Design Principles of the ToIP Stack

TRUST Over IP FOUNDATION

Feb 16, 2022 ToIP All Members Meeting

AGENDA

- Introduction: the context, the stack, the format
- The 17 design principles
  - The “dry” (technical) principles #1-7
  - The “wet” (humanistic) principles #8-14
  - The more general principles #15-17
- Q & A

Introduction

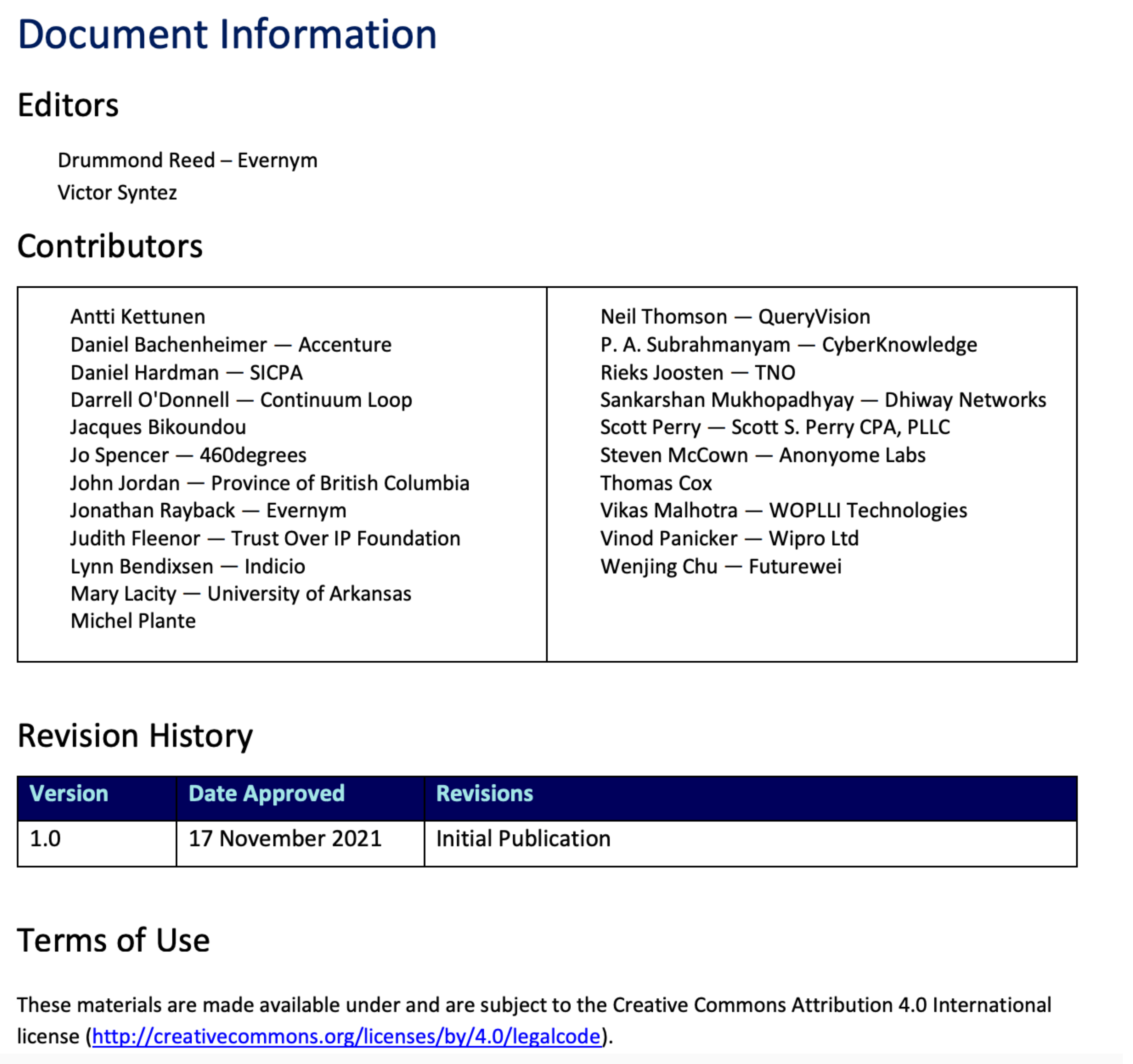
A critical core deliverable for ToIP in 2021!

TRUST Over IP FOUNDATION

Design Principles for the Trust over IP Stack

Version 1.0

17 November 2021



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

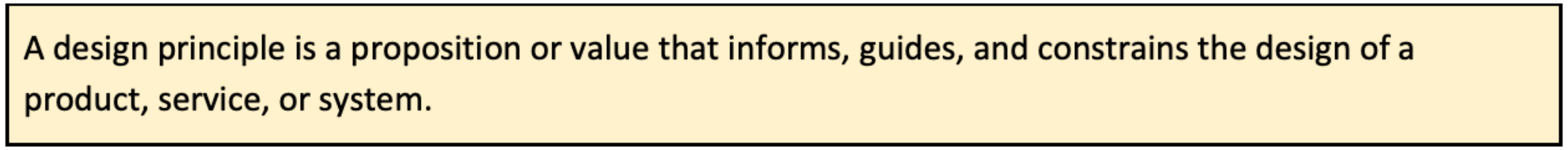
Version	Date Approved	Revisions
1.0	17 November 2021	Initial Publication

Terms of Use

https://trustoverip.org/wp-content/uploads/Design-Principles-for-the-ToIP-Stack-V1.0-2022-01-17.pdf

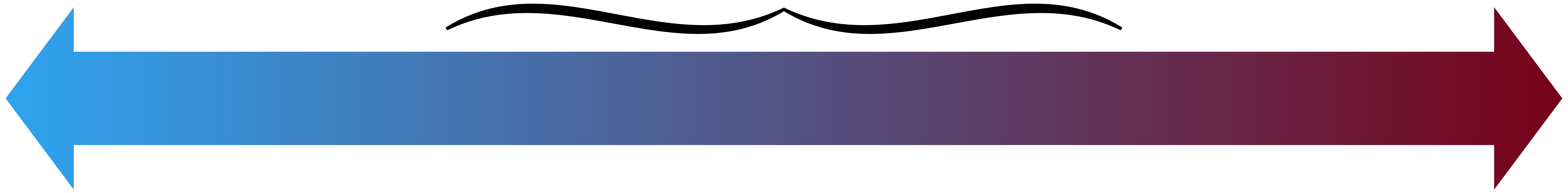
The context

- WHAT are design principles & WHY do we need them?



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

Design Principles

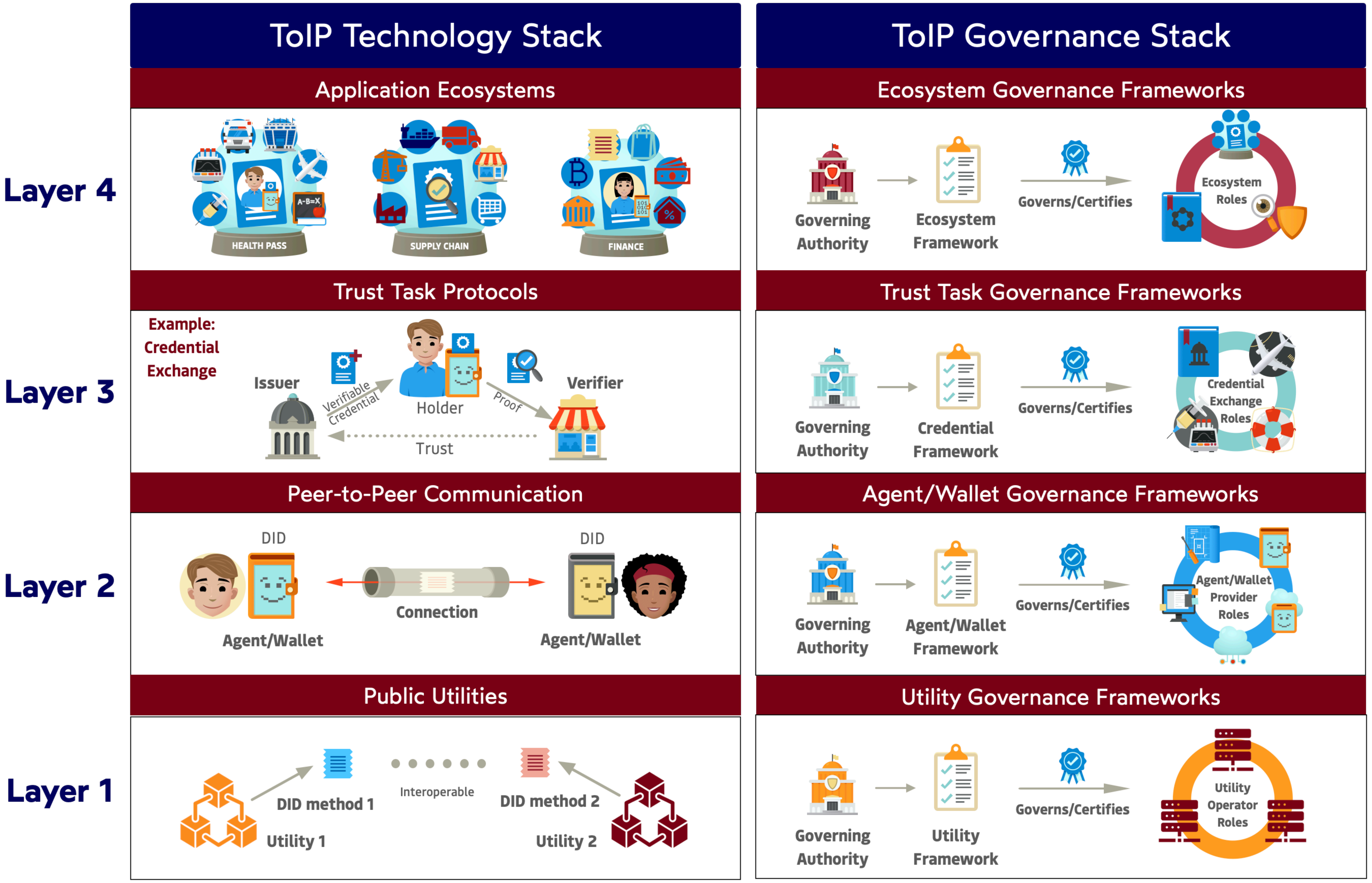


Scientific Laws

Best Practices

The stack

- WHAT are we designing ?



	ToIP Technology Stack	ToIP Governance Stack
Layer 4	Application Ecosystems	Ecosystem Governance Frameworks
Layer 3	Trust Task Protocols	Trust Task Governance Frameworks
Layer 2	Peer-to-Peer Communication	Agent/Wallet Governance Frameworks
Layer 1	Public Utilities	Utility Governance Frameworks

The format

- HOW are we presenting them?

1. A memorable name

#1: The End-to-End Principle

2. A concise statement

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.

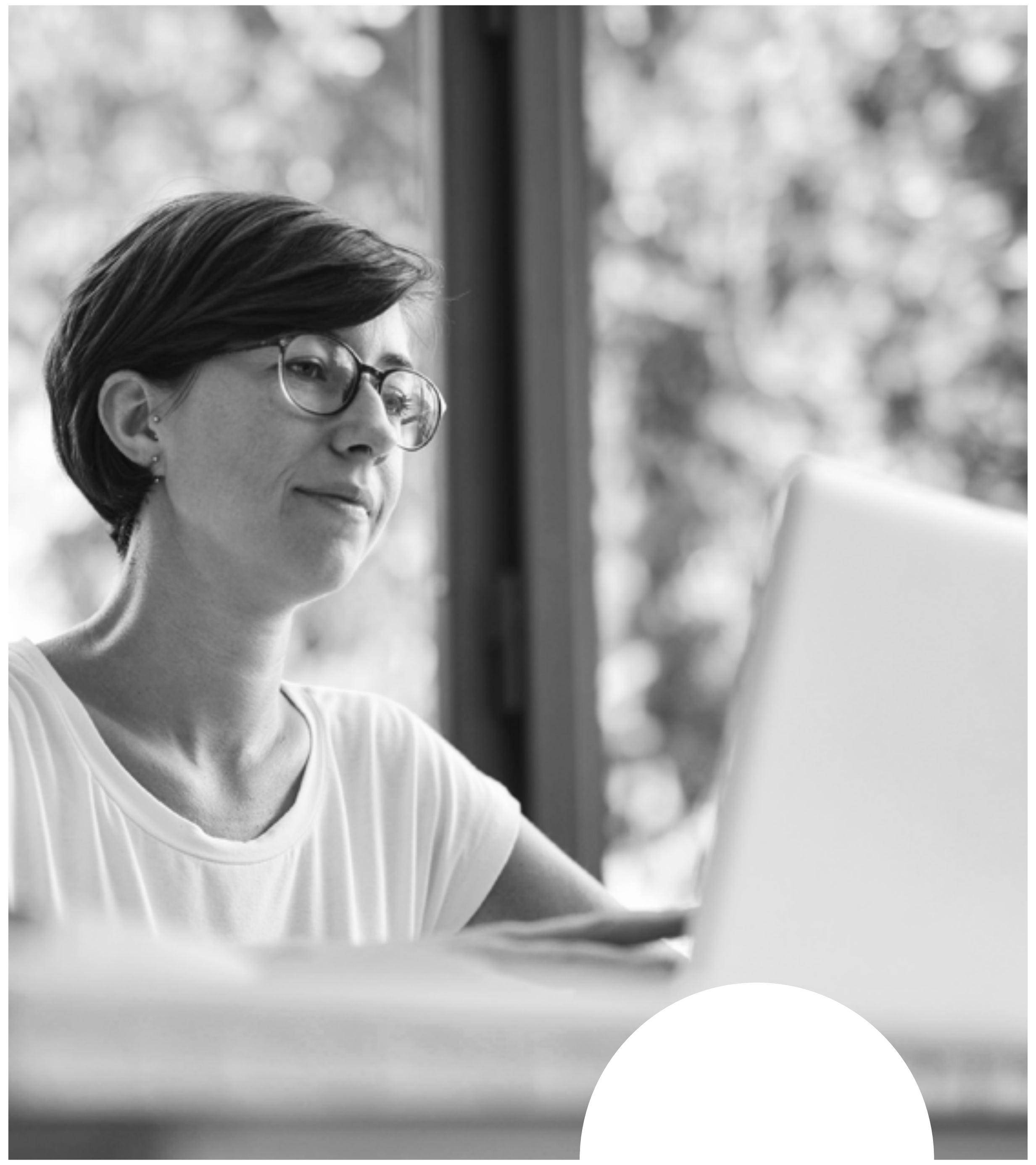
3. Further elaborate the principle

4. How it applies to the ToIP stack design

5. A table summary of how it applies to ToIP stack design

★★★★★	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

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.



Let’s get started

Design principle #1

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

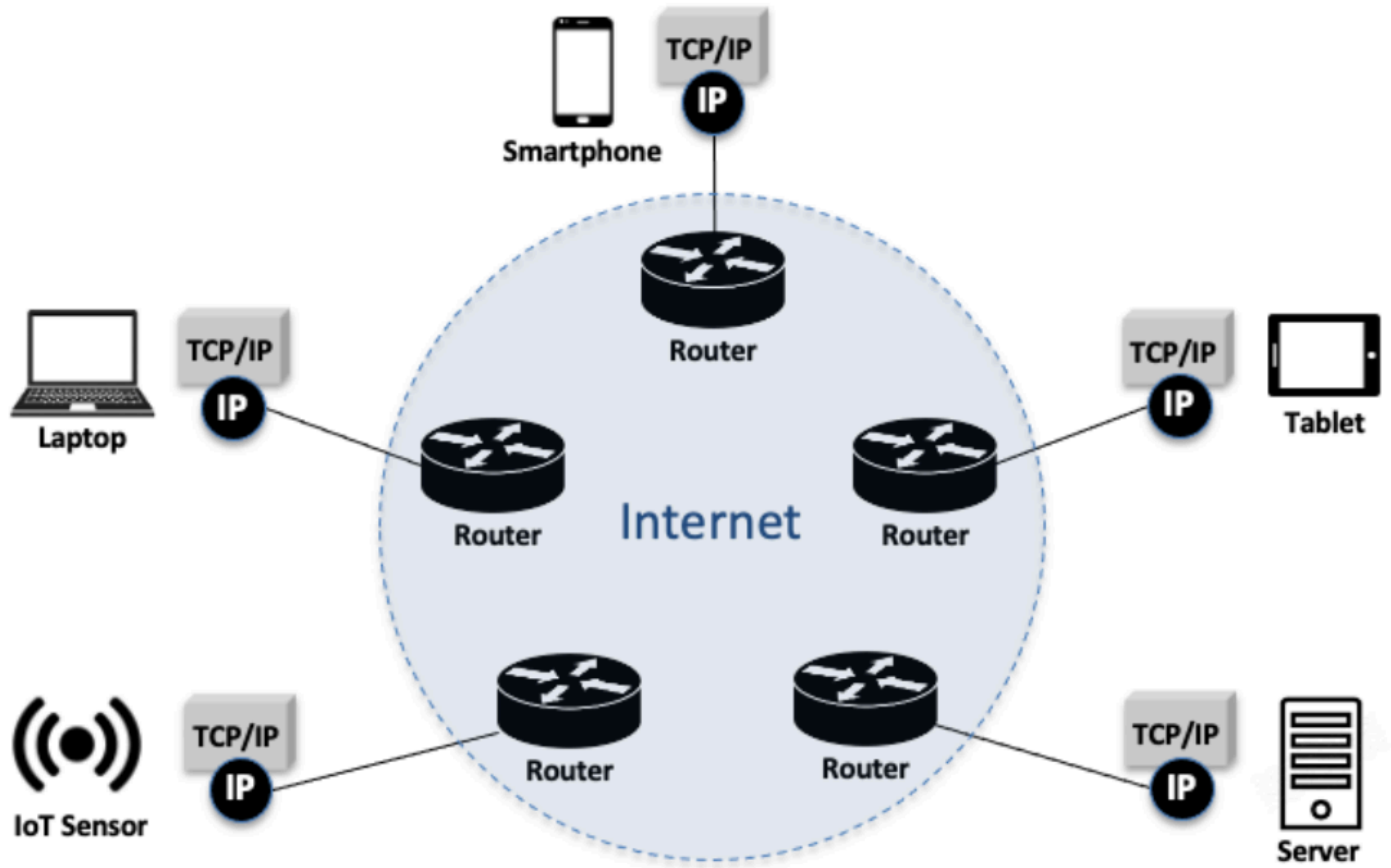


Figure 3. Every device on the Internet runs an instance of the TCP/IP protocol stack so data packets travel "end-to-end" from IP address to IP address

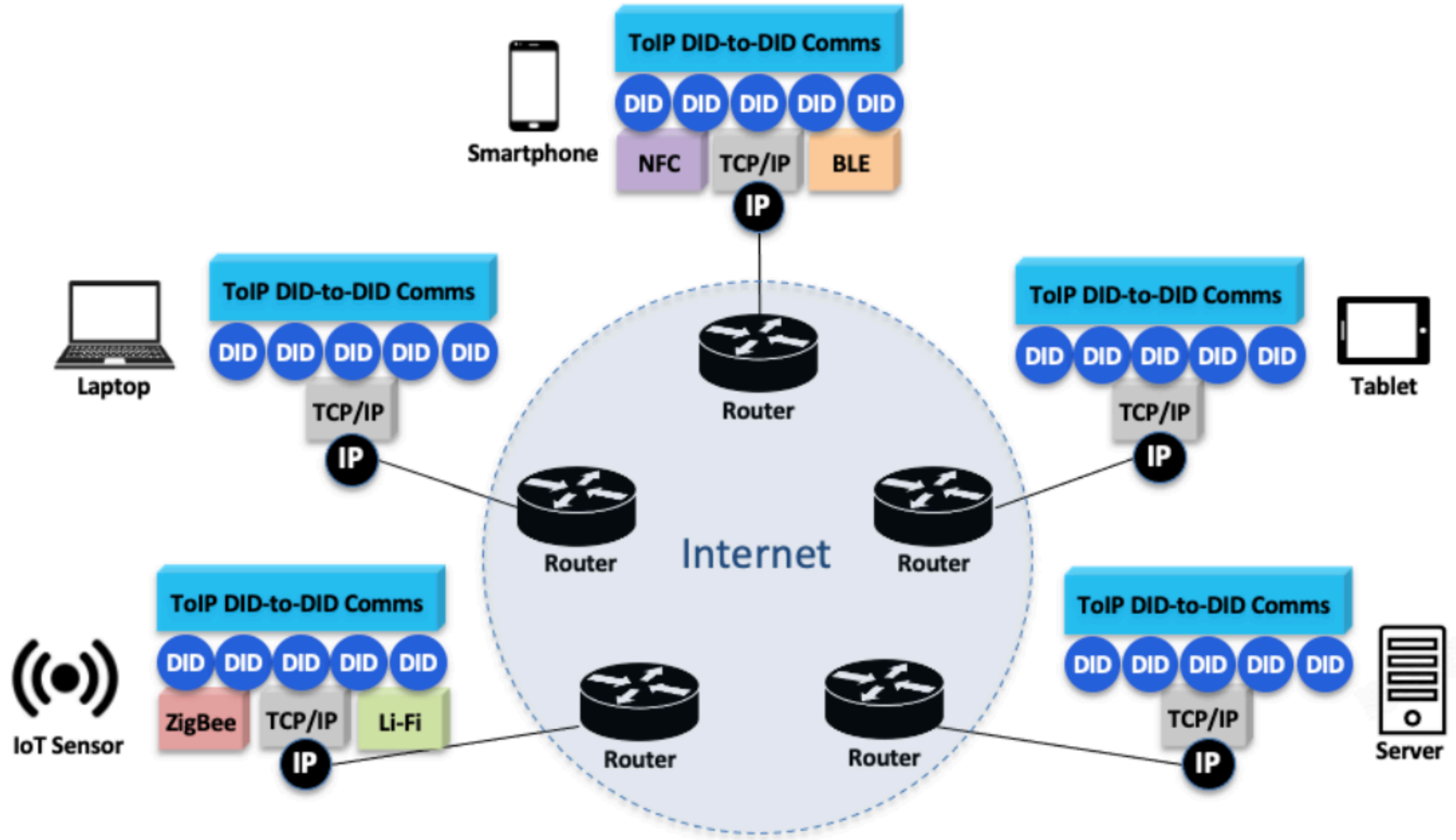
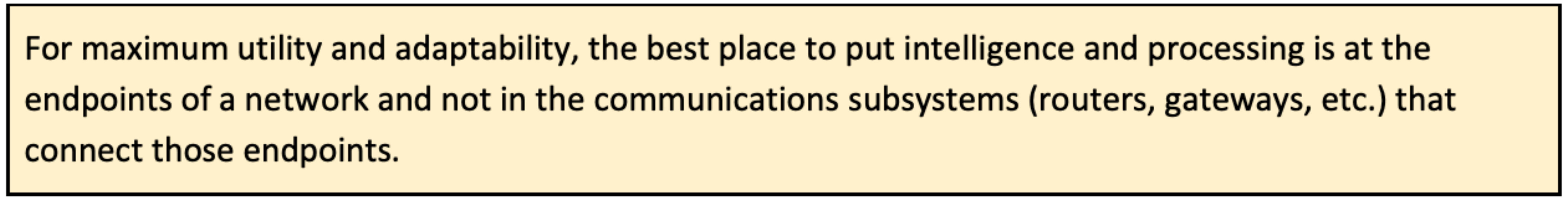


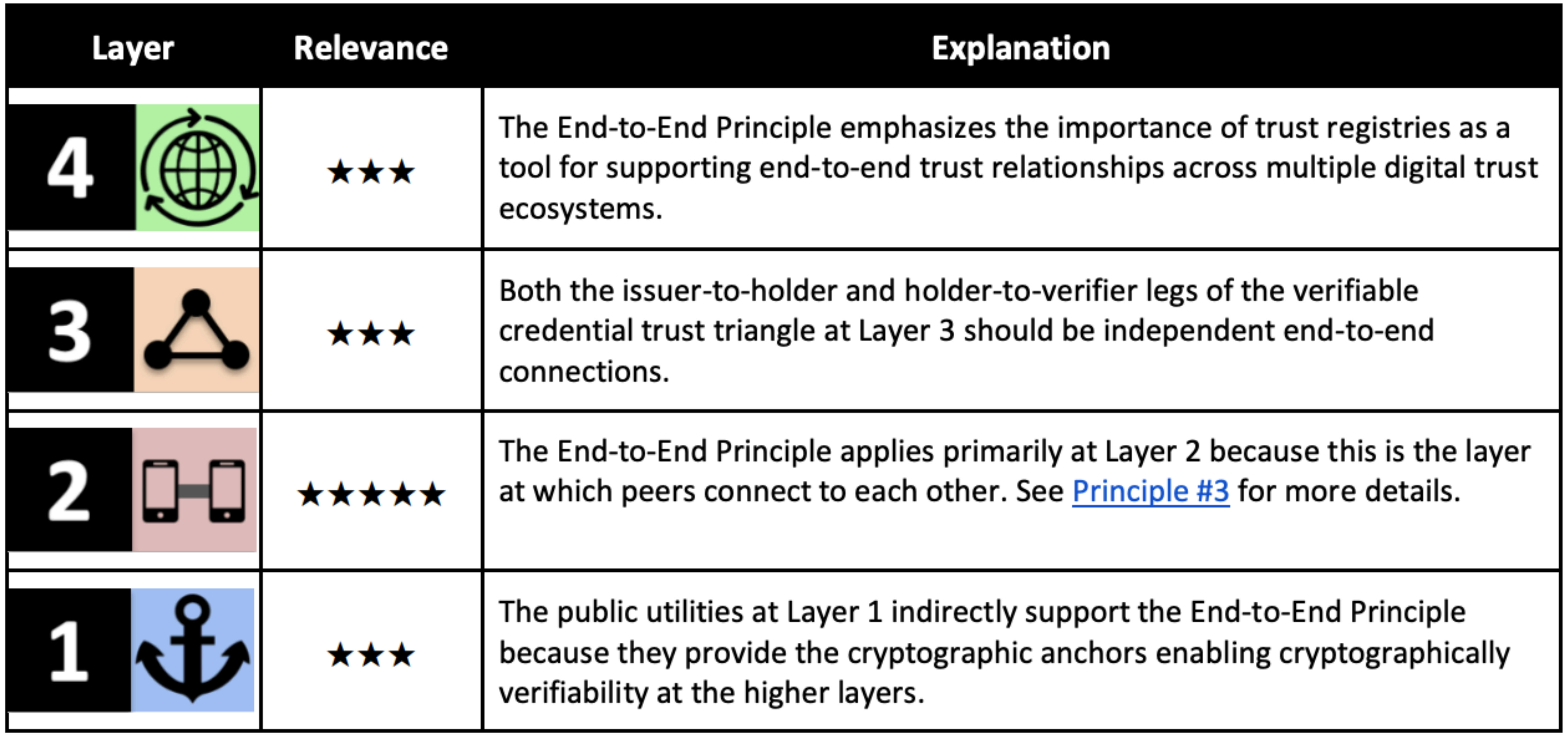
Figure 4. Every device in a ToIP digital trust ecosystem has at least one DID (and often multiple as shown here) and can securely communicate end-to-end between DIDs

Design principle #1



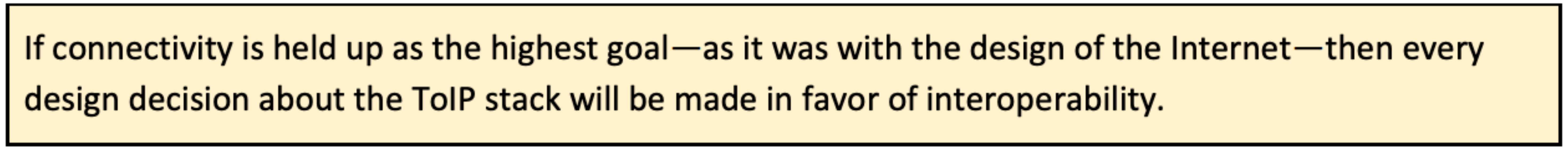
For maximum utility and adaptability, the best place to put intelligence and processing is at the endpoints of a network and not in the communications subsystems (routers, gateways, etc.) that connect those endpoints.

- The End-to-End Principle



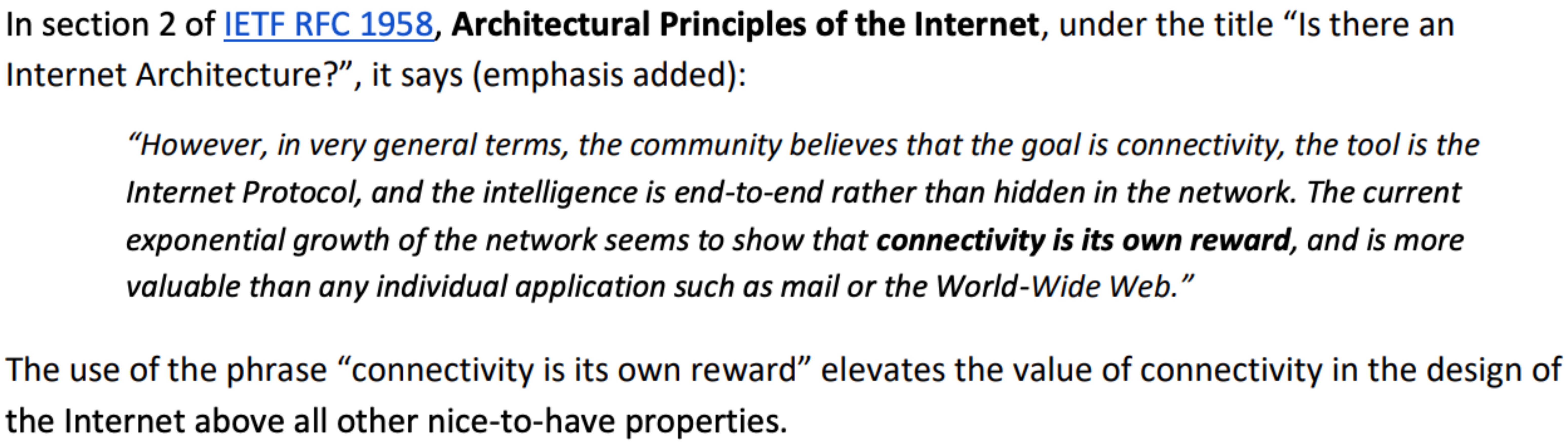
Layer	Relevance	Explanation
4	★★★	The End-to-End Principle emphasizes the importance of trust registries as a tool for supporting end-to-end trust relationships across multiple digital trust ecosystems.
3	★★★	Both the issuer-to-holder and holder-to-verifier legs of the verifiable credential trust triangle 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.

Design principle #2



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.

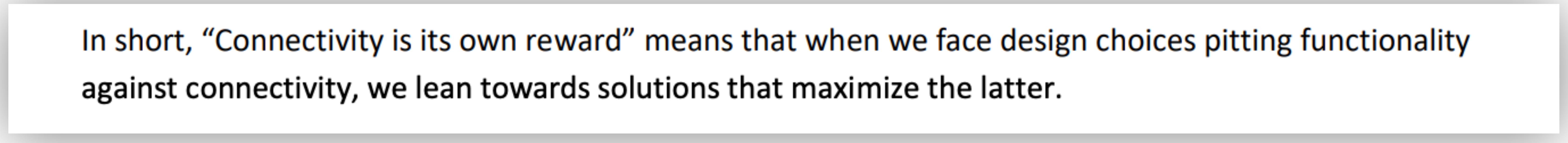
- Connectivity Is Its Own Reward



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

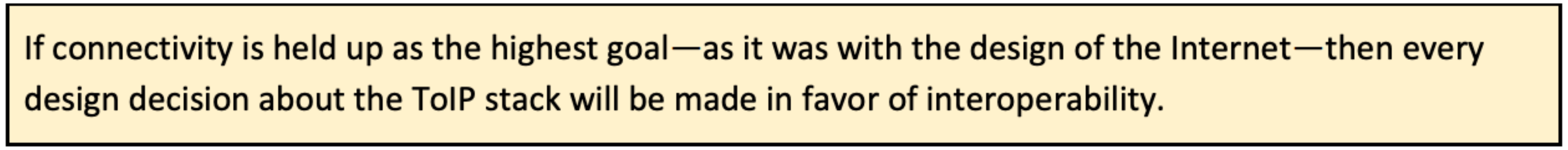
> *“However, in very general terms, the community believes that the goal is connectivity, the tool is the Internet Protocol, and the intelligence is end-to-end rather than hidden in the network. The current exponential growth of the network seems to show that* ***connectivity is its own reward****, and 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.



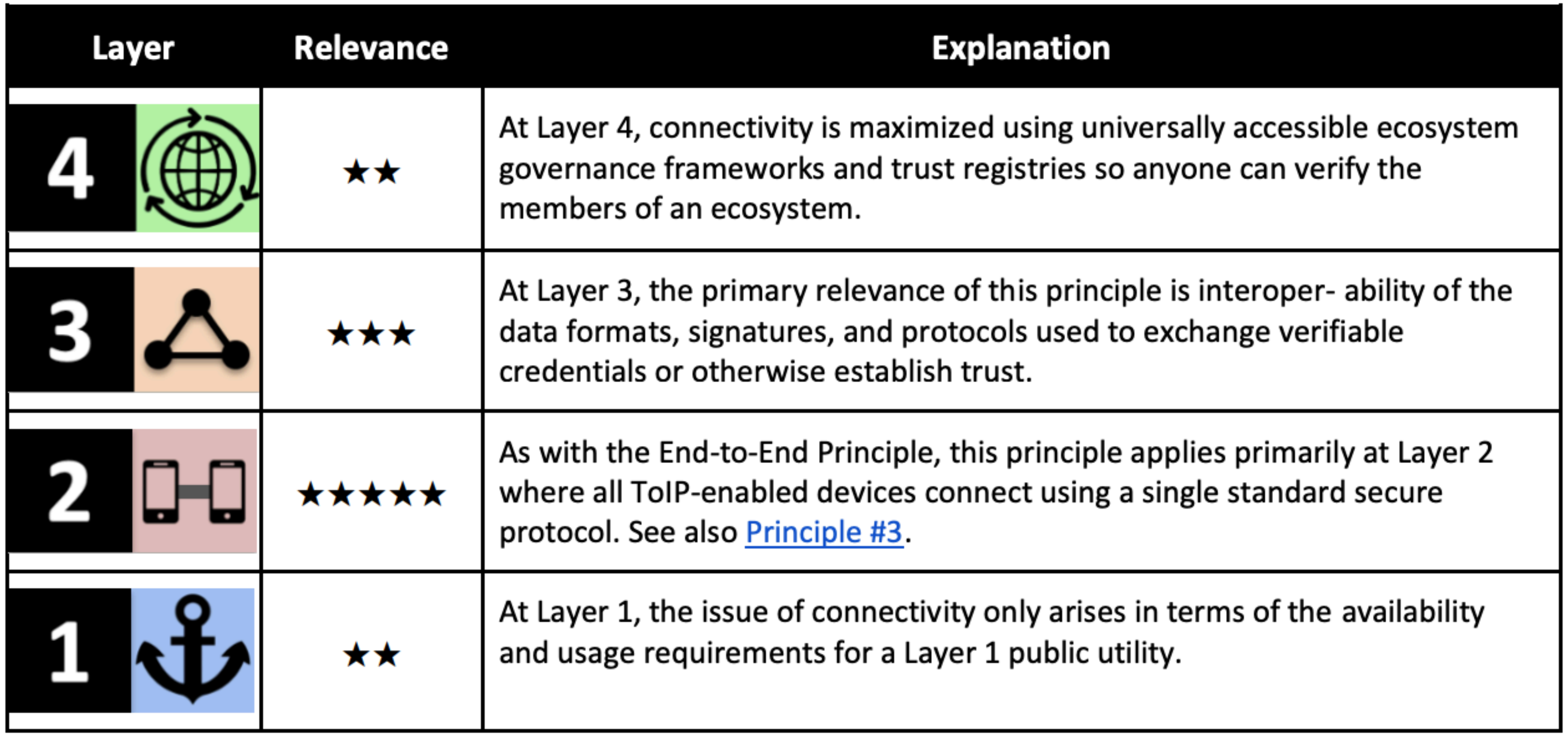
In short, “Connectivity is its own reward” means that when we face design choices pitting functionality against connectivity, we lean towards solutions that maximize the latter.

Design principle #2



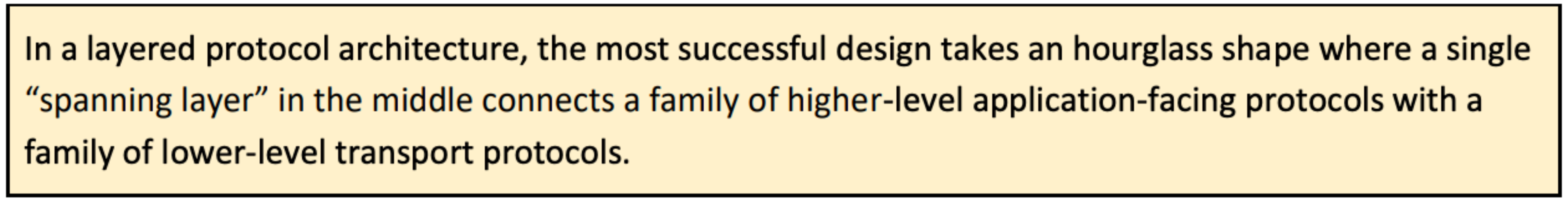
If connectivity is held up as the highest goal—as it was with the design of the Internet—then every design decision about the ToIP stack will be made in favor of interoperability.

- Connectivity Is Its Own Reward



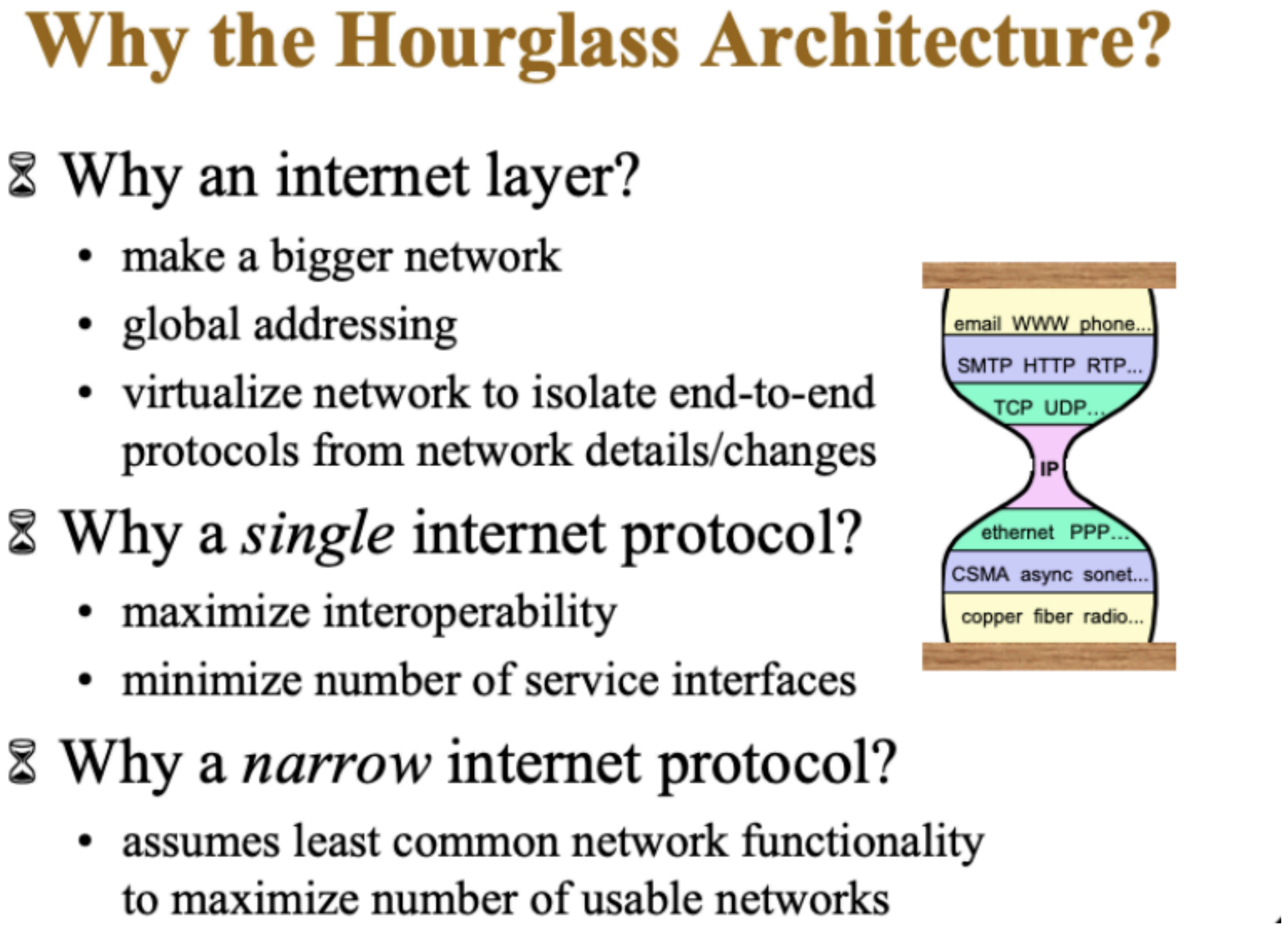
Layer	Relevance	Explanation
4	★★	At Layer 4, connectivity is maximized using universally accessible ecosystem governance frameworks and trust registries so anyone can verify the members of an ecosystem.
3	★★★	At Layer 3, the primary relevance of this principle is interoper- ability of the data formats, 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.

Design principle #3



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.

- The Hourglass Model



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

Design principle #3

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.

- The Hourglass Model

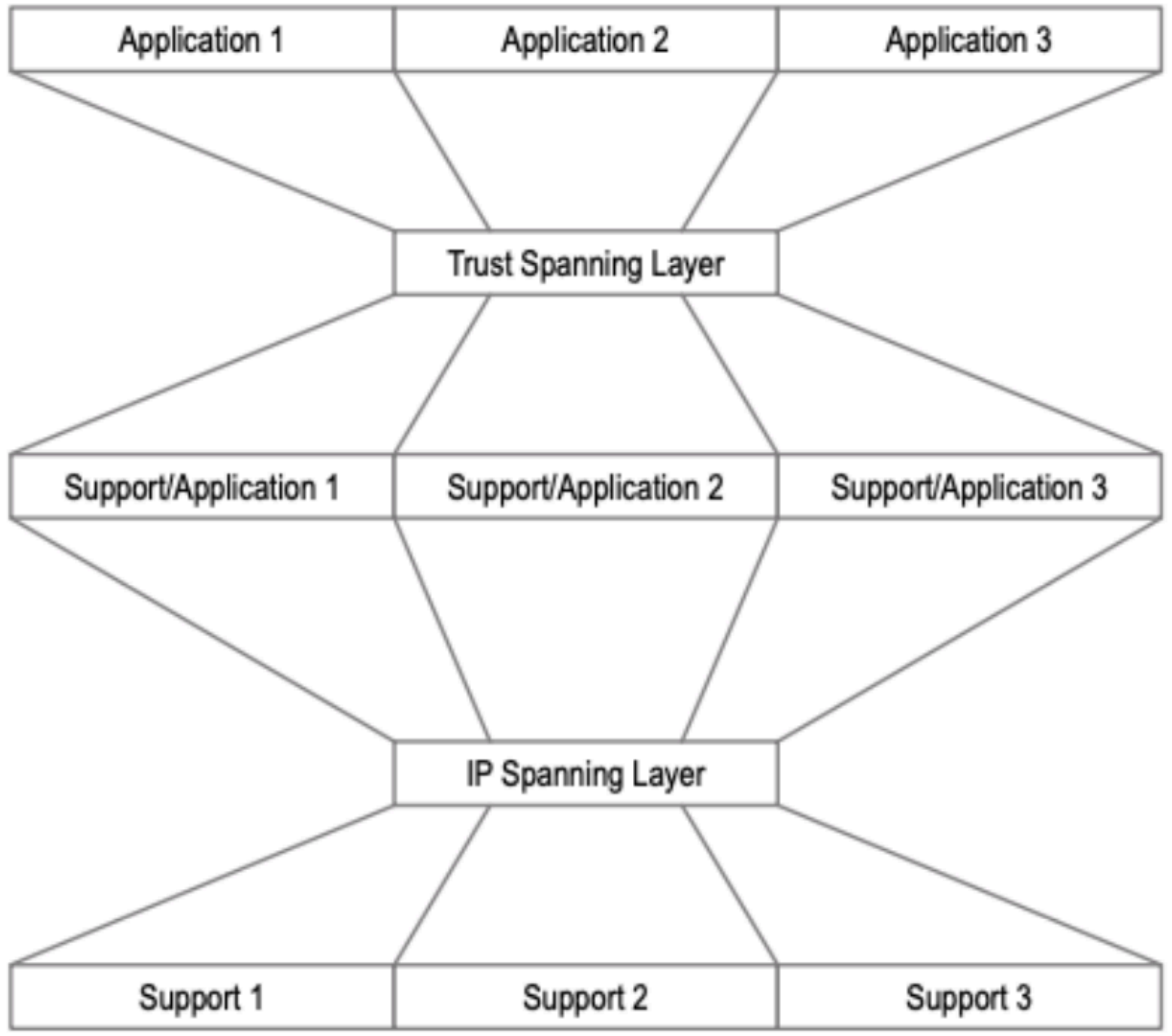


Figure 11. A trust spanning layer as a "secure neck" on top of the IP spanning layer "waist"

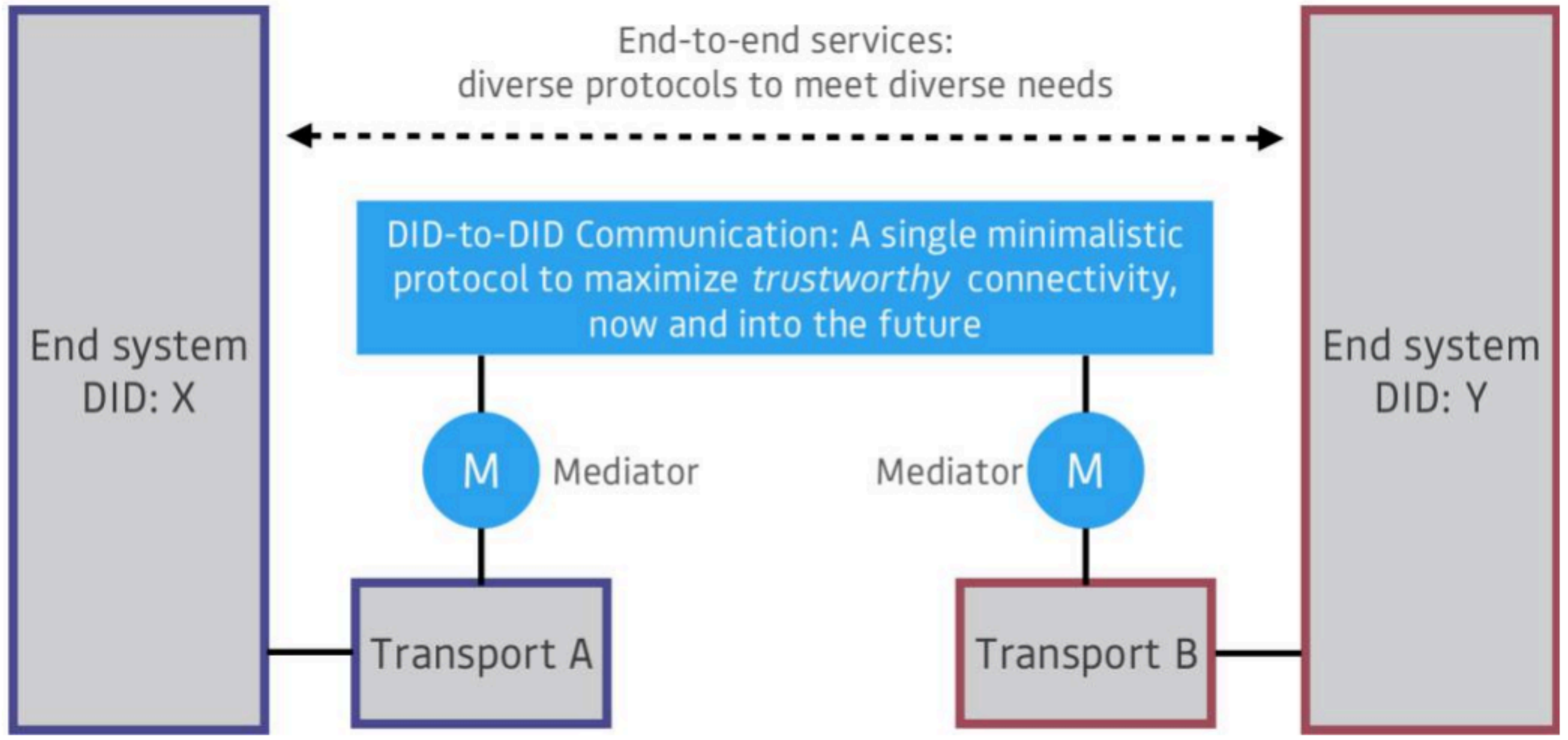

End-to-end services:
diverse protocols to meet diverse needs
End system DID: X
DID-to-DID Communication: A single minimalistic protocol to maximize *trustworthy* connectivity, now and into the future
End system DID: Y
M
Mediator
Mediator
M
Transport A
Transport B

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

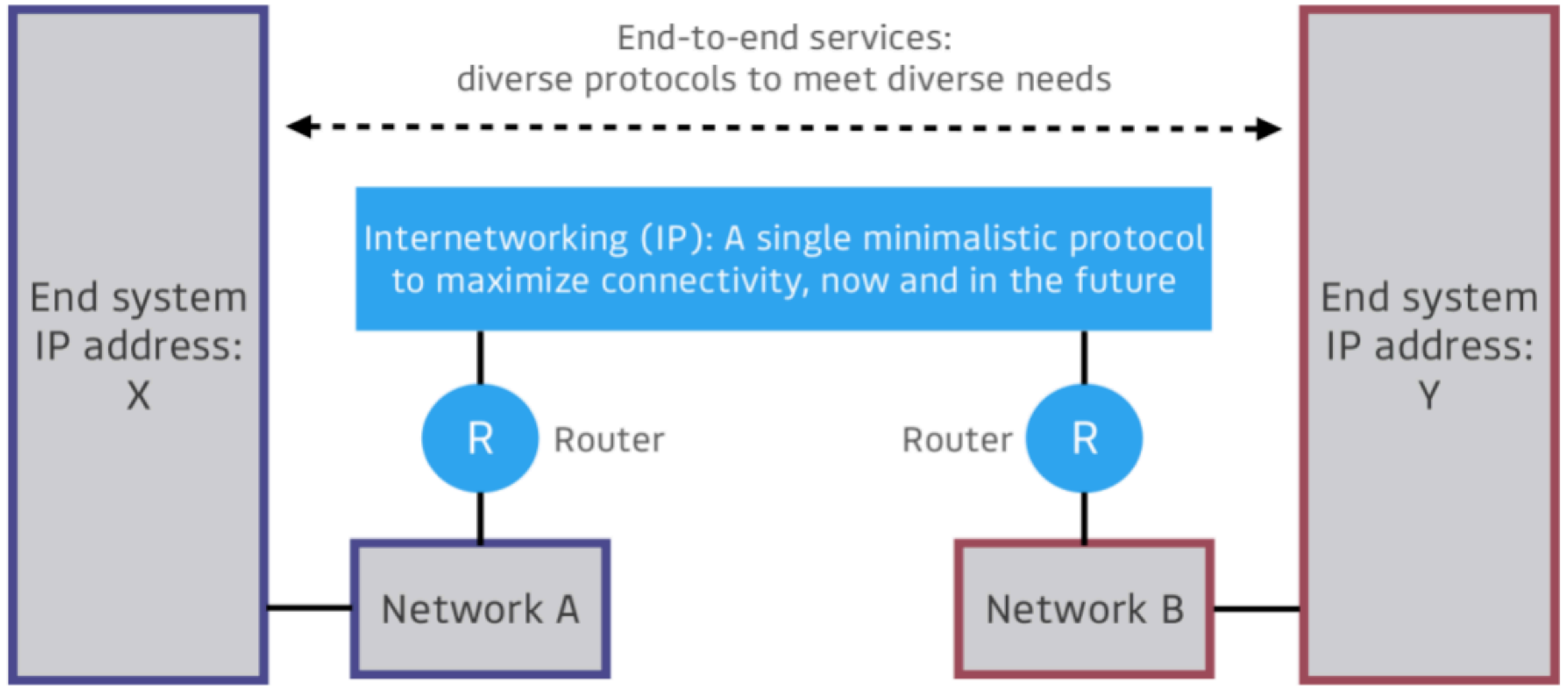
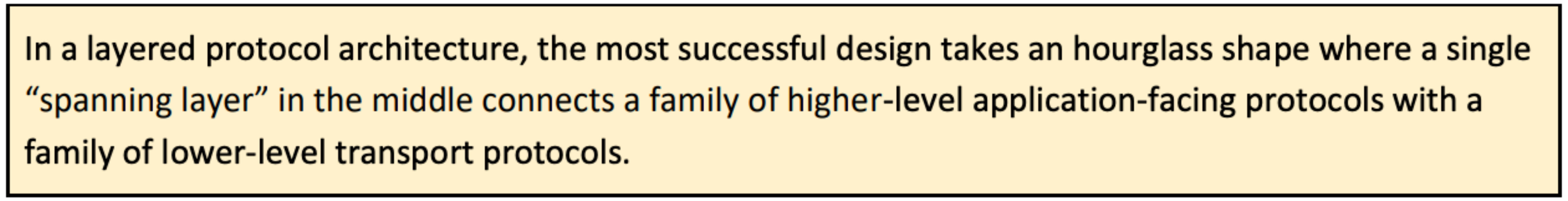
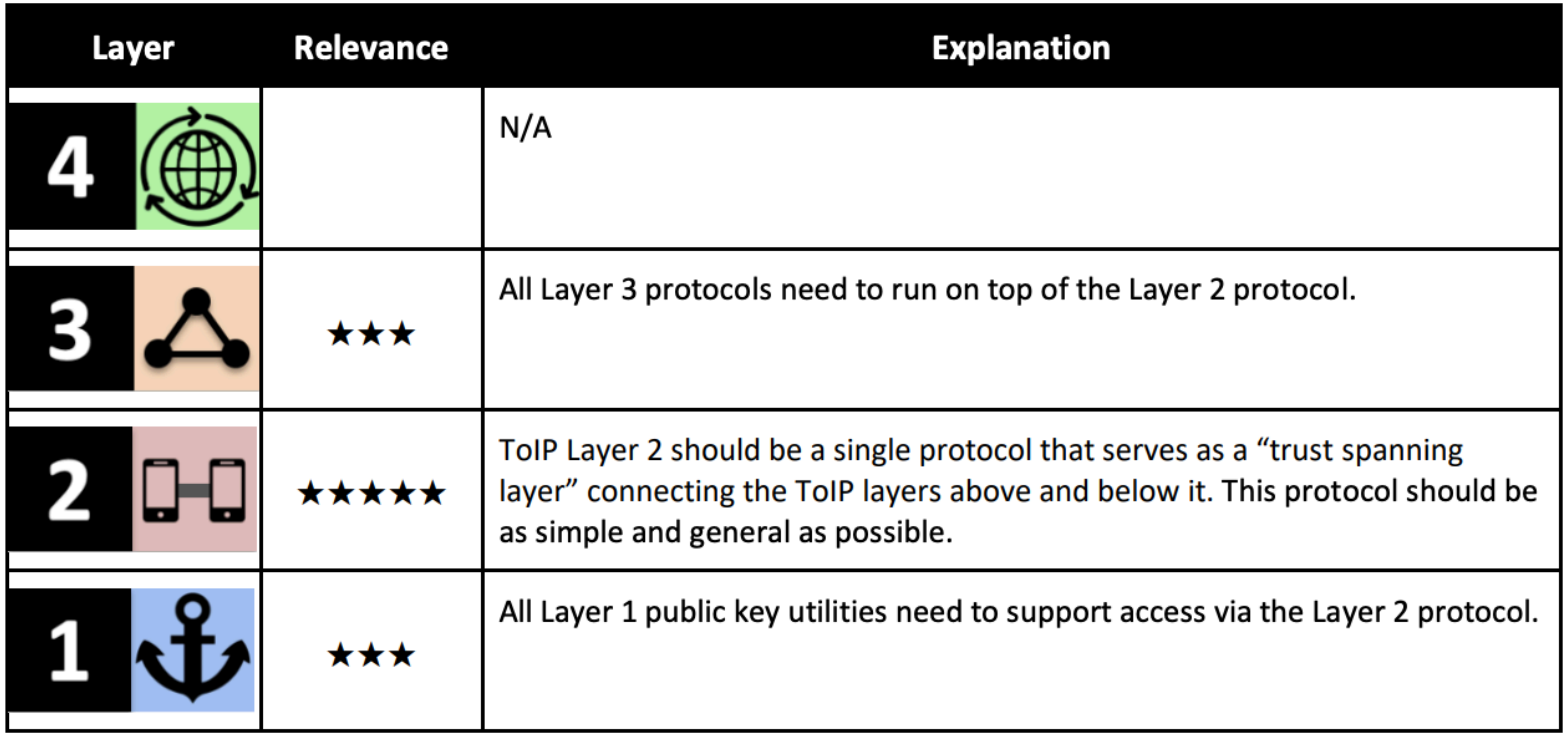

End-to-end services:
diverse protocols to meet diverse needs
End system IP address: X
Internetworking (IP): A single minimalistic protocol to maximize connectivity, now and in the future
End system IP address: Y
R
Router
Router
R
Network A
Network B

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

Design principle #3

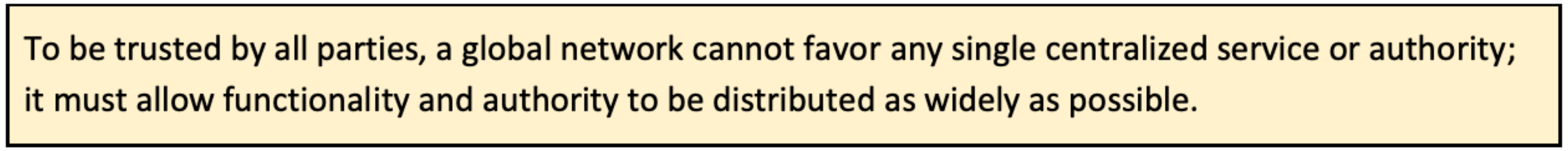

In a layered protocol architecture, the most successful design takes an hourglass shape where a single "spanning layer" in the middle connects a family of higher-level application-facing protocols with a family of lower-level transport protocols.

- The Hourglass Model

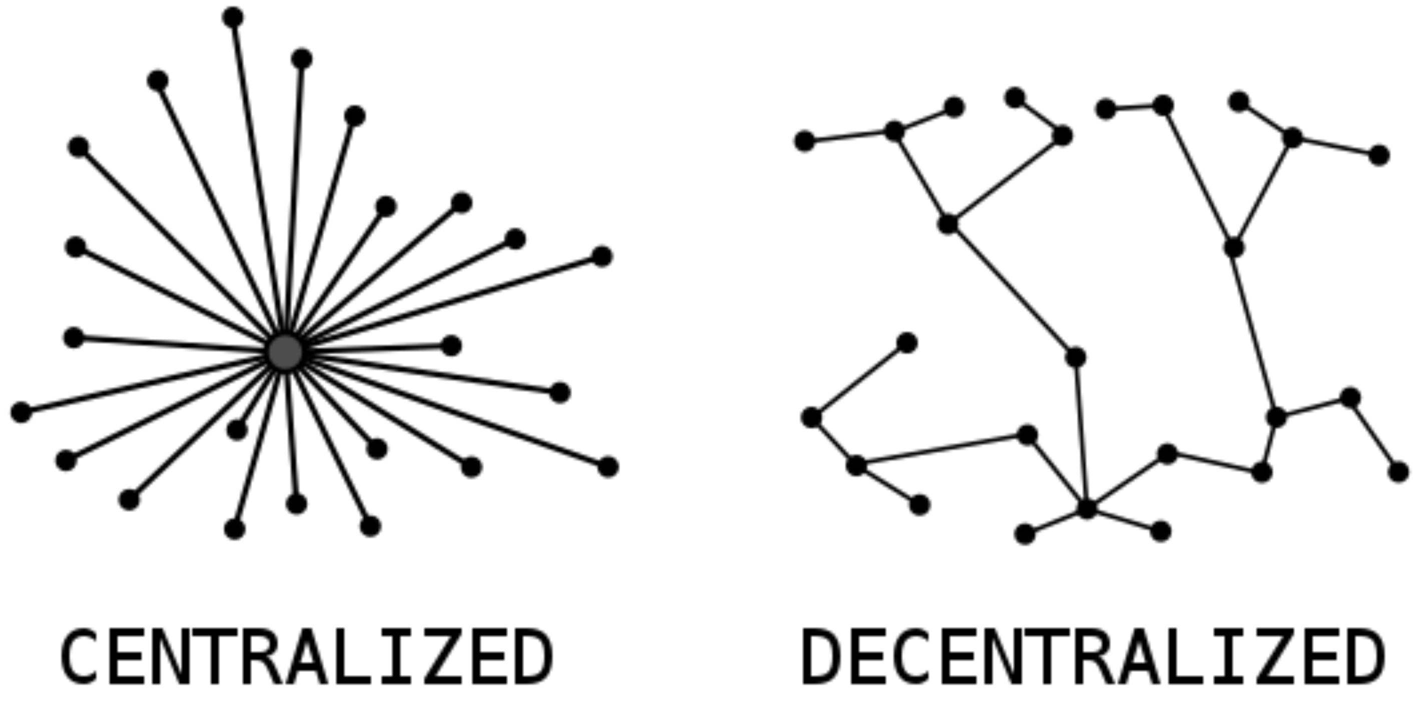


Layer	Relevance	Explanation
4		N/A
3	★★★	All Layer 3 protocols need to run on top of the Layer 2 protocol.
2	★★★★★	ToIP Layer 2 should be a single protocol that serves as a "trust spanning layer" connecting the ToIP layers above and below it. This protocol should be as simple and general as possible.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

Design principle #4


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.

Decentralization by Design and Default

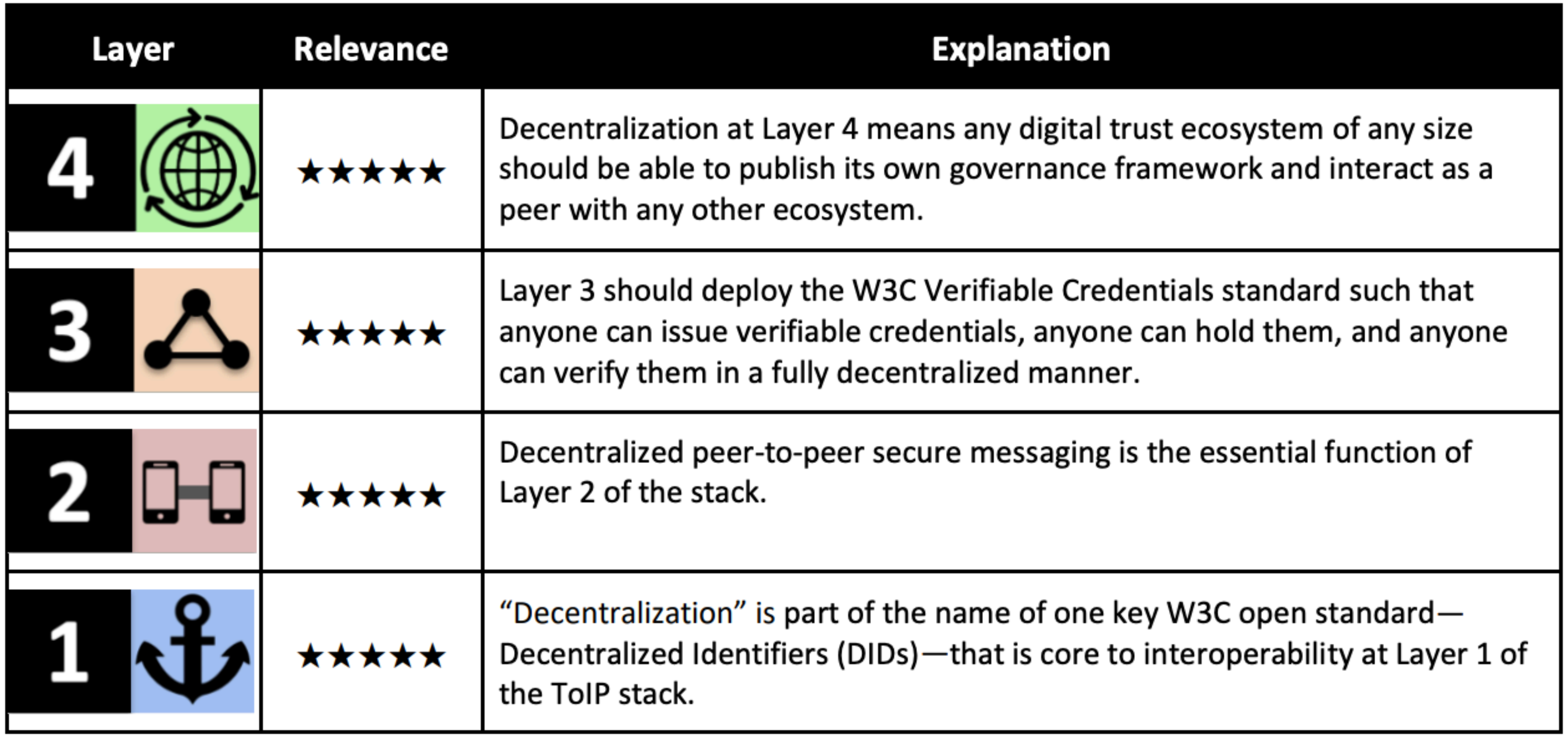


- Internet is decentralized by its original design
  - Reliability to withstand disasters
  - Peer-to-peer connectivity without controlling intermediaries
- … but the benefits of decentralization are eroding
  - The rise of large centralized service architectures where we spend most of our online time
  - They may have high ROI in the short term
  - But they introduce vulnerability, single points of failure
  - Distort communication among peers
  - Open up to market, cilvil liberty and other abuses
- … and that ultimately diminishes our trust in the Internet technology

Design principle #4

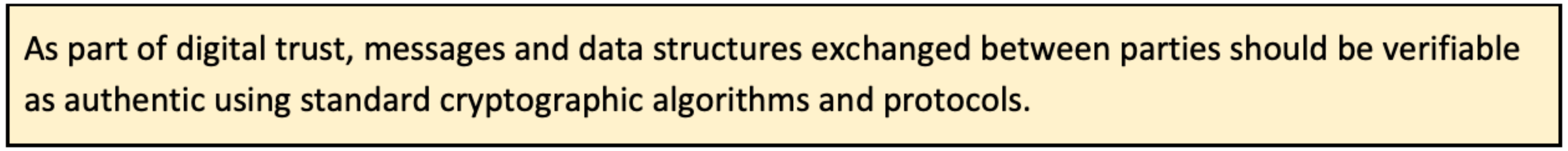
To be trusted by all parties, a global network cannot favor any single centralized service or authority; it must allow functionality and authority to be distributed as widely as possible.

- Decentralization by Design and Default



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.

Design principle #5



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

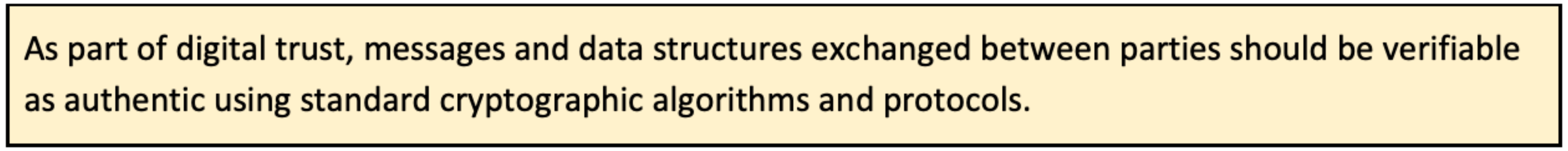
- Cryptographic Verifiability

Authenticity | Integrity

"On the Internet, nobody knows you're a dog."

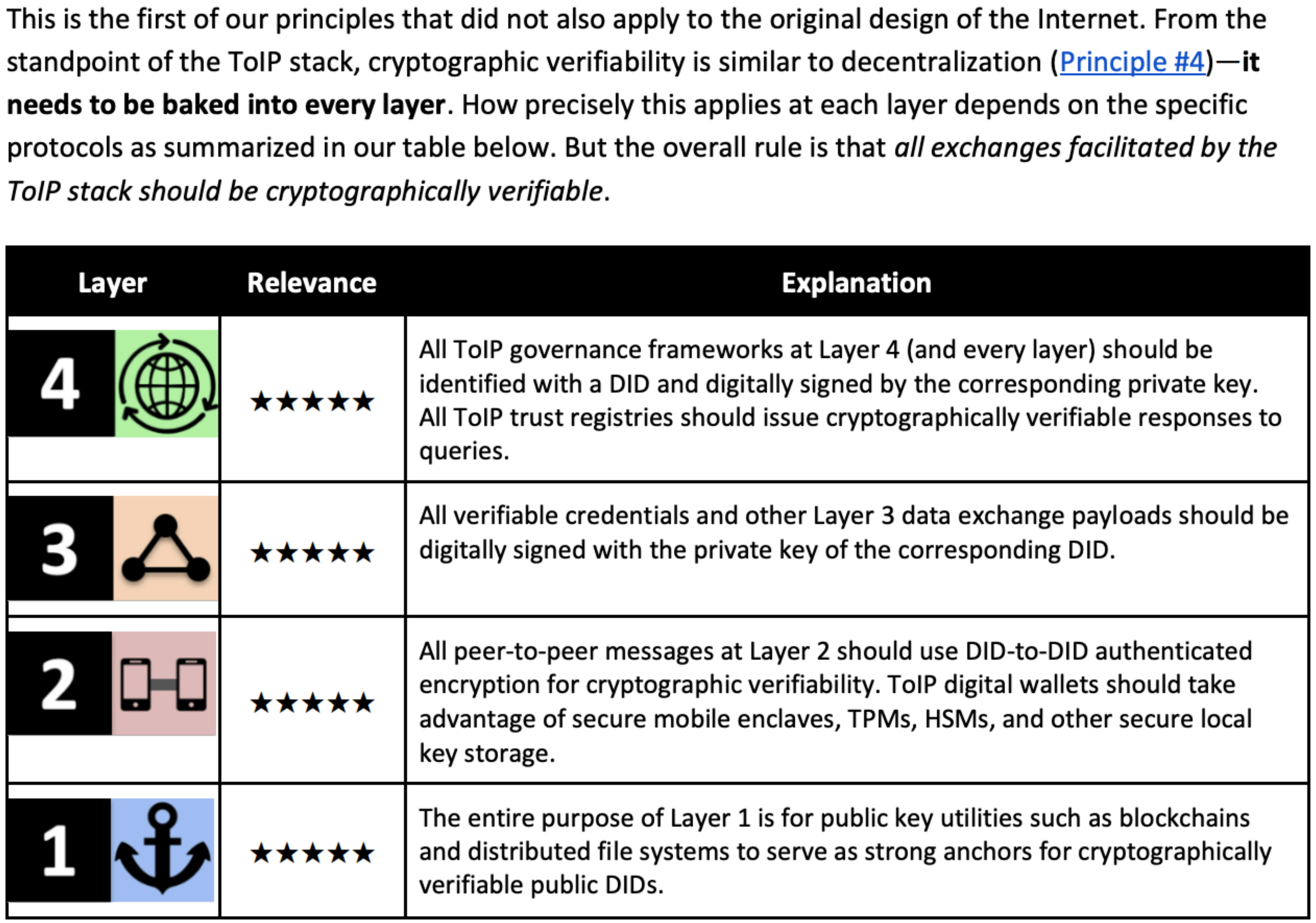
- The original Internet is inherently insecure
- Add-on enhancements are "overlays" for specific types of communications
  - E.g. IPSec, SSL/TLS, DNSSec, each has its own limitations
- And rely on centralized X.509 PKI
  - Has severe limitations for adoption beyond server-centric deployment
  - Shares many ills of a centralized system
- We need new authenticity and integrity capabilities that can be universally applied to common Internet services

Design principle #5



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

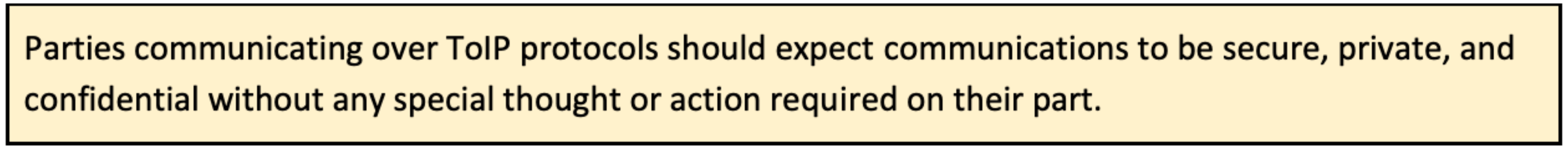
- Cryptographic Verifiability



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*.

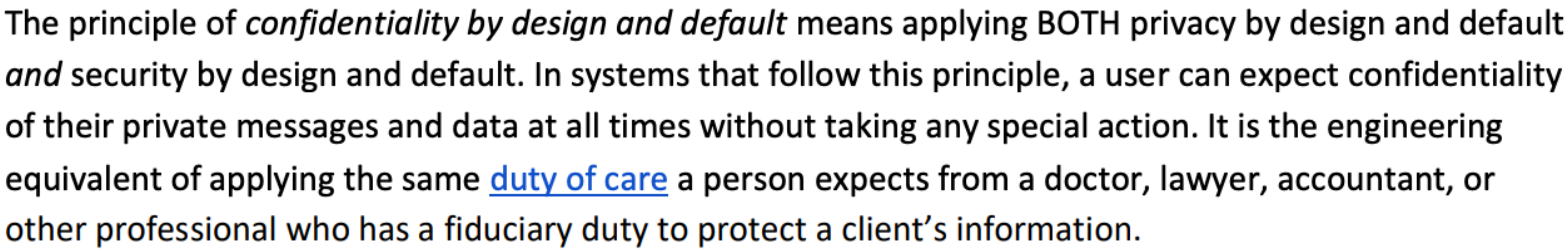
Layer	Relevance	Explanation
4	★★★★★	All ToIP governance frameworks at Layer 4 (and every layer) should be identified with a DID and digitally signed by the corresponding private key. All ToIP trust registries should issue cryptographically verifiable responses to queries.
3	★★★★★	All verifiable credentials 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.

Design principle #6



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

- Confidentiality by Design and Default

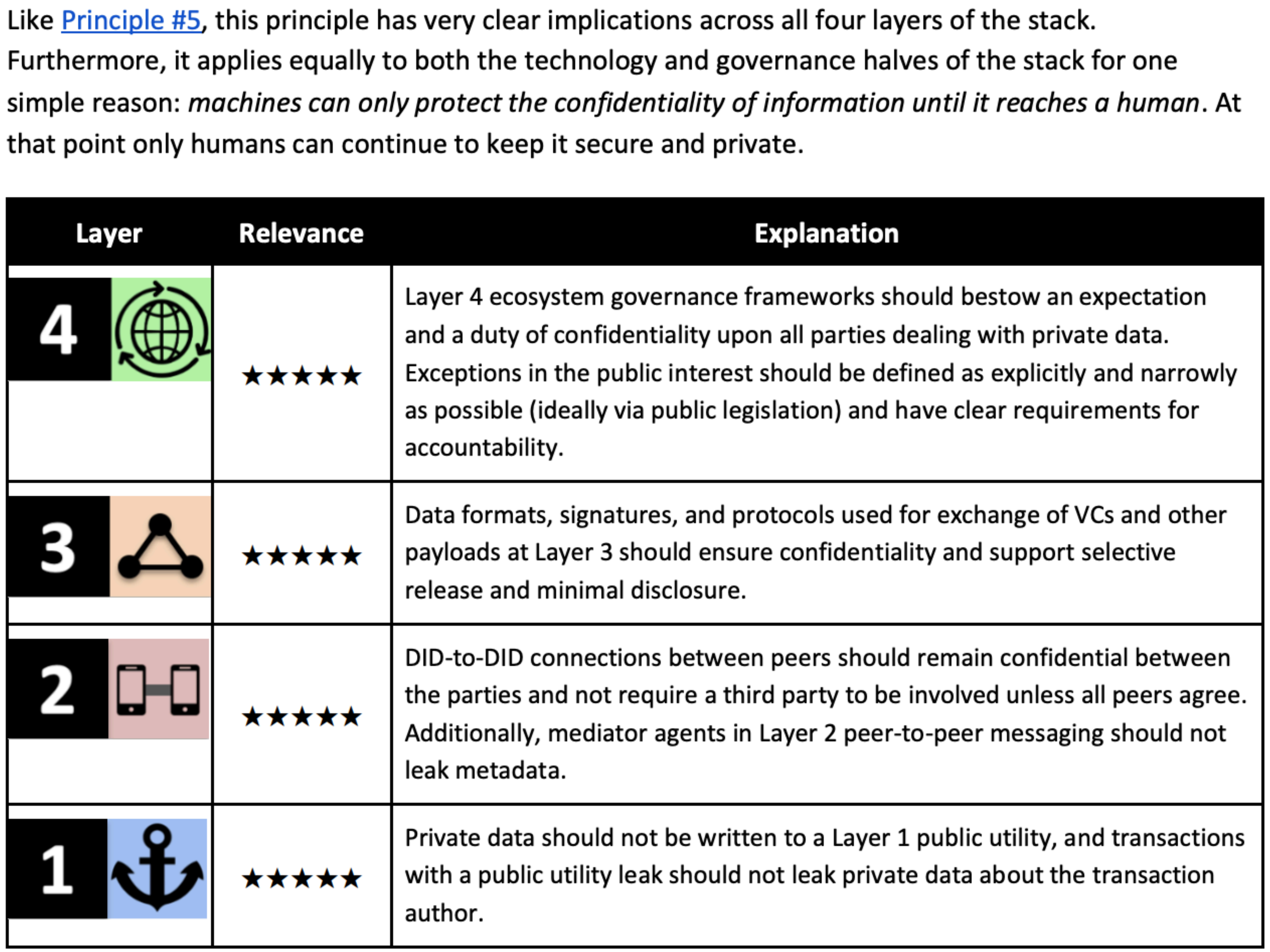


The principle of *confidentiality by design and default* means applying BOTH privacy by design and default *and* security by design and default. In systems that follow this principle, a user can expect confidentiality of their private messages and data at all times without taking any special action. 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.

Design principle #6

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

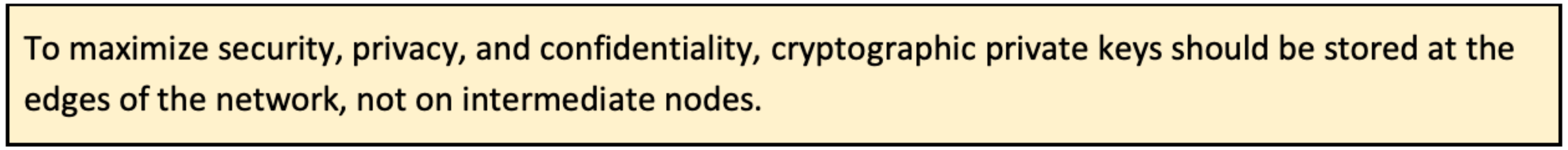
- Confidentiality by Design and Default



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.

Design principle #7



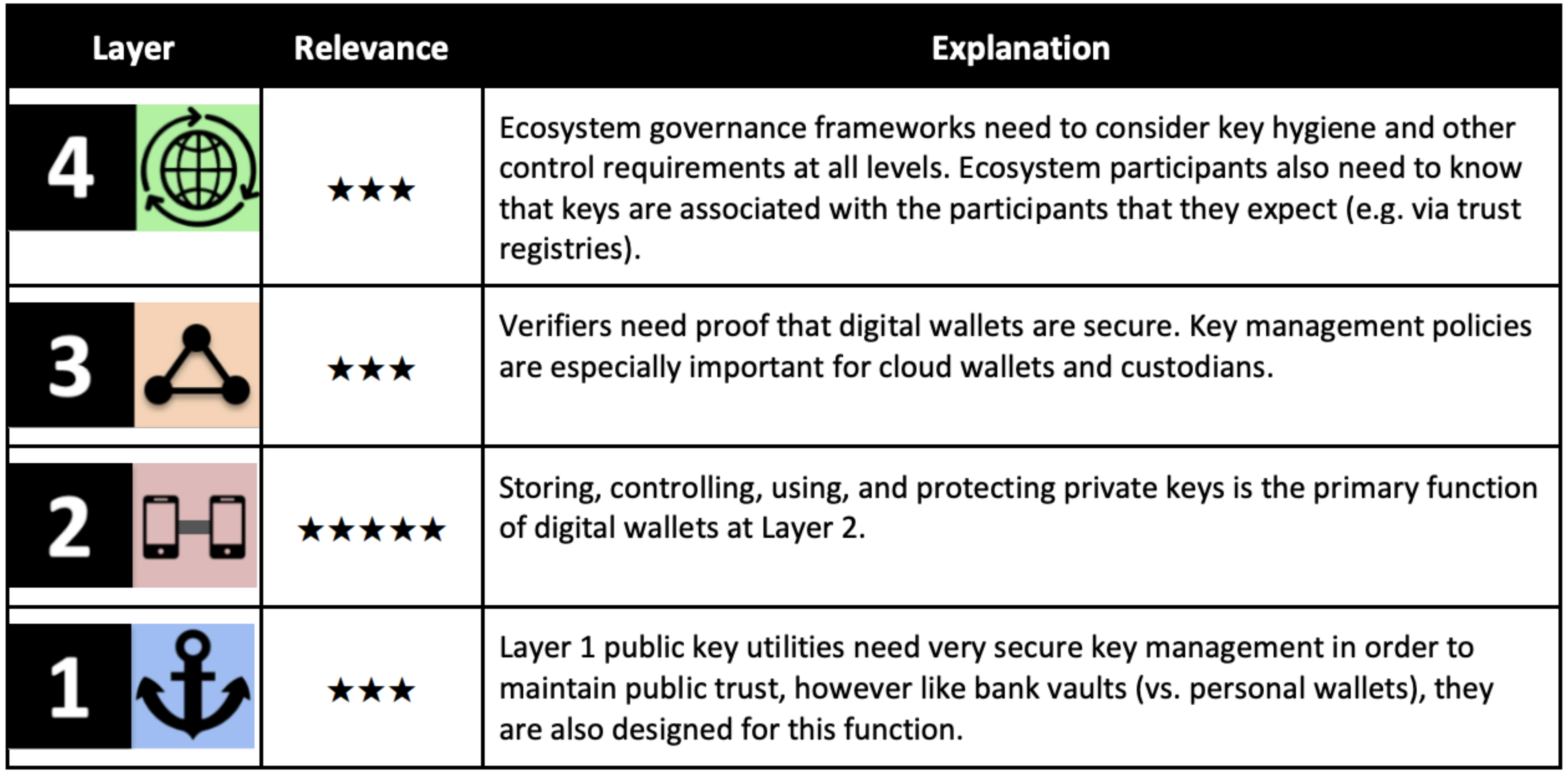
To maximize security, privacy, and confidentiality, cryptographic private keys should be stored at the edges of the network, not on intermediate nodes.

- Keys at the Edge
  - From the design principles of
    - End-to-End Principle
    - Cryptographic Verifiability
    - Confidentiality by Design and Default
  - We can logically conclude that keys should be kept at the edge
  - Edge Wallets vs. Cloud Wallets

Design principle #7

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

- Keys at the Edge



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.

Design principle #8

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

- Trust is Human

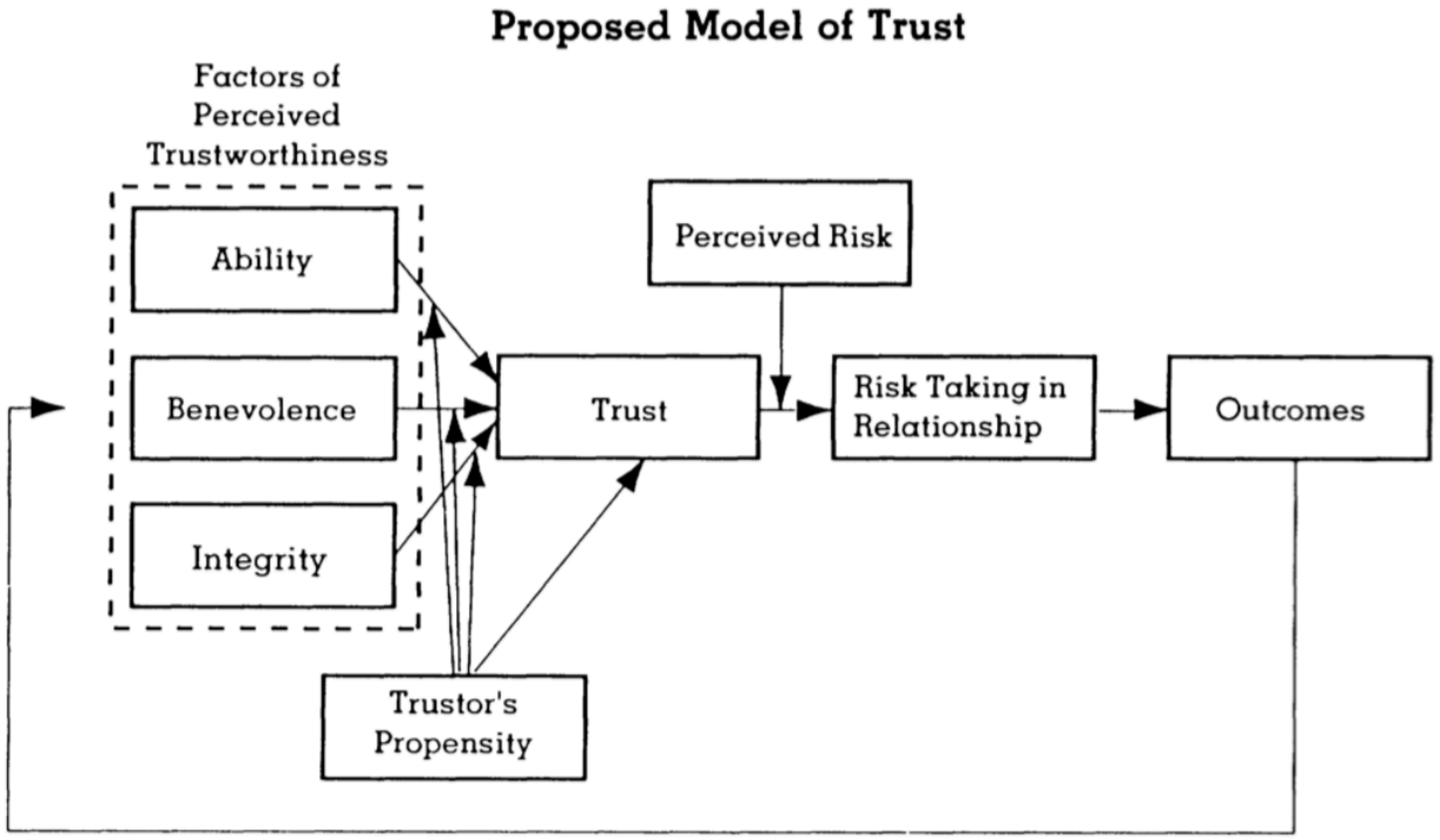
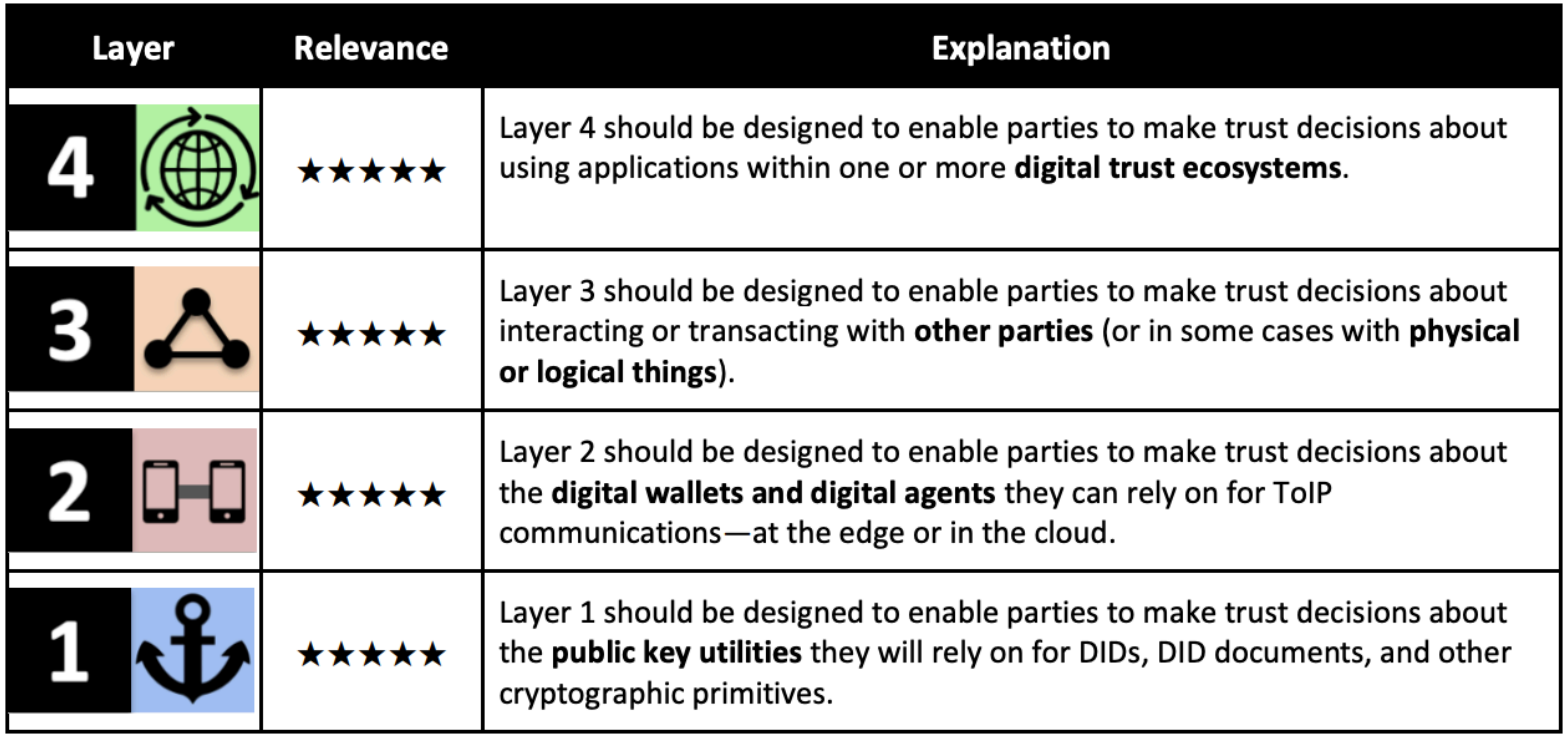


Figure 13. A model of organizational trust

Design principle #8

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

- Trust is Human



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.

Design principle #9

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.

- Trust is Relational

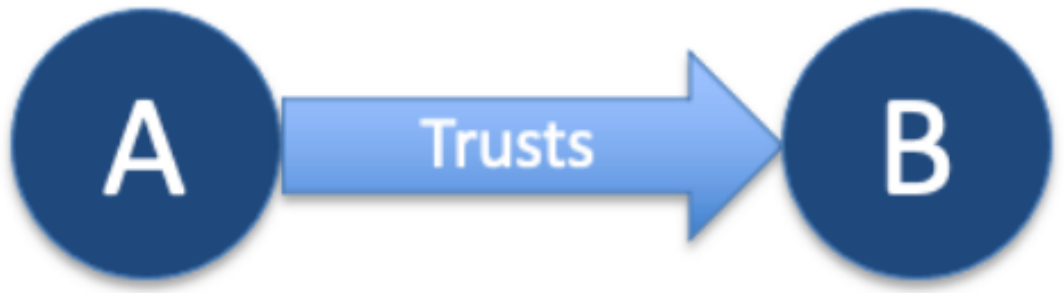


Figure 14. Trust is always between a subject and an object

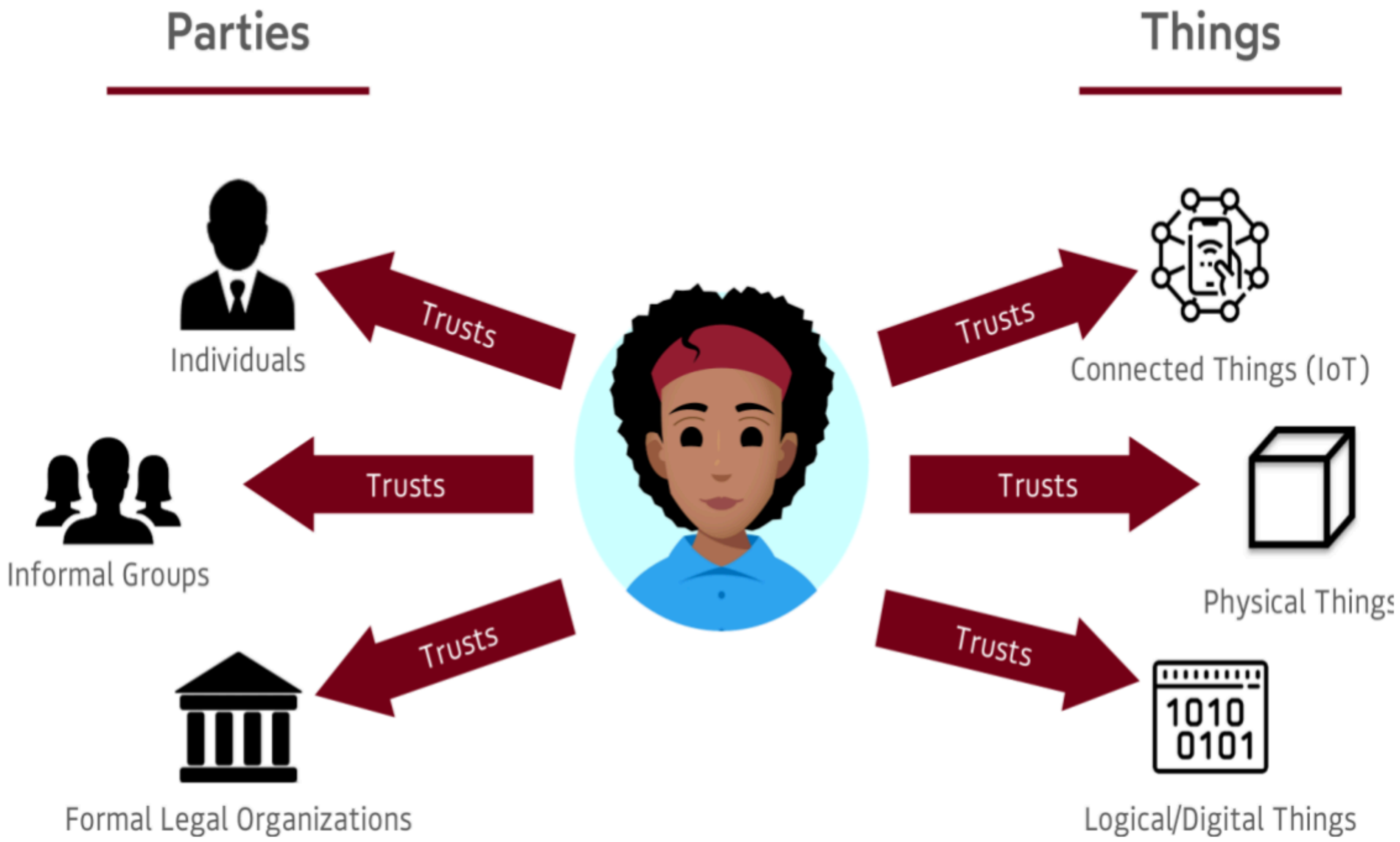
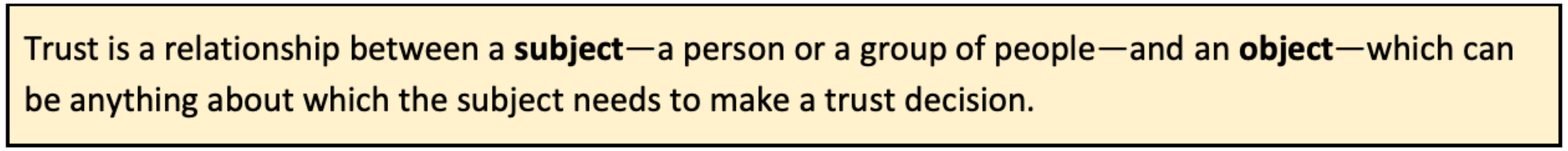


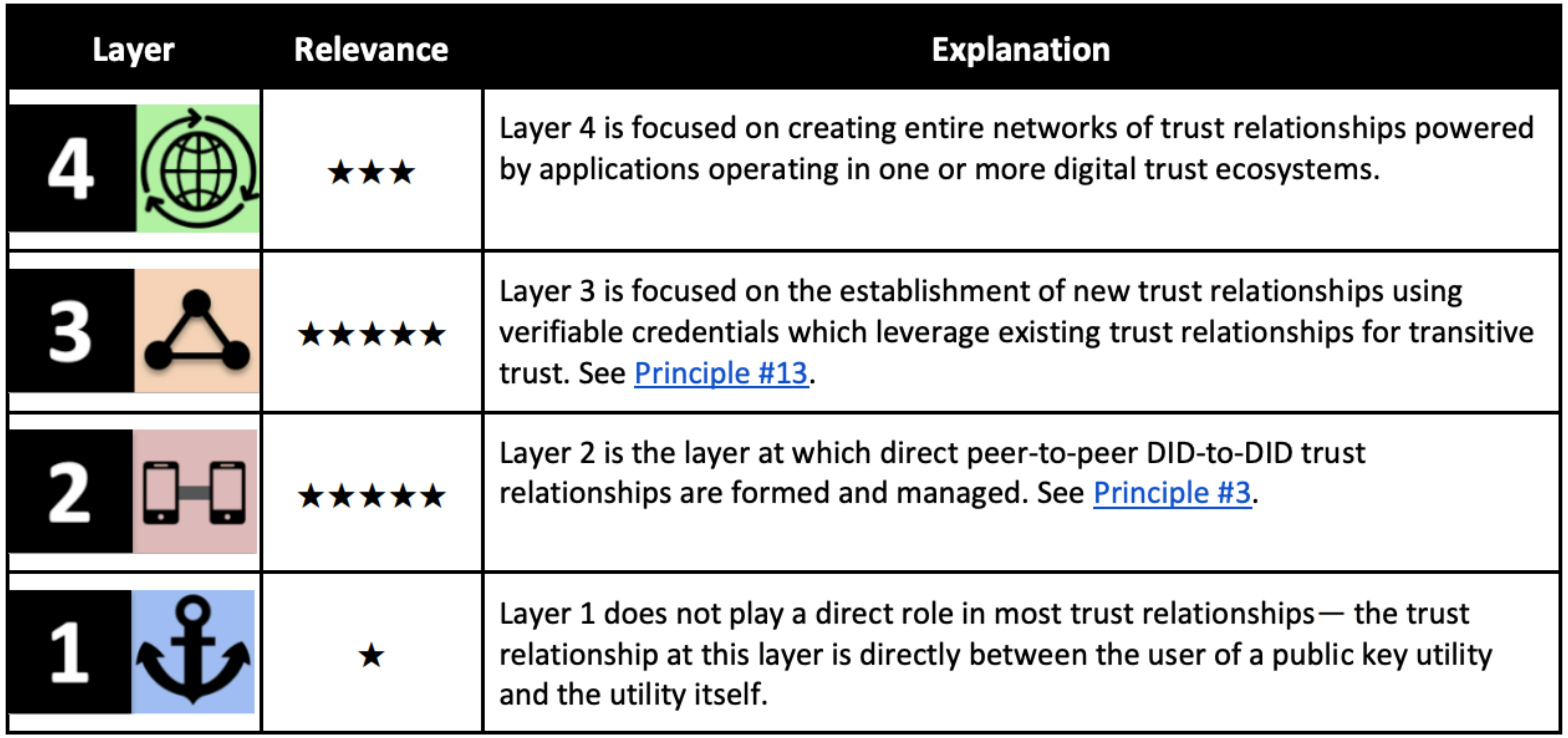
Figure 15. The object of a trust relationship can be anything about which the subject needs to make a trust decision

Design principle #9



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.

- Trust is Relational



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.

Design principle #10

> 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.

- Trust is Directional

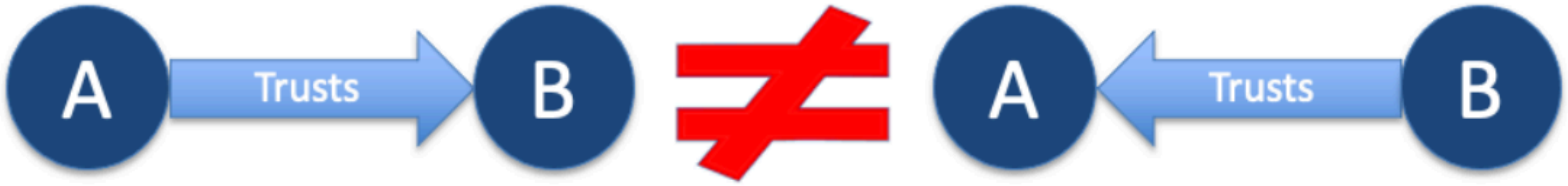
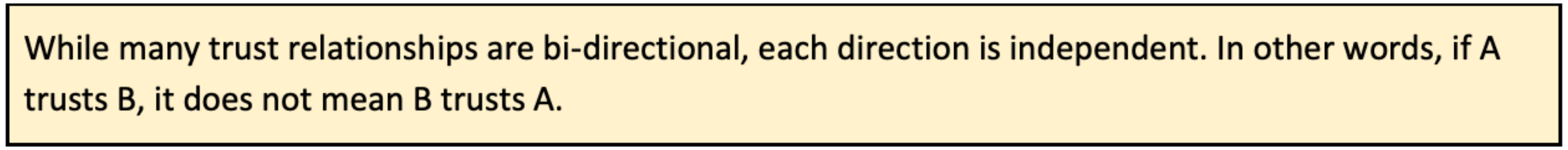


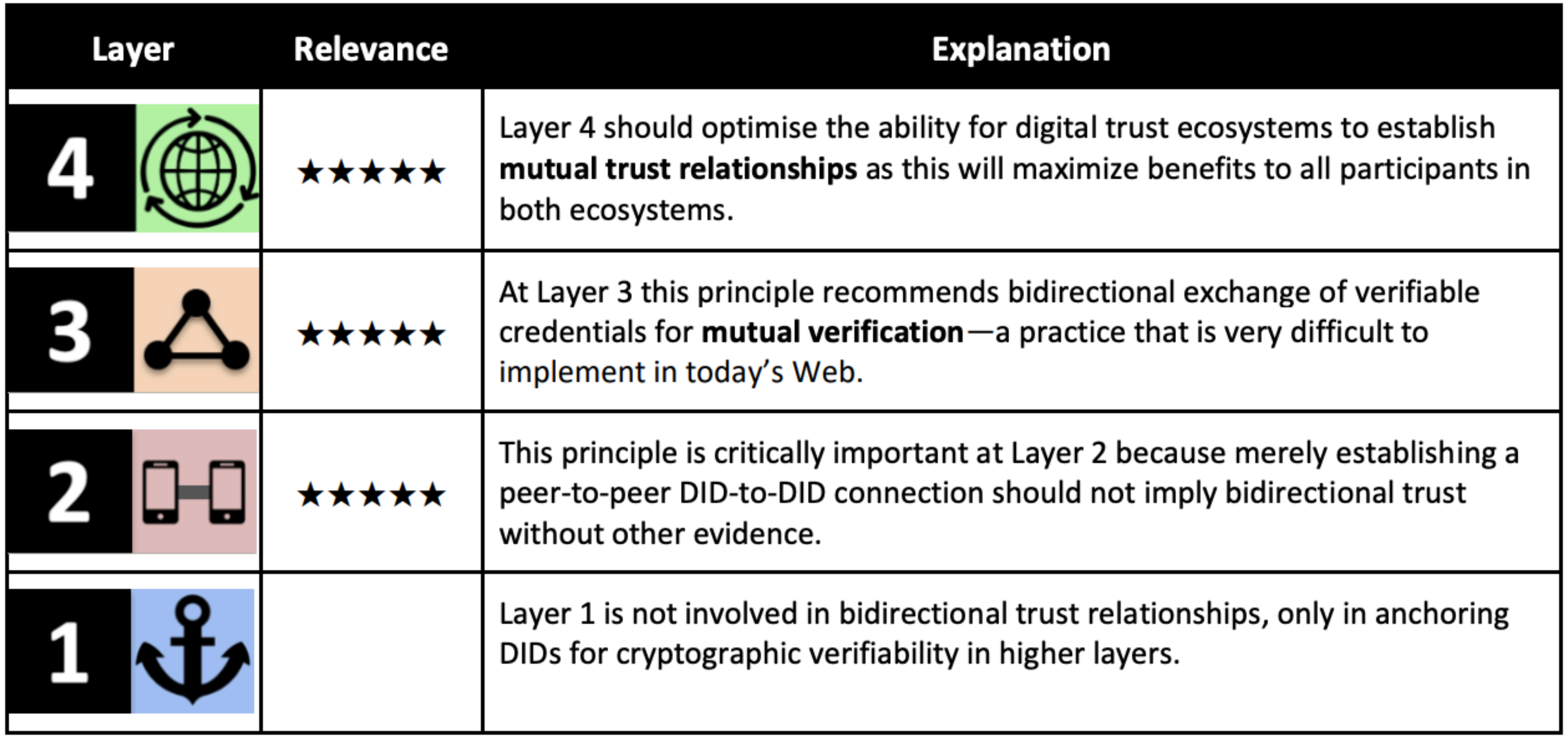
Figure 16. A single trust relationship is always unidirectional. A trust relationship MAY be bidirectional, but if so, it is composed of two unidirectional trust relationships

Design principle #10



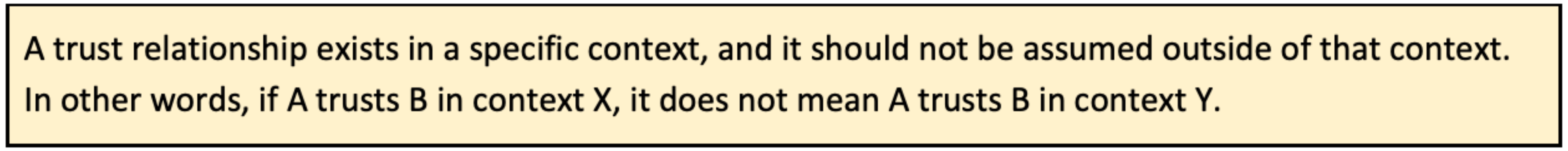
While many trust relationships are bi-directional, each direction is independent. In other words, if A trusts B, it does not mean B trusts A.

- Trust is Directional



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.

Design principle #11



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.

- Trust is Contextual

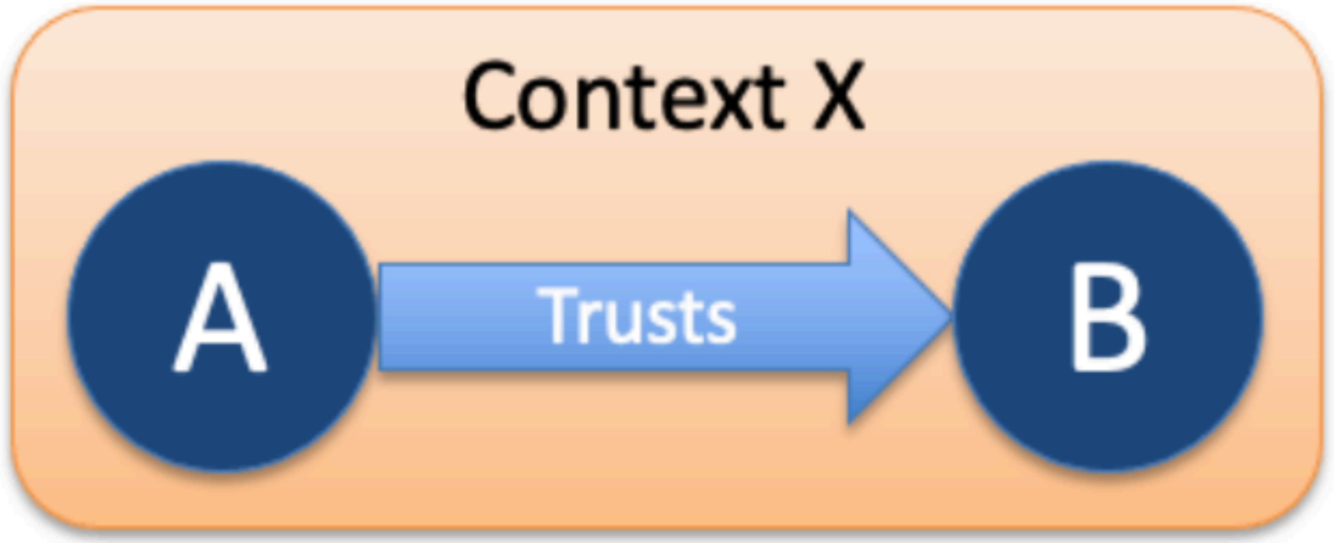
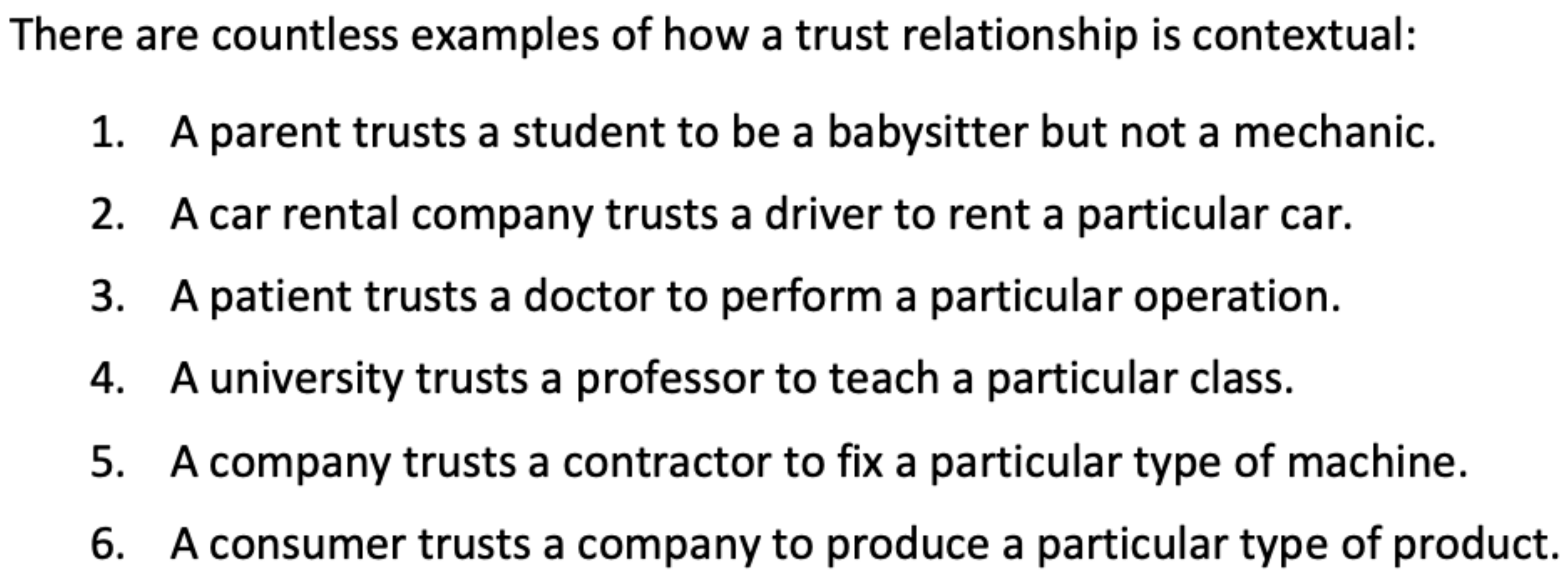


Figure 17. A trust relationship always exists in a specific context



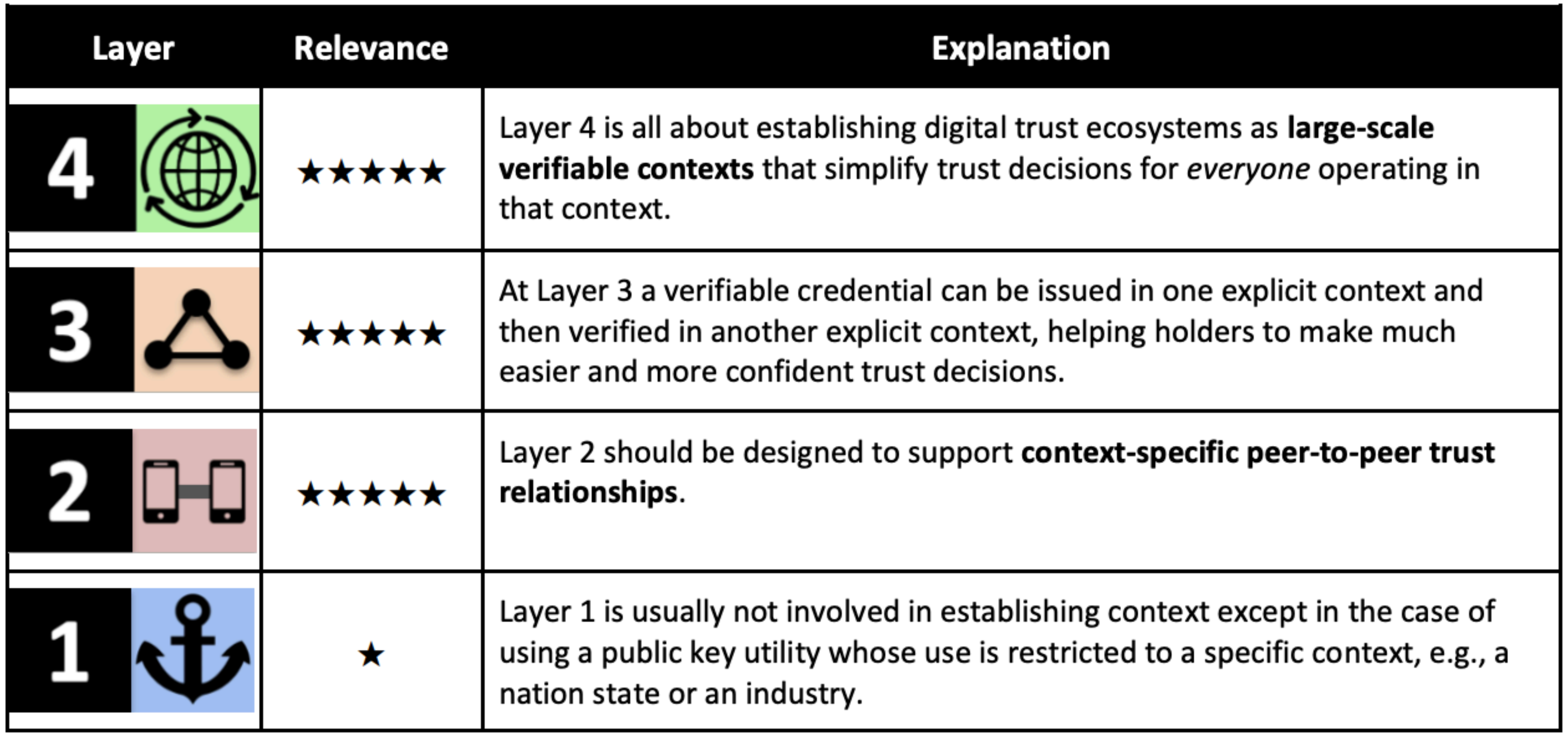
There are countless examples of how a trust relationship is contextual:

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.

Design principle #11

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.

- Trust is Contextual



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.

Design principle #12

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 has Limits

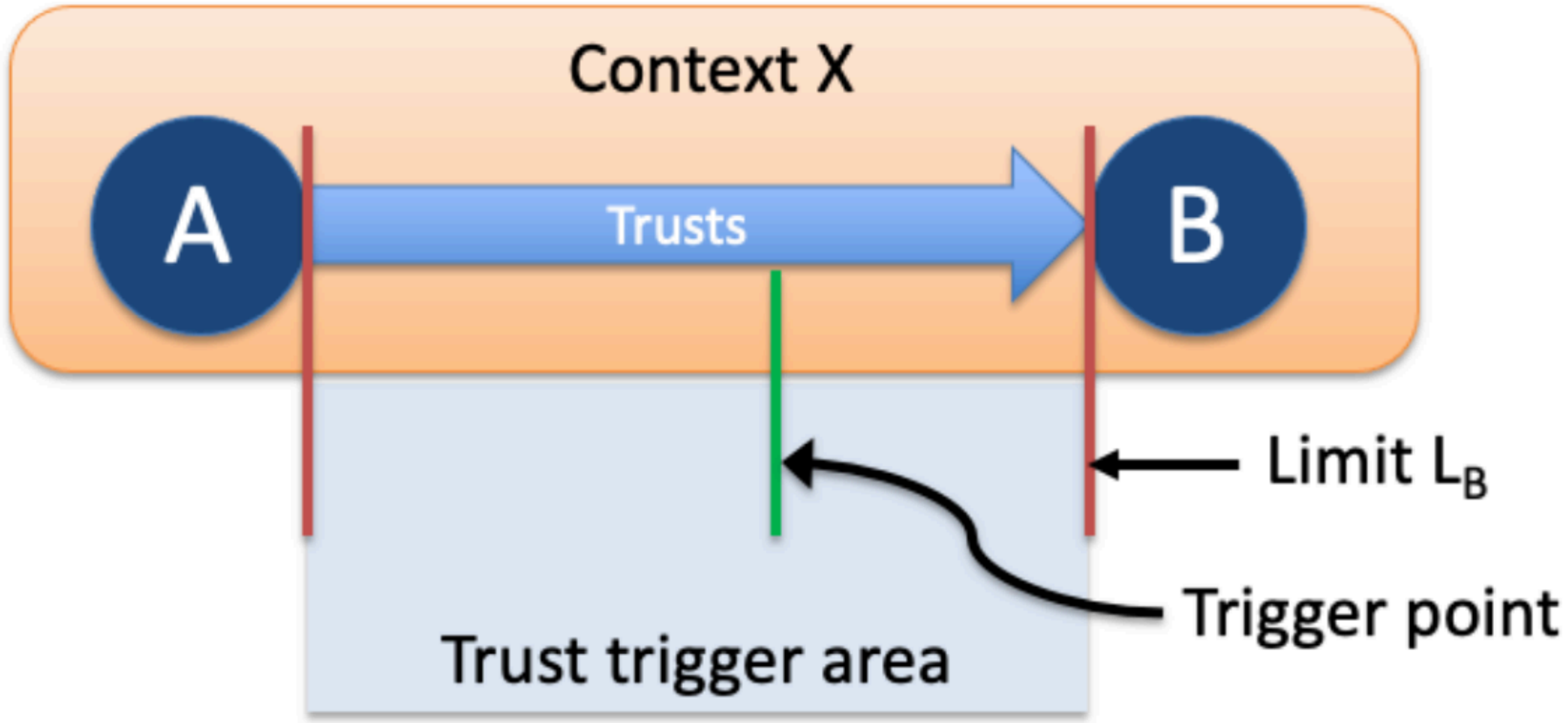


Figure 18. A trust relationship has a range of trigger points for making a trust decision up to a limit determined by the subject

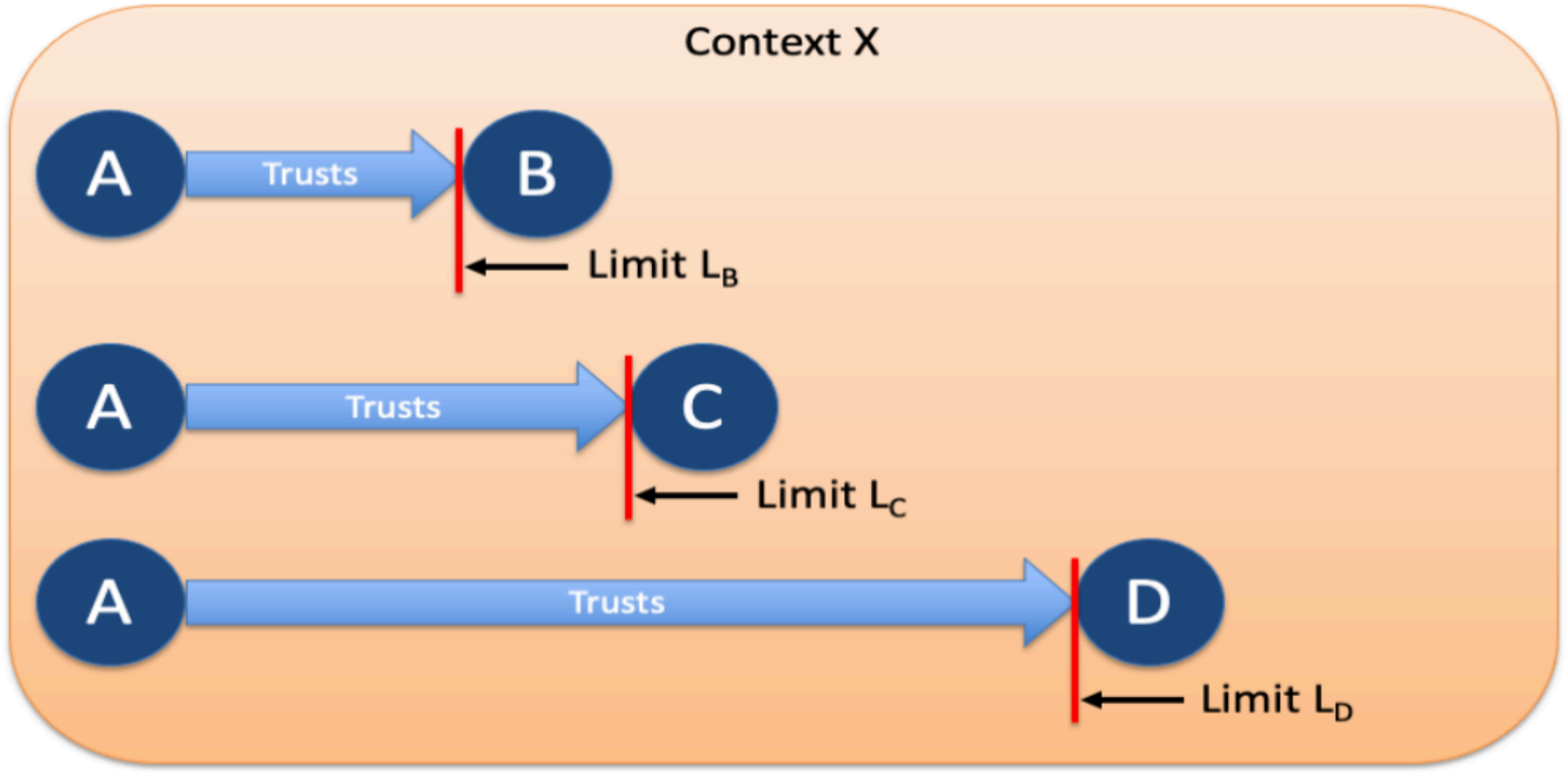
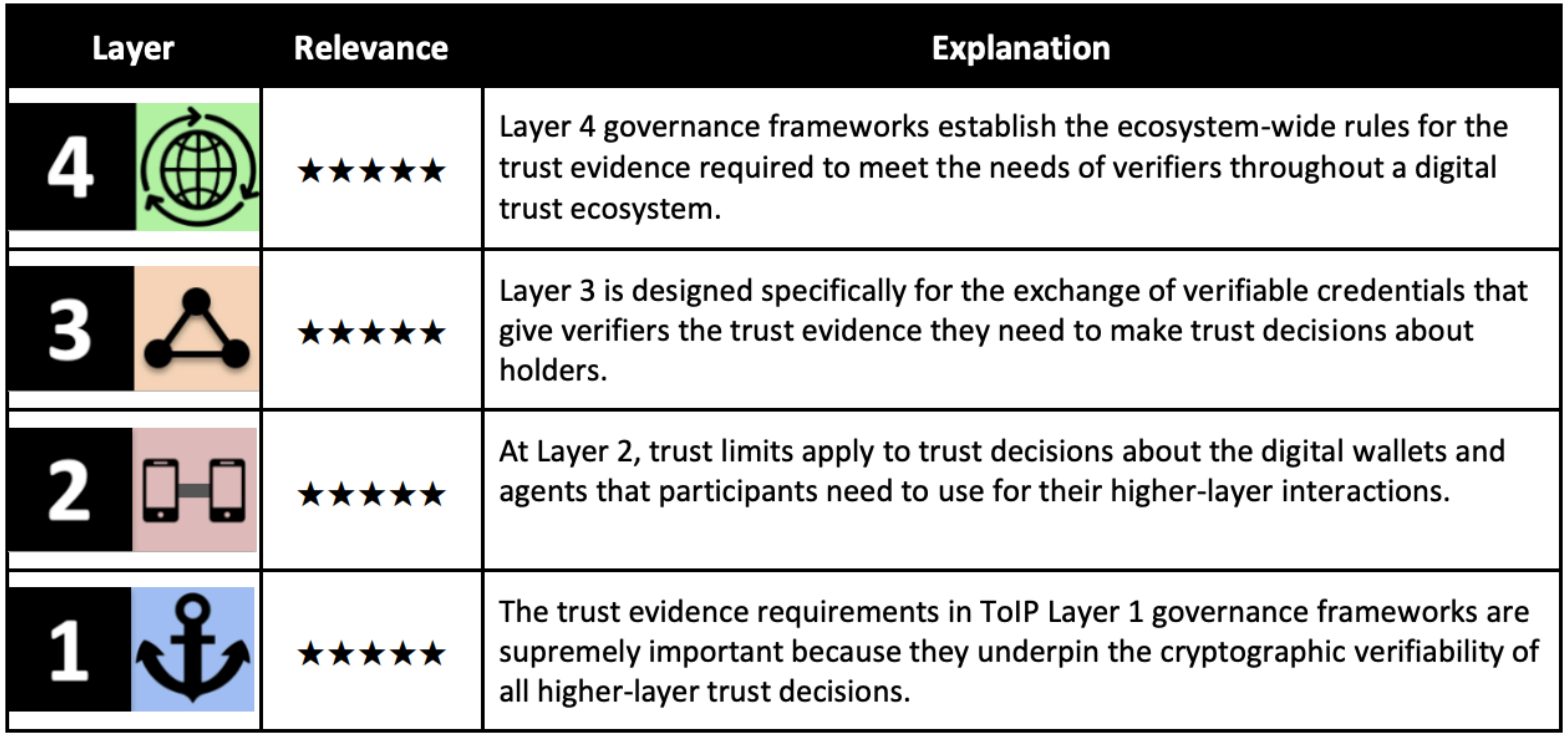


Figure 19. A can have different limits of trust in B, C, and D in the same context X

Design principle #12

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 has Limits



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.

Design principle #13

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.

- Trust can be Transitive

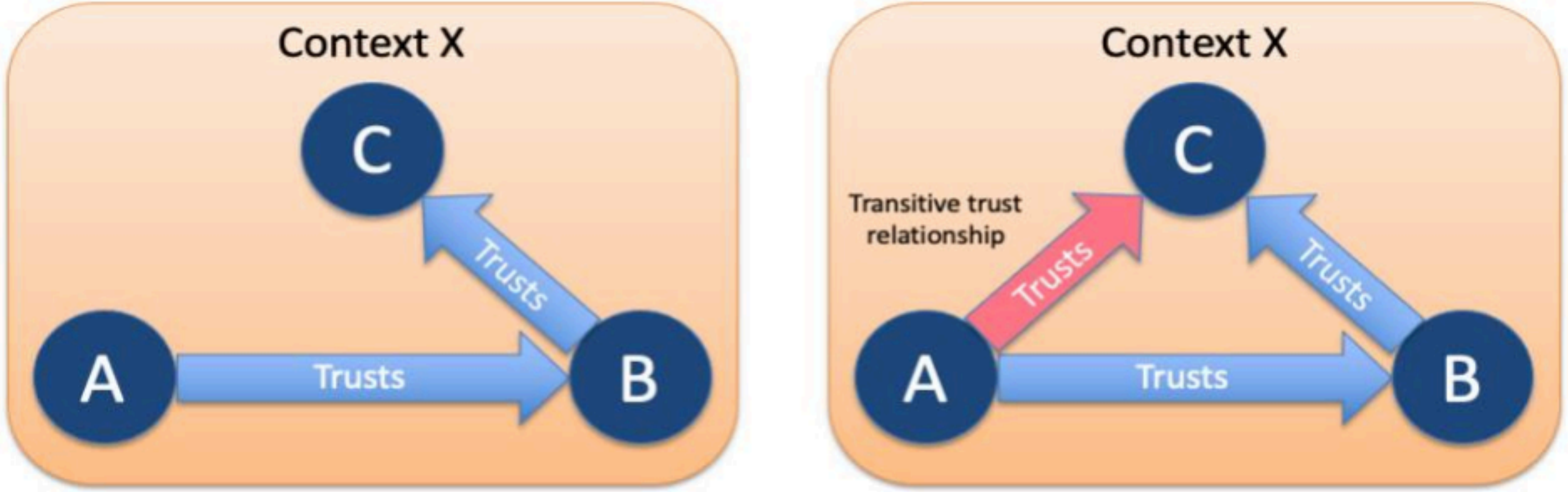


Figure 21. The "trust triangle". If A trusts B within context X and B trusts C within context X (left), then A can have transitive trust in C within context X (right)

Design principle #13

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.

- Trust can be Transitive

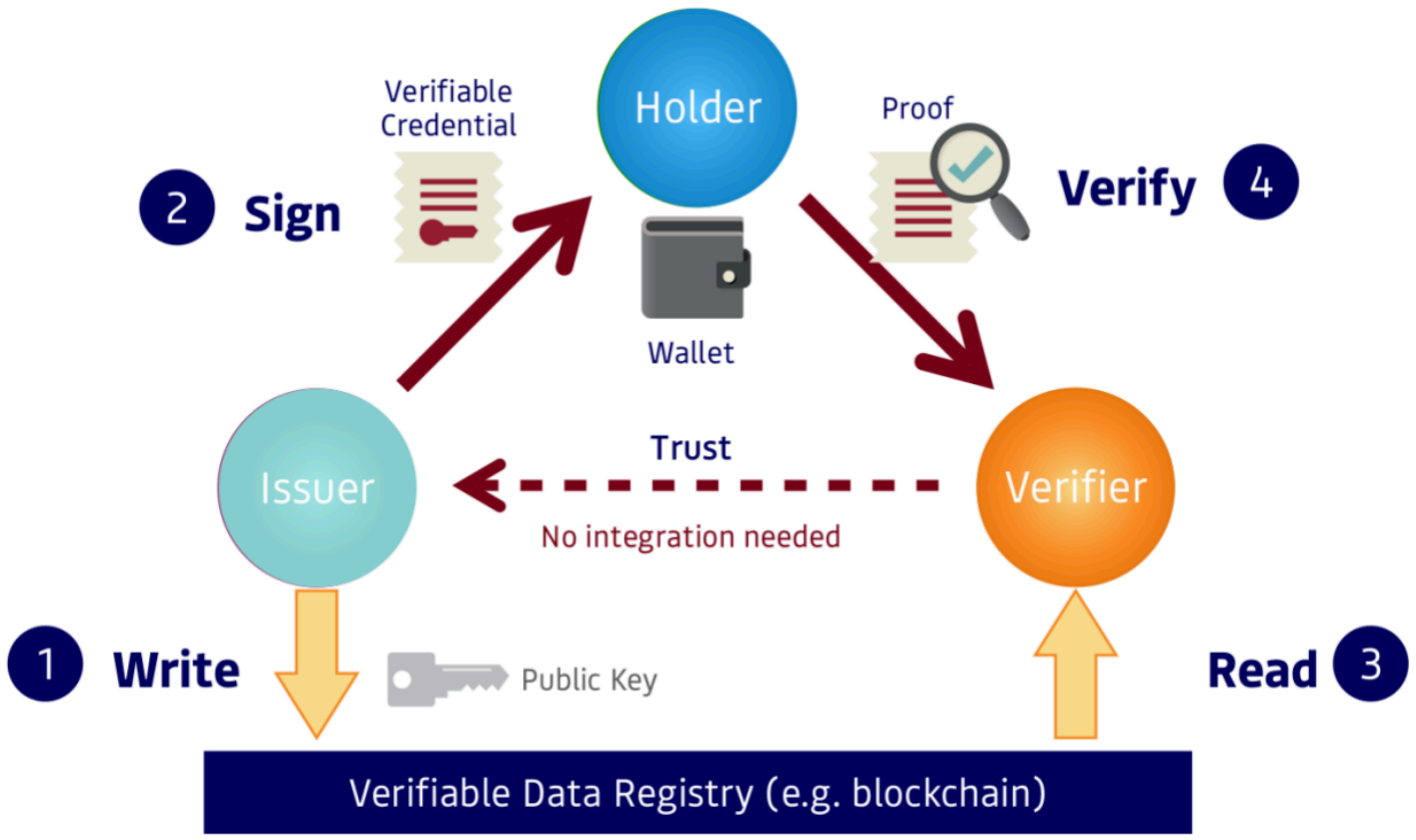


Figure 22. The verifiable credential trust triangle implemented at ToIP Layer 3

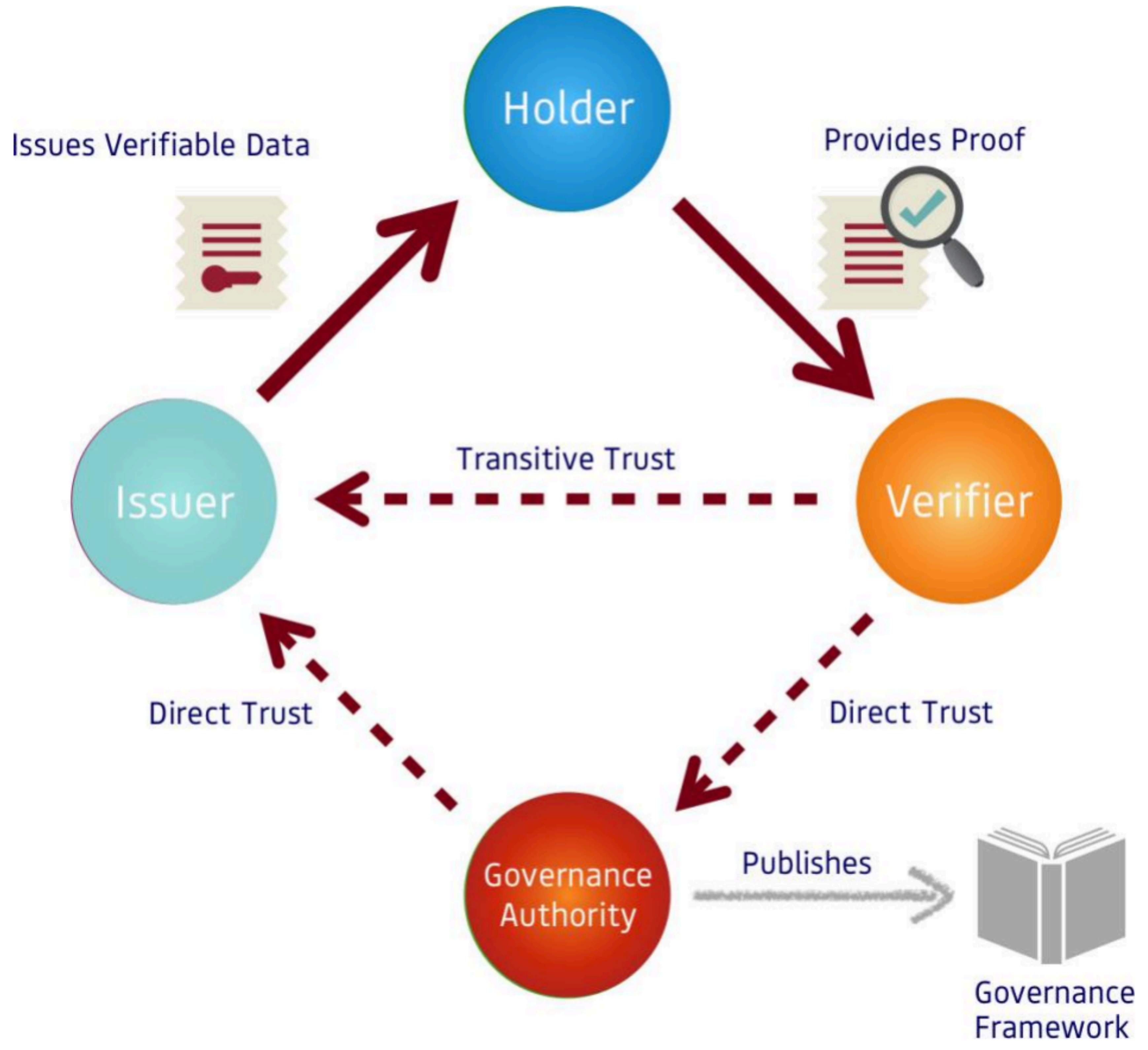
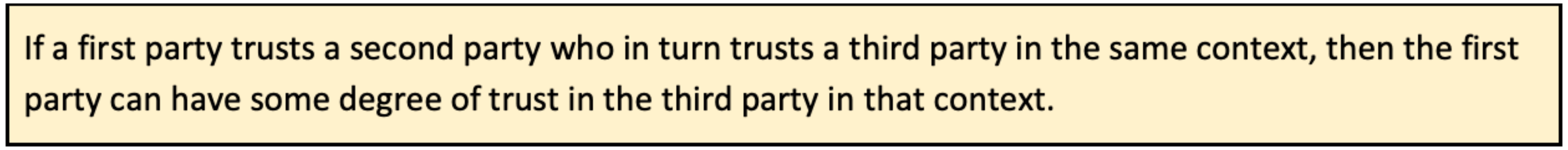


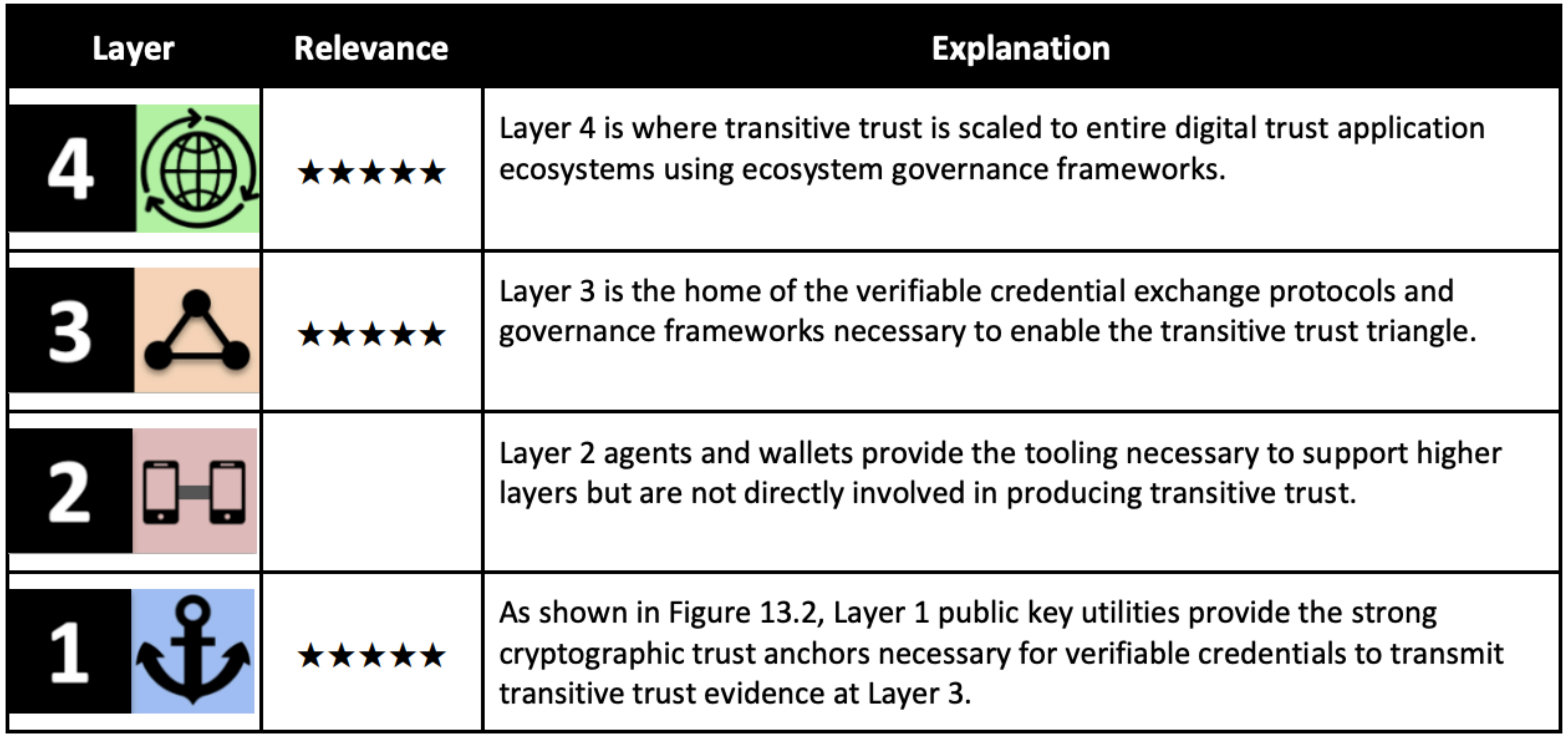
Figure 24. The ToIP governance trust diamond

Design principle #13



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.

- Trust can be Transitive



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.

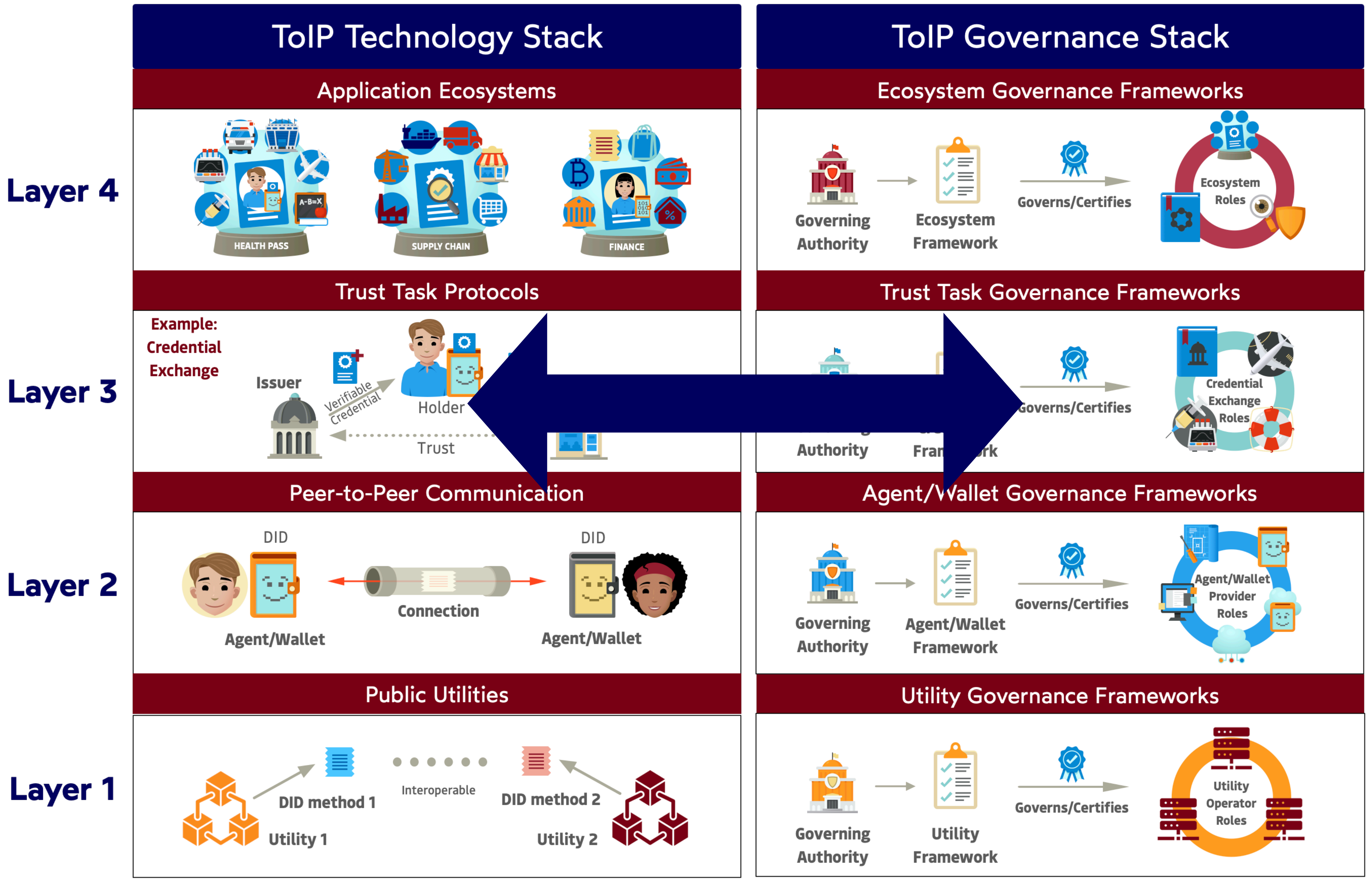
Design principle #14

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

• Trust and Technology has Reciprocal Relationship

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]

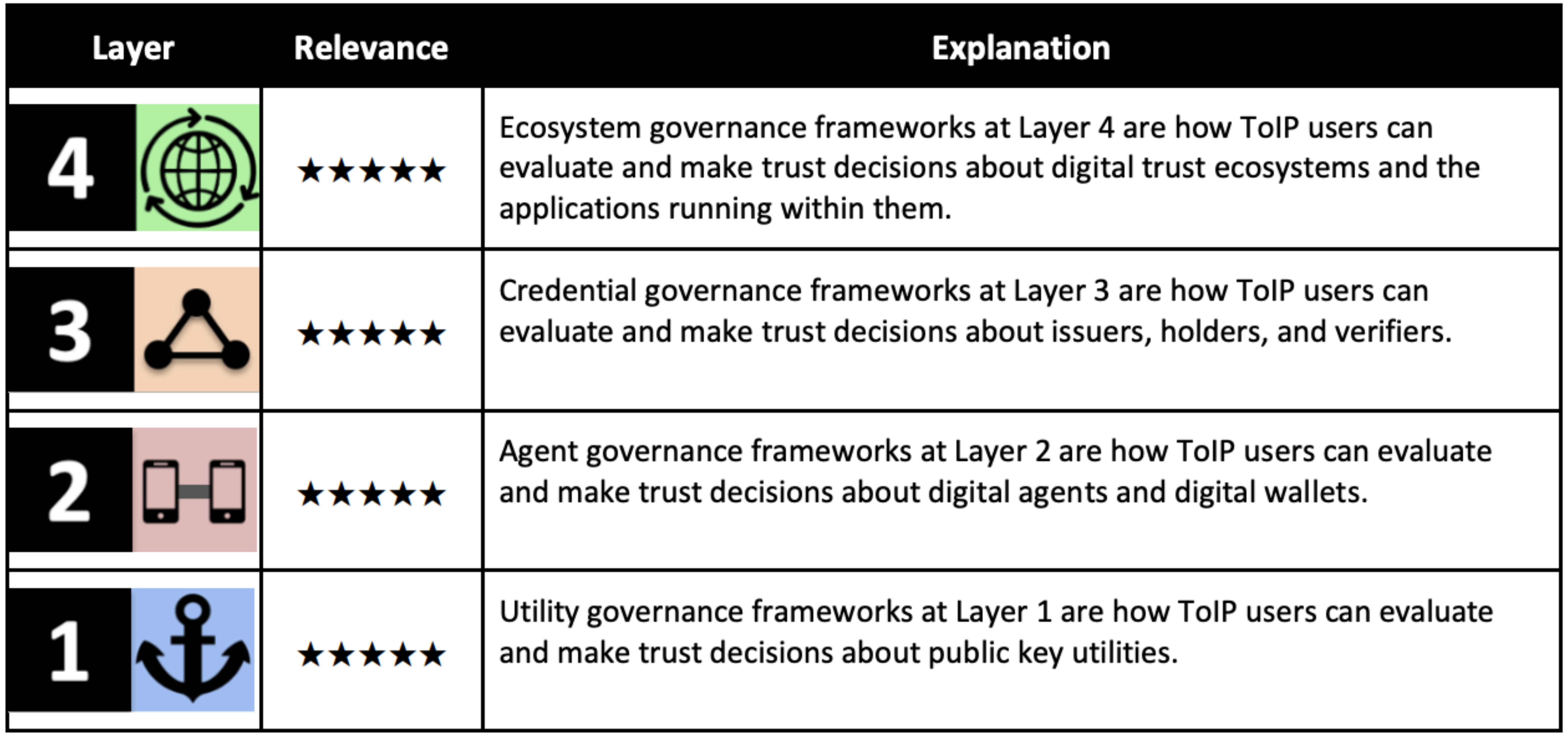


	ToIP Technology Stack	ToIP Governance Stack
Layer 4	Application Ecosystems	Ecosystem Governance Frameworks
Layer 3	Trust Task Protocols	Trust Task Governance Frameworks
Layer 2	Peer-to-Peer Communication	Agent/Wallet Governance Frameworks
Layer 1	Public Utilities	Utility Governance Frameworks

Design principle #14

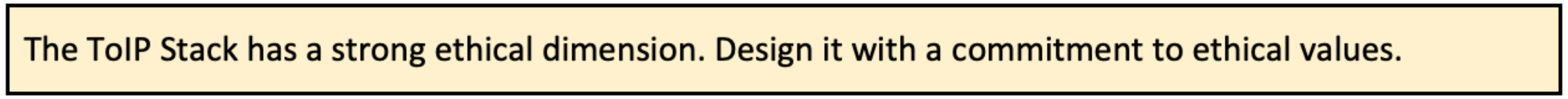
Technology can only help humans build trust if humans trust the technology.

- Trust and Technology has Reciprocal Relationship



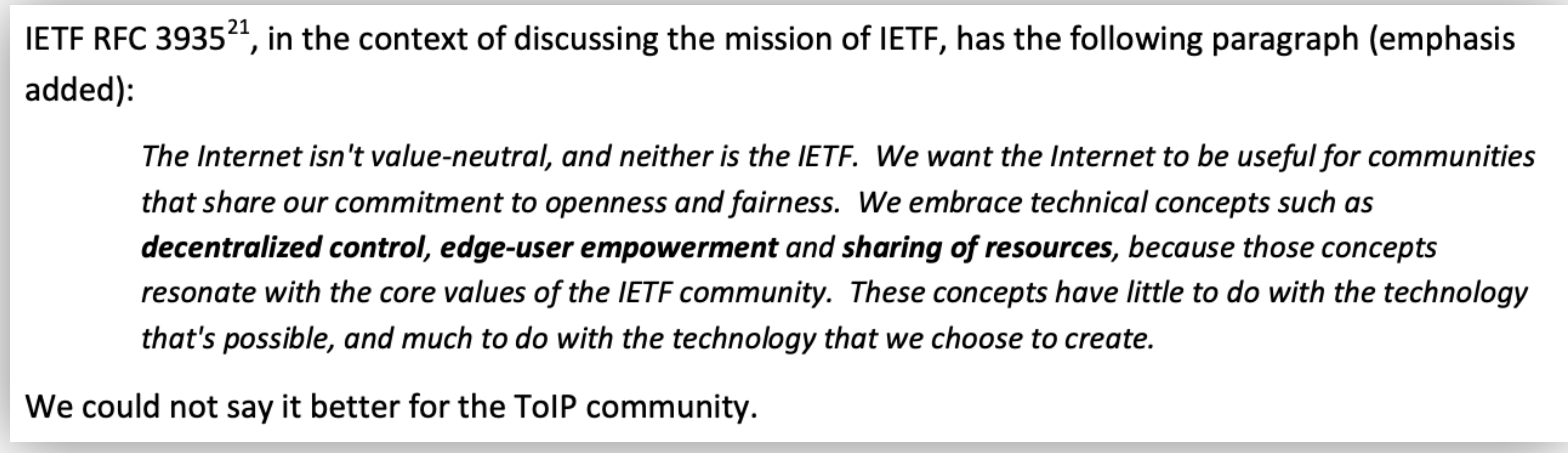
Layer	Relevance	Explanation
4	★★★★★	Ecosystem governance frameworks at Layer 4 are how ToIP users can evaluate and make trust decisions about digital trust ecosystems and the applications running within them.
3	★★★★★	Credential governance frameworks at Layer 3 are how ToIP users can evaluate and make 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.

Design principle #15



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

- Design for Ethical Values
  - It's a First Principle
  - Values of safety, privacy, autonomy
  - Values of accessibility, reliability, functionality & economics
  - Values of happiness, creative pursuits, and other inspirations



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

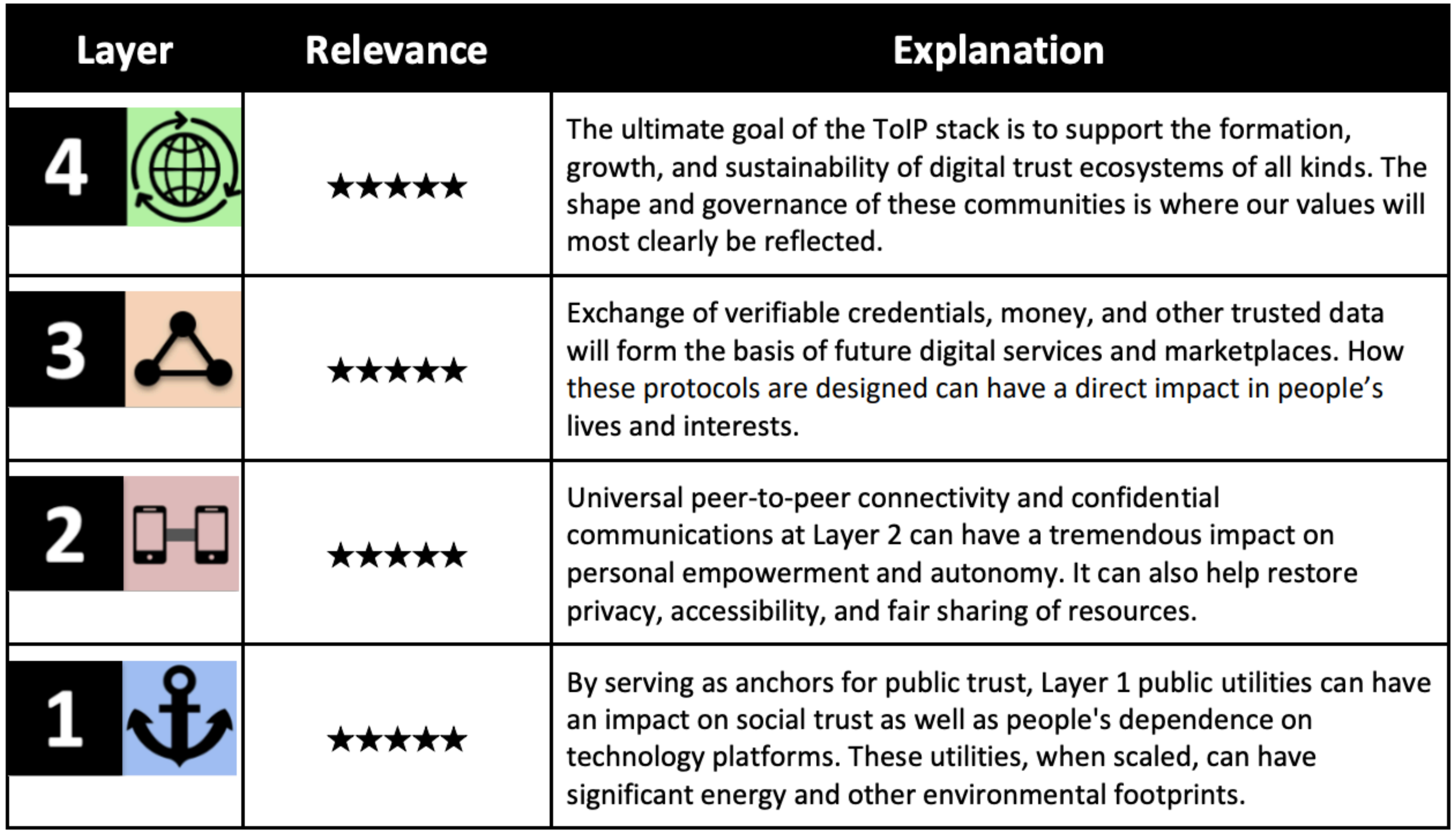
> *The Internet isn't value-neutral, and neither is the IETF. We want the Internet to be useful for communities that share our commitment to openness and fairness. We embrace technical concepts such as* ***decentralized control***, ***edge-user empowerment*** *and* ***sharing of resources***, *because 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.

Design principle #15

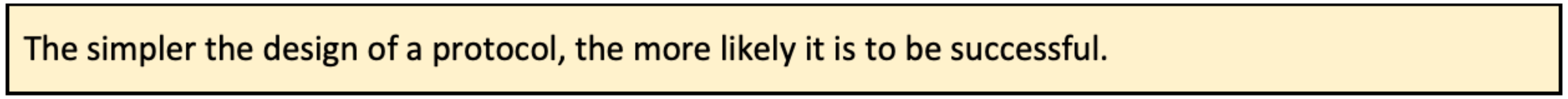
The ToIP Stack has a strong ethical dimension. Design it with a commitment to ethical values.

- Design for Ethical Values



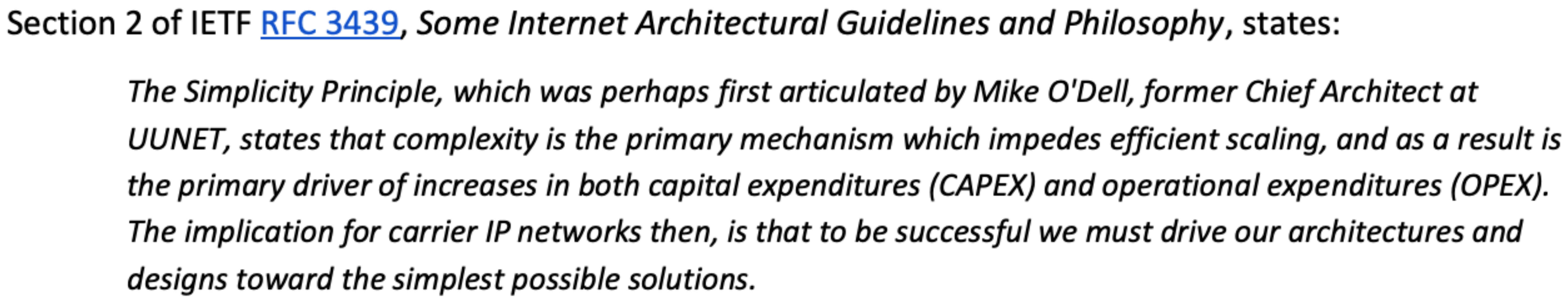
Layer	Relevance	Explanation
4	★★★★★	The ultimate goal of the ToIP stack is to support the formation, growth, and sustainability of digital trust ecosystems of all kinds. The shape and governance of these communities is where our values will most clearly be reflected.
3	★★★★★	Exchange of verifiable credentials, 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.

Design principle #16



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

- Design for Simplicity



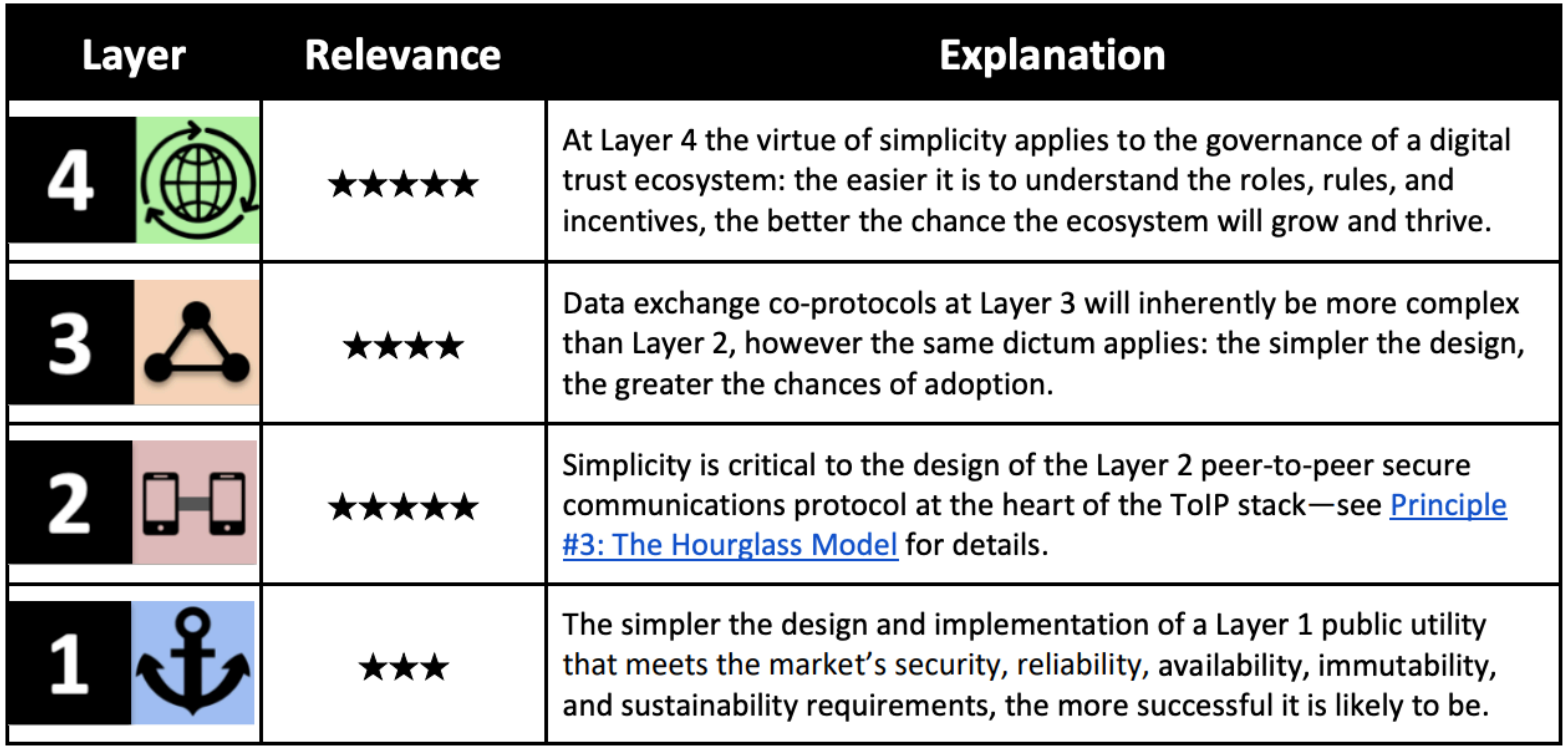
Section 2 of IETF RFC 3439, *Some Internet Architectural Guidelines and Philosophy*, states:

> *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.*

Design principle #16

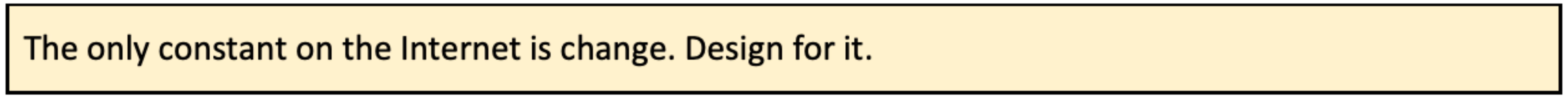
The simpler the design of a protocol, the more likely it is to be successful.

- Design for Simplicity

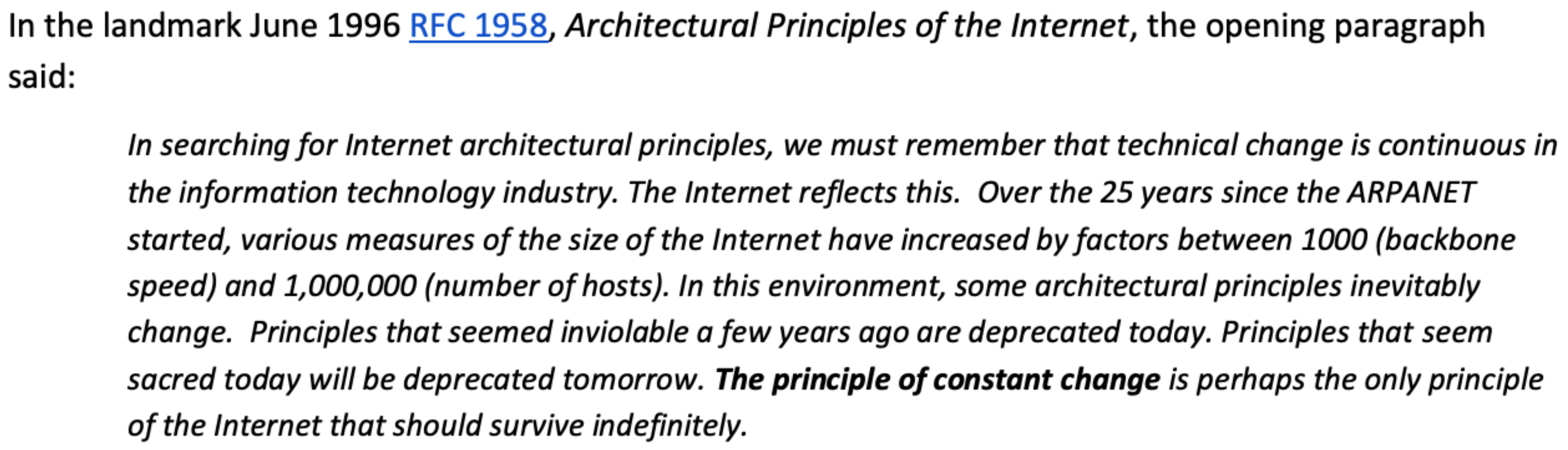


Layer	Relevance	Explanation
4	★★★★★	At Layer 4 the virtue of simplicity applies to the governance of a digital trust ecosystem: the easier it is to understand the roles, rules, and incentives, the better the chance the ecosystem will grow and thrive.
3	★★★★	Data exchange co-protocols at Layer 3 will 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.

Design principle #17


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

- Design for Constant Change

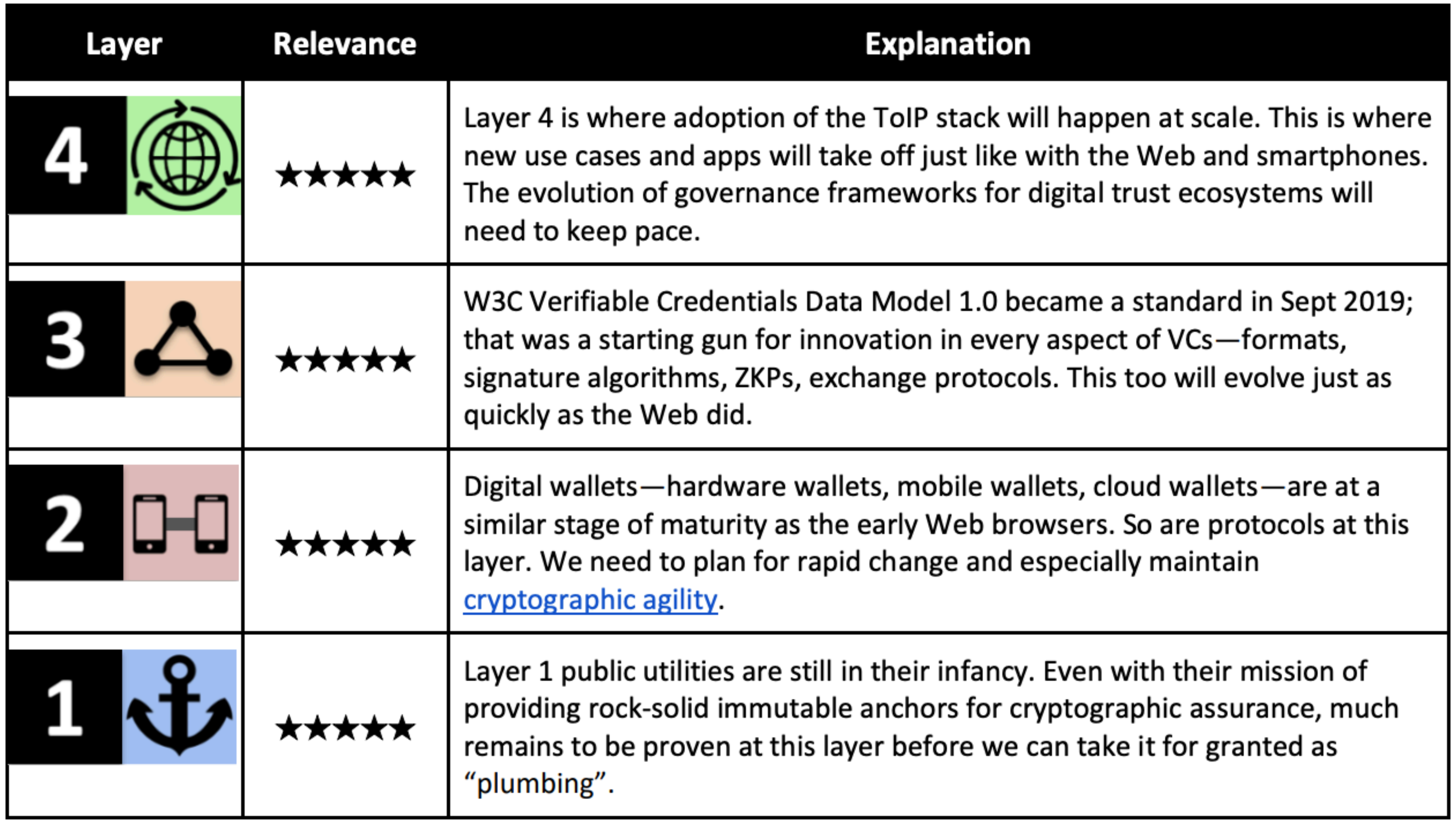

In the landmark June 1996 RFC 1958, *Architectural Principles of the Internet*, the opening paragraph said:

> *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.*

Design principle #17

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

- Design for Constant Change



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”.

Q & A

1. The End-to-End Principle
2. Connectivity is Its Own Reward
3. The Hourglass Model
4. Decentralization by Design and Default
5. Cryptographic Verifiability
6. Confidentiality by Design and Default
7. Keys at the Edge
8. Trust is Human
9. Trust is Relational
10. Trust is Directional
11. Trust is Contextual
12. Trust has Limits
13. Trust can be Transitive
14. Trust and Technology has a Reciprocal Relationship
15. Design for Ethical Values
16. Design for Simplicity
17. Design for Constant Change