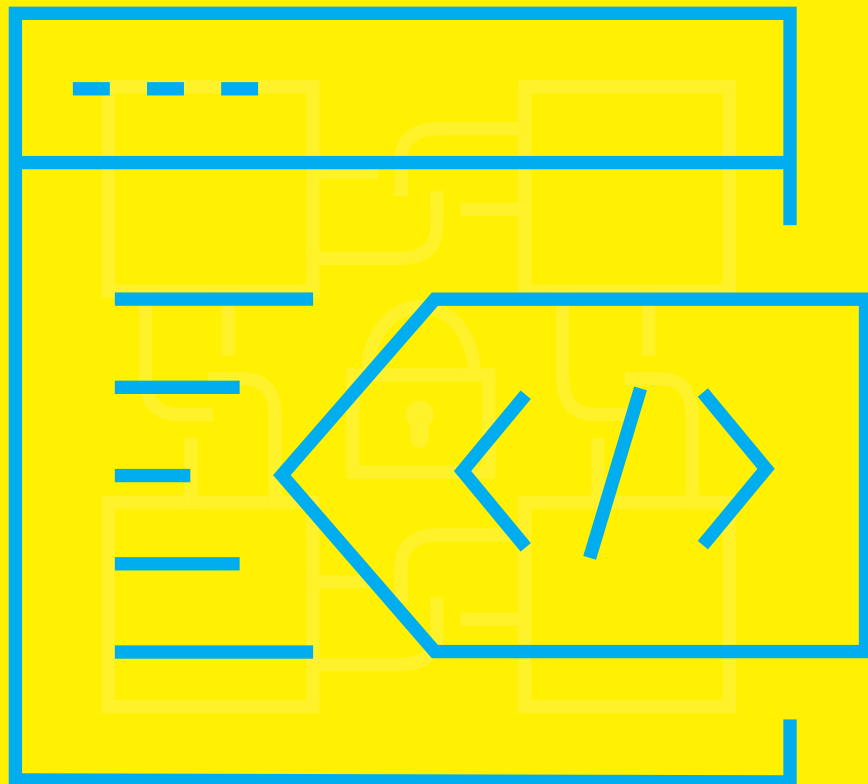




White Paper | Version 1.0 | May 2020

Specification: IDS Meta Data Broker



- Position Paper of members of the IDS Association
- Position Paper of bodies of the IDS Association
- Position Paper of the IDS Association
- White Paper of the IDS Association

Publisher

International Data Spaces
Association Anna-Louisa-Karsch-Str. 2
10178 Berlin
Germany

Editor

Sebastian Steinbuss,
International Data Spaces Association

Corresponding Author

Sebastian Bader, Fraunhofer IAIS

Authors & Contributors

Sebastian Bader, Fraunhofer IAIS
Fabian Bruckner, Fraunhofer ISST
Gernot Böge, FIWARE Foundation
Dennis Oliver Kubitzka, Fraunhofer IAIS
Jörg Langkau, nicos AG
Dileep Murthy, Fraunhofer IAIS
Ralf Nagel, Fraunhofer ISST

Copyright

International Data Spaces Association,
Dortmund, Germany, 2020

**Cite as**

Steibuss S., Bader S., et al. (2020): White Paper Specification. IDS Meta Data Broker. International Data Spaces Association. <https://doi.org/10.5281/zenodo.5675076>

Table of Contents

1	Introduction.....	4
1.1	Scope of the Document	4
1.2	Related Documents.....	4
1.3	Notational Conventions	4
2	General overview.....	5
3	IDS Meta Data Broker Requirements.....	7
3.1	Role of an IDS Meta Data Broker	7
3.2	Connector Requirements.....	9
3.3	Functional Requirements.....	11
3.4	Message Requirements.....	14
3.5	Behavioral Requirements	16
3.6	Business Requirements	17
3.7	Informational Requirements	18
3.8	Interface Requirements.....	19
3.9	Conditional Requirements.....	20
4	IDS Meta Data Broker Profiles	23
4.1	Advanced Information Profile	23
4.2	Usage Control Profile.....	25
5	Functions and Correlated Messages.....	26
5.1	Messages received by a Broker	26
5.2	Messages send by a Broker as Response.....	29
6	Annex.....	31
6.1	HTTP API.....	31
6.2	Removed Requirements	32
	Our Members.....	34
	Overview Publications	36

1 Introduction

1.1 Scope of the Document

This document describes the minimal features of an IDS Meta Data Broker, as an index-service running in conjunction with an IDS Connector. The description of the IDS Connector itself is not part of this document. This document does not describe data brokerage functionality for the International Data Spaces, but Meta Data Brokerage.

This document contains the specification of the IDS Meta Data Broker and acts as a foundation for the Certification criteria for the IDS Meta Data Broker. In addition to the minimal requirements, also two advanced Broker Profiles are described, enhancing the standard Broker functionalities by improved information management and usage policies.

1.2 Related Documents

The following public documents¹ are related to this document and should be considered as important:

- IDS-RAM 3.0
- IDS Certification Scheme 2.0
- IDS Certification Criteria

The following internal documents are related to this document:

- IDS Communication Guide
- IDS Handshake

1.3 Notational Conventions

B <XXX>	New Numbering introduced in Version 2.0
(B<XX>)	Deprecated Numbering for Requirements of Version 1.0 in Brackets
(X <XX>)	Reference Numbering for Requirements obtained from other Documents
<i>Italic</i>	Italic Text in italics represents source code or messages
Bold	Not used in this document

¹ The documents can be found at: <https://www.internationaldataspaces.org/ressource-hub/publications-ids/>

2 General overview

The International Data Spaces Reference Architecture Model (version 3.0) describes several components:

- International Data Spaces Connector
- International Data Spaces App Store
- International Data Spaces Meta Data Broker
- International Data Spaces Clearing House
- International Data Spaces Identity Provider

The IDS Meta Data Broker is a service for publishing and searching metadata of Connectors and resources between International Data Spaces Participants. In order to ensure the necessary interoperability and general interactions, an IDS Meta Data Broker (like the App Store) is also defined as a specialized IDS Connector. The communication between an IDS Connector and an IDS Meta Data Broker is based on the same principles as any other Connector-Connector communication within the International Data Spaces. Still, an IDS Meta Data Broker provides a collection of additional functionalities:

- Indexing services in order to effectively and efficiently respond to queries and present known Connectors and other resources
- Interfaces for Users or IDS-Messages to ensure access to the stored information.

This document serves as a specification defining what an IDS Meta Data Broker is, which functionalities it must support and which services other IDS components can expect. The described content is aligned with the explanations of the IDS-RAM 3.0 and complements the "The IDS Communication Guide"².

A software component is allowed to describe itself as an IDS Connector only after an IDS compliant Certification Body verified that the necessary capabilities are in place and operate in IDS conformant manners. The list of certification criteria (Certification Criteria - Components v2.1.0³) contains the respective requirements. The same procedure is valid for an IDS Meta Data Broker, with additional requirements for broker-functionalities as documented in this list. After a successful certification of the relevant Criteria of an IDS Connector and as an IDS Meta Data Broker, a software component can include the following RDF statement into its Self-Description:

```
<Connector/Broker URI>  
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
<https://w3id.org/idsa/core/Broker> .
```

However, no other entity is obliged to believe this claim without further proof. The IDS way for a trustworthy information provision is the Dynamic Attribute Token (DAT⁴), provided by IDS Dynamic Attributes Provisioning Service (DAPS⁵). A certified IDS Meta Data Broker can ask for a

² <https://industrialdataspace.jiveon.com/docs/DOC-2524>

³ <https://industrialdataspace.jiveon.com/docs/DOC-2811>

⁴ <https://github.com/International-Data-Spaces-Association/IDS-G/tree/master/glossary#dynamic-attribute-token-dat>

⁵ <https://github.com/International-Data-Spaces-Association/IDS-G/tree/master/glossary#dynamic-attribute-provisioning-service-daps>

DAT at its responsible DAPS. The necessary interactions are specified in the The IDS Communication Guide⁶ and The IDS Handshake⁷. Only if a correctly signed DAT from a trustworthy DAPS includes the above claim, other IDS entities should accept the claim of a potential IDS Meta Data Broker.

In accordance to the thereby directly included guaranteed features, an IDS Meta Data Broker may provide significantly more services. Several options to include broker features in an International Data Space are possible:

- only mandatory IDS Meta Data Broker features
- mandatory and profile features:
 - o any set of additional features from the “Usage Control Enabled” and “Advanced Information” profiles
 - o all additional features from the “Usage Control Enabled” or/and “Advanced Information” profiles.
- general features supporting publication, indexing and discovery which are not mentioned in this specification (not applicable as an IDS Meta Data Broker), or
- any combination of the categories above.

⁶ <https://industrialdataspace.jiveon.com/docs/DOC-2524>

⁷ <https://industrialdataspace.jiveon.com/docs/DOC-1817>

3 IDS Meta Data Broker Requirements

The following broker requirements represent the results from multiple workshops, presentations and discussions between 2016 and 2020. It contains the agreed understanding and the foundation for the upcoming developments. As such, features may be updated for future versions.

All requirements listed under the Headings: Connector, Function, Messaging, Behavior, Business, Information and Interface are normative and required for an IDS Meta Data Broker. All requirements listed under the heading Conditional Requirements have also to be met, if the specified condition holds. In addition, there are optional requirements listed under the Profiles “Advanced Information” and “Usage Control Enabled”. Any subset of these optional requirements can be included into an IDS Meta Data Broker. Finally, we provide a list of Functions and Correlated IDS Messages relevant for an IDS Meta Data Broker. This list serves just as an overview of requirements derived in The IDS Communication Guide⁸ with additional explanations.

3.1 Role of an IDS Meta Data Broker

The following statements can be counted as informal requirements and specify the role of the IDS Meta Data Broker within the IDS. As these requirements are just conceptual, they are not subject of any certification process.

B 001 (B1)	An IDS Meta Data Broker is an optional component of any International Data Space.
B 002 (B3)	An IDS Meta Data Broker provides an index service, which is a system for the publication of metadata sent by Connectors.
B 003	An IDS Meta Data Broker should allow the link-based discovery of data sources.
B 004 (B4)	The actual transfer of data, except Meta Data, between a provider and consumer Connector does not involve an IDS Meta Data Broker at any time.
B 005 (B6)	Multiple brokers may exist in International Data Spaces.
B 006 (B7)	There may be different Meta Data Broker implementations from different providers in International Data Spaces.

⁸ <https://industrialdataspace.jiveon.com/docs/DOC-1817>

B 007 (B10)	A Connector operator may independently define which (meta-)data partitions of its Connector is sent to one or multiple IDS Meta Data Brokers.
B 008	An International Data Spaces Participant may use a generally known or otherwise announced server to host metadata for IDS entities. The recommended procedure is to use an IDS Meta Data Broker for this.
B 009 (B13)	The communication between a Connector and An IDS Meta Data Broker is message oriented. There are two categories of broker messages: <ul style="list-style-type: none"> • Publishing Messages (delivery of Meta Data to the index services) • Query Messages (query of Meta Data from the index service)
B 010	Persistent storage can be achieved using triple stores or by any other suitable storage back ends.
B 011 (B24)	The persistence of an index service is not predefined. Here is a list of options: <ul style="list-style-type: none"> • File system • NoSQL • RDBMS • LDAP/ActiveDirectory • RDF
B 012	Passivating the Meta Data of Connectors and Resources is preferred over deleting. In case Connectors or Resources reappear, An IDS Meta Data Broker should be able to recognize their previous state and reactivate the entries.
B 013	The recommended RDF serialization for any metadata is JSON-LD.
B 014	An IDS Meta Data Broker should accept registration attempts from a valid IDS Connector.
B 015 (B28)	An IDS Meta Data Broker may parallelize the processing of Broker Messages, as long as this has no impact on query results.

3.2 Connector Requirements

The requirements listed here are required for the certification as a connector, like specified by Certification Criteria - Components v2.1.0. Criteria that are already covered by stronger requirements for the broker are not listed.

B 016 (B12)	The communication between a Connector and an IDS Meta Data Broker relies on the general IDS communication between two Connectors as specified by The IDS Communication Guide and The IDS Handshake2
B 017	An IDS Meta Data Broker must provide a Self-Description document at <scheme>://<authority>[:port][path/to/broker]/ according to the IDS Information Model.
B 018 (B27)	An IDS Meta Data Broker has to protocol all executed actions or queries. This log is used for error and problem report and is not public. Log request must be addressed to the admin and management team of the broker. Retention time of the log is specified by the International Data Spaces.
B 019 (COM 03)	Encryption and integrity protection is facilitated by means of mechanisms considered state of the art by BSI TR 02102-1, NIST SP 800-175b, or an equivalent crypto catalogue.
B 020	An IDS Meta Data Broker has a valid IDS Identity, an according X509 identity certificate and provides a valid DAT with each interaction.
B 021 (OS 01)	An IDS Meta Data Broker supports installation and execution of containers.
B 022 (OS 06)	System data backups as well as backups of data transferred are always encrypted before being stored outside system.
B 023 (APS 01)	A Broker supports only apps possessing a valid signature. This signature is the signed check sum of the software artefact, which was created by means of a private key of the app publisher.
B 024 (APS 02)	A Broker verifies signature after app was downloaded (from IDS-AppStore or other source) and before it is installed, and before every execution of app. Public key of app publisher is contained in an X.509v3 certificate signed by a Certification Authority accepted by data provider and data consumer

B 025 (APS 05)	A Broker supports apps delivered and installed as independent software containers (i.e. apps bring along possible dependencies of e.g. software modules themselves and can be used irrespective of Connector's configuration).
B 026 (APS 06)	A Broker receives apps from a central app store.
B 027 (AUD 01)	Connector logs each access control decision in the form of an integrity protected log entry in its domain.
B 028 (AUD 03)	Connector logs any changes made to its configuration in the form of integrity protected entries in its domain.

3.3 Functional Requirements

B 029	An IDS Meta Data Broker must provide capabilities to persistently store metadata conforming to the IDS Information Model. The conforming Information model has to be at least of version 3.0.0. ⁹
B 030 (B5)	The architecture of an IDS Broker may be based on the general Connector architecture. Therefore, an IDS Broker is a specialized IDS Connector, which might not implement some Connector specific features.
B 031	Several protocols can be implemented. The default and recommended protocol is HTTPS.
B 032 (B17)	An interaction not meant for meta-data retrieval with an IDS Meta Data Broker contains exactly one of the following functions (see also Figure 1): <ul style="list-style-type: none">a) Register (a new entry)b) Update (an existing entry)c) Passivate (an existing entry)d) Activate (an existing entry)e) Remove (an existing entry)
B 033	Only the according Sovereign (defined according to IDS RAM 3.0) may register, change, or delete Connector or Resource metadata. Following restrictions apply: <ul style="list-style-type: none">a) The Sovereign can authorize a third party with this task, for instance if the Sovereign itself does not operate an own IDS Connector. In this case, the requesting Connector must supply an IDS AuthorizationToken proofing his permission.b) An IDS Meta Data Broker Operator is allowed to administer the contained data. In particular, an IDS Meta Data Broker Operator is allowed to delete or passivate metadata if it holds trustworthy indications that for instance the respective Connector or Resource left an International Data Space or stopped responding. In any case, an IDS Meta Data Broker Operator must not manipulate metadata in any way resulting in wrong information.

⁹ <https://github.com/International-Data-Spaces-Association/InformationModel>

B 034 (B20)	The identifier of all IDS Resources contains a unique character sequence, which is assigned by an authorized Connector. Connectors must receive their identity key from an IDS Identity Provider. This key is based on an URI, which has to be resolvable for all IDS members (must not contain localhost or local IP addresses). This URI is called the IDS Identifier.
B 035	The IDS Identifier may be used to retrieve the linked metadata in case no Meta Data Broker is available. The URI is not used for direct addressing a Connector and therefore does not need to point to the specific Connector location.
B 036	By using authorities from the global Domain Name System (DNS), the IDS Identifier is unique between all members. The IDS recommends the usage of DNS URIs for identifiers.
B 037 (B29)	An index service must not store external RDF vocabularies, not belonging to the Information Model. See IDS-RAM 3.0 explanations for Vocabulary Hubs.
B 038 (B30)	<p>An IDS Meta Data Broker is not responsible to validate the content of broker messages. This contains following cases</p> <ul style="list-style-type: none"> a) a Connector can be reached as described by the metadata b) a Connector exposes a data endpoint as published in the metadata c) a RDF vocabulary is available at the given URL
B 039 (B31)	<p>An IDS Meta Data Broker will provide a self-information like any other Connector. In difference to other Connectors this self-information has to be supplied at a specified URL (<code>http(s)://<broker.authority>[:port]/<path to broker root></code>) and gives additional information about the index service. This information contains additionally:</p> <ul style="list-style-type: none"> a) supported (native) query languages b) supported identity providers (like Connector) c) available add-on service (like data endpoints) d) publication of local index service rules (→ B18)
B 040	The attributes of B31 are specified in the IDS Information Model and have to be delivered in an RDF serialization.

B 041	An IDS Meta Data Broker uses state of the art communication encryption based on TLS. In case no TLS encryption is possible, Connectors should refuse the connection.
-------	--

3.4 Message Requirements

B 042	An IDS Meta Data Broker must announce the supported IDS Information Model versions for outbound and inbound messages through its Self-Description document.
B 043	Both categories from B13 are based on request and response messages: <ul style="list-style-type: none"> a) Instance of an Request or Notification Message (from Connector to an IDS Meta Data Broker) and an according Response instance (including a possible Rejection Message) (from an IDS Meta Data Broker to a Connector) b) QueryMessage (from a Connector to an IDS Meta Data Broker) and an according ResultMessage or suitable RejectionMessage.
B 044 (B15)	Every Message contains a unique messageId (URI). The corresponding ResponseMessages must contain this messageId as a correlationMessage attribute.
B 045 (B16)	A DescriptionRequestMessage sent to An IDS Meta Data Broker targets at most one of the following IDS entities identified by an URI: <ul style="list-style-type: none"> a) Connector b) Resource <p>If not provided with an URI a DescriptionRequestMessage has to be interpreted as a Request of a Self Description of the Broker.</p>
B 046 (B21)	A QueryMessage send to An IDS Meta Data Broker contains exactly one action: <ul style="list-style-type: none"> a) Query (one or multiple meta datasets by a query expression)
B 047	An IDS Meta Data Broker must reject messages if they do not conform to the IDS Information Model specifications and should send a proper RejectionMessage in that case.
B 048	If an IDS Meta Data Broker accepts a registration attempt, it must acknowledge it using a MessageProcessedNotification. A MessageProcessedNotification must not be sent if the indexing led to an error.
B 049	Error Handling: An IDS Meta Data Broker must react with Rejection Messages and appropriate error messages as follows: ·

- Message syntax is wrong:
idsc:MALFORMED_MESSAGE (for instance: incorrect JSON-LD, incorrect Multipart)
- Security Token or other identity or security related claims are wrong/could not be verified:
idsc:NOT_AUTHENTICATED including a hint to the first corrupted attribute causing the rejection (e.g. "Invalid DAT." or "DAT attribute does not match requesting hostname.")
- Access not granted because any requirement was not met:
idsc:NOT_AUTHORIZED including a hint to the first corrupted attribute causing the rejection.
- Requested resource does not exist or requester has no permission to this specific resource:
idsc:NOT_FOUND
- Message Type is not supported:
idsc:MESSAGE_TYPE_NOT_SUPPORTED (e.g. for an ParticipantRegisterMsg to an IDS Broker).
- Header or Payload use an Information Model version which the Connector can not handle:
idsc:VERSION_NOT_SUPPORTED with a list of supported versions.
- The Broker faced an unexpected internal error:
idsc:INTERNAL_RECIPIENT_ERROR with contact details of the responsible operator. Additional requirements may be specified in the respective Information Model.

3.5 Behavioral Requirements

B 050	In case an invalid or corrupted message is received, An IDS Meta Data Broker must reject it.
B 051	An IDS Meta Data Broker must not present data of a removed or passivated component after its removing has been acknowledged to the requesting entity.
B 052	An IDS Meta Data Broker must allow updates of a registered IDS Connector if the update request was originally initiated by this Connector, or by the IDS entity controlling the Connector, if this relation has been made visible to the Broker.

3.6 Business Requirements

B 053	<p>In case An IDS Meta Data Broker accepts meta data supplied with usage restrictions (especially access restrictions):</p> <ul style="list-style-type: none">• An IDS Meta Data Broker must have the technical capabilities to enforce them and• enforce these data restrictions and• outline these restrictions to requesting Connectors.
B 054	<p>An IDS Meta Data Broker must not agree on a business interaction on behalf of any participant except the one controlling the broker or if it was explicitly given consent by the respective Data Sovereign (and this consent was not revoked in the meantime) to do so.</p>
B 055	<p>An IDS Meta Data Broker should outline usage restrictions and licenses as far as it is aware of them and allowed to share them.</p>

3.7 Informational Requirements

B 056	An IDS Meta Data Broker must outline its support for IDS Information Model versions.
B 057	An IDS Meta Data Broker must only support one IDS Information Model version for outgoing messages.

3.8 Interface Requirements

B 058	<p>An IDS Meta Data Broker must have at least one of the following:</p> <ul style="list-style-type: none">• an HTML-based UI at <code>https://<authority>[:port][/path/to/broker]/browse</code>• an HTTPS endpoint for IDS Multipart messages at <code>https://<authority>[:port][/path/to/broker]/infrastructure</code>. In this case a Broker must support the IDS messages as specified in Table Functions.• An IDS-CP socket at <code>idscp://<authority>[:port]/</code> responding to Broker-related messages.
B 059	<p>An IDS Meta Data Broker must be able to enforce the security of data transmission using state of the art communication technologies. See also B40++.</p>

3.9 Conditional Requirements

B 060	<p>If an IDS Meta Data Broker sells the access to its data content:</p> <ul style="list-style-type: none">• An IDS Meta Data Broker must outline how data can be bought and which usage restriction or license applies.• An IDS Meta Data Broker must provide »One Click« agreement• An IDS Meta Data Broker must be able to execute a Transaction Accounting• An IDS Meta Data Broker must be able to send notifications to an IDS Clearing House for Data Exchange Clearing <p>If a ContractAgreementMessage has been acknowledged by a Broker and another entity, an IDS Meta Data Broker must behave according to this Agreement.</p>
B 061 (B11)	<p>If multiple Meta Data Broker exist, the following options for data provision to an index service must be ensured:</p> <ol style="list-style-type: none">a) A Connector sends multiple identical data messages to different IDS Meta Data Brokerb) A Connector sends different data messages to different IDS Meta Data Brokerc) A Connector sends data to a singular IDS Meta Data Broker. The data is propagated to other Brokers by the initially targeted Meta Data Broker.
B 062 (B14)	<p>An IDS Meta Data Brokers' index service can refuse the processing of Publishing Messages based on locally defined rules. If such rules exist an IDS Meta Data Broker must respond with a RejectionMessage with a RejectionCode and an explanation message.</p> <p>Example rules are:</p> <ul style="list-style-type: none">• An IDS Meta Data Broker message is incorrect• the signature is missing or can't be validated• the sender is not authorized to send message to this index service

	<ul style="list-style-type: none"> • the sender of the message is different to the id-token in the data header • the number of queries in a specified time may be limited
B 063	An IDS Meta Data Broker may also block a distinct Connector for an arbitrary period to prevent DDOS attacks. If such a rule is triggered, no ResponseMessage should be send, not even a RejectionMessage.
B 064	SPARQL is the query language of the Semantic Web and Linked Data. IDS Brokers may accept SPARQL queries as payloads of QueryMessages but can also provide support for path-based query languages (JSON-Path, XPath, ...), other graph-related query languages (Gremlin) or any other standardized query language. If an IDS Meta Data Broker provides querying possibilities, it should indicate the supported languages in their Self-Description.
B 065	In case an IDS Meta Data Broker accepts an IDS Usage Contract describing usage restrictions targeting a stored metadata element, an IDS Meta Data Broker must also enforce the contained restrictions. If an IDS Meta Data Broker cannot enforce the Usage Contract, it must reject it.
B 066 (B25)	<p>Some persistence technologies provide native query languages. An index-service may allow native queries in BrokerQueryMessages which is not obligatory. Here is a list of native query languages:</p> <ul style="list-style-type: none"> a) SQL ==> RDBMS (B24 c) b) LDAP-Query ==> LDAP / AD (B24 d) c) SPARQL ==> RDF(B24 e) and as an additional component: d) full-text search ==> query engine like Apache Lucene (B24 a-e) <p>If such native query languages are implemented, they have to be stated within an IDS Meta Data Brokers Self Description.</p>
B 067	If an IDS-Meta data Broker allows the usage of HTTP as the interaction protocol, HTTPS must be enforced.

B 068	If IDS Meta Data Broker has the ability to subscribe to a data source, it must react to published messages targeting itself according to IDS specifications. See "Behavior".
B 069	If IDS Meta Data Broker requires an authorization token for incoming messages, this token must be provided and verified according to the latest specification.
B 070 (B8)	<p>If multiple Meta Data Broker index services exist within an IDS Meta Data Broker Following options of meta-data inventories for index services are possible:</p> <ul style="list-style-type: none"> a) all index service provide the same meta-data, and synchronize their local states accordingly, b) individual index services provide different meta-data, or c) combinations from a) and b)

4 IDS Meta Data Broker Profiles

4.1 Advanced Information Profile

B 071	An IDS Meta Data Broker should indicate why it rejects a request from a valid IDS Connector.
B 072	An IDS Meta Data Broker should keep track of removed components.
B 073	An IDS Meta Data Broker should constantly verify the existence of its indexed IDS components and synchronize its indexed meta-data with the components self-descriptions.
B 074	An IDS Meta Data Broker supports the Search for data sources offered by Connectors. Possible search criteria are key words, taxonomies, multi-criteria facets.
B 075 (B18)	An IDS Meta Data Broker should provide version indicators, outlining that metadata has been updated. If implemented, one of the following solutions has to be applied: <ul style="list-style-type: none">a) the unique metadata key (URI) contains a version number (not recommended)b) the metadata contains a version number, which is incremented by the Connector or an IDS Meta Data Broker
B 076	The recommended scheme for IDS Identifiers is HTTP. IDS Participants should use dereferenceable URIs according to the Linked Data Principles (Linked Data - Design Issues1) for increased interoperability.
B 077	In case any of the options in B30 does not hold, an IDS Meta Data Broker may update its indexed information but is not expected to.
B 078	An IDS Meta Data Broker may support more than one IDS Information Model versions for incoming messages.
B 079	An IDS Meta Data Broker may support external vocabularies for component or data resource descriptions. In this case, an IDS Meta Data Broker should provide links to further information about these vocabularies (cf. Vocabulary Hub).
B 080	An IDS Meta Data Broker should have:

	<ul style="list-style-type: none">- an HTML-based UI at /browse and- an HTTPS endpoint for IDS Multipart messages at /infrastructure
B 081 (B26)	An IDS Meta Data Broker may provide a history feature for changed or deleted metadata. If available, a QueryMessage may contain a timestamp or a version number. Also, if An IDS Meta Data Broker does not provide a history, these arguments can be ignored and marked as such in the response.

4.2 Usage Control Profile

B 082	An IDS Meta Data Broker may be able to negotiate or at least provide data exchange agreements, as long it has the legal rights to do so.
B 083 (B23)	An IDS Meta Data Broker may filter or prohibit access to indexed metadata if an IDS Meta Data Broker has indications that the respective Data Sovereign has an interest in doing so. Such an interest can be encoded through IDS Usage Control Contracts, limiting access also of metadata to certain constraints.
B 084	An IDS Meta Data Broker may implement Usage Control engines, which can interpret and enforce IDS Usage Contracts as specified by the IDS Information Model.
B 085	An IDS Meta Data Broker may indicate that a certain rule or contract inhibits access or pretend that the requested information does not exist.

5 Functions and Correlated Messages

5.1 Messages received by a Broker

Function	Description / Intention	Priority ¹⁰	IDS Message type	Explanation Message Type ¹¹
Register a new Connector	Event notifying the recipient(s) about the availability of a new Connector	Mandatory	ConnectorAvailableMessage	<p>Multipart Request Message with a Header and Payload</p> <ul style="list-style-type: none"> – Header contains the type of message, security token, Connector's id, along with the date-time stamp in json format – Payload contains the Connector's title, description, curator, maintainer, and its catalog in json format <p>NOTE: the Connector's id in the header (issuerConnector) and payload (id) should be the same</p> <p>Expected response :</p> <ul style="list-style-type: none"> – MessageProcessedNotification (on success) – RejectionMessage (on Failure)
Update whole/distinct Connector attributes	Event notifying the recipient(s) about a Connector's configuration change	Mandatory	ConnectorUpdateMessage	<p>Multipart Request Message with a Header and Payload</p> <ul style="list-style-type: none"> – Header contains the type of message, security token, Connector's id, along with the date-time stamp in json format – Payload contains updated Connector's self-description (similar to ConnectorAvailableMessage) in json format <p>NOTE: the Connector's id in the header (issuerConnector) and payload (id) should be the same</p> <p>Expected response :</p> <ul style="list-style-type: none"> – MessageProcessedNotification (on success) – RejectionMessage (on Failure)

¹⁰ mandatory, optional, implementation-specific

¹¹ Structure, content, expected responses

Unregister a Connector	Event notifying the recipient(s) that a Connector will be unavailable and never be available again	Mandatory	ConnectorUnavailableMessage	Multipart Request Message with only a Header <ul style="list-style-type: none"> Header contains the type of message, security token, Connector's id, along with the date-time stamp in json format Expected response : <ul style="list-style-type: none"> MessageProcessedNotification (on success) RejectionMessage (on Failure)
Register a *collection* of Connectors	Event notifying the recipient(s) about the availability of a new set of Connectors	implementation-specific		https://industrialdataspace.jive.com/docs/DOC-1817#jive_content_id_Broker_Registration
Update a *collection* of Connectors	Event notifying the recipient(s) about a set of Connector have changed	implementation-specific		
Unregister a *collection* of Connectors	Event notifying the recipient(s) that a set of Connectors will be unavailable	implementation-specific		
Register a new Resource	Event notifying the recipient(s) about the availability of a new Data Resource	mandatory	ResourceAvailableMessage	
Update a distinct Resource	Event notifying the recipient(s) about a resource' change	mandatory	ResourceUpdateMessage	
Unregister a distinct Resource	Event notifying the recipient(s) that a resource will be unavailable and never be available again	mandatory	ResourceUnavailableMessage	
Register a new *collection* of Resources	Event notifying the recipient(s) about the availability of a	implementation-specific		

	new set of resources			
Update a *collection* of Resources	Event notifying the recipient(s) about a set of resources has changed	implementation-specific		
Unregister a *collection* of Resources	Event notifying the recipient(s) that a set of resources will be unavailable	implementation-specific		
Query for a specific available Connectors/ Artifacts or all available Connector/Artifacts	Query message intended to be consumed by a component	mandatory	DescriptionRequestMessage	
Query Meta Data from the	Query message intended to be consumed by a component	optional	Query Message	<p>Multipart Request Message with a Header and Payload</p> <ul style="list-style-type: none"> – Header contains the type of message, security token, Connector's id, along with the date-time stamp in json format – Payload contains the sparql query in text format <p>Expected response :</p> <ul style="list-style-type: none"> – ResultMessage (on success) – RejectionMessage (on Failure)
Retrieve the component's self description document	Message requesting the self-description of an Infrastructure Component	mandatory	RequestMessage	<p>Expected response :</p> <ul style="list-style-type: none"> – DescriptionResponseMessage (on success) – RejectionMessage (on Failure)

5.2 Messages send by a Broker as Response

Returned Messages Interpretation	Description / Intention	Priority	IDS Message type	Explanation Message Type ¹²
Resultant message from Broker on success	Indicates that the incoming message has been processed and accepted.	Mandatory	MessageProcessed Notification	Always part of an Interaction Sequence. <ul style="list-style-type: none"> Header contains the type of message, security token, Broker's id, along with the date-time stamp in json format Payload contains the Broker's self description
Resultant message from Broker on failure	Indicates that the incoming message was not correct or could not be handled properly. The required 'rejectionReason' should help the sending Connector to understand the error/issue.	Mandatory	RejectionMessage	Always part of an Interaction Sequence. Contains only a Header with the following attributes: type of message, correlation message, rejection reason, along with the date-time stamp
Response to a QueryMessage from Broker	Contains the result of the query for the resultant QueryMessage	Mandatory	ResultMessage	Always part of an Interaction Sequence <ul style="list-style-type: none"> Header contains the type of message, security token, Meta Data Broker's id, along with the date-time stamp in json format Payload contains the query result in text format
Response to a DescriptionRequest from Broker	Message containing the self-description of an Infrastructure Component	Mandatory	ResponseMessage	Always part of an Interaction Sequence <ul style="list-style-type: none"> Header contains the type of message, security token, Meta Data Broker's id, along with the date-time stamp in json format

¹² Structure, content, expected responses

				<ul style="list-style-type: none">- Payload contains the Meta Data Broker's self-description in json format
--	--	--	--	---

6 Annex

6.1 HTTP API

At this moment, the IDS Meta Data Broker Interface of the reference Implementation consists of at least two HTTP endpoints. One for query operations and the other for data manipulating requests. All input and output of the Broker API is in multipart format with three parts: Meta Data, Broker Message, and Signature.

The API description in **OpenAPI 3.0** is publicly available and has the same normative meaning as all other requirements of this document as far as the HTTP binding is concerned. The currently valid version of the API description is too large for this document but can be found at this link:

<https://app.swaggerhub.com/apis/idsa/IDS-Broker/>

A **POSTMAN collection** further describes exemplary client-side requests against an IDS Meta Data Broker following the above HTTP API. In contrast to the OpenAPI description of the API, the POSTMAN collection is a non-normative resource supporting the interested reader. It is hosted at this link:

<https://idsa-redirects.postman.co/collections/8586393-e1b4de75-57ae-49d4-9912-8858373fc7ad?version=latest&workspace=a498d519-86ee-44e9-8946-6ec406d08a14>

6.2 Removed Requirements

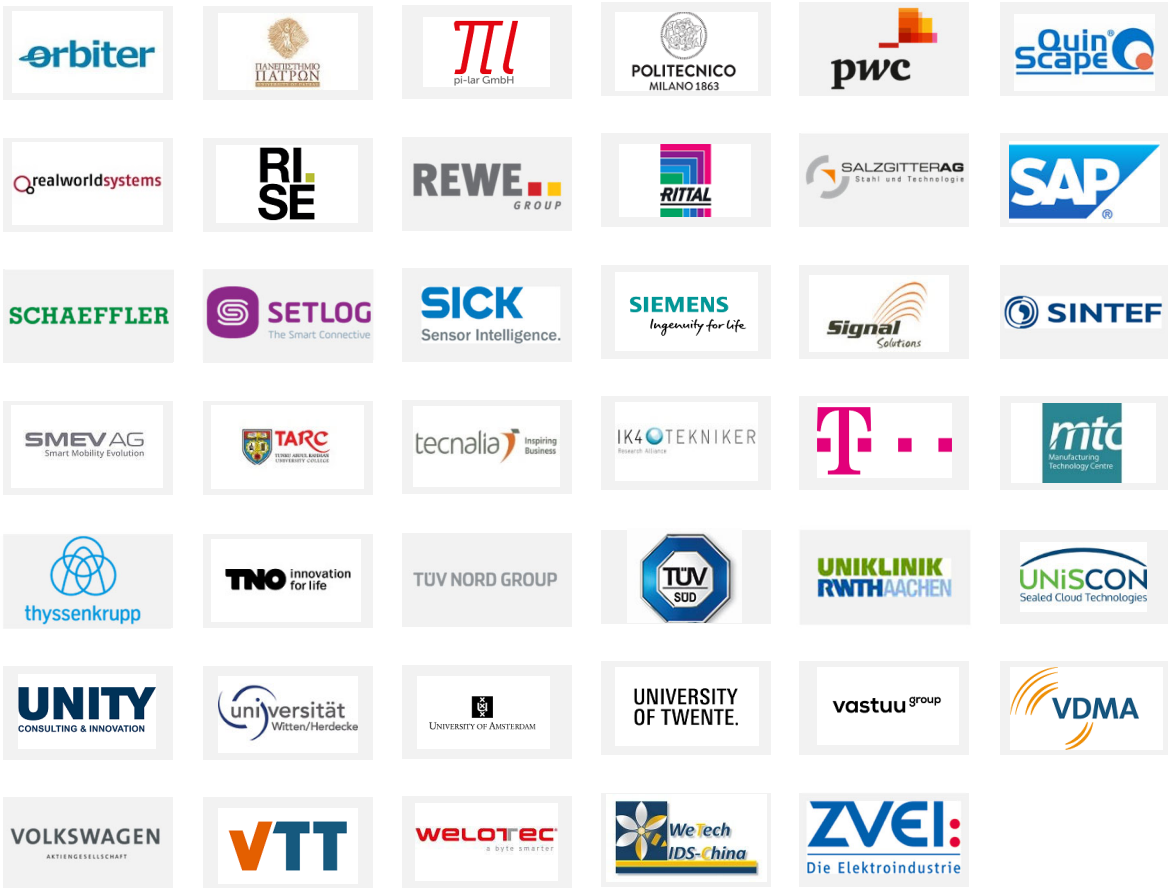
The following listed requirements have been introduced by version 1.0 of this document but needed to be removed in the further specification process. They are kept here for legacy reasons but have no normative meaning anymore.

No.	Description
B2	A broker can offer different services: <ul style="list-style-type: none">• Index service• Clearing• Marketplace
B9	If multiple broker index services exist <ol style="list-style-type: none">a) index services are isolated and ignore othersb) index services from a network and propagate data to othersc) combinations from a) and b) are allowed
B19	A Connector may only add or change his own metadata in the index service. The index service has to validate from the id token of the DataHeader, if the sender is authorized to request a specific action. Attention: An Identity provider may use different keys than the International Data Spaces. The index service therefor has to map IdP keys to IDS keys.
B22	A BrokerQueryRequest contains exactly one scope: <ol style="list-style-type: none">a) All (query all active or passive datasets)b) Active (query only active datasets)c) Access (query only active datasets, which are available for the questioner)
B32	An index service may also store metadata of data-apps.
B33	The BaseBroker is based on the BaseConnector and Ubuntu with Docker.

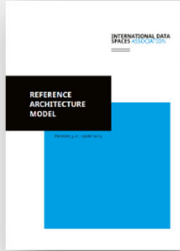
B34	The BaseBroker consists of this containers: a) Execution Core Container with ActiveMQ and Camel (App Store Image) b) Broker-API (BaseBroker Image) c) ApacheDS (DockerHub Image) d) Keycloak (DockerHub Image)
B35	ApacheDS loads default and additional IDS index service schema for RDF file storage.
B36	Running Keycloak in the Connector is optional. A broker may use any Identity Provider.
B37	Keycloak and index service do not use a shared LDAP. Keycloak just syncs with LDAP sources and uses an internal RDBMS for storage.
B39	Broker-API is a Spring Boot REST service, which parses incoming BrokerMessages and executed corresponding LDAP actions. The connection to LDAP is done with Spring LDAP (based on JNDI).
B41	A camel workflow connects the incoming queue BrokerRequest with Broker-API.
B42	The BaseBroker uses JMS Request/Response pattern for BrokerResponseMessage with a temporary queue.

OUR MEMBERS





OVERVIEW PUBLICATIONS



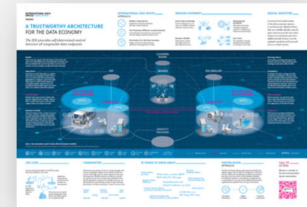
Reference Architecture Model



Executive Summary



Image Brochure



Infographic



Use Case Brochures



Study on Data Exchange



Position Paper Implementing the European Data Strategy



Position Paper GDPR Related Requirements and Recommendations



Position Paper Usage Control in the International Data Space



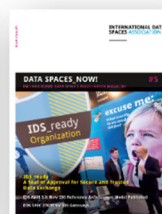
Position Paper IDS Certification Explained



White Paper Certification



White Paper Sharing data while keeping data ownership



Magazine Data Spaces_Now!

For these and further downloads: www.internationaldataspaces.org/info-package

Code available at: <https://github.com/industrial-data-space>

CONTACT


Head Office

INTERNATIONAL DATA SPACES ASSOCIATION

Emil-Figge-Str. 80
44227 Dortmund | Germany

phone: +49 231 70096 501
mail: info@internationaldataspaces.org

WWW.INTERNATIONALDATASPACES.ORG

 [@ids_association](https://twitter.com/ids_association)

 [international-data-spaces-association](https://www.linkedin.com/company/international-data-spaces-association)