

# KPIT

21-Aug-2020

# Architecture Diagram Simulink Data Dictionary

Document ID: NAV\_SDD\_DD

Version:2.4



Powertrain Practice

KPIT Technologies Ltd.

Pune, India

# Document Particular(s)

## Document Control

© KPIT Technologies Ltd. All rights are reserved. This document contains proprietary information meant only for the intended recipients. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of KPIT Technologies Ltd. (Hereafter referred to as 'KPIT').

## Revision Log

Sr.No.	Description	Version	Date	Prepared by	Approved by
1	Initial Version	1.0	17-May-19	Jayshri Burde	Ritesh Goyal
2	Added flow diagrams for current sprint(sprint 2) requirement	1.1	03-Jun-19	Jayshri Burde	Ritesh Goyal
3	Updated flow diagrams for current sprint(sprint 3) requirement	1.2	14-Jun-19	Jayshri Burde	Ritesh Goyal
4	Updated and added flow diagrams for current sprint(sprint 4) requirement	1.3	27-Jun-19	Jayshri Burde	Ritesh Goyal
5	Updated and added flow diagrams for current sprint(sprint 5) requirement	1.4	12-Jul-19	Jayshri Burde	Rishi Modi
6	Updated and added flow diagrams for current sprint(sprint 6) requirement	1.5	26-Jul-19	Jayshri Burde	Rishi Modi

# Document Particular(s)

## Document Control

© KPIT Technologies Ltd. All rights are reserved. This document contains proprietary information meant only for the intended recipients. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of KPIT Technologies Ltd. (Hereafter referred to as 'KPIT').

## Revision Log

Sr.No.	Description	Version	Date	Prepared by	Approved by
7	Updated and added flow diagrams for current sprint(sprint 7) requirement	1.6	10-Aug-19	Nikita Kasar	Jayshri Burde
8	Updated and added flow diagrams for current sprint(sprint 8) requirement	1.7	23-Aug-19	Nikita Kasar	Jayshri Burde
9	Updated and added flow diagrams for Sprint 9 requirements	1.8	6-Sep-19	Nikita Kasar	Rishi Modi
10	Updated as per Sprint 10 requirements	1.9	20-Sep-19	Nikita Kasar	Jayshri Burde
11	Added flow diagram for Find In Model functionality	2.0	04-Oct-19	Nikita Kasar	Jayshri Burde
12	Updated template	2.1	08-Nov-19	Nikita Kasar	Jayshri Burde

# Document Particular(s)

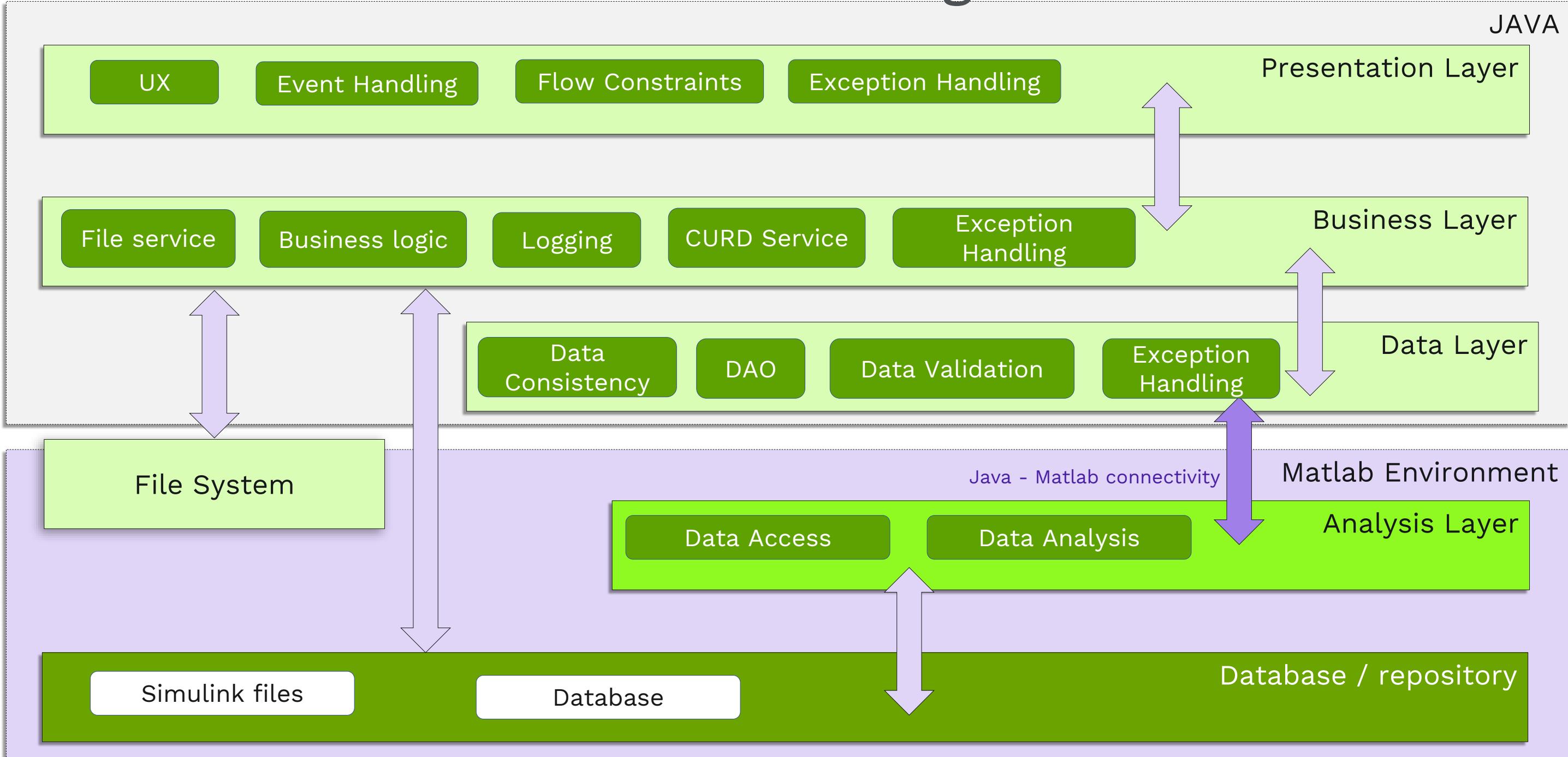
## Document Control

© KPIT Technologies Ltd. All rights are reserved. This document contains proprietary information meant only for the intended recipients. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of KPIT Technologies Ltd. (Hereafter referred to as 'KPIT').

## Revision Log

Sr.No.	Description	Version	Date	Prepared by	Approved by
13	Updated flow diagram for Excel Import Export requirement	2.2	29-May-20	Nikita Kasar	Shubhangi Mane
14	Added flow diagram for Apply CDF	2.3	26-Jun-20	Nikita Kasar	Shubhangi Mane
15	Added flow diagram for SLDD-95	2.4	21-Aug-20	Nikita Kasar	Shubhangi Mane

# Architecture Diagram



**Presentation Layer:**

Layer defining the user interface, providing user experience, capturing user actions and display required information. It is divided in 3 major modules.

**User Experience (UX) :**

- Define look and feel for the user interface based on various components like Icons/Images & Flash screens
- Provide methodology to play around with UI. e.g. resize of window, drag and drop of containers, hide of containers etc.
- Various methods to display fetched, searched and filtered information.

**Event Handling :**

- Responsible for capturing user actions on UI. Few operations
  - Capture click on any action button (like save, delete etc.)
  - Capture modification of any attribute in table
  - Capture selection of data elements, components etc.
- Provide captured information to business logic through list, callbacks etc.

**Flow Constraints:**

Constrain unwarranted actions/operations by user. Few example constraints are

- Do not allow open of new component before close of existed open component.
- Do not allow delete element operation for unsaved(dirty) component in GUI

**Exception Handling :**

- Handles all the unwanted or unexpected event, which occurs during the execution of the tool

**Business Layer:**

Responsible to take inputs and display response in presentation layer. It also communicate to Analysis layer through data layer for requesting information and actions

**Business logic:**

- Define operations such as filter , search, sort,
- Format data received from data layer and from presentation layer
- Render formatted information into UI

**CRUD Service:**

- Parse information related to CRUD operations like add, update , delete for data elements.
- Parse additional operation information (e.g. Input output compatibility check)
- Provide data flow to/from Data layer from/to business layer.

**File Service:**

- Communicate with file system to handle all the windows file system related operations like, Import Project, open project, close project etc.

**Logging:**

- Keep a trace of each and every operation perform by the user.
- Save any additional log info needed to understand tool's operation

**Exception Handling :**

- Handles all the unwanted or unexpected event, which occurs during the execution of the tool

Data Consistency

DAO

Data Validation

Exception  
Handling

Data Layer

## **Data Layer:**

Contains data, ensure consistency and validate data.

### **Data Consistency:**

- Ensure consistency of data transfer to and from the layer.

### **DAO:**

- Data Access Object
- Data persistency for the session

### **Data Validation**

- Validate user inputs with defined rules for various attributes
- Validate data being passed to sldd
- Validate data being rendered on UI

### **Exception Handling :**

- Handles all the unwanted or unexpected event, which occurs during the execution of the tool

**Analysis Layer:**

Interface with sldd/models, access data and analyze data.

**Data Access:**

- Fetch data from sldd
- Update data to sldd
- Fetch information from Simulink model
- Provide data in required format to data layer

**Data Analysis:**

- Analyze data from various sldd to perform specific operations e.g. :  
Input Output Compatibility
- Analyze information between sldd and model for consistency

# Data Dictionary Technology Stack

JAVAW

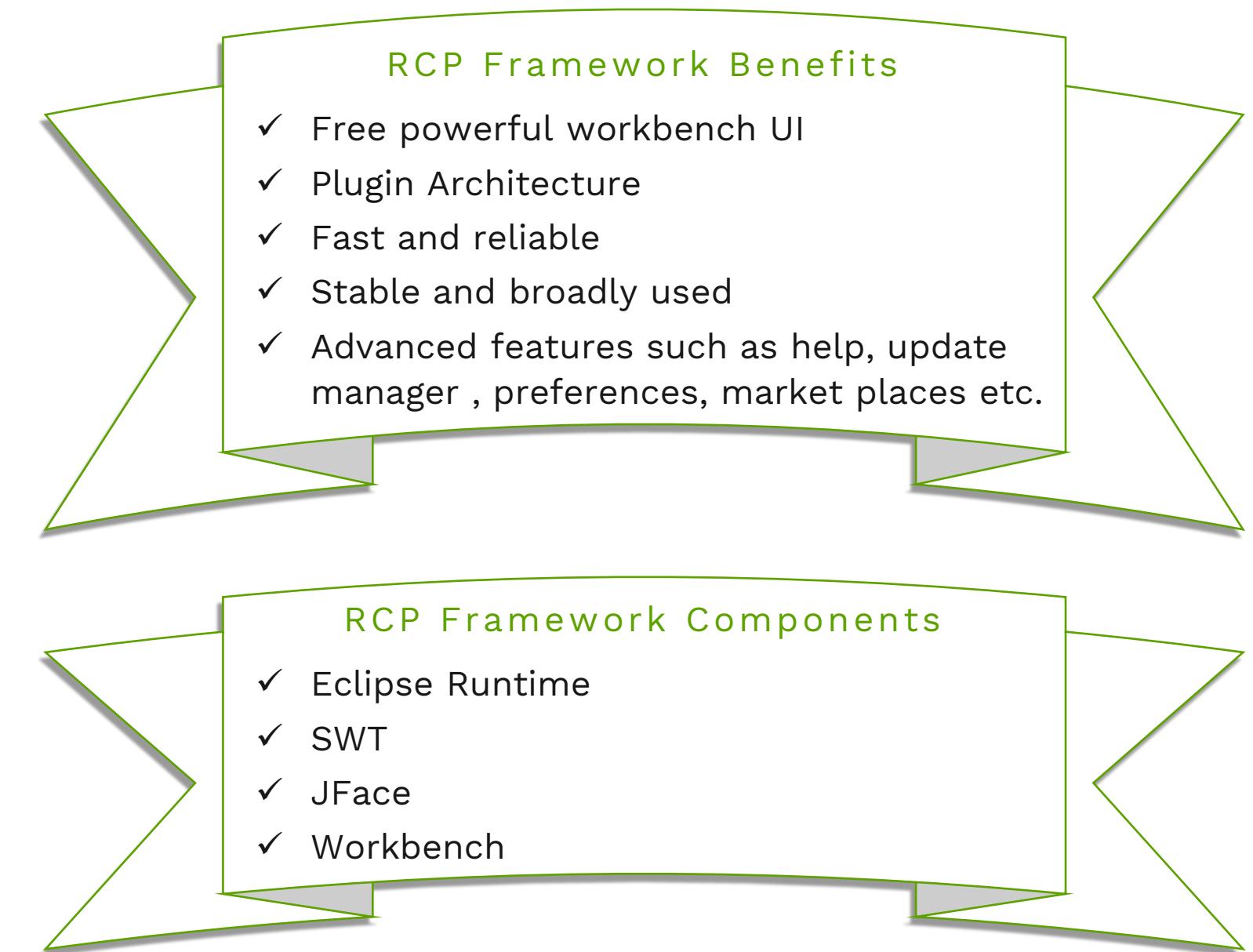
## Desktop Application



# Data Dictionary tool – RCP Framework

## ❖ Why RCP?

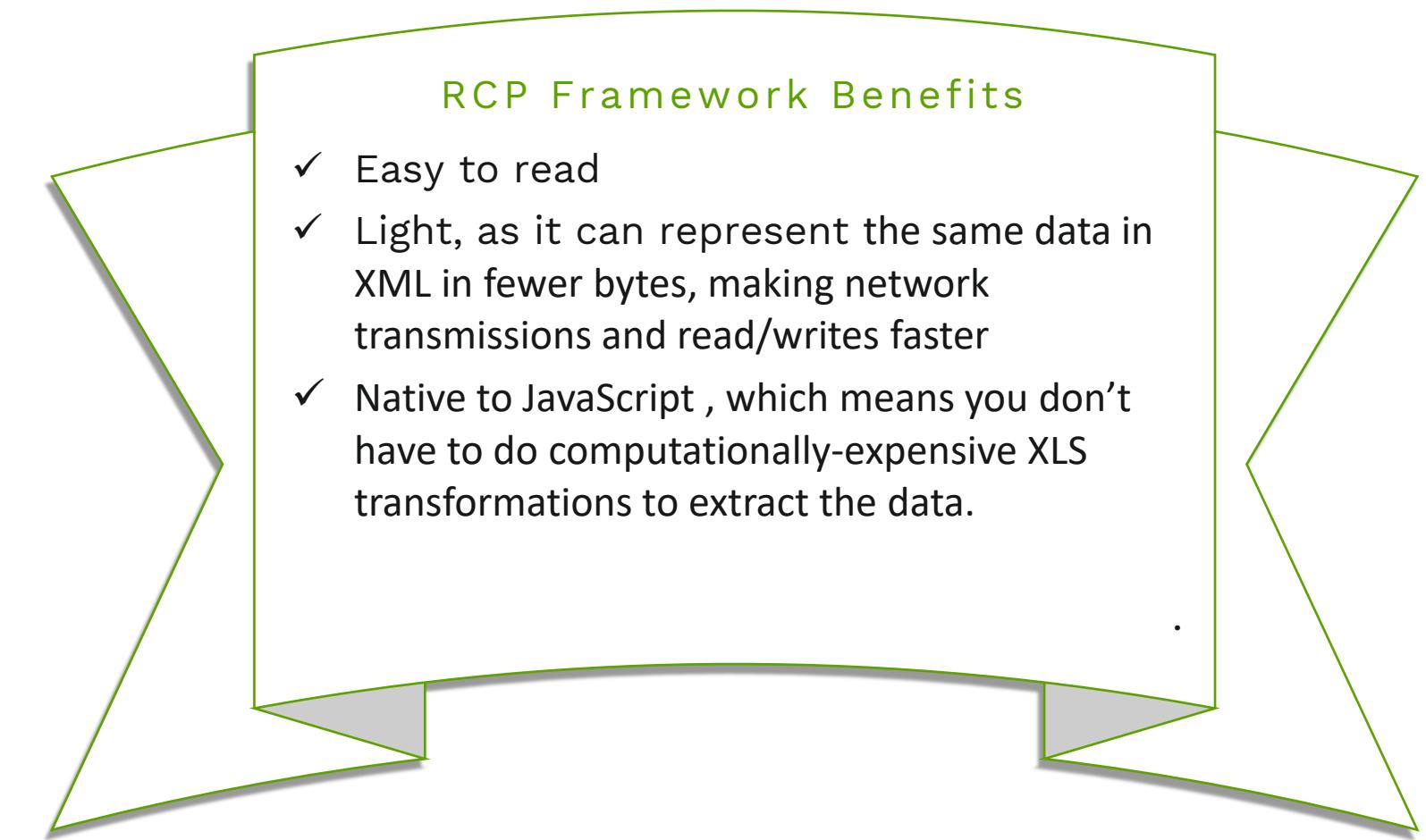
- All applications based on RCP guaranteed to have a native look and feel on all the supported execution platform(e.g. windows, MAC or Linux)
- RCP provides a set of basic functionality that is used or needed in many client applications a sophisticated help system, a well thought-out look-n-feel, a very efficient service oriented framework, window resizing and rearranging, Dock view etc.
- The application developed on top of RCP are portable and will run equally well on windows, MAC or Linux.



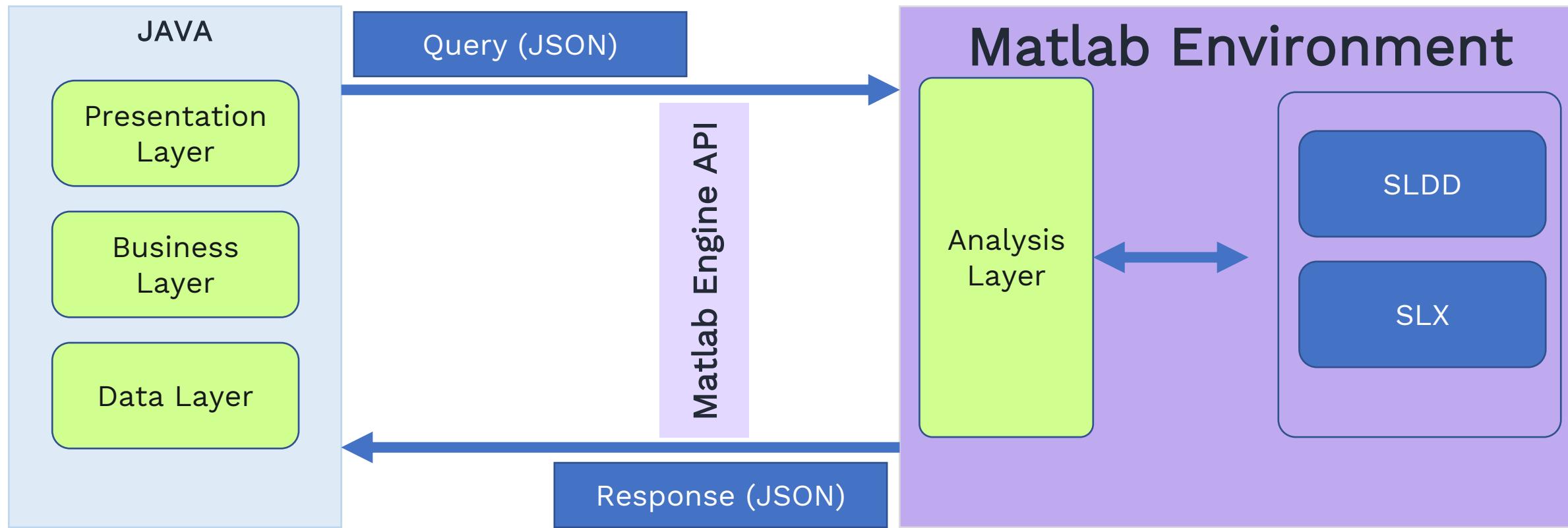
# Data Dictionary tool – JSON Format to exchange data

## ❖ Why JSON?

- JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data exchange format that is easy for humans and machines to read and write



# Java - Matlab connectivity



## Note:

- This GUI and MATLAB will be peer application. This will allow to utilize all benefits of MATLAB & JAVA during development.
- Java application will make a soft connection through MATLAB engine API and call MATLAB functions/Access data through queries.
- Standard queries will be defined during the implementation.

# Resources required

- **Development/Build environment :**

1. RAM : 4 GB or higher
2. OS : Windows 10 64-bit
3. Tools : Eclipse IDE Oxygen version with RCP plugin, SVNKit plugin installed, MATLAB R2020a Update2
4. Java version : JDK 1.8, JRE 1.8
5. System variable : PATH environment variable with MATLAB path set

- **User environment :**

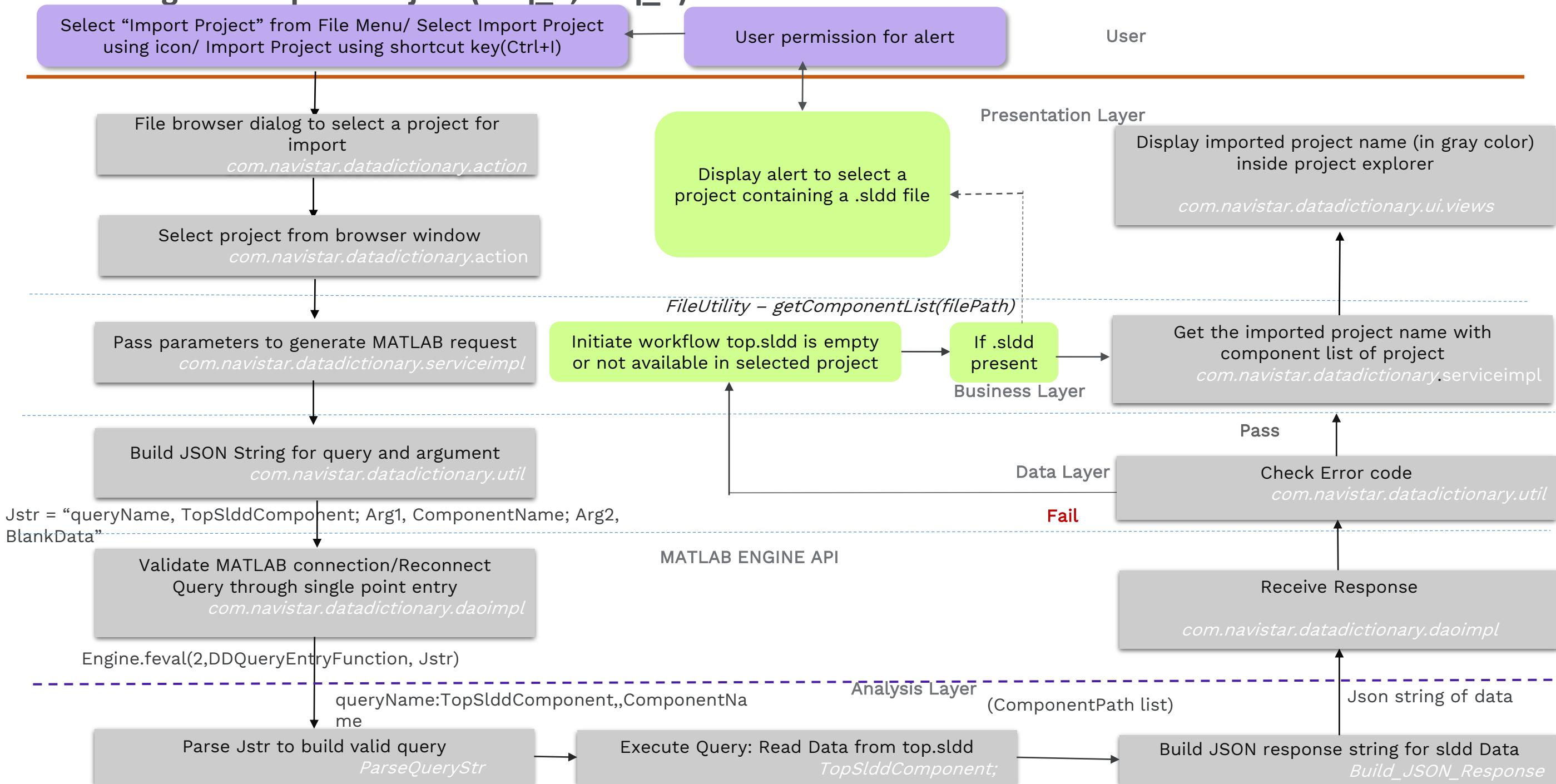
1. RAM : 4 GB or higher
2. OS : Windows 10 64-bit
3. Tools : MATLAB R2020a Update2
4. Java version : JRE 1.8
5. System variable : PATH environment variable with MATLAB path set

# KPI's

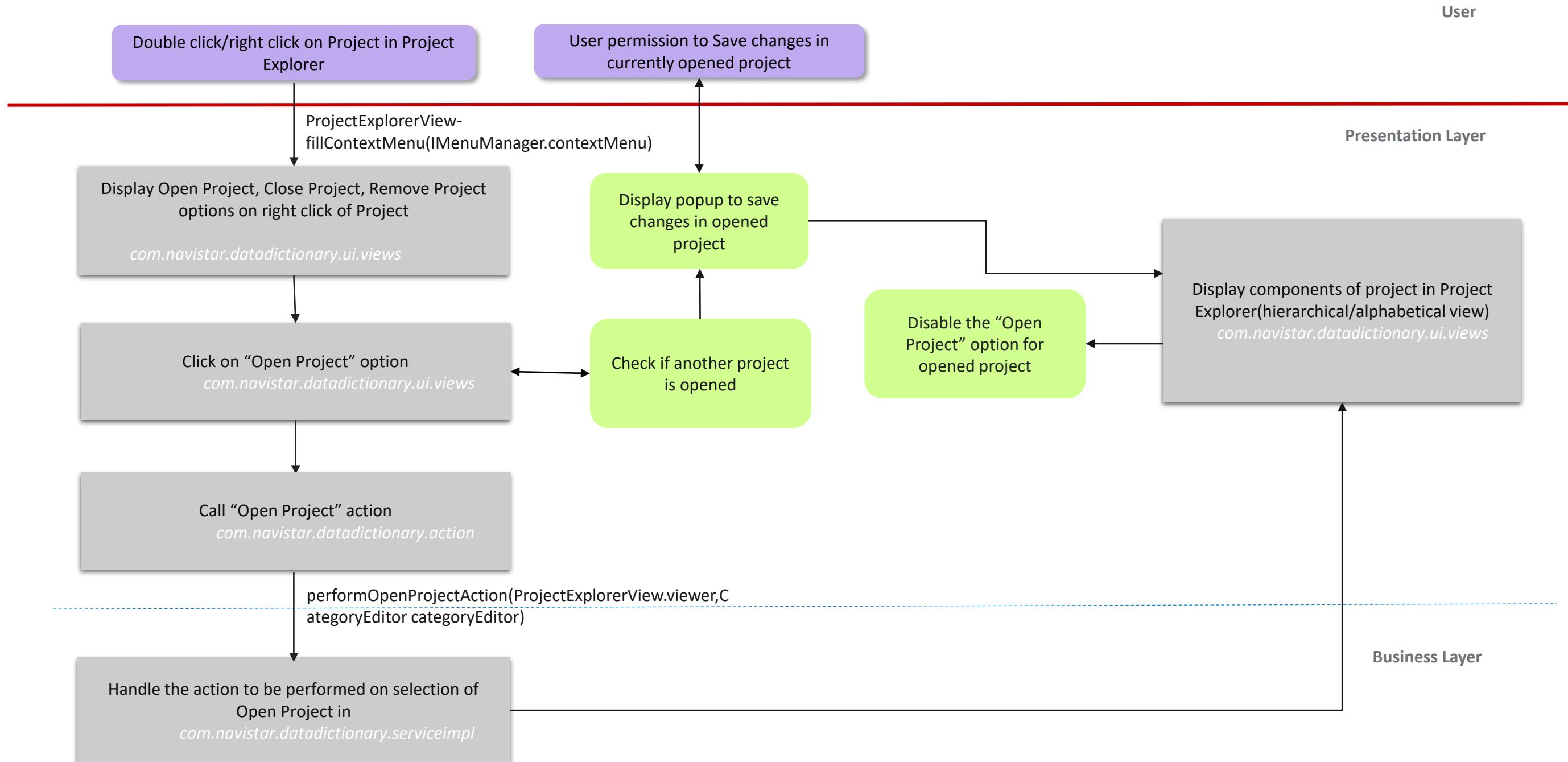
## Flow Diagrams



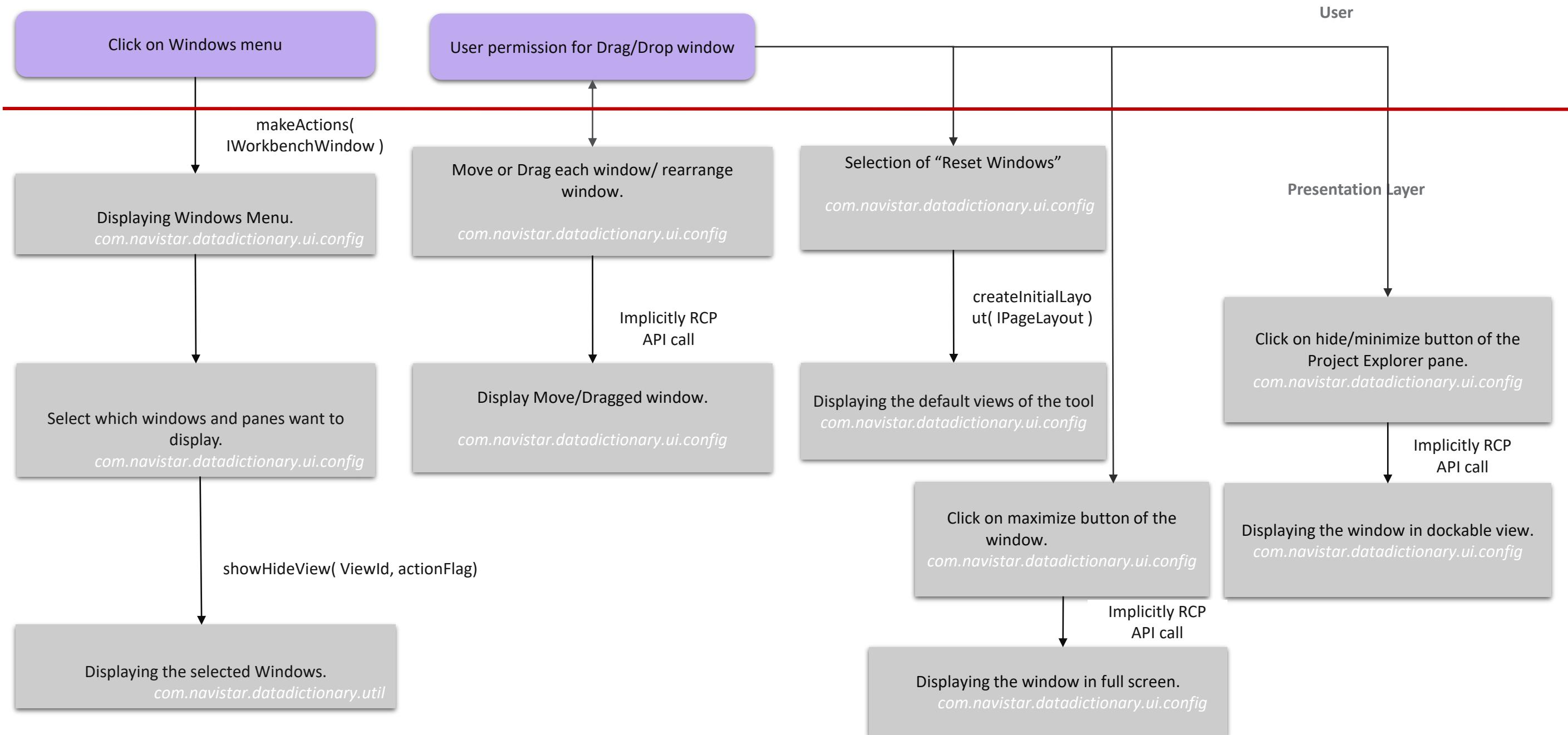
# Flow Diagram: Import Project (Req\_1, Req\_2)



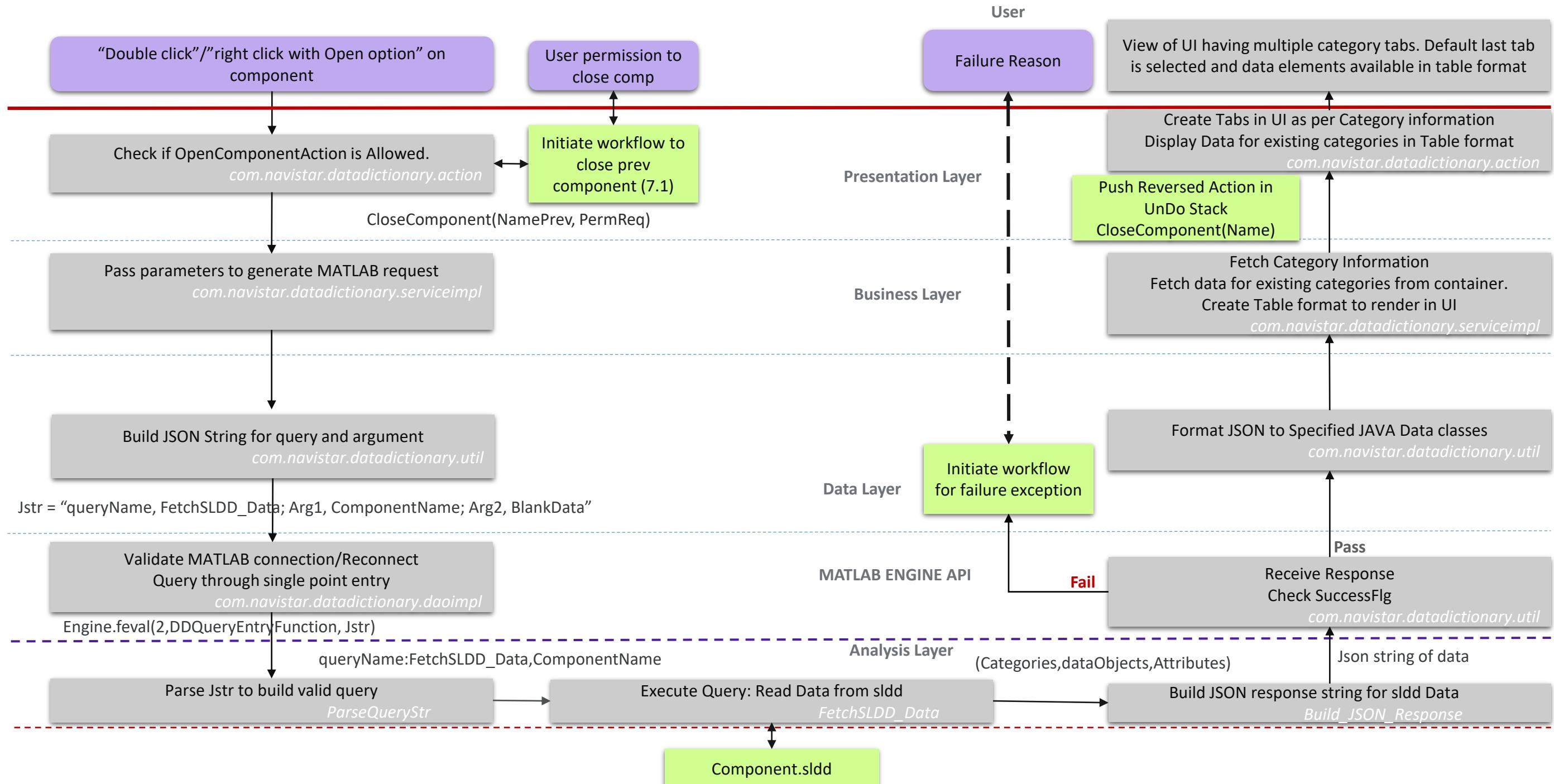
# Flow Diagram: Open Project(Req\_2.1, Req\_2.2, Req\_2.3, Req\_5)



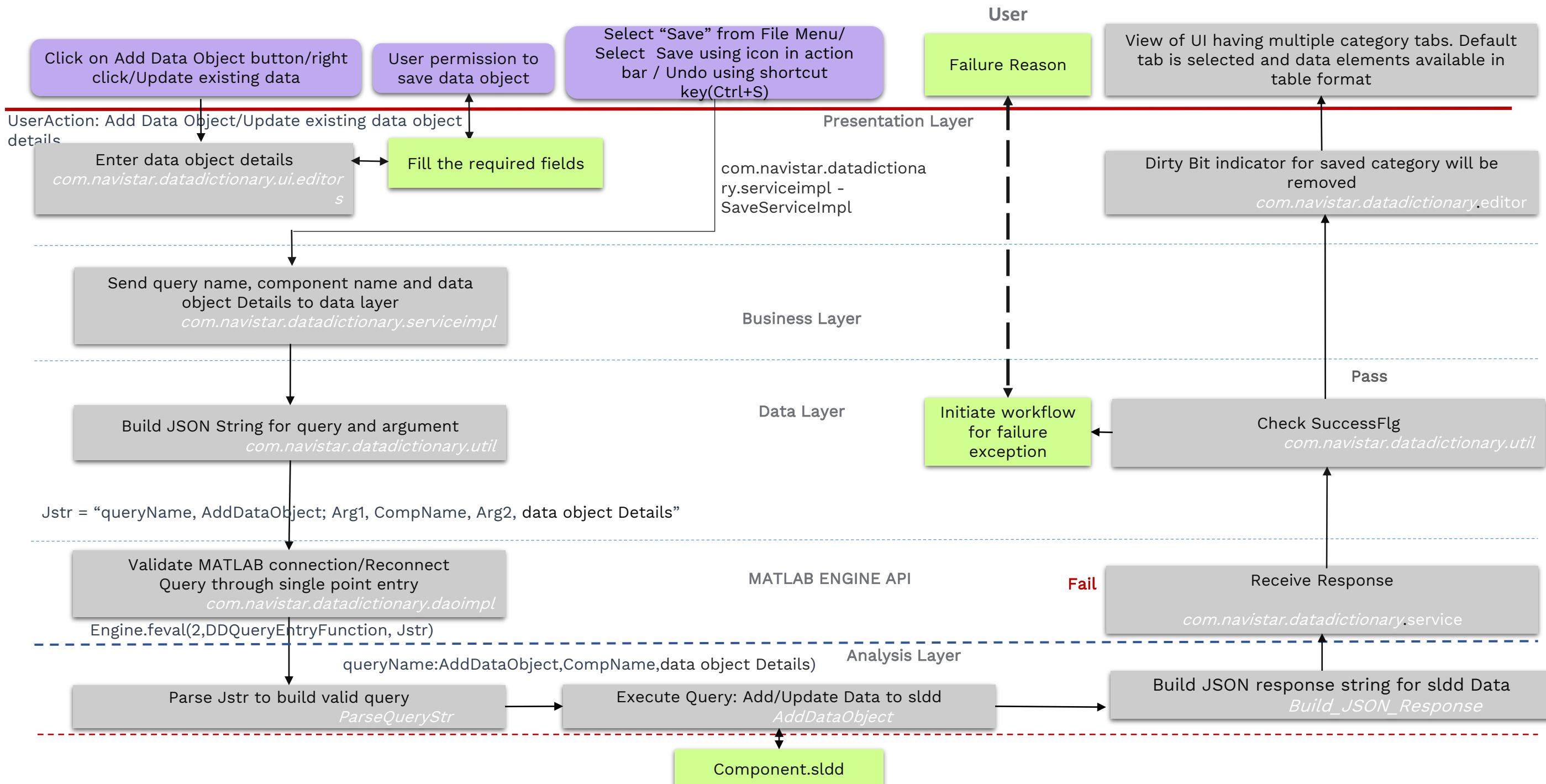
# Flow Diagram: Minimize, Maximize/ Reset/ Rearrange the Windows(Req\_27, Req\_28, Req\_29, Req\_30, Req\_32)



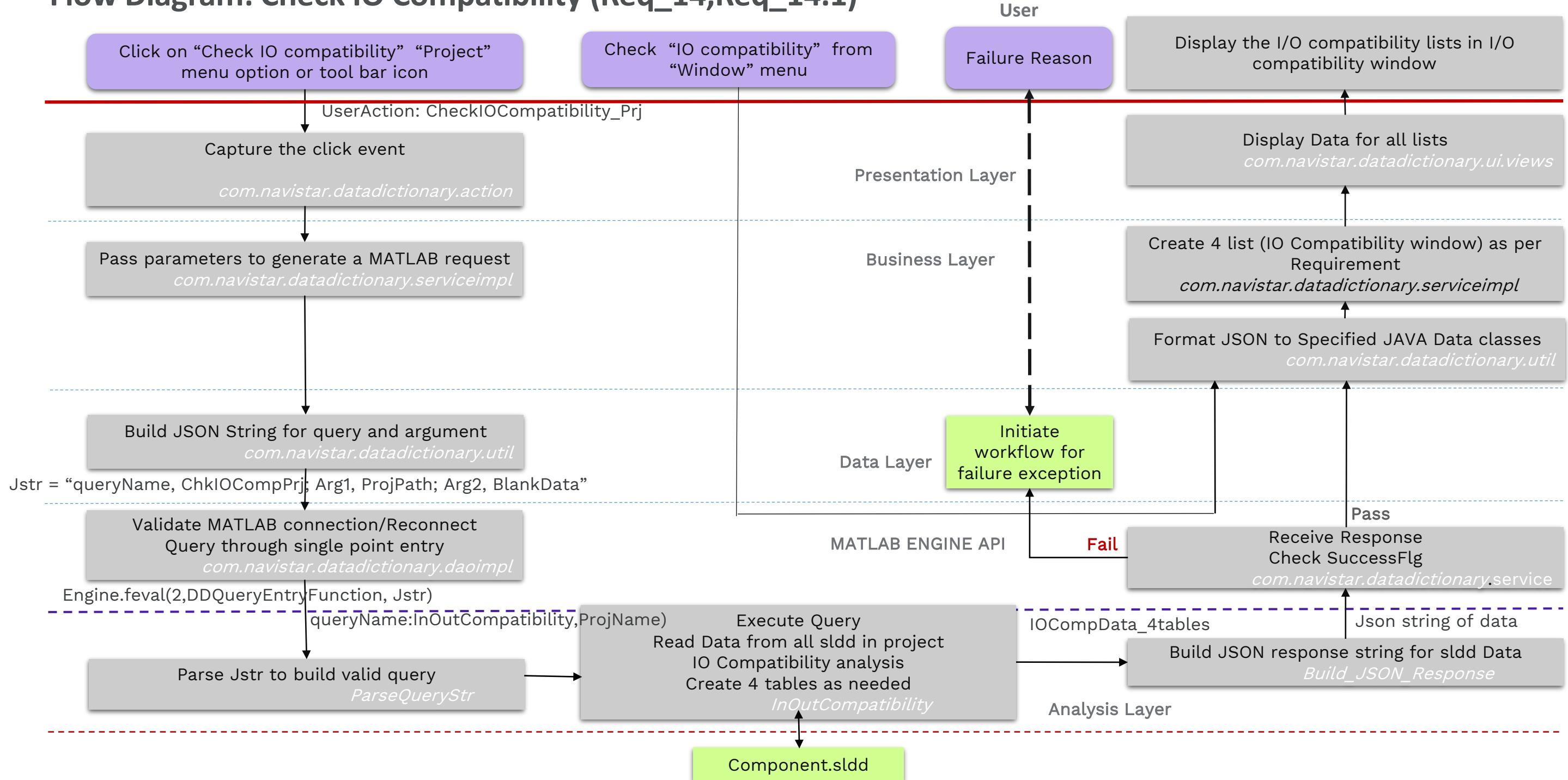
# Flow Diagram: Open Component (Req\_7, Req\_8, Req\_9)



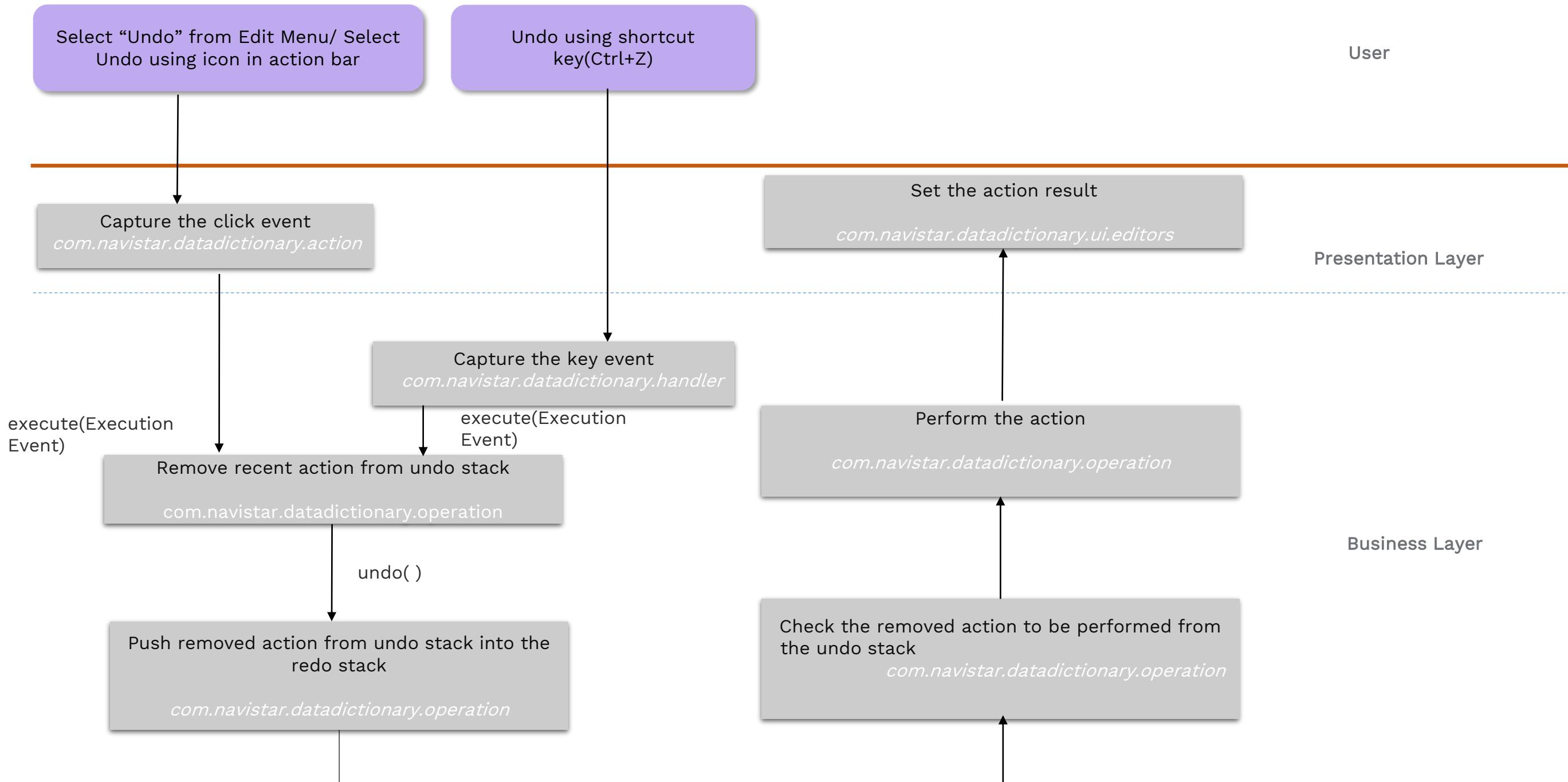
# Flow Diagram: Add Data Object (Req\_22)



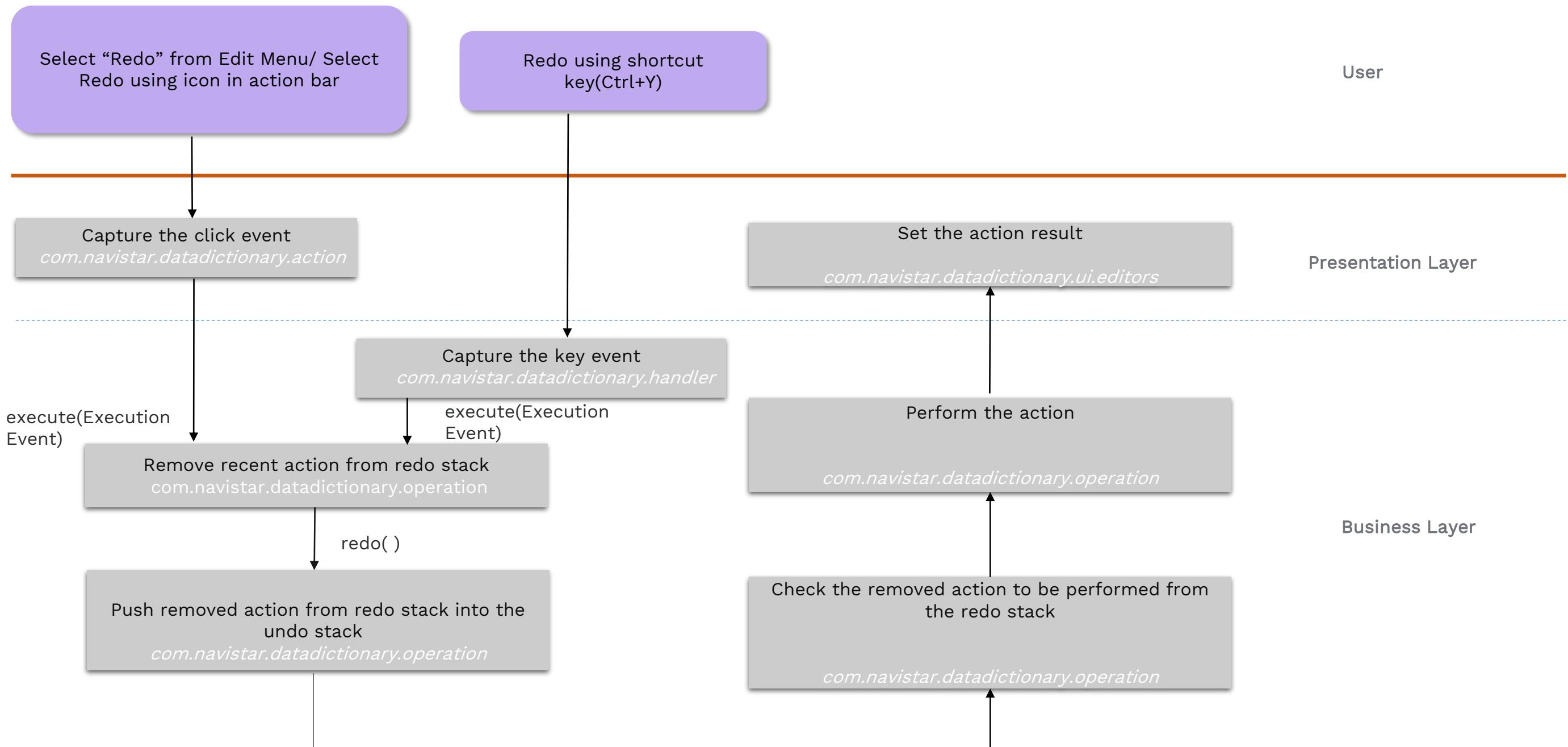
# Flow Diagram: Check IO Compatibility (Req\_14,Req\_14.1)



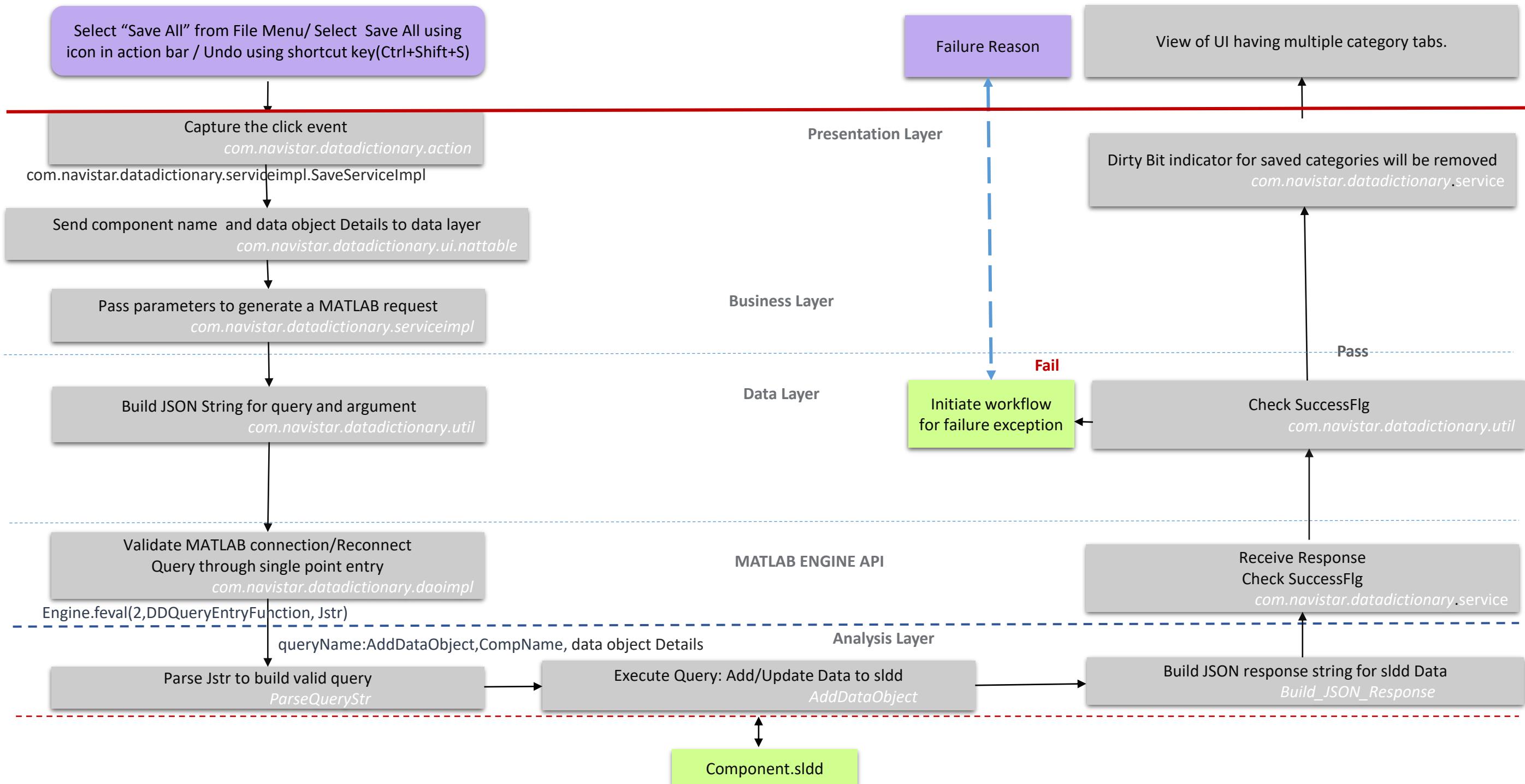
# Flow Diagram: Undo (Req\_26)



# Flow Diagram: Redo (Req\_26)



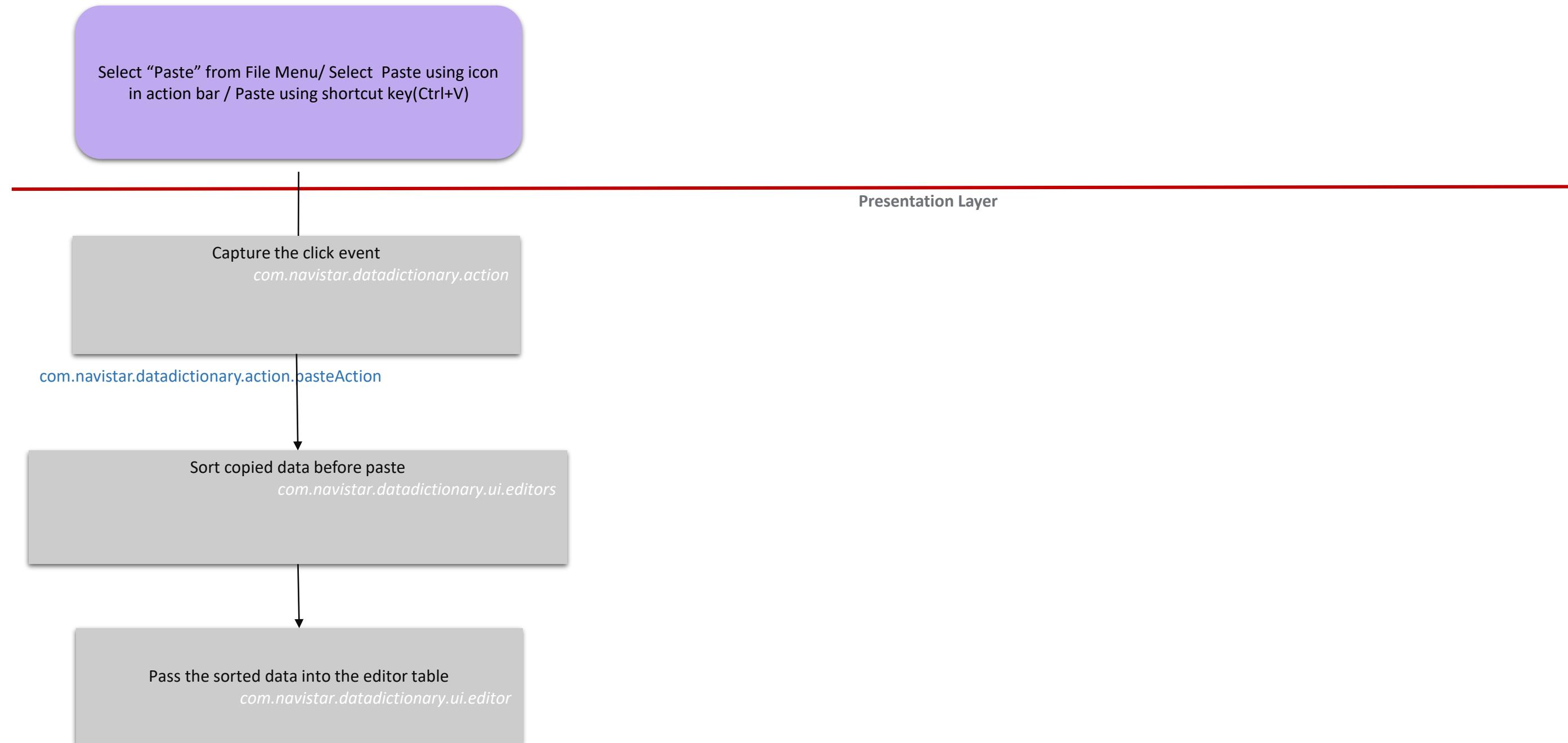
# Flow Diagram: Save All (Req\_25)



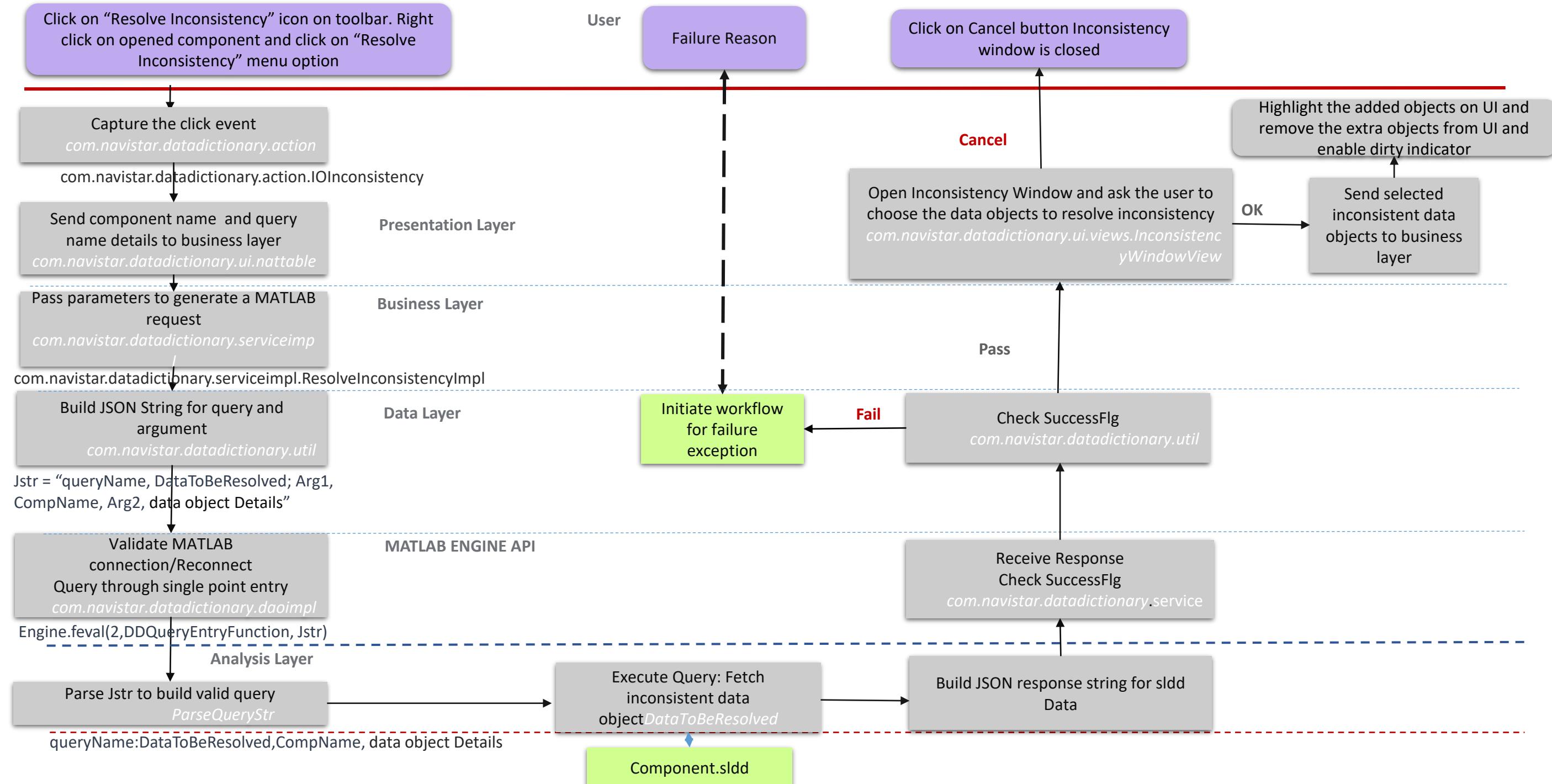
## Flow Diagram: Copy (Req\_26)



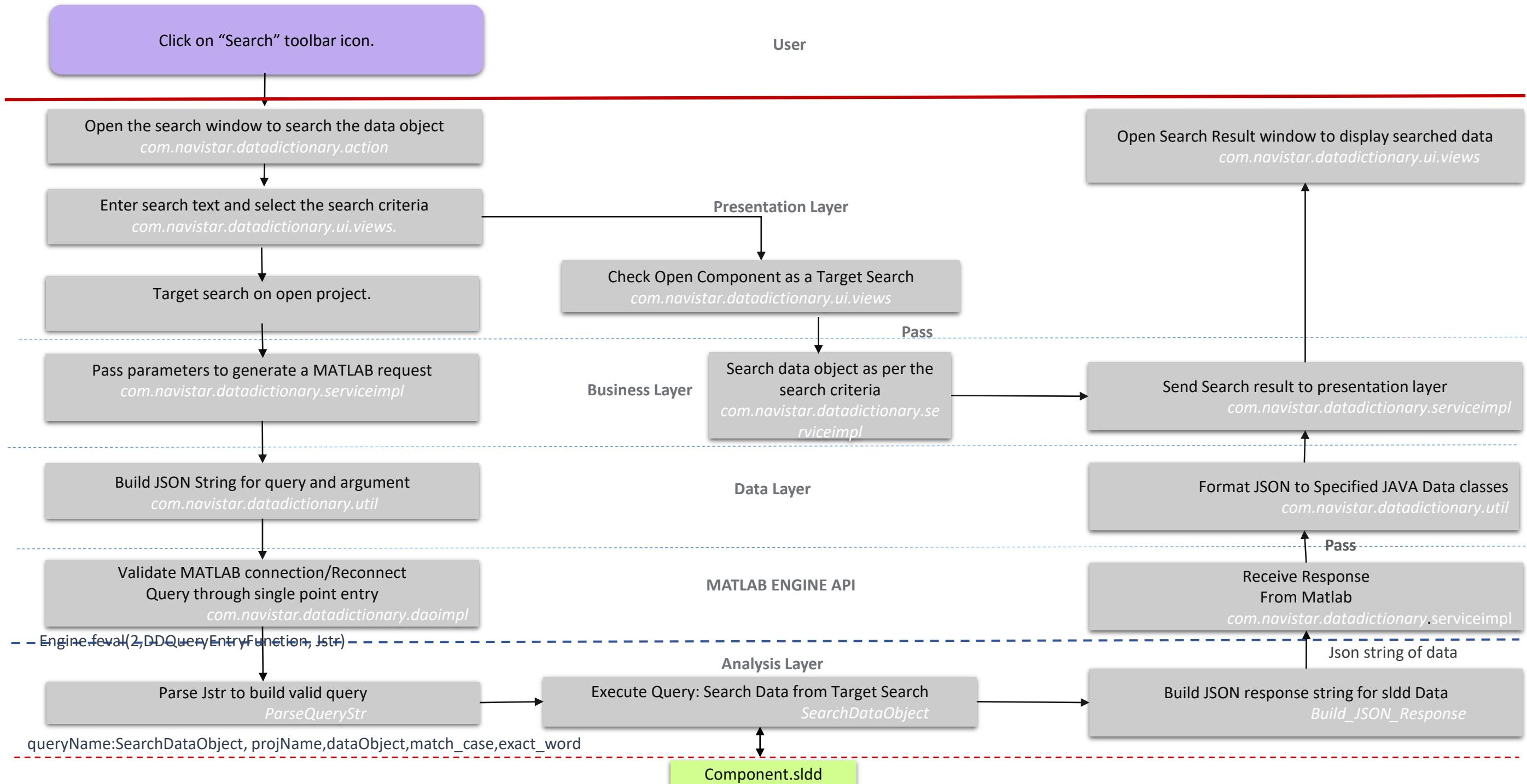
## Flow Diagram: Paste (Req\_27)



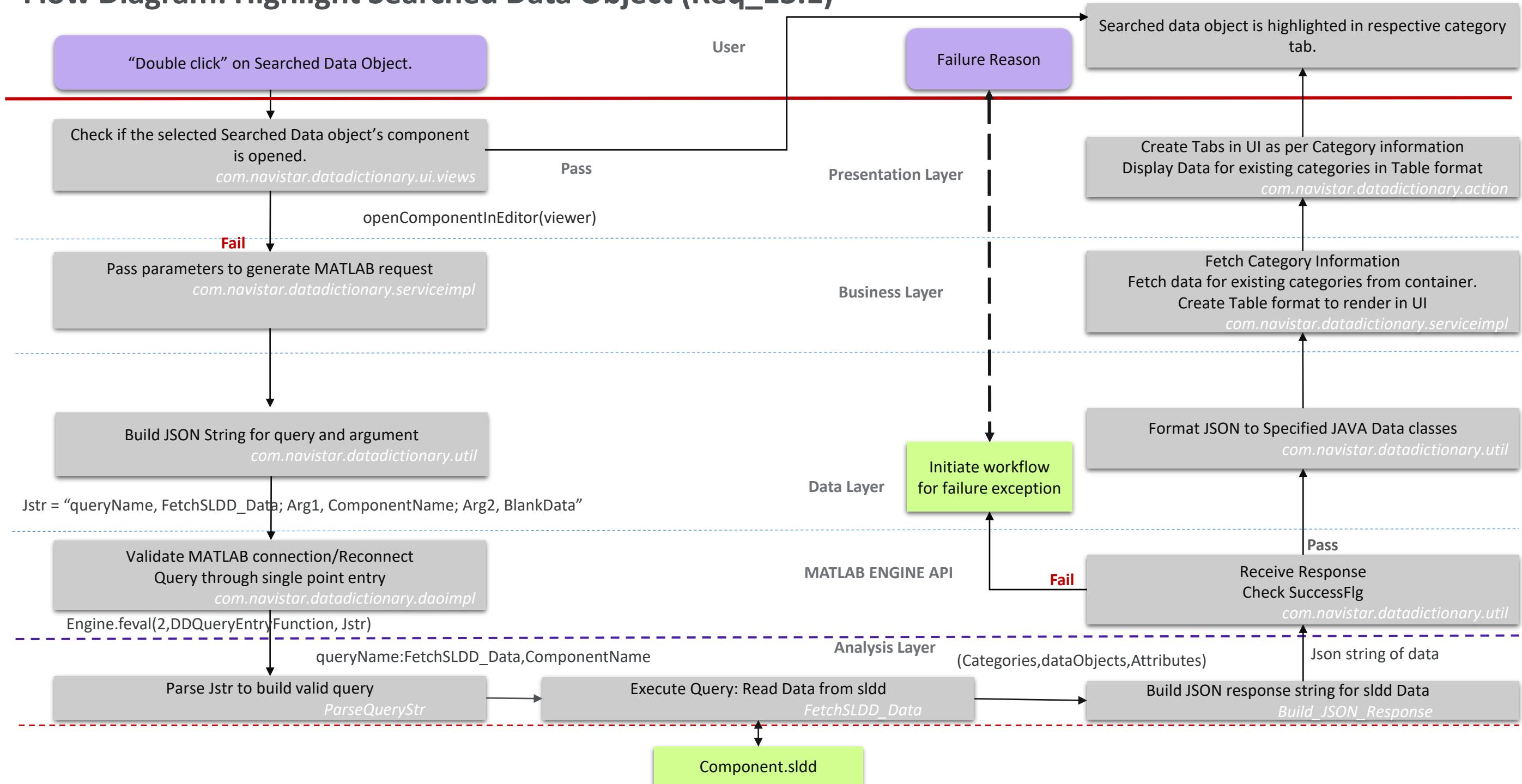
# Flow Diagram: Resolve Inconsistency (Req\_16, Req\_16.1)



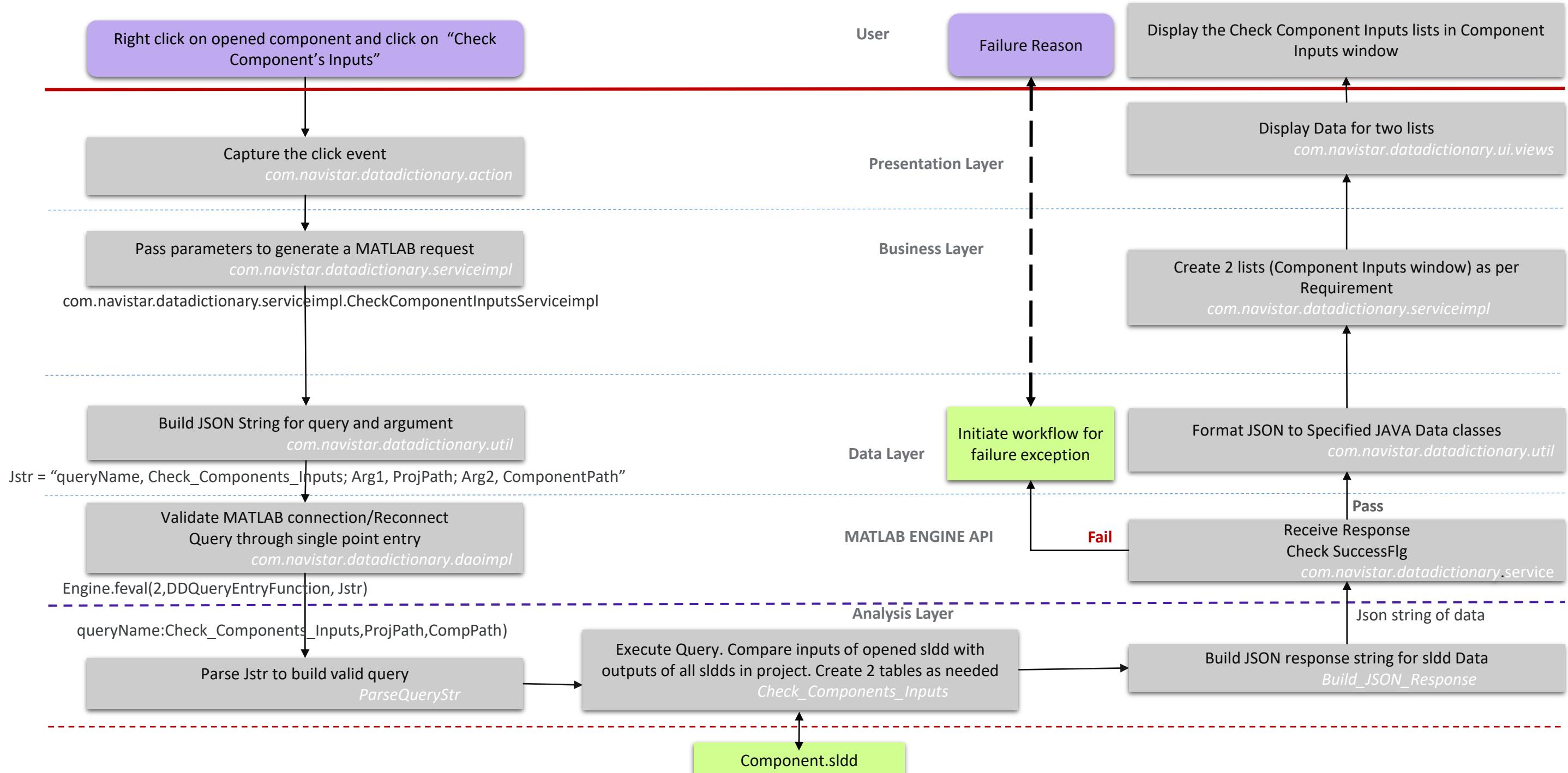
# Flow Diagram: Search Data Object (Req\_12, Req\_12.1, Req\_12.2, Req\_13)



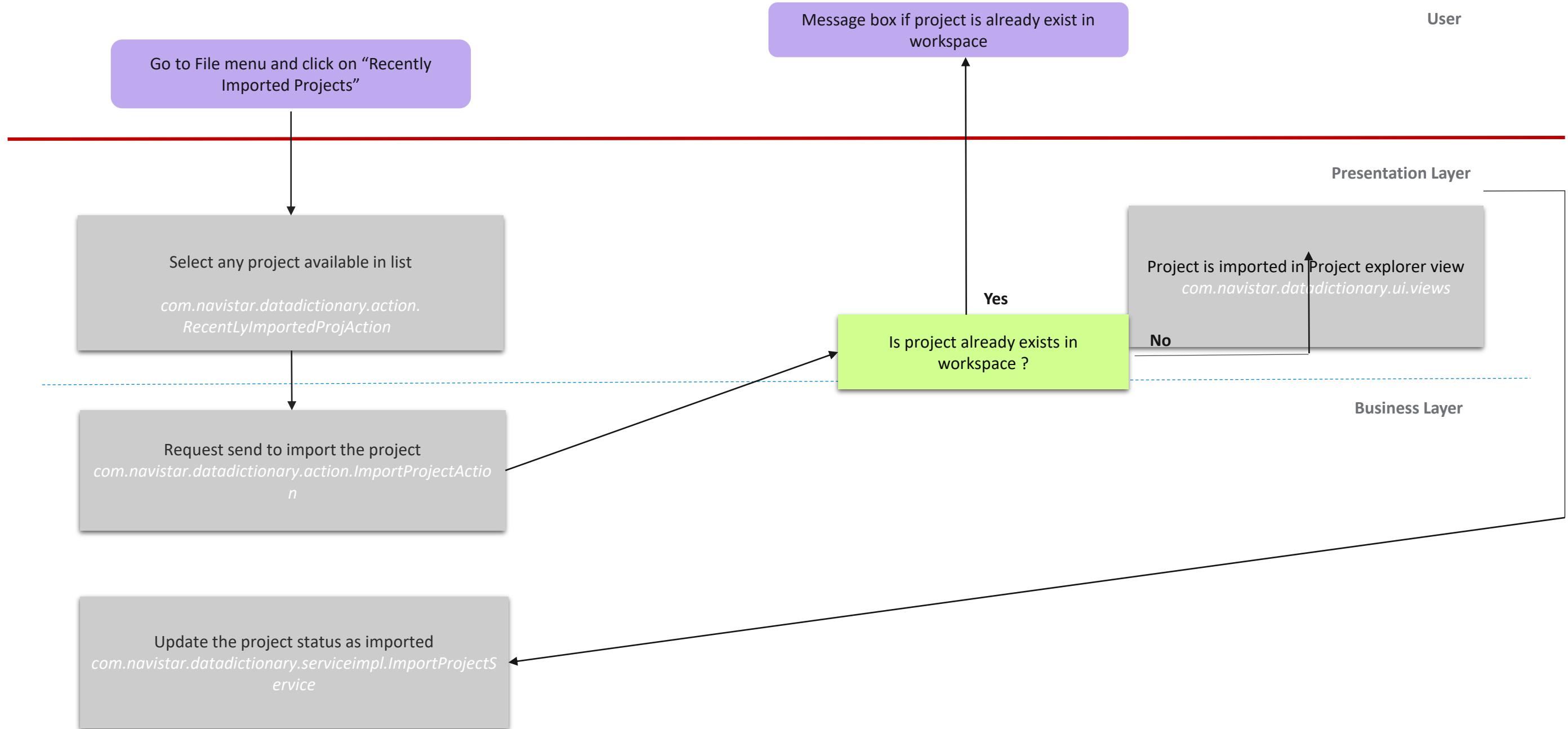
# Flow Diagram: Highlight Searched Data Object (Req\_13.1)



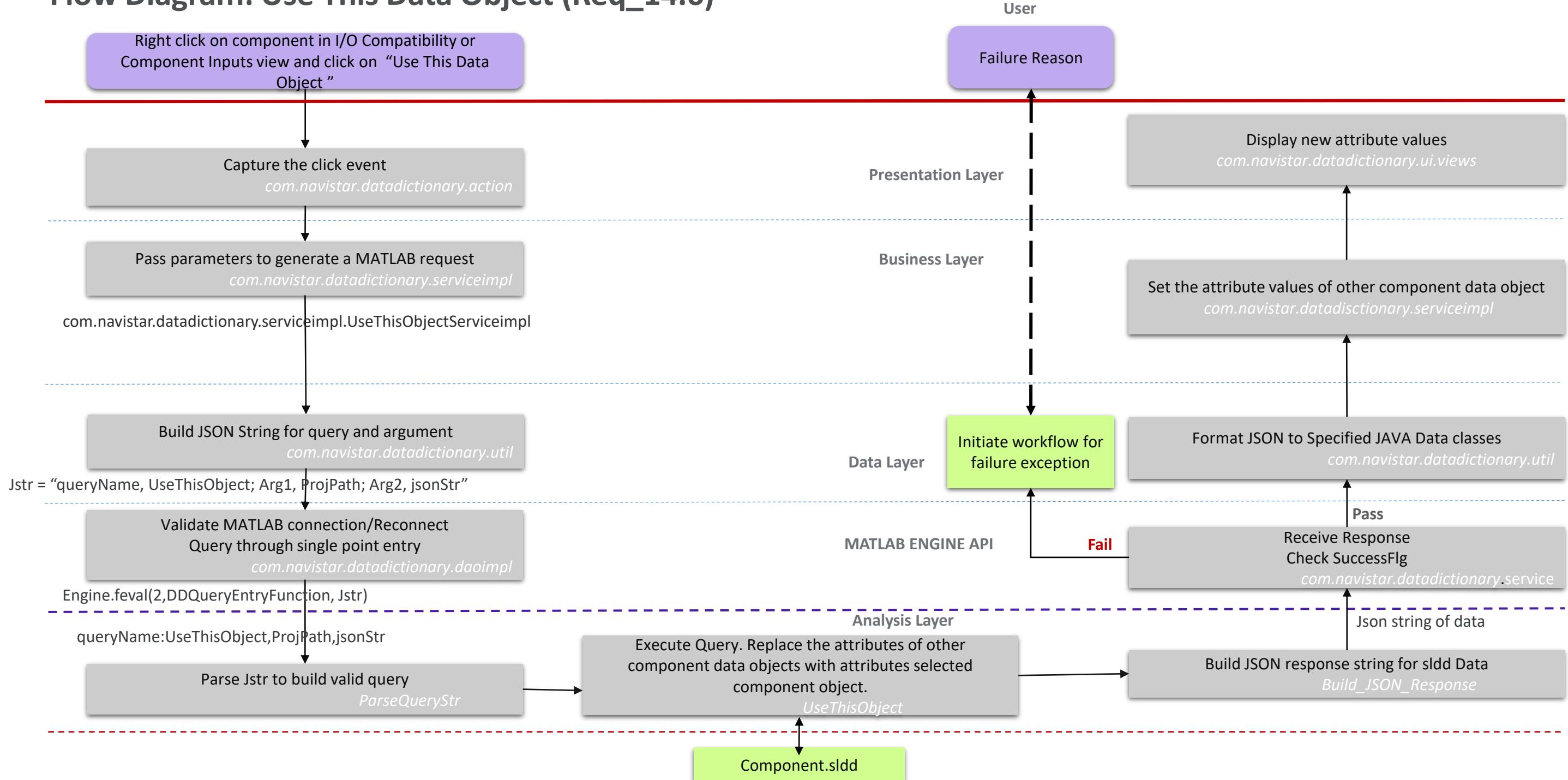
# Flow Diagram: Check Component's Inputs (Req\_14.4, Req\_14.5)



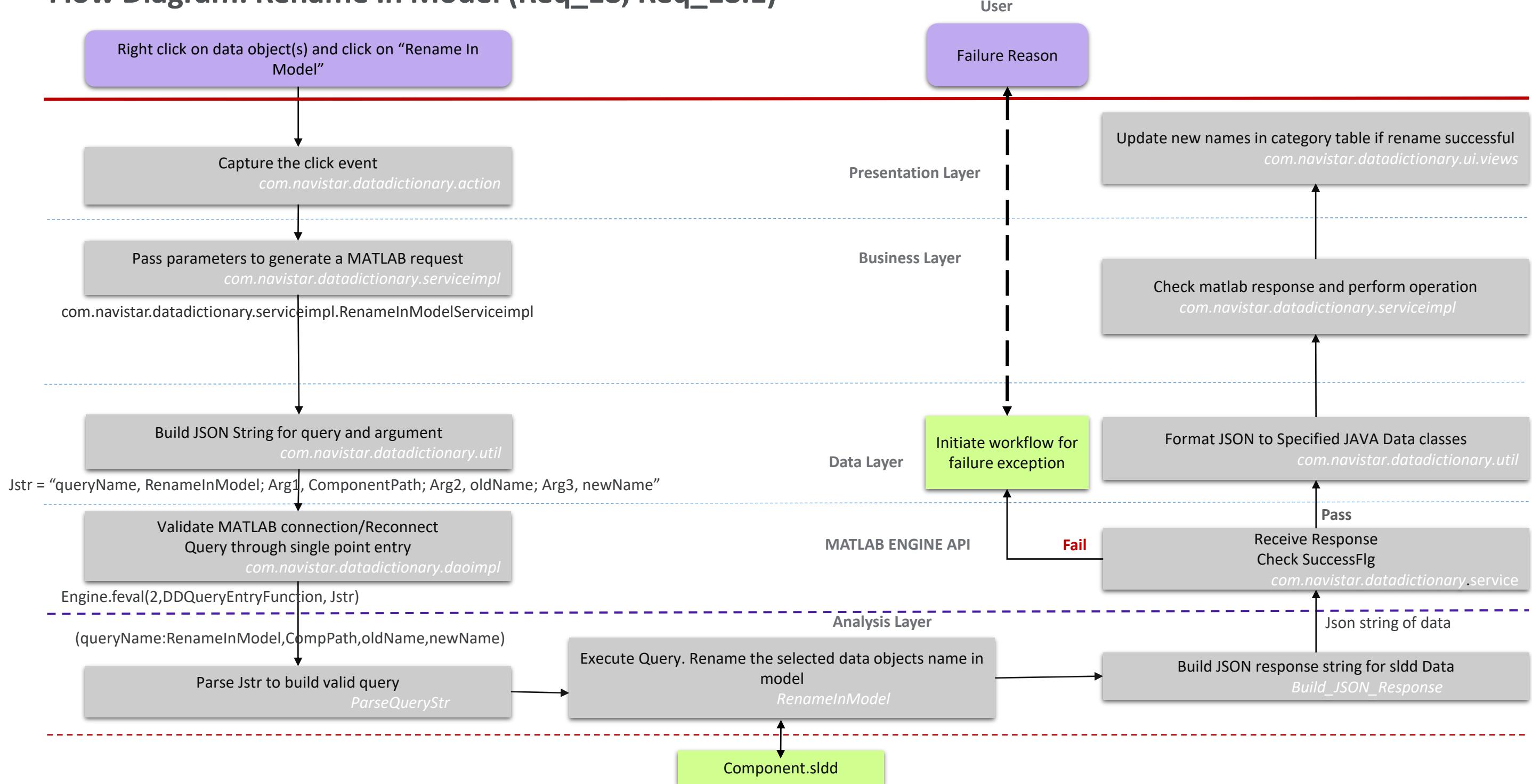
# Flow Diagram: Recently Imported Project(Req\_3.3, Req\_4)



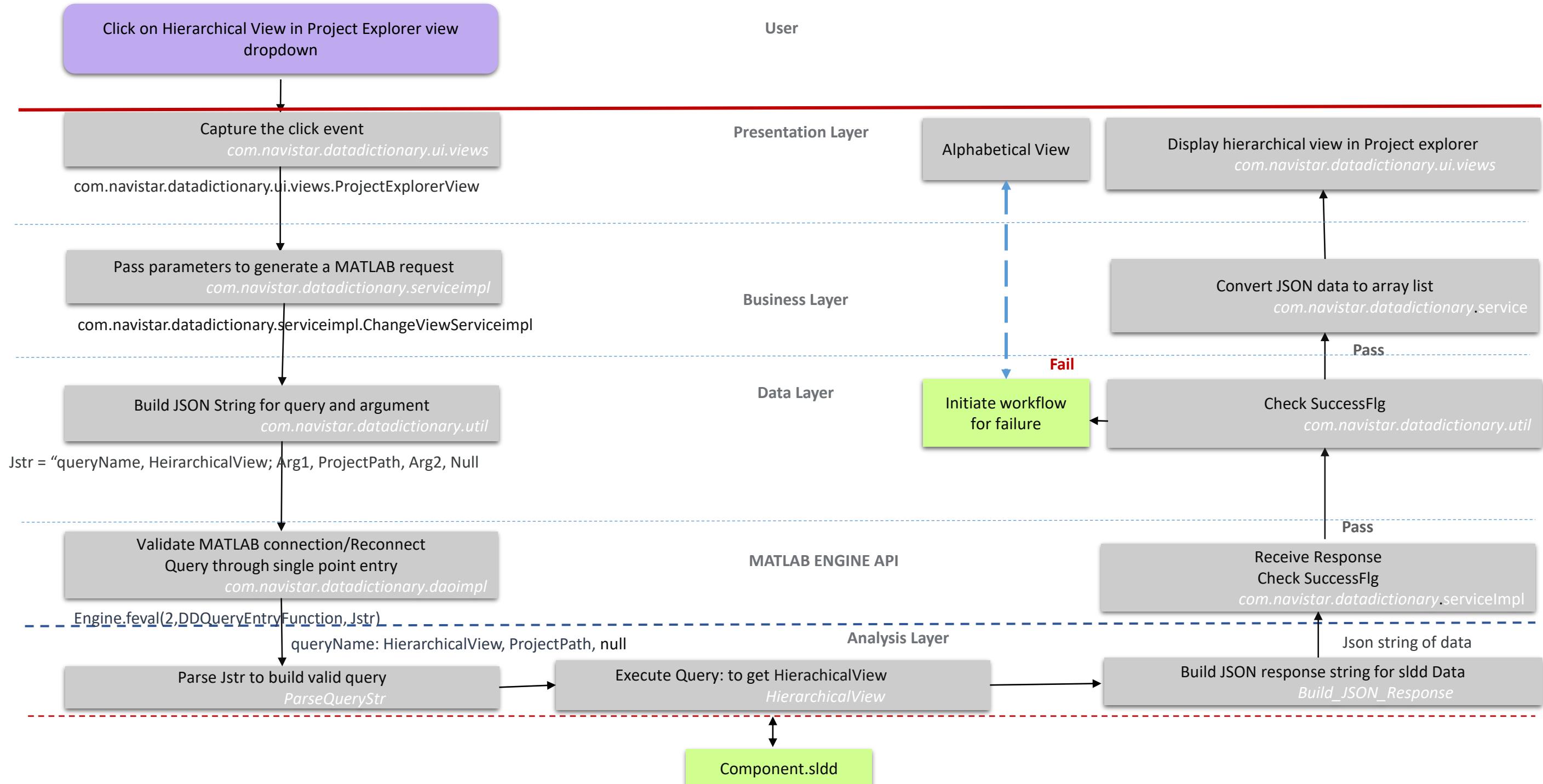
# Flow Diagram: Use This Data Object (Req\_14.6)



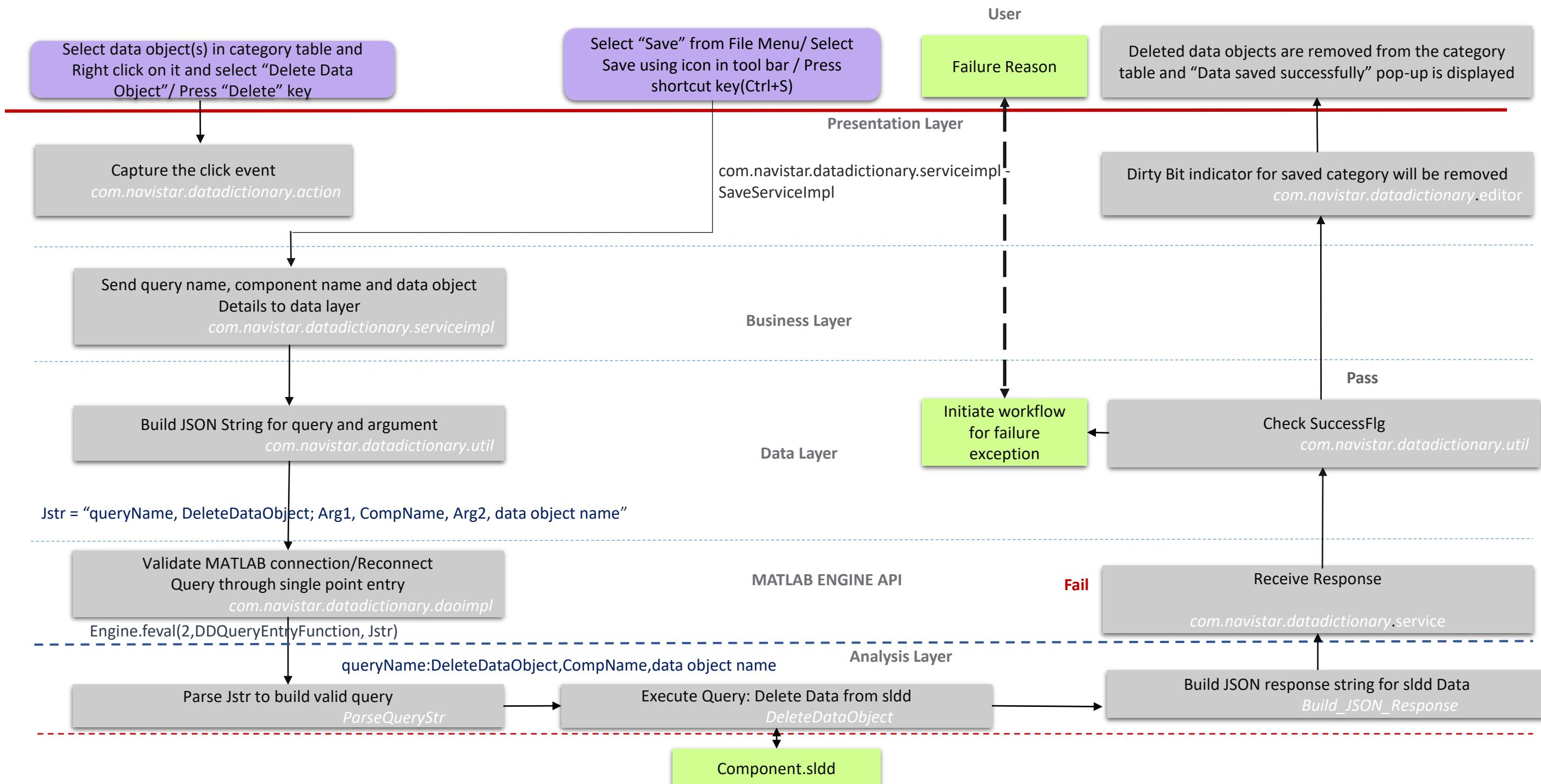
# Flow Diagram: Rename In Model (Req\_18, Req\_18.1)



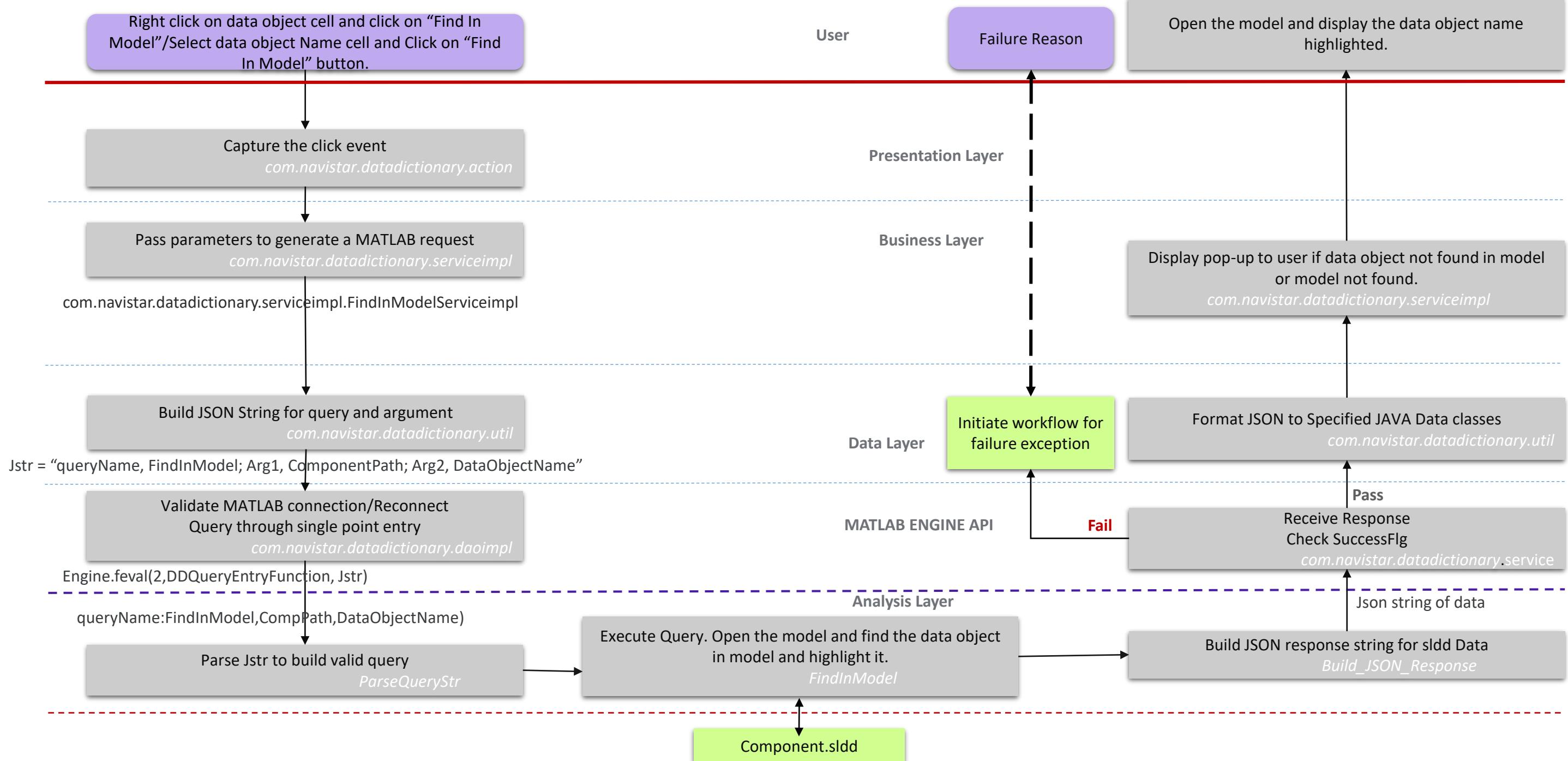
# Flow Diagram: Hierarchical View (Req\_6)



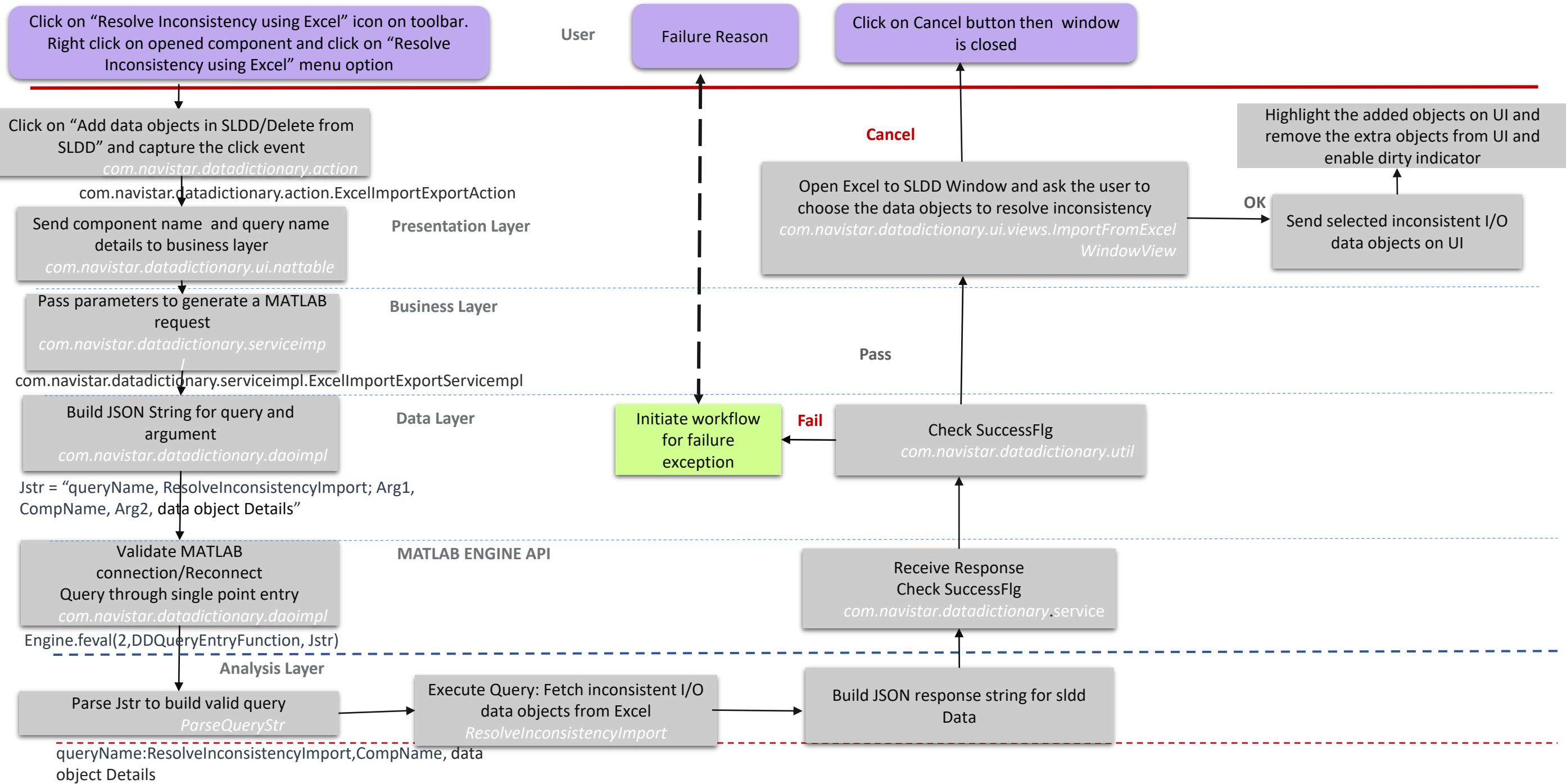
# Flow Diagram: Delete Data Object (Req\_24)



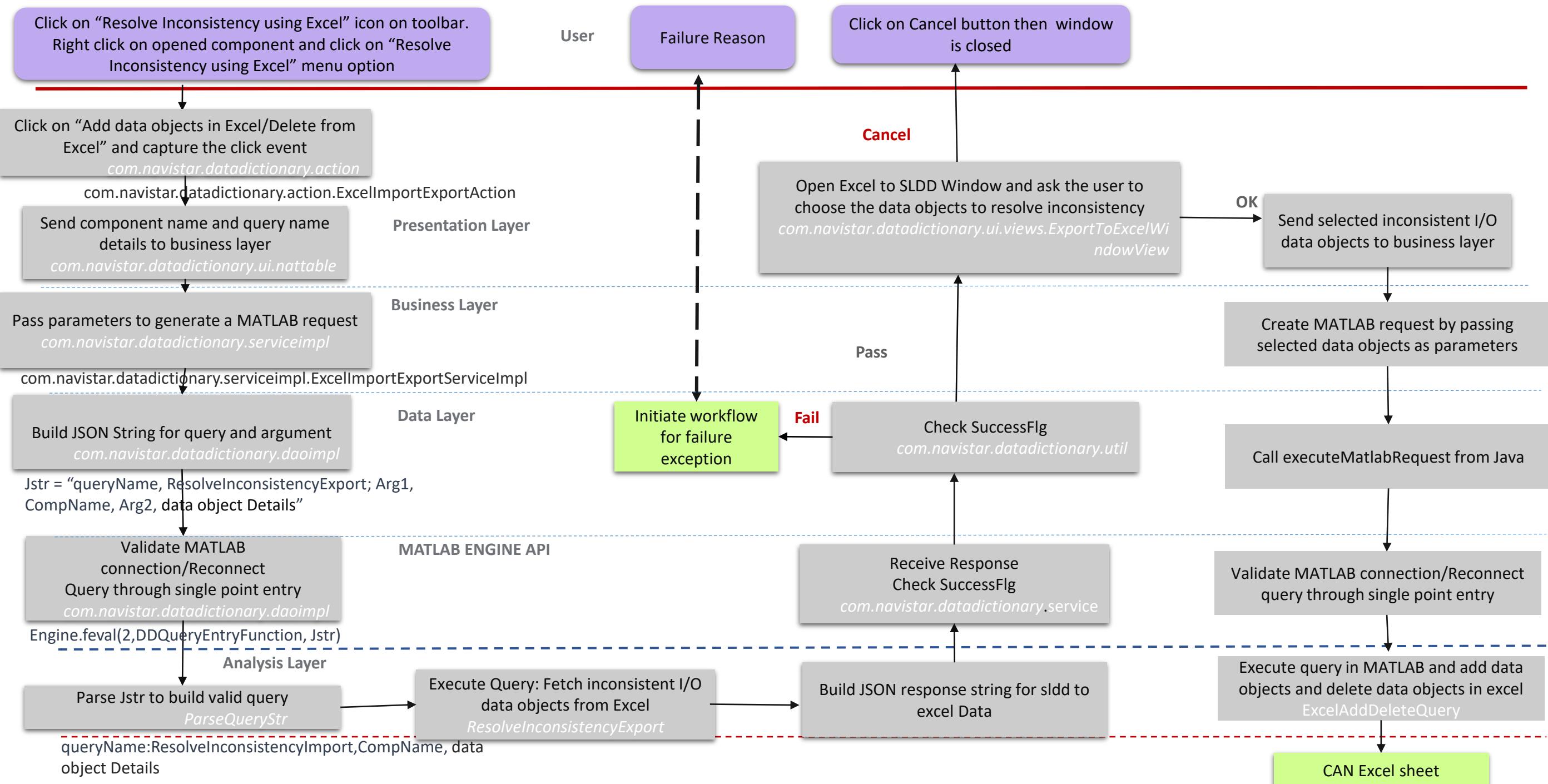
# Flow Diagram: Find In Model (Req\_17)



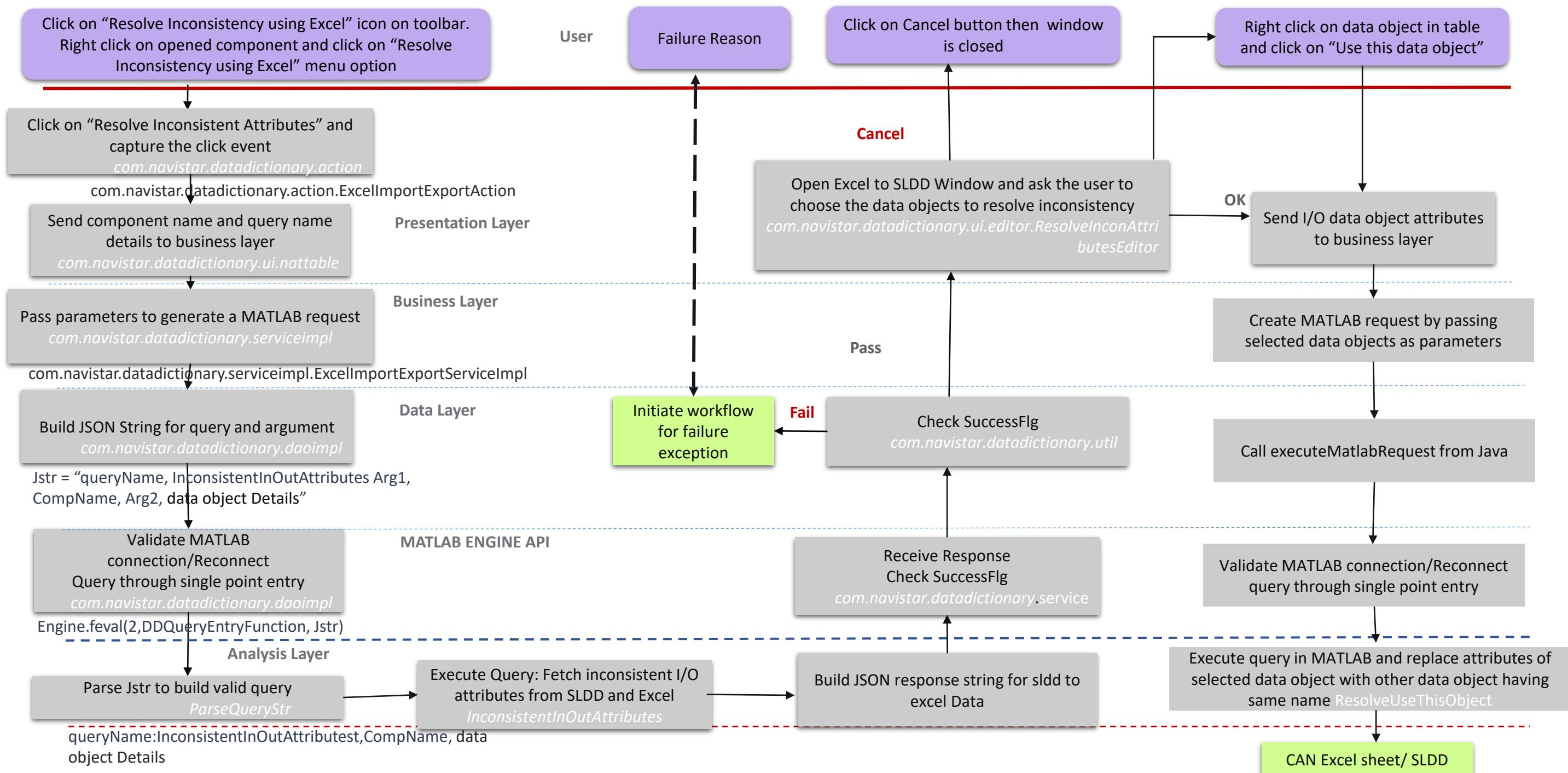
# Flow Diagram: Add data objects in SLDD/Delete from SLDD



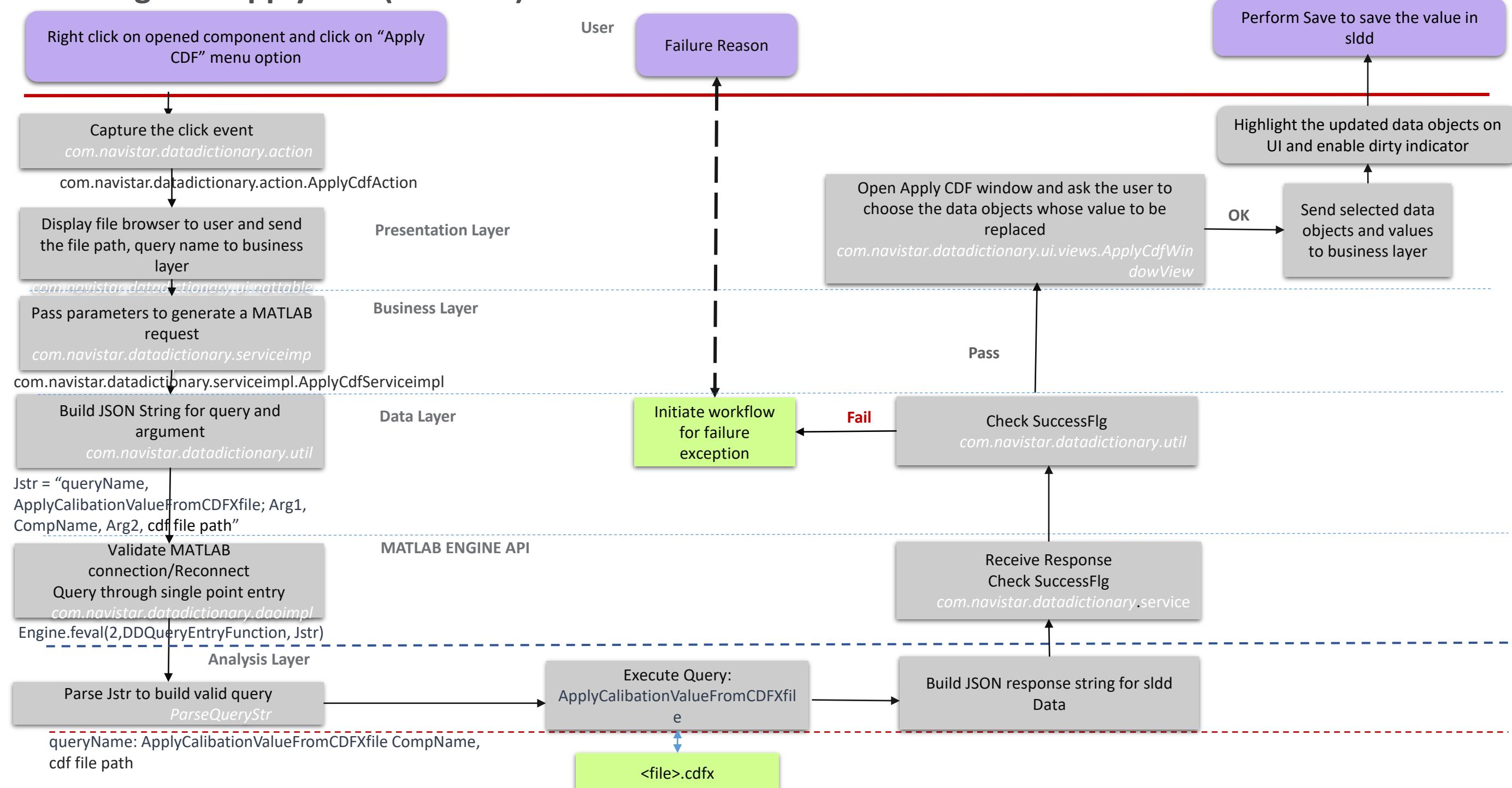
# Flow Diagram: Add data objects in Excel/Delete from Excel



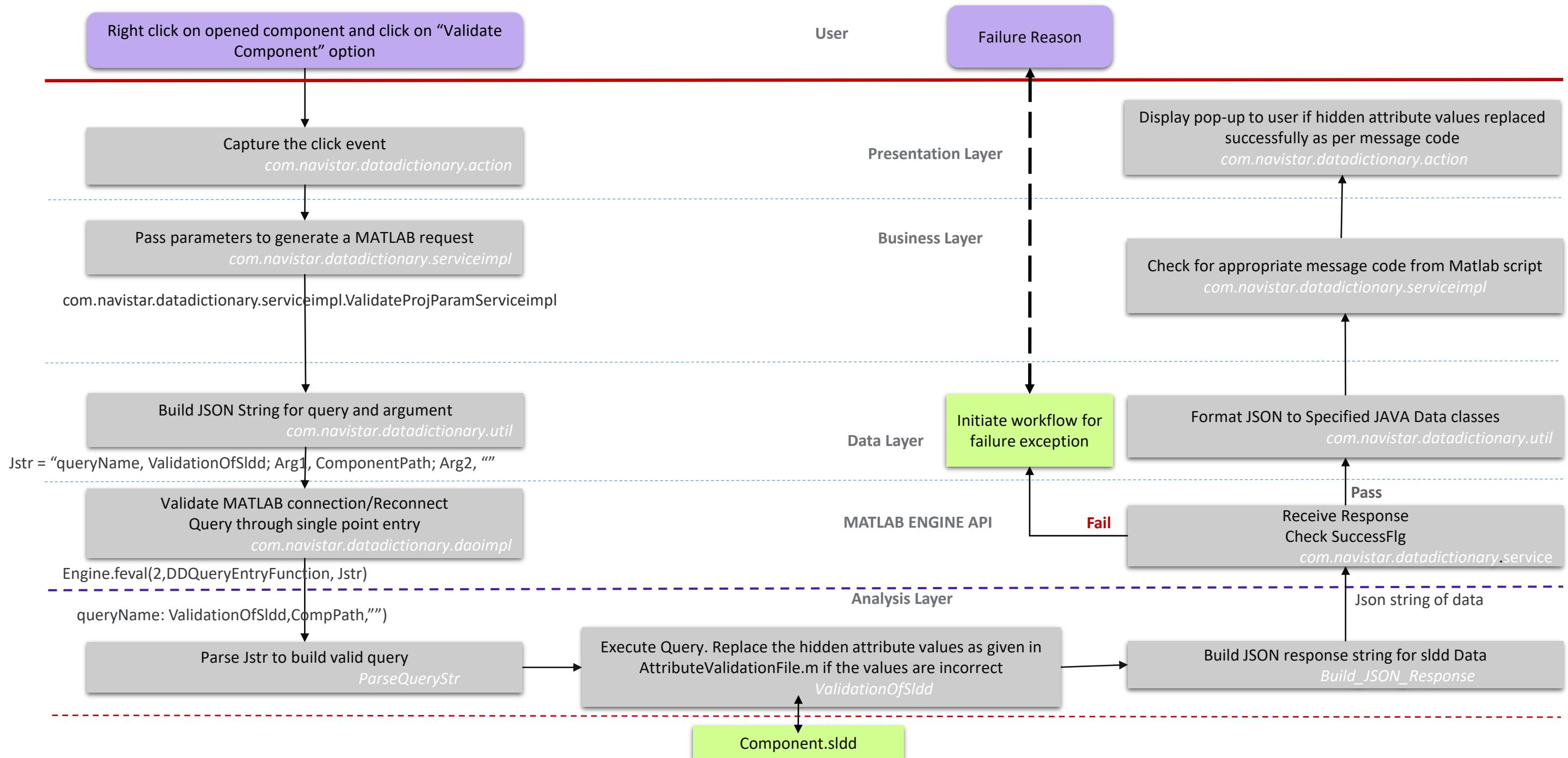
# Flow Diagram: Resolve Inconsistent Attributes using CAN Excel sheet



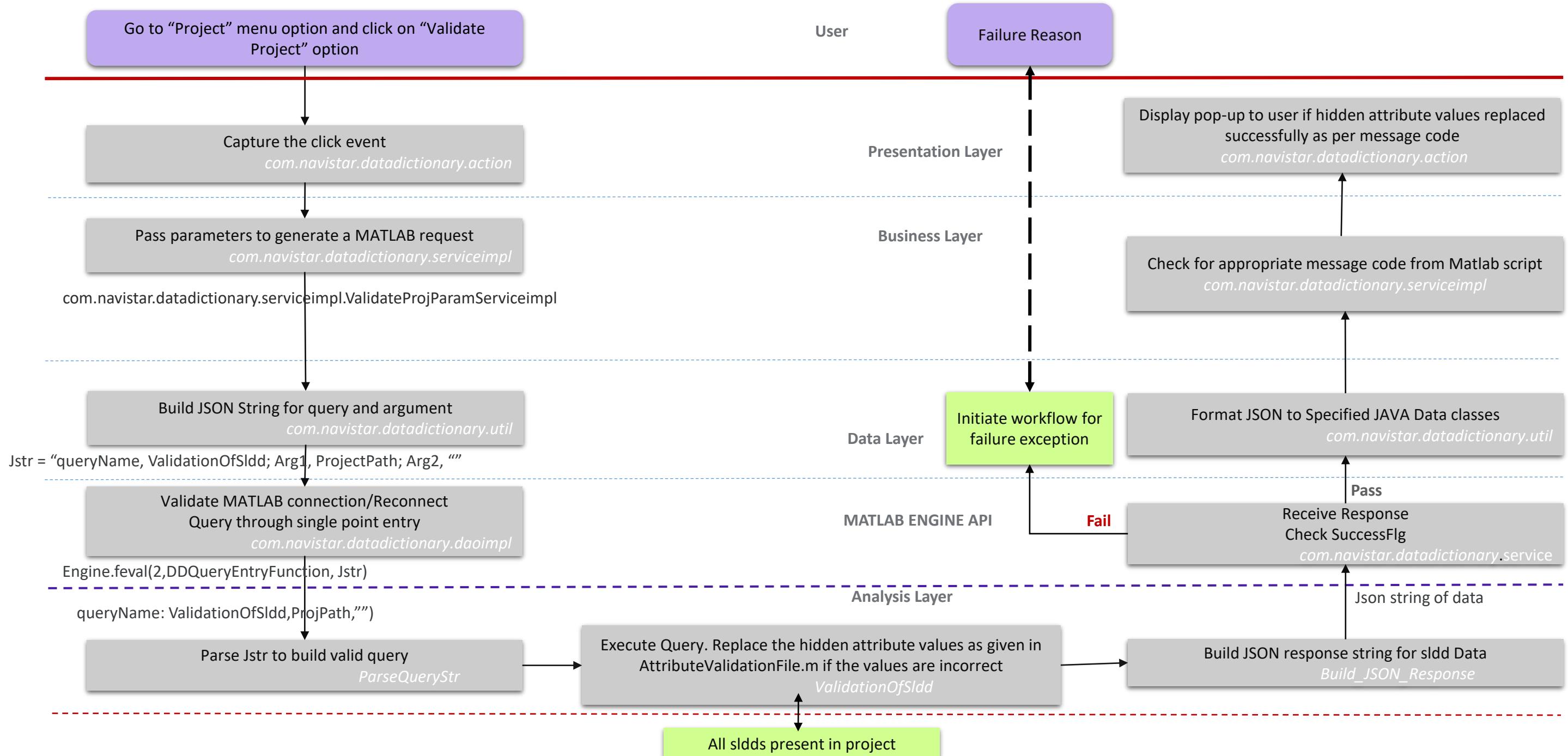
# Flow Diagram: Apply CDF (SLDD-78)



# Flow Diagram: Validate Component (SLDD-95)



# Flow Diagram: Validate Project (SLDD-95)





An aerial photograph of a coastal area. A two-lane asphalt road runs along the left side, with several cars and a motorcycle visible. The road is bordered by a concrete curb and some low walls. To the right of the road is a dense forest of tropical trees, including palm trees. Further right is a rocky coastline with clear turquoise water. In the bottom right corner, there is a white dotted grid pattern. The text "Thank You" is overlaid in the center-left area of the image.

Thank You