Replatform

Cloud Cost X-Ray — Sample Deliverable

Representative output shape. Numbers are illustrative; actual levers, magnitudes, and plans depend entirely on your environment.

This document shows the **structure** of every X-Ray we deliver — the 12 sections, the format of each, what you can expect to find on each page. The numbers below come from one anonymized engagement (India consumer commerce platform, BigQuery estate). They illustrate the **deliverable format**, not a forecast of your savings. The X-Ray's promise is structural, not numeric.

Page 1 — Executive summary

Every X-Ray opens with a one-page summary of: the warehouse covered, the engagement window, the count and aggregate magnitude of waste sources identified, and the recommended sequence of action. The format is constant across engagements; the specific numbers in your summary will reflect what the rule library found in your environment.

Illustrative example *(one engagement, anonymized — not a prediction of your outcome)*:

- Engagement: BigQuery cost diagnostic
- Window: 5 business days
- Annual warehouse spend at engagement start: in the low-six-figure USD range
- Annual waste identified: significant fraction of spend, distributed across slot reservations, model fan-out, and unread objects
- Recommended action: 30/60/90 reduction plan, no data model rewrite required

Your summary will look structurally identical with different specifics.

Page 2 — Methodology + scope

What we do

Read-only ingestion of:

- Query history (last 90 days)
- Slot reservation history + autoscaling logs (BigQuery / Snowflake / Databricks equivalent)

- IAM role + project / account structure
- dbt project (production manifest.json + catalog.json) if present
- Table access patterns + INFORMATION_SCHEMA equivalents

What we don't do

- Execute any queries against production tables
- Modify any infrastructure
- Access PII or business records
- Take any data offsite

Tools

- Codified rule library (proprietary; engineering details under NDA)
- Deterministic validation framework (reproducible against your environment after handover)

Methodology is constant across X-Rays. Per-engagement variation is in which rules fire, not in how they fire.

Page 3 — Top-N waste sources (named, ranked, owner-assignable)

Format you receive: ranked table of the waste levers found in your environment, named, with annual impact in your currency, assignable owner, and engineering effort estimate.

Illustrative example (one engagement):

#	Lever	Annual impact	Owner	Effort (eng-weeks)
1	Flat-rate slot reservations sized for peak that recurred 2% of the month → drop to autoscaling for off-peak hours	(mid-five-figure USD)	Data Platform Lead	0.5
2	dbt models re-running unchanged source data daily despite incremental materialization being viable	(mid-five-figure USD)	Analytics Eng	2
3	31% of warehouse tables not read in 90+ days → expire / archive	(low-five-figure USD)	Data Eng	1
4	dbt model fan-out — 6 downstream models multiplied 1 source-table scan into 14 effective scans	(low-five-figure USD)	Analytics Eng	1.5
5	Wide SELECT * patterns on partitioned tables ignoring partition pruning	(four-figure USD)	Data Eng (training)	0.5

Your top-N table will surface different levers in different quantities. The lever shapes we see most often: slot/warehouse rightsizing, dbt fan-out, unread objects, partition-pruning misses, materialization-strategy mismatches. Which of these dominate in your environment is what the X-Ray tells you.

Page 4 — Slot / warehouse rightsizing model

Format you receive: sensitivity table showing the cost shape at three commit positions against three workload assumptions, with the recommended commit position highlighted.

Illustrative example *(one engagement)*:

Commit shape	Current workload	+20% workload	+50% workload
1000 flat-rate slots, 1-yr commit	(baseline)	(over-committed)	(commit + on-demand top)
500 flat-rate slots + autoscale to 1500 on-demand peaks	**(recommended)**	(still in band)	(cross-over)
100% on-demand	(volatile high)	(higher)	(highest)

Your model will use your actual current commit shape, your actual workload, and your warehouse's pricing curve. The recommended position is environment-dependent.

Pages 5–6 — Highest-cost query patterns (with rewrites)

Format you receive: the five most expensive query patterns in your environment, each with: the pattern, why it's expensive in your warehouse's pricing model, the rewrite, and the projected slot-hour / DBU / compute-credit reduction. Verbatim SQL pre/post.

Pattern shapes we commonly see: implicit cross-joins on de-normalized tables, partition-pruning misses, repeated CTE materialization that should be table-materialized once, window functions over un-bounded partitions, recursive CTEs that should be iterative.

Which patterns dominate in your environment is the value of the X-Ray. We don't pre-guess.

Pages 7–8 — dbt model fan-out audit

(If you don't run dbt, this section is replaced with the equivalent for your transformation layer — Dataform, SQLMesh, hand-rolled stored procedures, etc.)

Format you receive: dependency-graph view of your transformation project highlighting:

- **Models nobody queries downstream** (downstream node count = 0, >30d unqueried) → candidates for retirement
- **Fan-out hotspots** (one source-table → N downstream models, each re-running unchanged data)
- **Models without incremental materialization** that should have it
- **Models with incorrect materialization** (incremental that should be view, table that should be incremental)

Output: prioritized list of model-level interventions ranked by warehouse-credit savings.

Page 9 — 30/60/90-day execution plan

Format you receive: time-phased plan separating quick wins (week 1, your team) from architectural changes (quarter 2+, your team or follow-on Execution).

Illustrative example *(one engagement)*:

Days 0–30 (quick wins, your team)

1. Switch slot commit shape

2. Expire / archive unread tables
3. Add partition pruning to identified SELECT * patterns

Days 30–60 (architectural)

1. Convert fan-out dbt models to incremental materialization
2. Refactor most expensive query patterns into materialized views
3. Implement table-access logging for ongoing unread-table identification

Days 60–90 (monitoring + governance)

1. Slot-utilization alerting
2. Per-model cost dashboard
3. Quarterly unread-table review cadence

Your plan will sequence to your team's roadmap, your tooling, and your risk tolerance.

Page 10 — Monitoring blueprint

Format you receive: specific alert thresholds and dashboard recommendations for the levers ranked in page 3, sequenced so that the cost from regressing post-engagement is structurally caught within hours, not quarters.

Tooling-agnostic: we adapt the blueprint to your stack (Grafana / Looker / dbt Cloud / Datadog / vendor-native dashboards / etc.). The structural shape — what to alert on, at what threshold, with what review cadence — stays constant.

Page 11 — What we did NOT find (transparency)

Mandatory section. We tell you what we audited and concluded was already tuned — so you don't spend further effort there.

Categories we commonly examine: IAM / role structure, cross-region transfer, dataset lifecycle policies, reserved-capacity contract terms, dataset-level partition strategy, query-tag hygiene, BI tool query patterns.

Telling you what's already right is part of the deliverable. It builds trust and saves your team time.

Page 12 — Engagement summary + credit terms

Format you receive:

- **Fee paid:** $100 (intro pricing, first 10 customers) or $4,950 (standard)
- **Engagement window:** [dates]
- **Deck delivered:** [date]
- **Recorded walkthrough:** [Loom / Drive link]
- **Rule library extract** (rules used in this engagement): [Notion / GitHub link, read-only access]
- **Credit terms:** - First-10 intro ($100): credited 100% against the first invoice of any Execution engagement signed within 60 days; permission requested to publish the anonymized findings shape (categories of waste + structural pattern only — no client identity, no data, no specifics) - Standard ($4,950): credited 100% against the first invoice of any Execution engagement signed within 60 days - If no Execution engagement: the deck is yours; no follow-on ask

Contact for follow-on: Yash Maheshwari, Replatform — yash@replatform.in / cal.com/replatform

---

The structural promise

What stays constant across every X-Ray:

- **12 sections in this order** (executive summary → methodology → top-N levers → rightsizing model → query patterns → transformation-layer audit → 30/60/90 plan → monitoring → transparency → engagement summary)
- **Rule library + deterministic validation framework** is the engine
- **Read-only metadata access only** — never production data
- **5 business days from credentials to deck**
- **Recorded 90-min walkthrough on day 5**

What varies entirely:

- **Which levers fire** in your environment
- **Magnitude of each lever** in your currency
- **Recommended plan sequence** for your team
- **Architectural change opportunities** specific to your stack

The X-Ray's value is in the matching of your environment to the rule library — the rule library is constant; the match is unique.

Document prepared by Replatform. Sample format only. Any actual X-Ray deliverable is confidential to the engaged client.