

Request for Proposals (RFP): Reverse-Engineer Undocumented PL/SQL, PowerBuilder & Oracle Forms Codebases into Specs

Instructions for RFP Author: This template is structured so your IT, Architecture, and Procurement teams can issue a rigorous, apples-to-apples Request for Proposals. Customize the bracketed [Modifiable] fields below or use the companion .md file on the solution portal.

1. Issuing Organization & Procurement Schedule (Modifiable Fields)

RFP Field / Milestone	Modifiable Entry / Target Value
Issuing Organization & Sponsor	[Insert Legal Organization Name, Executive Sponsor & Primary Contact Email]
Project Scope Summary	Transition from Undocumented Legacy Codebases (PL/SQL, PowerBuilder, Forms, Pro*C, T-SQL, RPG/COBOL) to AST Dependency Graphs, Business-Rule Catalogs, OpenAPI Specs & Target Code
RFP Issue / Q&A; / Proposal Due Dates	[Issue: YYYY-MM-DD] [Vendor Questions Due: +14 Days] [Proposals Due: +30 Days]
Target Contract Type	Fixed-Price Milestone Deliverables with Automated Parity & Cutover Acceptance Gates

2. Mandatory Technical Scope of Work (Modifiable Requirements)

- [REQ-ARCH-01] Stage 1 - Deterministic Compiler AST & Call-Graph Construction: We parse your complete source repository (`.pbl`/`.srw`/`.srd`, .fmb`/`.pll`/`.rdf`, DBA_SOURCE` PL/SQL, T-SQL, Pro*C, ColdFusion, C#) into a deterministic graph database mapping every UI control -> event trigger -> procedure call -> table/column CRUD operation.`
- [REQ-ARCH-02] Stage 2 - Runtime Telemetry & Dead-Code Pruning: By correlating the static AST call graph with database audit/AWR execution counts (`V$SQL` / `DBA_HIST_SQLSTAT` / table modification timestamps`), we isolate the 30% to 45% of legacy objects that are completely dead.
- [REQ-ARCH-03] Stage 3 - AST-Grounded AI Specification Synthesis: Each topologically sorted call subgraph is fed into our private OCI/Anthropic/OpenAI inference pipeline to produce human-readable Functional Requirements Documents (BRDs), state machines, and edge-case tables where every bullet cites the exact legacy file and line range.
- [REQ-ARCH-04] Stage 4 - Automated Target Scaffolding & Test Generation: From the verified AST and specification, we generate target Oracle APEX 24.2 blueprints, PL/SQL package stubs, OpenAPI 3.1 contracts, and `utPLSQL` / Playwright` regression test suites.
- [REQ-ARCH-05] Query Fan-Out Rationalization & Runtime/Inference Filtering: Instead of migrating cloned reports and DataWindows 1-for-1, our AST SQL parser clusters queries by their canonical core JOIN graph, builds governed Oracle 23ai views only for those core queries, and generates dynamic WHERE clauses, faceted filters, and ad-hoc report variations at runtime and via AI inference (APEX 24.2 Smart Filters & Select AI).

3. Component Conversion & Acceptance Matrix

Legacy Artifact / Workload	Required Target Architecture	Engineering Method & Acceptance Gate
Undocumented 5,000-Line PL/SQL or T-SQL Packages	Line-Cited Functional Specification + Modular Package Design	AST Control-Flow Graph + LLM Summarization
Spaghetti UI Triggers (PowerBuilder / Oracle Forms)	Extracted Business-Rule Catalog + State Transition Matrix	UI-vs-Domain Logic Separation Classifier
Orphaned Tables, Views & Uncalled Procedures	Verified Dead-Code Decommission Manifest	Static Call-Graph + <code>DBA_HIST_SQLSTAT`</code> Join

Implicit Foreign Keys (Enforced Only in App Code)	Explicit DDL Constraints & Interactive ERD Lineage Diagram	AST Join-Predicate Mining
Cloned Reports / Queries Differing Only by WHERE Clause or Group-By (Query Fan-Out)	Canonical Core SQL View + APEX 24.2 Runtime / AI Inference Filters	AST Join-Graph Clustering (75-90% Report Count Reduction)

4. Mandatory Vendor Q&A; Questionnaire & Evaluation Benchmark Key

Bidders must answer each technical question below. Evaluation committee members can compare vendor answers directly against the Reference Benchmark Key:

Vendor Question 1: How is Legacy Code X-Ray different from simply using GitHub Copilot or ChatGPT on our source files? [Score: ____ / 10]

Reference Evaluation Benchmark: LLMs alone cannot resolve cross-file inheritance, global variables, dynamic SQL, or database trigger chains across 2,000 files. We use real compiler parsers (including Oracle `PL/Scope` and our PBL/FMB AST engines) to build the exact mathematical dependency graph first, so the AI never guesses or hallucinates relationships.

Vendor Question 2: Does our source code ever train public AI models? [Score: ____ / 10]

Reference Evaluation Benchmark: Never. All AST parsing runs locally inside isolated enclaves, and any semantic summarization uses zero-data-retention enterprise API endpoints.

Vendor Question 3: What is included in the downloadable PDF Solution Brief? [Score: ____ / 10]

Reference Evaluation Benchmark: The PDF Solution Brief includes a sample Legacy Code X-Ray Diagnostic Output (showing an AST dependency map, extracted business-rule table with line citations, and dead-code scorecard) plus instructions for requesting a 2-week Phase 0 Codebase Assessment.

Vendor Question 4: How do you rationalize report/query fan-out and guarantee 100% customer code & data ownership? [Score: ____ / 10]

Reference Evaluation Benchmark: Vendor must demonstrate AST SQL clustering to collapse cloned queries into governed core views with runtime/AI-inference filters, and deliver 100% customer-owned SQL, PL/SQL, APEX, or cloud code with zero proprietary runtime lock-in.

5. Commercial Milestone Pricing & Evaluation Weighting (Modifiable)

Phase / Deliverable Gate	Required Acceptance Output	Bidder Fixed Fee	Eval Weight
Phase 0: AST Discovery & Blueprint	Complete code/schema AST inventory, dead-code manifest & core SQL fan-out map	\$ [Insert]	20%
Phase 1: Core Schema & Logic Uplift	Target cloud/DB provisioning, automated DDL/code conversion & unit tests	\$ [Insert]	30%
Phase 2: UI, Reports & UAT	Modernized UI/portals, runtime/AOP reports, security VPD/SSO & UAT sign-off	\$ [Insert]	30%
Phase 3: Cutover & Hypercare	Zero-downtime cutover, row/hash reconciliation & 30-day production warranty	\$ [Insert]	20%