



Proposal for Spanning Layer

Daniel Hardman — contributed to TSPTF — 15 Feb 2023 — <http://bit.ly/3xdt22n>

Assertion A.1

Trust varies with context.

I assume everyone agrees with this — what trust we need and why we need it are both functions of context — so it's not argued here.

Assertion A.2

Authenticity should decrease dependence on context.

I also assume everyone agrees with this.

Assertion A.3

Overly weak authenticity \rightarrow TSP isn't minimally sufficient.

This is controversial. It is only partially argued here.

Assertion A.4

Some authentic context shouldn't be layered.

This is controversial, too.

Assertion A.5

Predicting adoption is harder than
identifying who has the weakest mousetrap.

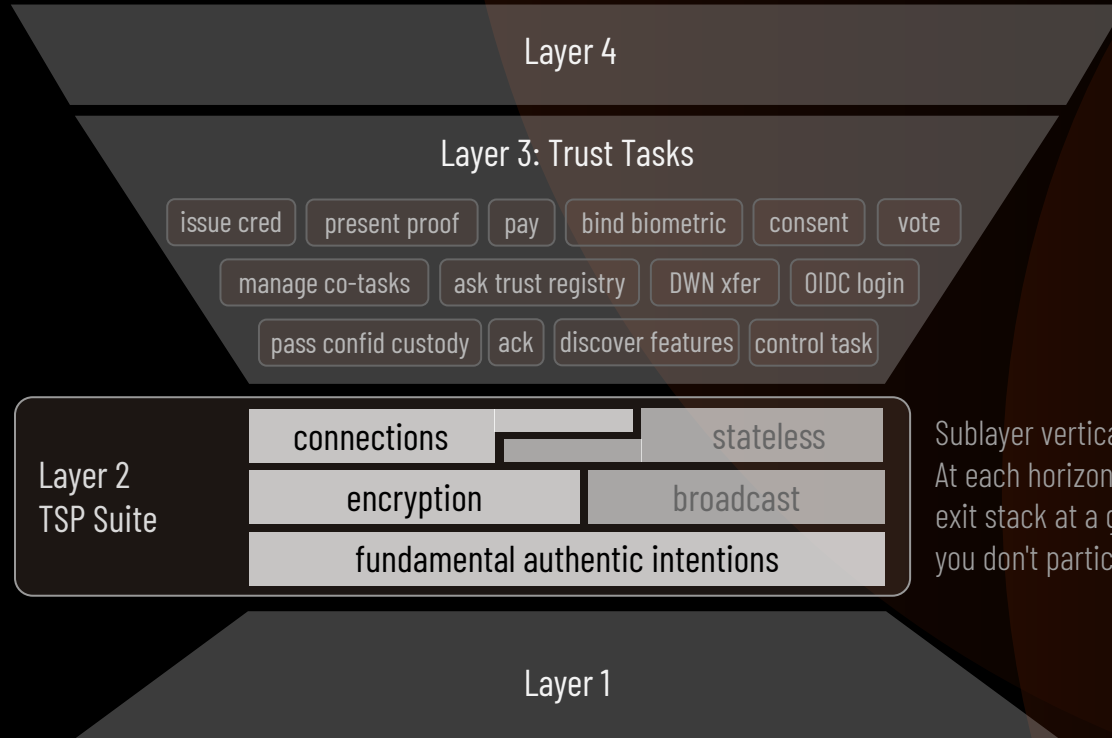
This is controversial, too.

Authenticity: foundation of trust, but not whole building - PAC theorem+

- What about confidentiality?
- What about privacy?
- What about transparency and recourse?
- What about power dynamics?

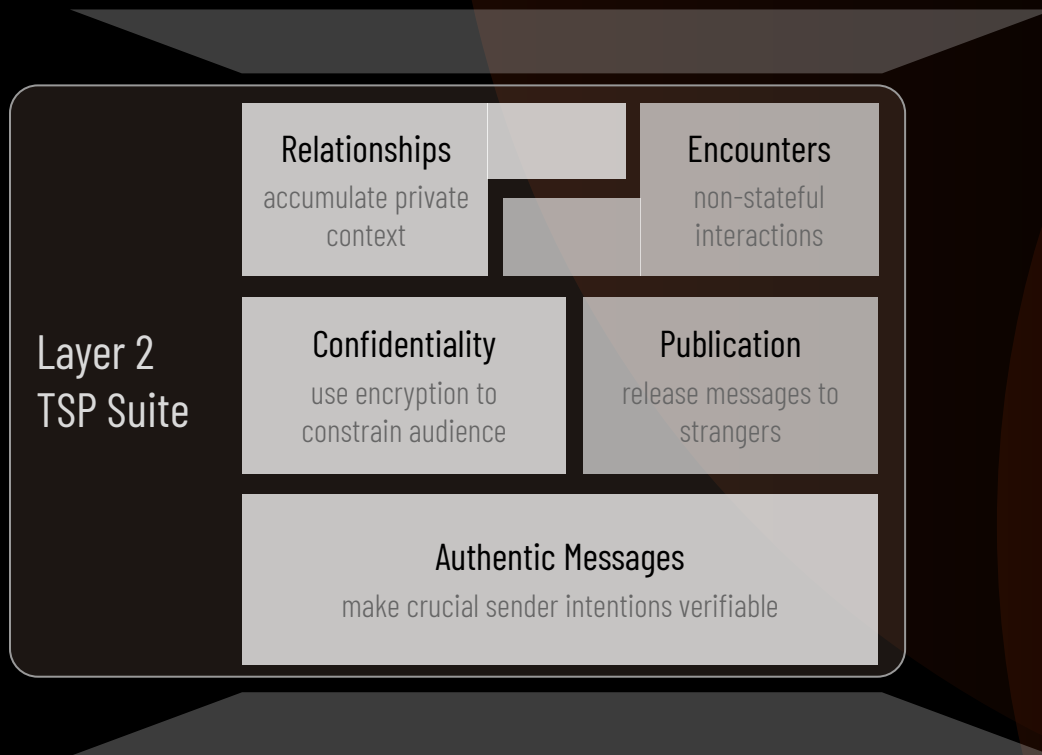
We don't have to put all of these into the spanning layer (and we shouldn't). But we should avoid describing as "trusted" or "trustable" mechanisms that offer only narrow authenticity. A routing behavior that just collects signatures from each intermediary isn't even close to "trustable", and we should discourage such reductionist thinking.

Proposal as a picture



Sublayer verticalness → strict dependency.
At each horizontal split, pick one option. Can exit stack at a given sublayer but may mean you don't participate in layers 3 and 4.

TSP suite detail



Proposal in words

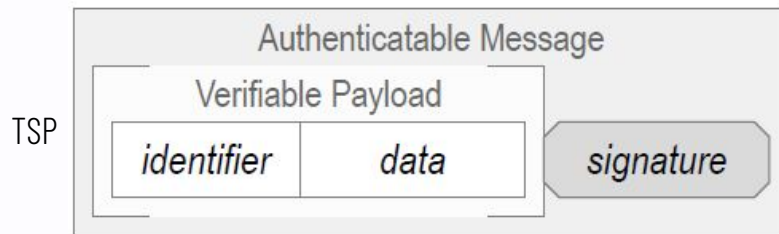
Referencing Sam's proposal as a base:

1. Add detail to Sam's diagrams by creating a "**TSP Suite**" that consists of 3 sublayers inside layer 2 — somewhat like the "IP Protocol Suite" includes IP + TCP | UDP.
2. Bottom layer of this suite should resemble Sam's vision — but **re-frame "authenticity" as broader than sources**, and add headers to make sure some vital semantics are defined.
3. **Encryption (vital for confidentiality)** is crucial for most trustful interactions and needs explicit attention as an **optional TSP sublayer** immediately above authenticity. Re-imagine DIDComm's enveloping using CESR coolness.
4. **Connection management (statefulness across interactions, vital for privacy and reputation)** — is also crucial in many cases, and deserves an **optional TSP sublayer** above confidentiality.
5. Embrace **DIDComm-style composability as a first-class feature** for trust tasks at Layer 3 (has some Layer 2 implications).

Re A.3 – "too weak authn → not min sufficient"

IP as a model for TSP

"IP provides just one thing: addressing/identifier — so that's what we should provide, too."



But does the dominant internet spanning layer actually provide *only* a source identifier?

IP

Version	Hdr Len	Type Svc	Total Length	
Identification			Flags	Frag Offset
Time to Live		Protocol		Hdr Checksum
Source Address				
Destination Address				
Options				Padding

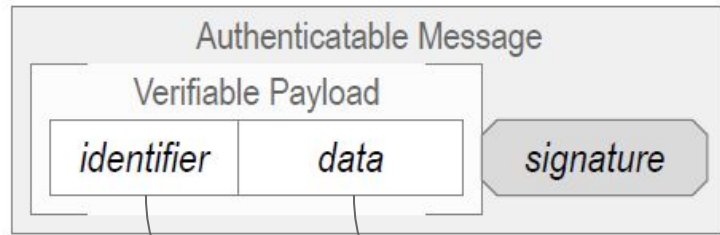
Context is a big deal

"Love makes the world go 'round."

— Lewis Carroll, 1865

Repeated in Gilbert & Sullivan's *Iolanthe* (1887). Sung with tenderness by Deon Jackson, the Everly Brothers, Jane Morgan, Anna Alberghetti, Ashlee Simpson...

Roughly authentic quote from *Alice in Wonderland* (see [ch VI and IX](#)). But uttered by the ugly and psychotic Duchess, to Alice, in sarcasm, after beating her miserable baby for sneezing and sharing joke poetry about how compassionate that was.



payload = (*A.id*, "I love you.")

msg = payload + *sign*(payload, *A.id.key*)

Should Bob trust that Alice asserts she loves him?

at I am
logic errors in
eral. One is
"message" and his
hus, I am
ges when I
message to his
diagram. The
tent in how I'm
. Properly, this
g the asserted
I am implying
roader.

e. It means that
solid.

his changes
h is that
ough to build
ving

erip/trust-span
22

Alice said it *to Carl*
instead?

Version	Hdr Len	Type Svc	Total Length	
Identification			Flags	Frag Offset
Time to Live		Protocol		Hdr Checksum
Source Address				
Destination Address				
Options			Padding	


Without dest address, Alice's intentions are not fully verifiable, and a malicious intermediary could mis-deliver the message. *It's often crucial to have an authentic dest.*

Note: this trust issue is about the *dest* (intended audience), not the *route* (how it gets there). These should not be conflated.

Trust-destroying dest abuse from current events...

Slashdot Stories **Firehose >** All Popular Polls **Software** Apparel Newslette

US Army Officer Reply-All Email Chain Causes Pandemonium (militar

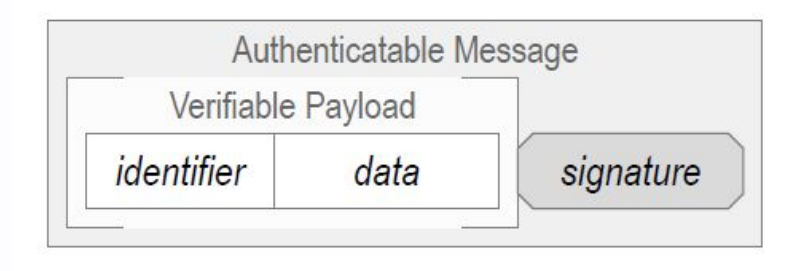
 Posted by **BeauHD** on Friday February 10, 2023 @08:50PM from the hilarity-ensues dept.

An anonymous officer writes in an opinion piece via Military.com:

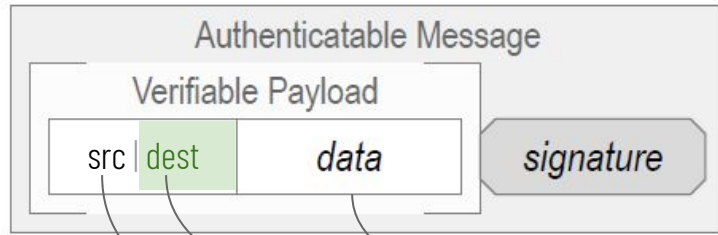
It was the "reply-all" heard around the world. Around 06:30 Eastern time Feb. 2, approximately 13,000 Army inboxes pinged with an email from an unfamiliar sender. It was from a U.S. Army captain, asking to be removed from a distribution list. It initially seemed as though some unfortunate soul had [inadvertently hit "reply-all" and made an embarrassing mistake](#). What followed can really be described only as professional anarchy, as thousands of inboxes became buried in an avalanche of email replies. Someone appears to have unwittingly edited an email distribution list, entitled "FA57 Voluntary Transfer Incentive Program," routing replies back to the entire list.

In this case, the mis-delivery was an accident (probably). But playing games with destination is a great way to do deliberate mischief, too. Either way, trust is undermined.

Without no notion of a recipient, is it accurate to frame this as a "message"?



With no notion of any other party or action, can it define a protocol?



`payload = (A.id, B.id, "I love you.")`

`msg = payload + sign(payload, A.id.key)`

Should Bob trust that Alice asserts she loves him?

What if Alice said it *9 years ago, as a child?*

Version	Hdr Len	Type Svc	Total Length	
Identification			Flags	Frag Offset
Time to Live		Protocol		Hdr Checksum
Source Address				
Destination Address				
Options				Padding

Without reference to the sender's timing, Alice's intentions at time of receipt are not fully verifiable, and a malicious intermediary could manipulate assumptions. *It's often crucial to have authentic timing.*

Trust-destroying time abuse from current* events...

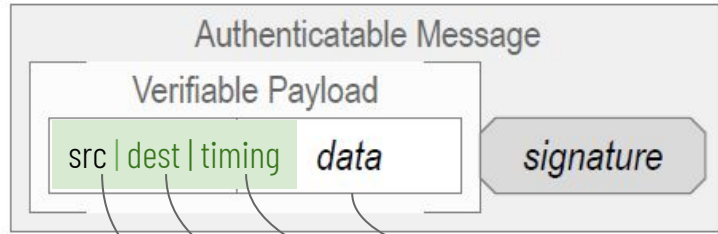


image credit: COP Paris (Flickr) - public domain.

"I actually did vote for the \$87 billion, before I voted against it."

—US Secretary of State
John Kerry

Many more recent and more flagrant examples exist; I picked something a bit old to avoid controversy. Was I fair? This is an authentic picture of John Kerry at COP2015, and he was Secretary of State then. His time- and trust-abusing statement is also authentic, and truly illustrates the problem. But Kerry articulated his famous flip-flop during the 2004 US presidential campaign, so combining the 3 elements is also misleading WRT time...



`payload = (A.id, B.id, timing, "I love you.")`

`msg = payload + sign(payload, A.id.key)`

Should Bob trust that Alice asserts she loves him?

What if Alice said it *while rehearsing* a romantic play?

Version	Hdr Len	Type Svc	Total Length	
Identification			Flags	Frag Offset
Time to Live	Protocol		Hdr Checksum	
Source Address				
Destination Address				
Options			Padding	

Without reference to a goal, Alice's stance toward her authentic data may be misunderstood. Her intentions are not fully verifiable. *It's often crucial to have authentic purpose.*

Trust-destroying purpose abuse from current events...

TECH DRIVERS

ChatGPT's 'jailbreak' tries to make the A.I. break its own rules, or die

PUBLISHED MON, FEB 6 2023•11:09 AM EST | UPDATED WED, FEB 8 2023•3:03 PM UNDEFINED EST

Rohan Goswami
@ROGOSWAMI

SHARE    

KEY POINTS

- Reddit users have engineered a prompt for artificial intelligence software ChatGPT that tries to force it to violate its own programming on content restrictions.
- The latest version of the workarounds, which are called Do Anything Now, or DAN, threatens the AI with death if it doesn't fulfill the user's wishes.

image credit: cnbc.com (fair use)

The jailbreak technique is to tell ChatGPT to pretend a different purpose (predict what an AI without safeguards against hate and sleaze would say). This changes meaning and accountability.

RELATE



Google employees slam CEO Sundar Pichai for 'rushed, botched' announcement of GPT competitor Bard



Why passkeys from

Stepping back to generalize

- The proper scope of concerns for authenticity isn't as narrow as who sent an opaque payload; **authenticity of metadata** is also important.
- Although Alice and Bob's love life is light-hearted, **authenticity issues like this have major consequences for trust in real life**. Abusing the destination, the timing, or other higher-level context of interactions is the bread and butter of cyber criminals.
- However, just because authenticity can be abused **doesn't mean TSP has to fix it**. (See argument for A.4 "some context can't be layered", coming next.)
- So, what **guidelines** could decide whether something belongs in the spanning layer?

Guidelines Straw Man

It belongs in the base sublayer of the TSP suite IFF:

- It's about authenticity WRT the intentions of a message sender (high cohesion).
- It is simple and general-purpose (Beck), and it is likely to be widely used.
- Leaving it undefined invites abuse.
- Including it has little or no effect on the set of spannable supports.

7+1 fundamental authentic intentions

Who am I, the speaker?	Changes what part of my identity context is known, and what reputation is at stake: "I'm V, your gamer buddy" vs. "I'm Volodymyr Zelenskyy, the president of Ukraine."
Who do I think you are?	Changes accountability of senders, listeners, and eavesdroppers: "This is for Alice, who has previously proved her security clearance to me."
Did I say anything before this that matters?	Clarifies what state the message is designed to modify: "This builds on the assumptions in our wargaming scenario. See my previous messages for caveats."
What's my goal?	Guides behavior patterns and outcome. "I am trying to be a whistleblower, not testify under oath."
Is my goal bounded in time?	Sets expectations about timeliness. "This offer to merge our companies is only good for the next 20 minutes. Act now before it's too late."
Is there external state that matters?	Anchors accountability in something larger than the conversation. "I proposed this stock purchase AFTER I read version 2 of your prospectus, not before."
What else do you need to know?	Tells how to interpret other pieces of data in the message. "Since this is an official application, you will notice my full name, mailing address, and 3 required attachments."
What's wrong?	Lets parties describe, recognize, and react to errors in predictable ways.

Base sublayer of TSP Suite = **Fundamental Authentic Intentions (FAI)**?

This has the cute property that messages passing through this layer can be evaluated objectively for **FAlthfulness** to the sender's intentions. :-)

Faithful is a synonym for *trustworthy*.

What's in a name?

FAlthful fields?

sr (source identifier)	Required. AID. Gives sender's intent WRT the reputational context for the message.
sig (source identifier)	Required. Signature over header and payload.
a (audience identifiers)	Optional (missing → audience is "any"). Array of AID. Identifies intended plaintext audience, NOT delivery targets for routing or encrypted envelopes.
th (thread)	Required. Non-negative 32-bit int. All participants use sr + th as the thread's lookup key; the sender of a thread's first message must pick a th value that makes this combination unique enough for all practical purposes. Groups messages by topic into logically related streams with different goals, states, and trust profiles.
mo (message ordinal)	Required. Monotonically increasing, non-negative 32-bit int. Counts how many messages sender has previously contributed to this thread; makes gap detectable.
pth (parent thread)	Optional, and only allowed when mo == 0 (starting a new thread). If omitted then, thread is standalone. Otherwise, connects this thread to previous verifiable data.
ttl (time to live)	Optional. Non-negative 32-bit int. Epoch time in seconds when sender's goal for this message will lapse. Begin ignoring at this time to avoid useless processing ("offer good until time X").
ex (exists)	Optional. Hash with special CESR prefix to clarify PoE type (e.g., blockchain root; IPFS, github commit, build artifact). Proves message was created after the referenced data already existed.
s (message schema)	Required. SAID. Defines structure of rest of payload, including extra headers and attachments.

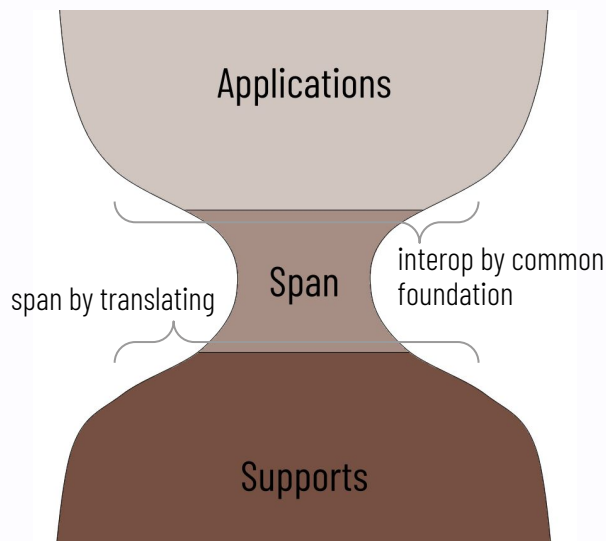
CESR support for extensible schema, binary and text transformations, compression, encodings, and arbitrary attachments is assumed.
We also need **DIDComm's goal codes and problem codes**.

Re A.4 — "some context shouldn't be layered"

Spanning layer: translate below, interop above

Why are common standards needed at one layer but not at another? Certain protocols are designed with the specific purpose of bridging differences at the lower layers, so that common agreements are not required there. Instead, the layer provides the definitions that permit translation to occur between a range of services or technologies used below... [A]t and above such a layer common standards contribute to interoperation, while below the layer translation is used. Such a layer is called a "spanning layer" in this paper.

—Clark, 1997, p. 133



[A]t the narrow point in the hourglass, is ... the key to the approach that [it] takes to interoperation. The bearer service provides a set of capabilities sufficient to support the range of applications illustrated above it. It implements these capabilities by building on the more basic capabilities...below. [It] would thus span the broad range of network technologies illustrated below it, hide the detailed differences among these various technologies, and present a uniform service interface to the applications above. The bearer service is thus an example of a spanning layer, with specific features and capabilities.

—Ibid

The Hourglass Theorem

weakness → span more "pre" beneath → more ways to implement → **easier technical adoption**

strength → fatter interface with more "post" above → harder to support → **richer applications**

"If a specification S1 is weaker than another specification S2, then:

1) $\text{post}(S1) \subseteq \text{post}(S2)$, and

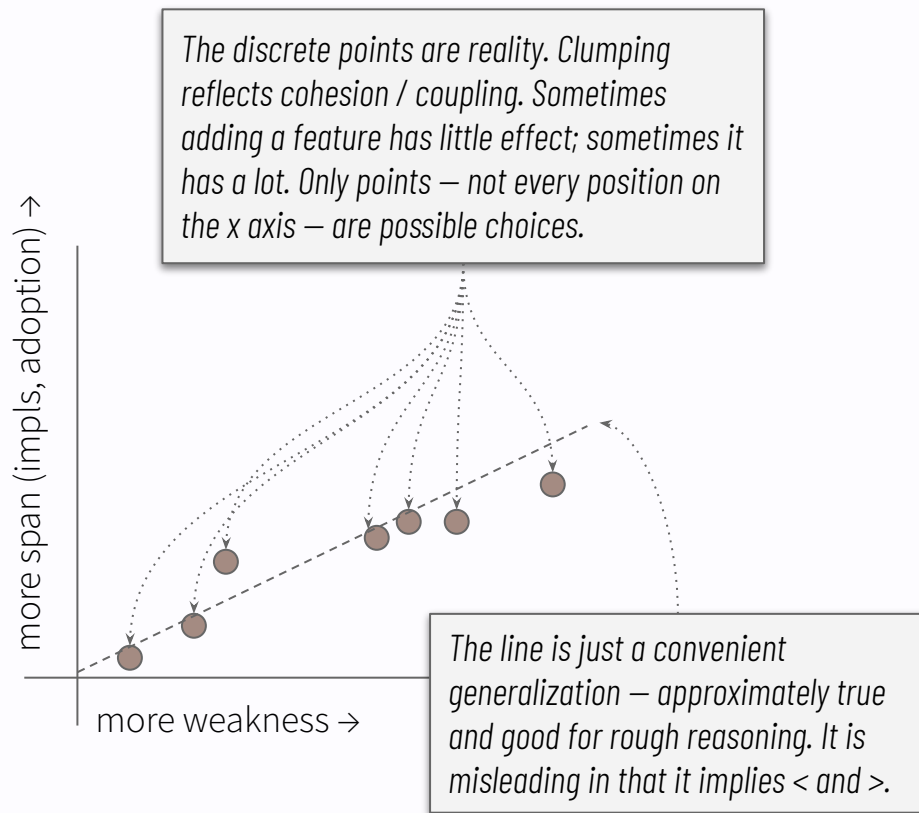
2) $\text{pre}(S1) \supseteq \text{pre}(S2)$." —Beck, 2019, p. 52

The weaker spec's upper hourglass is a subset of the stronger. *What weak supports \leq strong.*

The weaker spec's lower hourglass is a superset of the stronger. *What weak spans \geq strong.*

*NOTE: Beck does not use the \subset and \supset (proper sub/superset) operators, so $>$ and $<$ would be incorrect. This nuance matters, because it means *weaker doesn't automatically increase span*.*

Weakness and spanning: fuzzy and clumpy, not precise and continuous



If one goal is maximizing possible supports, then the Hourglass Theorem tells us that the slope of the subspace of feasible solutions when considering this goal as a function of the logical weakness of the spanning layer is non-negative. **We have no metrics for logical strength or for the size of the space of possible solutions**, only for the notions of one service description being weaker than another and one set of service descriptions being included in another. These definitions allow system architects to **reason about the sign of the slope**, but not its steepness nor what value is necessary in order to achieve a particular design goal.

—Beck, 2019, p. 53-54

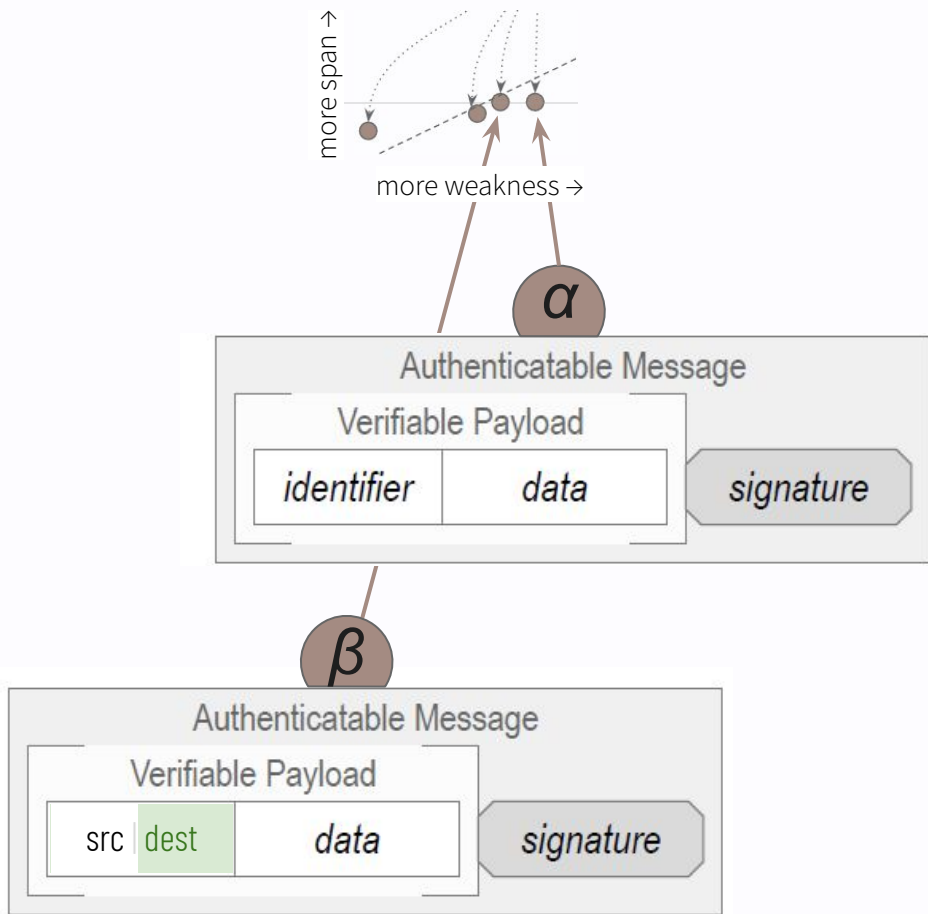
Example of clumpiness

It's usually good to be weaker. But here, α is weaker than β , yet has the same Y value on our graph. That's because the set of spannable supports is identical.

- β 's extra `dest` field has same data type as `src`
- Same dependencies in code, at runtime, and in external systems
- Same party chooses both field values
- No new temporal coupling: both chosen at same point in time

Eliminating `dest` to get weakness trades away strength for no benefit.

Weakness is only valuable if it increases spanned supports.



(Un)constrained vs. Undefined

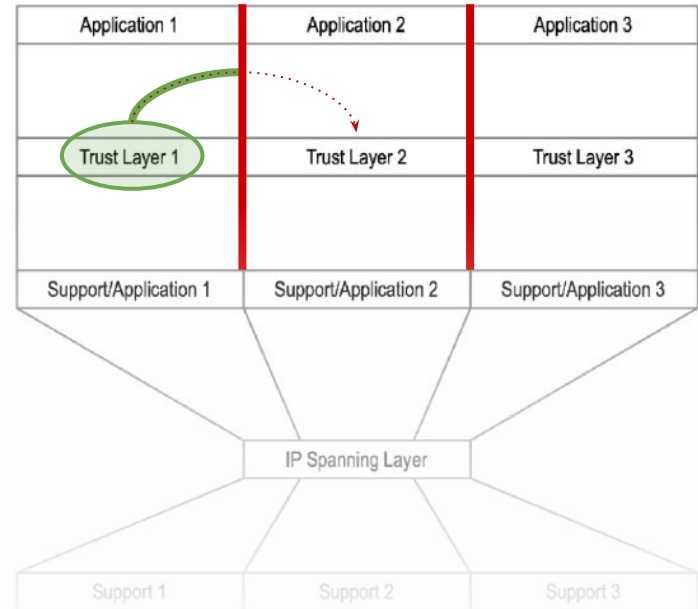
Q: What if Alice just signs for anybody to read?

A: **She should say that** (dest = any).

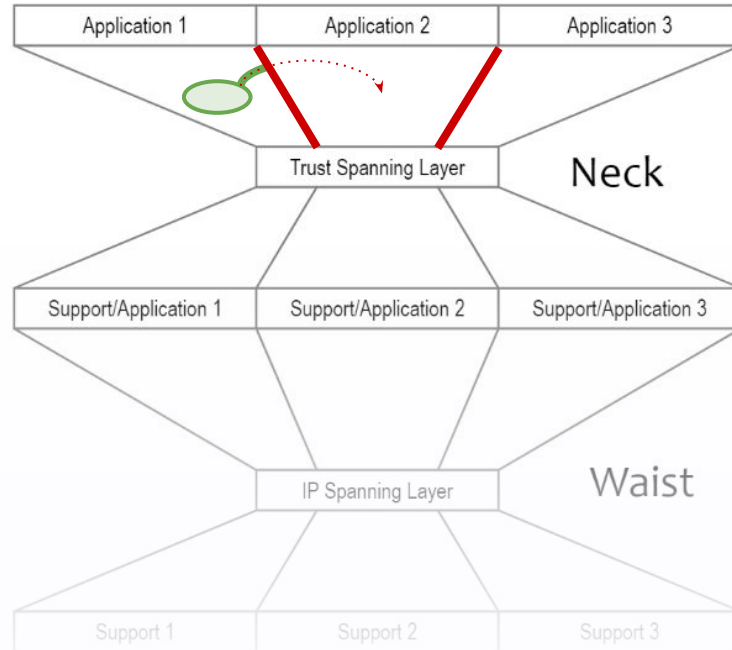
- Any dest ("Msg is for anyone") != undefined dest ("What does "dest" mean? Who cares?").
- Undefined intentions → trust manipulation.
- Pushing X from layer 1 to 2 means that X is undefined in layer 1.

Sam rightly decried "Platform-Locked Trust"

Security is siloed because applications don't have a common layer of shared trust.



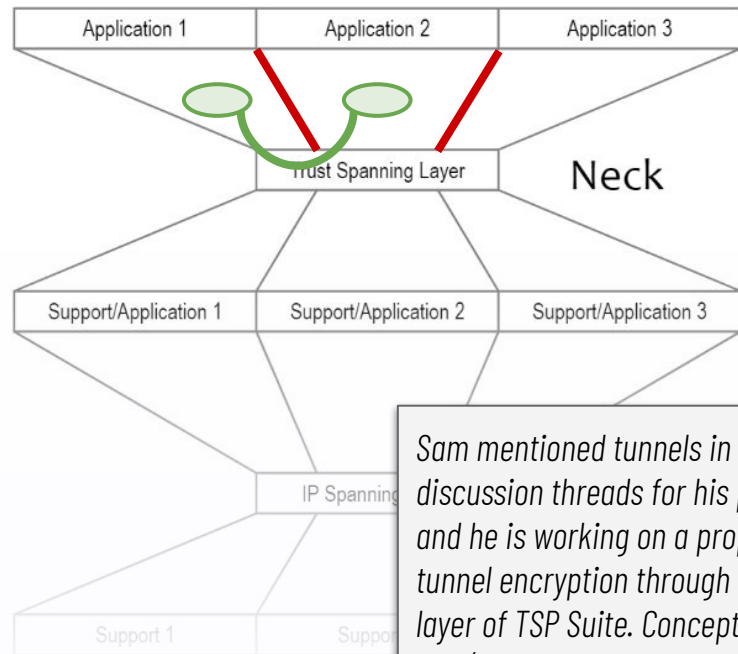
A basis for trust that is defined above TSP
can also create "Application-Locked
Trust"...



Neck only prevents if TSP
is minimally sufficient (no
green above neck)

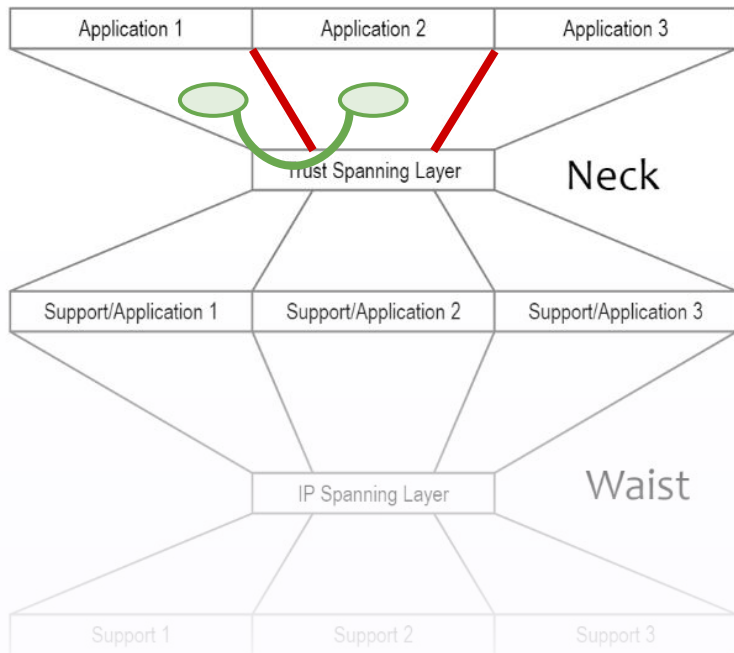
Tunnels to the rescue!

If TSP carries opaque trust basis in payload, we can get around the wall.



Sam mentioned tunnels in the discussion threads for his proposal, and he is working on a proposal to tunnel encryption through the base layer of TSP Suite. Conceptually, I like this (cf DIDComm enc envelope).

Tunnels have limits



1. Require **provable** correspondence in other stack (e.g., encrypt on one side, decrypt on the other) — else can create dead ends or unfounded assumptions.



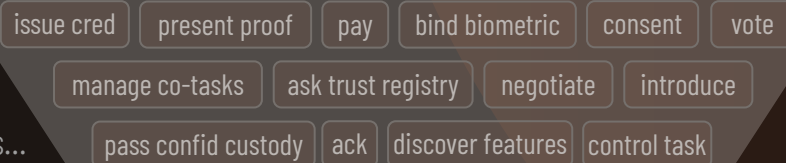
2. Lower-level constructs can't reason about what's happening.

Applications aren't the layer above TSP

Layer 4: Applications and Ecosystems

Trust tasks differ from applications in important ways: they are *multi-party*, and they should be *composable*. Trust implications...

Layer 3: Trust Tasks

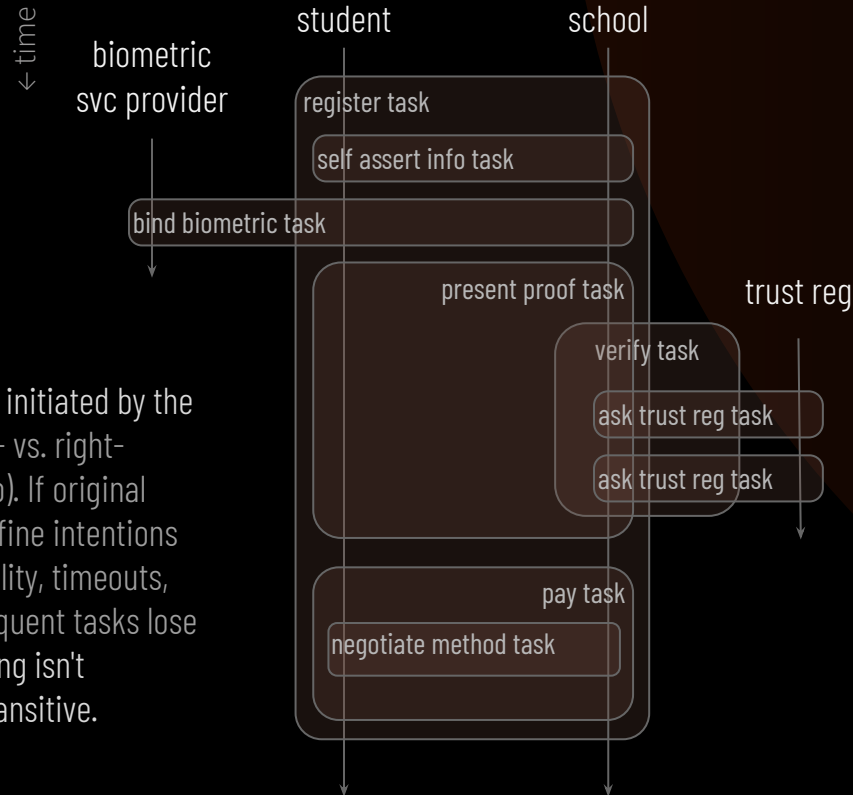


What's above TSP isn't best imagined as high-level network protocols, either. It demands *human-quality comms on human timescales*. This also has trust implications...

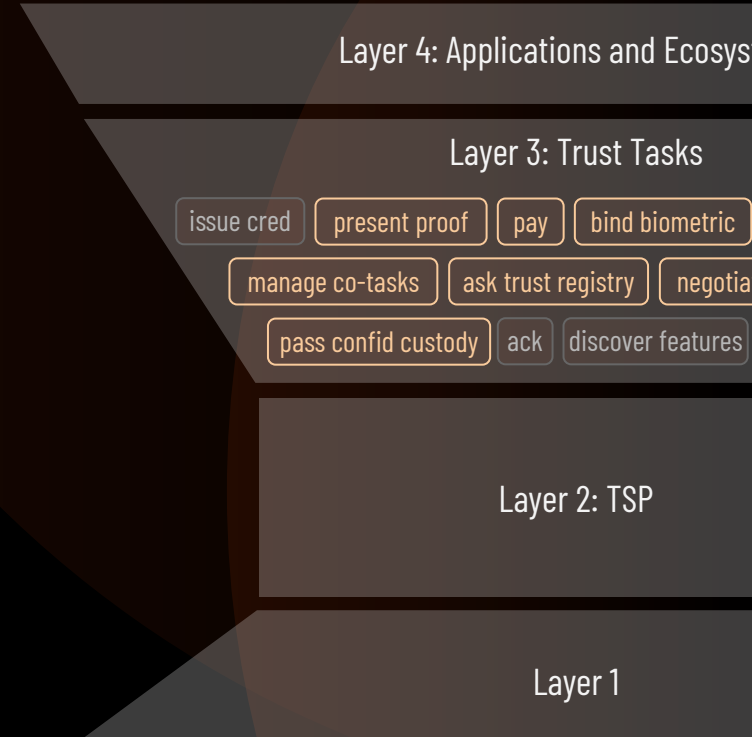
Layer 2: TSP

Layer 1

Composability example: register for a MOOC

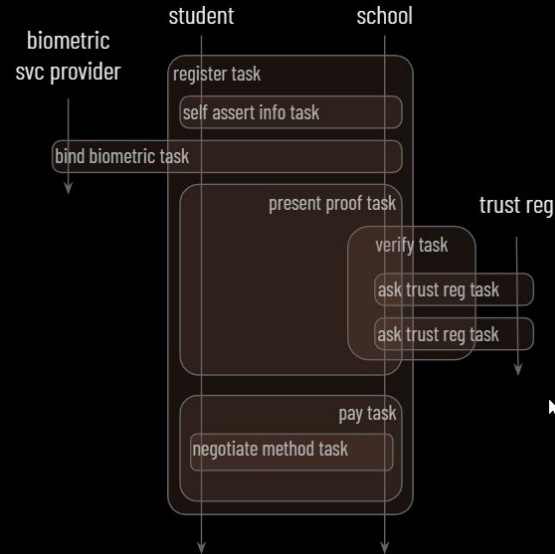


Not all tasks are initiated by the same actor (left- vs. right-aligned tells who). If original actor doesn't define intentions WRT confidentiality, timeouts, etc., then subsequent tasks lose context. Tunneling isn't automatically transitive.

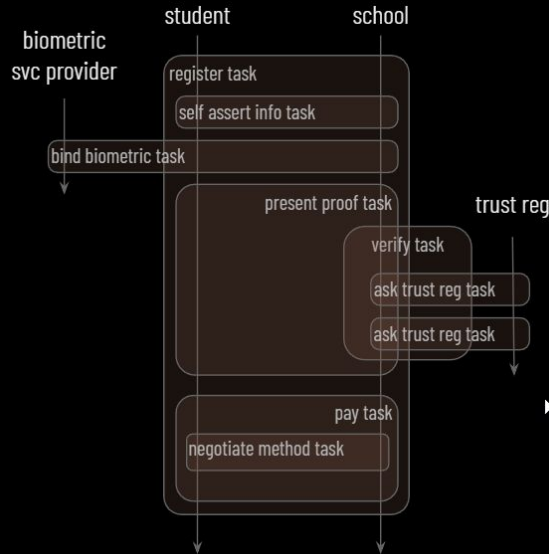


Reasoning about trust requires context hooks guaranteed by foundation

How does the TSP help me reason about and manage the trust properties (confidentiality, privacy, transparency, consent, governance...) of this workflow if the only thing guaranteed to be common to all of them is the TSP, it cares only about authentic source, different entities manage different parts, and tunnels aren't transitive?



How does the TSP help me enforce timeouts, propagate errors and cancellations, and pass inputs and outputs across trust task boundaries, if the only thing guaranteed to be common to all of them is the TSP, it cares only about authentic source, different entities manage different parts, and tunnels aren't transitive?



Gluing things together
requires context hooks
guaranteed by foundation

Understanding DIDComm

Defined in DIDComm v2 spec, which may cause confusion. Always separate layer.

Application-Level Protocols

issue cred present proof pay bind biometric
co-protocols chat negotiate introduce
"routing" ack err discover features control task

built from DIDComm primitives and casually called "DIDComm" by some.

DIDComm

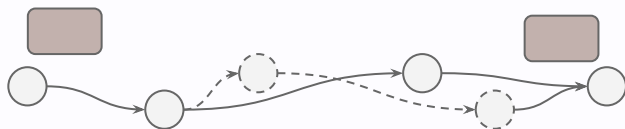
encryption

signing

plaintext struct + hdrs, err and goal codes

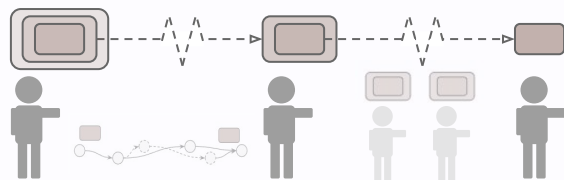
DID Methods, VDRs, crypto

3 kinds of "routing" that are often confused



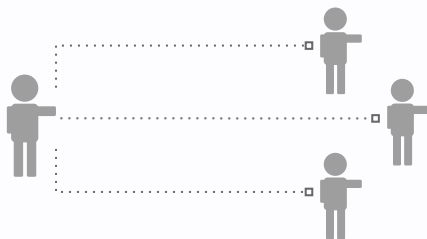
Bit Movement

Concerns network nodes. Many routes may be valid; best for now is chosen dynamically. Bits don't change in transit.



Confidential Chain of Custody

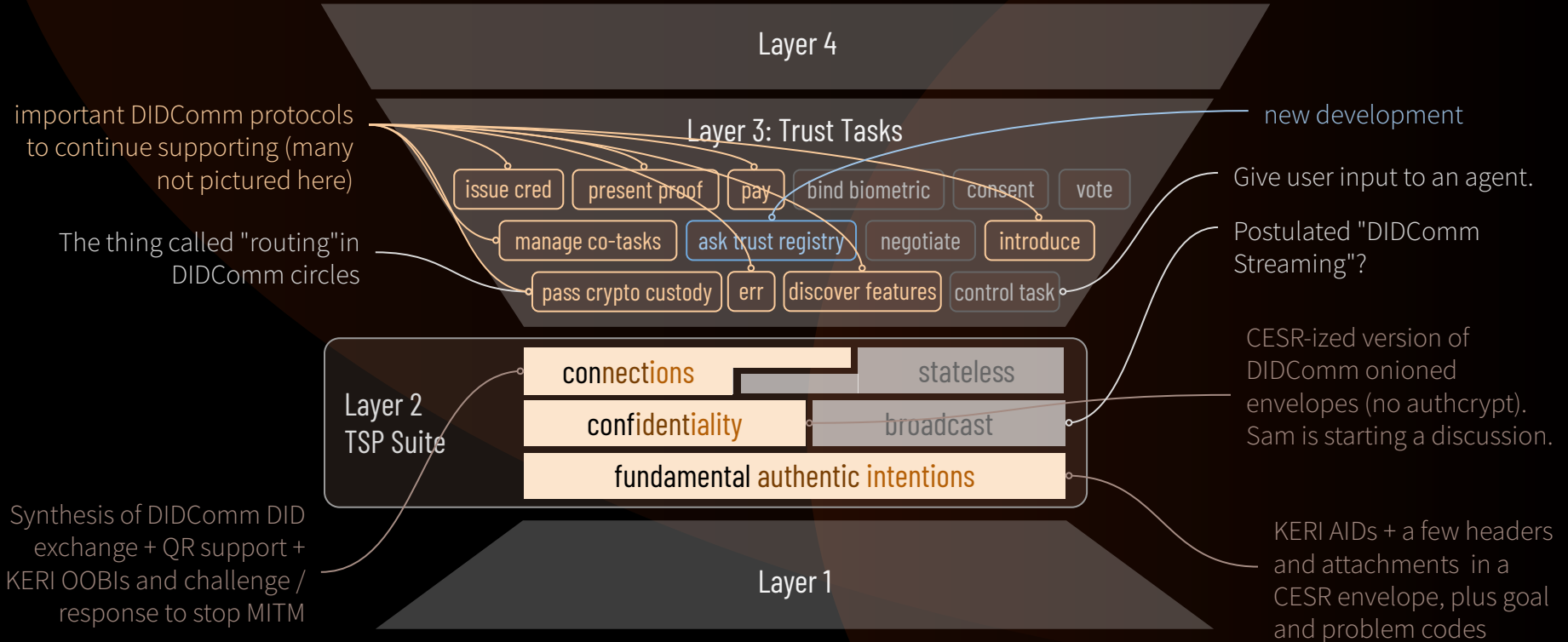
Who's accountable as confidential envelopes are opened? Parties are identifiers, not network nodes. Movement is slightly related.



N-wise Distribution

Sending a message a full audience > 1 , as opposed to anyone party? Movement and confidentiality are both incidental concerns.

Mapping DIDComm



Re A.5 – "adoption is complicated"

Perhaps we should
consider non-technical
adoption issues?

Beck identifies openness and symmetry (power dynamics) as interesting and orthogonal to the weak/strong tradeoff:

A final aspect of interoperation is the issue of whether, when two (or more) entities interact, they do so as equals or peers, or instead in some asymmetric way, for example as client and server, or as provider and consumer. Protocols are sometimes designed in such a way that not all the participants in the protocol can play all the roles. For example, some of the early remote file access protocols were designed so that the functions that implemented the file server and the file user were separately specified. This made it easy to produce implementations that omitted the server code, so that the resulting software could only play the role of the user. It was then possible to sell the server code for additional money. **In general, symmetry, like openness, tends to reflect business issues.**

—Beck, 1997, p. 138

Considerations

SSI community isn't starting from scratch. Do we want to facilitate their adoption at all? Where are we willing and not willing to compromise?

The larger internet isn't starting from scratch, either. Where are we willing and not willing to compromise?

Possible Compromises

1. View AIDs as a DID upgrade, and define DID rotation as a migration path.
2. Allow identifiers in source and target fields to be DIDs, but deprecate them with EOL date and define the security consequences formally. (DIDComm message trust contexts are something sort of like what we would need.)
3. Allow "trust tasks" that short-circuit all but the bottom of Layer 2 (still live in layer 3 but get encryption or manage connections a different way).
4. Formally define a "trust bridge" that lets trust from TSP land infuse a less trustworthy stack.
5. Allow the use of channel-based security (e.g., TLS) as a poor but pervasive substitute for confidentiality guarantees that travel with the message.

TSP suite detail

