RelationalAI Snowflake Native App

Architecture White Paper

Introduction

RelationalAI is on a mission to power every decision with intelligence. As the industry's first knowledge graph coprocessor for Snowflake's AI Data cloud, RelationalAI enables organizations to apply rule-based reasoning, advanced graph analytics, predictive analytics, and prescriptive analytics to their data, where it already lives, simplifying computing and powering better decisions. Learn more at Relational.ai.
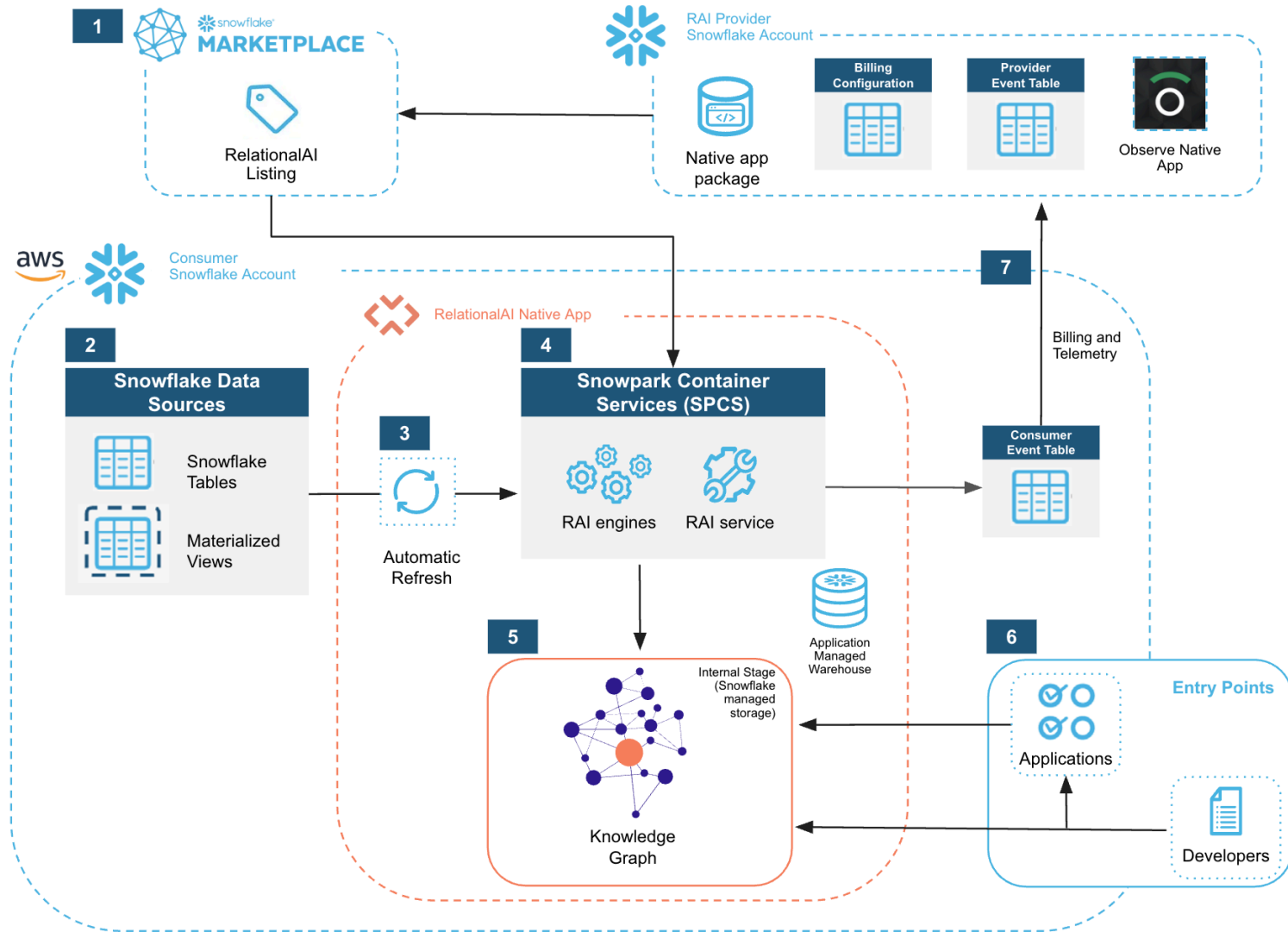
Why use RelationalAI in Snowflake?

Built on the Snowflake Native App Framework Integration with Snowpark Container Services (now generally available on AWS and public preview on Azure), RelationalAI enables customers to build and modernize intelligent applications using a data-centric architecture based on relational knowledge graphs.

As a Snowflake platform extension, RelationalAI augments Snowflake's AI capabilities to help enterprises make better, faster decisions. This makes it possible for customers to combine generative AI models with compound AI techniques for rule-based reasoning, graph analytics, prescriptive analytics, and predictive analytics in Snowflake. Customers' data stays within the security and governance perimeter of their Snowflake account, eliminating the need for manual data synchronization. Unlike legacy solutions that offer limited support for these workloads, RelationalAI is cloud-native and provides expressive query processing within Snowflake while maintaining the advantages of versioning, zero-copy cloning, scale, and consumption pricing.

With RelationalAI, customers can find a simplified, faster, and cost-effective path to advanced enterprise AI capabilities.

Architecture

1. The RelationalAI Native Application package is available via the Snowflake Marketplace, for accounts hosted on AWS (visit the RelationalAI Native App listing in the Snowflake Marketplace for the latest list of supported regions).
2. A prerequisite for using RelationalAI for the purpose of building a knowledge graph is to have data available that is either in Snowflake Tables or Views.
3. A knowledge graph exists within the RelationalAI Native app within the Snowflake internal, secure stage that is kept up to date via an automated continuous refresh process.
   - On the warehouse: developers used to execute queries against the RelationalAI Native App's Snowflake database. A dedicated warehouse is necessary to ensure that service operations against this database do not contend with other workloads. This warehouse is created automatically when the RelationalAI application is activated, and suspended automatically when the app is deactivated.
4. Developers can interact with the knowledge graph either via
   - Cloud notebooks such as Snowflake notebooks, etc.
   - Use a local installation of the RelationalAI Python package in combination with an IDE such as VSCode or Jupyter notebook
5. The RelationalAI Service is the core component of the RelationalAI Native App for Snowflake. It runs in a container on Snowpark Container Services and manages the RelationalAI engines that execute queries written with the RelationalAI Python package.
6. Query results can either be returned directly to the caller or be stored in Snowflake tables. Python functions can be exported as stored procedures, to be called from Snowflake SQL, either on demand or scheduled, using Snowflake Tasks.
7. Billing and telemetry
   - When you install the RelationalAI Native App, you are prompted to share continuous telemetry with RelationalAI. The continuous telemetry is written to your Snowflake account's active event table and contains operational information such as internal system logs, engine sizes, and usage data. Customer data and personally identifiable information (PII) are not included in continuous telemetry data.

To further explore RelationalAI and its full capabilities, visit our technical docs. There, you'll find comprehensive resources to guide you in building, deploying, and operating intelligent applications within the Snowflake AI Data Cloud.