
π - QLens

Version: 01

Author : PibyThree Developers

Table of Contents

Table of Contents	1
1. Introduction.....	2
2. Key Considerations	4
2.1 Scope	4
2.2 Out of Scope.....	4
2.3 Assumptions	5
2.4 Dependencies	5
2.5 Risks.....	6
3. Business Feature Details	7
3.1 Business Process Flow	7
3.2 RBAC and UBAC	7
3.3 Scan Configuration & Execution	8
3.4 Monitoring & Reporting	9
4. Technical Feature Details.....	10
4.1 Technical Architecture.....	10
4.2 Data Quality Engine.....	11
4.3 Account & Dataset Configuration	12
4.5 Security & Access Control.....	13
5. Technical Details – Frontend	14
5.1 Page Flow & Navigation.....	15
5.2 UI Screens.....	15
5.2.1 Connections Screen.....	15
5.2.2 Rules Screen	16
5.2.3 Evaluation Screen	16
5.2.4 Object Quality Workspace	16
5.2.5 Table Detail Page	17
6. Technical Details – Backend.....	18
6.1 Snowflake Container Services (SPCS) Deployment.....	18
6.5.1 Compute Pool Details	18
6.5.2 Snowflake Service Details.....	18
6.5.3 Service Specification.....	19

6.6 Manifest Configuration	19
6.6.1 Privileges	19
6.6.2 References	19
6.6.3 Grant Callback	19
7. Data Quality Engine Details	20
7.1 Execution Modes & Scanning Patterns	20
7.2 Core Quality Dimensions & Rule Catalog	20
7.2.1 Completeness	20
7.2.1 Uniqueness	21
7.2.2 Technical Validity	21
7.2.3 Consistency & Schema Drift	22
7.2.4 Freshness	23
7.2.5 Volume	23
7.3 Results Aggregation & Downstream Tracing	23
7.4 Snowflake Data Access Requirements	23
8. Troubleshooting	25
9. Glossary	26
10. Reference Artifacts & Editable Links	27
11. Appendix	29
11.1 Usage Guidelines for Contributors	29
11.2 Environment Deployment Guide	29
11.2.1 Prerequisites	29
11.2.2 Deployment Steps (Provider)	30
11.2.3 Installation Steps (Consumer)	30
11.2.4 Environment Configuration	30
11.2.5 Validation	31
11.3 Pi-QLens Setup Guide	31
11.4 End-to-End Demo Walkthrough	32
11.4.1 Create a Connection	32
11.4.2 Add Users and Assign Roles	33
11.4.3 Apply Example Rules	33
11.4.4 Run a Scan	33

11.4.5 Review Results and Failures	34
11.4.6 Navigate Across the Main Pages	34
12. Support & Contact Information	35
12.1 Technical Support	35
12.2 Business & Sales Inquiries	35
12.3 Documentation & Feedback.....	35

1. Introduction

This document outlines the requirements, architecture, and implementation details for the **Pi-QLens** solution for **Data Quality Monitoring**. The objective of this initiative is to deliver a secure, scalable, and reliable data quality framework—distributed as a **Snowflake Native App**—that enables organizations to continuously monitor, assess, and improve the quality of data within their Snowflake environments through automated quality checks, rule-based validations, anomaly detection, and comprehensive reporting.

The document serves as a single source of truth for all stakeholders involved in the project, including business users, data stewards, data engineers, solution architects, testers, and project managers. It captures both functional and non-functional requirements, along with detailed backend and frontend implementation approaches within the Snowflake Native App Framework.

The Pi-QLens solution leverages the Snowflake Native App distribution model and high-performance capabilities such as Snowpark Container Services (SPCS), Snowflake Image Repositories, and Reference Binding for secure access management and application configuration. This approach ensures that all processing is executed within an isolated, secure environment in the consumer account, maintaining strict data governance, data privacy, and operational efficiency.

This initiative is part of the broader **Pi-Snow** roadmap and is designed to improve data quality observability, governance, and trustworthiness across enterprise data assets through a packaged application model. The solution supports configurable data quality rules, scheduled and on-demand scans, quality scorecards, downloadable reports, and actionable insights while minimizing manual configuration for end users.

As part of the Pi-Snow roadmap, future releases of Pi-QLens are planned to expand support for additional enterprise data platforms and quality dimensions, providing a centralized and scalable data quality monitoring experience across the organization's data ecosystem.

2. Key Considerations

This section outlines the foundational elements that influence the scope, planning, and execution of the Pi-QLens project, including scope boundaries, assumptions, dependencies, and risks.

2.1 Scope

The scope of project includes the following capabilities:

- Data Quality Assessment across Snowflake databases, schemas, and tables.
- Execution of quality checks across multiple dimensions, including:
 - Completeness
 - Uniqueness
 - Technical Validity
 - Consistency
 - Freshness
 - Volume
 - Business Metric Validation
 - Schema Drift Detection
- Support for both Full Scans and Incremental Scans to accommodate varying data volumes, refresh frequencies, and operational requirements.
- Configuration of quality rules at multiple levels, including database, schema, and table granularity.
- Creation and management of custom business rules to address organization-specific data quality requirements.
- Centralized monitoring of data quality results through the Pi-QLens application interface.
- Generation and download of assessment reports in HTML and Excel formats for audit, compliance, and operational review purposes.
- Multi-account connectivity, enabling multiple Snowflake accounts to be connected and managed through a single Pi-QLens application instance.
- Historical tracking of quality assessment results to support trend analysis and continuous improvement initiatives.
- Role-based access controls to ensure governed access to quality configurations, scan execution, and reporting capabilities.
- Execution of all quality assessment and reporting workflows within the Snowflake Native App framework, leveraging Snowpark Container Services for secure and scalable processing.

2.2 Out of Scope

The following items are not included in the current scope:

- Automated correction, modification, or deletion of source data.
- Data ingestion, ETL/ELT processing, or data movement between systems.
- Real-time streaming data quality validation.
- Enterprise data cataloging and end-to-end data lineage management.
- Data quality assessment for systems outside configured Snowflake environments.

2.3 Assumptions

The successful deployment and operation of Pi-QLens is based on the following assumptions:

Sr. No.	Description	Implications if Not Met
1	Snowflake account with Snowpark Container Services (SPCS) enabled.	Application cannot be deployed or executed.
2	Required Snowflake roles and privileges are available for application installation and configuration.	Application objects and services cannot be created.
3	Users have appropriate access to the databases, schemas, and tables intended for quality assessment.	Data quality scans cannot be executed on selected data assets.
4	Sufficient warehouse and compute resources are available to support scan execution and report generation.	Scans may fail, timeout, or experience degraded performance.
5	The customer is responsible for defining and maintaining business-specific quality rules and thresholds.	Quality assessments may not accurately reflect business expectations.

2.4 Dependencies

Sr. No.	Description	Owner	Implications if Not Met
1	Application installation and required account-level privilege grants	Consumer Account Administrator	Application cannot be installed or configured successfully.
2	Snowpark Container Services (SPCS) and Compute Pool availability	Consumer Account Administrator	Pi-QLens services cannot be deployed or executed.
3	Warehouse availability for executing data quality scans and generating reports	Consumer Account Administrator	Scan execution and reporting functionality will be unavailable.
4	Access to target databases, schemas, and tables for quality assessment	Data Owner / Consumer Administrator	Data quality rules cannot be evaluated against the selected data assets.
5	Application role mapping and user access configuration	Consumer Account Administrator	Users will be unable to access Pi-QLens features based on their responsibilities.

2.5 Risks

Sr. No.	Description	Mitigation Strategy
1	Insufficient access privileges on target databases, schemas, or tables may prevent quality scans from executing.	Validate required permissions during onboarding and periodically review role assignments.
2	Large-scale scans on high-volume datasets may increase execution time and compute consumption.	Use incremental scans where applicable and optimize scan schedules based on business requirements.
3	Incorrectly configured custom rules or thresholds may produce inaccurate quality assessments.	Review and validate custom rules before enabling them in production environments.

3. Business Feature Details

This section describes the key business features of Pi-QLens, including data quality assessment, scan configuration and execution, access management, monitoring, and reporting capabilities. These features enable organizations to effectively manage and govern data quality across Snowflake environments.

3.1 Business Process Flow

Pi-QLens enables organizations to assess and monitor data quality across Snowflake environments through a configurable and governed workflow.

The business process consists of the following steps:

1. Connect one or more Snowflake accounts to the Pi-QLens application.
2. Add Users, assign roles and allocate resources.
3. Select the databases, schemas, and tables to be monitored.
4. Configure data quality rules and thresholds, including standard and custom business rules.
5. Execute full or incremental data quality scans based on business requirements.
6. Review quality assessment results across dimensions such as Completeness, Uniqueness, Technical Validity, Consistency, Freshness, Volume, Business Metrics, and Schema Drift.
7. Analyze identified issues and trends through the Pi-QLens dashboard.
8. Generate and download assessment reports in HTML or Excel format for operational review, governance, and compliance purposes.

3.2 RBAC and UBAC

Pi-QLens protects data using two complementary controls:

1. Roles determine what a user can do

- a. Users are assigned one of several roles.
- b. Each role defines what kind of queries or actions are allowed.
- c. Example role types:
 - i. **Admin**
 - full management rights
 - can configure connections, manage users, and control data access
 - can run and inspect queries across the platform
 - ii. **Data Steward**
 - focused on data operations and governance
 - can work with datasets, apply rules, and run scans
 - iii. **Viewer**
 - read-only access
 - can see available connections, datasets, and reports

2. Resource-level controls determine what a user can see

- a. Access is granted at the connection level first.
- b. Then, within a connection, access can be restricted by:
 - i. database
 - ii. schema
 - iii. table
- c. This means a user may have access to a connector but only see a limited subset of the data inside it.

3.3 Scan Configuration & Execution

Pi-QLens enables users to configure and execute data quality assessments across selected Snowflake environments.

Users can define scan configurations at the database, schema, or table level and associate one or more quality rules with the selected data assets.

The platform supports:

- Full Scans for comprehensive assessment of entire datasets.
- Incremental Scans for evaluating newly added or modified data.
- Rule-based execution using standard and custom quality rules.
- On-demand scan execution for immediate validation and analysis.

Quality assessments can be performed across multiple dimensions, including:

- Completeness
- Uniqueness
- Technical Validity
- Consistency
- Freshness
- Volume
- Business Metrics
- Schema Drift

Scan results are stored within the application and made available through dashboards, historical trend analysis, and downloadable reports.

3.4 Monitoring & Reporting

Pi-QLens provides centralized monitoring and reporting capabilities to enable continuous visibility into data quality across connected Snowflake environments.

The platform offers:

- Monitoring of scan execution status, including successful and failed scans.
- Visibility into quality scores and rule-level assessment results.
- Historical tracking of data quality trends across datasets and quality dimensions.
- Identification of failed checks and quality violations requiring attention.
- Scan execution history for operational review and auditing purposes.
- Downloadable assessment reports in HTML and Excel formats.
- Centralized dashboards for monitoring data quality posture across multiple Snowflake accounts.

These capabilities help data teams identify quality issues early, track remediation progress, and demonstrate compliance with organizational data quality standards.

4. Technical Feature Details

This section outlines the technical interpretation of business features. It translates business needs into technical components, ensuring alignment between business goals and system capabilities within the Snowflake Native App Framework.

4.1 Technical Architecture

The solution follows a Snowflake Native App architecture with a containerized web application:

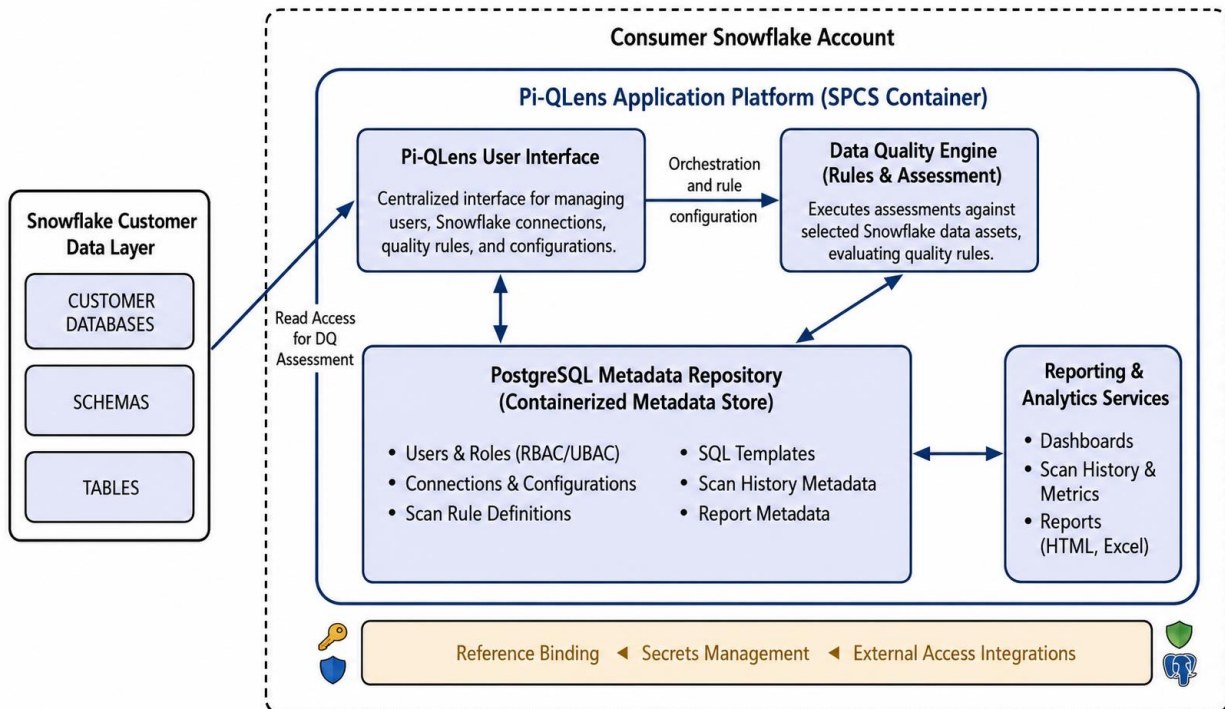


Fig 1 — Pi-QLens Architecture

Pi-QLens is deployed as a Snowflake Native App and utilizes Snowpark Container Services (SPCS) as its execution layer for data quality assessments. The application separates operational metadata management from scan execution to provide a scalable and governed architecture.

The architecture consists of the following key components:

Pi-QLens User Interface

Provides a centralized interface for managing users, Snowflake account connections, quality rules, scan configurations, and reporting.

- Framework: Next.js (React/TypeScript) with Tailwind CSS
- Port: 3000 (exposed via SPCS endpoint)
- Access: Consumers click “Open” in Snowsight to access the UI, already authenticated via Snowflake
- Client-Side: Tailwind CSS, React, Axios for API calls, and ExcelJS for Excel export

PostgreSQL Metadata Repository

Acts as the primary application database and stores:

- User and role information
- RBAC and UBAC configurations
- Snowflake account connections
- SQL templates
- Quality rule definitions
- Scan configurations
- Scheduling information
- Application metadata
- Report metadata

Data Quality Engine

Executes configured quality assessments and evaluates quality rules against selected Snowflake data assets.

Snowpark Container Services (SPCS)

Provides the execution environment for the Pi-QLens application. Scan orchestration, rule evaluation, query execution, and report generation are performed through services running within SPCS.

Snowflake Data Layer

Contains the customer databases, schemas, and tables selected for quality assessment. Pi-QLens executes quality checks against these resources based on configured permissions.

Reporting & Monitoring Layer

Provides dashboards, scan history, quality metrics, trend analysis, and downloadable HTML and Excel reports.

4.2 Data Quality Engine

The Pi-QLens Data Quality Engine is responsible for executing quality assessments against configured Snowflake data assets. The engine evaluates datasets using both system-defined and user-defined quality rules and generates detailed quality metrics and findings.

Key capabilities include:

- Execution of Full Scans and Incremental Scans.
- Evaluation of quality rules at the database, schema, and table levels.
- Support for configurable quality dimensions, including:
 - Completeness
 - Uniqueness
 - Technical Validity
 - Consistency
 - Freshness

- Volume
- Business Metrics
- Schema Drift
- Support for custom business rules tailored to organizational requirements.
- Rule-based quality scoring and result aggregation.
- Historical tracking of quality assessment results for trend analysis.
- Generation of detailed findings, exceptions, and quality metrics for reporting and monitoring.

The engine executes assessments against user-selected Snowflake data assets and stores the resulting metrics and execution metadata for subsequent analysis and reporting.

4.3 Account & Dataset Configuration

Pi-QLens enables users to connect and manage multiple Snowflake accounts through a centralized interface. This allows organizations to monitor and assess data quality across multiple environments from a single application instance.

The platform provides the following configuration capabilities:

Snowflake Account Management

- Register and manage multiple Snowflake account connections.
- Validate account connectivity before onboarding datasets.
- Maintain account-specific configurations and permissions.

Dataset Configuration

Users can define the scope of quality assessments by selecting:

- Databases
- Schemas
- Tables

Quality rules can be configured at different levels of granularity, enabling organizations to apply specific controls to selected data assets.

Rule Assignment

Pi-QLens supports:

- System-defined quality rules.
- User-defined custom quality rules.
- Dataset-specific rule configurations.
- Quality thresholds and validation criteria.

Scan Scope Management

Users can configure:

- Full dataset scans.
- Incremental scans.

This flexible configuration model enables organizations to align quality assessments with their business requirements, operational processes, and governance standards.

4.5 Security & Access Control

Pi-QLens implements Role-Based Access Control (RBAC) and User-Based Access Control (UBAC) to provide secure and governed access to application functionality and monitored data assets.

Application Roles

Pi-QLens supports the following application roles:

Role	Description
Admin	Manages application configuration, user access, Snowflake account connections, dataset onboarding, quality rules, and platform administration activities.
Data Steward	Configures and manages data quality rules, executes scans, reviews quality results, and monitors data quality performance for assigned resources.
Viewer	Provides read-only access to assigned resources, dashboards, scan results, and reports.

Access Levels

In addition to application roles, Pi-QLens supports resource-level access controls that determine the actions a user can perform on assigned databases, schemas, and tables.

Access Level	Description
READ	Allows users to view and browse assigned databases, schemas, tables, scan results, dashboards, and reports.
OPERATE	Includes READ permissions and allows users to execute scans, configure dataset rules, and perform operational activities on assigned resources.
MANAGE	Includes READ and OPERATE permissions and allows users to administer assigned resources, including configuration and access management activities.

Resource-Level Authorization

Access levels can be assigned at the following levels:

- Database
- Schema
- Table

This enables organizations to implement fine-grained security controls while ensuring users only have access to the resources required for their responsibilities.

Relationship with Snowflake Permissions

Pi-QLens access controls govern application functionality and resource visibility within Pi-QLens. They do not grant Snowflake object privileges.

The Snowflake user or role configured in the active connection must still have the required Snowflake privileges on the target warehouse, database, schema, and tables or views. If the underlying Snowflake permissions are missing, users may encounter incomplete metadata discovery, unavailable objects, failed scans, or missing quality assessment results.

5. Technical Details – Frontend

5.1 Page Flow & Navigation

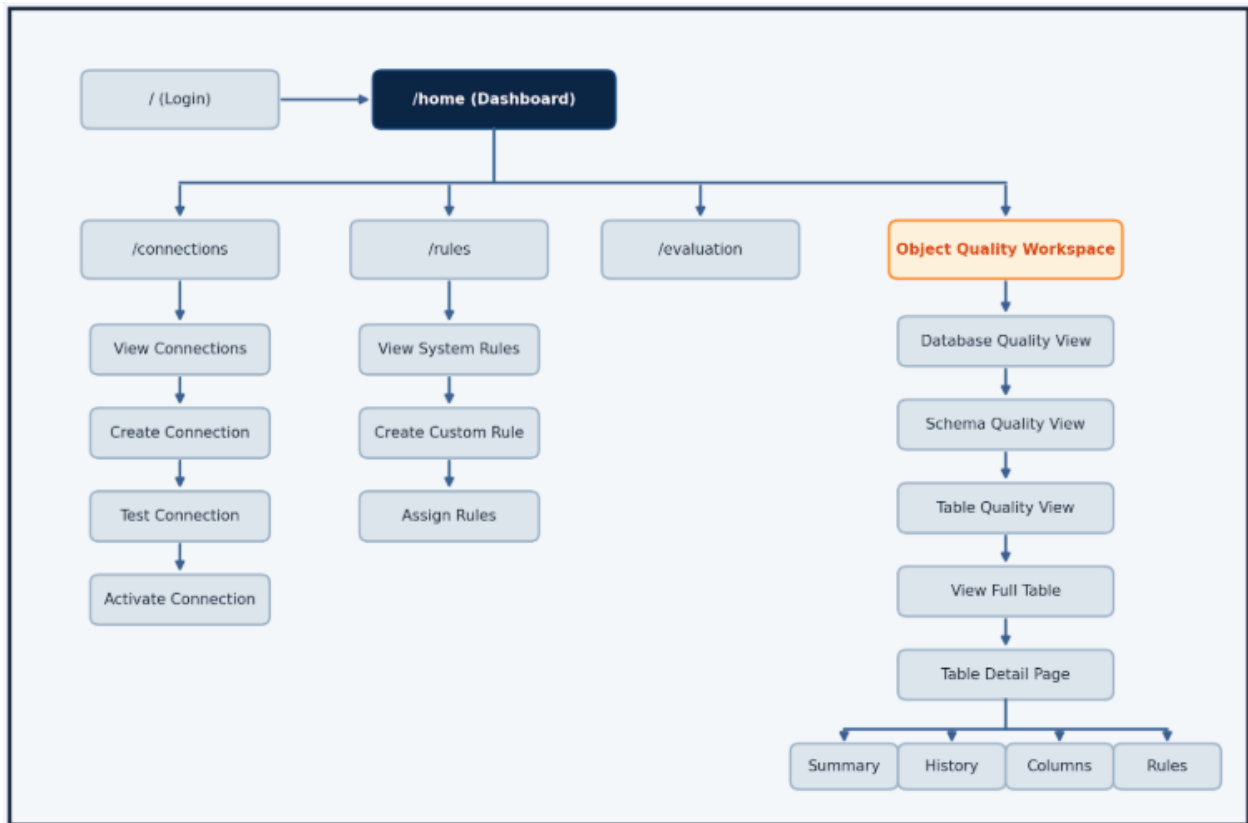


Fig 2 — Page Flow & Navigation through the Pi-QLens Web Application

5.2 UI Screens

Pi-QLens provides a web-based user interface for configuring connections, managing data quality rules, executing assessments, and reviewing quality results across Snowflake environments.

The application consists of a primary navigation menu and a set of functional workspaces designed to support the end-to-end data quality lifecycle.

5.2.1 Connections Screen

The Connections screen provides centralized management of Snowflake account connections used by Pi-QLens.

Available Actions

- View configured connections
- Create a new connection

- Edit connection details
- Test connection connectivity
- Activate a connection
- Deactivate a connection

Note: Multiple connections can be configured, but only one connection can be active at a given time.

5.2.2 Rules Screen

The Rules screen provides centralized management of system-defined and custom data quality rules.

Available Actions

- View available quality rules
- Search and filter rules
- Create custom rules through a popup interface
- Configure rule parameters
- Assign rules to datasets and tables
- Modify existing custom rules

5.2.3 Evaluation Screen

The Evaluation screen displays platform-level evaluation and automation information.

Available Actions

- Review platform evaluation metrics

5.2.4 Object Quality Workspace

The Object Quality Workspace provides hierarchical navigation through monitored Snowflake objects and their quality scores.

Navigation Hierarchy

Database



Schema



Table

Available Actions

- View database-level quality scores
- Drill down to schema-level quality scores
- Drill down to table-level quality scores
- Review quality metrics for selected objects
- Access detailed table analysis using **View Full Table**

5.2.5 Table Detail Page

The Table Detail Page is accessed through the **View Full Table** action available from the Table Quality View within the Object Quality Workspace.

Tab	Description
Summary	Displays overall quality score, quality dimensions, and latest scan results.
History	Displays historical scan executions and quality trends.
Columns	Displays column-level quality assessments and profiling details.
Rules	Displays rules assigned to the table and their execution results.

6. Technical Details – Backend

6.1 Snowflake Container Services (SPCS) Deployment

This section describes the deployment of the **Pi-QLens** application using Snowpark Container Services within the Snowflake Native App Framework.

- **Packaging:** The data quality profiling engines, metadata validation logic, and Next.js web UI are containerized using Docker and bundled directly into the Application Package.
- **Image Security:** Docker images are stored in an internal Provider Image Repository, scanned by Snowflake for security vulnerabilities prior to distribution.
- **Service Instantiation:** A Snowflake Service is created within the Application Instance to run the containerized Next.js web application.
- **Isolated Compute:** The service executes data profiling rules and audit log tracing within an isolated, Snowflake-managed compute environment.
- **Public Endpoint:** The service exposes a secure web UI endpoint accessible via Snowsight's "Launch App" button.
- **Reference-Based Configuration:** Encryption keys, External Access Integrations (EAI), and compute warehouses are managed through manifest references and service specifications.

6.5.1 Compute Pool Details

Sr. No	Object Name	Object Type	Description
1	PI_QLENS_COMPUTE_POOL	COMPUTE POOL	Account-level compute resource provisioned automatically by the <code>grant_callback</code> . Utilizes the CPU_X64_XS instance family with auto-suspend configured at 600 seconds.

6.5.2 Snowflake Service Details

Sr. No	Service Name	Object Type	Description
1	PI_QLENS_SERVICE	SERVICE	Containerized Snowflake Service created within the Application Instance to run and expose the secure web application UI.

6.5.3 Service Specification

The containerized service configuration is defined within the `service_spec.yaml` file:

- **Container Image:** `/PI_SNOW/PI_QLENS/PI_QLENS_IMAGE_REPO/piqlens:v1`
- **Resources:** 512MB–2GB memory allocations, 0.5–2 dedicated CPU cores.
- **Endpoint:** Port 3000, configured as a public endpoint named "ui".
- **Warehouse:** Bound dynamically via `REFERENCE('warehouse_ref')` to drive operational query execution.

6.6 Manifest Configuration

6.6.1 Privileges

Privilege	Description
CREATE COMPUTE POOL	Required to provision the account-level compute pool running the SPCS container.
BIND SERVICE ENDPOINT	Required to securely expose the public web UI endpoint for user access.

6.6.2 References

Name	Type	Used	Description
<code>piqlens_eai_ref</code>	EXTERNAL ACCESS INTEGRATION	Yes	Establishes network egress for external database and metadata connectivity.
<code>warehouse_ref</code>	WAREHOUSE	Yes	Binds the target query warehouse for data profiling and rule execution.

6.6.3 Grant Callback

The `grant_callback` property configured in the application manifest explicitly points to `CORE.GRANT_CALLBACK`. This stored procedure is automatically executed immediately after the consumer grants the required account privileges, performing the following lifecycle actions:

- **Validates CREATE COMPUTE POOL:** Automatically provisions and launches the `PIQLENS_COMPUTE_POOL`.
- **Validates BIND SERVICE ENDPOINT:** Instantiates the `CORE.PI_QLENS_SERVICE` using the specification file, injected dynamically with the authorized EAI and warehouse references.

7. Data Quality Engine Details

The **Pi-QLens Data Quality Engine** is the core execution framework responsible for profiling, assessing, and monitoring your Snowflake data assets. By evaluating datasets against both out-of-the-box system rules and custom user-defined assertions, the engine translates raw database checks into actionable business trust scores.

7.1 Execution Modes & Scanning Patterns

To balance performance cost with validation depth, the engine supports two distinct execution patterns:

- **Full Scans:** Evaluates the entire table or partition history to establish baselines, calculate historical rolling statistics, and perform complete historical audits.
- **Incremental Scans:** Utilizes configured **High-Watermark Columns** (e.g., updated timestamps, auto-incrementing IDs) to isolate and execute rules only on newly appended or modified records since the last validation run.

7.2 Core Quality Dimensions & Rule Catalog

Below is the complete, technically accurate mapping of the **28 pre-built system rules** extracted from your engine configurations, organized by their respective quality dimensions and execution levels (COLUMN vs. TABLE):

7.2.1 Completeness

Ensures that expected data is present and monitors null-value behaviors.

Rule Name	Level	Description
COMPLETENESS_CHECK	COLUMN	Check for NULL / empty values in a column.
CONDITIONAL_COMPLETENESS	COLUMN	Check completeness only in rows that satisfy a custom WHERE condition.
NULL_RATE_ANOMALY	COLUMN	Detect anomalies in the null rate compared to a rolling historical baseline.

7.2.1 Uniqueness

Validates that records and keys are distinct without unintended duplication.

Rule Name	Level	Description
UNIQUENESS_PRIMARY_KEY	COLUMN	Check for duplicate values, treating the selected column(s) as a primary key.
DUPLICATE_RECORD_CHECK	COLUMN	Check for duplicate records across a selected column set or the entire row.
COMPOSITE_UNIQUENESS_CHECK	COLUMN	Check uniqueness across a compound key of multiple columns.

7.2.2 Technical Validity

Validates that values conform to strict structural, format, and domain constraints.

Rule Name	Level	Description
VALIDITY_ALLOWED_VALUES	COLUMN	Check column values against a user-defined whitelist (allowed values).
VALIDITY_DATE_FORMAT	COLUMN	Validate date format, parsability, and structural integrity.
VALIDITY_DATE_RANGE	COLUMN	Check that dates fall within a valid min/max range (default: 1900 to present).
VALIDITY_EMAIL_FORMAT	COLUMN	Validate email address format (RFC 5322 compliant).
VALIDITY_PHONE_FORMAT	COLUMN	Validate phone number formats (supports international and national rules).
VALIDITY_NUMERIC_FORMAT	COLUMN	Validate numeric format, parsability, and optional precision/scale parameters.
VALIDITY_POSITIVE_NUMBER	COLUMN	Check that numeric values are strictly positive (> 0).
VALIDITY_NON_NEGATIVE_NUMBER	COLUMN	Check that numeric values are non-negative (≥ 0).

VALIDITY_COUNTRY_CODE	COLUMN	Validate country codes against the ISO 3166-1 standard.
VALIDITY_CURRENCY_CODE	COLUMN	Validate currency codes against the ISO 4217 standard.
ACCEPTED_RANGE_CHECK	COLUMN	Check if numeric or date values fall within a configured [min, max] interval.
STRING_LENGTH_CHECK	COLUMN	Check that string length is within configured minimum and maximum character bounds.
REGEX_PATTERN_CHECK	COLUMN	Validate string values against a user-supplied Regular Expression (Regex) pattern.
FUTURE_DATE_CHECK	COLUMN	Flag date or timestamp values that erroneously point to future dates.
PAST_DATE_THRESHOLD_CHECK	COLUMN	Flag dates that are older/earlier than a configured minimum historical threshold.

7.2.3 Consistency & Schema Drift

Enforces relational integrity, business logic consistency, and structural schema evolution.

Rule Name	Level	Description
CONSISTENCY_FOREIGN_KEY	COLUMN	Check foreign key referential integrity against a designated reference table.
CROSS_COLUMN_COMPARISON	COLUMN	Compare two columns in the same row using a relational operator (\$<, >, =, \ne\$, etc.).
SCHEMA_DRIFT_CHECK	TABLE	Detect unexpected schema changes (added, removed, or type-changed columns) vs. a stored baseline.

7.2.4 Freshness

Tracks data latency, execution SLAs, and ingestion pipelines.

Rule Name	Level	Description
FRESHNESS_DATA_RECENCY	COLUMN	Check data recency against a configured threshold (specified in days).
FRESHNESS_SLA_CHECK	COLUMN	Check maximum record age in minutes against a designated latency SLA budget.

7.2.5 Volume

Monitors volume thresholds and flags structural ingestion drops.

Rule Name	Level	Description
VOLUME_ROW_COUNT_THRESHOLD	TABLE	Check minimum (and optionally maximum) absolute row count thresholds.
VOLUME_ROW_COUNT_ANOMALY	TABLE	Detect volume anomalies in row count compared to a rolling statistical baseline.

7.3 Results Aggregation & Downstream Tracing

When the engine finishes executing these rules, the metadata and findings are processed as follows:

1. **Metric Scoring:** Individual column and table runs are aggregated into overall quality scores mapped directly to the **6 Core Quality Dimensions**.
2. **Audit Trail Logging:** Run results are committed to database audit logs, logging parameters such as execution timestamp, total rows evaluated, violation counts, and execution status.
3. **Exception Isolation:** For any failed assertions (such as invalid emails or orphaned foreign keys), the engine isolates the offending primary keys or entire records. This allows data engineers to inspect structural anomalies directly within the UI's **Record Trace Inspector** without having to query the raw source tables manually.

7.4 Snowflake Data Access Requirements

Pi-QLens application roles control what actions a user can perform within Pi-QLens, such as managing connections, configuring rules, executing scans, and viewing results. However, Pi-QLens roles do not replace or override Snowflake object-level permissions.

The Snowflake user or role configured in a connection profile must have the required privileges on the target data being monitored. At a minimum, the configured role should have:

- USAGE on the warehouse used for scans

- USAGE on the target database
- USAGE on the target schema
- SELECT on the target tables or views

Users can only view quality scores, scan results, and metadata for objects that are both:

1. Accessible through their assigned Pi-QLens resource permissions.
2. Accessible to the Snowflake user or role configured in the active connection.

For Steward users, Pi-QLens resource access should be granted only to the databases, schemas, and tables they are authorized to manage. The underlying Snowflake user or service account used by the connection must also have corresponding Snowflake privileges on those same objects.

Important: Pi-QLens application permissions do not grant access to Snowflake data. Missing Snowflake privileges may result in incomplete metadata discovery, missing objects in the Object Quality Workspace, failed quality scans, or unavailable quality assessment results.

8. Troubleshooting

This section outlines common technical issues that may arise during system operation along with recommended steps for diagnosis and resolution.

Issue	Description	Possible Causes	Recommended Actions
App UI Does Not Open	Pi-QLens UI fails to launch from Snowsight.	Required privileges not approved; EAI or warehouse reference not bound; service not running.	Verify privileges are approved, references are bound, and <code>CORE.PI_QLENS_SERVICE</code> is running.
User Management Shows Too Few Users	User list is incomplete.	<code>IMPORTED PRIVILEGES</code> on Snowflake database not granted.	Approve imported privileges and verify access to <code>SNOWFLAKE.ACCOUNT_USAGE.USERS</code> .
Connection Test Times Out	Connection validation fails or hangs.	Incorrect account details; invalid credentials; warehouse access issue; network restrictions.	Verify account URL, credentials, warehouse access, and EAI configuration.
Metadata Discovery Fails	Databases, schemas, or tables are not discovered.	Insufficient permissions on Snowflake objects.	Ensure the configured role has required metadata access and object privileges.
Quality Checks Return No Results	Scan completes without producing results.	Inactive connection; missing object permissions; warehouse unavailable.	Verify connection status, object permissions, and warehouse availability.
Scan Execution Performance Issues	Scans take longer than expected or consume excessive resources.	Large datasets; full scans on high-volume tables; undersized warehouse.	Use incremental scans where possible and review warehouse sizing.
Report Generation Fails	HTML or Excel reports cannot be generated or downloaded.	Scan results unavailable; service interruption; resource constraints.	Verify scan completion, service health, and retry report generation.
Missing Objects or Results	Expected databases, tables, or quality results are not visible.	Missing Pi-QLens resource access or insufficient Snowflake permissions.	Verify Pi-QLens access grants and ensure the active connection has required Snowflake <code>USAGE</code> and <code>SELECT</code> privileges on the target objects.

9. Glossary

Term	Definition
Pi-QLens	A Snowflake Native App for monitoring and assessing data quality across Snowflake environments.
Consumer Account	The Snowflake account where Pi-QLens is installed and operated.
Native App	A Snowflake application that runs directly inside the consumer's Snowflake account.
SPCS	Snowpark Container Services, Snowflake's managed platform for running containerized applications.
Compute Pool	The Snowflake-managed compute resource used to run SPCS services.
Data Quality Engine	The component responsible for executing data quality rules and assessments.
Schema Drift	Unintended changes in table structure, such as added, removed, or modified columns.
EAI	External Access Integration, a Snowflake object that enables secure outbound network connectivity.
Reference Binding	A Snowflake mechanism that securely associates consumer-owned resources (such as warehouses and integrations) with a Native App.
Grant Callback	A stored procedure automatically executed after privilege approval to complete application setup and provisioning.
Exception Record	A record that fails one or more configured data quality validations.

10. Reference Artifacts & Editable Links

This section provides direct access to editable project artifacts such as process diagrams, data mappings, and technical documentation. These resources are maintained in shared locations to ensure version control and collaborative updates.

Asset Type	Description	Official Documentation Link
Snowflake Platform Documentation	Official Snowflake documentation portal for SQL, architecture, security, administration, and advanced features.	Snowflake Documentation Home
Getting Started	Guide for onboarding and initial setup with Snowflake.	Getting Started with Snowflake
User Guides	Documentation covering Snowflake usage, warehouses, tables, data loading, and administration.	Snowflake User Guides
Architecture & Core Concepts	Explains Snowflake architecture, storage, compute, and platform concepts.	Snowflake Key Concepts & Architecture
Integration / Connectors	Documentation for connectors, drivers, APIs, and external system integrations.	Snowflake Integration Guides
Snowflake Container Services (SPCS)	Overview and concepts for running containerized workloads in Snowflake.	SPCS Overview
Snowflake Services	Documentation for creating, managing, and operating Snowflake services.	SPCS Services Guide
Compute Pools	Guide for configuring and managing compute pools for container execution.	Compute Pools Guide
Image Repository	Documentation for storing and managing Docker images within Snowflake.	Image Repository Guide

11. Appendix

This section includes supporting information, usage guidelines, and references.

11.1 Usage Guidelines for Contributors

- **Naming Conventions:** Follow established naming conventions for Snowflake objects, application roles, and manifest-defined references.
- **Security Scans:** All container images must undergo a mandatory security scan before publishing a new version.
- **Configuration Files:** Do not modify `setup.sql` or `manifest.yml` without an approved change request.
- **Credential Security:** Ensure no credentials, tokens, or encryption keys are hardcoded in the source code.
- **Local Validation:** Test all changes against a local Docker environment before pushing to the Snowflake image repository.

11.2 Environment Deployment Guide

11.2.1 Prerequisites

- **Snowflake Account:** Enterprise Edition (or higher) with Snowpark Container Services (SPCS) enabled in the target region.
- **Administrative Privileges:** ACCOUNTADMIN role, or the designated capability to grant CREATE COMPUTE POOL and BIND SERVICE ENDPOINT privileges.
- **Security Objects:** An encryption key secret (GENERIC_STRING) created directly in the consumer account.
- **Network Governance:** Network rules and External Access Integrations (EAI) configured for external database and metadata egress.
- **Warehouse:** An active X-Small or Small warehouse allocated for data profiling and query execution.

11.2.2 Deployment Steps (Provider)

1. **Build Docker Image:** Build the container image locally:
2. **Registry Push:** Tag and push the built image to your designated Snowflake image repository.
3. **Upload Stage Files:** Upload `manifest.yml`, `setup.sql`, `service_spec.yaml`, and `README.md` to the named application stage.
4. **Create Application Version:** Execute the version command:
`ALTER APPLICATION PACKAGE ... ADD VERSION V1_0 USING @stage;`
5. **Marketplace Submission:** Create the Snowflake Marketplace listing and submit the package for mandatory security review.

11.2.3 Installation Steps (Consumer)

1. **Marketplace Install:** Install the application from the Marketplace and name your application instance (e.g., PI_QLENS_APP).
2. **Configure Network Egress:** Define the network rules and create the corresponding EAI for external database/metadata hosts.
3. **Bind References:** Bind the EAI and the designated query warehouse via the Snowsight application configuration screen.
4. **Grant System Privileges:** Authorize the CREATE COMPUTE POOL and BIND SERVICE ENDPOINT privileges.
5. **Automated Provisioning:** Wait for the grant_callback procedure to automatically spin up the compute pool and instantiate the service.
6. **Launch Web UI:** Click the "Open" button in Snowsight to launch the Pi-QLens web interface.
7. **Refine Network Rules:** Update your network egress rules with specific database host endpoints as needed over time.

11.2.4 Environment Configuration

- **Service Isolation:** A single SPCS service handles all metadata profiling, validation rules, and tracing workloads.
- **Credential Management:** All configured external credentials are encrypted securely using a consumer-owned Fernet key.
- **Role Mapping:** The consumer maps the native APP_ADMIN application role to their internal Snowflake Role-Based Access Control (RBAC) hierarchy.
- **Resource Sizing:** Executes on a CPU_X64_S compute pool with the container memory footprint bounded between 512MB and 2GB.

11.2.5 Validation

Once the installation is complete, verify the setup using the following checklist:

- **Reference Verification:** Ensure all manifest references (EAI, Warehouse) report an 'ASSOCIATED' status.
- **Service Status:** Confirm the SPCS service status displays as RUNNING (monitored via *Snowsight* → *Monitoring*).
- **UI Access:** Click "Open" on the application instance and verify the Pi-QLens landing portal loads successfully.
- **Connection Test:** Navigate to the Connections tab, add a test source database, and confirm a "Connection successful" response.
- **End-to-End Run:** Onboard a dataset, configure validation rules, execute a data quality check run, and verify the resulting quality scores generate successfully[cite: 1].

11.3 Pi-QLens Setup Guide

Pi-QLens is configured from the Snowsight application setup screen after installation. Before you begin, make sure your Snowflake account can install Native Apps and that you have access to a warehouse and role with the required privileges. The setup flow is: install the app, approve privileges, open the UI, and then start configuring connections and rules.

Step 1: Install the App

1. Click **Get App** from the Snowflake listing.
2. Install the application in your Snowflake account.
3. Provide an application name, for example `PI-QLENS_APP`.
4. Open the installed application from Snowsight.

Step 2: Approve Privileges and Bind Objects

Approve the requested account privileges:

- **CREATE COMPUTE POOL:** Allows Pi-QLens to create the compute pool for its container service.
- **BIND SERVICE ENDPOINT:** Allows Snowflake to expose the Pi-QLens web UI endpoint.

Bind the required references:

- **External Access Integration:** Allows the container to reach Snowflake endpoints such as `*.snowflakecomputing.com:443`.
- **Warehouse:** Used for metadata discovery, quality checks, profiling, reporting, and application setup queries.

After the privileges and references are available, Pi-QLens creates its compute pool and service automatically.

Step 3: Open the Pi-QLens UI

After the service starts, click **Open** from the Snowsight app page.

You should see the Pi-QLens landing experience with the main navigation and initial setup state. The first user who opens the app is provisioned as the initial Pi-QLens admin. Additional users can be managed from the app admin screens.

11.4 End-to-End Demo Walkthrough

Use the following flow to demonstrate Pi-QLens from first connection to a completed scan and review. Each step below describes the exact click or action to take and the screen or result you should expect to see.

11.4.1 Create a Connection

1. Open the Pi-QLens UI and sign in with an admin or operator account.
2. In the top navigation, click **Connections**. You should land on the connection management page showing any existing Snowflake profiles.
3. Click **Create Connection**. A form opens for the new profile.
4. Enter a connection name such as *prod-snowflake*.
5. Enter the Snowflake account identifier, warehouse, role, and database you want Pi-QLens to inspect.
6. Click **Save**. You should see a success message confirming the profile was stored.
7. Click **Test Connection**. If the credentials and permissions are correct, you should see a confirmation that the connection is healthy.
8. Select it as the active profile. After this, the app will use this connection for catalog discovery, rules, and scans.

Important Notes

- If the connected Snowflake user is subject to an IP restriction, network policy, or network access rule, ensure that the policy allows connections originating from the network used by Pi-QLens before testing or activating the connection. This requirement applies to the Snowflake user configured in the connection profile.
- Pi-QLens automatically creates a system-defined **Self** connection representing the Snowflake account in which the application is installed. The **Self** connection is pre-configured with the local account identifier and cannot be deleted.
- Before the **Self** connection can be used, an administrator must configure the Snowflake username, authentication method, authentication token or credentials, warehouse, and any optional role, default database, or default schema settings.
- After configuration, use **Test Connection** to validate connectivity and then activate the connection for use within Pi-QLens.

11.4.2 Add Users and Assign Roles

1. In the main navigation, click **Admin** and then **Users**. You should see the user administration screen.
2. Click **Add User**.
3. Enter the new user's name, email, and temporary password or invitation details.
4. In the role dropdown, choose **Admin**, **Steward**, or **Viewer**.
5. Click **Save**. You should see the new user appear in the user list.
6. Sign out and sign in as that new user to confirm the correct menu options and permissions appear.

11.4.3 Apply Example Rules

1. In the top navigation, click **Rules**. You should see the rule library and the applied-rules area.
2. Select a dataset or table you want to monitor, for example `SALES_ORDERS`.
3. Click the rule you want to apply from the rule library. A configuration dialog opens.
4. For the first rule, choose a **Completeness** check and target the `customer_id` column. You should see the rule preview update to show the logic that will be executed.
5. Click **Apply Rule**. The rule should appear in the applied-rules section for that dataset.
6. Repeat the process for a **Uniqueness** rule on `order_id` and a **Freshness** rule on `updated_at`. Fill the appropriate values asked in different rules. example : threshold minutes , threshold days in freshness rules.
7. After each save, confirm the new rule appears in the table or dataset detail view so you can see that the policy is now active.

11.4.4 Run a Scan

1. Open the dataset or table detail page by clicking the relevant table from **Catalog** or **Quality**.
2. On the table page, locate the **Scan Now** click **Full Scan** or **Incremental Scan** button and click it.
3. While scanning **Scan Now** button turns as **Scan Disabled** this indicates a scan is running.
4. Wait for the run to complete. When it finishes,notification pops up **Scan completed**. The page should update with the latest scan time, number of checks executed, and a result summary.
5. If you are testing incremental behavior, choose **Incremental Scan** and verify that the watermark field is selected before you start the run.

11.4.5 Review Results and Failures

1. After the scan completes, Home / Dashboard will show result.
2. Review the summary cards to see how many checks passed, failed, or were skipped.
3. Click into the failed checks section to view the rule name, severity, and any row-level or validation details.
4. You can view granular results , by clicking on a particular table(in dataset section) > open full table page. Under rules you can even check logs of each rule.
5. Open **Evaluation** from the main navigation to see the quality score, trend, and coverage metrics for the monitored table or dataset.
6. Open Files from **Export Reports** to see generated outputs and export them as HTML or XLSX if needed.

11.4.6 Navigate Across the Main Pages

Use the top navigation to move through the product experience and see how each area behaves:

- **Home / Dashboard:** shows overall quality summary and latest health signals.
- **Connections:** lets you create, edit, test, and activate Snowflake connection profiles.
- **Rules:** lets you browse the rule library, configure rules, and apply them to datasets.
- **Evaluation:** shows quality scores, coverage trends, and recent scan results.
- **Reports:** displays generated reports and export options.
- **Catalog / Quality / Datasets:** lets you browse databases, schemas, tables, and drill into table-level details.
- **Admin → Users:** lets you add, edit, and remove user accounts and assign roles.

12. Support & Contact Information

12.1 Technical Support

For any issues, questions, or bug reports related to Pi-QLens, please reach out to the Pi-Snow Support Team using the contact information below:

- Email: Admin@Pibythree.com, Alliance@Pibythree.com

12.2 Business & Sales Inquiries

For questions related to the Pi-Snow roadmap, licensing, or upcoming Pi-QLens features:

- Contact: Alliance@pibythree.com
- Website: www.pibythree.com

12.3 Documentation & Feedback

To access the latest version of this document or to provide feedback on the application's performance:

- Documentation Portal: Pi Accelerators by PibyThree: Boost Cloud & AI Growth.