RED BEAR: ENTERPRISE DECISION INTELLIGENCE · DATA SHEET · C · DECISION SUPPORT AND CONTEXT

The questions that surface what a decision really costs

Context, precedent and the four Cartesian questions in front of the decider at the moment of choice.

KEY POINTS

- The smallest cluster and the one that changes the quality of decisions rather than their governance; it can be delivered before any intervention is switched on
- Context is assembled by reference, not stored, and absence is shown as prominently as presence
- Precedent is retrieved on the decision's shape, not its text; a forecast states its confidence or is not shown
- A recommendation is labelled, never pre-selected, and never in the position of the decision; the requester writes the four Cartesian answers, not the product

What it is

Almost every discussion of a proposal asks one question, and only one: what will happen if we do this. The paper is written to answer it, the meeting is convened to test it, and the approval turns on it. Three other questions are available, and they are where the cost usually hides.

Decision support puts in front of the decider, at the moment of choice, the four things the organisation already knows and rarely has to hand: the data, documents and applicable policy from the connected systems; the similar decisions it has taken before, who made them, on what reasoning and how they turned out; the likely effect of each option, with the confidence that forecast deserves; and, above a threshold the accountable director sets, the four Cartesian questions that force the missing half of the argument into the open while the decision is still live.

The pain point

The context exists and the decider does not have it: the data, the contract and the policy sit in four systems the decider does not open, so decisions are made on what is to hand rather than what is known. The organisation has decided this before, last quarter, with a known outcome, and nobody knows, so precedent is rediscovered at full price and inconsistency is invisible. Nobody can say what is likely to happen, so junior deciders defer upward rather than risk being wrong.

And a decision that is never taken produces nothing at all: no transaction, no record, and therefore no line on any report, no variance to explain and nobody to ask about it. The opportunity that expired while the paper went round is invisible to every system the organisation owns. A proposal that looked marginal against a baseline of standing still often does not look marginal once standing still has been priced.

How it works

Context assembly. The class defines its own context standard: what must be known, what is useful, and what is known to be unavailable. At the moment of decision the product gathers the data, documents and applicable policy from the connected systems against that standard, holds references and a snapshot digest rather than copies, and persists exactly what the decider was shown, so audit replay can reconstruct the information state and not just the choice. What could not be found is shown as prominently as what was.

Precedent retrieval. Similar past decisions are found on shape, the class, the drivers, the materiality band, the actor's role, not on the words used to describe them: who made them, on what reasoning, how they turned out, with a link to the full record and never a summary alone. The same decision is not taken from scratch repeatedly because nobody could find the last four like it.

Impact prediction and options. The likely effect of a pending decision is forecast from precedent and context, always with its confidence and the count of precedents behind it; below the configured count the product shows the precedents and declines to forecast, because a forecast drawn from forty decisions deserves less weight than one drawn from four thousand. Options are set out with their expected impact and a recommended course, labelled as a recommendation, carrying the name of what produced it, never pre-selected and never in the position of the decision. A recommendation is not a decision until a person stops reading before sending it; the product is built so that they cannot.

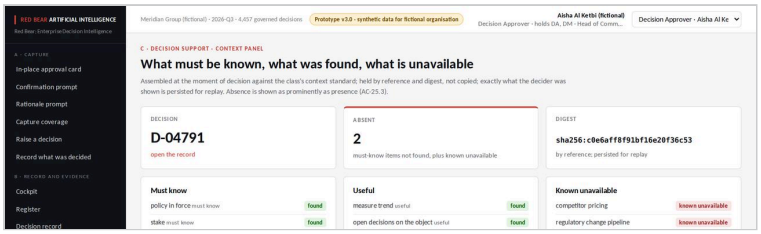
The four Cartesian questions. Above a risk threshold set by the accountable director, defaulting to the boundary of the significant class, the decider answers four questions before acting: *what happens if we do this; what does not happen if we do this; what happens if we do not; what does not happen if we do not*. Two concern acting and two concern not acting, and organisations are systematically good at the first pair and blind to the second. Laid out on two axes, act or not and happens or not, they form four quadrants, which is where the name comes from; the name is geometric and carries no attribution to Descartes. The requester writes the answers, not the product. Each answer is checked for relevance, and a thin one goes back to the requester with a note, without blocking. Each is checked for plausibility, and a doubtful one never goes back to the requester but to an available authorised person with the discrepancy shown side by side. The pattern across the four quadrants points to the lower-risk path and is read as a shape, never scored or totalled; the result is written to the decision record, before the

outcome is known, which is the only version of the reasoning a fair review can use. The depth of reasoning demanded scales with what the decision costs to get wrong: a routine decision is asked for none of it, a critical one for all of it plus the dissent.

What you get

Capability	What it does	Requirements
Context assembly	Data, documents and policy against the class's context standard; by reference; absence shown; what the decider saw persisted for replay	FR-SUP-01
Precedent retrieval	Similar past decisions found on shape, with decider, reasoning and outcome, linked to the full record	FR-SUP-02
Decision impact prediction	Expected effect per option with its confidence and precedent count; declines below the configured count; collapsed by default	FR-SUP-03
Options and recommendation	Options with expected impact; the recommendation labelled, attributed and never pre-selected	FR-SUP-04
The four questions at capture	Cartesian elements asked at the moment of commitment to the class's evidence standard; dissent and pre-mortem for the classes that require it	FR-CAP-10, FR-CAP-11, FR-CAP-12
The context standard	Must-know, useful and known-unavailable defined per class by the policy maker	FR-SUP-01

Screens



S-15 Context panel: The context panel: what the class says must be known, what was found, and what is known to be unavailable, on the record and on the approval card. Prototype render on synthetic data for fictional entities.

RETURN ON INVESTMENT

This cluster returns before governance does, which is why it can be sold first. **Better-informed decisions** is measured by the share of decisions with context viewed and evidence completeness at capture, both readable from Observe onwards. **Inaction is priced:** the cost of not deciding acquires an owner and a date instead of accruing to nobody. **Displacement is made visible:** what a commitment closes off is argued before it is committed, not discovered afterwards. **Fair review** becomes possible, because the reasoning is on the record as it stood before the result was known. One limit is worth stating plainly: the four-question examination is a discipline, not an automated feature. The product asks the questions, checks them and holds the answers, but it does not produce them. The quality of what is in the record is the quality of the argument the organisation was willing to have.

Platform engines

Decision support consumes the Research engine for context assembled from configured sources with citations and stated absence (RBP FR-RS-01, RBP FR-RS-02, RBP FR-RS-04), the Decision Learning engine for precedent and prediction (RBP FR-DL-02, RBP FR-DL-05, RBP FR-DL-06), the Decision Intelligence engine for options (RBP FR-DI-03), and the Cartesian Decision Analysis engine for the two axes and the quadrant read as a shape (RBP FR-CA-01, RBP FR-CA-02). Platform behaviour is specified in the platform functional specification, not here.