

Cracking the SAS Legacy Lock-In: The Exponentia Partnership

Partner: exponentia.ai	Asset Type: AE Battlecard	Target Outcome: Unlocking SAS Modernization Pipeline
---------------------------	------------------------------	---

1 The Opportunity

Unlocking the SAS Legacy Tax



\$700K – \$1M
Trapped Spend

This trapped annual spend, compounded by 8-12% YoY licensing escalation, is the primary barrier to funding new Databricks AI initiatives.



~38%
TCO Reduction

Projected 3-Year Total Cost of Ownership Reduction

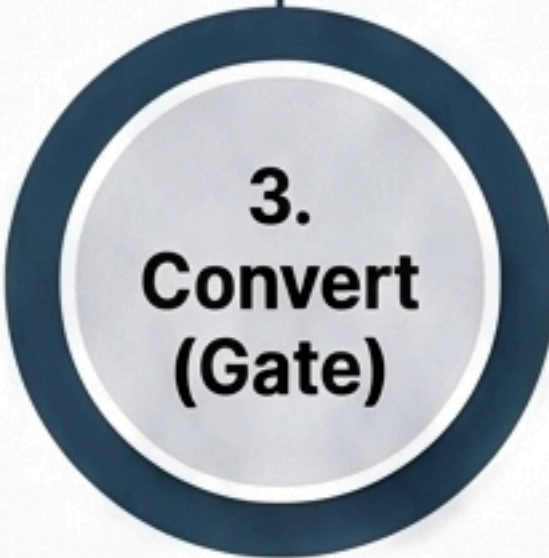
Converting legacy spend into Databricks consumption drives GenAI experimentation, real-time analytics, and modern governance.

3 The Engine

SAS to PySpark/SQL Automation

70–90%
Automated Translation

The MigrationSponent Engine uses a mechanized pipeline built specifically for regulated environments.



Process Flow: The mechanized steps of the MigrationXponent Engine

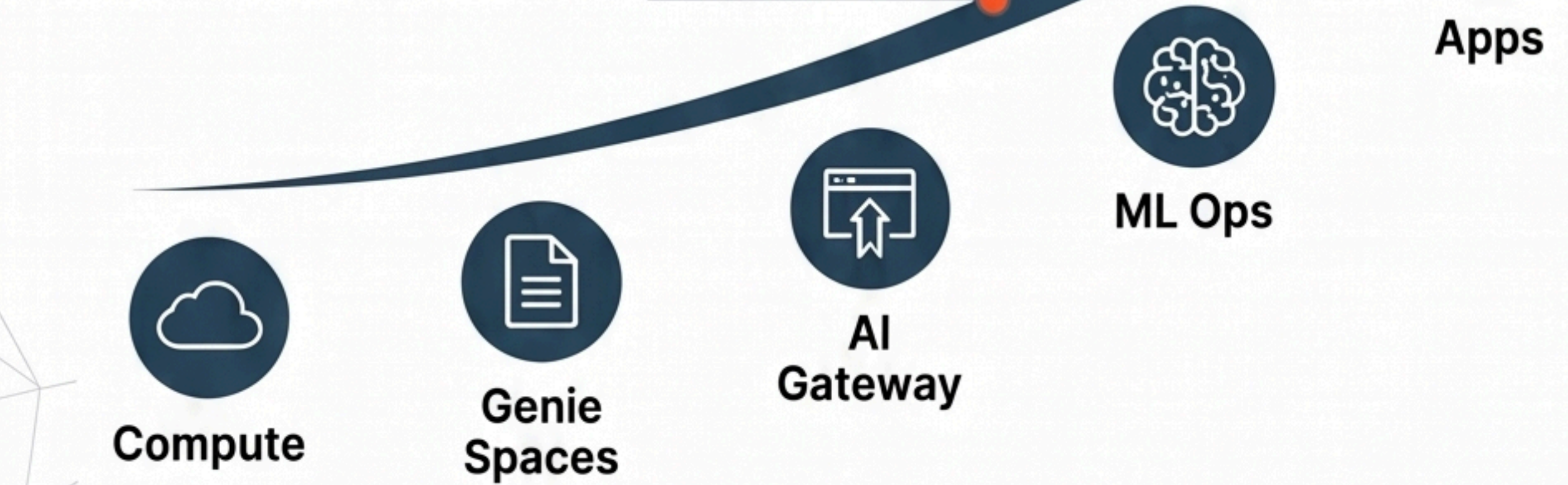
4 The AE Play

Start AI Activation in Parallel



The 6-Week AI Advisory Sprint

Don't wait for migration to finish; run advisory, use-case prioritization, and capability incubation immediately to drive consumption.



The Multiplier Effect
Accelerate Immediate Platform Consumption

De-risk execution to land data natively on Databricks, opening the path for Compute, Genie Spaces, AI Gateway, ML Ops, and Apps.



2 Unblocking Deals

Eliminate Migration Objections Systematically

Solving the 4 Structural Traps

Deals often stall due to executive fear of semantic data corruption or operational discontinuity; we replace theoretical confidence with mathematical proof.

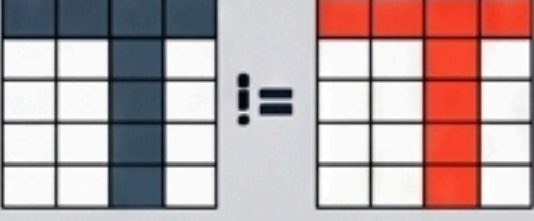
Trap 1: Non-deterministic coda behavior



Fix: Sort-order validation macros

Prevents unreliable outputs and uiigration risk by ensuring consistent execution results.

Trap 2: SAS MERGE != SQL JOIN



Fix: Custom DataFrame merge helper

Preserves exact 1:1, 1:M, and M:1 semontise to prevent semantic data corruption.

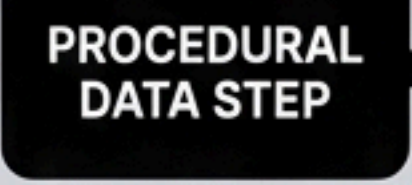
Trap 2: Missing values vs. ANSI NULL



Fix: Spark <=> operator and null-handling expressions

Eliminates risk of estaorial or regulatory failure by preserving the exact intent of legacy data handling.

Trap 4: Complex procedural Data Steps



Fix: Python UDTFs (User Defined Table Functions)

Translates black-box logic into debuggable, line-by-line code.