CARTOGRAPHAI BRD

Patrick Robak

CARTOGRAPH SOLUTIONS

Contents

I. Version Control

Version	Author	Details	Date
1.0	Patrick Robak	Version 1	12/5/2024

II. Introduction

A. Problem

The typical implementation of a Financial Crime Compliance (FCC) application lasts 12 to 24 months and costs between $5M to $10M. A disproportionate amount of that time and money is spent performing data analysis, mapping, and transformation. Though onerous, this exercise is incredibly important as the ultimate effectiveness of the system is highly dependent on the quality of the underlying data. CartographAI automates this costly, time-consuming, and error-prone effort by combining AI/ML technology with subject matter expertise.

Typically, business analysts seek out source data to satisfy the technical and functional requirements of the target FCC models. They create detailed mapping specifications for the development team to code. To do so, the analysts must have the below skills, but rarely possess more than 1 of these, let alone all 3.

- Experience with the specific FCC software and the underlying data model
- Understanding of the source data, or access to documentation and knowledgeable people
- Subject matter expertise in the products and services offered by the financial institution

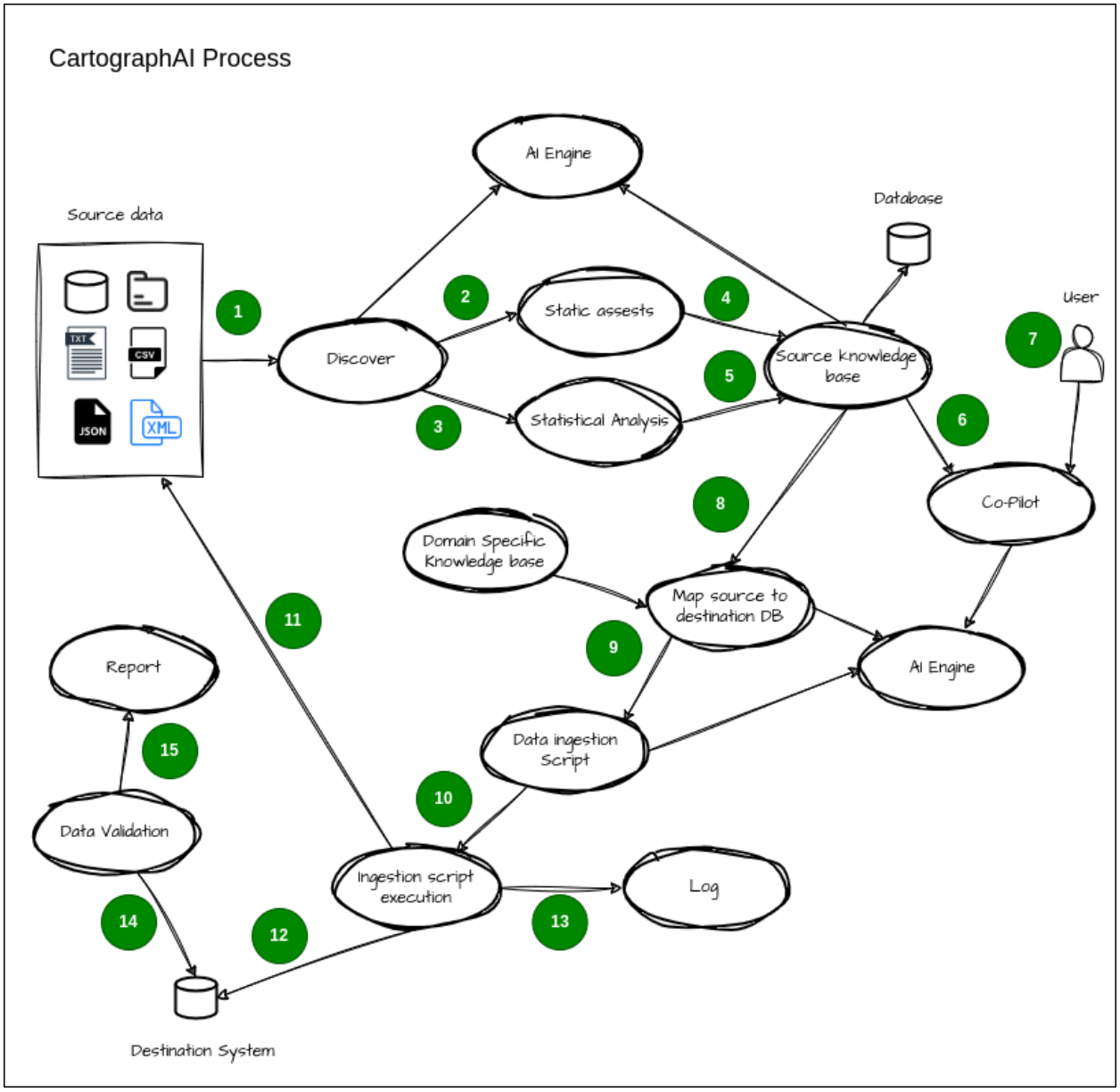
B. Solution

CartographAI is an automated data analysis, mapping, and transformation engine powered by AI that can accurately deliver these responsibilities in hours rather than months. Core functionalities include:

1. Profiling and Discovery
    a. CartographAI analyzes, reviews, and summarizes source data
    b. Understands structure, content, and interrelationships
    c. Assigns meaning to tables (ex. Transactions, Accounts, Customers, etc.)
    d. Assigns meaning to attributes (ex. Transaction ID, Account Number, Amount, etc.)
    e. Measures quality, accuracy, completeness, consistency, timeliness, and accessibility
2. Mapping and Transformation
    a. CartographAI extracts relevant data from the source
    b. Transformations adhere to target data types, constraints, and dependencies
    c. Data is loaded according to the specified timing and frequency
    d. Detailed documentation is automatically generated for transparency and traceability
    e. Mappings requiring manual intervention are flagged for review
    f. Perpetual monitoring ensures data quality does not degrade over time

C. Prototype

The following process diagram is representative of the initial CartographAI prototype. It is not intended to be a requirement, but rather a starting point for discussion.



D. Use Cases

1. New system implementations
2. Addition of new data sources
3. Platform data validations/justifications

E. Business Cases

1. Significantly accelerate project timelines
2. Reallocate resources towards other complex deliverables
3. Offer clients improved time-to-value on software investment

III. Requirements

A. Discover

The Discover module of CartographAI rapidly accelerates the discovery, analysis, and profiling of unknown and poorly documented data sources.

1. Sources

Discover has the ability to connect to and interpret data sources of many different types. This includes, but is not limited to RDMS, CSV, XML, etc.

a. RDBMS: A relational database is a collection of information that organizes data in predefined relationships where data is stored in one or more tables (or "relations") of columns and rows, making it easy to see and understand how different data structures relate to each other. Relationships are a logical connection between different tables, established on the basis of interaction among these tables.
b. NoSQL: The term NoSQL, short for “not only SQL,” refers to non-relational databases that store data in a non-tabular format, rather than in relational tables as relational databases do. NoSQL databases use a flexible schema model that supports a wide variety of unstructured data such as documents, key-value, wide columns, and graphs.
c. Lakehouse: A data lakehouse is a modern data architecture that creates a single platform by combining the key benefits of data lakes (large repositories of raw data in original form) and data warehouses (organized sets of structured data).
d. Apache Iceberg: Apache Iceberg is a high performance open-source format for large analytic tables. Iceberg enables the use of SQL tables for big data while making it possible for engines like Spark, Trino, Flink, Presto, Hive, Impala, StarRocks, Doris, and Pig to safely work with the same tables, at the same time.
e. CSV: A CSV (comma-separated values) file is a text file that has a specific format which allows data to be saved in a table structured format.
f. XML: Extensible Markup Language (XML) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality, and usability across the Internet.
g. JSON: JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is built on two structures:
    - A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
    - An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.
h. Parquet: Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval. It provides high performance compression and encoding schemes to handle complex data in bulk and is supported in many programming languages and analytics tools.

2. Inputs

After connecting to a source, Discover can consume various inputs to gain deeper insights into the nature of the data. The more inputs available to the Discover module, the more accurate the outputs are expected to be. However, input data quality is also an important consideration, as poor input quality could adversely affect results.

a. Source tables
   - Table names, schemas, indexes, etc.
     - Ex. CUSTOMER table name suggests this table stores customer-level information
     - Ex. An index on the CUSTOMER_NAME, DOB, and SSN columns suggests the combination of these attributes may uniquely identify a customer

b. Source attributes
   - Column names, data types, lengths, constraints, etc.
     - Ex. ACCT_NUM column name suggests this attribute is an account number
     - Ex. NUMERIC data type suggests this attribute can only be populated with numbers

c. Source values
   - Actual values populated in each column
     - Ex. 111-11-1111 value highly resembles a Social Security Number
     - Ex. 10-SEP-1987 value highly resembles a date

d. Source metadata
   - When available, structured data that provides information about other data
     - Ex. Text descriptions of source attributes

e. Source documentation
   - When available, documentation provides further information about source data
     - Ex. DOC, XLS, PDF, etc.

3. Outputs

After evaluating the available inputs, the Discover module outputs several key artifacts that detail the data source.

a. High-level functional descriptions
   - Clear, succinct, and accurate plain-English descriptions of each table and each attribute that assigns them meaning in the relevant business context
     - Ex. The CUSTOMER table stores critical reference data about the organization's customers. It contains information about each customer's name, address, type, status, tax identifier, and phone number.

b. Detailed data dictionary
   - Field names, data types, lengths, formats, defaults, descriptions, and examples for each table – using a DB agnostic format, such as DBML
     - Ex. Customer Number, VARCHAR2(50), Unique Customer Identifier, 749556711

c. Entity Relationship Diagram (ERD)

- An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

d. Data profiling report
   - Data profiling is an assessment of data that uses a combination of tools, algorithms, and business rules to create a high-level report of the data's condition. The purpose of data profiling is to uncover inconsistencies, inaccuracies, and missing data so that a data engineer can investigate and correct the source.
   - Data profiling reports are usually visualizations and graphs along with tables to display relevant metrics, such as the degree of duplication in the data set.
   - Data profiling is not only intended for troubleshooting risks to data quality and data integrity. It can also be a key process of discovery for analysts to uncover the structure, content, and relationships between different data sources.

4. Other Features

a. Confidence score
   - All insights generated by the Discover module will be measured against a configurable confidence score. Any output that does not meet the predetermined threshold will be flagged as such and will either require modified module inputs or manual verification.

b. Co-pilot
   - The Discover Co-pilot is an AI-powered virtual assistant. Users can interact with the co-pilot to clarify Discover insights and drill deeper into source databases, tables, attributes, and interrelationships.

c. Duplicate data
   - The Discover module will have the ability to identify duplicate or orphaned data, either within the same source or across different sources. This helps ensure the same transactions, customers, or accounts are not loaded into the target application more than once.

5. UI/UX

a. Discover Explorer
   - The Explorer is the primary screen of the Discover module, displaying all of the aforementioned outputs in a user-friendly and interactive manner.

b. Data source interface/configuration
   - The data source interface is a way to connect CartographAI to data sources of various types, typically by entering credentialed information like username, password, hostname, port, etc. The interface should be able to save the connection details for future use.

B. Map

The Map module of CartographAI leverages the Discover module output along with a knowledge base of mapping rules to propose source-to-target mappings for any target FCC application.

1. Inputs

After running data discovery, Map can consume the below inputs to propose detailed source-to-target mappings for analyst review, or immediate codification.

a. Mapping rule library
   - Functional mapping rules for each target attribute, along with data types, dependencies, and constraints that must be adhered to.

b. Discover module output
   - High-level functional descriptions, detailed data dictionaries, entity relationships, and data profiles will dictate which source tables and attributes are suitable candidates to be mapped to the target application.

2. Applications

CartographAI will include mapping rules for all significant Financial Crime Compliance applications, beginning with the most prevalent. The knowledge base will be expanded and improved over time.

a. NICE Actimize
   - NICE Actimize offers a suite of anti-money laundering (AML) software solutions designed to help financial institutions and others in the regulated sector to detect, investigate and report suspected financial crimes.
     - SAM (Suspicious Activity Monitoring) for Transaction Monitoring
     - KYC (Know Your Customer)
     - WLF (Watch List Filtering) for Sanctions and Payment Screening
     - CTR (Currency Transaction Reporting)

b. Oracle FCCM
   - Oracle Financial Crime and Compliance Management empowers anti–money laundering (AML) and compliance leaders with the right tools to protect their institution from illicit actors and regulatory fines while helping reduce their compliance costs.
     - AML (Anti-Money Laundering) for Transaction Monitoring
     - KYC (Know Your Customer)
     - CS (Customer Screening) for Sanctions Screening
     - TF (Transaction Filtering) for Payment Screening

c. Other FCC Applications (TBD)

3. Outputs

After evaluating the available inputs, the Map module outputs one key artifact.

a. Source-to-target mapping

- Detailed and exportable source-to-target mapping specifications for each critical application attribute, including:
    - Source table
    - Source attribute
    - Source data type
    - Target table
    - Target attribute
    - Target data type
    - Data transformations
    - Sample output
- The mappings will be available in two possible forms:
    - Full load – a full load of all source records meeting the mapping criteria, to be used on Day 0 and on a periodic/as-needed basis
    - Incremental load – an incremental load of delta records meeting the mapping criteria, typically used on a daily/ongoing basis

4. Other Features

a. Manual review/override
   - Each mapping produced by CartographAI can be manually reviewed in the Map Matrix UI. The mapping can either be approved or overwritten, if deemed necessary. All such actions would be fully audited.

b. Sample output
   - Each mapping will be accompanied by a real-time output sample to visualize the effects of any applied transformations. This will assist the user in validating the mappings and any necessary changes.

c. Output validation
   - Automated scripts that validate the mapping output to ensure it conforms to the necessary data types, constraints, and dependencies.

d. 4-eye review
   - A validation that requires at least 2 users to approve a mapping. These validations can be applied to specific target attributes, or across all mappings.

e. Confidence score
   - All insights generated by the Discover module will be measured against a configurable confidence score. Any output that does not meet the predetermined threshold will be flagged as such and will either require modified module inputs or manual verification.

f. Reverse mapper
   - Rather than use Discover insights and the rule library as inputs to produce source-to-target mappings, use existing ETL code and scripts as inputs to reverse engineer existing mappings. This is useful in cases where the current transformations and pipelines are poorly documented but need to be validated. The reverse mappings can also be compared to those CartographAI would typically generate to identify potential gaps for remediation.

5. UI/UX

a. Map Matrix
   - The Matrix is the primary screen of the Map module, displaying all of the aforementioned mappings in a user-friendly and interactive manner.

b. Target application interface/configuration
   - The target application interface is a way to filter the mapping rule library to only map those fields within the scope of each specific project. The user would specify the following parameters:
     - Application (ex. Actimize)
     - Modules (ex. SAM + WLF)
     - Rules (ex. Flow Through of Funds + Historical Behavior Comparison)
     - Usage (ex. Logic + Display)

C. Transform

The Transform module of CartographAI autonomously codifies the specifications produced by the Map module.

1. Inputs

After generating detailed data mapping specifications, Transform can consume the below inputs to autonomously generate ETL code/pipelines.

a. Map module output
   - Detailed and exportable source-to-target mapping specifications for each critical application attribute.

2. Outputs

After evaluating the available inputs, the Transform module outputs one key artifact.

a. ETL code/pipelines
   - A codification of the detailed mapping specifications that can be executed either manually or through a job scheduler to pull data from the data sources, transform it, and load it into the target.
   - The code will be available in two possible forms:
     - Full load – a full load of all source records meeting the mapping criteria, to be used on Day 0 and on a periodic/as-needed basis
     - Incremental load – an incremental load of delta records meeting the mapping criteria, typically used on a daily/ongoing basis

3. Formats

ETL pipelines come in different forms, all of which can be accommodated by CartographAI. Including, but not limited to:

a. SQL

- Structured Query Language (SQL) is a domain-specific language used to manage data, especially in a relational database management system (RDBMS). It is particularly useful in handling structured data.

b. Python
   - Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

c. Apache Spark
   - Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size.

d. Google Composer
   - Cloud Composer is a fully managed workflow orchestration service, enabling you to create, schedule, monitor, and manage workflow pipelines that span across clouds and on-premises data centers.

4. Integrations

In order to productionize ETL pipelines, integration with key related systems is a necessity.

a. Schedulers
   - TWS
     - The IBM Tivoli Workload Scheduler is a software automation tool that provides the backbone for automated workload management and monitoring.
   - AutoSys
     - Broadcom AutoSys Workload Automation is an automated job control system that lets you define, manage, and report on workload.
   - Apache Airflow
     - Apache Airflow is an open-source platform for authoring, scheduling and monitoring data and computing workflows. First developed by Airbnb, it is now under the Apache Software Foundation. Airflow uses Python to create workflows that can be easily scheduled and monitored.
   - API
     - Scheduling is also be available as an API call directly to the interface.

5. UI/UX

a. Transform Designer
   - The Designer is the primary screen of the Transform module. It is an easy-to-use web interface where users can create, modify, and manage processes.

D. Governance

The CartographAI data governance framework is a set of rules and processes that manage data within the application. The purpose of the framework is to ensure the integrity, security, and compliance of data, and to establish a standard for how data is collected, organized, stored, and used. This makes it easier to streamline and scale data governance, maintain policy and regulatory compliance, democratize data, support collaboration, and build trust.

1. Quality

Data quality involves processes to ensure data is accurate, complete, and reliable. CartographAI will measure and report the below data quality elements at both the source and target levels on a perpetual basis.

a. Accuracy
   - Data accuracy is a measure of how close data is to its true value, or how correct, precise, and error-free it is. It's a key part of data quality, and inaccurate data can lead to poor decisions, poor customer service, and operational inefficiencies.

b. Completeness
   - Data completeness is the degree to which a dataset has all the necessary information and attributes, and no missing or empty fields. It's important for accurate insights and decision-making.

c. Consistency
   - Data consistency is the state of data where all copies are the same across all systems and databases. It's important because it helps ensure that data is accurate, up-to-date, and coherent across different applications, platforms, and systems.

d. Timeliness
   - Data timeliness is the degree to which data is up-to-date and available when it needs to be used.

e. Accessibility
   - Data accessibility is the degree to which users can easily find, understand, use, and retrieve data within a given source.

2. Auditability

The framework includes robust auditing functionalities that provide detailed records of every action taken within CartographAI, whether conducted by a user or the system. This audit history helps ensure data security and regulatory compliance by monitoring when, how, and by whom changes are made.

3. Lineage

Data lineage process within CartographAI tracks the life cycle of data, including how it's generated, transformed, moved, and used across the system. It provides a detailed record of the data's origins, transformations, and movements. This lineage is presented to users as both a report and as a visualization within the CartographAI user interface.

E. Security

The data security framework protects digital information from unauthorized access, theft, or corruption.

1. IAM

Identity Access Management grants secure access to CartographAI to verified entities, ideally with a bare minimum of interference. The goal is to manage access so that the right people can do their jobs and the wrong people, like hackers, are denied entry.

There are two parts to granting secure access to an organization's resources:

a. Identity management
   - Identity management checks a login attempt against an identity management database, which is an ongoing record of everyone who should have access. This information must be constantly updated as users join or leave the organization, their roles and projects change, and the organization's scope evolves.

b. Access management (Role-based access control)
   - Access management, or role-based access control, keeps track of which modules and features the user has permission to access. Most organizations grant varying levels of access to resources and data and these levels are determined by factors like job title, tenure, security clearance, and project.

Granting the correct level of access after a user's identity is authenticated is called authorization. The goal of the IAM system is to make sure that authentication and authorization happen correctly and securely at every access attempt.

2. Logging

The framework records detailed information about data access, modifications, and movement within the system by tracking the "trace" of a data request as it travels through different components, allowing security teams to identify potential threats and suspicious activity by monitoring how data is being accessed and manipulated across the system, essentially providing a comprehensive view of data flow for enhanced security analysis.

3. Compliance

a. SOC 2
   - SOC 2 is a cybersecurity compliance framework that assesses how well a service organization manages customer data. It evaluates a service organization's controls related to security, availability, processing integrity, confidentiality, and privacy.
   - For more information, refer to https://secureframe.com/hub/soc-2/what-is-soc-2

b. ISO 27001
   - ISO 27001 is an international standard that helps organizations manage the security of their information. The requirements include:
     - Establishing information security objectives
     - Developing a plan to achieve those objectives
     - Identifying information security risks

    - Selecting appropriate controls to tackle those risks
- For more information, refer to https://www.iso.org/standard/27001

F. Deployment

CartographAI will follow cloud-native design and should be vendor agnostic. It will work both in public cloud infrastructure and private cloud infrastructure.

Clients are able to use CartographAI securely as a SaaS (Software as a Service) or it can be deployed in a cloud of their choice (AWS, Azure, GCP) with cloud formation templates and recommendations for the right sizing. The tool also provides a cloud cost estimate with the ability to adjust to the right sizing periodically.

G. Licensing

Access to CartographAI will be granted via subscription-based license. Every instance will require a separate license to run the application. CartographAI administrators will have the provision to revoke the license as governed by the terms and conditions.

IV. Traceability

ID	Section	Sub-Section	Group	Requirement
	I	N/A	N/A	Not a requirement
	II	A. Problem	N/A	Not a requirement
	II	B. Solution	N/A	Not a requirement
	II	C. Prototype	N/A	Not a requirement
	II	D. Use Cases	N/A	Not a requirement
	II	E. Business Cases	N/A	Not a requirement
BRD-1	III	A. Discover	1. Sources	a. RDMS
BRD-2	III	A. Discover	1. Sources	b. NoSQL
BRD-3	III	A. Discover	1. Sources	c. Lakehouse
BRD-4	III	A. Discover	1. Sources	d. CSV
BRD-5	III	A. Discover	1. Sources	e. XML
BRD-6	III	A. Discover	1. Sources	f. JSON
BRD-7	III	A. Discover	1. Sources	g. Parquet
BRD-11	III	A. Discover	2. Inputs	a. Source tables
BRD-12	III	A. Discover	2. Inputs	b. Source attributes
BRD-13	III	A. Discover	2. Inputs	c. Source values
BRD-14	III	A. Discover	2. Inputs	d. Source metadata
BRD-15	III	A. Discover	2. Inputs	e. Source documentation
BRD-16	III	A. Discover	3. Outputs	a. High-level functional descriptions
BRD-17	III	A. Discover	3. Outputs	b. Detailed data dictionary
BRD-18	III	A. Discover	3. Outputs	c. Entity relationship diagram (ERD)
BRD-19	III	A. Discover	3. Outputs	d. Data profiling report
BRD-20	III	A. Discover	4. Other Features	a. Confidence score
BRD-21	III	A. Discover	4. Other Features	b. Co-pilot
BRD-22	III	A. Discover	4. Other Features	c. Duplicate data
BRD-23	III	A. Discover	5. UI/UX	a. Discover Explorer
BRD-24	III	A. Discover	5. UI/UX	b. Data source interface/configuration
BRD-25	III	B. Map	1. Inputs	a. Mapping rule library
BRD-26	III	B. Map	1. Inputs	b. Discover module outputs
BRD-27	III	B. Map	2. Applications	a. NICE Actimize
BRD-28	III	B. Map	2. Applications	b. Oracle FCCM
BRD-29	III	B. Map	2. Applications	c. Other FCC Applications (TBD)
BRD-30	III	B. Map	3. Outputs	a. Source-to-target mapping
BRD-31	III	B. Map	4. Other Features	a. Manual review/override
BRD-32	III	B. Map	4. Other Features	b. Sample output

BRD-33	III	B. Map	4. Other Features	c. Output validation
BRD-34	III	B. Map	4. Other Features	d. 4-eye review
BRD-35	III	B. Map	4. Other Features	e. Confidence score
BRD-36	III	B. Map	5. UI/UX	a. Map Matrix
BRD-37	III	B. Map	5. UI/UX	b. Target application interface/configuration
BRD-38	III	C. Transform	1. Inputs	a. Map module output
BRD-39	III	C. Transform	2. Outputs	a. ETL code/pipelines
BRD-40	III	C. Transform	3. Formats	a. SQL
BRD-41	III	C. Transform	3. Formats	b. Python
BRD-42	III	C. Transform	3. Formats	c. Apache Airflow
BRD-43	III	C. Transform	3. Formats	d. Apache Spark
BRD-44	III	C. Transform	3. Formats	e. Google Composer
BRD-45	III	C. Transform	4. Integrations	a. Schedulers
BRD-46	III	C. Transform	5. UI/UX	a. Transform Designer
BRD-47	III	D. Governance	1. Quality	a. Accuracy
BRD-48	III	D. Governance	1. Quality	b. Completeness
BRD-49	III	D. Governance	1. Quality	c. Consistency
BRD-50	III	D. Governance	1. Quality	d. Timeliness
BRD-51	III	D. Governance	1. Quality	e. Accessibility
BRD-52	III	D. Governance	2. Auditability	a. Auditability
BRD-53	III	D. Governance	3. Lineage	a. Lineage
BRD-54	III	E. Security	1. IAM	a. Identity management
BRD-55	III	E. Security	1. IAM	b. Access management
BRD-56	III	E. Security	2. Logging	a. Logging
BRD-57	III	E. Security	3. Compliance	a. SOC 2
BRD-58	III	E. Security	3. Compliance	b. ISO 27001
BRD-59	III	F. Deployment	1. Deployment	a. Deployment
BRD-60	III	G. Licensing	1. Licensing	a. Licensing
	IV	N/A	N/A	Not a requirement