24

Futureproofing Rich Metadata File Ingestion with OSDU

T. Hewitt[1], R. Gadrbouh[1]

[1] CGG

Summary

Acting as a technology-agnostic, standards-based data platform, the OSDU has reduced energy data silos and provided the capability for applications developers to build new solutions and data ingestion services.

The current OSDU schemas are primarily created to store file metadata to allow users to query common business content that can be extracted from the files. We utilized a machine-learning and subject matter expert classification process to auto-generate detailed file metadata for millions of files and ingest them directly to the user OSDU instance with source files.

The file classification process currently generates a graph database representation of files and rich metadata labels at a data-object level. The classification results, alongside data lineage and quality, are stored in OSDU work product components and datasets and ready to migrate to the OSDU platform.

The process prevents users having to manually fill or supply the file manifests during file ingestion to their OSDU implementation. With over 700 distinct data types and 250,000 entities of subsurface terminologies, millions of ingested files can be enriched with highly granular metadata manifests that guarantee rapid data search and access to high-quality data.

Futureproofing Rich Metadata File Ingestion with OSDU

Introduction

Acting as a technology-agnostic, standards-based data platform, the OSDU has reduced energy data silos and provided the capability for applications developers to build new solutions and data ingestion services.

The current OSDU schemas are primarily created to store file metadata to allow users to query common business content that can be extracted from the files. We utilized a machine-learning and subject matter expert classification process to auto-generate detailed file metadata for millions of files and ingest them directly to the user OSDU instance with source files.

The file classification process currently generates a graph database representation of files and rich metadata labels at a data-object level. The classification results, alongside data lineage and quality, are stored in OSDU work product components and datasets and ready to migrate to the OSDU platform.

The process prevents users having to manually fill or supply the file manifests during file ingestion to their OSDU implementation. With over 700 distinct data types and 250,000 entities of subsurface terminologies, millions of ingested files can be enriched with highly granular metadata manifests that guarantee rapid data search and access to high-quality data.

Method and/or Theory

The current OSDU ingestion service provides two ingestion workflows: default files ingestion without metadata, and the manifest based ingestion. Manifest based ingestion allows multiple files ingestion with associated metadata. The metadata can be in the form of work-product-components, reference data, dataset, or master data.

We leveraged the manifest based ingestion with customizing the API to enrich the file ingestion workflow with the following manifests:

1- Document work-product-component - this provides high level file metadata including the document type (e.g., special core analysis report, thin section image…etc.), the main subject covered by the document and the document detected languages.
2- File dataset - we store the classification labels in an array using the name and description properties in the File.Generic dataset to provide detailed description of the file content and allow a rapid yet robust file search.
3- File uniqueness - processibility and quality is delivered in the DataQuality work-product-component.
4- As files are processed further and valuable images have been identified as part of the document body, image segmentation is treated as an activity and lineage is maintained between the source document and the derivative images with the Activity and ActivityTemplate work-product-components.

The document work-product-component (WPC) is linked to the created dataset and the lineage assertion property is utilized in all the additional WPC manifests.

During the ingestion process, these manifests are validated against their definition schema. Once the integrity of the manifests is ensured, the manifests are ingested with the file to the user's OSDU instance. Figure 1 illustrates a high-level architecture overview of the file ingestion service.

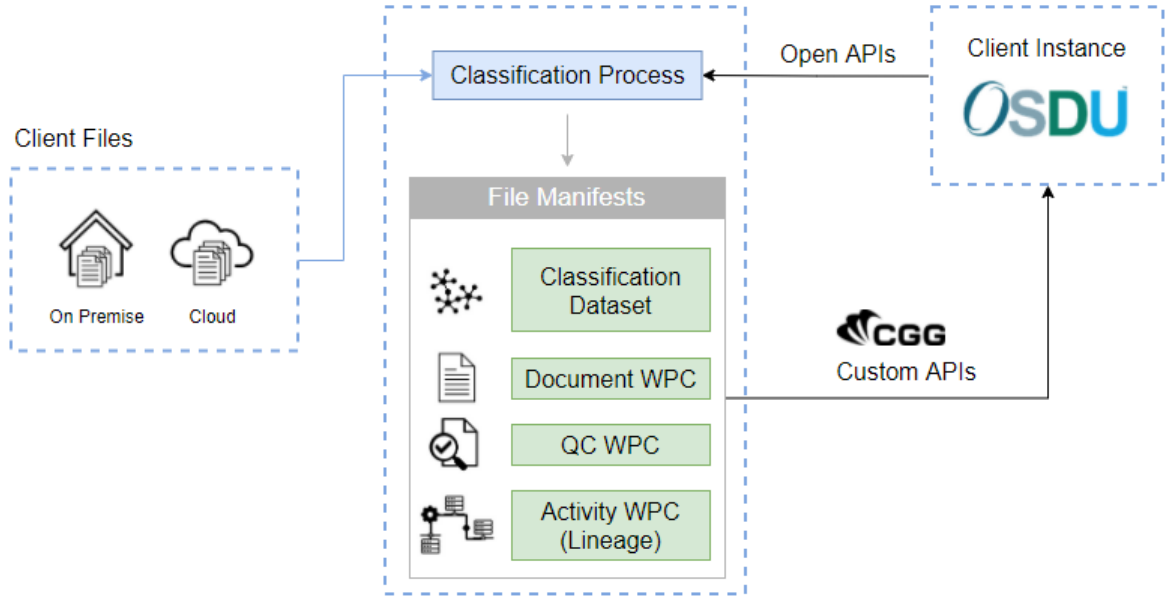


Figure 1 *High level architecture diagram of CGG Data Hub file ingestion service.*

Examples

The example file in Figure 2 is a special core analysis report according to the file title page. The classification process (Lun, et al., 2022) however resulted in more detailed context to the file as it contains geomechanical properties, XRD, and conventional core analysis. The image classification process has also identified thin section core photos that were segmented, cropped, and saved as individual derivative image files. The thin section images were processed with object classification machine learning workflow with the goal of predicting grain type and microfacies in addition to determining pore count and property statistics from thin section photomicrographs.

These file derivatives and the activities used to generate them must be recorded to maintain lineage and ensure data integrity. Hence, while the source document is ingested to the OSDU file storage, the classification process results are transformed into OSDU manifests and ingested with the source document (Figure 3).

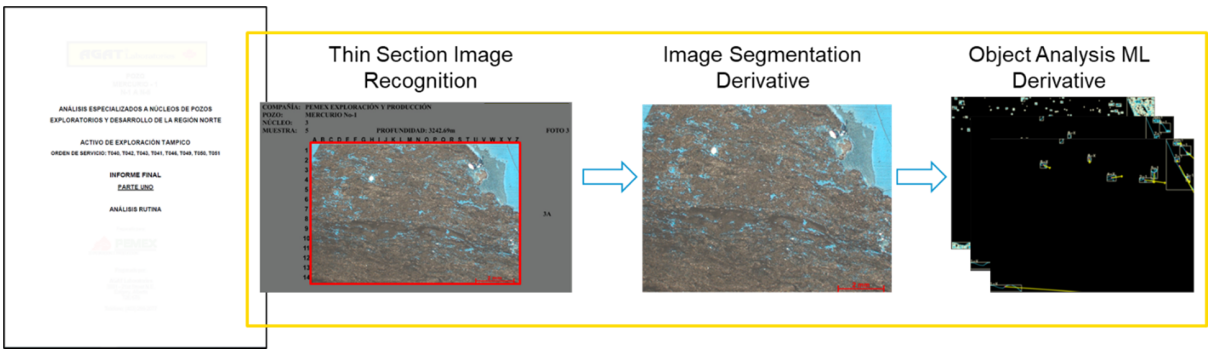
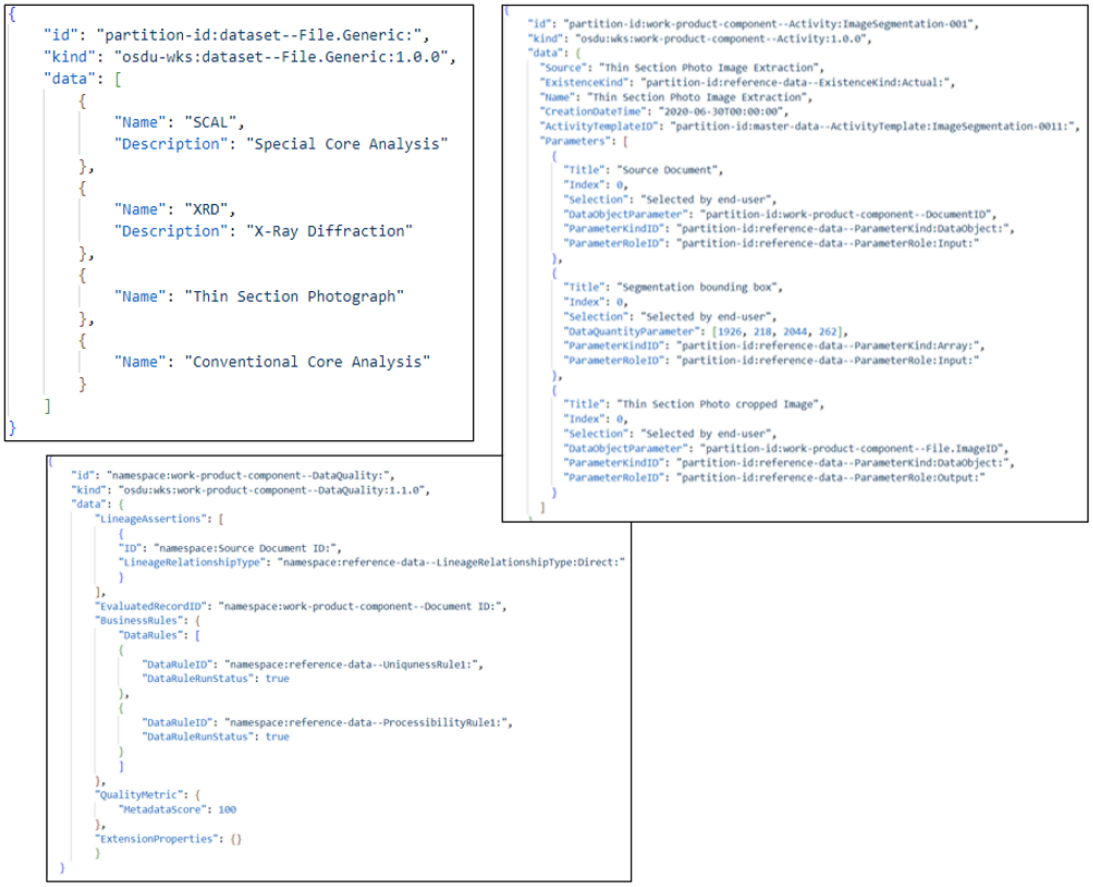


Figure 2 *File ingestion service example document. The document is a special core analysis report (in Spanish). The image classification process identified thin section images in the file, the images were segmented, cropped, and used in the object classification machine learning workflow. *Only important information (to this example) on the document title page are shown and the rest were hidden.*



```
{
    "id": "partition-id:dataset--File.Generic:",
    "kind": "osdu-wks:dataset--File.Generic:1.0.0",
    "data": [
        {
            "Name": "SCAL",
            "Description": "Special Core Analysis"
        },
        {
            "Name": "XRD",
            "Description": "X-Ray Diffraction"
        },
        {
            "Name": "Thin Section Photograph"
        },
        {
            "Name": "Conventional Core Analysis"
        }
    ]
}
```

```
{
  "id": "partition-id:work-product-component--Activity:ImageSegmentation-001",
  "kind": "osdu:wks:work-product-component--Activity:1.0.0",
  "data": {
    "Source": "Thin Section Photo Image Extraction",
    "ExistenceKind": "partition-id:reference-data--ExistenceKind:Actual:",
    "Name": "Thin Section Photo Image Extraction",
    "CreationDateTime": "2020-06-30T00:00:00",
    "ActivityTemplateID": "partition-id:master-data--ActivityTemplate:ImageSegmentation-0011:",
    "Parameters": [
      {
        "Title": "Source Document",
        "Index": 0,
        "Selection": "Selected by end-user",
        "DataObjectParameter": "partition-id:work-product-component--DocumentID",
        "ParameterKindID": "partition-id:reference-data--ParameterKind:DataObject:",
        "ParameterRoleID": "partition-id:reference-data--ParameterRole:Input:"
      },
      {
        "Title": "Segmentation bounding box",
        "Index": 0,
        "Selection": "Selected by end-user",
        "DataQuantityParameter": [1926, 218, 2044, 262],
        "ParameterKindID": "partition-id:reference-data--ParameterKind:Array:",
        "ParameterRoleID": "partition-id:reference-data--ParameterRole:Input:"
      },
      {
        "Title": "Thin Section Photo cropped Image",
        "Index": 0,
        "Selection": "Selected by end-user",
        "DataObjectParameter": "partition-id:work-product-component--File.ImageID",
        "ParameterKindID": "partition-id:reference-data--ParameterKind:DataObject:",
        "ParameterRoleID": "partition-id:reference-data--ParameterRole:Output:"
      }
    ]
  }
}
```

```
{
  "id": "namespace:work-product-component--DataQuality:",
  "kind": "osdu:wks:work-product-component--DataQuality:1.1.0",
  "data": {
    "LineageAssertions": [
      {
        "ID": "namespace:Source Document ID:",
        "LineageRelationshipType": "namespace:reference-data--LineageRelationshipType:Direct:"
      }
    ],
    "EvaluatedRecordID": "namespace:work-product-component--Document ID:",
    "BusinessRules": {
      "DataRules": [
        {
          "DataRuleID": "namespace:reference-data--UniqunessRule1:",
          "DataRuleRunStatus": true
        },
        {
          "DataRuleID": "namespace:reference-data--ProcessibilityRule1:",
          "DataRuleRunStatus": true
        }
      ]
    },
    "QualityMetric": {
      "MetadataScore": 100
    },
    "ExtensionProperties": {}
  }
}
```


Figure 3 *Example of three manifests that were ingested with the source file. The File.Generic dataset stores the classification results of different data types captured within the file, Activity WPC maintains the lineage properties to the thin section images segmentation, and the DataQuality WPC reflects the file uniqueness as the file has no duplicates. *Some IDs were removed from these manifests.*

Conclusions

The file ingestion service is the first service that leverages OSDU manifest based ingestion to provide a complete and enriched file metadata based on robust machine learning classification process with over 700 distinct data types and ensures a high integrity data search with quality and activity WPCs. OSDU has allowed us to futureproof our technology scale and speed and make it readily available to users with rapid data exchange between compatible platforms and software.

References

1. Open Subsurface Data Universe Software . Documentation . Wiki . Core Services Overview (opengroup.org).
2. Open Subsurface Data Universe Software . Data Definitions and Services . Data Definitions . Repository (opengroup.org).
3. Lun, C. H., Hewitt, T. & Hou, S., 2022. A Machine Learning Pipeline for Document Extraction. *First Break,* 40(2), pp. 73-78.