The ISP Column
A monthly column on all things Internet

Who are you?

June 2003

Geoff Huston

A Survey of Digital Identities

Well maybe you know who you are, but the rest of us are still trying to work it out!

The Who released the studio album "Who are you" in 1978. At this stage Pete Townshend's disillusion about stardom had set in and the band's musical performance is balanced by a set of introspective and cynical lyrics. The recording accompanied a film of the history of The Who, titled "The Kids are Alight". The Who completed the album and the film in 1978 with a concert held at Shepperton for Who fans on May 25th, 1978. This was to be the last recording of The Who with maniacal drummer Keith Moon. In September of that year Keith Moon died of an accidental overdose of pills he had been prescribed to control his alcoholism. The writing on the chair Keith is sitting on in the album cover reads "Not to be taken".

Whenever you travel across country borders, open a bank account, even enter your workplace, or perform numerous other tasks in your day, your identity is constantly challenged and you are requested to provide some form of credentials that support your identity claim. So your identity is not a trivial question to ask, and sometimes it can be tough to answer in terms of amassing the supportive evidence of your identity.

But if identity is a tough question to answer in human terms, its interesting to take the same fundamental question of identity into the world of the Internet and ask it in that context. How can I tell that its really you at the other end of my network session? What's your digital identity and how can I confirm it?

What attributes do Identity Systems share?

Before heading down this path of exploring various Internet-based identities, its useful to ask what sort of attributes we want from any useful identity system. Here's my list of useful identity attributes:

- Uniqueness
  It would be good to understand that your assertion of identity does not use the same identifier token as someone else.

- Consistency
  It would be good if identity was asserted within a consistent identifier space. That way we could avoid having your assertion of identity being interpreted by me as a meaningless sequence of 1's and 0's.

- Persistence
  It would also be good if the identity I used yesterday, or even last week was the same that I use today. Constantly changing identities are, at the very least, difficult to track.

- Trust
  It would be good if an assertion of a particular identity could withstand a challenge as to its validity. Others who would like to use this identity would like to be reassured that they are not being deceived.

- Robustness
  And lastly, it would be good if the identity realm was capable of withstanding deliberate or unintentional attempts to corrupt it in various ways.

So if these attributes are valuable in any digital identity system, the next step is to look at a few identity realms in the Internet and see how well they stack up.

A Hierarchy of Digital Identities

In networking there is a conceptual layering of functionality, starting at the layer of bits on the wire at the media access level and moving up a stack of layers through internetworking, end-to-end transport and application levels. Identity is expressed at every level. It would appear that from this perspective your digital presence within the Internet is not just a single identity, but an entire collection of various identities, used in a whole variety of contexts.

Your MAC Identity

Ethernet, the most common of the media level technologies uses a 48 bit Media Access Control layer address, and, in general, and certainly by intention, these addresses are unique. So one form of identity is this 48 bit MAC address used by my computer.

A MAC address identity certainly passes one of the more basic tests of identity, that of uniqueness. It is not the intention that two parties can assume the same identity, and use the same value as a unique identity. So in a LAN context a collection of devices can distinguish between each other by virtue of this 48 bit unique MAC address. A manufacturer of Ethernet devices is assigned a manufacturer's block of Ethernet MAC addresses and uniquely places one address in each device. The end consumer has no need to reconfigure the device with a new address, nor is there any need to alter MAC addresses each time the LAN changes with new devices being added. It also has a high utilization capability, in that a manufacturer can assign individual values to devices sequentially so that the identity space can be tightly packed. And, of course, the identity space is large, encompassing more than 260 trillion (10**12) individual addressed devices.

But beyond these attributes there are some real weaknesses in using a MAC address an identity outside the context of a LAN environment. The identity space is structured so that it can be readily asserted to be globally unique, but has few other distinguishing properties. The structure of the identity space reflects the manufacturer of the device, not its location within a particular network topology, so its of no assistance as a location token. In the context of equating a device identity to this network interface identity, the identity has limited persistence, in that it follows the interface hardware, not the host computer or its use. Changing WiFi cards in your laptop changes your MAC identity, as does switching from a wireless to a wired connection. The identity has no capability to express any linkage to any other identity domain. It has no internal structure of sub-fields that could be interpreted as pointers into other identity fields.

Even so, its a useful identity mechanism in the context of an identity space. Its been filled with 16 padding bits in order to be incorporated into the IEEE 64-bit EUI-64 global identifier structure, which in turn has been incorporated into the IPv6 address architecture as an interface identifier

Your IP Identity

So lets move up a level in the stack and look at an identity based on the IP address. The IPv4 address field is a 32 bit field where each connected IP interface has a unique value. IPv6 uses effectively the same construct, using a 128 bit identity domain in the place of the 32 bit field. In both cases the IP address is a structured identity space where there is a globally significant prefix that is used in the context of routing and forwarding outside to a particular local domain, and a local part that is used to deliver the packet to the correct interface of the associated device within the local network.

So, as an identity, an IP address should be unique. It is structured in such a way to be useful to forward packets to the addressed device, and it's well known, in that its not a secret value.

> Well not always unique. There is a form of aliasing a collection of addressed devices to share a single IP address called "anycast". Its been used in a number of contexts for collections of servers that provide identical services, most notably in recent times with a number of the root DNS servers. In this case its the routing system that effectively performs the function of balancing the traffic across multiple servers. In this case, anycast, your packets will not go to a particular identified location, as is normally the case with unicast, nor to all locations that share an address, as is the case with multicast, but will go to the 'closest location, where 'closest' is determined by the routing system. This works well when a uniquely address device wants to talk to any anycast address. but some care should be taken to ensure that an anycast source does not initiate a session using the anycast address as a source address - after all there's no telling which anycast server will receive the response

But an IP address not everything one would hope for in a identity. The IP address identifies an interface, not a device or its user. A device with multiple active interfaces has multiple IP addresses, and while its obvious to the device itself that it has multiple identities, no one else can tell that the multiple identities are in fact pseudonyms, and that the multiple addresses simply reflect the potential for multiple paths to reach the same endpoint. The IP address is structured in such a way that it can be used in routing and forwarding, which is helpful in the sense that there is no need to deploy a second identity system that refers only to locality within a network. An IP address has internal structure that is used to identify how to send IP packets to the addressed device, or, in other words, to reach you.

But an IP address suffers from semantic overload in attempting to carry both location and identity. If you change providers you are effectively changing your network location, and your IP address may change. And if the addressed device is mobile, then the address will change as its location within the Internet changes. The mobility folk have expressed the problem in terms of "home agents" and "care-of agents", and the essential issue in mobility is one of preserving your 'core' identity, or "home" IP address, while roaming to other parts of the network and being given a temporary address to use as a 'care-of' forwarding address. But this implies that an IP address is not necessarily a permanent association with a device, and impermanence is a weakness in any identity system.

Another issue with IP addresses, at least in version 4 of the protocol is that of its total span. While 32 bits is still a hefty size, encompassing some 4.4 billion unique addresses, there is an inevitable level of wastage in deployment, and a completely exhausted 32 bit address space may only encompass 1 or 2 billion IP devices. When this is coupled with a world of embedded IP devices in all kinds of industrial and consumer applications, 1 or 2 billion does not seem like such a large number after all. So, over the last decade we've seen the deployment of a number of technologies that deliberately set out to break any string binding of IP address with identity, and treat the IP address purely as a routing and forwarding token without any of the other attributes of identity, including persistence and public association. DHCP, or address-lending, is a commonly used method of extending a fixed pool of IP addresses over a domain where not every device is connected to the network at any time, or when devices enter and leave a local network over time and need addresses only for the time they are within the local network's domain. This has been used in LANs, dial-up, ADSL, WiFi service networks and a wide variety of applications. In this form of identity, the association of the device to a particular

IP address is temporary, and hence there is some weakening of the identity concept, and the dynamically-assigned IP address is being used primarily for routing and forwarding. This has been taken a further step with the use of Network Address Translation approaches, where a single device has a pool of public addresses to use, and maps a privately used address device to one of its public addresses when the private device initiates a session with a remote public device. The private-side device has no idea of the address that the NAT edge will use for a session, nor does the corresponding public-side device know that it is using a temporary identity association to address the private device.

> To achieve even higher 'packing' densities Port Translating NATS have been deployed, where each unique session is mapped to a unique port and IP address, and sessions from multiple private sources may share a common IP addresses, but differentiate themselves by having the NAT-PT unit assign port addresses such that the extended IP + port address is unique. How do you know if you are talking directly to a remote device, or talking through a NET filter, or multiple NAT filters, or NAT-PT filters? And if you are talking through a NAT, how do you know if you are on the 'outside' or the 'inside'?

These forms of changes to the original semantics of an IP address are uncomfortable changes to the concept of identity in IP, particularly in the area of NAT. The widespread adoption continues to underline the concept that as an identity token there is a lack of persistence, and the various forms of aliasing weaken its utility as an identity system. Increasingly an IP address, in the world of IPv4, is being seen as a locality token with a very weak association with some form of identity.

Of course that doesn't stop undue assumptions being made about the uniform equivalence of identity and IP address, however specious it may be in particular situations, and various forms of IP filter lists, whether they be various forms of abuse black lists or security permission lists all are evidence of this contradictory behavior of assuming that persistent identity and IP address are uniformly equivalent.

Version 6 of IP is an attempt to restructure the address field, and the 128 bits of address space represent a very large space in which to attempt to place structure. One of the more innovative concepts that was discussed within the development of IPv6 was extending the concept of the IPv6 interface identifier to be a globally unique identifier. This had some obvious connotations in being able to identify when the connectivity for a device has changed, as in such cases the globally unique identifier will remain constant while the routing prefix may have changed. There was also some interesting applications in the area of supporting multi-homed networks, where a network could be seen via different routing prefixes.

> This was a concept that, with some regret as I see it, was never truly developed in the context of IPv6 address architecture development, and today the interface identifier is defined by the IPv6 address architecture document to be one that has "global scope" but not global uniqueness. To me these terms are weakly synonymous, in that an identity that has global scope is one that would aspire to global uniqueness!

Your Service and Session Identity

It's a computer Science observation that its always possible to perform another level of abstraction. And of course its always possible to find yet another level of identity. In the TCP/IP protocol suite the next level of identity is that of the transport session. If you are reading this article from a web page then your system has set up a transport level connection with a server. Your system has combined the three fields: IP server address, TCP protocol identity, and the TCP web server local identity (port 80) into a compound identity that describes a particular service port on a particular device.

The port address concept, used in both TCP and UDP, represent generic identities for service rendezvous points. When combined with an IP address they become particular service points, or, identified service points, and these compound identification objects (IP address, Transport Protocol, Port) are service identifiers.

The identity concept for transport is further extended by including the sender's IP address and port address. The corresponding 5-tuple of (source IP address, destination IP address, Transport Protocol, Source Port, Destination Port) is an identifier for a particular instance of a session. Not only is this 5-tuple used at the destination point to correctly de-mux an incoming packet and send it to the correct local instance of the application, the session identity can also be used within the network to recognize a 'flow' of packets that require identical forwarding treatment and may require identical service treatment, if so configured. In the latter case the session identity is being used to trigger a particular service response.

Session identities are intended to be unique at any point in time, in that two distinct sessions will not share a common session identity. But their association over time is not unique, in that at a subsequent time a different session may use the same 5-tuple. As well as impermanence, session level identifiers exhibit a very fine level of granularity, and as such are often at a level of detail which is too fine to be a useful general identity token. One use is to allow a session to construct an identity that refers to itself that it can then hand into a quality of service policy controller to request a specialized service response for the session.

Your Domain Name

The previous set of identities had no particular human-visible aspects of their function. The identity tokens were structured to meet a particular purpose, and were not intended to be manipulated by humans. The Domain Name System, or DNS, was specifically intended to be a name realm that was suitable to be included in human discourse, yet at the same time admit enough structure to be manipulated by computer applications in a deterministic fashion. In its original incarnation the DNS was a simple replacement for the earlier 'hosts.txt' file. And what did 'hosts.txt' do? It mapped host names to IP addresses. Rather than saying "telnet 10.0.0.1" to an application, it would let you say "connect me to the mail server".

The DNS is essentially a hierarchical name space, where the hierarchical name structure allows the space to be efficiently searched and managed in a distributed fashion, but also supports one of the most desirable attributes for an identity space. The explicit hierarchy also assists in ensuring uniqueness, as DNS names are intended to be unique across the entire name string rather than just at the first component, so that "a.b.c" is a different identifier to "a.d.e"

There's been a lot of talk about domain names over the past few years, and many could be forgiven for gaining the impression that the Internet is nothing more than domain names.

The most common use of the DNS is to map domain names to IP addresses, but other uses are possible. The core of the DNS is a unique name space and a mapping capability that allows a query to be performed to retrieve the mapping information for a DNS name for a particular class of resource mapping.

> As a human-use oriented identity space its perhaps no surprise that the DNS has been one of the closest areas of attention for various forms of service provision related to identity functions. The DNS itself is now the subject of a massive service industry whose stock in trade is, simply, unique names. And its a strange industry. The service providers do not even have to think of any particular name - you do. Their value add to the proposition is to place it into the DNS, and there, by the virtue of picking a particular name hierarchy, you have attained uniqueness and some form of stability for your identifier. If you change IP addresses you can change the mapping without changing the name itself.

Domain Names are big business and big policy these days. How much personal information should be associated with domain names in the registration database? How to registrars distinguish themselves in this most basic of commodity markets? Why is there only one rooted point to this name hierarchy? Why do countries have points in the domain hierarchy yet there are also these various 'generic' name points at the top of the hierarchy, such as .com, .net and .org? How big can the DNS get?

The answer to this last question is a rather strange "it depends". The hierarchy of the DNS name structure is matched by a matching hierarchy of potential delegation and distribution of function in the DNS. At each point in the hierarchy it is possible to operate a separate name resolver that translates name queries into responses as driven from the local DNS mapping, or zone file. Couple this with the ability for name query agents, or 'forwarders' to cache the resolvers' responses under terms that are specified by the original authoritative server and you have e constructed with a massive identity space that can efficiently return responses to specific queries (well most of the time!

What about the association between this hierarchical string and the collection of resources that have been associated with the domain name? The question arises as to how could you update just that bit of the DNS that describes your device, or location-based identity, and do it every time you roam into a new connection realm, or each time to renew an address lease with DHCP. If you could undertake this easily and securely each time to change IP addresses, and have various caches quickly time out the old association and replace it with the current values, then the DNS could be used to fill in for the short-comings in the characteristics of the IP address world. And this is one of the intended outcomes of the combination of DNS security and DNS Incremental Update capabilities. This allows for a form of nomadism in the Internet world, where you can roam between addresses, and each time you secure a new address you can update your DNS records to reflect the new address association. In this model its the IP address that is now the variant part, and the domain name that provides and enduring and unique identity.

Its little wonder that the DNS is the subject of so much commercial and policy pressure, if it really has assumed the role of the major identity system within the Internet. Not only does it provide assistance to the underlying address framework, but it extends upwards into the application realm. And within the application realm the most common context of using these names appears to be in the context of various service identifiers, or URIs.

Your URI Identity

The URI space, as an identity space is very loosely defined, and its quite remarkable as to the extent to which it has spread across the world as a form of object identifier, or identity token.

Surprisingly, there are not many syntax rules to the Universal Resource Identifier space, nor a wealth of common semantic structure. The original IETF documentation, RFC 1640, refers quite simply to a syntax of a prefix word, a colon, and a following string. That's almost it, apart from the additional constraint that where there is hierarchy in the following string, slashes are used to delineate the hierarchical levels, and the hierarchy runs from left to right.

But the common usage of URIs has been more structured than that, and most URI schemes do not provide a single string that is an alias for an identity, but instead form an identity from the instructions that specify how to access the resource, in the same way as a postal address is often constructed from the instructions as to how to deliver a postal letter to you.

The prefix is an identifier that uniquely identifies service, or in terms of access, the protocol and port address to be used. The first, or top level of the hierarchical following string is either the DNS name of the server, or the DNS name coupled with some specific qualification, such as a mail address. Any subsequent hierarchical components represent service-specific instructions to be specified that lead you to the referenced object. Thus we have mailto:user@domain.example.com for a mail specification, or http://www.example.com/directory/hierarchy/index.html for a specific web page.

In this identity system uniqueness is keyed from the general use of a DNS name within the URI, and the wrapping around the DNS string is taking the DNS' general form of an alias for an IP address, and specifying a service point, and then arguments you need to provide to this service point to retrieve a digital resource. In that way a URI is closer to an algorithm description than a set of identifiers where the structure of the identifier is adapted to tasks such as sorting, searching or equivalence operations. There are issues with consistency here in that while the hierarchically structured

string set makes sense to one application it may not make any sense in the context of a different application. The persistence of URIs is an issue, in that the resource may change location over time, and the corresponding algorithm to locate the resource, or URI, must necessarily change as well. The other major difference between a structured identifier space and the URI approach is that the structured identifier space requires some form of lookup to apply the identity into a retrieval system. By changing the outcomes form the lookup operation, the identity owner can track changes in the location of the resource. In the URI scheme there is no way to understand how widely the identity has circulated, and it is not possible to update the in-circulation copies of the URI. The property of the DNS is that in itself the DNS identities are simple structured tokens, and they require a lookup operation to be performed in order to produce an algorithm that allows an application to refer to a particular object. While URLs are widely used service and resource identities, they pale in significance to DNS names. Perhaps its because a URL is often perceived as a domain name plus some formatting, and its the DNS that counts as the core of a named identity. It is also not surprising from this perspective to see the emergence of DNS objects that refer to URIs. In this approach the first DNS lookup retrieves one or more URIs that have been associated with the DNS name, and a second lookup is used to resolve any DNS names as may be referenced in the URI strings.

Your Phone Identity

One of the advantages (or major headaches, depending on your viewpoint) of the DNS was the move from using identity tokens that closely following network addressing primitives to names that had human significance. Now we could name our devices with identities that used names taken from the human world. This was going to leave the phone system, with its numeric notation of identity as an artifact of the past. Given such hopes of seeing the demise of the phone number as an identity realm, perhaps I should not be surprised to see the resurgence of the phone number playing the role of a personal contact identity!

The reincarnation of the phone number as a digital identity comes under the guise of the ENUM technology. The idea behind ENUM is that your phone number can be used by all kinds of applications that use a paradigm of individual communication. The ENUM model is that the application user provides the phone number of the individual to be contacted, and the DNS is used to map the phone number into a set of URIs that describe the various service points that may be used to reach the individual. This is an example of a DNS name that resolves into URIs which themselves use DNS names as part of the service specification.

The problems of this form of address space being used as an alias for a collection of service points is that it is an identity that has a strong legacy component in a different communications realm, and there are a number of constraints on the use of phone numbers in many regulatory regimes. While such a mapping from phone number to URI is highly convenient form of supporting peer discovery of VOIP services, it appears to fall short as a useful identity because of its longstanding association with a particular communications network, and mapping extensions of telephone service into the Internet domain does not automatically create a universally useful identity domain. SO while its a highly effective functional solution for a class of problems relating to mapping phone numbers into an Internet service realm, I'd offer the view that its not general enough to be a fully complete identity framework.

Your Secret Identity

Not all digital identities are intended to simply be assertions of a particular unique value within a larger domain of possibilities. One of the useful attributes of an identity is that of trust, and there is much that can be done if the two parties to a communication can definitely establish the identity of the other party.

Perhaps the simplest example is that of a password to a computer system. Having, at one point, established your right to access a system, how can you subsequently leave a trace of this relationship you've established so that subsequently you can validate your identity and reestablish access. One long-standing way is through a password, where validation is performed by a match of the provided password to that stored on the system. The beauty of this form of identity is that there is no particular need for uniqueness, as long as the space from which individual passwords are being drawn from is sufficiently large, and the passwords are sufficiently random, as the password is effectively a shared secret between the user and the system. Two parties using the same secret is no problem as long as the total space from which the secret is taken is so small that any password is readily guessable.

Its readily apparent that there are many risks in a simple password system, and also many limitations in its applicability. Not only are such systems susceptible to eavesdropping and password capture, there is the continual problem that passwords are a human system we're often quite forgetful. Its also the case that passwords are not (or at least should not) be transitive. Once you've established your credentials with one resource and obtained a password, how can you use this first trust relationship to substantiate further relationships? One solution is to move to various secret identities that have public counterparts, using public / private key pairs. The unique property here is that a secret created with the private key can only be unlocked with the public key, and vice versa. This can be viewed as a form of password - the secret password does not provide access to any particular resource, but instead allows you to prove an association with a corresponding public entity. The way in which this is used is to provide some form of verified public association between a particular public key and an identified individual or service. With this framework of private and public keys I can assert my identity to you by signing a message using my private key - only my public key can unlock the message. The identity is the combination of key pair. Your task is then to work out if you are prepared to believe the association of my public key with myself. Its here that public key infrastructure is necessary for such an identity framework to be truly useful as an digital identity realm.

Which "WHO" are you?

I couldn't let this question pass by without some form of passing reference to the venerable BBC television series "Doctor Who" that played from 1963 until 1996, using eight main actors and generated a cult following across a couple of generations.

Identity was never a problem in the early series, and simple assertion was as good as proof:

Taron: 'Are you trying to tell us that you are the Doctor?'
Doctor: 'That's right.'

Although the venerable Doctor himself was not quite so sure about himself:

Doctor: 'The more I know me, the less I like me.'

How good are any of these identities? Which one should we use? One observation certainly holds: if you happen to dislike one form of identity there's no shortage of alternatives that you can use!

Each of these digital identities have a context of usage, or realm of discourse if you want, and outside of that realm they tend to break down as a cohesive and useful identity tool. Offering my MAC address to you as an email point of contact, together with my username, makes little sense, even though it can be used to form a unique identity in the mail realm. Offering an identification at the appropriate level of abstraction that provides a description of the mode of contact and my identity in a form that matches the actions at this level is how we distinguish between identities. At the level of human interaction we swap email addresses using a domain and user name part. We do this because this is what you need to enter into your mail application in order to send me a message. In much the same way a telephone number is an identity that is formed from describing your actions if you want to call me.

But heading too far down the path of generating identity spaces based on algorithms of how to access the specific resource, trigger the particular application or contact a particular individual or role's network point of presence, or in other words the URI space, all have problems with maintaining referential integrity. The human world, and its digital counterpart, is far from static. Any identity system that aspires to be useful in a human space needs to be able to support a maintenance function that allows any implicit reference that is contained in an identity space to be updated and refreshed in a reliable, trustable and timely manner, Knowing who you were is a less important piece of information as compared to knowing who you are right now. That leads to consideration of structured identity spaces whose two major attributes are:

- sufficient structure to ensure that specific instances of the identity are unique, and
- appropriate structure to allow rapid lookup of the identity to be able to retrieve the current set of associated pointers within various specified realms.

This is a relatively precise description of the DNS, and its becoming clear that the major underpinning of useful and lasting digital identities rests within the framework of the DNS. Its a common cry in the IETF when considering various information distribution and reference structures to "just put it in the DNS!" and then just distribute the reference to the information as a DNS name. Its a seductively easy option in many cases, and maybe in this case the simple ease with which such identity spaces can be constructed in the DNS is further evidence that its a good answer. But its hard to advocate putting everything into the DNS. There is a general cautionary note about over-burdening the DNS and creating a single point of vulnerability and risk. There is also the observation that the DNS itself operates on top of a framework of addressing and routing, and some intractable circularities are created if identity realms in the routing and addressing domain rely on the stable operation of the DNS.

But it does appear that the level of attention that has been given to the practice and policies of the DNS operation has some justification, for, outside of the commercial opportunities in selling names to those who thought of the name in the first place, there is a more solid framework of a robust and carefully crafted identity space that allows for referential integrity and accuracy, and a means of separating the identifier from the object being identified.

But what about you and me? I don't think we are DNS objects, or not yet in any case. The general class of identity systems as they apply to you and me are based on a mix of what you know (secrets), what you are (unique tokens you generate), or what you possess (unique tokens you are assigned by some external entity). When applied to the Internet world you need to start with a trustable method of establishing identity (what you are), and generating a public confirmation (what you possess) that your digital identity is uniquely associated with you (what you know). If other words who you are is what you know applied to what you possess using what you are. And while R.D. Laing may have disagreed with the proposition itself, he may have found the underlying pattern appealing!

Ronald David Laing was one of the more controversial figures of 20th Century psychology and philosophy. His writings drew upon psychoanalysis, mysticism, existentialism and politics. He saw us as people entrapped by the pressures of social conformity. He explored notions of identity, self and sanity and wrote on a theme of how contemporary culture conspires to rob us of our individuality.

if it's not me
if it's me
if it's not me
if it's me
if it's not me, it's me
if it's not me, it's not me
it's not me if it's me
if it's me it's me
it's me if it's not me
if it's not me, it's not me
if it's not me, it's me
it's me if it's me
if it's not me it's not me
it's not me if it's not me
if it's me, it's me
if it's me, it's me
I am it
if it is not me
if it is not me, I am it, if I am not it, I am it, if I am it, I am not it

R. D Laing, Knots, 1969

Disclaimer

The author is a member of the Internet Architecture Board (IAB). The opinions expressed in this article are entirely those of the author, and are not necessarily shared by the IAB as a whole.

The above views do not represent the views of the Internet Society, nor do they represent the views of the author's employer, the Telstra Corporation. They were possibly the opinions of the author at the time of writing this article, but things always change, including the author's opinions!

About the Author

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also a member of the Internet Architecture Board, and is the Secretary of the APNIC Executive Committee. He was an inaugural Trustee of the Internet Society, and served as Secretary of the Board of Trustees from 1993 until 2001, with a term of service as chair of the Board of Trustees in 1999 – 2000. He is author of *The ISP Survival Guide*, ISBN 0-471-31499-4, *Internet Performance Survival Guide: QoS Strategies for Multiservice Networks*, ISBN 0471-378089, and coauthor of *Quality of Service: Delivering QoS on the Internet and in Corporate Networks*, ISBN 0-471-24358-2, a collaboration with Paul Ferguson. All three books are published by John Wiley & Sons.

E-mail: gih@telstra.net