verified by running these control programs along the data (public key). If it produces a valid result then the transaction is declared as valid. Using multiple keys for transactions will improve the level of security.

Ivy and Ivy Playground

Ivy is the high-level programming language developed by Chain for creating smart contracts in Chain core. The Ivy playground is an additional tool to create, compile and load the smart contract that can be run along the core.

As the number of blockchains is being created for different purposes, the importance of a tool like the core is evident. The security features like HSM and simplicity in blockchain management makes core an appropriate option for blockchain management

CoCo Framework

Coco (Confidential Consortium) is an open source blockchain framework designed by Microsoft. Microsoft announced the 'Coco' in August 2017 in their whitepaper 'Coco Framework Technical Overview'. The source code of the Coco framework is planned to publish in Github by 2018. Coco is not just a standalone blockchain protocol like Bitcoin or Ethereum rather it provides a platform for building trusted networks using any of the existing protocols. Of course, Coco is designed to be compatible with any existing blockchain protocols such as Ethereum.

Specialties of CoCo

In their whitepaper, Microsoft points out some of the problems with existing systems and how Coco solve the issues.The main drawbacks they pointout of existing systems are

Low transaction throughput

The average processing rate of the public Ethereum network is only 20 transactions per second, which is far behind to meet the requirement in an enterprise environment. Other blockchain networks also fail to meet the enterprise level transaction rate.

High latency

The average latency of a public Ethereum network is about 10-20 seconds and it is 10-15 minutes in bitcoin network. Such high latency will create a bottleneck effect in a business environment.

Lack of confidentiality

In a public blockchain networks, everyone is allowed to see every transaction. This is definitely not a welcome thing in the business environment where the competitors may also be the part of the network.

Lack of effective governance

Public blockchain networks are often self-governed or collectively governed by the users. The model is not suitable for many environments, especially for business level networks.

Low computational efficiency

As the network grows, the computational power required for mining also grows. Thus the energy required is very huge. The annual energy consumption of the bitcoin network is about 15 TWh !!!.

Many attempts were made to overcome these issues and new blockchain platforms like Fabric, Corda etc. also came into existence. But some of these are designed only to meet the requirements of a particular business domain. Some others provided an enterprise level control and security by employing complex algorithms but compromises on performance.Furthermore, whenever a new protocol is introduced to accommodate a feature, the user has to leave behind the technology that he expertise. And it will take some time to understand and work with the new system.

Benefits of CoCo

According to Microsoft, the COCO framework eliminates most of the drawbacks of the existing systems and it offer

- Acceptable throughput and latency for meeting enterprise needs
- Richer, flexible yet simpler confidentiality models
- Network policy management and distributed governance
- Facilitate non-deterministic transactions
- Reduced energy consumption

Coco achieves these performance indices through the use of Trusted Execution Environments (TEEs) like Intel's SGX or Windows Virtual Secure Mode (VSM).

This approach enables Coco to create a trusted networks of nodes and the distributed ledgers are run top of these.

The introduction of Microsoft coco framework is expected to make a big leap. As said earlier, the Coco is not a standalone blockchain protocol. Actually, it provides a foundation for building blockchain networks on top of it. Thus with Coco, it is possible to develop blockchains in any protocol and can integrate different blockchain technologies into a single project to satisfy different enterprise needs. And coco provides many additional features to ease and enhance the development process. In conclusion, from the information available so far, Coco has the potential to be the cradle of blockchain based enterprise applications

Tierion

The importance and vicinity of Blockchain are increasing on daily basis. More existing platforms and services are shifting towards the Blockchain technology by perceiving the advancement it makes. Consequently, different tools and associated services are also emerging in the background. Tierion is such an associated platform which can be used to create a verifiable database of any data on Blockchain. Or it is a Proof engine for data verification. Developers use Tierion to check integrity and timestamp of data or file or any process. The platform offers API and Developer tools to anchor data into a distributed ledger.

The capabilities of Tierion can be utilized by financial institutions, Insurance firms, etc. for safeguarding their critical data from unauthorized modifications. With Tierion, they can track each and every modification being made to the property titles, contracts, digital assets etc. Chain point, an open source protocol and distributed service developed to anchor data into the Bitcoin and Ethereum Blockchain, is the backbone of Tierion. The company is presently working with the Blockchain development projects of Philips and Microsoft to expand the application of it to ore areas.

Features of Tierion

Following are some of the features of Tierion which makes it an advanced tool for data verification.

Digital Receipts: The digital receipts issued by Tierion is a timestamp proof of a transaction took place.

Audit Trail: Tierion generates audit trials for data which are cryptographically verifiable. The trial will track a data from the origin onwards.

Immutable Records: Properly tracked data guarantees the immutable record keeping.

Secure Customer Data: They create verifiable customer data and reduce KYC and compliance cost.

Hash API: With Tierion 'Hash API' developers can anchor records with minimum cost.

Data Collection: Tierion is also used collect data from the web and mobile applications.

Integrate with other Apps: Zapier helps Tierion to integrate with other apps such as Gmail, Twitter, SalesForce etc.

Chainpoint

It contains all the information needed to verify the data without intermediaries. Chainpoint is the main component of Tierion which creates the timestamp proof of a Blockchain transaction. The initial version of Chainpoint was introduced in June 2015, and later versions Chainpoint 2.0 and 3.0 released in August 2016 and August 17 respectively. The ultimate proof from a Chainpoint or a 'chainproof' is a trail of operation cryptographically linking your data to one or more Blockchain Chainpoint Proof Creation Steps

Following steps are involved in Chainpoint proof creation.

- The user submits the hashed data to a Chainpoint.
- Chainpoint returns a hash_id (UUID) with a timestamp to the user.
- Chainpoint combine the submitted hash with UUID to obtain a new hash.
- The same hash is combined with a 'NIST Beacon' and a new hash is

created.
(This will make sure that the chain proof is created after the generation of hash_id)

- The new hash is sent to aggregation service.
- Aggregation service aggregates the hashes into Merkle trees.
- Then the Merkle root of the Merkle tree is sent to the Chainpoint calendar. (Various Chainpoint servers are kept in agreement to create a Chainpoint calendar. In fact, Chainpoint calendar is a Blockchain.)
- Calendar data is organized as blocks, and they are stored in a normal database called CockroachDB.Calendar blocks are then anchored to Bitcoin or Ethereum Blockchain.
- Now the Chainpoint starts to monitors the Blockchain. On each anchoring, if the transaction receives an adequate number of ‘Validation’, validated blocks are added to the calendar.
- Each validated blocks contains data to create the final Chainpoint proof.
- To finalize the proof, Chainpoint appends the partial proof with final data. And the final proof is created.

Benefits of Tierion

The major benefit of Tierion is that it eliminates the role of the third party in data verification. Anyone with the proof issued by Tierion can verify the entire transaction path of a data. It has a highly scalable architecture and better performance standard. The time stamp accuracy is achieved with Network Time Protocol (NTP) and National Institute of Standard and Technology (NIST) server. Immediate anchoring is another feature, Chainpoint anchors the data whenever a new hash is submitted to Chainpoint service. And indeed it is a cost-effective solution.

BigchainDB

The BigchainDB is a scalable distributed database which can be used for the blockchain technology. In a normal case, blockchain itself is the database. As in the case of bitcoin and many blockchain applications the blocks is providing the storage facility too. There are no additional databases. But the BigchainDB provides an alternative to this method. The BigchainDB will work as a distributed database with all characteristics of a blockchain

The BigchainDB was first introduced as a distributed database and later the characteristic of blockchain technology has added to it. Now BigchainDB has the features of both traditional blockchain (like bitcoin) and the distributed database and it supports both private and public networks. BigchainDB is a NoSQL(Non-SQL) database which provides a storage mechanism and data retrieval models other than the tabular relations used in relational databases.

The commonly used NoSQL types are Key-value stores, Document database, Wide column stores and Graph stores. Each of these NoSQL databases adopts different methods of data storage. The developer can select any of the above models according to the requirement and use case.

Why BigchainDB?

Normal blockchain networks like bitcoin suffer from several problems like low throughput, high latency, low storage capacity etc. In a bitcoin network, the latency before a single confirmed write is about 10 minutes and throughput is only a few transactions per second. The storage capacity is also not promising as it is still pegging at a few dozen GB. But in BigchainDB the throughput is about 1 million writes per second and latency is also significantly lower. The storage capacity of BigchainDB is that of a distributed database. Which means the capacity will increase as the number of nodes increases.

The BigchainDB has the following features.

- Decentralized control:-No central server for managing the database
- Immutability:-Once a change is made to the database it is immutable.
- Creation & Movement of Digital assets:- Digital assets can be created or manage the BigchainDB

BigchainDB vs Normal Blockchain and Distributed Database

	Traditional Blockchain	Distributed Database	BigchainDB
Throughput	Low (few transaction per second)	High (increase with nodes)	High (increase with nodes)
Latency	High (10 min)	Low	Low
Storage capacity	Low	High (increase with nodes)	High (increase with nodes)
Query capability	No	Yes	Yes
Rich permissioning	No	Yes	Yes
Decentralized Control	Yes	No	Yes
Immutability	Yes	No	Yes
Creation & Movement of Digital assets	Yes	No	Yes
Event Chain Structure	Merkle tree	-	Hash Chain

Models in BigchainDB

Three models namely Transaction model, Block model, and Vote model are the backbone of BigchainDB. These models give it the advantages of Blockchain as well as the normal database.

Transaction Models in BigchainDB

The basic component of BigchainDB is transactions. Every data stored in it will the details of the individual transaction. Two types of transaction models are used in BigchainDB

1) Creation Transaction
2) Transfer Transaction

The “Creation Transaction” is used to initialize the details of an asset in the blockchain and the “Transfer Transaction” is used to transfer ownership of the asset. A transaction in a JSON document will have the following structure
Id: Is the primary key. It will be the hash value of that particular transaction,
Version: It is the version number of that transaction model,
Fulfillments: Each fulfillment is a pointer to the unspent assets. It will point to the ownership of an asset,
Conditions: List of conditions that should be fulfilled by the transfer transactions,
Operation: String representation of the operation to be performed,
Timestamp: Transaction creation time in UTC. Provided by the user,
Hash: It is the hash value of the serialized payload,
Payload: It can be any JSON document. For a transfer transaction, it will be empty. All the transactions in the BigchainDB will be stored in the above mentioned structure only.

Block Models in BigchainDB

The blocks are also represented as JSON documents in the following structure,

```
{
“id”: “<hash of block >”,
“block”: {
“timestamp”: “<block -creation
timestamp >”,
“transactions”: [“<list of
transactions >”],
“node_pubkey”: “<public key of
the node creating the block >”,
“voters”: [“<list of federation
nodes public keys >”]
},
“signature”: “<signature of block >”,
“votes”: [“<list of votes >”]
}
```

Id: The primary key. It is the hash of the serialized block,

Timestamp: It is the time of creation of a block. It is given by the created the node,

Transactions: The list of transactions included in the block,

Node-pub key: The public key of the node, that created the block,

Voters: It is the list of public key of federation nodes existed in the system when the node is created,

Signature: Signature of the block by the node who created the block, It is generated by serializing the block data and using the private key,

Votes: list of votes given by the voters.

A vote has the following structure:

```
{
“node_pubkey”: “<the public key of the voting node
>”,
“vote”: {
“voting_for_block”: “<id of the
block the node is voting for >”,

    “previous_block”: “<id of the
block previous to this one >”,
“is_block_valid”: “<true|false
>”,
“invalid_reason”:
```

```
"<None|DOUBLE_SPEND|TRANSACTIONS_HASH_MISMATCH|
NODES_PUBKEYS_MISMATCH",
"timestamp": "<timestamp of the
voting action >"
},
"signature": "<signature of vote >"
}
```

Node_pubkey: It is the public key of the voting node.

Voting_for_block: Id of the block for which a node is voting

Previous_block: Id of the previous block

Is_block_valid: Vote for the block it can be true or false. I.e. positive or negative vote

Invalid_reason: Reason for invalidating or voting 'false'.

Timestamp: Time at which voting action takes place.

Signature: Signature for the vote.

Among many other blockchain related technologies BigchainDB is a unique one as it changes the very data storage mechanism of the blockchain. From the initial assessment, it is a promising technology, especially for handling the huge amount of data. It has the potential to leverage the blockchain technology in the domains like Big data analysis AI etc.