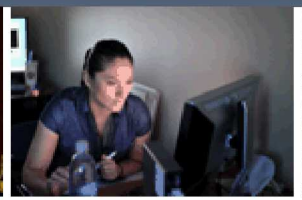
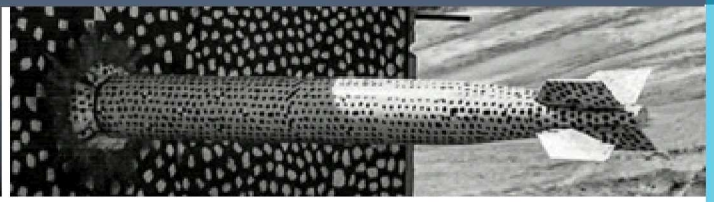
SAND2019-10967C

Citadel: Stockpile Evaluation Data System





PRESENTED BY

William DeRaad (SNL); Stephen Jackson (SNL)

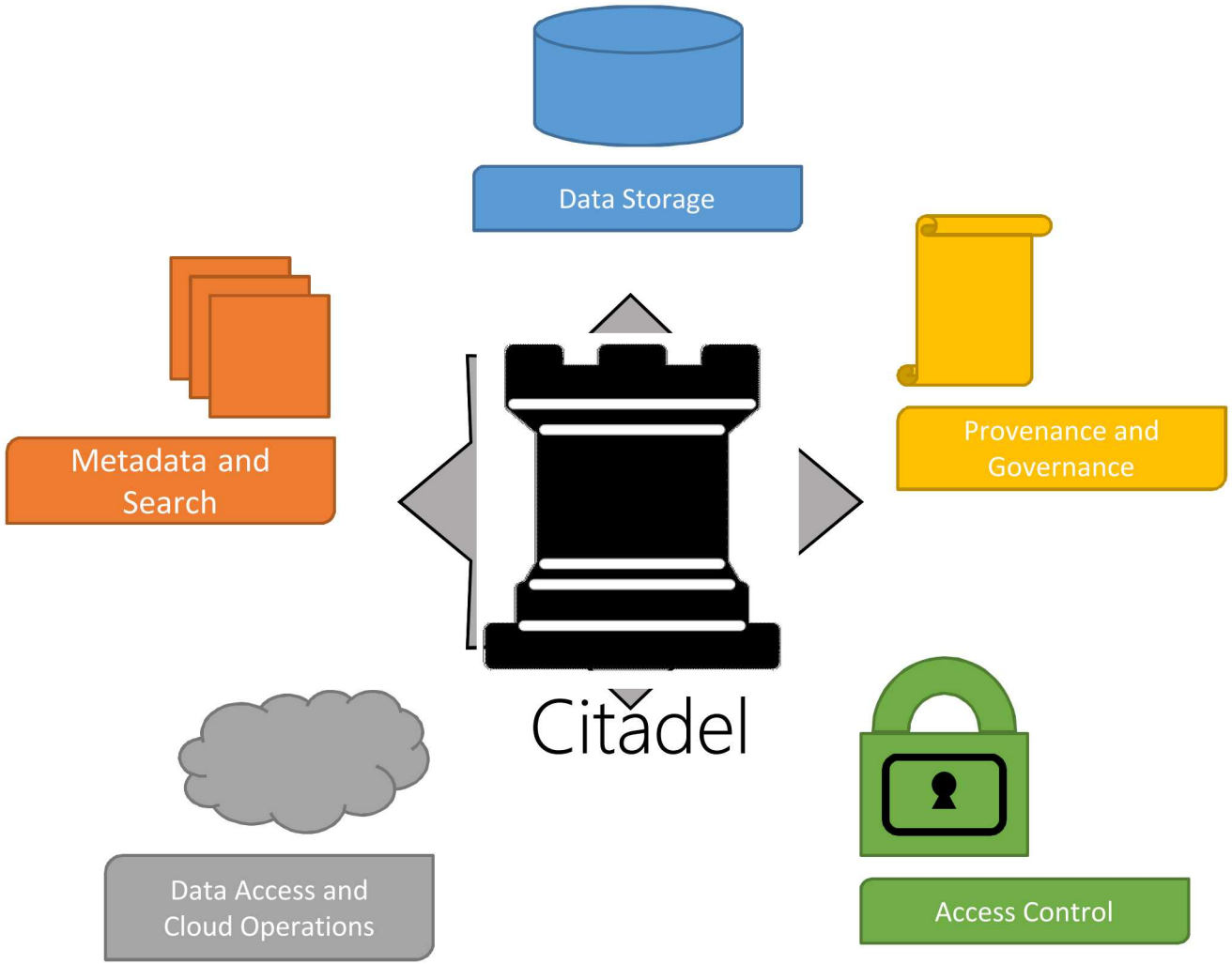
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Citadel

Citadel is a framework for making data engineering applications

Bootstrap organizing and storing data by providing:

- File Storage
- Provenance & Governance
- Authentication and Authorization
- Customizable record formats & Integrated search
- Cloud Operations



Primary Customers

Stockpile Evaluation Data System

Surveillance Data

Static record structures

Integrations with external data sources

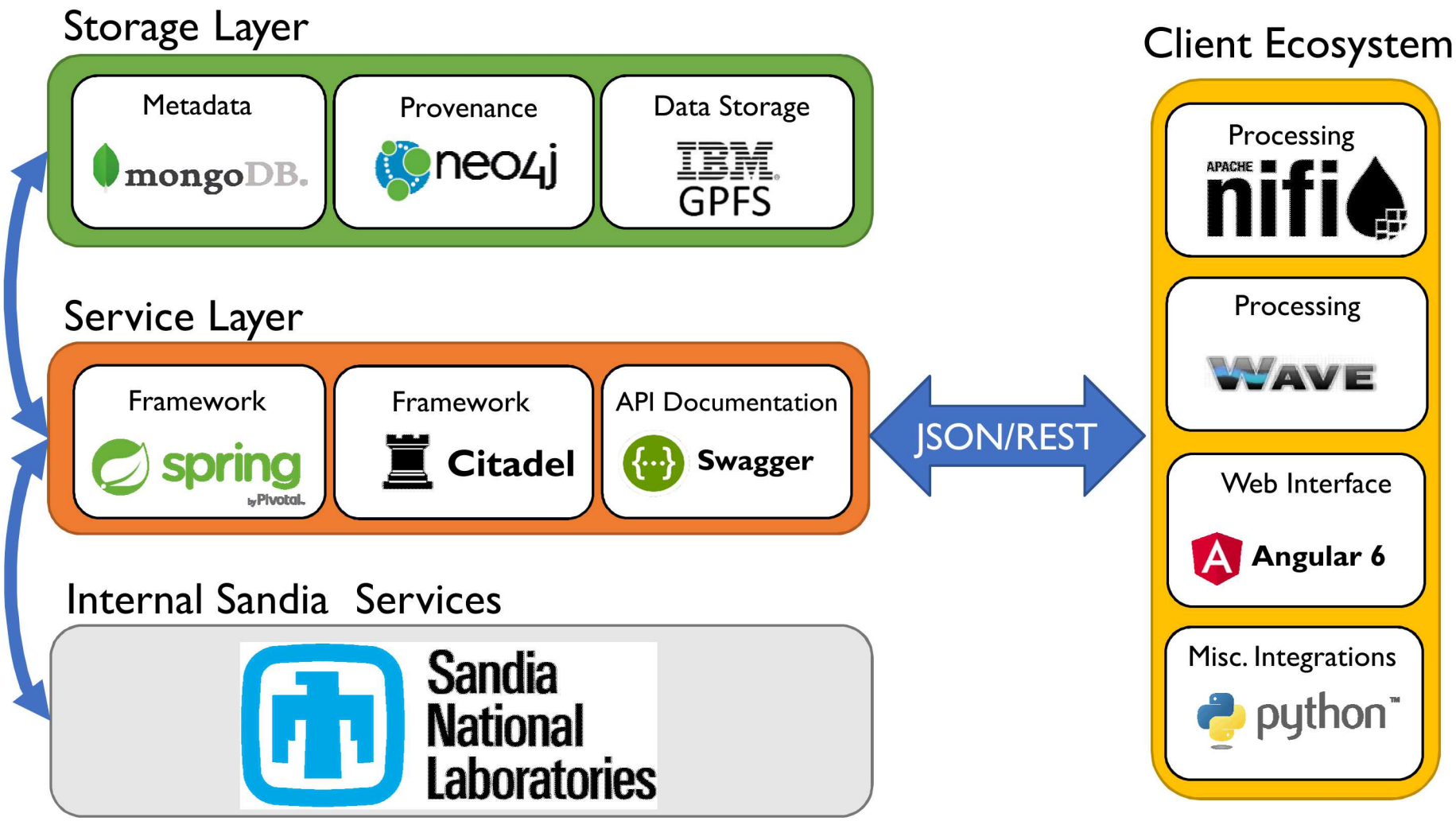
Data for Systems Engineering Applications

Development data

Primarily self-contained

In-the-field data collection

Full Citadel Stack



All Citadel services are provided by a JSON/REST API allowing customers to create their own clients to interact with Citadel applications.

Dev Ops

Continuous Integration

Continuous Deployment

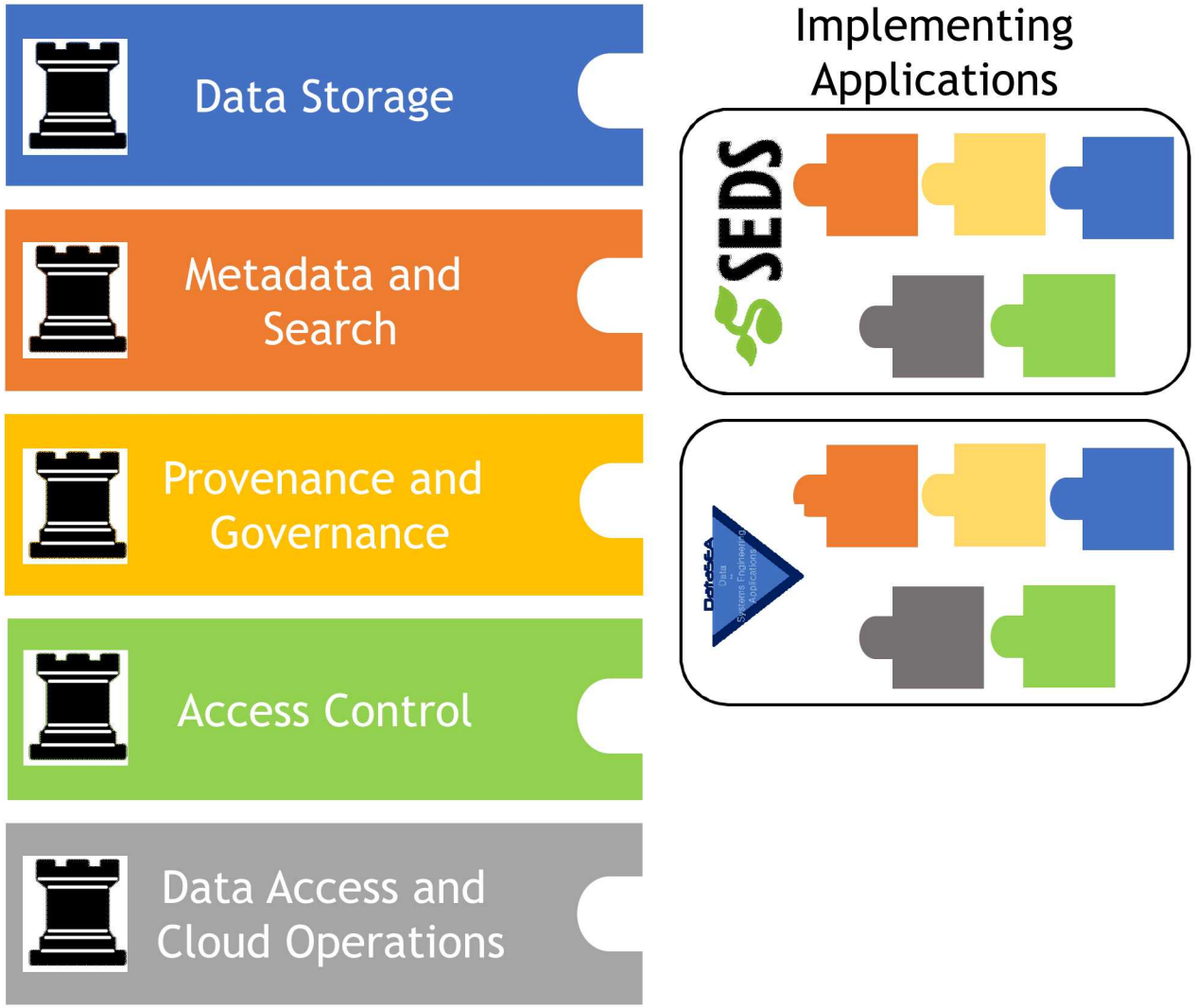
Supporting Features

Citadel Framework is extensible, supporting custom extensions to its sub-systems

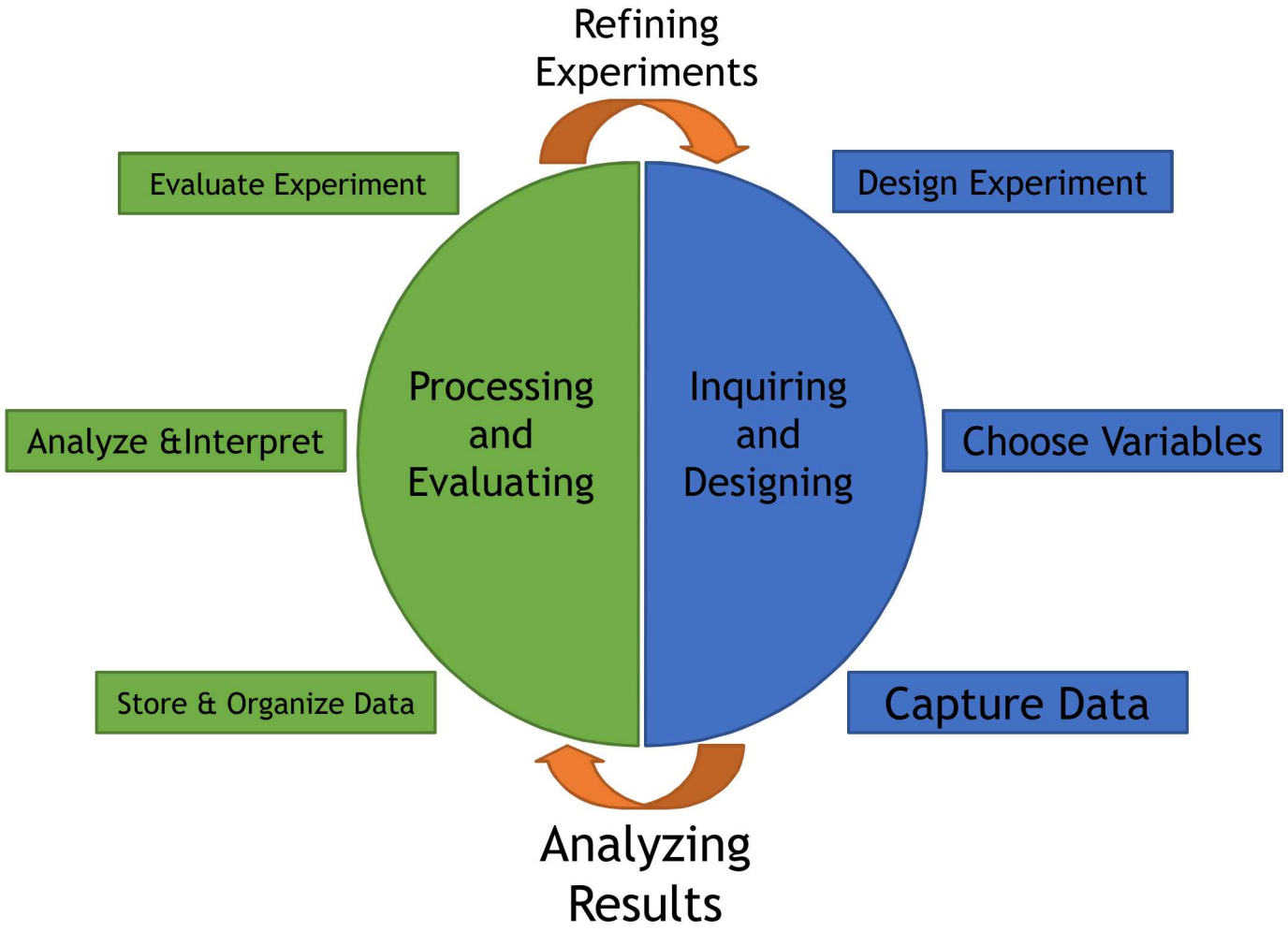
Applications can plug-n-play features they need as they need them

Out-of-the-Box Features:

- Storage system
  - Connectors to multiple mediums: NFS / Knox / HDFS.
- Authentication / Authorization:
  - LDAP.
  - OAuth2
- Data Access And Cloud:
  - Extract files to standard Parquet format.
- Provenance and Governance



Citadel Enables A Data Lifecycle



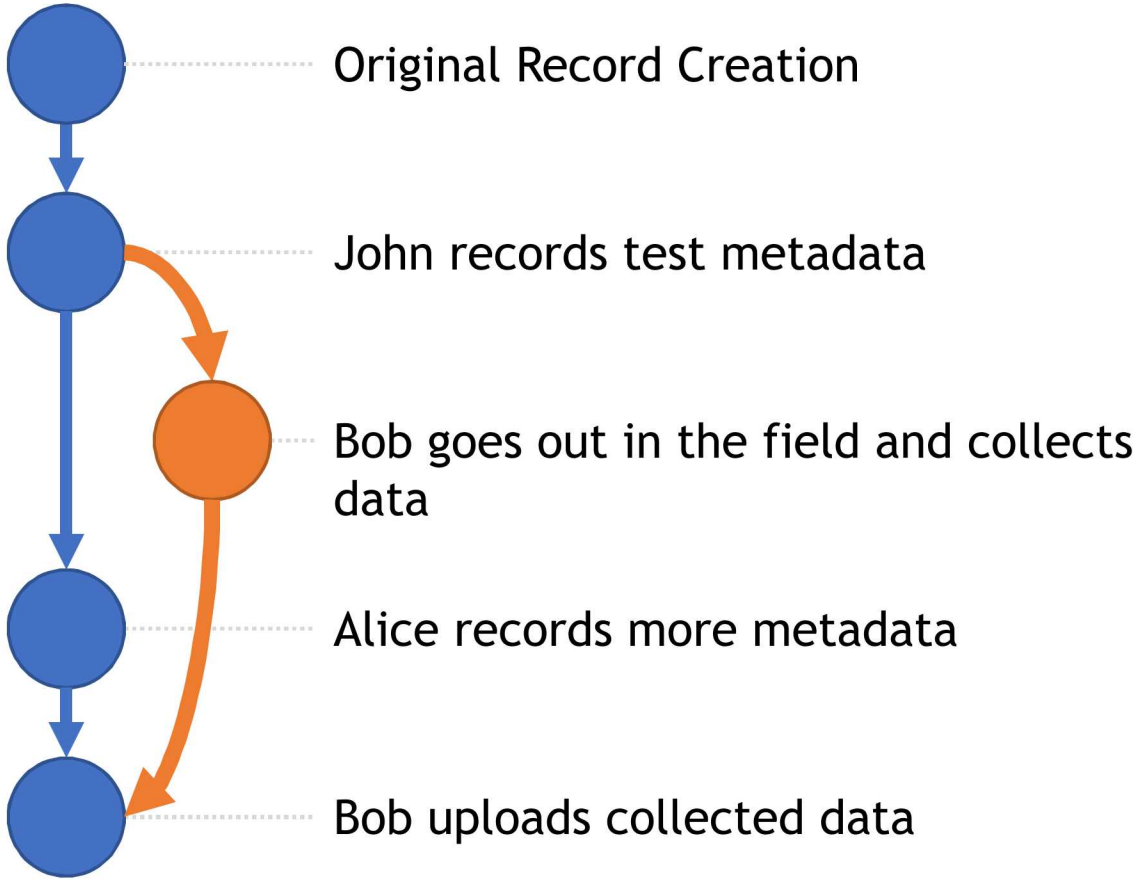
Provenance

Citadel stores data and associated metadata as "records."

We maintain a Directed Acyclic Graph of changes to a record

Using the graph we can inspect a record through time and use the graph to understand how and why a record changed

Citadel supports complex provenance graphs, including information changed or created in other instances.



Provenance

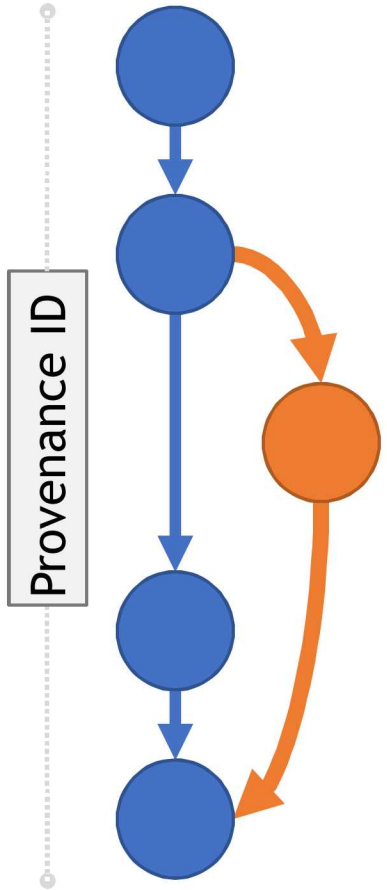
Open source graph database

Stores provenance information

- Branching structure
- Relationship between document revisions
- Reasons for changes

NoSQL Document-Oriented Database

Stores record contents



- Records are assigned a Provenance ID that are shared across all revisions of that record
- Individual revisions are assigned Revision IDs
- Revision IDs connect Neo4j nodes to Mongo Documents

Data Frames

Parquet is a columnar data storage format

- Allows compression schemes on a per-column level
- Future-proofed to allow adding more encodings

AVRO is a row-oriented data serialization library

AVRO is how a row is read from a Parquet file

- Self-Describing
- Supports code generation
- Reasons for changes

DataFrames can store anything that can be stored as a row in Avro

- In most cases Time/Answer pairs

To create a DataFrameGroup clients submit:

- A set of files along with which processing JAR to use

Processing JARs are isolated from the system

- Allows the use of legacy parsers without inheriting technical debt
- Provides extensibility to submit jobs to other processing frameworks in the future

Access Control

Built on Spring Security

Provides both authentication and authorization

Authorization determination made using a set of rules defined per record type

- Rules are stored in custom Permission Evaluators
- Permission Evaluator are selected based on tightest inheritance
- Permission Evaluators can Confirm, Deny, or Defer
- Permission types include Read, Create, Edit, Patch, Terminate, and Merge

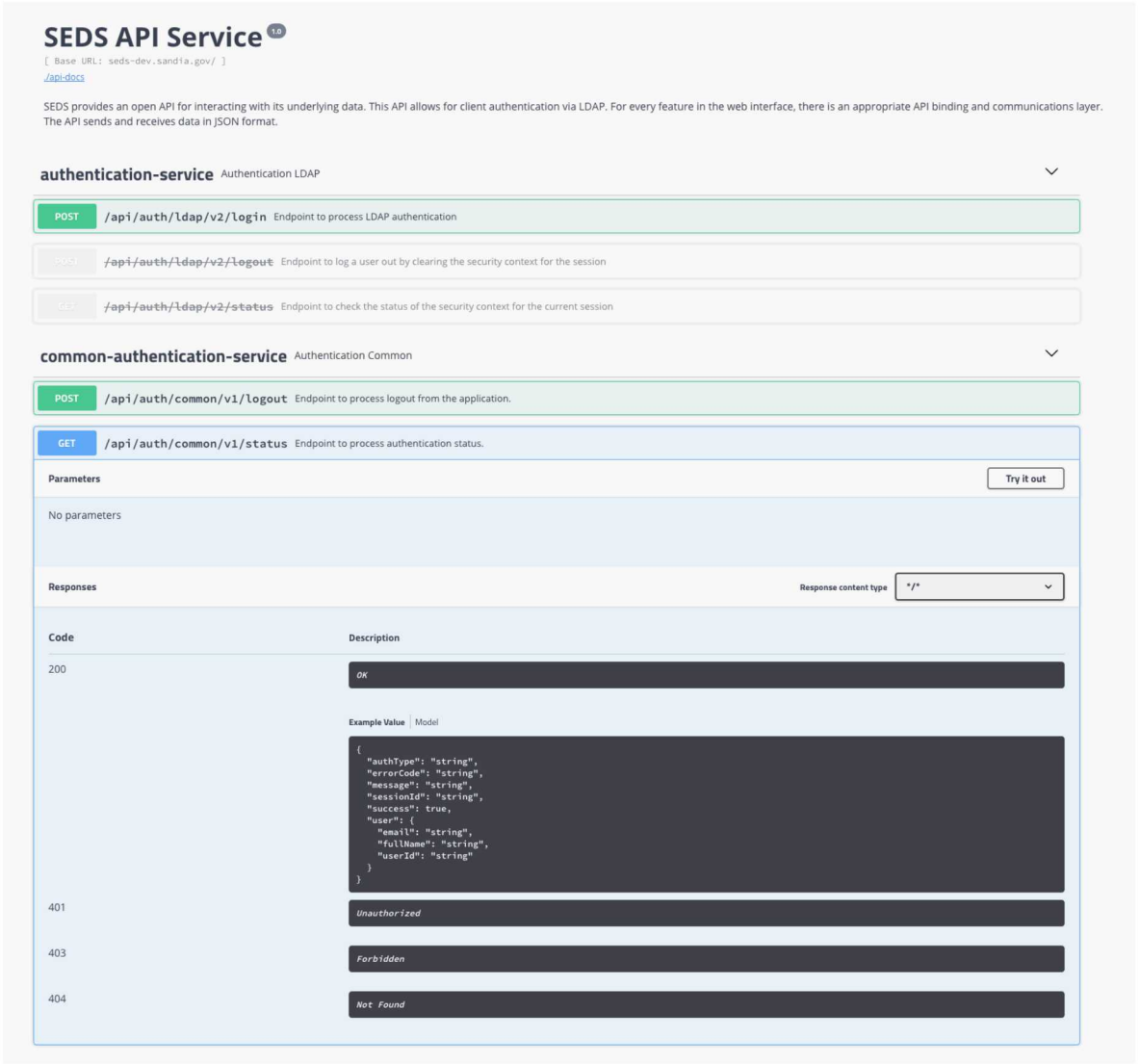
API documentation

API documented using the OpenAPI specification

Provided to clients via SwaggerUI Web Interface

- Provides clients a mechanism to test API endpoints
- Always in sync with our deployed API

Using OpenAPI enables client to autogenerate code to interact with our API in most modern languages



HDF5 Format and Library

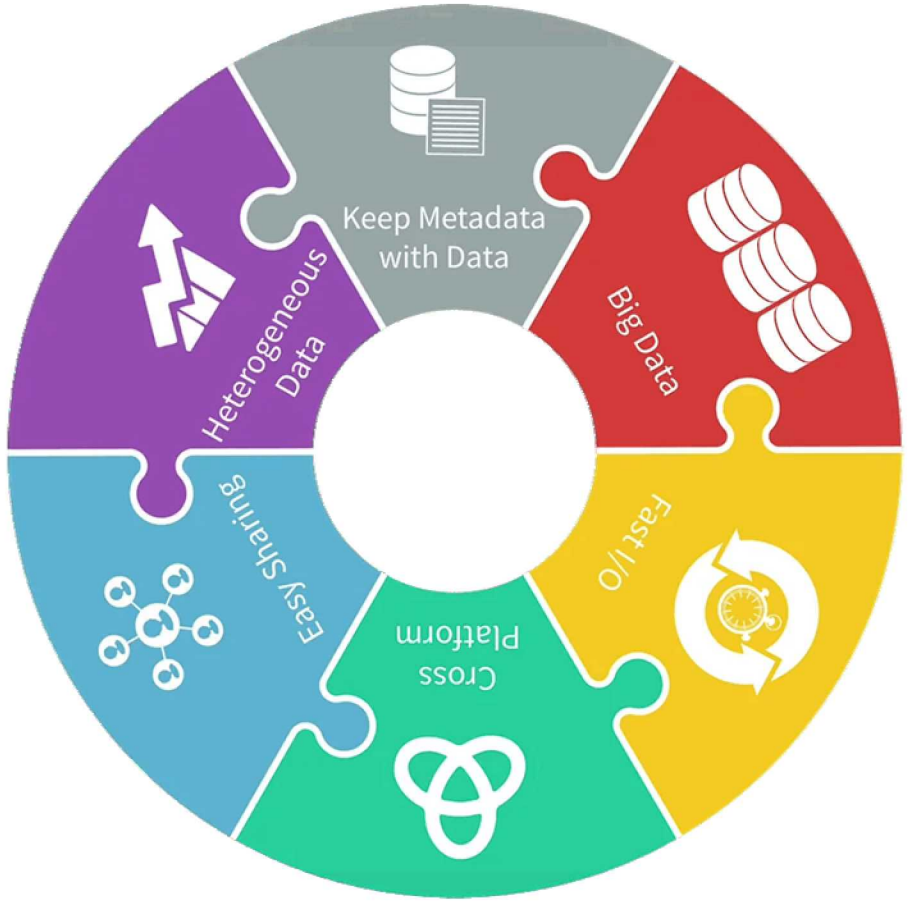
Helped our customer create an HDF5 specification and Java library

Enable development of future tools to capture rich metadata

Specification is represented in XSD

- Human readable
- Editable via the customer

Java Reader and Writer are then generated for the HDF5 format from the XSD files



SEDS Legacy Migration

Built in Java

All data is migrated through SEDS REST API

- API client generated in Java via Swagger Codegen
- Legacy data access via Hibernate ORM

Pros:

- Validate our API works for legacy data
- Provides additional layer to enforce data quality

Cons:

- Slower then database to database migration

Questions?