Reference Architecture for Trust over IP

Wenjing Chu

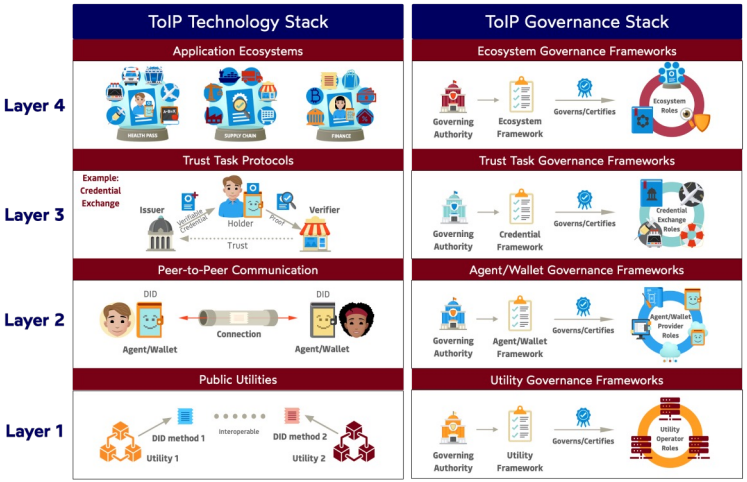
First Draft: April 20, 2022
Second Iteration: April 22, 2022
Third Iteration: April 25, 2022 - Internet Identity Workshop (IIW-34)
Revised for ToIP Dublin Mini Summit - September 11, 2022

https://github.com/windley/IIW_homepage/raw/gh-pages/assets/proceedings/IIW_34_Book_of_Proceedings.pdf

- This presentation was first given at the 34th IIW , April 26-28, 2022 in Mountain View, CA.
- Minor revisions for the Mini Summit of ToIP - September 13-16, 2022 - Dublin, Ireland.
- Also added a slide to update current work in progress at the ToIP Technology Architecture Task Force:
  - Github: https://github.com/trustoverip/TechArch
  - Slack: #tswg-tech-arch-tf
  - Wiki: https://wiki.trustoverip.org/display/HOME/TSWG+Technology+Architecture+Task+Force
- For a general introduction, I had a breakout session presentation on Wednesday 12:10 - OSSummit Dublin: https://sched.co/15z13.

Stack vs. Reference Architecture

The ToIP Stack View



Reference Architecture

?

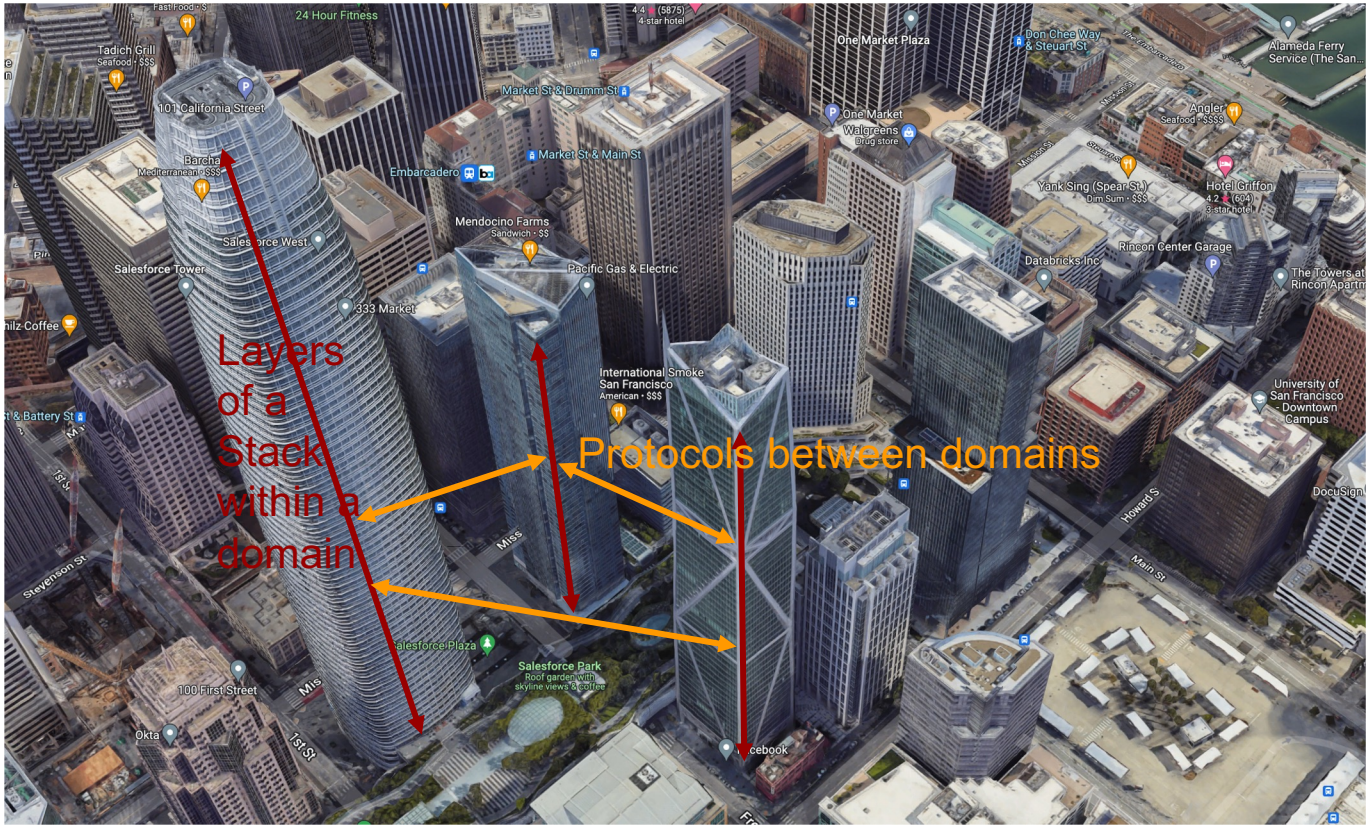
My Topic Today

- A Reference Architecture of a complex system is an abstract framework consisting of a list of component subsystems and interactions with each other and with external systems.

- It is a generalization of various viable solutions.

- It helps crystalize the most important architecture considerations while leave other details for substantiation (*by component specifications*).

A *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 a Domain where dependencies are clearly ordered. But it is not suitable to capture relationships between Domains. The Reference Architecture is a prerequisite to understand a Stack.



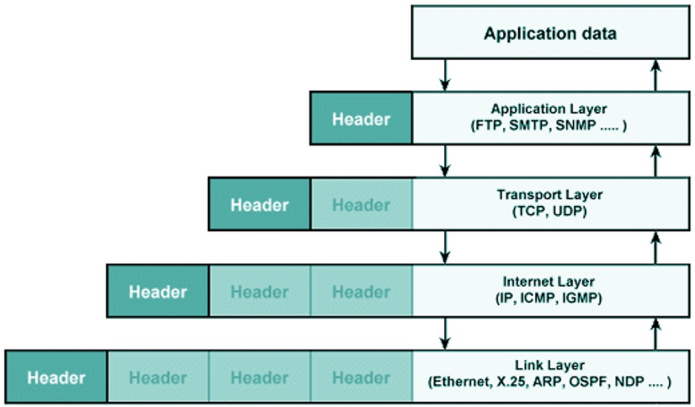
An analogy for a Reference Architecture

Wenjing Chu

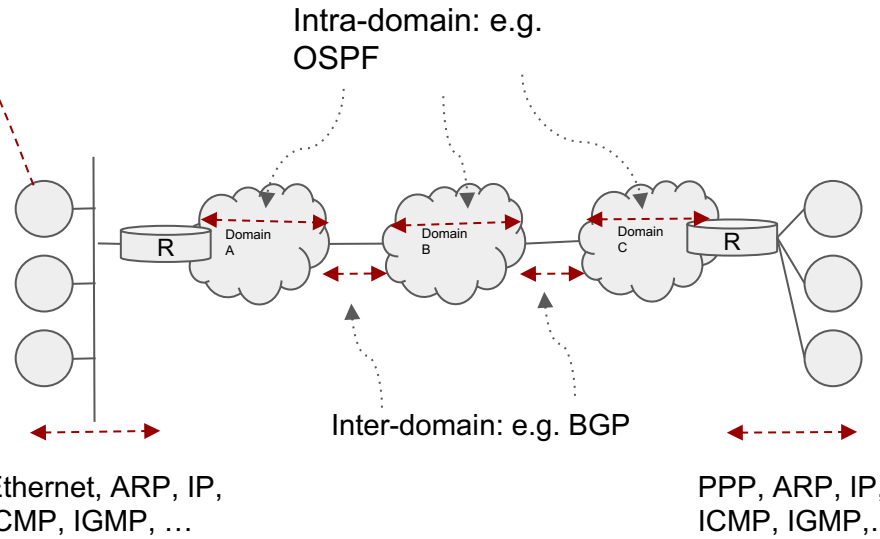
For example: Internet architecture

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

The stack on an end system



The Internet reference architecture

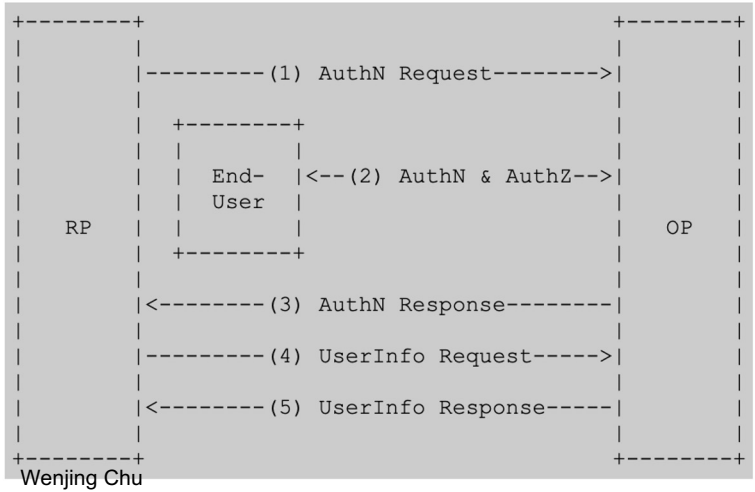


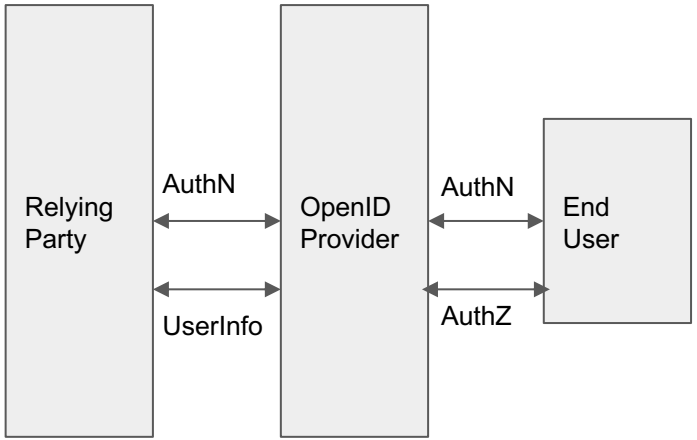
Another example: OIDC in the reference architecture view

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:





The Most Important Considerations for Trust over IP

- Universal Connectivity
  - aka Reachability, Interoperability
  - Hourglass
  - End-to-End
- Decentralization
- Authenticity
  - Verifiability
- Confidentiality, Privacy

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

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

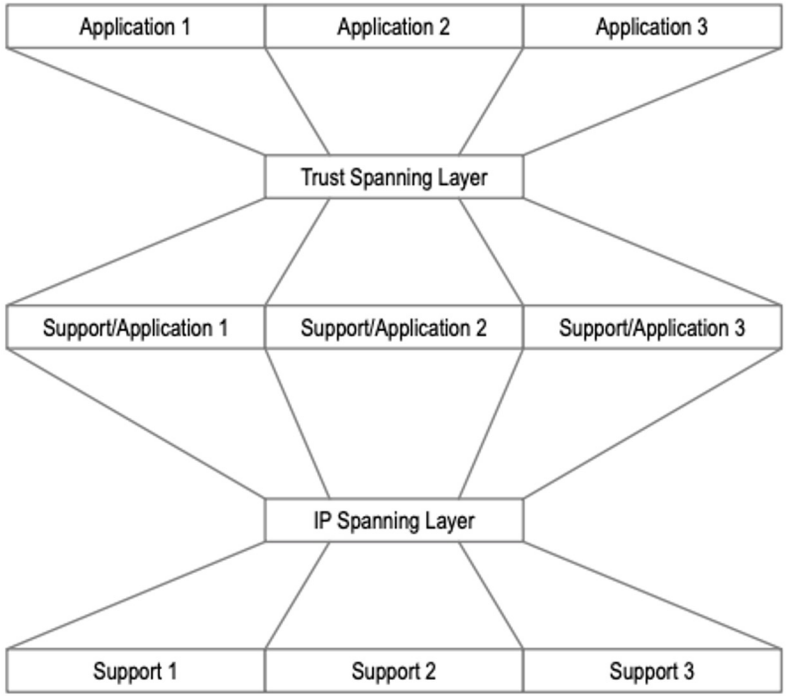
#1: The End-to-End Principle
#2: Connectivity Is Its Own Reward
#3: The Hourglass Model
#4: Decentralization by Design and Default
#5: Cryptographic Verifiability
#6: Confidentiality by Design and Default
#7: Keys at the Edge

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

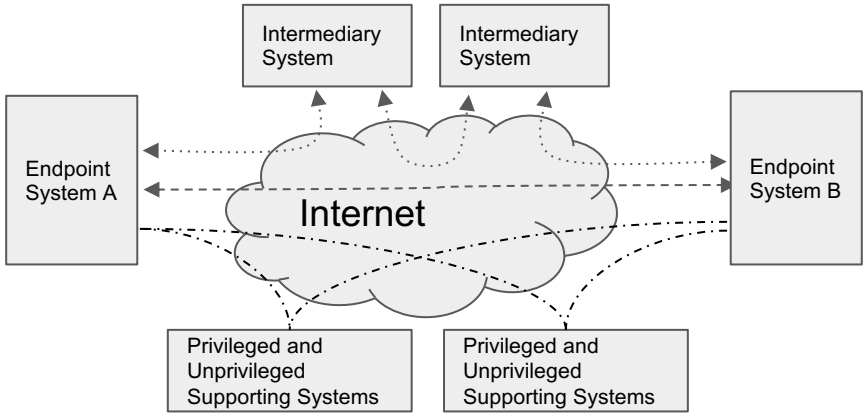
#8: Trust is Human
#9: Trust is Relational
#10: Trust is Directional
#11: Trust is Contextual
#12: Trust has Limits
#13: Trust can be Transitive
#14: Trust and Technology have a Reciprocal Relationship

Part Three: Overall Principles

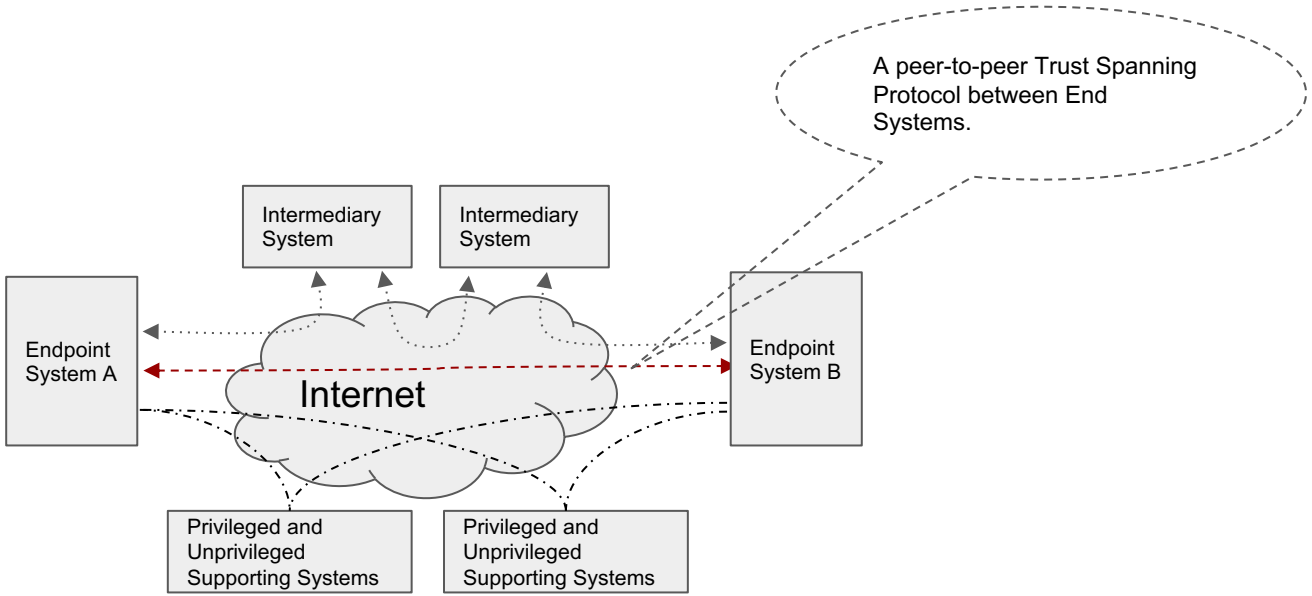
#15: Design for Ethical Values
#16: Design for Simplicity
#17: Design for Constant Change

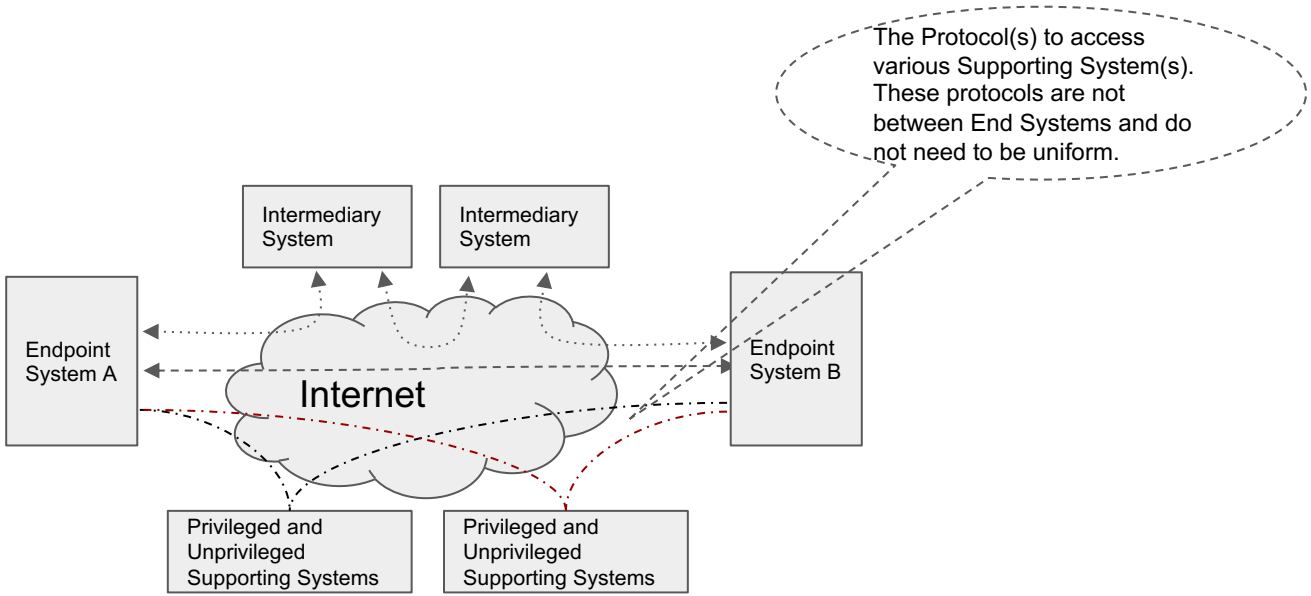

Application 1
Application 2
Application 3
Trust Spanning Layer
Support/Application 1
Support/Application 2
Support/Application 3
IP Spanning Layer
Support 1
Support 2
Support 3

Reference Architecture

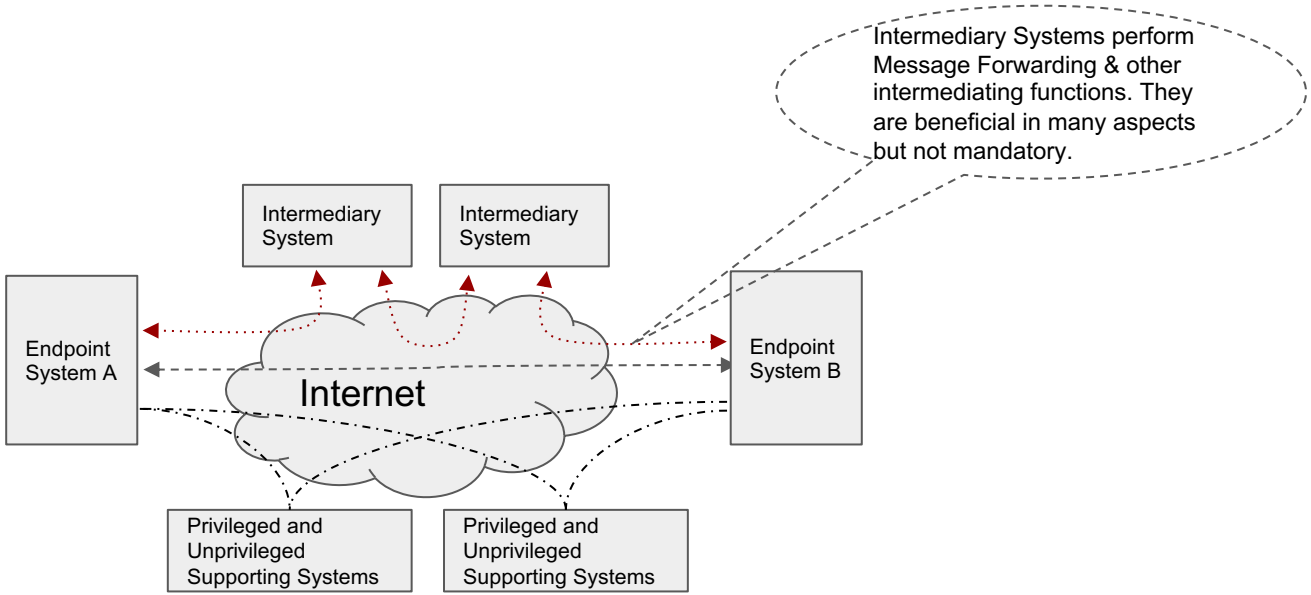


- Subsystems are delineated by **locus of control (domain)**
- They interact through **a set of protocols**, not just one.
- Each **type** of subsystems has a shared stack*, but the stack is not identical across different types of subsystems**.

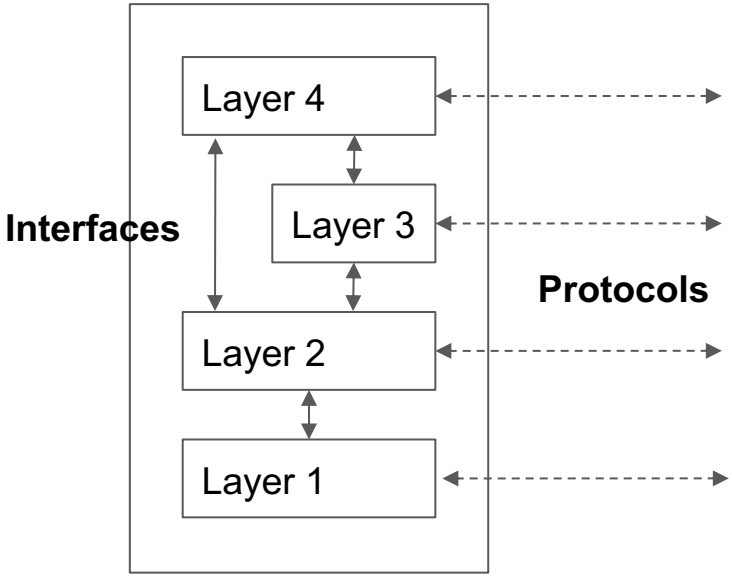

A peer-to-peer Trust Spanning Protocol between End Systems.
Intermediary System
Intermediary System
Endpoint System A
Internet
Endpoint System B
Privileged and Unprivileged Supporting Systems
Privileged and Unprivileged Supporting Systems



Examples of Supporting Systems: VDRs, Witness, Watcher, Accounting/Auditing, Reputation, Discovery, …

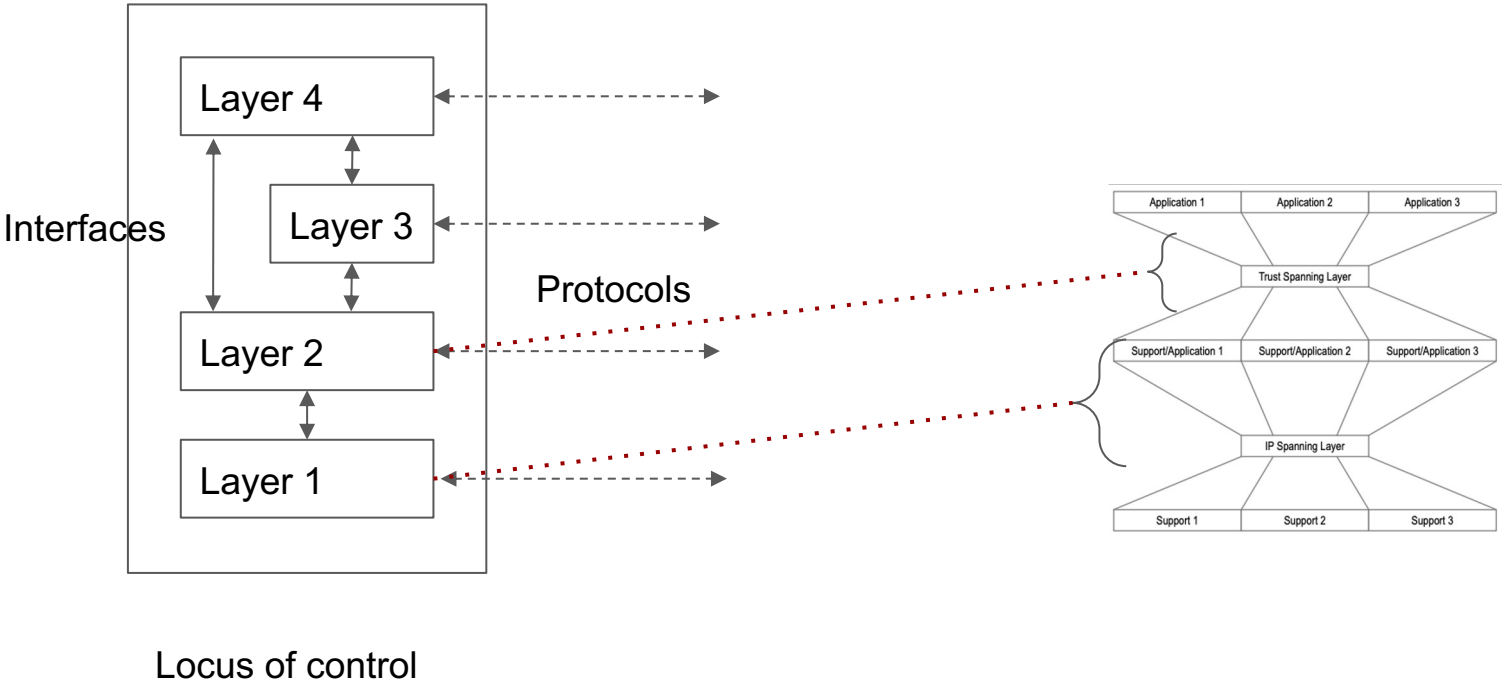

Intermediary Systems perform Message Forwarding & other intermediating functions. They are beneficial in many aspects but not mandatory.
Intermediary System
Intermediary System
Endpoint System A
Internet
Endpoint System B
Privileged and Unprivileged Supporting Systems
Privileged and Unprivileged Supporting Systems

An Endpoint System

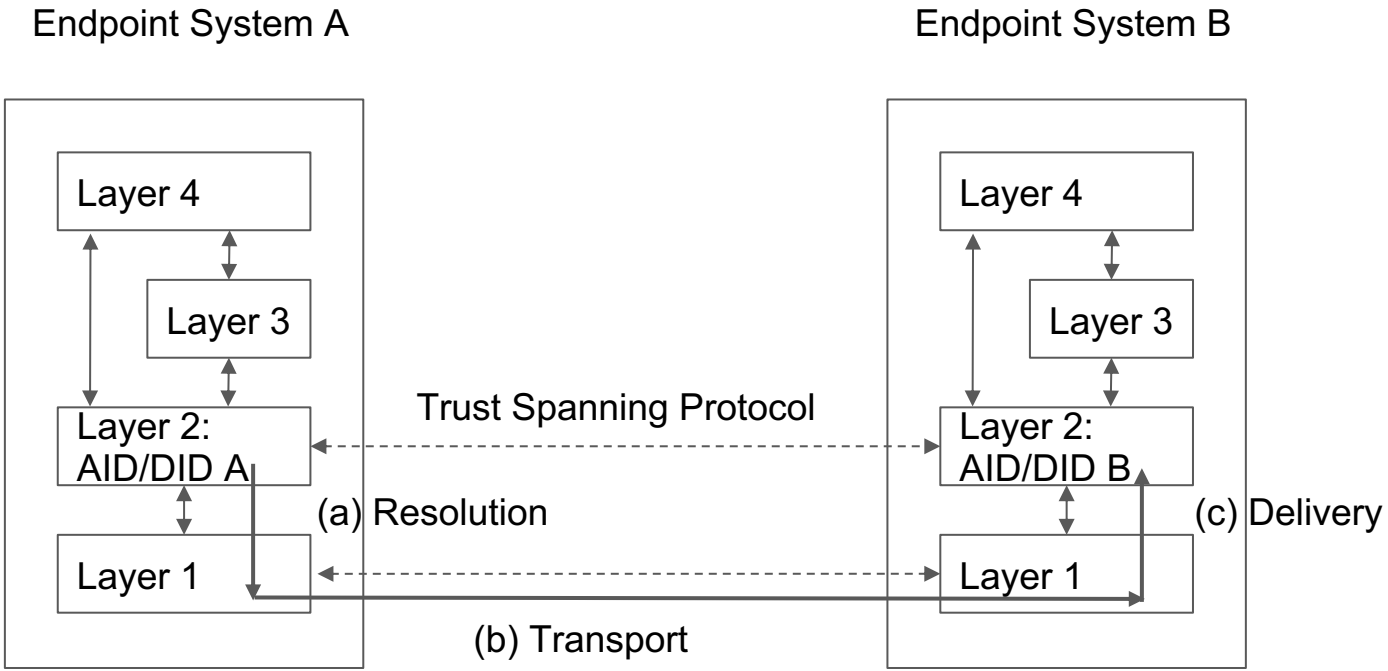


Locus of control

An Endpoint System



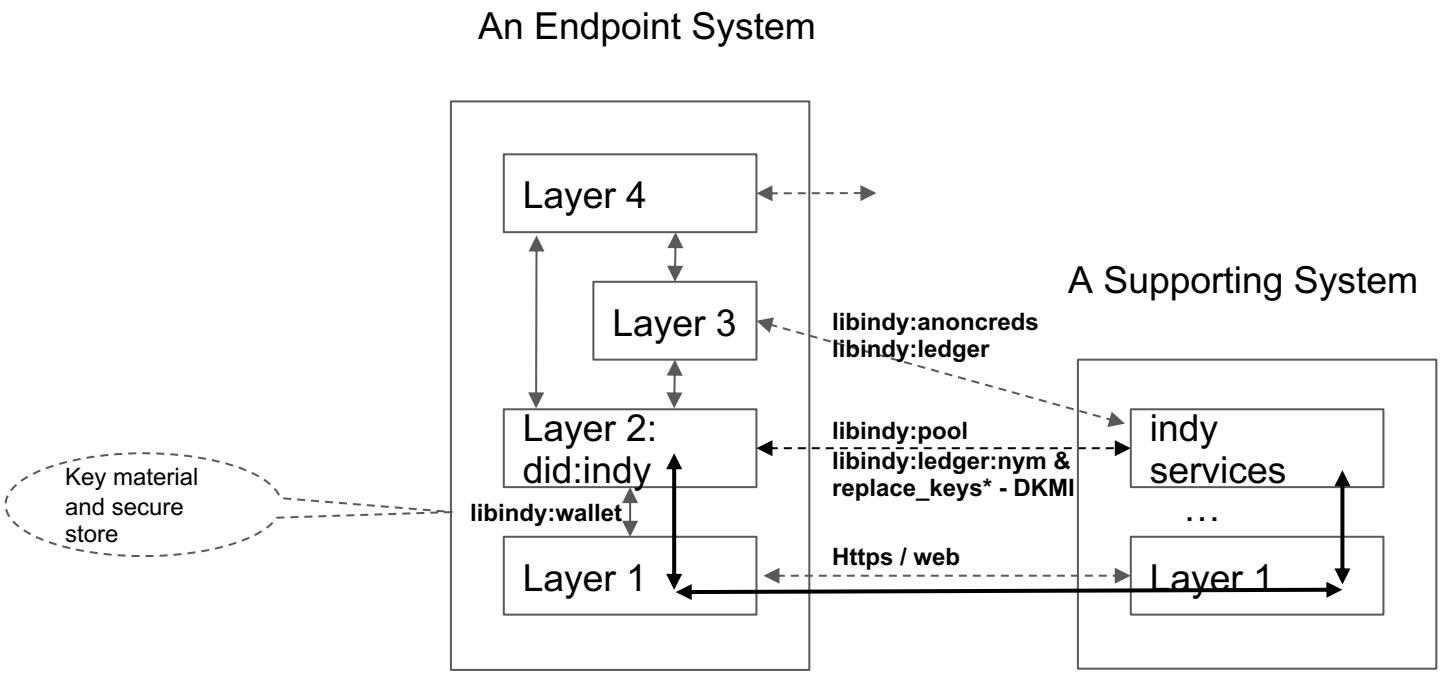
Locus of control



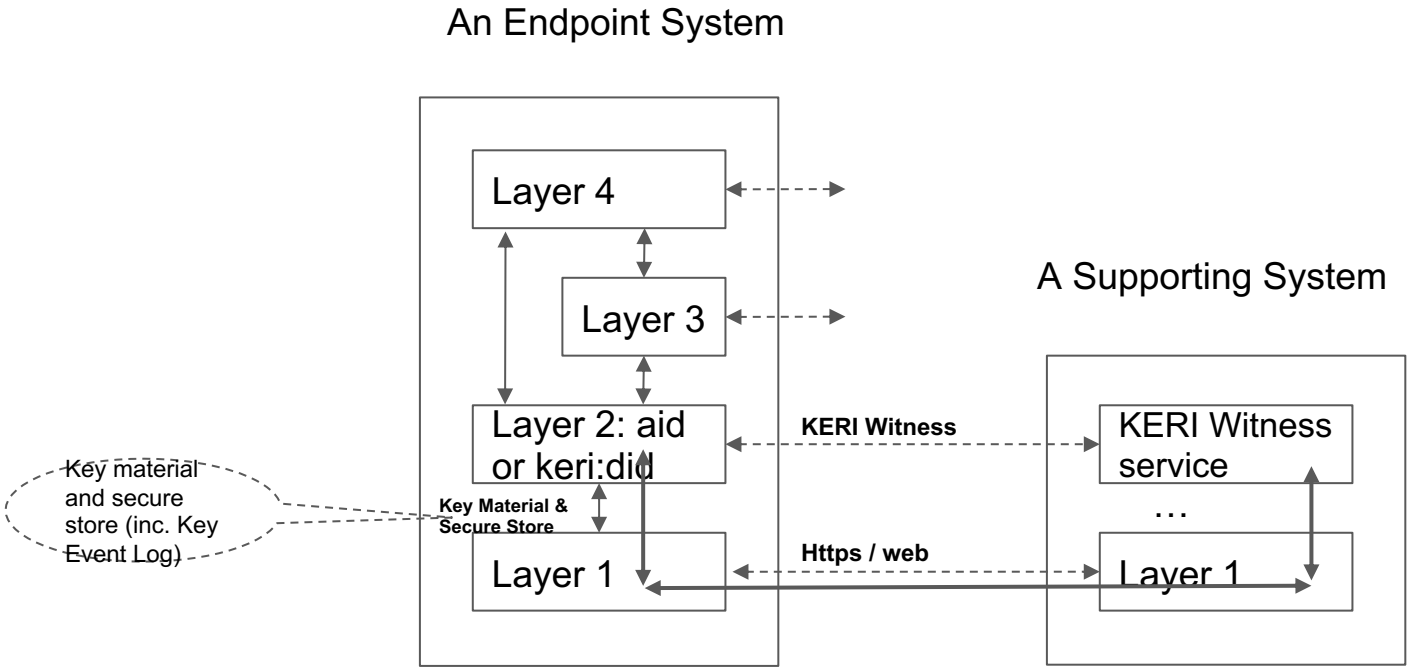
What Layer 2 needs and *only* needs: (1) AID/DID; (2) An end-to-end verifiable messaging protocol***. Of course, **with proper properties for each**: **(i) authenticity (ii) then confidentiality as needed (iii) then privacy as needed**

What Layer 2 needs (Trust Spanning Layer)

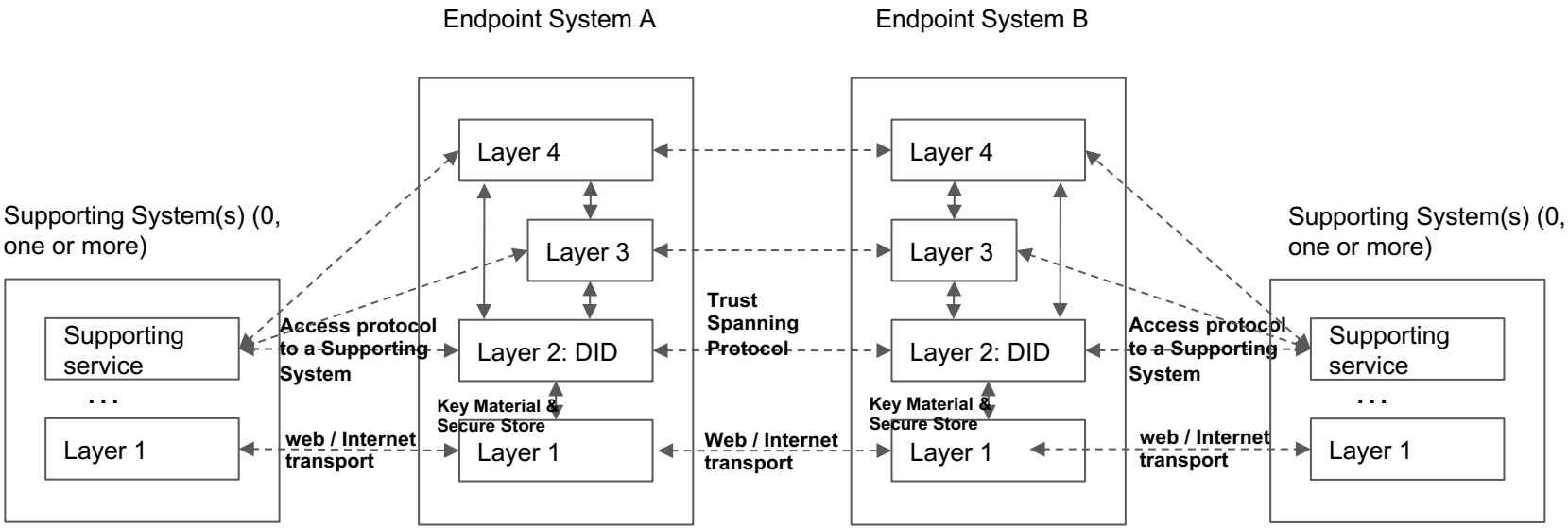
1. **AID/DID + An end-to-end verifiable messaging protocol**
2. **Properties:**
    a. **Authenticity (necessary and sufficient)**
    b. **Confidentiality by choice**
    c. **Privacy by choice**



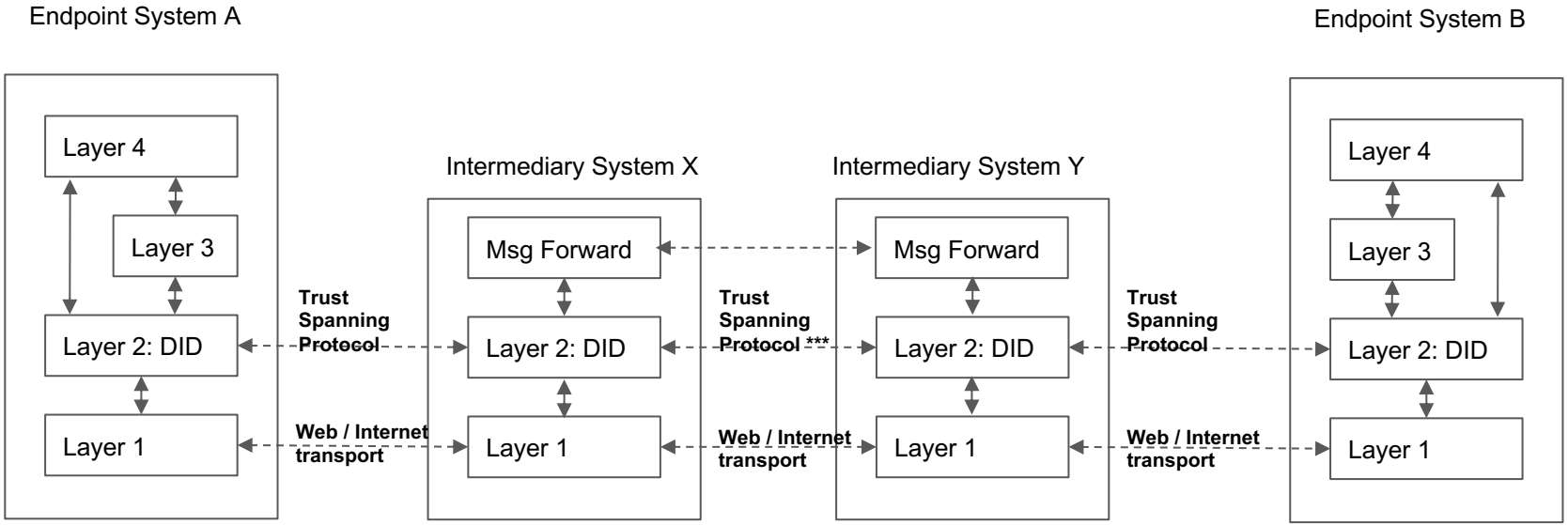
Indy-Aries example



- KERI example: Witnesses with Key Event Logs.
- KERI uses other supporting systems (e.g. Watcher for confirmation) in addition to Witness pool.
- As long as such services are required for the functioning of AID and E2E communication, they belong to Layer 2 and the pattern shown here should still work.

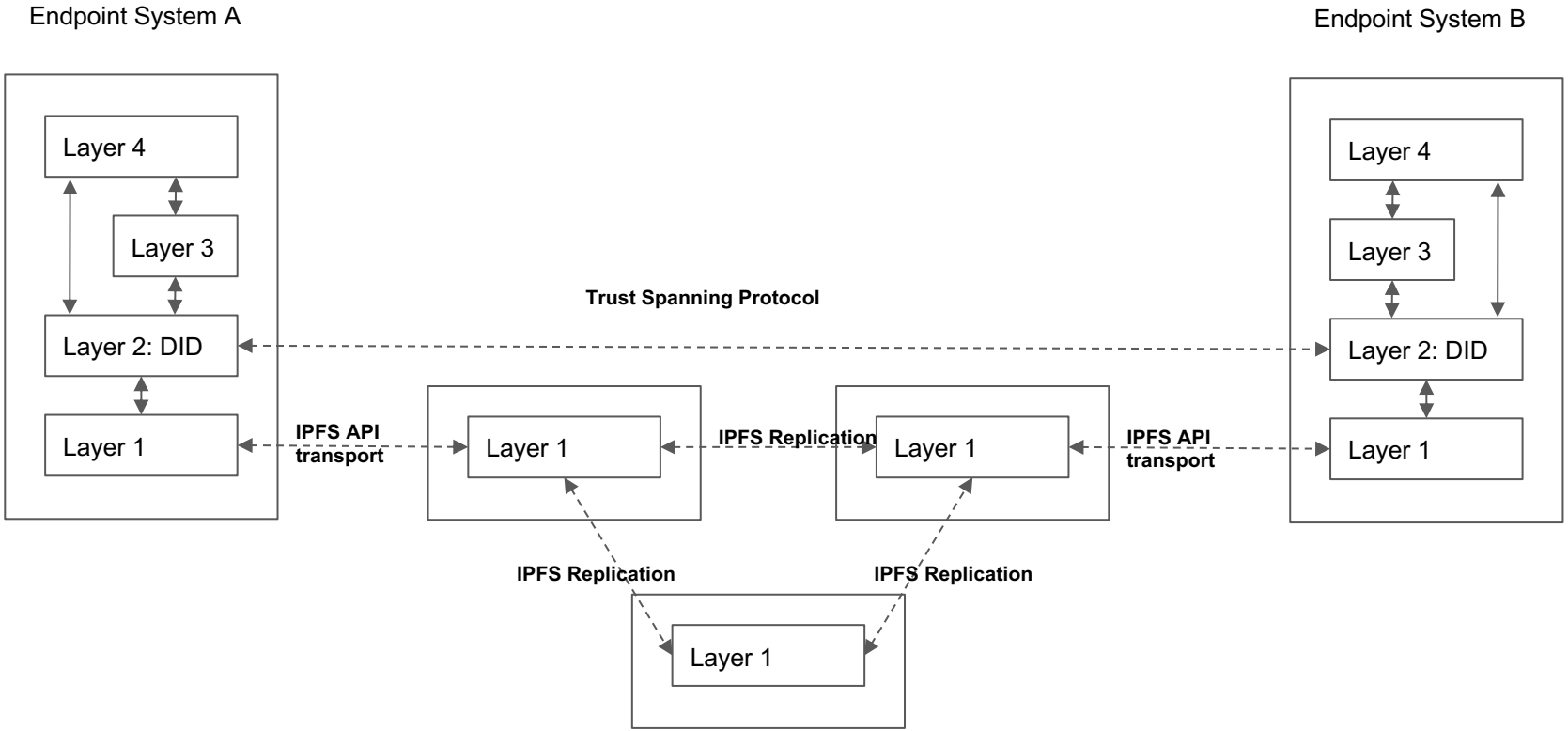


A Generalized Reference Architecture



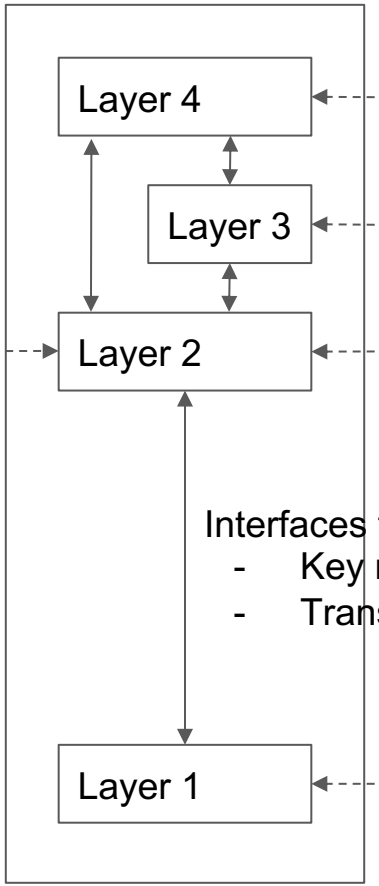
Intermediary Systems example:
DIDComm2

New Slide added June 23, 2022



Intermediary Systems example - DWN (aka Web5, TBDex)

An Endpoint System's Protocol Stack



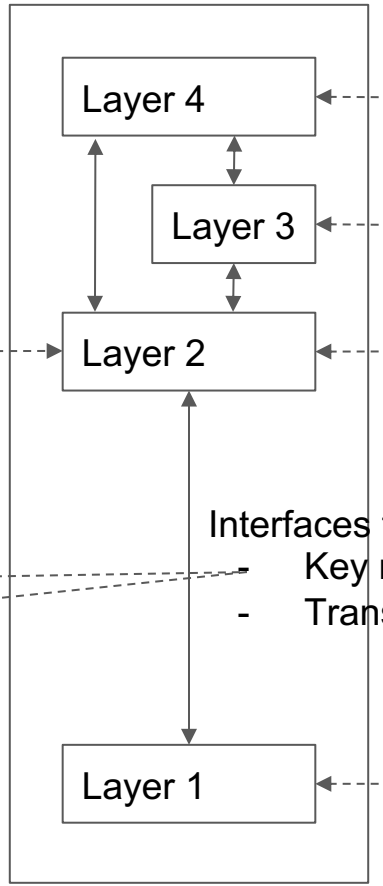
Zero, One or More Access Protocol(s) to Supporting System(s)

A Common Trust Spanning End System to End System Protocol

Interfaces to Layer 1
- Key material and storage***
- Transport over Internet

Transport over Internet

An Endpoint System's Protocol Stack



Zero, One or More Access Protocol(s) to Supporting System(s)

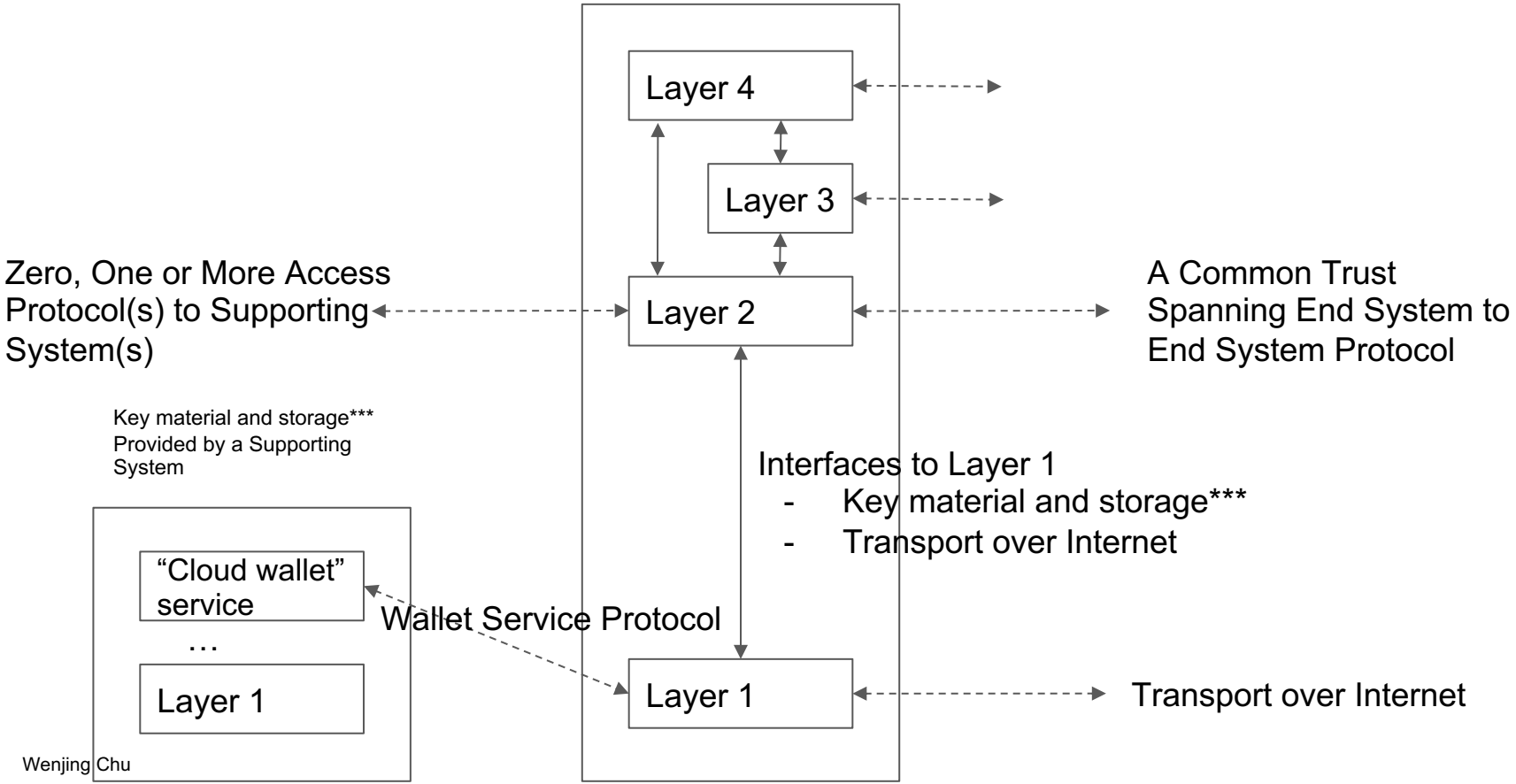
A Common Trust Spanning End System to End System Protocol

Interfaces to Layer 1
- Key material and storage***
- Transport over Internet

Interfaces can be realized as supporting services over the Internet (i.e. 'cloud wallet'):
- Key material and storage***

Transport over Internet

An Endpoint System's Protocol Stack



With this Reference Architecture, the details of each layer, interface, protocol can be specified one by one which, taken as a whole, completes the technical specifications.

- Current work in progress at the ToIP Technology Architecture Task Force.
- First draft available for review at:
  - Github: https://github.com/trustoverip/TechArch
- To give feedback or join the community:
  - ToIP Slack: #tswg-tech-arch-tf
  - Wiki: https://wiki.trustoverip.org/display/HOME/TSWG+Technology+Architecture+Task+Force

Questions?
Discussions?