White Paper

JACS

Just Another Communications Stack

Written by: Moustafa Amin CCIE, OCSA
Mohamed El-Dessouki MSc, Business Information Technology

Version: 2.0
Date: 16th of July 2019

viaBlock, LTD

Table of Contents

1 Abstract 5

2 Background 6

2.1 TCP/IP Overview 6

2.2 TCP/IP Alternatives 6

2.3 CLNS 6

2.4 IPv4 and IPv6 7

3 Introduction 9

3.1 Physical and Logical Addresses 9

3.2 Per-Interface IP Address 9

4 Problems 10

4.1 IPv4 Exhaustion 10

4.1.1 IPv4 Exhaustion - Reasons 10

4.2 Internet Routing Security 11

4.3 Central Authorities in IPv4 and IPv6 12

5 Solution Attempts 14

5.1 IPv4 Exhaustion 14

5.1.1 Before Exhaustion 14

5.1.2 After Exhaustion 14

5.1.3 Transition mechanisms 15

5.1.4 IPv6 perceived as a Long-term Solution 16

5.1.5 TUBA and CLNP 16

5.1.6 ATN and CLNP 17

5.2 Internet Routing Security 18

5.2.1 Resource Public Key Infrastructure, RPKI 18

5.2.2 IPchain: Blockchain-based Solution 18

5.3 Central Authorities 19

6 JACS 21

6.1 A little bit of history 21

6.2 CLNP versus IP 21

6.3 Limitations of previous attempts 22

6.4 Introducing JACS 22

6.5 Roadmap and development timeline 23

7 NSAP Address 24

7.1 NSAP Structure - A little bit of history 25

7.2 IPv4 to NSAP 27
7.3 JACS NSAP 27
8 Blockchain 30
8.1 General Functions of blockchain 31
8.2 Transactions, accounts and data storage 32
8.2.1 Unspent Transaction Output (UTXO) 32
8.2.2 Account balances 32
8.3 Blockchain for JACS 33
9 JACS and blockchain 34
9.1 Ethereum 34
9.1.1 Smart Contracts 34
9.1.2 Distributed Applications (dAPPs) 35
9.2 JACS over Ethereum 36
9.3 Utility token 37
9.4 JACS native blockchain 37
9.5 JACS rules 38
9.5.1 Abundance 38
9.5.2 Uniqueness 39
9.5.3 Aggregation 39
9.5.4 Anonymity 40
9.5.5 Fairness 40
10 JACS - Life Cycle 41
10.1 MVP (Q2, 2019) 41
10.2 Pre-Alpha version (Q4, 2019) 43
10.2.1 Scope & Address Allocation 43
10.2.2 One-time, Gas-free allocation fee 44
10.3 Alpha Version (Q1, 2020) 45
10.4 Beta Version (Q2,2020) 46
10.4.1 New Internet Goal 46
10.4.2 Adopters as Enablers 46
10.5 Official Release, Version-1 (Q4, 2020) 47
10.6 Official Release, Version-2 (Q1, 2021) 48
10.6.1 Lost Keys & Account Recovery 48
10.6.2 Block Production Rewards 49
10.7 Official Release, Version-3 (Q3, 2021) 49
11 Tokenomics 51
11.1 Proof of ownership 52
11.2 Adopters as Enablers 52
11.3 Team 52

11.4 Company **53**
11.5 Bounty **53**
11.6 Airdrop **53**
12 Crowdfund ***54***
12.1 Pre-ICO **54**
12.2 ICO **54**
13 General Disclaimer ***56***
14 JACS Token Legal and Crowdsale ***57***
14.1 General Information **57**
14.2 General Knowledge **57**
14.3 Risks **57**
14.4 Disclaimer **57**
14.5 Representation and warranties **59**
14.6 Governing law – Arbitration **59**
14.7 Parties with whom we may share your information **60**
14.8 What happens in the event of a change of control **60**
14.9 Forward - looking statements **60**

1 Abstract

In this white paper, we are presenting 'Just Another Communications Stack' (JACS) that aims to change the way data networks currently work.

JACS is a new communications stack that is totally different than the TCP/IP stack. JACS allows nodes to perform regular actions like surfing the Internet, interconnecting and much more.

At a very high level; JACS is the result of combining Blockchain and Connection-Less Network Services (CLNS), with its 160-bits NSAP addresses.

There are many drivers behind the creation of JACS, like: IPv4 address depletion, centralized address allocation and Internet routing system security.

The world ran out of IPv4 addresses in the first quarter of 2011.

IPv4's meager 32-bit address space of four billion is not even enough to give each person on earth a unique identifier.

IPv6 has come just in time, providing 340,282,366,920,938,463,374,607,432,768,211,456 possible 128-bit addresses, but still IPv6 shares many issues as the current IPv4, and its rollout rate (that has already been very slow over the past years) is getting slower than expected for a variety of reasons.

In addition, the current system to manage the global pool of IP addresses (this applies to both IPv4 and IPv6) is centralized and managed globally by Internet Assigned Numbers Authority (IANA), and by five Regional Internet Registries (RIR) responsible in their designated territories for assignment to end users and local Internet registries (LIR), such as Internet service providers.

Each of these RIRs manage the address pool for a large number of countries. Because the RIRs are private organizations, they are subject to the legal framework of the country where they are based. This configuration results in a jurisdictional overflow from the legal framework of the countries where the RIR is based to all the countries that the RIRs are serving (the countries served by the RIRs de facto become subjects of the legal system of the country where the RIR is hosted).

With JACS there will be no single top authority, no regional or local authorities. JACS addresses will be assigned and administrated over blockchain. Thus, fulfilling all general guidelines like administrative decentralization and data abstraction plus hardening the security of the Internet routing system.

Most importantly, there will be no maintenance or renewal (i.e. annual) fee for JACS blocks of addresses.

2 Background

2.1 TCP/IP Overview

Transmission Control Protocol/Internet Protocol (TCP/IP), is a suite of communication protocols used to interconnect network devices on the internet. TCP/IP can also be used as a communications protocol in a private network (intranet or extranet).
TCP/IP specifies how data is exchanged over the internet by providing end-to-end communications that identify how it should be broken into packets, addressed, transmitted, routed and received at the destination. TCP/IP requires little central management, and it is designed to make networks reliable, with the ability to recover automatically from the failure of any device on the network.

2.2 TCP/IP Alternatives

There used to be many alternatives to the TCP/IP protocol stack, few still exist and many more existed. But with the evolution of the Internet, a single unifying set of protocols was required to interconnect networks.
Some of those old alternatives included: DecNet, Appletalk, IPX, SNA, Apollo, Vines and others.
Apple used AppleTalk and AFP over a DDP transport, but they eventually migrated all of that to IP transport, instead.
Xerox Network Systems (XNS), based on PARC Universal Packet (PUP), was used in products like 3Com 3+Share and Ungerman Bass Net/One.
Novell modified XNS into the IPX/SPX protocol for their wildly successful Netware system. Banyan Networks also used a variant of XNS called Banyan VINES. Apollo/Domain also had a proprietary XNS variant, as well as a proprietary networking medium (Domain Ring).
Chaosnet was a packet-based protocol with a pretty small address space from MIT, used to interconnect LISP machines, developed around the same time as PUP and IP.
Digital Equipment Corporation has their own protocol, DECnet, to support their VMS machines. The classic version was called "Phase IV."
IBM maintained a primitive protocol called Systems Network Architecture (SNA) which ran over Synchronous Data Link Control (SDLC) for communications around their mainframes.
On top of these alternatives, came the OSI Connection-Less Network Services (CLNS).

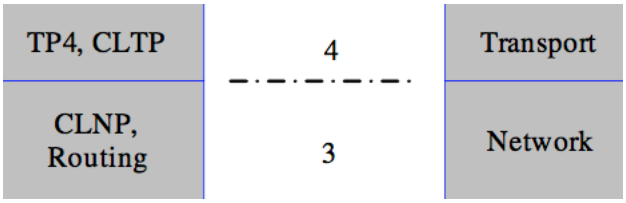
2.3 CLNS

Within the OSI CLNS protocol stack; Connection-less Network Protocol (CLNP) and Transport Protocol Class-4 (TP4) would be the closest analogy to TCP/IP.

The OSI protocol stack has a major advantage over the TCP/IP stack as it defines both the protocols and the APIs between the layers. CLNS is the API (the function calls that allow transport layers to exchange datagrams across the network). CLNP is the layer-3 protocol that implements CLNS.
CLNP provides fundamentally the same underlying service to a transport layer as IP. CLNP provides

essentially the same maximum datagram size, and for those circumstances where datagrams may need to traverse a network whose maximum packet size is smaller than the size of the datagram, CLNP provides mechanisms for fragmentation. Like IP, a checksum computed on the CLNP header provides a verification that the information used in processing the CLNP datagram has been transmitted correctly, and a lifetime control mechanism (Time to Live, TTL) imposes a limit on the amount of time a datagram is allowed to remain in the internet system.

While TP4, the transport protocol, offers error recovery, performs segmentation and reassembly, and supplies multiplexing and demultiplexing of data streams over a single virtual circuit. TP4 sequences PDUs and retransmits them or re-initiates the connection if an excessive number are unacknowledged. TP4 provides reliable transport service and functions with either connection-oriented or connectionless network service. It is the most commonly used of all the OSI transport protocols and is similar to the Transmission Control Protocol (TCP) in the TCP/IP protocol suite.

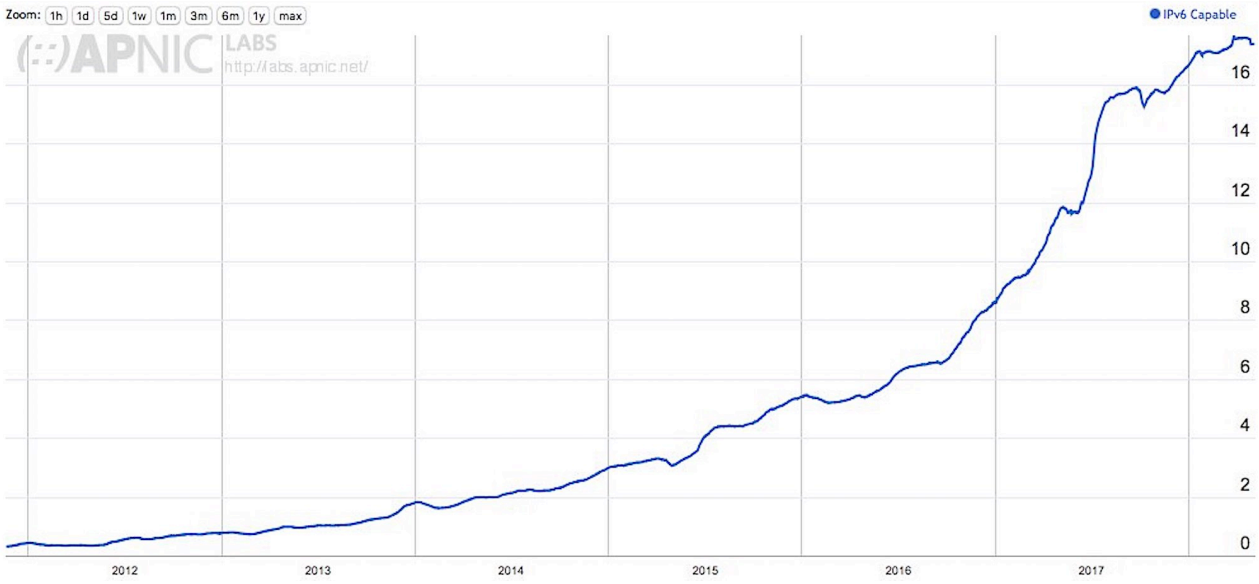


Protocols Analogy between CLNS and ISO OSI Model

2.4 IPv4 and IPv6

IPv4 has evolved significantly in the last three decades. It got all the nifty features that supposedly made IPv6 a better protocol (IPsec, QoS and autoconfiguration). There are also several decades of operational experience and gradual improvements that allowed building large-scale manageable secure networks.

IPv6 was definitely a better protocol than IPv4 in 1995, but then it got shelved for 15 years and while everyone was improving IPv4, nobody ported those improvements to IPv6, until the Internet community woke up and realized it was the right time to start dusting the arcane protocol that barely anyone touched for many years.



The figure clearly shows that the adoption of IPv6 is slowing and maybe tailing off.

3 Introduction

3.1 Physical and Logical Addresses

In Data Networking, there are two kinds of addresses; physical addresses (like MAC) and logical addresses (like IP). IP addresses and MAC addresses were developed around the same time but each was responding to a different problem.

Consider the case of an Ethernet cable/segment over which several devices could be physically connected, and all have visibility to the signals transmitted onto the cable. In other words; all devices can hear all other devices, but each device has a unique identifier (MAC Address) with which they could address their data. This address would not prevent everyone else on the cable from seeing the data, but it would provide an indicator as to which, specific device was the intended recipient of that data.

But because in the real-world nodes will not always be on the same segment, as the case of the public Internet, every data network node (host, router, or even a network printer) is assigned a logical global address (like IP) that is used to locate and identify the node in communications with other nodes.

IPv4 addressing provides 2^{32} (4,294,967,296) addresses (just less than 4.3 Billion). However, large blocks of IPv4 addresses are reserved for special uses and are unavailable for public allocation.

3.2 Per-Interface IP Address

In the IP world, if a device has several network interfaces, then each interface must have at least one distinct IP address assigned to it. For example, a laptop might have a wireless network interface and a wired network interface using a network cable, and this would require a total of two IP addresses, one per interface. Another example is a mobile phone with cellular data network and Wi-Fi.

Routers, by nature, have several network interfaces and typically have several IP addresses associated with them. It is also possible that an interface can be assigned more than one IP address for various reasons (secondary addresses).

The IPv4 addressing structure provides an insufficient number of publicly routable addresses to provide a distinct address to every Internet device or service. This problem has been mitigated for some time by changes in the address allocation and routing infrastructure of the Internet.

This started by the transition from classful network addressing to Classless Inter-Domain Routing (CIDR), that delayed the exhaustion of addresses substantially.

In addition, Network Address Translation (NAT) permits Internet service providers and enterprises to masquerade private network address space with only one publicly routable IPv4 address on the Internet interface of a customer premises router (CPE), instead of allocating a public address to each network device.

4 Problems

4.1 IPv4 Exhaustion

Address exhaustion is the depletion of the pool of unallocated IPv4 addresses. Because there are fewer than 4.3 billion addresses available, depletion has been anticipated since the late 1980s, when the Internet started to experience dramatic growth. This depletion is one of the reasons for the development and deployment other alternatives solutions, like IPv6.

The main market forces that accelerated IPv4 address depletion included the rapidly growing number of Internet users, always-on devices, mobile devices and recently the Internet of Things (IoT).

The Internet Engineering Task Force (IETF) created the Routing and Addressing Group (ROAD) in November 1991 to respond to the scalability problem caused by the classful network allocation system in place at the time. The anticipated shortage has been the driving factor in creating and adopting several new technologies, including NAT, CIDR in 1993, and IPv6 in 1998.

IPv6, the successor technology to IPv4 which was designed to address this problem, supports approximately 3.4×10^{38} network addresses.

Although as of 2008 the predicted depletion was already approaching its final stages, most providers of Internet services and software vendors were just beginning IPv6 deployment at that time.

The top-level exhaustion occurred on 31 January 2011. Four of the five RIRs have exhausted allocation of all the blocks they have not reserved for IPv6 transition; this occurred on 15 April 2011 for the APNIC, Asia-Pacific, on 14 September 2012 for RIPE NCC, Europe, Middle East and Central Asia, on 10 June 2014 for LACNIC, Latin America and the Caribbean, and on 24 September 2015 for ARIN North America.

Individual ISPs still had unassigned pools of IP addresses, and could recycle addresses no longer needed by their subscribers. Each exhausted its pool of available addresses at different times.

4.1.1 IPv4 Exhaustion - Reasons

While the primary reason for IPv4 address exhaustion is insufficient capacity in the design of the original Internet infrastructure, several additional driving factors have aggravated the shortcomings. Each of them increased the demand on the limited supply of addresses, often in ways unanticipated by the original designers of the network.

1- Mobile devices

As IPv4 increasingly became the *de facto* standard for networked digital communication and the cost of embedding substantial computing power into hand-held devices dropped. Mobile phones have become viable Internet hosts. Even new specifications of 4G devices require IPv6 addressing.

In today's Internet, the "things" in IoT, each needs a traditional IP address, because the things are mostly servers and switches, firewalls and routers, laptops, phones and tablets with IP to IP connectivity.

2- Always-on connections

Throughout the 1990s, the predominant mode of consumer Internet access was telephone modem dial-up. The rapid increase in the number of the dial-up networks increased address consumption rates. Although it was common that the modem pools, and as a result, the pool of assigned IP addresses were shared amongst a large customer base. By 2007, however, broadband Internet access had begun to exceed 50% penetration in many markets.

Broadband connections are always active, as the gateway devices (routers, broadband modems) are rarely turned off, so that the address uptake by Internet service providers continued at an accelerating pace.

3- Internet demographics

There are hundreds of millions of households in the developed world. In 1990, only a small fraction of these had Internet connectivity. Just 15 years later, almost half of them had persistent broadband connections. The many new Internet users in countries such as China and India are also driving address exhaustion.

4- Inefficient address use

Organizations that obtained IP addresses in the 1980s were often allocated far more addresses than they actually required, because the initial classful network allocation method was inadequate to reflect reasonable usage. For example, large companies or universities were assigned class A address blocks with over 16 million IPv4 addresses each, because the next smaller allocation unit, a class B block with 65,536 addresses, was too small for their intended deployments.

Many organizations continue to utilize public IP addresses for devices not accessible outside their local network. From a global address allocation viewpoint, this is inefficient in many cases, but scenarios exist where this is preferred in the organizational network implementation strategies.

Due to inefficiencies caused by subnetting, it is difficult to use all addresses in a block. The host-density ratio, as defined in RFC 3194, is a metric for utilization of IP address blocks, that is used in allocation policies.

4.2 Internet Routing Security

Internet routing security is one of the pressing issues in the Internet. It encompasses the correct announcement and propagation of IP prefixes between the domains or - using the Internet terminology - Autonomous Systems (AS).

BGP (Border Gateway Protocol) is the protocol that manages the advertisement and propagation of prefixes between the different domains.

BGP configuration is mostly done via out-of-band mechanisms where network operators tell each other which prefixes to announce among themselves. Hence, an accidental misconfiguration or a malicious attacker controlling a BGP router can divert traffic to networks which should not receive it or make ranges of IP addresses unavailable (and effectively denying global services). This attack is commonly known as BGP hijacking and can be accomplished forging BGP announcements and propagating them to neighboring AS's.

There were many incidents for BGP hijacking, one of the most recent one happened November 2018

On November 12th, 2018, between 1:00 PM and 2:23 PM PST, some customers in the USA noticed issues connecting to Google G-Suite, Google Search as well as Google Analytics. The reason was that the traffic (destined to Google) was getting routed and dropped at China Telecom!

That was a severe denial of service to Google services. Some analysis indicated that the origin of this leak was the BGP peering relationship between a Nigerian provider, and China Telecom. The Nigerian provider has a peering relationship with Google via IXPN in Lagos and has direct routes to Google, which leaked into China Telecom. These leaked routes propagated from China Telecom, via some transit ISPs.

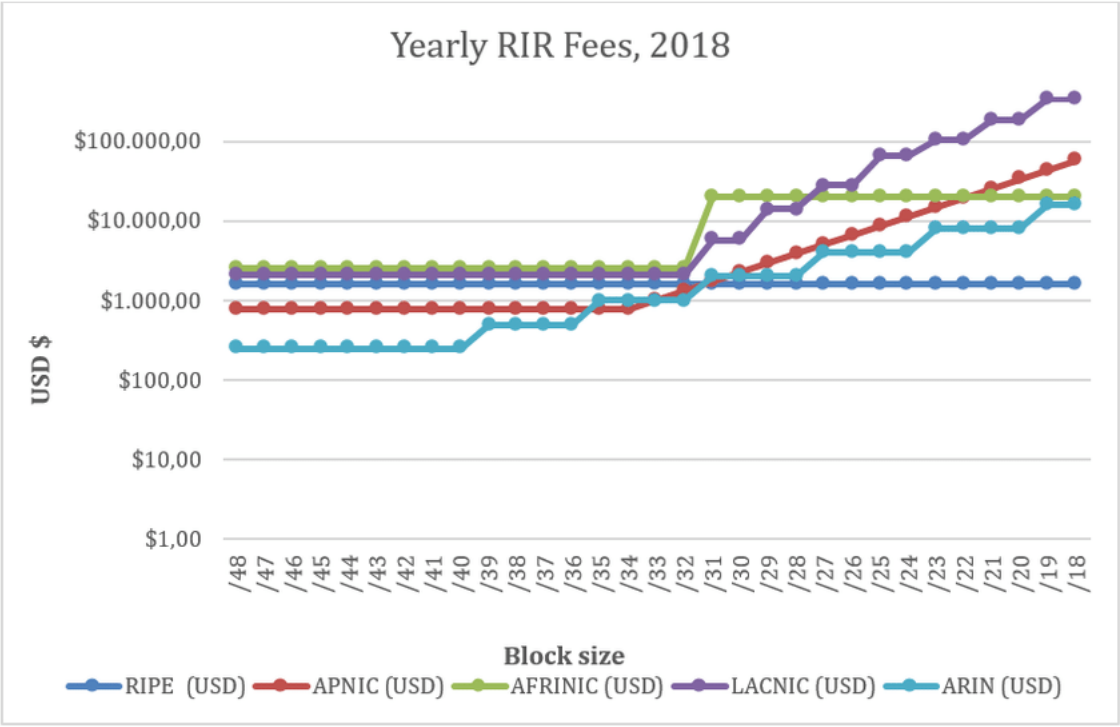
Numerous similar incidents taking place in 2017, were reported here:
https://www.internetsociety.org/blog/2018/01/14000-incidents-2017-routing-security-year-review/

4.3 Central Authorities in IPv4 and IPv6

The current centralized system to manage the global pool of IP addresses is centralized in five transnational organizations, the Regional Internet Registries (RIRs). Each of these RIRs manage the address pool for a large number of countries. Because the RIRs are private organizations, they are subject to the legal framework of the country where they are based. This configuration results in a jurisdictional overflow from the legal framework of the countries where the RIR is based to all the countries that the RIRs are serving (the countries served by the RIRs de facto become subjects of the legal system of the country where the RIR is hosted).

Beside the centralized nature of address allocation, there is always the burden of recurring maintenance or renewal fees as well as the relatively hard re-allocation of resources among the different RIRs.

The below figure shows the yearly RIR fees as of last year (2018):



5 Solution Attempts

5.1 IPv4 Exhaustion

5.1.1 Before Exhaustion

Efforts to delay address space exhaustion started with the recognition of the problem in the early 1990s and the introduction of a number of stop-gap refinements to make the existing structure operate more efficiently, such as CIDR, NAT as well as strict usage-based allocation policies.

Other technologies include:

- Use of NAT which allows a private network to use one public IP address and permitting private addresses in the private network
- Use of private network addressing
- Name-based virtual hosting of web sites
- Tighter control by regional Internet registries on the allocation of addresses to local Internet registries
- Network renumbering and subnetting to reclaim large blocks of address space allocated in the early days of the Internet, when the Internet used inefficient classful network addressing.

5.1.2 After Exhaustion

1- Reclamation of unused IPv4 space

Before and during the time when classful network design was still used as allocation model, large blocks of IP addresses were allocated to some organizations. Since the use of CIDR, IANA could potentially reclaim these ranges and reissue the addresses in smaller blocks.

ARIN, RIPE NCC and APNIC have a transfer policy, such that addresses can get returned, with the purpose to be reassigned to a specific recipient. However, it can be expensive in terms of cost and time to renumber a large network, so these organizations are likely to object, with legal conflicts possible. Even if all of these were reclaimed, it would only result in postponing the date of address exhaustion.

Similarly, IP address blocks have been allocated to entities that no longer exist and some allocated IP address blocks or large portions of them have never been used. No strict accounting of IP address allocations has been undertaken, and it would take a significant amount of effort to track down which addresses really are unused, as many are in use only on intranets.

Some address space previously reserved by IANA has been added to the available pool. There have been proposals to use the class E network range of IPv4 addresses (additional 268.4 million addresses to the available pool) but many computer and router OSs and firmware do not allow the use of these addresses. For this reason, the proposals have sought not to designate the class

E space for public assignment, but instead propose to permit its private use for networks that require more address space than is currently available through RFC 1918.

Several organizations have returned large blocks of IP addresses like Stanford University, who relinquished their Class A IP address block in 2000, making 16 million IP addresses available.

2- IP addresses Marketplaces

The creation of markets to buy and sell IPv4 addresses has been considered to be a solution to the problem of IPv4 scarcity and a means of redistribution. The primary benefits of an IPv4 address market are that it allows buyers to maintain undisrupted local network functionality.

The creation of a market in IPv4 addresses would only delay the practical exhaustion of the IPv4 address space for a relatively short time, since the Internet is still growing.

Case: Microsoft bought 666,624 IPv4 addresses from Nortel's liquidation sale for 7.5 million dollars.

5.1.3 Transition mechanisms

As the IPv4 address pool depletes, some ISPs will not be able to provide globally routable IPv4 addresses to customers. Nevertheless, customers are likely to require access to services on the IPv4 Internet. Below are some transition mechanisms used to overcome the exhaustion, as seen; some of these mechanisms have been developed for providing IPv4 service over an IPv6 access network.

1- ISP-level IPv4 NAT

ISPs may implement IPv4 NAT within their networks and assign private IPv4 addresses to customers. This approach may allow customers to keep using existing hardware.

However, the allocation of private IPv4 addresses to customers may conflict with private IP allocations on the customer networks. Furthermore, some ISPs may have to divide their network into subnets to allow them to reuse private IPv4 addresses, complicating network administration.

2- NAT64

It translates IPv6 requests from clients to IPv4 requests. This avoids the need to provision any IPv4 addresses to clients and allows clients that only support IPv6 to access IPv4 resources.

However, this approach requires a DNS server with DNS64 capability and cannot support IPv4-only client devices.

3- DS-Lite

'Dual-Stack Light' uses tunnels from the customer premises equipment (CPE) to a network address translator at the ISP. The CPE encapsulates the IPv4 packets in an IPv6 wrapper and sends them to a host known as the *AFTR element* that de-encapsulates the packets and performs network address translation before sending them to the public Internet. The NAT in the AFTR uses the IPv6

address of the client in its NAT mapping table. This means that different clients can use the same private IPv4 addresses, therefore avoiding the need for allocating private IPv4 IP addresses to customers or using multiple NATs.

4- Address plus Port

It allows stateless sharing of public IP addresses based on TCP/UDP port numbers. Each node is allocated both an IPv4 address and a range of port numbers to use. Other nodes may be allocated the same IPv4 address but a different range of ports. The technique avoids the need for stateful address translation mechanisms in the core of the network, thus leaving end users in control of their own address translation.

5.1.4 IPv6 perceived as a Long-term Solution

Deployment of IPv6 was perceived as the standards-based solution to the IPv4 address shortage.

IPv6 is endorsed and implemented by all Internet technical standards bodies and network equipment vendors. It encompasses many design improvements, including the replacement of the 32-bit IPv4 address format with a 128-bit address which provides an addressing space without limitations for the foreseeable future.

IPv6 has been in active production deployment since June 2006, after organized worldwide testing and evaluation in the '6bone' project ceased. Interoperability for hosts using only IPv4 protocols is implemented with a variety of IPv6 transition mechanisms as highlighted in the previous paragraph.

But despite being identified as the long-term solution, IPv6 is still on a low-adoption stage. It requires a significant investment of resources, and poses incompatibility issues with IPv4, as well as certain security and stability risks.

5.1.5 TUBA and CLNP

TCP & UDP with Bigger Addresses (TUBA) is based on CLNP (ISO 8473) and ES-IS (ISO 9542) control protocols by providing the operation of TCP transport and UDP over CLNP.

TUBA depended on a long-term migration proposal based on a gradual update of Internet Hosts (to run Internet applications over CLNP) and DNS servers (to return larger addresses). This proposal required routers to be updated to support forwarding of CLNP (in addition to their regular IP forwarding).

TUBA proposed that existing Internet transport and application protocols continue to operate unchanged, except for the replacement of 32-bit IP addresses with larger addresses. The use of larger addresses will have some effect on applications, particularly the DNS. TUBA did not mean having to move over to OSI completely. It would mean only replacing IP with CLNP. TCP, UDP, and the traditional TCP/IP applications would run on top of CLNP.

The main arguments in favor of TUBA are that routers already exist which can handle the network-layer protocol, that the extensible addresses offer a wide margin of "future-proofing" and that there is an opportunity for convergence of standards and products.

However, this proposal did not require encapsulation, translation of packets nor address mapping. IP addresses and NSAP addresses may be assigned and used independently during the migration period. Routing and forwarding of IP and CLNP packets may be done independently.

5.1.6 ATN and CLNP

More than a decade ago, there was a serious attempt to bring back CLNS protocol suites to life by the Indonesian air transportation authorities.

The Communication Navigation Surveillance/Air Traffic Management (CNS/ATM) standards set forth by the International Civil Aviation Organization (ICAO) for its contracting States, necessitates Indonesian air transportation authorities to implement the Aeronautical Telecommunication Network (ATN) for its air traffic management.

To provide a technological assistance, Badan Pengkajian dan Penerapan Teknologi (BPPT) started a joint research program with Swiss German University 'SGU' in 2007 to develop a complete ATN TP4/CLNP networking suite for GNU/Linux systems.

The BPPT-SGU team started developing the CLNP and the EIRP that were still in isolation in the network layer in 2007.

In 2008, the BPPT-SGU continued the program with the integrations and the testing of the CLNP and the EIRP as well as the design and the implementation of the CLTP and the COTP-TP4.

As a BPPT-SGU sub-team in charge of integration and testing of the CLNP, they linked the protocol with the data-link layer by utilizing the mechanisms to receive and transmit frames in the Ethernet 802.3 frame format having IEEE 802.2 LLC header. In addition, they developed a new communication domain PF_ATN for the BSD socket to link CLNP with the application layer.

Functions of the protocol are integrated as a single Linux kernel module which is coded according to the Linux kernel coding standards.

That was in accordance with the international agreement, both world-wide as well as in the Asia Pacific Region, on the technologies used for air traffic management, all countries signatories of the Convention on International Civil Aviation should conform to the new ICAO standards CNS/ATM, that was endorsed by the Tenth Air Navigation Conference in 1991. One of the reasons was to anticipate the increasing amount of air traffic in the future years without neglecting the flight safety aspects. The CNS/ATM operational concept encompasses two time periods.

The first period, up to 2015, were the development of scenarios based on the more realistic assumptions concerning applicability and technological capabilities and accommodates early implementation steps.

The second, from 2015 to 2025 will address scenarios which incorporate the more visionary options for air traffic management.

The attempt went well but their final conclusion was that the objective to build a complete ATN end-system necessitates further work to be done. Some functions of CLNP have yet to be developed. Besides that, the development of the CLTP and the COTP Class 4 are still in the initial stage that need further development. Moreover, correctness and conformance testing need to be conducted on the complete networking suite.

5.2 Internet Routing Security

5.2.1 Resource Public Key Infrastructure, RPKI

The IETF has come up with a solution to the Internet routing security: RPKI.

RPKI is a repository where the legitimate owners of IP prefixes, AS numbers and ROAs (Route Origin Authorization, a certificate to allow an AS to announce an IP prefix) are recorded.

Unfortunately, the global deployment of the RPKI is slower than expected with only one tenth (1/10) of the total Class-C subnets (/24) owned by the five RIRs being protected by the RPKI. The reasons of this have been mainly:

- Centralized operations: Certification Authorities (CAs) hold ultimate control of resources in the RPKI. Since IP addresses are a critical asset of RPKI's participants (especially ISPs), they would like to have a higher degree of control over them, but without losing the security of being certified by a CA, i.e. balanced power between users and CAs.
- Management complexity: PKIs are cumbersome to manage, like the case of key rollover. In addition, deploying these extensions is not trivial and requires trained staff and investment.
- Exposure of business relationships through peering agreements in the RPKI. In addition, the RPKI faces implementation and transparency challenges.

5.2.2 IPchain: Blockchain-based Solution

IPchain is a blockchain that stores IP addresses allocation and delegation data. IPchain eases the deployment of internet routing security mechanisms.

When compared to the RPKI, IPchain stands out with:

- The ability to create flexible trust models, providing a different balance of power between CAs and downstream users.
- Simplified management, especially with common operations like key rollover.
- Auditability: blockchain's append-only ledger can detect possible configuration errors even before a modification.

Like crypto tokens and coins; IP addresses share some fundamental characteristics such as uniqueness or divisibility. Thus, IPchain allows its participants to exchange IP prefixes just like tokens exchange happens in blockchain.

This way, an ISP can record its IP addresses and who can advertise them. When another ISP receives a BGP update including these addresses, it can tell if they come from the legitimate source.

IPchain existing prototype allows allocating and delegating IP prefixes by means of blockchain transactions and uses a Proof of Stake consensus algorithm to randomly select block signers among all the holders of IP addresses.

One argument against IPchain, or any similar initiative, is the 'what is in it' for the end customer to adhere to that framework. Though the benefits are obvious and enormous, but when it comes to an enterprise (or a service provider) work force who are always overwhelmed with their daily tasks; it would be a real luxury to take the decision, learn thoroughly and integrate their legacy operation procedures with a new mechanism, without the existence of an appealing rewarding structure, thing that is always being missed in the era of private blockchains.

5.3 Central Authorities

Among the initiatives born to overcome the central allocation of IP addresses blocks was: 'InBlock'.

InBlock is a blockchain-based distributed governance body that aims to provide decentralized management of IP addresses.

Beside decentralization, InBlock also aims to fulfill the same objectives as the current IP address allocation system, namely: uniqueness, fairness, conservation, aggregation, registration and minimized overhead.

InBlock is implemented as a Decentralized Autonomous Organization 'DAO', i.e. as a set of blockchain's smart contracts in Ethereum. Any entity may request an allocation of addresses to the InBlock registry by solely performing a crypto currency transfer to the InBlock. The fee

required, along with the annual renewal fee, serves as a mechanism to deter stockpiling and other wasteful practices.

It is worth mentioning that InBlock is not a public platform that is widely adaptable, instead it is meant to be a way to conduct experiments on distributed address management as a starting point to inform future directions in this area.

From the conclusion of the InBlock initiative, we can quote:

''We propose to experiment with InBlock for IPv6 address allocation. We believe this experiment will provide useful hand-on experience about how blockchains can be used for managing allocations.''

6 JACS

6.1 A little bit of history

If we had the luxury of starting over from scratch, most likely we would have based the Internet on a new datagram internet protocol with much larger multi-level address structure. In principle, there are many choices available for a new datagram internet protocol. For example, the current IP could be augmented by addition of larger addresses, or a new protocol could be developed. However, the development, standardization, implementation, testing, debugging and deployment of a new protocol (as well as associated routing and host-to-router protocols) would take a very large amount of time and energy, and is not guaranteed to lead to success.

6.2 CLNP versus IP

There is already such a protocol available. In particular, CLNP, that is very similar to IP, and offers the required datagram service and address flexibility, but that came with some differences as well.

CLNP addresses are assigned to nodes, not interfaces. This is the host-based addressing. All interfaces (even multi-access interfaces like LAN) are unnumbered and each node has a "loopback" interface with a single address (similar to the host /32 prefix in IPv4). While the concept was highly interesting and marginally more useful than subnet-based IP architecture, it also imposed additional burden on hosts and routers:

- Hosts and routers had to run a host-to-router protocol among themselves that enables routers to find adjacent hosts and helps hosts to find the nearest router, this is the ES-IS protocol that basically does a similar job to what Router Advertisement (RA) does in IPv6.
- ES-IS provides great failover, redundancy (in terms of first-hop routers) and hosts mobility
- Routers then have to propagate all hosts reachability information throughout the area so that each router in the area has to know the location of all hosts within the same area.

CLNP intra-area forwarding has some elements that could be similar to bridging (packets are forwarded based on host ID) but works as true routing (layer-2 encapsulation is changed and TTL is decreased whenever a packet is forwarded by a CLNP router). The number of hosts within an area is obviously limited by the routers' capabilities (and can be quite limited in some actual CLNP implementations); to scale, CLNP introduced a concept of areas, which are almost identical to IP summary routes.

Per-node addresses nicely solve intra-area multihoming. A host is always reachable through a single address, even if it has more than one interface. When an interface fails, the existing sessions are not disrupted (as they originate from an address that belongs to the node itself, not the failed interface).

In the IP world there are some solutions that could provide the same result for multi-homing, like: NIC bonding, multi-chassis port channel and others.

6.3 Limitations of previous attempts

All previous attempts to find alternatives to the TCP/IP stack were meant to solve part of the problem, like TUBA. TUBA ran TCP over CLNP on an end-to-end basis, thus one of the main applications of TUBA was to run it inside the same domain (intra-domain) and not between domains as there was no intention to translate or tunnel CLNP packets into IP packets.

As opposed to all previous attempts and suggestions; JACS is built as a full solution that details, not only, how intra-domain communications will occur but - more importantly - how communications will happen between the dispersed JACS islands over the public Internet that is purely based on IP.

In all previous attempts to change the addressing structure and when it was left to the standards bodies or even open-source initiatives, they haven't made the right intended progress.

One of the reasons was the end client, who are - regardless of their scale or even their burning heat – always required to put a lot of time and effort to test incomplete initiatives.

Consider the case of IPv6, even if a service provider or operator doesn't need to acquire IPv6 addresses, maybe because their current IPv4 blocks are still doing their intended work and there is no pressing issue to acquire IPv6, they are still required to do so and moreover, pay for high-end gear in order to be able to work around the whole IPv4; i.e. carrier-grade NAT devices, IPv6 into v4 tunneling and so forth.

6.4 Introducing JACS

With JACS, the hard work and cost of having a valid alternative to IPv4 that would be globally recognizable is done through JACS and its community.

If you're a valid IPv4 block owner, you're allowed to acquire JACS blocks totally free of charge, relative to your IPv4 holdings, that will be more than enough for your current and probably needs for three decades to come. (Proof of ownership that will be explained later in this paper)

If you are not a current IPv4 block owner, you can still acquire JACS blocks for your project, company, organization or even if you intend to run and operate a service provider or telco operator.

The cost associated with a JACS block is minimal compared to the cost of an IPv4 or even an IPv6 address, moreover you don't need to worry about any maintenance or operation cost, meaning that the cost per JACS block is a once-off, life-time cost.

You'll be associated block(s) according to your need over the blockchain.

There will be no refusal or need for more justification, you need it; you pay for it; you will then get it.

According to your usage history, and because it's all recorded permanently over the blockchain, your assigned block(s) may be banned globally if it turns out that you're abusing their usage intentionally.

There will be a code of usage and ethics that will be an integral part of the blockchain consensus, meaning it's and all its future edits are subject to the community, if the community agrees to change part of it or even change it all altogether, this will be done and applied accordingly.

There will be no central authority as IANA or the different regional and local registries with IP addresses, to request blocks from, justify to, accept their implied policies or whatever the process is, it's all done automatically and decentralized over the blockchain.

6.5 Roadmap and development timeline

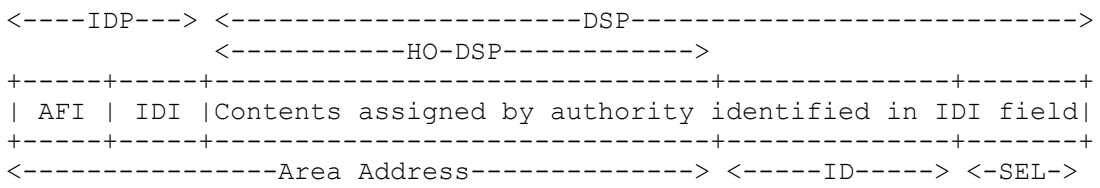
7 NSAP Address

The NSAP Address is formally defined in ISO/IEC 8348. It is the name of a Network Service Access Point (NSAP) located in an End System, and uniquely identifies that NSAP.

From a routing perspective (as in ISIS 'Intermediate System to Intermediate System' routing protocol for instance); ISO addresses are generally subdivided into:

- Area address
- System identifier (ID)
- NSAP selector (SEL)

The area address identifies both the routing domain and the area within the routing domain. Generally, the area address corresponds to the IDP plus a high-order part of the DSP (HO-DSP) as detailed below:



The ID field may be from one to eight octets in length but must have a single known length in any particular routing domain.

Each router is configured to know what length is used in its domain. The SEL field is always one octet in length. Each router is therefore able to identify the ID and SEL fields as a known number of trailing octets of the NSAP address.

The area address can be identified as the remainder of the address (after truncation of the ID and SEL fields). It is therefore not necessary for the area address to have any particular length -- the length of the area address could vary between different area addresses in a given routing domain.

As we discussed, there is a balance that must be sought between the requirements on NSAPs for efficient routing and the need for decentralized NSAP administration.

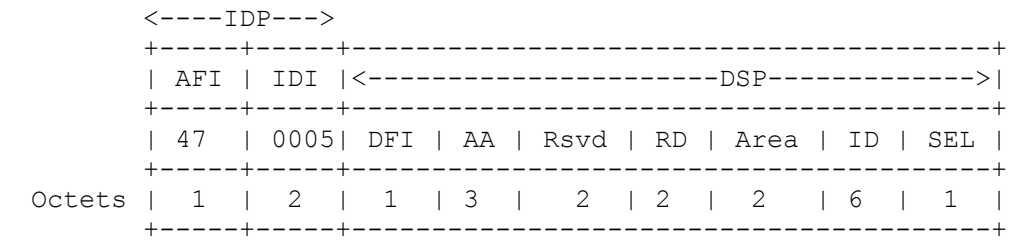
Note: There may be some confusion between an NSAP address and a NET (Network Entity Title) address, according to the ICAO ATN addressing specifications:

An NSAP address is a 20-octet string used to uniquely identify and locate a given NSAP (i.e. a network service user).

While a NET is a 20-octet string used to uniquely identify and locate a network layer entity of a system (Intermediate 'router' or end system) and thus, in networking terms, is used to identify the system itself. NET is formally defined in ISO/IEC 8348. NETs are syntactically identical to NSAP Addresses and they are allocated from the same address space. A NET differs from the NSAP address assigned to the same system only in the last octet, i.e. the network selector (N-Sel) field value.

7.1 NSAP Structure - A little bit of history

Let us examine the NSAP structure from GOSIP Version 2; it shows how the above mentioned balanced requirements might be met. The AFI, IDI, DFI and AA fields provide for administrative decentralization. The AFI/IDI pair of values 47.0005 identify the US Government as the authority responsible for defining the DSP structure and allocating values within it



IDP Initial Domain Part:

- **AFI** Authority and Format Identifier
- **IDI** Initial Domain Identifier

DSP Domain Specific Part:

- **DFI** DSP Format Identifier
- **AA** Administrative Authority
- **Rsvd** Reserved
- **RD** Routing Domain Identifier
- **Area** Area Identifier
- **ID** System Identifier
- **SEL** NSAP Selector

In addition to GOSIP Version 2 authority under 47.0005, there also existed the ANSI format under the Data Country Code for the US (DCC=840) as well as formats assigned to other countries and ISO members.

There were some cases, where an entity would prefer to use a country- or area- specific format rather than the US GOSIP format.

GOSIP Version 2 defines the DSP structure as shown (under DFI=80h) and provides for the allocation of AA values to administrations. Thus, the fields from the AFI to the AA, inclusive, represent a unique address prefix assigned to an administration.

ANSI specifies the structure of the DSP for NSAP addresses that use an Authority and Format Identifier (AFI) value of **39** decimal, which identifies the "ISO-DCC" (data country code) format, in which the value of the Initial Domain Identifier (IDI) is **840** decimal, which identifies the ANSI. This DSP structure is identical to the structure that is specified by GOSIP Version 2.

The AA field is called "org" for organization identifier in the ANSI standard, and the ID field is called "system". The ANSI format, therefore, differs from the GOSIP format illustrated above only in that the AFI and IDI specify the "ISO-DCC" format, and the "AA" field is administered by an ANSI registration authority rather than by the GSA. Organization identifiers may be obtained from ANSI.

In the low-order part of the GOSIP Version 2 NSAP format, two fields are defined in addition to those required by routing. These fields, RD and Area, are defined to allow allocation of NSAPs along topological boundaries in support of increased data abstraction. Administrations assign RD identifiers underneath their unique address prefix.

Routing domains allocate Area identifiers from their unique prefix. The result is:

* AFI+IDI+DFI+AA = Administration Prefix.

* Administration Prefix + Rsvd + RD = Routing Domain Prefix.

* Routing Domain Prefix + Area = Area Address.

This provides for summarization of all area addresses within a routing domain into one prefix. If the AA identifier is accorded topological significance (in addition to administrative significance), an additional level of data abstraction can be obtained.

On the other hand, in the EUR NSAP address registry for the ICAO ATN; we find that:

The IDP is always expressed as decimal digits. However, ISO/IEC 8348 permits NSAP Addresses in an ISO 6523-ICD domain to have either a binary or a decimal format for the remainder of the address - the Domain Specific Part (DSP). The format of the DSP is determined by the AFI.
All ATN NSAP Addresses have an AFI with the value **47** decimal. This AFI value is defined by ISO/IEC 8348 to imply an ISO 6523-ICD IDI with a binary format DSP.

All ATN NSAP Addresses have an IDI value of **0027** decimal. This value has been allocated by ISO to ICAO under the ISO 6523-ICD scheme.

An IDP of **470027** therefore forms the common NSAP Address Prefix to all ATN NSAP Addresses and NETs and effectively defines the ATN Network Addressing Domain, as a sub-domain of the Global Network Addressing Domain.

For the purposes of publication in a text format, ATN NSAP Addresses and NETs should be written as the character sequence "**470027+**", identifying the common prefix for all ATN NSAP Addresses, followed by the DSP expressed as a sequence of hexadecimal characters. The "+" sign is used as a separator between the decimal syntax IDP and the hexadecimal syntax DSP

7.2 IPv4 to NSAP

One of the most common ways that was used to convert IPv4 addresses to NSAP addresses was to:

* Take each octet of the quad-dotted notation of the IP address and add leading zeros so that we can add it to three digits, as follows:

Let us say the IPv4 address is: 192.168.1.2, where the first 2 octets are already 3 digits each, so we will prefix the last 2 octets with the required Zeros to make each octet at the 3 digits mark.

192.168.1.2 ➔ 192.168.**00**1.**00**2

We now have 12 digits, which you can easily rearrange into three groups of 4 digits, as follows:

192.168.**00**1.**00**2 ➔ 1921.6800.1002

The result in Hexadecimal is 6 octets or 48 bits in binary that can then be used as the unique ID of the resulting NSAP address.

The area prefix and NSEL suffix are then added to obtain the complete NSAP address, like for example:

FB.0000.1921.6800.1024.00

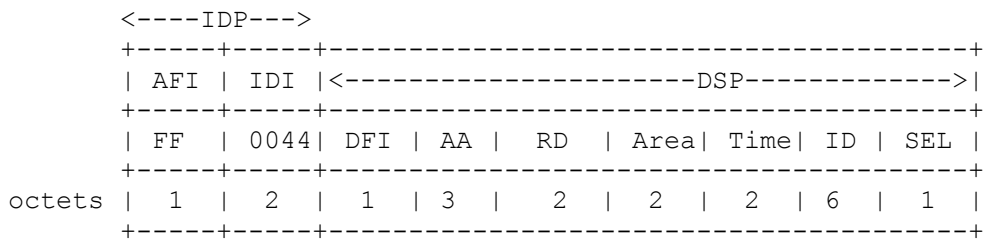
7.3 JACS NSAP

Inside the JACS platform, that will be detailed through the rest of this white paper starting from section 11.3; it is going to follow the GOSIP version 2 structure in terms of overall structure as well as treating the last 7 octets of the NSAP address as 6 octets ID and 1 last octet as the NSEL.

This leaves 13 octets prefix that JACS can manipulate for further NSAP addresses allocations.

JACS is starting from a clean slate in terms of NSAP addressing, so the abundance of addresses and the precise allocation scheme will reflect on a state-of-the-art allocation that is perfectly future and depletion proof.

Note: Unlike any previous specification of the NSAP address as a combination of decimal and hexadecimal representation; JACS will always specify the whole 20-octets NSAP address in Hexadecimal.



IDP Initial Domain Part:

- **AFI**: FF means JACS address allocated over JACS native blockchain (see roadmap).
- **IDI**: if set to 0000, it means that Geolocation allocation is disabled (as the case with all versions and releases of JACS over Ethereum), while if the IDI value is greater than zero; it indicates the country code that is based on country international codes found at: https://www.internationalcitizens.com/international-calling-codes/

DSP Domain Specific Part:

- **DFI**: Indicates what format of the Time field will be used. For JACS operation over Ethereum; it is the timestamp of the block, within which the address allocation transaction is included and confirmed. For JACS operation over its own blockchain; the Time will follow the MM-YY format indicating Month of the Year during which the address block is allocated to the requesting entity
- **AA**: assigned based on requesting entity identifier, it is derived from the Ethereum address of the requesting node (during JACS operation over Ethereum) so that it can preserve the contiguous assignment from the same block when that same entity requests more address blocks in the future. Similarly; the AA will be derived from JACS blockchain node IDs for JACS operation over its native chain
- **RD**: Routing domain identifier for all domains served by this same requesting entity
- **Area**: Area identified for all areas served by this same requesting entity
- **Time**: if the MM-YY format is used, for example '0520' means address being assigned during the month of May, 2020
- **ID**: System Identifier
- **SEL**: NSAP Selector

Version/Release	Blockchain	AFI	IDI
MVP	No	FA	0000
Pre-Alpha	Ethereum	FB	0000
Alpha	Ethereum	FC	0000
Beta	Ethereum	FD	0000
Official, version 1	Ethereum	FE	0000
Official, version 2	JACS	FF	Geo-enabled
Official, version 3	JACS	FF	Geo-enabled

Note: Detailed address structures for each version/release will be detailed in Section 10: JACS - Life Cycle

8 Blockchain

A blockchain is a decentralized and distributed digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the collusion of the network. This allows the participants to verify and audit transactions inexpensively. They are authenticated by mass collaboration powered by collective self-interests.

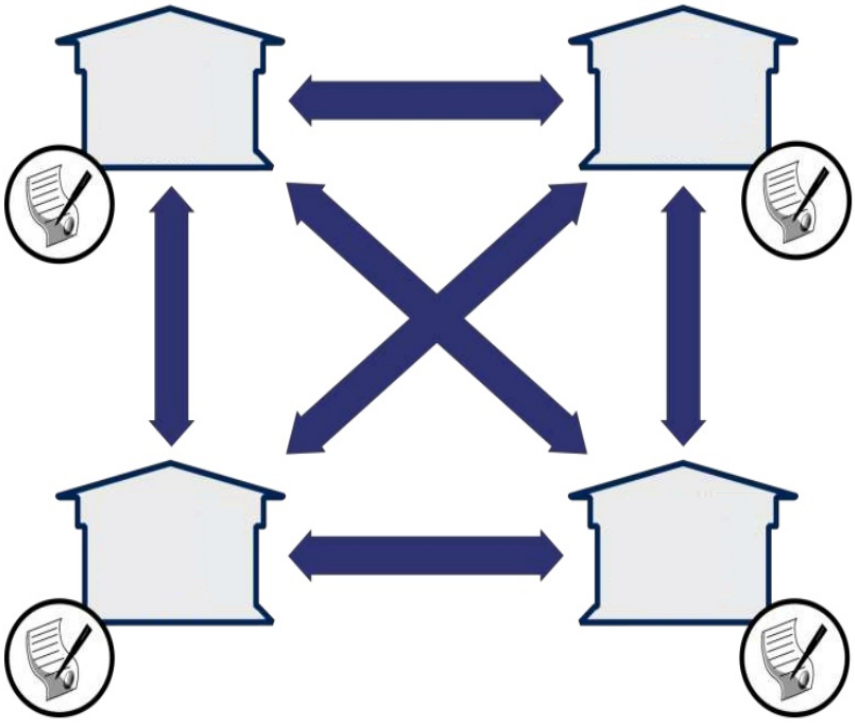
It is composed of a growing list of blocks, securely linked between each other through cryptography. Every block contains a hash pointer to a parent block, a timestamp and transactions' data.

There are two kinds of records: transactions and blocks. Blocks hold batches of valid transactions that are hashed and encoded into a Merkle tree. Each block includes the hash of the prior block in the blockchain, linking the two. The linked blocks form a chain. This iterative process confirms the integrity of the previous block, all the way back to the original genesis block (first block).

Blockchain has huge implications for a wide range of applications. It points the digital economy towards a new generation of internet - The Internet of Value - where people can send information and exchange valuable digital assets like currencies, property titles and identity information, in a trustworthy way. Blockchain transactions can happen without the involvement of centralised transaction management structures, like banks, stock exchanges and government agencies. It can store transactional history in a transparent and secure way, so that the chances of fraud, hacking and interference are eliminated. The technology could therefore revolutionize business transactions.

The addition of new valid block is determined through a distributed consensus mechanism. The consensus is an emerging artefact representing the agreement reached by over than thousands of nodes on the blocks added to the blockchain. The most popular consensus algorithms are Proof-of-Work (PoW) and Proof-of-Stake (PoS).

As opposed to PoW, that depends on the hashing power used to mine blocks; PoS depends on the number of stakeholders, the concentration of stake and also how much actual value is there in the blockchain.



Distributed Ledger where every node has the 'exact' copy of the ledger

8.1 General Functions of blockchain

Blockchain transactions do not carry only assets (or tokens) exchanged between the different nodes; they can carry any data or information that can be appended by the different nodes.

To elaborate, we can say that there are three main functions of blockchain:

- Storage for digital records (that is the feature of interest in this white paper)
- Exchanging digital assets (coins and/or tokens)
- Executing smart contracts:
    - Ground rules, terms and conditions recorded in code
    - The distributed network executes contract and monitors compliance
    - Outcomes are automatically validated without a third party

8.2 Transactions, accounts and data storage

8.2.1 Unspent Transaction Output (UTXO)

It was from Bitcoin's original paper created by 'Satoshi Nakamoto' where the world learnt about blockchain. Bitcoin state is represented by its global collection of UTXOs. The transfer of value in bitcoin is actioned through transactions. More specifically, a bitcoin user can spend one or more of their UTXOs by creating a transaction and adding one or more of their UTXOs as the transaction's input.

Firstly, bitcoin UTXOs cannot be partially spent. If a bitcoin user spends 0.5 bitcoin (using their only UTXO which is worth 1 bitcoin) they have to deliberately self-address (send themselves) 0.5 bitcoin in return change. If they don't send themselves change, they will lose the 0.5 bitcoin change to the bitcoin miner who mines their transaction.

Secondly, at the most fundamental level, bitcoin does not maintain user account balances. With bitcoin, a user simply holds the private keys to one or more UTXO at any given point in time. Digital wallets make it seem like the bitcoin blockchain automatically stores and organizes user account balances and so forth, but this is not the case.

In summary:

- The bitcoin blockchain does not hold account balances
- Bitcoin wallets hold keys to UTXOs
- If included in a transaction, an entire UTXO is spent, otherwise it's partially received back as change in the form of a new UTXO.

8.2.2 Account balances

On the other hand, for Ethereum (another public blockchain platform created to facilitate the development of smart contracts and applications); it is all about account balances.

The state of Ethereum is not an abstract concept; it is part of Ethereum's base layer protocol. It is a transaction-based 'state' machine; a technology on which all transaction-based state machine concepts may be built.

As with all other blockchains, the Ethereum blockchain begins life at its own genesis (first) block.

From this point onward, activities such as transactions, contracts, and mining will continually change the state of the Ethereum blockchain. In Ethereum, an example of this would be an account balance which changes every time a transaction, in relation to that account, takes place.

Importantly, data such as account balances are not stored directly in the blocks of the Ethereum blockchain.

There are two vastly different types of data in Ethereum; permanent data and ephemeral data. An example of permanent data would be a transaction. Once a transaction has been fully confirmed, it is recorded and is never altered. An example of ephemeral data would be the balance of a particular Ethereum account address. The balance of an account address is altered whenever

transactions against that particular account occur. It makes sense that permanent data, like mined transactions, and ephemeral data, like account balances, should be stored separately.

As discussed, Ethereum holds a set of accounts. Every account has an owner and a balance (some Ether).

Once identity is proved, any account holder can transfer Ether from their account to another. This is the Transaction.

8.3 Blockchain for JACS

Blockchain is a vital component in JACS platform, it performs many functions, like:

- Address allocation
- Address registry
- Route Origin verification and validation
- Security of allocation and advertisement, thus preventing any security breach in the Internet routing system, i.e. BGP Hijacking
- Crowdfunding (private Sale, Pre-ICO and public ICO)
- Rewarding for existing IPv4 blocks owners
- Incentivizing the community and early adopters

9 JACS and blockchain

JACS will use the Ethereum public blockchain in its early phases of operation as well as during the crowdfund.

The crowdfund will depend on deploying Smart Contracts with the locked value inside, so the escrow function will be transparent, and the outcomes will be clear as will be detailed in the crowd-fund section.

JACS phases that will leverage Ethereum in their operation will be:

- Pre-Alpha
- Alpha
- Beta
- Official Release, Version-1

9.1 Ethereum

Ethereum has a built-in Turing-complete programming language that allow developers to easily write smart contracts. Every operation in the network is triggered by transactions between accounts, that can be externally owned accounts (EOAs), owned and controlled by users, and contract accounts, associated to a smart contract which code and state stored with the account itself. Being a public blockchain, any party can create one or more EOAs and any party can run a smart contract in Ethereum.

Ethereum has implemented a PoW-based consensus mechanism. Miners are rewarded in Ether (Ethereum Cryptocurrency) for the storage and processing power they contribute to. Users who want to run a smart contract, or to interact with a smart contract, issue a transaction in the Ethereum network which includes a transaction fee payable to the miners. The value of the transaction fee is set by the user generating the transaction and should reflect the number of operation steps to be performed to accomplish a certain work and the priority that the user wants to get from the blockchain miners, as higher transaction fees imply that the transaction will be processed earlier by the miners. Transaction confirmation times are estimated around 10 to 15 seconds, depending on storage needs, code complexity and bandwidth usage.

9.1.1 Smart Contracts

The phrase smart contract was first coined by Nick Szabo. It refers to the verification, monitoring and execution of contracts including transfer of money using a software implementation. While it is possible to implement certain simple smart contracts in the original Bitcoin protocol, Ethereum was the first open Blockchain to be explicitly designed with programmability of smart contracts in mind.

It is designed to be a state transition machine with code being executed simultaneously by all miners on the decentralized Ethereum Virtual machine. Ethereum allows in principle for Turing-complete computations.

Smart contracts on Ethereum can be coded using a specialized programming language called Solidity and deployed through an initial transaction.

As indicated, smart contracts have their own specific address. A contract consists of state variables and functions. The functions can be called by additional transactions addressed to the contract to trigger changes in the state variables, make payments, etc. In order to support the storage, computation and communication costs of executing smart contracts there are fixed "gas" fees associated with each operation that must be paid by the caller.

9.1.2 Distributed Applications (dAPPs)

Ethereum is used to build dAPPs through smart contracts.

The smart contract executes some code when it receives transactions. It has a balance, some code, and some storage. This storage is persistent, and that's where the dAPP data resides.

For every transaction, the emitter needs to add some Ether, the *gas* (fuel of the Ethereum Virtual Machine, EVM). This is to motivate the miners to process the transaction. Miners ensure the network is reliable and are rewarded accordingly with some Ether.

There are instructions to read in storage, write in storage, etc. They all have a cost in gas, and that cost will constrain how much storage may be used.

The cost of each instruction in a Smart Contract will limit the amount of storage it uses. Ethereum allows for a theoretically infinite storage space, but we have to provide gas for every read/write operation.

This cost changes all the time: it depends on the network, the market and new developments of the Ethereum specs, but usually it is very costly to store 1 byte over the Ethereum network.

Most dAPPs running on Ethereum need to store and retrieve data, just like conventional (centralized) apps. The EVM does indeed allow saving variables and state in permanent storage.

9.2 JACS over Ethereum

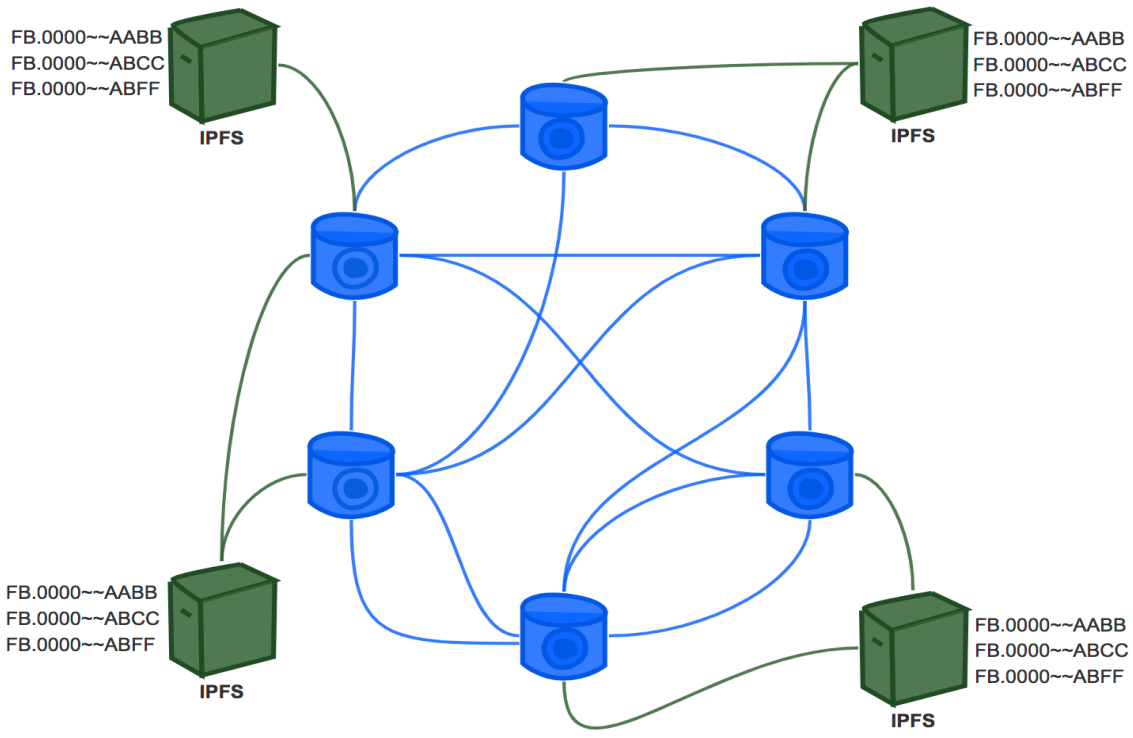
By using an existing blockchain, we can rapidly develop and deploy JACS as we only need to focus in the implementation of the registry service.

Due to the facilities provided to develop smart contracts, dAPPs and its relative maturity; Ethereum was our blockchain of choice.

When JACS operates over Ethereum, it needs to store the full range of JACS address blocks and ties the allocated blocks to their requesting nodes identifiers. As discussed; storing few bytes to the EVM could work well but for larger chunks of data; the costs are too high so we need to find a way where we can store data 'off-chain'. One way of doing this is using IFPS.

Inter-Planetary File System (IPFS) is a protocol designed to create a permanent and decentralized method of storing and sharing files. IPFS allows p2p storage and can be used as a distributed file system to store data.

JACS dAPP follows the off-chain data storage approach, thus instead of storing the data on-chain; JACS only stores the unique hash, that follows each JACS address block allocation, on the chain and then uses the hash to retrieve the data.



The requesting node will then receive the allocated address block that it can distribute it to all internal nodes, that it acts as their gateway, allowing each node to acquire unique global address out of that block (similar to the DHCP operation for IPv4 or IPv6 addresses).

9.3 Utility token

At the start of its operation, JACS will leverage the Ethereum public blockchain to perform all address allocation registry functions. This means all functionality lies in the blockchain, in the form of a group of smart contracts that run in the blockchain without human intervention.

Modifications to the information regarding the registry of JACS address blocks are triggered by blockchain transactions and subject to the consensus of the blockchain. Therefore, the smart contracts define what a valid transaction is and then all the nodes of the blockchain will enforce that only valid transactions modify the address allocation registry information. Once included in the blockchain, the allocation of a block is irrevocable.

As discussed; JACS is initially configured with the full blocks of globally routable JACS addresses to allocate. When a requesting node requests some address block, it uses its Ethereum account to perform this request. The request is basically a blockchain transaction that transfers a predetermined fee in JACS tokens to the smart contract address; hence comes the utility function of the JACS token.

JACS then verifies that the transaction is valid and that the fee has been correctly transferred.

Upon reception of the transaction, JACS code goes through its associated state (stored off-chain) and finds an address block that is not currently allocated. JACS associates the available block with the identity of the requesting entity and records the hash of the off-chain allocation on the blockchain (as explained in the previous section)

As opposed to IPv4 or IPv6 allocation, JACS address allocation is recorded permanently over the blockchain.

This way; the holder of the address blocks will never need to renew the allocation or abide to any renewal date.

There is a one-time fee to acquire an address block and there will never be any renewal or maintenance fees.

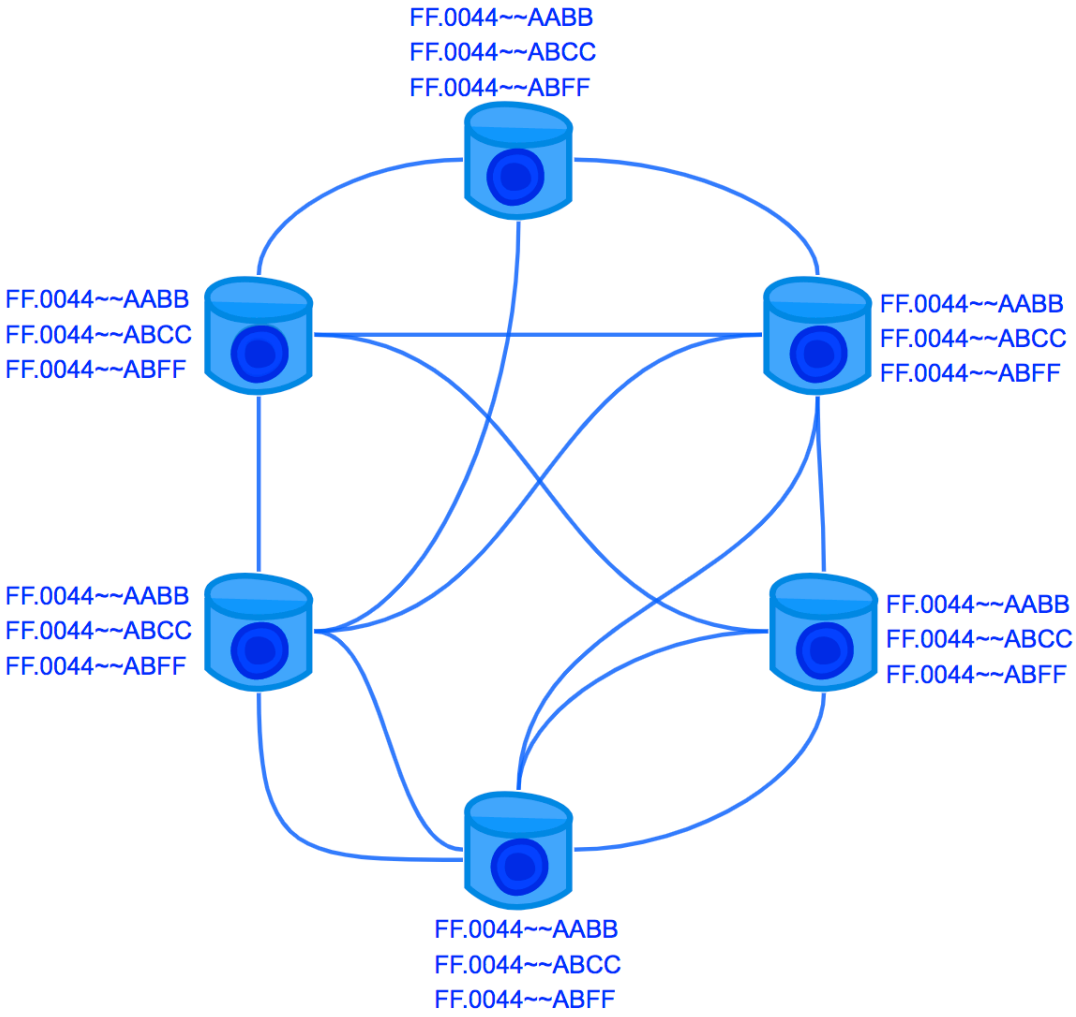
9.4 JACS native blockchain

Starting from its official release version 2; JACS will run over its native public blockchain instead of the Ethereum blockchain. At that time; all JACS tokens created as ERC-20 tokens will be swapped on 1-to-1 ratio to native JACS utility tokens over JACS native blockchain.

JACS, when running over its native public blockchain, will enjoy these features:

- No initial cost required to deploy any smart contract or dAPP

- No Transactional Fees. In the case of Ethereum; blocks miners determine which transactions to be included in the mined block according to the transaction fee offered.
- Delay: it could take minutes over the Ethereum blockchain for an NSAP block to be allocated and registered to the requesting entity, while over JACS chain; the delay will only depend on the block production rate, that is aimed to be very fast.
- JACS address blocks will be stored over the blockchain itself and there will be no need for IPFS



9.5 JACS rules

The blockchain is always the core component of JACS that will define the rules governing the allocation of JACS addresses.

These rules are as follows:

9.5.1 Abundance

Unlike all other address allocation mechanisms, JACS cares less about the conservation of its NSAP addresses or its depletion over time.

There will be no refusal or justification needed by any requesting entity; there are enough addresses for every single grain of sand on earth and even beyond and for all future needs.

That said, there will be no precautions taken by JACS to prevent or limit stockpiling, that is the accumulation of resources beyond the actual legitimate need of a requesting entity.

9.5.2 Uniqueness

JACS guarantees that the allocated blocks are unique as the allocation is purely done over blockchain.

Every time a requesting entity requests an address block via a proper transaction, the allocated block is tied to that entity and is stored permanently over the blockchain, so that it can never be assigned to another requesting entity. Moreover, the AA field of the JACS address block will be derived from the blockchain public address of the requesting node (Ethereum blockchain at the start of the operation then JACS native blockchain)

9.5.3 Aggregation

One of the critical factors of all address allocation mechanisms, is the need to preserve the size of global routing table.

Thus, addresses aggregation is crucial for the viability of the Internet routing system.

In the case of IPv4 or IPv6, the different RIR allocation policies promote aggregation through the preferred use of PA ‘Provider Assigned’ addresses, but they still allow PI ‘Provider Independent’ allocations.

These PI allocations are the main reason behind the explosion of the size of the global routing table as these PI entries cannot be efficiently aggregated most of the time.

In JACS, there is no central authority providing the addresses, like: IANA, RIRs, LIRs, Service Providers, ANSI or any Government, it is the public blockchain that assigns address blocks for all requesting entities, so there is no concept of PA or PI addresses as there is no ‘Provider’ in reality.

When a requesting entity requests a new block over blockchain (maybe for any operation expansion or future use); JACS will recognize the entity and can then grant the entity further blocks that are contiguous with its original holdings, thus allowing the efficient aggregation to happen on the Internet core routers.

It’s worth mentioning that part of the future development of JACS will explore how to leverage JACS native public blockchain to, possibly, replace the existing routing protocols that run over the Internet routers.

9.5.4 Anonymity

In JACS, blockchain identities are anonymous. Moreover, payments are done using JACS tokens.

This implies that JACS has no knowledge (hence no control) about the entity that receives the allocation. Requesting entities can still - voluntarily - attach their contact information or policies for each allocation as part of the metadata, however it’s not mandatory.

9.5.5 Fairness

Fairness in the context that the policies should be equally applied to all entities irrespectively of their location, nationality, size, or any other factor.

JACS naturally and inherently achieves this goal.

The role of the judge is taken by the blockchain consensus. The participating nodes verify the transactions and ensure that everything happens the way it should be.

10 JACS - Life Cycle

There will be many phases throughout the lifecycle of JACS, it starts with the Minimum Viable Product (MVP) then the Pre-Alpha that will be built as a dAPP over the Ethereum blockchain.

The Ethereum dAPP will still be the core component for the Apha version, Beta version and official release version 1.

Starting from official release version 2 JACS will leverage its own public blockchain.

10.1 MVP (Q2, 2019)

Where multiple nodes on the same LAN segment can communicate based solely on JACS/CLNP without requiring any IP addresses or TCP/IP protocol stack running on any node.

Nodes could be physical appliances or virtual machines. It's very important though to make sure there is a direct Ethernet segment linking all nodes.

CLNP is linked with the datalink layer by utilizing the mechanisms to receive and transmit frames in the Ethernet 802.3 frame format having IEEE 802.2 LLC header.

That said, we can expect that the following scenarios would not work:

- Public cloud instances running JACS between them (unless some feature like: 'dedicated host' is enabled so that the instances always reside on the same host)
- Nodes that hide behind firewalls or NAT devices

Required setup steps for the prototype could be found on Github: https://github.com/viaBlock

The MVP is based on the old ATN code with lots of modifications and enhancements to run the 10-years old code over the latest kernel and the latest Linux versions.

There are few scripts that help getting the code running on different nodes. Also, for the sake of facilitating testing the connectivity between nodes; a script exists that triggers a chat application between a specific node (server) and clients (all other nodes).

Messages could be sent from clients to the server, verified by the server and reflected back from the server to the clients.

As JACS NSAP addresses are 20 bytes long as detailed before, the IDP part of JACS addresses for the MVP is set as: **FA0000** (where AFI is: FA and IDI is: 0000; that means geolocation disabled)

Note that during the MVP, there is no blockchain involved and each participating node will append the MAC address of their interface (for example enp0s8 interface) over which they are communicating via CLNP to produce their full NSAP address.

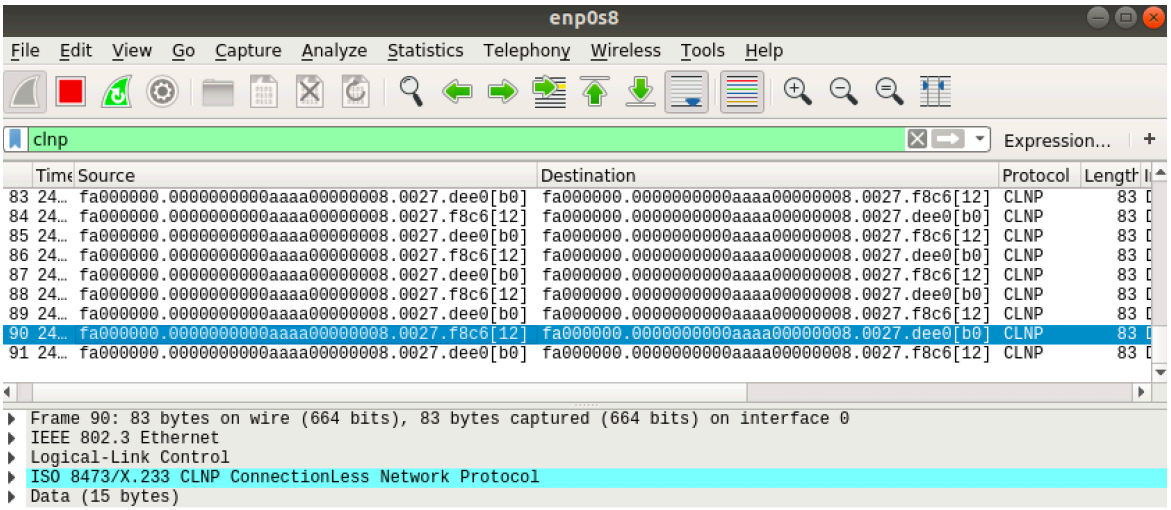
The DFI, AA, RD, Time and the first octet of the ID fields will all be Zeroed.

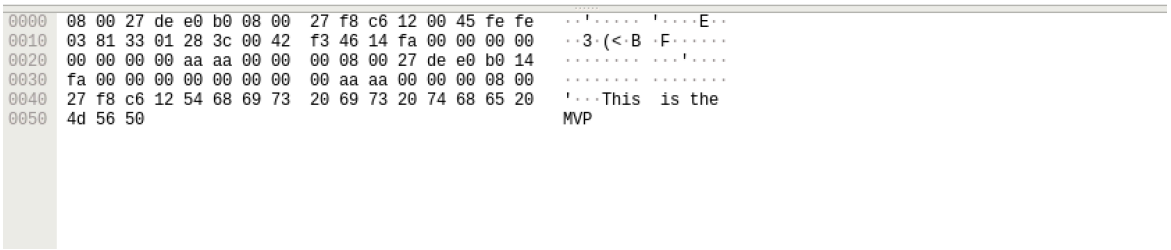
The Area will be set to **AAAA**, and the MAC address will be appended as the last 6 octets.

```
          <----IDP--->
          +-----+-----+-----------------------------------------+
          | AFI | IDI |<---------------------DSP------------->|
          +-----+-----+-----------------------------------------+
          | FA  | 0000| DFI | AA |  RD  | Area| Time| ID | SEL |
          +-----+-----+-----------------------------------------+
  octets  |  1  |  2  |  1  | 3  |   2  |  2  |  2  | 6  |  1  |
          +-----+-----+-----------------------------------------+
```

As seen below, a message '**This is the MVP'** was sent successfully from one client to the server.

Wireshark can be used to capture the CLNP packets, parse them and read the embedded messages as shown.





This shows that a typical NSAP address produced during the MVP will look like:

FA 0000 00 000000 0000 AAAA 0000 00 <MAC address>

10.2 Pre-Alpha version (Q4, 2019)

10.2.1 Scope & Address Allocation

In the Pre-Alpha version of JACS; nodes within a single domain will be able to communicate among themselves (**intra-domain**) leveraging purely JACS protocol stack.

JACS Pre-Alpha version will be supported over all major Linux distributions.

From a technical standpoint in order for a JACS-enabled domain to communicate with the public Internet; the domain must have its gateway node(s) that is/are dual-stack node(s). Each gateway is required to statefully translate JACS traffic to IP traffic to be forwarded over the Public Internet and vice versa. The gateway node could also have the function of a requesting node and/or a delegate as will be explained in this section.

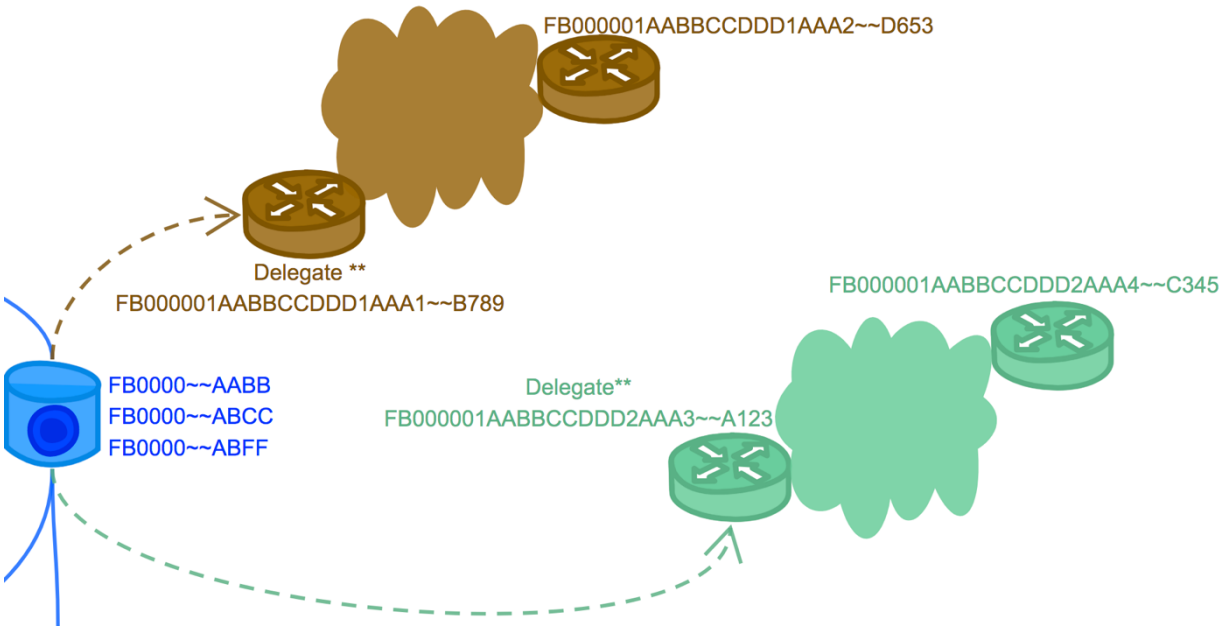
The operation normally starts by requesting some JACS address block to be allocated; this allocation is handled by the JACS dAPP.

A requesting node needs to send some JACS tokens (ERC-20) to a defined smart contract address for the contract to allocate an address block and tie it to the requesting node identifier.

The allocation fee per address block will be specified in US$; thus, the amount of JACS tokens that needs to be sent will vary according to the price of JACS token against US$ at the time of request.

The IDP part of JACS addresses, assigned during the Pre-Alpha version, is set as: **FB0000** (AFI is: **FB** and IDI is: **0000**; that means geolocation disabled) and the dAPP will assign the DSP part of the address as follows:

- DFI: Indicates that the Time field includes the timestamp of the Ethereum block.
- AA: derived from the Ethereum address of the requesting node
- RD: To be allocated and assigned locally by the requesting node to its domains' delegates. A domain delegate acts as their own domain gateway/exit and it is delegated the function of allocating and assigning JACS addresses from their parent requesting node
- Area: To be allocated and assigned locally by each domain delegate
- Time: the timestamp of the Ethereum block, within which the address allocation transaction is included and confirmed
- ID/SEL: Derived from the MAC address of a JACS-enabled interface of the node



As in figure, the requesting node (blue one on the left of the diagram) sent the proper fee in JACS tokens and acquired JACS address prefix: FB000001AABBCC~~ from over the blockchain.

As it is over the Ethereum blockchain, geolocation is disabled. The allocated address block covers lots of addresses that will be consumed by all downstream domains and areas hiding behind the requesting node, thus the requesting node is actually acting as their top 'parent' upstream node (gateway)

Each downstream domain has its own 'child' gateway node that will be the delegate of the parent gateway (the requesting node).

The green delegate node (FB000001AABBCCDDD2AAA3~~A123) is the gateway of the domain: DDD2 (green cloud).

The associated area of the delegate green node is: AAA3, and as we can see, for the other right-hand side green node, the same domain is: DDD2 (green cloud) but its area is different: AAA4

The same goes for the brown domain (DDD1), its delegate and other nodes.

As a general precaution and because there is no means for account recovery over the Ethereum dAPP; at least one backup requesting node should be specified to hold the same keys and account info as the main requesting node.

10.2.2 One-time, Gas-free allocation fee

Not only JACS relieves its users from any recurring maintenance or renewal fee, as explained earlier, but it also waives the gas amount that should be paid by a requesting node as it emits the 'request address' transaction that will trigger the operation of the relevant smart contract in the dAPP.

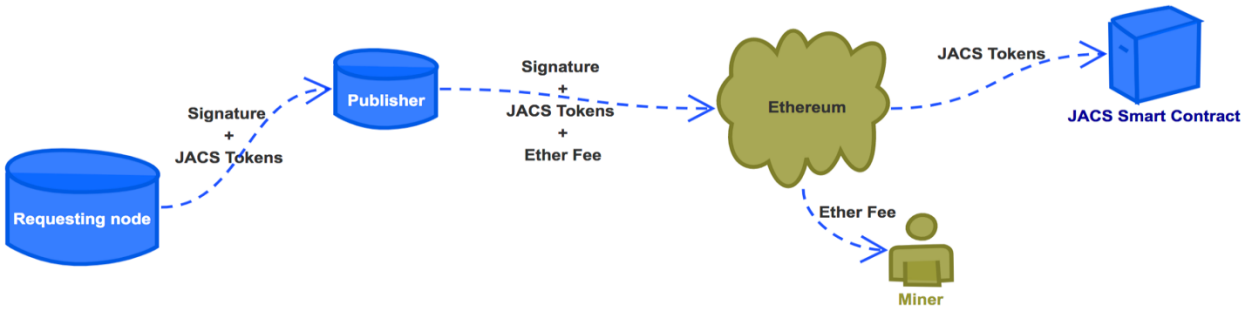
In other words; requesting nodes don't need to hold Ether in their Ethereum wallets in order to initiate any transaction over the blockchain, like for instance, exchanging JACS tokens for some address block(s).

To make this happen, there must be the ability to cryptographically prove that the requesting node is its actual account owner by giving their valid signatures, which will allow the transfer of JACS tokens from their account. The valid signature will be one argument of a transfer function that will be executed by a JACS intermediary node.

Getting the confirmation from a requesting node, on the form of their signature about their intent to send tokens, is a free operation.

The following sequence will take place:

- The requesting node signs the arguments to the transfer function (recipient, amount). That could be done by the node giving its signature on some data to an intermediary JACS payment node, named 'facilitator'
- The facilitator is an account that holds some Ether, thus able to do transactions on the Ethereum network, thus it can execute any function in the contract, like for example; transferViaSignature.
- This function takes the transfer data along with the requesting node's signature.
- The smart contract validates whether the signature is valid and does the token transfer.

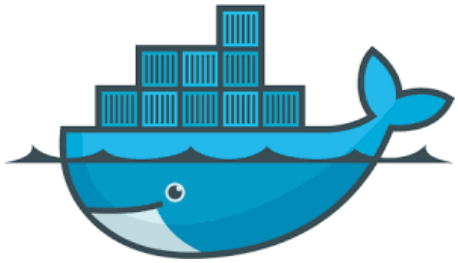


10.3 Alpha Version (Q1, 2020)

As in Pre-Alpha; the Alpha version of JACS will have an intra-domain scope as well as address allocation over the Ethereum dAPP.

Beside the operation over the major Linux distributions; Alpha version will allow the operation of JACS over latest Windows, MAC-OS operating systems as well as inside containers.

The IDP part of JACS addresses, assigned during the Alpha version, is set as: **FC0000** (AFI is: **FC** and IDI is: **0000**; that means geolocation disabled) and the dAPP will assign the DSP part of the address as the case with the Pre-Alpha version (requesting nodes and delegates).

10.4 Beta Version (Q2,2020)

10.4.1 New Internet Goal

Generally speaking, it is necessary for any change to be deployed in an incremental manner, allowing graceful transition from the current architecture without disruption of service

The ultimate goal of JACS involves transition to a new worldwide Internet which operates much as the current Internet, but with CLNP replacing IP and with JACS NSAP addresses replacing IP addresses.

One way to achieve the intended smooth transition would be to allow a solution, like TUBA to come into play as a transitional step, thus for a certain host in order to initiate communication with another host; the host will obtain a public address in the same manner as it normally does, except that the address would be larger (in our case: JACS address). In the legacy way, the host would contact the DNS server, obtain a mapping from the known DNS name to a public address, and send TCP or UDP traffic encapsulated in CLNP.

As we don't want to touch the existing Internet, like the need to allow the DNS servers to deal with JACS addresses or maybe the need for Internet routers to natively forward CLNP packets; we are going to depend on blockchain and some other mechanism as will follow.

Also, depending on dual-stack nodes as gateways for JACS enabled domains is not scalable at all. At the end of the day; the ability of a gateway, to handle the increasing traffic from and to its internal domain, is very limited, so the need for another scalable mechanism becomes a must.

10.4.2 Adopters as Enablers

How would a JACS-enabled site access the public internet that is solely and purely based on IP?

In this Beta version, the early adopters of JACS, will be allowed to avail their powerful nodes distributed around the globe, hence become enablers for JACS solution.

One critical requirement for any offered node, though, is to be a powerful 'dual-stack', meaning that it supports both JACS and IP natively.

Then comes the real value (what is in it?); that is to incentivize these enablers with JACS tokens.

On the technical side; we must tunnel the JACS traffic from a JACS-enabled site all the way through the Internet, to terminate on one of the offered nodes, from where it could get translated to access the public services. The procedure is somehow identical to how VPN services work nowadays.

Depending on the community for a viable solution, will allow the maximum adoption as opposed to the non-incentivized approaches like traditional transition plan (for example RPKI); where we saw how only one tenth (1/10) of the total Class-C subnets owned are being protected by the RPKI

Innovative ways exist on how to tunnel JACS traffic over the public Internet all the way to the enablers' nodes, and the good news is that blockchain is the control part of all of these ways.

Allowing blockchain to perform all the control plane functions of layer-2 tunneling techniques, while keeping the tunneling technique itself untouched, perfectly achieves the goal.

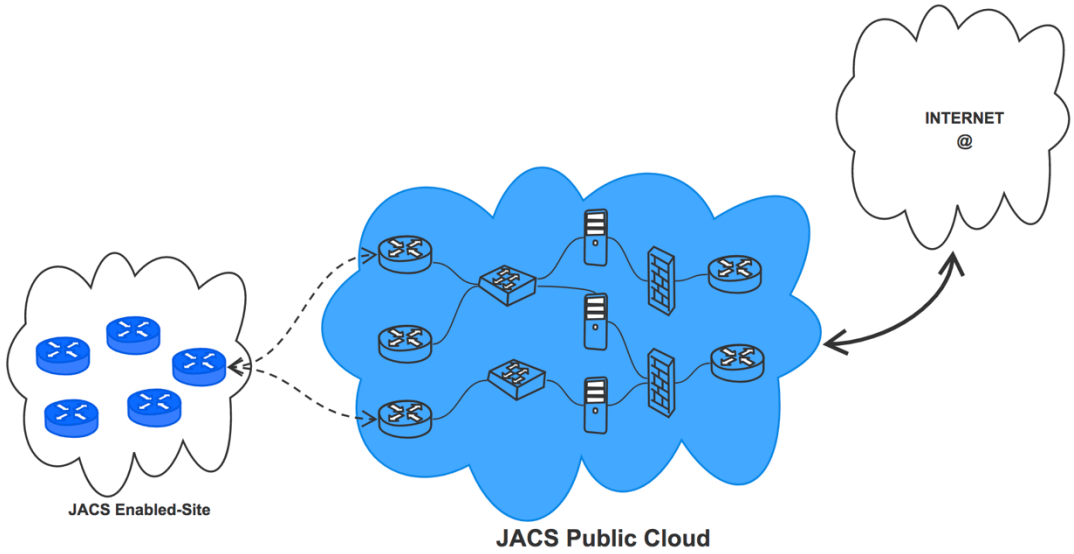
The latter solution would need another white paper to be explained in depth, but it was tested thoroughly and was practically proven.

Last but not least for the Beta version; the IDP part of JACS addresses, assigned during this version, is set as: **FD0000** (AFI is: FD and IDI is: 0000; that means geolocation disabled) and the dAPP will assign the DSP part of the address as the case with Pre-Alpha and Alpha versions

10.5 Official Release, Version-1 (Q4, 2020)

Starting from the first version of the official release; JACS-enabled sites will access all regular services from within the JACS global architecture without getting support from any enablers as the case with Beta version.

The launch of JACS native public cloud will allow this to happen seamlessly, where a robust underlying infrastructure and a bunch of compute and storage nodes that will intercept the tunneled traffic from all JACS-enabled sites and allow these sites to access all global public services.



It is worth as a reminder to state that any mechanism allowing JACS-enabled site to talk to the IP-based INTERNET requires the following:

- JACS-enabled site gateways must be dual-stack nodes in terms of their support of IP (to reach the public JACS cloud) as well as JACS for the internal communications within their site.
- JACS public cloud is mainly based on dual-stack nodes and the underlying infrastructure is capable of handling both IP and JACS traffic.

10.6 Official Release, Version-2 (Q1, 2021)

Starting from the second version of the official release; the Ethereum blockchain will be replaced by JACS native blockchain. JACS native blockchain characteristics and specifications will be shared on a timely manner.

With all the features and benefits indicated in its relevant section (JACS over native blockchain); a couple of interesting features need to be highlighted:

10.6.1 Lost Keys & Account Recovery

Over JACS native blockchain, it will be possible for a requesting node to recover its account even if loses its keys.

For the earlier versions of JACS over Ethereum, it is impossible for a requesting node to recover its account or lost keys, thus if that happens and there is no backup requesting node specified in advance to hold the same account info and keys; the requesting node will not be able to verify its

ownership of its acquired JACS blocks and also will not be able to request additional blocks in a contiguous manner.

The only option to resume normal operation is for the requesting node to create a new Ethereum account and request new JACS address block and resets all address delegation and allocation channels with its delegates.

Though it may appear daunting; but getting back into regular operation is not a hard process as it could be perceived.

10.6.2 Block Production Rewards

Over Ethereum, JACS with its earlier versions rewards its users in different ways, one of the most remarkable ways is the Adopters as Enablers rewarding mechanism.

Over its native blockchain, there is no need for any user's services (it was actually the case since the first version of the official release). So, there must be a way to continue the reward and incentive model.

This is done by rewarding the block producers over JACS blockchain. Here the word block refers to the blocks that are glued together to form the chain and not the address block.

All characteristics and eligibility for block production over JACS blockchain will be revealed on time, as well as the rewarding structure itself and its duration.

The rewarding period, during which there will be rewards for block producers, will highly depend on the rate of adoption of JACS. Even after this rewarding period; there will be other ways to get rewarded by participating as a community member of the JACS native chain.

10.7 Official Release, Version-3 (Q3, 2021)

Prior to the third version of the official release; regular (legacy) equipment could be used as long as the required software components are properly loaded on them.

The ideal solution that works inside JACS environment, so far, is Network Function Virtualization (NFV), where the client can use virtual machines (VMs) running on top of standard servers (instead of legacy hardware appliance) for each network function and apply required JACS software pieces on top of the VMs.

In this version, the Open Networking technology will be utilized (instead of NFV that will still exist) at its best and will be totally bundled with JACS solution.

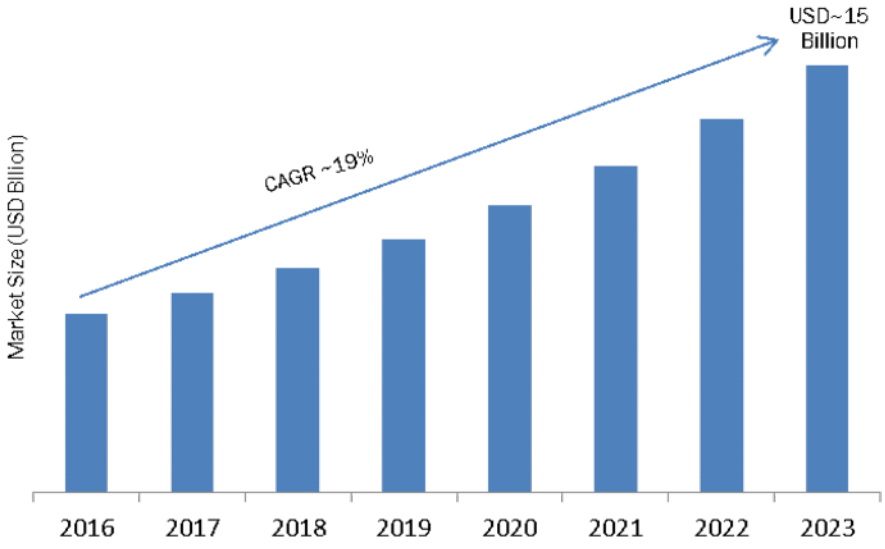
In the basic definition Open Networking; it allows network disaggregation to happen for the sake of the end customer who will be able to freely mix between different hardware and software

components. Disaggregation simply means separating the hardware from the software of the network gear. Hence the white box was born.

The white box is 100% plug-and-play. All configuration needed (that is minimal) will be fully automated. JACS bundled White Boxes will be commercially available in different sizes to satisfy all needs.

JACS will get in partnership with the big players in the Hardware and Software areas for its own white boxes; the white boxes will be privately labeled for JACS.

According to this analysis; the white box market size is forecasted to reach 15Billion US$ in 2023.



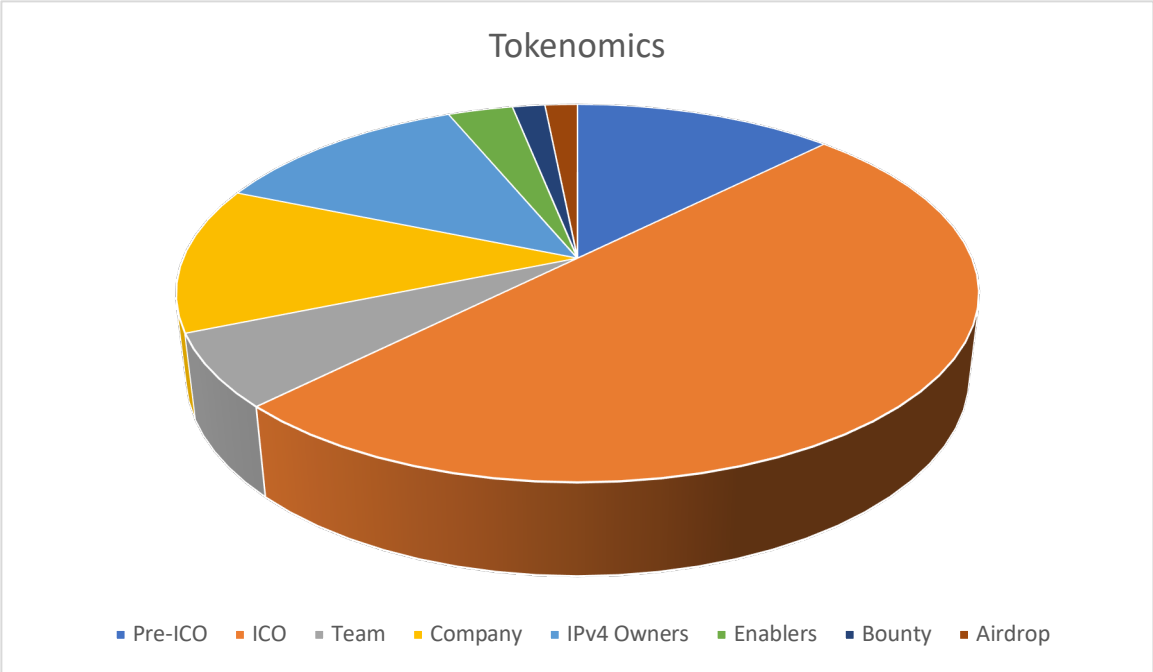
11 Tokenomics

JACS token will be used to request and own JACS address blocks. New users (individuals, organizations, providers, operators…) need to acquire JACS tokens to be able to acquire JACS address blocks.

Built over the Ethereum blockchain; JACS has a total of **2^{29} = 536,870,912** JACS tokens that will be created at once.

Limited finite supply ensures liquidity and value. The main categories and sub-categories are:

- Crowdfund (62.5%):
  - Pre-ICO (12.5%)
  - ICO (50%)
- Reserve (18.75%):
  - Team (6.25%)
  - Company (12.5%)
- Rewards (18.75%):
  - IPv4 Owners (12.5%)
  - Adopters as Enablers (3.125%)
  - Bounty (1.563%)
  - Airdrop (1.563%)



11.1 Proof of ownership

On the other hand, verified IPv4 blocks owners will be granted JACS tokens relative to their holdings to be able to claim their tokens for JACS blocks.

IPv4 blocks verification will be done by integrating JACS platform with the RPKI system (Resource Public Key Infrastructure) as well as the IPChain private blockchain.

There will be Four '4' JACS tokens for each IPv4 class-C subnet (/24), thus the total number of tokens that will be distributed to IPv4 subnets owners will be: **2^{26} = 67,108,864** JACS tokens, representing **12.5%** of the maximum supply of tokens.

11.2 Adopters as Enablers

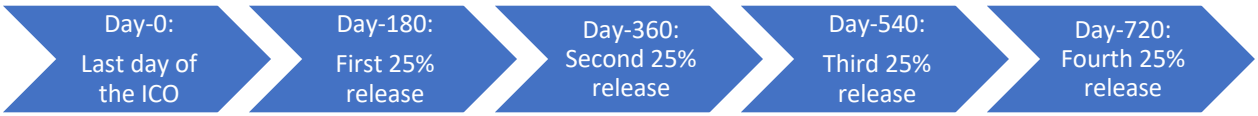
During the first year of JACS operation, **2^{24} = 16,777,216** JACS tokens will be rewarded to early adopters who are willing to acquire JACS address blocks in exchange for tokens and then be able to enable others to use JACS as detailed in the Beta version. This represents **3.125%** of the maximum supply.

11.3 Team

6.25% of the maximum number of JACS tokens will go to the founding team and the board of advisors.

These are 2^{25} = **33,554,432** JACS tokens that will be locked in the smart contract for 720 days (24 months).

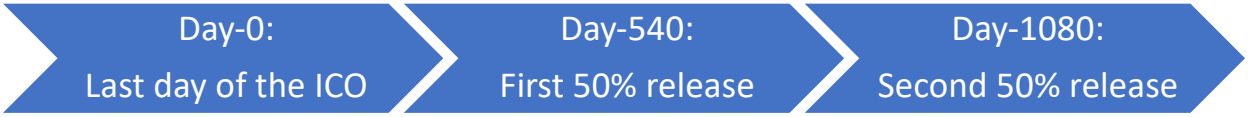
There will be a release of 25% of the locked tokens every 6 months following the conclusion of all crowdfund activities.



11.4 Company

12.5% of the maximum number of JACS tokens will be reserved for future development, Mergers and Acquisitions, IP rights, Patents registration and others.

These are 2^{26} = **67,108,864** JACS tokens that will be held in the smart contract for 1080 days (36 months) and released by 50% every 18 months following the conclusion of all crowdfund activities.



11.5 Bounty

2^{23} = **8,388,608** JACS tokens will go to bounty hunters. That represents **1.563%** of the maximum supply.

The related 'bitcointalk' bounty thread will include all details for bounty hunters on how to participate, the different areas required and the equivalent bounties.

11.6 Airdrop

2^{23} = **8,388,608** JACS tokens will be airdropped over the Ethereum blockchain. That represents **1.563%** of the maximum supply.

Airdrop dates and details will be shared over the different social medias as well as over the proper 'bitcointalk' ANN thread.

12 Crowdfund

There will be two distinct crowdfund phases; pre-ICO and public ICO where **62.5%** of the maximum supply of tokens will be available for purchase.

Contrarily to the common approach of availing 50% of the maximum supply of tokens during the crowdfund phases, JACS team decided to avail more tokens for the public community.

One of the main critical aspects of Tokens and crypto-based crowdfund is to allow the product to be perfectly decentralized and distributed across the global community.

Giving the chance to more people to invest and acquire even a single token, will fulfill this objective.

Also, because JACS is fully based on blockchain, the community will be the decision maker. There will be no central authority governing the process.

Even JACS team will have a limited capacity in terms of deciding the way forward and the fate of the project. Their function will be only to allow the platform to be fully developed, and everything will eventually and naturally be handed-over to the community.

12.1 Pre-ICO

The pre-ICO phase that will take place over 2-month period.

2^{26} = 67,108,864 JACS tokens, representing **12.5%** of the maximum supply, will be available for purchase with a fixed price of $12c per token.

Discounts are available for huge volume purchases.

12.2 ICO

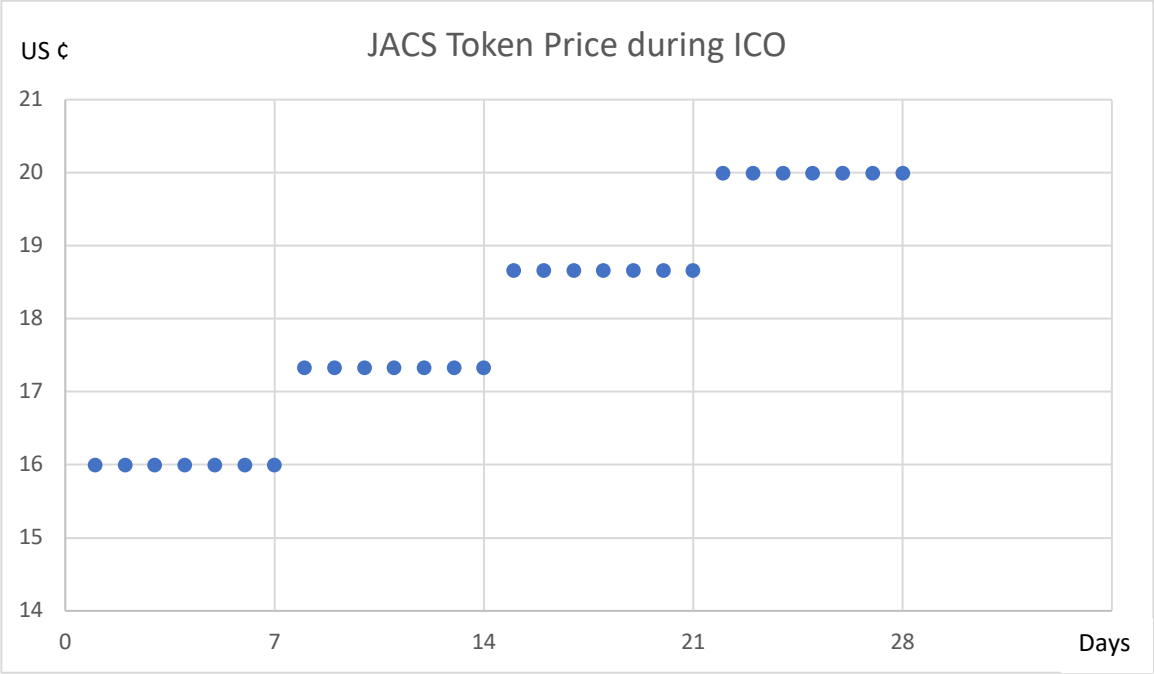
The ICO phase that will take place over 28-days (4 weeks) period.

2^{28} = 268,435,456 JACS tokens, representing **50%** of the maximum supply, will be available for purchase automatically during the ICO.

A smart contract will take care of the ICO activities over the public Ethereum blockchain.

The price per token during the ICO will follow a weekly-increasing step function where it will start with $16c for the first 7 days and will end with $20c for the last 7 days.

This will encourage investors to invest early enough during the ICO to benefit from the competitive starting price.



No discounts are available during the ICO, these are included in the ‘step’ function governing the price of token during the ICO.

13 General Disclaimer

PLEASE READ THIS DISCLAIMER SECTION CAREFULLY.
IF YOU ARE IN ANY DOUBT AS TO THE ACTION YOU SHOULD TAKE, YOU SHOULD CONSULT YOUR LEGAL, FINANCIAL, TAX, OR OTHER PROFESSIONAL ADVISOR(S).

The information set forth below may not be exhaustive and does not imply any elements of a contractual relationship. While we make every effort to ensure that any material in this whitepaper is accurate and up to date, such as products, services, technical architecture, token distribution, company timelines - such material could be subject to change without notice and in no way constitutes a binding agreement or the provision of professional advice.
viaBlock LTD. does not guarantee, and accepts no legal liability whatsoever arising from or connected to, the accuracy, reliability, currency, or completeness of any material contained in this whitepaper. Potential JACS Token holders should seek appropriate independent professional advice prior to relying on, or entering into any commitment or transaction based on, material published in this whitepaper, which material is purely published for reference purposes alone. JACS Tokens will not be intended to constitute securities in any jurisdiction.
This whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities or a solicitation for investment in securities in any jurisdiction. viaBlock LTD. does not provide any opinion on any advice to purchase, sell, or otherwise transact with JACS Tokens and the fact of presentation of this whitepaper shall not form the basis of, or be relied upon in connection with, any contract or investment decision. No person is bound to enter into any contract or binding legal commitment in relation to the sale and purchase of JACS Tokens, and no cryptocurrency or other form of payment is to be accepted on the basis of this whitepaper.

14 JACS Token Legal and Crowdsale

14.1 General Information

The JACS Token does not have the legal qualification of a security, since it does not give any rights to dividends or interests. The sale of JACS Tokens is final and non-refundable. JACS Tokens are not shares and do not give any right to participate to the general meeting of viaBlock LTD. JACS Token cannot have a performance or a particular value outside the viaBlock LTD. JACS network. JACS Token shall therefore not be used or purchased for speculative or investment purposes. The purchaser of JACS Token is aware that national securities laws, which ensure that investors are sold investments that include all the proper disclosures and are subject to regulatory scrutiny for the investors' protection, are not applicable.
Anyone purchasing JACS Token expressly acknowledges and represents that she/he has carefully reviewed this whitepaper and fully understands the risks, costs and benefits associated with the purchase of JACS Token.

14.2 General Knowledge

The purchaser of JACS Tokens undertakes that she/he understands and has significant experience of cryptocurrencies, blockchain systems and services, and that she/he fully understands the risks associated with the crowdsale as well as the mechanism related to the use of cryptocurrencies.
viaBlock LTD. shall not be responsible for any loss of JACS Token or situations making it impossible to access JACS Tokens, which may result from any actions or omissions of the user or any person undertaking to acquire JACS Tokens, as well as in case of hacker attacks.

14.3 Risks

Acquiring JACS Tokens and storing them involves various risks, in particular the risk that viaBlock LTD. may not be able to launch its operations and develop its blockchain and provide the services promised.
Therefore, and prior to acquiring JACS Tokens, any user should carefully consider the risks, costs and benefits of acquiring JACS Token in the context of the crowdsale and, if necessary, obtain any independent advice in this regard.
Any interested person who is not in the position to accept or to understand the risks associated with the activity (incl. the risks related to the non-development of the viaBlock LTD. JACS platform) or any other risks as indicated in the Terms & Conditions of the crowdsale should not acquire JACS Tokens.

14.4 Disclaimer

This whitepaper shall not and cannot be considered as an invitation to enter into an investment. It does not constitute or relate in any way nor should be considered as an offering of securities in any jurisdiction. The whitepaper does not include nor contain any information or indication that might be considered as a recommendation or that might be used to base any investment decision. This document does not constitute an offer or an invitation to sell shares, securities or rights

belonging to viaBlock LTD. or any related or associated company. The JACS Token is just a utility token which can be used only on the viaBlock LTD. JACS platform and is not intended to be used as an investment.
The offering of JACS Token on a trading platform is done in order to allow the use of the viaBlock LTD. JACS platform and not for speculative purposes. The offering of JACS Token on a trading platform is not changing the legal qualification of the token, which remains a simple means for the use of the viaBlock LTD. JACS platform and is not a security.
viaBlock LTD. is not to be considered as advisor in any legal, tax or financial matters. Any information in the whitepaper is given for general information purpose only and viaBlock LTD. does not provide with any warranty as to the accuracy and completeness of this information. Given the lack of crypto-token qualifications in most countries, each buyer is strongly advised to carry out a legal and tax analysis concerning the purchase and ownership of viaBlock LTD.'s Tokens according to their nationality and place of residence.
viaBlock LTD. today is not a financial intermediary according to United Kingdom's Law and is not required to obtain any authorization for Anti-Money Laundering purpose. This qualification may change in case viaBlock LTD. will offer services which are to be considered as qualifying a financial intermediation activity.
JACS Tokens confer no direct or indirect right to viaBlock LTD.'s capital or income, nor does it confer any governance right within viaBlock LTD.; a JACS Token is not proof of ownership or a right of control over viaBlock LTD. and does not grant the controlling individual any asset or share in viaBlock LTD., or in the viaBlock LTD. JACS network. A JACS Token does not grant any right to participate in control over viaBlock LTD.'s management or decision-making set-up, or over the viaBlock LTD. JACS network and governance to the purchasers.
Regulatory authorities are carefully scrutinizing businesses and operations associated with cryptocurrencies in the world. In that respect, regulatory measures, investigations or actions may impact viaBlock LTD.'s business and even limit or prevent it from developing its operations in the future. Any person undertaking to acquire JACS Token must be aware of the viaBlock LTD. business model, the whitepaper or Terms & Conditions may change or need to be modified because of new regulatory and compliance requirements from any applicable laws in any jurisdictions.
In such a case, purchasers and anyone undertaking to acquire JACS Token acknowledge and understand that neither viaBlock LTD. nor any of its affiliates shall be held liable for any direct or indirect loss or damage caused by such changes.
viaBlock LTD. will do its utmost to launch its operations and develop the viaBlock LTD. JACS platform. Anyone undertaking to acquire JACS Token acknowledges and understands that viaBlock LTD. does not provide any guarantee that it will manage to achieve it. On concluding the Commercial Operation, these tokens will be issued by a technical process referred to as a «Blockchain». This is an open source IT protocol over which the Company has no rights or liability in terms of its development and operation.
The token distribution mechanism will be controlled by a Smart Contract; this involves a computer program that can be executed on the Ethereum network or on a blockchain network that is compatible with Smart Contract programming language.
They acknowledge and understand therefore that viaBlock LTD. (incl. its bodies and employees) assumes no liability or responsibility for any loss or damage that would result from or relate to the incapacity to use JACS Tokens, except in case of intentional misconduct or gross negligence.

JACS Tokens is based on the Ethereum protocol. Therefore, any malfunction, unplanned function or unexpected operation of the Ethereum protocol may cause the viaBlock LTD. JACS network to malfunction or operate in a way that is not expected. Ether, the native Ethereum Protocol account unit may itself lose value in a similar way to JACS Tokens, and also in other ways.

14.5 Representation and warranties

By participating in the crowdsale, the purchaser agrees to the above and in particular, they represent and warrant that they:

- have read carefully the Terms & Conditions attached to the whitepaper; agree to their full contents and accept to be legally bound by them;
- are authorized and have full power to purchase JACS Token according to the laws that apply in their jurisdiction of domicile;
- are not a U.S. citizen, resident or entity (a "U.S. Person") nor are they purchasing viaBlock LTD. JACS Tokens or signing on behalf of a U.S. Person;
- are not resident in China or South Korea and nor are they purchasing JACS Token or signing on behalf of a Chinese or South Korean resident;
- live in a jurisdiction which allows viaBlock LTD. to sell JACS Tokens through a crowdsale without requiring any local authorization and are in compliance with the local, state, and national laws and regulations when purchasing, selling and/or using viaBlock LTD. JACS Tokens;
- are familiar with all related regulations in the specific jurisdiction in which they are based and that purchasing cryptographic tokens in that jurisdiction is not prohibited, restricted or subject to additional conditions of any kind;
- will not use the crowdsale for any illegal activity, including but not limited to money laundering and the financing of terrorism;
- have sufficient knowledge about the nature of the cryptographic tokens and have significant experience with, and functional understanding of, the usage and intricacies of dealing with cryptographic tokens and currencies and blockchain-based systems and services;
- purchase JACS Token because they wish to have access to the viaBlock LTD. JACS platform;
- are not purchasing JACS Token for the purpose of speculative investment or usage.

14.6 Governing law – Arbitration

The Client acknowledges and accepts that the viaBlock LTD. ICO operation is taking place within the United Kingdom's legal environment. The Parties agree to seek an amicable settlement prior to bringing any legal action. All disputes arising with the with papers provided, shall be resolved by arbitration in accordance with the United Kingdom's Rules.

JACS Tokens will not be listed on any regulated stock exchange.

14.7 Parties with whom we may share your information

viaBlock LTD. may share User Content and your information (including but not limited to, information from log files, device identifiers, location data, and usage data) with businesses that are legally part of the same group of companies that viaBlock LTD. is part of, or that become part of that group. Affiliates may use this information to help provide, understand, and improve the Service (including by providing analytics) and Affiliates' own services (including by providing you with better and more relevant experiences).

14.8 What happens in the event of a change of control

If we sell or otherwise transfer part or the whole of viaBlock LTD. or our assets to another organization (e.g., in the course of a transaction like a merger, acquisition, bankruptcy, dissolution, liquidation), your information collected through the Service may be among the items sold or transferred.

14.9 Forward - looking statements

This whitepaper contains forward–looking statements. The words or phrases "would be," "will allow," "intends to," "will," "shall," "will likely result," "are expected to," "will continue," "is anticipated," "estimate," "project," or similar expressions are intended to identify "forward–looking statements." All information set forth in this whitepaper, except historical and factual information, represents forward–looking statements.

This includes all statements about the project plans, beliefs, estimates and expectations. These statements are based on current estimates and projections, which involve certain risks and uncertainties that could cause actual results to differ materially from those in the forward-looking statements. These risks and uncertainties include issues related to: rapidly changing technology and evolving standards in the industries in which the company operates; the ability to obtain sufficient funding to continue operations, maintain adequate cash flow, profitably exploit new business, license and sign new agreements; the unpredictable nature of consumer preferences; and other factors.

Readers are cautioned not to place undue reliance on these forward–looking statements, which reflect management's analysis only as of the date hereof. The Company undertakes no obligation to publicly revise these forward–looking statements to reflect events or circumstances that arise after the date hereof. Readers should carefully review the risks and uncertainties described in other documents that the company publishes.