PROJECT SUMMARY V1.6

DRAFT

Disclaimer:

The purpose of this document is to present the Wire project to interested parties and to provide them with sufficient information to enable them to determine whether they wish to be further involved in the development of the project. In no way should this document be construed to constitute a prospectus or solicitation for investment. This document has not been prepared in accordance with any regulation or laws of any jurisdiction, and any interested parties should satisfy themselves that under the laws of their relevant jurisdiction they are legally able to engage or invest in such a venture.

This is an initial overview of the project. The contents of this paper are liable to change as is the direction, scope, progress, and prospects of the project. This paper does not form a contract or agreement in any way.

Copyright Statement:

Executive Summary

The Wire solution provides a billing, payment, and subscriber contracting system for Mobile Network Operators (MNO) using de-centralised smart contracts on a blockchain platform. Abstracting the contract and billing mechanism from the operator's traditional business support systems and moving it to a blockchain platform provides huge benefits for MNOs such as contract flexibility and transparency, additional security, cost reduction, and scalability.

There are over 7 billion global mobile service subscribers generating trillions of calls, messages, and data sessions annually. Servicing this demand whilst attempting to offer compelling propositions in a highly competitive market environment, is an on-going challenge for all Mobile Network Operators (MNO). In many cases, MNOs are constrained by ageing billing systems or traditional ways of forming a contract with the subscriber.

Legacy mobile business support systems were usually built around the ability to deliver a simple handset and combi tariff, generate a regular paper or electronic bill for the subscriber, and to take payment via a direct debit or payment card. Now though, there is an increasing need for MNOs to offer different products, to provide pricing and payment flexibility, and to provide a better and more immediate experience for the subscriber.

The Wire solution will lever the innate ability of blockchain technology to provide a secure and de-centralised environment, and the power of electronic smart contracts, to provide an alternative billing, payment, and subscriber contracting architecture for Mobile Network Operators (MNO).

The Wire Solution consists of three core elements:

The Wire Smart Contract – An electronic smart contract in the blockchain which contains the contract information such as contract term, products & tariffs, pricing, subscriber information.

The Wire Subscriber App – A mobile (or desktop) app through which a subscriber can view their bill and pay for services

The Wire Operator Control Centre – A portal through which the Operator can view and interact with the smart contract, communicate with subscribers, and write usage information into the smart contract.

The Wire solution aims to improve the experience in many key use cases in which Mobile Network Operators currently experience constraints, find difficult to deliver change, or which require significant overhead costs to service. The initial use cases for consideration have been drawn from the Project Team's extensive combined experience of working with MNOs. Further use applications for the solution will be developed as the project progresses and the team will seek MNO

partners to gather further insight into potential operational ‘pain-points’.

There are real-world problems which can potentially be resolved using blockchain and smart contract technologies, and there is huge potential for a solid business to be built from the Wire project. One of the key strengths of Wire is that it has been started by identifying tangible and immediate business problems that it believes can be addressed by current or forthcoming technology, rather than build a ‘template’ platform and then search for viable use cases.

The project is in the start-up phase and is now seeking investment and support to grow the team and continue to explore and build the solution.

Contents

39. Introductory Section

39.1. Purpose and status of Document

This document is an initial overview of the Wire intended to give potential partners and investors an overview of the objectives and possibilities of the project. It will describe the business problems that exist in the mobile telecoms sector which could be addressed by the application of blockchain technology. It will outline some of the use cases that the wire solution could support and provide a high-level summary of the Wire architecture. Finally, it will give a rough idea of the delivery plan for the Wire project

This is a 'living document' which will change and expand. At this early stage it cannot contain the full detail of the project; much of this will be defined as the project progresses and in consultation with project partners. The vision is that this document will evolve into a whitepaper which can be used by investors to assess the project in further funding stages.

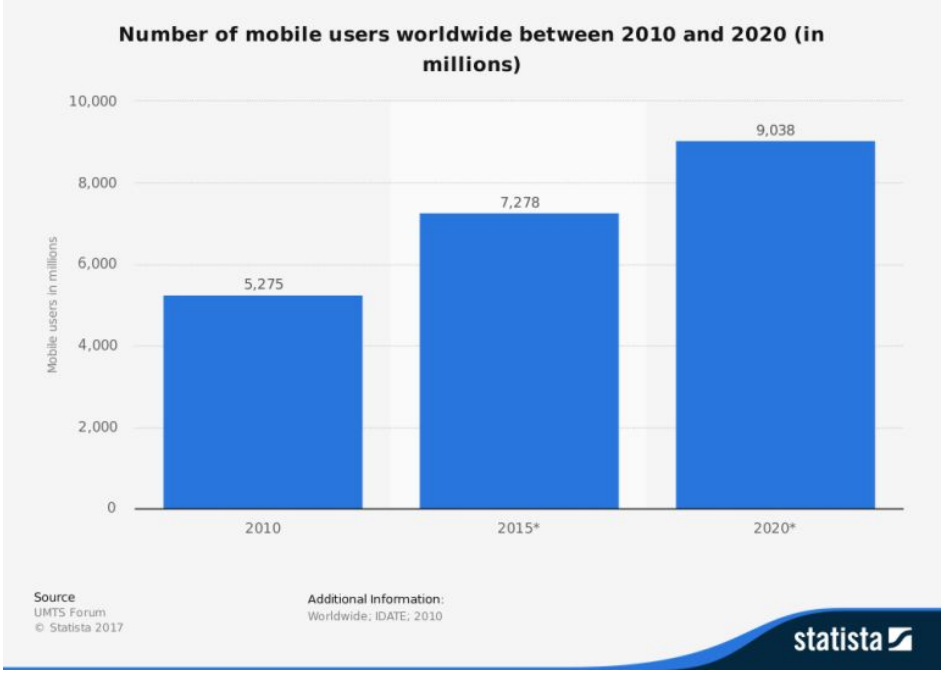
The document does seek to demonstrate that the fundamental market insights, project use cases, and solution mechanisms are sufficiently sound to present a robust basis for a positive investment and engagement decision.

39.2. The Market Size

The global mobile telecoms market continues to grow. In developed markets where handset and subscriber levels have plateaued, Mobile Network Operators are increasingly focussed on growing the average revenue per user by offering new products and improving customer satisfaction. Furthermore, MNOs are now looking to expand their future connection base from emerging technology such as the Internet of Things (IoT). In less developed markets, handset and tariff uptake remains strong.

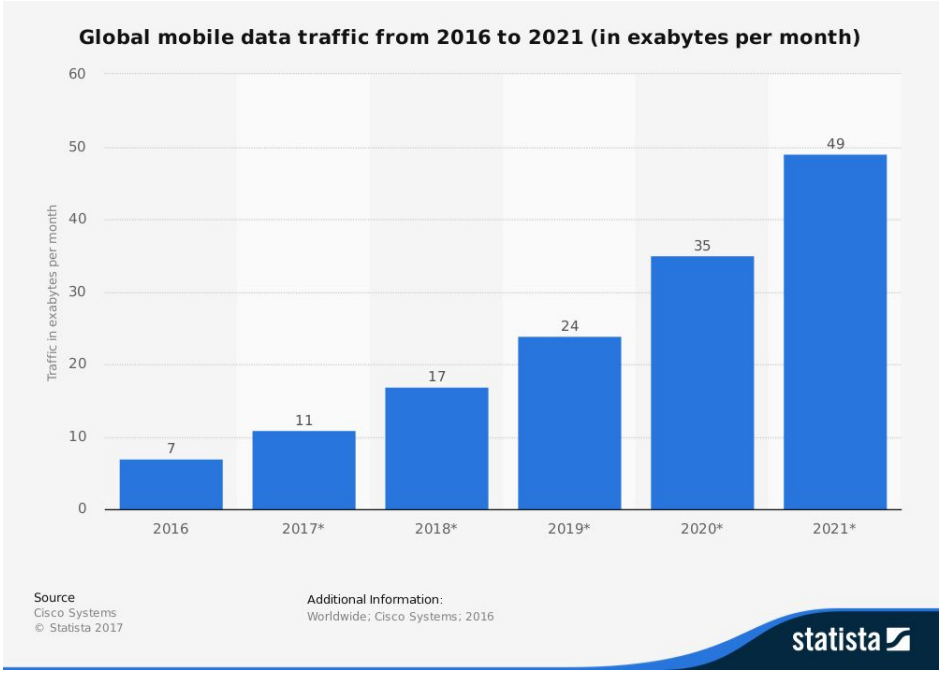
Key facts and figures

- Total mobile revenues reached $1.05 trillion in 2016[1].
- In 2015, the mobile ecosystem generated 4.2% of global GDP[1].
- Operator investments totalled around $880 billion between 2011 and 2015[1]
- The number of mobile phone Subscribers continues to grow:



- The average mobile phone Subscriber in the U.S and Europe is estimated to make over 8 calls per day or 3,000 phone calls per year
- In June of 2014, 561 billion text messages were sent worldwide, an average of 18.7 billion texts sent every day. This does not include app-to-app 'data' messages
- The amount of data traffic is forecast to continue to grow massively each year:

[1] Source: GSMA 'The Mobile Economy' 2017.



It is difficult to find any statistics for the number of usage events (calls, SMS, and data sessions) globally every year, but a calculation based on the above facts suggests a figure of over **300 trillion usage events** are completed every year.

39.3. Challenges in the Mobile Telecoms Industry (the Problem)

For many Mobile Network Operators, the key objective in the years immediately after market entry was to grow their subscriber base rapidly and to secure a significant subscriber share. In many cases, it has been difficult to introduce improvements to business support systems (BSS) up to date during this phase because of the possibility of disrupting operations and risking the velocity of subscriber growth. So many operators' BSS have become 'legacy' systems – increasingly unable to offer the flexibility needed to deliver new products and services, but difficult to change.

Billing systems in particular, are often problematic for MNOs. They often carry a high overhead in terms of the operating cost and the specialist resources needed to implement changes. In addition, some legacy billing systems are approaching capacity limits which could present a real problem if the operator wishes to enter the Internet of Things (IoT) and machine-to-machine (M2M) markets where the processing of large numbers of call records is required.

As the market has matured and the subscriber base has become increasingly saturated, many operators have shifted their strategy to focus on customer retention and to offer increasingly flexible and diverse propositions to their subscribers. However, they are often

hampered in their efforts to do so by the dated nature of BSS architecture and the inflexibility of traditional software solutions.

Traditional contract forms and tariff/service structures, enshrined in inflexible billing and pricing systems, are now handicapping the rollout of new proposition and product offerings; product marketing departments are developing new bundles, tariffs, and services but are frustrated by the time needed to get them to market as delivery projects must manage numerous complex system changes to implement even small product improvements.

In addition to the inflexibility and cost of many BSS systems, their bespoke or proprietary nature results in them often forming 'silos' of data, meaning that it is difficult to extract information and exposing it to other systems. Operators are under increasing pressure to improve the transparency of contracts and ensure their Subscribers are getting value for money. In most countries, there are regulatory targets around the accuracy of billing which can be difficult to meet. Most operators would like to offer better digital tools for customer service and reduce the need for expensive call centres, but are hampered from doing so by the need to integrate with complex BSS systems.

39.4. What Is Blockchain Technology?

Blockchain is a peer-to-peer technology which provides a de-centralised and secure distributed ledger of digital assets.

De-centralised - Data is not stored in a single (or small set of) location/s but instead is shared across the blockchain network. The network is fundamentally resilient and has no-single vulnerability for hackers to exploit. The abstraction of data from traditional architecture also means that it also far more accessible and in effect works like a shared distributed database

Distributed Ledger - Most blockchain platforms feature the concept of a distributed ledger, or record of transactions, which can be scrutinised by all members of the network. Transactions across the network need to be verified by a number of network 'nodes' before they are accepted as a valid transaction. This concept of consensus ensures that data in the blockchain is incorruptible and cannot be altered by any single entity which ensures security and resilience.

Secure - Data is secured using cryptographic keys. Although the transaction history of an asset can be seen, the contents can only be changed with an appropriate key and with blockchain technology this ledger cannot be compromised by a single entity, it requires many nodes to update a ledger, The confidential and secure client contracts which are bound between Business and individual clients securely.

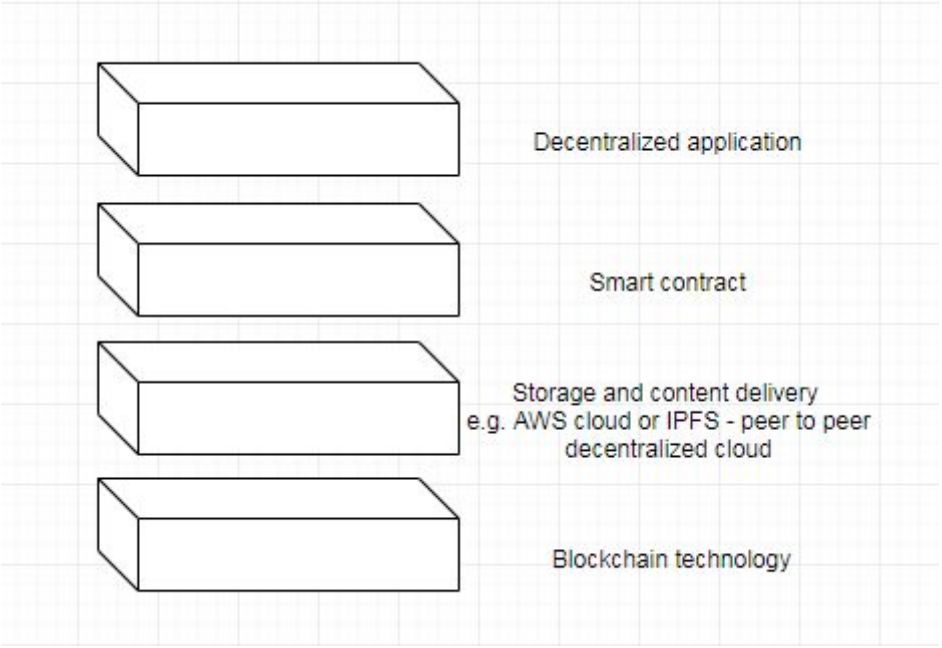
Blockchain technology was invented to fill in the gaps where trust was actioned by a centralised system, blockchain as a technology solved the problem by distributing the trust to a node in a decentralised approach. The most successful use of blockchain is Bitcoin a decentralised cryptocurrency.

In its initial iteration, Blockchain was conceived as method of providing a de-centralised store of value, or crypto-currency (e.g. Bitcoin), but developers quickly realised the potential for Blockchain to transact and store many different types of digital assets and information. In theory any information which can be digitised, from text to sound and image files, can be used as a digital asset.

There are many different blockchain technologies, we shall be looking at 2 main technologies as below:

1) Ethereum and
2) Hyperledger Fabric.

Blockchain architecture:



For choosing a blockchain, following point are the key to build a decentralised app (DApp) on a blockchain technology

- Platform
- Consensus algorithm
- Network technology
- Cloud storage
- Scalability

- Smart Contract
- Currency

Note	Ethereum	Hyperledger Fabric
Platform Type	Generic blockchain platform	Modular blockchain platform
Consensus algorithm	Proof-of-work	PBFT (Practical Byzantine Fault Tolerance)
Network	Permissionless, public or private	Consortium (Permissioned), private
Smart Contract	Smart contract code e.g. Solidity	Smart contract code e.g. Go, Java
Currency	Ether or Tokens via smart contract	Currency and tokens via chaincode.
Scalability	Scalable implementation of Ethereum is achieve business transactions	Consensus algorithms supports scalability.

Table 1 Generic difference between 2 Blockchain technologies.

Platform :

Hyperledger Fabric, Corda and Ethereum platforms have very different visions with respect to possible fields of application. Development of Corda is based on financial services applications. Fabric provides a modular and extendable architecture that various applications in different industries, from banking and healthcare over to supply chains.

Ethereum is independent and very generic for its field of application. It is not modularity in contrast to Hyperledger.

Consensus:

With Ethereum, all participants have to reach consensus over the order of all transactions that have taken place, irrespectively of whether a participant has taken part in a particular transaction or not. The order of the transactions is crucial for the consistent state of the ledger. If a definitive order of transactions cannot be established there is a chance that double-spends might have occurred, that is, two parallel transactions transfer the same coin to different recipients, thus making money out of thin air. As the network might involve mutually distrusting and anonymous parties, a consensus mechanism has to be employed that protects the ledger against fraudulent or adverse participants that attempt double-spends. In the current implementation of Ethereum, this mechanism is established by mining based on the proof-of-work (PoW) scheme. All

participants have to agree upon a common ledger and all participants have access to all entries ever recorded. The consequences are that PoW unfavorably affects the scalability. Concerning the data stored on the ledger, even though records are anonymized, they are nevertheless accessible to all participants, which is problematic for applications that require a higher degree of privacy.

In contrast to Ethereum, Fabric and Corda interpretation of consensus is more refined and does not merely boil down to mining based on PoW or a derivative thereof. Due to operating in a permissioned mode, Fabric and Corda provide a more fine-grained access control to records and thus enhance privacy. Furthermore, a gain in performance is achieved as only parties taking part in a transaction have to reach consensus.

Smart Contract:

Smart contracts (smart contract code) is designed to be written in Go or Java for Fabric, in Solidity for Ethereum and in Java or Kotlin for Corda. In Fabric, the term “chaincode” is used as a synonym for smart contract. As an illustrative example, the reader is reminded of the usage of a smart contract code in the consensus mechanism of Corda in order to ensure transaction validity. However, there is a notable difference between Fabric and Ethereum on the one hand and Corda on the other that is connected to the second way the “smart contracts” term is used. For Corda, smart contracts not only consist of code but additionally are allowed to contain legal prose. The Corda was explicitly designed to account for the highly regulated environment of the financial services industry. Both Fabric and Ethereum lack this feature.

Currency:

Ethereum is designed on a build-in crypto-currency called *Ether*. It is used to pay rewards to nodes that contribute to reach consensus by mining blocks as well as to pay transaction fees. There-fore decentralized apps (DApps) can be built for Ethereum that allow monetary transactions. Furthermore, a digital token for custom use cases can be created by deploying a smart contract that conforms to a pre-defined standard for one’s own currencies or assets.

Fabric and Corda do not require a build-in cryptocurrency as consensus is not reached via mining. With Fabric, however, it is possible to develop a native currency or a digital token with chaincode. For Corda, a digital currencies or tokens is not possible..

39.4.1. Scalability:

Scalability is a bottleneck for blockchain technologies where some of the applications requires thousands of tps (transactions per seconds). Following are the scalability seen on different blockchain platforms:

- Bitcoin: Scalability of 6 tps
- Ethereum: Scalability of 20 tps per node.
- Ethereum: 7 tps per node for ERC 20 (smart contract on the Ethereum blockchain).

The reason for the scalability bottleneck is due to the consensus algorithm used by the underlying platform, which provides the highest level of security, it obviously comes at a cost of scalability. Consensus algorithm is such that every node in the network can validate, authenticate and process every transaction on the blockchain. Recall that blockchains have one inherent critical characteristic — "decentralization" — which means that every single node on the network processes every transaction and maintains a copy of the entire state.

Scalability solution:

Ethereum in effort to provide scalable solution, it is proposing to provide two-layer solution to provide transactions and smart contract scalability.

- Layer 1 sharding
- Layer 2 sidechain (off-chain transactions) e.g. Plasma.

There are two main paths to improving blockchain scalability. The first ("sharding") involves creating better-designed base-layer blockchain protocols, which still maintain most of the desired decentralization and security properties of a blockchain that we see in the simple designs available today but only require a small percentage of nodes to see and process every transaction, allowing many more transactions to be processed in parallel at the same time. The second involves creating "layer 2" protocols that send most transactions off-chain and only interact with the underlying blockchain in order to enter and exit from the layer-2 system and in the case of attacks on the system.

One of the assets that can be housed in the Blockchain is Smart Contracts - essentially secure code containers which can not only store contractual information but can execute changes and operations depending on certain pre-set conditions. This 'self-execution' allows

the contracts to function outside of traditional architecture and therefore to be far more flexible and accessible.

The combination of the accessibility and security of Blockchain network platforms with the functionality of smart contracts has huge potential in many areas. Previously many projects have focussed on building the core Blockchain and Smart Contract platforms, but as this development matures new projects such as Wire are emerging which can look at how this combination of technologies can be applied to solve specific real-world problems.

39.5. How Wire will solve the problem through use of the blockchain

The Wire solution applies blockchain technology to overcome many problems around the cost, flexibility, and scalability of business support systems faced by MNOs, and to provide extensive opportunities to enhance product offerings and customer experience.

These include:

A shared view – the smart contract is abstracted from the Operator and exists on the blockchain allowing direct interaction by both Operator and Subscriber. Both parties can inspect the contract for full transparency and have a shared view of billing and contract situation; this will reduce misunderstandings and reduce billing enquiries.

New products, services and bundles - The abstraction of the contract from proprietary and 'hard-wired' BSS allows greater flexibility to build new deal types and offer different products to customers.

Easy billing – The Subscriber is billed through an app. No need to send out paper bills.

Easy and instant secure payments – Subscribers pay their bills directly in the app. Payments are made via the blockchain and the Operator receives the balance lightning fast. There shall be an option for secure payments via cryptocurrency.

Targeted and conditional messaging – The contract can be analysed for certain product and usage conditions, and targeted messaging sent to the subscriber app to promote new offerings, offer top-ups, etc.

A scalable solution – freeing the billing and contracting processing from physical hardware and constraints significantly improves the Operators ability to scale to explore new expansion opportunities.

The flexibility to respond to changing conditions – the contract mechanism give the operator the ability to change the contract to meet new market challenges and respond to new prospects.

Confidential and secure client contracts – Using consortium blockchain technology i.e. permissioned and private decentralised ledgers, the mobile operator shall be able to maintain the contract complete confidentiality with respect to other clients on same decentralised ledgers.

Secure communications – The DApp shall be securely connecting the Smart contract with the Subscriber app and Business app.

40. The Wire Solution – Selected Use Cases

Using its experience of mobile telecoms operations, the Wire project has identified several use cases. Further cases will be considered as a result of the project's collaboration with MNO partners.

40.1. Building a contract

Building a contract for a Subscriber will be easy using the Wire Solution. The solution allows simple tariff and product structures to be built directly into the Smart Contract. Whilst this simplification will challenge the complex interdependent tariff and usage relationships that many Operators have been bound to use by current BSS technology, it will hugely improve the flexibility to provide mobile contracts which offer better value for their customers, and the transparency of the solution will allow the Subscriber to have a far clearer understanding of how their mobile plan works.

When the Wire Subscriber App is first opened on the Subscriber's device, they are presented with a start-up wizard. The user enters basic information including some straightforward security responses and the app then checks whether the Subscriber already has an existing contract. If no existing contract is found, the app creates a blank smart contract on the Blockchain and populates it with some basic contact information.

The contract contains two key parts; the smart contract which contains the product and tariff structure for the subscriber's deal; and general terms and conditions.

When the Subscriber is ready to discuss a deal with an Operator, they provide the Operator Agent with the Operator key. This can be done by exposing a QR code to the Agent when in a face-to-face situation, e.g. a retail conversation, or by sending a secure message to the agent via the app. Using the Operator Key, the agent now has the ability to inspect their existing contract and offer a better deal for the Subscriber.

The Operator Agent can then build a deal directly with the customer using the Wire Agent Tool. The Agent drags and drops tariffs, booster packs, devices, and other products and services into a sample contract – a facsimile of a real contract showing exactly what the final agreement will contain. This is then sent to the Subscriber App for the Subscriber to consider. When a deal has been agreed, the customer signs the contract using their private Subscriber key and the deal is transposed from the 'sample' contract into the actual smart contract on the Blockchain.

The Operator's provisioning systems are informed automatically via the Wire API and the contract can be inspected to obtain the necessary

set-up information. Alternatively, the Operator can obtain details of all newly contracted customers via a report from the Wire search tool.

If an agreement is not reached and the subscriber clicks to reject the offer, the Operator Key is invalidated for security reasons. A new one is generated when the Subscriber next re-opens a deal discussion.

In theory, a deal can be discussed, built, signed and provisioned in minutes. The transparency of the contract allows the customer to obtain the best deal possible and ability of the Agent to build contracts rapidly and flexibly means that Operators can begin to offer far more tailored customer offerings.

40.2. Contract signature

When a contract deal has been agreed both the Subscriber and agent provide specific approval using the Subscriber app to form a contract. This approval is recorded in the contract with a timestamp. Both the Operator and Subscriber must provide their keys to record this approval. The Subscriber key is provided automatically by the Subscriber app (after the Subscriber has entered their pin code as additional validation). The Operator key is provided by the Operator agent in the form of a QR code which is scanned by the Subscriber app. The QR code would also provide additional information to be recorded in the contract, such as agent ID.

The contract is enforceable as soon as it is validated in the blockchain. The whole process from approval to validation should take minutes.

40.3. Subscriber Billing

The Subscriber bill is calculated directly within the Subscribe App using the product and usage event data extracted from the Smart Contract. The Subscriber can view their bill directly on their device and make the appropriate payment. No paper bill is issued, but the Subscriber App can send an electronic version of the Bill to the Subscriber's e-mail as an option.

The Subscriber App allows the Subscriber to view their allowances, usage, top-ups, etc. for complete bill and consumption transparency

The Operator has the option of including helpful notes against each tariff or product which may explain why it appears on the bill in the way that it does. For example, 'If you don't have our Worldwide Roaming add-on, this call might be chargeable.' This messaging will help to reduce billing enquiries to the Operator's call centre.

40.4. Credit and Billing Cycles, and Payments

The Smart Contract contains two values (set when the contract is signed) which determine the mechanism for billing and customer credit.

Bill Limit – The bill limit is a numeric value which sets a maximum time after a weekly or monthly bill has been generated by which the customer must pay the balance of the bill. For example, if the bill limit is set at 4 weeks the customer must pay their bill within 4 calendar weeks.

The application will provide reminders when 50%, 75%, and 90% of the limit has expired, and a final warning at 95%. If the bill limit expires, the application will set a flag in the contract to report the limit has been breached. The Operator can use this flag to search for all subscribers which have gone beyond their bill limit for collections purposes.

Setting a bill limit flag sets the app to 'bundle' billable events within a set period into a regular traditional bill.

If no value is specified for Bill Limit the Credit Multiplier is used.

Credit Limit – Setting the credit limit means that billing occurs on a rolling basis. Instead of a regular bill, the Subscriber is presented with a rolling balance. The credit limit determines 'unpaid balance' the customer can accumulate before a flag is set in the Smart Contract to say that the limit has been breached. Again, the application will provide multiple warning as the credit limit is approached.

Subscribers can pay their balances directly from the Subscriber App using the built-in currency wallet. The funds are sent directly to the Operator via the blockchain. As an option, the Subscriber app can e-mail a receipt for the payment to the Subscriber.

40.5. Billing and Contract Inspection

The Wire Solution includes a desktop and tablet App for Operator agents to view both the smart contract and billable events. As all contract and event information is contained in the smart contract, the agent will have exactly the same view as the Subscriber which facilitates easier billing query support.

The Subscriber can view their bills and recent usage using the Subscriber app.

If a change is made to the pricing or terms by the Operator, the Subscriber app notifies the Subscriber automatically.

40.6. Subscriber Messaging and Identification

The Wire solution provides a messaging centre app to the Operator through which they can send messages to Subscribers. These messages are displayed in the application. The messaging solution provides assistance in formatting the message so that it can be viewed correctly on a device and contains several pre-set responsive messaging templates.

Important messages can be marked as a 'recordable event' which forces the Subscriber to confirm that they have been read and this fact is recorded in the smart contract.

The Wire solution also provides a simple reporting tool which allows the Operator to pull-off list of Subscribers who have specific contract features such as tariffs, contract expiry dates, etc. These lists can be used with the messaging centre to easily send messages to specific Subscriber groups.

40.7. In-Life Contract Variation and Care

Network Operators do their utmost not to change pricing, allowances, or terms whilst the contract is in effect. However, it is not uncommon that a change in the economic or regulatory landscape occurs during the contract lifetime forces the Operator to alter the terms of the contract. This can often be a difficult exercise for the Operator, especially those using legacy or complex BSS systems. The challenges of identifying all impacted Subscribers (e.g. those on a specific tariff), of communicating the reasons for the change and its impact to those Subscriber, and in applying the contract change itself, can be extremely painful and risky.

The Wire solution makes this process far more straightforward. Firstly, Subscriber smart contracts can be readily inspected to determine which Subscribers have the specific terms or price-points which need to be altered. Then, appropriate messaging can be pushed to the Subscriber app for those Subscribers identified and an acknowledgement recorded that the Subscriber has read and understood that communication.

Finally, the changes can be applied directly to the smart-contract by the Operator (using the Operator key to access the contract). If required, a further communication can be sent to the Subscriber confirming the changes have been made, but in all cases the application will automatically alert the Subscriber that a change has been made to the contract.

In some circumstances, the Operator may need to the Subscriber to specifically accept or reject the proposed changes. This can be done via the app and a flag registered in the contract to record this event. If the Subscriber rejects the change, and if the legal terms of the contract allow them to do so, they may trigger a contract cancellation process.

In-life contract variation may also occur where an Operator Subscriber Care agent might (providing they have the appropriate internal approval) change the contract terms for the Subscriber to rectify a problem, or for example apply a discount to compensate the Subscriber for some inconvenience. In this case, the Subscriber can approve the contract variation in real-time using the app and feel confident that the issue will be rectified 'on the call'.

The Wire Solution provides a method for mass application of contract variations and a App through which care agents can apply changes for specific Subscribers.

All contract changes are logged in the smart-contract and can be viewed by either the Operator or Subscriber.

40.8. Lock/Suspend Billing

If a device is reported lost or stolen, or if the Operator needs to stop the Subscriber from being billed for another reason, the Operator can

set a 'lock' flag in the smart contract which stops any subsequent usage events being billed in the Subscriber app (until the flag is reset).

40.9. Allowance Alerts

The application will automatically monitor the Subscriber's usage and notify them when they approach their usage allowance limit. This will prevent users overspending and support the Operators efforts and obligations to ensure Subscribers do not fall into debt.

40.10. Portable Contracts

Because the smart contract is both de-centralised and is 'electronic' it is far more transparent.

The Subscriber can choose to expose the contract terms to other parties who can scrutinise their contract duration, allowances, and pricing, and who may be able to offer a better deal. To do this the Subscriber would unlock their application, visit the relevant contract screen, and opt to show the contract details. The app would then generate a Quick Reader (QR) code which could be scanned by the third-party. The QR would direct to a secure file which would provide contract and pricing information.

This case facilitates better competition in the market and ensures contracts to ensure that the Operator is providing the best value for the Subscriber. This is key industry issue in many countries with regulatory bodies placing increasing pressure on Operators to improve the transparency, simplicity and fairness of contracts.

40.11. Further Use Cases (The Waiting Room)

In addition to the use cases outlined above, the project will consider the following functions:

- Contract cancellation/termination (using conditional checks to determine whether a contract can be ended any subscriber cancellation charge)
- Advertising/promotion through the app based on the Smart Contract conditions, e.g. promoting a top-up as the user approaches their usage allowance cap.
- On demand premium content ordering and unlock.
- Credit checking using information from the smart contract
- Porting process initiation
- Out of Allowance Alerting
- Know Your Customer (KYC) and Data Protection (DPA) information – capture through the app and storage in the smart contract
- eSIM integration/interoperability

Furthermore, the project proposes to engage with MNO partners to refine and expand the use cases already identified, and to scope out new applications of the solution.

41. Technical Application Overview

41.1. Overview of The Wire Architecture

The Wire Architecture shall employ Client/Server architecture, the Wire Smart Contract on blockchain shall act as a Server and there shall be two different applications, the Business application and Subscriber application. Security is the essential part of blockchain technology and to meet the security and trust the Client/Server architecture shall use mutual authentication using PKI and certificates.

The Wire Architecture consists of three core components:

The Wire Smart Contract – contains the baseline contract information such as contract term, products & tariffs, pricing, subscriber information, and service information, and a record of usage (voice, SMS, data, etc.), contract amendment history, payments, key events, etc.

The Wire Subscriber App – the customers control panel for their mobile service. The app reads information from the Smart Contract to provide a view of usage, spend, payments, allowances, etc. The app takes care of billing for the Operator. The customer can also make payments for products and services via a built-in wallet.

The Wire Operator Control Centre –

The Operator Control Centre consists of are four core components**:**

API – the interface with the operator infrastructure through which the Operator can send usage information to the Smart Contract which is recorded in the contract and used by the Subscriber App as part of billing. The API can be used to write other information to the Smart Contract as part of controlled contract amendment events, and read key contract information for the purposes of customer care, payment tracking, etc.

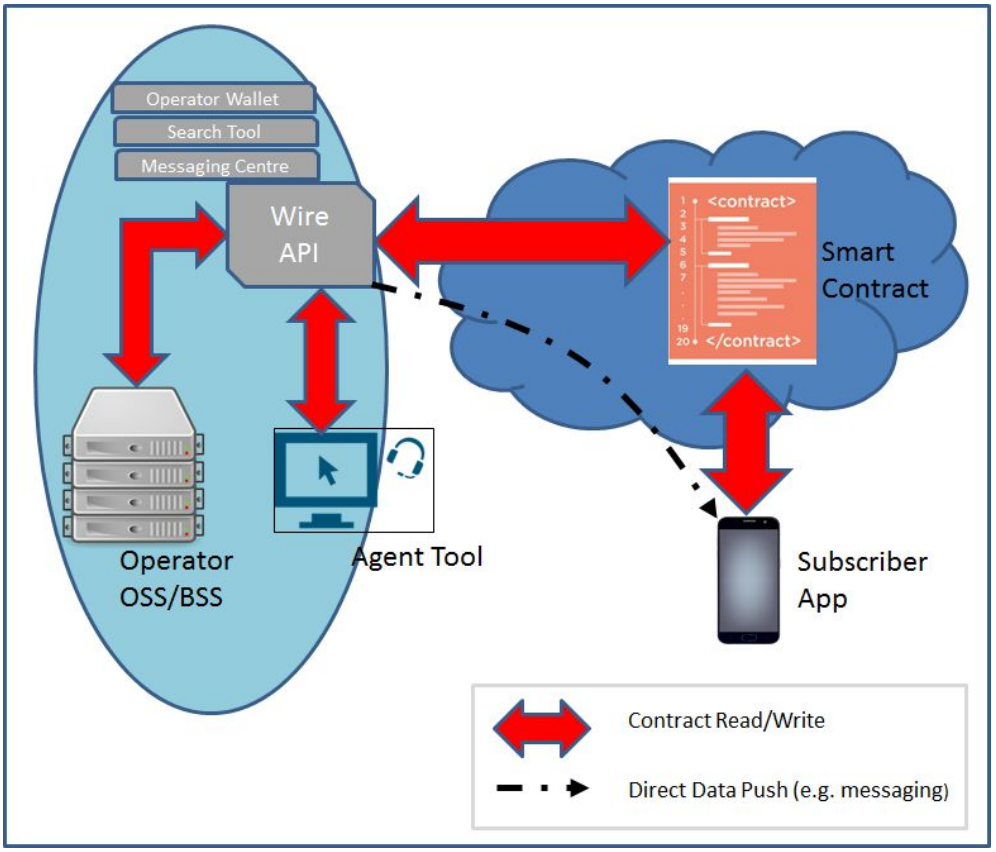
The Messaging Centre – capable of sending messages to subscribers.

The Search Tool – interrogates customer contracts to retrieve lists of customers with selected attributes

The Agent Tool – to provide customer care. The tool gives the Operator Agent a shared view of the Smart Contract and the Agent can make agreed changes to the Smart Contract. In addition, the Agent tool is used to build the initial contract with customer.

The Operator Wallet – allows the Operator to store Wire Tokens as payment for usage transactions. The operator loads funds in the form of Wire tokens into the wallet and the API will trigger the wallet to deduct the charge for transacting each billing event from

the wallet balance. The charge is transacted via the blockchain to the corporate Wire wallet.



41.2. The Smart Contract

41.2.1. Contract Design Principles

The intention is that the Smart Contract will be designed around Ricardian principles. One of the key benefits of Ricardian contracts is that they present electronic, cryptographically-secured contracts, in a way that they can still be read in a similar format to traditional paper-based agreements, i.e. whilst having all the features of a blockchain-based smart contract they can be outputted with the same legal structure and terminology as current contracts.

One of the challenges for the project might be possible resistance to the solution from Operator legal teams who may be uncomfortable with the completely new form of agreement type presented by a straight blockchain smart contract. By designing the contract in such a way that it still presents in a way that resembles a traditional agreement, the project hopes to mitigate this risk.

The project will also attempt to design the contract to be able to accommodate in-life structure changes. This will allow Wire to update the contract fields and schema to be adapted for new functionality, and new products and services that an operator may

want to introduce. Whilst preserving specific terms and conditions (with the Subscriber) and core data, expanded contracts can be deployed to the subscriber base with minimal impact.

41.2.2. Contract Content

The following file provides an outline of the information that might be contained in the Smart Contract.

ExampleContractContents.xlsx

41.2.3. Contract Access Control

The Wire contract can be changed with two keys; the Subscriber key, and the Operator key. Both keys have full read access to the contract for complete transparency between parties. Write access differs depending on the key type.

The table below shows a *sample* of different access cases and permissions:

	Build initial contract and deal[1]	Write usage data	Write personal and DPA data	Expose contract information (to third-party)	Update pricing and contract terms	Write 'Recordable Event'	Make a paymen (paymer event)
Subscriber	✓	✗	✓	✓	✗	✓	✓
Operator	✓	✓	✗	✗[2]	✓	✓	✓

[1] Both keys are needed to build the initial deal and form the contract.

[2] The Operator might request the Subscriber to expose contract information to chosen third-parties, but the Subscriber must give specific permission to do so. The mechanism for doing so is TBD.

41.3. The Subscriber Application

The functions of the Subscriber Application are outlined in the Use Cases in Section 2 of this document and include contract view and signature, bill and usage viewing, and payment (via the wallet)

41.3.1. The Subscriber Wallet

The Subscriber Wallet allows the Subscriber to make payment for their bills direct to the Operator via the blockchain. In common with the Smart Chain solution, it is likely that the project will integrate an existing wallet and payment solution to provide this functionality.

Preferably, the wallet system will allow payment in fiat currency as well as a variety of crypto-currencies.

41.3.2. App Maintenance and Updating

The application will be designed so that updates to introduce new features and optimisations can be made easily over the air (or via a download) without losing any key information such a keys or wallet balances.

41.3.3. App Skinning

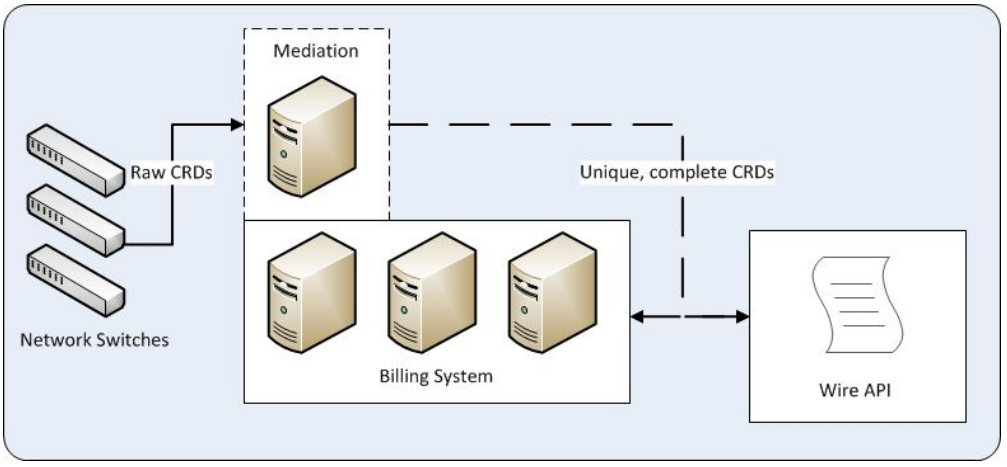
Although the general layout and functionality of the Subscriber Application will be common to all Operators, the app will allow the deployment of custom branding so that each Operator can make the app appear to be unique to them.

41.4. The Wire Control Centre

41.4.1. API

The Wire API provides the primary method for interfacing an Operator's core Network and Business Support Systems with the Wire smart contract on the blockchain. The API will translate usage data provided by the Operators 'back-end' systems and write it into the Smart Contract.

Usage data is supplied to the API in the form of unique (de-duplicated) and complete (non-partial) Call Data Records (CDR) from the Operators Business Support Systems. Typically, these CDRs would be served in ASN.1 format but some Operators may use an alternative standard or proprietary format, in which case customisation of the API will be required.



The API also serves as a conduit for writing/reading contract information as part of customer care and contract amendment operation.

Although the API allows straightforward integration between the Operator BSS, it will not be a plug'n'play component as each MNOs architecture and data standards is likely to be different and configuration of the API will be needed.

The operation, hosting, load balancing, security, monitoring, resilience, backup, and recovery of the API will be the responsibility of the MNO, but it is envisaged that the Wire team will provide remote support during a major incident as part of ongoing customer service provide by Wire.

41.5. The Blockchain (Transaction) Platform

The choice of blockchain platform will be determined early in the project. The initial view is that the solution will utilise an existing third-party blockchain platform. This provides many benefits:

- Reduces cost and time to market – no need to develop own platform
- Utilises tried and tested platform
- Expertise and support already in place

However, it is possible that the project may have to develop a proprietary blockchain if a third-party platform which meets the high-level solution requirements cannot be identified.

Primary considerations in choosing the platform will be:

- The platform must be able to support the Wire smart contact and dual key solution
- The platform must be able to support high-numbers of simultaneous transactions
- The platform must be capable of scaling significantly to handle future volumes of transactions.
- The ability to identify rejected (invalidated) write operations should be established
- The platform developers should have clear support processes, resources, and infrastructure defined.
- The project team need to be assured that the transactions can be processed in a 'reasonable' time and that future increased throughput will not compromise this speed. Although 'real-time' transaction speeds are not crucial to the service, customers will expect updates to be written to the Smart Contract within a few minutes.
- The cost associated with completing a transaction and the impact of low value transaction amounts on payment validation time.

42. Revenue Ecosystem

Wire will generate income by making a fractional 'micro-charge' (which could be for example, the equivalent of $0.005) to the operator for each usage event transacted to the Smart Contract. For simplicity, the micro-charge amount will be fixed for each usage and negotiated with each MNO customer as part of the commercial process (along with the period that the fixed amount will apply for before review).

Payment of the micro-charge is made using the Wire Token (see below). An operator can purchase wire tokens on the open market (via a cryptocurrency exchange) and the token can be loaded into a wallet which is provided as part of the Wire Control Centre. The micro-charge is deducted from the wallet when a usage event occurs and sent by the API to the Wire corporate wallet via the blockchain.

The Wire Token

The Wire Token is a simple utility token used by the operator to fund usage transactions, and thereby pay for the Wire solution and service.

The Wire Operator Control Centre associates a unique ID with each billable usage event, the ID is also attached to the wallet payment for the event. In this way usage events and payments can be reconciled for audit purposes.

A single Wire token will fund multiple micro-charges (usage events). For example, a single Wire Token could fund a block of 10,000 usage events.

Token volume and distribution

The project will create a total of 1,000,000,000 (1 billion) Wire tokens, i.e. this will be the maximum supply of tokens.

Approximately, 548,000,000 tokens will be released as part of the token mainsale, of which 150,000,000 will be allocated to seed and private investers. A further, 2,000,0000 will be provided as part of a bounty scheme.

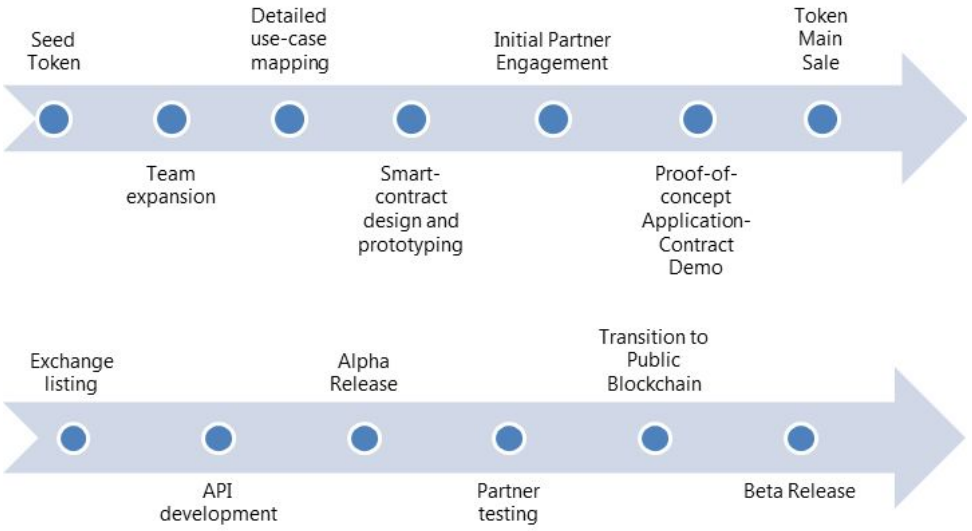
The project team retain 100,000,000 tokens to 'pump-prime' commercial agreements with MNO partners, i.e. to incentivise the take-up of the solution. So, early-adopter MNOs and those who work with the project as partners during the development phase will be provided tokens either free of charge or at a discounted rate to secure their continued usage and support for the project.

200,000,000 tokens will be retained to fund the ongoing development of the project. And a final 150,000,000 tokens will be allocated to the project team.

43. Project Delivery

43.1. The Wire Roadmap for 2018

The following diagram shows the high-level roadmap for the project:



44. Contacts.

For more information on the Wire project, please contact:

E-mail: matthew.armishaw@googlemail.com

Tel: +44 (0) 7786 932687