E1.2 DISEÑO DE LA ARQUITECTURA Y COMPONENTES

ESPACIO DE DATOS LINGÜÍSTICO (SER 15/23 OTT)

Resumen

Documento de diseño de la arquitectura y componentes del catálogo del espacio de datos lingüísticos. En este documento se describe a los componentes de espacio de datos de eclipse como componente fundamental sobre el cual se implementarán las extensiones particulares del espacio de datos lingüístico y el catálogo. Además, se incluye a ELG-Share como vocabulario de referencia para describir los activos del espacio de datos y ODRL para definir las políticas de uso y contratación. Teniendo en cuenta estos referentes y sus mecanismos de extensión se propone un diseño para el espacio de datos. Finalmente se presentan los prototipos creados para la interfaz de usuario del espacio de datos.

Andrés García-Silva

26 de febrero de 2024

Expert.ai Research Lab

Calle Poeta Joan Maragall, 3-5, Escalera Izquierda, Planta 1ª, Derecha, 28020, Madrid CIF: B-66425513, Inscrita en el Registro Mercantil de Madrid, en el Tomo 44.538, Folio 74, Hoja Número M-784613, Inscripción 1ª.

www.expert.ai

Historia de revisiones

revisión	Fecha	Descripción	Autor (Organización)
0.1	19/02/2024	Primera versión completa del documento	Andres Garcia Silva
0.2	21/02/2024	Revisión del documento	José Manuel Gómez
0.3	26/01/2024	Actualización diseño – reunión de empresas	Andrés García-Silva
1.0	26/01/2024	Versión 1.0	Andrés García-Silva

Contenido

Lista de Figuras

1 Introducción

Los componentes de espacios de datos de eclipse, EDC por sus siglas en inglés, implementan componentes extensibles que siguen las especificaciones de la asociación internacional de los espacios de datos (IDS). Estos componentes son flexibles y permiten que se añadan nuevas funcionalidades con solo implementar algunas de las interfaces que contiene la especificación EDC. En el espacio de datos lingüísticos EDL se va a usar la versión de INESDATA de los EDC para la implementación de los conectores y del catálogo del espacio de datos.

La principal extensión al conector EDC será la **adición de metadatos para describir los activos que se publicarán en el EDL**. Para definir estos metadatos se va a seguir el vocabulario ELG Share que se utiliza en el European Language Grid. **Los metadatos deben almacenarse de forma persistente y poder consultarse vía búsquedas tradicionales basadas en palabras clave y búsquedas por facetas**. Además, de forma análoga a la definición de políticas en EDC se seguirá el vocabulario ODRL.

El diseño del EDL se realiza teniendo en cuenta la arquitectura, el conector y el catálogo federado de EDC, el vocabulario ELG Share y los servicios en la nube de INESData. Además, en este documento se incluyen prototipos de la interfaz de usuario del catálogo del EDL y la consola del usuario donde se gestiona los activos, políticas y ofertas de contratos.

2 Eclipse Dataspace Components EDC

El proyecto Eclipse Dataspace Components EDC es una colección completa de bibliotecas y módulos, que se publican como artefactos Maven y que los desarrolladores pueden usar y ampliar. EDC implementa el estándar International Data Spaces (IDS), así como los protocolos y requisitos relevantes asociados con Gaia-X. EDC es extensible de tal manera que puede admitir protocolos alternativos. EDC separa los planos de control y de datos, lo que permite una forma modular y personalizable de construir espacios de datos. Debido a las interfaces comunes y al mapeo de estándares existentes, EDC agrega capacidades de negociación de contratos y manejo de políticas de manera interoperable.

2.1 Conector del Espacio de datos y el catálogo.

Un conector (ver Figura *1*) es un componente necesario para garantizar los principios subyacentes del espacio de datos incluyendo identidad, confianza, soberanía, e interoperabilidad. Los participantes en el espacio de datos usan el conector para asegurar la compartición de datos de acuerdo con estos principios. El conector provee capacidades para descubrir, conectar, negociar automáticamente contratos, aplicar políticas y auditar procesos. Un conector se integra en la infraestructura de cada participante y se comunica con otros.

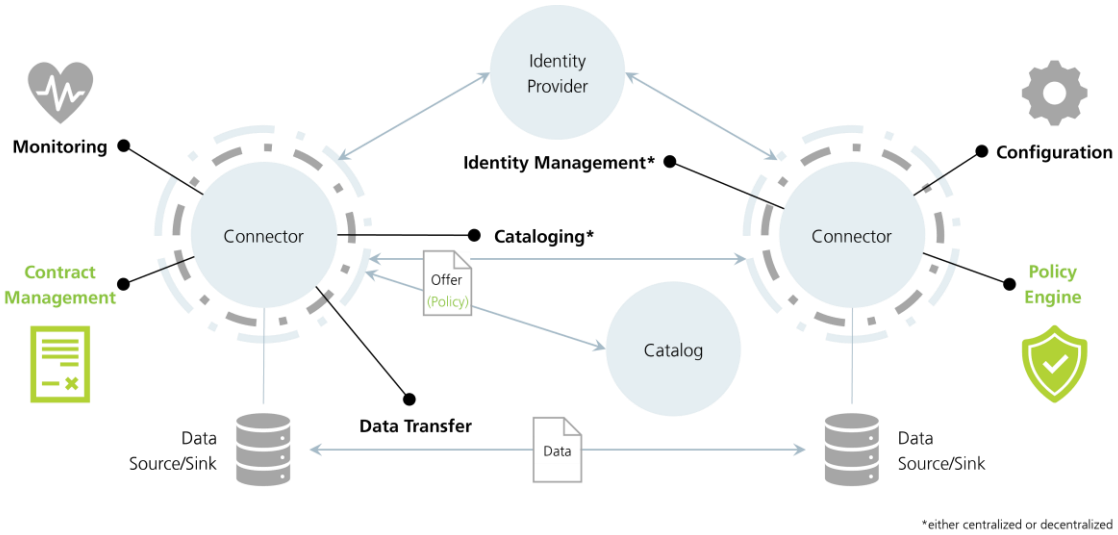

Figura 1. Características del conector del espacio de datos

Fuente: https://eclipse-edc.github.io/docs/#/README

2.2 El plano de control

El plano de control tiene entre sus tareas manejar solicitudes de protocolo y API, administrar varios procesos asincrónicos internos, validar políticas, realizar la autenticación de participantes y delegar la transferencia de datos a un plano de datos. Su trabajo es manejar (casi) toda la lógica empresarial. Para ello, está diseñado para favorecer la confiabilidad sobre la baja latencia. No transfiere datos directamente desde el origen al destino.

- **Activos:** Los activos son contenedores de metadatos, no contienen los bits ni bytes reales. El activo también contiene una dirección de datos, que puede entenderse como un "puntero al mundo físico".
- **Políticas:** Las políticas permiten expresar que ciertas condiciones pueden, deben o no cumplirse en determinadas situaciones. Las políticas se utilizan para expresar qué requisitos debe satisfacer un sujeto (por ejemplo, un interlocutor en la comunicación) para poder realizar una acción.
- **Definiciones de Contrato:** Las definiciones de contrato son donde se vinculan los activos y las políticas. Es la forma de expresar qué políticas están vigentes para un activo. Entonces, cuando se va a ofrecer un activo (o varios activos) en el espacio de datos, se una definición de contrato para expresar en qué condiciones se ofrece. Esas condiciones se componen de una política de contrato y una política de acceso. Es importante tener en cuenta que las definiciones de contrato son objetos internos, es decir, nunca salen del ámbito del proveedor y nunca se envían al consumidor.
    - **Políticas de acceso**: Determina si a un consumidor en particular se le ofrece un activo o no. Por ejemplo, es posible que se quiera restringir ciertos activos de modo que solo los consumidores dentro de una geografía particular puedan verlos. Los consumidores fuera de esa geografía ni siquiera los tendrían en su catálogo.
    - **Políticas de contrato**: Determina las condiciones para iniciar una negociación de contrato para un activo en particular. Satisface está política no garantiza automáticamente la creación exitosa de un contrato, simplemente expresa la elegibilidad para iniciar la negociación.

  Las definiciones de contrato también contienen un selector de activos. Esto es una expresión de consulta que define todos los activos que se incluyen en la definición. Con

eso es posible configurar el mismo conjunto de condiciones (= política de acceso y política de contrato) para una multitud de activos.

- **Negociaciones de contratos:** Si un conector cumple la política del contrato, puede iniciar la negociación de un contrato para un activo en particular. Durante esa negociación, ambas partes pueden enviar ofertas y contraofertas que pueden contener términos alterados (= política) como lo haría cualquier ser humano en una negociación, y la contraparte puede aceptarlas o rechazarlas. Las negociaciones contractuales tienen algunos aspectos clave:
  - apuntan a un activo
  - tienen lugar entre un proveedor y un conector de consumidor
  - el usuario no puede cambiarlos directamente
  - los usuarios sólo pueden rechazarlos o cancelarlos

  Las ofertas de contrato son objetos efímeros, ya que se generan sobre la marcha para un participante en particular y nunca persisten en una base de datos y, por lo tanto, no se pueden consultar a través de ninguna API.

  Las negociaciones contractuales son de naturaleza asincrónica. Eso significa que después de iniciarlos, se convierten en procesos con estado (potencialmente de larga duración) que avanzan mediante una máquina de estado interna. El estado actual de la negociación se puede consultar y modificar a través de la API de administración.
- **Acuerdos de contrato:** Una vez que la negociación de un contrato concluye con éxito, se "convierte" en un acuerdo contractual. El conector del proveedor es el que da la aprobación final. Los acuerdos contractuales son objetos inmutables que contienen la política final acordada, el ID del activo para el que se negoció el contrato y la fecha exacta de firma.
- **Catálogo:** El catálogo contiene las "ofertas de datos" de un conector y uno o varios puntos finales de servicio para iniciar una negociación para esas ofertas**.**
  Cada oferta de datos está representada por un objeto *Conjunto de datos* que contiene una política y uno o varios objetos de *Distribución*. Una Distribución debe entenderse como una variante o representación del Conjunto de Datos. Por ejemplo, si se puede acceder a un archivo a través de múltiples canales de transmisión desde un proveedor (HTTP y FTP), entonces cada uno de esos canales se representaría como una Distribución. Otro ejemplo serían los recursos de imagen que están disponibles en diferentes formatos de archivo (PNG, TIFF, JPEG). Un objeto *DataService* especifica el punto final donde el proveedor acepta las negociaciones y transferencias de contratos. En la práctica, este será el punto final DSP del conector.

2.3 Mecanismo de Extensión de EDC

El conector de EDC se ha diseñado para ser extendido fácilmente con nuevas funcionalidades. Para extender el conector se necesitan dos cosas:

- una clase que implementa la interfaz *ServiceExtension*.
- un archivo de complemento en el directorio src/main/resources/META-INF/services. Este archivo debe tener el mismo nombre que el nombre de clase completo de la interfaz y debe contener el nombre completo de la clase de implementación (clase de complemento).

En el conector el directorio SPI contiene todas las interfaces necesarias que deben implementarse, así como clases de modelo y enumeraciones esenciales. Básicamente, los módulos spi definen hasta qué punto los usuarios pueden personalizar y ampliar el código.

Una clase que implementa ServiceExtension debe incluir un objeto cuya clase implementa una interfaz en el directorio SPI.

3 Catálogo

Compartir datos entre participantes requiere el suministro de metadatos que los describan. La información sobre los datos debe publicarse con un vocabulario acordado para su consulta y con controles que regulen el acceso a los elementos del catálogo. Dos participantes pueden compartir datos comunicándose directamente o en línea sin la necesidad de un catálogo. Pero para un mayor número de participantes, una función de catálogo aumenta enormemente la capacidad de descubrimiento de los activos y servicios de datos. Los catálogos no proporcionan el activo de datos en sí, pero sí ofertas de contratos de datos (más sobre esto en la sección sobre intercambio de datos a continuación).

3.1 Arquitectura

Un catálogo es un componente común para implementar la capacidad de descubrimiento de datos. Puede ser implementado como un servicio administrado por uno o más participantes seleccionados, alojado por la autoridad del espacio de datos u operado de manera completamente descentralizada por cada participante que ofrece contratos de datos. El tipo de arquitectura de catálogo utilizada depende del diseño del espacio de datos, así como de las necesidades y capacidades de los participantes.

En un espacio de datos se prefiere una arquitectura federada en oposición a un catálogo centralizado. El catálogo centralizado presenta desventajas incluyendo baja disponibilidad ante fallos, así como riesgos en la neutralidad de quien administre el catálogo. Por este motivo, se propone un catálogo federado hibrido como el que se muestra en la Figura 2.

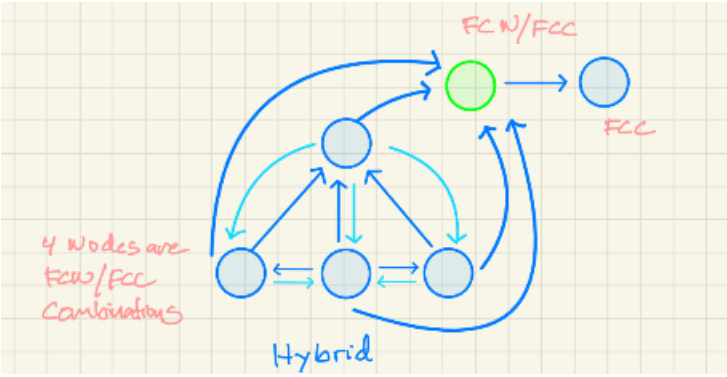

Figura 2. Catálogo Federado Hibrido.

(Fuente: https://github.com/eclipse-edc/FederatedCatalog/blob/main/core/docs/Catalog_Architecture.pdf)

En está arquitectura hay conectores que funcionan como Nodos del Catálogo Federado (FCN) y otros conectores asumen el rol de Cache del Catálogo Federado (FCC). Además, un conector puede actuar como ambos FCN y FCC.

A continuación, se listan las funcionalidades soportadas por cada subsistemas:

Nodo del catálogo Federado (FCN)	Cache del Catálogo Federado (FCC)
• Catálogo de cada participante • Soporta consultas basadas en políticas • Puede soportar múltiples protocolos • Se apoya en el índice de activos (AssetIndex) • Soporta distintos tipos de almacenamiento	• Sirve de cache de los nodos de catálogo federado y sus políticas • Ofrece una interfaz de consulta • Una entrada del catálogo se envía al conector de origen para su recuperación • Puede soportar múltiples protocolos • Soporta distintos tipos de almacenamiento

La arquitectura de la cache del catálogo federado FCC se muestra en la Figura 3. Hay dos componentes claramente separados: el subsistema de interfaz de consulta y el subsistema de crawling.

- El subsistema de crawling se encarga de consultar todos los participantes del espacio de datos en el servicio de registro y extrae la información de sus catálogos (FCN) para posteriormente almacenarlas en el almacenamiento del FCC.
- El subsistema de consulta expone un API rest para que los participantes del espacio de datos puedan consultar los datos disponibles en el espacio de datos e iniciar transacciones.

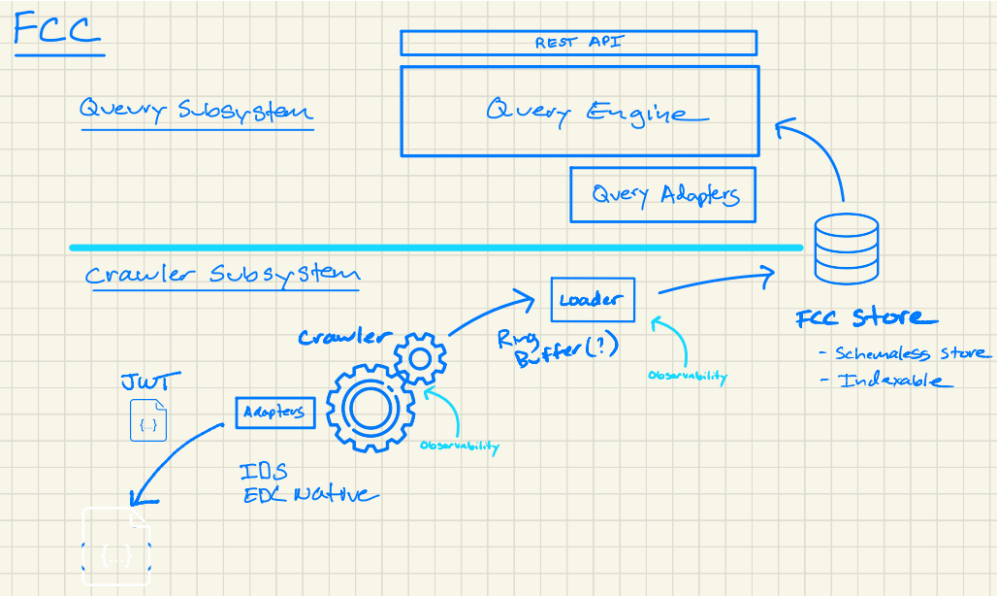

Figura 3. Componente de la cache del catálogo federado.

(Fuente: https://github.com/eclipse-edc/FederatedCatalog/blob/main/core/docs/Catalog_Architecture.pdf)

El nodo del catálogo federado es mucho más simple ya que solo almacena la información de un participante del espacio de datos. Para esto se apoya en índice de activos del conector que registra los activos del participante y un almacenamiento externo para almacenar políticas y ofertas de contrato.

4 Vocabularios

En cuanto los metadatos y vocabulario usados en el espacio de datos se van a usar los vocabularios propuestas en el EDC para la definición de políticas. Es decir, la definición de políticas se hará usando el vocabulario ODRL[1].

[1] https://www.w3.org/TR/odrl-model/

Para describir los activos del espacio de datos con metadatos estructurados se propone usar el esquema ELG-Share[2], también llamado ELG-Schema, que se usa en el European Language Grid ELG. ELG-Schema está especialmente creado para describir tecnologías del lenguaje incluyendo corpora, herramientas y servicios, modelos, y otros recursos lingüísticos.

Los principales conceptos de este vocabulario se visualizan en la Figura 4:

- **MetadataRecord**: Corresponde al ítem del catálogo y registra información relativa al proceso de registro, como quién creó el ítem y cuándo, si fue extraído de otro catálogo, quién es responsable de su curación (actualizaciones), etc.
- **DescribedEntity**: Corresponde a cualquier entidad que pueda ser descrita por un registro de metadatos. Puede ser un recurso lingüístico, una persona, una organización, etc. (consulte Tipos de elementos del catálogo y el cuadro verde en la imagen de arriba).
    - La clase **LanguageResource** se distingue además en uno de cuatro tipos de recursos: **ToolService**, **Corpus**, **LexicalConceptualResource** y **LanguageDescription**.
    - Un recurso de lenguaje se puede describir a través de un conjunto de elementos de metadatos comunes a todos los tipos, y un conjunto adicional que se ajusta a cada uno de estos cuatro tipos.
- **Distribución**: Corresponde a la forma física con la que se pone a disposición un Recurso Lingüístico a través del catálogo, p.e. como un archivo descargable, o un formulario al que se accede a través de una interfaz, etc.

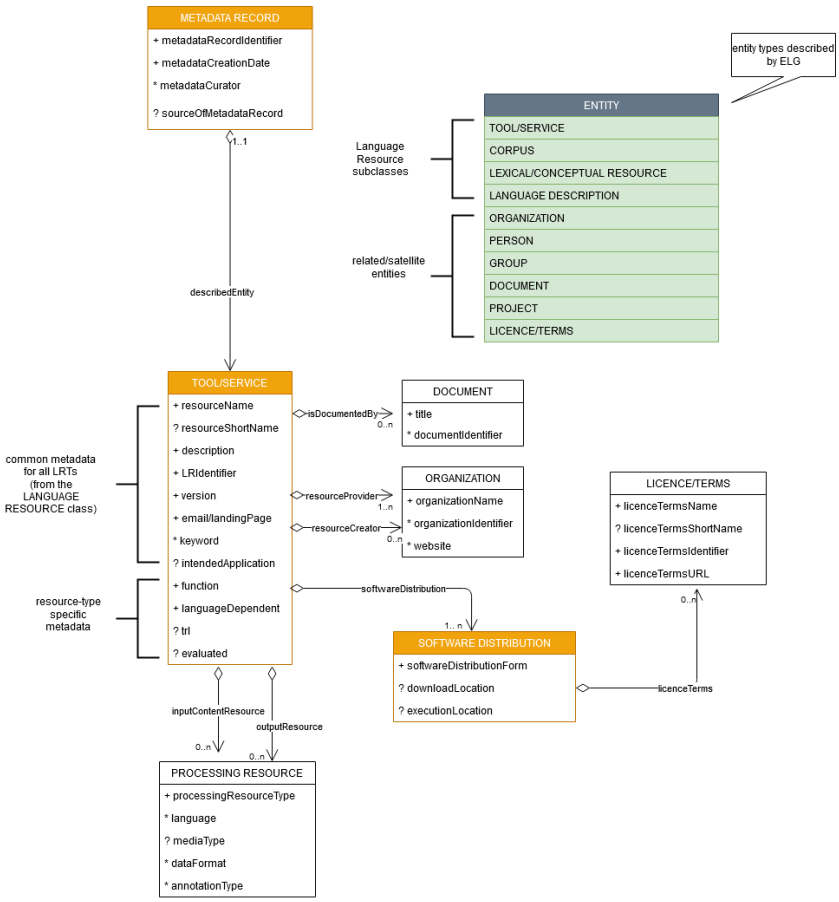

Figura 4. Principales conceptos de ELG-Share.

(Fuente: https://european-language-grid.readthedocs.io/en/stable/all/A2_Metadata/Metadata.html)

[2] https://european-language-grid.readthedocs.io/en/stable/all/A2_Metadata/Metadata.html

El esquema XSD completo, la documentación, así como plantillas y ejemplos de registros de metadatos para todos los tipos de recursos se pueden encontrar en el repositorio Git del esquema ELG SHARE: https://gitlab.com/european-language-grid/platform/ELG-SHARE-schema

La documentación completa del esquema está disponible en la siguiente web: https://european-language-grid.readthedocs.io/en/stable/all/A2_Metadata/Metadata.html

5 Diseño y arquitectura

En el espacio de datos lingüísticos se usará como base la **arquitectura definida por EDC** presentada en la sección 2 garantiza la compatibilidad con los estándares y normas definidos por IDSA (International Dataspace Association). Además, dado que EDC establece un mecanismo para extender los componentes (ver sección 2.3), es posible adaptar ciertos componentes a las necesidades particulares de este espacio de datos.

La principal extensión prevista es **incluir los nuevos metadatos para la descripción de los activos**. **Se espera que la versión INESData del conector persista los metadatos**, por ejemplo, en una base de datos PostgreSQL[3]. Además, **es necesario que la versión INESData del conector permita realizar búsquedas sobe los metadatos incluyendo búsquedas por palabras clave y búsquedas facetadas por los tipos de metadatos**.

En cuanto a los metadatos que se usarán para describir los activos, se seguirá el vocabulario ELG Share (ver sección4). En particular se va a usar la **versión mínima**[4] de este vocabulario que incluye solo los metadatos obligatorios y recomendamos. Estos metadatos han sido cuidadosamente seleccionados por varias razones:

- **identificación y cita**: nombre(s) del recurso; identificador(es); una breve descripción del contenido; información de versiones; un punto de contacto para obtener más información (correo electrónico o página de destino); datos de los proveedores de recursos y de los creadores de recursos; clasificación por dominio, palabras clave y aplicación prevista; cobertura lingüística (idioma y, si es necesario, dialecto); fecha de publicación
- **soporte**: enlaces a manuales, material de capacitación; muestras del recurso
- **uso/acceso**: Forma de distribución (por ejemplo, como archivo descargable, una distribución que se puede acceder a través de una interfaz, código fuente o archivo binario de software, etc.); condiciones de licencia; ubicación de acceso.

Se requieren elementos de metadatos adicionales, específicos de cada tipo de recurso, como el tamaño y formato de los archivos de datos, dependencias y requisitos técnicos para herramientas y servicios, etc.

El **catálogo del espacio de datos será federado** de acuerdo con la descripción presentada en la sección 3. La implementación del catálogo se hará tomando como base la implementación disponible en EDC[5]. Está implementación **será extendida para que toda la información del catálogo se almacene en una base de datos PostgreSQL**. Ya que la implementación del catálogo federado sigue la arquitectura EDC se pueden usar los mecanismos de extensión para añadir nuevas funcionalidades.

El portal que podrán usar los usuarios para acceder al espacio de datos (ver sección 6) se conectará al conector que corresponde a cada usuario para la gestión de activos, contratos, políticas, y negociaciones. En la Figura 5 se presenta el diagrama de secuencia que muestra la interacción entre usuario, portal, servicios en la nube de INESData, el conector del espacio de datos y el cache del catálogo federado.

Cuando el usuario inicia sesión en el portal, se identifica contra el servicio de autenticación de INESData, y este retorna un token de acceso. El portal solicita a la nube de INESData que ejecute la imagen del conector correspondiente al usuario. Para evitar largos periodos de espera en la ejecución de imágenes, la nube de INESData debe mantener un conjunto de imágenes en ejecución por ejemplo teniendo en cuenta la frecuencia de uso del conector por el usuario o las consultas que reciba de otros conectores.

[3] https://www.postgresql.org/

[4] https://european-language-grid.readthedocs.io/en/stable/all/A2_Metadata/MinimalVersion.html

[5] https://github.com/eclipse-edc/FederatedCatalog

Sí la ejecución de la imagen del conector en la nube INESData fue exitosa entonces el portal se conectará con el portal del conector y usará los servicios del conector para la gestión de activos, políticas, y ofertas de contratos. De forma similar, si el usuario quiere consultar el catálogo del espacio de datos, está consulta se realiza por medio del portal del conector correspondiente que tendrá acceso a la caché del catálogo federado usando el conector. Así, la cache del catálogo federado solo visualizará la información a la que tenga disponibilidad el conector del usuario.

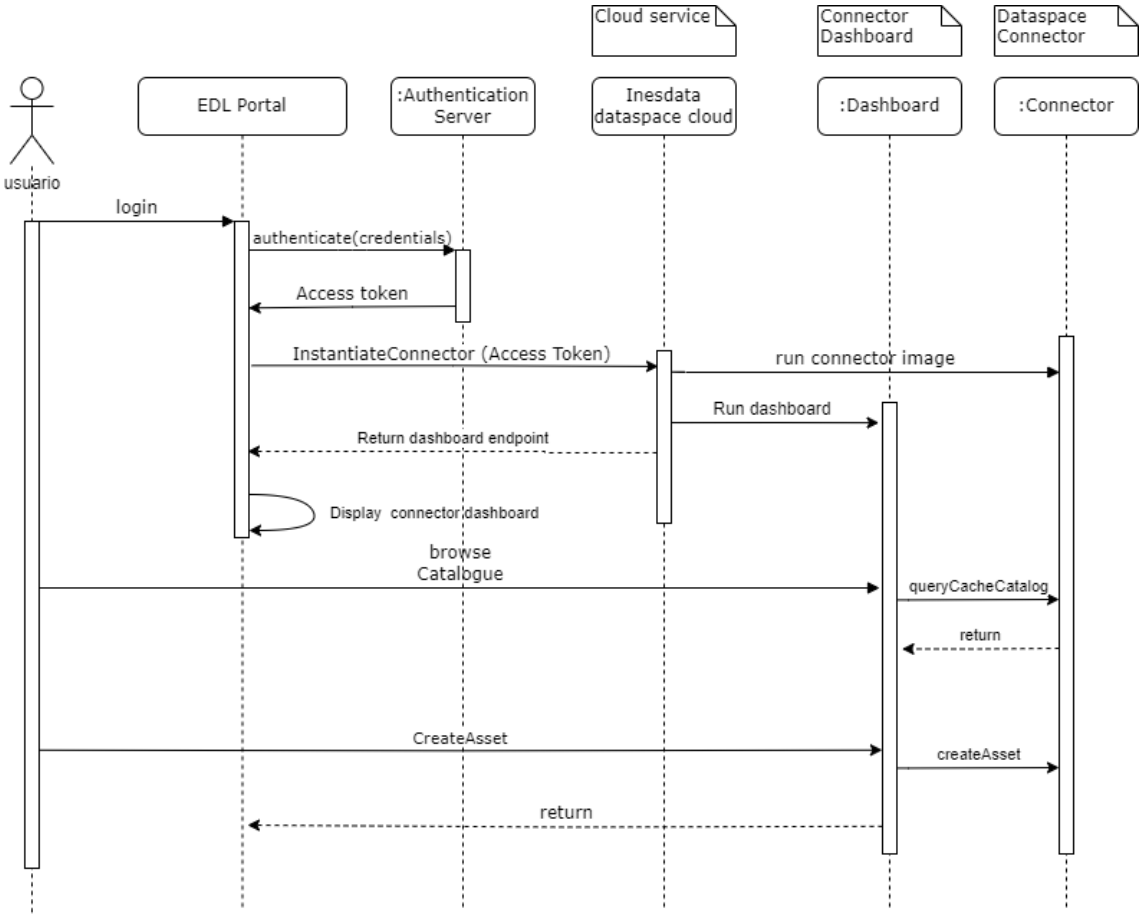

Figura 5. Diagrama de secuencia del espacio de datos.

6 Interfaz de Usuario

En la Figura 6 se presenta un mapa cognitivo de alto nivel de las acciones habilitadas en la interfaz del catálogo. El usuario alcanza la página de destino donde tiene acceso al catálogo. En la página del catálogo el usuario puede navegar los activos disponibles incluyendo datos y servicios. Además, el usuario puede filtrar los recursos disponibles, por ejemplo, de acuerdo con el tipo de activo o lenguajes soportados, y una vez se ha encontrado el activo de interés esté se puede seleccionar para ver los detalles o ejecutar una acción. Entre las acciones consideradas se tienen negociar un activo, descargarlo si no hay restricción, o probar un servicio. Por otra parte, en la página del catálogo el usuario puede cargar, actualizar y borrar activos.

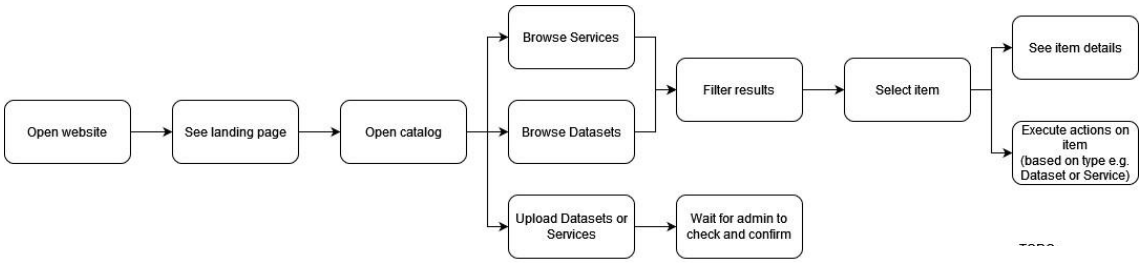

Figura 6. Mapa Cognitivo del Catálogo del Espacio de Datos

Para soportar las acciones descritas anteriormente la interfaz de usuario del catálogo se ha diseñado con dos vistas principales: a) la vista de navegación y búsqueda permite a los usuarios navegar por el catálogo aplicando diferentes filtros en la tipología de recursos y lenguajes soportados, y b) la vista de la consola del usuario donde pueden cargar activos en el catálogo, actualizarlos y borrarlos. La consola del usuario solo está disponible para usuarios registrados con sesión iniciada.

6.1 Vista de navegación y búsqueda del catálogo

Un prototipo de la vista de navegación y búsqueda del catálogo se presenta en la Figura 7. En la Figura 8 se ve el prototipo donde se han aplicado dos filtros para restringir los activos visualizados y en la Figura 9 se muestra el prototipo de la visualización detallada de un activo.

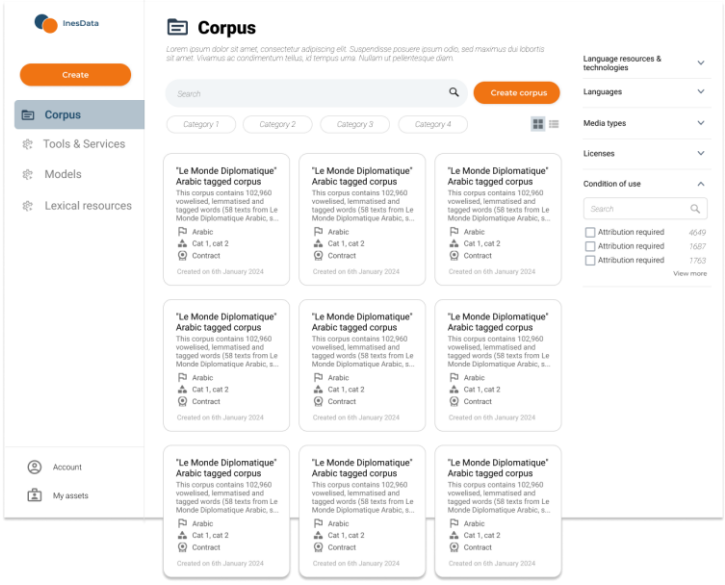

Figura 7. Vista de navegación y búsqueda

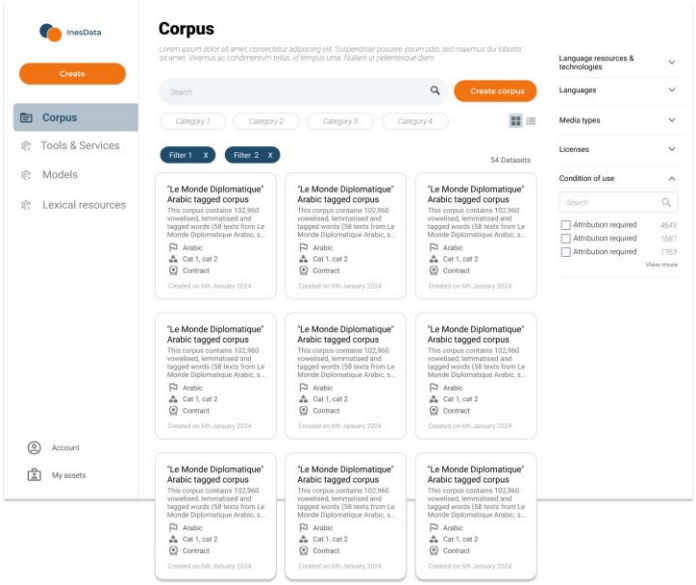

Figura 8. Filtros e la vista de navegación y búsqueda

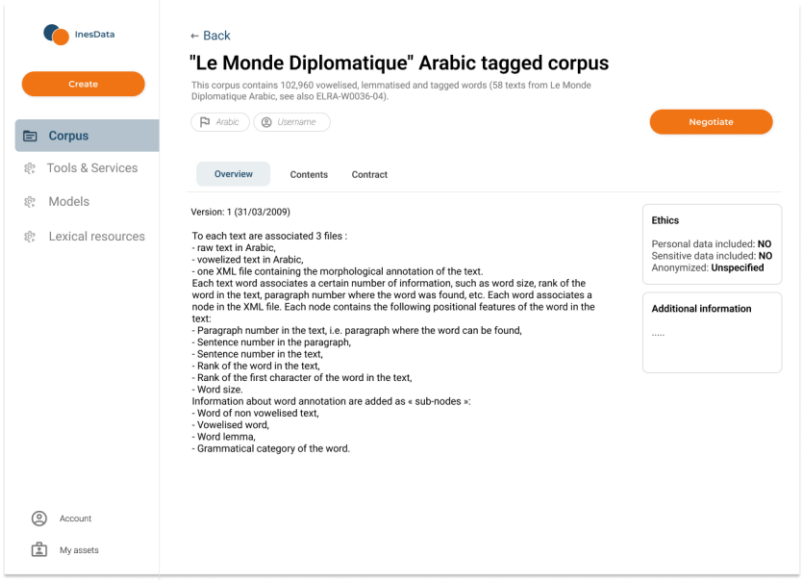

Figura 9. Detalle de un activo en el catálogo.

6.2 Consola del usuario

Un prototipo de la vista de la consola del usuario se presenta en la Figura 10. En está vista lo primero que se visualiza es un listado de los activos del usuario y estadísticas por tipo de activo. El menú incluye opciones para la creación de activos y ofertas de contrato, así como la visualización de los contratos negociados con otros usuarios del espacio de datos.

En la consola el usuario puede crear activos (ver Figura 11 y Figura 12) y asignarles una oferta de contrato previamente creada (ver Figura 13) y crear ofertas de contrato asignando políticas de acceso y de contratación previamente creadas (ver Figura 14).

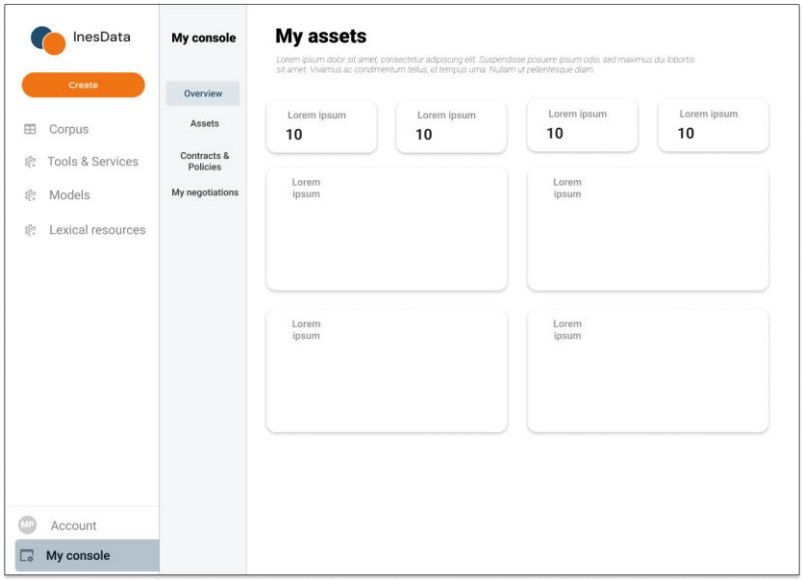

Figura 10. Vista de los activos del usuario en la consola de usuario.

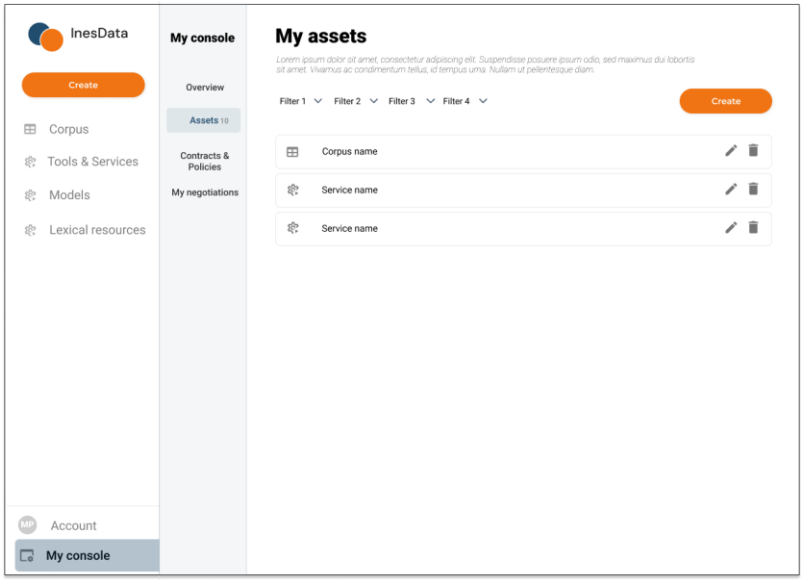

Figura 11. Interfaz de creación de activos por tipo en la consola del usuario.

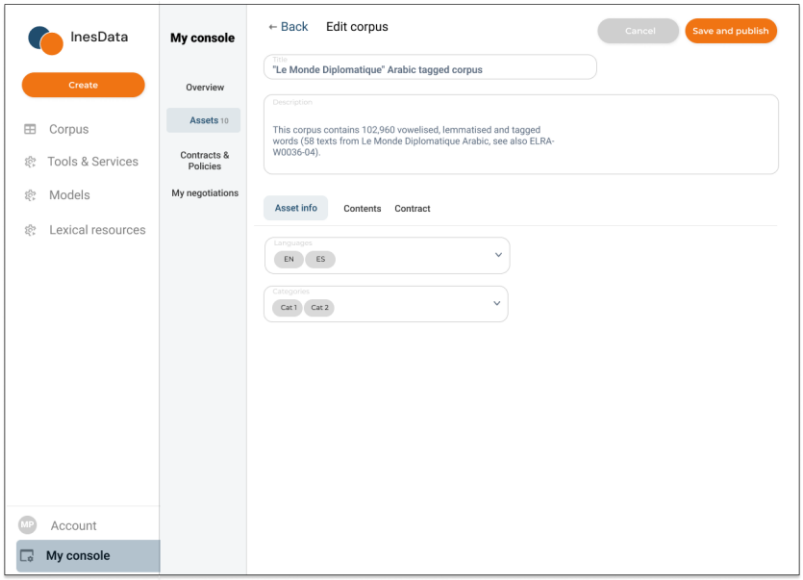

Figura 12. Ejemplo de creación de un activo de tipo corpus en la consola del usuario.

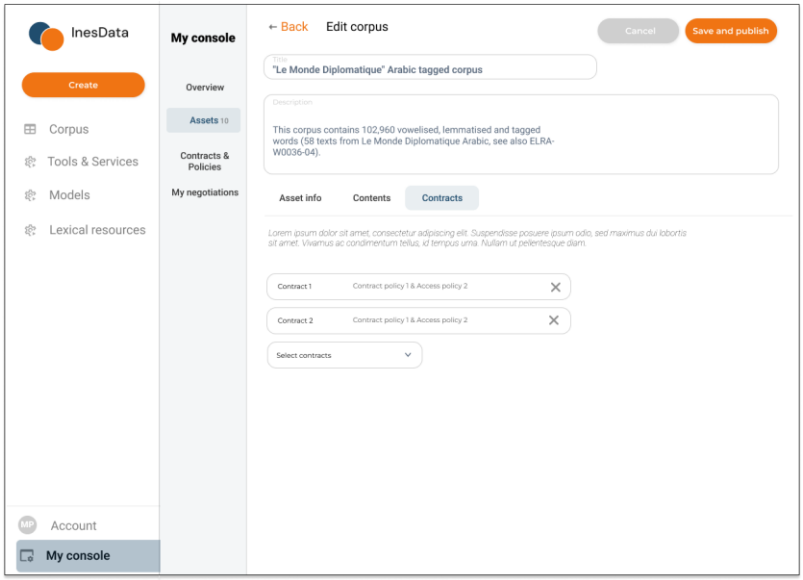

Figura 13. Ejemplo de asignación de un contrato de oferta al activo.

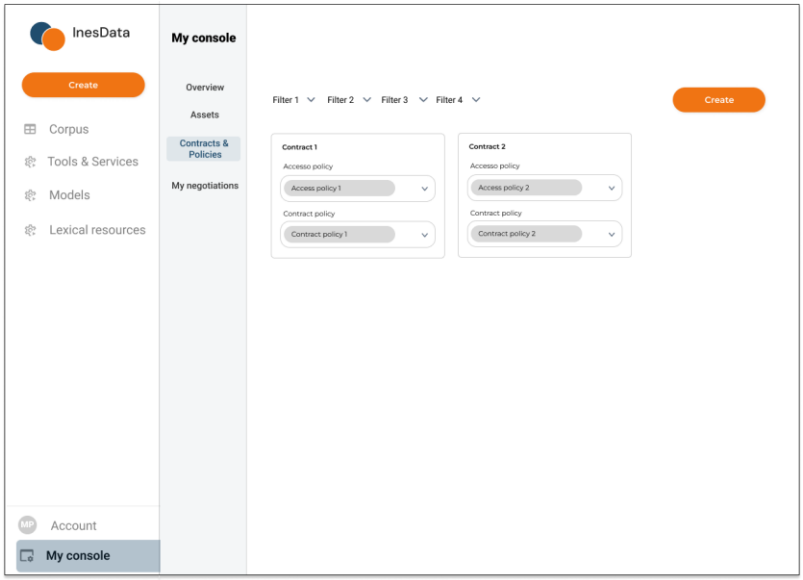

Figura 14. Creación de ofertas de contratos en la consola del usuario

6.3 Web del Espacio de Datos lingüístico

Además, se publicará una web con información del espacio de datos y como participar en el. De acuerdo con la planificación del proyecto está web se debe liberar en el mes 8 del proyecto. En este documento solo se presenta el diseño gráfico de la página web del espacio de datos. El contenido se definirá en el transcurso del proyecto teniendo en cuenta los avances en el cronograma y la fecha de liberación de la web.

Figura 15. Página principal del catálogo del espacio de datos.

7 Conclusiones

En este documento se presenta el diseño del espacio de datos lingüístico EDL. El diseño se basa en la versión INESData de los componentes de espacios de datos de eclipse EDC. Es una precondición que el conector de la versión INESData de EDC almacene en una base datos los activos, políticas, contratos y transacciones, así como los metadatos asociados a cada uno de estos ítems. Es clave que los metadatos se puedan almacenar y recuperar de forma eficiente tanto para la visualización en la interfaz de usuario como en el motor de búsqueda que soporta búsquedas por palabras claves y búsquedas facetadas por tipo de metadato.

La interfaz de usuario se compone de un portal donde el usuario podrá registrarse e iniciar sesión. Cuando el usuario acceda al catálogo se desplegará el panel (dashboard) de su conector que le dará acceso a todas las funcionalidades de búsqueda del catálogo y de gestión activos, políticas, contratos del conector.