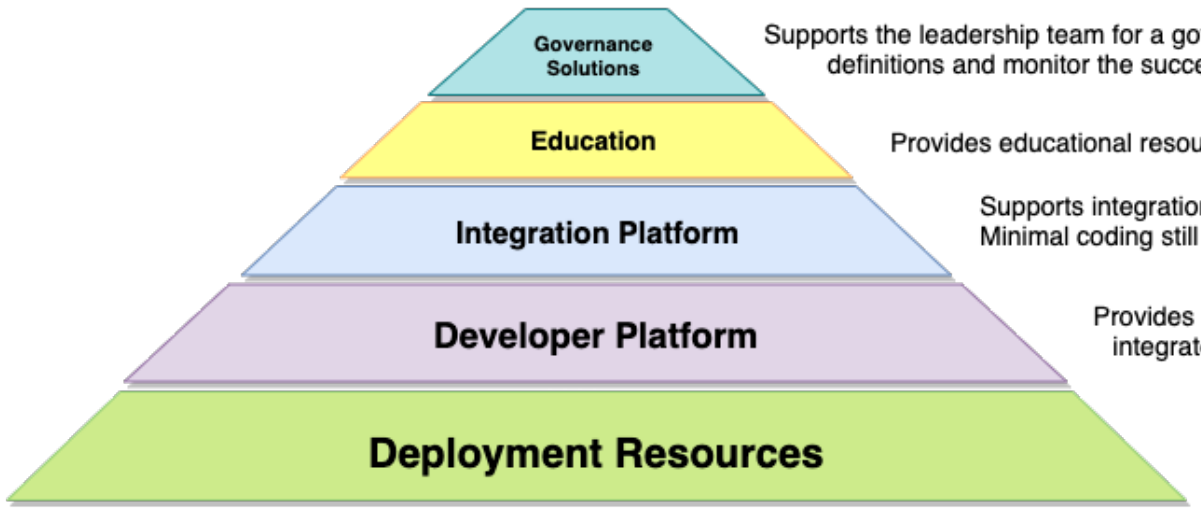
Egeria Subject Area OMAS

David Radley



Supports the leadership team for a governance program providing the ability to create common definitions and monitor the success of the governance efforts across the enterprise.

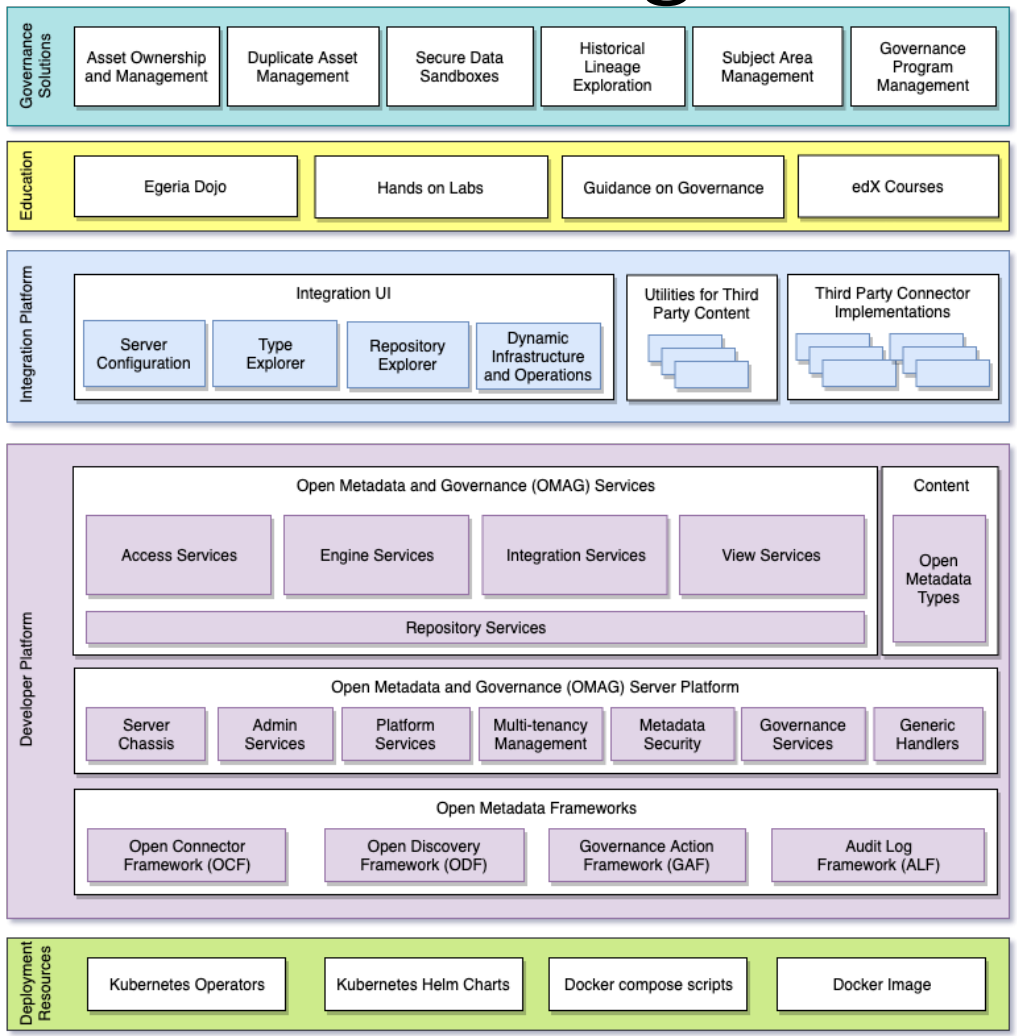
Provides educational resources for different personas and starting points.

Supports integration of popular technologies by installing and configuring Egeria. Minimal coding still required around unusual and home-grown tools and technologies.

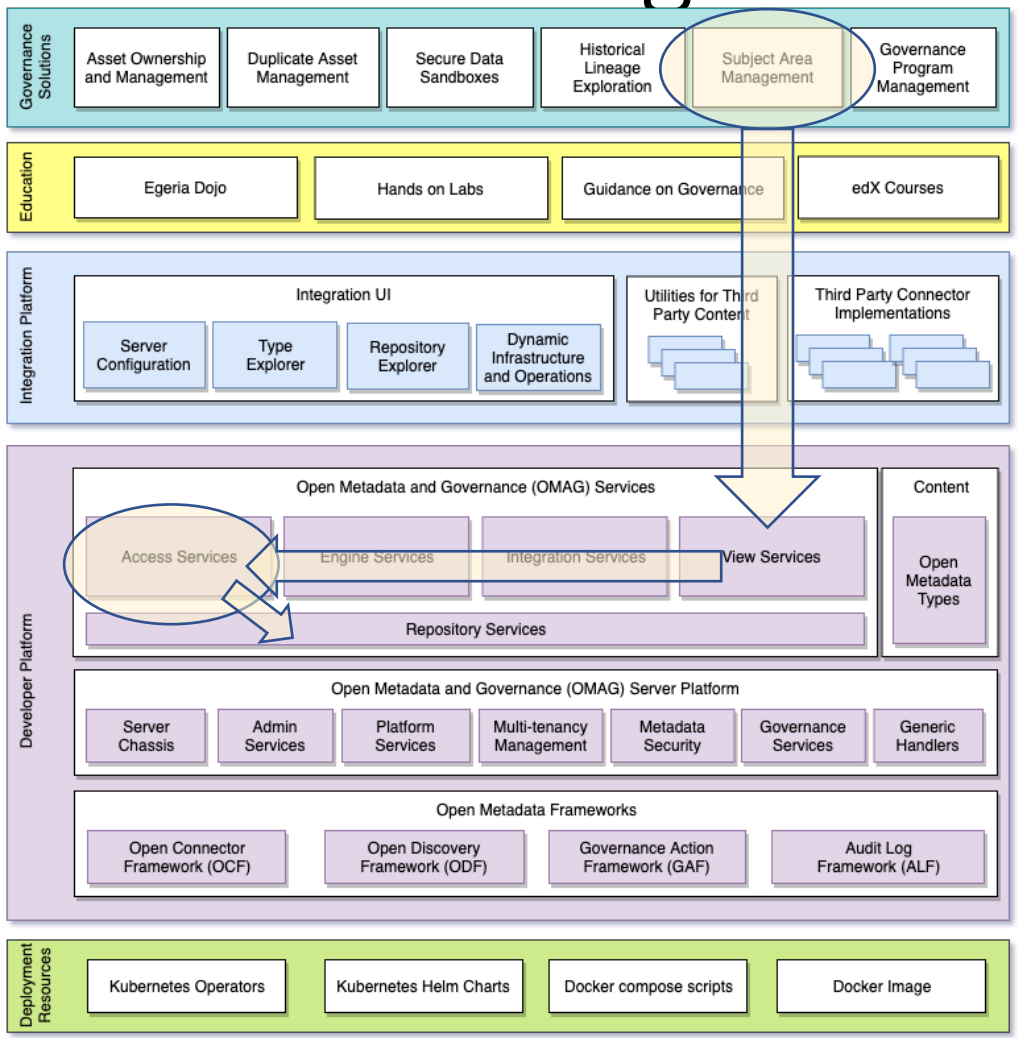
Provides frameworks, APIs, hosting platforms for building an integrated metadata and governance distributed solution.

Tools and components to ease the process of deploying Egeria into production.

Functional organisation



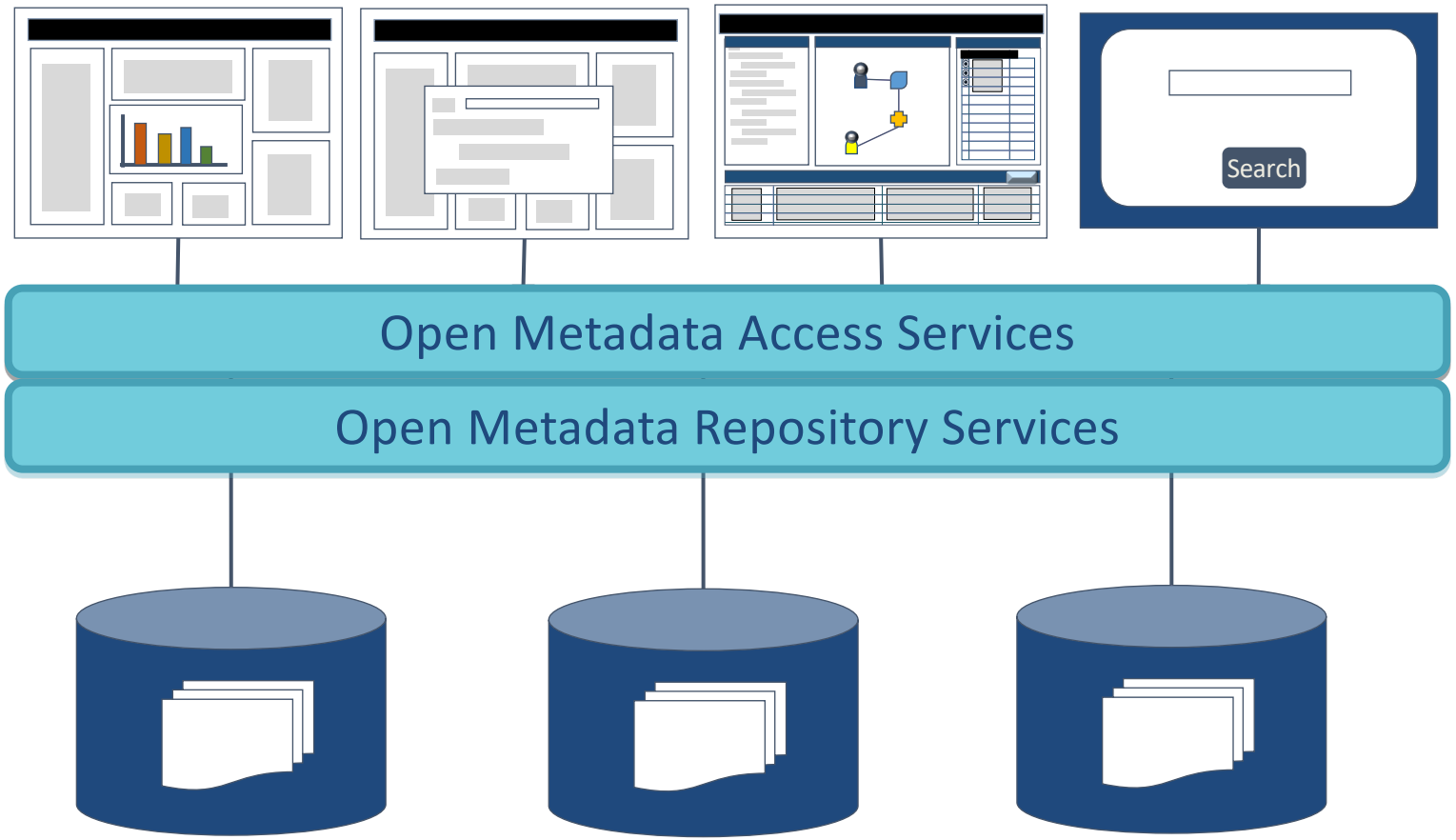
Functional organisation



OMAS = Open Metadata Access Service

Subject Area OMAS provides services for a Subject Area owner

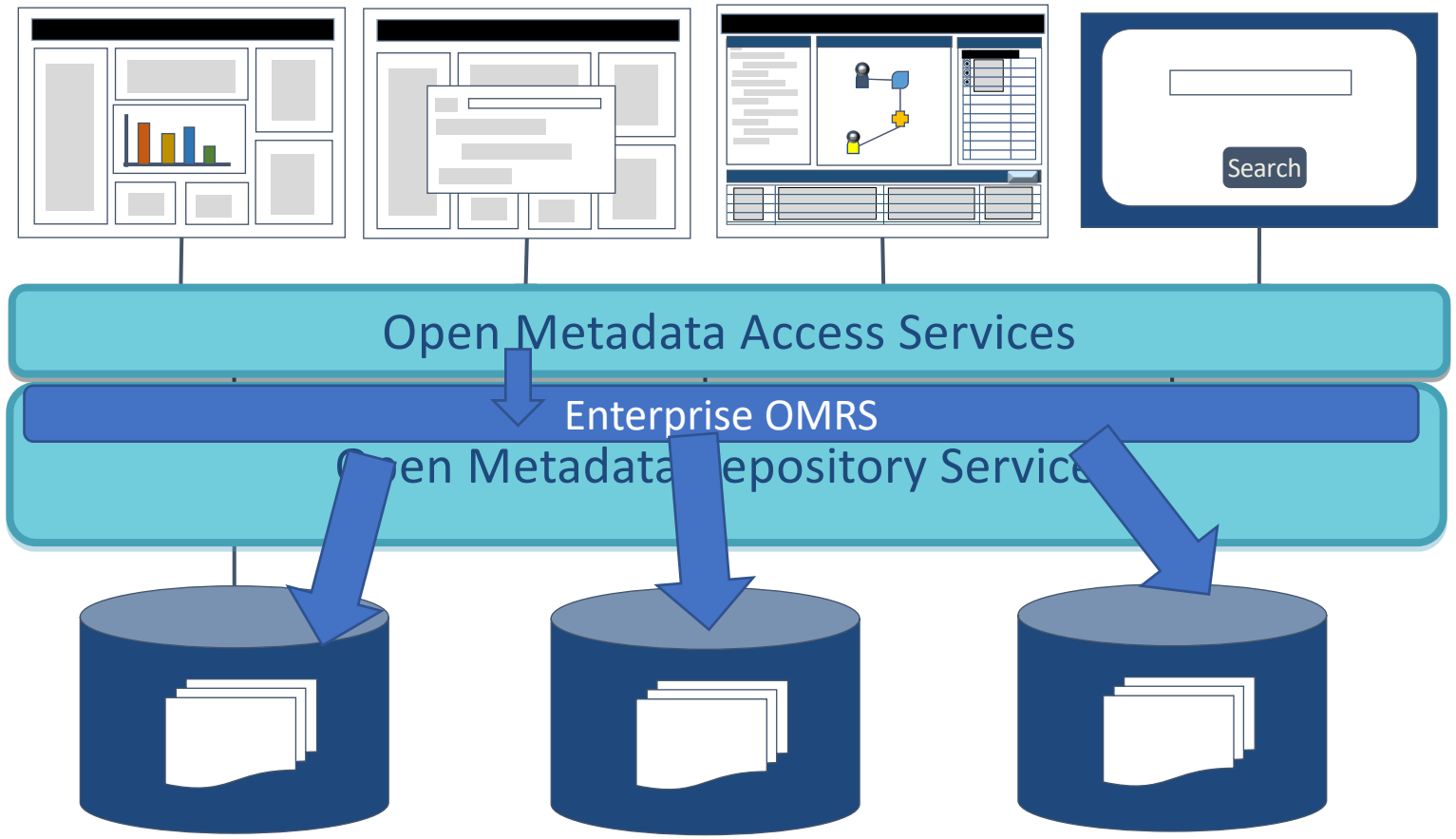
Egeria – vendor neutral integration



Use cases,
Personas,
Practitioners
input

Data
integration,
availability and
integrity best
practices

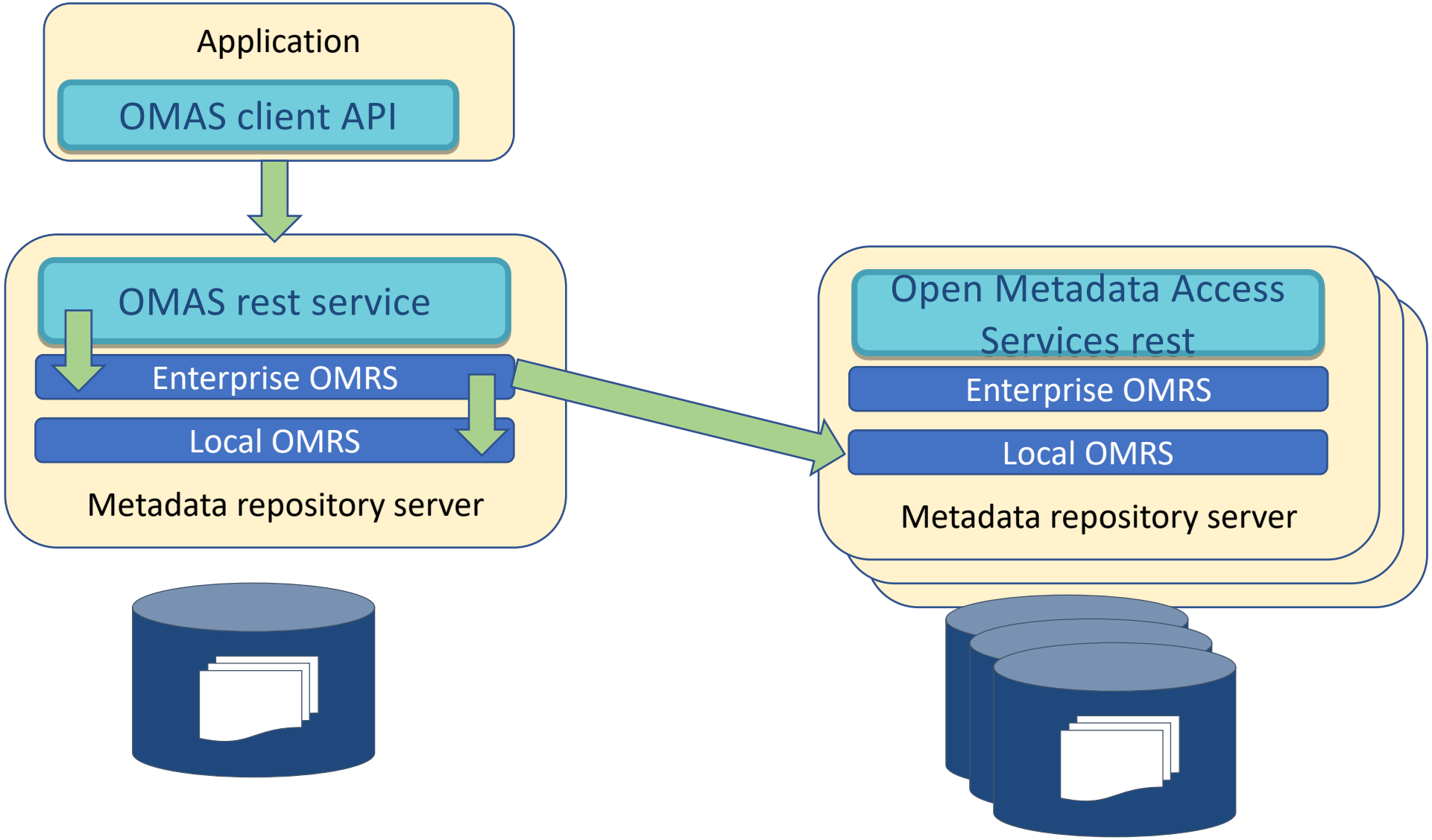
Egeria – vendor neutral integration



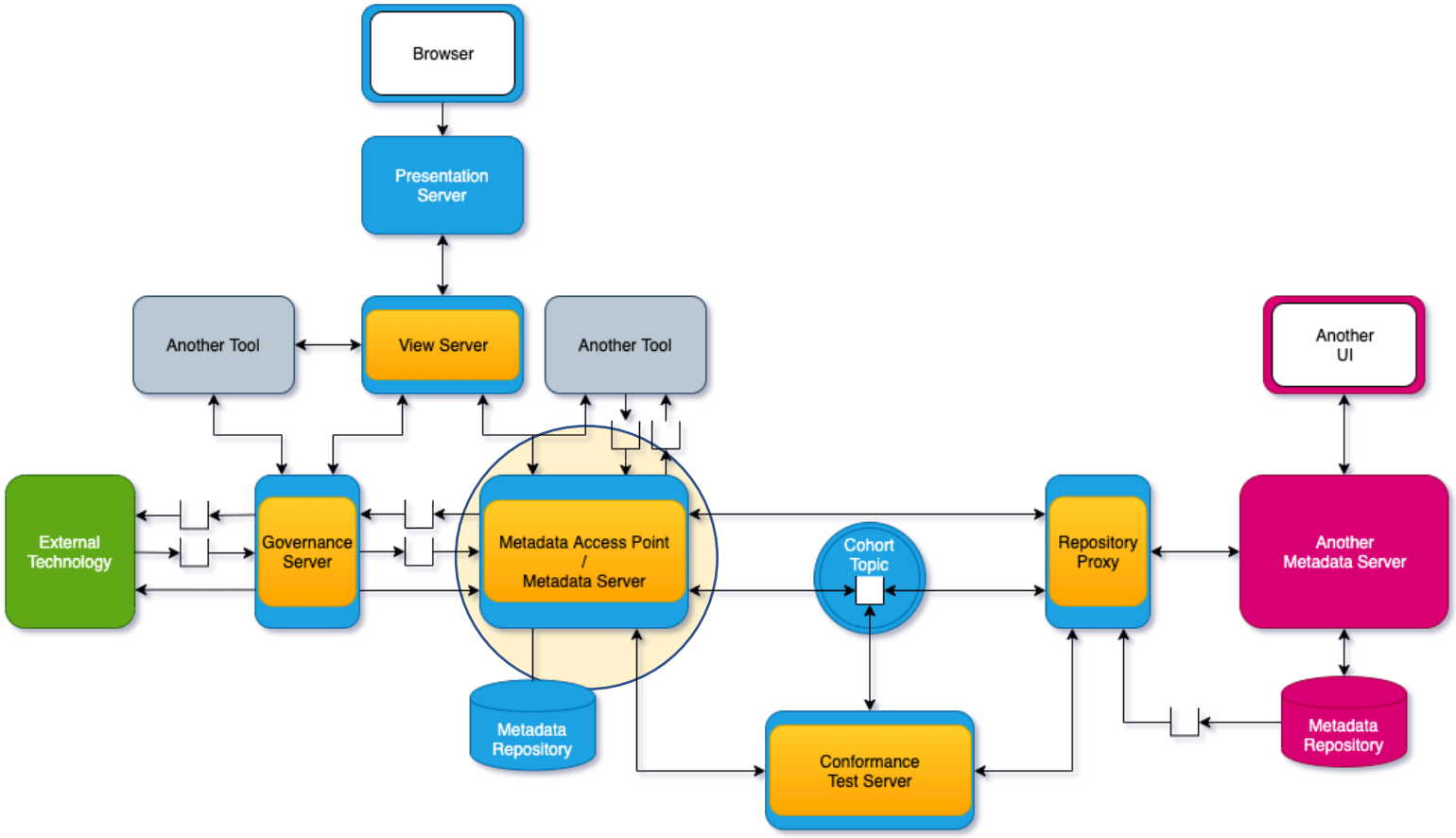
Use cases,
Personas,
Practitioners
input

Data
integration,
availability and
integrity best
practices

OMAS and OMRS

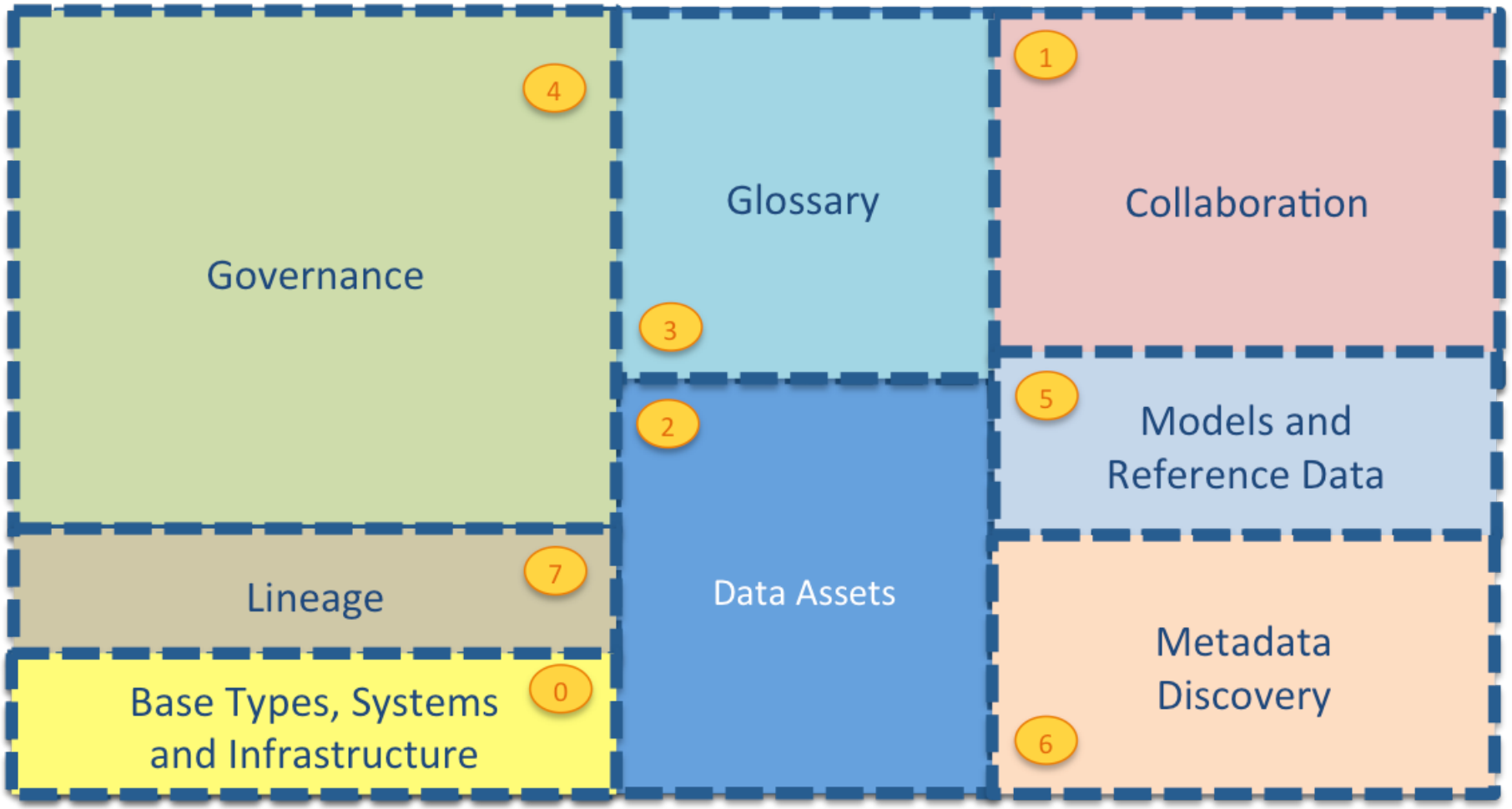


The high level architecture



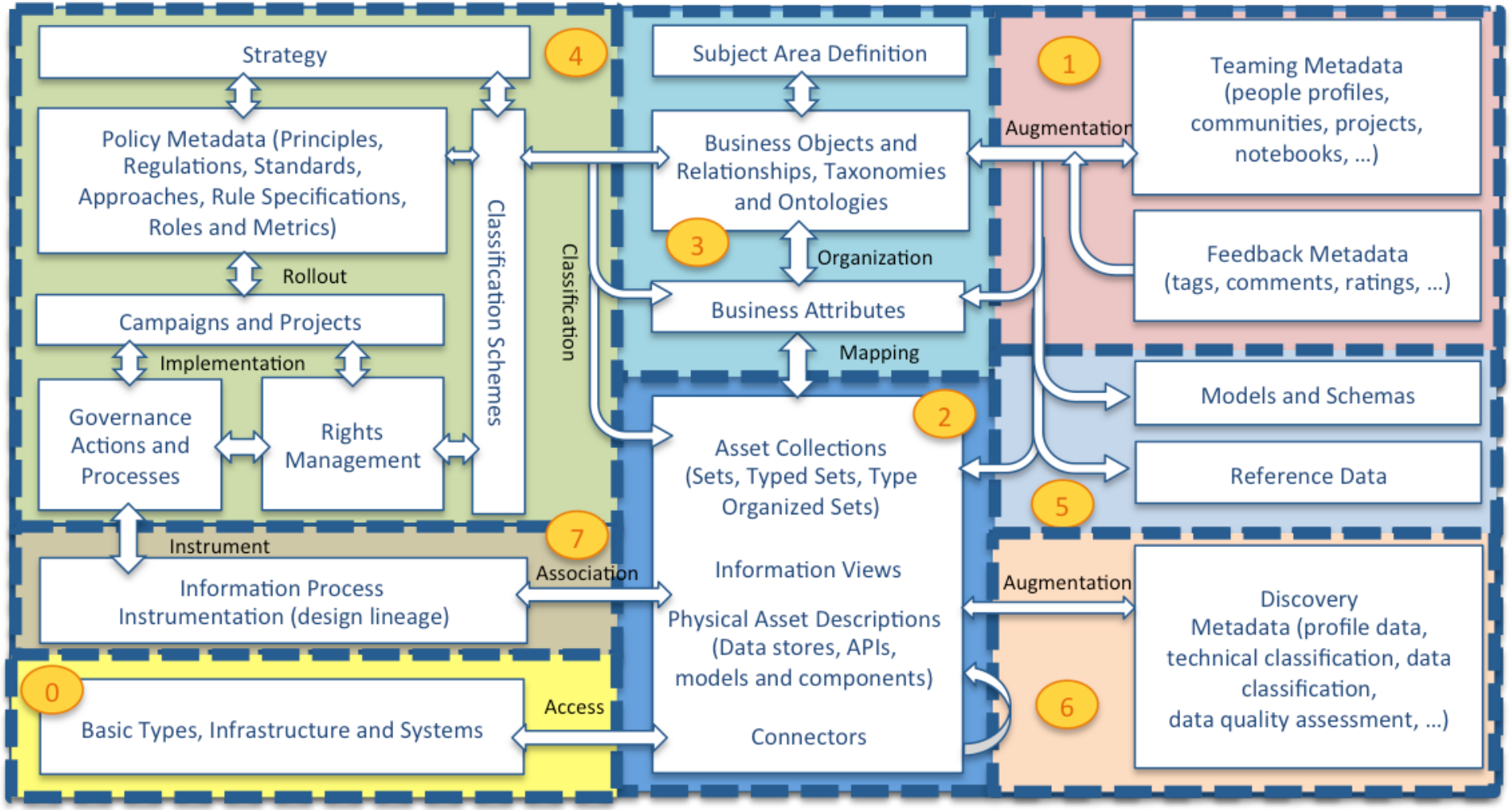
https://egeria.odpi.org/open-metadata-publication/website/planning-guide/

Open types



https://egeria.odpi.org/open-metadata-publication/website/open-metadata-types/

Open types in more detail



Subject Area
OMAS authors
Area 3 content

https://egeria.odpi.org/open-metadata-publication/website/open-metadata-types/Area-3-models.html

Open Metadata Access Service

Exposing a person or tools specific API, usually based on a subset of the open types

They should be designed outside in.

They are defined in configuration, accessed via a OMAGServerConfigStore connector.

The admin calls can be used configure an OMAS

They expose a Rest interface and optionally a java API.

They can produce events with OMAS orientated payloads (Not OMRS orientated)

Writes to the audit log

OMAS in Egeria

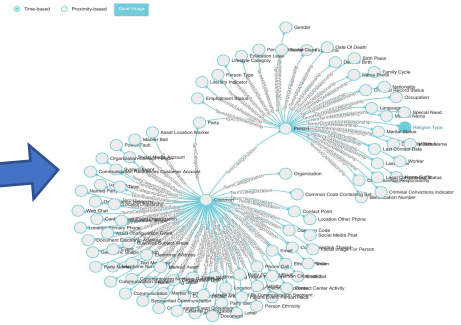
Startup of Subject Area OMAS in audit log

Thu May 13 11:21:18 BST 2021 cocoMDS1 Startup OMRS-AUDIT-0040 An enterprise OMRS connector has been created for the Subject Area OMAS
Thu May 13 11:21:18 BST 2021 cocoMDS1 Startup OMRS-AUDIT-0041 The enterprise OMRS connector for the Subject Area OMAS has started

Subject Area OMAS Config example


```
"class": "AccessServiceConfig",
"accessServiceId": 1020,
"accessServiceAdminClass": "org.odpi.openmetadata.accessservices.subjectarea.admin.SubjectAreaAdmin",
"accessServiceName": "Subject Area",
"accessServiceFullName": "Subject Area OMAS",
"accessServiceURLMarker": "subject-area",
"accessServiceDescription": "Document knowledge about a subject area",
"accessServiceWiki": "https://egeria.odpi.org/open-metadata-implementation/access-services/subject-area/",
"accessServiceOperationalStatus": "ENABLED",
"accessServiceInTopic": {
```

Subject Area - domain of expertise within an organisation



efficacy
drugs
patient
ethical
Code of practice
contagious
Audit trail
Adverse effect

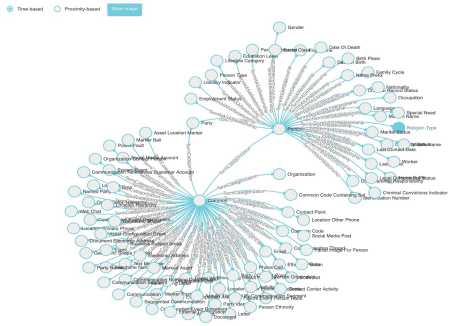
Tribal knowledge

Pharmaceuticals expert

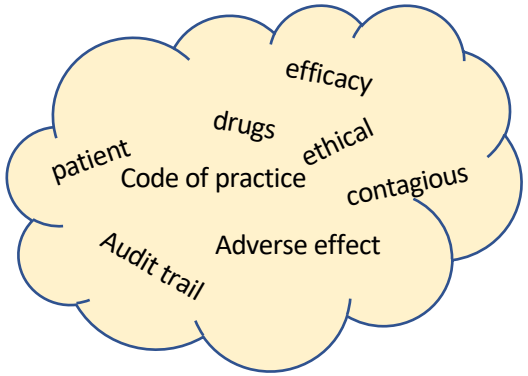
Invoice
Tax
Audit trail
client
budget
customer
sales

Finance expert

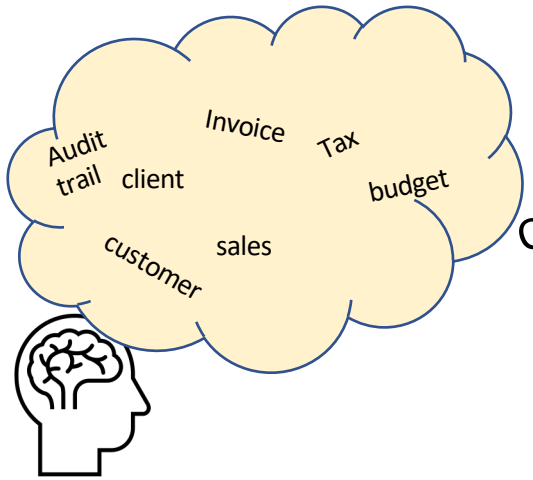
Capture the language and relationships



Mapping into Egeria allows you to store and connect using the open types



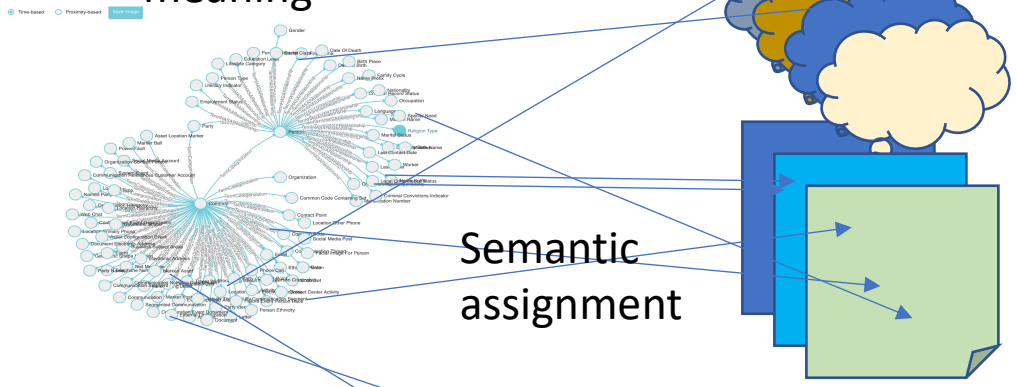
Pharmaceuticals expert



Finance expert

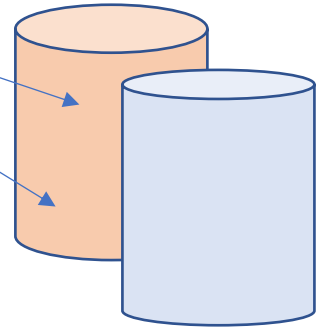
Capture terms & relationships

Context and meaning



Technical assets

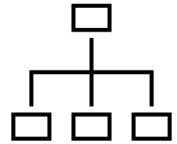
Semantic assignment



Egeria open types representing technical Assets and meaning allowing you to connect Across different metadata repositories

Subject Area use cases

Shared understanding 

Creation of category hierarchy 

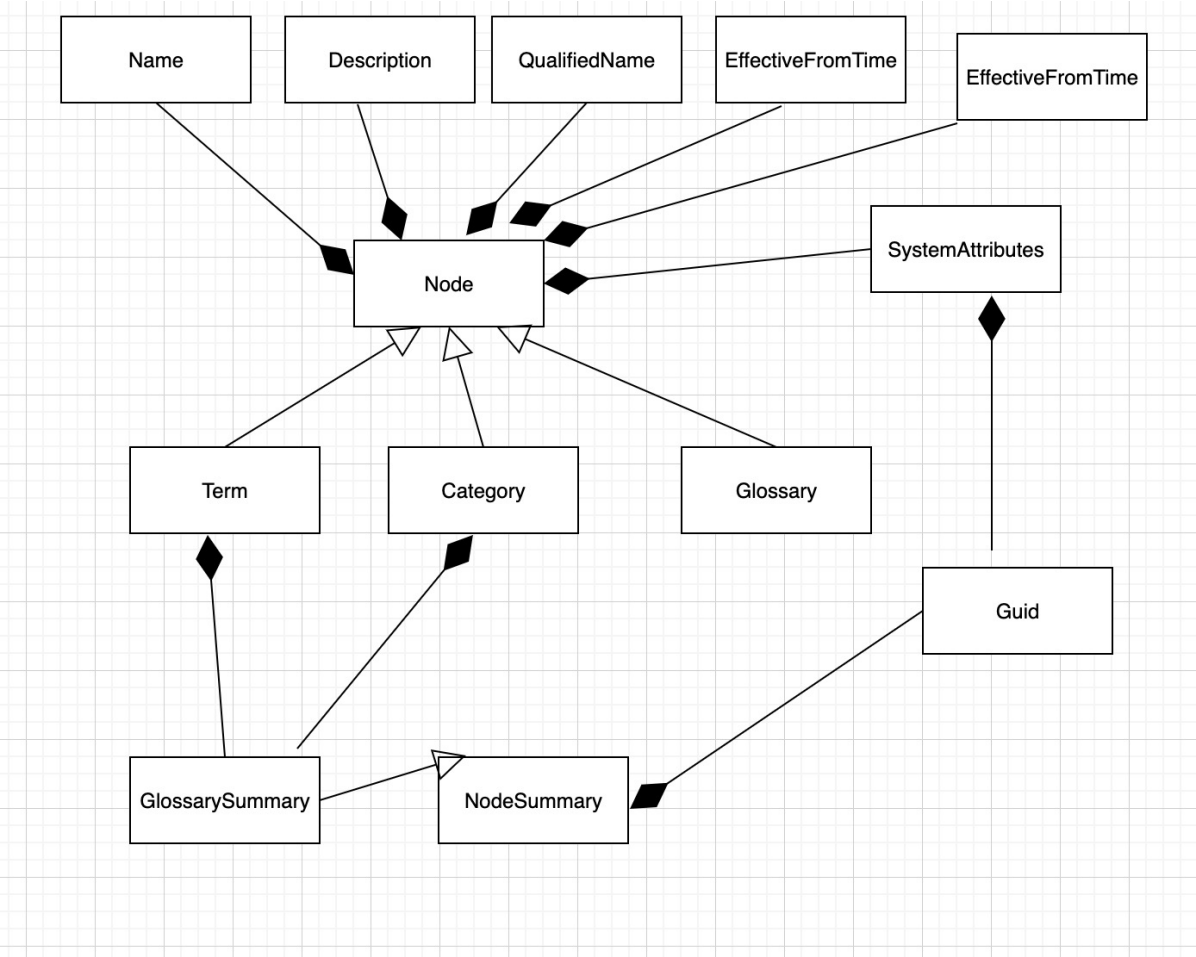
Relating terms to increase meaning 

Governance – confidentiality … 

Architectural positioning.

- Decision on scope of the subject area omas
  - Glossary only for scoping and capturing subject matter expertise
  - Or also get involve validation rule definition and valid values (sets and terms)
- Relationship with digital architecture and governance program
  - Governance program creates subject area definition
  - Digital architecture for implementation
- 2 ways of working
  - Subject area –subject area classification already set by governance program
  - Glossary author – basic operations.

Subject Area model repacks Area 3 content (and a little more) so it can be worked with naturally by Subject Area authors



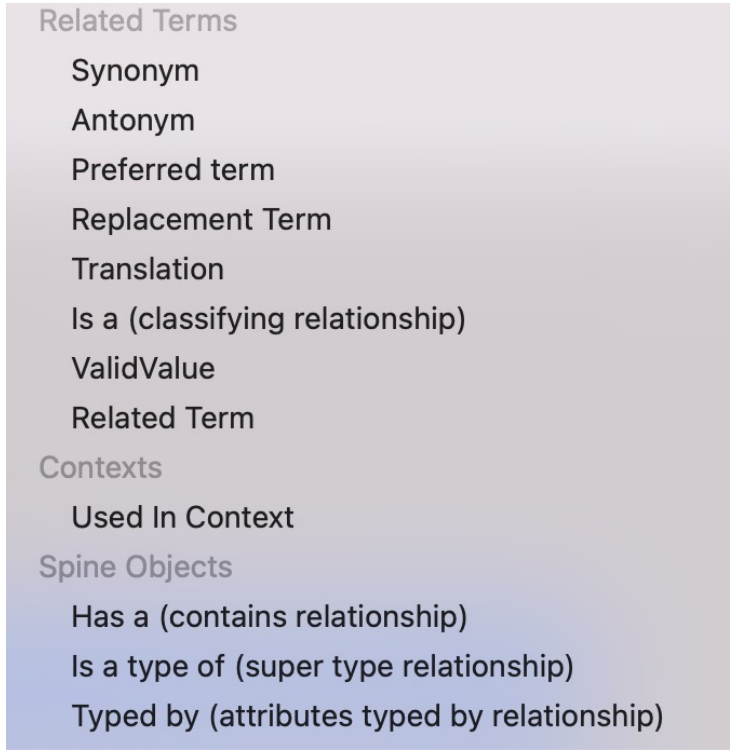
Part of the subject area model

- Node contains common attributes
- Term Category Glossary extend Node
- Terms and Categories are owned by one Glossary
- Glossary -> term, Category ->Category and Category - > Term relationships are managed by the OMAS logic.

Subject Area OMAS authoring

- Authoring semantic content
- Nodes
  - Glossaries
  - Categories
  - Terms
- Relationships
  - Relating terms
  - Hierarchy
  - Spine objects
  - ….

Types of term to term relationships



Semantic relationships linking terms together to add more meaning

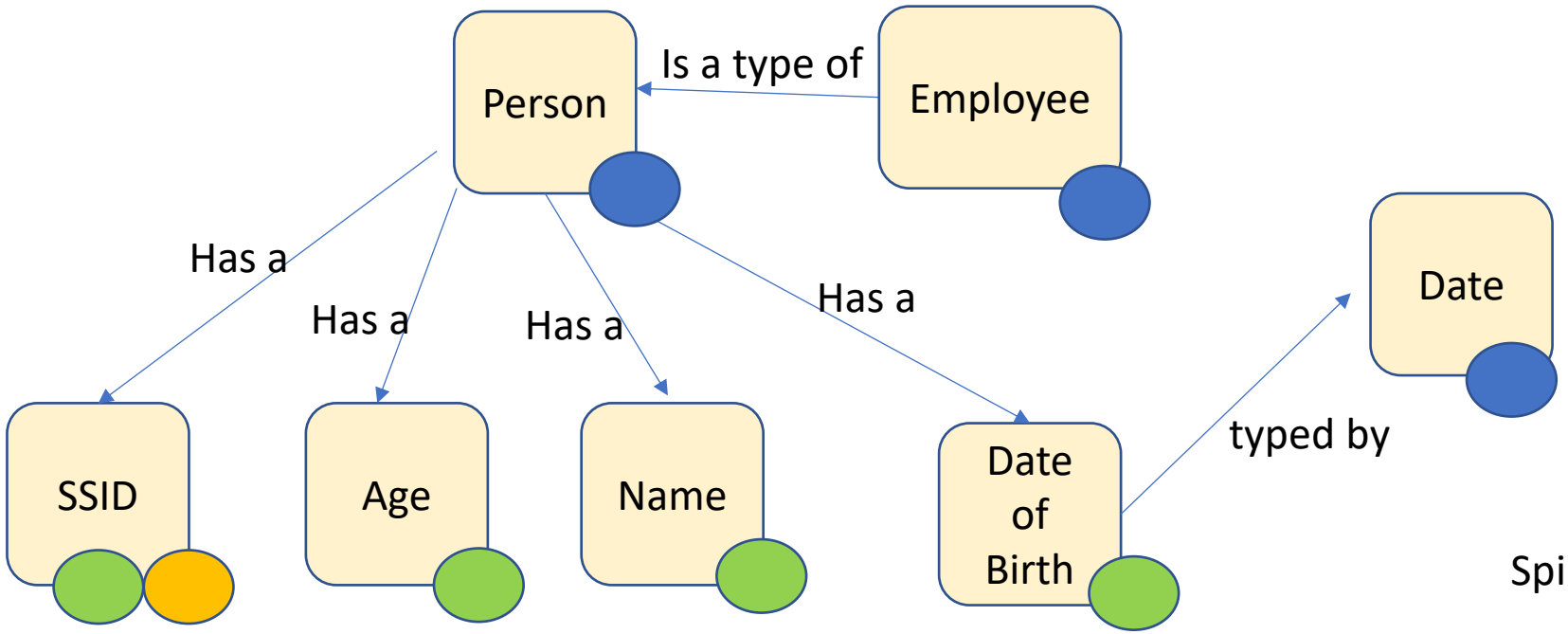
Defining and linking terms to be used in a particular context

Holding structure in the glossary

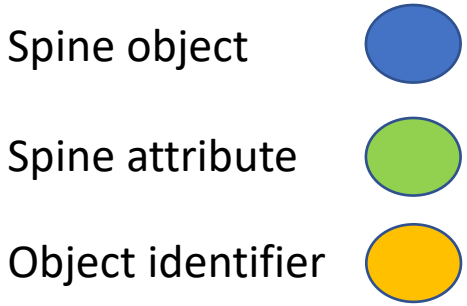
Why do we need Spine Objects?

- Technical information is often held in a structure e.g. a relational database table and it’s columns
- To add meaning to the column or table, we represent the technical artifacts in the glossary as terms.
- These technical Glossary terms can be linked to business orientated glossary terms. In this way you can relate technical meanings and business meaning.

Spine Objects and attributes



Spine objects, attributes and object identifiers are Boolean flags on Term in Subject Area OMAS



Python consumer of Subject Area OMAS – some helper functions

Can create Terms and Glossaries. Some useful functions you may not be aware of

- import to import Terms defined in a csv file:

```
createSimpleTermsFromCSVFile(serverName, serverPlatformName, serverPlatformURL,
                             userId, fileName, glossaryGuid, nameColumnHeader,
                             descriptionColumnHeader, exampleColumnHeader)
```

- valid values (reference data)

```
createValidValue(serverName, serverPlatformName, serverPlatformURL,
 userId, validValueForGuid, validValuesGuid)
```

- createSynonym

```
createSynonym(serverName, serverPlatformName, serverPlatformURL, userId, guid1, guid2)
```

- Semantic assignment (not Subject Area)

```
createSemanticAssignment(serverName, serverPlatformName, serverPlatformURL, userId,
 assetGuid, schemaTypeGuid, glossaryTermGuid
```

https://github.com/odpi/egeria/blob/master/open-metadata-resources/open-metadata-labs/common/environment-check.ipynb

Python consumer of Subject Area OMAS – some helper functions

Can create Terms and Glossaries. Some useful functions you may not be aware of

- import to import Terms defined in a csv file:

```
createSimpleTermsFromCSVFile(serverName, serve
                             userId, fileNam
                             descripti
```

- valid values (reference data)

```
createValidValue(
 userId, val                                    mURL,
```

```
                                    formName, serverPlatformURL, userId, guid1, guid2)
```

bject Area)

```
                                    ment(serverName, serverPlatformName, serverPlatformURL, userId,
                                    maTypeGuid, glossaryTermGuid
```

Incomplete – opportunity for Python programmers to contribute more functions

https://github.com/odpi/egeria/blob/master/open-metadata-resources/open-metadata-labs/common/environment-check.ipynb

Glossary Author View service consumer of Subject Area OMAS

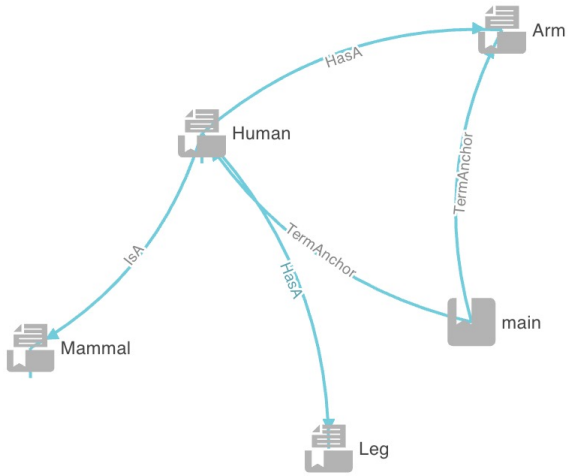
Mostly proxies the Subject Area OMAS API

Breadcrumb API to optimize breadcrumb length

Bulk term creation API.

Glossary Author UI

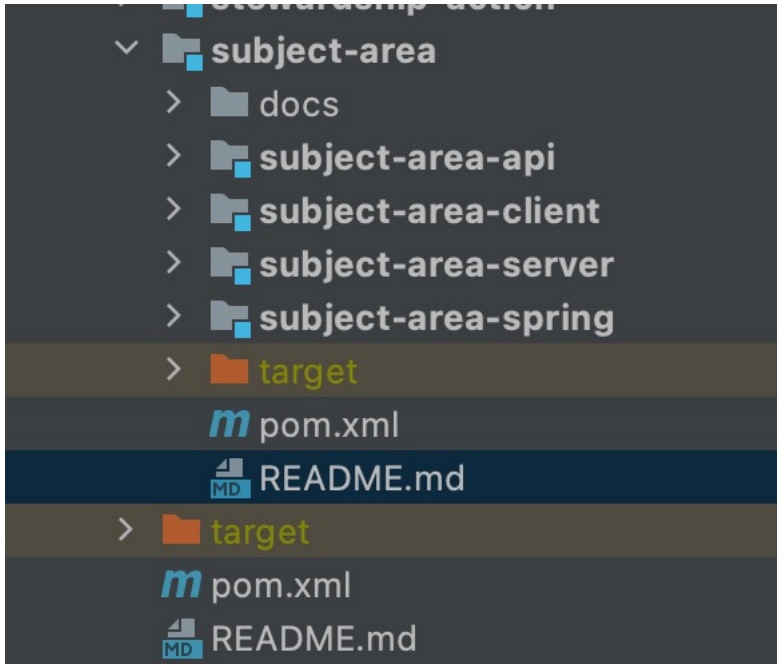
- For more details see previous community call presentation.
- * Update* Create relationship being developed.



See video

Subject Area OMAS code

Maven modules



API –Term Category Taxonomy etc beans and ffdc messages
Client – the Java OMAS API
Spring – the Spring rest interface
Server – the server code that communicates with OMRS

https://github.com/odpi/egeria/tree/master/open-metadata-implementation/access-services/subject-area

Subject Area architectural decisions

- Uses Generics
- Dates are longs
- Mappers
- Supports effectivity on CRUD
- Supports update and replace
- Supports soft and hard deletes.
- Does not yet use the generic OMRS handlers
- Does not yet support controlled terms yet.
- Graph interface to query and filter neighborhood
- Read only indicator

Subject Area OMAS FVT

A comprehensive test of the core of the Subject Area capability, in the build.
Very useful when making changes – ensuring no regressions
Testing framework to add to. Tests all of the following:

Feel free to add new tests to increase coverage

https://github.com/odpi/egeria/tree/master/open-metadata-test/open-metadata-fvt/access-services-fvt/subject-area-fvt

Importing

- Python
- Archive
  - Canonical model
  - CIM
  - …

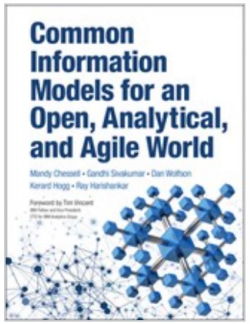
Use cases not implemented but in this scope.

- APIs to enable import from the Egeria React UI
- APIs to enable valid values management the Egeria React UI
- producing events
- rules
- subject area focused to APIs. Currently glossary focused.
- export
- ???

Information models.

https://www.informit.com/store/common-information-models-for-an-open-analytical-and-9780133366341

Home > Store



View Larger Image

Common Information Models for an Open, Analytical, and Agile World

By Mandy Chessell, Gandhi Sivakumar, Dan Wolfson, Kerard Hogg, Ray Harishankar

Published Apr 8, 2015 by IBM Press. Part of the IBM Press series.

eBook (Watermarked)
Your Price: $25.59
List Price: $31.99
Includes EPUB, MOBI, and PDF
About eBook Formats

Add to cart

Add To My Wish List

Share |

Register your product to gain access to bonus material or receive a coupon.

Description | Sample Content | Updates

Copyright 2015
Dimensions: 7" x 9-1/8"
Edition: 1st

eBook (Watermarked)
ISBN-10: 0-13-336634-0
ISBN-13: 978-0-13-336634-1

Maximize the Value of Your Information Throughout Even the Most Complex IT Project
Foreword by Tim Vincent, IBM Fellow and Vice President, CTO for IBM Analytics Group

To drive maximum value from complex IT projects, IT professionals need a deep understanding of the information their projects will use. Too often, however, IT treats information as an afterthought: the "poor stepchild" behind applications and infrastructure. *That needs to change. This book will help you change it.*