

Service Promise Protocol

VERSION 0.1 · PUBLIC WORKING DRAFT · 25 AUGUST 2026

A vendor-neutral record format and lifecycle for service promises in hospitality: the obligations a venue accepts towards a guest or represented party, and the evidence by which those obligations are settled.

Item	Detail
Status	Public Working Draft, published for review. This document is not a standard and claims no adoption.
Publisher	Invenio Group Holdings Limited. Invenio HALO is the reference implementation of this protocol.
Editor	Ahmed Rahim Khan
Feedback	pilots@inveniohalo.com
Licence	Text licensed under CC BY 4.0. The HALO name and any certification marks remain trademarks of Invenio Group Holdings Limited and are not licensed by this document.

1 Purpose and scope

1.1 This specification defines a vendor-neutral record format and lifecycle for service promises in hospitality operations. It exists so that any operator, of a single terrace or of a global group, can answer four questions from the record alone: what has been promised, who owns each promise, which promises remain outstanding, and which were objectively kept.

1.2 In scope: the service record, its request and promise components, the lifecycle and its transitions, ownership and transfer, intake and service budgets, evidence

and verification, settlement outcomes and their supersession, immutability and portability, privacy and worker protections, and conformance.

1.3 Out of scope: routing algorithms, user interfaces, hardware, pricing, staffing and scheduling, and any scoring or certification programme. The protocol defines what a trustworthy record is, not how work is organised.

1.4 Existing hospitality systems record tasks, tickets, messages, bookings or transactions. This protocol is designed specifically for an evidence-backed record of service obligations and their settlement, in which completion cannot be established solely by the party that performed the work.

2 Conformance language

The key words **MUST**, **MUST NOT**, **SHOULD**, **SHOULD NOT** and **MAY** are to be interpreted as described in RFC 2119. A system claims conformance at one of the levels defined in section 14. Text outside numbered clauses and rules is informative.

3 Definitions

Term	Meaning
Beneficiary	The party to whom a promise is owed: a guest, a group, an organiser, or a party they represent.
Issuer	The venue or operating entity that accepts obligations.
Request	An expressed need originating from a beneficiary, a staff observation, a policy or an external system. A request is not yet a promise.
Service record	The top-level object: a mandatory request component, a promise component created only by acceptance, and a settlement.
Acceptance	The issuer's act of taking on the obligation. Acceptance creates the promise component of the record.
Promise	The component of a service record created by acceptance: an obligation with exactly one owner, fixed budgets and a defined route to settlement.
Owner	The single accountable person, role or system currently responsible for the promise.
Intake budget	The declared window, running from <code>requested_at</code> , within which a request must be accepted or declined. Fixed in issuer policy before any request exists.
Service budget	A time window the issuer fixes at acceptance for a stage of the promise, such as arrive, settle or verify.
Event	An append-only entry recording a fact about the record, in

Term	Meaning
	total order.
Evidence	An event of a class defined in section 9 that supports settlement.
Verifier	The party or system, other than the performer alone, whose evidence settles the record.
Settlement	The record reaching a terminal outcome under section 10.
Settlement version	One authoritative statement of outcome. Adjudication records a superseding version; superseded versions remain visible.
Adjudication	Resolution of a dispute, producing a superseding settlement version.
Session	A pseudonymous identifier linking a beneficiary to a service point for a bounded period.

4 The record lifecycle

4.1 A conformant record moves through the following states. Optional informative events, such as progress notes, may occur inside any state without changing it.

State	Meaning
REQUESTED	A need has been recorded, with requested_at and an intake budget. No obligation exists yet.
ACCEPTED	The issuer has taken on the obligation. The promise component now exists, with an owner and budgets fixed.
FULFILMENT_CLAIMED	The performer asserts that the obligation has been discharged. A verification window opens.
SETTLED	Terminal. The record carries a settlement version with exactly one outcome from section 10.

4.2 Declines and cancellations are not separate states. They are settlements: a declined request settles with outcome DECLINED, and a withdrawn record settles with outcome CANCELLED. Every recorded need therefore has a terminal, reportable end.

4.3 A settled record is never reopened. A dispute records adjudication events and, where upheld, a superseding settlement version under rule R12 and clause 10.5.

5 Core rules

R1 A request **MUST NOT** be treated as an obligation until the issuer accepts it. Acceptance creates the promise component of the record.

- R2** Every request **MUST** reach acceptance or a recorded decline within its intake budget, which comes from pre-declared issuer policy and runs from requested_at. A lapse **MUST** be recorded as an intake breach and escalated under that policy.
- R3** Every accepted promise **MUST** have exactly one accountable owner at every moment until settlement.
- R4** Promise budgets run from accepted_at unless a budget declaration explicitly states another basis. Transfer of ownership **MUST NOT** reset or extend any budget.
- R5** Promise budgets **MUST** be fixed at or before acceptance and **MUST NOT** be amended retrospectively. Intake budgets are fixed in policy before any request exists.
- R6** A performer's claim of fulfilment **MUST NOT**, by itself, settle a record as kept. Verified settlement requires evidence of class E1 to E4 (dual-sided closure).
- R7** Every service record **MUST** reach a terminal outcome. No record may be deleted, and none may close silently, or by expiry, into a kept state.
- R8** A decline **MUST** record a reason attributed to a policy or protocol, never to a named individual.
- R9** Timestamps **MUST** be assigned by the recording system from a single authoritative clock, and the events of a record **MUST** be totally ordered.
- R10** The record is append-only. A correction is a new event referencing the event it amends; the amended event remains visible.
- R11** Where a breach triggers recovery, the recovery action **MUST** be recorded as a new service record that references the breached record.
- R12** A dispute **MUST NOT** alter an existing settlement version. Adjudication **MUST** record a superseding version that references the version it supersedes; superseded versions remain visible and the latest version is authoritative.

6 The service record

6.1 The top-level object is a service record: a mandatory request component, a promise component created only by acceptance, and a settlement. A declined request therefore carries no owner, no accepted_at and no promise budgets. Calling every record a promise would imply obligations that never existed.

6.2 Classification is two-axis. Criticality states the consequence if the obligation fails; urgency states how quickly action is required. Obligations concerning declared allergen exposure, medical safety or immediate risk of physical harm **MUST** be classified criticality SAFETY. Accessibility obligations **MUST** be classified

by the consequence and urgency of failure and **MUST NOT** be automatically downgraded because they originate as accessibility requests.

Field	Requirement	Meaning
record_id	MUST	Globally unique, stable identifier of the service record.
issuer	MUST	Venue or operating entity.
beneficiary	MUST	Pseudonymous session reference and its disclosure level. Personal identity appears only as permitted under section 13.
origin	MUST	Physical control, staff entry, beneficiary device, reservation, policy or API.
obligation	MUST	Obligation class plus a description of the requested outcome.
criticality	MUST	STANDARD, MATERIAL or SAFETY. Defaults from obligation-class policy.
urgency	SHOULD	ROUTINE, TIME_BOUND or IMMEDIATE.
request.requested_at	MUST	Moment the need was recorded. Intake clocks run from here.
request.intake_budget	MUST	Window for acceptance or decline, fixed in pre-declared issuer policy.
promise.accepted_at	MUST *	Moment of acceptance.
promise.owner	MUST *	Current owner and the full chain of custody.
promise.budgets	MUST *	Windows fixed at acceptance, running from accepted_at unless a budget declares another basis.
promise.conditions	MAY	Dependencies and represented context recorded at acceptance.
events	MUST	Append-only, totally ordered entries across the record.
evidence	MUST (L2+)	Entries of the classes in section 9.
verifier	MUST (L2+)	Party or system whose evidence settled the record.
settlement.outcome	MUST	Exactly one value from section 10, per settlement version.
settlement.settled_at	MUST	Moment the settlement version was recorded.

Field	Requirement	Meaning
settlement.version	MUST	Version number; a superseding version references its predecessor.

6.3 Fields marked with an asterisk exist only when the request was accepted.

7 Ownership, transfer and escalation

7.1 Acceptance assigns the first owner. Where policy allows, self-assignment by the accepting person is acceptance.

7.2 A transfer entry **MUST** record the previous owner, the new owner and the protocol authorising the transfer.

7.3 An escalation entry **MUST** reference its arming condition, for example a lapsed acknowledgement budget, and the escalation protocol identifier.

7.4 A routed assignee **MAY** decline an assignment. The decline is attributed to the filtering protocol, such as break protection or a load cap, and **MUST NOT** be published against the individual (section 13).

8 Intake and service budgets

8.1 The intake budget comes from pre-declared issuer policy per obligation class and runs from requested_at. Promise budgets are declared per stage, in seconds, at acceptance, and run from accepted_at (rule R4).

8.2 A budget lapse creates a breach event. Breach does not terminate the record; a breached promise can still settle KEPT_LATE. Budgets for obligations of criticality SAFETY **MUST** come from declared class policy and **MUST NOT** be loosened at acceptance by per-instance discretion.

9 Evidence classes

Class	Name	Meaning
E0	Performer claim	The owner or performer asserts completion. Never sufficient alone.
E1	Beneficiary confirmation	Explicit confirmation from the beneficiary session.
E2	Independent system event	Payment or POS line, door or lock event, dispatch or delivery record, or an equivalent external system fact.
E3	Second-party attestation	A staff member other than the performer attests from direct observation.

Class	Name	Meaning
E4	Instrumented observation	A sensor, proximity or presence signal defined in venue configuration.

9.1 Verified settlement **MUST** include at least one item of evidence of class E1 to E4. Class E1 **SHOULD** be sought where practical. Beneficiary silence is not evidence (clause 10.3).

10 Verification, closure and outcomes

10.1 When fulfilment is claimed, a verification window opens. The window is declared in the same manner as a budget.

10.2 Every settlement version records exactly one outcome:

Outcome	Meaning
KEPT	Verified complete within all budgets.
KEPT_LATE	Verified complete after a budget breach.
UNVERIFIED_COMPLETE	Performer claim with no class E1 to E4 evidence inside the verification window.
BROKEN	The obligation was not discharged: abandoned, or breached with no completion.
DECLINED	The issuer did not accept the request, with the reason recorded under rule R8.
CANCELLED	The beneficiary withdrew, or the represented context ended before fulfilment.

10.3 Absence of a beneficiary response **MUST NOT** be recorded as satisfaction or as class E1 evidence.

10.4 Conformant reporting **MUST** distinguish KEPT, KEPT_LATE and UNVERIFIED_COMPLETE. Aggregating them into a single completed figure is non-conformant.

10.5 A dispute does not alter the existing settlement. Adjudication **MUST** record a superseding settlement version, carrying one outcome from clause 10.2 and the rationale, and referencing the version it supersedes (rule R12).

10.6 UNVERIFIED_COMPLETE records an evidence limitation, not, by itself, a service failure. Conformant reporting **SHOULD** publish verification coverage, the share of settlements carrying class E1 to E4 evidence, alongside outcomes. Outcome, verification coverage and beneficiary satisfaction are distinct measures and **MUST NOT** be conflated.

11 Breach, decline, cancellation and dispute

- A breach entry records the lapsed intake or promise budget, the state of the record at the lapse, and any escalation or recovery it armed (rules R2 and R11).
- A decline entry records the policy reason (rule R8) and any alternative offered, itself recorded as a child record where pursued.
- A cancellation entry records who withdrew and at which state.
- A dispute **MAY** be opened by the beneficiary, the issuer or the owner within a declared dispute window, and resolves by adjudication under clause 10.5.

12 Immutability, amendment and portability

12.1 Records are append-only under rule R10. Level 3 systems **MUST** maintain tamper evidence across events, for example hash chaining.

12.2 The issuer **SHOULD**, and at Level 3 **MUST**, be able to export its complete service records, including settlement versions and amendment history, in the open interchange form of this protocol. Records belong with the operation, not with any vendor.

12.3 Access rights for beneficiaries and workers are baseline protections under clause 13.1 and apply at every conformance level.

13 Privacy and worker protections

13.1 Baseline protections. The following apply at every conformance level:

- Beneficiaries are pseudonymous sessions by default.
- Identity attributes are disclosed by consent and at the minimum level an obligation requires: a performer may see that a severe nut allergy is present and beneficiary-verified, not a profile.
- Attributes of criticality SAFETY on which an obligation depends **MUST** be presented to the performer at acceptance.
- A beneficiary **MUST** be able to obtain the records of promises made to their session. A worker **MUST** have access to events naming them and a route to contest them, recorded as adjudication entries.
- Reporting **MUST NOT** publish rankings or league tables of individual workers. Operational analytics are aggregated by default.
- Evidence containing personal data **SHOULD** be minimised and retained only as issuer policy and applicable law require.

13.2 A worker **MAY** export a credential derived from records naming them. Participation is opt-in, the export is controlled by the worker, and an employer **MUST NOT** condition employment on it. Failures arising from conditions outside the worker's control **MUST** be distinguishable in any derived credential.

14 Conformance levels

Level	Name	Requirements
Level 1	Recorded	Sections 4 to 8, the outcome vocabulary of section 10, clause 12.1, and the baseline protections of clause 13.1. Verification may be absent; every completion is then UNVERIFIED_COMPLETE.
Level 2	Verified	Level 1 plus sections 9, 10 and 11 in full: dual-sided closure, evidence classes, verification coverage and adjudication.
Level 3	Attested	Level 2 plus section 12 in full: tamper evidence and portability, published verification rules and audit readiness.

14.1 A system claims conformance at a named level. Baseline protections are not graduated; they apply at every level. Note the consequence of Level 1: a system without independent verification cannot report anything as kept. Honesty is enforced by vocabulary before it is enforced by audit.

14.2 Certification marks, including any HALO mark, are administered outside this specification and are not conferred by conformance claims.

15 Worked examples

15.1 A restaurant bill

A guest at table 14 requests the bill from a table control at 21:04:00. Class policy declares an intake budget of 30 seconds; at acceptance the promise budgets are fixed at settle 180 seconds and verify 120 seconds. The floor protocol accepts at 21:04:04, four seconds into the intake budget, and assigns Sofia as owner. Sofia claims fulfilment at 21:05:31 (E0). The POS records payment at 21:05:40 (E2) and the guest confirms at 21:05:42 (E1). Settlement version 1 records KEPT at 21:05:42: one hundred and two seconds after the request and ninety-eight seconds after acceptance, with two independent items of evidence.

15.2 A hotel early check-in

At booking, the hotel creates two linked service records. Record A is a promise: assess room availability and communicate the decision by 11:00 on the day, owned by the front office role. Record B is the early check-in request itself, room ready by 13:00, whose acceptance is conditional on the outcome of A. On the day, availability is confirmed at 10:52 and communicated: A settles KEPT on the communication event (E2). B is accepted, and key issuance at 12:41 (E2) settles B KEPT. In the alternative run the check fails: A still settles KEPT, because the assessment and the communication happened on time, and B settles DECLINED under the published

availability policy, with the offered alternative recorded as a child record. Nothing accepted is ever reclassified as unaccepted.

15.3 An event with a guaranteed service level

An organiser books a private event for 120 guests. The agreement declares budgets per obligation class and a recovery policy: any beverage promise breaching 240 seconds arms a supervisor recovery record referencing the parent, plus a fee credit entry. During service, one beverage promise breaches at 22:14. The recovery record is created automatically and settles KEPT in 95 seconds; the parent settles KEPT_LATE. The settlement report lists 212 service records: 205 KEPT, 4 KEPT_LATE, 2 UNVERIFIED_COMPLETE and 1 BROKEN with its recovery record. Under clause 10.4 these figures are never merged.

A Appendix A. Reference interface (non-normative)

This protocol is implementation-neutral. Conformance concerns records and rules, not transport. A venue with no integrations can conform through an implementation's own controls and surfaces; nothing in this specification requires an API integration at go-live. For systems that do integrate, the reference implementation exposes the lifecycle as follows.

POST /promises	create a request
POST /promises/{id}/accept	accept and fix budgets; assigns the owner
POST /promises/{id}/transfer	transfer ownership under a protocol
POST /promises/{id}/claim	claim fulfilment (E0)
POST /promises/{id}/evidence	attach evidence of class E1 to E4
POST /promises/{id}/dispute	open or resolve a dispute
GET /promises/{id}	full record with events and evidence

B Appendix B. Example record (non-normative)

```
{
  "record_id": "rec_9f24c1",
  "issuer": "aster-house",
  "beneficiary": { "session": "ses_t14_2104", "disclosure": "minimum" },
  "origin": "table_control",
  "obligation": { "class": "bill", "description": "Present and clear the bill" },
  "criticality": "MATERIAL",
  "urgency": "TIME_BOUND",
  "request": { "requested_at": "2026-08-25T21:04:00Z", "intake_budget_s": 30 },
  "promise": {
    "accepted_at": "2026-08-25T21:04:04Z",
    "owner": { "current": "sofia.r", "chain": ["floor_protocol", "sofia.r"] },
    "budgets": { "settle_s": 180, "verify_s": 120 }
  },
  "events": [ "...append-only, totally ordered..." ],
  "evidence": [
    { "class": "E2", "source": "pos.payment", "at": "2026-08-25T21:05:40Z" },
    { "class": "E1", "source": "beneficiary.confirm", "at": "2026-08-25T21:05:42Z" }
  ]
}
```

```
],  
  "verifier": "beneficiary_and_pos",  
  "settlement": { "version": 1, "outcome": "KEPT", "settled_at": "2026-08-  
25T21:05:42Z" }  
}
```

c Status and change log

- v0.1, 25 August 2026. Public Working Draft. Incorporates editor corrections before first publication: service-record structure with intake budgets, two-axis classification, linked-record conditions, baseline protections at every conformance level, and settlement supersession for disputes.

This document becomes a standard only through adoption, interoperability and credible governance, none of which it claims today. Comments, implementations and objections to pilots@inveniohalo.com.