

VNA-SB-002 | Application Stack X to Application Stack Y | Client/Server Desktop to Google Cloud Native | Redwood Stone

PowerBuilder to Google Cloud Migration (Cloud Run, React & AlloyDB)

When your enterprise cloud strategy standardizes on Google Cloud Platform (GCP), legacy Windows-only PowerBuilder apps block data center exit. We use our automated PowerBuilder AST reverse-engineering engine to extract all DataWindow schemas, SQL queries, and PowerScript business logic, generating containerized Cloud Run services, React/TypeScript frontends, and AlloyDB/BigQuery data pipelines.

Search Queries & Scope Covered: migrate PowerBuilder to Google Cloud | convert PowerBuilder to React and Cloud Run | PowerBuilder DataWindow to REST API | PowerBuilder Sybase to Google AlloyDB migration | reverse engineer PowerBuilder application to cloud

SOURCE LEGACY STACK (STACK X)		MODERNIZED TARGET ARCHITECTURE (STACK Y)	
PowerBuilder (.pbl, DataWindows, PowerScript) + Sybase / SQL Server / Oracle		Google Cloud Run, React/TypeScript, AlloyDB / Cloud SQL & BigQuery	
AST LOGIC RECOVERY	TARGET COMPUTE TIER	TRANSACTIONAL DATABASE	ANALYTICS OFFLOAD
100% PBL Coverage	Serverless Cloud Run	AlloyDB / Cloud SQL	BigQuery + Looker

1. Problem Space & Architectural Bottlenecks

- Organizations standardizing on Google Cloud Platform often discover dozens of departmental or core operational PowerBuilder applications tied to on-premises Sybase ASE, SQL Server, or legacy Oracle instances.
- Lifting and shifting Windows fat-client binaries into Google Compute Engine VMs with remote desktop streaming preserves every maintenance bottleneck while inflating cloud compute bills.
- Engineering teams proficient in TypeScript, Python, and Go lack the domain tooling to decipher binary PowerBuilder .pbl libraries, inherited DataWindow buffers (Primary!, Delete!, Filter!), and implicit transaction commits.

2. Technical Solution & Reference Architecture

- 2.1 Automated PBL Decompilation & Specification Blueprinting: Our reverse-engineering engine parses every .pbl library, producing OpenAPI 3.1 specifications, Entity-Relationship models, and human-readable business-rule specifications from legacy PowerScript.
- 2.2 DataWindow Buffer-to-REST/GraphQL Translation: PowerBuilder's stateful Primary!, Filter!, and Delete! DataWindow buffers are translated into stateless, optimistic-concurrency REST endpoints running on Google Cloud Run (Python FastAPI / Node / Go) backed by Cloud Tasks and Pub/Sub.
- 2.3 Database Modernization to AlloyDB or Oracle@Google Cloud: Underlying Sybase, SQL Server, or legacy Oracle schemas are migrated via Google Database Migration Service (DMS) or GoldenGate into AlloyDB for PostgreSQL or Oracle Database@Google Cloud.
- 2.4 Reporting DataWindows to BigQuery & Looker: Heavy reporting and composite DataWindows are offloaded from transactional tables into BigQuery medallion tables with scheduled queries and Looker Studio dashboards.

2.5 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. Artifact-by-Artifact Conversion & Modernization Matrix

Legacy Artifact / Baseline Workload	Modernized Oracle / Cloud Target	Engineering Method & Outcome
PowerBuilder DataWindow (.srd) CRUD & Buffers	React TanStack Table / AG Grid + FastAPI on Cloud Run	OpenAPI 3.1 Schema & JSON DTO Generation
PowerScript Non-Visual Objects (NVO)	Containerized Cloud Run Microservices (Python / TypeScript)	AST Business Rule Extraction + Pytest Suite
Sybase ASE / SQL Server Stored Procedures	AlloyDB for PostgreSQL PL/pgSQL or Oracle@GCP	Automated Dialect Transpilation & DMS Cutover
Composite Reporting DataWindows	BigQuery Partitioned Views + Looker / PDF Cloud Functions	SQL Pushdown & Serverless PDF Rendering
INI Files & Windows Registry Config	GCP Secret Manager + Cloud Identity / IAP SSO	Zero-Trust IAM & OAuth2 Token Wiring
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. Oracle APEX & Ancillary Engineering Toolchain

- Viscosity PBL/SRD AST Reverse-Engineering Parser & Blueprint Generator
- Google Cloud Run, Cloud Build, Artifact Registry, and Identity-Aware Proxy (IAP)
- AlloyDB for PostgreSQL, Oracle Database@Google Cloud, and BigQuery
- Oracle GoldenGate / Google Database Migration Service (DMS) for zero-downtime cutover
- Vertex AI & Gemini Code Assist for automated unit-test synthesis

Anonymized Production Proof: For a regional healthcare and diagnostics network migrating off on-premises Windows servers to Google Cloud, we parsed 180+ PowerBuilder modules and 600+ DataWindows, converting the transactional tier to Cloud Run + AlloyDB and offloading 90+ operational reports to BigQuery in 14 weeks.

5. Principal Architect Q&A

Q: Can we run Oracle Database natively inside Google Cloud if we want to keep our PL/SQL packages?

A: Yes. With Oracle Database@Google Cloud (Exadata Database Service and Autonomous Database running physically inside GCP data centers), you can pair Google Cloud Run / Vertex AI directly with low-latency Oracle databases or Oracle APEX.

Q: How do you validate that the new Google Cloud application matches legacy PowerBuilder calculations?

A: Our pipeline captures real transaction payloads and DataWindow result sets from the legacy app and replays them against the new Cloud Run API endpoints to prove 100% numerical parity before cutover.

Q: What is included in the downloadable PDF Solution Brief?

A: The PDF Solution Brief details the 5-stage PowerBuilder-to-GCP AST extraction pipeline, the DataWindow-to-React/Cloud-Run reference architecture, and a decision matrix comparing AlloyDB vs. Oracle Database@Google Cloud.