

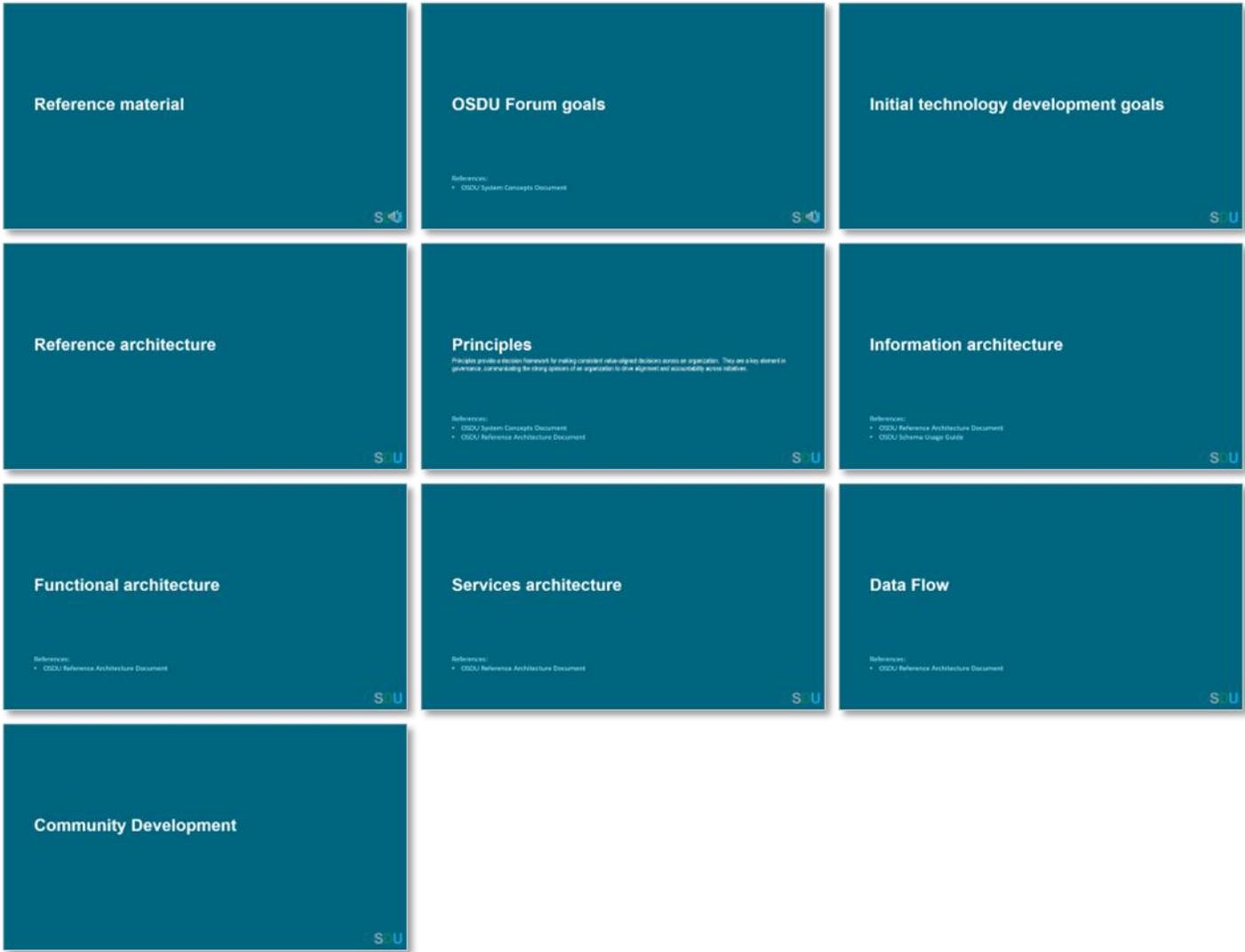


OSDU Data Platform Application Developer Training

Data Platform Overview



Topics



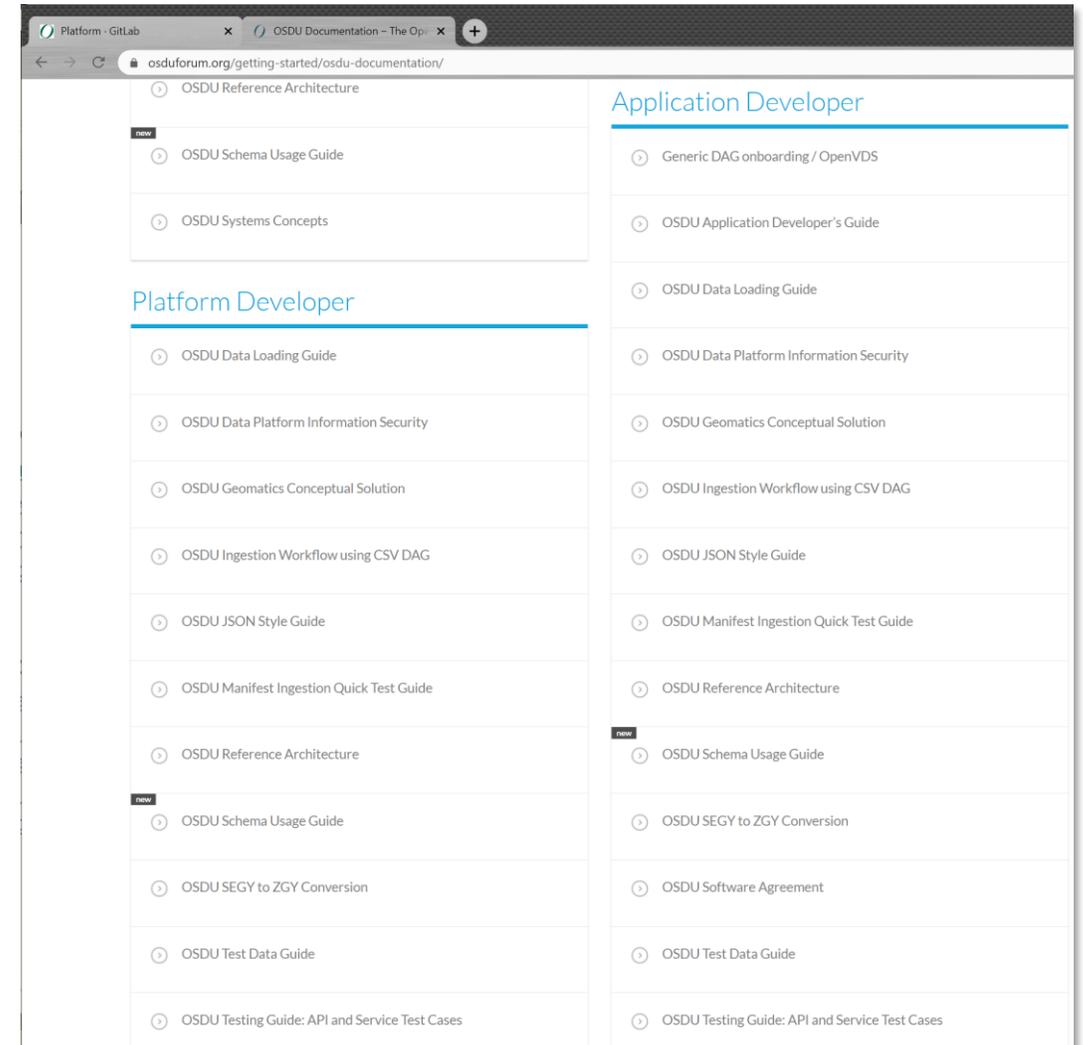
Reference material

Reference Material: OSDU Forum Documentation

OSDU Forum Documentation:

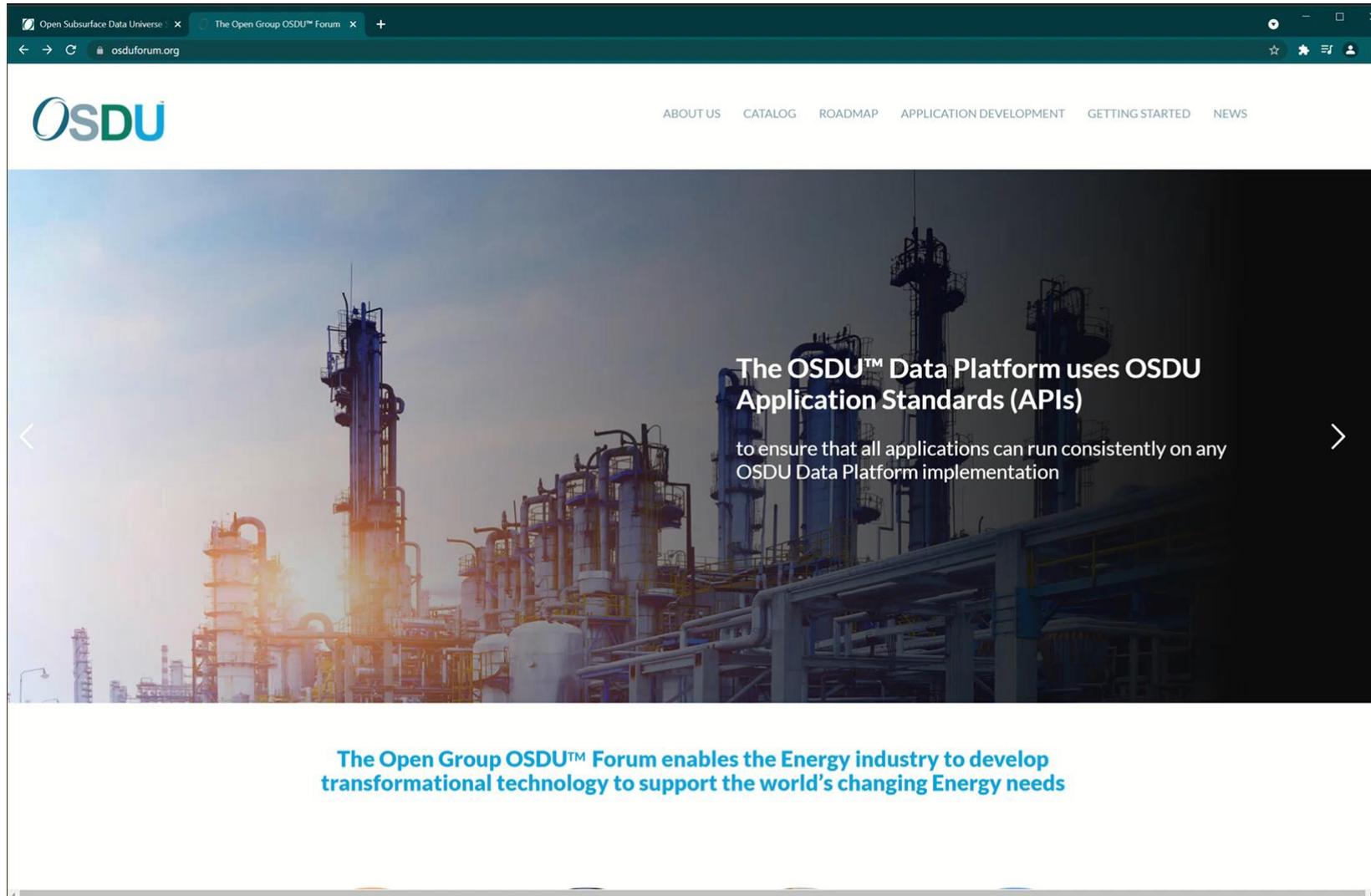
- [System Concepts*](#)
- [Reference Architecture*](#)
- [Schema Usage Guide*](#)
- [Testing Guide: API and Test Cases*](#)
- Etc.

<https://osduforum.org/getting-started/osdu-documentation/>



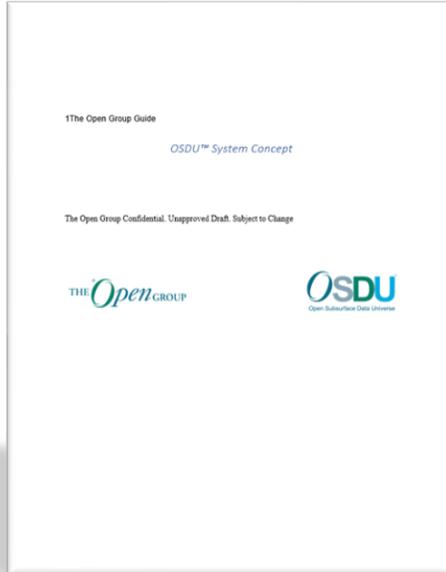
* Individual Documents Links will change as the standards and data platform progresses

Documentation at the osduforum.org website



Key Standards and Guides in the OSDU Forum

System Concept



Business context for OSDU Data Platform design and implementation

[Guide link](https://bit.ly/3aNrPXP)

<https://bit.ly/3aNrPXP>

Reference Architecture



Overall system architecture including design principles, data concepts, security and operations

[Ref. link](https://bit.ly/3dUmIr9)

<https://bit.ly/3dUmIr9>

Technical Standard

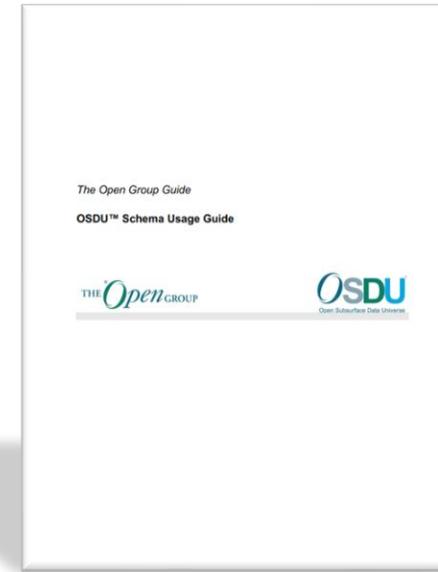


Guidelines, requirements, and recommendations that make up the OSDU Forum specification, so all applications can run on any OSDU Data Platform regardless of the Cloud Provider

[Standard link](https://bit.ly/3dGEUUQ)

<https://bit.ly/3dGEUUQ>

Schema Usage Guide



Design of OSDU Forum canonical schema and data definitions to ensure consistent usage in applications with goal of interoperability

[Guide link](https://bit.ly/3us3TKL)

<https://bit.ly/3us3TKL>

Application Development Guide



ISVs and developers guide on OSDU Data Platform concepts and approaches to build OSDU Data Platform based applications

[Guide link](https://bit.ly/3qSyRAH)

<https://bit.ly/3qSyRAH>

NB: Links point to draft copies: published links TBC

Copyright © The Open Group 2021

Implementation Space Documentation is in Gitlab

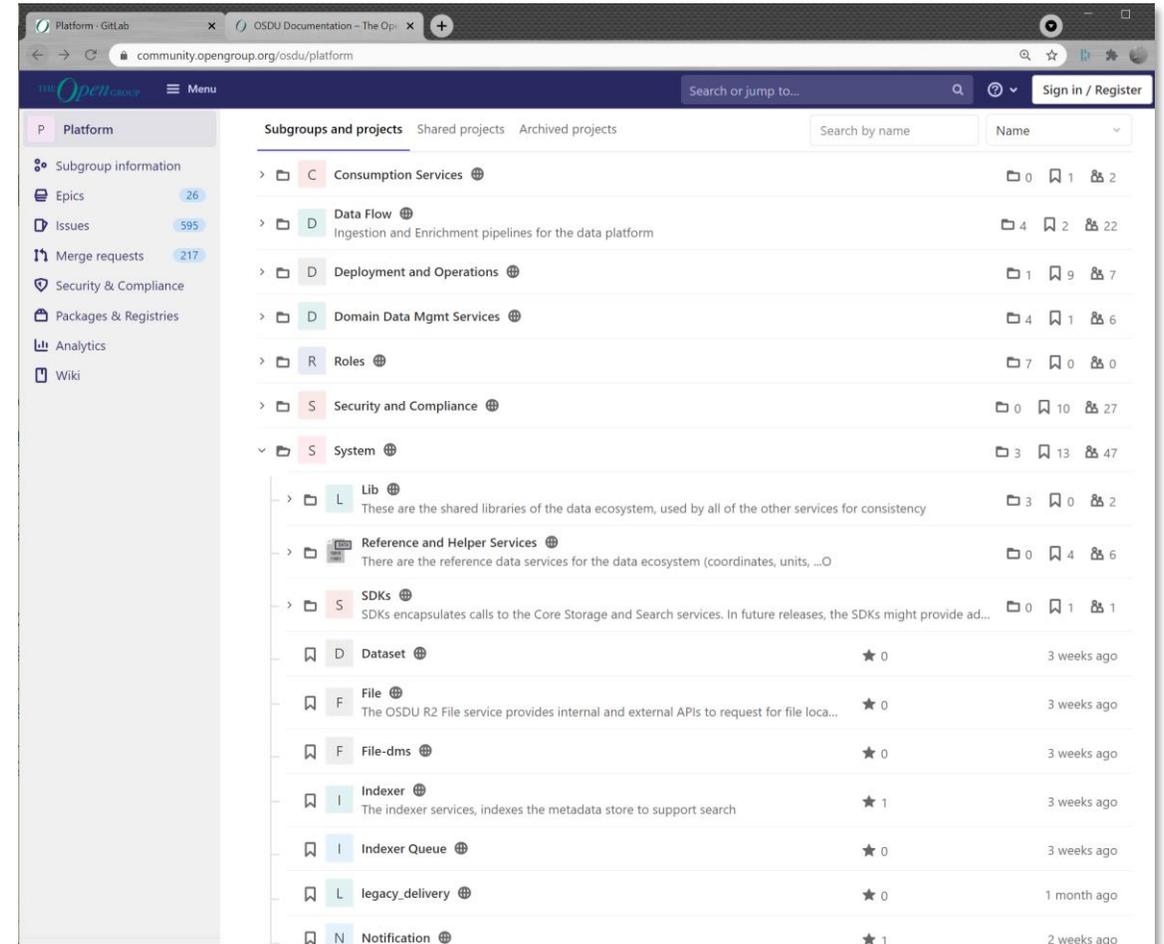


OSDU Data Platform Open-Source Gitlab

OSDU Data Platform Open Source GitLab**:

- [Data Platform](#)
- [System Services](#)
- [Data Services](#)
- [Domain Data Management Services](#)
- [Validation \(Test Cases\)](#)
- Etc.

<https://community.opengroup.org/osdu/>



** Versioned for all releases and contains latest developments

Community GitLab

The screenshot shows the GitLab interface for the 'Open Subsurface Data Universe Software' group. The left sidebar contains navigation options: Group information, Epics (38), Issues (697), Merge requests (227), Security & Compliance, Push Rules, Kubernetes, Packages & Registries, Analytics, Wiki, and Settings. The main content area displays the group name, a description, and recent activity statistics: 1257 Merge Requests opened, 451 Issues opened, and 61 Members added in the last 90 days. Below this is a list of subgroups and projects, including User Interfaces, Roles, Data Definitions and Services, Platform, Governance, Tutorials, Documentation, Runner Health Check, and OSDU Home.

Subgroup/Project	Stars	Last Updated
User Interfaces	0	1
Roles	2	0
Data Definitions and Services	0	2
Platform	7	9
Governance	0	1
Tutorials	0	8
Documentation	16	1 month ago
Runner Health Check	0	1 month ago
OSDU Home	10	2 months ago

Test Cases as an API Reference

<https://community.opengroup.org/osdu/platform/testing>

The screenshot displays the GitHub repository page for the OSDU Platform. The left sidebar shows the repository structure with 'Platform' as the main group. The main content area provides an overview of the platform, including recent activity statistics and a list of subgroups and projects.

Recent activity (Last 90 days):

Merge Requests opened	Issues opened	Members added
1256	445	53

Subgroups and projects:

Subgroup/Project	Stars	Last Updated
Consumption Services	0	1 month ago
Roles	7	0
Domain Data Mgmt Services	4	1 month ago
System (Owner)	3	13
Deployment and Operations (Owner)	1	9
Security and Compliance (Owner)	0	10
Data Flow (Owner)	4	2
Platform Validation	3	3 weeks ago
CI-CD Pipelines	5	2 weeks ago
Open Test Data (Developer)	2	4 months ago
Platform Architecture	0	4 months ago
Platform Home	5	1 month ago
bulk-loader	1	1 month ago
Retrospectives	0	1 year ago
Trusted MR Container	0	1 year ago
OSDU Data Definitions	0	11 months ago

OSDU Forum goals

References:

- [OSDU Data Platform System Concepts Document](#)

OSDU Forum Mission

Working with The Open Group to create certifiable standards to support interoperability

Using the Open Source model to promote collaboration

The Open Group OSDU Forum delivers an Open Source, standards-based, technology-agnostic data platform for the energy industry that stimulates innovation, industrializes data management, and reduces time to market for new solutions.

Starting in Subsurface Oil and Gas, but extending to support the wider needs of Energy Companies

“PolycLOUD” – runs on Public Clouds and on premises, taking advantage of the technology available on each

Not a data storage solution or a data management application - but a modern, cloud-native data platform

OSDU Data Platform: A Universal Data Repository

Supports *business workflows, applications and utilities*

- **With** discoverable, available, and usable data across all supported disciplines and business areas
- **Without** compromising the need for sufficient data quality and reliability
 - Generic APIs and Domain Optimized APIs
both work on the same basic data definition framework for identification and classification
- **Without** compromising the need for to achieve performance and scalability
 - Domain DDMS with Optimized APIs support workflow/application/DP interactions
 - Application/Data Platform interfaces are highly interactive in contrast to import/export/exchange

Acting as a true, single 'System of Record' the OSDU Data Platform is the preferred source for data access and for processes that create/derive new data.

Initial technology development goals

OSDU Data Platform

» Purpose:

- Break down barriers by making data available across domain and functional boundaries

... an integrated approach; breaking down barriers

» Scope: Upstream Oil & Gas

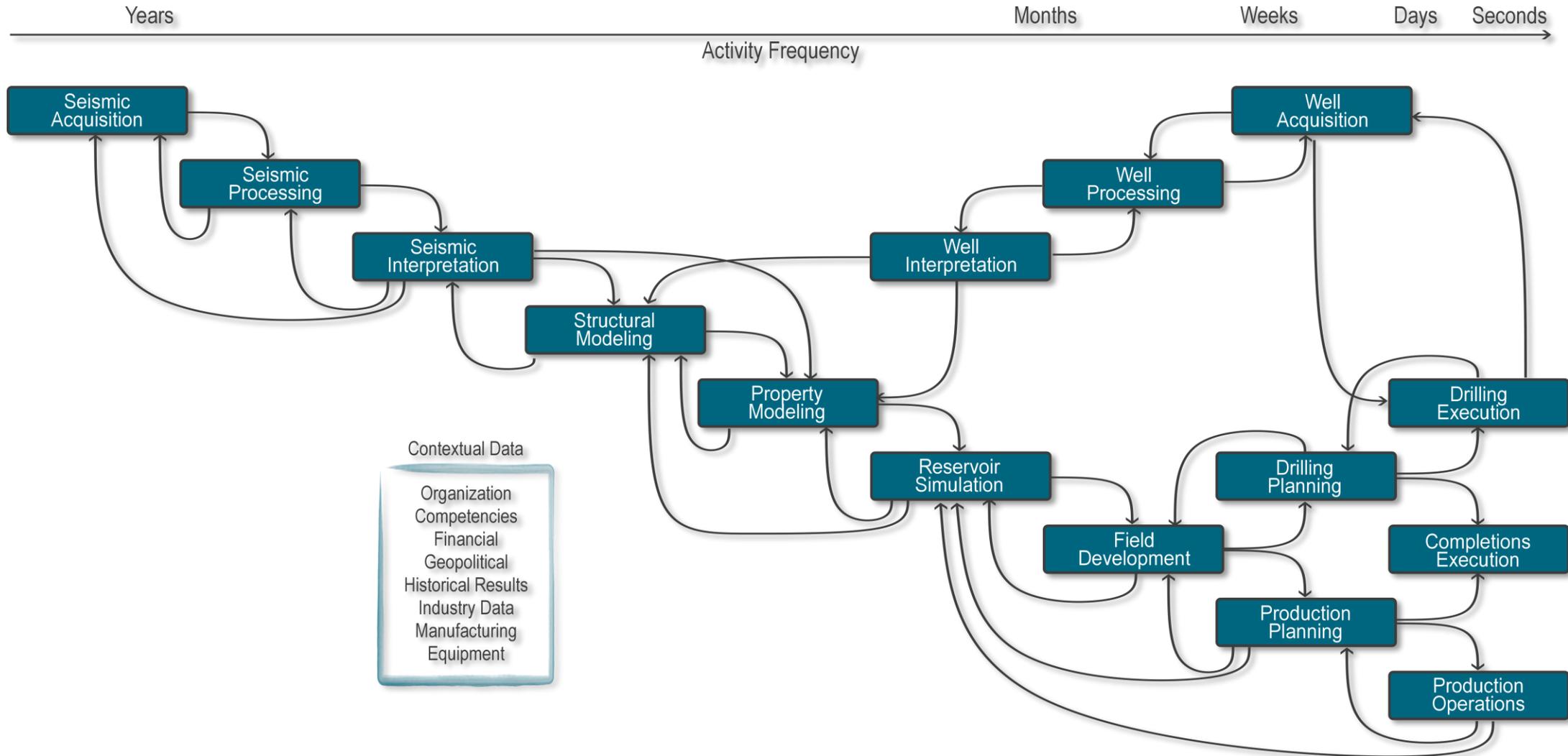
- Domains: Exploration to Production
- Functions: Planning and Operations

- among vertical silos;
- between planning and operations;
- between operators and suppliers; and
- past and future knowledge.

» Maturity:

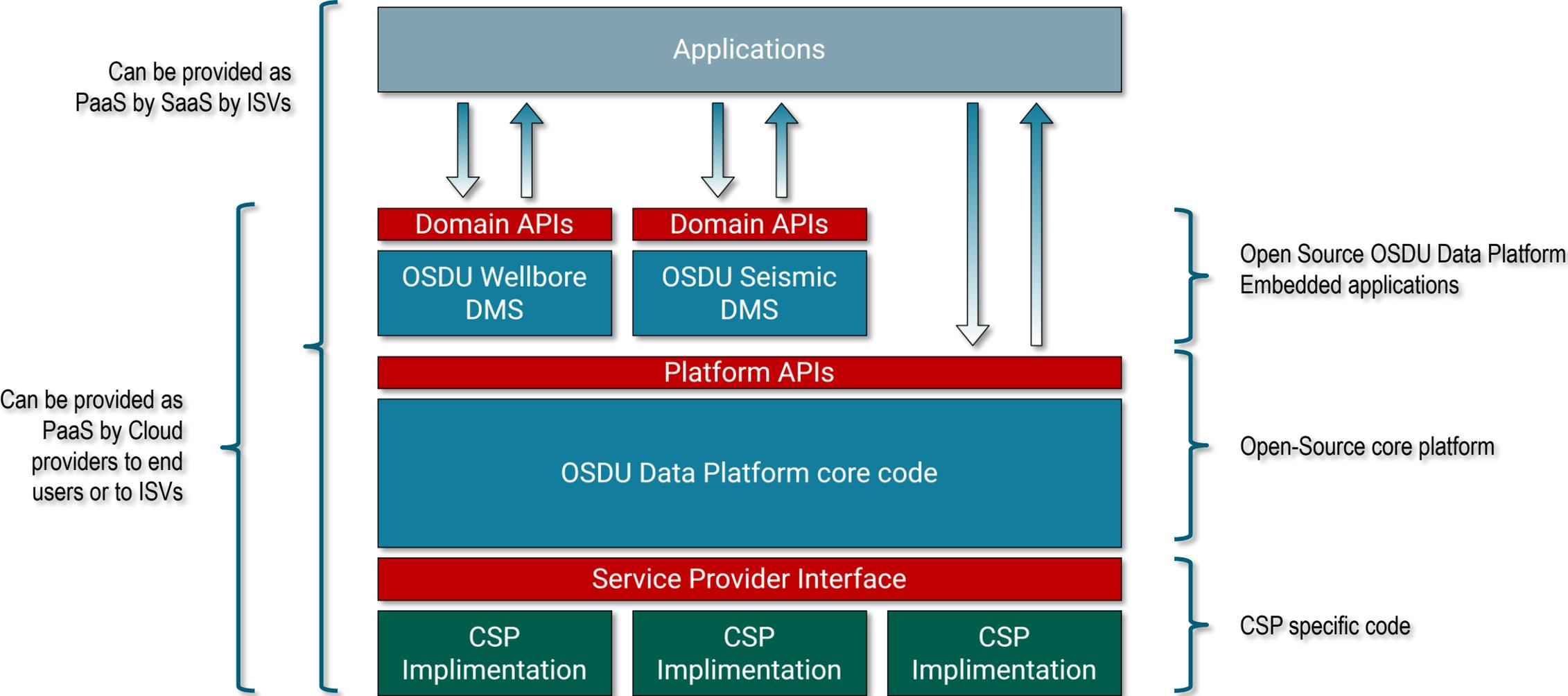
- In development since early 2015 on Google Cloud Platform (GCP)
- Platform technology was commercially available in DELFI and contributed to the OSDU Forum in 2019
- Aligned to OSDU Forum goals, Shell's initial contribution and other member contributions
- Ported to additional Cloud platforms by AWS, Microsoft and IBM
- Evergreen, continuous addition and improvement through continuous delivery

Initial data footprint driving design



Reference architecture

OSDU Data Platform Software Components



OSDU Data Platform Deployment Topology

Multi-Partition

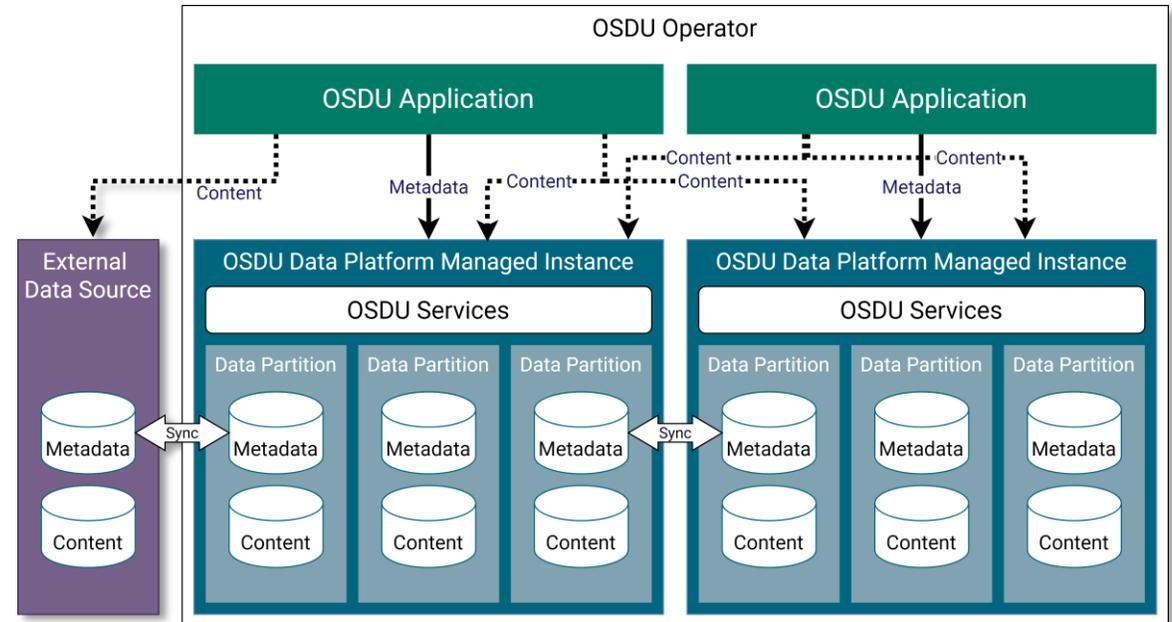
Data separated in multiple projects or resource groups for added security.

Multi-Region

Metadata (partially) replicated to support fast search; content data remains in region and managed by fully functional OSDU Data Platform instances.

External Data Source

Metadata (partially) replicated to support fast search; content data remains in external source and not managed by the OSDU Data Platform.



Principles

Principles provide a decision framework for making consistent value-aligned decisions across an organization. They are a key element in governance, communicating the strong opinions of an organization to drive alignment and accountability across initiatives.

References:

- [OSDU Data Platform System Concepts Document](#)
- [OSDU Data Platform Reference Architecture Document](#)

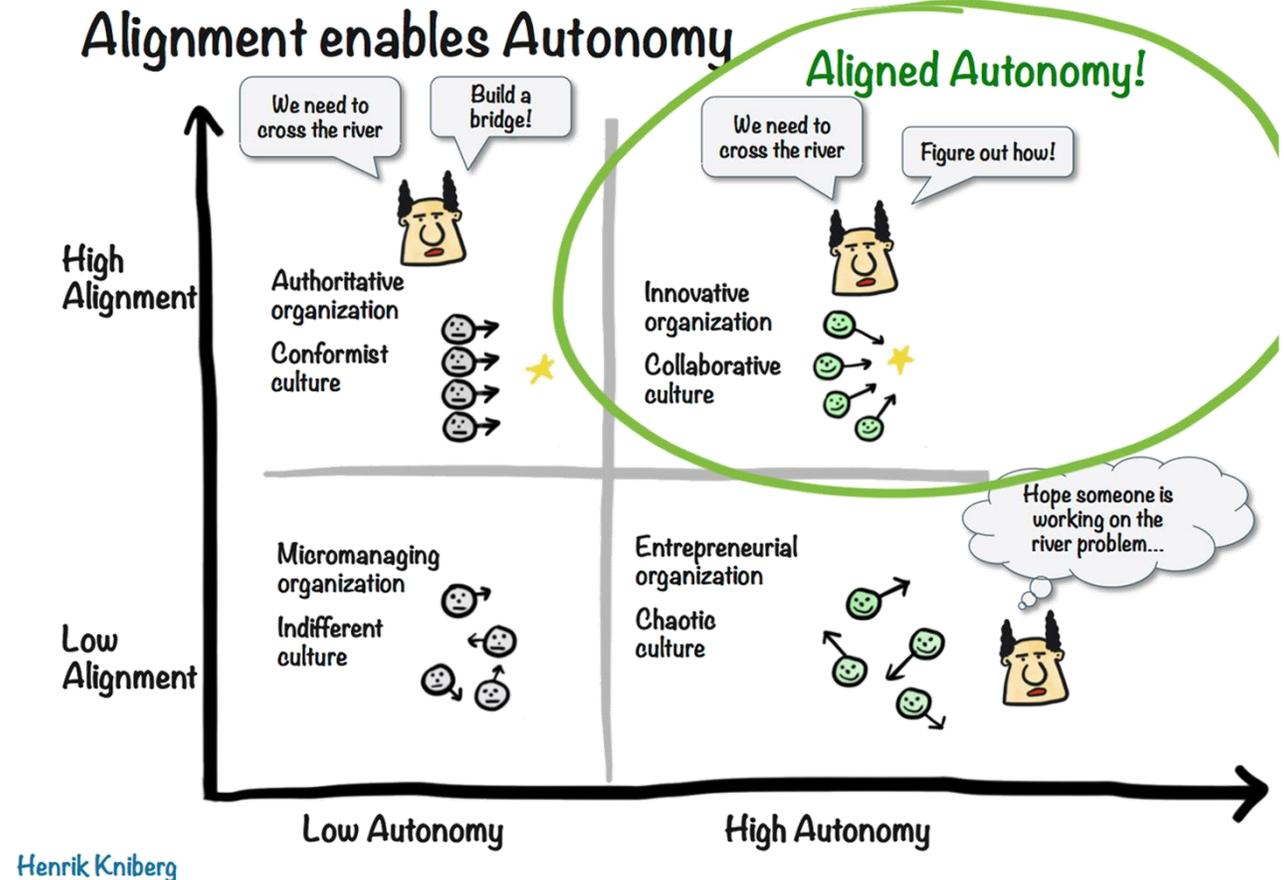
Open Source / Open Standards Principles

Aligned Autonomy

Maintain a balance between innovation and usability.

Intentional Architecture, Emergent Design

Always consider the future, but respond to current needs, context and feedback.



OSDU Data Platform: System Design Principles

Updated
05/2021

Agility

- Take a user centric approach
- Microservices architecture
- Aligned autonomous teams
- Continuous delivery and feedback

Security

- Make security everybody's concern
- Leverage cloud provider expertise
- Protect data as the most valuable asset
- Put platform Operator in control of data and users

DevOps

- Monitoring and Support
- DevOps/Delivery Infrastructure
- Robust and fault tolerant patterns
- Focus on fewest technologies: languages, frameworks, protocols, etc.
- Automation at every step of the way is key

Multi Cloud

- Avoid the lowest common denominator
- Leverage cloud native managed solutions where possible
- Adopt community (Open Source) APIs where applicable
- Clear separation between DP APIs and SPI through common code layer; intent to avoid CSP-specific functionality encroachment
- CSP implemented functions should be behaviorally symmetric

OSDU Data Platform: Data Principles

Value All Data

- Minimize friction on receipt of 'data'
- Data content and context as received into the data platform are preserved:
- Data in the Data Platform are immutable

Use Data Securely and Responsibly

- Data are access controlled
- Data are governed for right of use
- Data are owned and managed by the 'DP Operator' organization

Embrace Minimal Viable Governance

- Data are globally identifiable within and across data platform instances
- Data are classified
- Data are discoverable
- Data are consumable

Establish and Continuously Improve Data

- Data access with 2 modes:
 - Generic APIs for normal access
 - Domain DDMS APIs for optimized access within defined bounded context
- Improved data are new data: content and context
- Data versioning / lineage tracking is enabled

Information architecture

References:

- [OSDU Data Platform Reference Architecture Document](#)
- [OSDU Data Platform Schema Usage Guide](#)

Platform Data

Data within the platform can be segmented into the following forms.

- » **Content data:** Data that has business purposes outside of the data platform whether ingested from external systems or enriched within the data platform for external consumption.
- » **Resource Data:** Frequently referred to as Metadata and Policies, this data is produced and managed within the data platform for the purpose of managing and classifying Content data.
- » **Systems Data:** Data that is produced and managed within the data platform for the purposes of configuring, securing, and managing the behavior of the platform itself.

Content Data

Content is data that is produced and consumed in the course of business operations. It is unbounded in terms of scope and representations forcing the data platform to accommodate new content with minimal governance.

Definitions and representations are governed by the producing entity whether introduced from outside the data platform (ingestion), created inside the data platform (enrichment), or delivered to external users (consumption). When possible, these definitions and representations reflect the needs and consensus of a community of users to facilitate collaboration whether expressed as a formal standard or otherwise.

Examples:

- Geophysics formats (Seismic Measurements & Interpretations)
- Petrophysics formats (Wells, Wellbores, Logs)

Resource Data

Resource data, frequently referred to as Metadata, contains

system properties that are governed by the data platform and support system wide behavior and data lifecycle management.

```
},
"kind": {
  "description": "The schema identification for the OSDU resource object following the pattern {Namespace}:{Source}:{",
  "title": "Entity Kind",
  "type": "string",
  "pattern": "^[\\w\\-\\.]+:[\\w\\-\\.]+:[\\w\\-\\.]+:[0-9]+.[0-9]+.[0-9]+$",
  "example": "osdu:wks:master-data--Basin:1.0.0"
},
"version": {
  "description": "The version number of this OSDU resource; set by the framework.",
  "title": "Version Number",
  "type": "integer",
  "format": "int64",
  "example": 1562066009929332
},
"acl": {
  "description": "The access control tags associated with this entity.",
  "title": "Access Control List",
  "$ref": "../abstract/AbstractAccessControlList.1.0.0.json"
},
"legal": {
  "description": "The entity's legal tags and compliance status. The actual contents associated with the legal tags i",
  "title": "Legal Tags",
  "$ref": "../abstract/AbstractLegalTags.1.0.0.json"
},
"tags": {
```

data content properties that are defined by the content provider and provide a description of the content and relationships.

```
"properties": {
  "BasinID": {
    "description": "A system generated unique identifier.",
    "type": "string"
  },
  "BasinName": {
    "description": "Name of the Basin.",
    "type": "string"
  },
  "BasinNameAliases": {
    "description": "Alternative names, including historical, by which this basin is/has been known.",
    "type": "array",
    "items": {
      "$ref": "../abstract/AbstractAliasNames.1.0.0.json"
    }
  },
  "BasinDescription": {
    "description": "A textual description of the Basin.",
    "type": "string"
  },
  "ProspectFlag": {
    "description": "Indicator showing whether the basin is considered prospective for hydrocarbons",
    "type": "boolean"
  }
}
```

Data Basics: Supporting aligned autonomy

Autonomy through user defined “*Kinds*”

- » “*Kinds*” define the representation of data in the OSDU Data Platform, identified by `[Namespace [.partition],DefinitionAuthority-Source:GroupType.IndividualType:Version]`, Where:
 - **Namespace** identifies an Organizational instance
 - **DefinitionAuthority-Source** defines the organization with authority over the data definition; different companies often have specialized data/metadata model definitions even if based on the same industry/OSDU Forum reference/standard definitions
 - **GroupType** is the basic type of data: master, reference, work product component, etc. for the purpose of classification.
 - **IndividualType** is the specific type: Well, Well Log, Seismic 3D Survey, etc.

Alignment for consistency within each OSDU Data Platform instance through “*Well Known Schemas (WKS)*”;

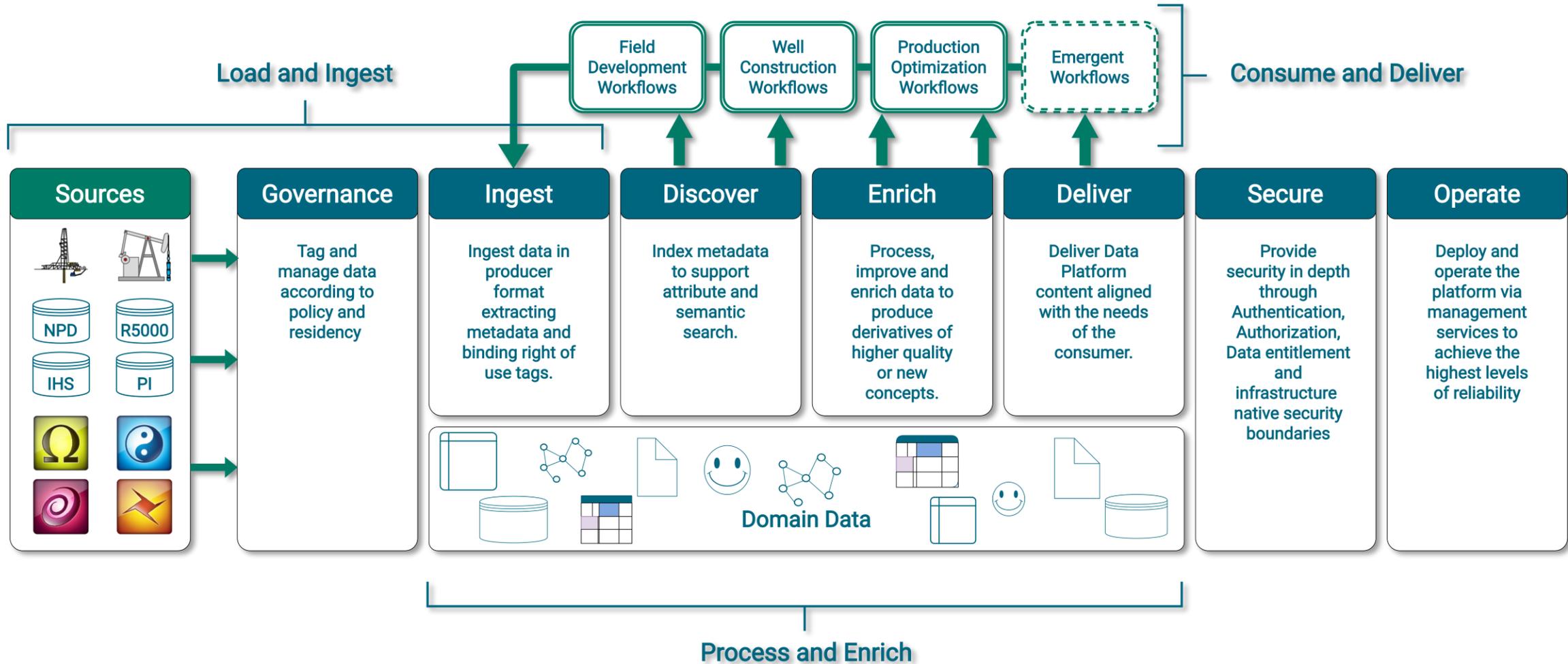
- » A *WKS* is a standardized data definition associated with a 'type' of the data to support understanding and/or processing data for discovery, exchange and consumption.
- » Non-*WKS* data definitions are useful to accommodate data defined for specific purposes: external licensed data, data from legacy systems, etc.

Functional architecture

References:

- [OSDU Data Platform Reference Architecture Document](#)

Functional Architecture

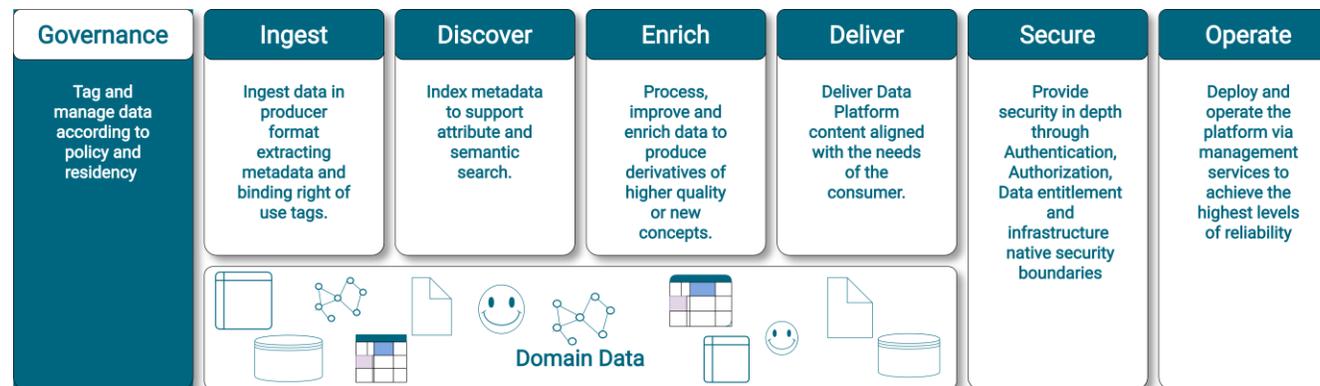


Governance (Entitlements and Obligations)

All data entering the Data Platform are tagged with attributes that allow us to make decisions on right of use.

These tags broadly capture data ownership, data security and information related to contractual obligations, data residence and trade compliance.

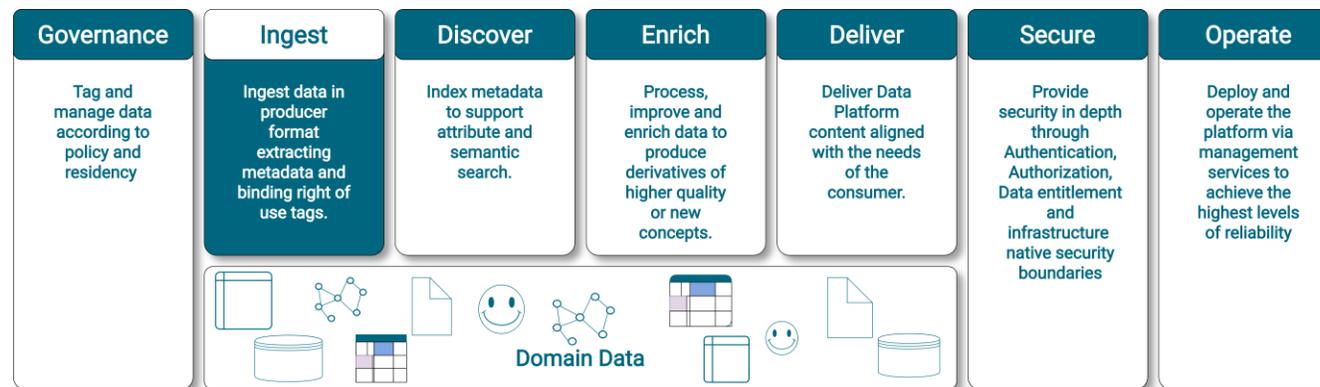
These tags, combined with contextual information about the user, location, the application, and other factors are used when searching and accessing data to determine whether to grant or block access.



Ingest

Ingestion is the act of absorbing information. It can be implemented as either registering data in (ingest by reference) or adding data (ingest by copy) to the OSDU Data Platform. The process focuses on minimizing friction and maximizing the amount of information that can be captured.

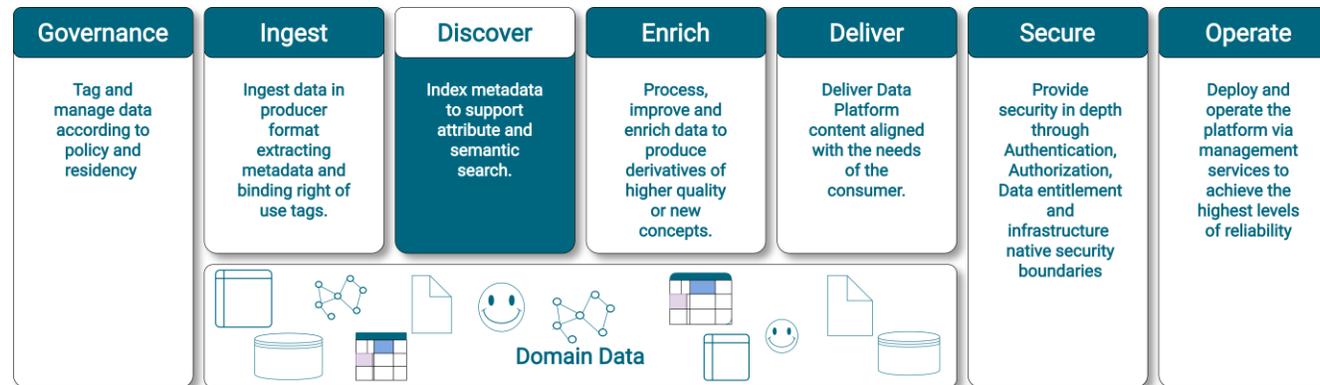
The act of ingestion and the logical layer representing ingestion to the OSDU Data Platform should not be mistaken for a singular implementation. Ingestion is a contract with well-defined rules. Any entry of the data in the system should follow this contract. If not, concerns such as compliance and lineage cannot be ensured.



Discover

Discovery is searching for data that has been ingested into the OSDU Data Platform. Discovery supports full text search on string fields, range queries on date, numeric and geo-spatial search.

The metadata that is searched is created by indexing documents that contain structured or unstructured data. Documents and indices are saved in a separate persistent store optimized for search operations. The indexer captures attributes defined in the data definition which is created at the time of record ingestion in OSDU Data Platform.

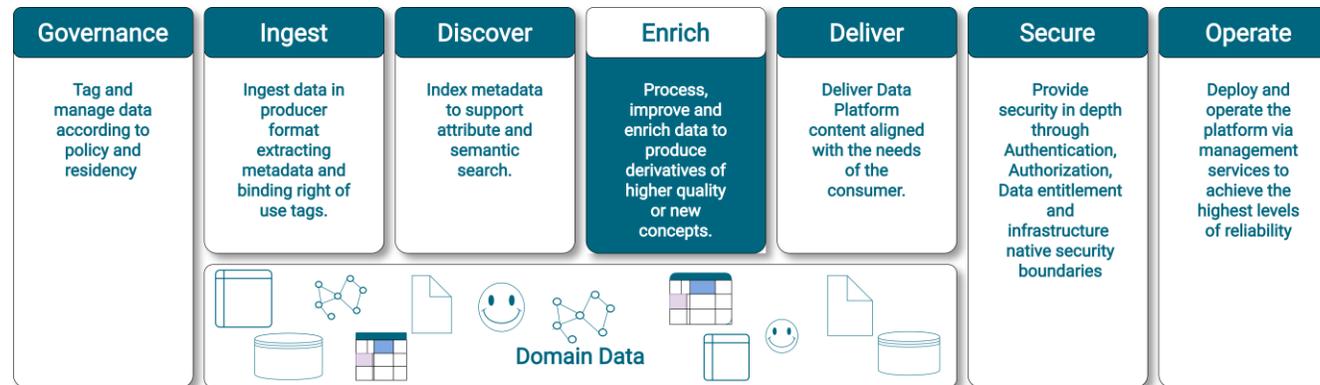


Enrich

Enrichment is creating new data from existing data. This can be as simple as normalizing frame of reference to support search to as complicated as creating new types by federating data from several different sources. In all cases, the new data is considered derived from existing data and thus treated as new data with lineage maintained between the derived data and its source

Examples of how enrichment is used in the OSDU Data Platform include:

- Normalizing frame of reference to support search.
- Creating higher quality records from multiple sources
- Optimize data definition and representation for consumption workflows.
- Fact extraction from documents.



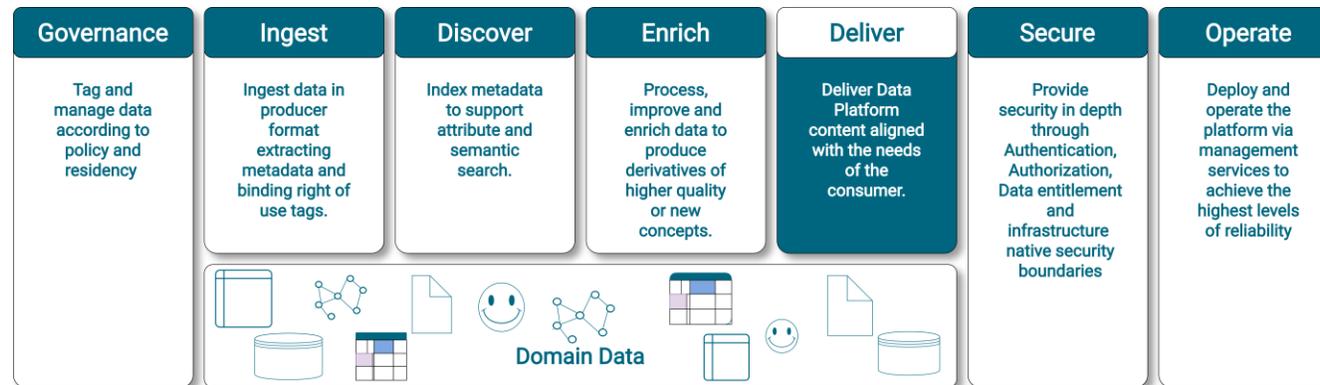
Deliver

The primary value of a data platform is realized at the time of consuming data.

Almost everything mentioned prior to this, whether it be ingestion of new data or enrichment of existing data, represents investment/cost for the purpose of making data available and usable.

There are multiple consumption patterns in the data platform, including:

- Content Delivery from (copied-in or referenced) files, etc.
- Consumption Zones
- Domain Data Management Service (DDMS)



Deliver: Content as a file

OSDU Data Platform resource and content data can be delivered as a set of packaged resources that will be taken outside the data platform. This “Pack N Go” consumption model allows external systems to access data which can then be ingested.

- » The advantages of this approach are that it
 - is highly flexible, allowing the addition of new data types without the need to introduce custom code that is type specific.
 - caters to applications that are not directly integrated with the platform providing a low-cost way of leveraging the data.
- » The disadvantages of this approach are that applications will not be updated with new or enriched data and frequently these applications are not obligated to bring data back to the platform.

Deliver: Consumption Zones

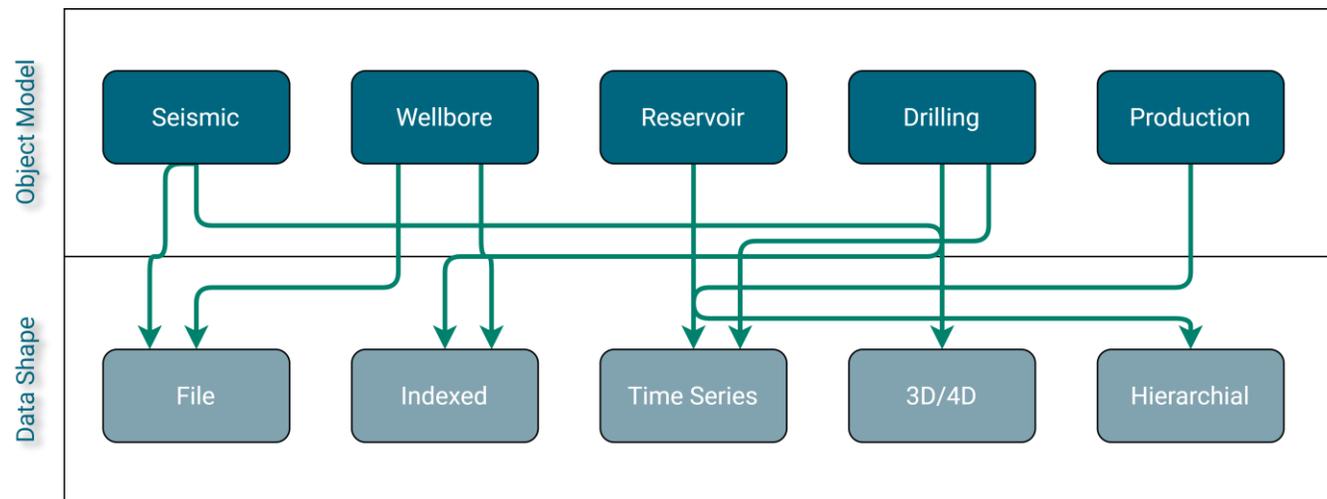
OSDU Data Platform resource and content data can be consumed via a read-only consumption zone. The data in this consumption zone is synchronized with the data platform and optimized for specific purposes. One example of this is search; where indexes are built from the metadata store and automatically updated as new data is ingested or existing data enriched.

- » The advantage of this approach is that it is fit-for-purpose consumption of data in the platform, which is maintained by the data platform itself, so information is kept up to date.
- » The disadvantages include:
 - Data is often duplicated to achieve the fit-for-purpose goal
 - Building consumption zones is frequently a development activity since it is exceedingly difficult to generalize fit-for-purpose.

Deliver: DDMS

A DDMS is a set of services, representations and APIs that are optimized for a specific domain and adheres to all the obligations of the OSDU Data Platform. They are introduced into the platform to simplify the development of applications running against the platform and to improve the experience of the user in terms of capability, performance, and scalability.

Unlike Consumption Zones, the DDMS provides both read and write capability while honoring platform obligations such as immutability, discovery, and entitlement.



Services architecture

References:

- [OSDU Data Platform Reference Architecture Document](#)

Architectural Style

The Architectural style of the OSDU Data Platform is microservices that have both common code to capture business logic and infrastructure specific code to optimize performance, cost, and sustainability.

Within the OSDU Data Platform, service boundaries are defined by domain driven design and each service is developed, deployed and versioned independently allowing asynchronous delivery and evolution.

Anatomy of an OSDU Data Platform Service

OSDU Application

An Application that consumes the OSDU Data Platform API

OSDU Data Platform API

An API Defined by the OSDU Forum and subject to a standard

OSDU Data Platform Common Code

Code that represents the behavior of the OSDU Data Platform that is common across all Providers

- The ability to understand and manage OSDU Data Platform (meta)data
- The ability to ingest and enrich OSDU Data Platform (meta)data
- The ability to index and search this OSDU Data Platform (meta)data
- The ability to entitle OSDU Data Platform (meta)data based on corporate or regulatory requirements
- The ability to deliver OSDU Data Platform (meta)data in a form that is consumable in Oil & Gas
-

Service Provider Interface

An API used by OSDU Data Platform Common Code to access Provider Specific Technologies

Provider Technologies

Technology deployed and often optimized for a specific cloud platform

- Storage
- Network (VPC)
- ...

Provider Operations

Cloud Operations (monitoring, recovery) that are optimized for Provider Technology

Structure of a Service Repository

API & Documentation

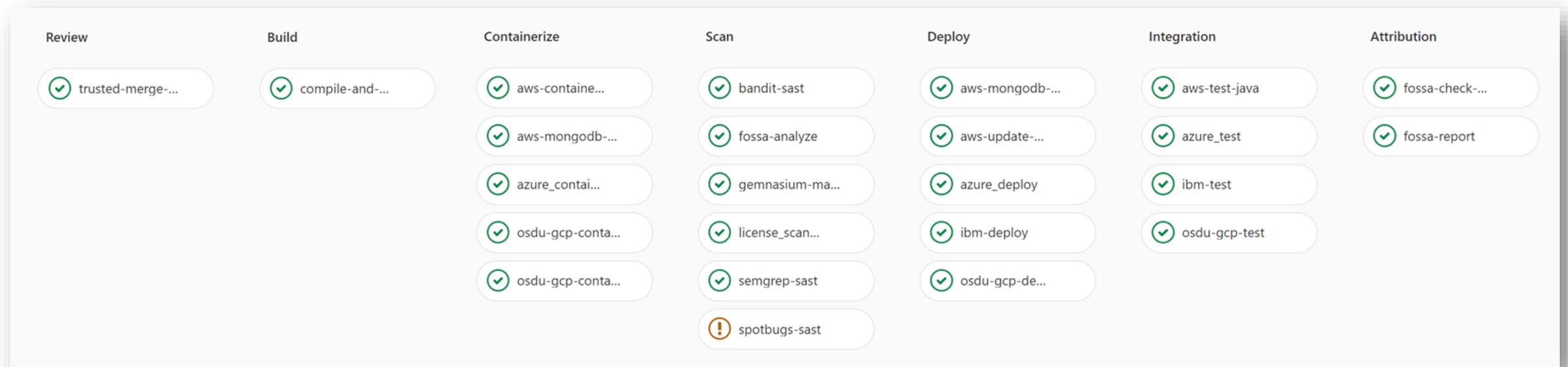
Service Provider Implementation

Common Code

Apache 2 License

```
3 // Licensed under the Apache License, Version 2.0 (the "License");
4 // you may not use this file except in compliance with the License.
5 // You may obtain a copy of the License at
6 //
7 // http://www.apache.org/licenses/LICENSE-2.0
8 //
9 // Unless required by applicable law or agreed to in writing, software
10 // distributed under the License is distributed on an "AS IS" BASIS,
11 // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
12 // See the License for the specific language governing permissions and
13 // limitations under the License.
14
15 package org.opengroup.osdu.storage.api;
16
17 import javax.validation.Valid;
18
19 import org.opengroup.osdu.core.common.model.storage.*;
20 import org.opengroup.osdu.core.common.model.storage.validation.ValidKind;
21 import org.opengroup.osdu.storage.service.BatchService;
22 import org.opengroup.osdu.storage.util.EncodeDecode;
23 import org.springframework.beans.factory.annotation.Autowired;
24 import org.springframework.http.HttpStatus;
25 import org.springframework.http.ResponseEntity;
26 import org.springframework.security.access.prepost.PreAuthorize;
27 import org.springframework.validation.annotation.Validated;
28 import org.springframework.web.bind.annotation.GetMapping;
29 import org.springframework.web.bind.annotation.PostMapping;
30 import org.springframework.web.bind.annotation.RequestBody;
31 import org.springframework.web.bind.annotation.RequestMapping;
32 import org.springframework.web.bind.annotation.RequestParam;
33 import org.springframework.web.bind.annotation.RestController;
34 import org.springframework.web.context.annotation.RequestScope;
35
36 @RestController
37 @RequestMapping("query")
38 @RequestScope
39 @Validated
40 public class QueryApi {
41
42     @Autowired
43     private BatchService batchService;
44
45     @Autowired
46     private EncodeDecode encodeDecode;
```

Service Build Pipeline



API and Service Landscape

Data Services

These are the data-oriented services of the data platform. They include:

- Helper services, such as Units and Coordinates
- Data Ingestion services
- Data Enrichment services
- Data Consumption services
- Domain Data Management Services

These services will expand as the data footprint of the OSDU Data Platform grows and can be continuously developed and deployed against an installed OSDU Data Platform

Platform Services

These are the core services of the data platform and

- Ensure the proper behavior of the OSDU Data Platform
- Are agnostic of data type and thus can be used to support any data

They include support for:

- Security and Compliance
- Metadata definition and storage
- Indexing and search
- Deployment and operations

These services are relatively few and have a strong stability promise

Data Services (Generic vs. DDMS)

Generic APIs are the foundation that our OSDU Data Platform is built on.

They

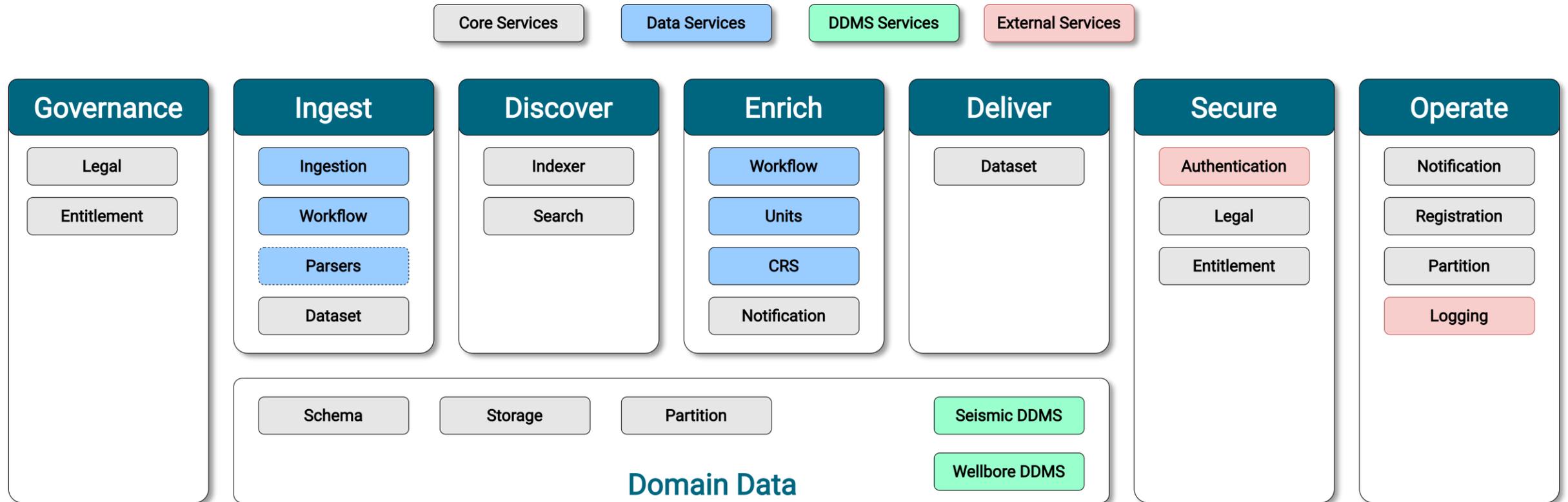
- » are published as open endpoints to access metadata and data as generic types
- » allow the continuous addition of new data types
- » support an ecosystem of:
 - System developers extending the platform
 - Systems integrators integrating the platform
 - Customers adopting the platform

Domain APIs provide an (E&P) domain specific way of accessing the system.

They carry semantics not possible in a generic API, making them a powerful tool for application developers who want the very best performance for accessing data.

We may need to update this type-safe optimized API every few years, but they allow storage optimization based on access patterns over generic APIs.

Mapping Services to Function



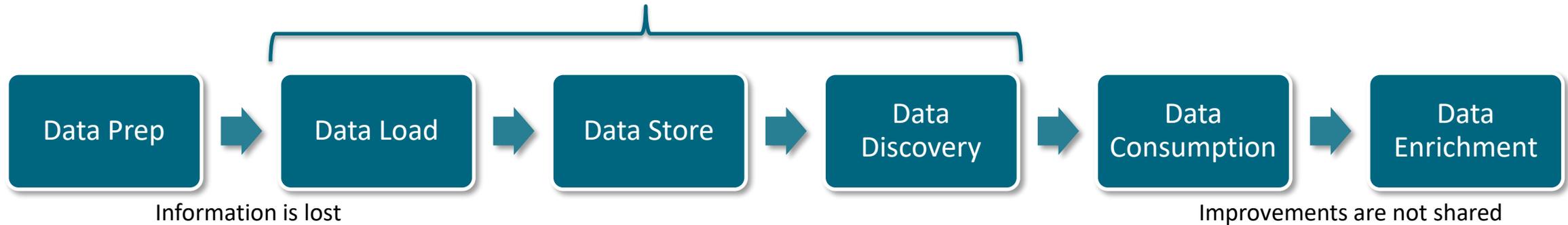
Data Flow

References:

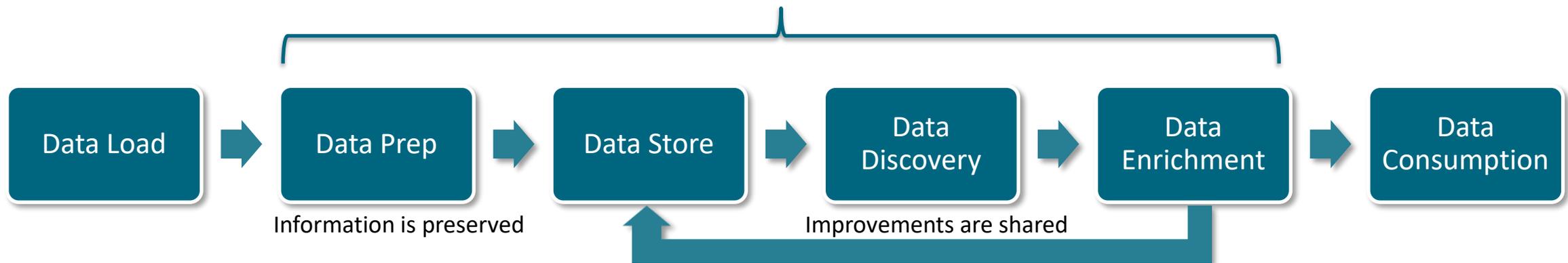
- [OSDU Data Platform Reference Architecture Document](#)

Data Flow Patterns (ETL & ELT)

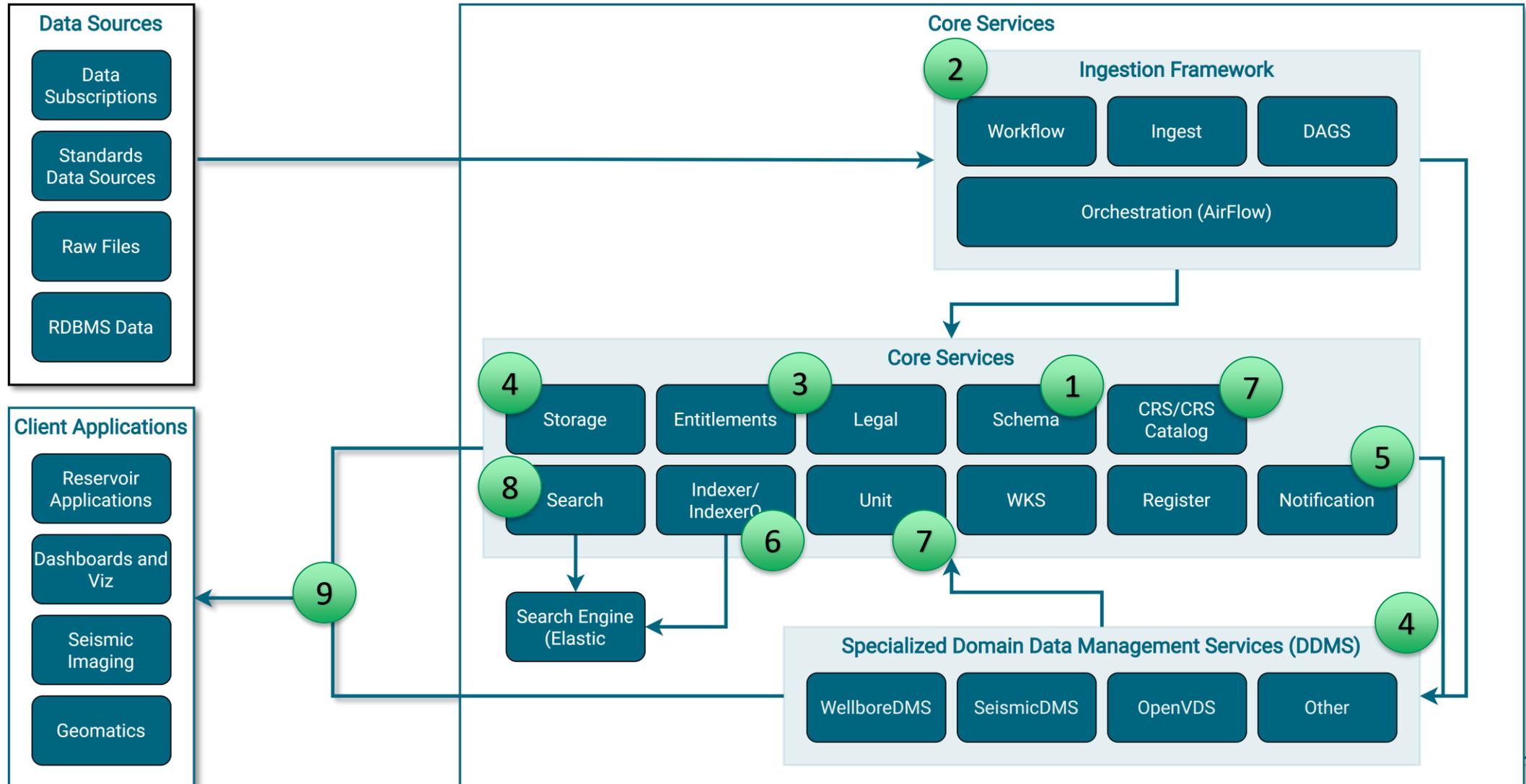
Corporate Repository Approach



The OSDU Data Platform Approach



Data Flow: Services Involved



Community Development

Open Source

The development of the data platform

- is licensed under Apache 2.0 ([Policy](#)),
- developed in GitLab ([Home Project](#)) as a set of microservices
- according to a [PMC governance model](#)

Contribution at Platform level

Active people ?

212

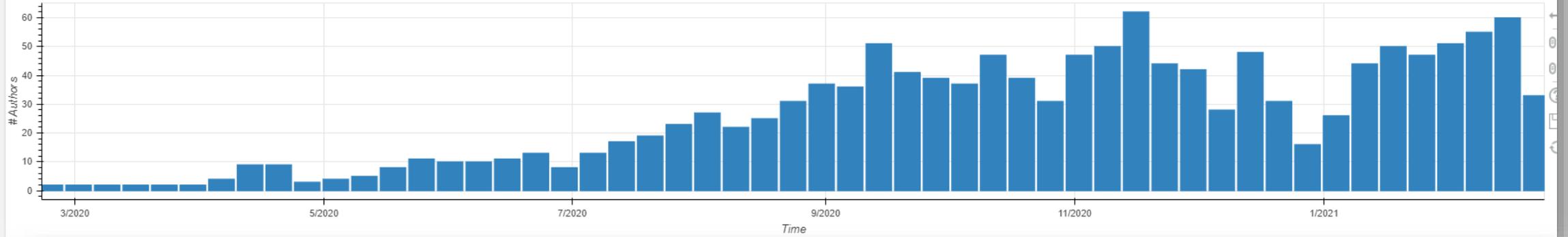
Git authors

Onboardings ?

204

Git authors

Active authors (Git)



Commits ?

6232

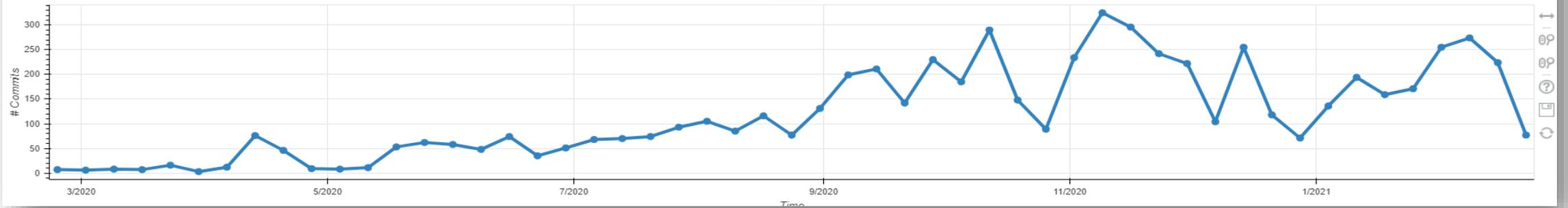
Lines/commit ?

729.01

Lines/commit/file ?

0.02

Commits



Training Agenda

Application Development Training Agenda

OSDU Data Platform Overview

OSDU Data Platform Lectures

- Core Platform APIs
- Ingestion Workflow
- DDMS APIs

OSDU Data Platform Labs

- Data Lifecycle Lab
- Wellbore DMS Lab
- Seismic DMS Lab