API Master Class

Knowing Your Verint API Options

Matthew Monahan, Sr. Director, Data Strategy
Jonathan Sant, Principal System Integrator, Advanced Solutions
Shannon Garrett, Solution Delivery Architect, Advanced Solutions
Ryan de Leon, Sr. Manager, Advanced Solutions

SEPTEMBER 9, 2025 | ORLANDO, FLORIDA

VERINT®

VERINT®

Session Goals

- Understand Integration Studio
- Understand API Authentication
- Gain Practical Experience with Verint APIs
- EDH DLE
- Best Practices
- Answer Questions

Agenda

1. Integration Studio
2. Authentication In Depth
3. Hands On Exercises with WFE APIs
4. EDH DLE data retrieval and processing
5. Troubleshooting and Best Practices

VERINT® IS THE CX AUTOMATION LEADER

Breaks

2 x 15 mins

Wireless

SSID: VerintEngage

PW: Engage25

Section 1: Integration Studio

Matthew Monahan

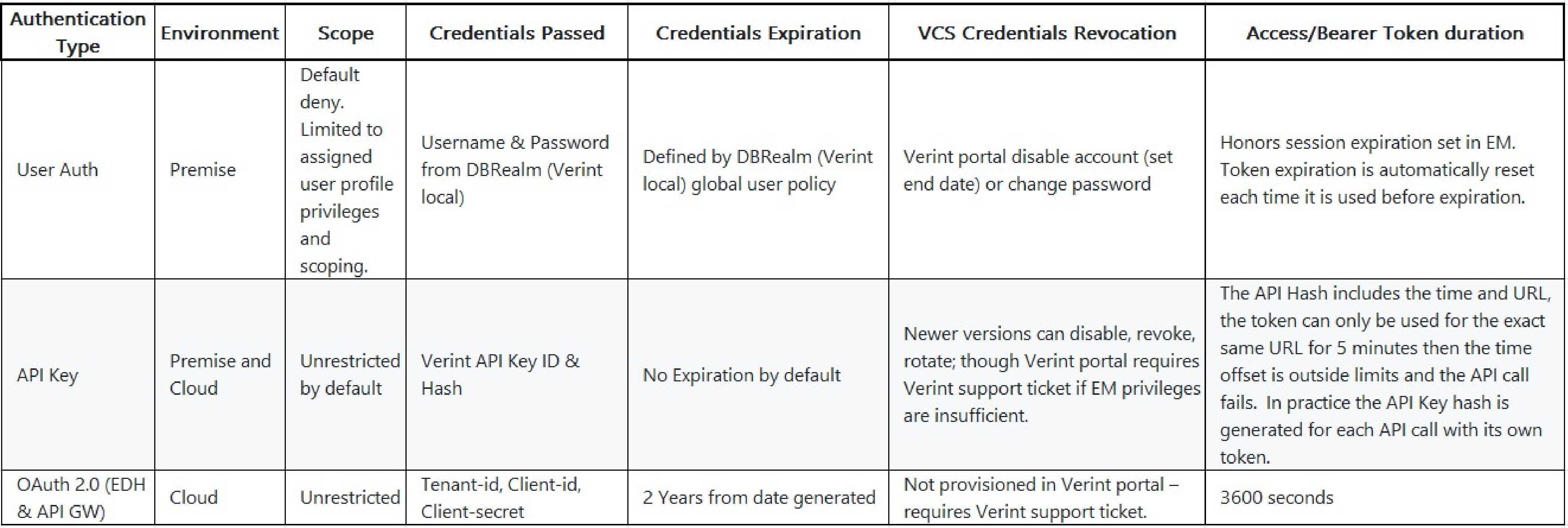
Section 2: Authentication In Depth

Jon Sant

Authentication - Overview

- History
  - DB Realm – Username and password, scope tied to user account
  - API Key (SWT) - Enhanced security, stateless authentication, token-based auth
  - Key Management API
    - Automation friendly, reduced credential exposure
    - Expiration, deletion, rotation, custom keys, granular access control via endpoint restriction
  - OAuth 2.0 - API GW endpoints such as EDH

Authentication Comparison



Authentication Type	Environment	Scope	Credentials Passed	Credentials Expiration	VCS Credentials Revocation	Access/Bearer Token duration
User Auth	Premise	Default deny. Limited to assigned user profile privileges and scoping.	Username & Password from DBRealm (Verint local)	Defined by DBRealm (Verint local) global user policy	Verint portal disable account (set end date) or change password	Honors session expiration set in EM. Token expiration is automatically reset each time it is used before expiration.
API Key	Premise and Cloud	Unrestricted by default	Verint API Key ID & Hash	No Expiration by default	Newer versions can disable, revoke, rotate; though Verint portal requires Verint support ticket if EM privileges are insufficient.	The API Hash includes the time and URL, the token can only be used for the exact same URL for 5 minutes then the time offset is outside limits and the API call fails. In practice the API Key hash is generated for each API call with its own token.
OAuth 2.0 (EDH & API GW)	Cloud	Unrestricted	Tenant-id, Client-id, Client-secret	2 Years from date generated	Not provisioned in Verint portal – requires Verint support ticket.	3600 seconds

User Authentication

API Token Authentication

Postman Pre-Request Script

Authentication - Key Management API

- Key Management API is a RESTful API that gives administrators a set of endpoints for managing API keys.
  - Create, list, retrieve, enable, disable, customize an API key
  - You cannot use this API to change the type of a key (custom, internal, or external), nor to change the user.

- Process flow
  - 1. The API client uses an existing API key to call the API Keys Management API.
  - 2. The API client makes a call to a specific API to perform an action.

- API Keys Management: What's new - Guides - Engagement Data Management - Verint Connect

Authentication References

- Developer Portal Reference

  https://connect.verint.com/developers/edm/w/started/42216/api-authentication

- Postman API Token Authentication

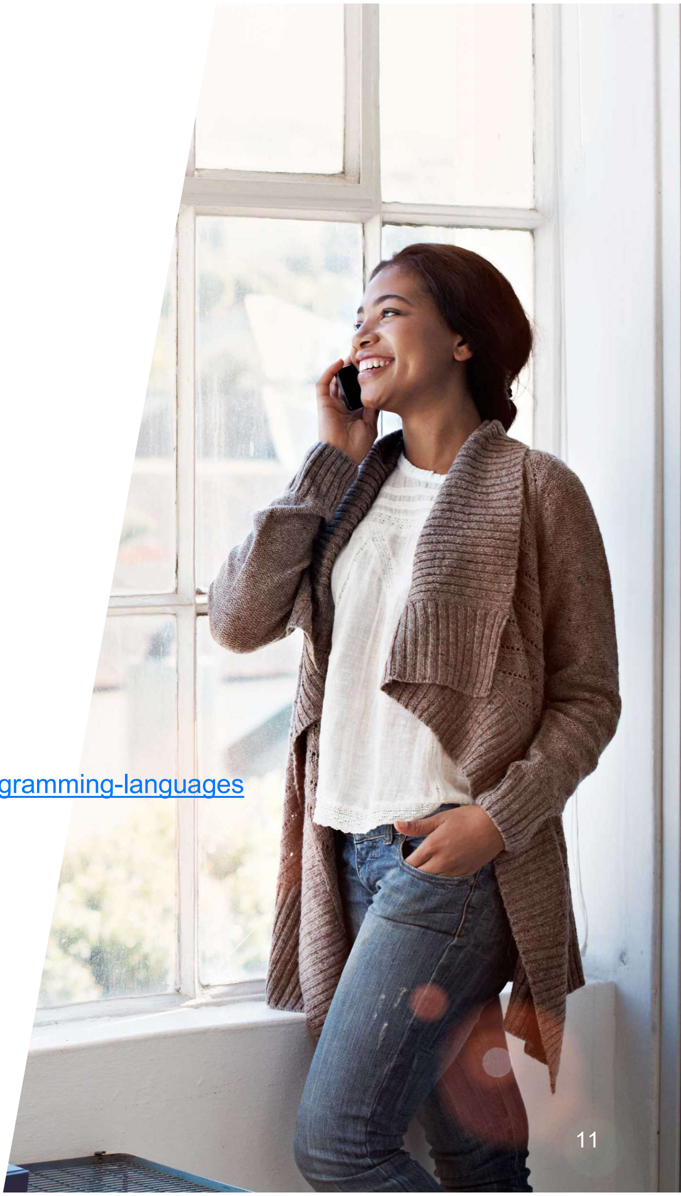
  https://connect.verint.com/developers/edm/w/started/12610/using-postman-with-api-token-authentication

- Other Languages

  https://connect.verint.com/developers/edm/f/discussions/28332/authenticating-with-verint-in-different-programming-languages

- Key Management

  https://connect.verint.com/developers/edm/w/guides/73796/api-apikeymanagement



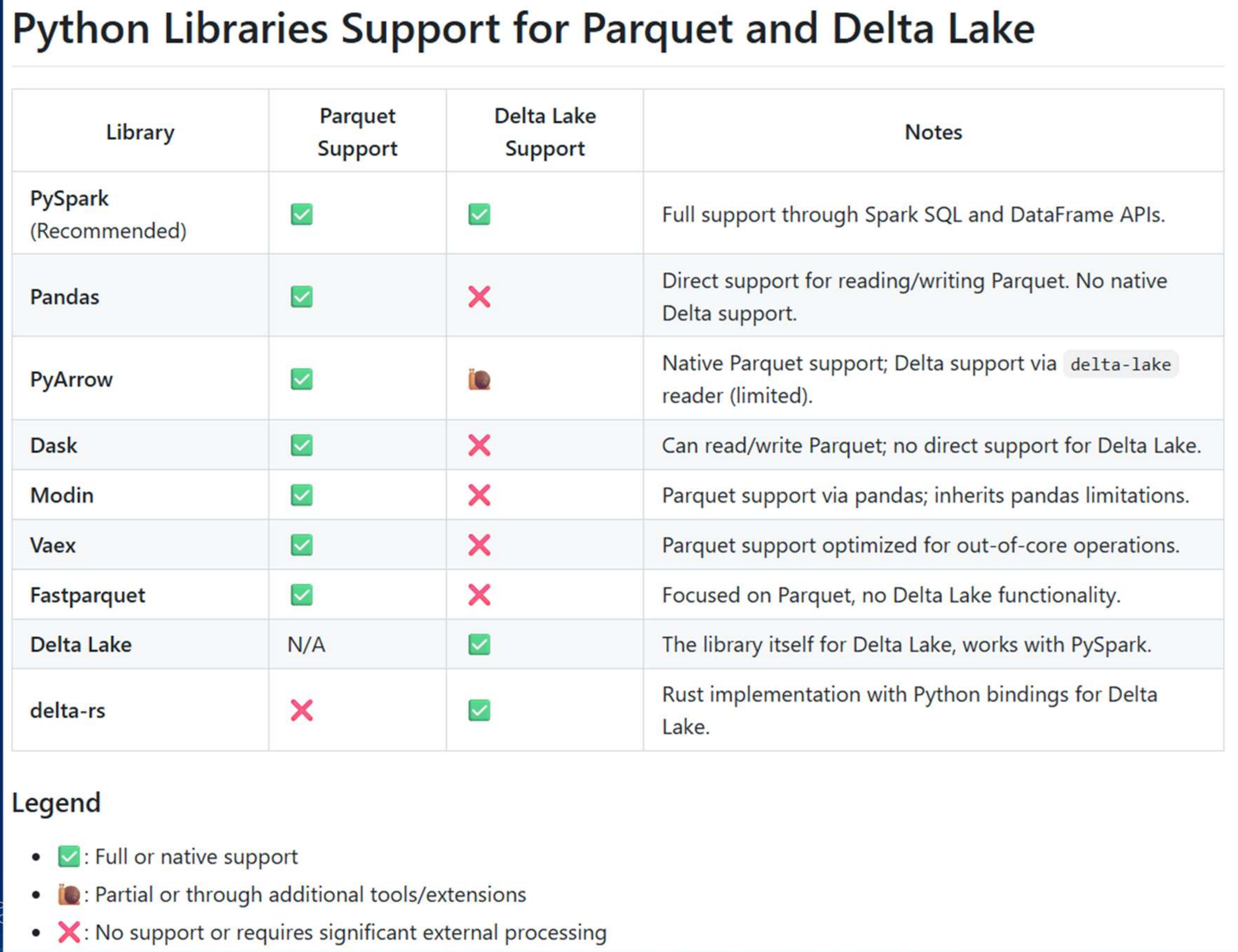
Section 3: Hands On Exercises with WFE APIs

Shannon Garrett

Efficient Searching With Employee & Interaction APIs

Section 4: EDH DLE API

Jon Sant



Python Libraries Support for Parquet and Delta Lake

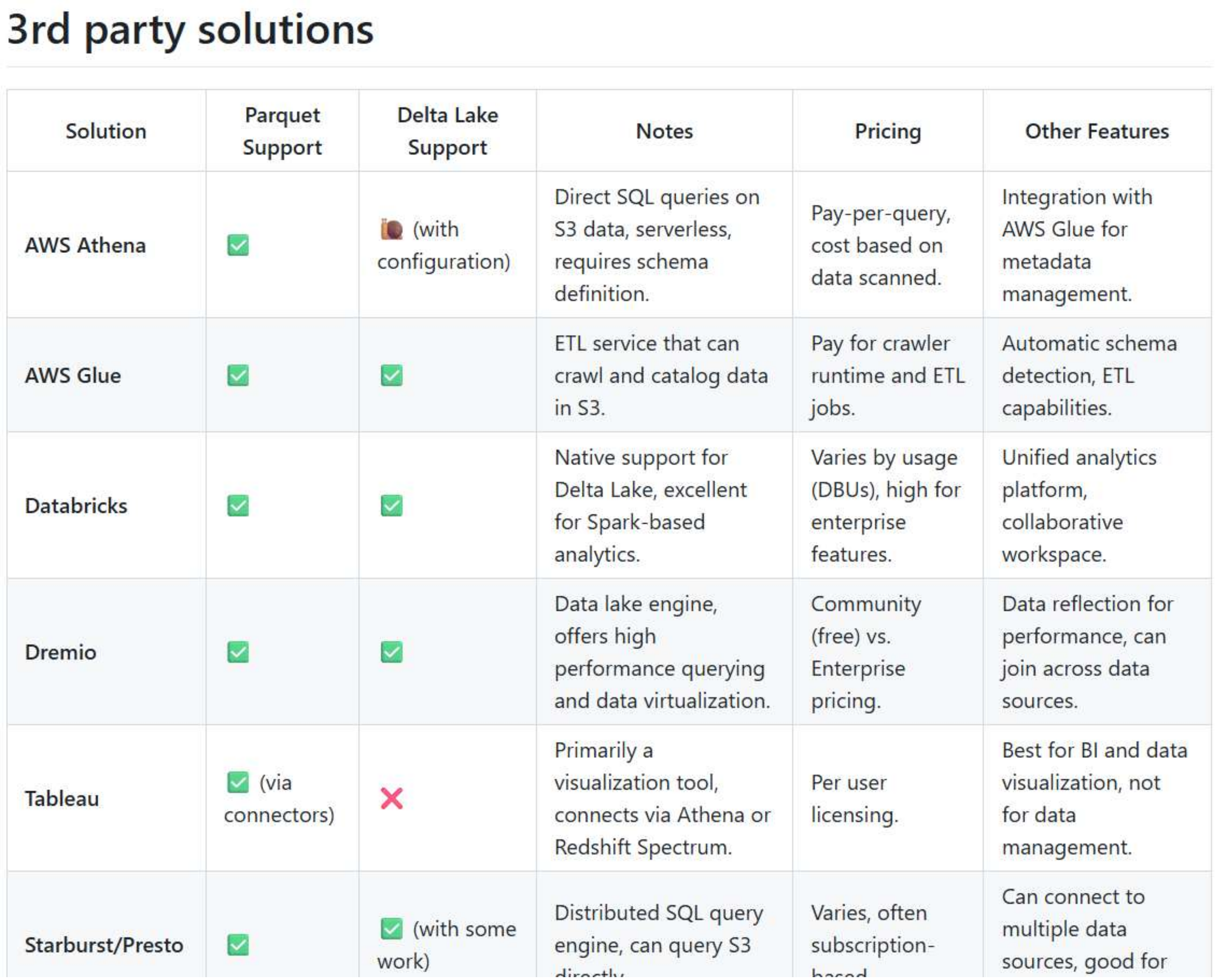
Library	Parquet Support	Delta Lake Support	Notes
PySpark (Recommended)	✅	✅	Full support through Spark SQL and DataFrame APIs.
Pandas	✅	❌	Direct support for reading/writing Parquet. No native Delta support.
PyArrow	✅	🐌	Native Parquet support; Delta support via `delta-lake` reader (limited).
Dask	✅	❌	Can read/write Parquet; no direct support for Delta Lake.
Modin	✅	❌	Parquet support via pandas; inherits pandas limitations.
Vaex	✅	❌	Parquet support optimized for out-of-core operations.
Fastparquet	✅	❌	Focused on Parquet, no Delta Lake functionality.
Delta Lake	N/A	✅	The library itself for Delta Lake, works with PySpark.
delta-rs	❌	✅	Rust implementation with Python bindings for Delta Lake.

Legend

- ✅: Full or native support
- 🐌: Partial or through additional tools/extensions
- ❌: No support or requires significant external processing

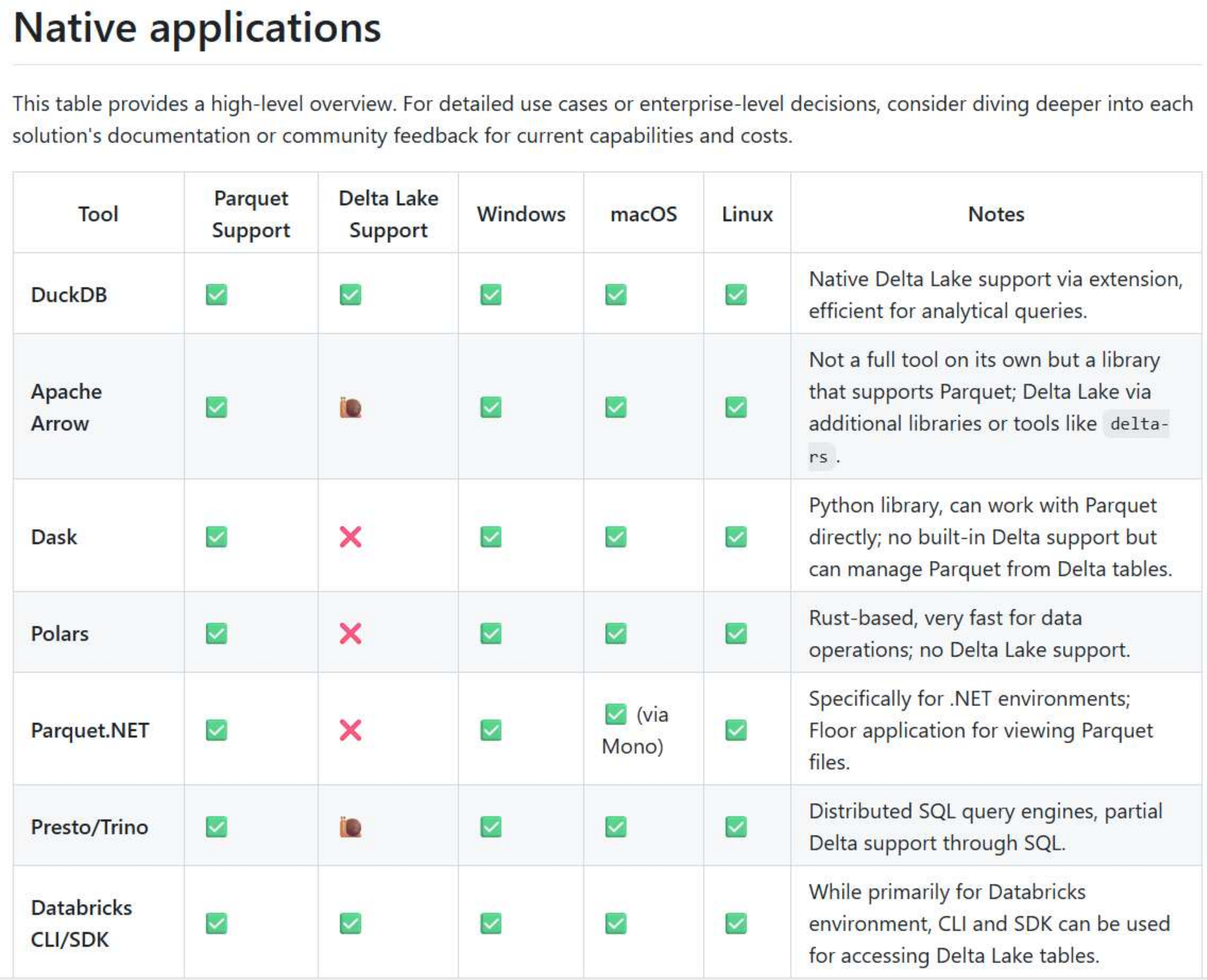
EDH – Python Libraries

- Parquet: Most libraries support Parquet due to its popularity as a columnar storage format.
- Delta Lake: Support mainly comes through integration with Spark or specific Delta Lake libraries like delta-rs.
- Integration: Many of these libraries can work together or use each other's capabilities (e.g., PyArrow is often used under the hood by other libraries for Parquet operations).



3rd party solutions

Solution	Parquet Support	Delta Lake Support	Notes	Pricing	Other Features
AWS Athena	✅	🐌 (with configuration)	Direct SQL queries on S3 data, serverless, requires schema definition.	Pay-per-query, cost based on data scanned.	Integration with AWS Glue for metadata management.
AWS Glue	✅	✅	ETL service that can crawl and catalog data in S3.	Pay for crawler runtime and ETL jobs.	Automatic schema detection, ETL capabilities.
Databricks	✅	✅	Native support for Delta Lake, excellent for Spark-based analytics.	Varies by usage (DBUs), high for enterprise features.	Unified analytics platform, collaborative workspace.
Dremio	✅	✅	Data lake engine, offers high performance querying and data virtualization.	Community (free) vs. Enterprise pricing.	Data reflection for performance, can join across data sources.
Tableau	✅ (via connectors)	❌	Primarily a visualization tool, connects via Athena or Redshift Spectrum.	Per user licensing.	Best for BI and data visualization, not for data management.
Starburst/Presto	✅	✅ (with some work)	Distributed SQL query engine, can query S3 directly	Varies, often subscription-based	Can connect to multiple data sources, good for



Native applications

This table provides a high-level overview. For detailed use cases or enterprise-level decisions, consider diving deeper into each solution's documentation or community feedback for current capabilities and costs.

Tool	Parquet Support	Delta Lake Support	Windows	macOS	Linux	Notes
DuckDB	✅	✅	✅	✅	✅	Native Delta Lake support via extension, efficient for analytical queries.
Apache Arrow	✅	🐌	✅	✅	✅	Not a full tool on its own but a library that supports Parquet; Delta via additional libraries or tools like `delta-rs`.
Dask	✅	❌	✅	✅	✅	Python library, can work with Parquet directly; no built-in Delta support but can manage Parquet from Delta tables.
Polars	✅	❌	✅	✅	✅	Rust-based, very fast for data operations; no Delta Lake support.
Parquet.NET	✅	❌	✅	✅ (via Mono)	✅	Specifically for .NET environments; Floor application for viewing Parquet files.
Presto/Trino	✅	🐌	✅	✅	✅	Distributed SQL query engines, partial Delta support through SQL.
Databricks CLI/SDK	✅	✅	✅	✅	✅	While primarily for Databricks environment, CLI and SDK can be used for accessing Delta Lake tables.

EDH DLE – Purpose

- Metadata
  - Defines tables and their respective columns
  - Table relationships, PK, FK
  - Data types and description
- Data
  - Primary purpose
  - Targeted download
    - Parquet, CDC, delta log
- Recommendation: Use the right tool for the job
  - If S3 is in scope, leverage Verint's S3 replication feature (free, customer owns s3 storage costs)

Live demo

Section 5: Troubleshooting and Best Practices

Jon Sant

Troubleshooting - Isolate first

- Cut the problem in half
  - Environment
    - Proxy interference
    - Connection pooling
    - TLS protocol, version, certificate CA trust
    - DNS/NAT can affect routing, TLS
    - Service or network maintenance (planned or unplanned)
  - Product
    - API key provisioning
    - User rights
    - Service or network maintenance (planned or unplanned)
    - EM is the source of truth for data and service availability

Troubleshooting – Common issues

- Authentication SWT signing
  - Compare to Postman outcome for the same endpoint and auth method
- Intermediary device is dropping or changing content or headers
- Performance: Load at scale
  - WFE: Check WFO-Server-ID distribution relative to request count
  - Premise or Hybrid: Ensure server role and configuration is set up
- API deprecation
  - Check API online and what's new section
- Schedule API – Request is validated only; updates happen afterwards
  - CallbackURL is not specified or callback event is ignored
- Imperva firewall

Best Practices

- Multi-threading
  - Use parallelism when the endpoint does not support batch processing and/or at large scale such as mass non-RT transcription retrieval
  - Response header WFO-Server-ID is key for WFE endpoints
    - Log it. Will expedite any Verint side troubleshooting.
    - Layer 7, http client session management can force load distribution (assuming L4 isn't deployed)
    - Ensure ratio is distributed according to app server count
- Handle authentication errors on the fly
  - Re-sign and retry
  - Be careful of AD account lockout if using LDAP on premise; password lockout policy applies
- Gracefully handle connection or service outages
  - 5xx errors should be retried
  - Verint service restart is typically between seconds to 15 minutes depending on the service affected, typically timing is closer the former.

Best Practices - Getting help

- Postman is our shared method of testing and validation
  - Language agnostic, allows isolating between API and code
  - Critical step for determining an issue's root cause
- Collect
  - Request/Response header and body
  - Network information
  - Response code, response time (with waterfall if available)
  - Authorization header
  - Is the problem intermittent or constant
  - When did the problem start and when did the functionality last succeed
- Report to appropriate resource
  - API Consultant
  - Support

Section 6: Wrap Up

Ryan

Session Goals

- Understand Integration Studio
- Understand API Authentication
- Gain Practical Experience with Verint APIs
- Answer Questions

Review

1. Integration Studio
2. Authentication In Depth
3. Hands On Exercises with WFE APIs
4. EDH DLE data retrieval and processing
5. Best practices

Thank you!

Matthew Monahan, Sr. Director, Data Strategy
Jonathan Sant, Principal System Integrator, Advanced Solutions
Shannon Garrett, Solution Delivery Architect, Advanced Solutions
Ryan de Leon, Sr. Manager, Advanced Solutions

Leave a Review

Master Class: Verint API Integration Techniques using Integration Studio

https://github.com/Verint-Engage-API-Masterclass