Design Principles for the Trust over IP Stack

Version 1.0

17 November 2021

This publicly available recommendation was approved by the ToIP Foundation Steering Committee on 17 November 2021.

The mission of the Trust over IP (ToIP) Foundation is to define a complete architecture for Internet-scale digital trust that combines cryptographic assurance at the machine layer with human accountability at the business, legal, and social layers. Founded in May 2020 as a non-profit hosted by the Linux Foundation, the ToIP Foundation has over 250 organizational and 100 individual members from around the world.

Please see the end page for licensing information and how to get involved with the Trust Over IP Foundation.

Table of Contents

Document Information

Editors

Drummond Reed – Evernym
Victor Syntez

Contributors

Antti Kettunen Daniel Bachenheimer — Accenture Daniel Hardman — SICPA Darrell O'Donnell — Continuum Loop Jacques Bikoundou Jo Spencer — 460degrees John Jordan — Province of British Columbia Jonathan Rayback — Evernym Judith Fleenor — Trust Over IP Foundation Lynn Bendixsen — Indicio Mary Lacity — University of Arkansas Michel Plante	Neil Thomson — QueryVision P. A. Subrahmanyam — CyberKnowledge Rieks Joosten — TNO Sankarshan Mukhopadhyay — Dhiway Networks Scott Perry — Scott S. Perry CPA, PLLC Steven McCown — Anonyome Labs Thomas Cox Vikas Malhotra — WOPLLI Technologies Vinod Panicker — Wipro Ltd Wenjing Chu — Futurewei

Revision History

Version	Date Approved	Revisions
1.0	17 November 2021	Initial Publication

Terms of Use

Introduction

This document is a joint deliverable of the Technical Stack Working Group and Governance Stack Working Group of the Trust Over IP (ToIP) Foundation. Its purpose is to enumerate the set of design principles that inform, guide, and constrain the design of the **ToIP stack**—the heart of the ToIP model for decentralized digital trust infrastructure (figure 1).

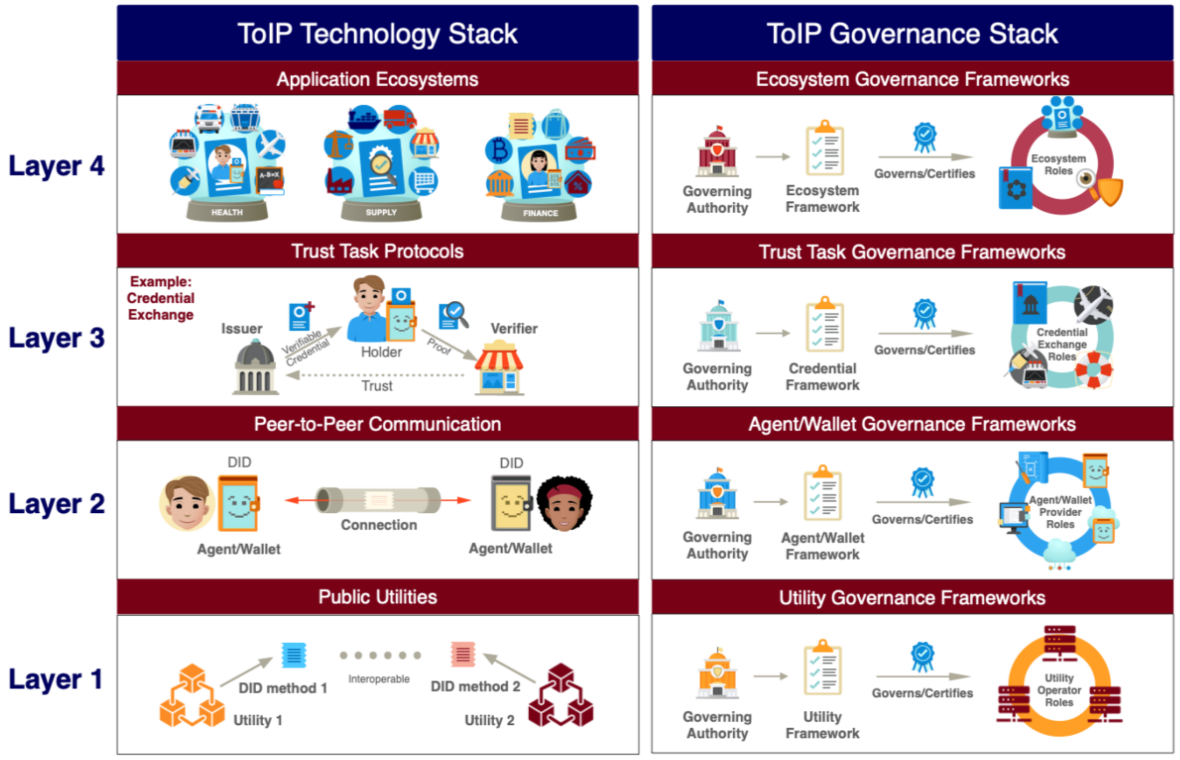


Figure 1. Overall architecture of the ToIP stack showing the four layers and two halves

The ToIP website now features a fully interactive graphic version of the ToIP stack—we recommend exploring this to gain a broader understanding of each cell in Figure 1.

We also recommend reading this document in conjunction with three other deliverables from the ToIP Foundation:

1. Introduction to ToIP—this white paper provides an overall introduction to the problem space addressed by decentralized digital trust infrastructure, the origin and basic structure of the ToIP stack, and the mission and activities of the ToIP Foundation.
2. ToIP Technology Architecture Specification defines the overall technical requirements for the four layers of the ToIP technology stack.
3. ToIP Governance Architecture Specification defines the overall requirements for ToIP-compliant governance frameworks.

As with all ToIP deliverables, the ToIP Foundation invites your feedback and suggestions. Please feel free to contact us via any of the channels listed on the ToIP Foundation website.

What are Design Principles?

A design principle is a proposition or value that informs, guides, and constrains the design of a product, service, or system.

The goal of any design principle[1] is to provide guidance to the designers of a product, service, or system so they can take advantage of lessons learned from the success or failure of previous designs. Design principles represent accumulated wisdom that falls in between the generality of scientific laws and the specialization of best practices.

When it comes to computer software and network design, design principles are closely related to *design patterns*. To quote from the Wikipedia article on this topic:

> *In software engineering, a software design pattern is a general, reusable solution to a commonly occurring problem within a given context in software design. It is not a finished design that can be transformed directly into source or machine code. Rather, it is a description or template for how to solve a problem that can be used in many different situations.*

Software and network design principles and patterns can be applied to systems of any type or scale. For example, in 1998 Sir Tim Berners-Lee published his Principles of Design for the entire World Wide Web. In the opening paragraph he explains:

> *Again and again we fall back on the folklore of the principles of good design…. Principles such as **simplicity** and **modularity** are the stuff of software engineering; **decentralization** and **tolerance** are the life and breath of [the] Internet.*

Even closer to home is the set of nine design principles published by the U.K. National Health Service (NHS) in 2018. Intended to guide the development of all NHS products, services, and systems, these principles are "inspired by the NHS Constitution that's steered the NHS for 70 years." They are concise enough to summarize in a single list:[2]

1. Put people at the heart of everything you do.
2. Design for the outcome.
3. Be inclusive.
4. Design for context.
5. Design for trust.
6. Test your assumptions.
7. Make, learn, iterate.
8. Do the hard work to make it simple.
9. Make things open. It makes things better.

[1] The term "design principle" can be applied broadly to many different types of design: graphic design, product design, software design, computer network design and so on. The Design Principles website maintains a catalogue of 195 examples of over 1000 design principles.

[2] We included this full list because several of these NHS design principles are directly relevant to ToIP.

For the purposes of this document, the contributing ToIP Working Groups have focused on principles that will help us inform, guide, and constrain the design of the ToIP stack as an Internet-scale architecture for decentralized digital trust. To that end, each of the design principles in this document concludes with the following table summarizing the specific ways in which that principle is applicable to the design of each of the four layers of the ToIP stack.

Layer	Relevance	Explanation
4		The ecosystem symbol represents the purpose of Layer 4 to support the applications needed to develop and sustain entire digital trust ecosystems.
3		The triangle symbol represents the Layer 3 verifiable creden-tial "trust triangle" of issuer, holder, and verifier that enables parties using the ToIP stack to establish transitive trust.
2		The symbol of two connected mobile phones represents the purpose of Layer 2 as a universal peer-to-peer secure privacy-routing DID-to-DID communications protocol.
1		The anchor symbol represents the purpose of Layer 1 public key utilities to provide strong anchors for Decentralized Identifiers (DIDs) and their associated public keys.

In the "Relevance" column, we assign star ratings for each layer as follows:

★★★★★	Highly relevant to the design of this layer
★★★★	Very relevant to the design of this layer
★★★	Moderately relevant to the design of this layer
★★	Somewhat relevant to the design of this layer
★	Only slightly relevant to the design of this layer
	Not relevant to the design of this layer

Our goal, much like that of Sir Tim Berners-Lee or the NHS, is to establish and maintain a set of principles against which each deliverable from the ToIP Foundation can be evaluated to see if it is consistent both in spirit and in substance.

Organization of these Principles

The principles in this document are organized in three categories based on an observation from cryptography pioneer Nick Szabo, who distinguished between two types of "code":[3]

1. Code written in a computer language expected to be executed by a machine ("dry code"), and
2. Code written in a human language, i.e., laws, regulations, rules, policies and other forms of governance expected to be followed by humans ("wet code").

Accordingly, our three categories are:

1. **Principles of computer network architecture**—the "dry code" principles that represent fundamental lessons learned about the design of large-scale computer networked systems, especially the Internet.
2. **Principles of human network architecture**—the "wet code" principles that represent fundamental truths about how trust relationships operate between humans—either individually or in groups.
3. **Overall design principles**—the remaining principles that apply to the overall design of the ToIP stack, "wet or dry".

Although each principle can be taken on its own, we recommend reading them in the order presented because they build upon each other, both within each category and as a whole.

[3] Lawrence Lessig makes a similar distinction in his 2006 book Code: Version 2.0, based on his original 1999 book Code and Other Laws of Cyberspace.

Terminology

When it comes to digital identity, security, privacy, and decentralized digital trust infrastructure, **terminology matters**. In particular, since people from different backgrounds tend to interpret terms differently, we need to find and adopt mechanisms that allow us to (a) establish a common understanding about concepts (ideas) that are relevant within the context of ToIP, and (b) a way for assigning them terms that can be used unambiguously. This is the purpose of the ToIP Foundation Concepts and Terminology Working Group.

In general, the design principles in this paper, since they apply the design of the ToIP stack as a whole, do not require specialized terminology. However, it is important to be precise about a handful of terms that describe relationships and interactions between *people* because these terms need to be understood in both a technical and legal context.

To address this, we use terms defined in the Parties-Actors-Actions model developed by the EU European Self-Sovereign Identity Lab (eSSIF-Lab) as part of the eSSIF-Lab Glossary work (licensed under CC BY-SA 4.0). As shown in Figure 2 below, this is a mental model[4] of how to precisely describe and categorize the entities that participate in human interactions (the notations and colors used in this diagram are explained on this eSSIF-Lab page).

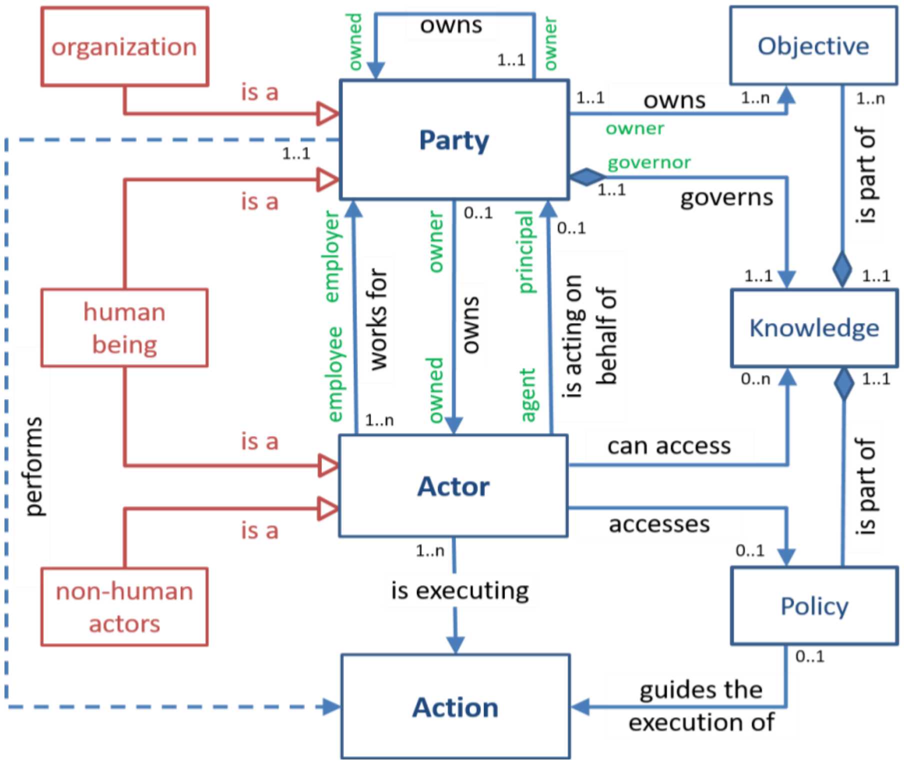


Figure 2. The eSSIF-Lab Parties-Actors-Actions mental model

[4] A mental model is a description of a set of concepts (ideas), relations between them, and constraints, that together form a coherent and consistent 'viewpoint', or 'way of thinking' about a certain topic.

The key terms we will be using from this model are explained (and hyperlinked to their definitions) in the following description from the eSSIF-Lab site:

> *[The model] shows that parties (humans, organizations) perform actions for the purpose of realizing their objectives. Parties are not considered to actually execute such actions; they have (human and non-human) actors that work for them, execute such actions, using the party's knowledge as the (authoritative) guidance for executing the actions (as well as any other relevant knowledge they can access).*
>
> *Perhaps the most important idea in this pattern is that a party is not considered to (be able to) act directly—they need actors (i.e. entities that can act) to act on their behalf and thus make them perform. This does, however, not preclude having entities that are both party and actor—e.g. humans—and that such entities can act on their 'own' behalf. And we can continue to use the commonly used form of speech in which a party performs some action by realizing that this means that there is (at least) one actor that is actually executing that action.*
>
> *This works well for human beings, which are both a party and an actor. So a human being can act, implying itself as an actor, and using its personal knowledge as guidance. The model also works when a human being (as a party) may hire someone else (as an actor), e.g. to fill in a tax return form. This other is guided by the knowledge of the human being that hired her, and uses its own knowledge for the details of filling in the tax form.*
>
> *It also works well for organizations, which are typically companies, enterprises, governments or parts thereof, i.e. groups of human beings and possibly other actors that, as a group, fit the criteria for being a party. This group of actors would typically work to realize the organization's objectives, being guided by the organization's knowledge (registrations, policies, etc.). Like human beings, an organization may (have an appropriate actor) decide to hire or fire actors for longer or shorter periods.*

About "Technology Independence"

There has been some confusion about whether the ToIP stack is "technology independent" or whether it presumes specific technologies.

Neither is true. These questions stem from misconceptions about the role of the ToIP stack. Keep these points in mind as you read this document:

- **First, the ToIP stack is still being designed.** A primary purpose of this document is to deeply elucidate the principles governing design choices made by the ToIP Technical Stack Working Group and ToIP Governance Stack Working Group.
- **Secondly, the ToIP Technology Stack is fundamentally a protocol stack.** The technology half of the ToIP stack functions like other protocol stacks such as the TCP/IP stack. Therefore, the process of completing specifications for the ToIP Technology Stack will result in a suite of protocols against which interoperability test suites and—if needed—certification programs can be developed.
- **Every protocol in the ToIP Technology Stack must be implementation-independent.** This is true of any truly open standard—the programming languages and toolsets used to implement that standard can (and will) vary.
- **As of this writing, no specific protocol choices have been "hard-coded" into the design of the ToIP Technology Stack.** Certain data model and digital signature specifications—such as W3C Verifiable Credentials (VCs)—and certain protocols—such as DIDComm—have been explicitly designed for decentralized digital trust infrastructure. So these are obvious candidates. However the Technical Stack Working Group is not locked into these choices, nor are any of these choices necessarily exclusive. Again, the main purpose of this document is to define the design principles that will govern the ultimate choices.

A special note about DIDs. At the time this paper was written, the W3C Decentralized Identifiers (DIDs) specification for cryptographically-verifiable identifiers enjoys broad support. Since this new class of identifiers serve as an atomic building block of ToIP architecture, in this paper we will assume DIDs will be supported. However that does not mean they are the only cryptographically verifiable identifier in the ToIP stack—for example KERI autonomic identifiers (AIDs) may also be supported.

If you have a specific interest in participating in the design work for the ToIP Technology Stack, please consider joining the ToIP Technology Stack Working Group (and in particular the Technology Architecture Task Force). If your interest lies in designing governance for decentralized digital trust infrastructure, please consider joining the ToIP Governance Stack Working Group.

Part One: Computer Network Architecture ("Dry Code") Principles

This first category of principles is derived in large part from lessons learned about the design of the primary network over which ToIP is designed to operate—the Internet. The first four of these principles are based directly on architectural principles of the Internet itself. The other three deal with elements that were missing from the original design of the Internet but which are now clearly needed to establish the "missing trust layer".

#1: The End-to-End Principle

For maximum utility and adaptability, the best place to put intelligence and processing is at the endpoints of a network and not in the communications subsystems (routers, gateways, etc.) that connect those endpoints.

The End-to-End Principle is often cited as the most fundamental principle in the design of the Internet itself. Although the "endpoints" in ToIP architecture are at a different layer of abstraction from those at the Internet layer (see figures 3 and 4 below), the End-to-End Principle is just as important—for the same reasons. A cogent summary of this principle is given on the Wikipedia page of the same name:

> *The end-to-end principle is a design framework in computer networking. In networks designed according to this principle, application-specific features reside in the communicating end nodes of the network, rather than in intermediary nodes, such as gateways and routers, that exist to establish the network.*

The End-to-End Principle is the primary focus of IETF RFC 1958, **Architectural Principles of the Internet**. In section 2.1 it says:

> *The key to global connectivity is the inter-networking layer. The key to exploiting this layer over diverse hardware providing global connectivity is the "end to end argument".*

Section 2.3 goes on to explain:

> *It is also generally felt that end-to-end functions can best be realised by end-to-end protocols.*
>
> *The end-to-end argument is discussed in depth in [Saltzer][5]. The basic argument is that, as a first principle, certain required end-to-end functions can only be performed correctly by the end-systems themselves. A specific case is that any network, however carefully designed, will be subject to failures of transmission at some statistically determined rate. The best way to cope with this is to accept it, and give responsibility for the integrity of communication to the end systems. Another specific case is end-to-end security.*
>
> *To quote from [Saltzer], "The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be*

[5] https://datatracker.ietf.org/doc/html/rfc1958#ref-Saltzer

useful as a performance enhancement.")

This principle has important consequences if we require applications to survive partial network failures. An end-to-end protocol design should not rely on the maintenance of state (i.e. information about the state of the end-to-end communication) inside the network. Such state should be maintained only in the endpoints, in such a way that the state can only be destroyed when the endpoint itself breaks (known as fate-sharing). An immediate consequence of this is that datagrams are better than classical virtual circuits. The network's job is to transmit datagrams as efficiently and flexibly as possible.

Everything else should be done at the fringes.

From an architectural design standpoint, the implications of the End-to-End Principle are summarized by this extensive article on the principle on Devopedia.org. The article begins:

When a function has to be supported in a networked system, the designer often asks if it should be implemented at the end systems; or should it be implemented within the communication subsystem that interconnects all the end systems. The end-to-end argument or principle states that it's proper to implement the function in the end systems. The communication system itself may provide a partial implementation but only as a performance enhancement.

The architecture and growth of the Internet was shaped by the end-to-end principle. It allowed us to keep the Internet simple and add features quickly to end systems. The principle enabled innovation.

More recently, the principle has been criticized or violated to support features such as network caching, differentiated services, or deep packet inspection.

As a final reference, *every* one of the sources cited above cite the same seminal 1981 paper[6] by J.H. Saltzer, D.P. Reed and D.D. Clark from the MIT Laboratory for Computer Science entitled **End-to-End Arguments in System Design**. Devopedia.org describes the paper this way (emphasis added):

Given a distributed system, the paper gives guidance on where to place protocol functions. For example, ***end systems should perform recovery, encryption and deduplication****. Low-level parts of the network could support them only as performance enhancements. Phil Karn, a well-known Internet contributor, comments years later that this is "the most important network paper ever written".*

The highlighted passage emphasizes another principle, *Keys at the Edge*, that has special relevance when it comes to digital trust (see Principle #7).

[6] https://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf

How this principle applies to the ToIP stack

The specific relevance of the End-to-End Principle to ToIP architecture can be summarized as follows:

- **The trust layer of the Internet provides the same benefits of end-to-end protocol design as the Internet itself**—for all the same reasons articulated above.
- **The only way to deliver end-to-end security and end-to-end privacy to parties communicating over the Internet is with end-to-end protocols.** As soon as a protocol requires an intermediary to decrypt and re-encrypt messages, end-to-end security and privacy guarantees can no longer be made. This is why one candidate for the Layer 2 peer-to-peer communications protocol, DIDComm, is transport agnostic and derives its security and privacy properties from DIDs rather than from intermediate infrastructure. This also improves overall security because access to a message's contents is reserved for the endpoints (sender and receiver).
- **Only end-to-end protocol design can shield applications from needing to deal with network state.** With end-to-end protocols, applications at the edge only need to deal with application state. This is particularly important when it comes to complex security operations such as data normalization and canonicalization prior to encryption and decryption. Leaving these to edge applications means developers and test suites can focus on determining state locally and avoid the potential explosion in complexity of having to deal with state somewhere else in the network.
- **Parties can focus their trust decisions on the endpoints with which they will interact and not on intermediaries.** With end-to-end protocols, all that matters is the level of trust assurance a party can achieve in the party at the other endpoint in the context of a specific interaction at a specific point in time. Parties can use all the facilities of the ToIP stack—DIDs, verifiable credentials, governance frameworks, trust registries—to accumulate the evidence necessary to make this trust decision.

The following two diagrams provide a simple way to visualize the application of the End-to-End Principle to ToIP architecture. Figure 3 illustrates how the End-to-End Principle is applied to the Internet today: every device has an endpoint identified with an IP address and runs an instance of the TCP/IP protocol stack in order to send and receive data packets.

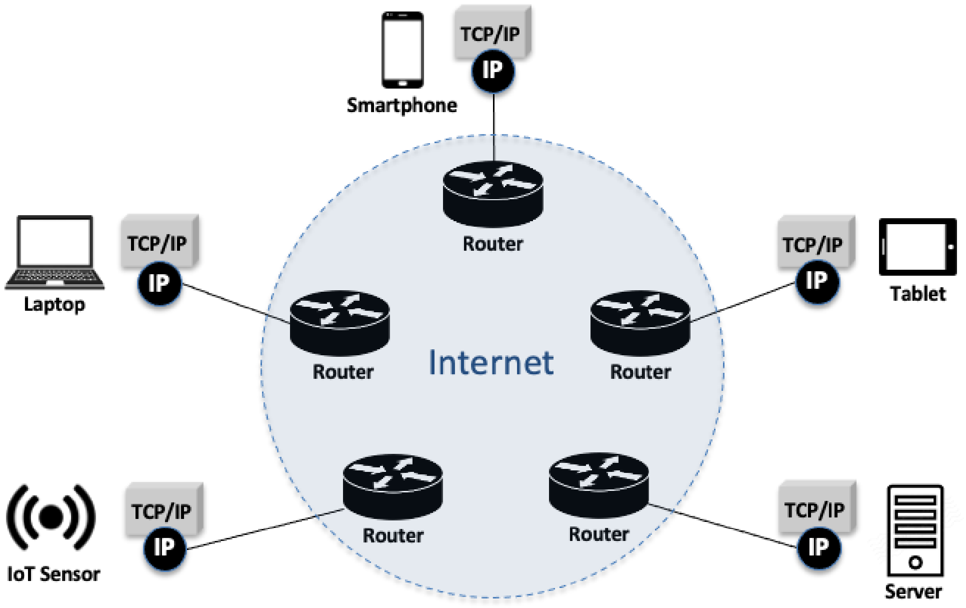


Figure 3. Every device on the Internet runs an instance of the TCP/IP protocol stack so data packets travel "end-to-end" from IP address to IP address

Figure 4 illustrates the same principle applied to ToIP architecture except at a higher layer. Every entity interacting within a digital trust ecosystem has at least one endpoint identified with a DID so it can send and receive secure messages from that endpoint using a DID-to-DID communications protocol at Layer 2 of the ToIP stack.

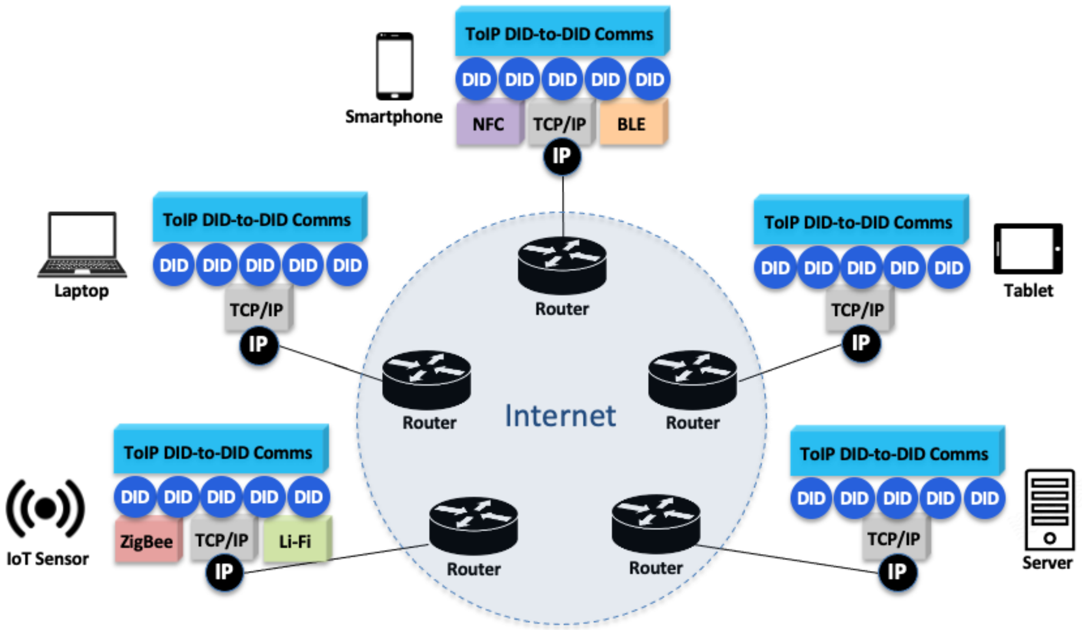


Figure 4. Every device in a ToIP digital trust ecosystem has at least one DID (and often multiple as shown here) and can securely communicate end-to-end between DIDs

Note this key difference in privacy architecture between the TCP/IP layer and the ToIP layer: at the TCP/IP layer, a device is typically assigned a single IP address for all communications to/from that device during an Internet session (even if the IP address is assigned dynamically). At the ToIP layer, *DIDs do not need to be tied to the underlying transport layer*. Rather, they can be assigned purely on the basis of trust relationships between the participants.[7] This means that, in order to maintain privacy and separation of contexts, an individual might maintain tens, hundreds or thousands of different DID endpoints on the same device.[8] Further, since the ToIP layer functions on top of the underlying transmission protocol layer(s), it can use other transport protocols such as Bluetooth Low Energy (BLE), Near-field Communications (NFC), Zigbee, and so on.

Our concluding table emphasizes that, while the End-to-End Principle is particularly relevant to ToIP Layer 2, end-to-end trust relationships need to be supported by all four layers of the ToIP stack.

Layer	Relevance	Explanation
4	★★★	The End-to-End Principle emphasizes the importance of trust registries as a tool for supporting end-to-end trust relationships across multiple digital trust ecosystems.
3	★★★	Both the issuer-to-holder and holder-to-verifier legs of the verifiable credential trust triangle at Layer 3 should be independent end-to-end connections.
2	★★★★★	The End-to-End Principle applies primarily at Layer 2 because this is the layer at which peers connect to each other. See Principle #3 for more details.
1	★★★	The public utilities at Layer 1 indirectly support the End-to-End Principle because they provide the cryptographic anchors enabling cryptographically verifiability at the higher layers.

[7] This is the design goal of the did:peer method, for example.

[8] It might seem self-defeating from a privacy standpoint to use different DIDs on a device if they can potentially be correlated to a single underlying IP address. However, there are numerous solutions for redirection or onion-routing to prevent correlation of communications at the Internet layer.

#2: Connectivity Is Its Own Reward

If connectivity is held up as the highest goal—as it was with the design of the Internet—then every design decision about the ToIP stack will be made in favor of interoperability.

In section 2 of IETF RFC 1958, **Architectural Principles of the Internet**, under the title "Is there an Internet Architecture?", it says (emphasis added):

> *"However, in very general terms, the community believes that the goal is connectivity, the tool is the Internet Protocol, and the intelligence is end-to-end rather than hidden in the network. The current exponential growth of the network seems to show that **connectivity is its own reward**, and is more valuable than any individual application such as mail or the World-Wide Web."*

The use of the phrase "connectivity is its own reward" elevates the value of connectivity in the design of the Internet above all other nice-to-have properties.

The value of connectivity has been demonstrated in some of the most fundamental disruptions that the Internet has brought—for example, moving telecommunication from landlines to mobile phones and then to the Internet with Voice-over-IP (VoIP). With mobile phones, we were no longer chained by the phone cord, nor were we forced to live with one phone number shared by all of the household. We were given a new freedom to be reached anywhere, anytime, person-to-person.

However, the true value of transitioning from landlines to mobile wasn't realized until the smartphone revolution. This is when we started to see value creation move from the telecom networks to the devices (in accordance with the End-to-End Principle). VoIP disrupted single-purpose telecom networks with more flexible, powerful data infrastructure—the Internet. Now multi-purpose smart devices could provide many ways of communicating "over the top" with apps like Skype, WhatsApp, Twitter and others. The underlying infrastructure—the TCP/IP network layer—was the same for all the edge devices, but applications created an exponential value increase which created an explosion in connectivity. Jupiter Research estimates that by the end of 2021 there will be over 46 billion devices connected to the Internet.[9]

[9] https://www.juniperresearch.com/researchstore/key-vertical-markets/internet-of-things

How this principle applies to the ToIP stack

This principle applies to the ToIP stack the same way the End-to-End Principle does. In other words, the goal of the ToIP stack (literally, "Trust over the Internet Protocol") is to add trust functions to the Internet architecture that will only further enhance global connectivity.[10] Connectivity in this context should be understood as *the ability to exchange trustworthy information between any two endpoints*.

The analogy to the smartphone communications revolution clearly applies here. Today, your ability to have a trusted private channel with another party depends on either:

1. You hosting your own Internet server with your own TLS digital certificate, or
2. You and your counterparty using the same private communications platform (e.g., WhatsApp, Signal, Telegram, etc.)

This is like being tied down to a shared landline telephone. With ToIP infrastructure, everyone and everything will be able to enjoy their own trusted private communications channels—as many as they need—without having to run their own servers and without having to use a proprietary third-party platform. In the MyData ecosystem, this is referred to as the "API of me"—an API controlled fully by the individual, so it is much less susceptible to spam or other unwanted messages and only works with applications and parties you trust.

The specific implications of the *Connectivity Is Its Own Reward* principle for the design of the ToIP stack include:

1. **The ToIP stack should be able to work with any kind of Internet-connected device.** We should not be satisfied by merely having ToIP apply to some systems with some set of limited applications. Our goal should be to be able to enable trustworthy communications between any two endpoints that need them—everything from host machines to mobile devices to Internet of Things (IoT) sensors.
2. **All ToIP endpoints should be able to connect using a single standard protocol.** This solution for maximizing connectivity at ToIP Layer 2 is discussed at length in Principle #3: The Hourglass Model.
3. **Maximizing connectivity also means maximizing backwards compatibility and adaptability.** The power of the Internet protocol stack was not just that it enabled any two network endpoints to connect with each other, but that it could be adapted to all the existing local area networks and gateways that were already operating. The same applies to the ToIP stack—it needs practical and low-cost methods of integrating it to work with existing centralized and federated identity management systems and protocols such as OpenID Connect, SAML, and OAuth.

[10] Despite its name, the ToIP Layer 2 protocol is not limited to running over TCP/IP. It can also run over other transport protocols such as Bluetooth, NFC, 5G, mesh protocols, etc.

In short, “Connectivity is its own reward” means that when we face design choices pitting functionality against connectivity, we lean towards solutions that maximize the latter.

Layer	Relevance	Explanation
4	★★	At Layer 4, connectivity is maximized using universally accessible ecosystem governance frameworks and trust registries so anyone can verify the members of an ecosystem.
3	★★★	At Layer 3, the primary relevance of this principle is interoper- ability of the data formats, signatures, and protocols used to exchange verifiable credentials or otherwise establish trust.
2	★★★★★	As with the End-to-End Principle, this principle applies primarily at Layer 2 where all ToIP-enabled devices connect using a single standard secure protocol. See also Principle #3.
1	★★	At Layer 1, the issue of connectivity only arises in terms of the availability and usage requirements for a Layer 1 public utility.

#3: The Hourglass Model

In a layered protocol architecture, the most successful design takes an hourglass shape where a single "spanning layer" in the middle connects a family of higher-level application-facing protocols with a family of lower-level transport protocols.

After the End-to-End Principle, the hourglass model is perhaps the most widely recognized hallmark of the Internet protocol suite. The "hourglass" metaphor is evident from looking at a diagram of the TCP/IP stack such as Figure 5 from a presentation by Steve Deering of Cisco given at the IETF 51 meeting in London in August 2001.

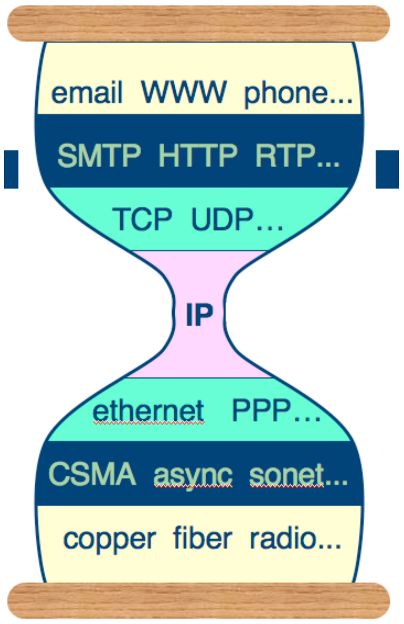
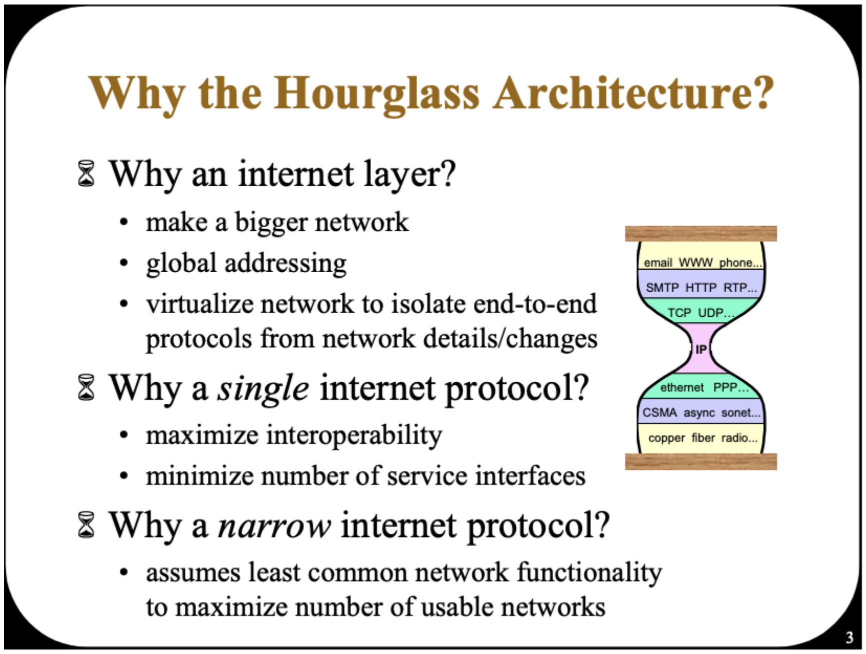


Figure 5. The Internet protocol suite forms the shape of an hourglass

The single "spanning layer" at the "waist" of the hourglass is the IPv4 protocol. A 2019 paper from Communication of the ACM called On the Hourglass Model calls this the "distinguished layer" and summarizes the rationale for it this way (emphasis added):

> *The hourglass model of layered systems architecture is a visual and conceptual represent-ation of an approach to design that seeks to support a great diversity of applications and allow implementation using a great diversity of supporting services. At the center of the hourglass model is a distinguished layer in a stack of abstractions that is chosen as the sole means of accessing the lower-level resources of the system. This distinguished layer can be implemented using services that are considered as lying below it in the stack as well as other services and applications that are considered as lying above it.* ***However, the components that lie above the distinguished layer cannot directly access the services that lie below it.***

The presentation from Steve Deering cited above goes on to state the case this way:



Why the Hourglass Architecture?

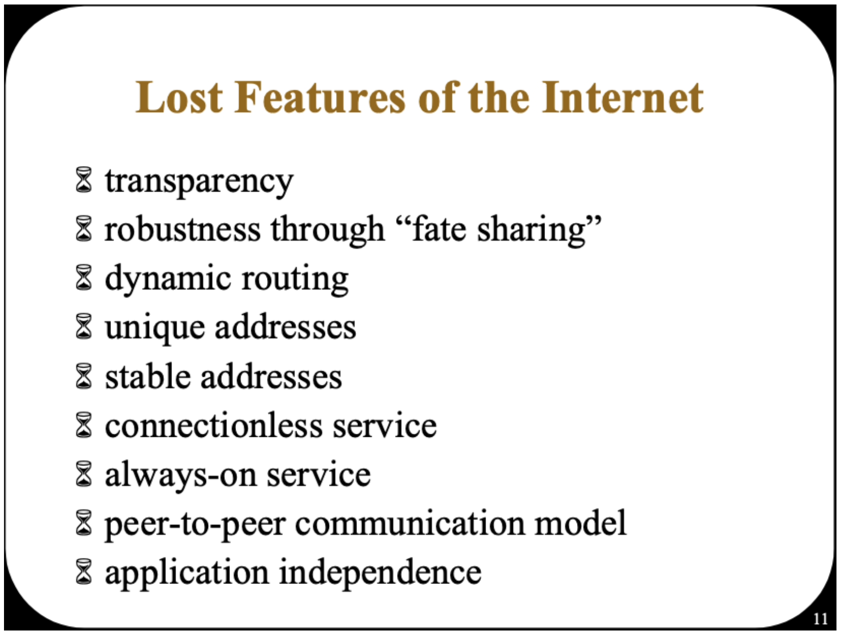
⌛ Why an internet layer?
- make a bigger network
- global addressing
- virtualize network to isolate end-to-end protocols from network details/changes

⌛ Why a *single* internet protocol?
- maximize interoperability
- minimize number of service interfaces

⌛ Why a *narrow* internet protocol?
- assumes least common network functionality to maximize number of usable networks

Figure 6. A summary of the rationale for the hourglass model

The rest of Mr. Deering's presentation delves into various efforts to "fatten" the "waist" of the hourglass by "squeezing other protocols into the middle". He strongly recommends resisting this. Figure 7 is his slide summarizing the Internet features that would be lost if the simplicity and universality of the IP protocol was not maintained at the waist.



Lost Features of the Internet

⌛ transparency
⌛ robustness through "fate sharing"
⌛ dynamic routing
⌛ unique addresses
⌛ stable addresses
⌛ connectionless service
⌛ always-on service
⌛ peer-to-peer communication model
⌛ application independence

Figure 7. Features that would be lost if the hourglass model is not maintained

The hourglass model is not an accident. A 2011 Georgia Institute of Technology study called How the Internet architecture got its hourglass shape and what that means for the future[11] used a computer model called EvoArch (evolution of architecture) that simulated the evolution of network protocols as shown in Figure 8.

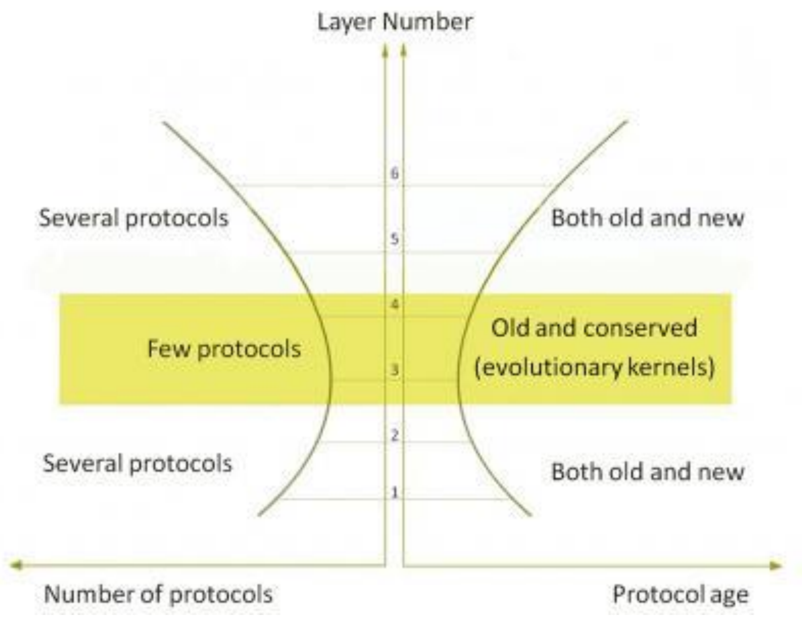


Figure 8. The EvoArch model simulated how protocol stacks naturally evolve

The authors of the study concluded:

EvoArch showed that even if future Internet architectures are not built in the shape of an hourglass initially, they will probably acquire that shape as they evolve. Through their simula-tions, Dovrolis and Akhshabi found that while the accuracy of the structure improved with time, the basic hourglass shape was always formed -- no matter what shape it started in.

"Even though EvoArch does not capture many practical aspects and protocol-specific or layer-specific details of the Internet architecture, the few parameters it is based on -- the generality of protocols at different layers, the competition between protocols at the same layer, and how new protocols are created -- reproduced the observed hourglass structure and provided for a robust model," said Dovrolis.

How this principle applies to the ToIP stack

If the spanning layer for the Internet—the IP protocol—is already firmly established (and not going anywhere for decades or centuries), then how can the hourglass model be applied to the ToIP stack—which by definition runs on top of the TCP/IP stack (as well as other transport protocols)?

[11] https://phys.org/news/2011-08-internet-architecture-hourglass-future.html

The answer is by implementing a second **higher-level spanning layer** that serves the same purpose for universal trusted communications that the Internet's IP layer serves for universal data communications. To illustrate this, let's start with Figure 9 that illustrates the function of the Internet's IP protocol spanning layer.

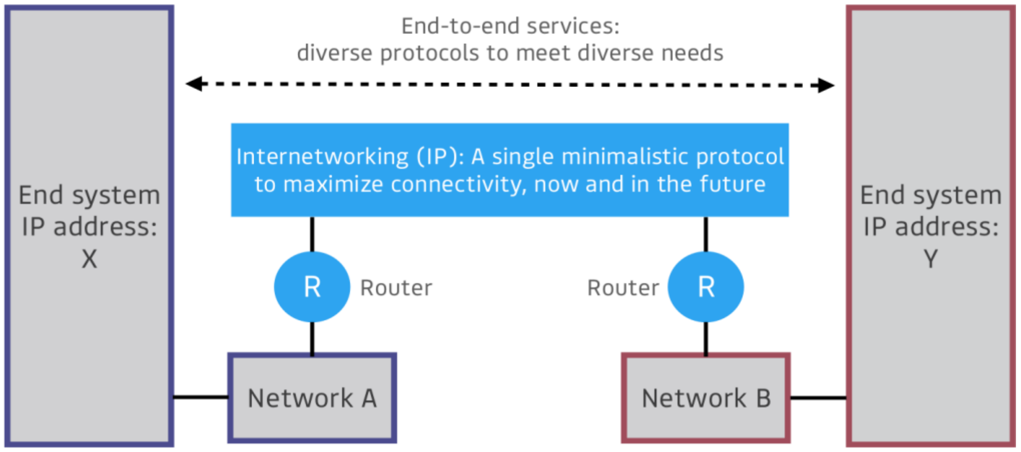

End-to-end services:
diverse protocols to meet diverse needs
Internetworking (IP): A single minimalistic protocol to maximize connectivity, now and in the future
End system IP address: X
End system IP address: Y
R
Router
Router
R
Network A
Network B

Figure 9. The role of the Internet Protocol as the spanning layer for the Internet

To maximize connectivity (Principle #2), the Internet Protocol supports a basic datagram service without reliability guarantees. It uses a globally unique endpoint identification scheme (IP addresses) that is routable and scalable, but otherwise very simple. Its requirements for both end systems and network routing equipment are minimal, thus all types of devices can be connected to the Internet and scaled economically. The Internet Protocol is also agnostic as to what kinds of physical networks are used to interconnect systems. It demands very little from the transport medium, which means all existing network technologies could be used easily and efficiently to construct the Internet.

Now look at Figure 10 illustrating that the ToIP Layer 2 protocol should be designed to fulfill the same types of requirements for the ToIP stack.

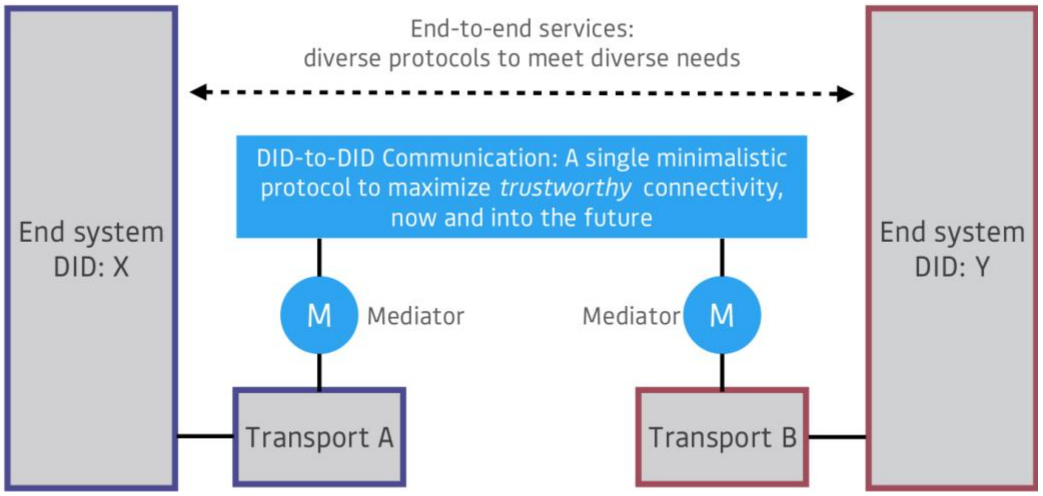

End-to-end services:
diverse protocols to meet diverse needs
DID-to-DID Communication: A single minimalistic protocol to maximize trustworthy connectivity, now and into the future
End system DID: X
End system DID: Y
M
Mediator
Mediator
M
Transport A
Transport B

Figure 10. The role of the ToIP Layer 2 DID-to-DID communications protocol as the spanning layer for trustworthy communications over any transport network

This means the ToIP Layer 2 protocol should follow a set of design considerations similar to the Internet Protocol:

- **It should be a simple secure messaging service** that provides the basic ingredients higher layers will need to construct trust relationships and exchange trusted data: mutual authentication, message integrity, and privacy (no leakage of unnecessary information).
- **It should use a globally-unique end point identification scheme** (such as DIDs) that is decentralized (Principle #4) and supports cryptographically verifiability (Principle #5) and confidentiality (Principle #6), but otherwise is simple, flexible, and scalable.
- **Messages should be inherently routable** similar to how emails are routed.
- **It should be designed so existing computing devices can easily support it**. Less powerful devices such as small sensors on batteries should be able to proxy their implementation to a more capable companion because it is a simple messaging service.
- **It should be agnostic to the transport protocol below it.** The current TCP/IP protocol suite, or one of its many transport services, should obviously be one such transport layer. However, non-Internet methods, such as Bluetooth, NFC, QR, or even plain old mail services could also serve as viable transport services as needed.

DIDComm is one example of a protocol that has been specifically designed to meet these requirements. In DIDComm, all endpoints are DIDs. They can be anchored on ToIP Layer 1 public utilities when such anchoring is needed to establish sufficient trust. However DIDComm does not require DIDs to be anchored in ToIP Layer 1—it also supports other DID methods such as did:peer and did:key which operate entirely at Layer 2. DIDComm messages can be exchanged directly peer-to-peer or they can be exchanged end-to-end over the network using message routing agents called “mediators”.

However, even if a protocol such as DIDComm is adopted as the universal trust spanning layer protocol at ToIP Layer 2, that does not reduce the need for integration with other legacy identity and access management (IAM) or federation protocols such as OpenID Connect, OAuth, SAML, or Shibboleth. This remains an essential task of ToIP Foundation Working Groups.

In conclusion, the evolution of a higher-level trust spanning layer was anticipated by Dr. Sam Smith in his 2019 paper Key Event Receipt Infrastructure (KERI) Design. In section 5, “Trust Spanning Layer”, he says:

> *Because an identity system security overlay necessarily uses protocols above the IP layer, [it] cannot span the internet at the IP layer but must span somewhere above it. This gives rise to a double waisted or “waist and neck shape” where the security overlay is the neck. This is shown in the following diagram [Figure 11].*

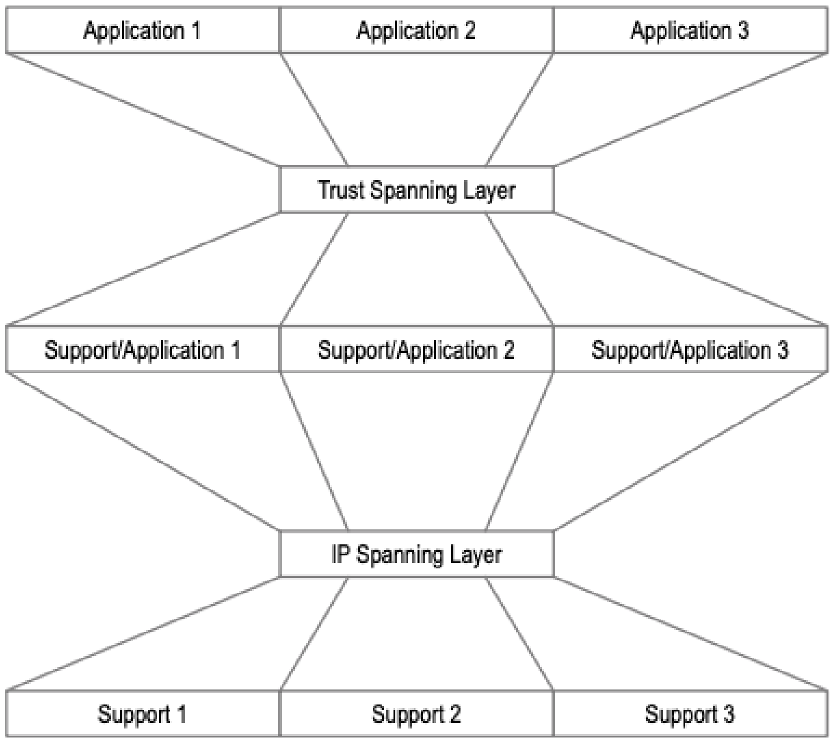


Figure 11. A trust spanning layer as a “secure neck” on top of the IP spanning layer “waist”

In other words, the trust spanning layer—the “neck”—needs to serve the same purpose within the ToIP stack that the IP protocol layer—the “waist”— serves within the TCP/IP stack. This illustrates perfectly how the Hourglass Model applies directly to the design of the ToIP stack—and why it may be the most prescriptive of all the recommendations in this paper. As our concluding table indicates, it applies primarily to ToIP Layer 2.

Layer	Relevance	Explanation
4		N/A
3	★★★	All Layer 3 protocols need to run on top of the Layer 2 protocol.
2	★★★★★	ToIP Layer 2 should be a single protocol that serves as a “trust spanning layer” connecting the ToIP layers above and below it. This protocol should be as simple and general as possible.
1	★★★	All Layer 1 public key utilities need to support access via the Layer 2 protocol.

#4: Decentralization by Design and Default

> To be trusted by all parties, a global network cannot favor any single centralized service or authority; it must allow functionality and authority to be distributed as widely as possible.

The Internet is often held up as one of the most successful examples of a decentralized system. A 2018 Hackernoon article put it this way:

> *[The Internet] was a project by the US DoD (Department of Defense) to establish a computer data communications network that could withstand unforeseen events and disasters like war. Therefore it must be decentralized so that if one part of the system fails the rest can still function. It must also be able to communicate using peer to peer interconnectivity without relying on a single computer. Another important consideration is that the computers must be interoperable among dissimilar systems, so that more devices can be a part of the network.*

The Mozilla Foundation maintains a site called The Internet Health Report that provides ongoing coverage of five key topics they deem essential to the health of the Internet: open innovation, digital inclusion, decentralization, privacy and security, and web literacy. On the principle of decentralization, they say:

> *Decentralization means the Internet is controlled by many. It's millions of devices linked together in an open network. No one actor can own it, control it, or switch it off for everyone.*
>
> *The Internet and the World Wide Web remain the biggest decentralized communication system humanity has ever seen. This was very much a part of the design: the inventors of the Web wished for all people to be able to create and access information.*

Having defined this goal, the Mozilla site goes on to sound this warning:

> *But the benefits of a decentralized Internet are eroding. When we concentrate our online activity on just a few social networks and messaging apps – as billions of us do – it narrows our experience of the Web to one where we are pointed only at content that appeals to our likes in search results and social media streams. Here, we are consumers rather than creators.*

These statements acknowledge the reality that, despite the decentralized architecture of the Internet, many software platforms have organized into large, centralized service architect-ures where their respective business services are administered and disseminated from a central service environment. This type of architecture often increases business return on investment. However, these types of systems are also at risk from supply-chain attacks that leverage single points of failure that can cascade into outages for downstream business customers. Examples of supply-chain attacks include the SolarWinds[12] Attack (2020) and the Kaseya Ransomware Attack[13] (2021) that both affected a wide range of commercial and government organizations. In addition to cyberattacks, other types of bugs or errors can result in interruptions in service. As an example, the October 2021 Facebook outage[14] was caused,

[12] https://www.csoonline.com/article/3613571/the-solarwinds-hack-timeline-who-knew-what-and-when.html

[13] https://us-cert.cisa.gov/kaseya-ransomware-attack

[14] https://en.wikipedia.org/wiki/2021_Facebook_outage

not by an attack, but by a misconfiguration of Facebook's Domain Name servers.

Almost every discussion on the topic of decentralization and the Internet revolves around this same theme: the surveillance economy and entrenched interests of Big Tech are "recentralizing" the Internet—and to counter this, we need to restore the facets of Internet design that minimize centralized control points and maximize distributed authority.

How this principle applies to the ToIP stack

For a trust layer for the Internet, decentralization is even more essential than for the Internet itself. In fact the End-to-End Principle demands it: if trust is to be end-to-end, directly between parties, it cannot require a dependence on intermediaries (other than for the trans-mission of secure messages, just as the Internet relies on routers to transmit data packets).

But how exactly does one design a system to be decentralized? In 2019, the Sovrin Foundation, the governing authority for a ToIP Layer 1 public utility, defined Decentralization by Design as one of 12 core principles of the Sovrin Governance Framework. They listed eight component principles, summarized in the following table:

Diffuse Trust	Shall not concentrate power in any single individual, organization, jurisdiction, industry sector, or other special interest to the detriment of the network as a whole.
Web of Trust	Shall be designed to not favor any single root of trust.
Censorship Resistance	Shall be designed to resist censorship of any Entity while remaining compliant with all applicable laws.
High Availability	Shall be designed and implemented to maximize availability.
No Single Point of Failure	Shall be designed and implemented not to have any single point of failure.
Regenerative	Shall be designed so that failed components can quickly and easily be replaced by other components.
Distributive	Shall be designed and implemented such that authority is vested, functions performed, and resources used by the smallest or most local part of the community includes all relevant and affected parties.
Innovation at the Edge	Shall encourage innovation to take places at the edges of the network among members of the community most directly involved or impacted.

There are many other points of view on how to best ensure the decentralization of a system. For example, the W3C Decentralized Identifiers (DID) Working Group tackled the challenge of trying to define what "decentralization" means in the context of evaluating a particular DID method by publishing the DID Method Rubric V1.0. It "presents a set of criteria which an evaluator can apply to any DID Method based on the use cases most relevant to them".

The eSSIF-Lab project, a European Horizon 2020 project to further interoperability and standardization of SSI infrastructure, publishes another perspective of decentralization as part of its eSSIF-Lab Glossary cited earlier. It defines the term "authority" as follows:

> *An Authority is a party of which certain decisions, ideas, rules etc. are followed by other parties. We distinguish between two kinds of authority:*
>
> - ***centralized authority**, also known as the power or right to give orders, make decisions that other parties must follow, and enforce obedience. This kind of authority ignores the natural autonomy of other parties.*
> - ***decentralized authority**, also known as the power or right that is freely endowed by other parties to the authority, to make decisions, phrase ideas, set rules, etc., which these parties will adopt and follow because they think it is in their own interest to do so.*

In other words, decentralized authority is not enforced by rule of law, but via governance that participants willingly follow for their mutual benefit. Phil Windley, author of the O'Reilly book Digital Identity and co-founder of the Internet Identity Workshop, emphasizes the unique role of governance in decentralization in his 2018 essay on the topic:

> *One of the ironies of decentralized systems is that they **require better governance than most centralized systems**. Centralized systems are often governed in an ad hoc way because the central point of control can easily tell all participants what to do. Decentralized systems, on the other hand, must coordinate across multiple parties, all acting independently in their own self-interest. This means that the rules of engagement and interaction must be spelled out and agreed to ahead of time, with incentives, disincentives, consequences, processes, and procedures made clear.*

Given the inherent relationship between decentralization and authority, Ashwin Mathew of the School of Information at the University of California makes a particularly relevant design recommendation in his article The Myth of the Decentralized Internet published in the October 2015 special issue of Internet Policy Review called Doing Internet Governance:

> *Public debates about the internet often decry the present, in which the freedoms offered by the internet are under attack by nation states and monopolistic corporations. These debates invoke a past in which the internet is said to have been freer, and more decentralised, and imagine a possible future in which freedoms might be ensured once more through the development of decentralised technologies, immune to control.*
>
> *However, as I have shown, if the internet encapsulated certain freedoms in the past, it was by no means as a consequence of intrinsically decentralised technology. All systems have centres of power, whether as global administrative functions (such as centralized administrative control in the ARPANET, or the NSFNET PRDB) or as concentrations of power in topology (such as the more central networks which control routing across specific geographical scales).*

There are other realities that balance the amount of decentralization that is possible within or across systems. For example, within the ToIP stack, there can still be points of centralization involved for where service providers are supporting specific use cases such as:

- Credential issuer services (especially issuing on behalf of another party or issuing secondary credentials)
- Cloud wallet / web wallet / custodial wallet hosting services
- Credential verifier services that embed business logic into the use of verifiable credentials to pursue specific commercial outcomes.

What is important is that these service providers should be limited to involvement to the specific digital trust ecosystems they are serving and not be allowed to constrain an overall evolution toward a decentralized ecosystem-of-ecosystems. In his article, Mr. Mathew goes on to make a very important point: that focusing on technology alone as the solution to such centres of power is a mistake (emphasis added):

> *The danger inherent in imagining and designing future internet technologies as though they are decentralised is that the inevitable centres of control required for **governance** will go unremarked upon, and in doing so will become susceptible to political capture by the very powers that these technologies seek to evade. **It is critical to anticipate and design for the functions and accountability of centres of power in internet technologies.***

Given this advice, it should not be surprising that the Decentralization by Design and Default principle applies to every layer from top to bottom of the ToIP stack.

Layer	Relevance	Explanation
4	★★★★★	Decentralization at Layer 4 means any digital trust ecosystem of any size should be able to publish its own governance framework and interact as a peer with any other ecosystem.
3	★★★★★	Layer 3 should deploy the W3C Verifiable Credentials standard such that anyone can issue verifiable credentials, anyone can hold them, and anyone can verify them in a fully decentralized manner.
2	★★★★★	Decentralized peer-to-peer secure messaging is the essential function of Layer 2 of the stack.
1	★★★★★	"Decentralization" is part of the name of one key W3C open standard—Decentralized Identifiers (DIDs)—that is core to interoperability at Layer 1 of the ToIP stack.

#5: Cryptographic Verifiability

As part of digital trust, messages and data structures exchanged between parties should be verifiable as authentic using standard cryptographic algorithms and protocols.

For various reasons discussed throughout this paper, the original design of the Internet protocol suite did not include mechanisms for establishing and maintaining digital trust relationships. The Wikipedia article on Internet security summarizes it this way:

> *The Internet is an inherently insecure channel for information exchange, with high risk of intrusion or fraud, such as phishing, online viruses, trojans, ransomware and worms.*

Fixing these problems required developing other protocols that either replaced or were layered over the TCP/IP protocols to add the necessary security features. Examples include:

- IPsec for the network layer security.
- DNSSec for domain name layer security.
- Secure Sockets Layer (SSL), succeeded by Transport Layer Security (TLS) for web layer security.
- Opportunistic TLS, DMARC, and Pretty Good Privacy (PGP) for email.

Many of these protocols have had significant success in the market. Some—like SSL/TLS— have even become de facto market requirements for specific applications such as browsing.

The challenge is that all of these protocols are "overlays" designed to solve the security issues for a specific type of Internet communication. In addition, most of them rely on X.509 PKI infrastructure, which itself has several barriers to ubiquitous adoption. In short, none of them are poised to achieve the level of adoption necessary to become the default for all Internet communications.

The principle of cryptographic verifiability maintains that for strong digital trust relationships, all parties need to be able to verify the authenticity of *any message or data structure* exchanged between them. This breaks down into two requirements:

1. Verifying that the party identified as the sender is the authentic sender and is not being impersonated by another party.
2. Verifying that the message or data structure has not been tampered with en route.

NOTE: The second requirement is proof of the **integrity** of a message. This is separate from (but closely related to) the **confidentiality** of the message. See Principle #6: Confidentiality by Design and Default.

How this principle applies to the ToIP stack

This is the first of our principles that did not also apply to the original design of the Internet. From the standpoint of the ToIP stack, cryptographic verifiability is similar to decentralization (Principle #4)—**it needs to be baked into every layer**. How precisely this applies at each layer depends on the specific protocols as summarized in our table below. But the overall rule is that *all exchanges facilitated by the ToIP stack should be cryptographically verifiable*.

Layer	Relevance	Explanation
4	★★★★★	All ToIP governance frameworks at Layer 4 (and every layer) should be identified with a DID and digitally signed by the corresponding private key. All ToIP trust registries should issue cryptographically verifiable responses to queries.
3	★★★★★	All verifiable credentials and other Layer 3 data exchange payloads should be digitally signed with the private key of the corresponding DID.
2	★★★★★	All peer-to-peer messages at Layer 2 should use DID-to-DID authenticated encryption for cryptographic verifiability. ToIP digital wallets should take advantage of secure mobile enclaves, TPMs, HSMs, and other secure local key storage.
1	★★★★★	The entire purpose of Layer 1 is for public key utilities such as blockchains and distributed file systems to serve as strong anchors for cryptographically verifiable public DIDs.

#6: Confidentiality by Design and Default

Parties communicating over ToIP protocols should expect communications to be secure, private, and confidential without any special thought or action required on their part.

In the initial design of the Internet, confidentiality of Internet communications was not a priority. It was enough of a challenge to create a global internetwork that could transmit data between any two local networks and survive a nuclear war. However, as described at length in a five-part 2015 Washington Post series entitled Net of Insecurity: A Flaw in the Design, this lack of attention to security and privacy resulted in a set of vulnerabilities whose dangers grew as fast as the Internet did.

Since then, two sets of design principles have become widely established in the industry:

1. Privacy by Design is an approach to engineering privacy directly into software, services, and systems. The seven core principles were initially developed by Ann Cavoukian in 1995, formalized as a design framework in 2009, and adopted by the International Assembly of Privacy Commissioners and Data Protection Authorities in 2010.
2. Security by Design is the corresponding approach to security engineering, i.e., considering security requirements from the very start, then building them into systems at every layer following well-known patterns for authentication, authorization, confidentiality, data integrity, accountability, availability, safety, and non-repudiation.

As the popularity of these approaches has grown, they have frequently been extended from “...by Design” to “...by Design **and Default**”. The importance of the latter is articulated in the second principle of Privacy by Design: “Privacy as the default setting”.

> *Privacy by design seeks to deliver the maximum degree of privacy by ensuring that personal data are automatically protected in any given IT system or business practice. If an individual does nothing, their privacy still remains intact. No action is required on the part of the individual to protect their privacy — it is built into the system, by default.*

The same applies to security: by making it the default for all actions taken by a user within a system, users do not need to acquire special knowledge or take special actions to maintain security. It is just “built in”.

The principle of *confidentiality by design and default* means applying BOTH privacy by design and default *and* security by design and default. In systems that follow this principle, a user can expect confidentiality of their private messages and data at all times without taking any special action. It is the engineering equivalent of applying the same duty of care a person expects from a doctor, lawyer, accountant, or other professional who has a fiduciary duty to protect a client’s information.

How this principle applies to the ToIP stack

Like Principle #5, this principle has very clear implications across all four layers of the stack. Furthermore, it applies equally to both the technology and governance halves of the stack for one simple reason: *machines can only protect the confidentiality of information until it reaches a human*. At that point only humans can continue to keep it secure and private.

Layer	Relevance	Explanation
4	★★★★★	Layer 4 ecosystem governance frameworks should bestow an expectation and a duty of confidentiality upon all parties dealing with private data. Exceptions in the public interest should be defined as explicitly and narrowly as possible (ideally via public legislation) and have clear requirements for accountability.
3	★★★★★	Data formats, signatures, and protocols used for exchange of VCs and other payloads at Layer 3 should ensure confidentiality and support selective release and minimal disclosure.
2	★★★★★	DID-to-DID connections between peers should remain confidential between the parties and not require a third party to be involved unless all peers agree. Additionally, mediator agents in Layer 2 peer-to-peer messaging should not leak metadata.
1	★★★★★	Private data should not be written to a Layer 1 public utility, and transactions with a public utility leak should not leak private data about the transaction author.

#7: Keys at the Edge

To maximize security, privacy, and confidentiality, cryptographic private keys should be stored at the edges of the network, not on intermediate nodes.

This principle builds directly on several of the preceding principles: end-to-end communications (Principle #1), cryptographic verifiability (Principle #5), and confidentiality of all messages and data exchanges (Principle #6). To be consistent with these, the only logical place end-users of ToIP infrastructure can safely hold and use their private keys is *at the endpoints*. If they had to rely on intermediate nodes to hold and use their private keys for encryption, decryption, or message signing, it would mean—*by definition*—that confidentiality of communications cannot be maintained end-to-end because third parties (known or unknown) would have to be brought into the trust relationship.

This principle reflects the inherent tension between two prevailing models for digital wallets:

1. **Edge wallets**—also called **mobile wallets** or **non-custodial wallets**—operate on local devices such as smartphones, tablets, laptops, smart TVs, smart cars, and so on. Such devices are not expected to be always online and may have uneven, intermittent connectivity.
2. **Cloud wallets**—also called **hosted wallets** or **custodial wallets**—operate in the cloud and are generally intended to be online and accessible 24/7.

IMPORTANT: There is a critical distinction between cloud wallet hosted by a third party but containing private keys intended to be under the exclusive control of the end-user, and a third party acting as a guardian for the end-user under a guardianship arrangement. In the latter case, the guardian performs actions with the guardian's own private keys (typically using guardianship credentials of some kind).

The distinction between edge wallets and cloud wallets is different from the distinction between **hot wallets** and **cold wallets**. As the Wikipedia article on digital wallets explains:

> *Hot wallets are connected to the internet while cold wallets are not. Most digital wallet holders hold both a hot wallet and a cold wallet. Hot wallets are most often used to make quick payments, while a cold wallet is generally used for storing and holding your money and has no connection to the internet.*

For the purposes of the *Keys at the Edge* principle, it does not matter whether a digital wallet is hot or cold. What matters is **who controls the private keys where**. Are the private keys under the direct control of the controlling party acting at the edge of the network? Or does some intermediary in the network ultimately control the private keys?

If the former, the keys are "at the edge". If the latter, the controlling party must rely on a trust relationship with a third party.

How this principle applies to the ToIP stack

This principle does not prohibit the use of cloud wallets or custodial services for private keys. However, it does mean:

1. **ToIP architecture and governance frameworks should always give parties the choice to directly control their private keys using edge wallet(s) of their choice.** A digital trust ecosystem whose architecture or governance *requires* the use of a third party does not meet this principle.
2. **Parties should also have the choice of using a cloud wallet if they prefer**. Some-times this is simply the most convenient option; in other cases, it is the only option. In either case, the party still directly controls their own private keys. If a cloud wallet is used exclusively, extra caution should be taken with security and authentication.
3. **Parties should have the choice of entrusting control to a guardian of some kind**—a third party who controls the private keys in order to take actions on behalf of the first party. In this case the applicable governance framework should ensure that: a) the third party has a fiduciary duty to act in the interests of the first party, and b) the first party has the right to change guardians or switch to using an edge wallet.

As with #3: The Hourglass Model—this principle applies primarily to ToIP Layer 2.

Layer	Relevance	Explanation
4	★★★	Ecosystem governance frameworks need to consider key hygiene and other control requirements at all levels. Ecosystem participants also need to know that keys are associated with the participants that they expect (e.g. via trust registries).
3	★★★	Verifiers need proof that digital wallets are secure. Key management policies are especially important for cloud wallets and custodians.
2	★★★★★	Storing, controlling, using, and protecting private keys is the primary function of digital wallets at Layer 2.
1	★★★	Layer 1 public key utilities need very secure key management in order to maintain public trust, however like bank vaults (vs. personal wallets), they are also designed for this function.

Part Two: Human Network Architecture ("Wet Code") Principles

Our second set of principles pertain to "wet code"—the policies, procedures, rules, and laws that people write down and follow to create healthy sustainable societies and economies. These principles describe fundamental truths about how trust relationships operate between humans—either individually or in groups. They reflect the fact that **technology by itself is never sufficient to produce trust** because trust is a psychological belief of humans.

In digital trust infrastructure, this "wet code" is formally known as a **governance framework** or **trust framework**.[15] It consists of the policies, processes and practices that the members of a trust community agree to follow in order to achieve their mutual trust objectives. The fact that the ToIP stack has two halves—one representing technology and one representing governance—reflects how tightly integrated these two dimensions are in achieving the overall goal of the ToIP Foundation:

To develop tools and specifications to help communities of any size use digital networks to build and strengthen trust between participants.

To highlight the difference between "dry code" (cryptographic logic) and "wet code" (human trust) principles, look at the two images in Figure 12 below.[16]

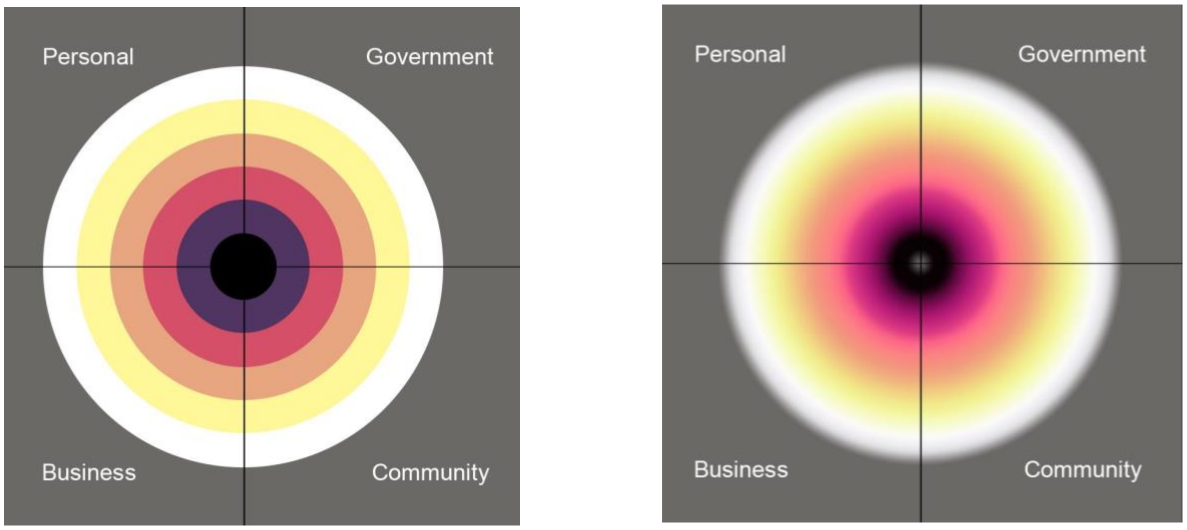


Figure 12. A visualization of the differences between cryptographic logic and human trust.

[15] The term "trust framework" is most often used in the context of digital identity systems. The term "governance framework" is broader—it includes trust frameworks as well as other types of policies and procedures necessary for governance of digital trust communities.

[16] Note that as we use more machine learning (ML) to assess trust, dry and wet coding start to converge.

The image on the left is a visualization of how cryptographic logic is applied: deterministicly. If a condition is met, the machine executes a function; if the condition is not met, the machine does not execute the function. This allows us to encode precise rules describing the inputs, output, and thresholds necessary to produce certain outcomes.

The image on the right illustrates human trust. It is much fuzzier. It is based on an unlimited number of conditions, some of which we cannot define verbally or objectively (or sometimes even consciously). Human trust can be vague and subjective yet is adaptable enough to consider conditions that are not yet known. It can also change over time—trust can be built up over multiple interactions and then eroded or destroyed by a single action. Thus human-centered trust principles should enable us to address such important governance aspects as inclusion, diversity, equitability, and redemption.

A single trust relationship can progress through various levels of trust depending on the risk assessments of both parties. Take the example of Alice, the CEO of Acme Technologies, a minority-owned business that won a contract with a local government agency. Here is Alice's journey to sign the contract:

1. Alice arrives at the government office building, enters through the main doors, and needs to pass through a physical security check that includes confirming her citizen identity by showing a driving license or passport.
2. Alice is escorted to the agency office and is introduced to the local government official and the legal officer. They exchange business cards, sharing their professional identities with each other.
3. Next they go to a conference room to review the terms of the contract. Once everything is ready, Alice signs the contract as the CEO of Acme, and the government legal officer does a verification of:
    a. Alice's personal credentials for her legal identity;
    b. Acme Technologies incorporation papers and business licenses to verify its identity and status as a registered legal entity; and
    c. Alice's appointment as CEO to verify she is authorized to execute the contract.
4. Alice's signature is notarized by two witnesses who also provide their notary credentials.

In Alice's short walk, she has progressed through a sequence of interactions that require increasing levels of trust. The more risk involved, the more rules and regulations are needed. Entering the building is relatively low-risk, but falsely signing a government contract carries much higher risk—so more must be done to protect it.

Similarly, every digital transaction should be backed by governance that spells out the business, legal, and technical policies that apply to the transaction. In Part 2, we will cover seven more design principles that apply what Nick Szabo called "wet code", i.e., the written rules, policies, laws, and regulations that are "executed" in the human brain to produce the actions we take to achieve trust.

#8: Trust is Human

Trust is a psychological belief held by people who individually or collectively need to act on that belief in order to make risk decisions.

Although "trust" as an abstract concept can be exceedingly hard to define, for our purposes we can focus on two aspects of how trust works:

1. **Trust is fundamentally a belief of either a single person or a group of people acting together.** As explained in our Terminology section, the eSSIF-Lab Glossary term for both options is a party. Trust is part of the knowledge of the party.
2. **The purpose of this belief is to help a party make a decision relative to a specific action.** In other words, while trust in the abstract is a good subject for philosophers and ethicists, in the context of the ToIP stack what matters is enabling parties to make trust decisions about a specific interaction or transaction of some kind.

There are hundreds of academic papers on the factors that influence trust decisions. A good example is a 1995 paper called An Integrated Model of Organizational Trust[17] that cites three key "factors of perceived trustworthiness" as shown in this diagram from the paper:

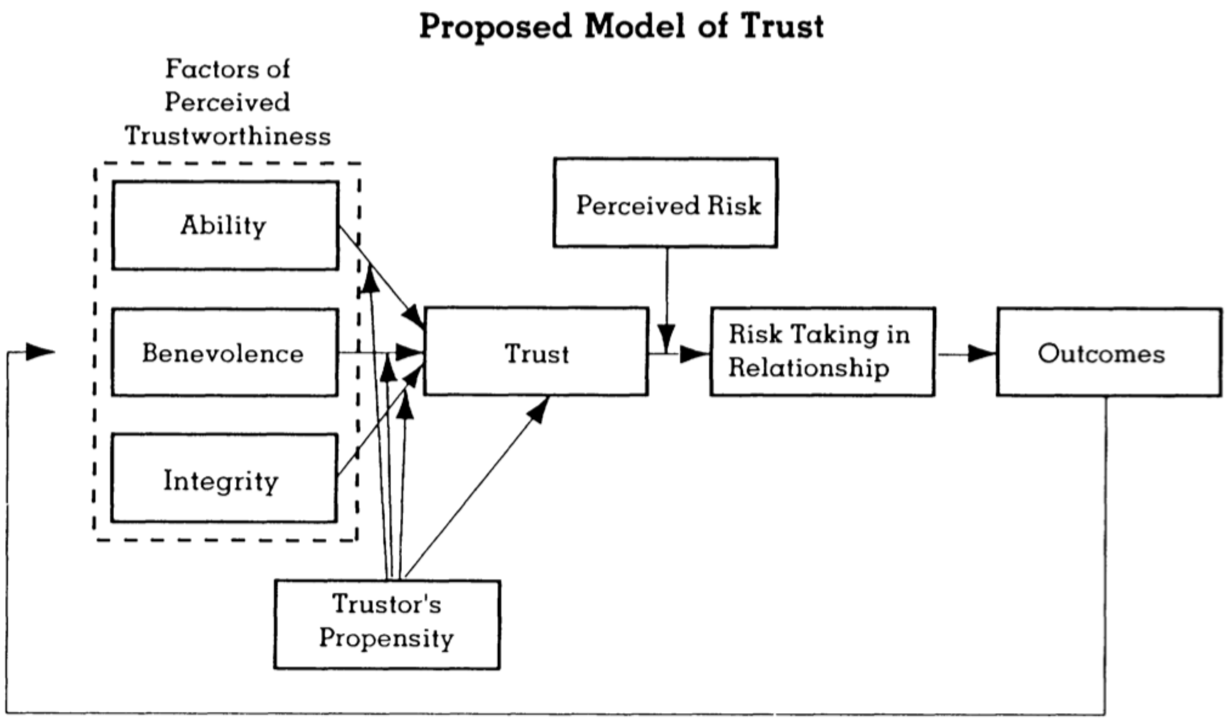


Figure 13. A model of organizational trust

[17] Mayer, Roger C., James H. Davis, and F. David Schoorman. "An Integrative Model of Organizational Trust." *The Academy of Management Review* 20, no. 3 (1995): 709–34. https://doi.org/10.2307/258792.

In a scenario where A wants to make a trust decision about B, the paper cites three key factors—*ability*, *benevolence*, and *integrity*—that together will determine the perceived trustworthiness of B to A. In addition, A will have to have a certain degree of *propensity* to trust. The combination of perceived trustworthiness of B to A and A's propensity to trust will determine if A makes a decision to trust B.

A more recent paper about trust as a human belief that is highly relevant to the design of the ToIP stack is Facets of Trustworthiness in Digital Identity Systems published in May 2021 by the Alan Turing Institute. To quote from the executive summary:

> *This work presents the different pillars of trustworthiness for digital identity management systems. We specifically consider the **trustworthiness** of these systems, rather than whether or not such systems are **trusted**. The latter is an important area that informs our work but is not the subject of this work. Whereas **trusted** can be defined as a belief in the integrity, ability or character of an entity, **trustworthiness** of an entity regards the extent to which it is deserving of trust.*
>
> *The emphasis in this work is placed on the various features and mechanisms for each of the dimensions that can be used to capture the trustworthiness assurance levels (TAL) of identity-related functions. This presentation outlines the preliminary assurance features and mechanisms in each of the pillars of **security, privacy, ethics, robustness, reliability and resiliency** in digital identity systems.*

How this principle applies to the ToIP stack

The primary implication of this principle is wonderfully summed up by the first and fifth of the nine design principles of the U.K. National Health Service: "*Put people at the heart of everything you do*", and "*Design for trust*".

From a ToIP architecture standpoint, the implication is straightforward: **every layer should be designed to enable parties to make trust decisions**. What kind of trust decision—about what kinds of parties and what kinds of actions—depends on the layer as summarized here:

Layer	Relevance	Explanation
4	★★★★★	Layer 4 should be designed to enable parties to make trust decisions about using applications within one or more **digital trust ecosystems**.
3	★★★★★	Layer 3 should be designed to enable parties to make trust decisions about interacting or transacting with **other parties** (or in some cases with **physical or logical things**).
2	★★★★★	Layer 2 should be designed to enable parties to make trust decisions about the **digital wallets and digital agents** they can rely on for ToIP communications—at the edge or in the cloud.
1	★★★★★	Layer 1 should be designed to enable parties to make trust decisions about the **public key utilities** they will rely on for DIDs, DID documents, and other cryptographic primitives.

#9: Trust is Relational

Trust is a relationship between a **subject**—a person or a group of people—and an **object**—which can be anything about which the subject needs to make a trust decision.

A simple way to illustrate this principle is the one-dimensional directed graph in Figure 14.

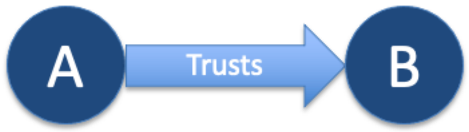


Figure 14. Trust is always between a subject and an object

While this principle may seem obvious, from a design perspective it has two lessons:

1. Trust can only be described, modeled, and acted upon in the context of a relationship between a subject—the party that needs to make a trust decision—and an object.
2. The potential objects of a trust relationship can be anyone or anything about which the subject might need to make a trust decision.

To illustrate the second point, Figure 15 categorizes the objects of trust relationships into two broad categories—parties (which can have legal standing)—and *things* (which do not have legal standing).

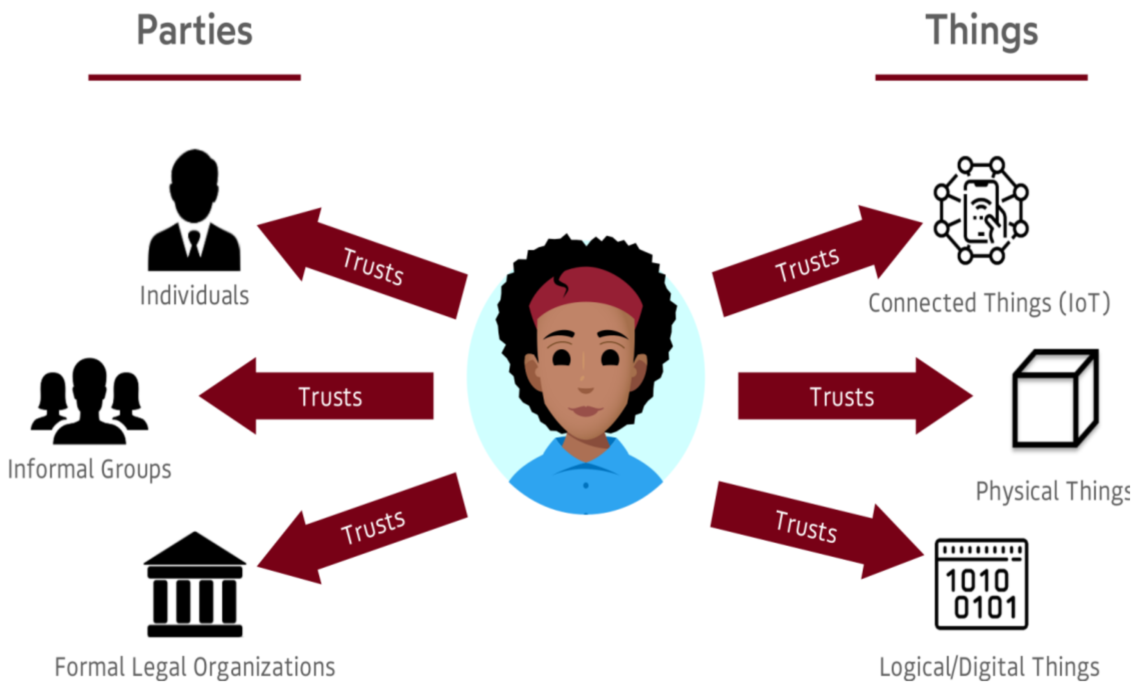


Figure 15. The object of a trust relationship can be anything about which the subject needs to make a trust decision

How this principle applies to the ToIP stack

The key implication of this principle is that the ToIP stack should be designed **to support any type of trust decision**. It is not just about building trust between people, groups, and organizations (the left half of Figure 9.2). It's also about building trust with the **things** in our lives (the right half of Figure 9.2). This includes:

1. **Connected things** in the Internet of Things (IoT): vehicles, appliances, drones, sensors, etc.
2. **Unconnected physical things**: food products, drugs, tools, buildings—any non-digital product that has health, safety, or other ethical considerations.
3. **Logical/digital things:** websites, software programs, messages, documents, bots—any digital artifact that may be an attack vector.

This category is particularly important when it comes to artificial intelligence (AI) bots, programs, or algorithms. These raise the question of who or what is actually "making a decision" or "taking an action". Until our legal systems begin to recognize AI programs as legal actors, some person or organization will be held responsible and accountable, and this is where the ultimate trust relationship must reside.

Layer	Relevance	Explanation
4	★★★	Layer 4 is focused on creating entire networks of trust relationships powered by applications operating in one or more digital trust ecosystems.
3	★★★★★	Layer 3 is focused on the establishment of new trust relationships using verifiable credentials which leverage existing trust relationships for transitive trust. See Principle #13.
2	★★★★★	Layer 2 is the layer at which direct peer-to-peer DID-to-DID trust relationships are formed and managed. See Principle #3.
1	★	Layer 1 does not play a direct role in most trust relationships— the trust relationship at this layer is directly between the user of a public key utility and the utility itself.

#10: Trust is Directional

While many trust relationships are bi-directional, each direction is independent. In other words, if A trusts B, it does not mean B trusts A.

This principle may also seem obvious, yet it still has important lessons for the design of decentralized digital trust infrastructure. Of course, this principle only applies when both participants in the trust relationship are parties capable of making trust decisions. The subject party is the **trustor** and the object party is the **trustee**. Figure 16 illustrates that a trustee/trustor relationship may NOT be bidirectional. Each party makes its own trust decision.

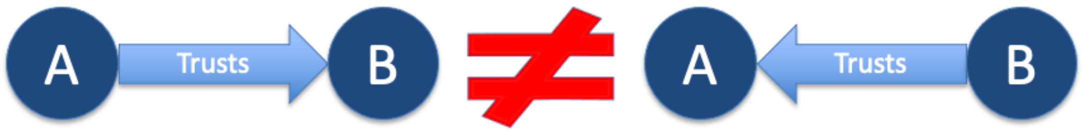


Figure 16. A single trust relationship is always unidirectional. A trust relationship MAY be bidirectional, but if so, it is composed of two unidirectional trust relationships

How this principle applies to the ToIP stack

The lesson here is simple: ToIP architecture should **never** assume bidirectional trust, but always treat each direction in a trust relationship separately. The goal should be to make it as easy and safe as possibly for each party to make its own independent trust decision.

Layer	Relevance	Explanation
4	★★★★★	Layer 4 should optimise the ability for digital trust ecosystems to establish **mutual trust relationships** as this will maximize benefits to all participants in both ecosystems.
3	★★★★★	At Layer 3 this principle recommends bidirectional exchange of verifiable credentials for **mutual verification**—a practice that is very difficult to implement in today's Web.
2	★★★★★	This principle is critically important at Layer 2 because merely establishing a peer-to-peer DID-to-DID connection should not imply bidirectional trust without other evidence.
1		Layer 1 is not involved in bidirectional trust relationships, only in anchoring DIDs for cryptographic verifiability in higher layers.

#11: Trust is Contextual

A trust relationship exists in a specific context, and it should not be assumed outside of that context. In other words, if A trusts B in context X, it does not mean A trusts B in context Y.

This starts to reveal the more complex dimensions of trust. It suggests that in our previous simple trust relationship diagrams, we left out an important component—the **context** in which any trust relationship exists. This is shown as the outer box in Figure 17.

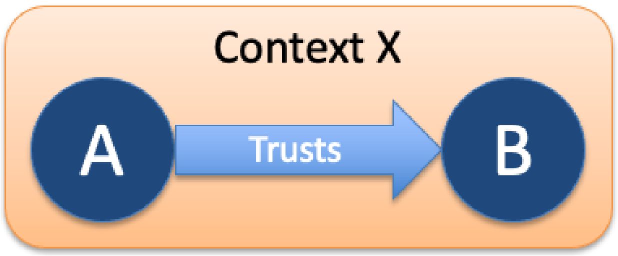


Figure 17. A trust relationship always exists in a specific context

There are countless examples of how a trust relationship is contextual:

1. A parent trusts a student to be a babysitter but not a mechanic.
2. A car rental company trusts a driver to rent a particular car.
3. A patient trusts a doctor to perform a particular operation.
4. A university trusts a professor to teach a particular class.
5. A company trusts a contractor to fix a particular type of machine.
6. A consumer trusts a company to produce a particular type of product.

How this principle applies to the ToIP stack

As simple and intuitive as it seems, this principle has two very important implications:

1. **ToIP architecture should enable context to be modeled as an integral component of trust relationships.** This can be exceedingly important in the design of everything from user interfaces to policy engines. Supplying the appropriate context can make it much easier for parties to make more confident trust decisions.
2. **ToIP architecture should help parties guard against taking trust out of context.** Most phishing attempts do just that: they try to mislead the target about the true context of a message or a link. Surfacing the true context explicitly can often defeat that type of attack.

Because of these factors, context plays a very significant role at every layer of the ToIP stack except Layer 1.

Layer	Relevance	Explanation
4	★★★★★	Layer 4 is all about establishing digital trust ecosystems as **large-scale verifiable contexts** that simplify trust decisions for *everyone* operating in that context.
3	★★★★★	At Layer 3 a verifiable credential can be issued in one explicit context and then verified in another explicit context, helping holders to make much easier and more confident trust decisions.
2	★★★★★	Layer 2 should be designed to support **context-specific peer-to-peer trust relationships**.
1	★	Layer 1 is usually not involved in establishing context except in the case of using a public key utility whose use is restricted to a specific context, e.g., a nation state or an industry.

#12: Trust has Limits

In the human perception of trust, every trust decision has a trigger point along a continuum that ends at a limit point. The limit point where risk exceeds reward.

Trust is almost never unlimited. As shown in Figure 18, party A may trust party B up to a certain limit that depends on the context. For example, a parent may trust a student to be a babysitter for an afternoon—but not overnight. Or a company may trust a contractor to fix a broken copier, but not to replace it with a new copier. Or a bank may trust a customer up to a certain credit limit but not to a higher one.

Different parties make different trust decisions in different ways, applying different factors, all depending on their experience and perceptions. Some parties need to consider many different pieces of **trust evidence** to make a trust decision; others operate more quickly and intuitively with much less evidence. In Figure 18 we illustrate that, in a given context, a trust decision has a range of trigger points along a continuum. When the evidence weighed by party A (the subject) passes the trigger point but is still inside the limit with regard to party B (the object), A will decide to trust B. When the evidence pushes the trigger point beyond the limit L_B, A will decide not to trust B (at least for this decision in this context).

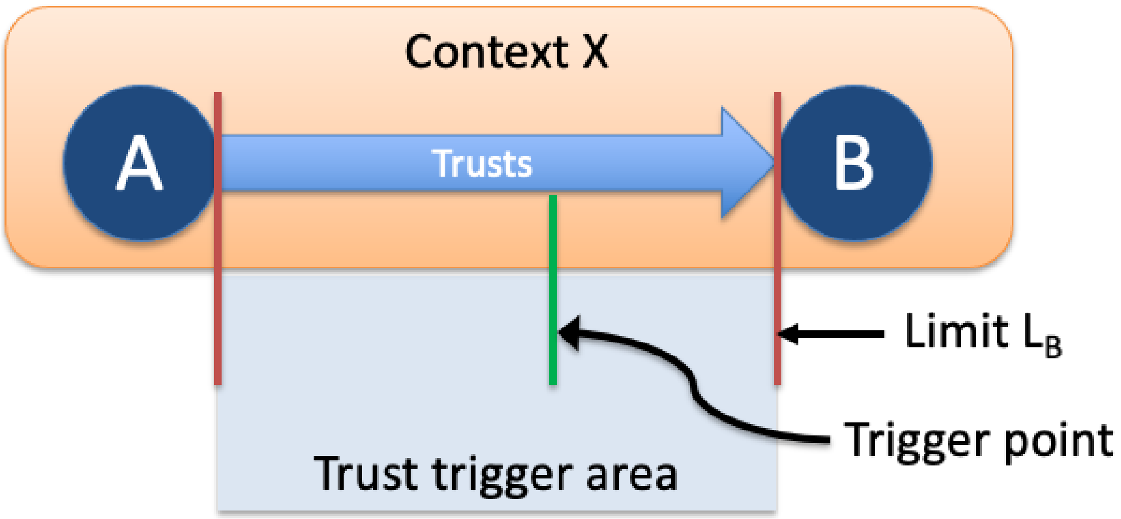


Figure 18. A trust relationship has a range of trigger points for making a trust decision up to a limit determined by the subject

Of course, the limit may vary with the object of the trust decision. As shown in Figure 19, parent A may only trust student B to babysit for an afternoon, but trust neighbor C to babysit overnight, and trust cousin D to babysit for a week. If A is an equipment rental company, it could require different levels of insurance for customers B, C, and D depending on their rental history, business volume, project type, and so on.

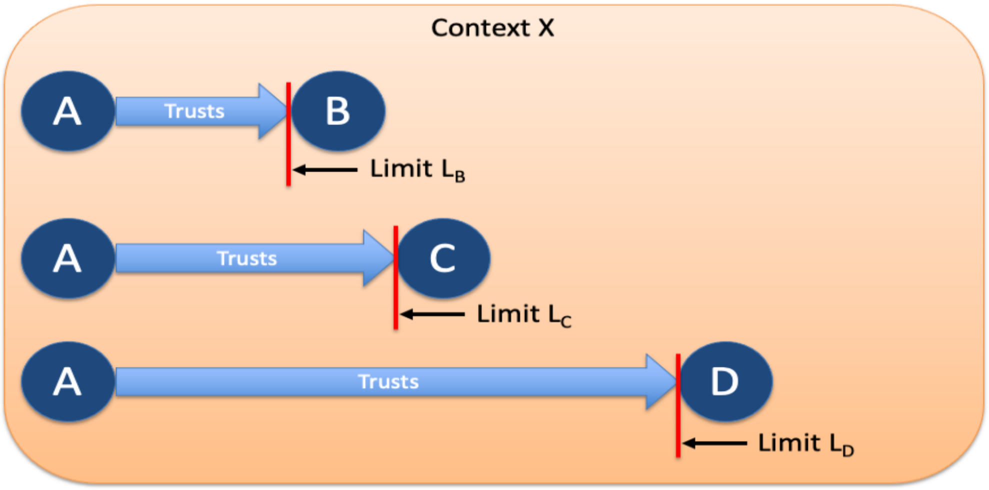


Figure 19. A can have different limits of trust in B, C, and D in the same context X

Risk assessment is a science. When a large enough population needs to make a certain type of trust decision, the results will lie along a bell curve as shown in Figure 20. As parties receive more information, the number of parties making the decision will increase to a point where the vast majority reach **reasonable assurance**. The far right-hand side of the bell curve is where parties have 100% of the information they need to reach **absolute assurance**. The margin between these two points is the **window of error**.

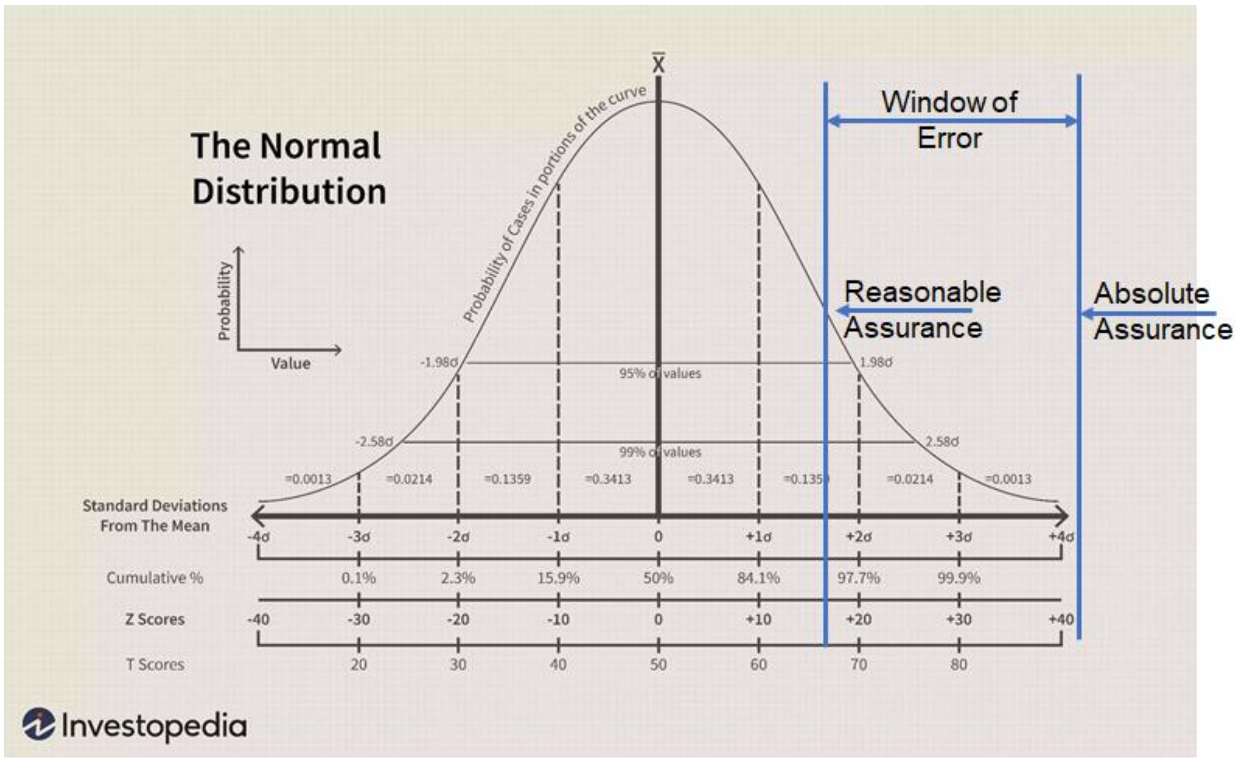


Figure 20. The "Bell Curve" or Normal Distribution applied to trust decisions[18]

[18] Image by Julie Bang © Investopedia 2019

How this principle applies to the ToIP stack

One of the primary purposes of governance frameworks in ToIP architecture is to **establish consistent and transparent requirements for trust evidence** for the members of a digital trust community. What those particular requirements are depends on the trust community and the layer of the ToIP stack. For example, the trust evidence requirements for a ToIP Layer 1 public key utility operated by the provinces of a nation state will be very different from the requirements for a ToIP Layer 4 digital trust ecosystem for patient medical records.

Secondly, **ToIP architecture should target reasonable assurance, not absolute assurance.** Trust communities do not operate in a utopian setting. They operate in the real world where cost-benefit analysis and risk assessment must drive decision making. Therefore under-standing, accepting, and planning for a window of error between reasonable assurance and absolute assurance should be a standard best practice.

Thirdly, **ToIP architecture should facilitate defining levels of assurance (LoA) for different types of trust evidence** as necessary to satisfy the need for different participants to make their own trust decisions based on their own trust limits. Common examples at ToIP Layer 3 include defining levels of identity proofing and identity authentication such as those in NIST 800-63. However, ToIP architecture should not stop there; governance frameworks should apply LoA to any type of trust evidence where it will assist verifiers in making better trust decisions.

Lastly, **limits apply at all levels of the ToIP stack**. Overall trust in an interaction is defined by the combination of these limits. For example, if the trust evidence is within the subject's limits for all layers except for a particular ToIP Layer 1 public key utility, or a particular ToIP Layer 2 digital wallet, or a particular ToIP Layer 3 credential, or a particular ToIP Layer 4 trust registry, then the interaction will not proceed. Parties need to be able to make trust decisions at every layer.

Layer	Relevance	Explanation
4	★★★★★	Layer 4 governance frameworks establish the ecosystem-wide rules for the trust evidence required to meet the needs of verifiers throughout a digital trust ecosystem.
3	★★★★★	Layer 3 is designed specifically for the exchange of verifiable credentials that give verifiers the trust evidence they need to make trust decisions about holders.
2	★★★★★	At Layer 2, trust limits apply to trust decisions about the digital wallets and agents that participants need to use for their higher-layer interactions.
1	★★★★★	The trust evidence requirements in ToIP Layer 1 governance frameworks are supremely important because they underpin the cryptographic verifiability of all higher-layer trust decisions.

#13: Trust can be Transitive

If a first party trusts a second party who in turn trusts a third party in the same context, then the first party can have some degree of trust in the third party in that context.

This is the design principle that introduces the “trust triangle” at the heart of the ToIP stack. The basic geometry is shown in Figure 21. On the left, A has a direct trust relationship with B in context X, and B has a direct trust relationship with C in the same context. A can then have some amount of transitive trust in C in the same context as shown on the right.[19]

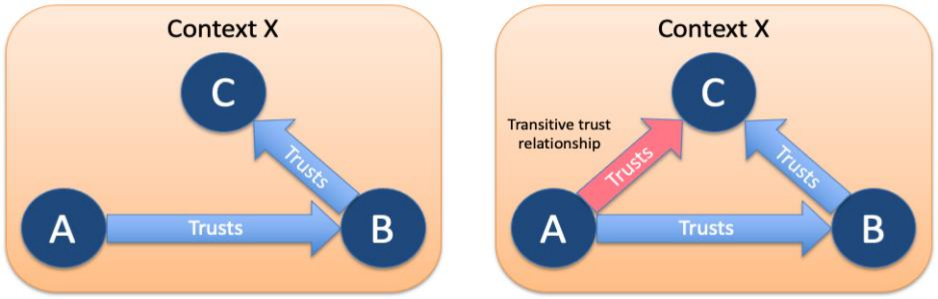


Figure 21. The “trust triangle”. If A trusts B within context X and B trusts C within context X (left), then A can have transitive trust in C within context X (right)

Like many of the other “wet code” principles, transitive trust is a natural part of human experience. We “learn” trust from others. For example, if you (A) trust your friend Bob (B) to refer a babysitter and Bob recommends Cathy (C), then you are likely to trust Cathy to be a good babysitter. Some of your trust in Bob has been transferred to Cathy by referral rather than gained through direct experience. This also applies in an organizational context, e.g., if Acme (A) trusts Bonfix (B) to fix copy machines, and Bonfix (B) trusts CopyCity (C) with regard to copy machines, then Acme is likely to trust CopyCity to fix its copy machines—but maybe not to fix its delivery trucks.

A trust relationship that generated through (or incorporates) transitive trust is still subject to all the other “wet code” principles. In other words:

- **It is not automatic**. It is still a trust decision made by the subject party.
- **It is directional**. If A trusts B and B trusts C, that does not give C a reason to trust A.
- **It is still context-bound**. Transitive trust in a babysitter may not have any bearing on trusting that same person as an auto mechanic.
- **It has limits**. In fact, because a transitive trust relationship originates indirectly via another party, it usually starts with a lower limit than a direct trust relationship.
- **It is only one factor** in making a trust decision. The subject party may also consider many other types of trust evidence.

[19] Transitive trust is neither automatic nor absolute. In other words, the right half of Figure 2, A *might* decide to trust C to some *limit*. It does not mean A automatically trusts C to the same limit that B does.

•

How this principle applies to the ToIP stack

In many respects the entire purpose of the entire ToIP stack is to **enable ToIP users to leverage transitive trust relationships to gain the trust evidence they need** to make faster, more informed, more confident trust decisions. At the heart of this design is a specific version of the transitive trust triangle called the **verifiable credential trust triangle** shown in Figure 22.

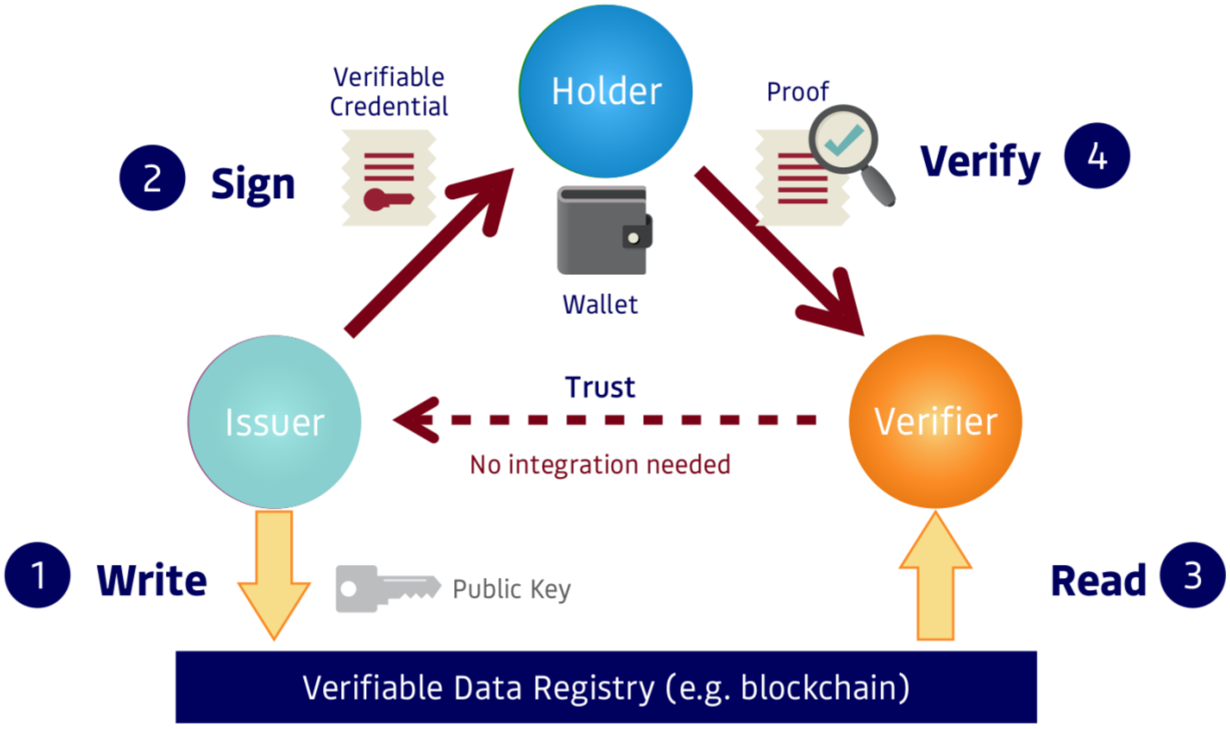


Figure 22. The verifiable credential trust triangle implemented at ToIP Layer 3

For a complete explanation of this trust triangle, please see our Introduction to ToIP white paper. Note that the verifiable credential trust triangle reverses the order of the parties shown in Figure 21 to the order shown in Figure 23:

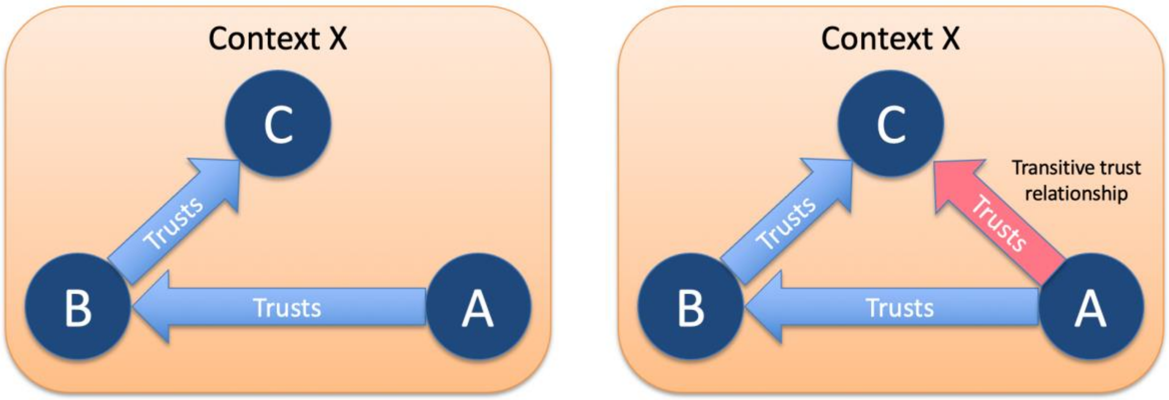


Figure 23. The transitive trust roles in the verifiable credential trust triangle

In other words, if Verifier A trusts Issuer B, and Issuer B trusts data it is willing to issue in a verifiable credential to Holder C, then if Holder C can present a proof of that verifiable credential to Verifier A, and Verifier A can use that data to make a transitive trust decision about Holder C.

But the design of the ToIP stack does not stop with a single trust triangle. To build scalable digital trust ecosystems, it adds a second trust triangle to form the **governance trust diamond** shown in Figure 24.

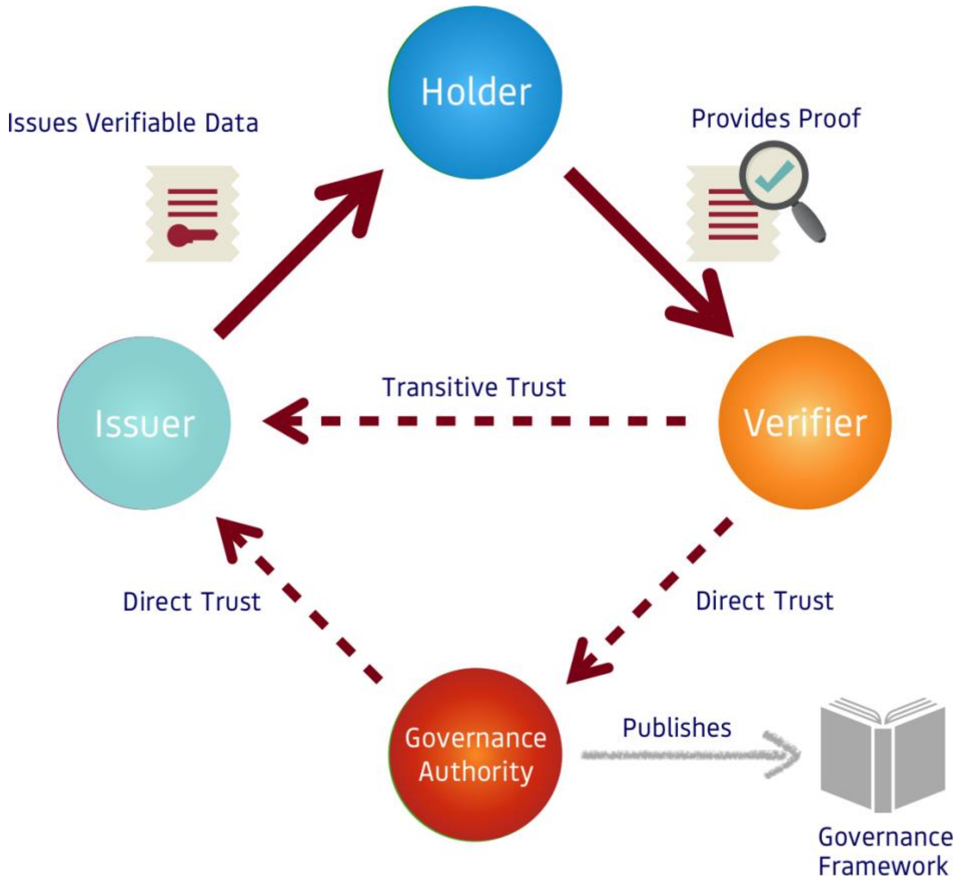


Figure 24. The ToIP governance trust diamond

In the second trust triangle in the lower half of Figure 23, the Verifier trusts the Governing Authority and the Governing Authority trusts the Issuer, therefore the Verifier can have some degree of transitive trust in the Issuer. This of course is the model already used in some of the largest trust networks in the world, such as the Mastercard and Visa payment networks. With ToIP architecture, this model can be applied by any digital trust community of any size for any purpose.

In summary, transitive trust is the "DNA" around which the entire ToIP stack is designed. In fact, the only layer where it does not apply is Layer 2 where the connections are peer-to-peer.

Layer	Relevance	Explanation
4	★★★★★	Layer 4 is where transitive trust is scaled to entire digital trust application ecosystems using ecosystem governance frameworks.
3	★★★★★	Layer 3 is the home of the verifiable credential exchange protocols and governance frameworks necessary to enable the transitive trust triangle.
2		Layer 2 agents and wallets provide the tooling necessary to support higher layers but are not directly involved in producing transitive trust.
1	★★★★★	As shown in Figure 13.2, Layer 1 public key utilities provide the strong cryptographic trust anchors necessary for verifiable credentials to transmit transitive trust evidence at Layer 3.

#14: Trust and Technology have a Reciprocal Relationship

Technology can only help humans build trust if humans trust the technology.

This final "wet code" principle ties together all of the principles in Parts 1 and 2. In a nutshell, it explains the situation we are in with trust on the Internet today. Because the necessary security, privacy, and confidentiality protections were not built in at the outset (again we recommend the Washington Post series Net of Insecurity: A Flaw in the Design to understand why this happened), many of us distrust the very network on which we are becoming more dependent every day.

This feeds a vicious cycle: the less trust we have in the Internet, the harder it is to use it as a tool to build trust—and the worse the problem becomes. This is exacerbated every time we add another new technology that has its own trust issues, for example:

- **Facial recognition technologies** that cause people to believe they can no longer control their biometrics.
- **Social media algorithms** that amplify disinformation because it produces the highest user engagement rates.
- **Artificial intelligence (AI) bots** whose algorithms become so complex and dynamic that humans cannot explain them.[20]

In short, the bigger the "trust gap" we develop with technology, the harder we make it for technology to help us solve the problem.

How this principle applies to the ToIP stack

This principle summarizes the very purpose of the ToIP stack as a whole: to reverse the decline and start a **virtuous cycle** where the more we can trust the technology, the more we can use it to build trust in our digital lives.

The most important application of this principle is that ToIP architecture must address BOTH trust in technology AND trust in humans. This is fundamentally the reason for designing the ToIP stack in two halves as shown in Figure 25.

[20] For further discussion of the unique trust challenges that AI systems pose, see Draft NIST-IR 8332: "Trust and Artificial Intelligence". For an interesting study of about how to leverage decentralized trust technologies to help meet AI's socio-technical challenges, see the research paper "A Decentralized Approach Towards Responsible AI in Social Ecosystems".

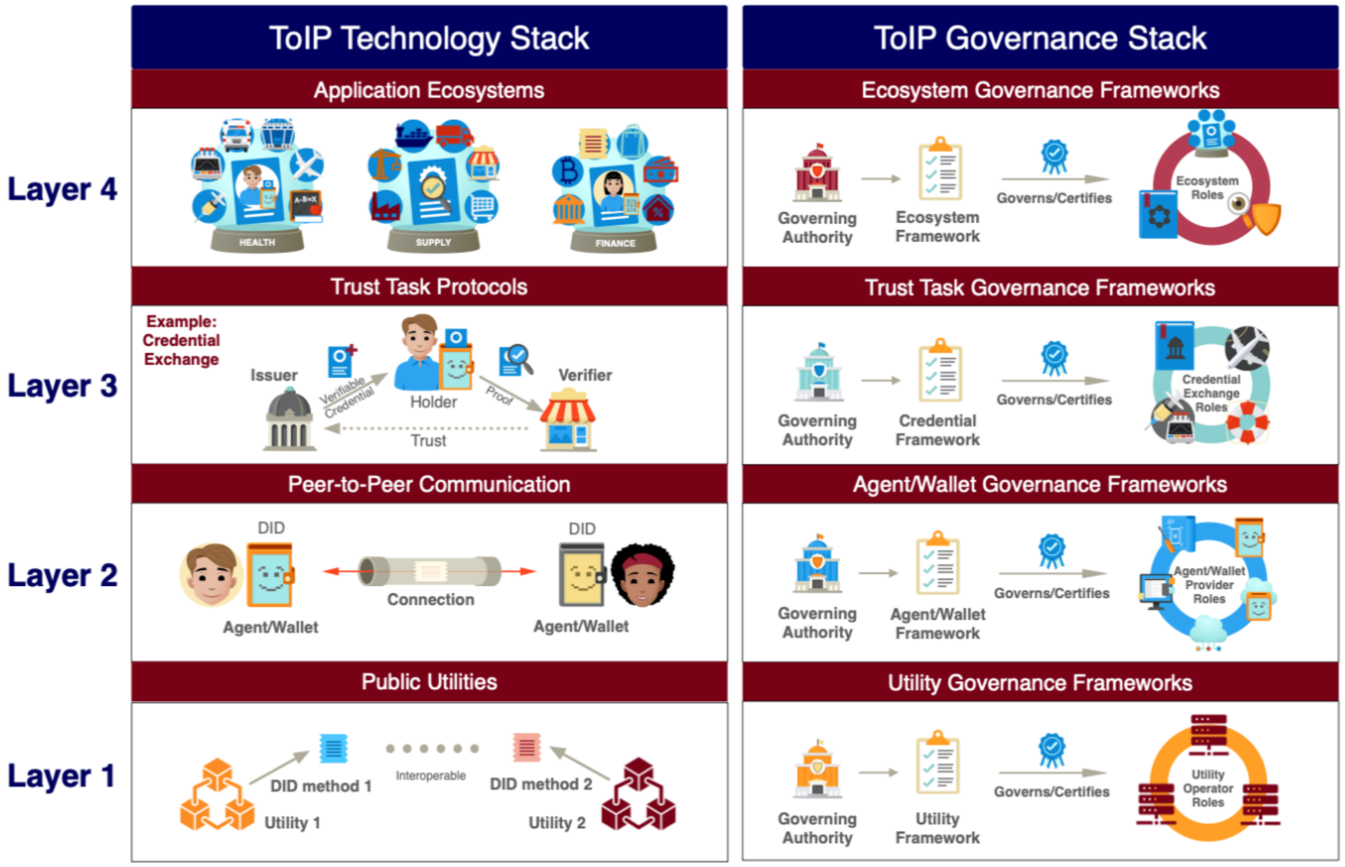


Figure 25. The two-half design of the ToIP stack reflects the reciprocal relationship of trust and technology

Each of the four layers needs governance (the right half of figure 25) because **humans need to trust the technologies deployed at each layer**. For layers 1 and 2, this is primarily trust in cryptographically verifiable communications across a decentralized global network. For layers 3 and 4, this is primarily trust in the protocols used to exchange verifiable credentials within and between digital trust ecosystems.

This principle also circles back to the principles in Part 1—in particular to #2: Connectivity Is Its Own Reward—because the easier it becomes to use the ToIP stack to connect and build trust relationships between any two peers—and the more confidence those peers have in using the ToIP stack—the faster we reverse the cycle and start building new trust relationships, new trust communities, and new digital trust ecosystems.

Layer	Relevance	Explanation
4	★★★★★	Ecosystem governance frameworks at Layer 4 are how ToIP users can evaluate and make trust decisions about digital trust ecosystems and the applications running within them.
3	★★★★★	Credential governance frameworks at Layer 3 are how ToIP users can evaluate and make trust decisions about issuers, holders, and verifiers.
2	★★★★★	Agent governance frameworks at Layer 2 are how ToIP users can evaluate and make trust decisions about digital agents and digital wallets.
1	★★★★★	Utility governance frameworks at Layer 1 are how ToIP users can evaluate and make trust decisions about public key utilities.

Part Three: Overall Principles

This final set of design principles apply to the whole of the ToIP stack, “wet or dry”, because they reflect the values of the ToIP Foundation in undertaking this work and the impact we hope it will have on the future of the Internet and the Web.

#15: Design for Ethical Values

The ToIP Stack has a strong ethical dimension. Design it with a commitment to ethical values.

We live in an age of technology. As a Foundation, we believe the kind of technology we develop at ToIP can have a significant impact on people, society, and the environment. This is especially true as we work on both the Technology Stack and the Governance Stack to arrive at an integrated architecture to bring trust to the Internet.

IETF RFC 3935[21], in the context of discussing the mission of IETF, has the following paragraph (emphasis added):

> *The Internet isn't value-neutral, and neither is the IETF. We want the Internet to be useful for communities that share our commitment to openness and fairness. We embrace technical concepts such as* ***decentralized control****,* ***edge-user empowerment*** *and* ***sharing of resources****, because those concepts resonate with the core values of the IETF community. These concepts have little to do with the technology that's possible, and much to do with the technology that we choose to create.*

We could not say it better for the ToIP community.

Ethics in technology can be a complex and difficult subject. However, that should not prevent us from making the commitment to ethical values. Such commitment extends to situations where there is no simple clear cut answer or consensus to an ethical question. It requires us to dedicate conscious effort to raise, consider, and debate these questions when and where they arise—and then guide our decisions based on ethical values. It also extends to the whole life cycle of technology standard development, from formulating a need, to choosing among possible solutions, to evaluating it after the technology is deployed, and finally to improving it in the next cycle.

The Design for Ethical Values principle should be seen as a first principle from which other principles may be derived, at least in part. It applies to both the “dry code” and “wet code” principles. Part of the rationale for the Confidentiality by Design and Default principle and the Keys at the Edge principle is the ethical value we ascribe to *privacy*. And the ethical value of *autonomy*, *independence,* and *agency* is one reason for the Decentralization by Design and Default principle (as well as the Key at the Edge principle). Similarly, the entire set of the trust principles in Part 2 are in part derived from ethical values both from an individual level and a societal level that reflect our understanding of human needs and the kind of society we want to live in.

[21] https://datatracker.ietf.org/doc/html/rfc3935

In addition to base level[22] ethical values in technology such as *safety*, *privacy,* and *autonomy*, philosophers and ethicists also consider second level values, for example, *accessibility*, *reliability*, *functionality,* and *economics*. These values reflect respect for a person's time, labor, and limited economic resources. If a technology is very costly to deploy, it has less ethical value than an alternative that costs less (at least on this account). Another ethical value we uphold is *fairness*. Technology designs often bring into question fairness concerns even if they are unintentional. For example, a verifiable credential solution that only works on high-end smartphones will disadvantage people who cannot routinely afford such devices, thereby increasing societal inequality.

From the perspective of higher-level human ethical values, we also want happiness, creative pursuits, and other inspirations. These values respect universal human needs for fulfillment and reaching a person's full potential. Technology design can open doors as well as close doors. Technological products can be delightful or dreadful. It is an ethical responsibility that we consider these factors when we develop new technology. This is a primary reason that the ToIP Foundation has, for example, established a Human Experience Working Group dedicated to make ToIP based technology not just accessible, but whenever possible delightful for everyone to use.

Finally, sustainability is a critical and mandatory ethical consideration at a time when human activities' environmental footprint is threatening the entire planetary ecosystem and our very survival. Technology has contributed significantly to that footprint in the past and continues to do so today. It is our responsibility to put the ToIP stack on a sustainable path. It requires us to carefully evaluate and project potential environmental impacts throughout the design process—especially with ToIP Layer 1 public utilities—as well as promote this value in the deployment of digital trust ecosystems we help incubate.

[22] See Maslow's Hierarchy of Needs for considering different levels of needs from which we derive ethical values. https://en.wikipedia.org/wiki/Maslow's_hierarchy_of_needs

How this principle applies to the ToIP stack

Layer	Relevance	Explanation
4	★★★★★	The ultimate goal of the ToIP stack is to support the formation, growth, and sustainability of digital trust ecosystems of all kinds. The shape and governance of these communities is where our values will most clearly be reflected.
3	★★★★★	Exchange of verifiable credentials, money, and other trusted data will form the basis of future digital services and marketplaces. How these protocols are designed can have a direct impact in people's lives and interests.
2	★★★★★	Universal peer-to-peer connectivity and confidential communications at Layer 2 can have a tremendous impact on personal empowerment and autonomy. It can also help restore privacy, accessibility, and fair sharing of resources.
1	★★★★★	By serving as anchors for public trust, Layer 1 public utilities can have an impact on social trust as well as people's dependence on technology platforms. These utilities, when scaled, can have significant energy and other environmental footprints.

#16: Design for Simplicity

The simpler the design of a protocol, the more likely it is to be successful.

Section 2 of IETF RFC 3439, *Some Internet Architectural Guidelines and Philosophy*, states:

> *The Simplicity Principle, which was perhaps first articulated by Mike O'Dell, former Chief Architect at UUNET, states that complexity is the primary mechanism which impedes efficient scaling, and as a result is the primary driver of increases in both capital expenditures (CAPEX) and operational expenditures (OPEX). The implication for carrier IP networks then, is that to be successful we must drive our architectures and designs toward the simplest possible solutions.*

"Do the hard work to keep it simple" is one of the nine design principles published by the U.K. National Health Service (NHS) in 2018. It is especially appropriate to the design of large-scale decentralized networks and other complex adaptive systems. Equally appropriate is this famous quote about simplicity from John Gall in his 1975 book *Systemantics*:[23]

> *A complex system that works is invariably found to have evolved from a simple system that worked. A complex system designed from scratch never works and cannot be patched up to make it work. You have to start over with a working simple system.*

Simplicity is so prized in systems design that Steve Jobs, the late CEO of Apple and the visionary behind the iPhone, put it this way:

> *That's been one of my mantras — focus and simplicity. Simple can be harder than complex. You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains.*

Simplicity has an extra special place in the design of trust systems because the easier it is to understand the design of the system, the easier it is to build it, test it, use it, and develop confidence in it.

All that said, there is also the dictum widely attributed to Einstein: "Everything should be made as simple as possible, but not simpler." In other words, there is a minimum threshold of complexity not just to the technology and cryptography required to implement the ToIP stack, but for the "wet code" required for governance frameworks at each layer. Thus the quote that is perhaps the most relevant to designing for simplicity in the ToIP stack is from Oliver Wendel Holmes:[24]

> *"I would not give a fig for the simplicity on this side of complexity, but I would give my life for the simplicity on the other side of complexity."*

[23] https://en.wikipedia.org/wiki/John_Gall_(author)
[24] https://www.planplusonline.com/simplicity-side-complexity/

PlanPlus Online created this illustration of Holme's meaning:

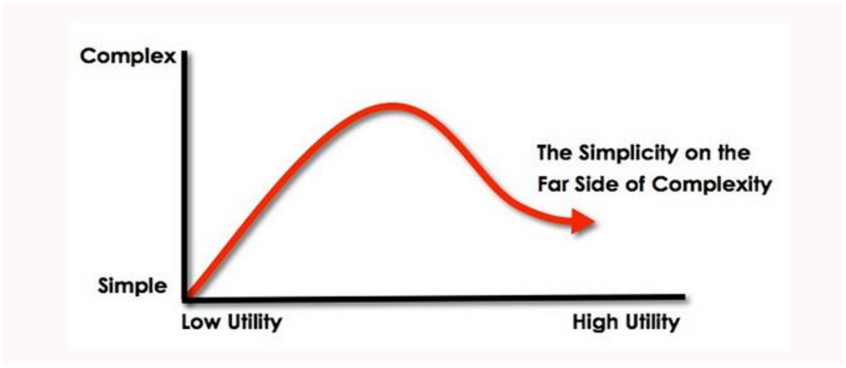


Figure 26. The simplicity on the far side of complexity

Simply put, the design goal for the ToIP stack is to reach *the simplicity on the other side of complexity*.

How this principle applies to the ToIP stack

Layer	Relevance	Explanation
4	★★★★★	At Layer 4 the virtue of simplicity applies to the governance of a digital trust ecosystem: the easier it is to understand the roles, rules, and incentives, the better the chance the ecosystem will grow and thrive.
3	★★★★	Data exchange co-protocols at Layer 3 will inherently be more complex than Layer 2, however the same dictum applies: the simpler the design, the greater the chances of adoption.
2	★★★★★	Simplicity is critical to the design of the Layer 2 peer-to-peer secure communications protocol at the heart of the ToIP stack—see Principle #3: The Hourglass Model for details.
1	★★★	The simpler the design and implementation of a Layer 1 public utility that meets the market's security, reliability, availability, immutability, and sustainability requirements, the more successful it is likely to be.

#17: Design for Constant Change

The only constant on the Internet is change. Design for it.

In the landmark June 1996 RFC 1958, *Architectural Principles of the Internet*, the opening paragraph said:

> *In searching for Internet architectural principles, we must remember that technical change is continuous in the information technology industry. The Internet reflects this. Over the 25 years since the ARPANET started, various measures of the size of the Internet have increased by factors between 1000 (backbone speed) and 1,000,000 (number of hosts). In this environment, some architectural principles inevitably change. Principles that seemed inviolable a few years ago are deprecated today. Principles that seem sacred today will be deprecated tomorrow.* ***The principle of constant change*** *is perhaps the only principle of the Internet that should survive indefinitely.*

The 25 years since this RFC was published have demonstrated the wisdom of this principle. In many respects the Internet of today is barely recognizable from the Internet of 1996, let alone the Internet of 1976. The changes are legion:

- **Compute power**: Moore's Law—the observation that the number of transistors in a dense integrated circuit (IC) doubles about every two years—has held true *since 1970*!
- **Transmission speeds** have gone from 50 Kbps in 1984 to an average of over 200 Mbps today—a *4000% increase*.[25]
- **The number of Internet users** has grown from 2.6 million in 1990 to 3.4 billion in 2016—a *1300% increase*.[26]

Beyond just size, speed, and power, usage of the Internet has gone through several revolutions. First there was the pre-Web Internet, when usage was confined primarily to academic, military, and government needs. Then the Web emerged and started to go through its own evolutionary stages:

1. **Web 1.0** refers the first generation of the Web consisting primarily of static websites navigated using basic search engines like Alta Vista and directories like Yahoo.
2. **Web 2.0** was coined in 1999 to describe the second-generation Web that was driven by user-generated content, self-publishing, social networking, and interoperability with other products, systems, and devices.
3. **Web 3.0** was originally a term coined by Tim Berners-Lee to describe the Semantic Web. However, it has since evolved into much wider usage as a shorthand for the movement to a much more decentralized version of the web powered by blockchain and other decentralized technologies.

[25] https://www.eff.org/deeplinks/2021/07/future-symmetrical-high-speed-internet-speeds
[26] https://ourworldindata.org/internet

Another term for Web 3.0 is the *Decentralized Web*, which was described by Matthew Hodgson in an October 2019 Techcrunch article:

> *The Decentralised Web envisions a future world where services such as communication, currency, publishing, social networking, search, archiving etc are provided not by centralised services owned by single organisations, but by technologies which are powered by the people: their own community. Their users.*
>
> *The core idea of decentralisation is that the operation of a service is not blindly trusted to any single omnipotent company. Instead, responsibility for the service is shared: perhaps by running across multiple federated servers, or perhaps running across client side apps in an entirely "distributed" peer-to-peer model.*
>
> *Even though the community may be "byzantine" and not have any reason to trust or depend on each other, the rules that describe the decentralised service's behaviour are designed to force participants to act fairly in order to participate at all, relying heavily on cryptographic techniques such as Merkle trees and digital signatures to allow participants to hold each other accountable.*

It is no coincidence that this description of this latest evolutionary step for the Web aligns directly with the mission of the ToIP Foundation. Our job is to standardize the technology stack and governance stack that will make the Decentralized Web possible.

However it does not take a rocket scientist to see that *this will not be the last evolutionary step for the Internet or the Web*. Even if we are completely successful in our mission at the ToIP Foundation, *change is constant*. Our design for the ToIP stack needs to not just solve the issues in digital trust we have today, but anticipate as much as possible the new issues we will have tomorrow—and for decades to come.

How this principle applies to the ToIP stack

Layer	Relevance	Explanation
4	★★★★★	Layer 4 is where adoption of the ToIP stack will happen at scale. This is where new use cases and apps will take off just like with the Web and smartphones. The evolution of governance frameworks for digital trust ecosystems will need to keep pace.
3	★★★★★	W3C Verifiable Credentials Data Model 1.0 became a standard in Sept 2019; that was a starting gun for innovation in every aspect of VCs—formats, signature algorithms, ZKPs, exchange protocols. This too will evolve just as quickly as the Web did.
2	★★★★★	Digital wallets—hardware wallets, mobile wallets, cloud wallets—are at a similar stage of maturity as the early Web browsers. So are protocols at this layer. We need to plan for rapid change and especially maintain cryptographic agility.
1	★★★★★	Layer 1 public utilities are still in their infancy. Even with their mission of providing rock-solid immutable anchors for cryptographic assurance, much remains to be proven at this layer before we can take it for granted as "plumbing".

Additional Relevant Principles

In addition to the other references in this document, these sets of design principles are cited as particularly relevant to the overall work of designing the ToIP stack and ToIP digital trust ecosystems.

The Laws of Identity

This famous set of seven principles, authored in 2004 by Kim Cameron when he was Chief Architect of Identity for Microsoft, set out the requirements for what Kim called the "identity metasystem".

The Principles of SSI

These twelve principles were developed in 2020 in a cross-community volunteer effort facilitated by the Sovrin Foundation in order to arrive at a consistent definition of SSI.

eSSIF-Lab Principles

These six principles capture what the European Self-Sovereign Identity Lab believes is the essence of SSI—that it refers to all concepts/ideas, architectures, processes and technologies that aim to support (autonomous) parties as they negotiate and execute electronic (business) transactions with one another.

Data Design Principles

This is a work-in-progress authored by the ToIP Inputs and Semantics Working Group that covers a collection of different principles about data design and why it is a critical component of a successful digital trust ecosystem.

The Trust Over IP Foundation (ToIP) is hosted by the Linux Foundation under its Joint Development Foundation legal structure. We produce a wide range of tools and deliverables organized into five categories:

- ❖ Specifications to be implemented in code
- ❖ Recommendations to be followed in practice
- ❖ Guides to be executed in operation
- ❖ White Papers to assist in decision making
- ❖ Glossaries to be incorporated in other documents

ToIP is a membership organization with three classes—Contributor, General, and Steering.

The work of the Foundation all takes place in Working Groups, within which there are Task Forces self-organized around specific interests. All ToIP members regardless of membership class may participate in all ToIP Working Groups and Task Forces.

When you join ToIP, you are joining a community of individuals and organizations committed to solving the toughest technical and human centric problems of digital trust. Your involvement will shape the future of how trust is managed across the Internet, in commerce, and throughout our digital lives. The benefits of joining our collaborative community are that together we can tackle issues that no single organization, governmental jurisdiction, or project ecosystem can solve by themselves. The results are lower costs for security, privacy, and compliance; dramatically improved customer experience, accelerated digital transformation, and simplified cross-system integration.

To learn more about the Trust Over IP Foundation please visit our website, https://trustoverip.org.