Deliver Secure SQL Access for Enterprise APIs

Dipak Patel, Principal Product Manager and Co-founder of OpenAccess, Progress
Dennis Bennett, Principal Sales Engineer, Progress
Audio Bridge Options & Question Submission
Deliver Secure SQL Access for Enterprise APIs

Dipak Patel, Principal Product Manager and Co-founder of OpenAccess, Progress
Dennis Bennett, Principal Sales Engineer, Progress
Agenda

- Background and Use cases for SQL Access
- How to deliver SQL Access for REST APIs?
- Demo
- Best Practices
How Have BI Solutions Evolved?

1st wave – 1996
Technical BI

IT  User

2nd wave – 2008
Self Service BI

Analyst  User

3rd wave
End-User BI

© 2017 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
SQL Is a Common Approach
SQL Is a Common Approach

- .NET
- Python
- PHP

Enterprise API Layer / Data Access Layer

- ODBC
- SSIS

Authentication
- Access Policy
- Data Encryption
- Data Masking
- Audit Policy
- Data Lineage
- Replication

Flat Data: XML, CSV, Excel, Log, PDF

Big Data: NoSQL

Operational Databases

Data Warehouses

Real-time / Batch APIs
How Do We Get Back to 3rd Wave?

1st wave – 1996
Technical BI

IT
User

2nd wave – 2008
Self Service BI

Analyst
User

3rd wave
End-User BI

Gartner
OpenAccess Gets You Back to 3rd Wave
Survey Question 1: Which Enterprise BI tools are you using in your organization today?

- Oracle Business Intelligence (OBIEE)
- Microsoft Business Intelligence (SSIS, SSAS, SSRS)
- SAP Business Objects
- IBM Cognos
- Other (Please share through comments)
Survey Question 2: What solutions are you leveraging to develop your Enterprise API layer?

- CA API Management
- IBM API Management
- Oracle API Manager
- Mulesoft
- Other (Please share through comments)
SQL Standards remain popular for access data

API Landscape

Source: 2017 Data Connectivity Outlook Survey
Other popular use cases we see need for virtual SQL access

- **Application / Business Logic**
  - Business Logic Layer: CRM, Finance and other applications are integrated

- **Abstraction**
  - Expose single interface distributed across large objects in NoSQL database vs transactional records stored in SQL database

- **Support Multi-Tenancy**
  - Enterprises offer multi-tenant architecture
  - Our customers are using our SDK to enforce tenant level security at the driver level without touching the hosted architecture.
Survey Question 3: Which of the following do you need SQL Access to?

- Enterprise API Layer / Data Access Layer
- Business Logic Layer
- Single/Multiple Data Stores
- Multi-Tenant Architecture
- Other (Please share through comments)
Agenda

- Background and Use cases for SQL Access
- How to deliver SQL Access for REST APIs?
- Demo
- Best Practices
Customized SQL Connectivity

It is the same code whether you are supporting ODBC, OLE DB, JDBC or .NET.

OpenAccess SDK provides the majority of the code needed to SQL enable a data source. You generate the small amount of code (IP) residing between the SQL engine and your data store. This code processes the results generated by the SQL engine.

The IP can be written in C, C++, Java or .NET.
Steps to Implement a Driver

**Design**
- Define the view of API data – schema
- Determine how API will interact with SQL engine – row-based or query-based

**Setup**
- Determine features you will support – CRUD?
- Implement the IP API functions by starting with template or one of our examples

**Test**
- Test sample queries through provided ODBCISQL application
- Package for redistribution
OpenAccess can plug in to your enterprise security

Oracle Access Manager

Entrust Get Access

RSA Access Manager

CA SiteMinder

CA API

OAM API

SAML PDP

RSA API

Unsecured Traffic

Authenticated Traffic

API Gateway

Image source: https://docs.oracle.com/cd/E55956_01/doc.11123/administrator_guide/content/admin_existing.html
Translating between SQL and REST

OpenAccess SDK SQL Engine

ipExecute

IP CODE

{JSON}

REST Call

© 2017 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.
SQL Support

Broad SQL92 and SQL99 support
- DDL – Create / Alter / Drop Tables, Views and Indexes
- DML – Select, Insert, Update, Delete
- Stored procedures – Call
- Joins, unions, nested queries
- Order By, Group By, Scalar Functions

Join options
- Push down to the server
- SQL Engine performs Joins
- Multiple Join Order options
A large Financial Company is using OpenAccess for ODBC, JDBC Access

- Technology Needs:
  - BI team needed access to the data access layer from Microstrategy and IBM Cognos.
  - New applications need to be authorized by internal authentication and security layer.

- Challenge:
  - They developed an in-house custom JDBC driver that wasn’t performing well. The driver had compatibility issues and maintenance was very expensive.
  - Now, they also need an ODBC driver to support a new BI tool within a strict timeline.

- Solution:
  - With OpenAccess, they could deploy both an ODBC and JDBC driver in one go.
  - OpenAccess worked seamlessly with all of their 3rd party authentication and security tools.
  - Their alternative was to find an ODBC expert, develop the ODBC driver from scratch
Agenda

▪ Background and Use cases for SQL Access
▪ How to deliver SQL Access for REST APIs?
▪ Demo
▪ Best Practices
Agenda

- Background and Use cases for SQL Access
- How to deliver SQL Access for REST APIs?
- Demo
- Best Practices
Best Practices: Support Multiple APIs – ODBC, JDBC, ADO.NET

- Some BI/Data Integration tools are Java and use JDBC, others are Native and use ODBC
- Having to implement and test single interface layer saves in cost of ownership
- Compatibility with enterprise BI and Data Integration tools – not all drivers are created equal
Best Practices: Authentication

- Where to include code to integrate with IAM provider
- How to handle OAuth flows and UI?
- Authentication happens at execution time with REST APIs
Best Practices: How to create data model from APIs

- How to handle dynamic vs static metadata in API
- Map API and fields to relational schema
- Handling metadata for semi structured JSON responses
Best Practices: Optimize Query Performance

- Design enterprise API to allow search push-down, paging results, meta-data access
- Push down filtering and other operations, possibly joins
- Perform iterative design – simple to more complex
- Implement rich SQL support because BI and ETL tools can generate very complex SQL queries
Next Steps

1. **Schedule a meeting with our Product team** [www.progress.com/company/contact]

2. Or Learn More:
   - [OpenAccess Overview Guide](#)
   - [Download 30-day trial of OpenAccess](#)
   - [Tutorial to Build a Custom Driver over a REST API under 2 hours](#)