AXTRIA

INGENIOUS INSIGHTS

IC Reports Framework

Presented By

Date: 30-Apr-2025

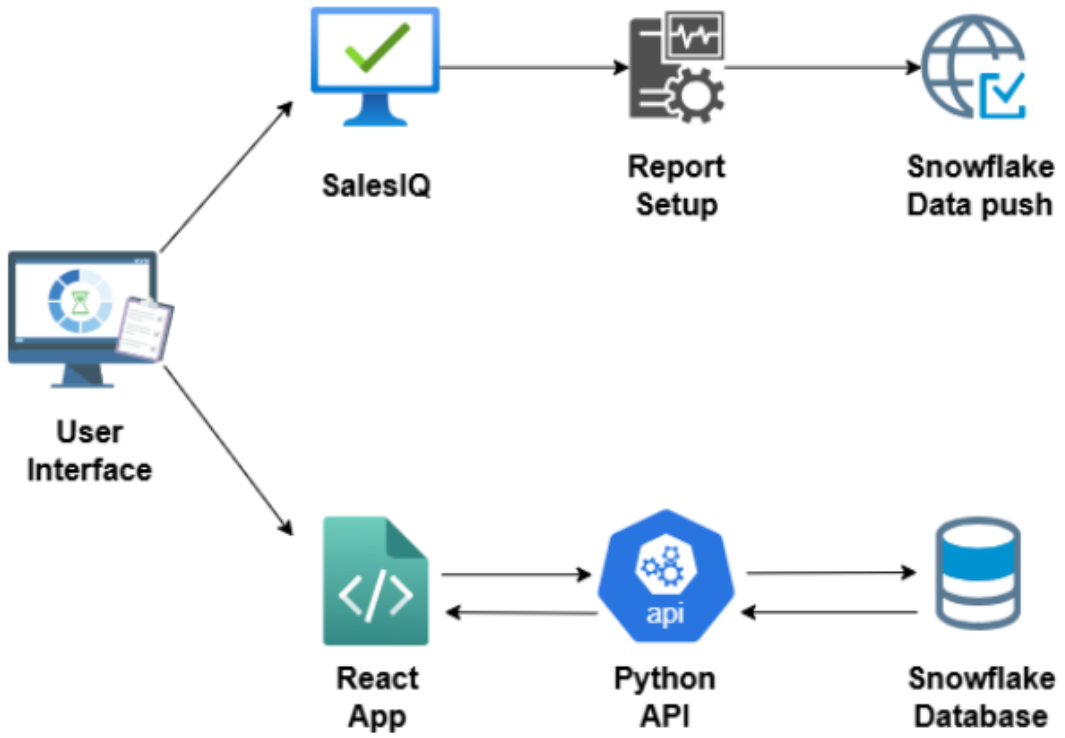
Agenda

- Framework Overview
- Components
- Rollout Strategy
- Flexibility and Ongoing Maintenance
- Skillset Required and Pre-requisite
- Benefits, Effort Estimation and Cost
- Compare and Contrast with Power BI
- Transition Plan from PBI to ReactJS

Framework Overview

- **The Framework** is built on the powerful **React.js** framework and leverages **Highcharts** for data visualization. This approach ensures high performance, interactive reports, and seamless user experience.
- React.js is a JavaScript library for building user interfaces, while Highcharts is a charting library that provides a wide variety of interactive charts and graphs

Architecture



Remarks: *It integrates seamlessly with the existing IC configuration and report setup. There is no need for additional configuration or changes to the current setup of 'Report Setup' in SalesIQ.*

Components

The framework uses a component-based architecture for efficient report creation. Reports are broken into reusable React Components, simplifying management and updates. This modular approach enhances maintainability and scalability, with components applicable to both SalesIQ web and UFE.

Steps to Follow

Current Components

HeaderSection1

- Showcases essential report metadata like report title, reporting period, user information, and logos

HeaderSection2

- Shows essential report metadata and allows users to switch between different views, e.g. ‘My View’, ‘Team Summary’

RHSPanel

- To present specific content related to the report’s details

AdditionalNotes

- To display list of additional notes

Components (cont.)

DataTable

- For rendering tables with various interactive features, allowing users to view and manage data efficiently

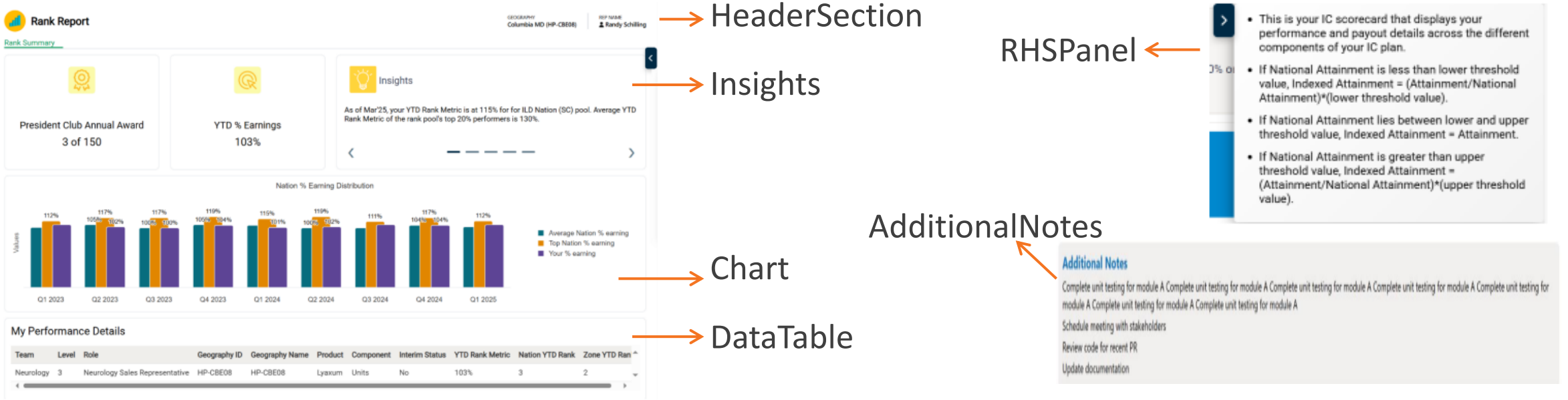
Insights

- Provides a dynamic and interactive way to display key insights related to the report

Chart

- To display various charts using Highcharts, allowing users to visualize data in an interactive way

Example



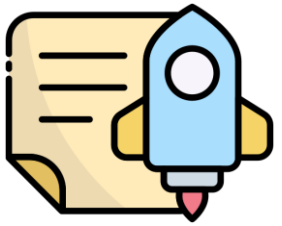
Rollout Strategy

Identify Data Sources

Develop Report Components

Integrate with IC Configuration

Test the Report



Deploy the Report

Examples of Current Implementations

BI-US Pilot: A pilot report for the Business Intelligence team in the US. The following React reports have been developed:

- Weekly Performance Report
- Scorecard
- Goalcard
- Rank Report
- HCO HCP Report
- What If Calculator

Ferring SIQ Implementation: A report for the Ferring SIQ project. The following React reports have been developed:

- Scorecard
- Goalcard

Flexibility and Ongoing Maintenance

If there are changes to the plan, such as new requirements or updates to the existing reports, follow these steps:

Assess Impact

Update Component

Test Updates

Inform all stakeholders about change

Ongoing maintenance involves regular updates to the framework and reports to ensure they remain efficient and accurate. This includes:

Monitor Performance of Reports

Address Issues

Update Framework

Provide Training

Skillset Required and Pre-requisites

Associates working on these reports must have the following skills:

- Knowledge of IC Reports
- React.js Skills
- Highcharts Skills
- Snowflake Knowledge
- Python for APIs
- Problem Solving
- Communication Skill

Before starting the development of new reports, ensure the following prerequisites are in place:

- iPad (For Testing)
- Salesforce Org Details
- Server Details
- IIS Server Enabled
- Report Data Published
- Node.js Installed on Server
- SMFT Credentials
- Snowflake Credentials
- Domain to Host Python API and React App

Benefits, Effort Estimation and Cost

High Performance

Cost-Effective

Look and Feel

Easy to Customize and Reuse

Implementation Effort (when all Reports are in Legacy Setup):

Task	Effort
Initial Setup (IT Tasks)	5
Type (Per Report Effort)	**Effort**
React Report with DTR report setup (as per the components present in Library)	5
React Report with Non DTR report setup (as per the components present in Library)	6
React Report with small changes	7
React Report with medium changes	10
React Report with complex changes	On Demand

Changes	Category
Font and color changes	Small
New chart configuration (time for POC)	Medium
Calculation logic change (IC Calculator)	Complex

Implementation Effort (when Reports are migrated from Golden Image with no Changes)

Task	Effort
Initial Setup (IT Tasks)	5
Type (Per Report Effort)	**Effort**
React Report with DTR report setup (as per the components present in Library)	6
React Report with Non DTR report setup (as per the components present in Library)	6

Maintenance Effort	Cost Implications
Any changes or enhancements will be managed and implemented by React Development team	• Requires two dedicated resources- front-end (React) and back-end developments • Infrastructure access, server access, snowflake access, IIS admin access

Compare and Contrast with Power BI

Feature	React (With Components)	Power BI Embedded
Export to PDF/Excel	✅ Included in **Datatable** component	✅ Native
Print Views	❌ Currently not supported.	✅ Available out-of-box
Custom Format Downloads	✅ Advanced filter functionality to the **Datatable** component	⚠️ Limited to standard formats
Chart Variety	✅ **Chart** component	✅ Wide but predefined
Custom Design	✅ Full freedom (branding, interactions)	⚠️ Customization limited to themes
Dynamic/Interactive UIs	✅ 100% possible	✅ Built-in for standard interactions
Hover Details	✅ Fully customizable	✅ Native tooltips
Drill-down Navigation	✅ You control logic (e.g., via router/state)	✅ Built-in
Cross-chart Filtering	✅ If implemented across components	✅ Ready-made
Row-Level Security	✅ Possible via backend logic	✅ Native RLS support
User-Specific Filters	✅ We're receiving user-specific filtered data in React from Snowflake.	✅ Supported

Compare and Contrast with Power BI (cont.)

Feature	React (With Components)	Power BI Embedded
Whitelisting/IP Controls	✅ We can handle it in salesforce	✅ Controlled via Azure
Custom UX & Layout	✅ Pixel-perfect freedom	⚠️ Limited to PBI visual framework
Dev Time	🚧 Relatively high	🧩 Low, mostly configuration
Reusability	✅ Each component is reusable.	⚠️ Limited reusability/custom visuals
Embedded Analytics Flow	✅ Seamless with React routing/state	✅ Via iframe or API
Filter Panels	✅ Custom design, can save state	✅ Slicers available
Bookmarks/Views	✅ Can build via user config saving	✅ Native support
Cross-Filtering	✅ Possible with state sync logic	✅ Native
Per-user config (e.g., column select)	✅ Easy via JSON configs/state mgmt.	⚠️ Complex via bookmarks/APIs
White-labeling	✅ Fully skinnable	⚠️ Limited
Multi-tenant Custom Logic	✅ Possible with smart component design	⚠️ Needs complex Power BI setups

Transition Plan from PBI to ReactJS

Assessment of Existing PBI Report

Review Power BI report to understand data, KPIs, visuals, calculations, and filters.

Identify Data Sources

- Identify Power BI data sources.
- Ensure data structure compatibility with ReactJS.

Report Configuration

The IC Team will handle all Power BI setup changes, report conversions, and data structure updates.

React Development Phase

Fetch Snowflake data using Python APIs and replicate the Power BI layout in ReactJS with components.

Testing and Validation

Test data rendering & UI performance with large datasets, ensuring a responsive, user-friendly experience on web & iPad.

Report Deployment Strategy	Follows the existing process
Multi-Tenancy Considerations	Follows the existing process
Customer Customization Capability	Includes business team-driven data configurations, these changes can be easily managed based on business requirements

Do you have a question for us?

AXTRIA
INGENIOUS INSIGHTS