RED BEAR: ENTERPRISE DECISION INTELLIGENCE · DATA SHEET · A · CAPTURE

The decisions you never knew were made

You cannot govern a decision you never saw. Capture arrives where the decision is taken and costs the decider seconds.

KEY POINTS

- Six routes in, one record out: meetings and calls, a decision inbox, business applications, a custom connector, portal and mobile, and an API agents call before they act
- Coverage, not cleverness, sets the ceiling on everything else: the register can only hold what reaches it
- Detection proposes, the named decider confirms; nothing enters the register on a machine's say-so alone
- The decision is captured, not the conversation: consent first, redaction before storage, and one tap to discard

What it is

A consequential decision leaves behind a transaction, a thread, or nothing at all. A commitment made on a Teams call is gone when the call ends. An approval given by email sits in one person's mailbox, findable by that person alone. A discount applied in CRM records the amount, the date and the approver, and not one word of the judgement that produced it.

Capture is the set of routes by which a decision enters Red Bear: Enterprise Decision Intelligence. Whichever route it arrives by, a decision normalises into the same record: what was decided, who or what decided it, which class it belongs to, the policy in force at that moment, and the reasoning. Nothing downstream needs to know which door it came through, which is the point of one capture layer rather than a governance feature inside each system.

The pain point

Each of those is a decision the organisation has made and cannot see. It cannot be counted, so nobody knows how many there are. It cannot be evaluated against policy, because policy was never shown it. It cannot be reviewed six months later against what actually happened, because there is nothing to attach the outcome to. The decision had consequence; the record has none.

This is the constraint on everything else the product does. The register can only hold what reaches it, policy can only evaluate what it is shown, escalation can only interrupt a decision it knows about, and analysis can only learn from decisions that were written down. So the layer is built on one assumption: people will not come to a new screen to record a decision they have already made. Capture has to arrive where the decision is actually taken, in the meeting, in the mail thread, in the application, in the agent's own code path.

How it works

Six routes, one contract. Every route terminates in a single normalised decision candidate; no route writes to the register directly, and consent is evaluated before content is read. Productivity connectors capture in Teams, Outlook, SharePoint and Office and let the decider approve in place. The messaging connector reads business chat for the commitments agreed there, with consent recorded. Enterprise application connectors read the decision at the transaction in ERP, CRM, HR and supply chain, hold it where policy requires, and write the outcome back later. A custom connector lets the customer administrator wire anything with an API, against a published contract, without a vendor release. Voice and call intake extracts the decision and its spoken reason from the transcript, and discards the transcript. The portal and mobile app are a capture route with the same contract as every connector, not a privileged path. And the agent API is called before the agent acts, so the decision is evaluated at the moment of intent rather than reconstructed from logs.

Proposed, then confirmed. Most consequential decisions are never declared as decisions: "let's go with supplier B" is a commitment, and nobody files it as one. Decision detection finds the implicit decision in conversation, meetings and mail and proposes it to the owner, who confirms, edits or discards before anything is recorded. A candidate not confirmed within the class's window is discarded, not held. Rationale prompting asks for the missing Cartesian elements at that moment, briefly and by voice where typing is not practical, scoped to the class's evidence standard: a routine decision is asked for none, a critical one for all plus dissent. Every captured decision is classified on arrival, so guardrails, ownership and target time apply from the first second.

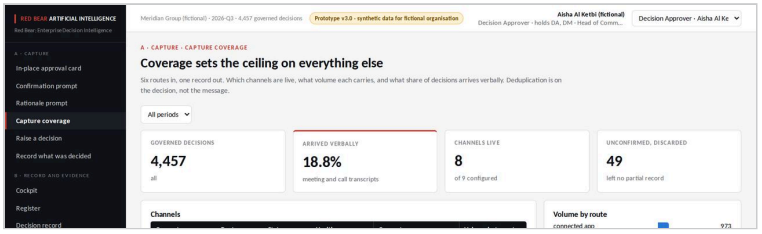
Clean and respectful. One decision routinely appears in three channels: discussed on a call, confirmed in chat, executed in the ERP. The product deduplicates on the decision, not the message, so it becomes one record with three sources. And capture that feels like surveillance gets switched off, by staff, by a works council, or by a general counsel who reads the retention policy. So the layer captures the decision, not the conversation: consent is recorded before anything is taken from a channel, personal data is redacted, what is not decision content is discarded, and the acting identity comes from the verified sign-in rather than the text of a

message. The test is simple. If an employee shown their own captured record would say it describes a decision they made and nothing else about their day, capture is doing its job.

What you get

Capability	What it does	Requirements
Productivity and messaging connectors	Capture in Teams, Outlook, SharePoint, Office and business chat; approve in place; consent recorded	FR-CAP-01, FR-CAP-02
Enterprise and custom connectors	Read the decision at the transaction; hold it when policy says; write the outcome back; wire any API without a vendor release; a connector store and partner SDK	FR-CAP-03, FR-CAP-04, FR-CAP-06
Voice and call intake	The decision and its spoken reason extracted from the transcript, after consent, before the transcript is discarded	FR-CAP-05
Decision capture and detection	One record per decision, deduplicated on the decision; implicit decisions proposed and confirmed by the owner	FR-CAP-07, FR-CAP-08
Rationale, dissent and point-of-decision prompts	The missing Cartesian elements, the minority view and "what would make this wrong", asked at the moment of commitment, confirm-first	FR-CAP-10, FR-CAP-11, FR-CAP-12
Raise, decide, send and record	Raise a decision in the portal; decide in place where dial and guardrails permit; send to whoever holds authority; record a decision already taken, marked retrospective	FR-CAP-13, FR-CAP-14, FR-CAP-15, FR-CAP-16
Legacy harvesting	Historic approvals, minutes and mail archives imported once, flagged harvested, so precedent exists from day one	FR-CAP-09

Screens



S-04 Capture coverage view: The capture coverage view: which channels are live, what volume each carries, and what share of decisions arrive verbally. Prototype render on synthetic data for fictional entities.

RETURN ON INVESTMENT

Two measures carry this pillar at pilot, both baselined before anything is switched on and neither carrying a target in this document: **the share of decisions in a governed class that reach the register**, and **the time from a decision being taken to it being captured**. What capture makes possible is a count of decisions in a governed class rather than an estimate; policy evaluated at the moment of the decision rather than at the audit; the reason captured while it is still true; an agent's intent registered before it acts; and outcomes attached to decisions, because there is something to attach them to. Capture is the Connect phase of the adoption journey, and a 90-day pilot on one decision class ends with an honest answer to a question most organisations cannot currently answer: how many decisions of this kind do we make, who makes them, and how many did anyone ever write down.

Platform engines

Capture consumes the Red Bear Platform's Decision Intelligence engine (the decision record and candidate contract, RBP FR-DI-01), the Connectors service (RBP FR-CN-01 to RBP FR-CN-03), and the Privacy and aggregation service for consent and

redaction (RBP FR-PA-02). Platform behaviour is specified in the platform functional specification, not here.