Software Requirements Specification

for

Gaia-X Federation Services

Compliance
Continuous Automated Monitoring
CP.CAM

Published by

eco – Association of the Internet Industry (eco – Verband der Internetwirtschaft e.V.)
Lichtstrasse 43h
50825 Cologne
Germany

Table of Contents

List of Figures

List of Tables

1. Introduction

This document specifies the architecture, and functional and non-functional requirements of the Continuous Automated Monitoring (CAM) core service within Gaia-X. Continuous automated monitoring (CAM) involves consistently gathering and assessing compliance-relevant information about Service and Node operations by a GAIA-X monitoring body to validate whether they continuously adhere to GAIA-X criteria. The term continuous automated auditing (CAA) builds on top of the continuous automated monitoring, by including an accredited external auditor. The role of the auditor is to validate the chain of evidence generated by the CAM (and the CAM itself), based on a recognized conformance assessment methodology.
Continuous automated auditing represents a disruptive change because it provides stakeholders with ongoing, up-to-date feedback about relevant Monitoring Areas. This in comparison with conventional auditing approaches, which assess a cloud service only at a specific point in time.

To get general information regarding Gaia-X and the Gaia-X Federation Services please refer to [TAD].

1.1 Purpose

The innovative continuous monitoring concept has recently gained importance in cloud service contexts.
Prior research has already proposed conceptual architectures and techniques to continuously monitor and audit services and providers that one can categorize as either ***test-based*** or ***monitoring-based*** approaches.

Test-based approaches	GAIA-X monitoring bodies access the cloud service infrastructure and test cloud service components directly.
Monitoring-based approaches	Service or Node Providers monitor their service infrastructure to collect and provide monitoring-relevant information to GAIA-X monitoring bodies.

By using ***test-based methodologies***, GAIA-X monitoring bodies directly access the cloud service infrastructure to examine cloud service components and operations. Typically, test-based techniques produce evidence by controlling some input to the Service and evaluating the output, such as calling a Service's RESTful API and comparing responses with expected results. Prior research has shown that GAIA-X monitoring bodies can apply test-based approaches to verify the integrity of multiple cloud users' data, assess data location, validate adherence to security criteria, and so on. However, GAIA-X monitoring bodies cannot easily apply test-based approaches in practice because they need access to the cloud infrastructure. Service Providers may refuse required infrastructure access due to organizational issues (e.g., resistance to integrate untrustworthy techniques of monitoring bodies) and regulatory issues (e.g., data protection laws). In addition, performing test-based monitoring requires GAIA-X monitoring bodies to configure and adjust applied test-based techniques in accordance with the cloud infrastructure and heterogonous data formats. This adaptation is challenging in cloud service environments because cloud infrastructures exhibit dynamic characteristics (i.e.,

dynamic reassignment of resources) and feature fast technology lifecycles and ongoing technical changes (i.e., due to agile software development), which ultimately limits the extent to which monitoring bodies can apply test-based approaches.

Monitoring-based strategies provide auspicious means to overcome these challenges. When using monitoring-based approaches, cloud service providers monitor their cloud service infrastructure to collect data by themselves and then provide compliance-relevant information to the GAIA-X monitoring bodies. These bodies then assess Service and Node compliance with GAIA-X principles based on the transmitted data. For example, researchers developed a prototypical monitoring-based infrastructure (called *"CUMULUS"*) to, for instance, verify database user identification to validate criteria of cloud service certifications. Likewise, prior research has shown that third parties can use various monitoring metrics and key performance indicators (e.g., availability and resource management indicators and hypervisor security metrics) for monitoring-based purposes. Monitoring-based approaches do not require invasive cloud infrastructure access from monitoring bodies in contrast to test-based approaches. More importantly, unlike in test-based scenarios, Providers can independently alter their cloud infrastructure while ensuring that they still transmit monitoring-relevant data to monitoring bodies. Despite these benefits of monitoring-based over test-based approaches, Providers' sending monitoring-relevant data has one challenging drawback: the risk that they will deliberately manipulate data. Providers may euphemize provided data to assure GAIA-X criteria compliance; therefore, Providers and GAIA-X monitoring bodies must prove that malicious Providers do not manipulate monitoring data. Likewise, Providers must set up sophisticated monitoring systems that aggregate monitoring-relevant data across implemented monitoring technologies and format relevant data in accordance with monitoring bodies' needs.

While both methodologies have advantages and disadvantages, test- and monitoring-based methodologies complement each other because GAIA-X monitoring bodies can use them in parallel to collect diverse evidence about GAIA-X criteria compliance.

1.2 Intended Audience and Reading Suggestions

The document is intended to a technical reader, i.e., a software engineer or architect. Thus, the reader is expected to be familiar with the general concepts of micro-services-based Cloud architectures. Additionally, the reader might need to be familiar with the overall concepts of Gaia-X [TAD], although references will be given to individual terms.

1.3 References

Abbreviation	Description/ Title	Link
[TAD]	The Gaia-X Architecture identifies and describes the concepts of the targeted federated open data infrastructure as well as the relationships between them.	Please refer to annex "Gaia-X_Architecture_Document_2103"

[PRD]	Gaia-X Policy Rules intend is to identify clear controls to demonstrate European values of Gaia-X, such values including Openness, Transparency, Data Protection, Security and Portability.	Please refer to annex "Gaia-X_Policy Rules_Document_2104"
[TDR]	Gaia-X Federation Services Technical Development Requirements	Please refer to annex "GXFS_Technical_Development_Requirements"
[NF.SPBD]	Gaia-X Federation Service Non-functional Requirements Security & Privacy by Design	Please refer to annex "GXFS_Nonfunctional_Requirements_SPBD"

2. Overall Description

In the following, the CAM core service is described from different perspectives.

2.1 Product Perspective

The *Continuous Automated Monitoring (CAM)* component is a core service within the Gaia-X Federation Service. Its main goal is to provide transparency to the users of Gaia-X about the compliance of individual services offered in the Gaia-X Catalogue. The basis for this compliance are certain requirements and rules that Gaia-X itself has imposed on its system, for example requirements coming from the field of security, such as encryption, data privacy or interoperability. Often, existing standards such as the BSI C5 or EUCS are used as a point of reference. The purpose of the CAM service is to automatically gather evidence that can indicate the compliance or non-compliance of a certain Gaia-X service as a whole or by a concrete instantiation of a particular service by a user. This is especially necessary if the offered services are dealing with sensitive data and thus need to have a higher assurance level, e.g., compared to the assurance level "high" in the EU Cybersecurity Certification.

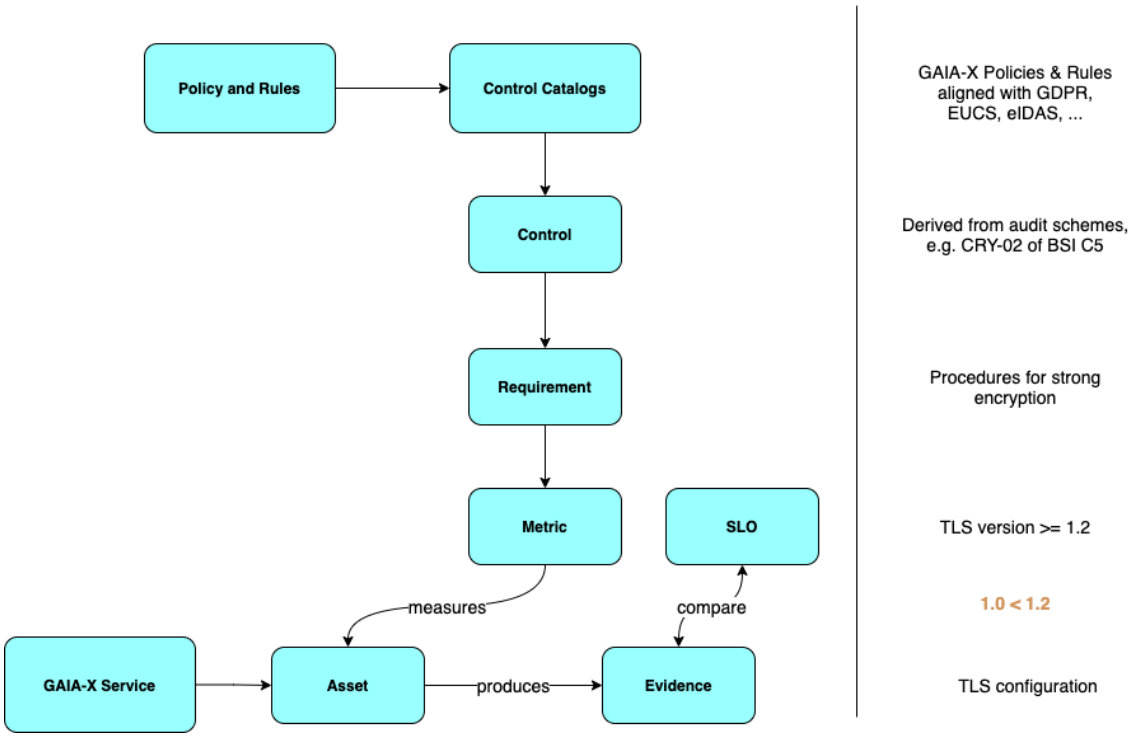


Figure 1: *An overview of the used terminology*

Fehler! Verweisquelle konnte nicht gefunden werden. shows an overview of the terminology that is used t hroughout this document. The basis for the compliance analysis are the Gaia-X *Policy and Rules*, which refer to individual *Controls*, based in a *Control Catalog*, e.g., EUCS[1]. Each Control usually has a set of *Requirements*, detailing the control, usually referring to different assurance levels. Up to the level of a requirement, even though modelling in a language such as OSCAL is preferred, textual descriptions of the controls are used. However, for automated monitoring of security controls, a machine-readable representation is needed. For that reason, a *Metric*, is associated to each requirement. Technically speaking, a metric is a definition of a measurement, that results into a standardized result.

In this context, a metric is applied to properties of a resource and produces results that satisfy the fulfilment of a particular control. For example, this is achieved by automatically interacting with the service-under-test using standardized protocols and interfaces to retrieve technical evidence. For example, to check for the fulfillment of requirements regarding transport encryption, the CAM service might interact with the service using the TLS protocol and gather technical evidence regarding the used TLS version as well as employed cipher suites. This evidence is then later compared, i.e., evaluated, against a set of common best practices also referred to in the compliance catalogs. In the example, best practices from the BSI state that at least TLS version 1.2 should be used.

[1] https://www.enisa.europa.eu/publications/eucs-cloud-service-scheme

2.2 Product Functions

The main function is to evaluate whether a certain target system adheres to the Gaia-X principles. It does so by carrying out technical checks against the target system or by retrieving information from its APIs to monitor its state.

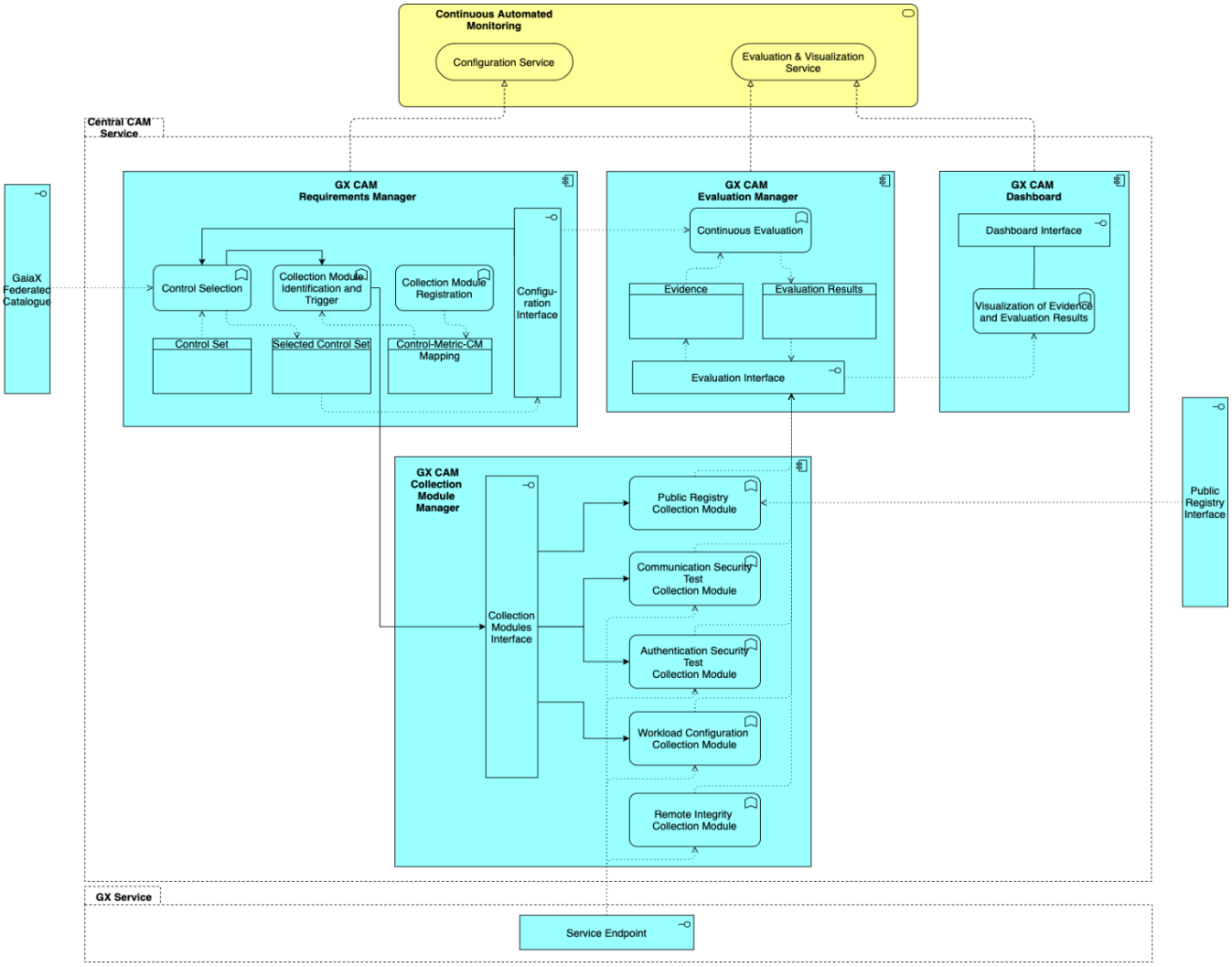


Figure 2: *The architecture of the Continuous Automated Monitoring*

Fehler! Verweisquelle konnte nicht gefunden werden. shows an overview of the envisioned overall a rchitecture of the CAM and its components, according to the ArchiMate standard. In the following, each major component is described briefly. A more detailed description and the requirements of each manager can be found in individual subsections of Section 4.

2.2.1 GX CAM Requirements Manager (GX-CAM-RM)

The main purpose of this component is to manage requirements and to initiate the continuous monitoring process accordingly. Therefore, the GX-CAM-RM needs to hold a set of possible controls that are suitable for

monitoring as well as a mapping to the technical Collection Modules (see below) that implement the measurement of a certain metric for a particular control (see Section 2.6.4). The GX-CAM-RM also provides a functionality to register and deregister Collection Modules.

2.2.2 GX CAM Collection Module Manager (GX-CAM-CMM)

This component can be seen as a collection of different Collection Modules. Each collection module is responsible to collect technical evidence for the fulfilment of a control according to a specific *metric*. We differentiate between different classes of collection modules.

Within the first iteration of this specification, a minimum set of five classes is specified:

- *Public Registry Collection Module*, which gathers additional security and certification related information from public registries (including the Gaia-X Federated Services Catalogue), e.g., additional security certification a service already holds.
- *Communication Security Collection Module*, which uses test-based approaches to assess the communication security of critical parts of the Gaia-X service, e.g., by measuring the quality of a TLS connection.
- *Authentication Security Collection Module*, which may use test- or monitoring-based approaches to assess the quality of employed authentication and authorization techniques by the cloud service. Possible metrics can include correct implementation of state-of-the-art protocols, such as OAuth/OpenID, which also would fulfill requirements with regards to interoperability.
- *Remote Integrity Collection Module,* which gathers information regarding the software stack currently running the Gaia-X service instance. The software stack is described by a list of the deployed system components. These details can be utilized to assess the trustworthiness of the service instance, check for known vulnerabilities, and verify the integrity of the software stack.
- *Workload Configuration Collection Module*, which gathers security- and privacy-related configuration information about a particular instantiation of a Gaia-X service. Whereas the first four modules gather information that applies to a service as a whole, the last module depends on the actual instantiation of the service for a particular user of the Gaia-X ecosystem. It uses test-based approaches and available standard APIs offered by the cloud provider, such as OpenStack or Kubernetes APIs.

A more detailed description of each collection module and the minimum set of metrics that they need to implement in this first release is stated in Section 4.2. All collection modules in common produce technical *evidence*, which is then transferred to the evaluation interface of the GX CAM Evaluation Manager.

2.2.3 GX CAM Evaluation Manager (GX-CAM-EM)

After collecting the technical evidence by the GX-CAM-CMM, the evidence needs to be evaluated with regards to which degree it demonstrates the fulfilment of a control or requirement. For example, the technical evidence might conclude that TLS version 1.0 is used. However, a control or requirement would state that only state-of-the-art TLS versions, i.e., > 1.2, should be in use. In this case, the evaluation would

fail and an *evaluation result* representing this result would be generated. The result can then be queried, either by the dashboard or other interested and authorized parties.

2.2.4 GX CAM Dashboard

This component is used to visualize the evaluation results using a modern web application. To this end, the GX CAM Dashboard retrieves evaluation results from the GX-CAM-EM and visualizes them, making them available to authorized parties.

Note, that there is no central persistence layer demonstrated in this approach. This is intentional since each component of the service can be distributed and should work independently of the others. Thus, it is within the scope of each component to address the necessary persistence. Specific components, such as the collection modules may even choose not to persist information at all. However, common requirements for storage are described in Section 2.6.

2.3 Evidence Flow

In the Requirements Manager, concrete metrics are provided including their scale and target values (also see Section 0). For instance, a metric regarding the configured TLS version of an endpoint may define a scale of *1.0; 1.1; 1.2; 1.3* and a target value, i.e., a compliant value, of *1.3*. This allows to easily adjust target values when they change.

Collection modules have to be developed in a way that they provide evidence stating results according to the scale defined by the respective metric, allowing the Evaluation Manager to compare the measured value against the target value.

Consequently, the security requirements, that services should adhere to, are defined in a uniform way, and stored in the Requirements Manager, while their measurement is implemented in modularized components, i.e., the Collection Modules. The evaluation, i.e., the comparison of previously defined target values against the measured ones, is then performed in the Evaluation Manager.

2.4 User Classes and Characteristics

In this document, two primary user classes are considered:

- Cloud service providers using Gaia-X resources to run their services. More specifically, representatives of the service provider, e.g., responsible for security and operations. These are the primary *users* denoted in this document.
- Furthermore, we consider a second class called *administrators*. They belong to the organization running the instance of the CAM, e.g., the Gaia-X AISBL or a third-party contractor. These have additional permissions within the system, e.g., to add and remove additional modules.

2.5 Operating Environment

Please refer to the Technical Development Requirements in [TDR].

2.6 Design and Implementation Constraints

In the following, general design and implementation constrains are detailed, that need to be considered throughout the system.
Note that throughout the document, mandatory requirements use the verb *must*, while optional requirements use the verb *should*.

2.6.1 Authentication and Authorization

ID	Description	Acceptance Criteria
AUTH-C-01	All interface functions must have authentication	/
AUTH-C-02	To be compliant with standards and best-practices OAuth 2.0 in combination with JWT tokens should be used for all components and interfaces that need authentication.	Conformance testing of interfaces to the OAuth 2.0 standards, Integration Tests
AUTH-C-03	The components must avoid authentication configurations that are considered to be deprecated, e.g., OAuth 2.0 implicit flow.	Conformance testing of interfaces to the OAuth 2.0 standards
AUTH-C-04	JWT tokens must have sensible security defaults, e.g., with regards to expiration date	Tokens older than 24h must be rejected

Table 1: *Design and Implementation Constraints Authentication and Authorization*

2.6.2 Interfaces

ID	Description	Acceptance Criteria
INTFC-C-01	While in general, the components must follow the Gaia-X approach for REST-based APIs, REST may not suffice for certain scenarios when transmitting data within the service. Therefore, when needed (see INTFC-C-02 and INTFC-C-03), other RPC protocols, such as gRPC[2], must be	Documentation with criteria why a certain RPC standard was chosen

[2] https://grpc.io

	chosen for transmitting data between sub-components within the CAM.	
INTFC-C-02	Functionalities within an interface that relate to triggers and events cannot be sufficiently represented in REST, whereas an RPC call would allow for easy interaction and also subscribing to certain messages. Therefore, those functionalities must be implemented in the RPC mechanism chosen in INTFC-C-01.	Clear documentation which interface definitions are intentionally not REST-based
INTFC-C-03	Functionalities within an interface that relate to streaming of data cannot be efficiently represented in REST. Therefore, those functionalities must be implemented in the RPC mechanism chosen in INTFC-C-01.	Clear documentation which interface definitions are intentionally not REST-based
INTFC-C-04	The choice of a non-REST interface should at least provide capabilities to convert or expose them as REST interfaces. For example, through projects such as the gRPC Gateway[3], certain gRPC interfaces can easily be exposed through a REST API, for those interfaces, where it makes sense, i.e., the dashboard.	/

Table 2: *Design and Implementation Constraints Interfaces*

[3] https://github.com/grpc-ecosystem/grpc-gateway

2.6.3 Programming Languages and Technology

ID	Description	Acceptance Criteria
TECH-C-01	The chosen programming language must be a language familiar and often used in the Cloud and web context. All manager components should be written in the same language and framework, unless there is specific reason to diverge from this, e.g., within the individual collection modules or when they are based on pre-existing work. Because of the requirements specific to the databases and interfaces, suitable candidates are Java, Kotlin or Golang. Please refer to [TDR] for more information.	Documentation on why a language was chosen, as well as why certain modules deviate from a common programming language
TECH-C-02	A common database layer should be used across the different services to ensure easier maintainability. An appropriate solution could be the use of a database abstraction layer such as Hibernate[4] (for JVM-based components) or gorm[5] (for Golang-based components).	Documentation on which database layer was selected.

Table 3: *Design and Implementation Constraints Programming Languages and Technology*

2.6.4 Metrics and Controls

The collection modules must gather certain kinds of evidence, e.g., from APIs or endpoints provided by a service or from public registries. In the following, metrics are described including their scale and target values.

This way, the collection modules can be developed against the required metrics and provide evidence that can easily be evaluated by the GX Evaluation Manager (see also Section 2.3).

ID	Description	Acceptance Criteria
MC-C-01	Controls must be persisted using the properties specified by the OSCAL model (see the OSCAL documentation[6])	Documentation
MC-C-02	Metrics must be persisted in the format specified in this document (see below) and be transmitted in the format specified in 3.3.1.	Documentation

[4] https://hibernate.org

[5] https://gorm.io

[6] https://pages.nist.gov/OSCAL/documentation/schema/catalog-layer/catalog/#catalog-organization

Table 4: *Design and Implementation Constraints Metrics and Controls*

Selected Metrics for Implementation

The collection modules specified in Section 4.2 need to at least support the metrics, described in the following tables.

Name	**SystemComponentsIntegrity**
Description	This metric is used to assess whether the deployed system components were started without modifications.
Source	EUCS
Domain	A7: Operational Security
Control	OPS-21: Managing Vulnerabilities, Malfunctions and Errors - System Hardening (especially OPS-21.3)
Scale	Boolean
Target Value	True
Interval	1 day
Target	Gaia-X services that aim for EUCS assurance level “high”

Table 5: *Metric "SystemComponentsIntegrity"*

Name	**CyberSecurityCertification**
Description	This metric is used to assess if the service provider or service holds a valid cyber security certification, such as ISO 27001.
Source	TBD
Domain	TBD
Control	TBD
Scale	[valid; invalid; not available]
Target Value	valid
Interval	1 week
Target	Certification as provided in the Gaia-X Federated Catalogue

Table 6: *Metric "CyberSecurityCertification"*

Name	**OAuthGrantTypes**
Description	This metric is used to assess that no deprecated grant types are used in an OAuth 2.0 configuration.
Source	EUCS
Domain	A8: Identity, Authentication and Access Control Management
Control	IAM-07: Authentication Mechanisms
Scale	[authorization_code, implicit, password, client_credentials, device_code, refresh_token]
Target Value	NOT [password, implicit]
Interval	5 minutes
Target	Resources that offer single sign on authentication

Table 7: *Metric "OAuthGrantTypes"*

Name	**TlsVersion**
Description	This metric is used to assess if up-to-date encryption protocols are used for traffic served from public networks.
Source	EUCS
Domain	A9: Cryptography and Key Management
Control	CKM-02: Encryption of Data in Transit
Scale	[1.0; 1.1; 1.2; 1.3]
Target Value	1.3
Interval	5 minutes
Target	Resources that accept traffic from public networks

Table 8: *Metric "TlsVersion"*

Name	**TlsCipherSuite**
Description	This metric is used to assess if strong encryption mechanisms are used for traffic served from public networks.
Source	EUCS
Domain	A9: Cryptography and Key Management
Control	CKM-02: Encryption of Data in Transit
Scale	String specifying encryption algorithm
Target Value	[TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, TLS_CHACHA20_POLY1305_SHA256]
Interval	5 minutes
Target	Resources that accept traffic from public networks

Table 9: *Metric "TlsCipherSuite"*

Name	**AtRestEncryption**
Description	This metric is used to assess that object storages, i.e., blob storages, are encrypted at rest.
Source	EUCS
Domain	A9: Cryptography and Key Management
Control	CKM-03: Encryption of Data at Rest
Scale	[disabled; enabled]
Target Value	enabled
Interval	1 hour
Target	Resources that offer storage

Table 10: *Metric "AtRestEncryption"*

The mapping between collection modules, metrics and controls is shown in Table 1111.

	EUCS OPS-21	**TBD**	**EUCS IAM-07**	**EUCS CKM-02**		**EUCS CKM-03**
	SystemComponentsIntegrity	**CyberSecCert**	**OAuthGrantTypes**	**TlsVersion**	**TlsCipherSuite**	**AtRestEncryption**
Public Registry CM		X				
Communication Security CM				X	X	
Authentication Security CM			X			
Remote Integrity CM	X					
Workload Security CM						X

Table 11: *The mapping between collection modules, metrics and controls*

2.7 User Documentation

The following documentation MUST be provided:

- **Deployment Manual:** The contractor must provide documentation describing deployment procedures for GX- CAM, including roll-out and roll-back.
- **Operations Manual:** The contractor must provide documentation describing all modes of operation including error handling, instructions for secure operation, and means of recovery.
- **Software Architecture:** The contractor must provide an overview of GX- CAM's software architecture, including a component view, a process view, and implementation considerations.
- **Security Concept:** The contractor must provide a security concept for the secure operation of GX-CAM according to the defined security requirements.

Further requirements regarding the documentation can be found in [TDR].

3. Interface Requirements

3.1 User Interfaces

The GX-CAM service offers two user interfaces: one interface for configuring the monitoring, and one interface that visualizes the evaluation results.

3.1.1 Configuration interface

In the configuration interface, users select the services they want to monitor, and the controls they want to apply, for instance a user may select a storage service that she uses, and select a control regarding at-rest encryption to be monitored on that service.

ID	Description	Acceptance Criteria
CI-F-01	The configuration user interface must present the user with the following configuration options: • *Monitorable Services*, i.e., services that can be monitored by the user. • Controls that can be monitored per monitorable service.	Documentation, Test Cases
CI-F-02	The configuration user interface must offer the user the possibility to configure the collection modules where applicable, e.g., when a collection module requires a credential for accessing an API	Documentation, Test Cases

Table 12: *Functional requirements of the Configuration user interface*

3.1.2 Dashboard interface

The Dashboard interface visualizes evaluation results for the user. Note: Specific requirements for the design of the dashboard component can be found in Chapter 0 (System features for GX CAM Dashboard).

ID	Description	Acceptance Criteria
DI-F-01	The Dashboard user interface must present the user with the following information: • Compliance or non-compliance per selected control per selected service	Documentation, Test Cases
DI-F-02	The Dashboard user interface must allow the user to visualize the information specified in [DI-F-01] over time in a time frame of at least 30 days.	Documentation, Test Cases

Table 13: *Functional requirements of the Dashboard user interface*

3.2 Hardware Interfaces

Not applicable.

3.3 Software and Communication Interfaces

In the following, the necessary software and communication interfaces are described on a level of functions and object properties. Both can either use basic types, such as Integers, Strings or Duration, as well as references to other Object types (defined in 3.3.1), marked in bold, e.g., **GX.CAM.Evidence**.

3.3.1 Object Representation

When transmitted via an interface, the following object representations and naming conventions should be used.

GX.CAM.Control
The model of a control should follow the control definition of the OSCAL[7] standard.

GX.CAM.Metric

Property	**Type**	**Special Remarks**
ID	Int	
Name	String	
Description	String	
Control	Identifier	*Identifier according to OSCAL model*
Scale	String	
Target Value	Any	*Depends on the scale*
Interval	Duration	

Table 14: *GX.CAM.Metric*

GX.CAM.Evidence

Property	**Type**	**Special Remarks**
ID	Int	
Name	String	
TargetService	Service (or identifier)	
TargetResource	String	*Optional, specific to the service. Can be a resource identifier within the service*
GatheredUsing	**GX.CAM.Metric** (or identifier)	
GatheredAt	Time	

[7] https://github.com/usnistgov/OSCAL

Value	Any	*Depends on the type of evidence*
RawEvidence	String or Binary	*Optional, could be a JSON representation of the raw underlying evidence*

Table 15: *GX.CAM.Evidence*

GX.CAM.Evaluation

Property	**Type**	**Special Remarks**
ID	Int	
Metric	**GX.CAM.Metric** (or identifier)	*The metric that was used in gathering*
Evidence	**GX.CAM.Evidence** (or identifier)	
Status	Boolean	*Whether the evaluation was successful or not*
Time	Time	

Table 16: *GX.CAM.Evaluation*

GX.CAM.Compliance

Property	**Type**	**Special Remarks**
ID	Int	
ControlID	String	*The control to check for compliance, reference to OSCAL model.*
Evaluations	List of **GX.CAM.Evaluation** (or identifier)	*A list of references to evaluations of metrics which are associated to this control.*
Status	Boolean	*Whether the compliance status is successful or not*
Time	Time	

Table 17: *GX.CAM.Compliance*

GX.CAM.CollectionModule

Property	**Type**	**Special Remarks**
ID	Int	*A unique ID*
Name	String	*The name of the module*
Description	String	*A description*
Metric	**GX.CAM.Metric** (or Identifier)	*The metric according to which this module is gathering evidence*

Table 18: *GX.CAM.CollectionModule*

GX.CAM.ServiceConfiguration

Property	Type	Special Remarks
RawConfiguration	String	*A service specific configuration, such as specific access tokens for the provisioned service. It should be a JSON serialized format of the specific configuration. This can be used by the collection module to retrieve additional information.*

Table 19: *GX.CAM.ServiceConfiguration*

3.3.2 GX CAM Configuration Interface

Function Name	Parameters	Return Type	Description
StartMonitoring	service_id (url) control_ids (list)	-	*Needs to be exposed as a REST-API endpoint to users.*
StopMonitoring	service_id (url)	-	*Needs to be exposed as a REST-API endpoint to users.*
GetMonitoringStatus	service_id (url)	list of **GX.CAM.Control** (or its ids)	*Needs to be exposed as a REST-API endpoint to users.*
ListMetrics	-	list of **GX.CAM.Metric**	
GetMetric	metric_id	**GX.CAM.Metric**	
ListCollectionModules		list of **GX.CAM.CollectionModule**	
FindCollectionModule	metric_id	**GX.CAM.CollectionModule**	
AddCollectionModule	**GX.CAM.CollectionModule**		*Administrators only*
RemoveCollectionModule	module_id (int)		*Administrators only*

Table 20: *GX CAM Configuration Interface*

3.3.3 GX CAM Evaluation Interface

Function Name	Parameters	Return Type	Description
SendEvidences	evidence_stream (stream of **GX.CAM.Evidence**)		*Client-side streaming of evidence from the client (a specific collection module) to the server (evaluation manager)*
GetEvaluation	service_id (url) metric_id (int)	**GX.CAM.Evaluation**	*Retrieves the latest evaluation result for a particular metric. Needs to be exposed through a REST-API*
StreamEvaluations	service_id (url)	stream of **GX.CAM. Evaluation**	*Starts a continuous stream of evaluation results for a particular service*
GetCompliance	service_id (url) control_id (int)	**GX.CAM.Compliance**	*Retrieves the current compliance result for a particular control of a service. Needs to be exposed through a REST-API*
ListCompliance	service_id (url)	list of **GX.CAM.Compliance**	*Retrieves all current compliance results for a particular service. Needs to be exposed through a REST-API*

Table 21: *GX CAM Evaluation Interface*

3.3.4 GX CAM Collection Modules Interface

Function Name	Parameters	Return Type	Description
StartCollecting	service_id (url) metric_id (int) configuration (**GX.CAM.ServiceConfi**	Identifier	*Triggers the collection module manager to start the collection of evidence, optionally including a service-*

	guration) eval_manager (url)		*specific configuration, if the collection module needs it. The collection manager will forward the request to the collection module, which will then stream evidence to the evaluation manager specified*
StopCollecting	collection_id (int)	-	*Stops the collection*

Table 22: *GX CAM Collection Modules Interface*

4. System Features

Next to the requirements stated in this document, the requirements regarding the Technical Environment/ Development [TDR] must be also met.

4.1 GX Requirements Manager

4.1.1 Description and Priority

The Requirements Manager (GX-CAM-RM) manages the selection of Gaia-X requirements. To that end, it persists a mapping of collection modules to the metrics and controls they can measure. It offers an interface to the user to select security controls to be monitored for a particular set of services, that the user has permissions to.

4.1.2 Stimulus/Response Sequences

4.1.2.1 Monitoring Start

1. User accesses Requirements Manager via interface, either directly via API or through a user interface, according to Section 0.
2. User selects controls to be monitored for a particular service, identified by a unique URL (referring to the federated catalogue), starting the monitoring process.
3. System saves selection and triggers the Collection Module Identification and Trigger (CMIT) functionality
4. CMIT does
    a. lookup metrics that are associated to the fulfillment of the set of selected controls.

b. lookup collection modules that are suited to measure the particular metrics.

5. Lastly, the system invokes the *StartCollecting* function of the Collection Manager interface (see Section 3.3.4) with the appropriate parameters, handing over the workflow to the Collection Module Manager.
6. Optional: Some Collection Modules might need additional service-specific configuration, such as user credentials. In this case, additional flows to the user might be necessary to exchange these.

4.1.2.2 Collection Module Registration

1. Administrator accesses requirements manager via interface
2. System presents overview of registered collection modules (CMs)
3. Developer adds/modifies/deletes a CM registration.
4. Change is persisted.

4.1.3 Functional Requirements

ID	Description	Acceptance Criteria
RM-F-01	The mapping of user-selected controls and services must be persisted in a database.	Common database technology used (see [*TECH-C-02]*); documentation; test cases.
RM-F-02	The external-facing interface functionalities (see 0) of the RM are implemented as a web service / REST service.	Documentation; test cases showing use of REST API using JSON.
RM-F-03	The configuration interface must provide the set of monitorable controls to authenticated users.	Documentation; test cases; interface must return the set of controls per user.
RM-F-04	The user must be able to select a set of controls per service to be monitored continuously.	Documentation; test cases
RM-F-05	Where necessary, the user must be able to configure the monitoring of a certain control, e.g., providing a credential for the access to a certain API.	Documentation; test cases
RM-F-06	The configuration interface must provide an (internal) endpoint for the Evaluation Manager to retrieve the selected controls per service for a user.	Documentation; test cases
RM-F-07	The RM must identify registered CMs that are needed to monitor the selected controls in the selected services.	Documentation; test cases
RM-F-08	The RM must trigger the CMs identified (see [RM-F-06]) via the CM Manager Interface (see 3.3.4).	Documentation; test cases
RM-F-09	The Configuration Interface must provide a functionality to register, deregister, and modify CMs and the controls and services they can monitor, and persist this mapping in a database.	Common database technology used (see [*TECH-C-02]*); documentation; test cases.

RM-F-10	The Configuration Interface should provide an API that allows to manually register, deregister, and modify CMs and the controls and services they can monitor, and persist this mapping in a database.	Common database technology used (see [*TECH-C-02]*); documentation; test cases.
RM-F-11	The RM must preserve a mapping of available controls to associated metrics, and a mapping of metrics and collection modules which implement the respective metrics.	Documentation; test cases

Table 23: *Functional Requirements for the GX Requirements Manager component*

4.2 GX Collection Module Manager

4.2.1 Description and Priority

The GX Evidence Collection Module Manager (CMM) is a central component that is responsible for the measurement of security metrics, i.e., for gathering evidence. It employs a modular structure that allows to deploy various types of collection modules for different purposes, such as testing APIs, retrieving configuration information from a service, and retrieving security-relevant information from public registries.

The CMM can collect evidence based on different publicly available sources:

- (Geo)-Location of Services, TLS configuration, network configuration, OAuth configuration
- Information in public registries, such as the Gaia-X federated catalogue.

The CMM can also collect evidence based on internal sources of a service, ideally in a standardized format / via open APIs:

- Configuration data, e.g., from Kubernetes, OpenStack
- Log data, vulnerability scans (e.g., in STIX 2.0 format)
- Remote integrity checks

4.2.2 Stimulus/Response Sequences

4.2.2.1 Starting the Measurement of a Metric

1. Requirements Manager triggers the measurement of a metric via the CMM interface.
2. The CMM triggers the corresponding collection module.

4.2.2.2 Stopping the Measurement of a Metric

1. Requirements Manager triggers the deactivation of a metric via the CMM interface.
2. The CMM triggers the deactivation corresponding collection module.

4.2.3 Functional Requirements

Requirements for the CM Manager (GX-CAM-CMM) and general requirements for the collection modules are listed in the following:

ID	Description	Acceptance Criteria
CMM-F-01	The CMM must offer an API for the Requirements Manager to start/stop the execution of collection modules, according to Section 3.3.4.	Documentation, Test Cases
CMM-F-02	The CMM must manage, i.e., start or stop, the execution of a CM according to the trigger induced by the Requirements Manager.	Documentation, Test Cases
CMM-F-03	Any CM must collect evidence in high-frequency intervals as specified in the respective metric (see Table 111). They must stream it to the evaluation manager using the function *SendEvidences* (see Section 0).	Documentation, Test Cases
CMM-F-04	CMs must function (e.g., retrieve and store evidence, scale up and down) independently from each other.	Documentation, Test Cases
CMM-F-05	The gathered measurements must be transformed into a unified evidence format (see Section 3.3.1).	Documentation, Test Cases

Table 24: *Functional Requirements for the Collection Module Manager component*

In the following, the single collection modules are described with their individual requirements.

4.2.3.1 Public Registry Collection Module

The Public Registry Collection Module (PRCM) is a module that retrieves information from public registries, such as available certifications as provided in the Gaia-X Federated Catalogue, i.e., in the self-description.

ID	Description	Acceptance Criteria
PRCM-F-01	The PRCM must be able to retrieve evidence from public registries via standard APIs.	Documentation, Test Cases
PRCM-F-02	The PRCM must be able to retrieve certification information stored in the Gaia-X federated Catalogue.	Documentation, Test Cases

Table 25: *Functional Requirements for the Public Registry Collection Module*

4.2.3.2 Communication Security Test Collection Module

The core purpose of this module is to gather evidence of the service related to communication security. In the first iteration, this is focused on TLS security offered by endpoints of the service. If possible, existing open-source tools should be re-used to interact with the TLS endpoint to gather the necessary information specified in the following.

ID	Description	Acceptance Criteria

ComSec-F-01	The CCSM must be able to collect the following metrics from TLS endpoints exposed by Gaia-X services: • TLS version • TLS cipher suites • Certificate path validity	Documentation, Test Cases
ComSec-F-02	The CSCM should be able to detect common weaknesses in a TLS configuration.	Documentation, Test Cases
ComSec-NF-01	Existing and license-compatible open-source tools should be re-used.	Documentation, Test Cases

Table 26: *Functional Requirements for the Communication Security Test Collection Module*

4.2.3.3 Authentication Security Test Collection Module

The purpose of the Authentication Security Test Collection Module is to apply authentication metrics to exposed APIs via active testing.

ID	Description	Acceptance Criteria
AuthSec-F-01	The AuthSecCM must be able to collect authentication information of APIs exposed by Gaia-X services.	Documentation, Test Cases
AuthSec-F-02	The AuthSecCM must be able to at least measure the following metrics regarding an OpenID configuration: • Supported grant types • Supported encryption algorithms • Supported signing algorithms	Documentation; Test Cases

Table 27: *Functional Requirements for the Authentication Security Test Collection Module*

4.2.3.4 Remote Integrity Collection Module

The purpose of the Remote Integrity Collection Module is the assessment of the utilized software stack.

ID	Description	Acceptance Criteria
RemInt-F-01	The RemIntCM must be able to collect the expected values for the integrity of system components based on information provided in the GX.CAM.ServiceConfiguration.	Documentation, Test Cases
RemInt-F-02	The RemIntCM must be able to collect information regarding the integrity of system component from the service instances.	Documentation; Test Cases
RemInt-F-03	The RemIntCM must support the verification of provided integrity proofs against the expected values for the system components (Remote Attestation).	Documentation; Test Cases

RemInt-F-04	Existing open-source implementations (under Apache 2.0 license) shall be used.	Documentation

Table 28: *Functional Requirements for the Remote Integrity Collection Module*

4.2.3.5 Workload Configuration Collection Module

The purpose of the Workload Configuration Collection Module is to use service-provided APIs to retrieve configuration information about deployed resources, such as encryption and authentication settings.

ID	Description	Acceptance Criteria
WCCM-F-01	The WCCM must be able to collect configuration profiles from resources via Kubernetes APIs.	Documentation, Test Cases
WCCM-F-02	The WCCM should be able to collect configuration profiles from resources via OpenStack APIs.	Documentation, Test Cases
WCCM-F-03	The WCCM must be able to collect configuration information about at least the following metrics: • Encryption-at-rest settings where applicable, e.g., encryption algorithms used. • Encryption-in-transit settings where applicable, e.g., allowed transmission protocols. • Authentication settings where applicable, e.g., role management • Access control settings where applicable, e.g., inbound/outbound traffic restrictions	Documentation; Test Cases
WCCM-F-04	The WCCM should reuse existing open-source tools, such as Clouditor[8].	Documentation; Test Cases

Table 29: *Functional Requirements for the Workload Configuration Collection Module*

4.3 GX Evaluation Manager

4.3.1 Description and Priority

The GX Evaluation Manager (EM) is a central component that stores evidence, evaluates them according to the controls selected by the user, updates the compliance status, and stores the evaluation and compliance results in a database.

4.3.2 Stimulus/Response Sequences

[8] https://github.com/clouditor/clouditor

4.3.2.1 Update Evaluation and Compliance Status

The EM may either evaluate results in predefined intervals, or based on a trigger, i.e., incoming streamed evidence. In this sequence, the EM does not interact with any user.

1. As a basis for the evaluation, the EM retrieves the user-selected controls for the particular service from the Requirements Manager (RM) using the interface specified in 0
2. First, all metrics associated to each selected control are fetched from the RM.
3. Gathered evidence for the combination of the selected metric and service ID are queried from the evidence store – or filtered directly from the stream of evidence.
4. Each evidence is evaluated against the target value of the metric using a Boolean expression and an evaluation object with the result is created.
5. All evaluation object results associated to a control are then combined using a Boolean AND expression and a compliance object result is created.
6. The EM then stores the evaluation and compliance results in a database.

4.3.2.2 User Interaction for Visualization

1. User login to dashboard
2. The dashboard uses the REST-API endpoints of the interface specified in 0 to retrieve the evaluation and compliance results from the EM

4.3.3 Functional Requirements

ID	Description	Acceptance Criteria
EM-F-01	The EM must retrieve the user-selected mapping of controls and services from the RM.	Documentation; test cases
EM-F-02	The EM must evaluate evidence against the mapping of controls and services in high-frequency intervals of few minutes.	Documentation; test cases
EM-F-03	The EM must store evaluation results in a database.	Common database technology used (see [*TECH-C-02]*); documentation; test cases.
EM-F-04	The EM must represent evaluation results for at least the following cases: • Complete compliance with a control • Non-compliance with a control • Conflicting evidence • Missing evidence • No recent evidence available	Documentation; test cases
EM-F-05	The EM should represent abstract KPIs about the evaluation, such as: • Percentage of selected controls that are compliant or non-compliant.	Documentation; test cases

	• Percentage of selected controls that have been compliant or non-compliant over a user-selected time frame.	

Table 30: *Functional Requirements for the Evaluation Manager component*

4.4 GX Dashboard

4.4.1 Description and Priority

The GX Dashboard is a central component that visualizes evaluation results and makes them available to users.

4.4.2 Stimulus/Response Sequences

The Dashboard retrieves evaluation results from the Evaluation Manager via its provided interface, either periodically or triggered by a user accessing the dashboard.

4.4.3 Functional Requirements

ID	Description	Acceptance Criteria
DA-F-01	The Dashboard must retrieve the evaluation results from the provided endpoint of the Evaluation Manager. See GetEvaluation call in 0	Documentation; test cases
DA-F-02	The Dashboard must visualize at least the following information to the user: - Compliance status per control per service - Compliance status per service The necessary API calls are described in 0 (GetCompliance / ListCompliance)	Documentation; test cases; the visualization follows a commonly used format, such as a tree diagram
DA-F-03	The Dashboard should represent KPIs about the evaluation results, such as: - Percentage of selected controls that are compliant or non-compliant - Percentage of selected controls that have been compliant or non-compliant over a user-selected time frame Basis for the KPIs are the selected controls retrieved by GetMonitoringStatus (see 0) as well as the individual evaluation results GetEvaluation call in 0)	Documentation; test cases; visualization in a pie chart or time series graph

Table 31: *Functional Requirements for the Dashboard component*

5. Other Nonfunctional Requirements

5.1 Performance Requirements

ID	Description	Acceptance Criteria
PE-NF-01	GX-CAM components that monitor, evaluate, or access evaluation results must be able to do so in high-frequency intervals as specified in the respective metrics.	Documentation; test cases

Table 32: *Performance requirements for the GX-CAM components*

5.2 Safety Requirements

Not applicable.

5.3 Security Requirements

ID	Description	Acceptance Criteria
SEC-NF-01	Collected evidence must be stored in encrypted form using state-of-the-art encryption algorithms.	Documentation; test cases
SEC-NF-02	The access to data stored by the GX-CAM components, e.g., evidence and user data, must be restricted according to the least-privilege principle.	Documentation; test cases
SEC-NF-03	The transmission of evidence between GX-CAM components, as well as between GX-CAM components and external components, such as the target services, must be encrypted and authenticated using state-of-the-art protocols and algorithms.	Documentation; test cases
SEC-NF-04	Each Gaia-X Federation Service MUST meet the requirements stated in the document “Specification of non-functional Requirements Security and Privacy by Design” [NF.SPBD].	Documentation; test cases

Table 33: *Security requirements for the GX-CAM components*

5.4 Software Quality Attributes

ID	Description	Acceptance Criteria
SQ-NF-01	Collection Modules must be extensible to allow for the measurement of new metrics.	Documentation; test cases
SQ-NF-02	Databases, especially for stored evidence, must be highly available.	Documentation; test cases
SQ-NF-03	User interfaces must be easily usable.	Documentation; test cases

SQ-NF-04	Components must be fault tolerant such that the temporary unavailability of components does not lead to a non-compliant evaluation result.	Documentation; test cases
SQ-NF-05	Components must be scalable with the number of users, and the amount of services that are monitored.	Documentation; test cases

Table 34: *Software Quality Attributes for the GX-CAM components*

Appendix A: Glossary

The glossary is part of the Gaia-X Architecture Document [TAD].

Appendix B: Overview GXFS Work Packages

The project "Gaia-X Federation Services" (GXFS) is an initiative funded by the German Federal Ministry of Economic Affairs and Energy (BMWi) to develop the first set of Gaia-X Federation Services, which form the technical basis for the operational implementation of Gaia-X.

The project is structured in five Working Groups, focusing on different functional areas as follows:

Work Package 1 (WP1): Identity & Trust
Identity &Trust covers authentication and authorization, credential management, decentral Identity management as well as the verification of analogue credentials.

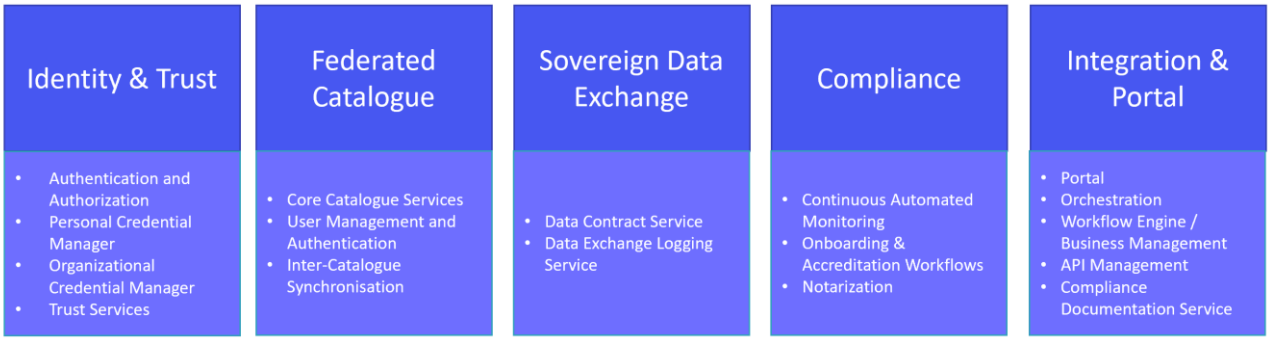
Work Package 2 (WP2): Federated Catalogue
The Federated Catalogue constitutes the central repository for Gaia-X Self-Descriptions to enable the discovery and selection of Providers and their Service Offerings. The Self-Description as expression of properties and Claims of Participants and Assets represents a key element for transparency and trust in Gaia-X.

Work Package 3 (WP3): Sovereign Data Exchange
Data Sovereignty Services enable the sovereign data exchange of Participants by providing a Data Agreement Service and a Data Logging Service to enable the enforcement of Policies. Further, usage constraints for data exchange can be expressed by Provider Policies as part of the Self-Description

Work Package 4 (WP4): Compliance
Compliance includes mechanisms to ensure a Participant's adherence to the Policy Rules in areas such as security, privacy transparency and interoperability during onboarding and service delivery.

Work Package 5 (WP5): Portal & Integration
Gaia-X Portals and API will support onboarding and Accreditation of Participants, demonstrate service discovery, orchestration and provisioning of sample services.

All together the deliverables of the first GXFS project phase are specifications for 17 lots, that are being awarded in EU-wide tenders:



Further general information on the Federation Services can be found in [TAD].