ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Contribution

Attestation Results at the commitment point

FG-TIDA WG1 (CDRA)

Requirements and tests

Proposed text for FG-TIDA WG1 Deliverable 2



Proposed text for FG-TIDA WG1 Deliverable 2

ABSTRACT

Before a service commits something it cannot take back on an AI agent's instruction (cost, a payment, a secret or access to a production system), it needs to know what the agent is executing at that moment, which its last appraisal may no longer show. RFC 9334 separates when a claim's value was generated from when the Relying Party performs the operation, and AR4SI leaves the expiry of individual Trustworthiness Claims undefined. This contribution proposes text for WG1 Deliverable 2 on that moment, the commitment point: a declared freshness deadline, a measurement age per required aspect carried in the Attestation Result, age rules for evidence measured before use or only at launch, and negative tests in which only the appraisal separates two builds, with the measurement of context and memory left to Deliverable 2. Annex C shows how it combines with the timeliness requirements of contribution 12, and Annex D notes binding and disclosure points for other clauses. It defines no protocol or format.

HISTORY

Version	Document	Approval	Focus Group	Working Group
0.0	Attestation Results at the commitment point	Not yet adopted	TIDA	CDRA

To access this Document as a PDF, open the published documentation site in your web browser and click the PDF download link, or navigate directly to https://kingyewchoo.github.io/attestation-results-at-the-commitment-point/specification.pdf.

AUTHORS

Name	Affiliation	Contact
King Yew Choo (proposed editor)	True Primary	k.choo@trueprimary.com

KEYWORDS

Remote attestation, Attestation Results, freshness, measurement age, commitment point, Relying Party, AR4SI, disclosure.

FOREWORD

Agentic AI systems now act as autonomous entities capable of reasoning, planning, and executing tasks across services, infrastructures, and organizational boundaries. This fundamentally changes the nature of identity in digital systems: identity is no longer limited to human users and static machine instances but must also encompass autonomous, dynamic, and goal-oriented agents. And identity alone is insufficient — the core question extends beyond "what an entity is" to "whether, and under what conditions, that entity should be trusted to act".

The Focus Group on Trust and Identity for humans and agentic AI (FG-TIDA) addresses trust management and interoperable digital identity infrastructure for humans and for agentic AI, supporting the development of secure, trustworthy digital ecosystems in which humans and agentic AI can safely interact and collaborate.

FG-TIDA was established under Recommendation ITU-T A.7 (Focus groups: Establishment and working procedures), with ITU-T Study Group 17 as its parent group. Further details are available in the group's terms of reference.

For more news and updates about the TIDA Community, including its supporter organizations, please refer to the community's published resources.

NOTE

This draft proposes input to the WG1 Deliverable 2 outline: requirements for the clause on when fresh Evidence is due and how long Attestation Results stay valid, with Annex D's notes on binding and disclosure for other clauses; the section Requested decision and milestones states the decision requested.

CONTENTS

Proposal . . . 1
- 0.0.1 Scope and motivation . . . 1
- 0.0.2 What this item adds . . . 1
- 0.0.3 Terms . . . 2
- 0.0.4 Requirements . . . 2
- 0.0.5 Tests . . . 3
- 0.0.6 Where the requirements land . . . 4
- 0.0.7 Relations to other work . . . 4
- 0.0.8 Out of scope . . . 5
- 0.0.9 Requested decision and milestones . . . 5

Annex A (informative): one paid request . . . 5
Annex B (informative): what common evidence can establish . . . 6
Annex C (informative): how this item and contribution 12 combine . . . 6
Annex D (informative): notes for the binding and disclosure clauses . . . 7
References . . . 8

Proposal

0.0.1 Scope and motivation

The WG1 draft charter starts from one decision: before a service releases a payment, a secret or access to a production system to an agent, it needs to know continuously what is actually being executed (charter, section 2). This item calls the moment of release the **commitment point**: the Relying Party commits something it cannot take back, whether cost (work, compute, a payment), a secret or access. In the time model of RFC 9334 (Appendix A), it is the event OP, "operation performed".

An agent changes at runtime: its model, code, tools, policies and context can all change (charter, section 2). An appraisal that was fresh when a session began may not describe what sends the instruction at the commitment point. RFC 9334 leaves the freshness threshold to local policy (section 10) and notes that claim values may be generated long before they are signed (section 10.4). AR4SI says a Relying Party may track when a Verifier expires its confidence in individual Trustworthiness Claims, but defines no mechanism for it (section 2.4). No requirement yet ties the age of each required aspect's measurement to the commitment point.

This item proposes such requirements for WG1, with Deliverable 2 as their home. It covers when, and against what limits, measurements are checked; how each part of an agent is measured is the dynamic-attestation work of Deliverable 2 itself. It defines no protocol, message or document format (charter, section 4).

The Attestation Result is one input to the Relying Party's enforcement decision, which belongs to WG2 (charter, sections 4 and 7). This item states only what the result carries and when it is checked. Whether a request starts, how retries are deduplicated, and whether an authority is valid all stay outside it.

0.0.2 What this item adds

- **The commitment event as a freshness deadline.** The Relying Party declares the event at which it commits; each required aspect must be within its age limit at that event. It complements the risk-keyed triggers the charter lists (section 4), which say when to refresh; the deadline says by when.
- **A measurement age per required aspect, carried in the Attestation Result.** The Verifier reports each age relative to the result's own generation, so the check needs no clock shared by the Verifier and the Relying Party and works with nonce-based freshness (R9). It sets requirements for the per-claim expiry AR4SI leaves open; how a format carries it is a question for IETF RATS.
- **Age rules that match real evidence.** Where every component is measured before it loads, a fresh report over the complete log shows everything loaded up to that report, and the aspect's age is the report's age. Where measurements are taken only at launch, an aspect the launch state keeps fixed takes the age of the freshest report showing that launch; one that can change takes its age from the issue of an unpredictable Verifier value bound into the evidence at launch, and otherwise has no age. Parts that change without being loaded, such as context and memory, need the runtime measurement Deliverable 2 defines, which this item asks of it. Annex B shows what four common evidence types can establish.
- **Tools loaded mid-task.** An agent that loads a tool during a task changes what it executes after its last appraisal; under R5 the tool appears in the next report, and the age limit bounds how long it can go unseen.
- **Negative tests that isolate attestation.** Two builds share every other input, so only the appraisal can separate them, and a gate refusing everything fails.

0.0.3 Terms

Terms from RFC 9334 and AR4SI are used as defined there. This item adds four, offered to WG0's terminology work (charter, section 7).

Commitment event
: The event that marks the commitment point, declared by the Relying Party for each kind of request it accepts; an instance of RFC 9334's event OP, operation performed.

Measurement age
: For an aspect, the time from the latest point at which the Evidence establishes the aspect's state to the generation of the Attestation Result, time(RG). Evidence establishes the state at its own freshness point where every load into the aspect since launch is in the log it covers (R5), or where the measured launch state keeps the aspect fixed (R6). Otherwise, where the launch measurement covers the aspect, the age runs from the issue of an unpredictable value the Verifier issued before launch and the Evidence binds; without one, or for an aspect the launch measurement does not cover, such as context and memory, there is no age until Deliverable 2 defines its measurement (R6, R7). A freshness point is placed by a value whose issue time the Verifier knows: its own nonce, an epoch ID or a trusted timestamp (RFC 9334, section 10).

Measured before use
: An aspect whose every component is measured into a log before it loads, so fresh Evidence covering the complete log shows every component loaded up to that moment. It shows what has loaded; changes inside a loaded component do not appear.

Accepted exposure
: An aspect past the Relying Party's age limit, or with no measurement age, on which the Relying Party's enforcement decision (WG2) proceeds anyway. This item asks only that the Relying Party records it as exposure, never as a freshness pass (R12).

0.0.4 Requirements

Each requirement names who must do what, in ITU-T form ("is required to"). Together they are the proposed input to the Deliverable 2 clause on freshness and validity, and they stand on their own. Annex D records what they assume about binding and disclosure, for the clauses that own those.

What the Attestation Result carries

R1. For each kind of request, the Relying Party is required to declare, with its challenge or in a policy it publishes, the commitment event, the aspects it requires at that event (for example executables, configuration, model, tools, policy and context) and a maximum age for each.

R2. Before it accepts requests, the Relying Party is required to fix the Verifiers it accepts, their trust anchors, and the version of its own appraisal policy.

R3. For each required aspect, the Verifier is required to report its measurement age, counted from the point at which that age starts under R5 or R6, or that it has none (R7), and, where an AR4SI Trustworthiness Claim covers the aspect (for example executables for agent and tool code, configuration for configuration and policy, file-system for model files, sourced-data for context drawn from external systems), that claim's Trustworthiness Tier (AR4SI, section 3.2). Aspects no AR4SI claim covers, such as context the agent builds itself (conversation state and memory), are a gap for Deliverable 1 and IETF RATS, and until it closes they have no tier.

R4. When it re-signs a claim, the Verifier is required to keep the start of its measurement age, so the age runs from that start to the new result's generation and is never reduced.

R5. For an aspect measured before use, the Verifier is required to take as the measurement age the age of the freshest Evidence covering the complete log.

R6. For an aspect measured only at launch, the Verifier is required to take as the measurement age the age of the freshest Evidence showing that launch measurement, where the measured launch state includes enforcement keeping the aspect fixed afterwards (for example a read-only, integrity-verified file system). Where nothing keeps it fixed and the Evidence binds an unpredictable value the Verifier issued before launch, such as its nonce, the age runs from that value's issue (Annex B); otherwise the aspect has no measurement age.

R7. The Verifier is required to report an aspect with no measurement age as having none, never as fresh.

R8. The Verifier is required to show in the Attestation Result when a required aspect is missing, negative or incomplete, and to identify the Reference Values and appraisal policy it applied, so that a withdrawn Reference Value can be matched to the results that relied on it. With staleness found under R10 and R11, the enforcement decision can act on all four conditions (charter, section 4).

When it is checked

R9. The Relying Party is required to establish the Attestation Result's age by a mechanism of RFC 9334 section 10: its own nonce carried into the result, an epoch ID, or a timestamp only where it trusts the Verifier's clock. A nonce makes the result specific to that Relying Party.

R10. At the commitment event, the Relying Party is required to add the Attestation Result's age to each required aspect's reported measurement age and check the sum against that aspect's limit.

R11. The Relying Party is required to treat an aspect past its limit, or with no measurement age, as stale. Whether it then refuses the request or proceeds is its enforcement decision (WG2); if it proceeds, the aspect is accepted exposure.

R12. The Relying Party is required to record accepted exposure as exposure, never as a freshness pass.

R13. In long-running work, the Relying Party is required to repeat R9 to R12 at each further commitment event.

Gap for Deliverable 2: context and memory

This item asks Deliverable 2 to define how the context and memory an agent acts on for a request are measured at that request and bound to it, so that the aspect can have a measurement age. Until then the Verifier reports the aspect as having none (R7).

0.0.5 Tests

The negative tests the charter asks for (section 4). Each declares, before it runs, its expected outcome, its trust inputs and which components are real or simulated.

a. Two builds share every input other than attestation, all valid for both. Each run of this test names the appraised measurement and rule that separate the two builds. The accepted build passes and the excluded build fails, so a gate refusing everything fails (R1, R3).

b. An old measurement re-signed is aged from its original start, and past the limit it counts as stale (R4, R11).

c. An aspect measured before use passes with fresh Evidence over the complete log, and a tool loaded after that Evidence appears in the next (R5).

d. An aspect measured only at launch passes with each fresh report when its launch state keeps it fixed; otherwise it passes while within its limit when a Verifier-issued launch value is bound in, and has no age when none is (R6, R7).

e. Work paused past an age limit before a later commitment event is rechecked at that event (R13).

f. A result bound to one Relying Party's nonce fails at another (R9).

g. An Attestation Result whose own age, and the measurement age it reports for an aspect, are each within the limit the Relying Party declared for that aspect fails when their sum exceeds that limit (R1, R10).

h. An aspect with a negative appraisal, or missing from the Evidence, is shown as such in the Attestation Result, never omitted, and the result identifies the Reference Values it was appraised against (R8).

i. A run that proceeds on a stale aspect records it as exposure, never as a freshness pass (R11, R12).

j. A result from a Verifier, or under a trust anchor, that the Relying Party did not fix before accepting requests fails (R2).

Every requirement has at least one test. No interoperability is claimed for any mapping the tests do not cover.

0.0.6 Where the requirements land

Requirements	WG1 deliverable
R1 to R8	Deliverable 2: when fresh Evidence is due, how long Attestation Results stay valid, trustworthiness tiers per aspect
R9 to R13	Deliverable 2: refresh and validity at the commitment event
Gap for Deliverable 2	Deliverable 2: Evidence about an agent's context, which the charter places in scope (section 4)
Annex D notes	Deliverable 2, binding and disclosure (charter, sections 4 and 8); Deliverable 3, where instances share an environment
Gaps found	Deliverable 1: building blocks and gaps; Deliverable 4, if confirmed: conveying tiers and ages

0.0.7 Relations to other work

- **RFC 9334** supplies the roles, the passport model (section 5.1) and the time model this item builds on (section 10 and Appendix A).
- **AR4SI** (draft-ietf-rats-ar4si-11) supplies Trustworthiness Claims and Tiers. Its AR-augmented Evidence (section 4.2) binds a result to the Relying Party's own proof of freshness and lets the Relying Party calculate the time between the Verifier's proof of freshness and its own; R9 and R10, with the request binding Annex D notes, carry that to each aspect, checked at the commitment event. The per-aspect expiry its section 2.4 leaves open goes to IETF RATS as a question; WG1's part is the requirement on Relying Parties.
- **Epoch Markers** (draft-ietf-rats-epoch-markers-08) give a shared notion of freshness without a local clock: one way to place a runtime measurement in time.
- **EAR** (draft-ietf-rats-ear-05), which embeds AR4SI's trustworthiness vector in an EAT (RFC 9711), is the natural carrier for the tier and age per aspect; this item defines neither a format nor an extension.
- **Contribution 12** (fg-tida-contribution-12), a proposed Technical Specification on security requirements for trusted environment measurement and attestation of AI agents, also declared for WG1, gives each measurement report a measurement time and a validity window (notBefore, notAfter) and requires re-measurement before the window ends or when the environment changes (its REQ-TM-01 to REQ-TM-04). The two fit as layers: contribution 12 sets each report's measurement time and validity window where Evidence is produced and checked; this item ages each aspect in the Attestation Result and has the Relying Party check that age at the commitment event. Annex C sets out one combined reading.
- **WG2** makes the enforcement decision this result feeds.
- **Closest IETF drafts**, which guard the same boundary for one operation but do not bound how old each aspect's measurement is there: Action Evidence Boundary (-07), an executor-side model binding evidence and authority to one exact effect; Attestation-Bound Execution Finality (-04), which makes a protected effect reachable only through a verifying boundary.

The item grew from use case 35 in WG0's repository, also recorded on the DataTracker as a WG0 use case. Annex A works its central case through the requirements.

0.0.8 Out of scope

How each part of an agent is measured; enforcement decisions and whether a request starts (WG2); the validity of authorities; payment authorization; new protocols, message or document formats; liability and certification.

0.0.9 Requested decision and milestones

At Paris: allocation to WG1 and adoption as a time-boxed item, recorded as input to the Deliverable 2 outline due at Geneva (WG1 draft charter, sections 5 and 6), where the clause on when fresh Evidence is due and how long Attestation Results stay valid would start from these requirements; Annex C offers one reading of how they combine with contribution 12's timeliness requirements, for WG1 and contribution 12's authors to settle. Annex D's notes go to the binding and disclosure clauses and the Deliverable 1 gap list. At Geneva the item folds into that outline or closes. The author would serve as an editor of that clause once WG1 appoints its Deliverable 2 editors.

Date	Output
Paris, 30 November to 1 December 2026	Allocation to WG1 and adoption as a time-boxed item; the first test list.
December 2026	Requirements text and tests; the expiry question to IETF RATS.
Geneva, 26 to 27 January 2027	In the Deliverable 2 outline, or closed.

Annex A (informative): one paid request

This annex works one case through the requirements. It adds no requirement to them.

A deployer runs one agent build for about 17,000 consumers. Each orders one £15 answer a week from a seller of expert time, paying on their own card within £20 per-purchase and £100 weekly limits. The case is hypothetical; amounts and volumes are illustrative.

The seller's acceptance rule concerns the build: one appraisal of it governs about 17,000 purchases a week, while each user's spending limits bind only that user. Payment approval, even with a card token bound to a registered agent, shows which agent may spend; it says nothing of the state that agent runs in. A fault in one build reaches every one of those purchases at once.

Requirement	In this case
Commitment event (R1)	Dispatch of the expert task
Required aspects (R1)	Executables and configuration of the agent build, its tools, and its context
Age rule (R5, R6)	Executables and tools measured before use; configuration measured at launch and kept fixed by a read-only, integrity-verified file system
Request binding (Annex D, D1)	A stable purchase identifier, signed with the seller's challenge
Spending authority	Bound to the request at the authorization interface (charter, section 7), a dependency of this item
Age check (R10)	At dispatch, after payment approval and before work starts

The outcome: executables and tools pass with each fresh report. Configuration passes with each fresh report showing the same launch, with no relaunch needed. Context has no measurement age today, so it counts as stale, and the seller either refuses or proceeds on accepted exposure (R11). For it to pass, it needs the measurement this item asks of Deliverable 2 (the context the agent acts on for this request, measured at the request and bound to it) and a tier under R3.

Payment authorization itself stays with the payment actors. Whether a payment method can carry these items is for its authors to say.

Annex B (informative): what common evidence can establish

How the age rules (R5 to R7) apply to four widely used evidence types. Each binds a challenge into fresh Evidence. They differ in whether components loaded after launch are measured. None carries a launch time, but three can bind an unpredictable value fixed at launch, which places launch after that value was issued.

Evidence	Freshness anchor	Launch measurement	Measured before use, after launch	Launch placed in time
TPM 2.0 quote	Qualifying data (nonce) in the quote	PCRs from measured boot	PCRs with an event log, such as Linux IMA	Not carried; clock and reset counters are relative
Intel TDX quote	REPORTDATA	MRTD	RTMRs with an event log	Through a value bound in MRCONFIGID
AMD SEV-SNP report	REPORT_DATA	MEASUREMENT	None in the report; a virtual TPM can add it	Through a value bound in HOST_DATA
Arm CCA realm token	Challenge	RIM	REMs with an event log	Through a value bound in the Realm Personalization Value

What follows for the age rules:

- **Measured before use** (R5): where the event log covers every component loaded since launch, a challenge-fresh report gives each such aspect the age of that report. TPM with IMA, TDX with RTMRs and CCA with REMs can meet this.
- **Launch only** (R6, R7): where the measured launch state keeps the aspect fixed, such as a read-only, integrity-verified file system, each fresh report showing the same launch measurement gives it that report's age. Where nothing keeps it fixed and the launcher binds an unpredictable Verifier-issued value at launch, age runs from that value's issue; without it, the aspect has no measurement age. The binding costs one Verifier round trip per launch, and some confidential-container stacks already use these fields for policy digests, so the value may have to share the field. An SEV-SNP report alone covers nothing loaded after launch, unless the deployment adds a runtime measurement source such as a virtual TPM.
- **Changing without loading**: Linux IMA can re-measure some kernel-held state when it changes (critical data, such as device-mapper state). None of the four measures an agent's context or memory; those aspects need the runtime measurement Deliverable 2 defines.
- **Epoch markers** can place a runtime measurement after a known point without a local clock, giving it a measurement age where a nonce round trip is impractical.

Annex C (informative): how this item and contribution 12 combine

Contribution 12, a proposed Technical Specification of its own, and this item answer two halves of one timeliness question. Contribution 12 works where Evidence is produced and checked: when it was measured and how long a report stays valid. This item works on the Attestation Result and the Relying Party's check: how old each aspect's measurement is when it commits. The table gives one combined reading, offered for WG1 and the authors of contribution 12 to settle.

Question	Contribution 12	This item	Combined
When was it measured?	measurementTime from a trusted time source, on each report (REQ-TM-01)	An age per aspect (R3)	Absolute times stay in Evidence to the Verifier; ages go in the Attestation Result to the Relying Party (Annex D, D4)
How long is it valid?	notBefore and notAfter, set by environment type and risk (REQ-TM-01, REQ-TM-02)	A maximum age per aspect, declared by the Relying Party (R1)	The report's window bounds reuse of Evidence; each aspect's age is checked at the commitment event (R10)
When is it refreshed?	Before the window ends and on environment-change events (REQ-TM-03, REQ-TM-04)	The commitment event as the deadline (R1, R10)	Their triggers say when to refresh; the commitment event says by when
Does trust fall within the window?	Linear or step-wise decay of trust levels within the validity period (REQ-TD-05)	No decay: each aspect is within its limit or stale at the commitment event (R10, R11)	Decay can grade trust inside the window; the commitment-event check is the outer limit
A launch measurement in a new report	The new report gets a new measurementTime (REQ-TM-01)	Keeps its start, or has no age if no Verifier value was bound at launch, unless the launch state keeps the aspect fixed (R4, R6)	To settle together: whether a re-reported launch measurement restarts the window or keeps its start

Annex D (informative): notes for the binding and disclosure clauses

These notes are not proposed text for the freshness clause. They record what its requirements assume about binding a result to a request and about disclosure, for the Deliverable 2 clauses that own those (charter, sections 4 and 8) and for the Deliverable 1 gap list.

D1. **Request binding.** The instance signs the Relying Party's challenge and the request identifier together, with a key held in the measured component and identified in the Attestation Result. That the measured code signs only requests originating inside the agent is a property the appraisal of that code has to establish.

D2. **Shared environments.** A deployment that shows only where the key is held claims only that, and instances sharing one environment are identified as that environment.

D3. **Scoped identifiers.** The instance key and the request identifier are scoped to the Relying Party that sees them, unless that Relying Party needs linking across Relying Parties and the principal the agent acts for has agreed.

D4. **Ages, and rounding their start.** The Attestation Result carries ages, never absolute measurement or launch times. A Verifier can also round each age's start down to a whole multiple of a granularity it publishes, so rounding only lengthens an age and every result about one measurement gives the same rounded time; a result generated soon after a measurement still narrows it, so rounding reduces linkage and some remains. A Relying Party would then fix (R2) only Verifiers whose granularity, added to the result age and the measurement age it expects, stays within its smallest age limit.

D5. **Raw Evidence.** The Relying Party reads raw Evidence only for a claim its policy names and the Attestation Result does not carry. In the background-check model (RFC 9334, section 5.2), it relays Evidence it cannot read, for example Evidence encrypted to the Verifier.

D6. **Verifier linkage.** A Verifier states whether and when it links one instance's appraisals across Relying Parties, for example to warn them of a compromise, and how long it keeps appraisal records. Anonymous attestation, such as TPM 2.0 ECDAA, hides the attestation key, though the clock fields in each TPM 2.0 quote can still link one TPM's quotes.

Disclosure, as charter section 8 asks:

Item	Received by	Linkable across Relying Parties	A failed appraisal reveals
Tier per aspect	Relying Party	No, beyond the build's tier	Which aspect failed
Attestation Result	Relying Party	It can, if it carries values unique to the instance; a nonce stops replay to another Relying Party and leaves linkage open (R9)	Which aspects it covers
Measurement age	Relying Party	Yes, possibly, where ages are exact; partly, where they start from a rounded time shared by instances measured in the same interval, since a result generated soon after a measurement narrows that interval (D4); a timestamp on the result can add a link (R9)	That an aspect is stale
Instance key	Relying Party	No, where scoped per Relying Party (D3)	Nothing further
Request identifier	Relying Party	No: one request, one Relying Party	Nothing further
Raw Evidence	Relying Party, only under D5 (relayed unreadable in the background-check model)	Yes, possibly: full software state	The named claim's value
Raw Evidence, and each Relying Party's nonce	Verifier	It could, as it sees every appraisal for every Relying Party; D6 asks it to say when it does	The full appraised state

Tests these notes suggest, for the clauses that take them up: a signature requested by a caller outside the measured agent, or made by a key the result does not identify, fails (D1); two instances sharing one environment receive results that identify the environment and neither instance (D2); one instance presenting to two Relying Parties cannot be linked by key or request identifier (D3); a result carries no absolute measurement or launch time, and, for an age whose start falls late in a grid interval, its generation time minus that age equals the start rounded down to the published granularity (D4).

References

IETF

- RFC 9334, Remote ATtestation procedureS (RATS) Architecture, January 2023.
- RFC 9711, The Entity Attestation Token (EAT), April 2025.
- draft-ietf-rats-ar4si-11, Attestation Results for Secure Interactions, 31 October 2026.
- draft-ietf-rats-ear-05, EAT Attestation Results, 2 November 2026.
- draft-ietf-rats-epoch-markers-08, Epoch Markers, 2 November 2026.
- draft-schrock-action-evidence-boundary-07, The Action Evidence Boundary for Consequential Agent Effects.
- draft-das-rats-attestation-bnd-execution-finality-04, Attestation-Bound Execution Finality for GPU, AI Accelerator, DPU, SmartNIC, and Confidential-Computing Infrastructure.

FG-TIDA

- Terms of Reference, ITU-T Focus Group on Trust and Identity for Humans and Agentic AI.
- WG1 draft charter, Continuous and Dynamic Remote Attestation for Agentic AI (CDRA), version 0.5, 3 November 2026.
- WG2 draft charter, Identity, Behaviour and Enforcement for AI Agents (IBEA), version 0.4, 3 November 2026.
- Contribution 12, Security requirements for trusted environment measurement and attestation of AI agents (proposed Technical Specification, WG1), TIDA DataTracker, version 0.0, 16 November 2026.
- WG0 use case 35.

LIST OF TABLES

LIST OF FIGURES