IQONIQ

GROUP

IQONIQ BLOCKCHAIN TECHNICAL PAPER

IQONIQ

BECOME ONE

Scope

The following scope of work is for the Infrastructure setup- Core and Horizon Server Setup, Integration of API, a creation of digital asset, Client App Integration. The implementation phase will be carried out only for the defined activities as mentioned in this document.

The set of activities which clearly defines the scope are:

- Infrastructure Setup
- Set Up Blockchain Network In Lines With IQONIQ Blockchain
- Create Native Digital Asset Of The Blockchain Network
- Set Up Blockchain Network Node, Create Root Account
- Setting Up API Layer For Blockchain Network
- Decide Initial Number Of IQQ Coins
- Flush Root Account With Initial Number Of Coins
- Creating Network Performance Dashboard
- Creating Asset Performance Dashboard
- Creating Blockchain Explorer

IQONIQ Blockchain Overview

The challenges in the current market scenario, especially in disruptive technology, can be addressed with IQONIQ Blockchain, the Blockchain technology platform on which IQONIQ Remit, the remittance product operates. There are other products as well which works on IQONIQ Remit platform- IQONIQ Commerce, IQONIQ Market Maker, and IQQ. The Blockchain network is trusted by Banks, currency exchanges, corporate houses, Fintechs, Global Merchants and payment networks.

Introspecting into deeper aspects of IQONIQ Blockchain, the platform works on a decentralized network. A decentralized network consists of peers that can run independently of each other. The power to transmit information is distributed among a network of servers, instead of being driven from one primary source. This means that the IQONIQ Blockchain is independent of multiple entities and work on a single entity. The idea is to have as many independent servers participate in the network as possible so that the network will still run successfully even if some servers fail. The ledger within IQONIQ Blockchain records lists of all the balances and transaction in a similar way to that of the traditional ledger. A complete copy of the individual ledger is hosted on each server that runs IQONIQ Blockchain. Any entity can run the IQONIQ Blockchain server. The servers all together form a decentralized network, allowing the ledgers to be distributed as much as possible. The server's sync and validate the ledger by consensus mechanism. The servers communicate and sync with each other to ensure that transactions are valid and get applied successfully to the global ledger.

This entire process of coming to a consensus on this network occurs approximately every 3-5 seconds, which is a real-time settlement of the assets. The real-time settlement occurs with any of the assets present on the Blockchain network. The assets can be the IQONIQ Blockchain Native asset-IQQ, Fiat Currencies, USD, EUR, Cryptocurrencies like BTC, ETH etc. and Central Bank issued cryptocurrencies. The Anchors play an important role in IQONIQ Blockchain. Anchors are simply entities that people trust to hold their deposits and issue credits into the IQONIQ Blockchain for those deposits. They form a bridge between different currencies and the IQONIQ Blockchain. All money transactions in this network occur in the form of credit issued by anchors.

Anchors do two things:

- They take the deposit and issue the corresponding credit to the individual’s account address on the IQONIQ Blockchain ledger.
- One can make a withdrawal by bringing them credit they issued

One has to trust the anchor to honor their deposits and withdrawals of credit it has issued. Anchors exist in the traditional payment system. For example, to use a wallet, you deposit money in from your bank account, prefunding. The wallet then gives you credit the wallet. You can now send that wallet credit to anyone that trusts the wallet, anyone who trusts the wallet. Someone that received your wallet credit can convert it to fiat money using the wallet by withdrawing it to the bank. Anchors portrays almost the same functionality there. The difference is, all the wallets and other anchors are operating on the same network so they can all transact with each other now – this makes the system way more powerful. People can now easily send and exchange all these different anchor credits with each other.

The IQONIQ Blockchain is flexible in terms of operability. It is interoperable with other networks as well. IQONIQ Blockchain interoperates with Blockchain networks and domestic payment networks. It works with other platforms as well, for instance, Ripple, Stellar, Corda, Sawtooth, Fabric etc. The IQONIQ Blockchain ledger is able to store offers that people have made to buy or sell currencies. Offers are public commitments to exchange one type of credit for another at a predetermined rate. The ledger becomes a global marketplace for offers. These offers are defined to what is known as order book. There is an order book for each currency/issuer pair. For instance, if you are wanting to exchange Commerz Bank-EUR for Bitstamp-BTC you should look at the particular order book in the ledger to see what people are buying and selling it for. This allows people to not only buy and sell currencies in a way as the authorized dealers work but also to convert currencies seamlessly during transactions. This network also allows you to send any currency you hold to anyone else in a different currency through the built-in distributed exchange. People can receive any currency through an anchor they added. Here are a few possible ways the transaction can happen:

- The network finds an offer on the internal USD/AED exchange for someone wanting to buy AED for USD and automatically makes the exchange between the two parties.
- Using IQQ as an intermediary currency, IQONIQ Blockchain will look for offers on the network asking for USD in exchange for IQQ (the native, purely digital currency). It will simultaneously look for an offer asking for IQQ in exchange for AED. The network makes those exchanges and sends beneficiary the credit.

If there is no explicit relationship between offers to buy and sell, IQONIQ Blockchain tries to find offers from the network that will lead a chain of conversions from AED to USD. For example, AED to AUD, AUD to BTC, BTC to XLM, XLM to USD

Summary

The IQONIQ Blockchain is the decentralized network which facilitates the transaction on a realtime basis with a visibility on the documentation on a real-time basis. The distributed ledger technology makes the documents sharing more transparent and secured. The transactions which involved a lot of trusted parties and documents can be
transacted Blockchain technology.

Technical Specifications

FBA, a crucial part of IQONIQ Blockchain is the first provably safe consensus mechanism to enjoy four key properties simultaneously:

Decentralized control

Anyone is able to participate and no central authority dictates whose approval is required for consensus.

Low latency

In practice, nodes can reach consensus at timescales humans expect for web or payment transactions—i.e., a few seconds at most.

Flexible trust

Users have the freedom to trust any combination of parties they see fit. For example, a small non-profit may play a key role in keeping much larger institutions honest.

Asymptotic security

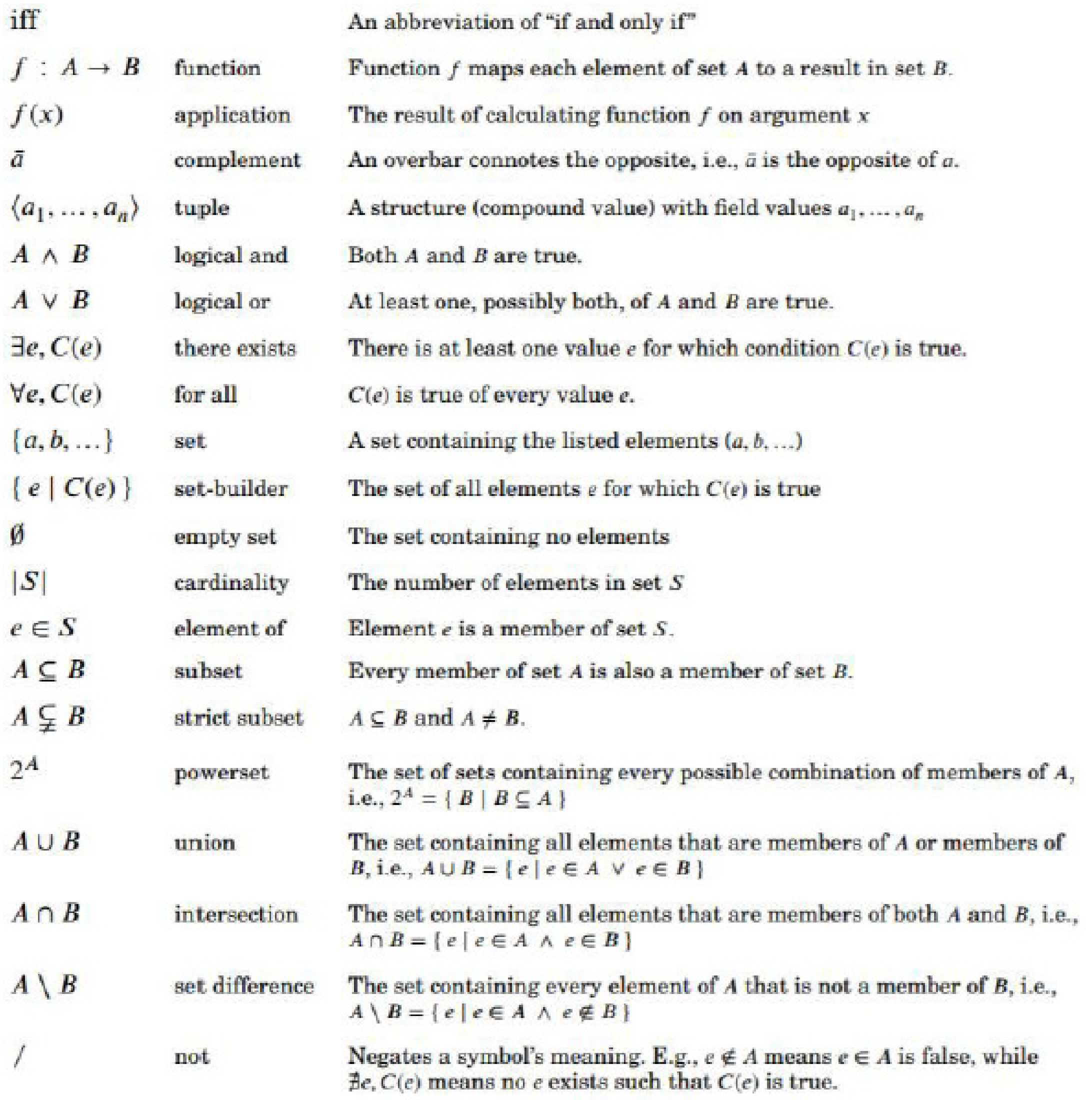
Safety rests on digital signatures and hash families whose parameters can realistically be tuned to protect against adversaries with unimaginably vast computing power.

Low latency

In practice, nodes can reach consensus at timescales humans expect for web or payment transactions—i.e., a few seconds at most.

Federated Byzantine Agreement Systems

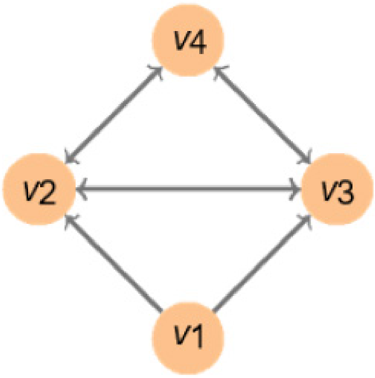
Like non- federated Byzantine agreement, FBA addresses the problem of updating replicated state, such as a transaction ledger or certificate tree. By agreeing on what updates to apply, nodes avoid contradictory, irreconcilable states. We identify each update by a unique slot from which inter-update dependencies can be inferred. For instance, slots may be consecutively numbered positions in a sequentially applied log. A mandatory to mention here, is the glossary of notations that one might need to go through, before diving into the realm of FBA. The picture below must be referred to:



iff		An abbreviation of "if and only if"
$f : A \to B$	function	Function f maps each element of set A to a result in set B.
$f(x)$	application	The result of calculating function f on argument x
$\bar{a}$	complement	An overbar connotes the opposite, i.e., $\bar{a}$ is the opposite of a.
$\langle a_1, \ldots, a_n \rangle$	tuple	A structure (compound value) with field values $a_1, \ldots, a_n$
$A \wedge B$	logical and	Both A and B are true.
$A \vee B$	logical or	At least one, possibly both, of A and B are true.
$\exists e, C(e)$	there exists	There is at least one value e for which condition $C(e)$ is true.
$\forall e, C(e)$	for all	$C(e)$ is true of every value e.
$\{a, b, \ldots\}$	set	A set containing the listed elements $(a, b, \ldots)$
$\{ e \mid C(e) \}$	set-builder	The set of all elements e for which $C(e)$ is true
$\emptyset$	empty set	The set containing no elements
$\lvert S \rvert$	cardinality	The number of elements in set S
$e \in S$	element of	Element e is a member of set S.
$A \subseteq B$	subset	Every member of set A is also a member of set B.
$A \subsetneq B$	strict subset	$A \subseteq B$ and $A \neq B$.
2^A	powerset	The set of sets containing every possible combination of members of A, i.e., $2^A = \{ B \mid B \subseteq A \}$
$A \cup B$	union	The set containing all elements that are members of A or members of B, i.e., $A \cup B = \{ e \mid e \in A \vee e \in B \}$
$A \cap B$	intersection	The set containing all elements that are members of both A and B, i.e., $A \cap B = \{ e \mid e \in A \wedge e \in B \}$
$A \setminus B$	set difference	The set containing every element of A that is not a member of B, i.e., $A \setminus B = \{ e \mid e \in A \wedge e \notin B \}$
/	not	Negates a symbol's meaning. E.g., $e \notin A$ means $e \in A$ is false, while $\nexists e, C(e)$ means no e exists such that $C(e)$ is true.

An FBA system runs a consensus protocol that ensures nodes agree on slot contents. A node v can safely apply update x in slot i when it has safely applied updates in all slots upon which i depends and, additionally, it believes all correctly functioning nodes will eventually agree on x for slot i. At this point, we say v has externalized x for slot i. The outside world may react to externalized values in irreversible ways, so a node cannot later change its mind about them.

A challenge for FBA is that malicious parties can join many times and outnumber honest nodes. Hence, traditional majority-based quorums do not work. Instead, FBA determines quorums in a decentralized way, by each node selecting what we call quo- rum slices. The next subsection defines quorums based on slices. The following subsection provides some examples and discussion. Finally, we define the key properties of safety and liveness that a consensus protocol should hope to achieve.



Q.v_1/ = ^^v_1; v_2; v_3“
Q.v_2/ = **Q**.v_3/ = **Q**.v_4/ =
^^v_2; v_3; v_4“

Fig. 2. v_1’s quorum slice is not a quorum without v_4.

Quorum slices

In a consensus protocol, nodes exchange messages asserting statements about slots. We assume such assertions cannot be forged, which can be guaranteed if nodes are named by public key and they digitally sign messages. When a node hears a sufficient set of nodes assert a statement, it assumes no functioning node will ever contradict that statement. We call such a sufficient set a quorum slice, or, more concisely, just a slice. To permit progress in the face of node failures, a node may have multiple slices, any one of which is sufficient to convince it of a statement. At a high level, then, an FBA system consists of a loose confederation of nodes each of which has chosen one or more slices. More formally:

Definition (FBAS)

A federated Byzantine agreementsystem, or FBAS , is a pair (V, Q) comprising a set of nodes V and a quorum function QV⊆22v \{0} specifying one or more quorum slices for each node, where a node belongs to all of its own quorum slices—i.e., Vv E V, Vq E Q(v), v E q. (Note 2X denotes the power set of X.)

Definition (quorum)

A set of nodes U ⊆ V in FBAS (V, Q) is a quorum iff U ≠ 0 and U contains a slice for each member—i.e., Vv E U, 3q E Q(v) such that q ⊆ U .

A quorum is a set of nodes sufficient to reach agreement. A quorum slice is the subset of a quorum convincing one particular node of agreement. A quorum slice may be smaller than a quorum. Consider the four-node system in Figure 2, where each node has a single slice and arrows point to the other members of that slice. Node v1's slice {v1, v2, v3} is sufficient to convince v1 of a statement. But v2's and v3's slices include v4, meaning neither v2 nor v3 can assert a statement without v4's agreement. Hence, no agreement is possible without v4's participation, and the only quorum including v1 is the set of all nodes {v1, v2, v3, v4}.

Traditional, non-federated Byzantine agreement requires all nodes to accept the same slices, meaning Vv1, v2, Q(v1) = Q(v2). Because every member accepts every slice, traditional systems do not distinguish between slices and quorums. The downside is that membership and quorums must somehow be pre-ordained, precluding open membership and decentralized control. A traditional system, such as PBFT, typically has 3f + 1 nodes, any 2f + 1 of which constitute a quorum. Here f is the maximum number of Byzantine failures—meaning nodes acting arbitrarily— the system can survive .

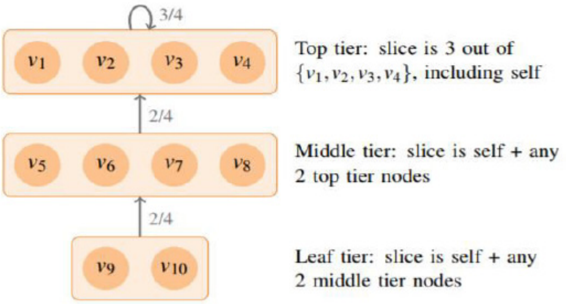

3/4
v_1
v_2
v_3
v_4
Top tier: slice is 3 out of $\{v_1, v_2, v_3, v_4\}$, including self
2/4
v_5
v_6
v_7
v_8
Middle tier: slice is self + any 2 top tier nodes
2/4
v_9
v_{10}
Leaf tier: slice is self + any 2 middle tier nodes

Figure 3: Tiered quorum structure example

FBA, introduced by this paper, generalizes Byzantine agreement to accommodate a greater range of settings. FBA's key innovation is enabling each node v to chose its own quorum slice set Q(v). System-wide quorums thus arise from individual decisions made by each node. Nodes may select slices based on arbitrary criteria such as reputation or financial arrangements. In some settings, no individual node may have complete knowledge of all nodes in the system, yet consensus should still be possible.

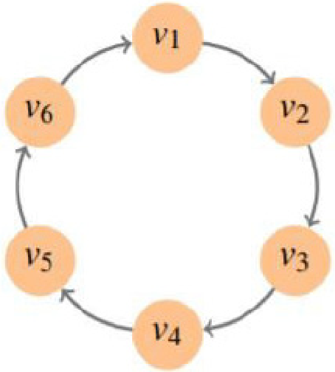
Examples and discussion

Figure 3 shows an example of a tiered system in which different nodes have different slice sets, something possible only with FBA. A top tier, comprising v1, … , v4, is structured like a PBFT system with f = 1, meaning it can tolerate one Byzantine failure so long as the other three nodes are reachable and well-behaved. Nodes v5, … , v8 constitute a middle tier and depend not on each other, but rather on the top tier. Only two top tier nodes are required to form a slice for a middle tier node. (The top tier assumes at most one Byzantine failure, so two top tier nodes cannot both fail unless the whole system has failed.) Nodes v9 and v10 are in a leaf tier for which a slice consists of any two middle tier nodes. Note that v9 and v10 may pick disjoint slices such as {v5, v6} and {v7, v8}; nonetheless, both will indirectly depend on the top tier. In practice, the top tier could consist of anywhere from four to dozens of widely known and trusted financial institutions. As the size of the top tier grows, there may not be exact agreement on its membership, but there will be significant overlap be- tween most parties' notions of top tier. Additionally, one can imagine multiple middle tiers, for instance one for each country or geographic region.

This tiered structure resembles inter-domain network routing. The Internet today is held together by individual peering and transit relationships between pairs of networks. No central authority dictates or arbitrates these arrangements. Yet these pair- wise relationships have sufficed to create a notion of de facto tier one ISPs. Though Internet reachability does suffer from firewalls, transitive reachability is nearly complete—e.g., a firewall might block The New York Times, but if it allows Google, and Google can reach The New York Times, then The New York Times is transitively reachable. Transitive reachability may be of limited utility for web sites, but it is crucial for consensus; the equivalent example would be Google accepting statements only if The New York Times does.

If we think of quorum slices as analogous to network reachability and quorums as analogous to transitive reachability, then the Internet's near complete transitive reach- ability suggests we can likewise ensure worldwide consensus with FBA. In many ways, consensus is an easier problem than inter-domain routing.

While transit consumes re- sources and costs money, slice inclusion merely requires checking digital signatures. Hence, FBA nodes can err on the side of inclusiveness, constructing conservative slices with greater interdependence and redundancy than typically seen in peering and transit arrangements. Another example not possible with centralized consensus is cyclic dependency structures, such as the one depicted in Figure 4. Such a cycle is unlikely to arise intention- ally, but when individual nodes choose their own slices, it is possible for the overall system to end up embedding dependency cycles. The bigger point is that, compared to traditional Byzantine agreement, an FBA protocol must cope with a far wider variety of quorum structures.



$$\mathbf{Q}(v_i) = \{\{v_i, v_{(i \bmod 6)+1}\}\}$$

Figure 4: Cyclic quorum structure example

Safety and liveness

We categorize nodes as either well-behaved or ill-behaved. A well-behaved node chooses sensible quorum slices (discussed further in Section 4.1) and obeys the protocol, including eventually responding to all requests. An ill-behaved node does not. Ill-behaved nodes suffer Byzantine failure, meaning they behave arbitrarily. For instance, an ill- behaved node may be compromised, its owner may have maliciously modified the soft- ware, or it may have crashed. The goal of Byzantine agreement is to ensure that well-behaved nodes externalize the same values despite the presence of such ill-behaved nodes. There are two parts to this goal. First, we would like to prevent nodes from diverging and externalizing different values for the same slot. Second, we would like to ensure nodes can actually externalize values, as opposed to getting blocked in some dead-end state from which consensus is no longer possible. We introduce the following two terms for these properties:

Definition (safety)

A set of nodes in an FBAS enjoy safety if no two of them ever externalize different values for the same slot.

Definition (liveness)

A node in an FBAS enjoys liveness if it can externalize new values without the participation of any failed (including ill-behaved) nodes.

We call well-behaved nodes that enjoy both safety and liveness correct. Nodes that are not correct have failed. All ill-behaved nodes have failed, but a well-behaved node can fail, too, by waiting indefinitely for messages from ill-behaved nodes, or, worse, by having its state poisoned by incorrect messages from ill-behaved nodes.

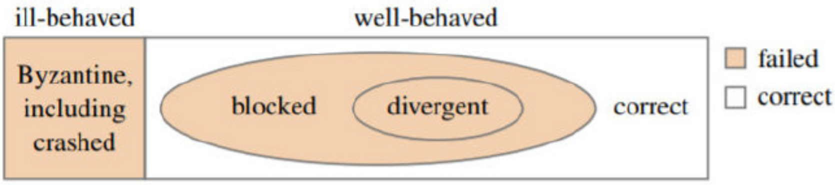


Figure 5: Venn diagram of node failures

Figure 5 illustrates the possible kinds of node failure. To the left are Byzantine failures, meaning the ill-behaved nodes. To the right are two kinds of well-behaved but failed nodes. Nodes that lack liveness are termed blocked, while those that lack safety are termed divergent. An attack violating safety is strictly more powerful than one violating only liveness, so we classify divergent nodes as a subset of blocked ones. Our definition of liveness is weak in that it says a node can externalize new values, not that it will. Hence, it admits a state of perpetual preemption in which consensus remains forever possible, yet the network continually thwarts it by delaying or reordering critical messages in just the wrong way. Perpetual preemption is inevitable in a purely asynchronous, deterministic system that survives node failure. Fortunately, preemption is transient. It does not indicate node failure, because the system can recover at any time. Protocols can mitigate the problem through randomness or through realistic assumptions about message latency. Latency assumptions are more practical when one would like to limit execution time or avoid the trusted dealers often required by more efficient Randomized algorithms. Of course, only termination and not safety should depend upon message timing.

OPTIMAL RESILIENCE

Whether or not nodes enjoy safety and liveness depends on several factors: what quorum slices they have chosen, which nodes are ill-behaved, and of course the concrete consensus protocol and network behavior. As is common for asynchronous systems, we assume the network eventually delivers messages between well-behaved nodes, but can otherwise arbitrarily delay or reorder messages.

This section answers the following question: given a specific (V, Q) and particular subset of V that is ill-behaved, what are the best safety and liveness that any federated Byzantine agreement protocol can guarantee regardless of the network? We first discuss quorum intersection, a property without which safety is impossible to guarantee. We the introduce a notion of dispensable sets—sets of failed nodes in spite of which it is possible to guarantee both safety and liveness.

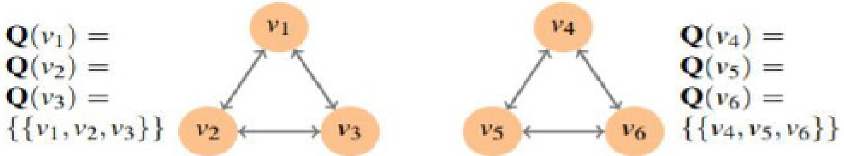


Figure 6: FBAS lacking quorum intersection

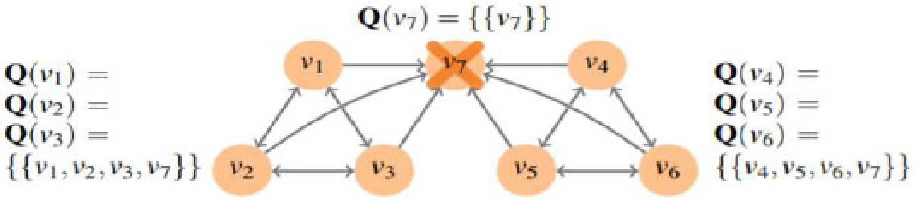


Figure 7: Ill-behaved node v_7 can undermine quorum intersection.

Quorum intersection

A protocol can guarantee agreement only if the quorum slices represented by function Q satisfy a validity property we call quorum intersection. Definition (quorum intersection). An FBAS enjoys quorum intersection iff any two of its quorums share a node—i.e., for all quorums U1 and U2, U1 ∩ U2 ≠ 0.

Figure 6 illustrates a system lacking quorum intersection, where Q permits two quorums, {v1, v2, v3} and {v4, v5, v6}, that do not intersect. Disjoint quorums can independently agree on contradictory statements, undermining system-wide agreement. When many quorums exist, quorum intersection fails if any two do not intersect. For example, the set of all nodes {v1, … , v6} in Figure 6 is a quorum that intersects the other two, but the system still lacks quorum intersection because the other two do not intersect each other.

No protocol can guarantee safety in the absence of quorum intersection, since such a configuration can operate as two different FBAS systems that do not exchange any messages. However, even with quorum intersection, safety may be impossible to guarantee in the presence of ill-behaved nodes. Compare Figure 6, in which there are two disjoint quorums, to Figure 7, in which two quorums intersect at a single node v7, and v7 is illbehaved. If v7 makes inconsistent statements to the left and right quorums, the effect is equivalent to disjoint quorums. In fact, since ill-behaved nodes contribute nothing to safety, no protocol can guarantee safety without the well-behaved nodes enjoying quorum intersection on their own. After all, in a worst-case scenario for safety, ill-behaved nodes can just always make any possible (contradictory) statement that completes a quorum. Two quorums overlapping only at ill-behaved nodes will again be able to operate like two different FBAS systems thanks to the duplicity of the ill-behaved nodes. In short, FBAS (V, Q) can survive Byzantine failure by a set of nodes B ⊆ V iff (V, Q) enjoys quorum intersection after deleting the nodes in B from V and from all slices in Q. More formally: Definition (delete). If (V, Q) is an FBAS and B ⊆ V is a set of nodes, then to delete B from (V, Q), written (V, Q) B means to compute the modified FBAS (V \ B, QB) where QB (v) = { q \ B I q ∈ Q(v) }.

It is the responsibility of each node v to ensure Q(v) does not violate quorum intersection. One way to do so is to pick conservative slices that lead to large quorums. Of course, a malicious v may intentionally pick Q(v) to violate quorum intersection. But a malicious v can also lie about the value of Q(v) or ignore Q(v) to make arbitrary assertions. In short, Q(v)'s value is not meaningful when v is ill-behaved. This is why the necessary property for safety—quorum intersection of well-behaved nodes after deleting ill-behaved nodes—is unaffected by the slices of ill-behaved nodes.

Suppose Figure 6 evolved from a three-node FBAS v1, v2, v3 with quorum intersection to a six-node FBAS without. When v4, v5, v6 join, they maliciously choose slices that violate quorum intersection and no protocol can guarantee safety for V. Fortunately, {v4,v5,v6} deleting the bad nodes to yield (V, Q) restores quorum intersection, meaning at least {v1, v2, v3} can enjoy safety. Note that deletion is conceptual, for the sake of describing optimal safety. A protocol should guarantee safety for v1, v2, v3 without their needing to know that v4, v5, v6 are ill-behaved.

Dispensable sets (D Sets)

We capture the fault tolerance of nodes' slice selections through the notion of a dispensable set or D Set. Informally, the safety and liveness of nodes outside a D Set can be guaranteed regardless of the behavior of nodes inside the D Set. Put another way, in an optimally resilient FBAS, if a single D Set encompasses every ill-behaved node, it also contains every failed node, and conversely all nodes outside the D Set are correct. As an example, in a centralized PBFT system with 3f + 1 nodes and quorum size 2f + 1, any f or fewer nodes constitute a D Set. Since PBFT in fact survives up to f Byzantine failures, its robustness is optimal. In the less regular example of Figure 3, {v1} is a D Set, since one top tier node can fail without affecting the rest of the system. {v9} is also a D Set because no other node depends on v9 for correctness. {v6, … , v10} is a D Set, because neither v5 nor the top tier depend on any of those five nodes . {v5, v6} is not a D Set, as it is a slice for v9 and v10 and hence, if entirely malicious, can lie to v9 and v10 and convince them of assertions inconsistent with each other or the rest of the system.

To prevent a misbehaving D Set from affecting the correctness of other nodes, two properties must hold. For safety, deleting the D Set cannot undermine quorum intersection. For liveness, the D Set cannot deny other nodes a functioning quorum. This leads to the following definition:

Definition (D Set).

Let (V, Q) be an FBAS and B ⊆ V be a set of nodes. We say B is a dispensible set, or D Set, iff:

1. (quorum intersection despite B) (V, Q) B enjoys quorum intersection, and
2. (quorum availability despite B) Either V \ B is a quorum in (V, Q) or B = V

Quorum availability despite B protects against nodes in B refusing to answer re-quests and blocking other nodes' progress. Quorum intersection despite B protects against the opposite—nodes in B making contradictory assertions that enable other nodes to externalize inconsistent values for the same slot. Nodes must balance the two threats in slice selection.

All else equal, bigger slices lead to bigger quorums with greater overlap, meaning fewer failed node sets B will undermine quorum intersection when deleted. On the other hand, bigger slices are more likely to contain failed nodes, endangering quorum availability.

The smallest D Set containing all ill-behaved nodes may encompass well-behaved nodes as well, reflecting the fact that a sufficiently large set of ill-behaved nodes can cause well behaved nodes to fail. For instance, in Figure 3, the smallest D Set contain ing v5 and v6 is {v5, v6, v9, v10}. The set of all nodes, V, is always a D Set, as an FBAS (V, Q) vacuously enjoys quorum intersection despite V and, by special case, also enjoys quorum availability despite V. The motivation for the special case is that given sufficiently many ill-behaved nodes, V may be the smallest D Set to contain all ill-behaved ones, indicating a scenario under which no protocol can guarantee anything better than complete system failure.

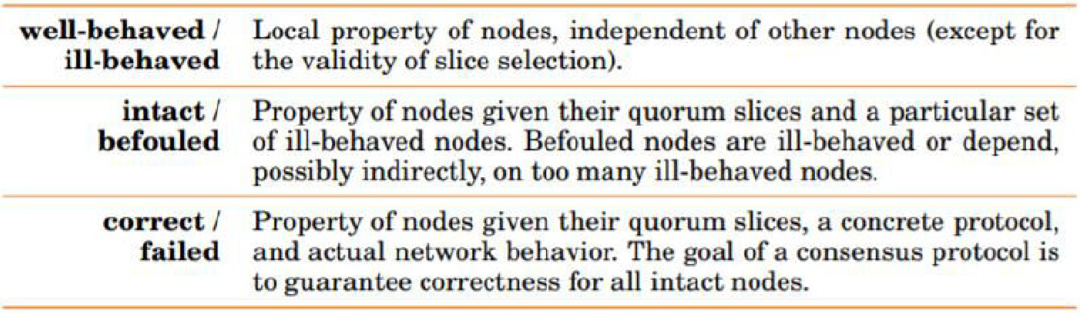
The D Sets in an FBAS are determined a priori by the quorum function Q. Which nodes are well- and ill-behaved depends on runtime behavior, such as machines getting compromised. The D Sets we care about are those that encompass all ill-behaved nodes, as they help us distinguish nodes that should be guaranteed correct from ones for which such a guarantee is impossible. To this end, we introduce the following terms:

Definition (intact).

A node v in an FBAS is intact iff there exists a D Set B containing all illbehaved nodes and such that v ⊆ B.

Definition (befouled).

A node v in an FBAS is befouled iff it is not intact. A befouled node v is surrounded by enough failed nodes to block its progress or poi- son its state, even if v itself is well-behaved. No FBAS can guarantee the correctness of a befouled node. However, an optimal FBAS guarantees that every intact node remains correct. Figure 8 summarizes the key properties of nodes. The following theorems facilitate analysis by showing that the set of befouled nodes is always a D Set in an FBAS with quorum intersection.



well-behaved / ill-behaved	Local property of nodes, independent of other nodes (except for the validity of slice selection).
intact / befouled	Property of nodes given their quorum slices and a particular set of ill-behaved nodes. Befouled nodes are ill-behaved or depend, possibly indirectly, on too many ill-behaved nodes.
correct / failed	Property of nodes given their quorum slices, a concrete protocol, and actual network behavior. The goal of a consensus protocol is to guarantee correctness for all intact nodes.

Fig. 8. Key properties of FBAS nodes

THEOREM 1

Let U be a quorum in FBAS (V, Q), let B ⊆ V be a set of nodes, and let U ‘ = U \ B. If U ‘≠ ∅ then U t is a quorum in (V, Q)B.

PROOF:

Because U is a quorum, every node v ∊ U has a q ∊ Q(v) such that q ⊆ U . Since U ‘ ⊆ U , it follows that every v ∊ U ‘ has a q E Q(v) such that q \ B ⊆ U ‘. Rewriting with deletion notation yields ∀v ∊ U ‘, ∃ q ∊ QB (v) such that q ⊆ U ‘, which, because U ‘ ⊆ V \ B, means that U ‘ is a quorum in (V, Q)B.

THEOREM 2

If B1 and B2 are D Sets in an FBAS (V, Q) enjoying quorum intersection, then B = B1 ∩ B2 is a D Set, too.

PROOF:

Let U1 = V \ B1 and U2 = V \ B2. If U1 = , then B1 = V and B = B2 (a DSet), so we are done. Similarly, if U2 = ∅ , then B = B1, and we are done. Otherwise, note that by quorum availability despite D Sets B1 and B2, U1 and U2 are quorums in (V, Q). It follows from the definition that the union of two quorums is also a quorum. Hence V \ B = U1 u U2 is a quorum

and we have quorum availability despite B.

We must now show quorum intersection despite B. Let Ua and Ub be any two quorums in (V, Q)B. Let U = U1 ∩ U2 = U2 \ B1. By quorum intersection of (V, Q), U = U1 ∩ U2 ≠ . But then by Theorem 1, U = U2 \ B1 must be a quorum in (V, Q) B1. Now consider that Ua \ B1 and Ua \ B2 cannot both be empty, or else Ua \ B1 and Ua \ B2 cannot both be empty or else Ua \ B = Ua would be. Hence, by Theorem 1, either Ua \ B1 is a quorum in (V,Q B) B1 = V,Q B1 or Ua \ B2 is a quorum in V,Q B2 or both. In the former case, note that if Ua \ B1 is a quorum in (V, Q) B1, then by quorum intersection of (V, Q) B1, (Ua \ B1) ∩ U ≠ ; since (Ua \ B1) ∩ U = (Ua \ B1) \ B2, it follows that Ua \ B2 ≠ , making Ua \ B2 a quorum in (V, Q) B2. By a similar argument, Ub \ B2 must be a quorum in (V, Q) B2. But then quorum intersection despite B2 tells us that (Ua \ B2) ∩ (Ub \ B2) ≠ ⍰, which is only possible if Ua ∩ Ub ≠ ∅.

THEOREM 3

In an FBAS with quorum intersection, the set of befouled nodes is a D Set.

PROOF:

Let Bmin be the intersection of every D Set that contains all ill-behaved nodes. It follows from the definition of intact that a node v is intact iff v ∉ Bmin. Thus, Bmin is precisely the set of befouled nodes. By Theorem 2, D Sets are closed under intersection, so Bmin is also a D Set.

FEDERATED VOTING

This section develops a federated voting technique that FBAS nodes can use to agree on a statement. At a high level, the process for agreeing on some statement a involves nodes exchanging two sets of messages. First, nodes vote for a. Then, if the vote was successful, nodes confirm a, effectively holding a second vote on the fact that the first vote succeeded.

From each node's perspective, the two rounds of messages divide agreement on a statement a into three phases: unknown, accepted, and confirmed. Initially, a's status is completely unknown to a node v—a could end up true, false, or even stuck in a permanently indeterminate state. If the first vote succeeds, v may come to accept a. No two intact nodes ever accept contradictory statements, so if v is intact and accepts a, then a cannot be false.

For two reasons, however, v accepting a does not suffice for v to act on a. First, the fact that v accepted a does not mean all intact nodes can; a could be stuck for other nodes. Second, if v is befouled, then accepting a means nothing—a may be false at intact nodes. Yet even if v is befouled—which v does not know—the system may still enjoy quorum intersection of well behaved nodes, in which case, for optimal safety, v needs greater assurance of a. Holding a second vote addresses both problems. If the second vote succeeds, v moves to the confirmed phase in which it can finally deem a true and act on it.

Voting with open membership

A correct node in a Byzantine agreement system acts on a statement a only when it knows that other correct nodes will never agree to statements contradicting a. Most protocols employ voting for this purpose. Well-behaved nodes vote for a statement a only if it is valid. Well-behaved nodes also never change their votes. Hence, in centralized Byzantine agreement, it is safe to accept a if a quorum comprising a majority of well-behaved nodes has voted for it. We say a statement is ratified once it has received the necessary votes.

In a federated setting, we must adapt voting to accommodate open membership. One difference is that a quorum no longer corresponds to a majority of well-behaved nodes. Another implication of open membership is that nodes must discover what constitutes a quorum as part of the voting process. To implement quorum discovery, a protocol should specify Q (v) in all messages from Q (v).

Definition (vote). A node v votes for an (abstract) statement a iff

1. v asserts a is valid and consistent with all statements v has accepted, and
2. v asserts it has never voted against a—i.e., voted for a statement that contradicts a—and v promises never to vote against a in the future.

Definition (ratify)

A quorum Ua ratifies a statement a iff every member of Ua votes for a. A node v ratifies a iff v is a member of a quorum Ua that ratifies a.

THEOREM 4

Two contradictory statements a and a- cannot both be ratified in an FBAS that enjoys quorum intersection and contains no ill-behaved nodes.

PROOF:

By contradiction. Suppose quorum U1 ratifies a and quorum U2 ratifies a- By quorum intersection, ∃v ∊ U1 ∩ U2. Such a v must have illegally voted for both a and a-, violating the assumption of no ill-behaved nodes.

THEOREM 5

Let (V, Q) be an FBAS enjoying quorum intersection despite B, and suppose B contains all ill-behaved nodes. Let v1 and v2 be two nodes not in B. Let a and a- be contradictory statements. If v1 ratifies a then v2 cannot ratify a-.

PROOF:

By co)n tradiction. Suppose v1 ratifies a and v2 ratifies a-. By definition, there must exist a quorum U1 containing v1 that ratified a and quorum U2 containing v2 that ratified a. By Theorem 1, since U1 \ ≠ ⍰and U2 \ B ≠ ⍰, both must be quorums in

(V, Q)B, meaning they ratified a and a- respectively in (V, Q)B. But (V, Q)B enjoys quorum intersection and has no illbehaved nodes, so Theorem 4 tells us a and a- cannot both be ratified.

THEOREM 6

Two intact nodes in an FBAS with quorum intersection cannot ratify contradictory statements.

PROOF:

Let B be the set of befouled nodes. By Theorem 3, B is a D Set. By the definition of D Set, (V, Q) enjoys quorum intersection despite B. By Theorem 5, two nodes not in B cannot ratify contradictory statements.

Blocking sets

In centralized consensus, liveness is an all-or-nothing property of the system. Either a unanimously well-behaved quorum exists, or else ill-behaved nodes can prevent the rest of the system from accepting new statements. In FBA, by contrast, liveness may differ across nodes. For instance, in the tiered quorum example of Figure 3, if middle tier nodes v6, v7, v8 crash, the leaf tier will be blocked while the top tier and node v5 will continue to enjoy liveness.

An FBA protocol can guarantee liveness to a node v only if Q(v) contains at least one quorum slice comprising only correct nodes. A set B of failed nodes can violate this property if B contains at least one member of each of v's slices. We term such a set B v-blocking, because it has the power to block progress by v.

Definition (v-blocking)

Let v E V be a node in FBAS (V, Q). A set B ⊆ V is v-blocking iff it overlaps every one of v's slices—i.e., ∀q ∈ Q(v), q ∩ B ≠ ⍰.

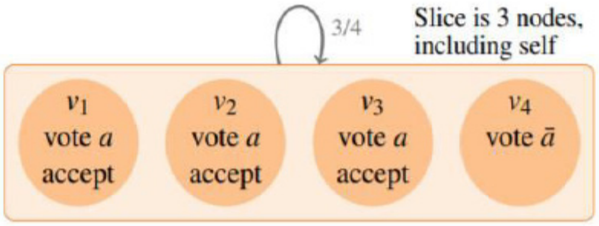


Figure 9: v_4 voted for $\bar{a}$, which contradicts ratified statement a.

THEOREM 7

Let $B \subseteq V$ be a set of nodes in FBAS (V, Q). (V, Q) enjoys quorum availability despite B iff B is not v-blocking for any $v \in V \setminus B$.

PROOF:

“$\forall v \in V \setminus B$, B is not v-blocking” is equivalent to “$\forall v \in V \setminus B, \exists q \in Q(v)$ such that $q \subseteq V \setminus B$.” By the definition of quorum, the latter holds iff $V \setminus B$ is a quorum or $B = V$, the exact definition of quorum availability despite B. As a corollary, the D Set of befouled nodes is not v-blocking for any intact v.

Accepting statements

When an intact node v learns that it has ratified a statement, Theorem 6 tells v that other intact nodes will not ratify contradictory statements. This condition is sufficient for v to accept a, but we cannot make it necessary. Ratifying a statement requires voting for it, and some nodes have voted for contradictory statements. In Figure 9, for example, v4 votes for a- before learning that the other three nodes ratified the contradictory statement a. Though v4 cannot now vote for a, we would still like it to accept a to be consistent with the other nodes. A key insight is that if a node v is intact, then no v-blocking set B can consist entirely of befouled nodes. Now suppose B is a v-blocking set and every member of B claims to accept statement a. If v is intact, at least one member of B must be, too. The intact member will not lie about accepting a; hence, a is true and v can accept it. Of course, if v is befouled, then a might not be true. But a befouled node can accept anything and vacuously not affect the correctness of intact nodes.

Definition (accept)

An FBAS node v accepts a statement a iff it has never accepted a statement contradicting a and it determines that either

1. There exists a quorum U such that $v \in U$ and each member of U either voted for a or claims to accept a, or
2. Each member of a v-blocking set claims to accept a.

Though a well-behaved node cannot vote for contradictory statements, condition 2 above allows a node to vote for one statement and later accept a contradictory one.

THEOREM 8

Two intact nodes in an FBAS that enjoys quorum intersection cannot accept contradictory statements.

PROOF:

Let ⍰ V, Q ⍰be an FBAS with quorum intersection and let B be its D Set of befouled nodes (which exists by Theorem 3). Suppose an intact node accepts statement a. Let v be the first intact node to accept a. At the point v accepts a, only befouled nodes in B can claim to accept it.

Since by the corollary to Theorem 7, B cannot be v-blocking, it must be that v accepted a through condition 1. Thus, v identified a quorum U such that every node claimed to vote for or accept a, and since v is the first intact node to accept a it must mean all nodes in U \B voted for a. In other words v ratified a in ⍰ V, Q ⍰B. Generalizing, any statement accepted by an intact node in ⍰ V, Q ⍰must be ratified in ⍰ V, Q ⍰ B. Because B is a D Set, ⍰ V, Q ⍰ B enjoys quorum intersection. Because additionally B contains all ill-behaved nodes, Theorem 4 rules out ratification of contradictory statements.

Safety

Consider an FBAS (V, Q) in which the only quorum is unanimous consent—i.e., ∀v, Q(v) = {V}. This ought to be a conservative choice for safety—don’t do anything unless everyone agrees. Yet since every node is v-blocking for every v, any node can single-handedly convince any other node to accept arbitrary statements.

The problem is that accepted statements are only safe among intact nodes. But, the only condition necessary to guarantee safety is quorum intersection of well-behaved nodes, which might hold even in the case that some well- behaved nodes are befouled. In particular, when Q(v) = {V}, the only D Sets are ⍰ and V, meaning any node failure befouls the whole system. By contrast, quorum intersection holds despite every B ⊆ V.

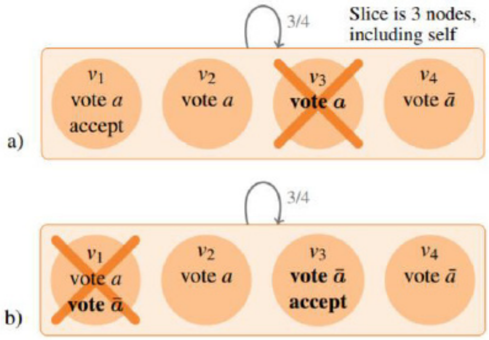


Figure 10: Scenarios indistinguishable to v_2 when v_2 does not see bold messages

Liveness

Another limitation of accepted statements is that other intact nodes may be unable to accept them. This possibility makes reliance on accepted statements problematic for liveness. If a node proceeds to act on a statement because it accepted the statement, other nodes could be unable to proceed in a similar fashion. Consider Figure 10a, in which node v3 crashes after helping v1 ratify and accept statement a. Though v1 accepts a, v2 and v4 cannot. In particular, from v2's perspective, the situation depicted is indistinguishable from Figure 10b, in which v3 voted for a- and is well-behaved but slow to respond, while v1 is ill-behaved and sent v3 a vote for a- (thereby causing v3 to accept a-) while illegally also sending v2 a vote for a.

To support a protocol-level notion of liveness in cases like Figure 10a, v1 needs a way to ensure every other intact node can eventually accept a before v1 acts on a. Once this is the case, it makes sense to say the system agrees on a.

Definition (agree)

An FBAS (V, Q) agrees on a statement a iff, regardless of what subsequently transpires, once sufficient messages are delivered and processed, every intact node will accept a.

Comparison to centralized voting

To understand why the above issues arise in federated voting, consider a centralized Byzantine agreement system of N nodes with quorum size T . Such a system enjoys quorum availability with $fL = N - T$ or fewer node failures. Since any two quorums share at least $2T - N$ nodes, quorum intersection of well-behaved nodes holds up to $fS = 2T - N - 1$ Byzantine failures. Centralized Byzantine agreement systems typically set $N = 3f + 1$ and $T = 2f + 1$ to yield $fL = fS = f$, the equilibrium point at which safety and liveness have the same fault tolerance. If safety is more important than liveness, some protocols increase T so that $fS > fL$. In FBA, because quorums arise organically, systems are unlikely to find themselves at equilibrium, making it far more important to protect safety in the absence of liveness.

Now consider a centralized system in which, because of node failure and contradictory votes, some node v cannot ratify statement a that was ratified by other nodes. If v hears $fS + 1$ nodes claim a was ratified, v knows that either one of them is well- behaved or all safety guarantees have collapsed. Either way, v can act on a with no loss of safety. The FBA equivalent would be to hear from a set B where B, if deleted, undermines quorum intersection of wellbehaved nodes. Identifying such a B is hard for three reasons: one, quorums are discovered dynamically; two, ill-behaved nodes may lie about slices; and three, v does not know which nodes are well-behaved. Instead, we defined federated voting to accept a when a v-blocking set does.

The v-blocking property has the advantage of being easily checkable, but is equivalent to hearing from fL + 1 nodes in a centralized system when we really want fS + 1.

To guarantee agreement among all well-behaved nodes in a centralized system, one merely needs fL + fS + 1 nodes to acknowledge that a statement was ratified. If more than fL of them fail, we do not expect liveness anyway. If fL or fewer fail, then we know fS + 1 nodes remain willing to attest to ratification, which will in turn convince all other well-behaved nodes. The reliance on fS has no easy analogue in the FBA model. Interestingly, however, fL + fS + 1 = T , the quorum size, suggesting a similar approach might work with a more complex justification.

Put another way, at some point nodes need to believe a statement strongly enough to depend on its truth for safety. A centralized system offers two ways to reach this point for a statement a: ratify a first-hand, or reason backwards from fS + 1 nodes claiming a was ratified, figuring safety is hopeless if they have all lied. FBA lacks the latter approach; the only tool it has for safety among well-behaved nodes is first-hand ratification. Since nodes still need a way to overcome votes against ratified statements, we introduced a notion of accepting, but it provides a weaker consistency guarantee limited to intact nodes.

Statement confirmation

Both limitations of accepted statements stem from complications when a set of intact nodes S votes against a statement a that is nonetheless ratified. Particularly in light of FBA's non-uniform quorums, S may prevent some intact node from ever ratifying v. To provide v a means of accepting a despite votes against it, the definition of accept has a second criterion based on vblocking sets. But the second criterion is weaker than ratification, offering no guarantees to befouled nodes that enjoy quorum intersection.

Now suppose a statement a has the property that no intact node ever votes against it. Then we have no need to accept a and can instead insist that nodes directly ratify a before acting on it. We call such statements irrefutable.

Definition (irrefutable)

A statement a is irrefutable in an FBAS if no intact node can ever vote against it.

Theorem 8 tells us that two intact nodes cannot accept contradictory statements. Thus, while some intact nodes may vote against a statement a that was accepted by an intact node, the statement an intact node accepted a is irrefutable. This suggests holding a second vote to ratify the fact that an intact node accepted a.

Definition (confirm)

A quorum Ua in an FBAS confirms a statement a iff ∀v ∊ Ua, v claims to accept a. A node confirms a iff it is in such a quorum.

Nodes express that they have accepted statement a by stating “accept (a),” an abbreviation of the statement, “An intact node accepted a.” To confirm a means to ratify accept

(a). A well-behaved node v can vote for accept (a) only after accepting a, as v cannot assume any particular other nodes are intact. If v itself is befouled, accept (a) might be false, in which case voting for it may cost v liveness, but a befouled node has no guarantee of liveness anyway.

THEOREM 9

Let (V, Q) be an FBAS enjoying quorum intersection despite B, and suppose B contains all ill-behaved nodes. Let v1 and v2 be two nodes not in B. Let a and a- be contradictory statements. If v1 confirms a, then v2 cannot confirm a-.

PROOF:

First note that accept (a) contradicts accept (a-)—no well-behaved node can vote for both. Note further that v1 must ratify accept (a) to confirm a. By Theorem 5, v2 cannot ratify accept (a-) and hence cannot confirm a-.

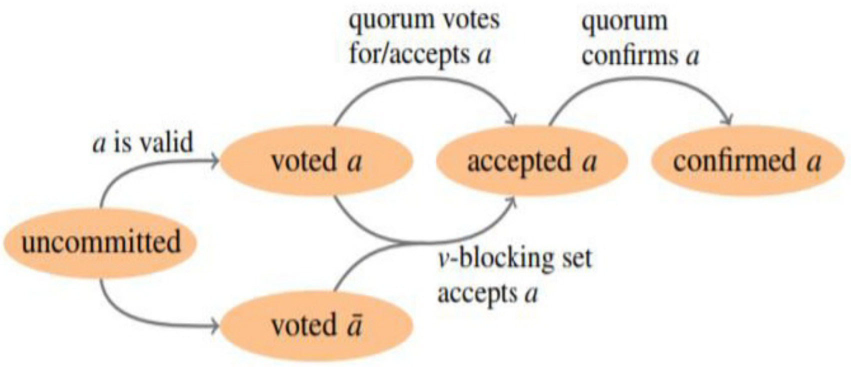


Figure 11: Possible states of an accepted statement *a* at a single node *v*

THEOREM 10

Let B be the set of befouled nodes in an FBAS (V, Q) with quorum intersection. Let U be a quorum containing an intact node (U <J B), and let S be any set such that U ⊆ S ⊆ V. Let S+ = S \B be the set of intact nodes in S, and let S− = (V\S)\B be the set of intact nodes not in S. Either S− = ⍰, or ∃v ∈ S− such that S+ is v-blocking.

PROOF:

If S+ is v-blocking for some v ∈ S−, then we are done. Otherwise, we must show S− = ⍰. If S+ is not v-blocking for any v ∈ S−, then by Theorem 7 either S− = ⍰ or S− is a quorum in (V, Q) B. In the former case we are done, while in the latter we get a contradiction: By Theorem 1 U \ B is a quorum in (V, Q) B. Since B is a D Set, as proven by Theorem 3, (V, Q) B must enjoy quorum intersection, meaning S− ∩(U \ B) ≠ ⍰. This is impossible, since (U \ B) ⊆ S and S− ∩ S =.

THEOREM 11

If an intact node in an FBAS (V, Q) with quorum intersection confirms a statement a, then, whatever subsequently transpires, once sufficient messages are delivered and processed, every intact node will accept and confirm a.

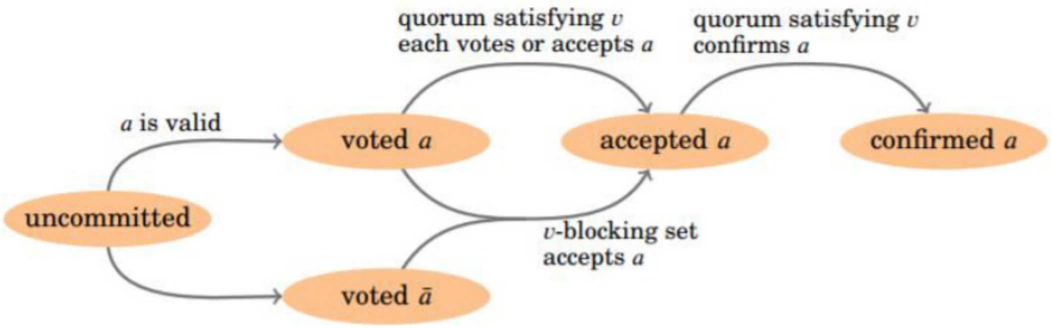


Fig. 11. Possible states of an accepted statement a at a single node v

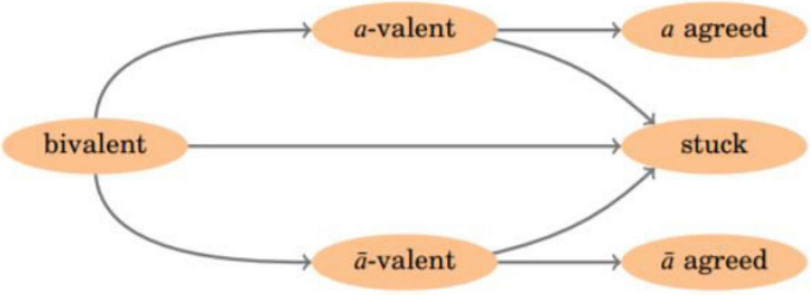


Fig. 12. Possible system-wide status of a statement a

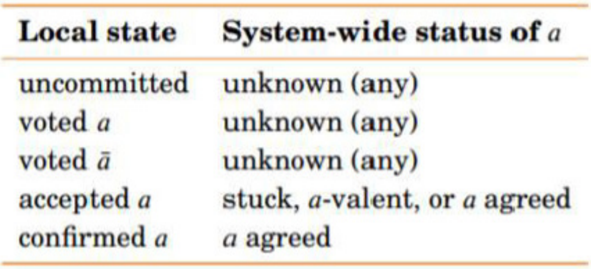
PROOF:

Let B be the D Set of befouled nodes and let U ⊈ B be the quorum through which an intact node confirmed a. Let nodes in U \ B broadcast accept (a). By definition, any node v, regardless of how it has voted, accepts a after receiving accept (a) from a v-blocking set. Hence, these messages may convince additional nodes to accept a. Let these additional nodes in turn broadcast accept (a) until a point is reached at which, regardless of future communication, no further intact nodes can ever accept a. At this point let S be the set of nodes that claim to accept a (where U ⊆ S), let S+ be the set of intact nodes in S, and let S– be the set of intact nodes not in S.

S+ cannot be v blocking for any node in S–, or else more nodes could come to accept a. By Theorem 10, then, S– = 0, meaning every intact node has accepted a. Figure 11 summarizes the paths an intact node v can take to confirm a. Given no knowledge, v might vote for either a or the contradictory a-. If v votes for a-, it cannot later vote for a, but can nonetheless accept a if a v-blocking set accepts it. A subsequent quorum of confirmation messages allows v to confirm a, which by Theorem 11 means the system agrees on a.

Liveness and neutralization

The main challenge of distributed consensus, whether centralized or not, is that a statement can get stuck in a permanently indeterminate state before the system reaches agreement on it. Hence, a protocol must not attempt to ratify externalized values directly. Should the statement "The value of slot i is x" get stuck, the system will be forever unable to agree on slot i, losing liveness. The solution is to craft the statements in votes carefully. It must be possible to break a stuck statement's hold on the question we really care about, namely slot contents. We call the process of obsoleting a stuck statement neutralization.



Local state	System-wide status of a
uncommitted	unknown (any)
voted a	unknown (any)
voted $\bar{a}$	unknown (any)
accepted a	stuck, a-valent, or a agreed
confirmed a	a agreed

Fig. 13. What an intact node knows about the status of statement a

More concretely, Figure 12 depicts the potential status a statement a can have system-wide. Initially, the system is bivalent, by which we mean there is one sequence of possible events through which all intact nodes will accept a, and another sequence through which all intact nodes will reject a (i.e., accept a statement a- contradicting a). At some point, one of these two outcomes may cease to be possible. If no intact node can ever reject a, we say the system is a valent; conversely, if no intact node can ever accept a, we say the system is a-valent.

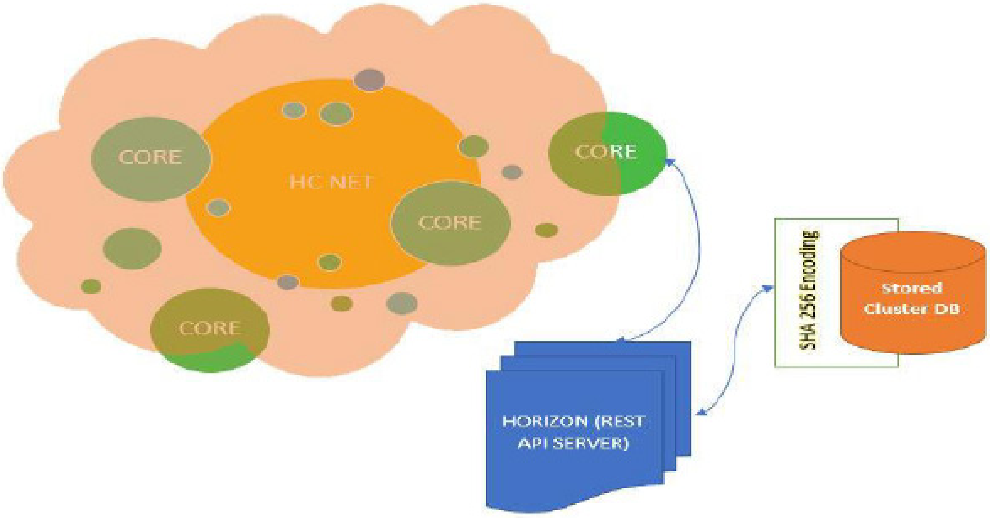
At the time an FBAS transitions from bivalent to a-valent, there is a possible out- come in which all intact nodes accept a. However, this might not remain the case. Consider a PBFT like four-node system {v1, … , v4} in which any three nodes constitute a quorum. If v1 and v2 vote for a, the system becomes a-valent; no three nodes can ratify a contradictory statement. However, if v3 and v4 subsequently vote for a- contradicting a, it also becomes impossible to ratify a. In this case, a's state is permanently indeterminate, or stuck.

As seen in Figure 10a, even once an intact node accepts a, the system may still fail to reach system-wide agreement on a. However, by Theorem 11, once an intact node confirms a, all intact nodes can eventually come to accept it; hence the system has agreed upon a. Figure 13 summarizes what intact nodes know about the global state of a statement from their own local state.

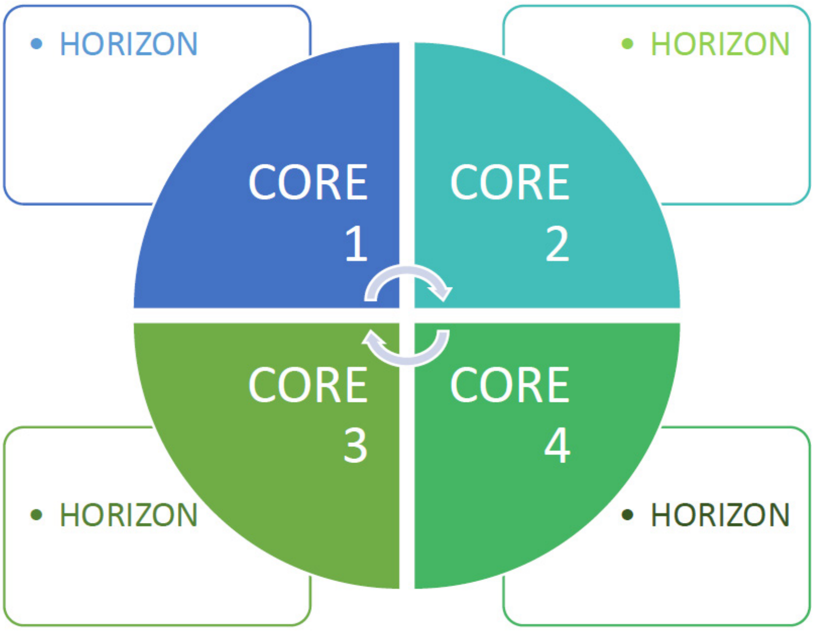
To preserve the possibility of consensus, a protocol must ensure that every statement is either irrefutable, and hence cannot get stuck, or neutralizable, and hence cannot block progress if stuck. There are two popular approaches to crafting neutralizable statements: the view-based approach, pioneered by viewstamped replication and favored by PBFT ; and the ballot-based approach, invented by Paxos . The ballot-based approach may be harder to understand .Compounding confusion, people often call view stamped replication "Paxos" or assert that the two algorithms are the same when they are not.

View-based protocols associate the slots in votes with monotonically increasing view numbers. Should consensus get stuck on the ith slot in view n, nodes recover by agreeing that view n had fewer than i meaningful slots and moving to a higher view number. Ballot-based protocols associate the values in votes with monotonically increasing ballot numbers. Should a ballot get stuck, nodes retry the same slot with a higher ballot, taking care never to select values that would contradict prior stuck ballots. This work takes a ballot-based approach, as doing so makes it easier to do away with the notion of a distinguished primary node or leader. For example, leader behavior can be emulated.

Blockchain Architecture



Blockchain Core & Horizon Integration



Network Details

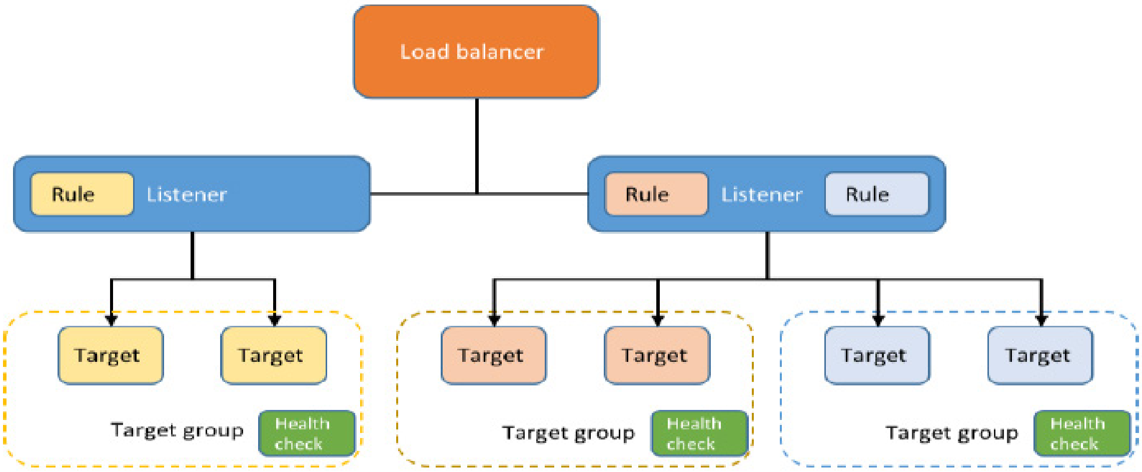
Features	Details
Transaction Per Second (TPS)	3000
Thread Mechanism	De-coupled web service & Multithreaded Response supporting 100000 concurrent service executions per second
Scaling Support	Multiple nodes supporting Horizontal Scaling
Request – Response Caching Caching	Caching enabled on the request node
Encryption Standard	AES -256 Encryption support with MD5 for data decryption
Key Management	Exchange maintains the Private &amp; Public Keys with Oracle in separate instance
Consensus Algorithm	FBFT (Forward Byzantine Fault Tolerance) Algorithm for consensus

Infrastructure Scalability Using Cloud & Load Balancer

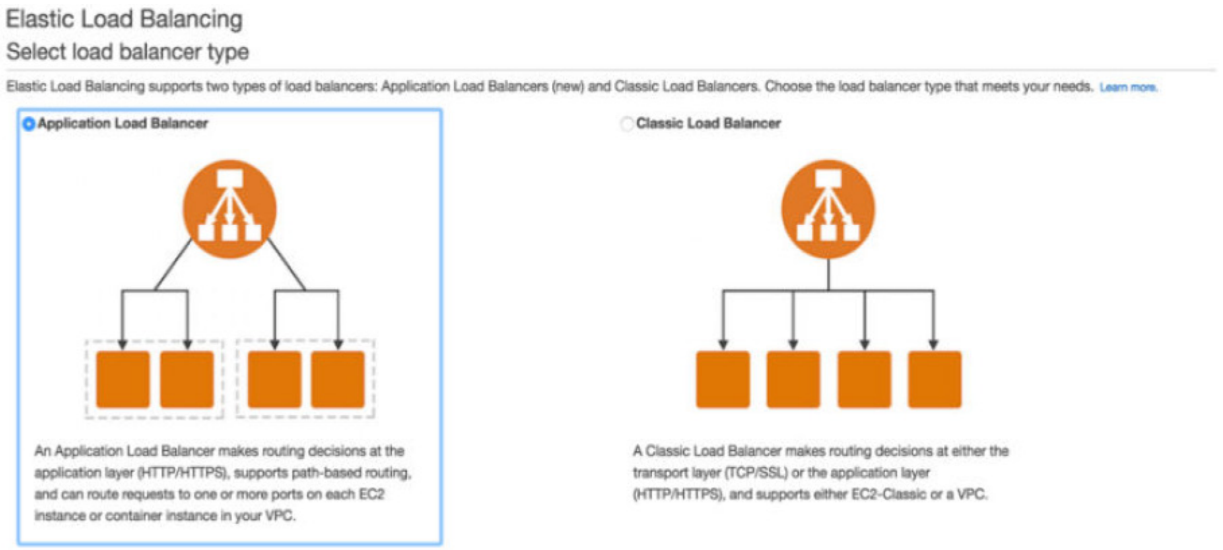
Key features of Application Load Balancers include:

- Path-based routing – URL-based routing policies enable using the same ELB URL to route to different microservices
- Multiple ports routing on the same server
- AWS integration – Integrated with many AWS services, such as ECS, IAM, Auto Scaling, and Cloud Formation
- Application monitoring – Improved metrics and health checks for the application

Proposed Infrastructure Component Details



Proposed Load Balancer Type



Elastic Load Balancing

Select load balancer type

Elastic Load Balancing supports two types of load balancers: Application Load Balancers (new) and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more.

● Application Load Balancer

An Application Load Balancer makes routing decisions at the application layer (HTTP/HTTPS), supports path-based routing, and can route requests to one or more ports on each EC2 instance or container instance in your VPC.

○ Classic Load Balancer

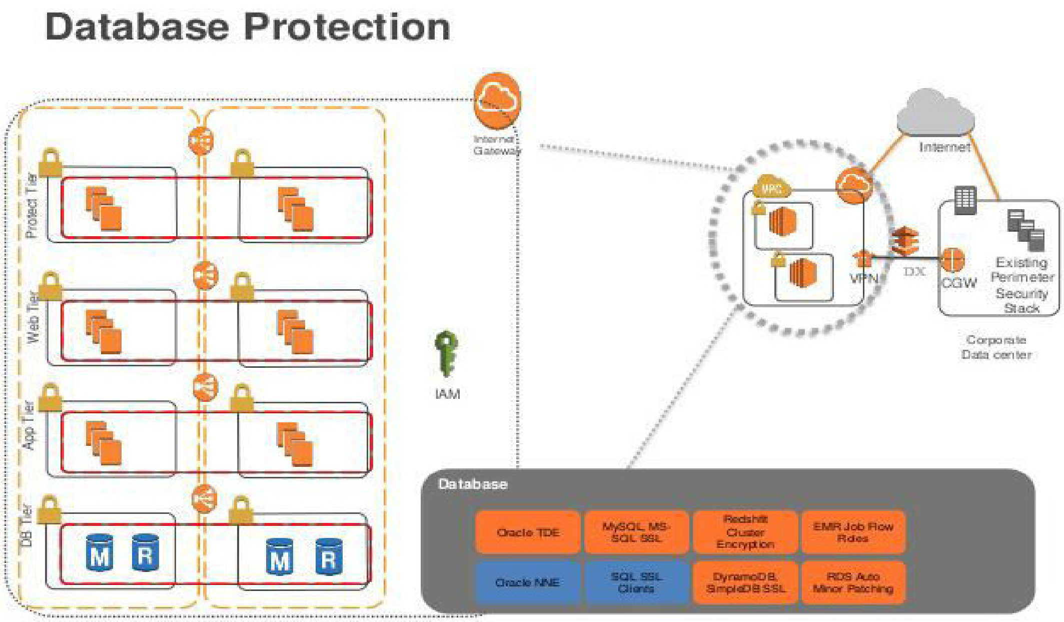
A Classic Load Balancer makes routing decisions at either the transport layer (TCP/SSL) or the application layer (HTTP/HTTPS), and supports either EC2-Classic or a VPC.

Load Balancer reduces the response latency and can achieve approximately 50,000 TPS. With multiple Load balancer, the TPS can be scaled above 1,50,000 TPS. Considering the below Minimum Hardware and Network specifications

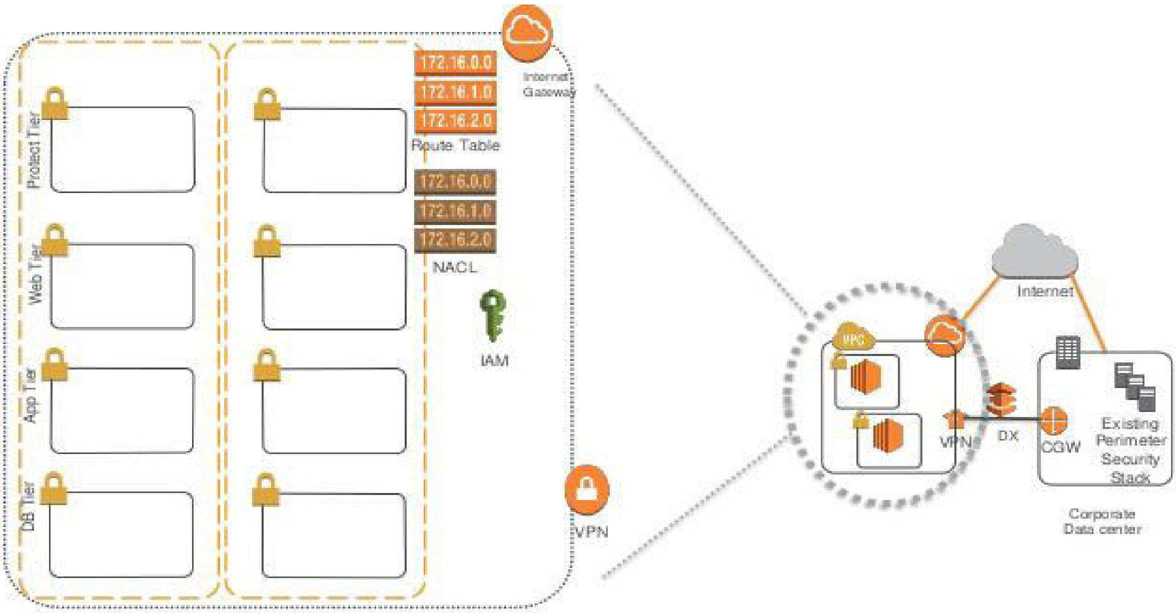
System Requirement Specifications

SEGMENT	MINIMUM REQUIREMENT
CPU	4CORES
INTERNAL MEMORY	16 GB
CACHE	2 GB
BANDWIDTH	3.5 Gbps Dedicated EBS bandwidth
OS	LINUX
DATABASE	Postgress & Oracle (Dev Edition)

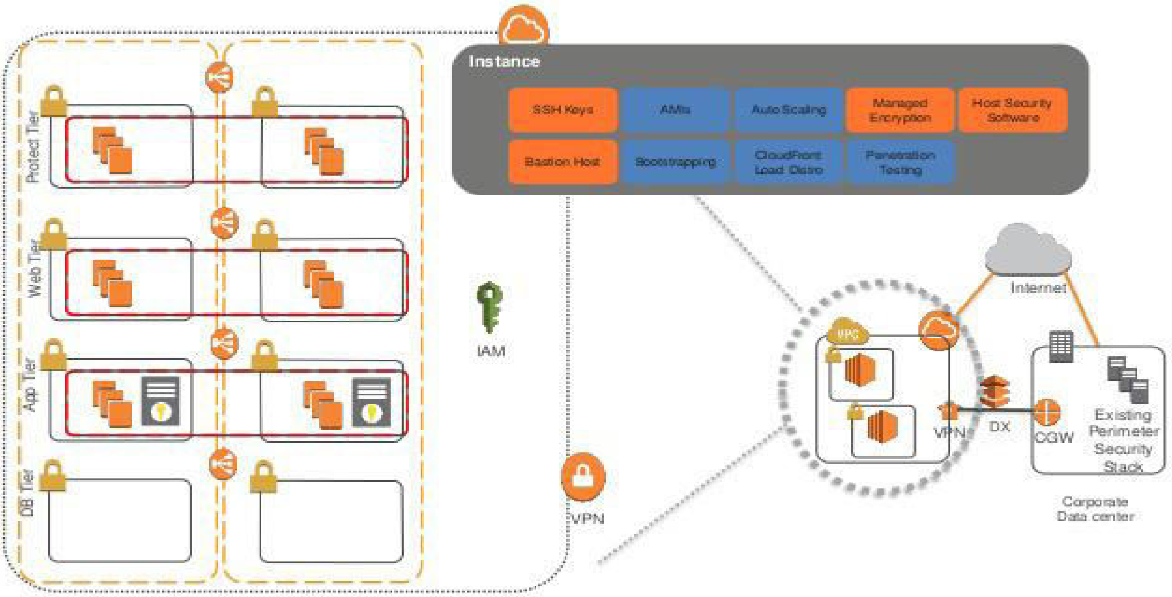
Security & Compliance Integration

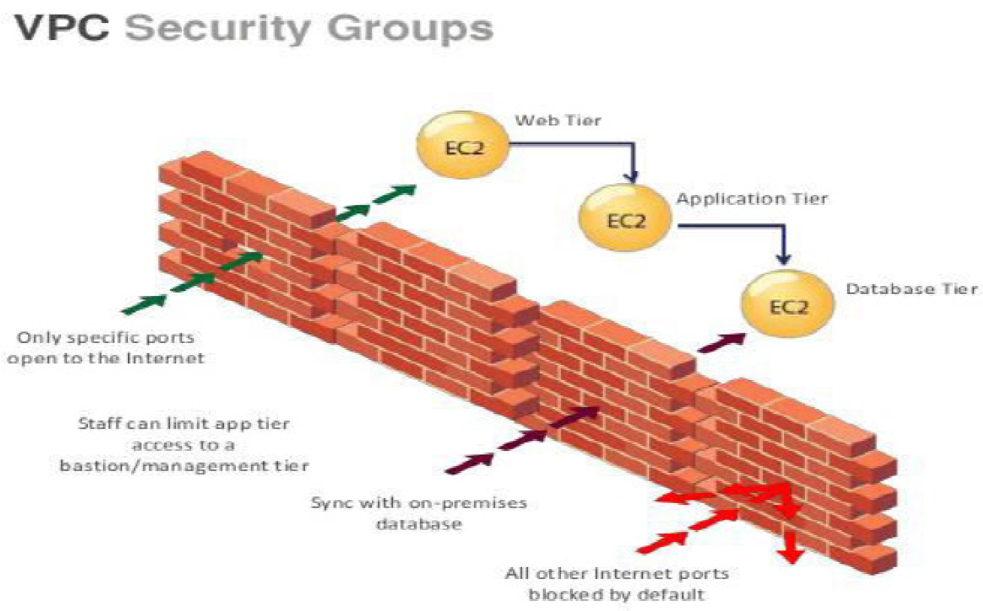


Network Protection



Instance Protection




VPC Security Groups
EC2
Web Tier
EC2
Application Tier
EC2
Database Tier
Only specific ports open to the Internet
Staff can limit app tier access to a bastion/management tier
Sync with on-premises database
All other Internet ports blocked by default

IQQ

Introduction to Conventional Remittances

The business on this digitized world happens mostly with a financial intermediary in place.Internet commerce has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. For most of the transactions, the system works well, but it still suffers the weaknesses of the trust-based model. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust-based model. Complete non-mediation by financial institutions is not possible since there will always exist some chance of reversibility of transaction. The mediation cost increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for nonreversible services. The need for trust spreads with the possibility of reversal. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. Fraud to a certain extent can be accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

The electronic payment system needs cryptographic proof instead of trust, thus allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a digital asset to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

Blockchain Technology

Companies in financial sectors are exploring and experimenting innovative ways to execute transactions quicker for an enhanced customer service, ensure cost efficiency in its operations, and assure transparency to customers and regulators. With large volumes of data getting generated regularly owing to the digitization of records, it becomes crucial for every organization to effectively manage the security threats and achieve significant cost efficiencies. This is where Blockchain, with its promises of decentralized ownership, immutability and cryptographic security of data, is catching the attention of the C-suite executives. Multiple use cases are also getting explored across industries as everyone has started realizing the disruptive potential of this technology.

IQQ- The crypto on IQONIQ Blockchain Network

IQQ forms the native assets on IQONIQ Blockchain Network. An asset is defined as an item value that is stored on the ledger. One IQQ forms the unit of digital currency like any other digital currency such as Bitcoin. The IQQ forms the medium to move money around the world and to construct transactions between different currencies quickly and securely.
IIIQQ is further fragmented at the base level in units called JOTs. A JOT can be defined as the one-tenth million of IQQ, i.e., 10 millionth of IQQ equals to a JOT. The IQONIQ Blockchain Network platform offers all of the innovative features of a shared public ledger on a distributed database—often referred to as Blockchain technology. The native asset of IQONIQ Blockchain Network, IQQ broadly serves two purposes:

A. IQQ will play a small antispam role

Each transaction costs a minor fee—0.0001 IQQ—associated with it. The fee is levied to prevent users with malicious intentions to flood the network. IQQ works mostly as a secured token, mitigating attacks which attempt to generate large numbers of transactions or consume large data space in the ledger.

Additionally, the IQONIQ Blockchain Network requires all accounts to hold a minimum balance of 20 IQQ. This requirement ensures that accounts are genuine and which facilitates the network maintain a seamless flow of transactions.

B. IQQ may facilitate multi-currency transactions.

IQQ sometimes facilitate trades between pairs of currencies between which there is not a sizeable direct market, acting as a bridge. This function is possible when there is a liquid market between the IQQ and each currency involved.

Transacting using IQQ

The IQONIQ Blockchain Network is free to use. If a person has to trade on the live network, the person needs IQQ or the native cryptos to ensure coverage of the base fees of the transaction. Eventually, the transaction on IQONIQ Blockchain Network platform is very low. The initial IQONIQ Blockchain Network will hold 100 billion IQQ at its root account. Then there will be an allocation to different exchanges with an initial amount of say 1miilion to each exchange across geographies. One should be aware at this point about the risk associated with all digital currency including complete loss of value.

When a transaction is initiated on the IQONIQ Blockchain Network using IQQ, the transaction draws IQQ from IQONIQ Blockchain Network, leaving the transaction fee regarding JOT to IQONIQ Blockchain Network platform. The transaction then reaches to an exchange, for instance, PayBito India or PayBito US where the traders trade. Based on the reserve of IQQ the exchanges are holding, and the position of the traders, the demand in the market will determine the value. When the majority of the traders take an extended position in the market, there will be a reduction in IQQ in IQQ reserve in the exchanges and consequently, the IQQ value will be more. On the contrary, if there are more short positions in the market, that is, the traders are selling more IQQ cryptos, there will be more IQQ cryptos available in the market with the exchanges and eventually the value of IQQ will reduce. The market mechanism determines the value of IQQ.

IQONIQ

BECOME ONE

IQQ Distribution

IQQ can be used in banks, currency exchanges, Corporate houses, wallet providers, payment service provider and E-commerce merchant icon. On looking into deeper aspects, with IQQ, Banks do cross-border settlements in real-time at a lower cost without the need for locking funds in Pre-funded Nostro Accounts. With IQQ, Currency Exchanges manage liquidity efficiently, settle funds in real-time for their customers and discover new remittance corridors. The corporate can benefit with the IQQ, PSPs can operate on this network without the need for an acquiring bank. For the E-commerce merchants, With IQQ, merchants can cut down the cost of receiving payments considerably and receive money instantly. IQQ provides an uber efficient, ultra-low cost, zero turnaround time alternative to conventional settlement methods.

Why IQQ: IQQ as an alternative digital crypto?

The points which differentiate IQQ and makes it more tradable are the following parameters:

I. Real-time Settlement: IQQ settles payments in 2-4 seconds in 1000+ node environment

II. Scalable: IQQ can handle 3000 concurrent transactions (transactions per second) and 100+ million transactions a day. At 75,000 transactions per second, the network scalability exceeded the throughput of current payment networks such as Visa, in a test environment, with the implementation of SegWit.

III. Wide Applications: IQONIQ products which operate on IQONIQ Blockchain are used for Remittance, Corporate Payment, Trade Finance, Commerce and Forex deal. These are mainly: IQONIQ Remit, IQONIQ Commerce, IQONIQ Market Makers and IQONIQ Commerce.

IV. Wide Acceptance: IQONIQ Products are used by Global Banks, Currency Exchanges, Payment Service Providers, Corporate Houses and E-commerce Merchants.

IQONIQ

BECOME ONE

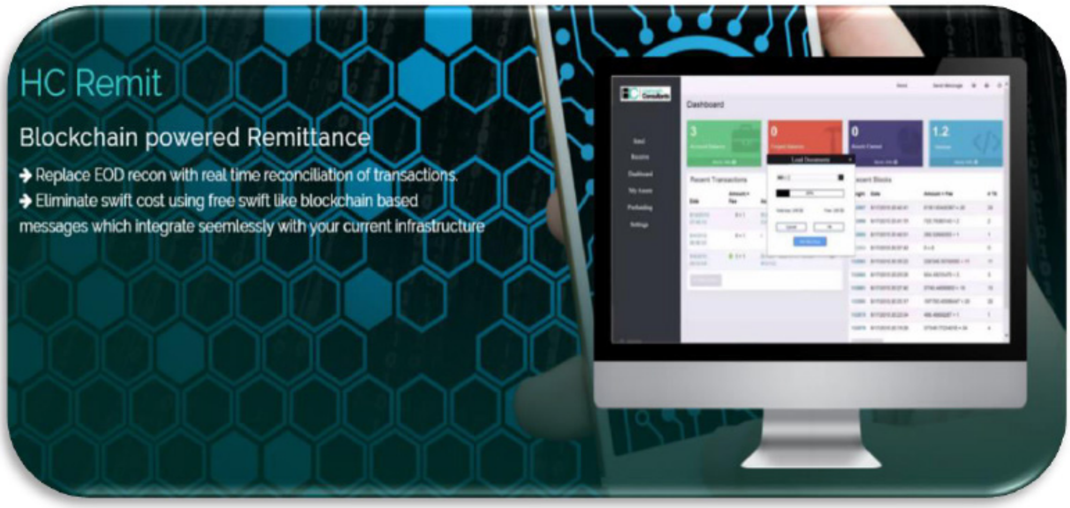
IQONIQ Remit

Introduction

The following paragraphs describe the prospect of the implementation of the IQONIQ Remit product for a real-time settlement for remittances using Blockchain. The scope of this document is limited to a pilot phase and production implementation between Banks and a Remittance Partner. The proposed implementation leverages a Blockchain platform to exchange remittance messages real time between banks and remittance exchanges. The settlement is carried out by debiting a prefunded account maintained by the receiving bank. Invoices and AML/KYC information can also be exchanged securely by recording the bytecode over Blockchain and sharing the document using No SQL databases. The implementation is expected to significantly reduce transaction costs and increase efficiency owing to instant reconciliation, visibility of prefunded account and elimination of Swift messaging costs.

IQONIQ Remit Product description

IQONIQ Remit is Infinite Holding's product for faster low-cost cross-currency payments.



HC Remit

Blockchain powered Remittance

→ Replace EOD recon with real time reconciliation of transactions

→ Eliminate swift cost using free swift like blockchain based messages which integrate seemlessly with your current infrastructure

IQONIQ

BECOME ONE

Product Benefits

The points which differentiate IQQ and makes it more tradable are the following parameters:

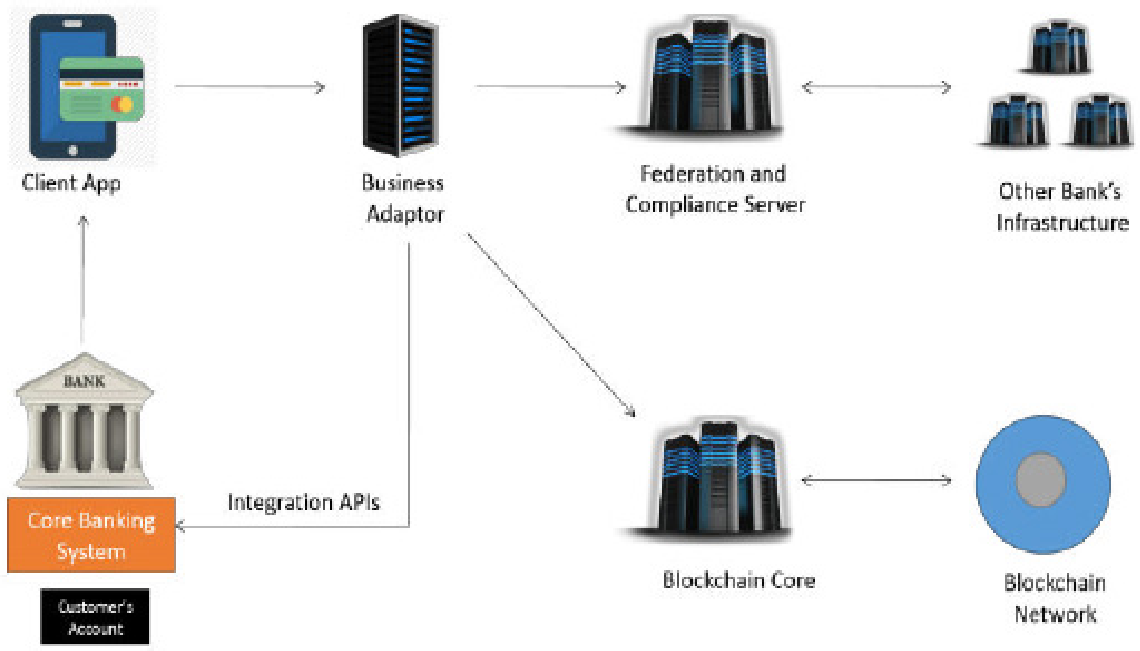
- Eliminate higher transaction cost arising from using SWIFT networks and Payment Gateways.
- Onboard new banks to your cross-border network faster as opposed to traditional remittance systems.
- Automate critical business processes using Smart Contracts.
- Lower maintenance and recurring costs for your remittance system.
- Safe, secure and authentic transaction handling.

Product Features

The points which differentiate IQQ and makes it more tradable are the following parameters:

- Live tracking of transactions with all parties using the Remittance Dashboard
- Monitor your funds and NOSTRO accounts through real-time tracking.
- Cross currency Remittance.
- New banks can be easily on-boarded to the network.

High Level Architecture



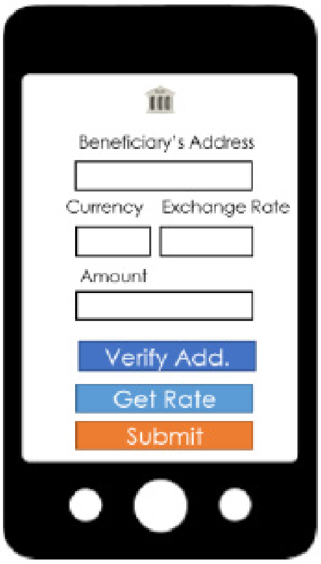
International Remittance in Beneficiary Country

The below use case enables customers registered with Bank to remit to Beneficiary country. The settlement is real-time and messaging on SWIFT system is replaced by API services and Blockchain transactions.

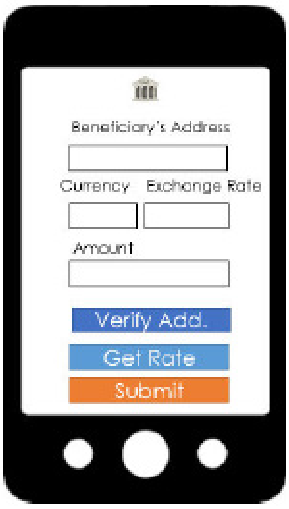
Customer Flow

1. Customer registers with an account with a Bank and is assigned a friendly ID such as Bob@alrajhi.com

2. He goes to the retail remittance page and is shown a form as below:



3. He enters the beneficiary address for the transaction. Instead of Bank Account No.,Beneficiary Name, Routing Number, SWIFT Code, IFSC Code etc., the proposed system requires that user only provides the receiver's ID that is provided to him when signing up.This ID is issued by the bank or can be asked from the user through a registration process.An existing identifier such as Bank Account No., Unique Identification Number, the Mobile number can be used.



Types of Beneficiary Address

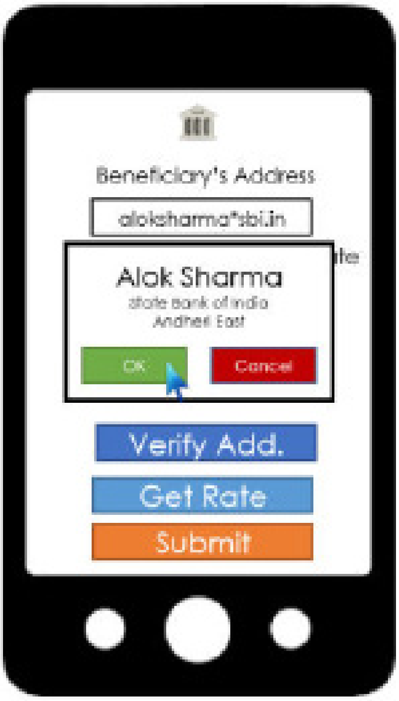
Address
aloksharma*bank.in

Bank Account No.-
9002812331-
ICIC000040*bank.in

Unique Identity Number
100088123311З*bank.org

Mobile Number
9008903291*bank.org

4. The beneficiary address that is submitted can also be verified before the actual transaction is submitted for remittance. This helps avoid reworking on the customer and bank's part if the transaction details are incorrect. The customer is shown the beneficiary details before submission and the transaction only goes through once the details are verified.

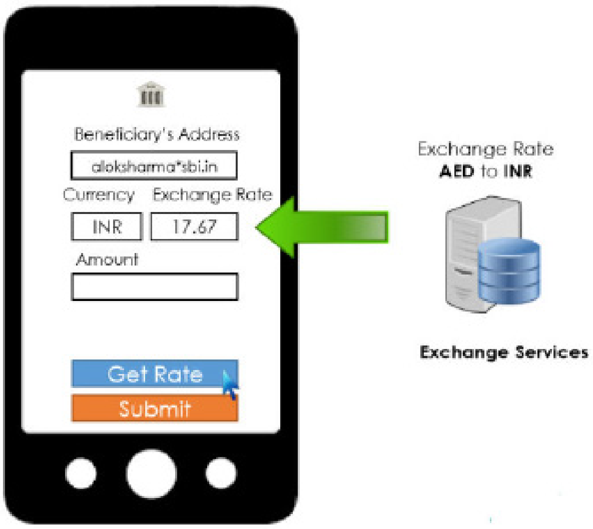


Beneficiary's Address

alokshama*sbi.in

Alok Sharma
State Bank of India
Andheri East

5. The customer enters the currency to be remitted and the amount to be remitted. He is shown the current exchange rate from the bank's APIs or Global Distributed Exchange's order book if so desired.



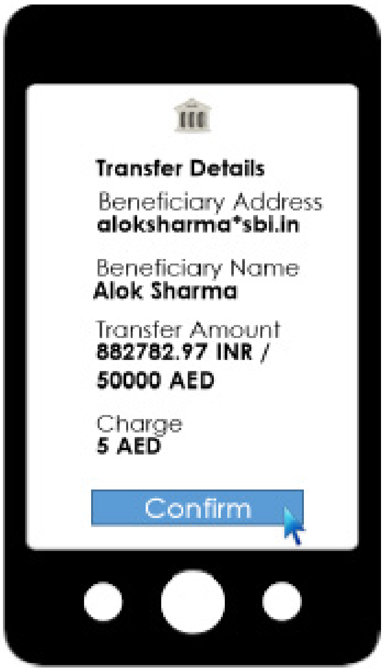
Beneficiary's Address: aloksharma*sbi.in

Currency: INR

Exchange Rate: 17.67

Amount:

6. The customer is shown the amount that will be credited to the beneficiary bank account and the fees that he needs to pay for the remittance. He needs to acknowledge the details to proceed with the remittance.



Transfer Details

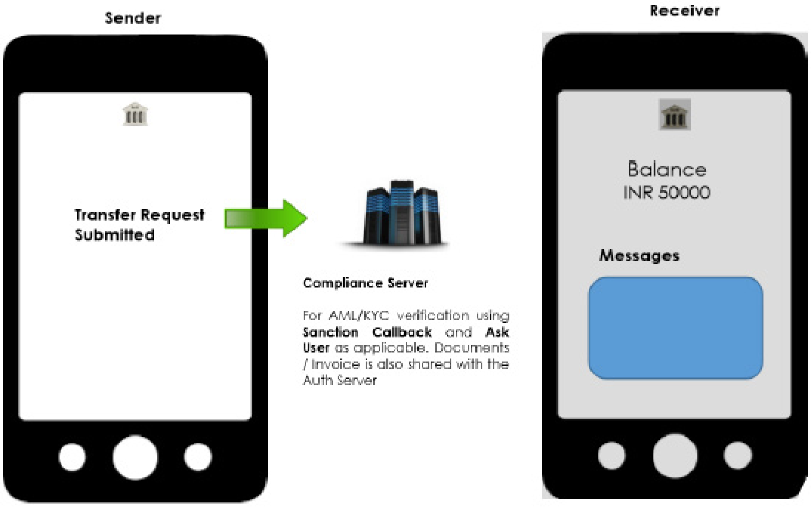
Beneficiary Address
aloksharma*sbi.in

Beneficiary Name
Alok Sharma

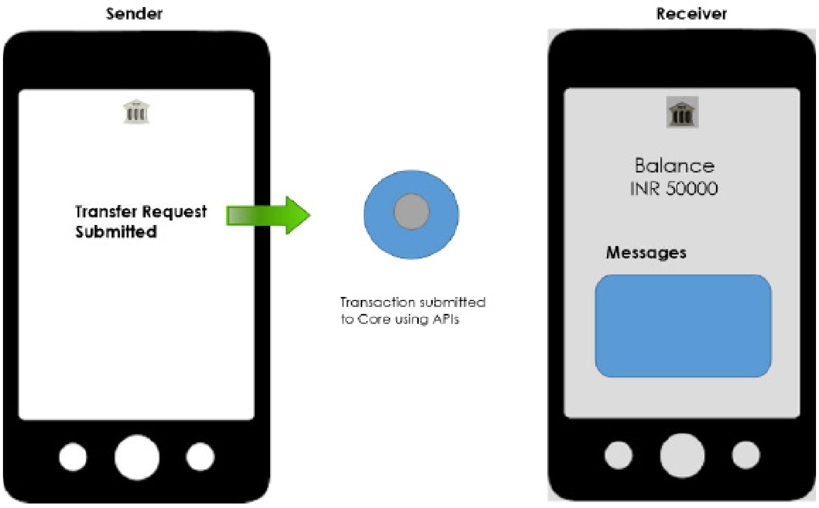
Transfer Amount
882782.97 INR / 50000 AED

Charge
5 AED

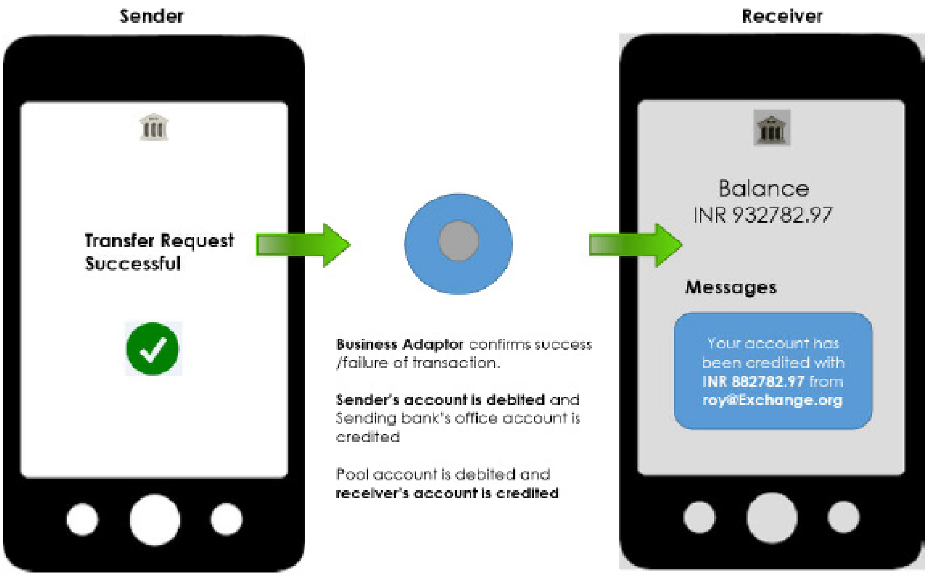
7. Once the customer confirms the details, a compliance check for the Sender and Receiver of the transaction is carried out. The sender's compliance server verifies the sender's compliance details such as KYC/AML/CTF. The remittance initiating party then shares the details of the transaction with the receiving party. The receiving party's compliance server then does a complete compliance check for the beneficiary's details. If all the compliance requisites are met, both the parties agree to initiate the transaction.


Sender
Transfer Request Submitted
Compliance Server
For AML/KYC verification using **Sanction Callback** and **Ask User** as applicable. Documents / Invoice is also shared with the Auth Server
Receiver
Balance
INR 50000
Messages

8. Once compliance checks are completed the transaction is submitted to the Blockchain.


Sender
Transfer Request Submitted
Transaction submitted to Core using APIs
Receiver
Balance
INR 50000
Messages

9. The receiving party receives the transaction on the Blockchain. It verifies with the compliance server that the incoming transaction has been authorized and all compliance requirements have been met. Once these details are verified, the receiving end debits the office account and credits the beneficiary's account.



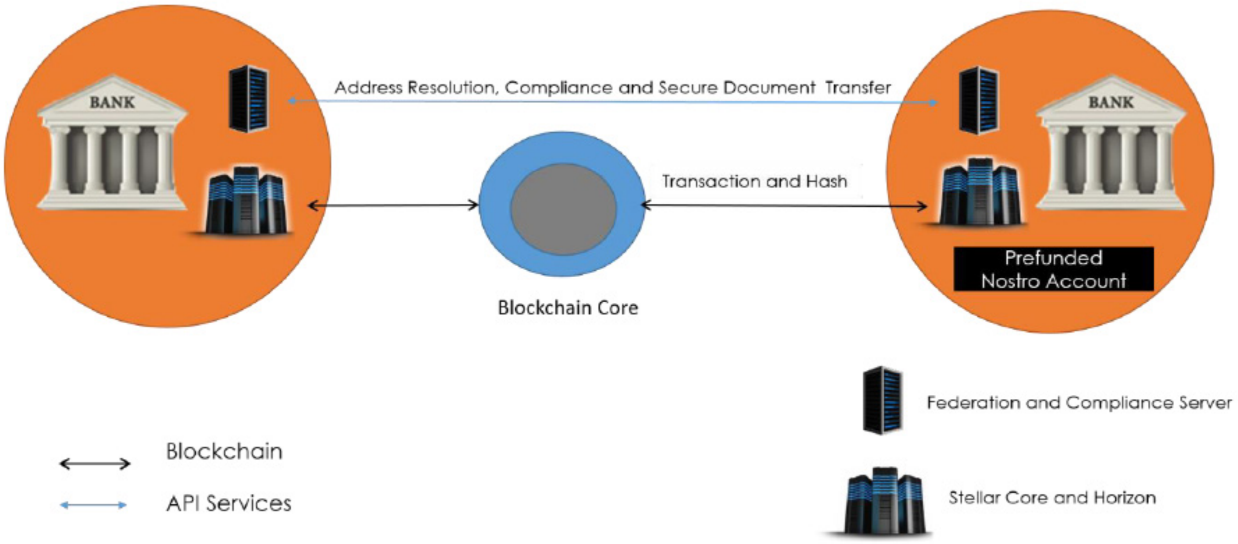
Your account has been credited with **INR 882782.97** from **roy@Exchange.org**

Business Adaptor confirms success /failure of transaction.

Sender's account is debited and Sending bank's office account is credited

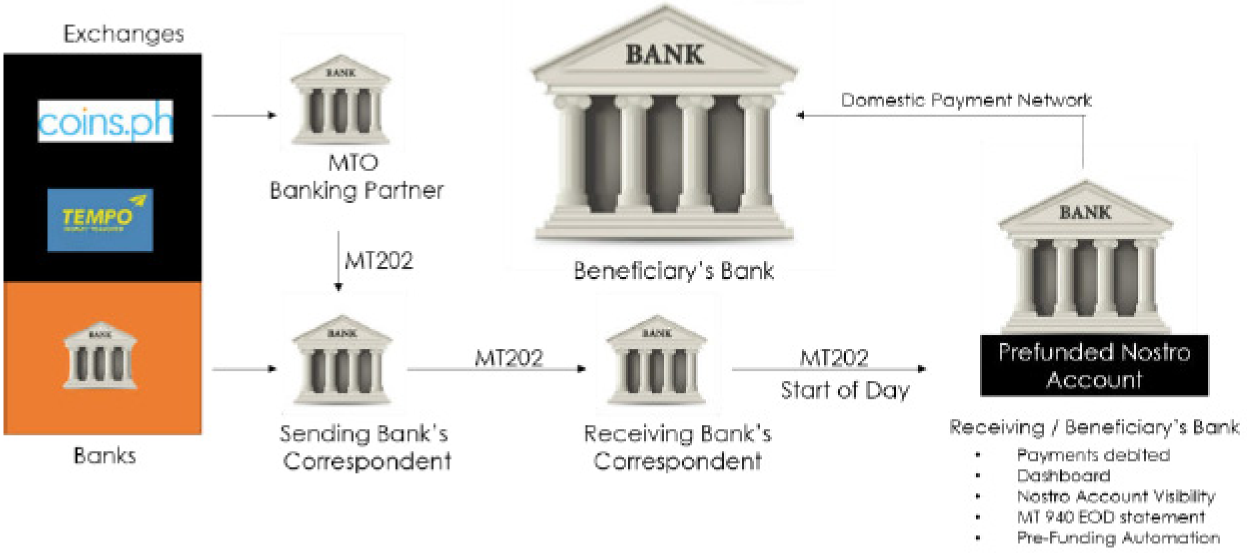
Pool account is debited and **receiver's account is credited**

Proposed Model



Movement of Fiat Currency

Actual Settlement between the transacting parties happens through established conventional means. the sending party holds pre-funded Nostro accounts either through direct or correspondent relationship with the receiving party. Alternatively, transacting parties can net off transactions and then settle either at the end of the day or any other designated time period. Transaction to domestic banks not on the network can be initiated through the domestic payment network in Beneficiary Country (Money Transfer Solutions).



Project Benefits

- Remittances to other countries using UIN's / Mobile Number / UPI VPAs
- Faster Reconciliation and KYC and AML verification
- Elimination of SWIFT costs
- Generation and Transfer of Nostro/Prefunded Account Statement
- Data stored on Blockchain for auditory purpose
- Secure document transfer
- Operational cost reduction

Platform Benefits

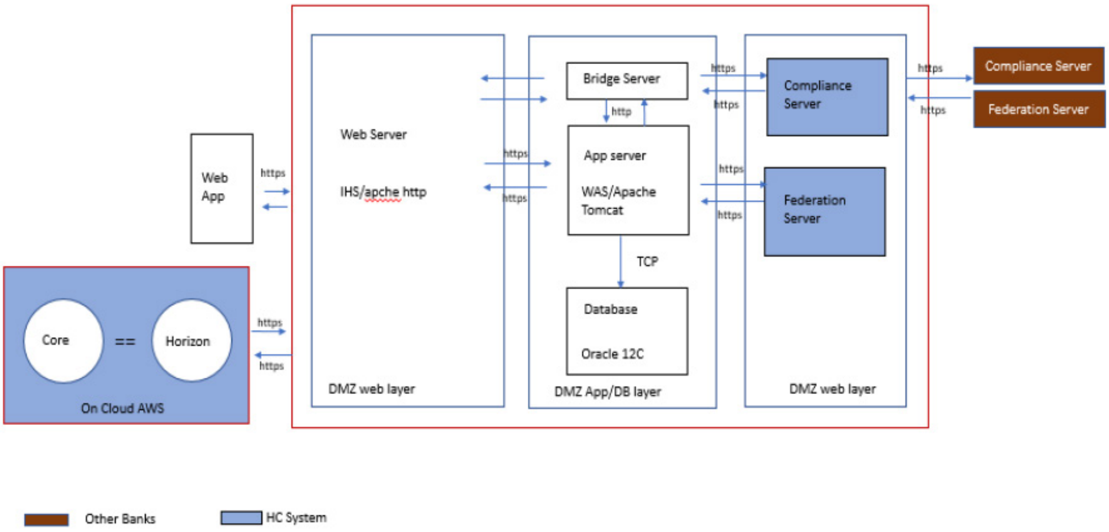
- Greater transaction per second on the network - 3000 TPS and 290 million transactions per day
- Low cost per transaction and Open Source Platform
- Customer data is anonymous on the Blockchain
- Proven scalability as operates a global public network

Applications

- Retail Remittances
- International E-commerce payments
- Payment disbursements

Proposed Architecture

1. 3-point architecture with separation of Web, App and DB servers
2. All rest services communication protocol are https
3. Information shared over the cloud setup does not contain any customer data



IQONIQ

BECOME ONE

IQONIQ Commerce

Introduction

The following paragraphs shed light on the implementation of the IQONIQ Commerce product for a real-time settlement for remittances and Trade Finance Operations such as processing Letter of Credit, Bank Guarantees, pre-shipment loans and invoice discounting using Blockchain. The scope of this document is limited to a pilot phase and production implementation between Banks and a Remittance Partner.

The proposed implementation leverages a Blockchain platform to exchange remittance messages real time between banks and remittance exchanges. The settlement is carried out by debiting a prefunded account maintained by the receiving bank. Invoices and AML/KYC information can also be exchanged securely by recording the bytecode over Blockchain and sharing the document using No SQL databases.

The implementation is expected to significantly reduce transaction costs and increase efficiency owing to instant reconciliation, visibility of prefunded account and elimination of Swift messaging costs.

Product description

IQONIQ Commerce is Infinite Holding’s product for faster low-cost cross-currency payments and for processing Letter of Credit, Bank Guarantees, pre-shipment loans and invoice discounting using Blockchain.

Product Benefits

- Eliminate higher transaction cost arising from using SWIFT networks and Payment Gateways.
- Onboard new banks to your cross-border network faster as opposed to traditional remittance systems.
- Automate critical business processes using Smart Contracts.
- Lower maintenance and recurring costs for your remittance system.
- Safe, secure and authentic transaction handling.

IQONIQ

BECOME ONE

- Reduce processing time through secure, online exchange of documents and instant settlement
- Manage working capital efficiently through live tracking of an account balance
- Real-time reconciliation and information sharing with other parties on the network
- Automatically reconcile transactions with other financial institutions
- Automate critical business processes through smart contracts
- Protect your data through encryption and Blockchain immutability

Product Features

- Live tracking of transactions with all parties using the Remittance Dashboard.
- Monitor your funds and NOSTRO accounts through real-time tracking.
- Cross currency Remittance.
- New banks can be easily on-boarded to the network.
- Online Purchase Order and Invoice creation and exchange
- Share Trade and Shipping documents securely through encryption requirement
- Banks can instantly verify and settle payments
- Can be expanded to include other trading partners or banks

Project Benefits

- Remittances to other countries using UIN’s / Mobile Number / UPI VPAs
- Faster Reconciliation and KYC and AML verification
- Elimination of SWIFT costs
- Generation and Transfer of Nostro/Prefunded Account Statement
- Data stored on Blockchain for auditory purpose
- Secure document transfer
- Operational cost reduction

IQONIQ
BECOME ONE

IQONIQ Corporate Payment

Introduction

The following paragraphs reflect the prospectus for the implementation of the IQONIQ Corporate Payment product for a real-time settlement for remittances and payments using Blockchain. The scope of this document is limited to a pilot phase and production implementation between Banks and a Remittance Partner.

The proposed implementation leverages a Blockchain platform to exchange remittance messages real time between banks and remittance exchanges. The settlement is carried out by debiting a prefunded account maintained by the receiving bank. Invoices and AML/KYC information can also be exchanged securely by recording the bytecode over Blockchain and sharing the document using No SQL databases.

The implementation is expected to significantly reduce transaction costs and increase efficiency owing to instant reconciliation, visibility of prefunded account and elimination of Swift messaging costs.

IQONIQ Corporate Payment Product description

IQONIQ Corporate Payment is Infinite Holding's product for faster low-cost cross-currency payments and cross-border remittances.

Product Benefits

- Reduce processing time through secure, online exchange of documents and instant settlement
- Manage working capital efficiently through live tracking of an account balance
- Real-time reconciliation and information sharing with other parties on the network
- Automatically reconcile transactions with other financial institutions
- Automate critical business processes through smart contracts
- Protect your data through encryption and Blockchain immutability

Product Features

- Online Purchase Order and Invoice creation and exchange
- Share Trade and Shipping documents securely through encryption requirement
- Banks can instantly verify and settle payments
- Can be expanded to include other trading partners or banks

Use Cases Of Different Blockchain Based Products

USE CASE 1: Wealth Management

IQONIQ Blockchain Initiative

The distributed ledgers, Blockchain technology can be implemented in a number of pragmatic ways within the wealth and asset management lifecycle. The highly flexible Blockchain technology would enable the client to ensure an onboarding process without significant friction. These distributed ledgers can streamline management of model portfolios, speed the clearing and settlement of trades while easing the compliance burdens concerning Anti-Money Laundering (AML) and Know Your Customer(KYC). This would enhance the client experience by abandoning redundancy and reducing operational expenses.

Blockchain can be well implemented to reconcile information through already existing systems and create opportunities for new markets and products. It can be included in transactions where assets are moved between parties or contracts are executed- rollovers, trusts, estates, and insurances. The distributed ledgers support to validate and execute complex transactions in near real time. The smooth operation that Blockchain technology offers will enrich the client experience by streamlining different processes while reducing operation costs, significantly.

Introduction

The recent exploration and experimentation concerning innovative ways to enable quicker transactions and an enhanced customer service by the organizations in the financial sector, also, put a high priority in ensuring cost efficiency in its operations as well as maintaining utmost transparency that satiates the regulators and the customers.

Digitization of records has given birth to large volumes of data that makes it utmost important for any organization to defy security threats while achieving superior cost efficiencies. This particular prospectus invites Blockchain, an entity boasting decentralized ownership allure C-suite executives. With its immutability and cryptographic security of data, Blockchain has enabled the realization of the disruptive potential of this technology.

The following paragraphs will discuss in depth, the different ways through which wealth and asset management firms can harness the benefits of Blockchain technology through real-world applications. The paper will also give an insight into approaching Blockchain innovation as well as highlighting near-term practical applications of the same.

An Overview of IQONIQ Blockchain

The challenges in current market scenario especially in disruptive technology can be addressed with IQONIQ Blockchain, the blockchain technology platform on which IQONIQ Remit, the remittance product operates. There are other products as well which works on IQONIQ Remit platform- IQONIQ Trade Finance, IQONIQ Commerce, IQONIQ Market Maker and IQQ. The Blockchain network is trusted by Banks, currency exchanges, corporate houses, Fintechs, Global Merchants and payment networks.

An Overview of IQQ

IQQ forms native assets on IQONIQ Blockchain. Asset is defined as an item value that is stored on the ledger. One IQQ forms the unit of digital currency like any other digital currency such as Bitcoin. The IQQ, forms the medium to move money around the world and to construct transactions between different currencies quickly and securely. IQQ is further fragmented at the base level in units called JOTs. A JOT can be defined as the one -tenth million of IQQ, i.e., 10 millionth of IQQ equals to a JOT. The IQONIQ Blockchain platform offers all of the innovative features of a shared public ledger on a distributed database—often referred to as blockchain technology.

Scope1: KYC and AML Management

Blockchain technology can be brought in to build and enrich a client profile that will surpass excellence. All confidential data points such as net worth, social media profiles, account information, preferences and profile data are allowed to be given access to with proper permission - to edit, read and write the data. Each data point and individual block of data is stored securely and can only be accessed through authorized permission.

Driving Factors

Client onboarding has never been easier than with Blockchain. Wealth managers must go through a tedious verification of client identification by taking into consideration, sources of wealth, business interests, political affiliation, residency address proof and many more. Verifying and collecting such volume of data often needs a time investment of days or even weeks.

Challenges:

- Strict onboarding requirements
- Proof of identification Residency
- Marital status Sources of wealth Occupation
- Business interests

Milestones:

- Profile stored on a blockchain/distributed ledger
- Trusted parties are granted access to all or part of the profile based on cryptography
- New relationships would be initiated by profile owner
- The system inherently enables an audit trail for tracking changes to the chain. As a result, requiring fact-checking, such as AML, are simplified
- Integrate blockchain technologies into onboarding and ACH and ACAT systems and processes

Benefits:

- Can facilitate many key
- Functions of onboarding Client and risk profiling Financial planning
- Anti-money laundering checks and money movement
- Can enhance or possibly replace traditional systems, such as ACH and ACAT
- Enables near-instantaneous transfers of assets between financial institutions with authenticated provenance of tracked changes

Scope 2: Open Architecture Investment Management

Open architecture, in its definition, is the ability of a financial institution to offer its clients proprietary as well as external services and products. The perspective offers investment firms to get past the conflict of interest or favouritism, practicing which the firm may only recommend proprietary products.

Driving Factors:

Wealth managers often face challenges regarding open architecture investment offerings. The rapidly multiplying facet of an open architecture as well as the third party investment models which are placed in separately managed accounts often is the harbinger of complex operational challenges. On the other hand, a distributed ledger technology enables the portfolio managers to instantly notify all subscribed clients if there are any changes in the portfolio. It also allows real-time views of cash flows and individual account performances. Smart contracts also let the management of fees paid by sponsors relatively easy by transacting a payment every time the model is downloaded or used.

Challenges:

- A wealth and asset manager using different platforms and data architectures causes difficulties in distributing, monitoring and updating third- party models.
- Firms must support redundant model management systems.
- Managers are often required to email models to program sponsors or use proprietary portals.

Milestones:

- Investment managers would create and maintain a model — similar to how they do it today.
- Models could be transmitted through a blockchain to various subscribed brokers.
- Individual accounts can be invested according to the model.
- Customization for restrictions and other account-level constraints can be stored and applied.

Benefits:

- Will allow other account transactions and trades to be shared more easily
- Can provide near-real-time performance, portfolio risk and drift data, allowing managers to observe more easily and have greater insights
- Can reduce the amount of reconciliation needed by moving from the current segregated master ledger to a secure, distributed one
- Reduces the need for some intermediaries responsible for settling and executing trades

Scope 3:

Boasting synonymous terminologies like automated trading, system trading, mechanical trading systems or algorithmic trading, the Automated Trading System allows traders to execute clearly conditioned rules for trade entries and exits. Once programmed, these norms and regulations can be executed via any computer. The norms set for the trade entries and exits deal with conditions such as simplistic moving an average crossover, or, it can be an amalgamation of complex strategies which involve a deep understanding of the programing language of a particular user's trading platform. It may from time to time demand the expertise of a soughtafter programmer.

Driving Factors:

Automated Trading Systems enables the traders to stick to a particular plan by eliminating significant chances of emotions being present in the trading process. As a precaution, the traders can always use pre-decided norms on pre-existing data before putting out money in a real world, live trading situation. The opportunity of back testing lets the traders examine the legitimacy of a trading idea. It also presents to the traders, a chance to determine the system's expectancy, that is, the chances of winning or losing an average amount, per unit of risk.

Challenges:

- Automated trading is a sophisticated method of trading, yet not infallible. Depending on the trading platform, a trade order could reside on a computer – and not a server

- It is possible for an automated trading system to experience anomalies that could result in errant orders, missing orders, or duplicate orders
- Over-optimization refers to excessive curve-fitting that produces a trading plan that is unreliable in live trading
- Centralized order books and trade management results to delayed reconciliation and opaque accounting statements that fails to provide clarity to the traders regarding the position settlements and cost details

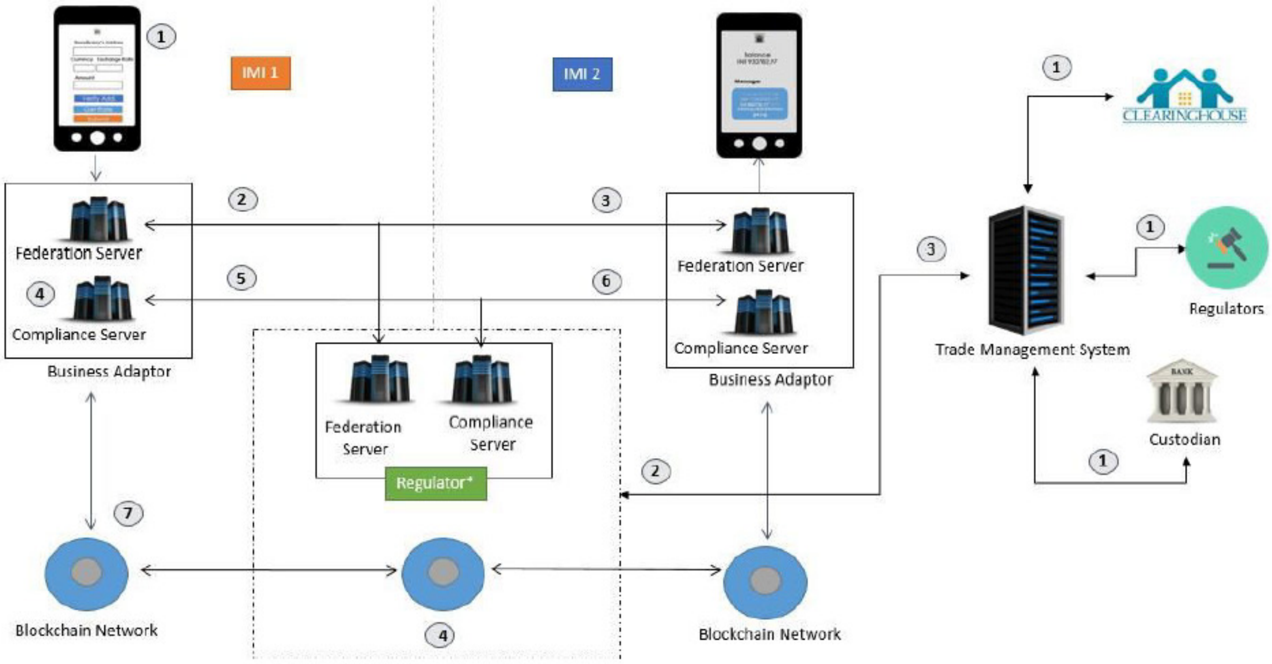
Milestones:

- Creating dedicated hosted infrastructure for trade & order management
- Setting up cluster databases for record management with loopback facilities
- Setting up procedural back testing environment for calibrating the trading algorithms with live trading system
- Creating sequencer for every securities to optimize the flow of orders in the trade matching engine
- Distributed ledger for reflecting the updated transaction records to every stakeholders in the trade life cycle process
- Decentralized ecosystem for trade management resulting in seamless and transparent accounting & ledger updating

Benefits:

- Increased efficiency and reduced downtime of Trade Processing System
- Discarding the opportunity for the data loss or redundant data
- Immutable ledgers for restricting errant orders placement in Trade Processing system
- Optimized OMS supporting higher order processing without creating any scope for process deadlock
- Instant reconciliation enabling faster settlement of trade processing
- Transparency in accounting will result in providing the exact cost of transactions and its associated charges for every stakeholders in the ecosystem

Process Framework



USE CASE 2: Insurance

Abstract:

Companies in financial sectors are exploring and experimenting innovative ways to execute transactions quicker for an enhanced customer service, ensure cost efficiency in its operations, and assure transparency to customers and regulators. With large volumes of data getting generated regularly owing to digitization of records, it becomes important for every organizations to effectively manage the security threats and achieve significant cost efficiencies. This is where Blockchain, with its promises of decentralized ownership, immutability and cryptographic security of data, is catching the attention of the C-suite executives. Multiple use cases are also getting explored across industries as everyone has started realizing the disruptive potential of this technology. This article will discuss how insurance & reinsurance firms are seeking out opportunities to harness the benefits of blockchain as well as key challenges to adopting this technology. Further, the article will highlight near-term practical applications for wblockchain and how to approach blockchain innovation.

Introduction to Blockchain

Blockchain can be described as a data structure that holds transactional records and while ensuring security, transparency, and decentralization. You can also think of it as a chain or records stored in the forms of blocks which are controlled by no single authority. A blockchain is a distributed ledger that is completely open to any and everyone on the network. Once an information is stored on a blockchain, it is extremely difficult to change or alter it.

Each transaction on a blockchain is secured with a digital signature that proves its authenticity. Due to the use of encryption and digital signatures, the data stored on the blockchain is tamper-proof and cannot be changed. Blockchain technology allows all the network participants to reach an agreement, commonly known as consensus. All the data stored on a blockchain is recorded digitally and has a common history which is available for all the network participants. This way, the chances of any fraudulent activity or duplication of transactions is eliminated without the need of a third-party.

Global Acceptance Of Blockchain – Diverse Industries Using Blockchain

The insurance industry is in the middle of an exciting transformation. Technology advancements coupled with changes in customer lifestyles and expectations are driving a hitherto unwitnessed level of innovation. Insurer's intent on realigning their business models with evolving customer needs have new allies in the form of blockchain and Internet of Things (IoT) that have the ability to change the course of the industry.

1. Government to record in a transparent way citizens' votes, or politicians' programs (for verifying if promises made have been kept) or to enable autonomous governance systems.
2. Intellectual property to certify the proof of existence and authorship of a document.
3. Internet to reduce censorships, by exploiting the immutability of data stored in the blockchain.
4. Finance to transfer money between parties without having to rely on banks.
5. Commerce to record goods' characteristics as well as their ownership, especially for luxury goods, thus reducing the market of counterfeit/stolen items.
6. Internet of Things (IoT) by exploiting smart contracts to automatically process data coming from sensors, in order to let intelligent machines interact with each other and autonomously take actions when specific situations occur.
7. Education to store information on qualifications acquired by learners.

Global Market Of Insurance Industry

The global insurance industry will grow more strongly than the global economy in 2018 and 2019. This year and next, we expect global premium to grow by more than €460bn in all. This is equivalent to average annual premium growth of 5.3% (in real terms, i.e. adjusted for inflation: 3.7%), whereas global GDP is expected to grow by only 4.9% (3.3% in real terms).
Life insurance, in particular, looks set to return to strong annual premium growth of 5.6% (3.9% in real terms) after a weak 2017. Property-casualty insurance is benefiting from the currently favorable economic environment. In this segment, we are expecting annual growth rates of close to 5% (3.3% in real terms). Emerging countries are the primary growth drivers, but somewhat stronger growth rates in high-volume industrialized countries are also contributing to this positive development.

The long-term outlook for the insurance industry is even more pleasing. In 2030, we expect premium volume to be close to €8tn – almost double what it is today.

Property-casualty insurance should benefit from the positive economic trend. In2018/2019, we expect global premium in this segment to grow at the same rate as the global economy – nominally by an average of 4.9% and in real terms by 3.3%.

Prospects are also good for the long-term outlook. In 2030, premium volume will most likely amount to around €7.9tn – almost double the volume of €4.2tn in 2017. Of the additional premium of €3.7tn by 2030, some €1.2tn is expected to come from China alone. Almost twothirds of this is likely to derive from life and health insurance business, with the rest coming from property-casualty insurance.

Despite these high growth rates, in 2030 the USA will still be the world's biggest insurance market, with a market share of almost 24%. China will move into second place, overtaking Japan.

Impact Of Insur-techs On Market Development

In saturated markets, i.e. the industrialized countries, InsurTech companies have grownsignificantly in importance over the past three to four years, but the high level of saturation there means that additional premium growth is not very likely.

We also think it unlikely that start-ups will squeeze out traditional insurance companies to any major extent, as capital requirements and regulations in the industrialized countries make it difficult for new players to acquire an insurance licence. Most InsurTechs currently operate as digital brokers or in partnerships with traditional insurance and reinsurance companies. Although InsurTechs, in their role as digital brokers, constitute competition for traditional sales channels – which could lead to a shift in market shares – the overall impact on premium volume across the market is most likely to be quite small in the industrialized countries.

In developing countries and emerging markets, however, InsurTechs play a different role. FinTechs in general are expected to see major growth in the emerging markets. According to studies1, in these markets alone, FinTechs could generate additional GDP of US$ 3.7tn by 2025, as mobile phones facilitate access to financial accounts. By 2025, low per-capita-income countries like Ethiopia, India or Nigeria could augment their GDP by 10–12% through FinTechs, while middle-income countries like Brazil, China or Mexico could still add 4–5%.

These possible developments also hold additional potential for the insurance industry. This is because rising per-capita income also leads to increased insurance penetration; this is currently still low in emerging markets, especially in developing countries, and offers tremendous catch-up potential.

Blockchain In Insurance:

Traditionally, the insurance sector has been slow to adopt new technology and is often the last financial sector to incorporate any technological evolution, and Blockchain is no exception. However, it is important to note that the concept of Blockchain is being looked at as one of those rare innovations, which has the potential to disrupt the insurance industry much ahead of the trend. The insurance industry is all about managing financial risks and includes a high volume of financial transactions on a day-to-day basis. This makes the industry vulnerable to a potentially large number of intrusions, attacks, and fraudulent transactions.

The insurance space is also highly complex with composite contracts between multiple stakeholders that require a large processing potential. In this thought paper, we will review some of the current challenges in the insurance space, and analyze if Blockchain can provide the much-needed solutions.

1. Fraud Detection and Risk Prevention - By moving insurance claims onto an immutable ledger, blockchain can help eliminate common sources of fraud in the insurance industry.

2. Claims Prevention & Management - Within claims prevention, new data streams can enhance the risk selection process by combining location, external risk and analytics. A distributed ledger can enable the insurer and various third parties to easily and instantly access and update relevant information (e.g., claim forms, evidence, police reports and third-party review reports).

3. Distribution & Payment Models - Global insurers can use blockchain to cut asset management costs by reducing the hedging fees they pay to protect themselves from currency fluctuations in international transactions.

4. Reinsurance - By securing reinsurance contracts on the blockchain through smart contracts, the blockchain can simplify the flow of information and payments between insurers and reinsurers.

Benefits

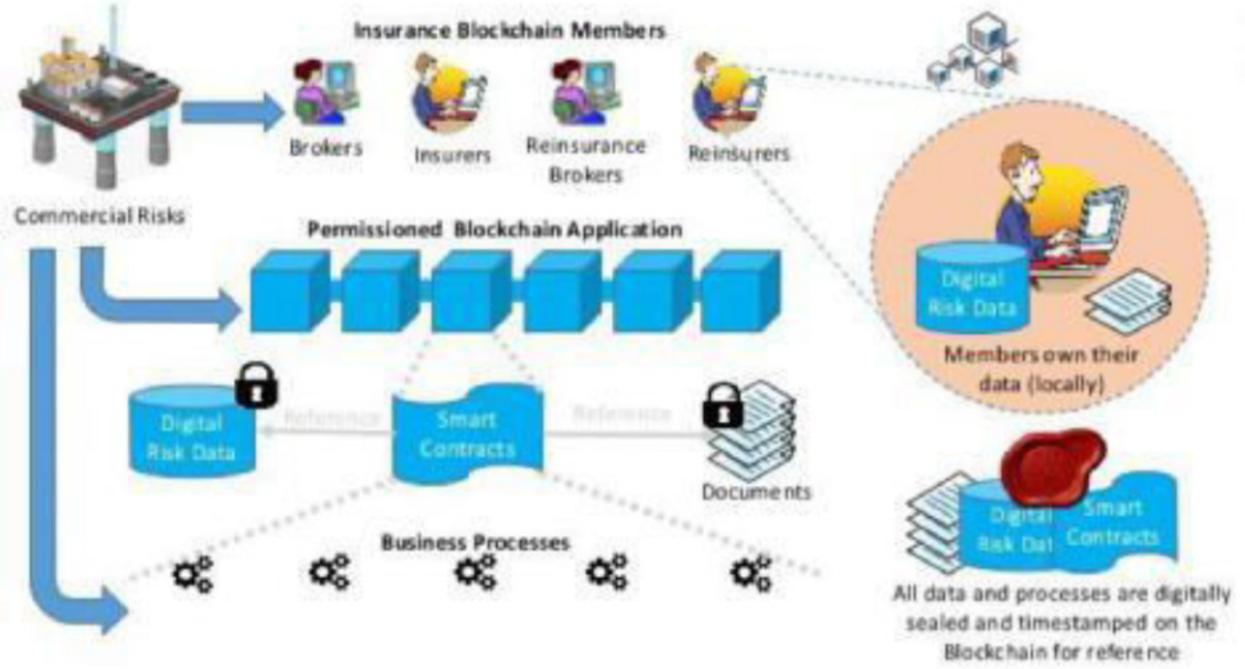
With IQONIQ Blockchain, insurance companies can convert multiple policies into "smart contracts" giving a single, consolidated view of policy data and documentation in real time. The solution allows visibility into coverage and premium payments, delivering automated notifications to network participants following payment events.

A transparent blockchain solution allowing multiple companies to collaboratively assemble relevant records can streamline claims recovery. Its shared ledger capabilities can help insurers agree on claims, build trust that evidence is being shared and improve the overall customer experience.

IQONIQ Blockchain can be the vital link across a vast ecosystem of third-party administrators and service provider networks. Its shared ledger transparency can help employers reduce errors resulting in improved claims processing, better provider management and lower operational expense.

IQONIQ Blockchain helps ensure contract certainty and improve risk-handling capabilities — from managing contracts among reinsurers to maintaining shared accounts and managing claims payments. With transparency across the reinsurance value chain, blockchain can eliminate the need for participating companies to regularly reconcile their reinsurance accounts.

Proposed Architecture


Insurance Blockchain Members
Brokers
Insurers
Reinsurance Brokers
Reinsurers
Commercial Risks
Permissioned Blockchain Application
Digital Risk Data
Reference
Smart Contracts
Reference
Documents
Business Processes
Digital Risk Data
Members own their data (locally)
Digital Risk Dat
Smart Contracts
All data and processes are digitally sealed and timestamped on the Blockchain for reference

USE CASE 3: Gaming Industry

Overview Of The Current Gaming Industry

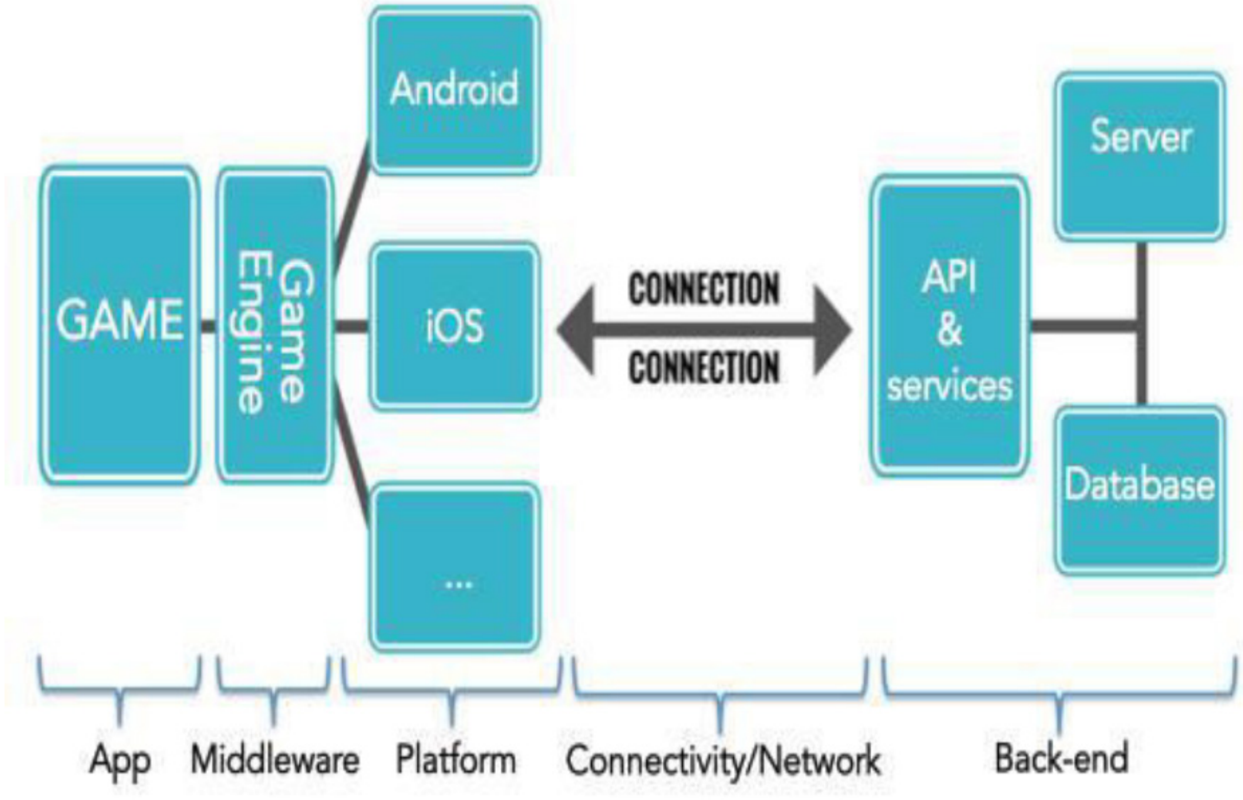
The computer and video game industry has grown from focused markets to mainstream. They took in about US $9.5 billion in the US in 2007, 11.7 billion in 2008, and 25.1 billion in 2010. The international games industry will hit an astronomical $108.9 billion in 2017, and continue to grow to $128.5 billion by 2020, market analyst firm predicts.

Global Games Market Report delivers some fascinating and mind-blowing predictions for the games industry, especially for mobile gaming. According to the report, 2.2 billion gamers are expected to a staggering $108.9 billion in games market revenues in 2017 alone, which represents a year-over-year increase of 7.8% (or $7.8 billion). The report also forecasts that digital will account for 87% of total revenues, or $94.4 billion, but remember that digital includes all facets of digital gaming--not just game sales: subscriptions and subscription- based services, micro transactions/DLC/add-ons, live services, freemium mobile games, full game downloads across all platforms (PSN, Xbox LIVE, Steam, GoG, etc), and other content.

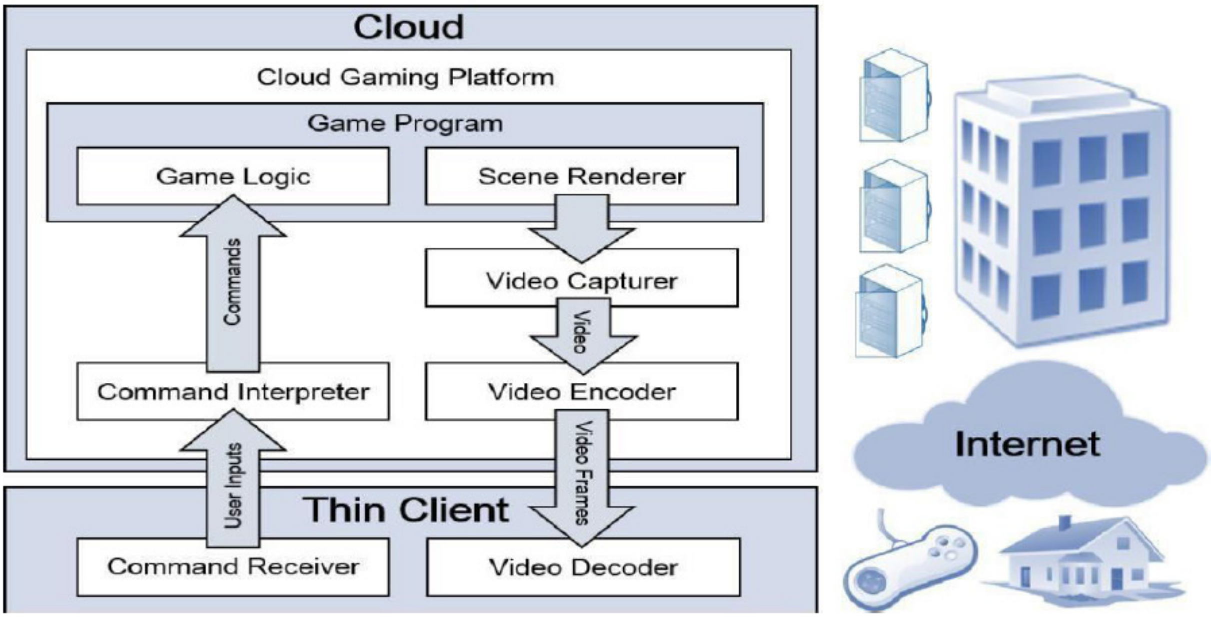
The analyst firm says that mobile is by far the most lucrative segment in global gaming, and is projected to make up more than half of the games market by 2020. In 2017, mobile gaming (smartphone and tablets) is expected to hit $46.1 billion, or 42% of all revenues. This represents a 19.3% year-over- year increase. As mobile is digital-only, it's one of the major prime movers for the digital sector, and crosses over between full game downloads and in-game purchases.

Current Gaming Architecture:

Online play comes in several forms, such as session-based multiplayer matches, massively multiplayer virtual worlds, and intertwined single-player experiences.



In the past, games using a client-server model required the purchase and maintenance of dedicated on-premises or co-located servers to run the online infrastructure, something only large studios and publishers could afford. In addition, extensive projections and capacity planning were required to meet customer demand without overspending on fixed hardware. With today's cloud-based compute resources, game developers and publishers of any size can request and receive any resources on demand, helping to avoid costly up-front monetary outlays and the dangers of over or under provisioning hardware.



Introduction to Blockhain:

Blockhain technology, also known as distributed ledgers, has a number of potential use cases within the wealth and asset management life cycle. Distributed ledgers are highly flexible; once implemented, they can be used to remove friction from the client onboarding process, streamline management of model portfolios, speed the clearing and settlement of trades, and ease compliance burdens associated with anti-money laundering (AML) and know your customer.

The result is elimination of redundant functions, reduced operational expenses and increased opportunities to enhance the client experience. While Blockhain technology is unlikely to replace current systems, it may be used to reconcile information across them or enable new infrastructure for new markets and products.

By extension, these concepts can expand to broader applications, such as rollovers, trusts, estates, gaming, wealth & asset management, insurance and other transactions where assets are moved between parties or contracts are executed. A distributed ledger supports the validation and execution of a transaction in near real time. The client experience is enhanced and the process streamlined, and costs are reduced.

Blockhain in Gaming

The collision of Blockhain technology and gaming holds great promise for the growth of both industries. Innovations in the budding field of Blockhain gaming have pushed the limits of non-fungible assets and are poised to keep supplying novel developments in other areas like scalability.

Gamers were some of the early adopters of cryptocurrencies as they were already familiar with many in-game virtual currency models and saw the benefits of integrating cryptocurrency networks into the domain. The eSports industry is booming, and it is only a matter of time before its rise is meaningfully coupled with cryptocurrency payment systems and decentralized models.

Driving Factors:

- Continuous and Parallel Gaming Universes
- Using encrypted Blockhain ledgers to store digital assets will guarantee safe storage of game items, permanently.
- Items could be customized by players allowing them to re-create and trade upgraded assets with unique properties.
  The ability to take micro-payments quickly, easily and affordably will give
- developers new ways to monetize their games.
- Democratization using voting features will make the Gaming World a Better Place
- Increasing the functional, economic & social value of Gaming Items
- The public Blockhain ledger will allow game developers to create rare virtual items, and prove their scarcity.
- Decentralized payment gateways will allow users to make payments with a fraction of the fees that are currently paid to credit card companies
- Drastically reducing fraud and ending lost revenue

IQONIQ Strikers- White Label Blockhain in line with IQONIQ Blockhain:

A ledger that drives the immutability, trust and transaction capability between the gaming platforms, the gaming Exchange and the gaming Wallet. The Blockhain network tracks the movement of the native gaming Tokens within the gaming ecosystem. The choice to use public Blockhain technology for the ecosystem player interface and an “off-chain” technology choice for the gaming platform interface is based primarily on the need for higher transaction speeds in the backend processes.

Use Case: Decentralized Gaming Platform & Own Crypto-Collectibles

Hurdles:

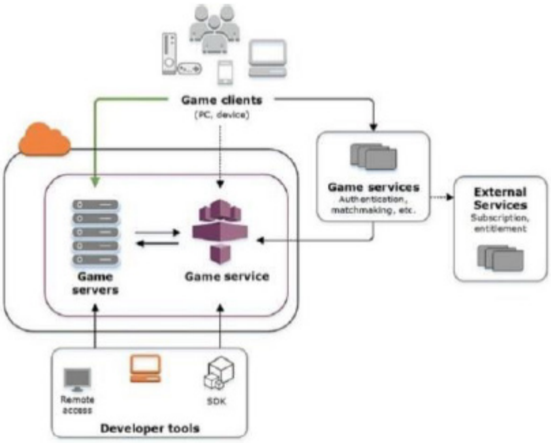
- With the emergence and trend of VR, AR, A.I., and smart technology, the dependency on new gaming consoles are also emerging. Thus it becomes challenging for gamers to keep on upgrading their gaming consoles with match to the gaming specifications.
- Oversaturated Market for Gamers, Developers and the Content providers because of centralized ecosystem is hampering the functional, economical & social value of gaming industry.
- Lack of incentivization or disproportionate rewarding mechanism for gamers resulting in reduced engagement.
- Lack of accessibility for gamers due to lack of universal index or recommendation generator of games.

The Ultimate Encounters

Setting up dedicated cloud infrastructure for gamers to experience boundless arena of games.

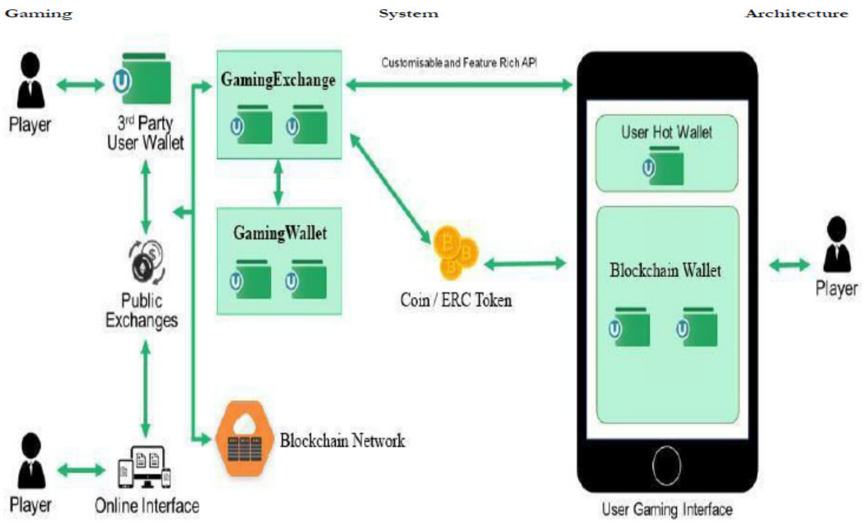
It will provide with the universal index of games that gamers would prefer to play. It will also minimize the overwhelmingly stressful thought of gamers over the gaming console or other hardware dependencies that reduces the level of gamer’s participation in the gaming platform.

- IQONIQ Blockhain assists in setting up the cloud infrastructure for Gaming Providers
- IQONIQ Blockhain assists in setting up security configurations such as IAM, ports routing, IP Tracking, DDoS & Malware protection, Wallet Security using AES 256 & Md5 encryption standard & SSL Configuration



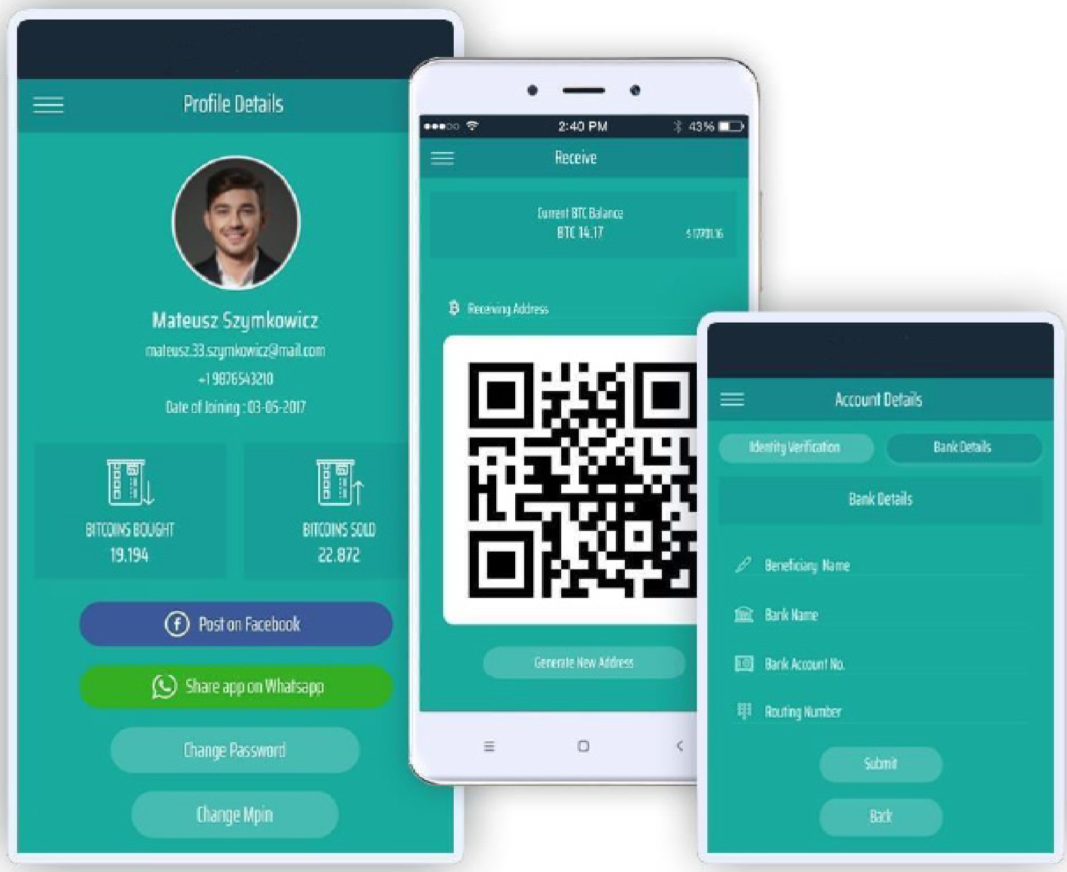
Setting up of Decentralized Gaming Ecosystem for Gamers to participate

- Assisting in setting up Blockhain network in line with IQONIQ Blockhain
- Assisting in setting up Blockhain network node. The Blockhain core is compatible with other leading Blockhain networks. Thus supporting other coins and ERC tokens to be easily integrated with the network
- Setting up the Blockhain wallet
- Setting up compliance server for managing KYC details
- Assisting up API layer for Blockhain network to integrate with the gaming platform
- Creating Network performance dashboard for monitoring the performance of the Blockhain nodes
- Creating Blockhain IO for monitoring the transactions ledger



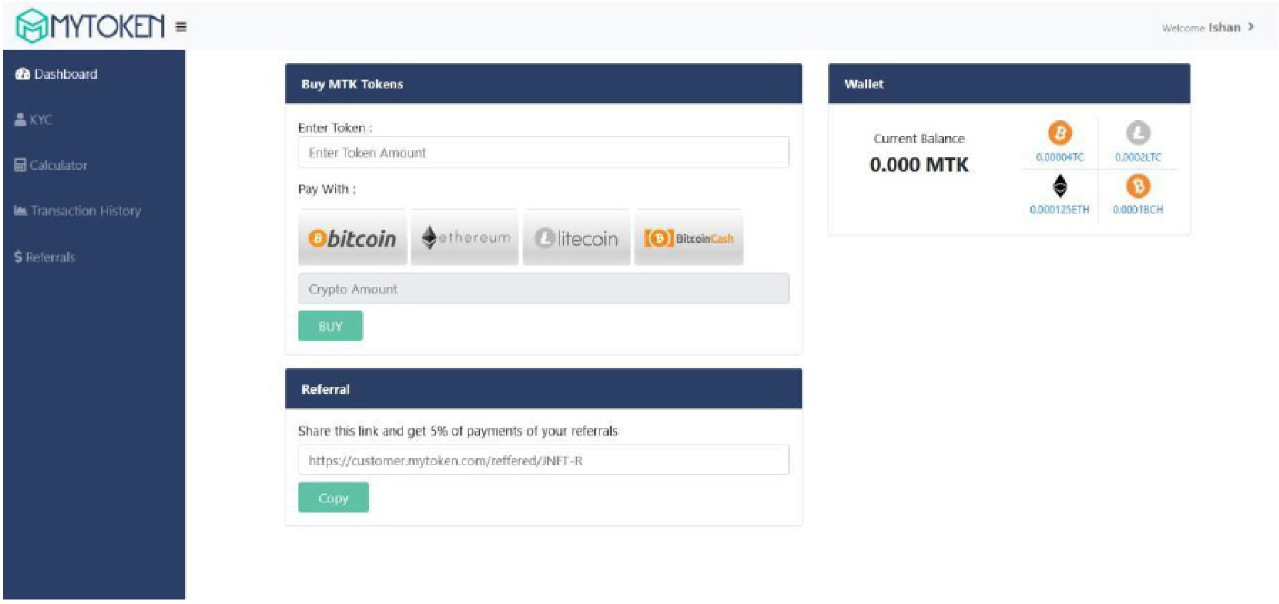
Create / Integrate your Gaming Token

- Create the native digital asset on the Blockhain or integrate your created ERC token on the network
- Create root account or integrate the contract's owner account with the Blockhain wallet
- Flush root accounts with initial number of coins / tokens in wallet
- Exposing API's for Sending and Receiving Coin / Token



ICO

- Integrating the web-wallet with the Blockhain wallet for coin distribution
- Creating Referals and Promotional programs for ICO distribution
- Assisting in dissemination of coins / tokens to ICO participants
- Assisting in listing of coins / tokens in leading & affiliate exchanges



Buy MTK Tokens

Enter Token : Enter Token Amount

Pay With :

Crypto Amount

Wallet

Current Balance
0.000 MTK

0.00004TC
0.0002LTC
0.000125ETH
0.0001BCH

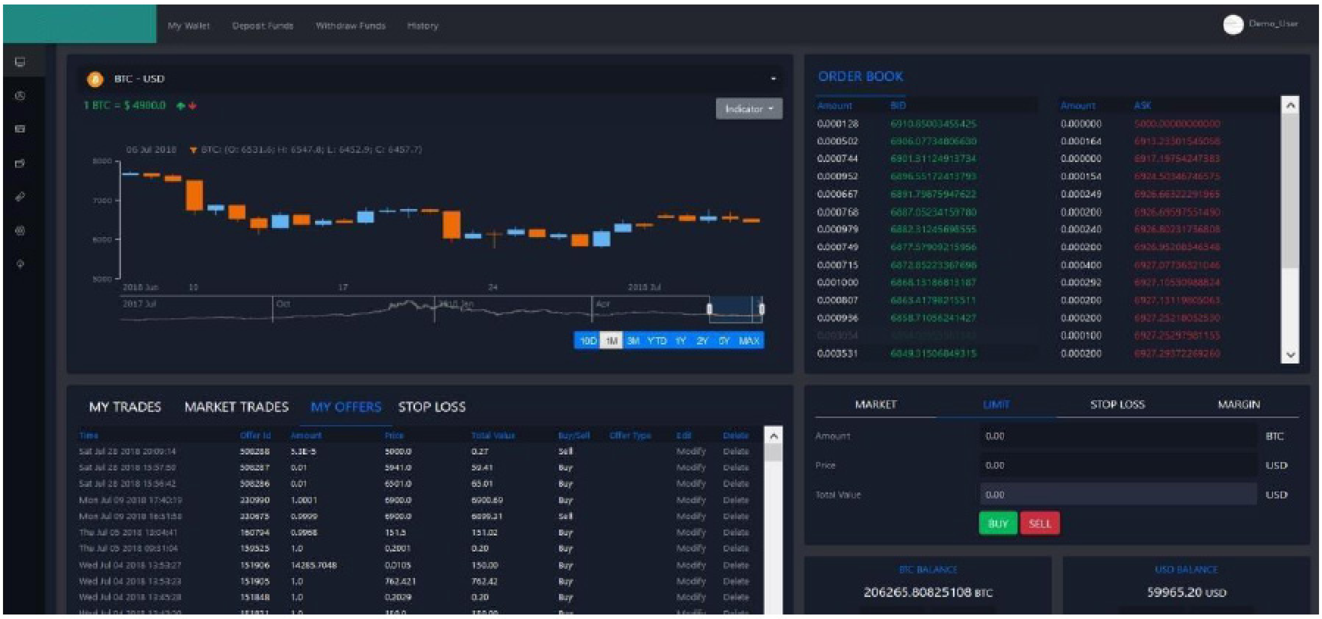
Referral

Share this link and get 5% of payments of your referrals

https://customer.mytoken.com/reffered/JNFT-R

Gaming Exchange and Integration with Gaming Platform

- Setting up the white label gaming exchange
- Pairing with base currency like Fiat (USD) or Crypto (BTC /ETH /Crypto of your choice) for coin or token to trade
- Setting up features like Market, Limit, Stop Orders, Margin for placing orders and increasing the volume of trade
- Setting up of Admin Console to manage gamers, developers and other stakeholders
- Setting up Multiple Access Control levels for Admin console
- Setting up Admin Panel for managing & creating trading pairs
- Setting up incentivization and reward modules for gamers
- Exposing exchange API's for integrating with the Gaming Platform



1 BTC = $ 4900.0

ORDER BOOK

Amount	BID	Amount	ASK
0.000128	6910.85003455425	0.000000	5000.000000000000
0.000502	6906.07734806630	0.000164	6913.23301545058
0.000744	6901.31124913734	0.000000	6917.19754247383
0.000952	6896.55172413793	0.000154	6924.50346746575
0.000667	6891.79875947622	0.000249	6926.66322291965
0.000768	6887.05234159780	0.000200	6926.69597551490
0.000979	6882.31245698555	0.000240	6926.80231756808
0.000749	6877.57909215956	0.000200	6926.95208546548
0.000715	6872.85223367696	0.000400	6927.07736321046
0.001000	6868.13186813187	0.000292	6927.10530988824
0.000807	6863.41798215511	0.000200	6927.13119805063
0.000936	6858.71056241427	0.000200	6927.25218052530
0.003054	[illegible]	0.000100	6927.25297981155
0.003531	6849.31506849315	0.000200	6927.29372269260

MY TRADES MARKET TRADES MY OFFERS STOP LOSS

Time	Offer Id	Amount	Price	Total Value	Buy/Sell	Offer Type	Edit	Delete
Sat Jul 28 2018 20:09:14	506288	5.3E-5	5000.0	0.27	Sell		Modify	Delete
Sat Jul 28 2018 15:57:50	506287	0.01	5941.0	59.41	Buy		Modify	Delete
Sat Jul 28 2018 15:56:42	506286	0.01	6501.0	65.01	Buy		Modify	Delete
Mon Jul 09 2018 17:40:19	230990	1.0001	6900.0	6900.69	Buy		Modify	Delete
Mon Jul 09 2018 16:51:58	230875	0.9999	6900.0	6899.31	Sell		Modify	Delete
Thu Jul 05 2018 12:04:41	180794	0.9968	151.5	151.02	Buy		Modify	Delete
Thu Jul 05 2018 00:31:04	150525	1.0	0.2001	0.20	Buy		Modify	Delete
Wed Jul 04 2018 13:53:27	151906	14285.7048	0.0105	150.00	Buy		Modify	Delete
Wed Jul 04 2018 13:53:23	151905	1.0	762.421	762.42	Buy		Modify	Delete
Wed Jul 04 2018 13:45:28	151848	1.0	0.2029	0.20	Buy		Modify	Delete
Wed Jul 04 2018 13:43:20	151831	1.0	150.0	150.00	Buy		Modify	Delete

MARKET LIMIT STOP LOSS MARGIN

Amount 0.00 BTC

Price 0.00 USD

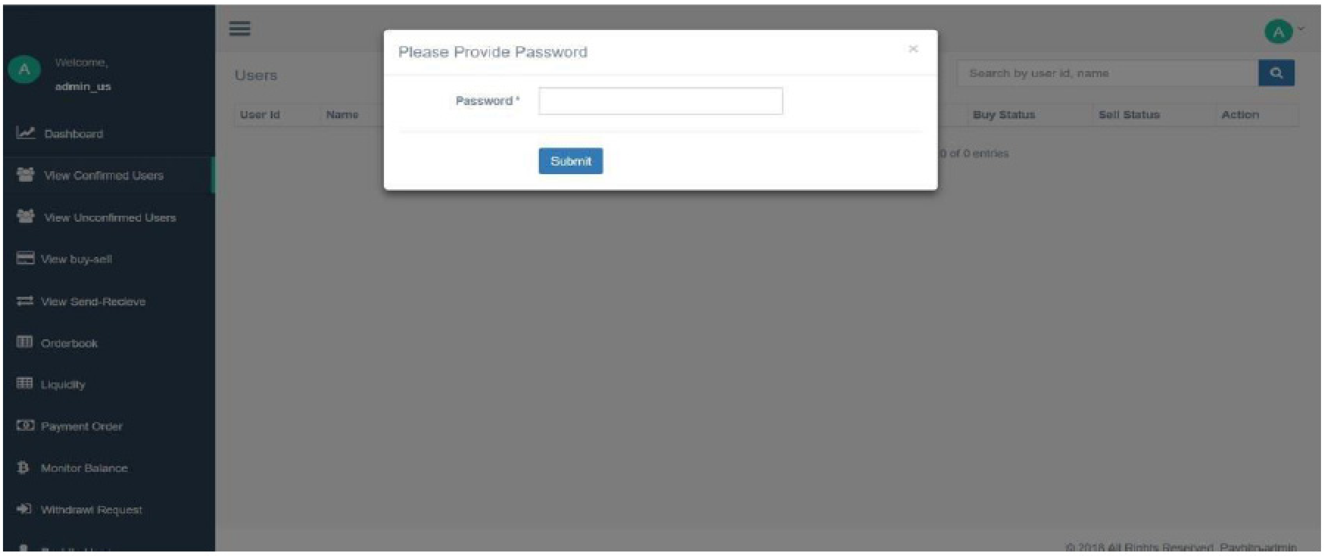
Total Value 0.00 USD

BUY SELL

BTC BALANCE
206265.80825108 BTC

USD BALANCE
59965.20 USD

White Label Exchange for Gaming



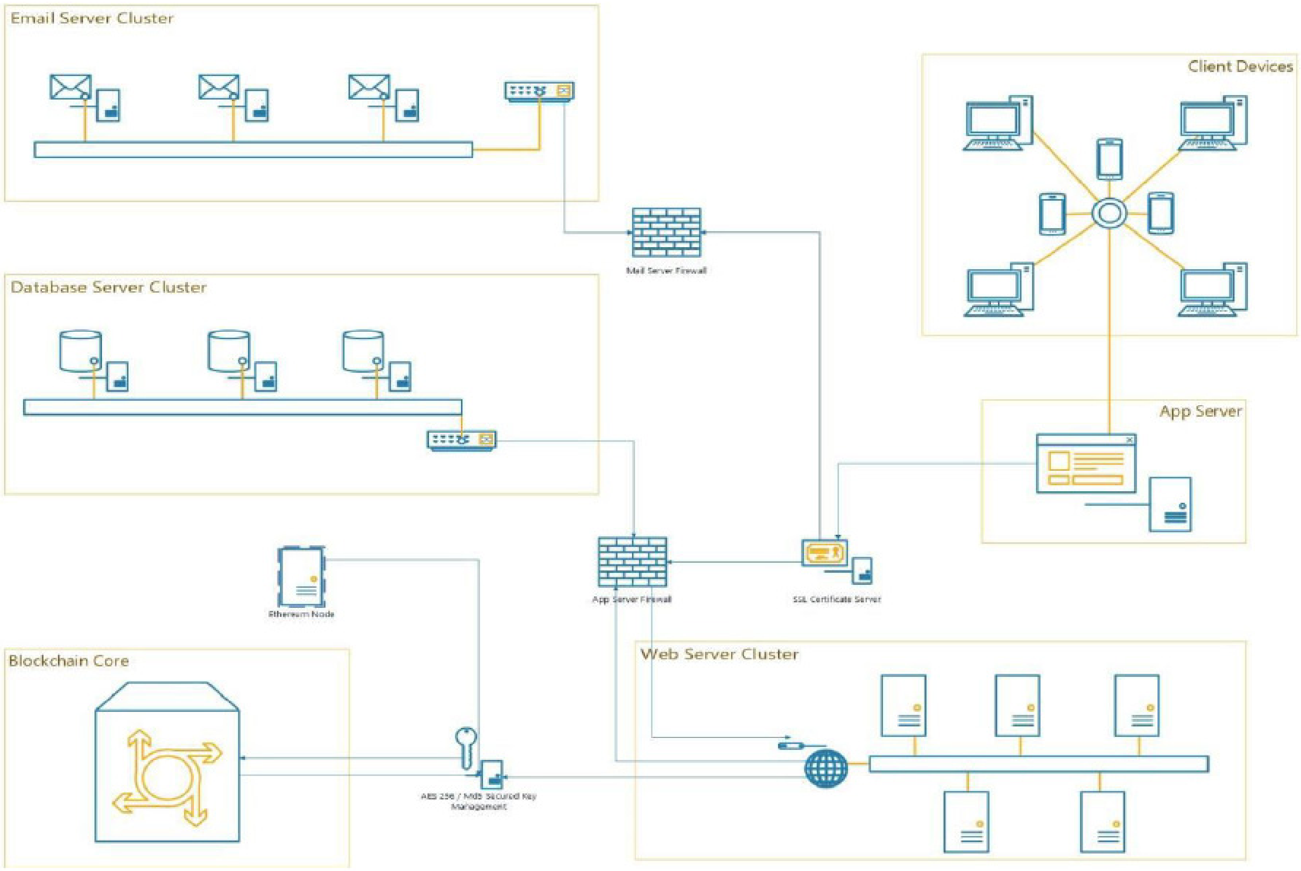
Please Provide Password

Password *

Submit

Exchange Admin Console for Backend Operations

STRATEGY - The Network Playground



Benefits

- Simple, Secure and Fun to use
- Trustworthy mechanism for storing and moving funds
- Support for ICO
- Manage transaction costs using Admin Console
- Support for cold wallet, hot wallet and hardware wallet integration with multi-signature security
- Compatible with ERC tokens and other Blockhain coin integration
- Customized module for gaming payment options
- Support for customized KYC and IAM to honor regulatory norms of

- Dedicated infrastructure for supporting 99.9% uptime
- Manage transaction costs using Admin Console
- Exchange margin features enabled for higher transaction
- Well defined access controls to manage funds
- Full functionality accessible from smart devices
- Support for coin / token dissemination to ICO participants
- Easy to integrate with third party exchanges for liquidity management
- Support for third party wallet integration
- Support for single Sign-In for Exchange, Admin console and gaming interfaces
- Customized module for incentivization and reward mechanism
- Support for high value concurrent transaction
- Customized Reporting for detailed user information's, transaction details, promotional details and network details
- Integrated security measures such as AES 256 & Md5 encryption support, DDoS & Malware protection, Firewall, Port Management, IP Tracking & Auditing enabled, BIP 32, Firebase & SegWit
- Supports for both Fiat and Crypto base to trade

USE CASE 4: Trade Finance

Trade Finance Landscape:

Providing delivery and payment assurance to buyers and sellers, trade finance by banks and other financial institutions is a vital function in the international commerce industry. It helps to close the gap concerning trade cycle funding for the parties as mentioned earlier. Only a robust financial mechanism and easy availability can ensure the growth and sustenance of the $16 trillion international trade market. This particular facet of trade finance has earned it the accolade of being the fuel for global commerce. It must be mentioned here, that, trade participants can be vulnerable to business risks and uncertainties that has its roots deeply embedded in several factors that include variance and fluidity in trade regulations and requirements across geographies, the operational and logistical complexities that arise when a large number of entities interact and process inefficiencies. According to a recent survey by the International Chamber of Commerce, there has been seen, an increasing trend in litigation and fraud related to trade financing over the last few years.

An example of such a mishap would include the trade and receivable financing fraud which includes the $1.1 billion lawsuit against Citigroup that was a result of financing falsified receivables. Besides, the loss of hundreds of millions of dollars to various banks in the Qingdao port metal financing fraud involving multiple invoices secured against the same collateral also offers a comprehensive view of how these problems can affect different sectors of International commerce.

The other challenges include payment and delivery delays due to process overheads along with a lack of insight into the movement of goods. It also takes a vast effort and engagement to manage counterparty due diligence and contractual compliance processes.

Have been taken the instance of banks, it can be seen that these omnipresent obstacles are bound to increase risks and costs that would lead to unfavorable financing terms, especially for small businesses. An estimated 60% of trade finance applications from small and medium-sized enterprises (SMEs) are rejected by banks globally. Besides, as put by the Asian Development Bank, the total value of unmet trade financing demand can be estimated at a whopping USD$1.6 trillion. In addition, a study conducted by International Finance Corp. reflects the financing gap for global micro, small and medium-sized enterprises at USD$2.6 trillion. Adversely affecting growth in global commerce, these risks and inefficiencies have limited the size of the trade finance market, which currently stands at $4 trillion to $5 trillion.

Pain Areas Of Trade Finance

1. **MANUAL CONTRACT CREATION** –The financial agreement provided by the importer has to be first reviewed by the import bank manually and then separately it has to send financials to the correspondent bank

2. **INVOICE FACTORING** - Invoices are used by exporters to achieve short-term financing from multiple banks which adds additional risk in the event of the delivery of goods fails

3. **DELAYED TIMELINE** - Multiple checks by intermediaries and numerous communication points significantly delays the shipment of goods

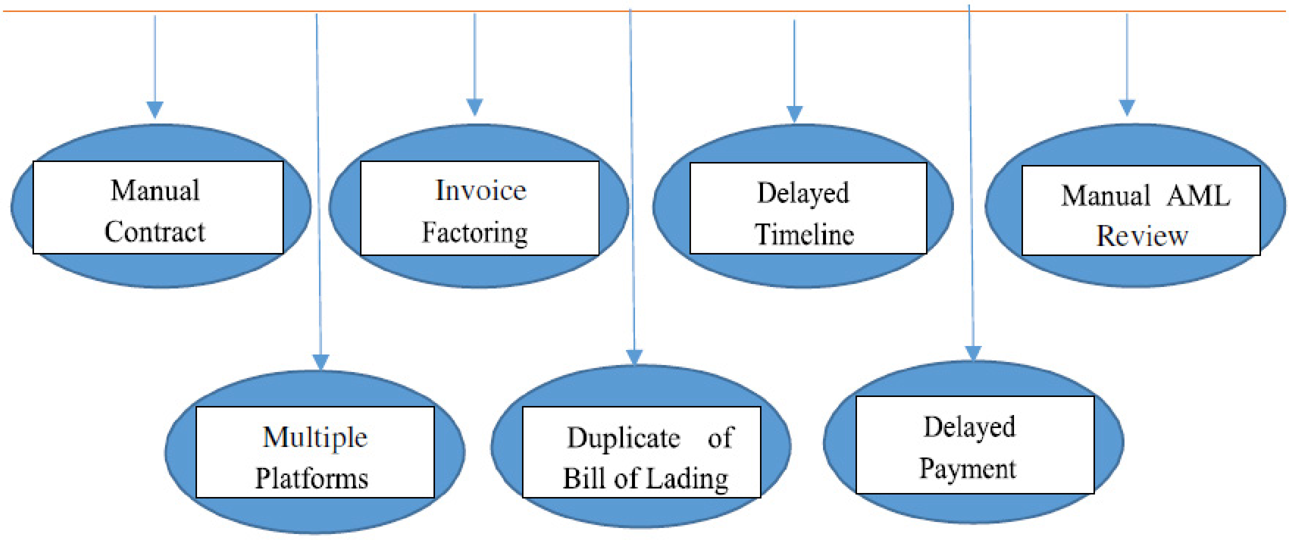
4. **MANUAL AML REVIEW** – A rigorous chore, the export bank must manually conduct AML checks using the financials provided by the import bank

5. **MULTIPLE PLATFORMS** - The chances of miscommunication is often and the propensity for fraud is high, since each party across countries operates on different platforms

6. **DUPLICATION OF BILL OF LADING** - Inability of banks to verify the authenticity leads to bills of lading being financed multiple times

7. **DELAYED PAYMENT** - Prior to the disbursement of funds to the exporting bank, multiple intermediaries must verify that funds have been delivered to the importer as agreed upon previously.

Pain Areas of Trade Finance



Blockchain In Trade Finance

The new sought-after goal for businesses in the contemporary trade scenario is a better connected, highly automated and far more open infrastructure, which will also enable more efficient trade finance solutions for customers. An answer to this can be achieved by incorporating a set of trusted and permissioned interactions between corporations, B2B networks, service providers and other financial institutions.

The entrance of 'smart contracts' will help businesses to automatically trigger commercial actions based on defined criteria. Further boosting efficiency by streamlining processes, this will also ease time and the cost of transactions. The improved traceability which an indelible audit trail provides is one of the main benefits of blockchain application in trade finance. Assets are automatically checked, owing to the new verification levels. Thus, businesses can reduce fraud and compliance costs by ensuring that each transaction is recorded sequentially and indefinitely. Blockchain has the ability to allow simple, secured share trade-related data between different financial institutions that nonetheless, enhances every aspect of security. Using independently verified complex cryptography, blockchain verifies every transaction that is verified within the network.

Unprecedented levels of trust can be injected in the trade and finance systems very easily by including some much-needed commercial transparency to the mix that would thrive on solving and eliminating problem s such as delays and sharing data between parties. Notions like authenticity, transparency, and simplicity are rapidly becoming the new language in the trade finance market.

Blockchain Approach

1. The agreement of sale between the importer and exporter is shared with import bank using a Smart Contract on the Blockchain at the very moment of purchase

2. The import bank will have the capability to review purchase agreement, draft terms of credit and submit an obligation to pay to the export bank in real time using Blockchain technology

3. A Smart Contract will be generated on the Blockchain to cover terms & conditions and lock-in obligations as soon as the Export bank will review the provided payment obligation and give the approval.

4. The exporter will digitally sign a Blockchain-equivalent letter of credit within the Smart Contract to initiate shipment after receiving the obligations.

5. Goods will be inspected by 3rd parties and the customs agent in the exporting country - with all the involved entities providing their respective digital signature of approval on the Blockchain Smart Contract

6. Blockchain offers a quick traceability during transit where goods will be transported from Country A to Country B

7. The importer will digitally acknowledge receipt of goods and trigger payment upon the confirmed delivery

8. Using provided acknowledgment, Smart Contract will enable the Blockchain network to automate payment from importer to exporter

Benefits

1. Real Time Review: Blockchain reduces the initiation time of the shipment as financial documents linked and accessible through Blockchain are reviewed and approved in real time

2. Transparent Factoring: Invoices accessed on Blockchain provide a real-time and transparent view into subsequent short-term financing

3. Disintermediation: Banks facilitating trade finance through Blockchain do not require a trusted intermediary to assume risk, eliminating the need for correspondent banks

4. Reduced Counterparty Risk: Bills of lading can be tracked through Blockchain, eliminating the potential for double spending

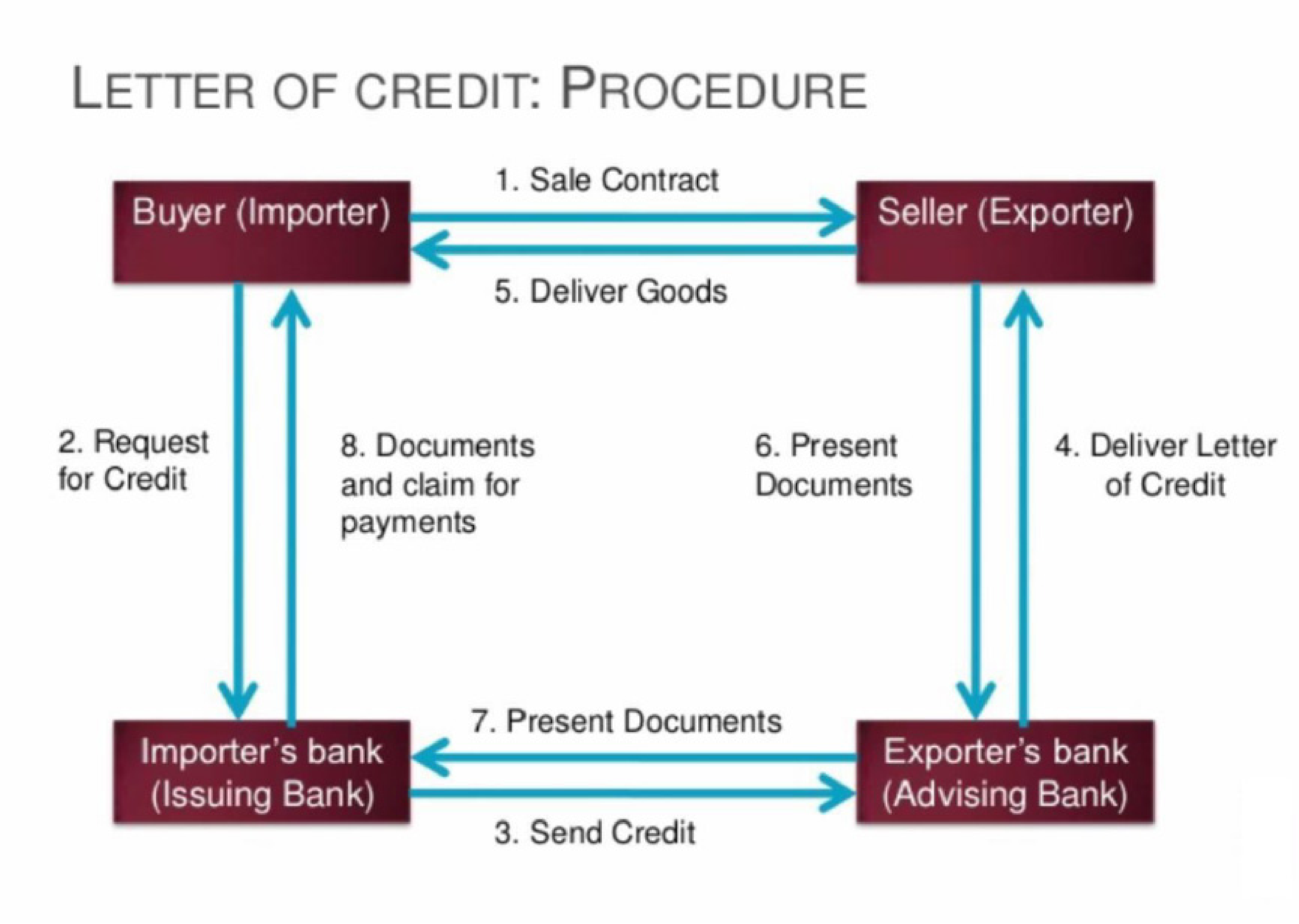
5. Decentralized Contract Execution: As contract terms are met, status can be updated on Blockchain in real time, reducing the time and human resources required to monitor the delivery of goods

6. Proof of Ownership: The title available within Blockchain provides transparency into the location and ownership of the goods

7. Automated Settlement and Reduced Transaction Fees: Contract terms executed via Smart Contract eliminate the need for correspondent banks and additional transaction fees

8. Regulatory Transparency: Regulators are provided with a real-time view of essential documents to assist in enforcement and AML activities

Architecture



LETTER OF CREDIT: PROCEDURE
Buyer (Importer)
Seller (Exporter)
1. Sale Contract
5. Deliver Goods
2. Request for Credit
8. Documents and claim for payments
6. Present Documents
4. Deliver Letter of Credit
Importer's bank (Issuing Bank)
Exporter's bank (Advising Bank)
7. Present Documents
3. Send Credit