TSP is an Inter-Trust Domain Protocol (ITDP)

Wenjing Chu

Feb 21, 2023
March 21, 2023 (Part 2: Sections 5 & 6)

Public Reference to this doc should be in its entirety and as:
{Wenjing Chu, "TSP is an Inter-Trust Domain Protocol (ITDP)", Feb 21, 2023. https://github.com/wenjing/Inter-Trust-Domain-Protocol}

A Quick Bio:

- I had participated and developed many Internet/network protocols in my career with several Startups and Cisco, e.g. TCP/IP stack, Internet routing protocols (BGP, OSPF, ISIS, RIP…), IP Multicast, IP Mobility, MPLS/ATM/FR, Wi-Fi/IEEE 802.11, and 4G/5G mobile networks...
- A Senior Product Manager for Nokia in security.
- A Distinguished Engineer in Dell/VMware and currently at Futurewei, where I am Sr. Director of Technology Strategy.
- Have been actively involved in many open source, standard and other communities: ToIP, OpenWallet (OWF), LF Edge, LF Network, C2PA, W3C, IETF, IEEE, ETSI ... over the years.
- With ToIP: a founding Steering Committee member, primary contributor of TechArch Spec, Co-chair of TSP task force, Co-chair of AI and metaverse task force, etc.

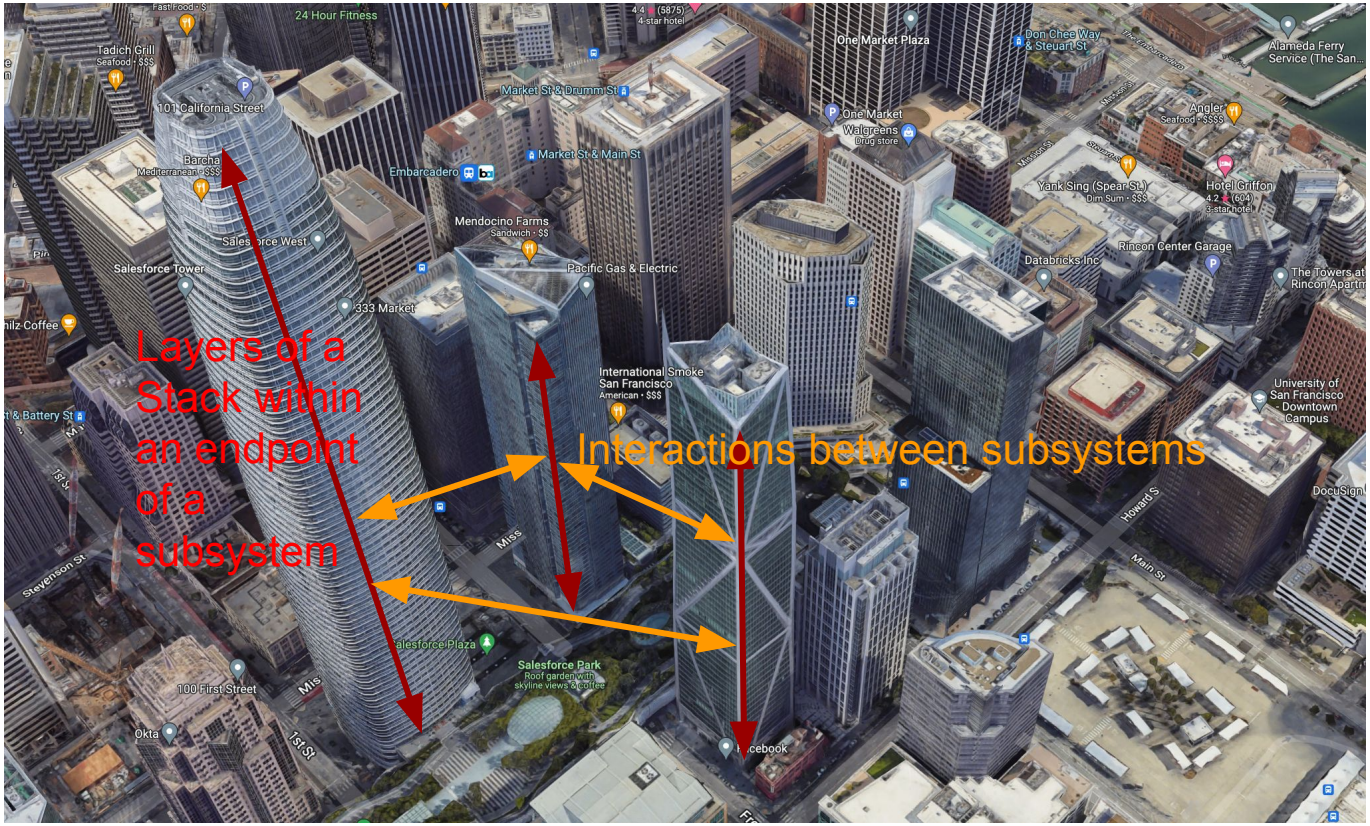
My proposal

1. A protocol should be understood under a Reference Architecture.
    a. I will re-present a short overview of my Reference Architecture that was first presented at IIW April 2022 and then contributed to the TechArch TF.
2. What is TSP, exactly?
3. Less is More - what the TSP should and shouldn't include.
4. What the Protocol Specification defines
5. How various trust tasks can be built on top of ITDP (TSP)? How does ITDP relate to other protocols (forerunners)?
6. How lower layer Support Functions may implement what ITDP needs
7. Q&A

Part 2 is to cover these sections that I didn't have time to present in the first time.

(1) A Quick Review of the Reference Architecture

A protocol stack is to view the decomposition vertically in functionality, where each higher layer incrementally adds functionality above the layer(s) below it. It is suitable within an endpoint where dependencies are clearly ordered. But it is not suitable to capture relationships between different sub-systems. **The Reference Architecture is a prerequisite to understand a protocol stack.**



An analogy for a Reference Architecture

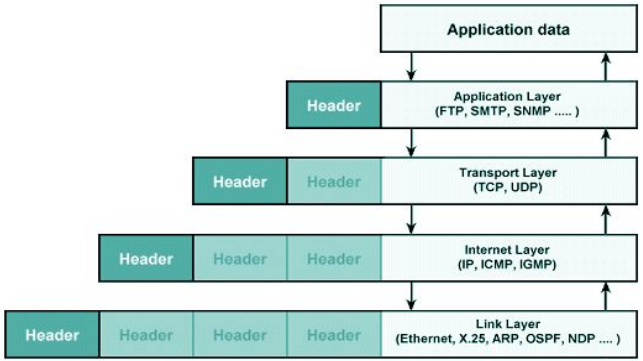
For Example: Internet Reference Architecture

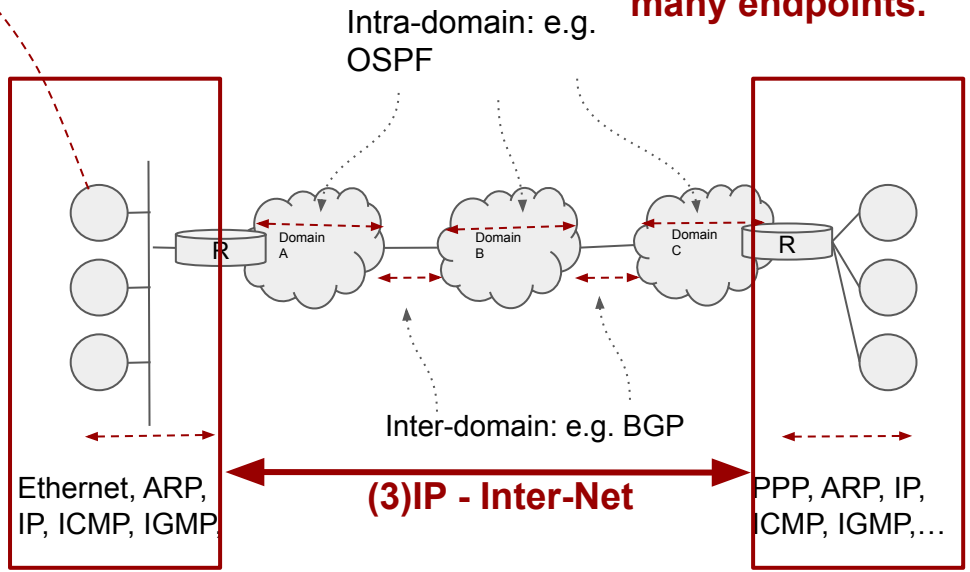
The familiar Internet *stack view* is usually focused on ***end systems***

The Internet reference architecture

(2)Another local area network with many endpoints.

(4)The protocol stack on each endpoint





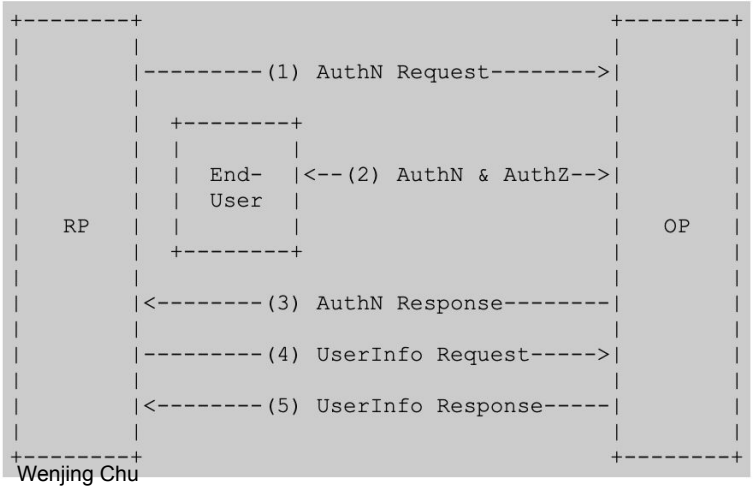
(1) One local area network with many endpoints.

Another Example: OIDC Reference Architecture

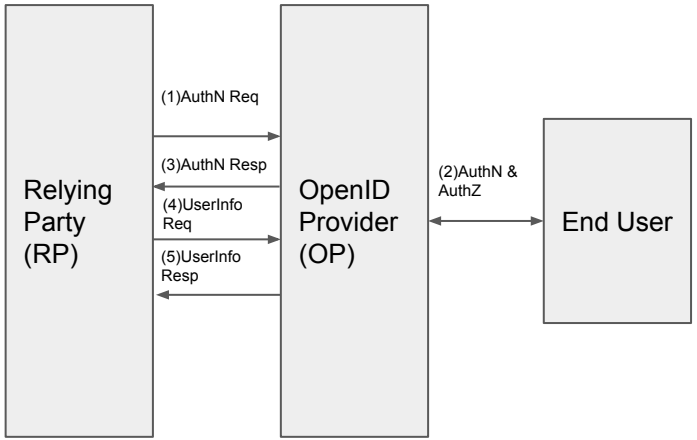
The OpenID Connect protocol, in abstract, follows the following steps.

1. The RP (Client) sends a request to the OpenID Provider (OP).
2. The OP authenticates the End-User and obtains authorization.
3. The OP responds with an ID Token and usually an Access Token.
4. The RP can send a request with the Access Token to the UserInfo Endpoint.
5. The UserInfo Endpoint returns Claims about the End-User.

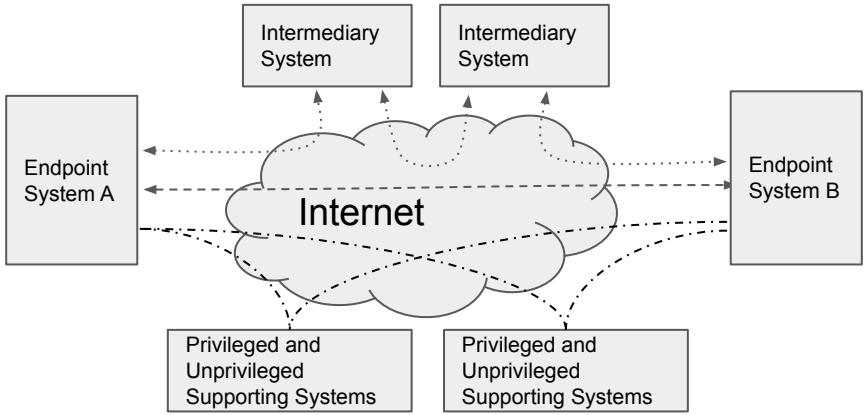
These steps are illustrated in the following diagram:



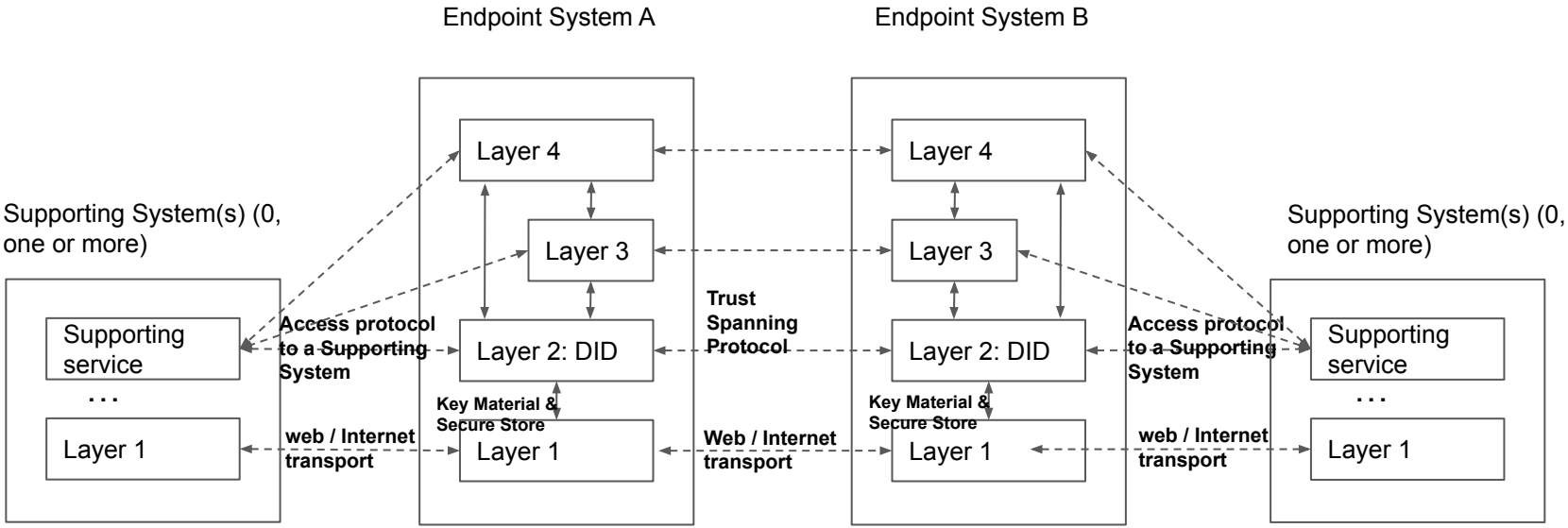
Same picture but unfolded.



The proposed Reference Architecture (RA)

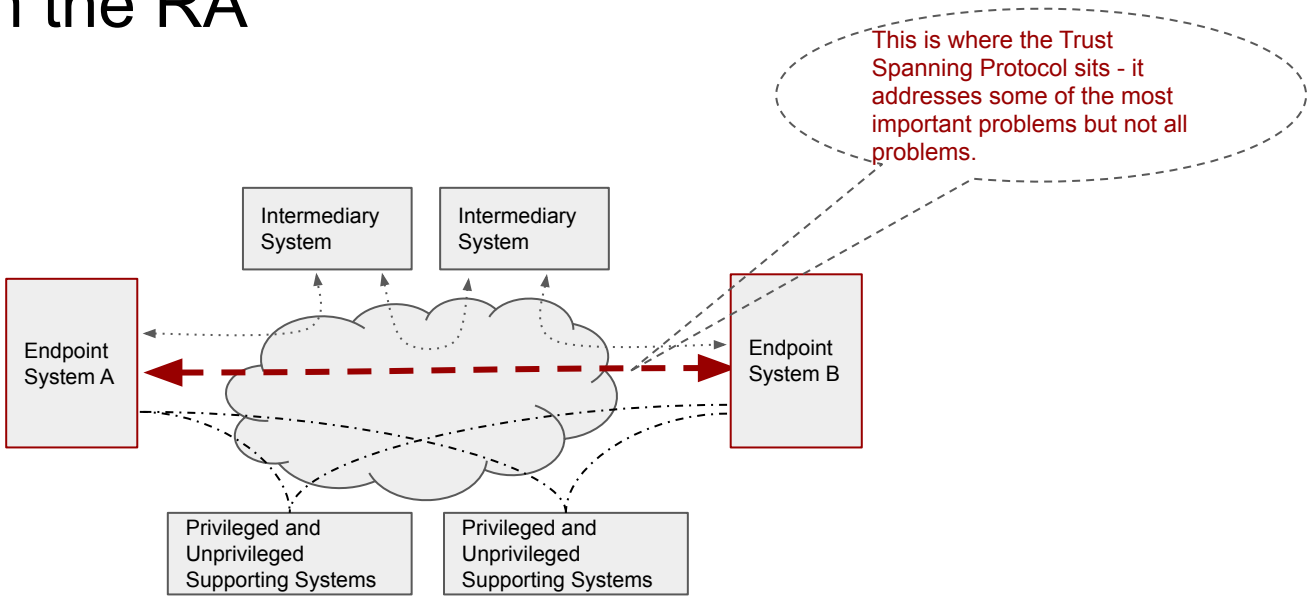


- Subsystems are delineated by **locus of control (domain)**
- They interact through **a set of protocols**, not just one.
- Each **type** of subsystems has a shared stack architecture*, but the stack architecture is not identical across different types of subsystems**.

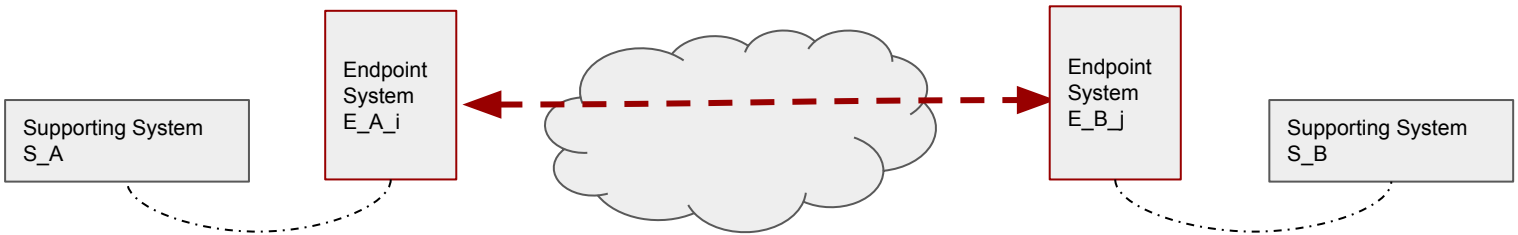


A Generalized Reference Architecture (with Intermediary Systems removed for clarity)

How TSP fits in the RA

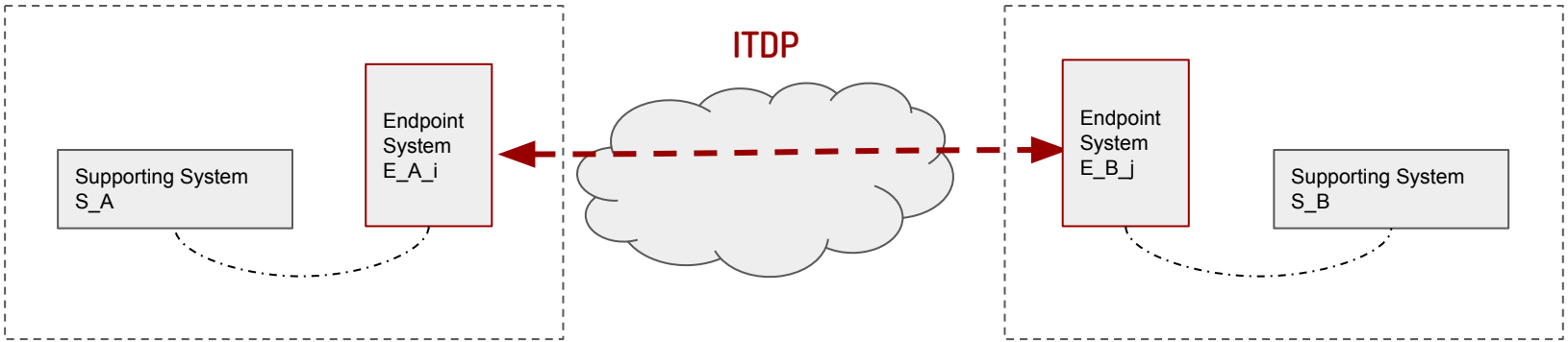


How TSP fits in the RA



(2) What is TSP, exactly?

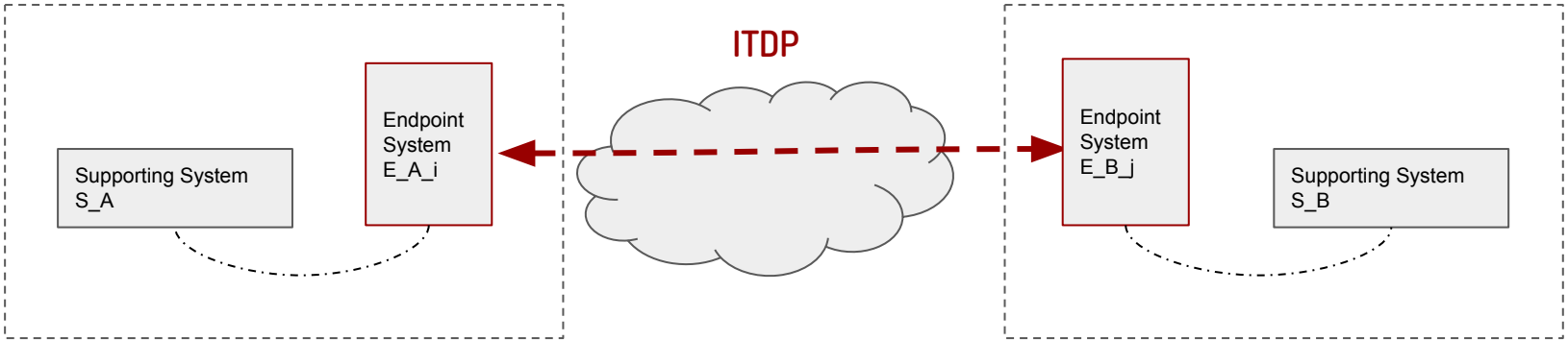
TSP is an Inter-Trust Domain Protocol (ITDP)



(0)How to bridge (or span) trust signals between Trust Domains is the problem that TSP needs to solve. Other problems are either the concerns of other protocols, layers of the same stack or other stacks entirely.

TSP is an Inter-Trust Domain Protocol (ITDP)

(1)ITDP enables any Trust Domain x to any Trust Domain y interoperability and supports as many types of Trust Domains as possible.

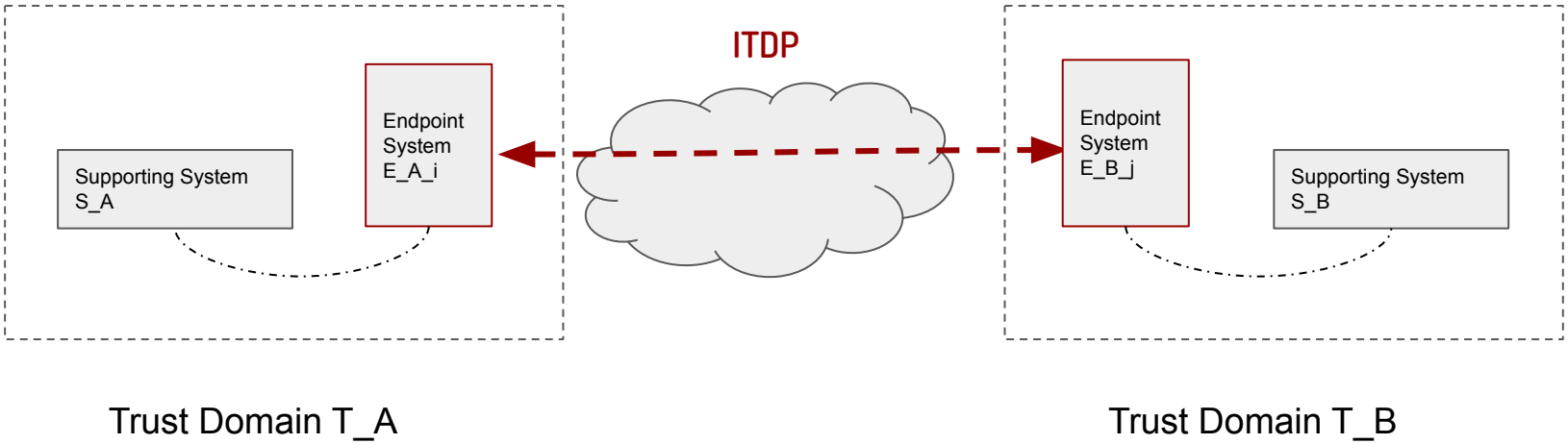


Trust Domain T_A

T_x examples… non-exhaustive at all	**S_x**	**E_x**
DID Method x	DID Method x impl. (eg blockchains)	Any except…
AID/KERI	KERI impl. (eg Witness pool)	Any except…
Centr. VID	Enterprise VID (eg IAM)	Enterprise issued accts
Fed. VID	OpenID VID (eg OIDC)	Any except…

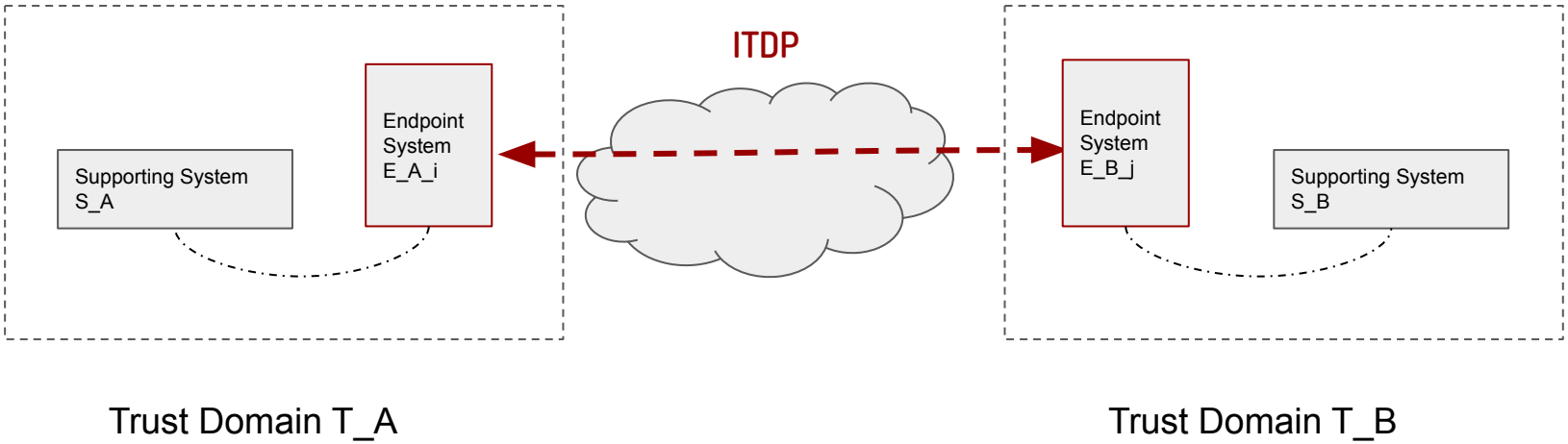
Trust Domain T_B

TSP is an Inter-Trust Domain Protocol (ITDP)



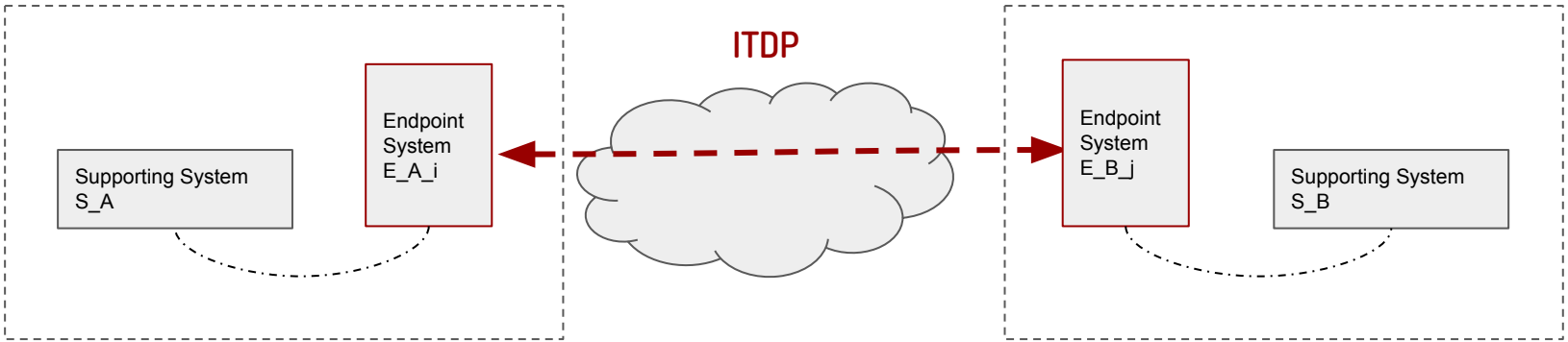
(2)The ITDP only bridges trust signal that EXISTS in trust domain A to trust domain B. It does not create new signals.

TSP is an Inter-Trust Domain Protocol (ITDP)



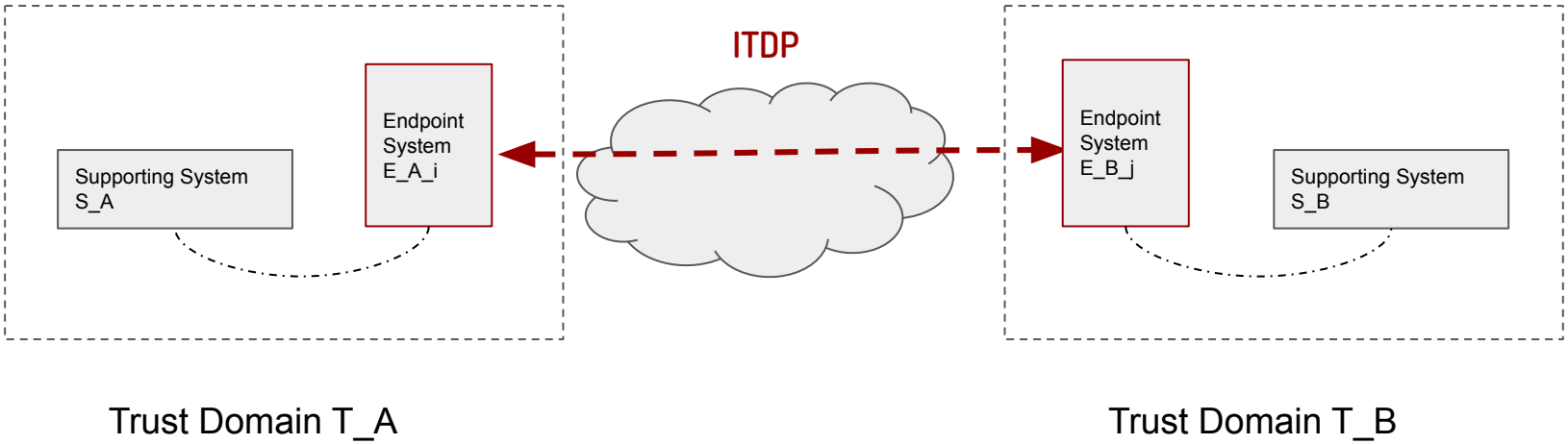
(3)The ITDP only enables the conveying of the trust signal. It is up to the endpoints to decide if those signals are indeed trustworthy or sufficient within the context that only they know. (the End to End principle).

TSP is an Inter-Trust Domain Protocol (ITDP)



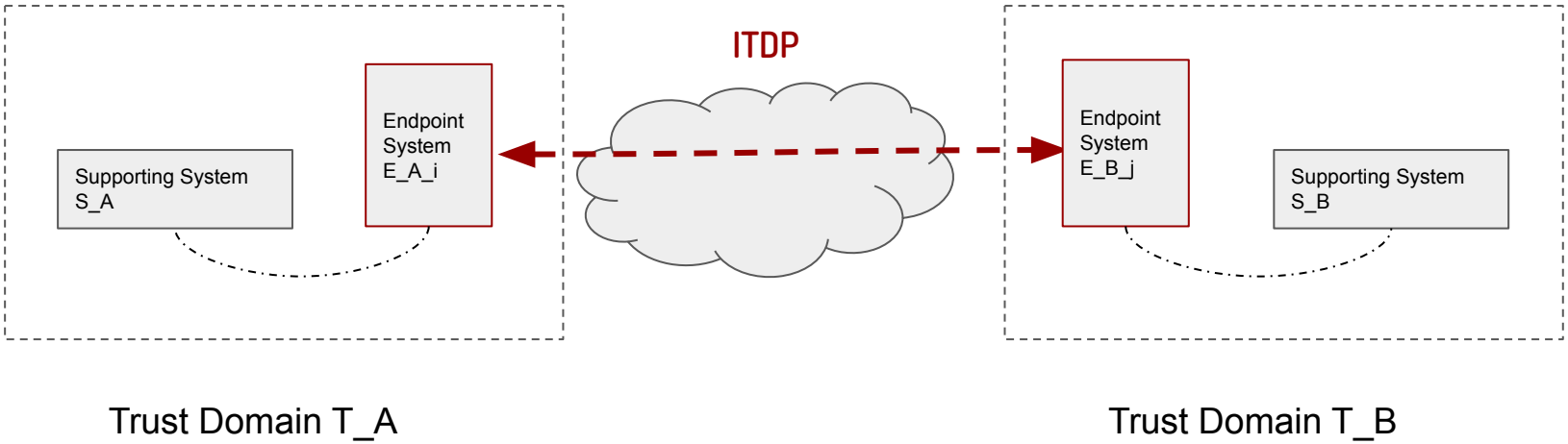
(4)ITDP only enables bootstrap of minimal level of trust between trust domains such that it's sufficient for they to continue to expand to more sophisticated levels of trust as they see fit.

TSP is an Inter-Trust Domain Protocol (ITDP)



(5)After the bootstrap of minimally sufficient trust between trust domains, the ITDP is also the channel and language in which they can maintain &/or expand to more sophisticated trust levels as needed.

TSP is an Inter-Trust Domain Protocol (ITDP)



(6)Speakers of ITDP are any generic endpoints without much constraints - mobile, web, IoT, physical vs. virtual, human associated vs. objects, large vs. small vs. tiny, individualized vs. clusters vs. platforms vs. clouds, nor of their roles ... could be any of above, all of above, none of above or any mix we can't yet imagine.

(3)Less is More - What ITDP should and shouldn’t do.

When I say what ITDP shouldn’t do…

It doesn’t mean it shouldn’t be done or discussed.

It doesn’t mean it’s unimportant or uninteresting.

It doesn’t mean it’s not related.

It just means better let others/protocols do it.

“Other protocols or layers of the same stack”

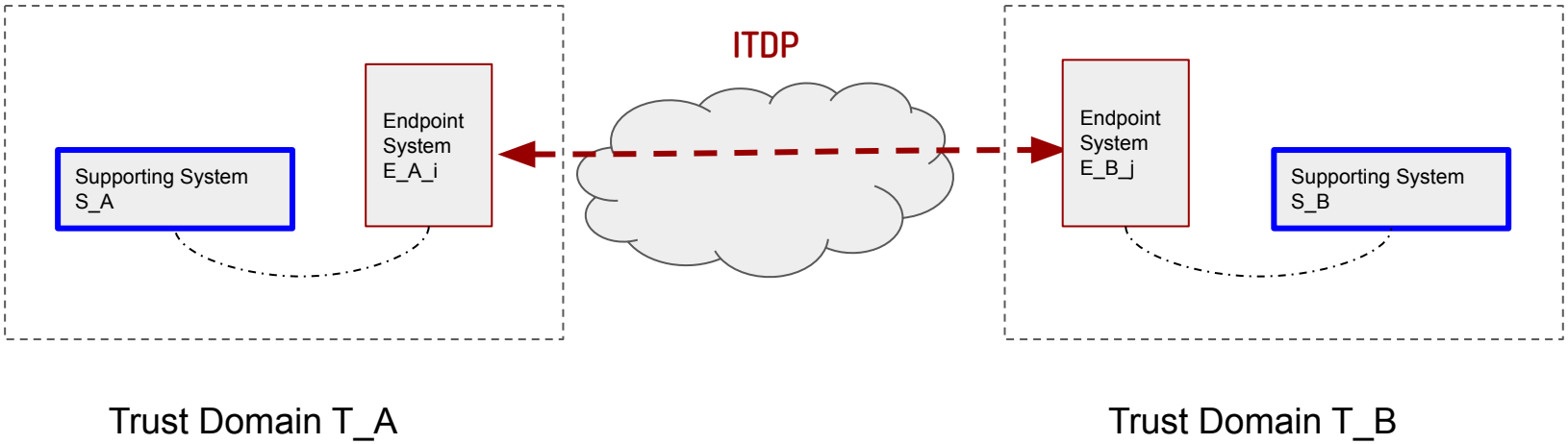
“Or other stacks entirely”

“Could be in a separate ‘guide’ document”

“The principle of Separation of Concerns.”

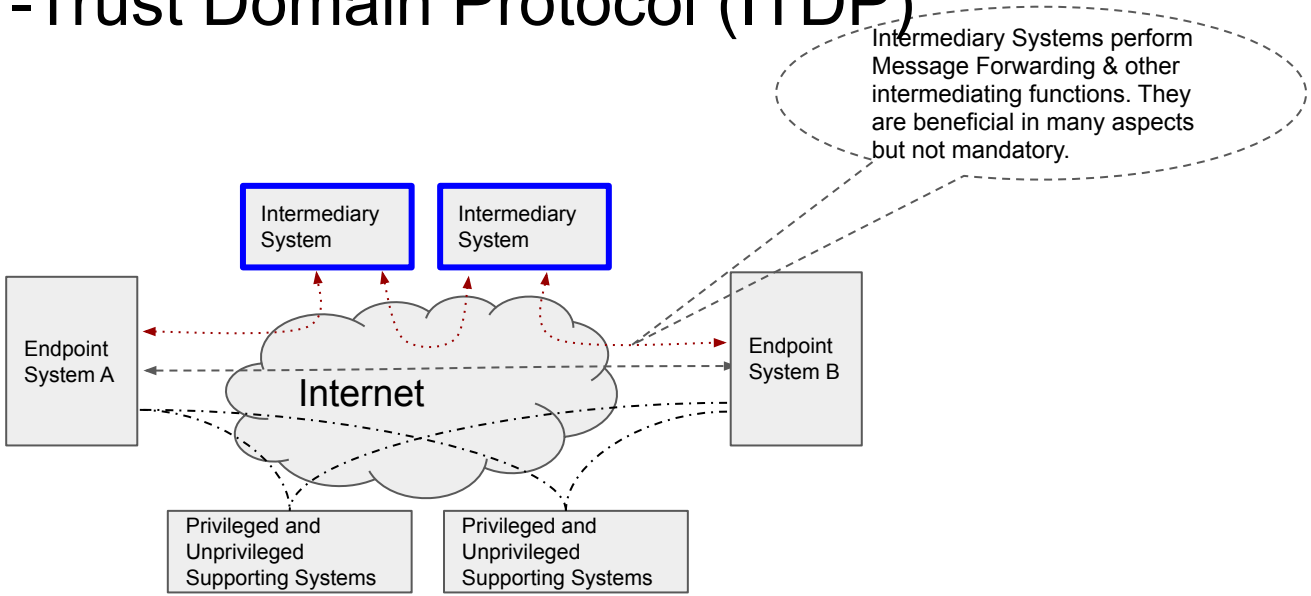
“The Hourglass principle”

TSP is an Inter-Trust Domain Protocol (ITDP)



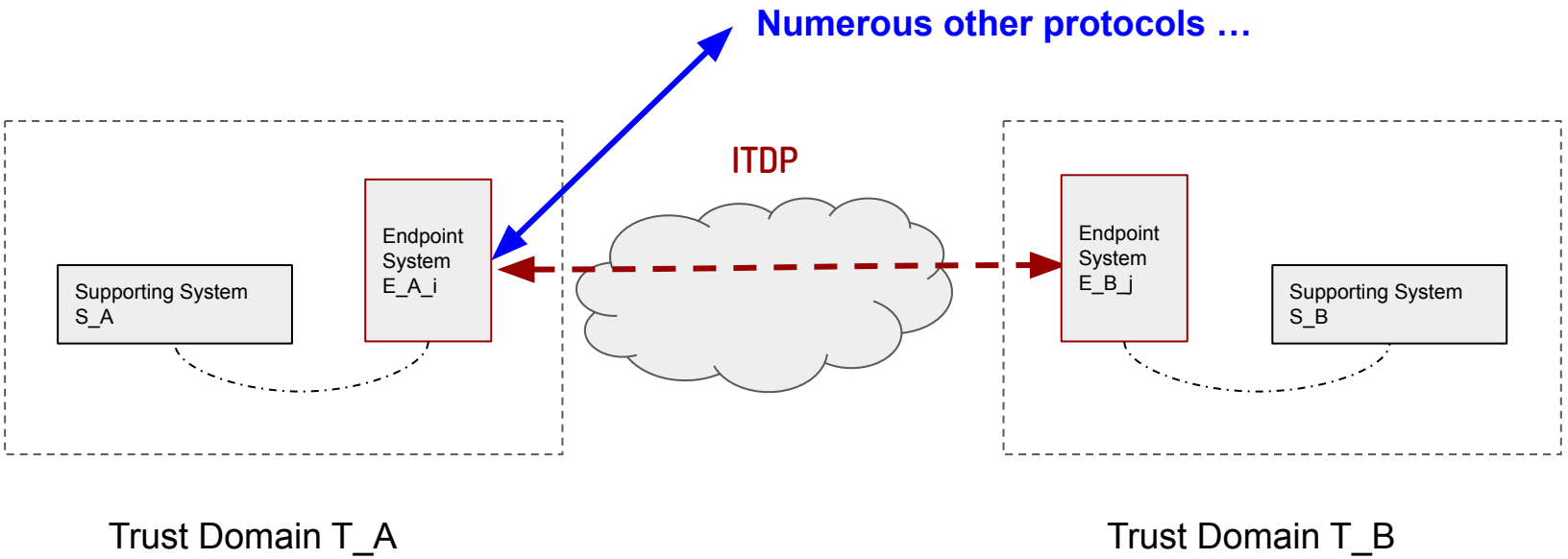
(7)Supporting systems themselves (e.g. blockchains, KERI witnesses, trust registries, reputation...) are out of scope, except the interface to them by the endpoints.

TSP is an Inter-Trust Domain Protocol (ITDP)



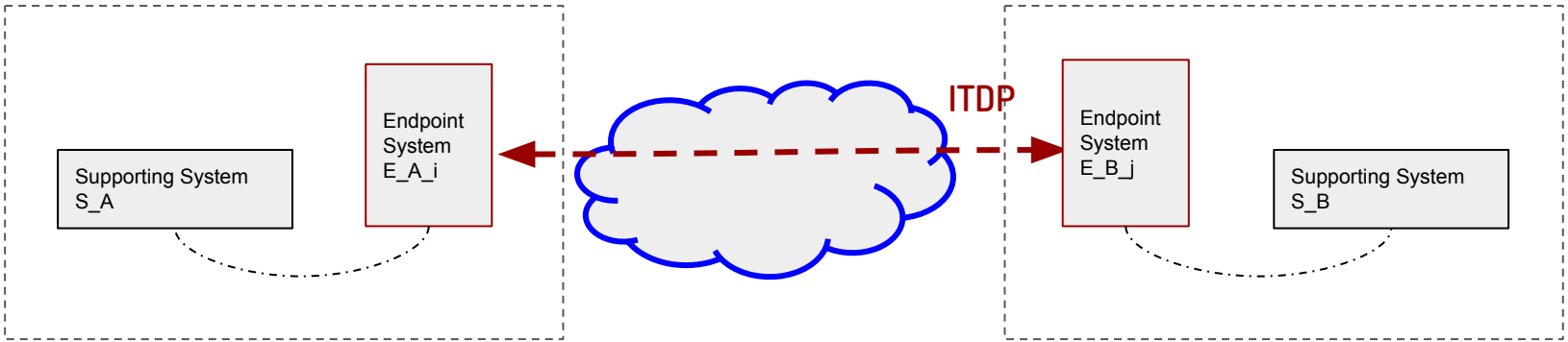
(8)Intermediary systems themselves (e.g. message routing, message storage, discovery...) are out of scope, except the interface to them by the endpoints.

TSP is an Inter-Trust Domain Protocol (ITDP)



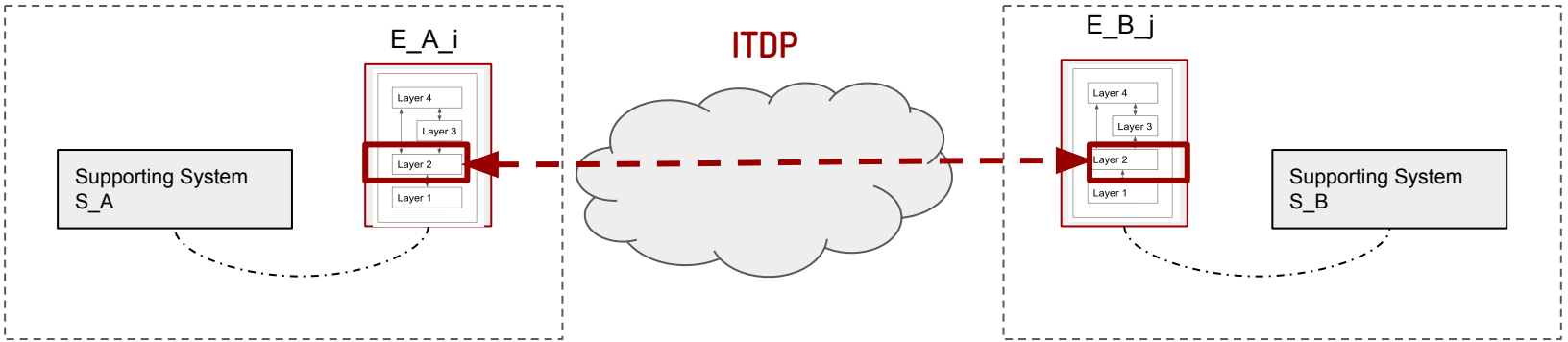
(9)The endpoints are likely to use numerous other protocols not directly concerned with bridging trust - they are all out of scope, and not depicted in my diagrams if not strictly necessary.

TSP is an Inter-Trust Domain Protocol (ITDP)



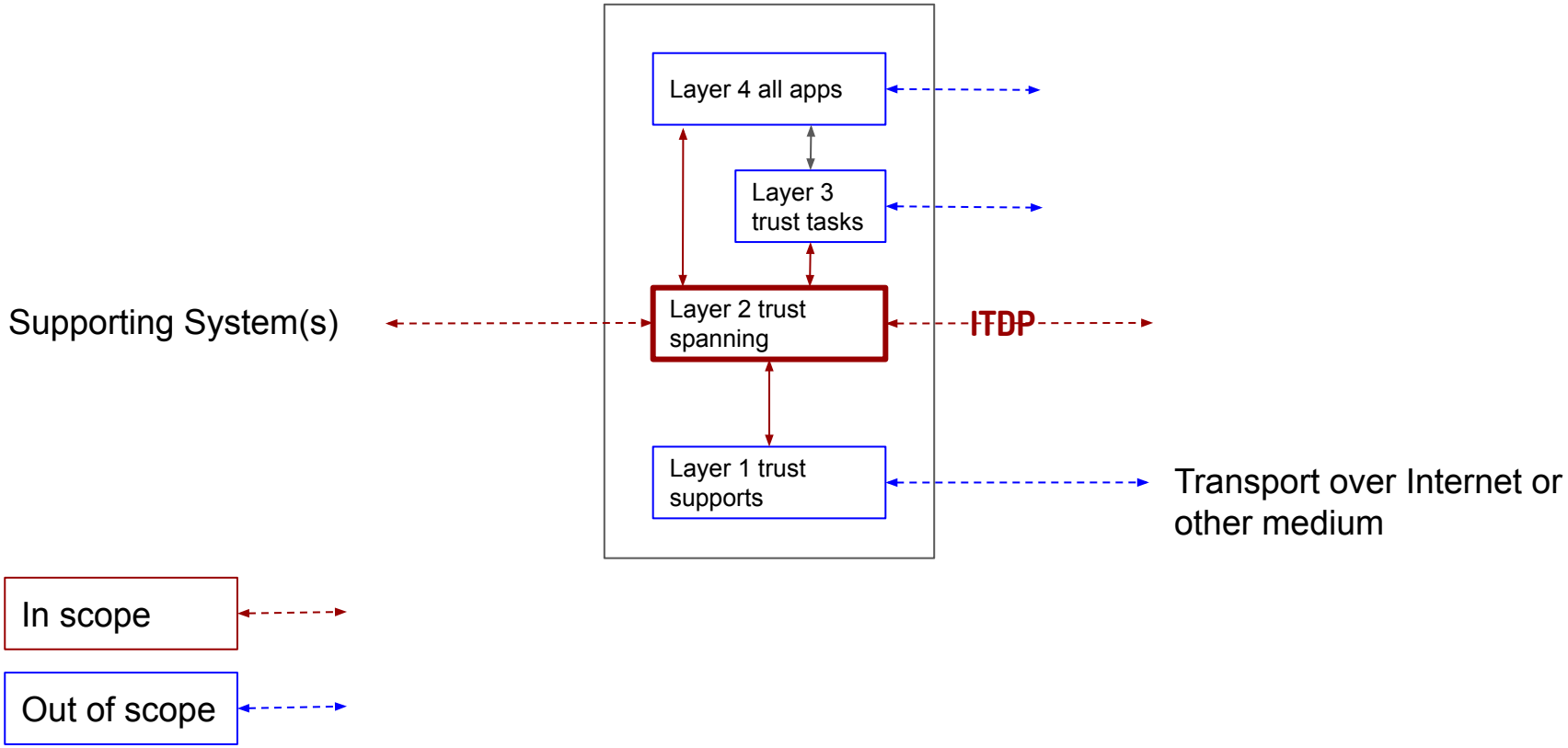
(10)The Internet (or other networking technologies) supports numerous communication or messaging functions - all of them are available for ITDP to use the lower layer regardless of their own layers within the Internet. ITDP uses them but they themselves are out of scope except the interface to use them.

TSP is an Inter-Trust Domain Protocol (ITDP)

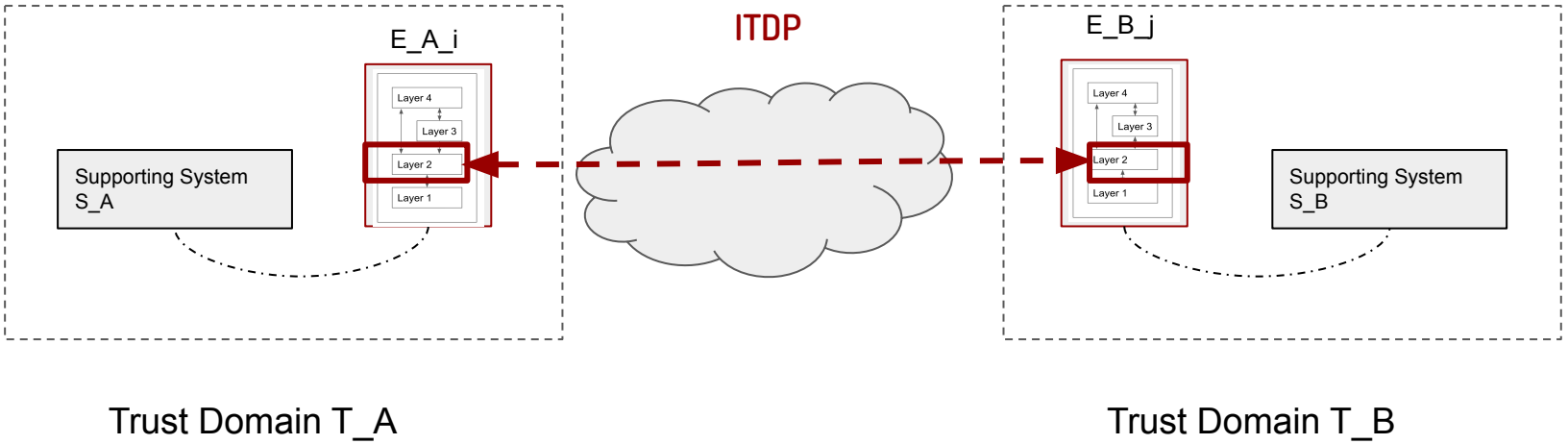


Now we can narrow down to what is in scope within the stack of an endpoint and specifically in the trust spanning layer (or layer 2).

An Endpoint's Trust Protocol Stack

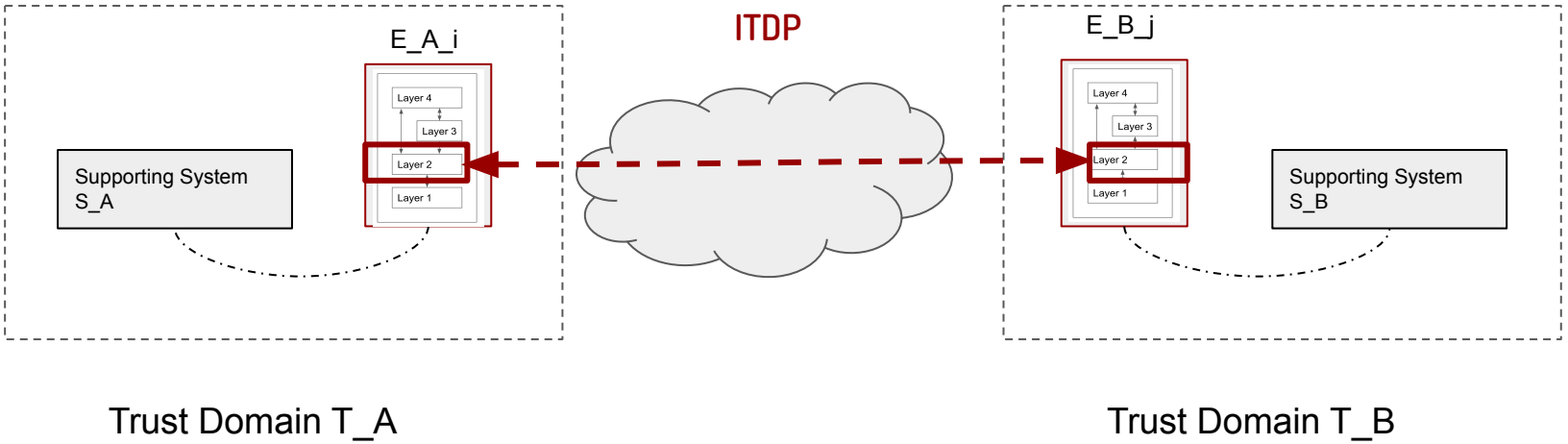


TSP is an Inter-Trust Domain Protocol (ITDP)



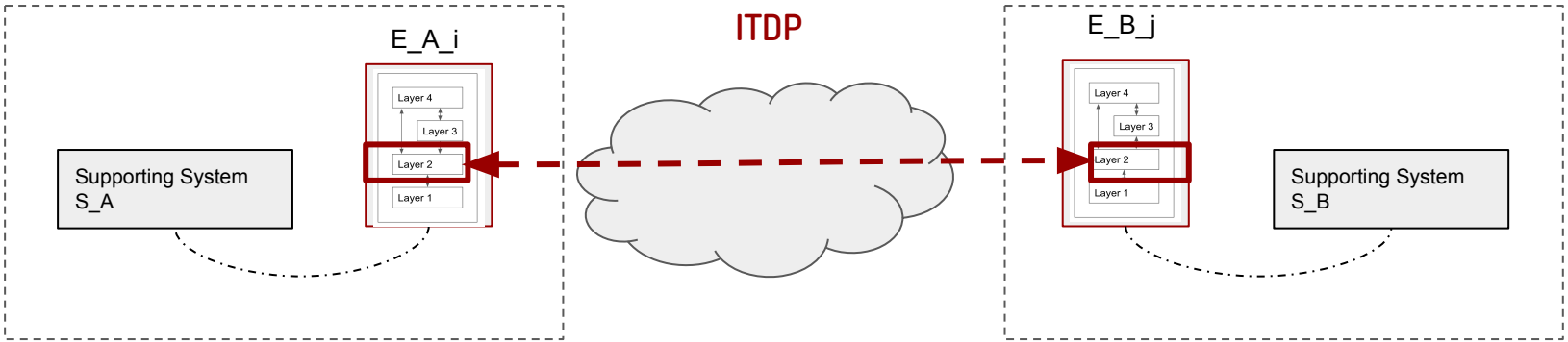
(11)ITDP needs to support a *generic identification* that can be efficiently mapped to trust domain specific addresses AND can be verified for authenticity (*verifiable*). Both are required and should be efficient. The actual implementations of such generic verifiable identification are out of scope.

TSP is an Inter-Trust Domain Protocol (ITDP)



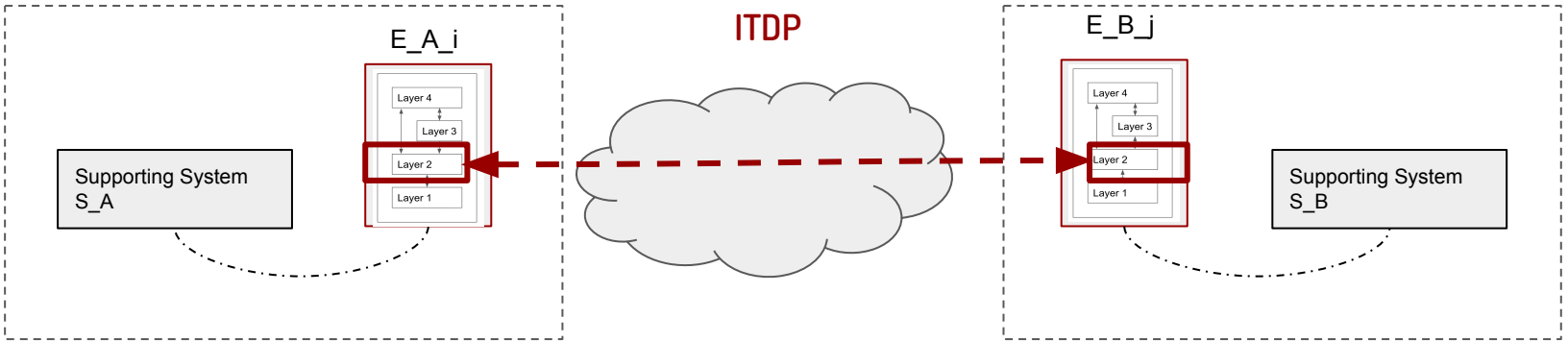
(12)ITDP needs to have a *generic language* to express necessary trust signals and handshake protocols (i.e. primitives and messages) to bootstrap, maintain and enable further exchange for additional trust signals. The actual transport of messages and additional trust exchange protocols (e.g. VC issuance and presentation/proof) are out of scope.

TSP is an Inter-Trust Domain Protocol (ITDP)



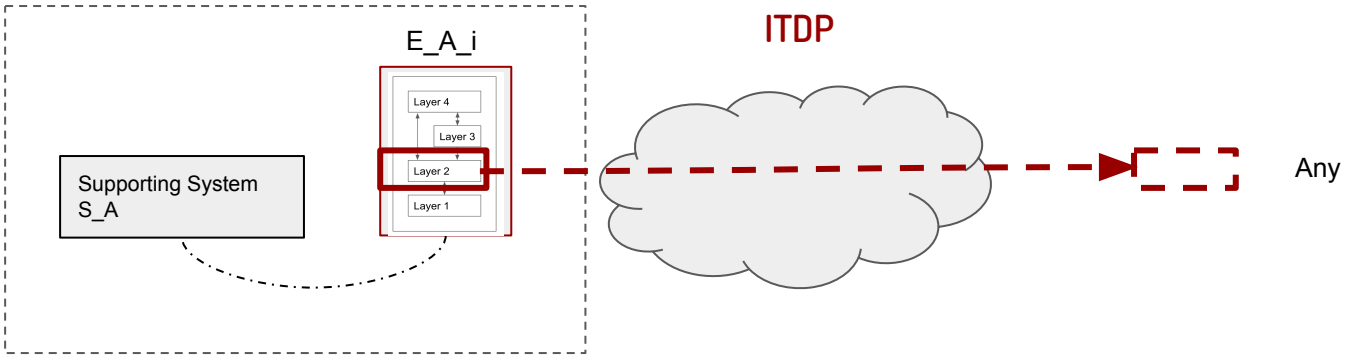
(13)ITDP establishes the minimum level of trust between endpoints, but it is not a tunneling protocol (unlike IP). The trust related information exchanged through ITDP can be used by the endpoints in any way they like, including creating tunnels or connections. Some can be, but Many data exchanges are not tunneled through or encapsulated in ITDP. State it in another way, ITDP is a new tool that those data exchange protocols can use for trust.

TSP is an Inter-Trust Domain Protocol (ITDP)



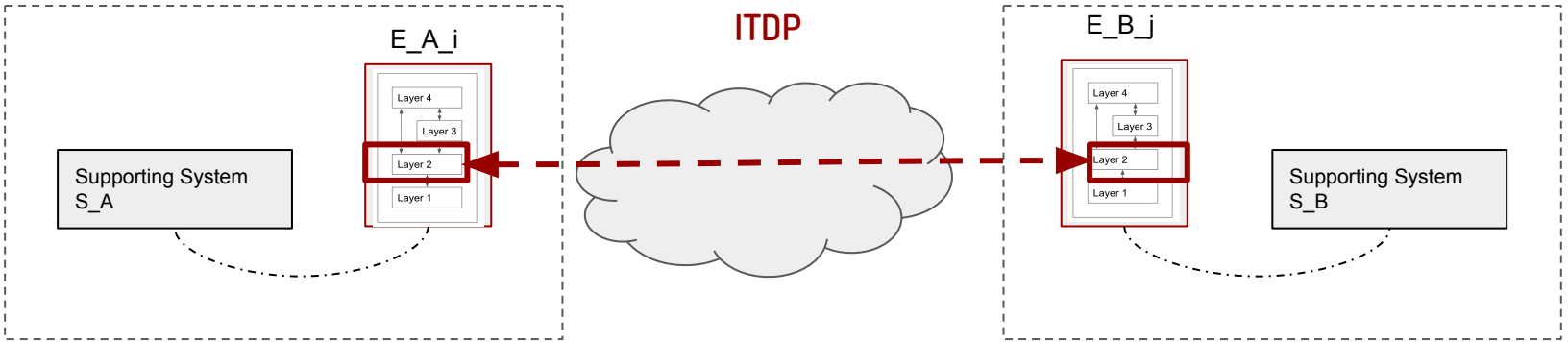
(14)ITDP supports multicast, anycast, broadcast in addition to unicast. An intermediary with an VID (i.e. destination) can perform these distribution functions.

TSP is an Inter-Trust Domain Protocol (ITDP)



(15)ITDP may function without a destination VID. If used in this way, it should mean "To whom it may concern". And verification is one-direction only. It will be delivered to an intermediary (well known or pre-configured or known in other ways) whose interpretation of the message's handling is out of scope.

TSP is an Inter-Trust Domain Protocol (ITDP)

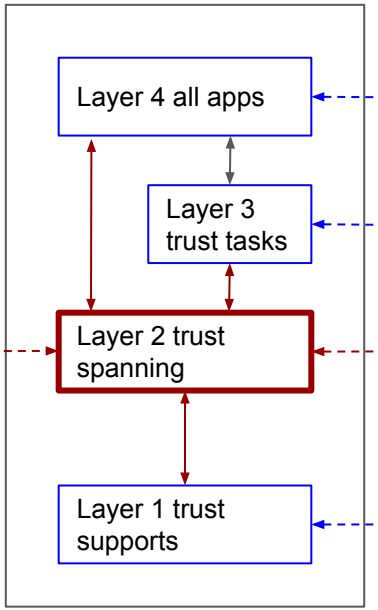


(16)ITDP can sit on top of any message delivery functions with wide/global reachability. Implementing those delivery functions are out of scope. There are numerous examples of such functions with global reachability.

(4)What the ITDP specification defines.

Recap

An Endpoint's Trust Protocol Stack



Supporting System(s)

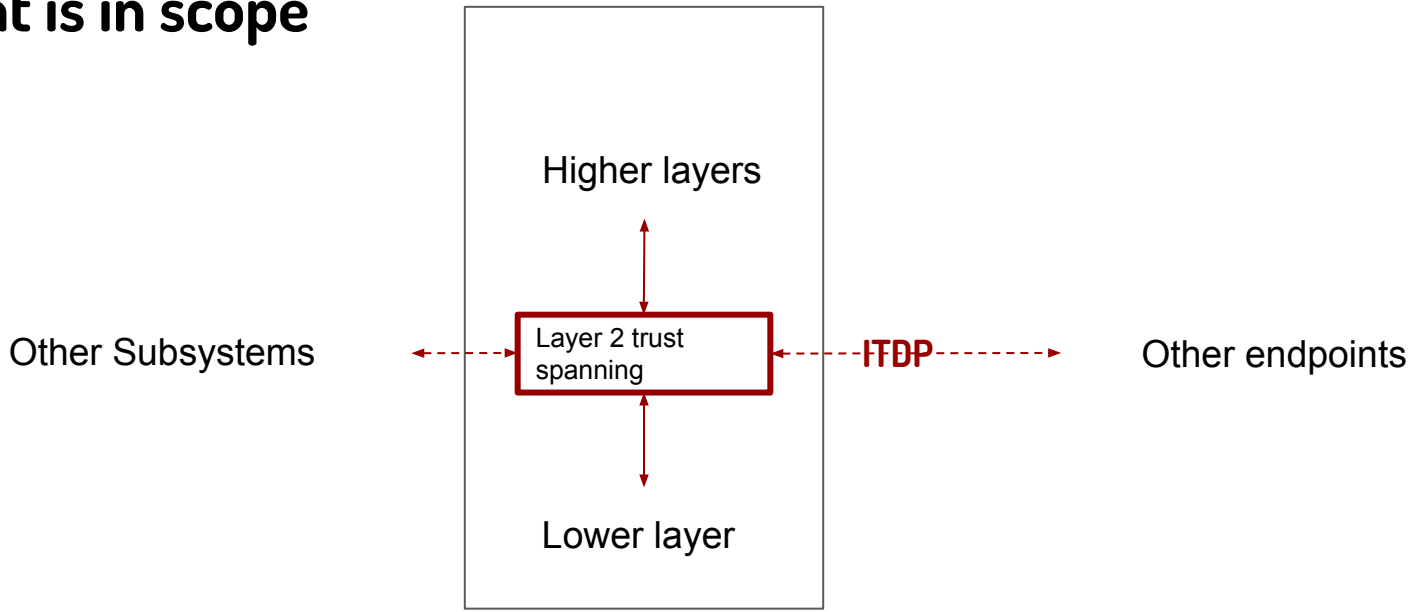
ITDP

Transport over Internet or other medium

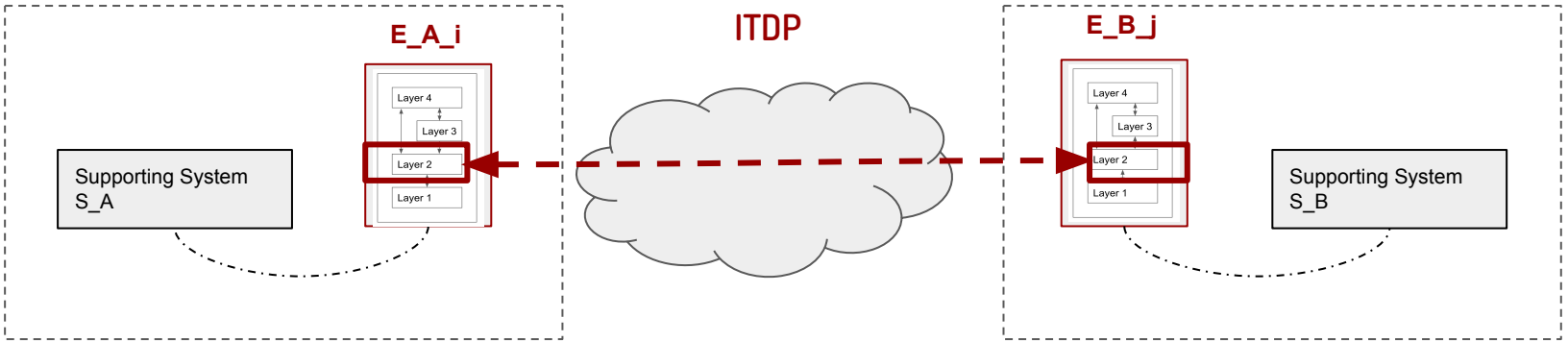
In scope

Out of scope

What is in scope

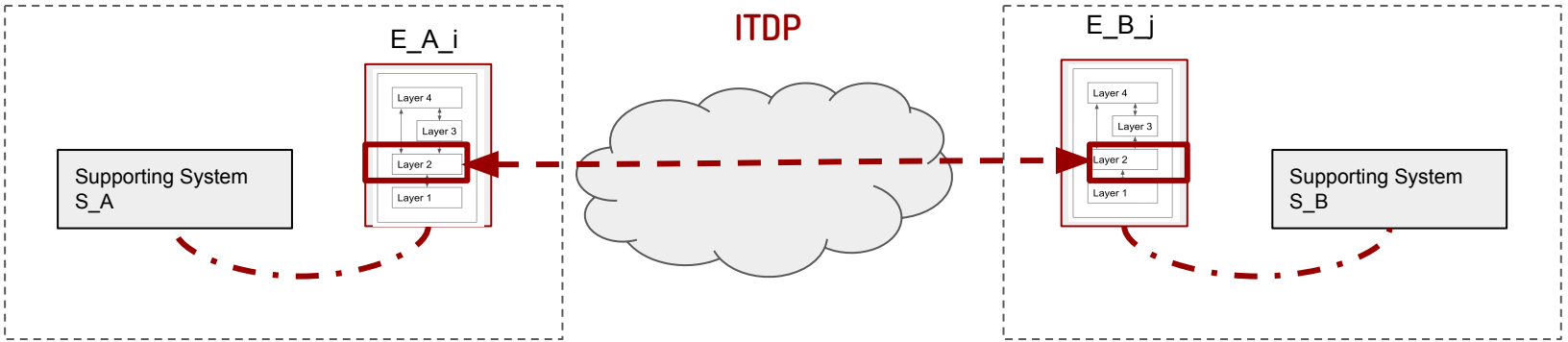


TSP is an Inter-Trust Domain Protocol (ITDP)



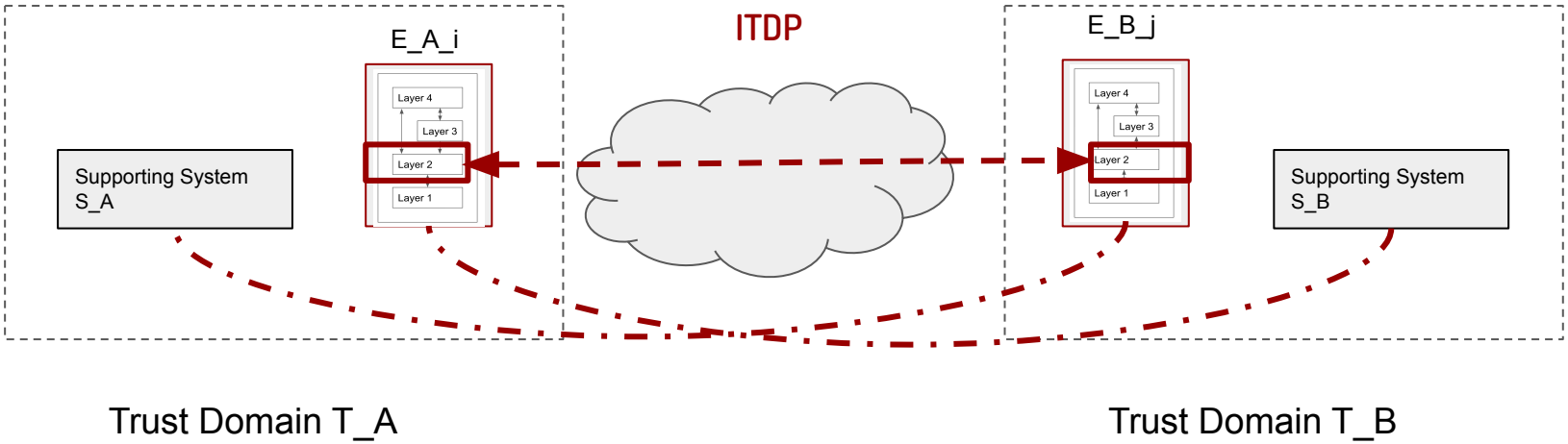
(17)ITDP defines a generic, trust domain agnostic, adaptable, multi-format identification scheme. This scheme must satisfy most common schemes existed today and leave flexibility for future adaptation. (e.g. DID core...)

TSP is an Inter-Trust Domain Protocol (ITDP)

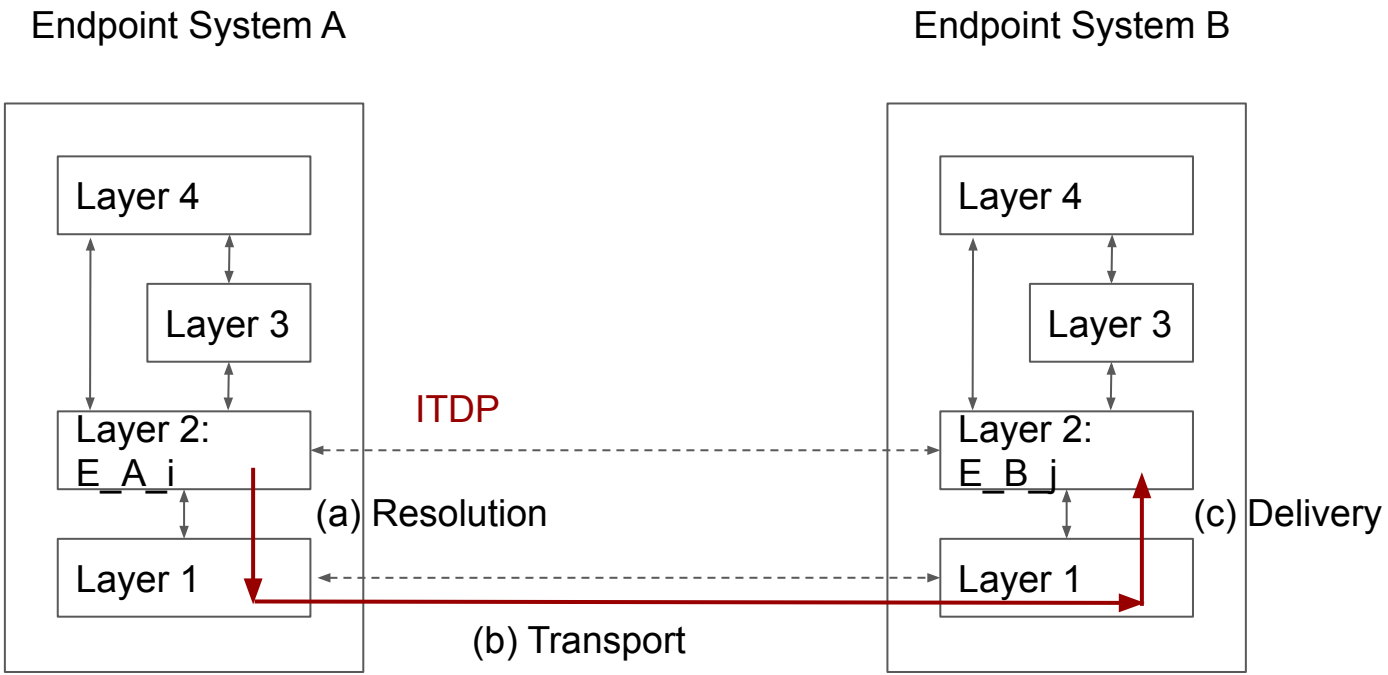


(18)ITDP defines a generic interface for management of identifications that supports all types of identifications. (e.g. major DID methods, AID/KERI, OpenID, other central/federated/distributed verifiable IDs.) This includes create, renewal/rotation, delete... operations.

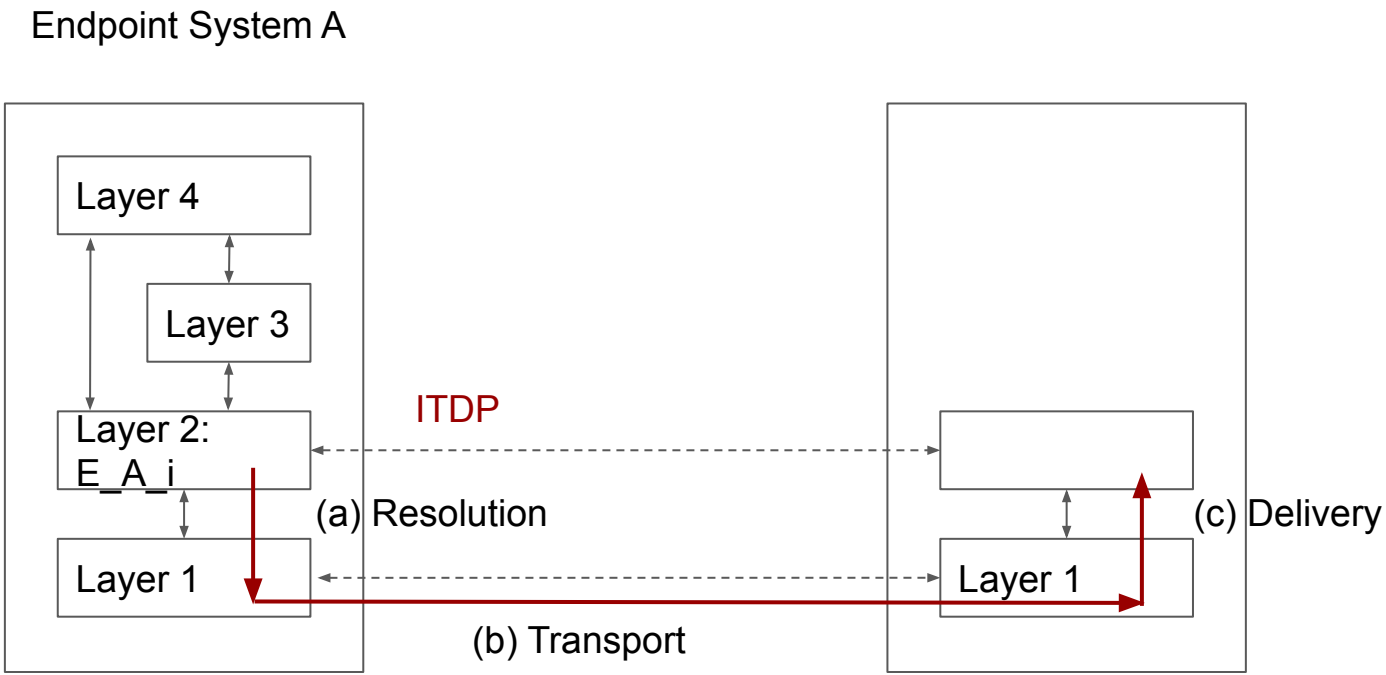
TSP is an Inter-Trust Domain Protocol (ITDP)



(19)ITDP defines a generic interface for verification of authenticity of all supported types of identifications.

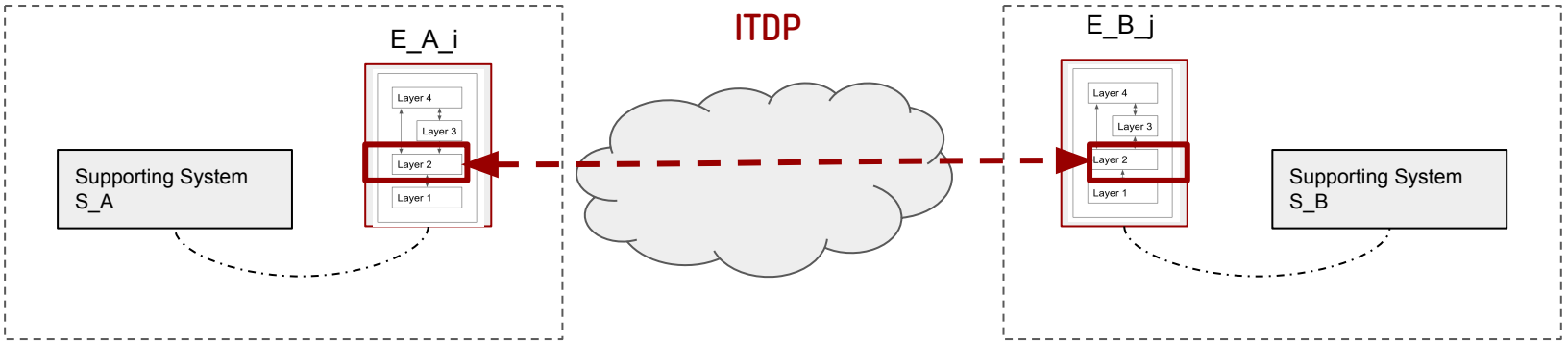


(20)ITDP defines a generic identification resolution, transport and delivery (RTD) procedure that can be efficiently realized in existing implementation methods and flexible enough for future adaptation. Note: this generic procedure is able to accommodate numerous variations of how it can be satisfied. Please do not assume it is a simple process as depicted in the generic diagram above.



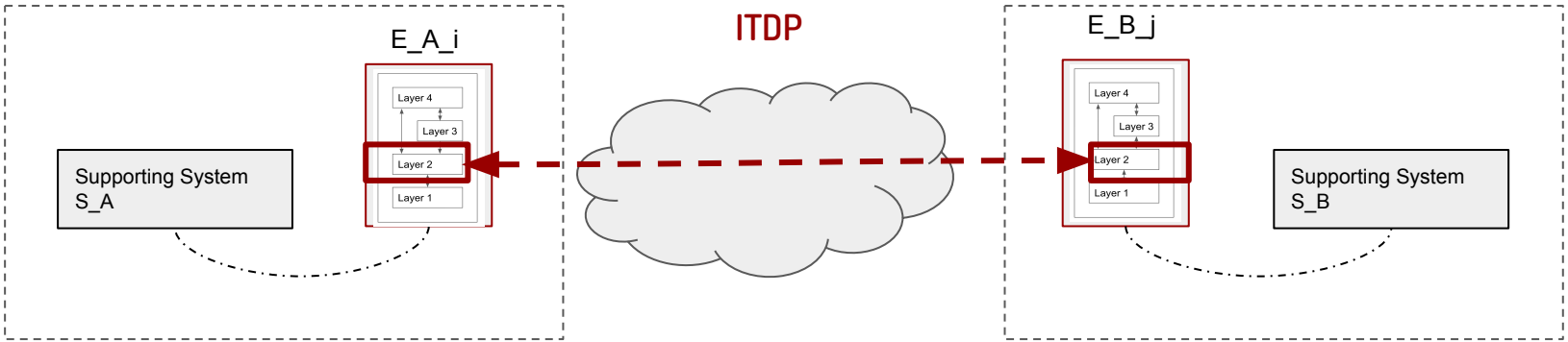
(21)As a special case, this RTD procedure can also support implicit destinations (e.g. without destination ID), including the default Internet without trust enhancement, or intermediaries. Note: if the destination does have a ID with special meaning, then it's the same procedure - may include additional handling.

TSP is an Inter-Trust Domain Protocol (ITDP)



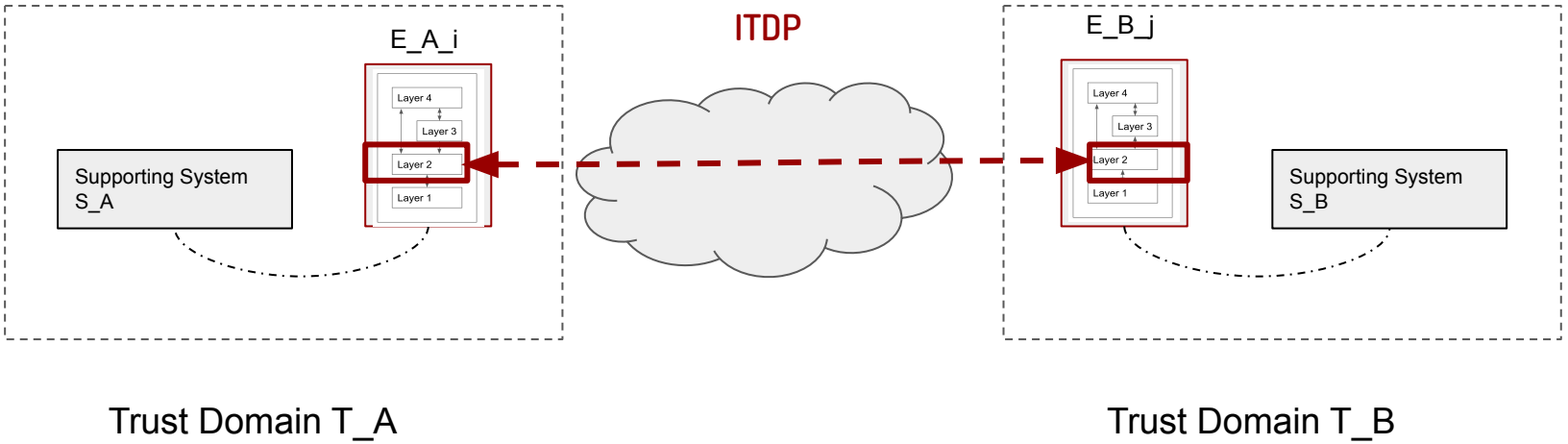
(22)ITDP defines a set of message types, format and how to use them in a dialogue (i.e. the language). This set is extendable. The minimum set is required which enables (a) the initial bootstrap (or introduction), (b) to initiate additional steps if needed, (c) to renew/update, delete, debug, or the equivalent/similar operations existed in various methods. (d) a generic type for upper layers. This minimum set must be defined in ITDP.

TSP is an Inter-Trust Domain Protocol (ITDP)



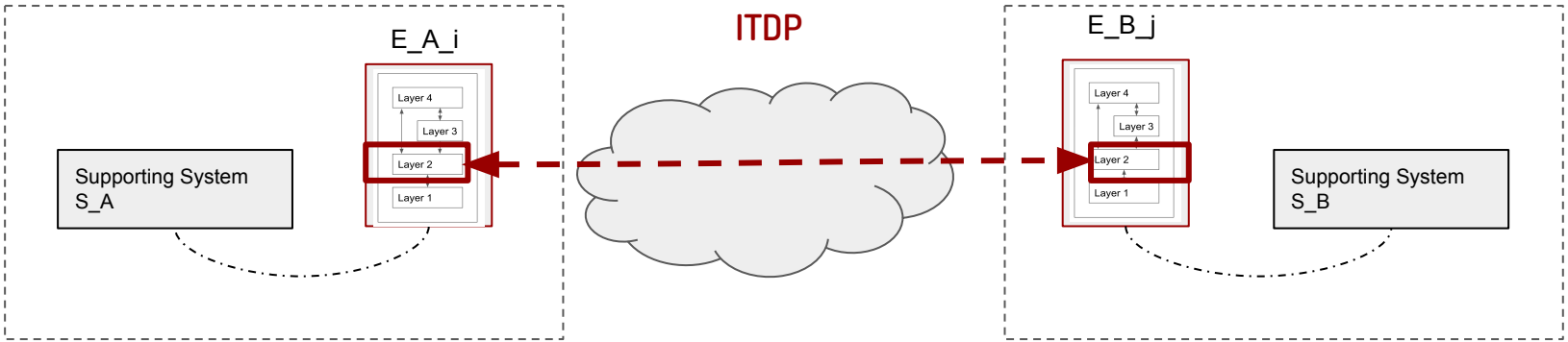
(23)ITDP can use any transport/messaging mechanism – including encoding schemes – existed today or created in future – in any of layers on Internet or any other means. The messaging mechanism may have any types of semantics as long as it can support the minimum unidirectional best effort delivery (similar to IP or UDP's assumptions in this regard). These transport/encoding mechanisms themselves are out of scope.

TSP is an Inter-Trust Domain Protocol (ITDP)



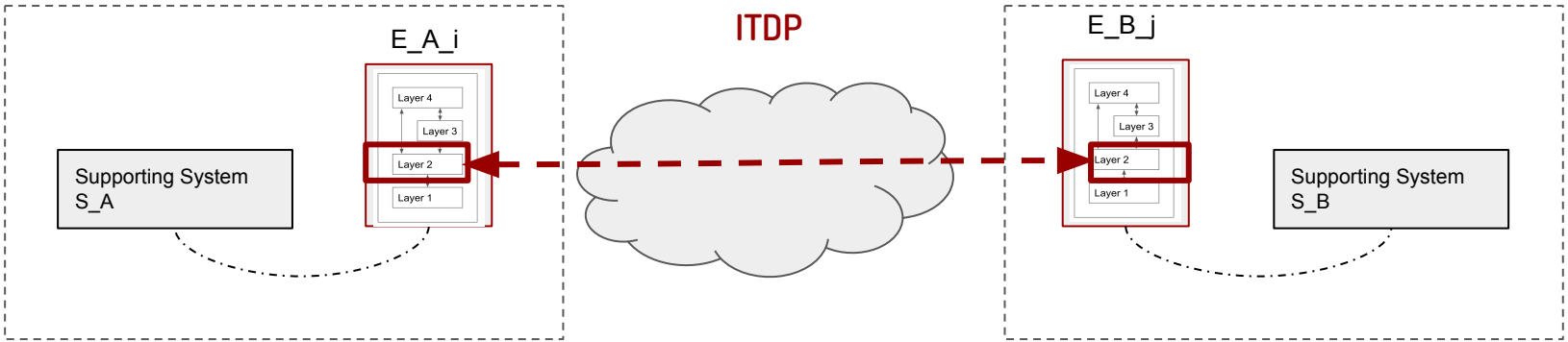
(24)ITDP messages incorporate primitives to protect their own content for message integrity, confidentiality and privacy. Some aspects of these may be left out optionally in some context. Note: again ITDP is not a tunneling or data path protocol.

TSP is an Inter-Trust Domain Protocol (ITDP)



(25)All used algorithms etc should be specified in a generic format for future evolution.

TSP is an Inter-Trust Domain Protocol (ITDP)



(26)In summary, ITDP is a decentralized, distributed, peer-wise protocol designed for establishing and managing public key based trust of authenticity across different trust domains, including trust domains that utilize different trust frameworks such as centralized, federated, decentralized or future evolutions of them.

(5)How various trust tasks can be built on top of ITDP (TSP)? How does ITDP relate to other protocols (forerunners)?

Recap on Principles & Architecture

Connectivity is the Goal, TSP is the tool, Apps are for End to End

2. Is there an Internet Architecture?

2.1 Many members of the Internet community would argue that there is no architecture, but only a tradition, which was not written down for the first 25 years (or at least not by the IAB). 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. This connectivity requires technical cooperation between service providers, and flourishes in the increasingly liberal and competitive commercial telecommunications environment.

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".

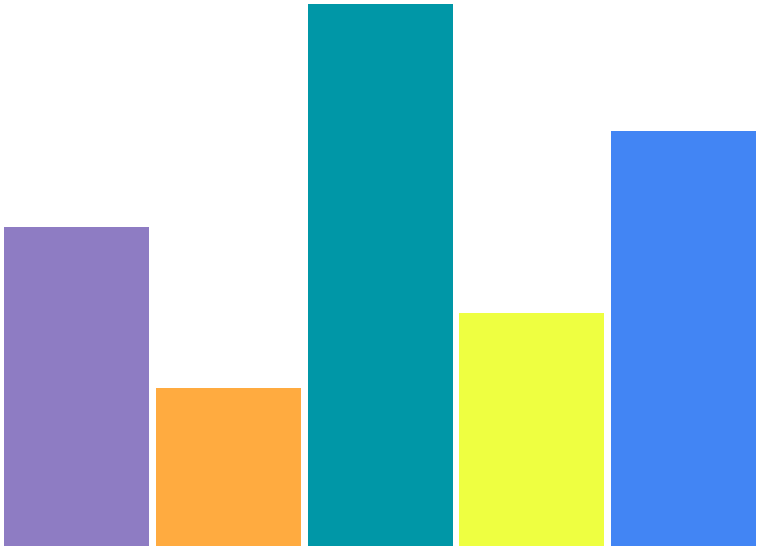
RFC 1958: “Architectural Principles of the Internet”.

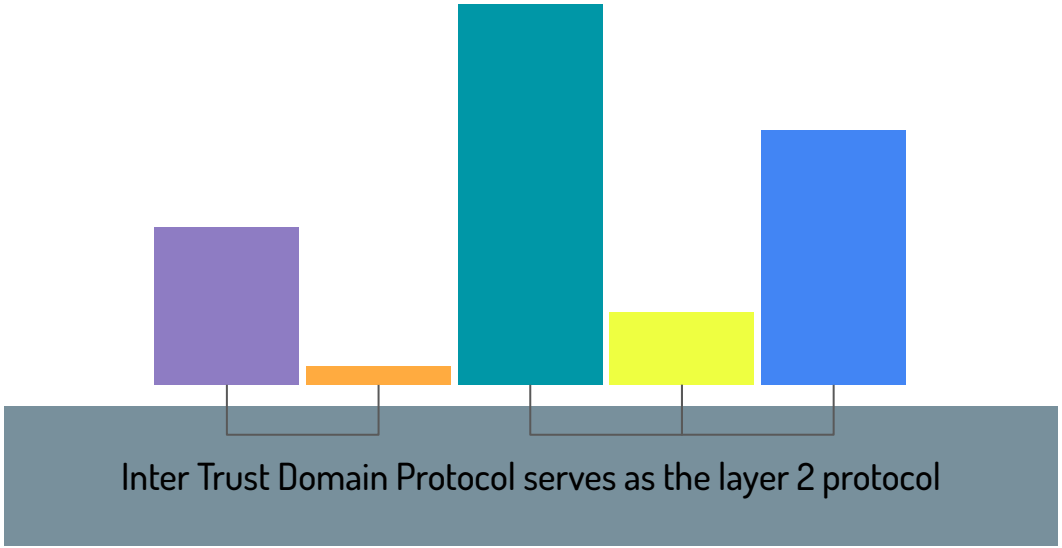
In our context: “Our aim is to maximize trustable connectivity, to achieve that we should design an inter-connect protocol with minimum foundational trust to bridge different trust domains together, and leave more intelligent features (i.e. other goals) to endpoints in higher layers.”

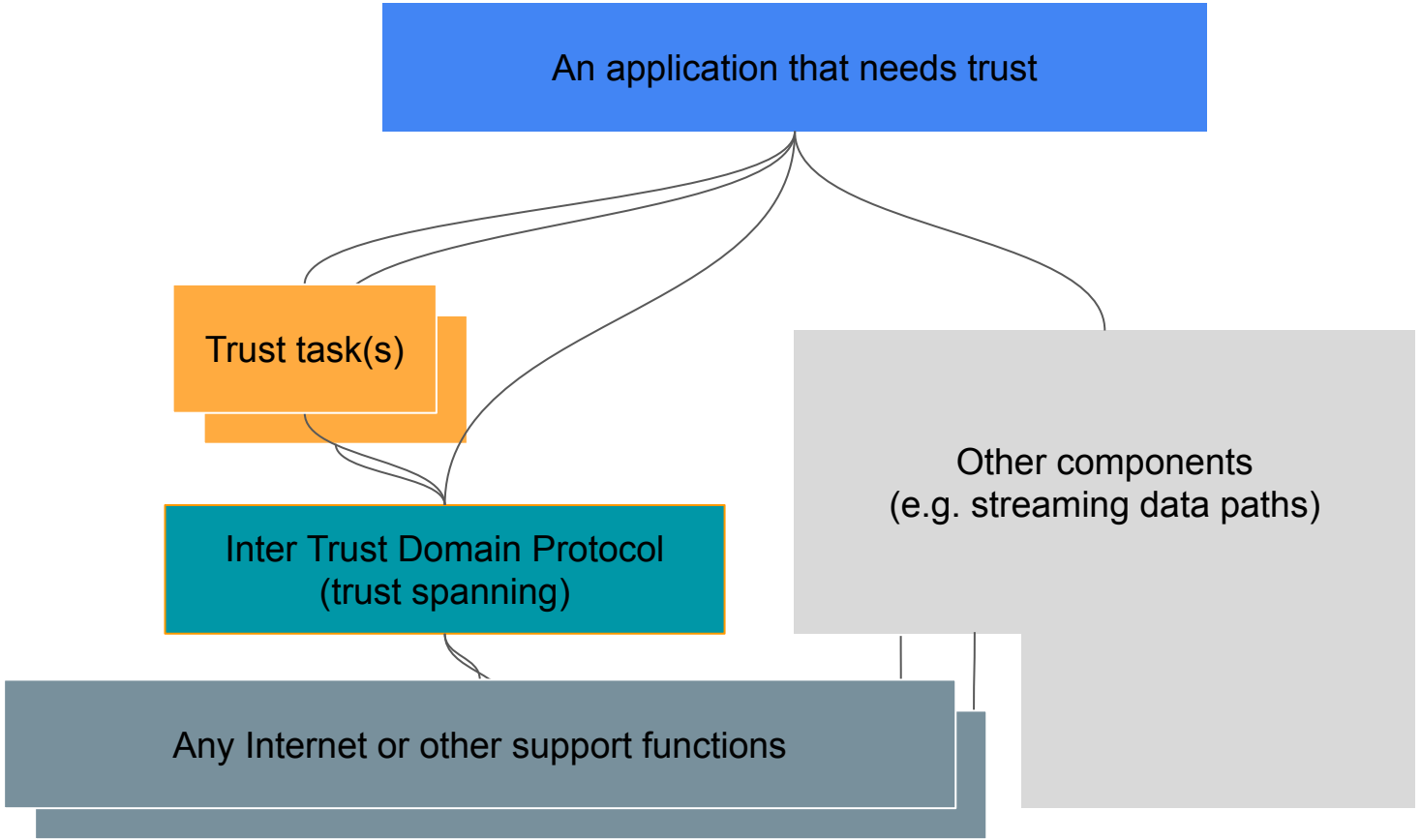
Backward compatibility or interoperability is a MUST.

Application level trust is an End to End concern.

Keep forward freedom!




Inter Trust Domain Protocol serves as the layer 2 protocol


An application that needs trust
Trust task(s)
Inter Trust Domain Protocol
(trust spanning)
Other components
(e.g. streaming data paths)
Any Internet or other support functions

Inter Trust Domain Protocol inter-connects trust domains with interoperability.
Interoperability provides “liquidity”.
And “liquidity” provides “pressure” towards better trust featured solutions that applications need.
ITDP itself does not dictate which solution is better.

ITDP is that Inter-Connect protocol. Therefore, when I say what ITDP shouldn't do X or X is out of scope…

It doesn't mean X shouldn't be done or discussed.

It doesn't mean X is unimportant or uninteresting.

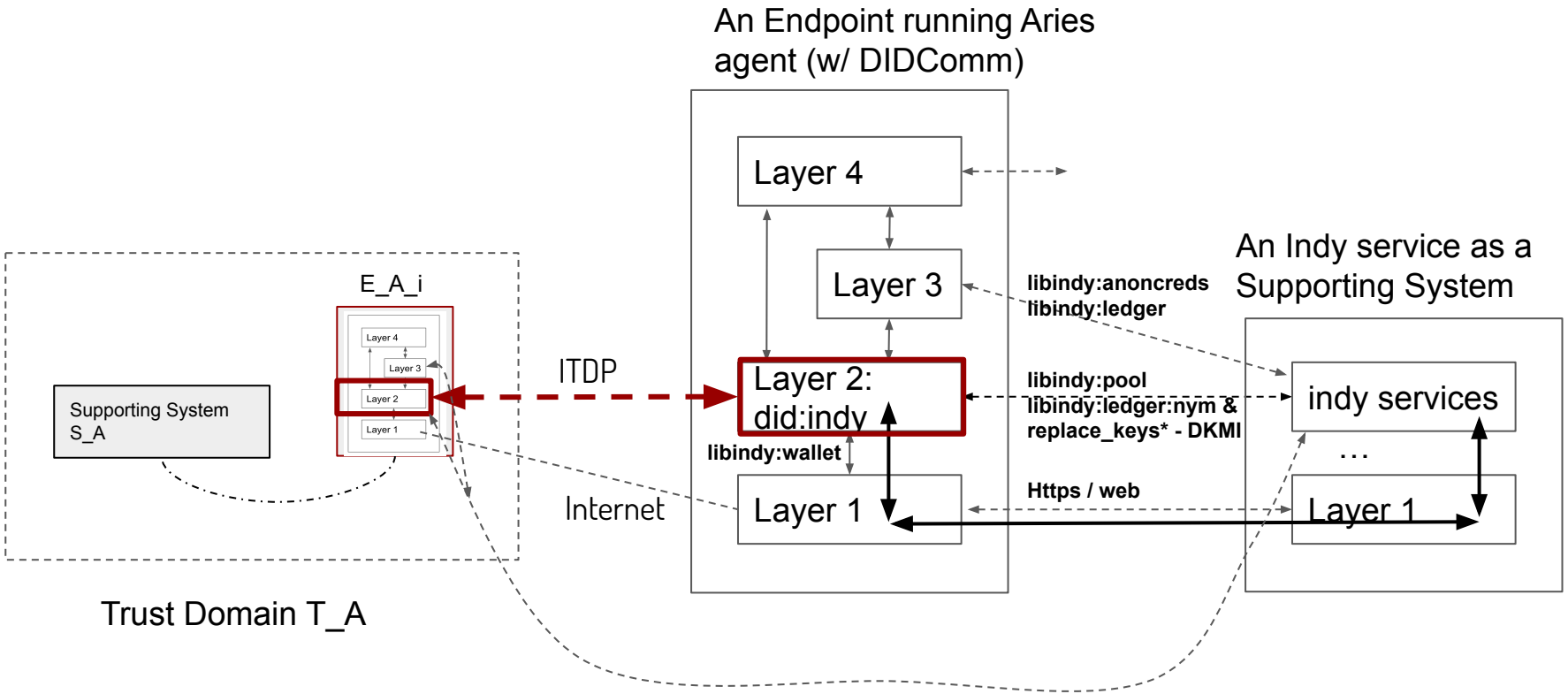
It doesn't mean X is not related.

It just means X should not be included in ITDP proper because it is not serving the goal of maximizing trustable connectivity or X is not minimally required.

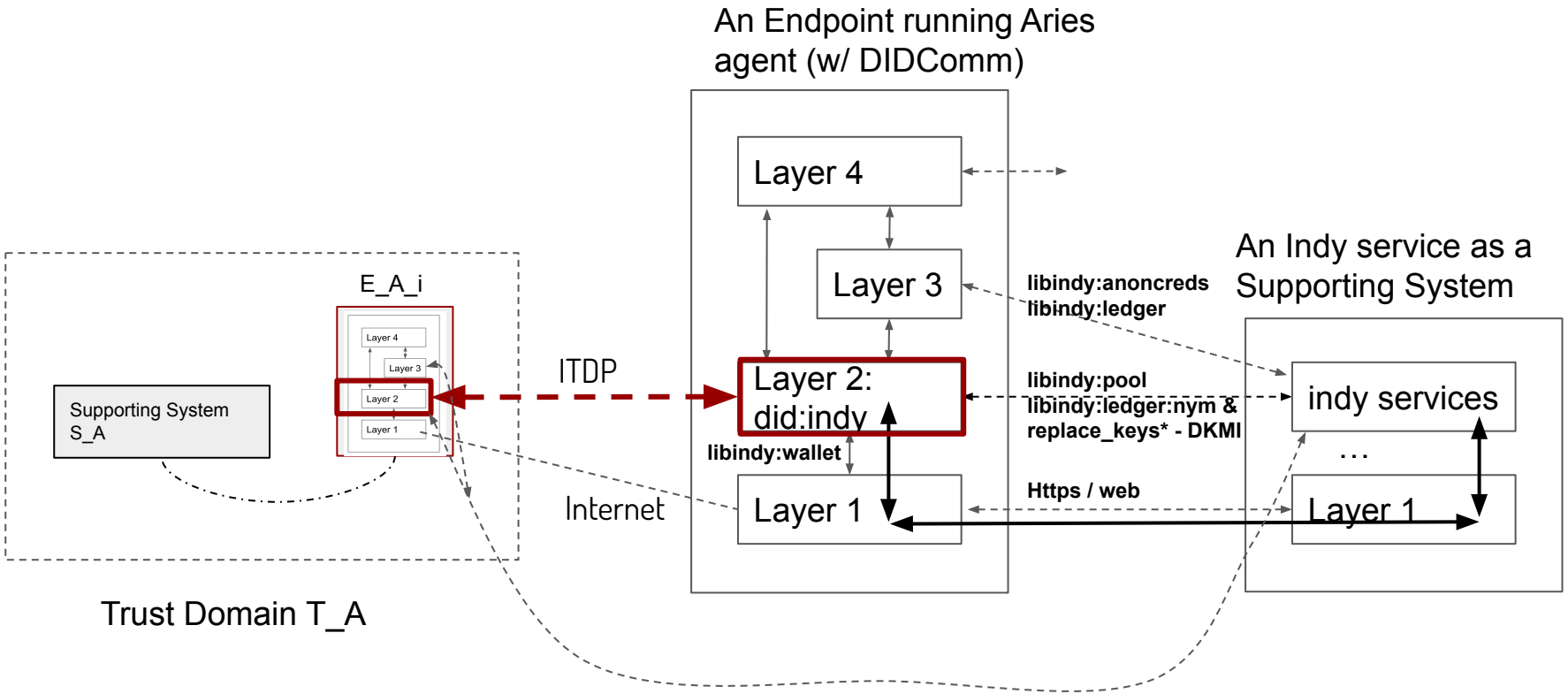
Less is More
(again)

Recap on ITDP
(go back to Slide 21 for Section 3
And Slide 35 for Section 4)

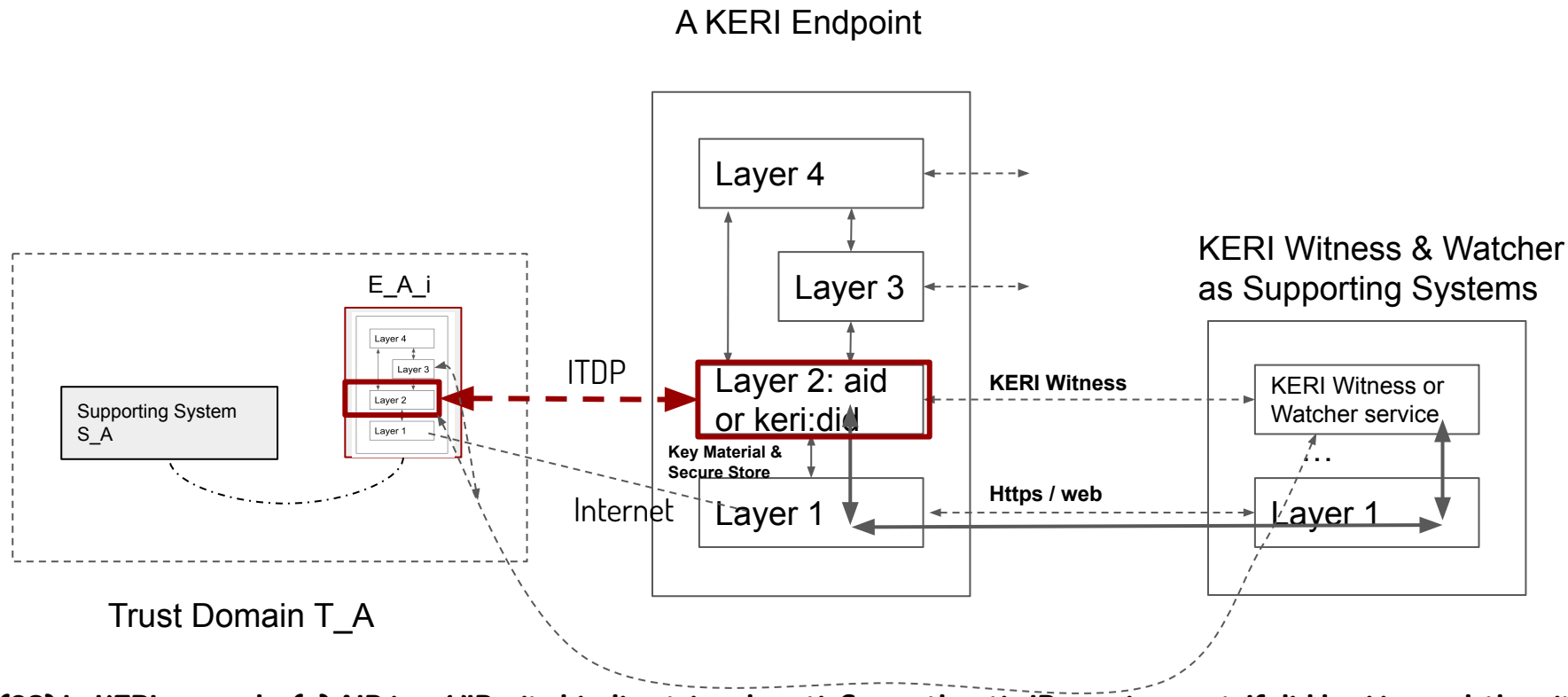
Case Studies



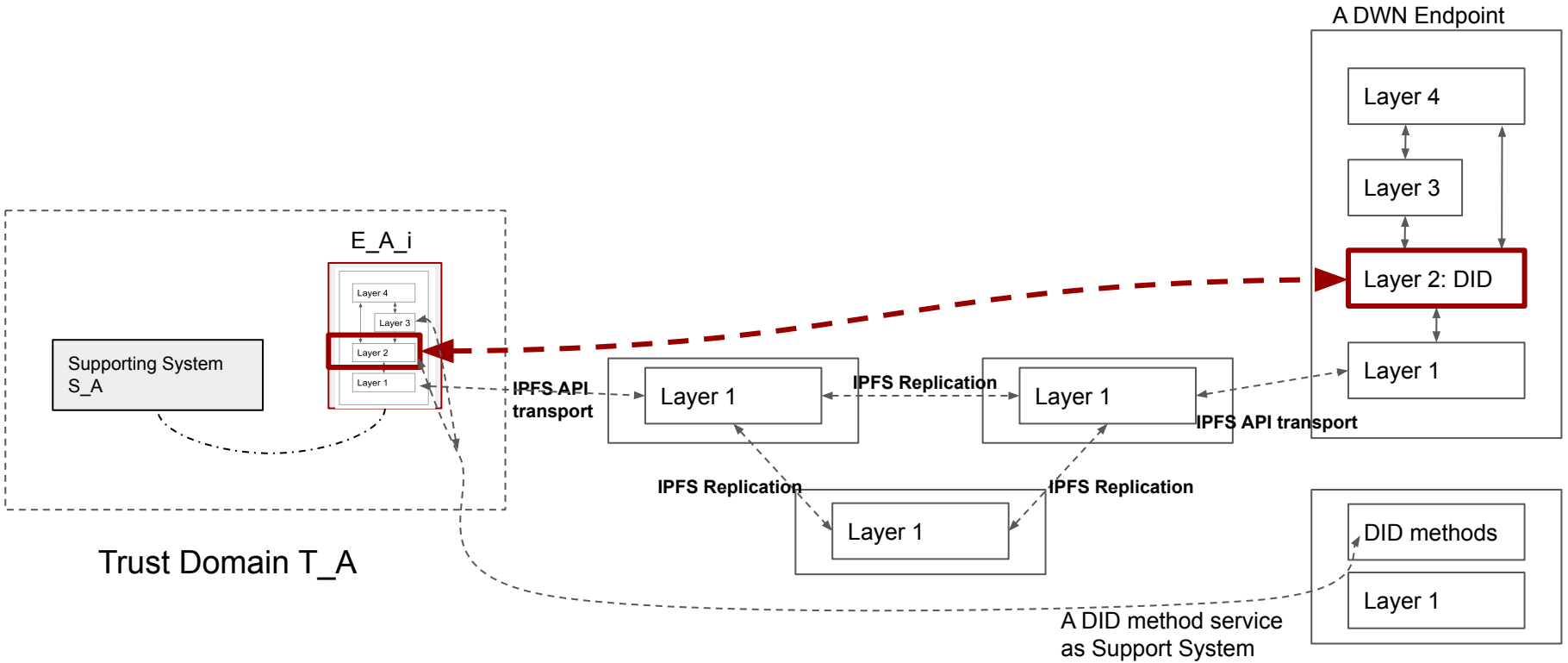
(27) In Aries example, (a) did:indy is-a DID which is-a VID. DID has a general resolution procedure. Aries uses DIDComm Messaging (v1 or v2) which uses JOSE (JWM, JWS, JWE, JWK…) messages so it also satisfies the authenticity requirement. (b) But DIDComm is tied to DID (does not interop with non-DIDs); it only support unicast i.e. tied to communication patterns; it often ties to Web/DNS; and also has unnecessary extras (e.g. routing Intermediary). A future iteration of DIDComm could be decomposed/refactored to be based on ITDP + other parts.



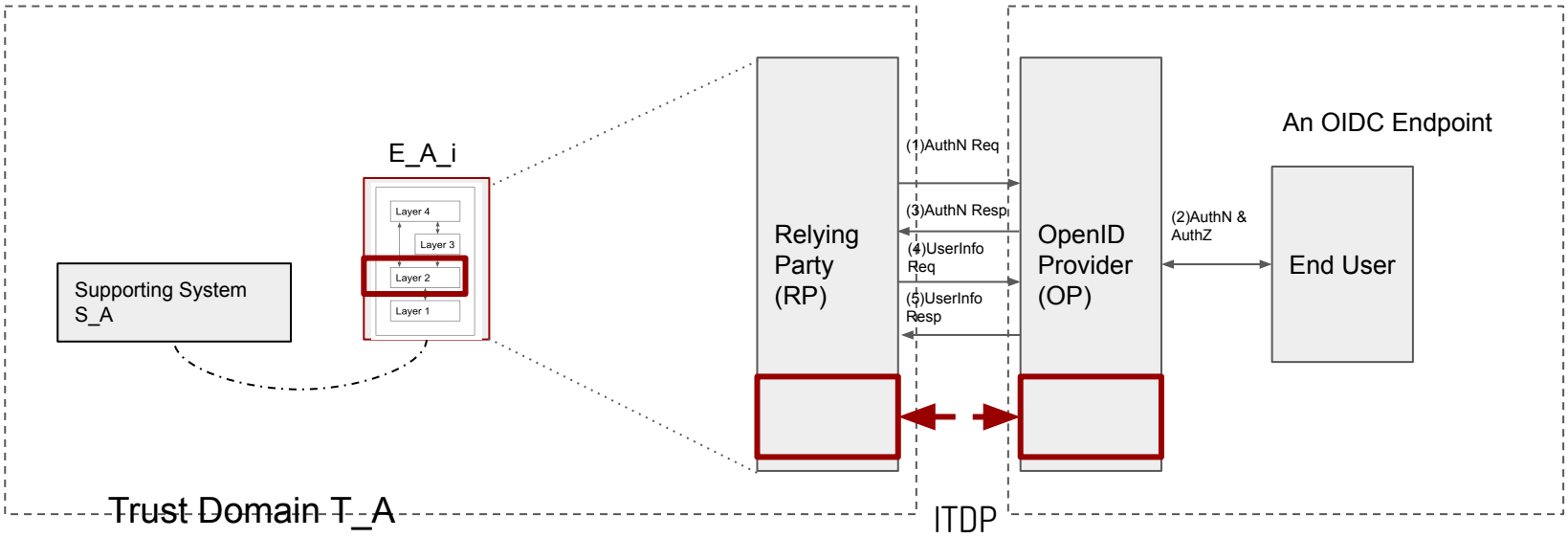
(28) In Aries example, DIDComm (v1 or v2) provides DID and JOSE messages to higher layers (layer 3 or 4). The messages support extensions and generic types (e.g. attachments) to the higher layers. Anoncreds can be implemented as a layer 3 protocol. So can VC issuance and presentation be implemented in similar ways. Aries libraries (at least older versions) however do not cut this separation cleanly between layers, but a refactoring can be done with modest efforts. All drawbacks of layer 2 of course spread to higher layers as well.



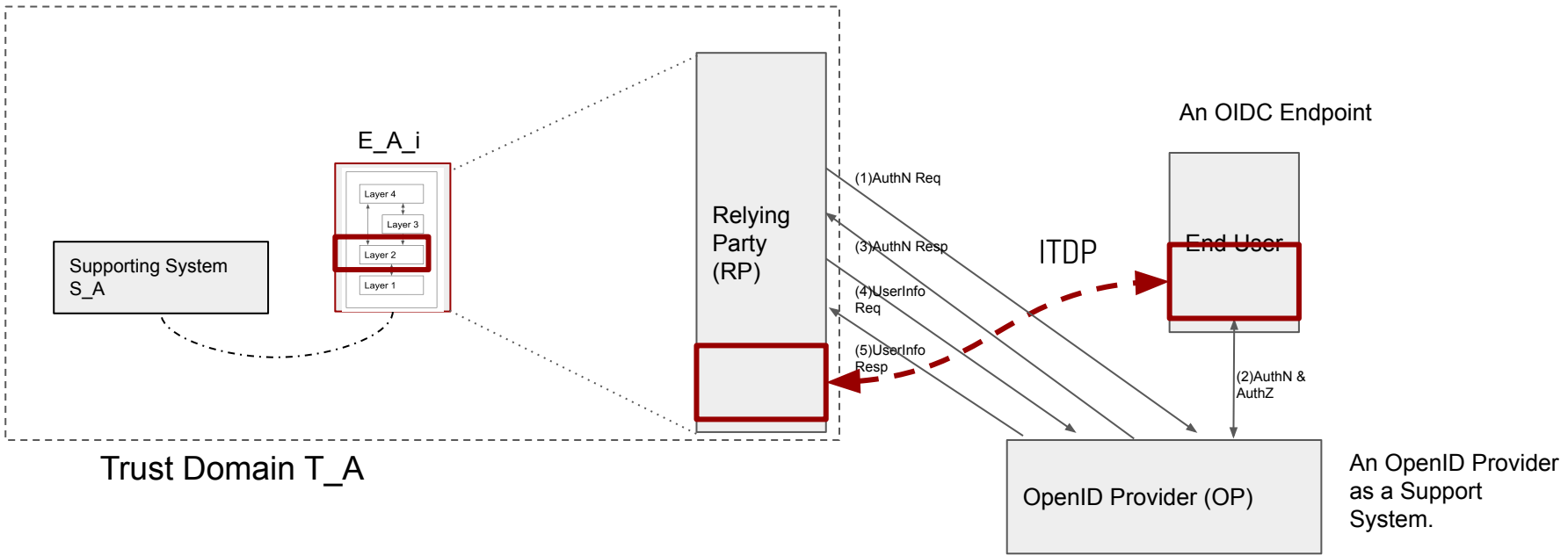
(29) In KERI example, (a) AID is-a VID - its binding triangle satisfies authentic ID requirement. If did:keri is used, then it has a general resolution procedure. The KERI authencatible message meets authenticity requirement & compatible. (b) Other aspects of KERI focuses on stronger properties of AID - those are orthogonal to ITDP & compatible. (c) KERI does not support non-AIDs, ITDP can be the inter-connect protocol for the endpoint to talk to a non-KERI party. (d) CESR is in layer 1 that can support ITDP (compatible). What we need is a generic AID verification & to agree on shared msg formats (e.g. self-desc). KERI currently does not specify non-unicast communication patterns.



(30) In DWN example, (a) It uses DID, so it satisfies authentic ID requirement & has a general resolution procedure. (b) All DWN messages are also JOSE messages - these definitions can be easily compatible with ITDP. (c) DWN chose IPFS DAG on CBOR - this is a transport choice and is fully compatible with layer 1 as shown above. (d) The Records in DWN is a trust task in layer 3. The same is true for DWN Protocols, Permissions, Hooks etc. In summary, DWN can be refactored to be built on top of ITDP, and by doing so it obtains trust interoperability to other implementations or functions.



(31) In OIDC - SIOP example, (SIOP: Self-Issued OpenID Provider), the RP is one endpoint, the SIOP+End User is the other endpoint. As shown above, the RP and OP can be refactored to use ITDP for interoperability with other non-SIOP endpoints. In addition the OIDC protocol interactions (1,3,4,5) become layer 3 trust tasks (authn & authz). (Pros & Cons)



(32) In OIDC (or other centralized or federated solutions), interoperability through ITDP can also be achieved with a change to the endpoints while keeping OP infrastructure unchanged. In this design, the OP serves as a Support System for an OIDC endpoint - this matches its original function. The other endpoint integrates RP protocol (1,3,4,5, authn & authz) on top of ITDP to provide this level of interoperability. (Pros & Cons)

In summary, ITDP serves the role of TSP very well. I reach this conclusion by studying some of the common solutions we are familiar with in terms of its ability to support respective trust tasks in upper layer and interoperability through ITDP with endpoints outside of its designed trust domain.

ITDP has the minimum required foundational trust upon which other stronger or diverse types of trust tasks and applications can be built. It is minimum - because removing either of its two main requirements will break the above property.

(6)How lower layer support functions may implement what ITDP needs?

ITDP Maintains High Degree of Layer 1 Endpoints & Other Support & Intermediary Systems' Implementation Freedom

ITDP to Layer 1 Support Functions - Brief Summary

- Generic wallet interface
- Network transport
  - Transport independent
  - Communication pattern independent (e.g. does not require a separate protocol to support multicast, streaming etc.)
  - Message encoding independent
  - DWN-style cloud medium very well supported
- Can be extremely efficient to support Mobile, IoT/sensor and other battery powered devices
- Can be readily integrated into Metaverse/XR devices
- Can be readily integrated into virtual Cloud Native services
- Backward compatible
- Forward freedom

Further discussion on implementation is for a future presentation.

(7) Q&A

- Public Reference to this doc should be in its entirety and as: {Wenjing Chu, "TSP is an Inter-Trust Domain Protocol (ITDP)", Feb 21, 2023. https://github.com/wenjing/Inter-Trust-Domain-Protocol}.
- Presentation on the Reference Architecture at IIW-34:
  - https://youtu.be/QZssxxZ9f88
  - https://github.com/windley/IIW_homepage/blob/gh-pages/assets/proceedings/IIW_34_Book_of_Proceedings.pdf
- ToIP TechArch specification: https://github.com/trustoverip/TechArch