Eusko Jaurlaritzaren
Informatika Elkartea

Sociedad Informática
del Gobierno Vasco

EJIE-2026-105

Solución de virtualización de datos

Pliego de Condiciones Técnicas

Julio 2026

Mediterráneo, 14 • 01010 Vitoria-Gasteiz • Posta-kutxatila / Apartado: 809 • 01080 Vitoria-Gasteiz • Tel. 945 01 73 00* • Fax. 945 01 73 01 • www.ejie.eus

Este documento es propiedad de Eusko Jaurlaritzen Informatika Elkartea – Sociedad Informática del Gobierno Vasco, S.A. (EJIE) y su contenido es CONFIDENCIAL. Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

Versión	Fecha	Resumen de cambios	Elaborado por:	Aprobado por:
1.0	03/06/2026	Versión inicial	Leire Rojo Aitor Cabero	Iratxe Montoya
1.1	18/06/2026	Modificaciones en todos los apartados	Leire Rojo Aitor Cabero	Iratxe Montoya
1.2	21/07/2026	Apartado 4.1 donde se cambia licencias por suscripción y se incorpora un resumen del alcance Apartado 5 donde se incorpora REDIS	Leire Rojo Aitor Cabero	Iratxe Montoya

Índice

1 Introducción 4

2 Situación Actual 5

3 Objeto del contrato 6

3.1 Objeto y alcance del producto y servicio 6

4 Requisitos de la solución 8

4.1 Conectividad – Fuentes de datos 8
4.2 Modelado de datos 10
4.3 Entrega de datos 12
4.4 Catálogo de datos 13
4.5 Seguridad /Gobernanza de datos 14
4.6 Optimización y rendimiento 16
4.7 Caché/ Replicación Selectiva 19
4.8 Semántica y Gobernanza de Datos 20
4.9 Procedimiento por lotes 21
4.10 Desarrollo y Operaciones (DevOps) 21
4.11 Despliegues 23
4.12 Requisitos de seguridad, cumplimiento y certificaciones 23
4.13 Licenciamiento y onboarding 24

1 Introducción

EJIE, Eusko Jaurlaritzaren Informatika Elkartea – Sociedad Informática del Gobierno Vasco, es la Empresa pública de servicios de las tecnologías de la información y las comunicaciones (TIC), cuya razón de existir es contribuir a la consecución de un Sector Público Vasco, moderno y eficiente, en el Marco Legal establecido por el Gobierno, con la seguridad y calidad necesarias y con el debido respeto al medio ambiente.

EJIE tiene como meta final la consecución de la satisfacción de sus clientes, siendo el instrumento común de prestación de servicios TIC en el Sector Público Vasco, y comprometiéndose en:

- Construir y mantener con eficiencia y calidad la infraestructura de los Sistemas de Información, posibilitando su continuidad y seguridad.
- Garantizar la interoperabilidad entre las distintas administraciones.
- Servir de apoyo a las necesidades de planificación y realización de la función informática de los Departamentos y Organismos Autónomos del Gobierno, asegurando la cobertura de sus demandas con el compromiso y profesionalidad adecuados a las relaciones contractuales que se establezcan.

Por tanto, EJIE debe ser un instrumento común de referencia para la prestación de servicios TIC en el Sector Público Vasco:

- Aportando valor añadido.
- Proporcionando soluciones competitivas.
- Transmitiendo confianza a sus clientes.
- Contando con personas cualificadas y comprometidas.

Se puede obtener información más detallada y extensa en nuestra dirección de Internet http://www.ejie.eus

2 Situación Actual

En la actualidad existen distintas iniciativas en el ámbito del dato estructurado y no estructurado, correspondientes tanto a EJGV como a otros organismos / entidades convergentes donde las arquitecturas centralizadas no son suficientes para responder a las cambiantes necesidades de negocio, las regulaciones de privacidad o la vertiginosa evolución de la tecnología, entre otras.

La clave para dar respuesta a todas las necesidades de información de las organizaciones hoy en día está en las arquitecturas distribuidas y lógicas. La virtualización de datos es la base de esas arquitecturas:

- Distribuidas: hoy en día, los volúmenes de datos son demasiado grandes y éstos están demasiado distribuidos. No es factible (ni deseable) tener todos los datos en un solo sistema. La realidad es que los datos residirán en múltiples sistemas/ubicaciones. Además, las necesidades analíticas y operacionales modernas son demasiado diversas, por lo que no es realista pensar en un solo sistema ("one size never fits all").

- Lógicas: los consumidores de datos se benefician al acceder a los datos a través de modelos semánticos, desvinculados de la ubicación real de los datos y los esquemas de datos físicos. A través de este enfoque semántico, los modelos de datos se adaptan a las necesidades de los consumidores y no al revés. Además, a través de un modelo semántico es posible aplicar políticas comunes sobre fuentes de datos dispares, que de otro modo no es posible a un coste y esfuerzo razonables, si se tienen que aplicar en todas y cada una de las fuentes. Finalmente, este desacoplamiento proporciona independencia y autonomía a los propietarios de la información, que pueden evolucionar la tecnología y realizar cambios en su infraestructura sin problemas, por ejemplo, mínima afección en un proceso de migración o iniciar una transición a la nube.

Así mismo, hoy en día EJIE cuenta con una plataforma de Gobierno del Dato y un Marketplace donde tanto EJIE, como sus clientes pueden publicar productos de datos. Estos productos de datos se ponen a disposición de diversas maneras, tales como, API REST, descarga directa o acceso a base de datos. La virtualización del dato permite proporcionar estos productos de datos independientemente de dónde resida el dato a compartir y la tecnología que sea necesaria para acceder al mismo.

Por ello, los beneficios de la virtualización del dato se pueden resumir en:

- Acceso unificado a datos.
- Menor latencia de acceso al dato para proyectos analíticos
- Reducción de costos de almacenamiento al no requerir replicarlos
- Mayor agilidad para el negocio
- Gobierno, semántica y seguridad centralizados
- Mejora de la calidad y consistencia de los datos
- Soporte para arquitecturas híbridas y multicloud.

El objeto de contratación del presente pliego se encuadra dentro de la prestación de servicios a contratar mediante encargo, por parte de las entidades integrantes del sector público de la Comunidad Autónoma de Euskadi, a través de sociedades que tengan la condición de medio propio de acuerdo con la legislación vigente. EJIE, como sociedad pública, órgano gestor tecnológico y medio propio, tiene encomendada la gestión centralizada de contratación, de conformidad con el Decreto 36/2020, de 10 de marzo, por el que se regula el Modelo de Gestión de las Tecnologías de la Información y la Comunicación en el Sector Público de la Comunidad Autónoma de Euskadi.

3 Objeto del contrato

3.1 Objeto y alcance del producto y servicio

El objeto del contrato es dotarse de una solución, en concreto de una plataforma SaaS, de virtualización de datos que permita integrar y acceder de forma unificada, segura y eficiente a la información procedente de diferentes sistemas y fuentes de datos, sin necesidad de replicarla físicamente, facilitando así su explotación para fines operativos, de compartición y analíticos.

La solución deberá proporcionar una capa única de acceso a los datos corporativos, mejorando la interoperabilidad entre aplicaciones, optimizando los procesos de integración de información, reduciendo la complejidad tecnológica asociada a la gestión de datos y favoreciendo una visión transversal de la información.

Asimismo, se persigue fortalecer las capacidades de gobierno del dato, garantizar la calidad, trazabilidad y seguridad de la información, y apoyar las iniciativas de transformación digital, transparencia y mejora continua de los servicios públicos prestados.

Implementar una solución de virtualización de datos que permita el acceso unificado, seguro y en tiempo real a la información corporativa distribuida en múltiples sistemas, mejorando la interoperabilidad, el gobierno del dato y la capacidad de análisis y toma de decisiones.

La solución deberá cumplir con los principios y obligaciones establecidos en materia de protección de datos personales y seguridad de la información, asegurando en todo momento la confidencialidad, integridad, disponibilidad y resiliencia de los datos tratados.

La plataforma de virtualización de datos deberá proporcionar las siguientes características:

1. **Abstracción:** las aplicaciones de negocio podrán acceder a los datos a través de un único punto lógico de acceso. Dicho punto esconderá la complejidad de las diferentes tecnologías y modelos de información subyacentes.

2. **Capa semántica unificada**: será posible definir las entidades de negocio en esta capa representando una definición común, y exponerlas como servicios de datos que cualquier aplicación puede consumir evitando inconsistencias y mejorando la calidad general de la información.

3. **Federación de datos**: la plataforma deberá permitir combinar datos de diferentes orígenes, federando el acceso a ellos a través de la capa de virtualización de datos, sin copiarlos en un repositorio centralizado. Evitar de esta manera replicar datos de un origen a otro.

4. Llevar el **procesamiento a los datos**: la plataforma deberá aprovechar al máximo los orígenes de datos especializados, explotando sus capacidades de procesamiento (por ejemplo, bases de datos analíticas que pueden ejecutar agregaciones sobre grandes volúmenes de datos de manera eficiente, motores de lagos de datos que pueden ejecutar consultas sobre colecciones de ficheros almacenados en la nube, etc.). La capa de virtualización actuará como un nodo maestro que distribuye la ejecución de la consulta entre los diferentes orígenes.

5. **Gestión de cambios**: los cambios en las fuentes de datos o en las tecnologías subyacentes se podrán acomodar fácilmente en esta capa, minimizando el impacto sobre las aplicaciones consumidoras. Las entidades, entes, organismos o departamentos tendrán total independencia para evolucionar sus sistemas con nuevos componentes tecnológicos.

6. **Capa de seguridad y gobernanza unificada**: la plataforma deberá poder definir, administrar y hacer cumplir reglas de seguridad y gobernanza de datos desde un solo punto, y definirlas en términos del modelo canónico de entidades de negocio comunes.

7. **Facilidad de uso**: el proceso de conectarse a fuentes de datos, combinarlas para crear entidades de negocio y publicarlas a través de múltiples protocolos, se debe poder hacer de modo gráfico.

8. **Flexibilidad**: la plataforma debe ser flexible para dar cabida a nuevos modelos y adaptarse de modo ágil a las necesidades de negocio.

9. **Rendimiento**: la plataforma deberá proporcionar rendimiento óptimo mediante la combinación de técnicas de optimización avanzadas, posibilidad de uso de caché y aceleración de consultas. La plataforma deberá tener un motor de optimización basado en costes para seleccionar automáticamente, en tiempo de ejecución, la mejor estrategia entre todas las posibles, sin depender de configuración ad-hoc.

Para la solución de virtualización se requieren licencias por suscripción y necesidad de onboarding como parte de la instalación inicial, tal y como se solicita en el apartado 4.13 Licenciamiento y onboarding.

En resumen, el alcance del contrato comprenderá:

- La configuración e integración de la solución con los sistemas corporativos de EJIE.
- Los servicios de formación mediante eLearning del producto a personal de EJIE los usuarios designados por EJIE.
- Los servicios de soporte técnico, mantenimiento correctivo, evolutivo y adaptativo durante la vigencia del contrato.
- La actualización de la solución para garantizar su adecuación a cambios normativos o tecnológicos.

4 Requisitos de la solución

Los requisitos descritos en este apartado deben estar disponibles en la solución en el momento de la presentación de la oferta.

4.1 Conectividad – Fuentes de datos

La solución deberá ofrecer en cuanto a conexiones e integraciones se refiere. En el listado se identifican las principales tecnologias, plataformas o servicios, sin perjuicio de que cabe la posibilidad de que se amplien:

1. Acceso a todo tipo de fuentes de datos, incluidas bases de datos relacionales (Oracle, PostgreSQL, y platafetc.), bases de datos paralelas y distribuidas, bases de datos multidimensionales (SAP BW, etc.), almacenes de datos en la nube (Amazon Redshift, S3, etc.), aplicaciones empaquetadas (SAP, etc.), SaaS (Salesforce, etc.), servicios web (SOAP, REST, OData), hojas de cálculo Excel, XML, JSON, ficheros planos, ficheros de log, colas JMS, fuentes en streaming (Kafka, etc.), bases de datos NoSQL (MongoDB, Amazon DynamoDB, etc.) y componentes de almacenamiento en Hadoop como HDFS, Parquet, Avro, etc.

2. Integración con formatos abiertos y nuevas fuentes de datos. Compatibilidad con formatos de tablas abiertas, como Apache Iceberg, Delta Lake, etc.

3. Integración nativa utilizando controladores y APIs proporcionados por los fabricantes.

4. Integración nativa y de altas prestaciones con fuentes cloud, incluyendo, pero no limitado a Redshift, Synapse, BigQuery, etc. Soporte como fuente y como caché.

5. Extensibilidad. Para aquellos casos en los que no haya un conector de fábrica, la herramienta debe proporcionar un SDK para el desarrollo de conectores personalizados.

6. Uso de APIs nativas de las fuentes de datos para obtener e integrar descripciones de columnas y otros metadatos, así como para la obtención de estadísticas e índices que puedan ser utilizados para mejorar el rendimiento. Además, la plataforma deberá utilizar APIs nativas de carga (bulk data load APIs) de las fuentes de datos (tanto en lectura como en escritura en el caso de la caché) para proporcionar alto rendimiento.

7. Conectividad con:

    a. bases de datos relacionales, incluyendo Oracle, SQLServer, MySQL, PostgreSQL, SAS, Vertica, Oracle Exadata, Sybase IQ, tanto “on premise” como en la nube. Bases de datos en memoria, incluyendo HANA, Oracle in-memory y Redis.

    b. Lagos de datos (data lakes) y lakehouses basados en almacenamiento de objetos, open formats tipo Apache Iceberg o Delta Lake y metadatado.

    c. Data warehouses, data lakes y lakehouses en nube, incluyendo Redshift, Snowflake, AWS RDS, Databricks Delta, Azure Synapse, Google BigQuery

d. Ficheros delimitados, logs, Excel, SAS

e. Ficheros XML, JSON

f. Ficheros en almacenes de objetos distribuidos, incluyendo HDFS, AWS S3, Azure ADLS, Google Cloud Storage

g. APIs SaaS, incluyendo Salesforce SFDC, Google Analytics, ServiceNow, o Dynamics 365, entre otras.

h. APIs Web Services (SOAP, REST, OData) con formatos de representación XML/JSON

i. Aplicaciones paquetizadas, incluyendo SAP, Oracle E-Business

j. Sistemas multidimensionales, incluyendo SAP BW, Oracle Analytics Server, MS SSAS

k. Bases de datos NoSQL databases incluyendo, MongoDB, Amazon DynamoDB, Redis

l. LDAP / Active Directory

m. Fuentes Streaming/IoT, incluyendo Kafka, JMS

n. Extensibilidad: SDK para desarrollo de conectores ad-hoc

8. Soporte de lectura/escritura en las fuentes

9. Integración fluida con los principales LLMs de mercado. Compatibilidad para Mistral, OpenAI, Amazon Bedrock, Azure OpenAI, Google AI Studio y NVIDIA NIM, debe permitir a las organizaciones elegir el LLM adecuado para sus necesidades. Debe disponer de un SDK de IA optimize los procesos de generación de SQL con revisión en múltiples pasos, corrección de errores y uso eficiente de tokens.

10. Enriquecimiento de datos con GenAI en SQL. Capacidad de generación de SQL integradas para que los usuarios puedan hacer consultas en lenguaje natural ayudándose de LLMs. Asimismo, desarrollo de consultas hibridas para transformar contenido no estructurado dentro de consultas estándar. Desde la extracción de sentimientos en comentarios de clientes hasta la generación de resúmenes de documentos y la traducción de texto, facilita el enriquecimiento de datos con información impulsada por IA Generativa utilizando herramientas familiares en el consumo.

11. Recomendaciones de IA Generativa. Debe incorporar recomendaciones de modelado de joins impulsadas por IA Generativa, lo que ayuda a acelerar la creación de nuevas vistas de datos, especialmente cuando no existen relaciones predefinidas.

12. Servidor MCP para participar en el desarrollo de agentes**:** El servidor MCP desempeña un papel crucial en la habilitación de agentes, permitiendo que estos interactúen de manera dinámica con herramientas y fuentes de datos externas. MCP (Model Context Protocol) es un marco que proporciona acceso estructurado a herramientas, API y datos, ayudando a los sistemas de IA a realizar tareas complejas de forma autónoma.

4.2 Modelado de datos

4.2.1 Características conceptuales

1. Permitir la creación de vistas integradas que combinen datos de cualquier fuente/tipo de datos, ya sea en la nube, en las instalaciones del cliente o distribuidos en diferentes áreas geográficas.

2. Disponer de una herramienta de modelado gráfica, con facilidad de uso, cero códigos.

3. Posibilidad de tener las mismas capacidades ejecutando código.

4.2.2 Requisitos requeridos

1. La plataforma debe proporcionar una interfaz de usuario común para el diseño y desarrollo, incluidos todos los estilos de entrega de datos disponibles, con una configuración tipo point-and-click y zero coding. Se esperan interfaces gráficas de usuario basadas en contexto para todas las fases de diseño, incluyendo conectividad de datos, transformación, calidad de datos, optimización de consultas, análisis de impacto, etc.

2. Las interfaces disponibles deben proporcionar menús sensibles al contexto, drag and drop, autocompletar, ayuda en línea, validación y herramientas de depuración para un diseño y prueba rápidos y seguros.

3. El producto debe proporcionar un enfoque de modelado relacional utilizando asistentes gráficos para operaciones comunes de SQL (uniones, uniones, agregaciones, proyecciones, aplanamiento, etc.)

4. Las IU deben permitir a los usuarios diseñar completamente una capa de datos virtual. El proceso de conectarse a fuentes de datos, combinarlas para crear vistas derivadas y publicar vistas debe manejarse a través de un conjunto gráfico de menús, asistentes y acciones de arrastrar y soltar. Los usuarios crearán vistas derivadas, que consistirán en especificar combinaciones / transformaciones entre la vista base y otras vistas derivadas, a través de operaciones de Join, Intersection, Union, Minus y otras operaciones de álgebra relacional. La definición de estas vistas derivadas se debe poder hacer gráficamente desde la herramienta diseño o utilizando un lenguaje de consulta de virtualización compatible con ANSI SQL. También debe ser posible gestionar la capa de datos virtuales, p. ej. creando usuarios y roles, configurando y monitoreando el servidor, o ajustando el procesamiento de consultas, completamente a través de la IU.

5. Entorno de desarrollo 100% web.

6. Proporciona una alternativa de secuencia de comandos a la interfaz gráfica para el modelado empleando una sintaxis compatible con SQL que incluya sentencias DDL (utilizadas para el desarrollo de modelos) y sentencias DML (que se utilizarán para consultar esos modelos).

7. Soporte nativo para tipos de datos complejos (registros y matrices) para modelar estructuras jerárquicas como APIs de servicios web.

8. La solución debe soportar diferentes tipos de datos para representar la información obtenida de los orígenes de datos. Tipos básicos, incluidos numéricos (por ejemplo, integer, long, float, double types), booleano, fecha / hora (por ejemplo, fecha, hora, tipos de timestamp), dinero, xml, valores binarios, clob / blob, tipos de datos geoespaciales, etc. También tipos compuestos para representar matrices, registros y enumeraciones.

9. La plataforma de virtualización debe permitir el descubrimiento automático de metadatos y esquemas en la más amplia gama de fuentes de datos. La metainformación extraída de los orígenes de datos debe incluir el tipo de dato en origen, claves primarias y foráneas, restricciones de NOT NULL, índices y estadísticas de campo (número de valores únicos por columna, número de nulos, número total de filas, tamaño promedio de columna, valores mínimos y máximos por columna).

10. También representación nativa de tipos de datos compuestos (por ejemplo, datos jerárquicos de XML, servicios web, bases de datos orientadas a documentos, SAP BAPI, etc.). Esto permite modelar estructuras jerárquicas como las que devuelven la mayoría de las fuentes de datos NoSQL y API de forma nativa, sin obligar al usuario a aplanarlas en un formato tabular. De este modo se eliminan las transformaciones personalizadas para acoplar, requiere menos vistas para representar correctamente los datos, admite consultas jerárquicas, se ejecuta más rápidamente y facilita la publicación de servicios web.

11. Soporte de tipos para almacenar valores de texto grandes y valores binarios. Soporte para valores de texto grandes como parámetros de entrada o salida. Los valores binarios pueden incluir imágenes, documentos PDF, documentos de Microsoft Office, documentos de texto, etc.

12. Perfilado de datos: La herramienta debe importar índices, PK, FK, estadísticas (información básica de perfilado), descripciones, etc. de fuentes de datos

13. Soporte de modelado Bottom-up y Top-down. El diseño top-down permitirá al usuario definir vistas que representen un “contrato” con las aplicaciones consumidoras. Este proceso dará como resultado un conjunto de entidades comerciales (implementadas como vistas “contrato”) que no están vinculadas ni vinculadas a ningún modelo físico. Una vez que las entidades comerciales se definen como vistas “contrato”, el diseñador proporciona una implementación para ellas. Dado que la vista de implementación y la vista de interfaz están desacopladas, es fácil generar datos para probar antes de la creación de la implementación final. Estas vistas facilitarán, llegado el momento, el reemplazo de unos orígenes de datos por otros sin que las aplicaciones consumidoras tengan que darse cuenta de dicho cambio.

14. Modelado de relaciones entre fuentes

15. Integración con herramientas de modelado de terceros. La solución debe ser capaz de importar modelos de datos de referencia / estándares de industria y definiciones de esquemas de XSD, WSDL, archivos XML y de herramientas de modelado de terceros como CA ERwin, Embarcadero ER / Studio, IBM Infosphere Data Architect (Rational) o SAP PowerDesigner, entre otros.

16. Amplia librería de funciones de transformación de datos. La solución debe proporcionar una amplia biblioteca de funciones de transformaciones out-of-the-box que incluyan: aritmética, trigonometría, procesamiento de texto / fecha, conversión de tipo, estadística / analítica, espacial, cifrado, pivotante (transformación de columna a fila y de fila a columna), condicional, comparación, XML, XPath / XQuery, XSLT, similitud textual, búsquedas y mapeos, operadores de nest / flatten para manipulación de datos jerárquicos, expresiones regulares y más.

17. Esta librería de operadores debería ser extensible, de forma que se puedan desarrollar y conectar fácilmente al servidor, trabajando posteriormente como ciudadanos de primer nivel dentro de la plataforma mediante un SDK para funciones personalizadas (UDF) y transformaciones.

18. Soporte de funciones geoespaciales. Soporte integral para datos geoespaciales en la capa de virtualización de datos que pueden exponerse a herramientas de visualización con visualización de datos geoespaciales.

19. Permitir la asignación de etiquetas (“tags”) a vistas y columnas. Estas etiquetas son términos asignados a vistas y servicios web que ayudan a describirlos y permiten encontrarlos navegando o mediante búsquedas. Las etiquetas son una forma de clasificar los elementos del catálogo de acuerdo con algunos criterios (por ejemplo, vistas “validadas” y “públicas”).

20. Integración nativa con sistemas de control de versiones (por ejemplo, SVN y Git) para desarrollo colaborativo y control de versiones.

4.3 Entrega de datos

4.3.1 Requisitos conceptuales

1. Debe estar diseñado para múltiples opciones de entrega de datos para satisfacer las necesidades del cliente a través de interfaces SQL (JDBC, ODBC y ADO.Net), MDX para análisis OLAP, API.

2. Capa completa de servicios de datos con soporte OAuth 2.0, SAML, OpenAPI, OData 4, GraphQL, para una fácil interoperabilidad con los sistemas en nube.

3. Capacidad para desarrollar y desplegar APIs de datos (DaaS) fácilmente y sin código.

4.3.2 Requisitos requeridos

1. Múltiples interfaces de entrega de datos. Entrega de datos sincrónica a través de interfaces SQL como JDBC, ODBC y ADO.NET, y API como SOAP, REST (representación XML, JSON, HTML o RSS), OData y GraphQL Web Services. Entrega asincrónica a través de la integración con colas de mensajes compatibles con JMS y Kafka.

2. Creación y despliegue de APIs (data services) con cero códigos, incluyendo soporte para SOAP, REST, OData 2 y OData 4.

3. Soporte para operaciones de IDU (inserción, borrado, actualización). El producto soporta operaciones IDU a través de todas las interfaces disponibles (SQL, SOAP, REST)

4. Documentación automática de servicios web publicados, utilizando OpenAPI (Swagger).

5. La solución debe poder ser utilizada con cualquier aplicación, software analítico o de visualización compatible con cualquiera de los métodos estándar previamente citados. Entre ellos: PowerBI, SAP

BO, Oracle Analytics Server, Qlik, SQL Server Reporting Services (SSRS), Amazon Quick Suite, Tableau, Looker, Excel, entre otros clientes de terceros.

6. Soporte integrado para GraphQL. GraphQL es un lenguaje de consulta para APIs de datos que permite al usuario especificar los datos que necesita mediante consultas tipo gráfico y simplifica enormemente el acceso a múltiples endpoints REST. Soporte integrado para GraphQL que permita que las vistas virtuales puedan invocarse vía GraphQL out-of-the-box con una aproximación zero-coding y, que, además, se encargue de la combinación de datos desde múltiples puntos finales para que el rendimiento también se mejore.

7. Notebook integrado para científicos de datos. Los científicos de datos suelen dedicar la mayor parte de su tiempo a tareas de preparación e integración de datos, en lugar de crear y perfeccionar sus modelos de ciencia de datos. Con la introducción de la virtualización de datos se busca minimizar este problema al proporcionar una forma muy ágil de exponer vistas de datos y servicios de datos en formatos que son amigables para los científicos de datos, y al tiempo que los abstrae de complejidades como dónde se encuentran los datos, las tecnologías nativas que se utilizan en cada dato, fuente y otras complejidades de integración de datos.

8. Además, para mayor facilidad de acceso a los datos por parte de los científicos de datos, sería deseable una integración con una herramienta de Notebook, para que los científicos de datos puedan acceder fácilmente a cualquier vista de datos de la capa de virtualización. Se espera que exista también integración en cuanto a seguridad, incluido el soporte SSO, de forma que todas las políticas de seguridad y gobernanza definidas en la capa de virtualización se apliquen cuando los científicos de datos utilicen el Notebook.

4.4 Catálogo de datos

4.4.1 Requisitos conceptuales

1. Herramienta de catálogo de datos que proporcione capacidades de descubrimiento de datos y autoservicio para usuarios de negocio, que les permita explorar tanto metadatos como datos en una única herramienta web.

2. Integración con catálogos de datos de terceros y herramientas de visualización e informes.

3. Catálogo de datos habilitado para AI/ML que proporcione funciones de descubrimiento interactivo, capacidades de colaboración y recomendaciones personalizadas.

4.4.2 Requisitos requeridos

1. La solución de virtualización debe integrar un Catálogo de Datos que permita a los usuarios de negocio y desarrolladores buscar y explorar los conjuntos de datos disponibles, incluyendo, pero no limitado a, búsqueda en los metadatos, búsqueda en los datos, previsualización de datos, interfaz de consulta, opciones de colaboración, extensión de metadatos, etc.

2. Interfaz 100% web.

3. Catálogo de datos 100% sincronizado con la plataforma de entrega de datos. Esto permitirá no solo descubrir, catalogar y localizar conjuntos de datos disponibles, sus metadatos y su linaje, sino que también permitirá una rápida integración de todas las fuentes de datos empresariales.

4. Mostrar estadísticas de uso de los diferentes conjuntos de datos publicados en el Catálogo de Datos. Uso de técnicas de AI/ML para sugerir automáticamente conjuntos de datos a los usuarios basados en dichas estadísticas.

5. Funciones de colaboración: los usuarios podrán respaldar conjuntos de datos o registrar comentarios o advertencias sobre ellos. Esto ayudará a los usuarios a contextualizar aún más el uso del conjunto de datos y comprender mejor cómo los experimentan otros usuarios.

6. Una experiencia sólida para descubrir datos. El Catálogo de Datos debe tener con una interfaz intuitiva al estilo del comercio electrónico que facilita a los usuarios de negocio la exploración y el acceso a productos de datos confiables sin depender de equipos técnicos. Al mostrar metadatos significativos e información sobre el uso, el mercado ayuda a los usuarios a encontrar rápidamente los datos adecuados para impulsar decisiones e innovación.

7. Trazabilidad de extremo a extremo para una visibilidad completa. Debe incluir la trazabilidad de los datos para incluir informes y dashboards creados en herramientas Power BI, Oracle Analytics Server y Amazon Quick Suite.

8. Personalización del idioma. Cada usuario puede personalizar el idioma de la interfaz, lo que fomenta una mayor adopción y una experiencia de autoservicio más inclusiva.

4.5 Seguridad /Gobernanza de datos

4.5.1 Requisitos conceptuales

1. Capa unificada de gobernanza y seguridad. La plataforma permite definir, administrar y hacer cumplir las reglas de seguridad y gobierno de datos desde un único punto. Además, dichas políticas se definen en términos de un modelo canónico de entidades de negocio.

2. Soporte completo de gobernanza y seguridad, incluyendo enmascaramiento de datos avanzado y control de acceso basado en roles (RBAC). Disponible para todos los activos de datos, a través de un único punto de control y administración.

4.5.2 Requisitos necesarios

1. La virtualización de datos ofrece un único punto lógico de acceso, evitando las conexiones punto a punto de las aplicaciones consumidoras a las fuentes de información. Como punto único de acceso a datos para aplicaciones, es el lugar ideal para hacer cumplir las restricciones de seguridad de acceso que se pueden definir en términos del modelo canónico con una granularidad muy fina.

2. La solución de virtualización debe asegurar el acceso de las aplicaciones consumidoras a las fuentes de datos finales de extremo a extremo a través de conexiones SSL / TLS entre el

consumidor y la capa de virtualización y mediante el protocolo de seguridad de fuente de datos específico entre la capa de virtualización y las fuentes de datos (por ejemplo, SSL / TLS, HTTPs o sFTP).

3. La solución debe admitir mecanismos de autorización y autenticación integrados de usuarios y roles con permisos a nivel de esquema (por ejemplo, para acceder a bases de datos y/o vistas concretas) y permisos a nivel de datos (por ejemplo, para acceder a ciertas filas o columnas de una vista).

4. La solución debe poder delegar la autenticación a un servidor LDAP o Active Directory externo, de modo que no sea necesario definir usuarios en el sistema de administración de usuarios incorporado. El sistema LDAP / AD proporciona el rol(es) del usuario, y éstos se usan para restringir el acceso a cualquier base de datos o vista dentro del servidor de virtualización de datos.

5. Soporte para Single sign-on (SSO), incluyendo, pero no limitado a OAuth, OpenID, Kerberos y SAML.

6. Soporte autenticación en dos pasos.

7. Integración con password vaults.

8. Soporte para autorizaciones basadas en usuarios y / o roles (RBAC).

9. Control de acceso de grano fino. Restricciones a nivel de columna y fila. Seguridad a nivel celular.

10. La solución de virtualización puede imponer permisos estrictos y detallados de usuario y basados en roles para todos y cada uno de los elementos definidos en él. Las políticas de autorización pueden implementar la seguridad de filas y columnas. Estas políticas se especifican por vista, por fila (especificadas con una condición de selección), por columna o por combinación de fila y columna (es decir, filas restringidas, pero solo para columnas restringidas), y se evalúan antes de la ejecución de cada consulta para determinar si el cliente está Permitido ver resultados particulares o no. Los datos no autorizados pueden ser 'filtrados' (es decir, eliminados de los resultados de la consulta) o 'enmascarados'.

11. Permitir políticas de seguridad globales basadas en semántica, las cuales son más fáciles de administrar ya que cuando la misma política de seguridad se aplica a varias vistas, puede definirse una vez en lugar de tener que asignarla a cada una de esas vistas. Las políticas globales a menudo se usan junto con "etiquetas", estas etiquetas se pueden asignar a vistas y columnas. Por ejemplo, se puede definir una política global para que solo los usuarios con un determinado rol puedan consultar las vistas que tienen la etiqueta "secreto". También puede definir una política global que enmascara todas las columnas que tienen la etiqueta "confidencial".

12. Autenticación:

    a. Integración con LDAP / Active Directory / OKTA

    b. Autenticación avanzada para servicios web y APIs (p. ej., SAML, OAuth2)

    c. Inicio de sesión único (SSO) en todos los clientes

d. Transferencia de credenciales a fuentes (pass-through) para aprovechar las políticas de seguridad definidas en los orígenes de datos

e. Integración con password vaults externos para almacenar las contraseñas para acceder a los orígenes de datos

f. Soporte para almacenar atributos de la sesión (por ejemplo, el departamento del usuario) durante la autenticación para usarlos más tarde para la autorización dinámica basada en esos atributos, o para inicializar los atributos de la conexión a las fuentes de datos.

13. Autorización: Autorización basada en roles RBAC

    a. Soporte para herencia de roles

    b. Control de acceso a nivel de detalle. Restricciones a nivel de columna y fila. Seguridad a nivel de celda.

    c. Políticas de seguridad basadas en semántica, es decir, en etiquetas asignadas a los datos (por ejemplo, “PII”, “HIPAA”, “confidencial”). Por ejemplo, soporta la definición de políticas de seguridad globales para que solo los usuarios con un determinado rol puedan consultar las vistas que tienen la etiqueta "PII". El sistema puede importar las etiquetas de un sistema externo.

    d. Políticas de seguridad coherentes en todos los mecanismos de entrega de datos. El producto permite la definición de políticas de seguridad una vez y las aplica a todas las interfaces (por ejemplo, SQL, API REST, catálogo de datos)

    e. Privacidad: enmascaramiento de datos. La herramienta permite elegir cómo enmascarar los "valores confidenciales", por ejemplo, los primeros/últimos caracteres, establecer los valores numéricos en 0 o 1, aplicar expresiones personalizadas o establecer valores nulos.

    f. Cifrado de datos en reposo / en movimiento

    g. SDK para definir políticas de autorización personalizadas

4.6 Optimización y rendimiento

4.6.1 Requisitos conceptuales

1. Disponibilidad de técnicas de optimización de consultas de última generación para lograr el máximo rendimiento. La plataforma deberá maximizar la delegación de consultas a las fuentes de datos para lograr un buen rendimiento sin tener que mover los datos a la plataforma de virtualización o copiar los datos en un repositorio central.

2. Optimizador dinámico basado en costes que seleccione el mejor plan de ejecución para cada consulta, sin depender de ajustes manuales por parte de los desarrolladores.

3. Disponibilidad de técnicas inteligentes de aceleración de consultas, basadas en IA, para escenarios analíticos complejos. Soporte de aggregation awareness, ejecución paralela (MPP).

4.6.2 Requisitos necesarios

1. Especialmente críticas son las capacidades de la plataforma para optimizar el trasiego de datos y la ejecución de las consultas de manera óptima. La plataforma de virtualización deberá disponer al menos de las características de optimización que se describen a continuación, priorizando al máximo la ejecución en el propio origen de datos, y minimizando el trasiego de datos a la capa de virtualización siempre que sea posible.

2. La plataforma debe permitir consultas ad hoc, y por tanto es necesario optimizar la ejecución con planes de ejecución dinámicos para cada consulta. Esto es especialmente relevante para el BI de autoservicio, donde no es posible confiar en el ajuste manual del rendimiento.

3. La plataforma aplicará tanto técnicas basadas en reglas estáticas (heurísticas) como técnicas de optimización basadas en costes. Para ello, deberían tenerse en cuenta datos estadísticos como el tamaño de las tablas, el número de campos, el tamaño de cada campo, valores mínimo y máximo, número de valores diferentes, etc.; así como capacidades específicas del origen de datos como el tipo (que determina sus capacidades de procesamiento, por ejemplo, base de datos o ficheros), modo (por ejemplo, fuentes de solo lectura o que permite mover una cantidad pequeña de datos externos a ella para delegar por completo una consulta) o tamaño (por ejemplo, número de nodos de una fuente MPP). Todo lo anterior se debe tener en cuenta por el optimizador basado en costes para seleccionar automáticamente el plan de ejecución.

4. Optimizador basado en costes (CBO) multifuente

5. Aceleración de consultas en escenarios analíticos, incluyendo soporte para datos pre-agregados (aggregation awareness / summaries) e integración con MPP (Multi Parallel Processing) para ejecutar consultas sobre grandes volúmenes de datos.

6. Disponibilidad de técnicas de optimización de consultas de última generación (optimización de predicados, reescritura de consultas) que incluyen, entre otras, la aggregation push-down, join reordering, branch pruning, etc.

7. Delegación de operaciones de agregación (completo y parcial) para escenarios analíticos, por ejemplo, consultas con “join” y “group by” en esquemas en estrella distribuidos y/o particionados entre varios orígenes de datos. Por ejemplo, el optimizador mueve las operaciones “group by”, comunes en consultas multidimensionales, por debajo de operaciones de “join” y “union” para maximizar la delegación.

8. Simplificación de uniones particionadas (partition pruning): cuando los datos de una vista de unión se distribuyen en varias fuentes de datos y una consulta solo necesita datos de una rama de la unión, el optimizador elimina las ramas que no participarán en la consulta, permitiendo delegar posteriormente las operaciones sobre la vista requerida.

9. Data movement: Técnica indicada para situaciones en las que un pequeño conjunto de datos en una fuente necesita combinarse con un gran conjunto de datos en otra fuente, especialmente cuando los resultados de la combinación deben agregarse por encima. El optimizador debe poder mover automáticamente los datos de la fuente con el conjunto de datos más pequeño a una tabla temporal en la fuente de datos con el conjunto de datos más grande, para entonces delegar la ejecución de la consulta completa, logrando mejoras de rendimiento significativas.

10. Protección de datos sensibles en las optimizaciones. Debe permitirse excluir datos sensibles de técnicas de optimización como el movimiento de datos en tiempo real o Data Movement, cuando dichas acciones pueden violar requisitos regulatorios. Esto protege la información sensible y facilita el cumplimiento de normativas de seguridad, sin importar dónde resida tu información.

11. Aceleración inteligente de consultas. El optimizador debe lograr mejoras significativas en el rendimiento utilizando de manera inteligente resúmenes agregados previamente (“summaries”) de datos transaccionales para optimizar múltiples consultas posteriores.

12. Los "summaries" deberán ser creados por el administrador bajo las sugerencias que le haga la solución de forma automática. Al analizar los diferentes patrones de consulta en un escenario dado y el rendimiento de consultas pasadas, el sistema aplicará un modelo de Machine Learning para identificar “summaries” adecuados que ayuden aacelerar consultas futuras. Los summaries se podrán almacenar en la caché o en una fuente de datos externa, definida por el usuario.

13. Se utilizarán datos de monitorización (por ejemplo, métricas de uso, tiempos de ejecución, distribución y volúmenes de datos, etc.) para alimentar el motor de recomendaciones. Este motor se deberá utilizar para ayudar a los administradores a ajustar el rendimiento.

14. Capacidad de cargar resúmenes directamente desde la fuente de datos al destino cuando ambos son la misma base de datos, es decir, sin pasar por la capa de virtualización de datos.

15. Procesamiento en paralelo. La solución de virtualización debe incluir integración nativa con motores MPP para la aceleración de consultas en escenarios complejos de big data. La solución llevará a cabo el procesamiento paralelo de operaciones dentro de la capa de virtualización en un clúster MPP que puede basarse en tecnologías de procesamiento paralelo como Spark. Además, es deseable que también se proporcione almacenamiento en caché MPP para un acceso rápido a los datos usados recientemente.

16. El mecanismo de aceleración de consultas MPP debe estar integrado en el modelo de optimización basado en costes de la solución. La plataforma toma la decisión final sobre si ejecutar el procesamiento final en memoria o delegar el procesamiento al clúster MPP.

17. Optimización inteligente basada en costos para subconsultas

18. Lecturas vectorizadas de Parquet

19. Almacenamiento en caché avanzado

20. Optimizaciones avanzadas para aprovechar tablas que existen en más de una fuente de datos (fuentes alternativas en la ejecución de una consulta)

4.7 Caché/ Replicación Selectiva

4.7.1 Requisitos conceptuales

1. Capacidades de replicación selectiva de datos para dar respuesta a los diversos casos de uso de un Data Fabric, incluido el almacenamiento en caché, tablas temporales y tablas permanentes. Posibilidad de utilizar dichos mecanismos tanto en los orígenes de datos como en otros sistemas (por ejemplo, un data warehouse o un data lake).

2. Soporte versátil para almacenamiento en caché, con un doble propósito: mejora el rendimiento ya que los datos se recuperan directamente desde la caché en vez de desde los orígenes de datos; evitar acceder a los orígenes de datos, que podrían estar sobrecargados en momentos de pico de trabajo.

4.7.2 Requisitos necesarios

1. Capacidades de replicación de datos para casos de uso de Data Fabric: tablas externas. Esta característica es útil, por ejemplo, cuando el acceso directo a la fuente no es óptimo por razones de rendimiento. La solución deberá poder cargar fácilmente los datos en una zona de “landing” en algún lugar distinto del caché, haciendo que la transición sea completamente fluida. También es útil para almacenar resultados intermedios en tablas de bases de datos externas para usarlas más tarde, por ejemplo, para simplificar los flujos de científicos de datos.

2. La solución deberá permitir el uso de caché. La posibilidad de emplear una caché ofrece dos ventajas: por un lado, mejora el rendimiento ya que los datos se recopilan directamente de la caché, lo que evita la consulta de fuentes de datos y el posterior tiempo de procesamiento de consultas en la capa de virtualización de datos. Por otro lado, limita el impacto sobre las fuentes de datos que podrían estar sobrecargadas en determinados momentos.

3. Permite habilitar la caché para una vista concreta

4. La tecnología de almacenamiento en caché deberá ofrecer múltiples modos de configuración que permitan una configuración granular (vista a vista).

5. Múltiples tecnologías. La solución no debe proporcionar un único (o un número muy limitado) soporte de caché, debe soportar el almacenamiento en caché distribuido y flexible basado en tecnologías de base de datos como Oracle, SQL Server o MySQL. Bases de datos paralelas como Vertica o bases de datos en memoria como SAP HANA, Oracle, Redis. Bases de datos en la nube, como AWS RedShift. MPPs como Spark.

6. El cacheado no se hará en el propio producto, sino en una caché relacional externa para poder hacer cache co-location con los datos existentes para maximizar el push-down de consultas. Esta estrategia aprovecha la ubicación conjunta (co-location) para reducir las combinaciones entre los datos almacenados en caché y los datos no almacenados en caché que se almacenan en el mismo sistema.

7. Múltiples modos de funcionamiento. Incluyendo, pero no limitado a, completa, incremental.

8. Soporte para cargas incrementales de la caché.

    1. Capacidad de usar vistas como flujos tipo ELT para la replicación directa dentro de una fuente (por ejemplo, lago de datos)

    2. Capacidad de mecanismos de carga masiva nativos de la base de datos para un almacenamiento en caché más rápido

9. Gestión automatizada de dependencias para replicación. Capacidad para la orquestación de trabajos basada en grafos acíclicos dirigidos (DAG) en el planificador, analizando automáticamente las dependencias de las tablas y secuenciando las tareas de replicación para un rendimiento óptimo. Esto es especialmente útil para gestionar zonas curadas en data lakehouses o implementar estrategias de almacenamiento en caché.

4.8 Semántica y Gobernanza de Datos

4.8.1 Requisitos conceptuales

1. Capacidades de modelado semántico, incluidos asistentes gráficos fáciles de usar, captura de linaje de datos, detección de cambios y análisis de impacto.

4.8.2 Requisitos necesarios

1. La plataforma permite agregar información semántica a los modelos, en forma de etiquetas a vistas o columnas específicas de una vista. En base a dicha información semántica, la plataforma permitirá establecer políticas y controles generales de acceso que se aplicarán a todas las vistas/columnas etiquetadas con una determinada etiqueta.

2. La solución proporcionará capacidades de análisis gráfico completo de "linaje de datos" y "uso de datos". El linaje en ambos sentidos, es decir, desde los orígenes de datos a las vistas de negocio y desde éstas a los orígenes de datos de dónde provienen se puede consultar gráficamente.

3. Detección de cambios en las fuentes. Para soportar escenarios donde los esquemas de origen cambian de manera que afecten las vistas virtualizadas, la solución debe proporcionar funciones de actualización de los orígenes de datos y análisis de impacto de cambios, de modo que se pueda regenerar el modelo virtual.
4. Integración con herramientas de gestión de datos de terceros. Posibilidad de integración con soluciones de gestión de metadatos de terceros como Informatica Data Catalog y Metadata Command Center o Anjana Data.

4.9 Procedimiento por lotes

4.9.1 Requisitos conceptuales

1. La herramienta deberá proporcionar capacidades para la ejecución planificada de tareas habituales como carga en caché, replicación selectiva, monitorización de cambios en las fuentes, etc.

4.9.2 Requisitos necesarios

1. La solución debe proporcionar un programador de tareas para ejecutar tareas batch. Los usos típicos serían exportar datos a una base de datos, a ficheros delimitados, CSV, ficheros planos, MS Excel, índices, cargas de caché y tareas de administración periódicas (recopilación de estadísticas de datos, datos tareas de actualización de caché, buscar cambios en las fuentes, etc.).

2. Integración: la solución ofrece un API para la poder invocar la ejecución de tareas desde herramientas de planificación de terceros, por ejemplo, Control-M
3. La herramienta permite enviar notificaciones/alertas por correo electrónico sobre la ejecución de tareas
4. La herramienta deberá soporta la definición de dependencias entre tareas y reintentos de tareas

5. La herramienta deberá soporta la programación de tareas batch, para diferentes propósitos:

    - Tareas de carga/actualización de caché

    - Tareas de indexación de contenido, para búsqueda basada en palabras clave

    - Tareas de gestión de metadatos: alertas de cambio de fuente, recopilación de estadísticas, mantenimiento de caché (borrado automático de datos obsoletos), etc.

    - Exportación de datos a archivos o bases de datos externas

4.10 Desarrollo y Operaciones (DevOps)

4.10.1 Requisitos conceptuales

1. La plataforma proporciona una consola centralizada para la gestión de despliegues completos de la solución, incluidos los entornos de desarrollo, preproducción y producción.

2. Gestión automatizada de infraestructura para despliegues en la nube para despliegues en entornos híbridos y en nube

3. Monitorización en tiempo real de todos los componentes de forma integrada.

4. Auditoría de todas las consultas, parametrizaciones y actividades realizadas en el sistema.

5. Integración con soluciones de monitorización de terceros.

4.10.2 Requisitos necesarios

1. Gestión centralizada de despliegues complejos, con múltiples clústeres. Gestión centralizada de actualizaciones, claves de licencia, parámetros específicos de diferentes entornos (por ejemplo, parámetros de conexión a los orígenes de datos), monitorización, registros de log y auditoría.

2. Soporte gráfico para promociones de modelos entre los diferentes entornos

3. Soporte para despliegues automatizados en proveedores cloud, incluyendo pero no limitado a, AWS, Azure y GCP. Gestión automática de clusters, balanceadores de carga, monitorización integrada de clusters, migración entre entornos, autoescalado, etc.

4. Soporte para despliegues en clúster y en alta disponibilidad.

5. Gestión automatizada de la infraestructura para la nube: soporte que incluye configuración de clústeres (TLS, balanceo de carga, escalado automático, etc.), controles de start / stop, instalación automática de actualizaciones y monitorización integrada.

6. Automatización: Gestión automatizada de infraestructura en nube. Automatización de la instalación, configuración, implementación y actualización de la solución en entornos cloud (AWS, Azure y GCP)

7. Integración: API para la integración con sistemas de gestión del ciclo de vida de terceros, por ejemplo, Jenkins.

8. Monitorización: Herramienta de monitorización web integrada, incluyendo parámetros de CPU, memoria, sesiones, consultas, caché, etc.

9. Monitorización: Los logs se pueden guardar en disco o en base de datos, en on premise o en la nube (Amazon S3 y Azure Blob Storage), para su posterior análisis e integración con herramientas de análisis de logs de terceros

10. Monitorización: Posibilidad de cargar logs para "retroceder en el tiempo" y representar gráficamente el estado del sistema en un momento dado

11. Monitorización: La información de monitorización se expone a través de JMX para la integración con herramientas de monitoreo de terceros. Integración con servicios de monitorización en la nube, incluidos, entre otros, AWS CloudWatch y Azure Monitor

12. Auditoría Finops: Registro completo de todas las consultas y actividades realizadas en el sistema, incluyendo usuario, IP de origen, hora, consulta, tablas involucradas, etc.

13. Auditoría Finops: Registro de intentos de inicio de sesión (exitosos y fallidos)

14. Auditoría Finops: Uso del histórico de consultas para alimentar procesos de automatización basados en IA.

15. Desarrollo: Integración nativa con sistemas de control de versiones (SVN y Git) para desarrollo colaborativo

4.11 Despliegues

4.11.1 Requisitos conceptuales

1. Soporte de despliegues en on-premise, en los hiperescalares disponibles en EJIE (AWS, Azure o GCP, IBM y Oracle), así como despliegues híbridos (on-premise + nube).
2. Soporte para despliegues distribuidos geográficamente, por rendimiento o cumplimiento de regulación
3. Soporta despliegues en contenedores
4. El software deberá proporcionar las mismas funcionalidades en la nube pública/privada y en instalaciones on-premise.
5. La solución permitirá incluir, sin coste adicional, un motor MPP basado en Presto totalmente integrado en la plataforma y con soporte incluido.
6. Soporta despliegues en contenedores (p. ej., Docker), incluida la compatibilidad con Kubernetes (K8s)
7. Soporta despliegues en nube, incluido AWS, Azure y GCP.
8. Soporta despliegues híbridos y multinivel
9. Compatibilidad con despliegues en contenedores en nube, incluido EKS (Amazon Elastic Kubernetes Service), AKS (Azure Kubernetes Service) y GKE (Google Kubernetes Engine)
10. Soporta escalado automático
11. Despliegue directo desde los marketplaces de AWS, Azure y Google Cloud

4.11.2 Arquitectura y despliegue

- Acceso mediante:
    - Interfaz gráfica (web).
- Modalidades de despliegue con recursos dedicados y posibilidad de segmentar en múltiples clientes
- La solución deberá ofrecer el servicio desde CPDs ubicados en regiones de la Unión Europea.

4.12 Requisitos de seguridad, cumplimiento y certificaciones

La solución deberá cumplir con altos estándares en materia de seguridad de la información, privacidad y cumplimiento normativo, acreditables mediante certificaciones oficiales vigente, las cuales se detallan en este apartado.

4.12.1 Cumplimiento normativo

La solución deberá facilitar el cumplimiento de:

- A nivel europeo cumplir con el Reglamento General de Protección de Datos (RGPD).
- A nivel nacional cumplir con la LOPDGDD que adapta la europea y detalla aspectos específicos.
- NIS 2 – Normativa marco de ciberseguridad en torno a redes y seguridad de la información

4.12.2 Arquitectura segura

El sistema deberá:

- Estar desplegado en entornos protegidos (VPC, firewalls, control de puertos).
- Despliegues en entornos cloud
- Garantizar comunicaciones seguras entre componentes propios de la solución, así como con componentes ajenos como, por ejemplo, bases de datos operacionales
- Asegurar que los accesos administrativos se realizan mediante canales seguros (ej. SSH).

4.13 Licenciamiento y onboarding

El licenciamiento por suscripcion deberá incluir:

- **Configuración**, al menos una configuración anual mínimo de 8 cores, productos de datos incluidos al menos 100 y Volumen de datos incluidos al menos 7.5 TB/año
- **Documentación**, oficial de la solución
- **Nuevas versiones**, actualizaciones y mejoras funcionales
- **Soporte fabricante,** 24x7 local o international support center telephones for severity 1 Service Request (SR)
- Al menos dos sesiones de **asesoría** por mes con expertos del producto sobre temas específicos.
- Al menos dos **evaluaciones** por año de contrato.
- **Onboarding a la solución**, donde el fabricante de la solución deberá realizar las siguientes tareas:

4.13.1 Configuración e integración con los sistemas corporativos

Configuración de la solución e integración de la misma con los sistemas corporativos de EJIE, los cuales comprenden:

- Configuración de comunicación segura entre la solución en cloud y el cloud híbrido de EJIE, que comprende tanto cloud pública como infraestructura on premise en CPDs propiedad de EJIE
- Configuración de la solución para ofrecer servicio a múltiples clientes con aislamiento a nivel de accesos y permisos, orígenes de datos, plantillas, etc.
- Configuración e integración con solución de identidades corporativas tanto de EJIE, como de cliente. A fecha de elaboración de este documento se contemplan Okta y Microsoft EntraID
- Configuración de conexión contra una base de datos relacional on premise, y otra en cloud pública
- Configuración del API de la solución para que se permita el uso desde desarrollos desplegados tanto en on premise, como en cloud pública
- Integración con la herramienta corporativa de Gobierno del Dato (Informatica IDMC).

4.13.2 Formación en la solución

Con la suscripción deberá estar incluída la formación y certificaciones del fabricante al personal de EJIE:

- Acceso a todo el catalogo eLearning del producto para los usuarios con email del dominio de EJIE.
- Certificaciones en el catálogo de formación del producto