Trust State Protocol (TSP)

A Framework for Trust Mechanics in Digital Systems

Karim Lahnin

www.karimlahnin.com

June 2026

Abstract

This paper defines a framework for how trust is measured and updated in digital systems. Trust State Protocol (TSP) sets out the rules by which a trust value is created, changed over time, and kept within a defined range. The framework is not tied to any platform, identity system, or legal requirement. It is intended as a reference for engineers and system designers who need a shared model for trust that works across different types of software.

1. Introduction and Core Terminology

The Trust State Protocol (TSP) defines how a trust value is created, updated, and read within a software system. It covers only the rules that govern how trust changes based on confirmed interaction results within a set boundary. It does not cover user interfaces, platform rules, identity systems, or legal requirements. TSP is a low level layer that handles trust mechanics only.

1.1 Key Terms

Term	Definition
Trust	A number that measures how reliable an entity has been, based on confirmed past interactions within a given context. It does not predict future behavior.
Trust State	The recorded trust value for one entity in one context at a given point in time. It changes over time and only applies within its own context.
Context	A defined area in which trust is measured. For example, a payment service or a delivery platform. Each context has its own rules and outcome criteria.
Entity	Any participant that can hold a trust value. This may be a person, a system, or an automated agent. It is not the same as a verified identity.

Each context is kept separate from others. Trust built in one area does not carry over to another. There is no single global trust value. This separation prevents trust from one area from incorrectly influencing another.

2. Design Objectives

TSP was created to fix known problems in how trust is handled in existing systems. The objectives below define what the protocol is meant to do and what it leaves out. The protocol covers trust mechanics only. It does not set policy, enforce rules, or decide how results are used by other systems.

Trust as a Protocol Variable

TSP treats trust as a core part of the protocol. Different systems can share a common model for how trust changes over time, rather than each building their own private logic.

Context Separation

Trust in one area does not apply to another. Each trust value must be tied to a specific context to prevent it from leaking into unrelated areas.

Trust Can Go Up or Down

Trust values may rise or fall based on confirmed results. They also decrease over time when no new information is available. Trust is not fixed or permanent.

Works Across System Types

TSP defines abstract rules, not implementation details. It can be used by centralized, shared, or distributed systems without requiring a common authority.

Privacy and Auditability

The protocol does not require real names or ongoing monitoring. Trust changes come from resolved outcomes. The process can be checked without exposing personal data or past behavior.

3. The Trust State Model

The Trust State Model defines how trust is represented in TSP. It describes a value that changes over time based on confirmed interaction results within a defined context. The model does not encode reputation, rankings, or social opinion.

3.1 Representation

A trust state is a single bounded number assigned to one entity in one context at one point in time. Its value reflects current confidence based on past confirmed interactions of the same type. The same number in two different contexts does not mean the same thing.

3.2 Starting Value

TSP does not set a fixed starting value. Each context must define its own starting point in a clear and repeatable way. The starting value represents a baseline level of uncertainty. It does not assume the entity is reliable or unreliable.

3.3 What a Trust State Does Not Mean

A trust state does not predict future behavior, express intent, or make a moral judgment. It records only the level of confidence supported by past confirmed results in a narrow context. It is a signal, not a verdict.

3.4 How Trust Changes

Trust changes through fixed rules based on confirmed results. For the same starting value, result, and time, every system using TSP must produce the same new trust value. Over time, when no new confirmed results arrive, trust moves toward the baseline as older information becomes less useful.

3.5 Context and Identity

Trust exists only within the context where it was created. It does not carry over to other contexts by default. There is no combined global trust value. Trust may be attached to a name, a code, or a system identifier. It does not require a verified real world identity.

4. Event Model and State Transition Rules

The Event Model defines what counts as an event in TSP and explains how events change the trust value. An event is a single confirmed result within one defined context. Only final results that meet the resolution criteria of that context may affect trust. Events represent outcomes, not actions or intentions.

4.1 Event Classification

Each event is given an impact on the trust value. Positive events raise confidence. Negative events lower it. Neutral events leave it unchanged. The context must define this classification in a fixed and repeatable way. Events are recorded at the time of confirmation, so trust changes follow confirmed results and not the start of an interaction.

4.2 State Transition Formula

The trust value is updated using the following rule:

$$T(ti^{+}) = (1 - \alpha i) \cdot T(ti^{-}) + \alpha i \cdot Ii$$

Variable	Meaning
$T(ti^{+})$	The trust value immediately after the event is processed
$T(ti^{-})$	The trust value immediately before the event is processed
αi	A weight between 0 and 1. A higher value gives more importance to the new event. A lower value gives more weight to the existing trust history.
Ii	The impact value of the event. This is set by the context based on the outcome type (positive, negative, or neutral).

This rule keeps the trust value within its allowed range. Each event moves trust by a weighted step. The new value is always a blend of the existing trust and the incoming event impact. The result is always kept within the bounds T_min and T_max.

4.3 Consistency

Given the same starting trust value, the same sequence of events, the same impact values, the same weights, and the same time steps, every system using TSP will produce the same trust values. The process is fully repeatable across independent systems.

5. Decay Function and Verification Model

5.1 The Decay Function

The decay function defines how the trust value changes when no new confirmed events arrive. Decay is a required part of the protocol. It reflects growing uncertainty over time, not a negative judgment. When no recent confirmed information is available, older results become less useful for predicting future interactions.

The standard form is exponential decay toward a baseline value:

$$D(x, \Delta t) = T_0 + (x - T_0) \cdot e^{(-\lambda \Delta t)}$$

Variable	Meaning
$D(x, \Delta t)$	The trust value after time Δt has passed since the last confirmed event
x	The trust value before decay is applied
T_0	The baseline uncertainty value defined by the context. Trust decays toward this value over time.
λ	The decay rate. Must be greater than 0. A higher value causes trust to fall faster. A lower value causes it to fall more slowly.
Δt	The amount of time that has passed since the last confirmed event

Decay is applied before any event update. This order prevents manipulation through late reporting. The purpose of decay is to stop old trust from building up without new evidence.

5.2 The Verification Model

The verification model defines what it means for an event to be confirmed. It does not bind the protocol to any specific tool, authority, or technology. Confirmation does not mean absolute truth. It means that an outcome has been checked according to declared and repeatable criteria.

An event must be confirmed before it can affect the trust value. Results that are not confirmed are ignored, regardless of how significant they appear. Each confirmed event is linked to a verification context that describes how strong the confirmation is. This may include mutual agreement, cryptographic proof, third party review, or system level guarantees.

The protocol does not favor any single method. It only requires that the verification context be declared, fixed, and repeatable. Confirmation is limited in scope. Even strongly confirmed results are still subject to decay and bounded influence. Confirmation does not directly change the trust value. It allows an event to enter the update process. The direction and size of the change are determined by the transition rules and decay, not by the confirmation itself.

> Verification within TSP is designed to be auditable without exposure. Systems may show that verification criteria exist and are applied consistently, without revealing identities, past behavior, or private processes. This supports accountability while keeping data retention to a minimum.

6. Security, Threat Model and Privacy

This section defines the threat setting for TSP. The protocol assumes that participants may act in their own interest and that some may try to distort trust signals for personal gain. It does not assume that all participants are honest or that a central authority is present. The main risk is manipulation of trust signals, not attacks on the underlying system.

6.1 Known Threats and Mitigations

Threat	How TSP Addresses It
Trust Inflation	Bounded updates, weighted steps, and required decay prevent trust from growing without limit. Repeated low impact events cannot push trust to its maximum. Trust falls over time without continued meaningful interaction.
Trust Laundering	Strict context separation prevents trust from moving between areas. Trust values, events, confirmation contexts, and decay settings are all tied to a single context. The protocol has no way to move trust across contexts.
Sybil Amplification	Trust growth is tied to confirmed results, not to the number of accounts. Creating more accounts does not produce trust unless those accounts generate confirmed events. The cost of maintaining many trusted accounts rises naturally.
Replay and Late Events	Trust updates are anchored to the time of confirmation. Events affect trust only when confirmed, and decay is applied up to that moment. This prevents past events from being used to manipulate current trust values.

6.2 Privacy

TSP is designed to work with as little data as possible. Trust changes depend only on confirmed events, their classification, the associated weights, and time. The protocol does not require raw interaction data, behavior logs, content records, or ongoing observation.

Trust values may be linked to names or codes that do not reveal real identity. Context separation limits the ability to combine trust information across areas, which is a common source of privacy problems in rating systems. Required decay also reduces how long past behavior affects trust values.

TSP is compatible with privacy principles but does not encode any specific legal requirements. Its focus on minimal data and outcome based trust aligns with common data protection principles.

7. Versioning, Extensibility and Contrast with Other Systems

7.1 Versioning Model

TSP uses clear version numbers to tell apart different editions of the specification. Each version defines a complete and self contained set of rules. Version numbers are fixed and unambiguous. A trust value produced under one version must be read using the rules of that same version. A new version is backward compatible if trust values created under the earlier version still hold their meaning under the rules that governed them. Any change that alters how trust values evolve, how decay works, or how events are read is a breaking change and requires a new version number.

7.2 Extensibility

The protocol can be extended through context settings rather than changes to the core rules. Contexts may define event types, confirmation assumptions, weight functions, and decay settings, as long as they follow the protocol constraints. TSP separates its core elements, which are trust value representation, event driven updates, decay requirements, bounded values, and repeatability, from the extension space, which includes context definitions, parameter choices, and optional metadata. Extensions must not redefine or bypass the core rules.

7.3 Contrast with Reputation Systems

Reputation systems collect subjective or platform defined opinions and are often global and permanent. TSP does not use reputation and does not model social perception. Moderation refers to platform level actions such as bans or restrictions. TSP does not moderate behavior. It produces trust signals that other systems may choose to use. Compliance refers to meeting legal or regulatory requirements. TSP is neutral on compliance and policy.

7.4 Governance Neutrality

TSP does not define a governance model for how versions are managed. Decisions on when to publish new versions, accept extensions, or retire features are outside the protocol specification. This neutrality allows different communities to adopt and evolve the protocol without embedding authority into it. Retired features remain valid within the versions in which they were defined. Retiring a feature does not change how existing trust values are read.

8. Industries and Applications

TSP applies in digital environments where trust must be measured over time, outcomes can be confirmed, and privacy or the absence of a central authority limits the use of standard methods. It works best where interactions repeat, the context is clearly defined, and results can be resolved.

8.1 Peer to Peer Marketplaces

Local classifieds, service platforms, rental services, and exchange systems often involve participants with no lasting formal relationship. TSP represents trust as a context bound value that changes with confirmed outcomes and does not rely on global reputation or permanent classification.

8.2 Gig Economy and On Demand Services

Gig platforms depend on repeated interactions between participants with limited shared history. TSP addresses the limits of averaged ratings by modeling trust as a changing value derived from confirmed service outcomes and reduced over time, with a separate context for each service type.

8.3 High Risk Payments and Alternative Finance

Digital asset exchanges, remittance networks, and high risk payment processors can use TSP to represent confidence drawn from settlement outcomes without embedding policy choices or requiring broad identity profiling. Time based decay keeps trust aligned with current behavior.

8.4 Distributed and Decentralized Platforms

Federated services and decentralized networks must coordinate trust without a central authority. TSP provides a framework based on confirmed outcomes and context separation, supporting trust reasoning where coordination and auditability are required.

8.5 Limitations

TSP is not suited to all systems. It does not apply where interactions are one time, cannot be repeated, or are already governed by external regulation. It does not replace identity verification, legal compliance, or dispute resolution.

9. Authorship and Statement of Independence

The Trust State Protocol was created by programmer Karim Lahnin. It is an original contribution to the design of trust mechanisms in digital systems, based on independent study of distributed systems, trust modeling, and privacy focused design.

Origin

TSP was not derived from any existing standard, product, or institutional research program. Its design treats trust as a protocol level concern, separate from reputation systems, identity frameworks, and policy enforcement. The author defined its core ideas, design principles, and trust value model, including bounded context, state transition rules, time based decay, context separation, and protocol neutrality.

Publication Status

Trust State Protocol is not a patented protocol. It is an independently published technical reference available for study and use by engineers and system designers.

Independence

TSP does not represent the position, policy, or endorsement of any organization, standards body, platform, or regulatory authority. Adoption and interpretation remain the responsibility of each implementing system.

Attribution

When referencing TSP in academic, technical, or professional work, the protocol name and author should be cited. Citations should identify the specific version of the specification used.

Karim Lahnin — Author, Trust State Protocol
www.karimlahnin.com

> All definitions and design objectives in this specification come from the author's independent work. Any later contributions, extensions, or revisions by others must be clearly attributed and versioned according to the protocol's versioning rules.

10. Prototype Development and Contact

A prototype integration of the Trust State Protocol is currently in development. The goal is to demonstrate how TSP can be applied in a real system for context based trust tracking, event confirmation, and trust value processing across digital platforms.

Current Status

TSP is at an early integration stage. The core rules and model are defined in this specification. Work on a reference implementation is ongoing. The aim is to provide a working example that engineering teams can study and build on.

Contact

Inquiries from engineering teams, product leads, and investors are welcome. All discussions are handled directly by the author.

Karim Lahnin — Protocol Owner
www.karimlahnin.com