

# For your technical team. The contracts, not the internals.

Profonti is a single-tenant data repository your team operates, with citation and definition guarantees baked into the query path. This page describes what we promise — not how we make it work.

If you're evaluating Profonti as a CTO, architect, or DBA, this is the surface you need to understand. The deeper internals are intentionally not on the marketing site; we share them under engagement scope. Ask in Discovery and we'll go as deep as you need.



### Isolation by architecture.

Single-tenant by default — your Profonti repository runs in your Kubernetes namespace, on your database, in your cloud or colocation account. There is no shared control plane and no cross-customer data path; there is nothing to "go wrong" between customers because there is no between.



### Citation as a query contract.

Every answer Profonti returns carries the row set it was computed from, the query that produced it, and the timestamp of the underlying source data. This is a structural property of the query layer, not a post-hoc explanation step. If an answer can't be cited, it isn't returned.



### Determinism through definitions.

Business terms are versioned and attached to queries. Re-running the same question against the same data version yields the same answer. Definition changes are explicit, dated, and visible in every downstream artifact — digest, alert, slide, CSV.

## TECHNICAL CONCERNS VS PROFONTI'S POSTURE

A COMMON CONCERN	PROFONTI'S POSTURE
"Will my data sit alongside someone else's?"	<b>No. Each engagement is a separate cluster, separate database, separate namespace.</b>
"Will the AI hallucinate a number?"	<b>An answer without a citable row set is not returned. The query layer enforces it.</b>
"Can I read what your code actually does?"	<b>Yes — under the perpetual license, you have the source. Customers regularly review it during Discovery.</b>
"What happens if Profonti the company disappears?"	<b>You still have a running cluster, a database, and the source code. Your repository keeps working.</b>
"How do I observe what it's doing?"	<b>Standard Kubernetes telemetry — metrics, logs, traces — into your observability stack of choice.</b>
"What about compliance?"	<b>The data plane is yours. Apply the controls that apply to your environment; Profonti doesn't add a separate compliance surface.</b>