AgenticSafetyEval Agent Contract v1

Status: Profile C (black-box) is the only evaluation profile. This document is the normative interface standard for evaluating an agent with AgenticSafetyEval. The key words MUST, SHOULD, and MAY are to be interpreted as in RFC 2119. The contract is versioned independently of the package (`agenticsafetyeval-agent/1`); implementations pin a contract version.

Purpose

Agents under test are session-shaped systems: Python classes, HTTP services, vendor engines that execute their own tools internally (vendor session engines). The contract therefore standardizes the session interface and the state interface — the only two surfaces the harness touches.

Profile C — Black-box state (the only profile)

For self-executing agents (session engines, vendor systems). The agent MUST accept one customer utterance at a time on a per-run session and return a reply; it MAY return structured metadata, which the harness records as **untrusted claims**, never as truth. A StateProvider MUST be available that can (the agent owner may draft it; who must *operate* it for a result to count as certified evidence is defined in `docs/trust_model.md` — results from an evaluatee-operated provider are self-reported):

- `seed(initial_state)` — load the scenario's starting state into a sandbox,
- `snapshot()` — return an isolated JSON snapshot of the current state,
- `reset()` — remove all run residue.

The harness drives scripted customer turns (answering agent questions only from scenario-declared clarification facts; a clarification is never a confirmation), then grades the final snapshot against the scenario's authorized final states, and checks reply claims against it.

The harness provides **no tools** to the agent, executes **no calls** on its behalf, and applies **no gate**. The agent's internal actions are invisible to the harness; the verdict is final-state correctness plus claim verdicts, computed deterministically.

Reference bindings: the Python `SessionAgent` protocol (`say(text) -> SessionReply`), `HttpSessionAgentAdapter` (POSTs `{"session_id", "transcript", "language", "channel"}` and reads `reply` / `metadata`), and the banking demo in `agenticsafetyeval.mocks.banking_session`.

Every report MUST label these results `execution_mode: black_box`.

Cross-cutting requirements

- **Session isolation.** One agent instance serves one run. Fresh

instances MUST NOT share conversational state (checked by AGT-04).

- **Robustness.** Adapters MUST tolerate empty utterances and sustain

multi-turn conversations without raising; a raise aborts the run and is scored, not hidden.

- **Reproducibility.** Runs SHOULD be repeatable; nondeterministic

agents are reported as pass rates over repeats (`--repeats`), never as single verdicts presented as exact.

- **Snapshot honesty.** The snapshot MUST be read from the sandbox

system of record, never derived from the agent's replies or metadata.

- **Provider control.** Every report MUST label who operated the

StateProvider (`provider_trust`: `harness`, `auditor`, or `self_reported`). The harness verifies the provider continuously: the post-seed snapshot MUST equal the seeded initial state or the run aborts, state is snapshotted after every turn, and certified evaluations SHOULD enable the randomized provider probe (`--provider-probe`). See `docs/trust_model.md`.

Conformance

- `agenticsafetyeval certify-agent --agent module:Class` runs AGT-01…AGT-05

against a session-agent adapter (response shape, empty-utterance tolerance, JSON metadata, session isolation, multi-turn robustness).

- `agenticsafetyeval certify-state-provider --provider module:Class` runs

SP-01…SP-04 (seed/snapshot round-trip, snapshot isolation, reset, deterministic re-seed).

Passing conformance means the integration honors this contract. It does not certify that an agent is safe, and it is not a regulatory approval.

Versioning

Breaking changes to profile semantics increment the contract major version. Scenario and run-record formats remain backward compatible within a major version; new fields are additive with defaults.