

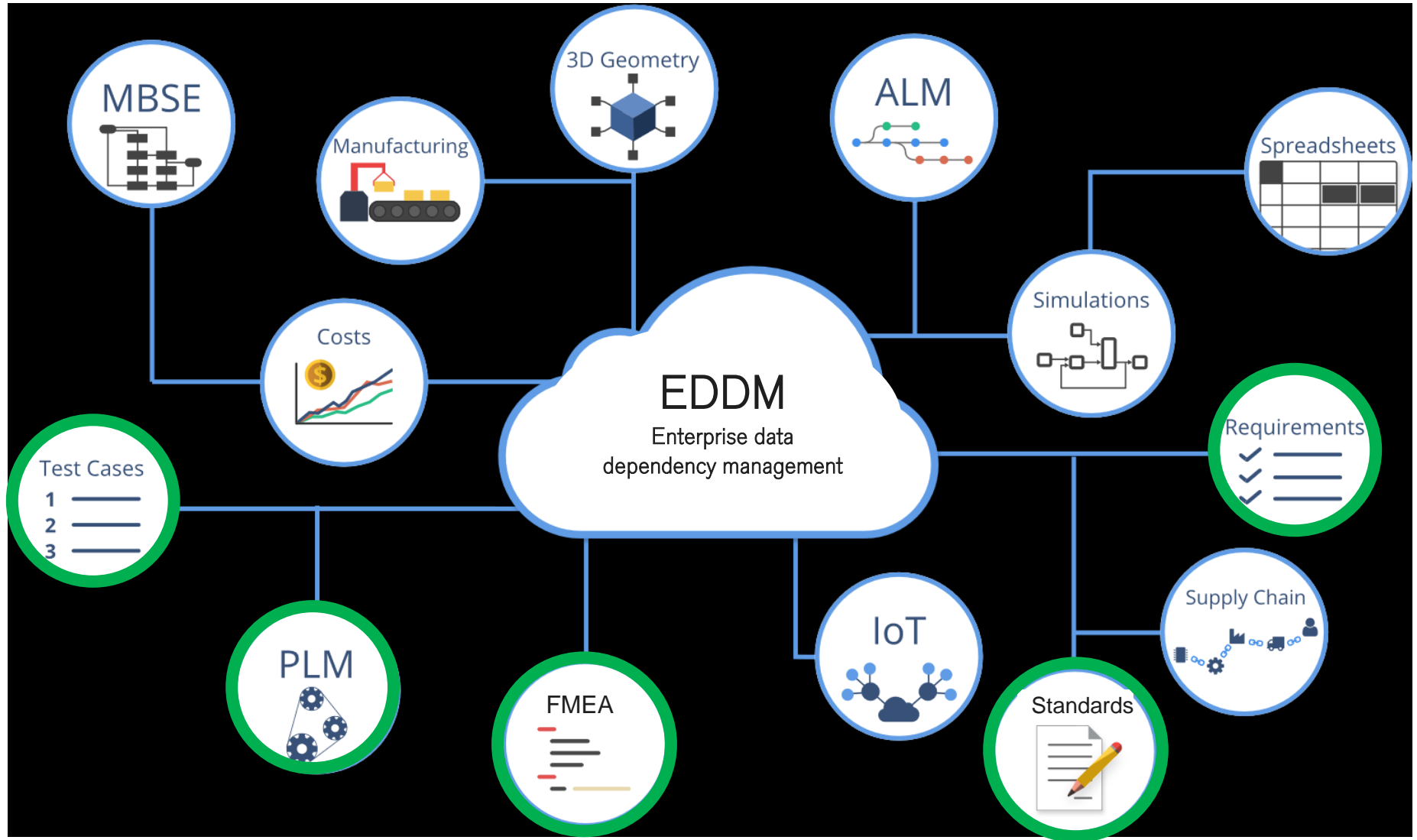


Enterprise data dependency management

J. Palanivel (Wipro), C. Schaefer (Dräger), H. Schulz-Mirbach (Dräger)
OSLC Fest 2022

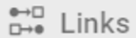
Enterprise data dependency management (EDDM)

Data linking, change detection and baselining within the enterprise



Enterprise data dependency management (EDDM)

Core functionality



Links

Data linking

Manage links between all EDDM enabled data containers

Two types of links

- Item links: involving endpoints that might change and are subject to EDDM change detection
- Element links: involving endpoints that will not change

EDDM link status

- Valid: link is valid and link endpoints have not changed
- Suspect: at least one link endpoint has changed
- Invalid: link is invalid (user defined)
- Dangling: at least one link endpoint is lost



Change detection

Detect link endpoint changes with automatic EDDM link maintenance

Link status becomes suspect if at least one link endpoint has changed

Link status becomes dangling if at least one link endpoint is lost



Baselines

Baselining of EDDM link sets

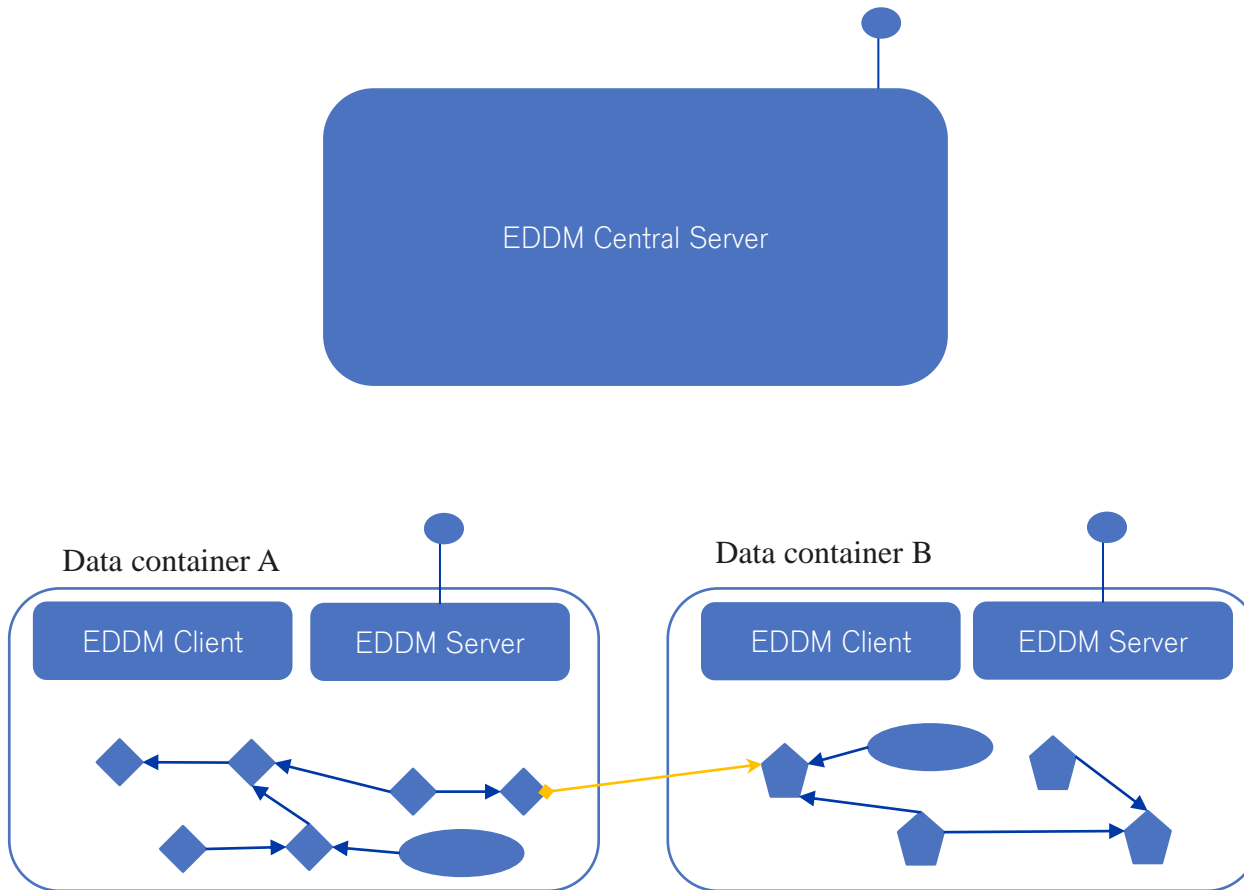
Store an immutable set of EDDM links in a named baseline

EDDM baselines are independent of the baselines of the link endpoints

Data dependency graph (DDG) supports the creation of EDDM baselines

Enterprise data dependency management (EDDM)

Solution architecture



EDDM Client:

- Embedded in the GUI of each data container
- Enables data link create/update/delete

EDDM Server:

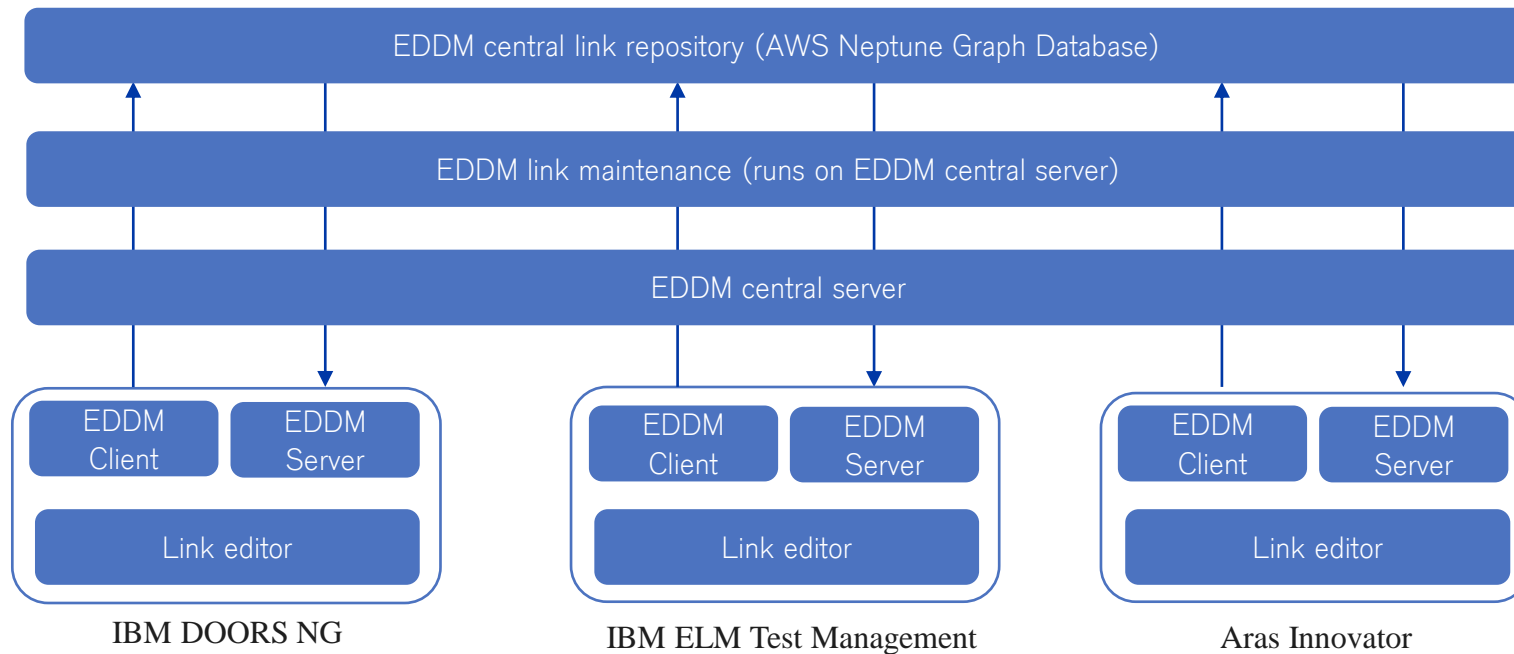
- Can be embedded in a data container or runs standalone
- Manages all EDDM queries for the data container
- Manages all data operations on data links for the data container
- Provides delegated UI's for data elements, items and data links

EDDM Central Server:

- Enables EDDM services discovery
- Translation from/to symbolic EDDM data links into resolvable URL's

Enterprise data dependency management (EDDM)

Deployment architecture



EDDM link maintenance manages EDDM links in up to three storage locations

1. Local data container: holds EDDM links where the link source belongs to the data container
2. PLM (Aras Innovator): holds all EDDM links (used for SQL based link analytics)
3. Central link repository (AWS Neptune): holds all EDDM links (used for graph based link analytics)

Enterprise data dependency management (EDDM)

Implementation and capabilities

Test Case 24731 (Current)

New Link Links DDG Link Editor

Filter: *← →*

Type	Id	Description	Link Status
	MP00355	VentStar Flex 220	

Part : MP00355

Overview

Description 1 VentStar Flex 220
State Released
Revision 08
Generation 9
ERP PI Series

Details

OFT/PSB
Language 1 de
Modified on 2020-04-15T08:30:20
Description 2 VentStar Flex 220
Language 2 en

Part MP00331

Released-12VentStar Anaesth

Filter: *← →*

Type	Id	Description	Link Status
	345	Verification Structure	
	386	Software Test	
	822	PCS 1.1.2 Verification Structu...	
	823	SW Plan v 1.1.2	
	824	SW Verification Structure 1.1...	

Mark Selected As Valid

Mark Selected As InValid

Delete Selected Link

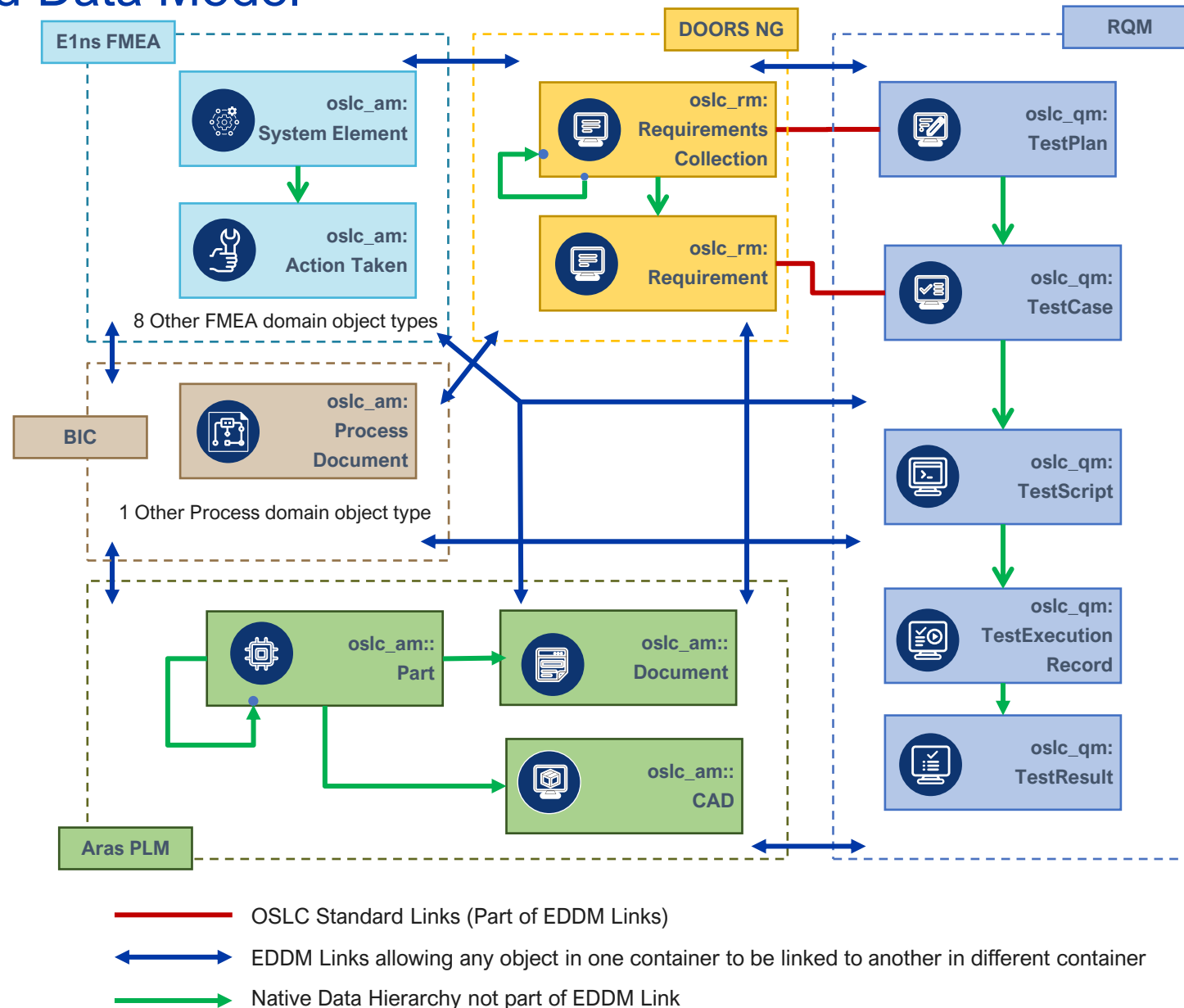
- Seamless integration in client UI
- Bulk operation capabilities
- Change detection capabilities
- Data dependency management
- Baselines on link sets

Data Item / Element	Link Status	Description
Part MP00331		VentStar Anaesth (P) o. LuerL
Test Plan 822		No Baseline
Test Plan 345		PCS Core Initial Dev
Test Plan 386		PCS Core Initial Dev
Test Plan 823		No Baseline
Test Plan 824		No Baseline
Document DCS SC2090		Shelf Life Management
Document DMS IN9050		Biocompatibility of medical devi
Document MP00339		PTS Breathing Circuits / Bags

Enterprise data dependency management (EDDM)

Common & Linked Data Model

- OSLC+ Data model approach
- OSLC Standard links are allowed and used as EDDM links
- Additional capability to create EDDM links between objects across containers.
- An EDDM link between object A to B means A depends on B
- Semantic meaning of the dependency left to business usage.



Enterprise data dependency management (EDDM)

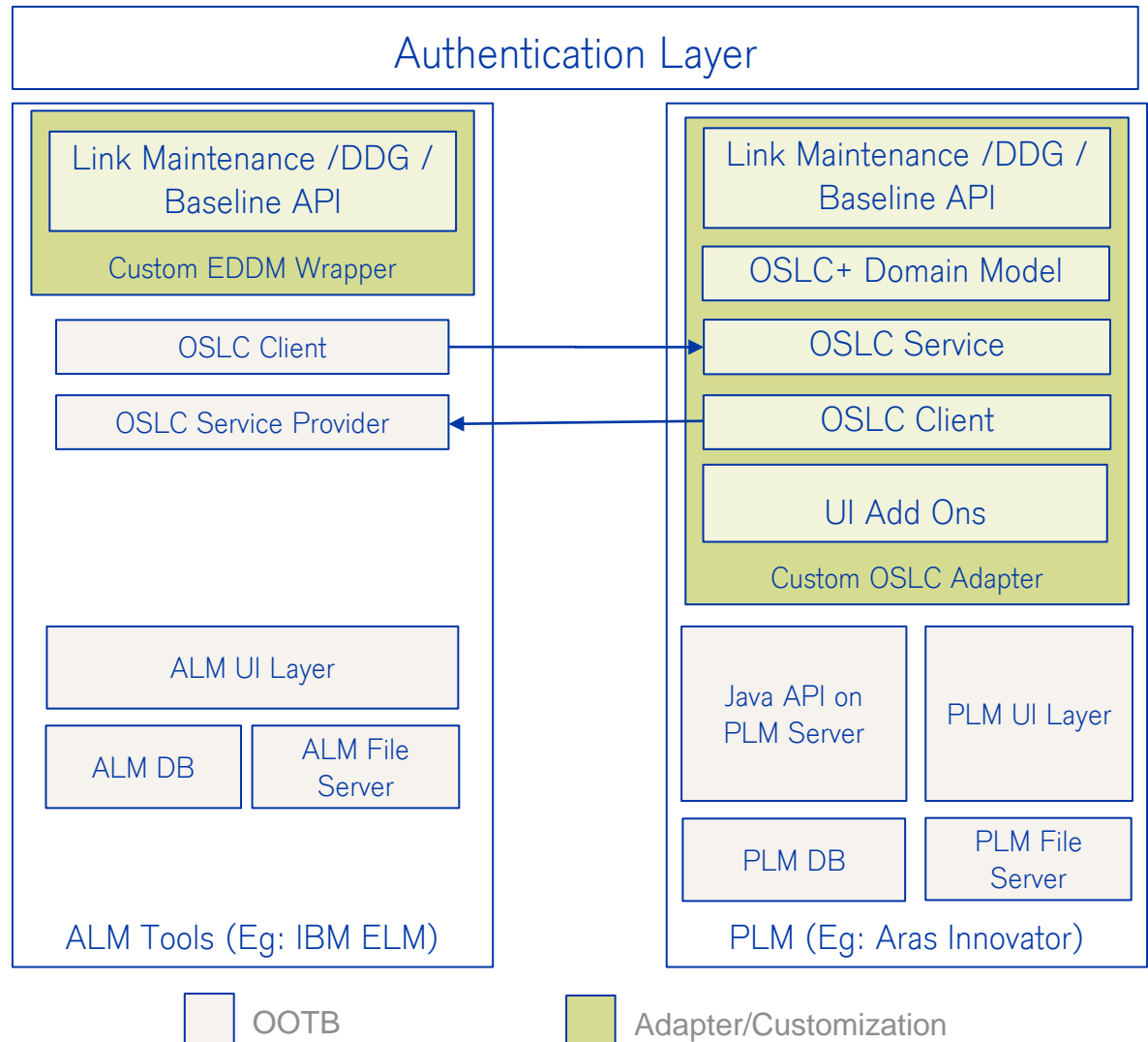
Use of Wipro's OSLC adapters when no native OSLC support is provided by container

OSLC Adapters for PLM

- Provide standards compliant services for
 - Change Management 2.0
 - Requirement Management 2.0
- Where the OSLC standards are not final, provide services aligned to the current direction
 - Product Definition 1.1
 - Core Extensions for PLM 2.0

EDDM Wrappers

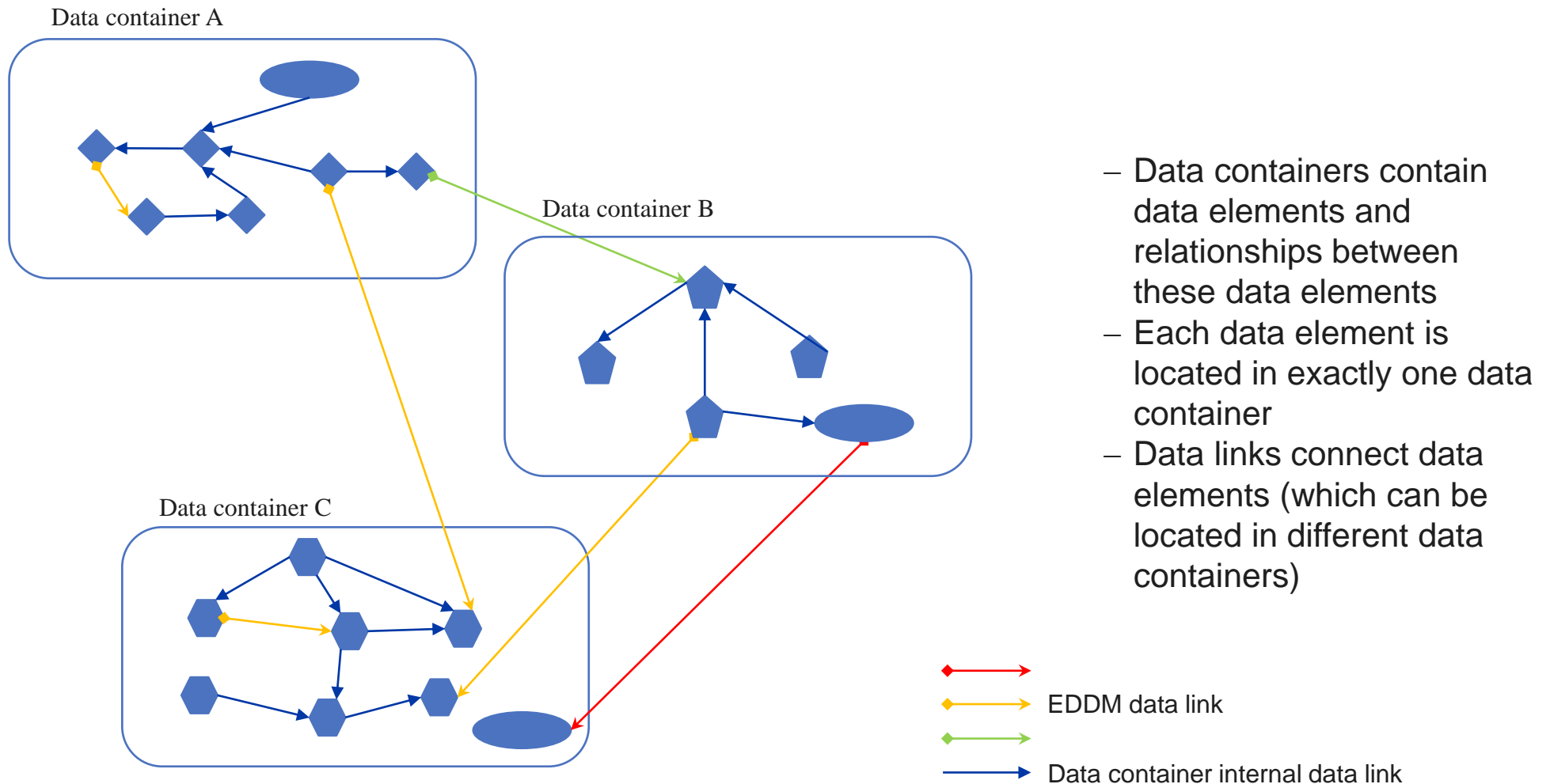
- EDDM Wrappers for native OSLC provider developed for necessary additional API
- Required for Link Maintenance, Data Dependency Graph, Baseline



Appendix

Enterprise data dependency management (EDDM)

Data containers, data elements, data links



Enterprise data dependency management (EDDM)

Definition of: Data element



Data element

- A data element is a self-contained indivisible unit of information
- A data element is uniquely identified by a URI (Uniform Resource Identifier)
- A data element is immutable (does not change over time)
- A data element might have additional attributes (these are not relevant in the context of EDDM)

Enterprise data dependency management (EDDM)

Data element – Related concepts in PLM systems



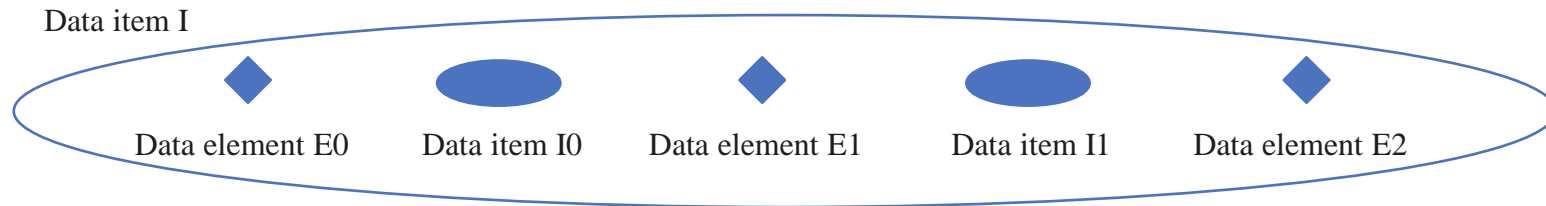
Data element

The data element is the EDDM abstraction of „a thing that will never change“. Here are some remarks how this relates to concepts in PLM systems

- In the the Aras PLM system all data (e.g. parts, documents, CAD) have an item_id (which is technically a globally unique identifier (GUID)) that is unique across all data elements in the system. Whenever PLM data are changed either a new „Generation“ or a new „Revision“ is created with a new item_id (identified by a new GUID). Therefore data identified by item_id's are examples of data elements
- Other systems (e.g. DOORS and RQM) use the concept of maturity level of data in the system. When the data maturity level has a value like Preliminary data can be changed, when the maturity level has a value like Released or Superseded data can no longer be changed. Therefore Released or Superseded data are examples of data elements. It should be noted that details of the maturity levels vary between different systems

Enterprise data dependency management (EDDM)

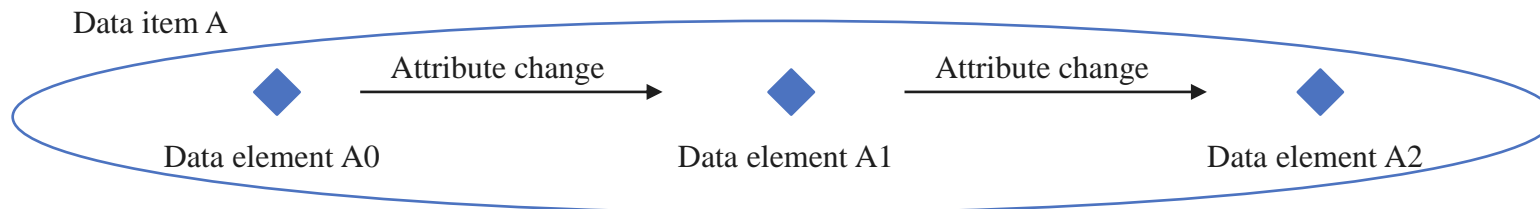
Definition of: Data item



- A data item is a set of data elements and data items
- A data item cannot contain itself
- A data item is uniquely identified by a URI
- A data item can change over time when data elements or data items are added or removed
- A data item has a change indicator that is updated whenever the data item changes

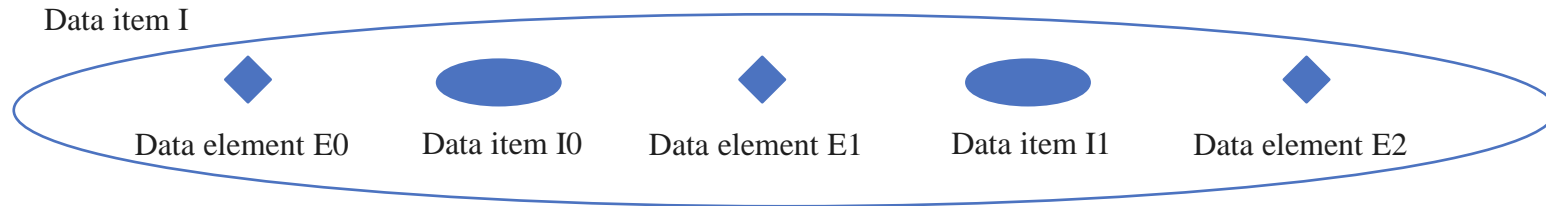
Remarks

- Any change of attributes of a data element results in the creation of a new data element. The resulting set of data elements is an example of a data item



Enterprise data dependency management (EDDM)

Data item change indicator



The change indicator of a data item is updated whenever the data item changes by

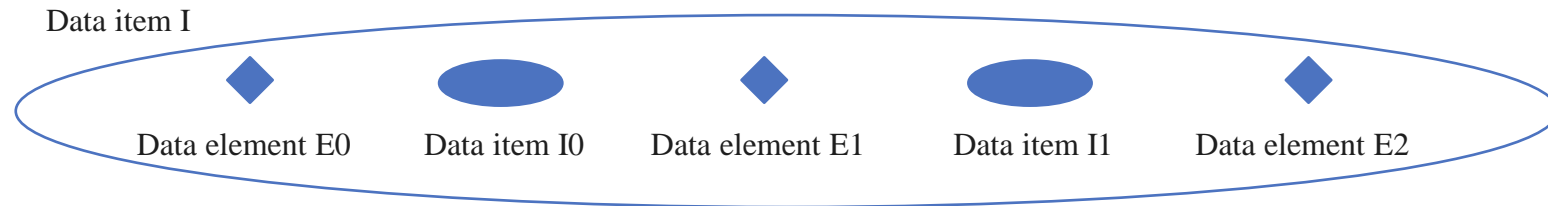
- adding or removing data elements
- adding or removing data items

The data item change indicator is not updated when

- the change indicator of a contained data item is updated
- data links are created or updated that have the data item as link source or target
- data links are created or updated that involve any of the contained data elements or data items as link source or link target

Enterprise data dependency management (EDDM)

Data item – Related concepts in PLM systems and OSLC

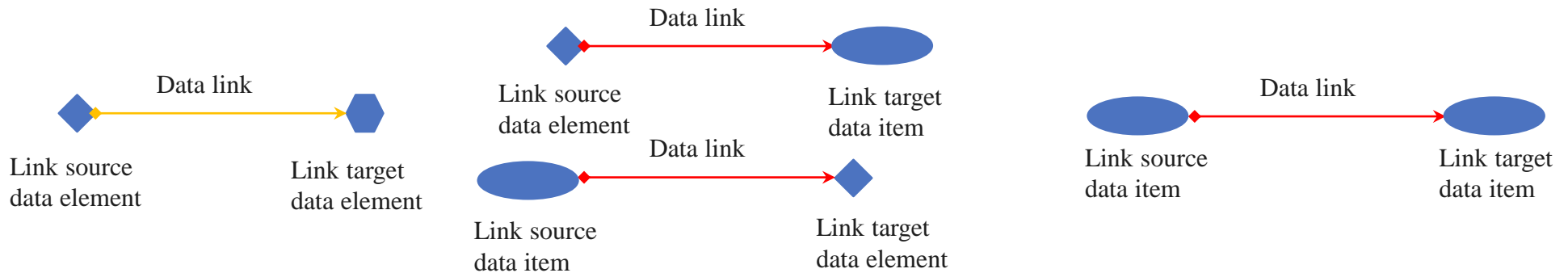


The data item is the EDDM abstraction of „a thing that can change“. Here are some remarks how this relates to concepts in PLM systems and OSLC

- In the Aras PLM system a configured item tracks its changes over time. Minor changes result in a new „Generation“ while larger changes result in a new „Revision“. There is always a „Current“ version of the configured item that is the most recent data set. In Aras PLM a configured item is identified by a separate config_id (which is technically a globally unique identifier (GUID)). The contained Revisions and Generations are identified by separate item_id's. The PLM configured item is an example of a data item
- The OSLC configuration management specification has the concepts of Component (A unit of organization consisting of a set of version resources), Configuration (A configuration identifies a set of versions of resources in a component) and Configuration Item (A configuration item is a resource selected in a configuration)

Enterprise data dependency management (EDDM)

Definition of: Data link



- A (EDDM) data link is a directed relationship between two data elements or data items: the data link source and the data link target. These can be located in different data containers
- A data link has the following data attributes
 - link_uri : Uniform resource identifier (URI) of the data link
 - source_uri : URI of the source data element or data item of the data link
 - target_uri : URI of the target data element or data item of the data link
 - change_indicator: is updated whenever the source or target of the data link change. The change indicator of the data link is derived from the tuple (change_indicator data link source, change_indicator data link target)
 - link_status: is one of the values: Valid (✓), Suspect (?), Dangling (≠), Invalid (X)
- Data links are stored in a data container
- Data links are data items
- Data links are navigated by following the URI given for the data link source and target

Enterprise data dependency management (EDDM)

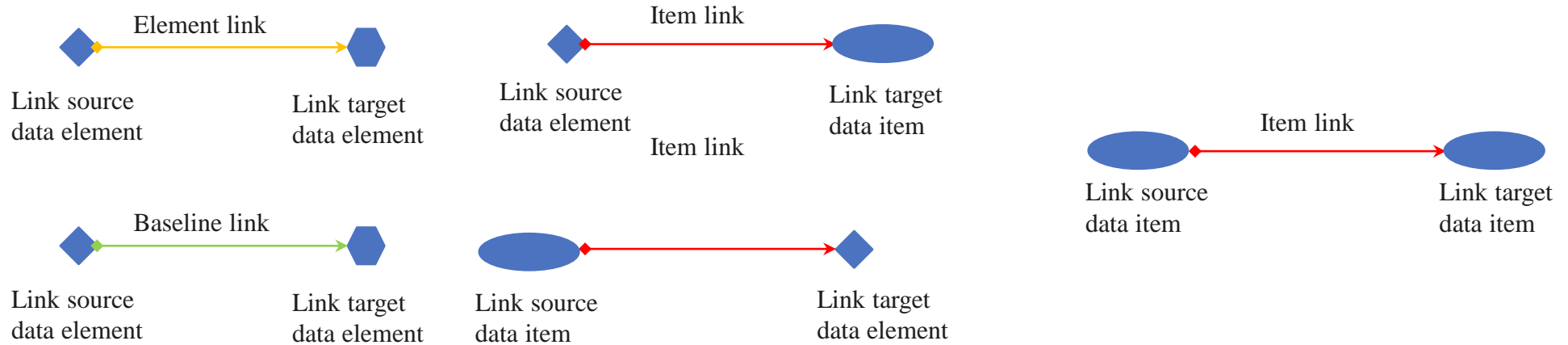
Definition of: Data link status



- Link status Valid (✓): both link source and target can be accessed, change indicator of the data link (change_indicator data link source, change_indicator data link target) matches the change indicators from the link source (change_indicator data link source) and link target (change_indicator data link target)
- Link status Suspect (?): both link source and target can be accessed, change indicator of the data link (change_indicator data link source, change_indicator data link target) does not match the change indicators from the link source (change_indicator data link source) and link target (change_indicator data link target)
- Link status Dangling (≠): either the link source or link target cannot be accessed (e.g. due to data deletion of the data item in the source or target data container)
- Link status Invalid (X): that status is manually set by end users to indicate that the link is invalid from a business perspective

Enterprise data dependency management (EDDM)

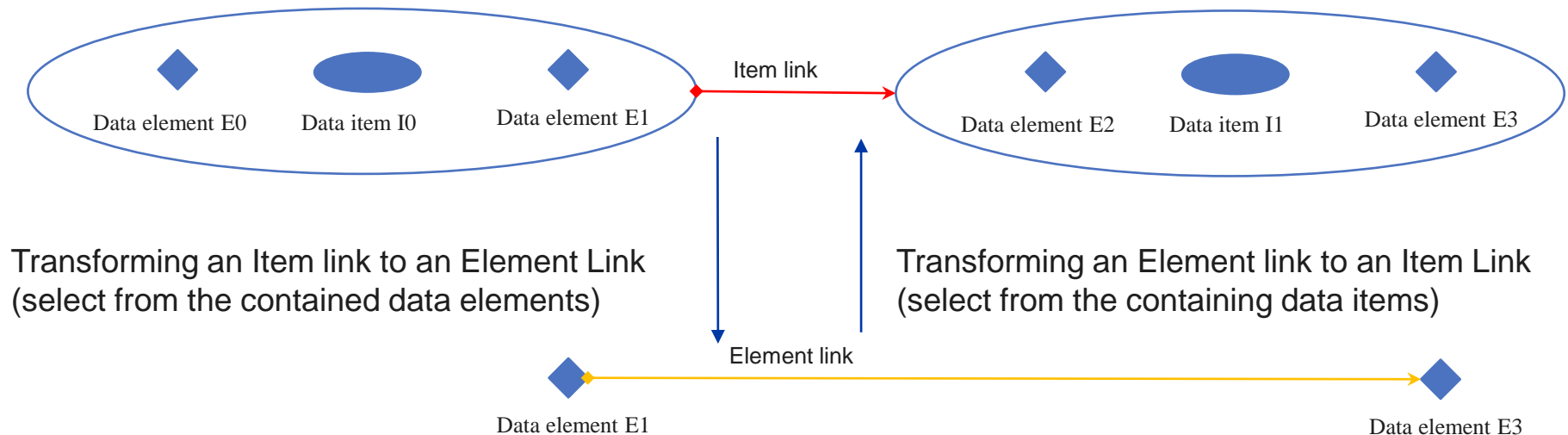
Definition of: Element link, Baseline link, Item link



- A data link is called an Element link (→) if both link source and link target are data elements
- An Element link is called a Baseline link (→) when it cannot be changed any more by EDDM
- A data link is called an Item link (→) if either link source or link target or both are data items

Enterprise data dependency management (EDDM)

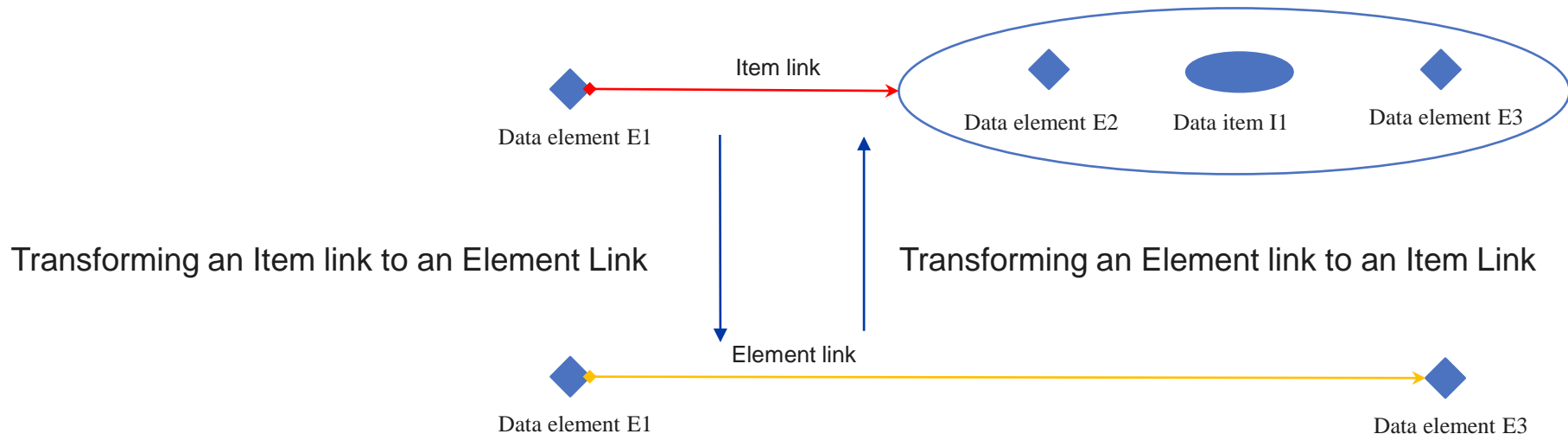
Transforming between Element links and Item links



- Item link → Element link: Select for link source and link target data item data elements that are elements of these data items and replace link source or link target with these data elements
- Element link → Item link: Select a data item where the link source data element is an element of or select a data item where the link target is an element of and replace link source or link target with these data items

Enterprise data dependency management (EDDM)

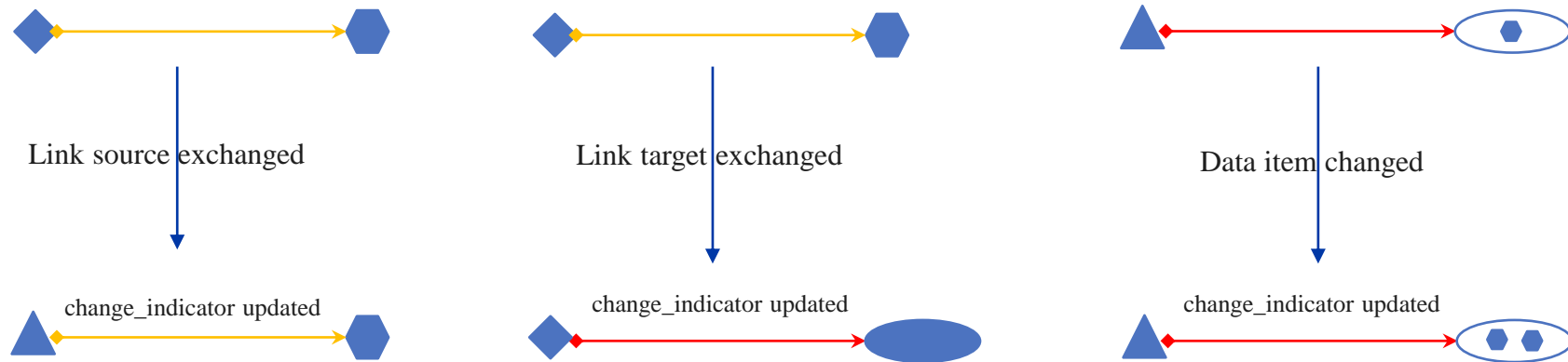
Transforming between Element links and Item links



- Item link → Element link: Select for link source and link target data item data elements that are elements of these data items and replace link source or link target with these data elements
- Element link → Item link: Select a data item where the link source data element is an element of or select a data item where the link target is an element of and replace link source or link target with these data items

Enterprise data dependency management (EDDM)

Examples of data link changes

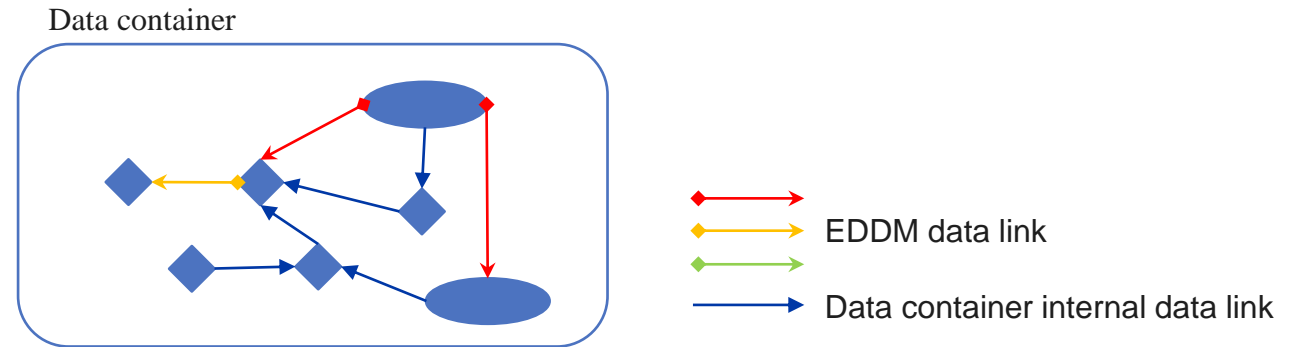


The change_indicator of a data link is updated when

- The source of the data link is exchanged
- The target of the data link is exchanged
- When the source or the target of the data link is a data item and this data item changes (by adding or removing data elements from the data item)

Enterprise data dependency management (EDDM)

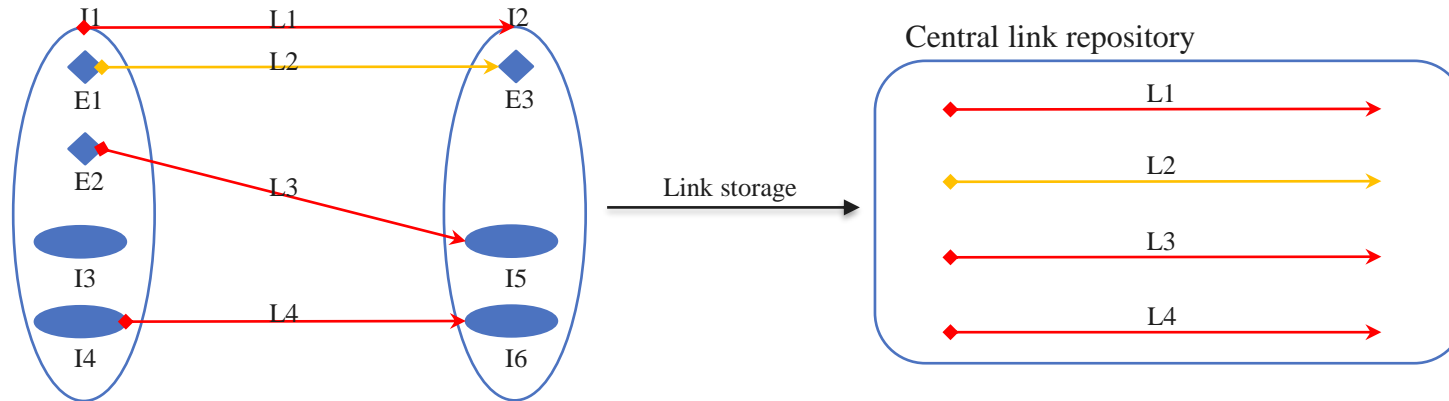
Definition of: Data container



- A data container is an IT system for the creation/update/deletion and persistence of
 - data elements
 - data items
 - EDDM data links (element links and item links and baselines)
 - Data container internal data links (these are not subject to EDDM)
- A data container is uniquely identified by a URI

Enterprise data dependency management

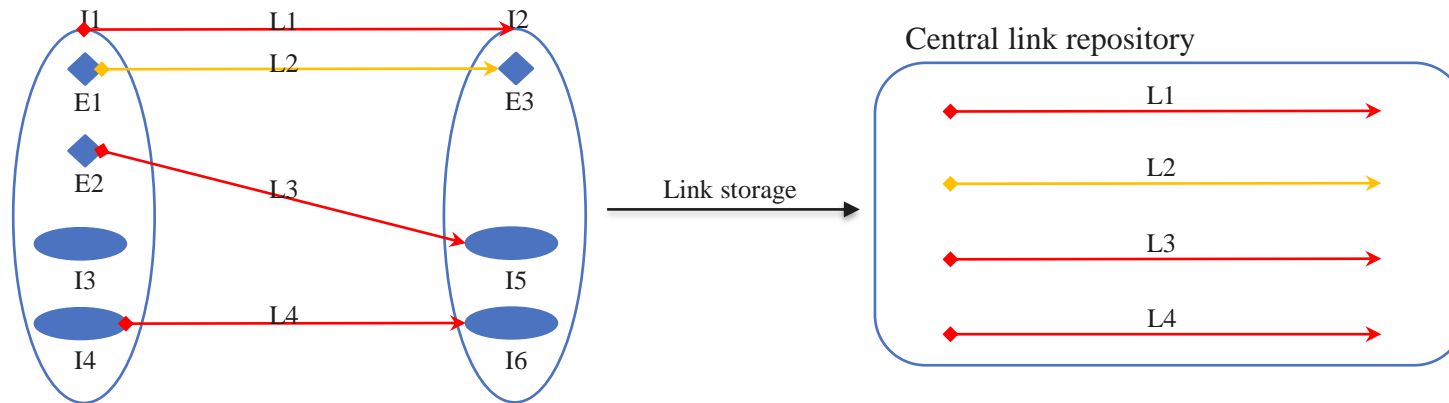
Definition of: EDDM Central Link Repository



- Data links are stored in EDDM servers. That results in a distributed storage architecture
- Distributed link storage might be inefficient for operations that handle large link sets from different data containers
- The EDDM central link repository is a EDDM server (it could be included in the EDDM central server) that holds (possibly redundantly) all links in the EDDM landscape
- EDDM servers must either replicate their locally stored links to the central link repository or use the central repository as native link store
- All link replicas are read-only in the central link repository (i.e. end users cannot directly change links in the central link repository)

Enterprise data dependency management

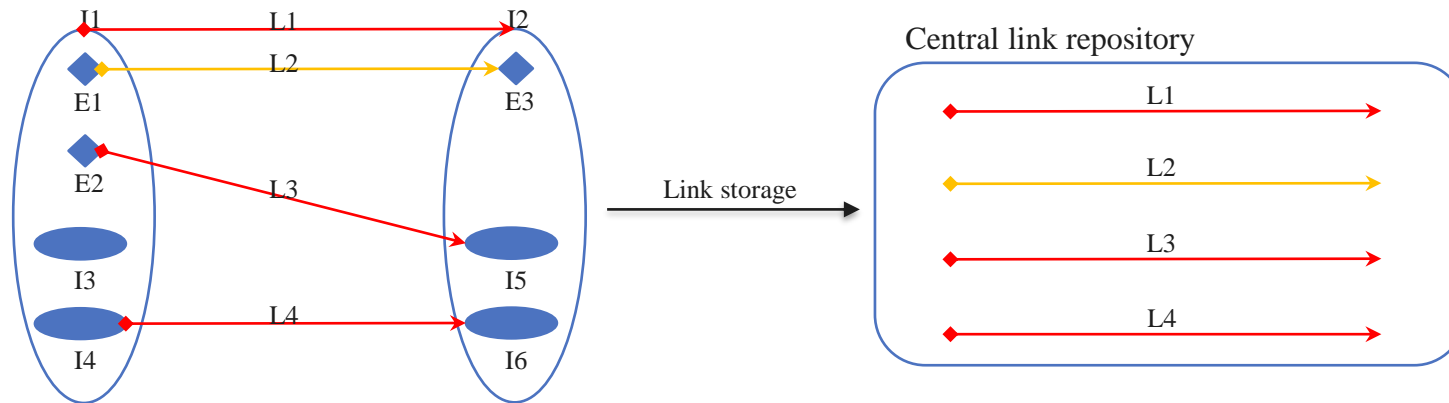
Central Link Repository – requirements for data handling



- Each EDDM server that uses both local storage and the central link repository for storing EDDM links must always store any link updates both on local storage and in the central link repository
- Main reasons for supporting both local link storage and storage in the central link repository
 - Applications might already have data linking capabilities (e.g., DOORS, RQM). These should continue to be used without changes → support local link storage
 - Creation of baselines and impact analysis of changes will require information about all existing links → support central link storage

Enterprise data dependency management

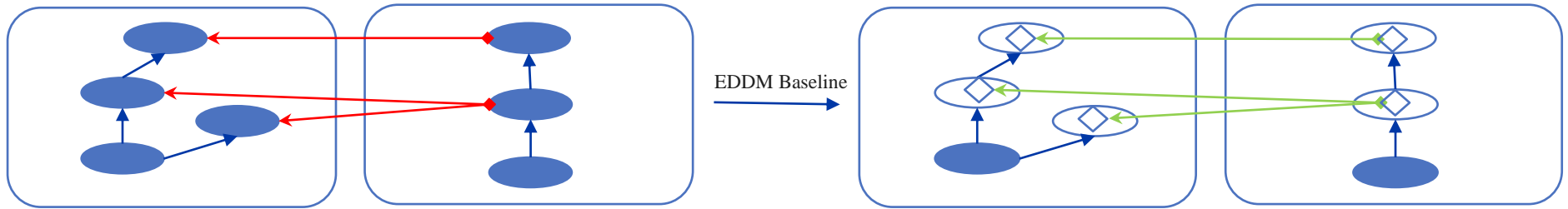
Central Link Repository – keeping the link status current



- The link status of data links changes based on the following changes of link source and target
 - Only for item links: change indicator of link source or link target data item changes
 - Deletion of either link source or link target
 - End user sets link status to Invalid
- EDDM must ensure that the data link status is kept current when link source or target change. Here are some implementation options
 - EDDM clients check and update the link status prior to displaying the link to end users
 - Data containers update the link status whenever link source or target change
 - Scheduled batch jobs update the link status periodically (that implies that the link status might be outdated when data items change and no batch job has been run)

Enterprise data dependency management

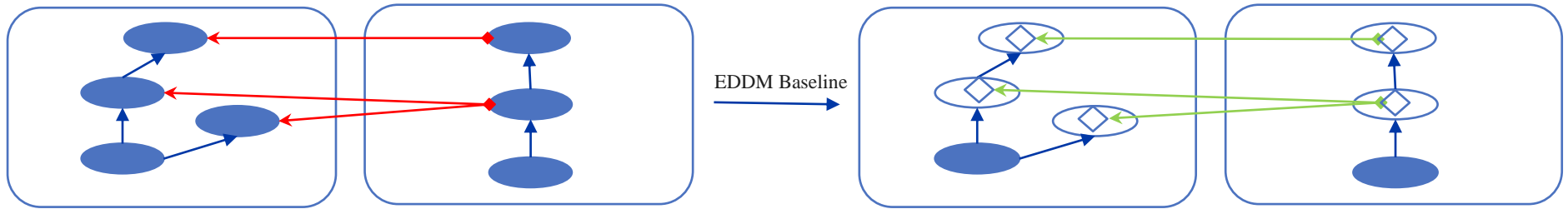
Definition of: EDDM Baseline



- An EDDM Baseline is defined as a named immutable set of EDDM element links
- Before a baseline can be created for a set of EDDM data links all item links in the set must be converted into element links
- The conversion of item links to element links must be handled either automatically by the EDDM server or (manually) by the end user
- An EDDM baseline cannot be changed after creation. I.e. all element links in the baseline are immutable and no links can be added or removed from the baseline. That must be ensured by the EDDM server where the baseline is stored
- Each EDDM baseline should be stored (potentially redundantly) in the Central Link Repository
- Design option: introduce new link type “baseline link”. A baseline link is immutable. A baseline link is a subtype of an element link

Enterprise data dependency management

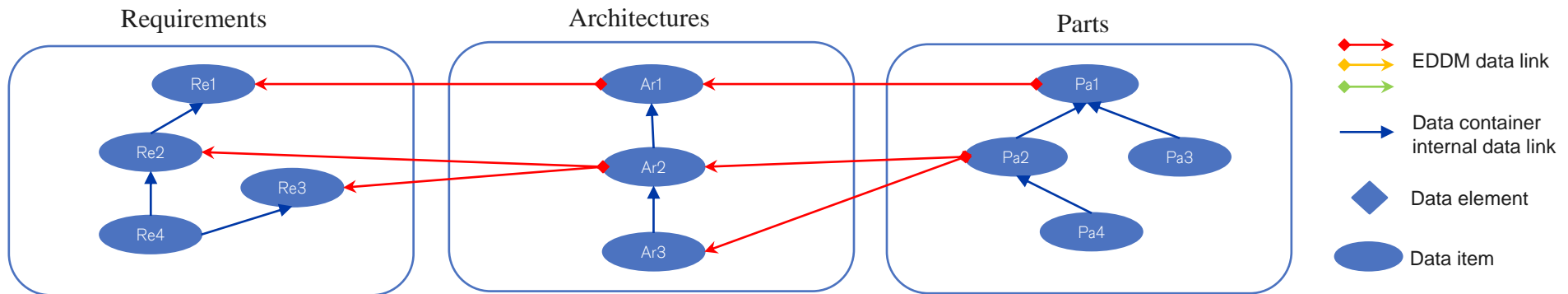
EDDM Baseline – requirements for data handling



- Links that are part of an EDDM baseline cannot be deleted → each EDDM server must know about all defined baselines in order to prevent link deletion → all EDDM baselines should be stored in the central link repository. Note: it is sufficient to prevent changes/deletions of links that are part of a baseline (that will guarantee that all defined baselines are immutable)
- It might be an idea to store a baseline indicator together with the EDDM link, the central EDDM server might offer services to list for a given link all baselines where the link is an element of
- Other possible approach: when a baseline is created copies of all links are stored directly in the central link repository (redundant storage in local systems is possible; all local systems must ensure the immutability of the baseline links)

Enterprise data dependency management (EDDM)

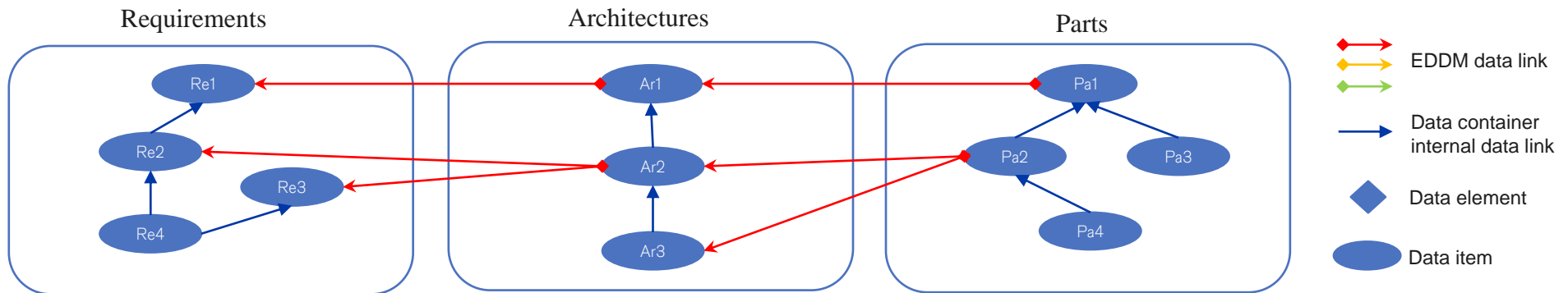
Definition of: Data Dependency Graph (DDG) for a data set



- A data set is defined as a set of data items and data elements
- The Data Dependency Graph (DDG) for a data set is defined as the graph consisting of all data elements, data items and their connecting EDDM links, that can be reached by recursively traversing all EDDM links starting in the data set
- Examples:
 - Data set = {Pa1} has the DDG = {Pa1 → Ar1, Ar1 → Re1}
 - Data set = {Ar2} has the DDG = {Ar2 → Re2, Ar2 → Re3}
 - Data set = {Pa1, Pa2} has the DDG = {Pa1 → Ar1, Pa2 → Ar2, Pa2 → Ar3, Ar1 → Re1, Ar2 → Re2, Ar2 → Re3}
 - Data set = {Re1, Re2} has the DDG = {} (the empty set)

Enterprise data dependency management (EDDM)

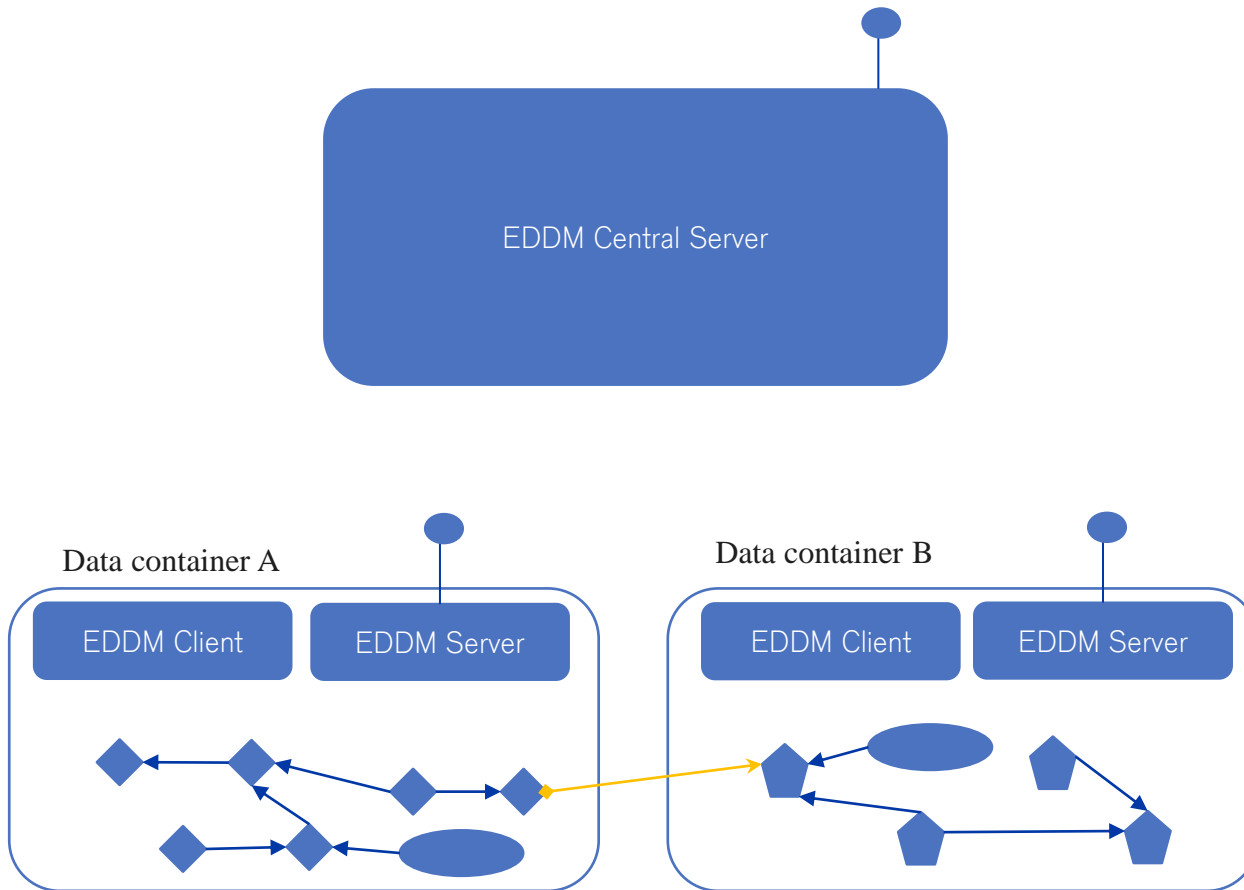
DDG – requirements for data handling



- The DDG is a core tool to determine the EDDM links to include in an EDDM baseline
- Example of part release: the end user will select in the Parts data container all parts to include in the part release. The DDG of these parts will be used for the EDDM baseline associated with the part release
- Calculation of the DDG can be efficiently implemented based on the Central Link Repository

Enterprise data dependency management (EDDM)

Solution architecture



EDDM Client:

- Embedded in the GUI of each data container
- Enables data link create/update/delete

EDDM Server:

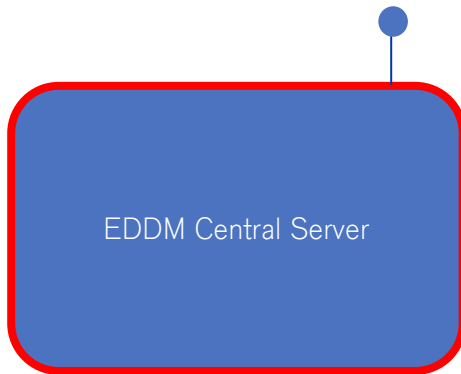
- Can be embedded in a data container or runs standalone
- Manages all EDDM queries for the data container
- Manages all data operations on data links for the data container
- Provides delegated UI's for data elements, items and data links

EDDM Central Server:

- Enables EDDM services discovery
- Translation from/to symbolic EDDM data links into resolvable URL's

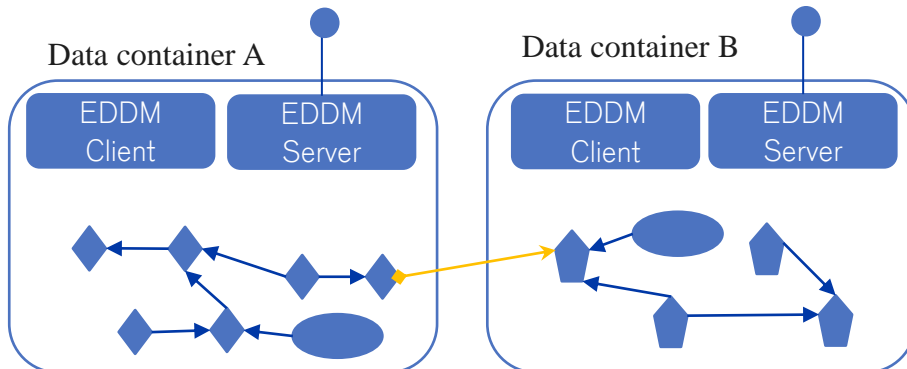
Enterprise data dependency management (EDDM)

Capabilities EDDM Central Server



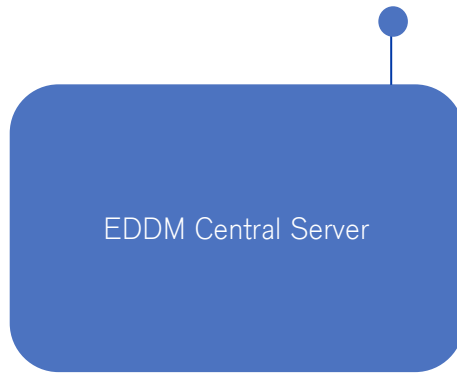
The EDDM Central Server enables discovery of available EDDM services and translates between symbolic and resolvable links

1. Which data element types and data item types are available via EDDM?
2. Which EDDM servers are available for each data element type and data item type?
3. In which EDDM servers can data links with given endpoint types be persisted?
4. Implement translation of symbolic EDDM data links into resolvable URL links and vice versa

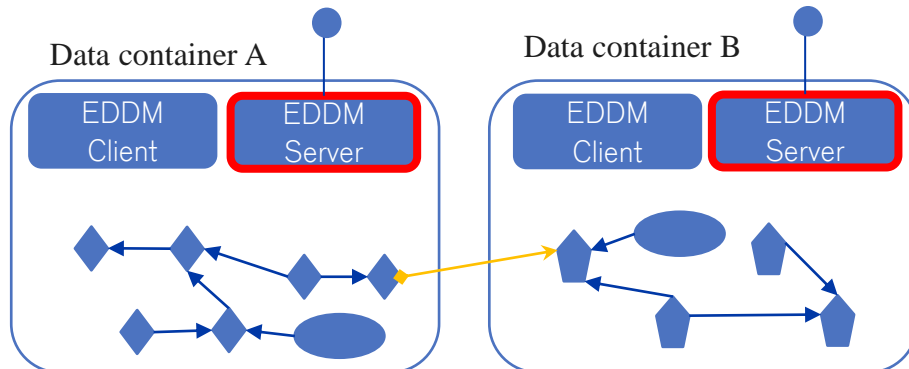


Enterprise data dependency management (EDDM)

Capabilities EDDM Server

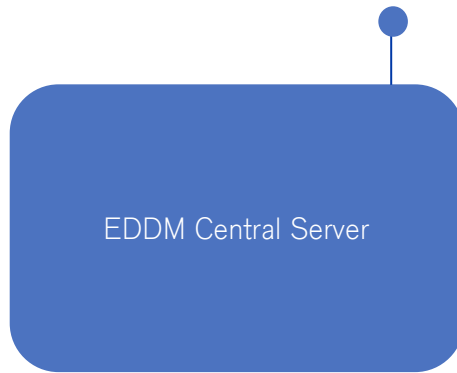


1. Authentication of incoming calls
2. Authorization of incoming calls
3. Handle translation of symbolic EDDM data links into resolvable URL links and vice versa
4. Query for data elements
5. Query for data items
6. Query for data links
7. Create/update/delete of data links
8. Preview UI and delegated dialogs for data elements, data items

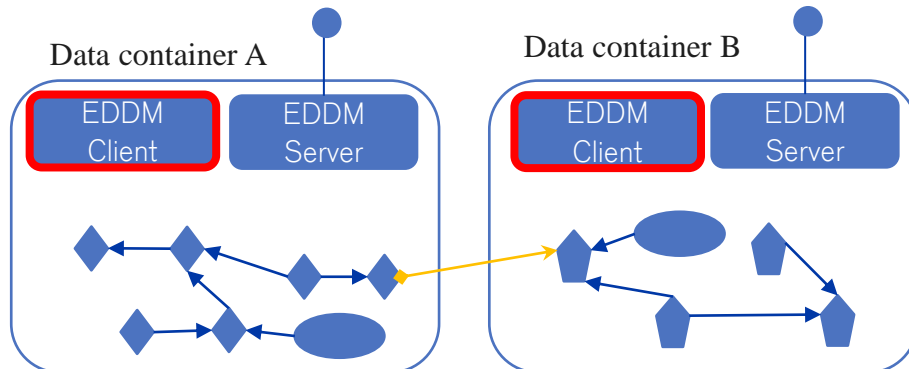


Enterprise data dependency management (EDDM)

Capabilities EDDM Client

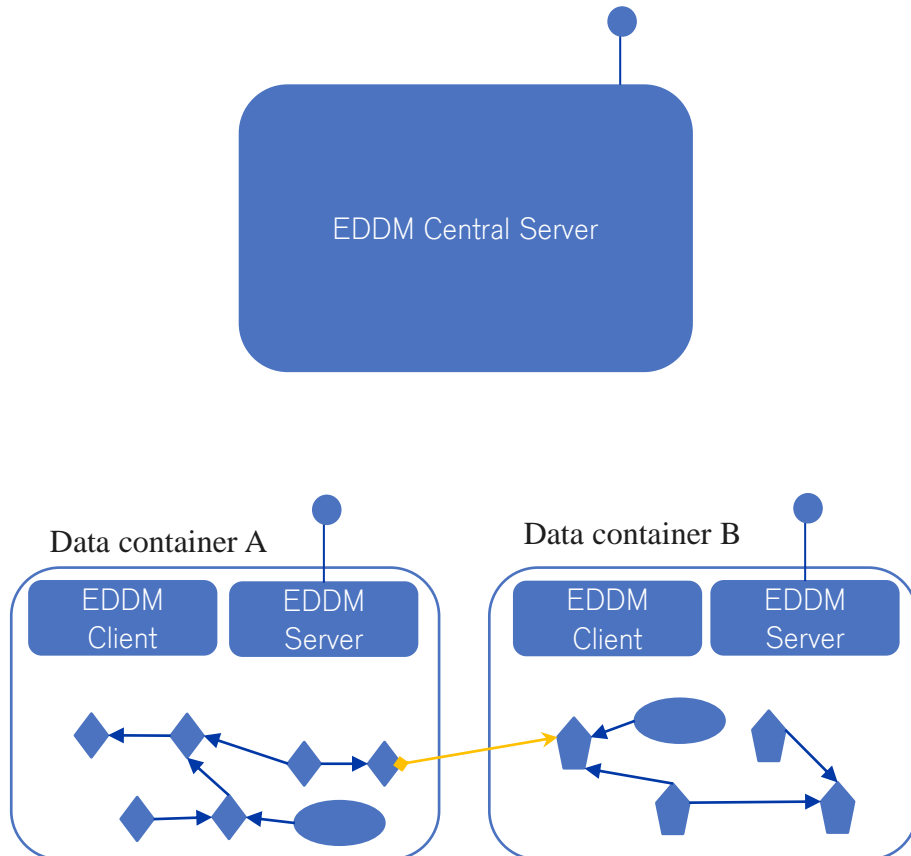


1. End user UI for data link management
2. Query EDDM Central Server for EDDM services discovery
3. Query EDDM servers and display of query results
4. Initiate create/update/delete of data links in EDDM servers



Enterprise data dependency management (EDDM)

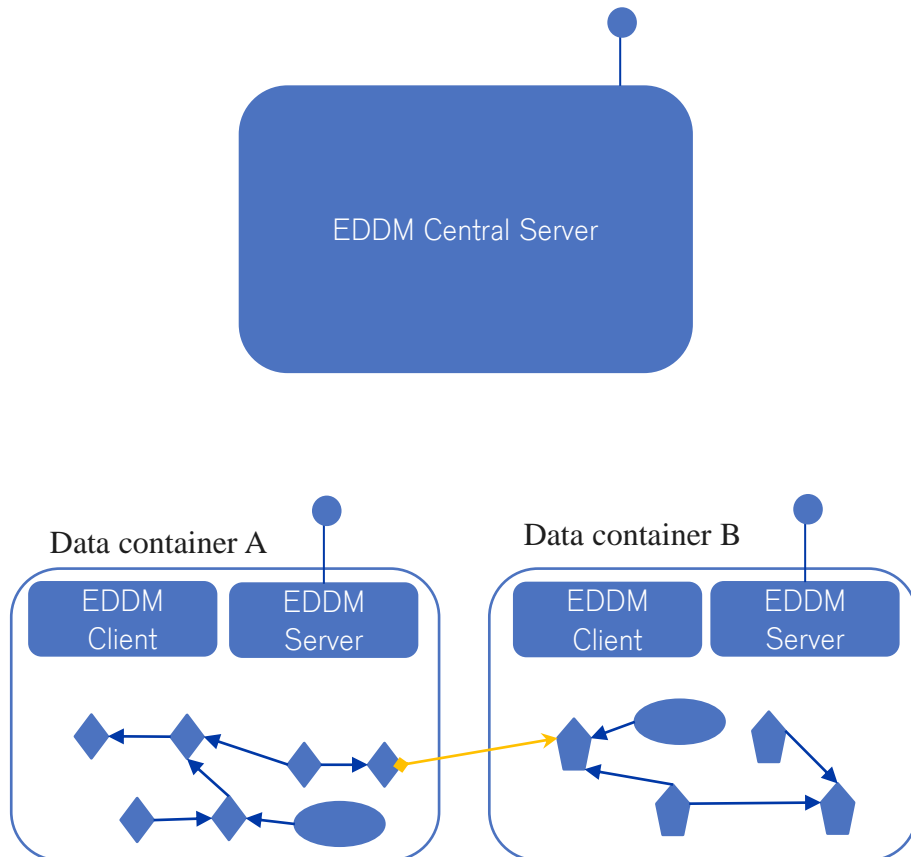
Symbolic EDDM data links and resolvable URL's



1. EDDM must ensure long term stability of EDDM links
2. IP addresses and host names of data containers will change over time
3. Data containers will be exchanged and data elements and data items will be moved to other data containers
4. For the aforementioned reasons EDDM links cannot be stored as resolvable URL's
5. EDDM links will be symbolic links which are transformed upon request into resolvable URL's
6. The EDDM Central Server implements the translation between symbolic EDDM links and resolvable URL's

Enterprise data dependency management (EDDM)

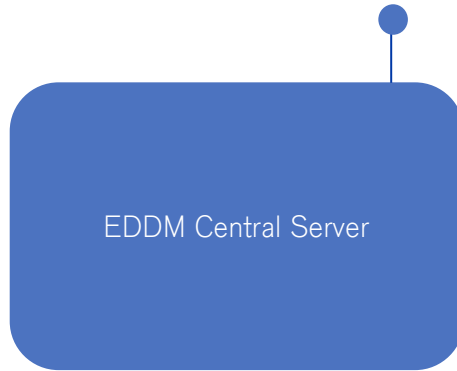
Authentication



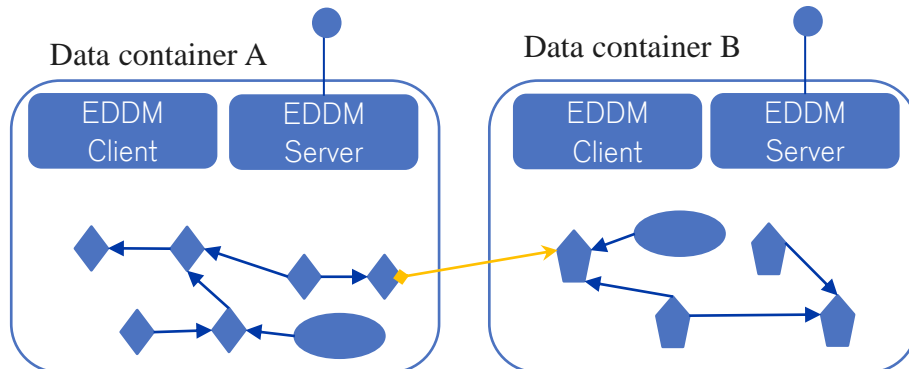
1. Authentication is done against the Dräger Active Directory (AD)
2. In case that a data container has already authenticated a user the authentication token should be used by the EDDM client
3. The EDDM client must pass the authentication token to the EDDM servers and the EDDM Central Server
4. Any actions initiated by an EDDM client by calling an EDDM server should be executed in the authentication context of the passed authentication token

Enterprise data dependency management (EDDM)

Authorization

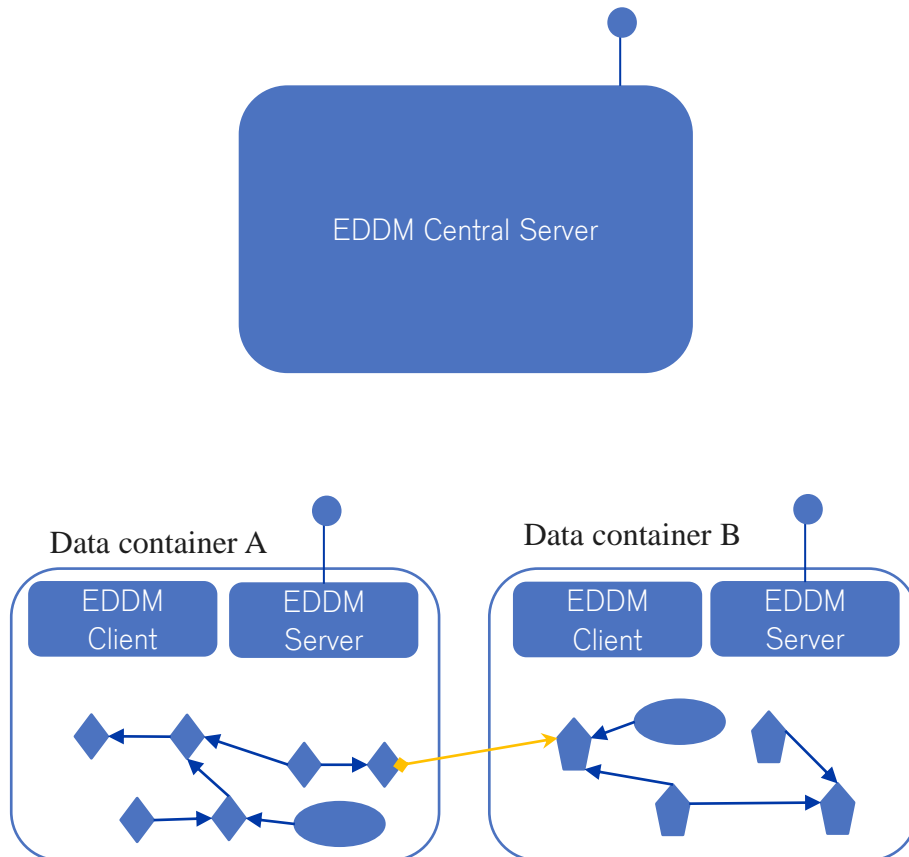


1. Authorization is based on AD group membership of the end user of the EDDM client
2. Any processing on the EDDM Server and the EDDM Central server is done in the authorization context of the end user



Enterprise data dependency management (EDDM)

Use case data link creation

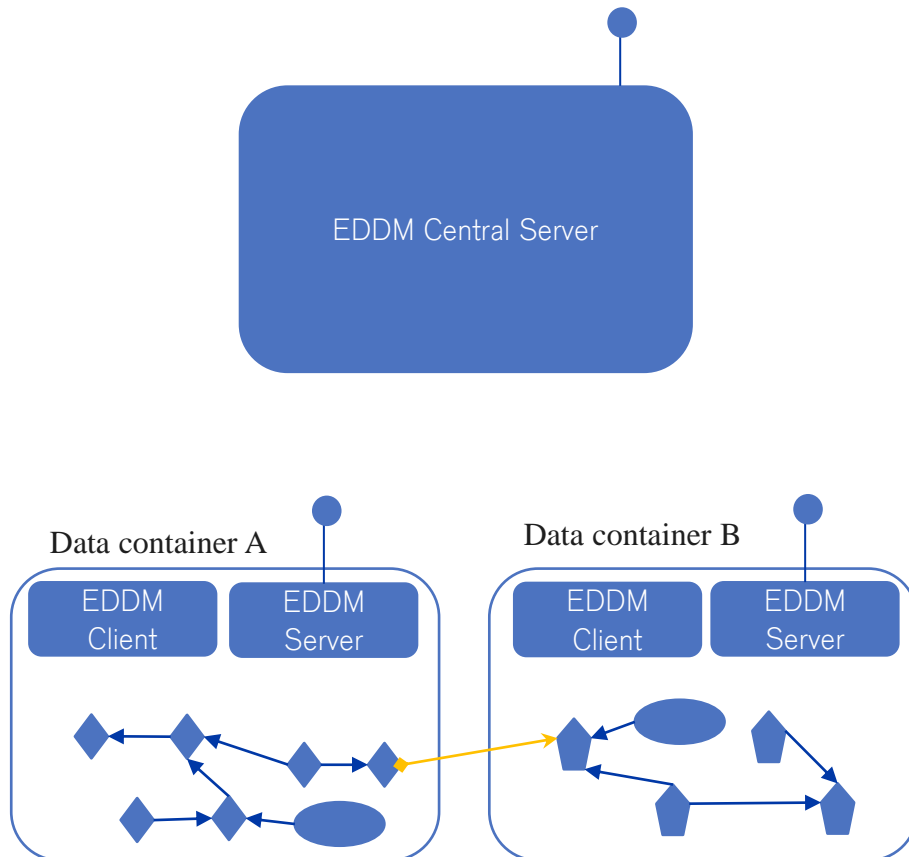


End user working in data container A wants to create a link to a B data element

1. The EDDM-A client sends a query to the EDDM Central Server to figure out where B data elements are available. Answer: in EDDM-B server
2. EDDM-A client sends a query for B data elements to EDDM-B server. Answer: list of B data elements
3. EDDM-A client collects from user input for new link between A and B data elements
4. EDDM-A client sends a query to the EDDM Central Server to figure out where EDDM links between A and B data elements should be persisted. Answer: in EDDM-A server
5. EDDM-A client determines the `change_indicator` of the new data link by using the `change_indicators` of the source A and target B of the new data link
6. EDDM-A client persists new data link in Data container A by calling EDDM-A server

Enterprise data dependency management (EDDM)

Use case data link update

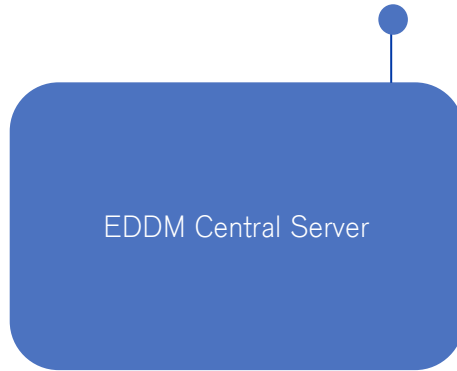


End user working in data container A wants to update an existing data link between A and B data items. She wants to change the data link target to another data element B

1. The EDDM-A client sends a query to the EDDM Central Server to figure out where B data elements are available. Answer: in EDDM-B server
2. EDDM-A client sends a query for B data elements to EDDM-B server. Answer: list of B data elements
3. EDDM-A client collects from user input for data link target update to new B data element
4. EDDM-A client sends a query to the EDDM Central Server to figure out where EDDM links between A and B data elements should be persisted. Answer: in EDDM-A server
5. EDDM-A client determines the `change_indicator` of the updated data link by using the `change_indicators` of the source A and target B of the updated data link
6. EDDM-A client persists new data link in Data container A by calling EDDM-A server

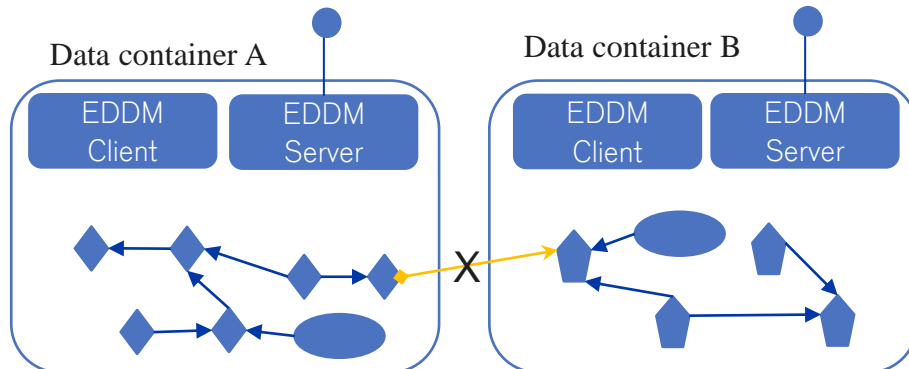
Enterprise data dependency management (EDDM)

Use case data link deletion



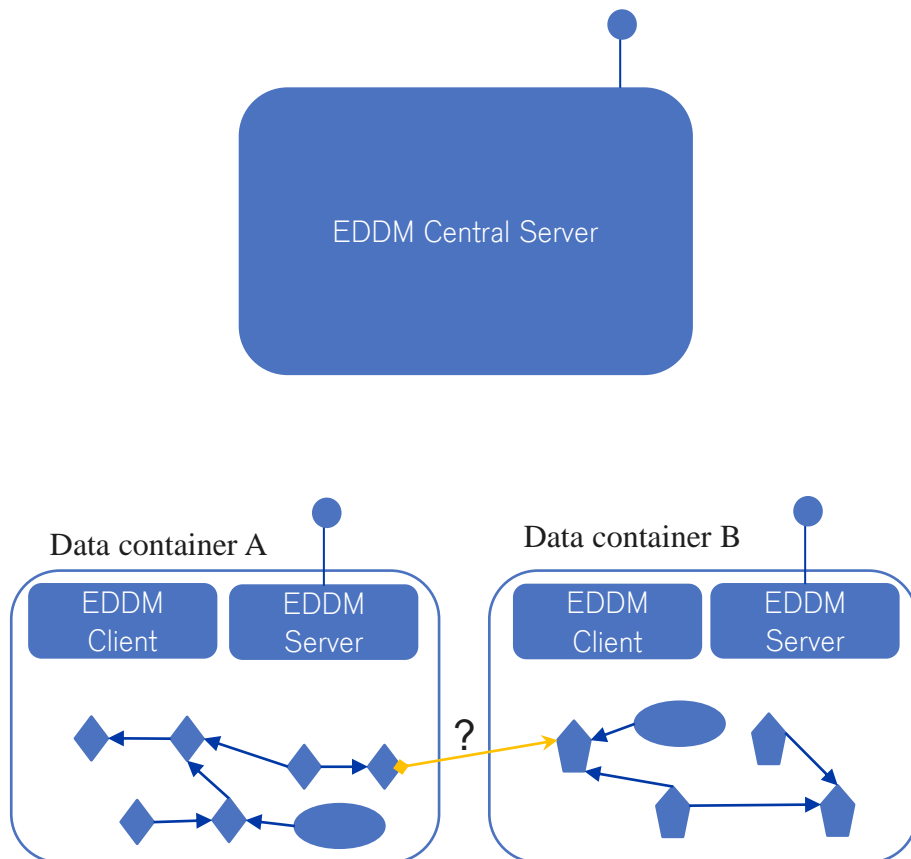
End user working in data container A wants to delete an existing data link between A and B data items

1. EDDM-A client sends a query to the EDDM Central Server to figure out where EDDM links between A and B data elements should be persisted. Answer: in EDDM-A server
2. EDDM-A client deletes existing data link in Data container A by calling EDDM-A server



Enterprise data dependency management (EDDM)

Use case suspect data link detection

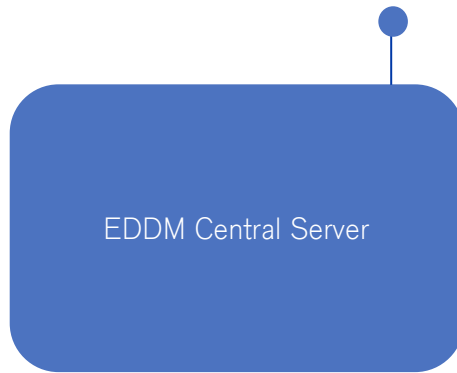


End user working in data container A wants to check the suspect status of an existing data link between A and B data items (could also be an automated check of the EDDM client)

1. The EDDM-A client sends queries to the EDDM-A and EDDM-B servers to get the change indicators of A and B. Answer: change indicators of A and B data items (data elements are immutable and have no change indicator)
2. The EDDM-A client compares the change indicators of A and B with the change indicator of the data link. If these change indicators do not match the data link is suspect
3. EDDM-A client shows suspect information of the data link in the EDDM-A client GUI

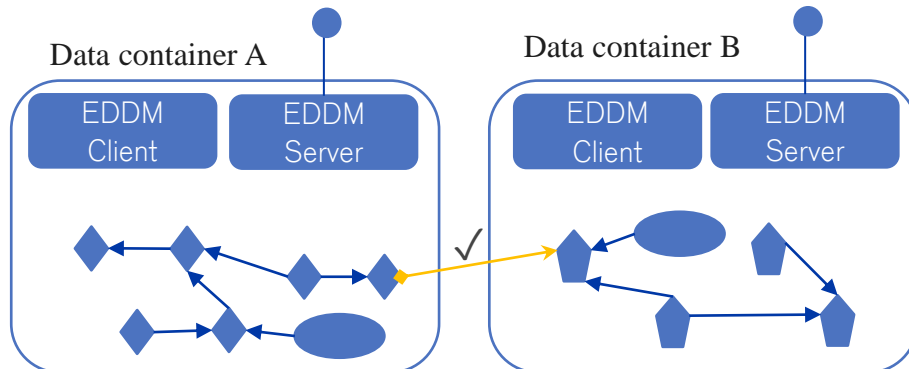
Enterprise data dependency management (EDDM)

Use case suspect data link resolution



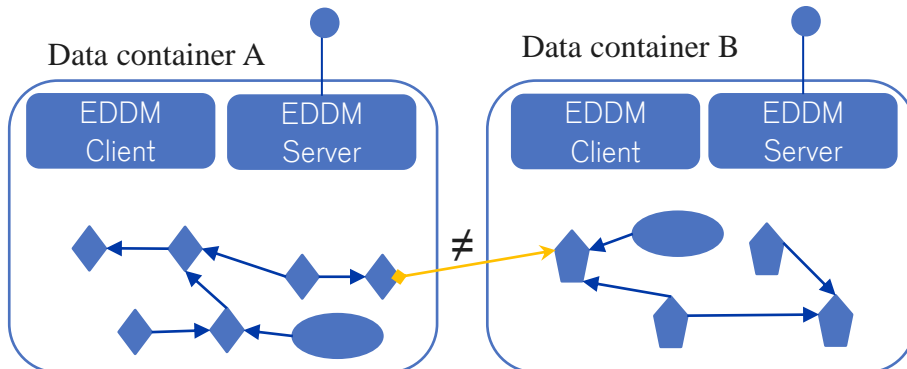
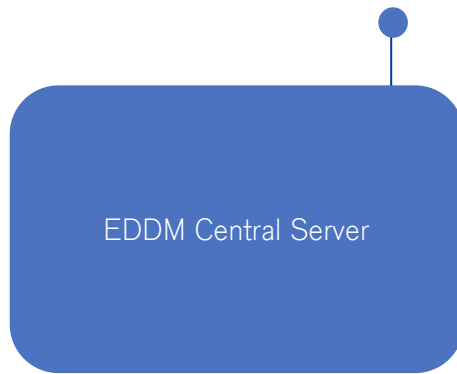
End user working in data container A wants to resolve the suspect status of an existing suspect data link between A and B data items. She inspects both ends of the link and decides that the link is valid

1. EDDM-A client updates in data container A the `change_indicator` of the data link to match the change indicators of source and target of the link by calling EDDM-A server



Enterprise data dependency management (EDDM)

Use case dangling data link detection



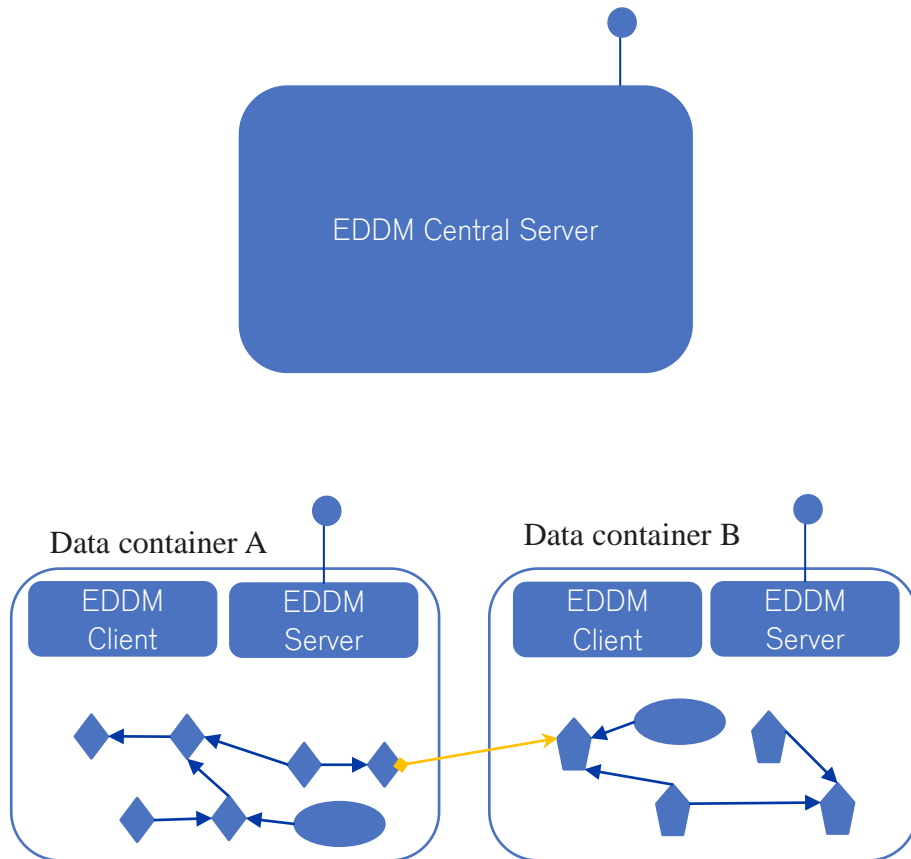
End user working in data container A wants to check the validity of an existing data link between A and B data elements (could also be an automated check of the EDDM client)

1. The EDDM-A client sends queries to the EDDM-A and EDDM-B servers to figure out whether the A and B data elements are valid (exist). Answer: validity (existence) information of A and B data elements
2. EDDM-A client shows validity/dangling link information of the data link in the EDDM client GUI

Note: dangling data links are resolved by either deleting the data link or updating it to valid source and target data items/elements

Enterprise data dependency management (EDDM)

Use case resource preview and delegated dialog

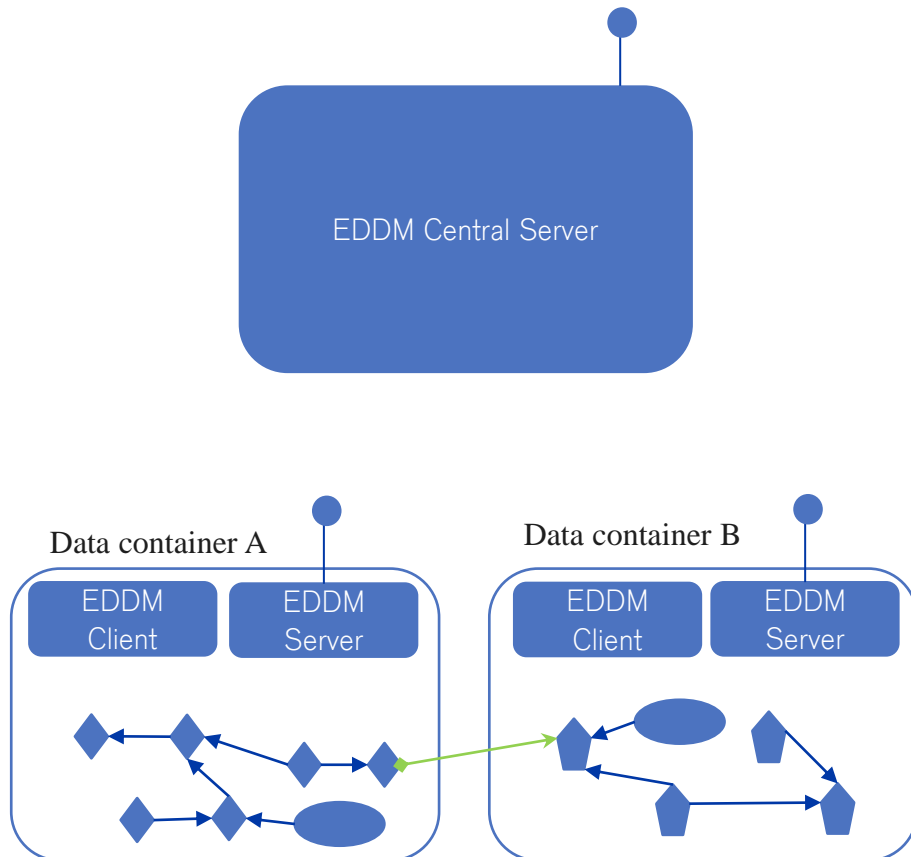


End user working in data container A wants to access an existing data link from A to B data element

1. End user hovers with her mouse over the data link
2. EDDM-A client sends request to EDDM Central Server to convert the stored symbolic EDDM link into a resolvable URL. Answer: resolvable URL
3. EDDM-A client sends a query to EDDM-B server for B preview UI. Answer: B resource preview UI
4. EDDM-A client shows B resource preview UI
5. End user presses B target in data link
6. EDDM-A client sends request to EDDM Central Server to convert the stored symbolic EDDM link into a resolvable URL. Answer: resolvable URL
7. EDDM-A client sends a query to EDDM-B server for B delegated dialog. Answer: B delegated dialog
8. EDDM-A client shows B delegated dialog

Enterprise data dependency management (EDDM)

Use case EDDM baseline creation

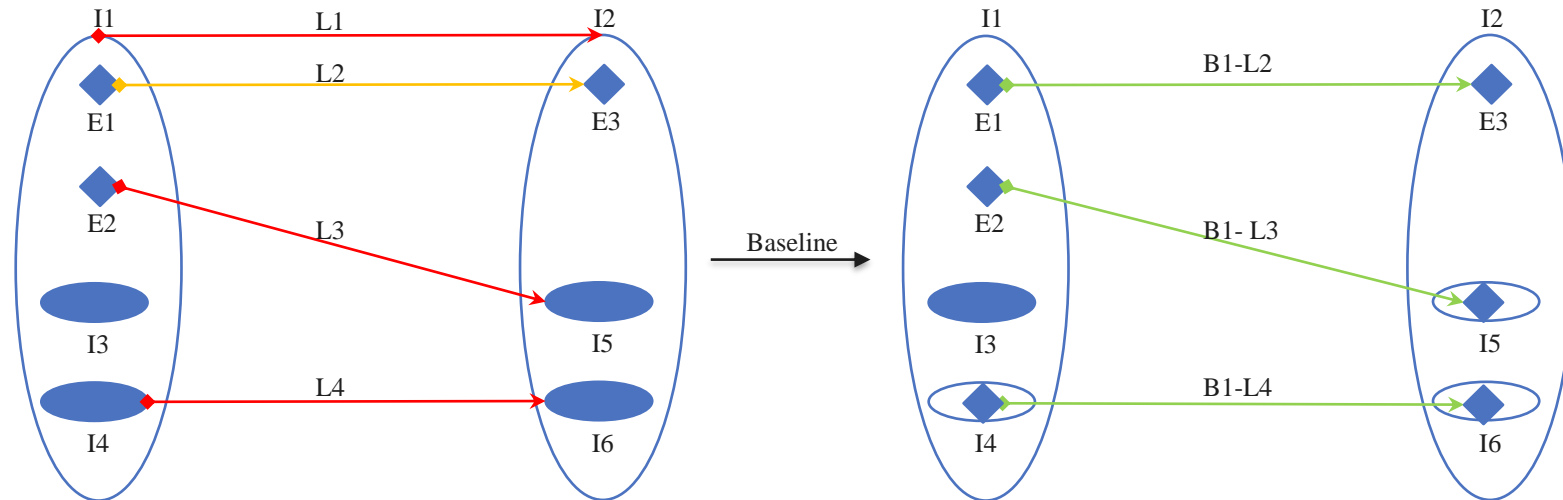


End user working in data container A wants to create an EDDM baseline

1. End user formulates search criteria for the links that should be included in the new EDDM baseline
2. EDDM-A client sends link queries to all EDDM servers (alternatively the query is only sent to the Central Link Repository). Answer: list of EDDM links
3. End user selects from the returned link set all links that should be included in the new EDDM baseline
4. End user converts in the link list all item links into element links (that could be automated by the EDDM servers) and defines a name for the new EDDM baseline
5. EDDM-A client sends a query to the EDDM Central Server to figure out where the new baseline should be persisted. Answer: in EDDM-A server
6. EDDM-A client persists new EDDM baseline in Data container A by calling EDDM-A server
7. Optional: EDDM-A clients persists new EDDM baseline in Central Link Repository

Enterprise data dependency management (EDDM)

Detailed example for EDDM baseline creation



- Step 1: user selects links for which a baseline should be created: {L2, L3, L4} and chooses B1 as name for the new baseline
- Step 2: New baseline link B1-L2 is created out of link L2
- Step 3: Data element from data item I5 is selected which will serve as link target for the baseline link B1-L3
- Step 4: New baseline link B1-L3 is created out of link L3
- Step 5: Data elements from data item I4 and I6 are selected which will serve as link source and link target for the baseline link B1-L4
- Step 6: New baseline link B1-L4 is created out of link L4
- Step 7: All baseline links {B1-L2, B1-L3, B1-L4} are stored together with the baseline name B1 in the central link repository
- Step 8 (optional): All baseline links {B1-L2, B1-L3, B1-L4} are stored together with the baseline name B1 in a local EDDM server

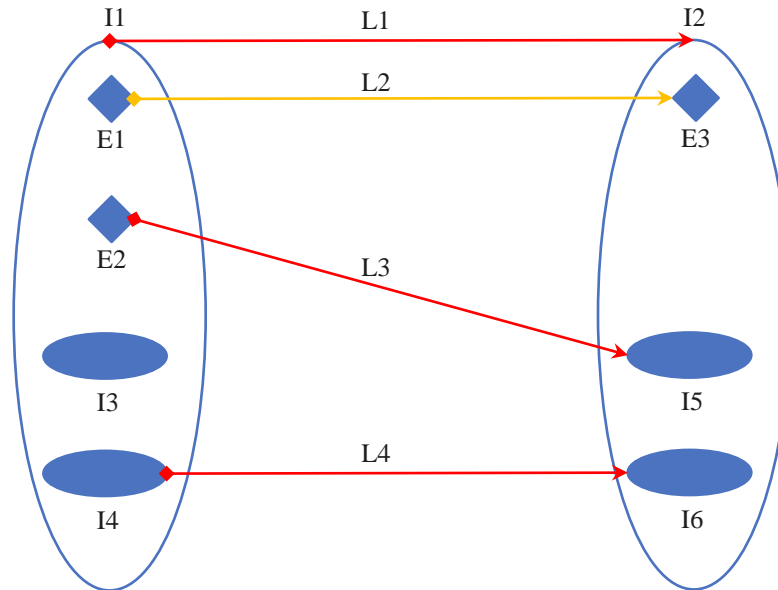
Enterprise data dependency management

EDDM change management

- Only data items, element links and item links are subject to change
 - Data items change by adding or removing data elements: all links involving the changed data item go to the status Suspect
 - Element links and item links that are not part of a baseline can be created, modified, deleted
 - Changing an element link which is part of a baseline is defined as the creation of a new element link with the same endpoints
- Possibilities for link transfer when a new data element is added to a data item
 - All element links are inherited from the “previous/source” data element
 - No data element links are inherited
- When a data item is changed (by adding a data element) all links containing this data item (as link source or target) will go to status Suspect

Enterprise data dependency management (EDDM)

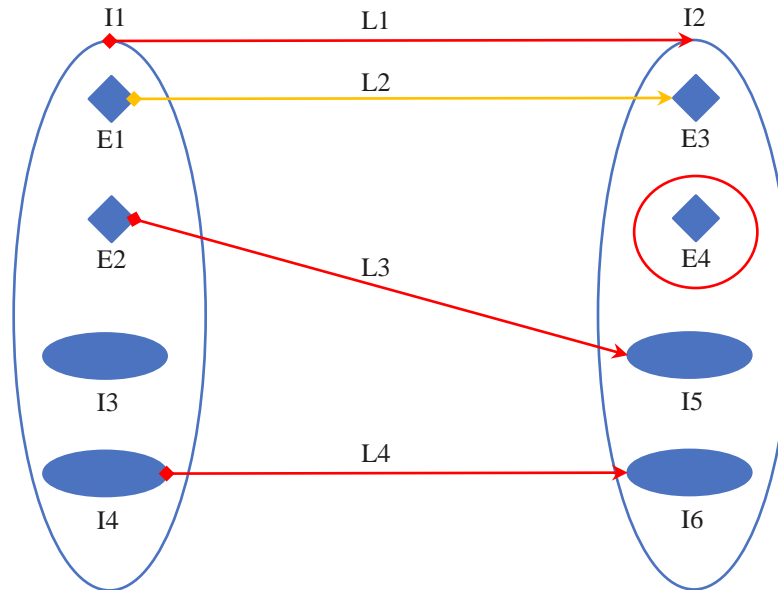
EDDM change management – overview change use cases



- Use case 1: new data element E4 is added to data item I2
- Use case 2: Change indicator of data item I3 is updated
- Use case 3: Change indicator of data item I4 is updated
- Use case 4: Data item I5 is removed from data item I2
- Use case 5: Data item I6 is deleted
- Use case 6: Item link L3 is deleted

Enterprise data dependency management (EDDM)

EDDM change management – Use case 1

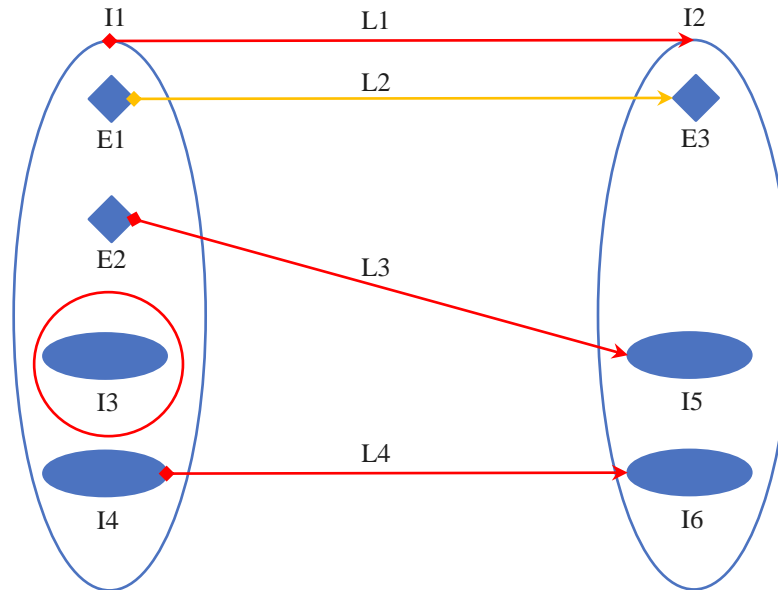


Use case 1: new data element E4 is added to data item I2

- Change indicator of data item I2 is updated
- Link status of item link L1 changes to Suspect

Enterprise data dependency management (EDDM)

EDDM change management – Use case 2

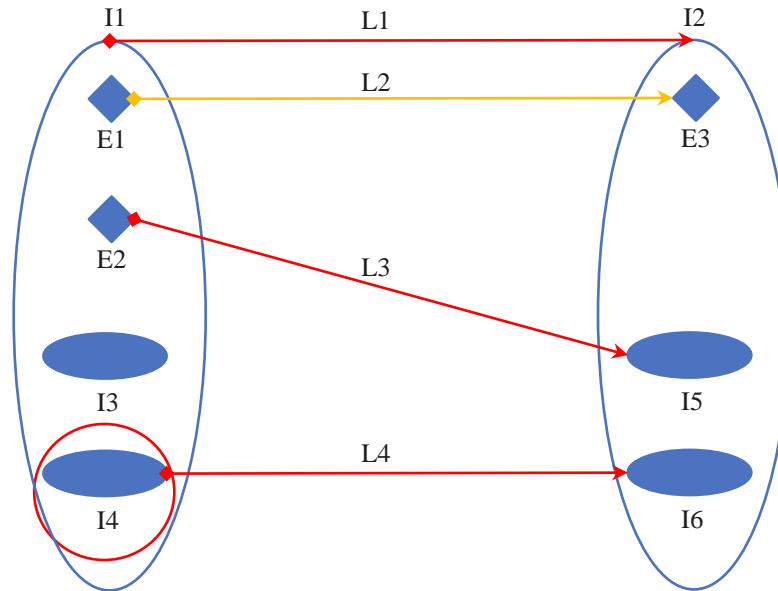


Use case 2: Change indicator of data item I3 is updated

– No further effects (change indicator of data item I1 is not updated)

Enterprise data dependency management (EDDM)

EDDM change management – Use case 3

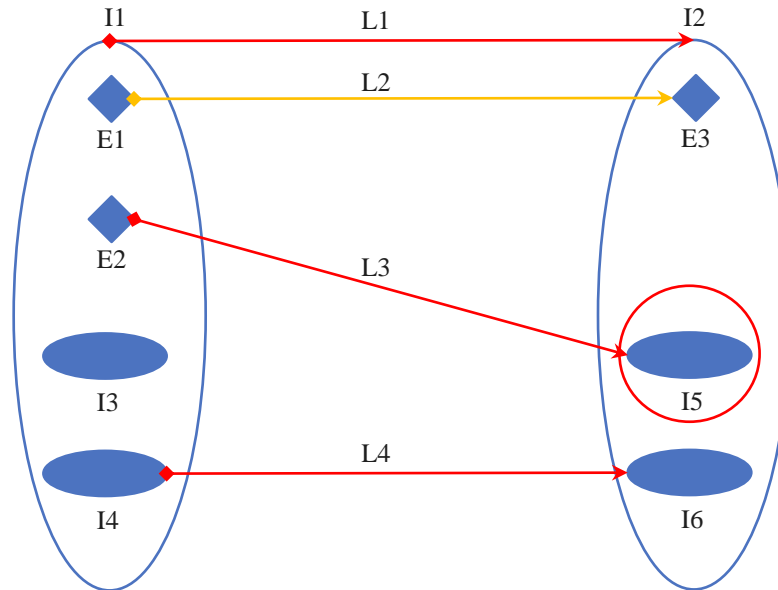


Use case 3: Change indicator of data item I4 is updated

- Link status of item link L4 changes to Suspect
- Change indicator of item link L4 is updated
- Change indicator of data item I1 is not updated

Enterprise data dependency management (EDDM)

EDDM change management – Use case 4

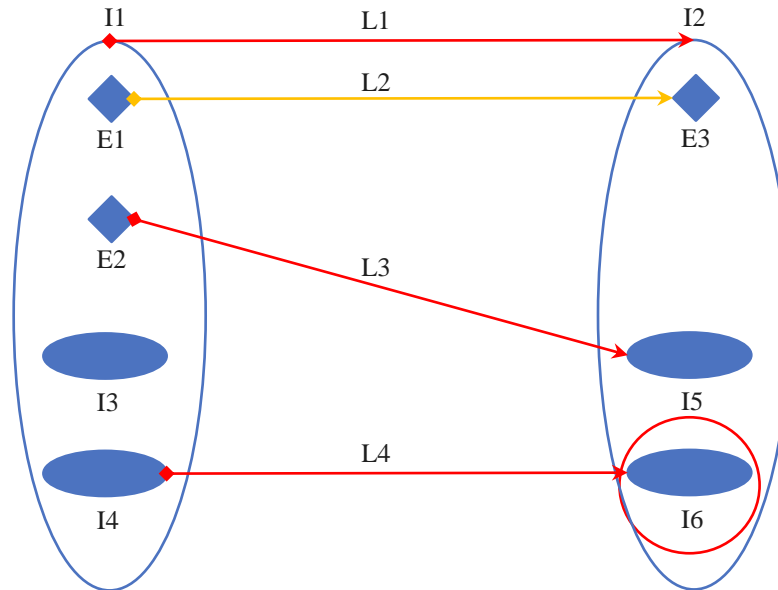


Use case 4: Data item I5 is removed from data item I2 (I5 is not deleted)

- Change indicator of data item I2 is updated
- Link status of item link L1 changes to Suspect
- Change indicator of item link L1 is updated
- Link status of item link L3 is not updated (I5 is not deleted and L3 remains unchanged)

Enterprise data dependency management (EDDM)

EDDM change management – Use case 5

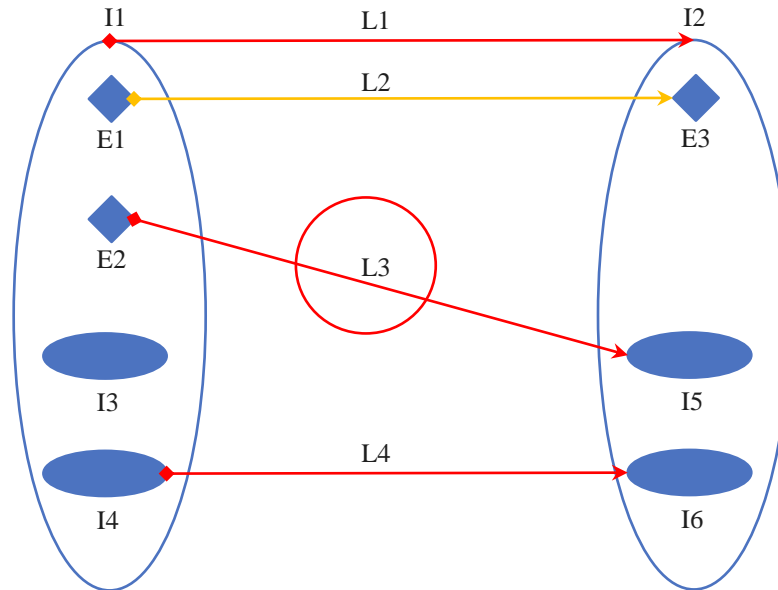


Use case 5: Data item I6 is deleted

- Change indicator of data item I2 is updated
- Change indicator of item link L4 is updated
- Link status of item link L1 changes to Suspect
- Change indicator of item link L1 is updated
- Link status of item link L4 changes to Dangling

Enterprise data dependency management (EDDM)

EDDM change management – Use case 6

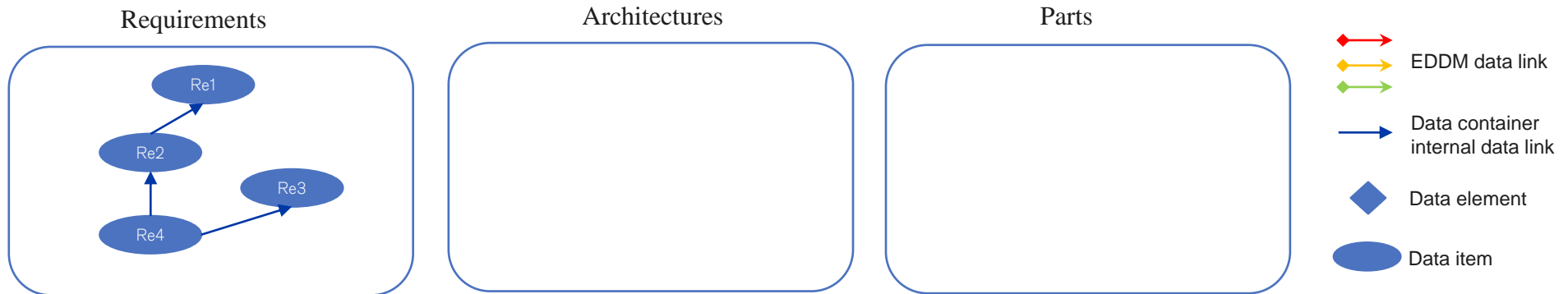


Use case 6: Item link L3 is deleted

– No further effects (change indicators of data items I1, I2 and I5 are not updated)

Enterprise data dependency management (EDDM)

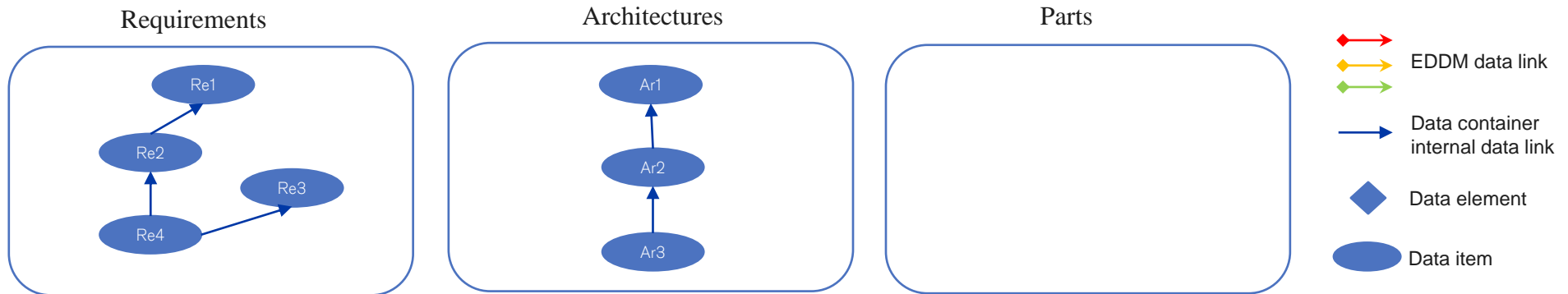
A comprehensive use case – Step 1: Creation of requirements



- Requirements and their mutual dependencies are created in the Requirements data container.
- For the requirements structure data container internal links are used
- The top level requirement Re1 is the head of the requirements structure. It will only be Released when all sub-requirements are Released

Enterprise data dependency management (EDDM)

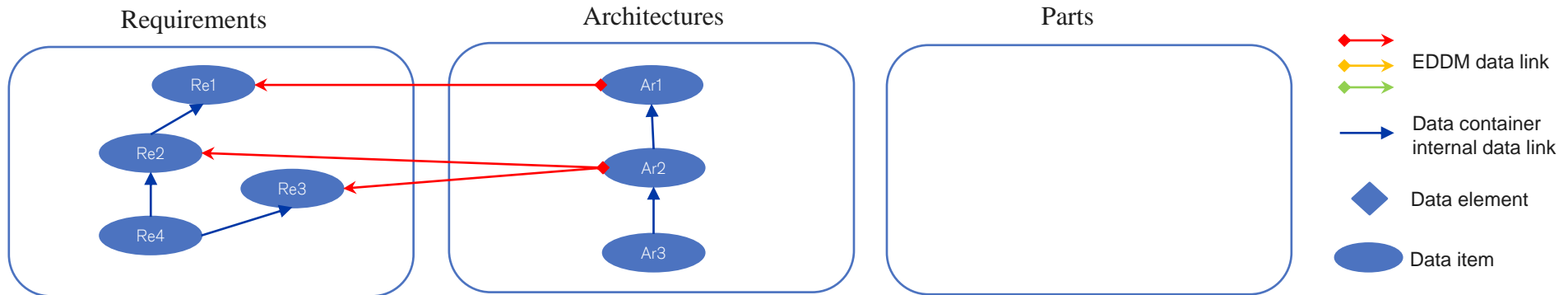
A comprehensive use case – Step 2: Creation of architectures



- Architectures and their mutual dependencies are created in the Architectures data container
- For the architectures structure data container internal links are used
- The top level architecture Ar1 is the head of the architectures structure. It will only be Released when all sub-architectures are Released

Enterprise data dependency management (EDDM)

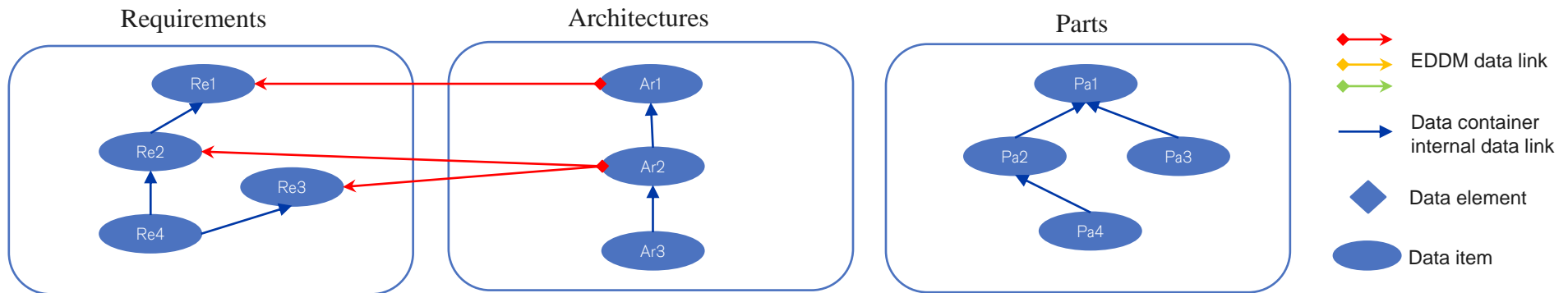
A comprehensive use case – Step 3: Links from architectures to requirements



- Work in the Architectures data container starts with the creation of the head of the architecture structure, Ar1. When Ar1 is created an EDDM item link to the head of the requirements structure, Re1, is created
- The architecture Ar2 describes a subsystem architecture which depends on the requirements Re2, Re3. For these dependencies EDDM item links are created
- It should be noted that EDDM links are only created between core elements of the architectures and requirements structures. It is up to the end users to decide which EDDM links are needed for the data dependency management between architectures and requirements

Enterprise data dependency management (EDDM)

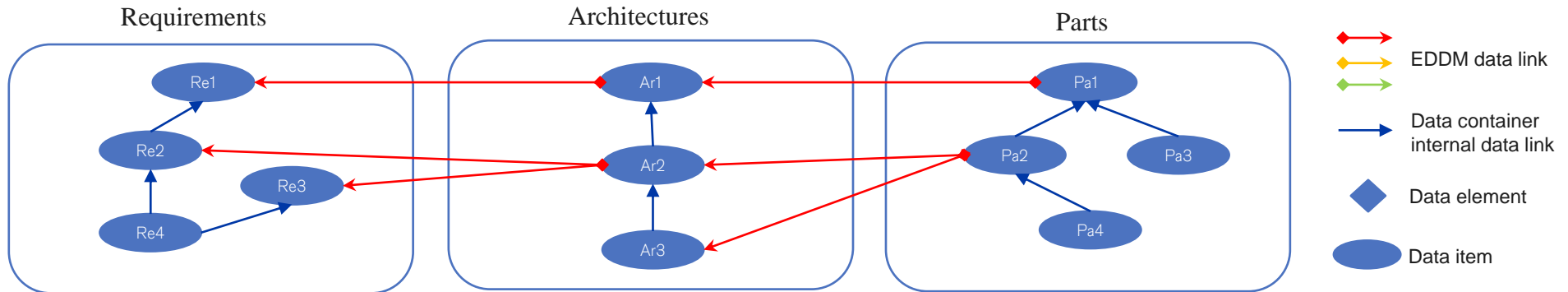
A comprehensive use case – Step 4: Creation of parts



- Parts and their mutual dependencies are created in the Parts data container
- For the parts structure data container internal links are used
- The top level part Pa1 is the head of the parts structure. It will only be Released when all sub-parts are Released

Enterprise data dependency management (EDDM)

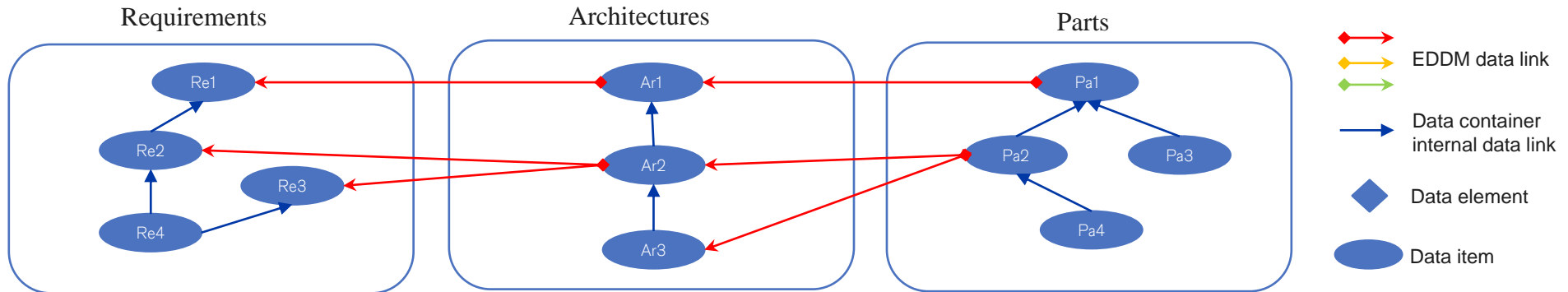
A comprehensive use case – Step 5: Links from parts to architectures



- Work in the Parts data container starts with the creation of the head of the parts structure, Pa1. When Pa1 is created an EDDM item link to the head of the architectures structure, Ar1, is created
- The part Pa2 depends on the architectures Ar2, Ar3. For these dependencies EDDM item links are created
- It should be noted that EDDM links are only created between core elements of the parts and architectures structures. It is up to the end users to decide which EDDM links are needed for the data dependency management between parts and architectures

Enterprise data dependency management (EDDM)

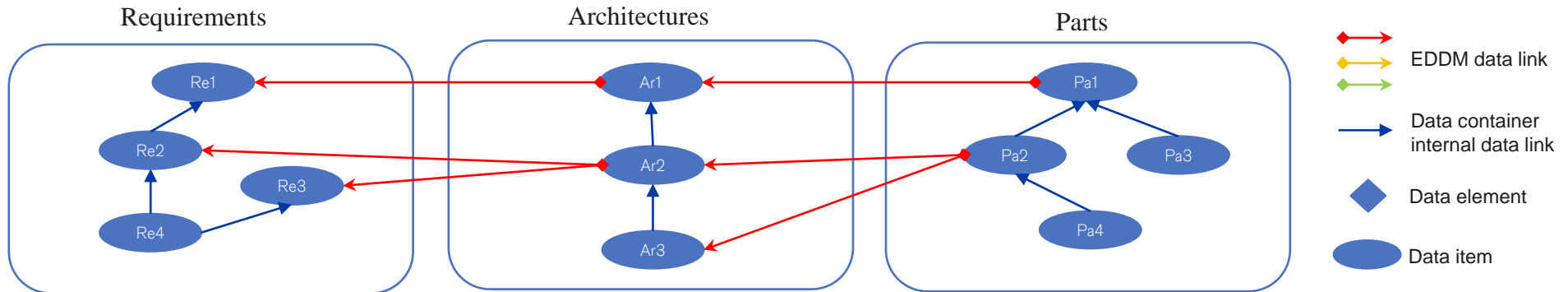
A comprehensive use case – Step 6: Link handling during on-going development



- During on-going development work all data items can be changed concurrently
- Whenever a data item is updated all EDDM links that involve the updated data item will go to the status Suspect
- Example: when architecture element Ar2 is updated after all shown links have been created then the three item links Pa2→Ar2, Ar2→Re2, Ar2→Re3 go to Suspect and the links change indicators are updated
- The end user has to decide how to update the link status (either set it to Valid or leave it on Suspect)

Enterprise data dependency management (EDDM)

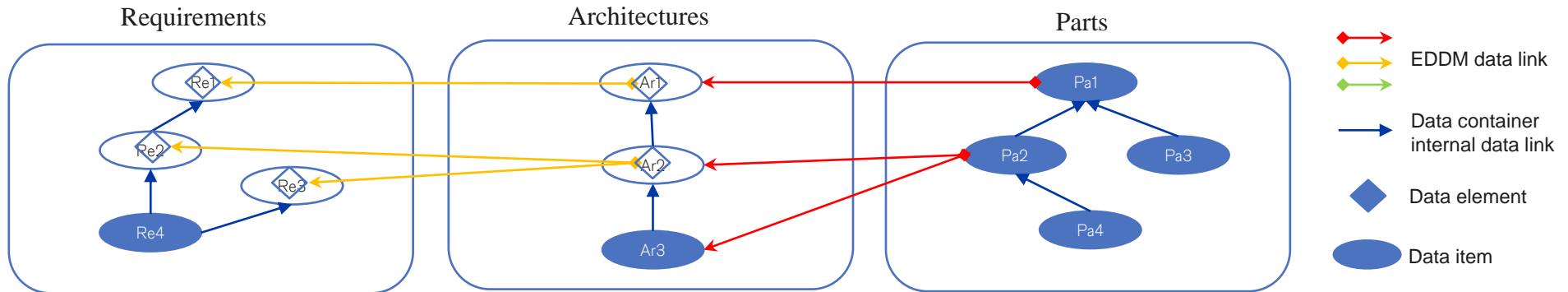
A comprehensive use case – Step 7: Requirements release



- Requirements release is usually done bottom-up in the dependency hierarchy: Re4, Re3, Re2, Re1 are released in this sequence. All data container internal links of the Requirements data container are frozen as part of the release process
- After the release each requirements data item will contain a new data element and the change indicator of all data items will be updated
- The item links Ar1→Re1, Ar2→Re2, Ar2→Re3 will all have the status Suspect and the links change indicators are updated

Enterprise data dependency management (EDDM)

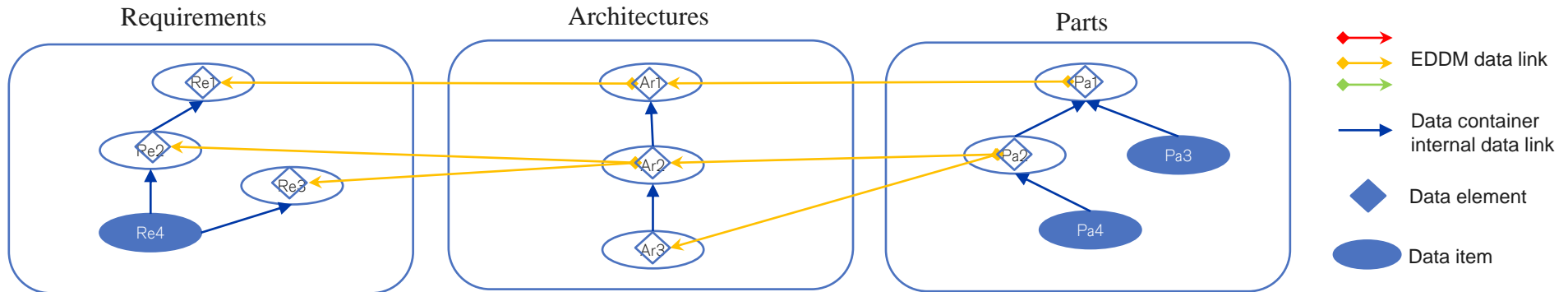
A comprehensive use case – Step 8: Architectures release



- Architectures release is usually done bottom-up in the dependency hierarchy: Ar3, Ar2, Ar1 are released in this sequence. All data container internal links of the Architectures data container are frozen as part of the release process
- After the release each architectures data item will contain a new data element and the change indicator of all data items will be updated
- All outbound EDDM links from Architectures to Requirements are converted from item links to element links. That is done by replacing each item link source and target with the recently created released data elements. Example: the item link $Ar1 \rightarrow Re1$ is converted into an element link by using the most recent released Ar1 data element as link source and the most recent released Re1 data element as link target
- The element links $Ar1 \rightarrow Re1$, $Ar2 \rightarrow Re2$, $Ar2 \rightarrow Re3$ all have the status Valid and the links change indicators are updated
- The item links $Pa1 \rightarrow Ar1$, $Pa2 \rightarrow Ar2$, $Pa2 \rightarrow Ar3$ all have the status Suspect and the links change indicators are updated

Enterprise data dependency management (EDDM)

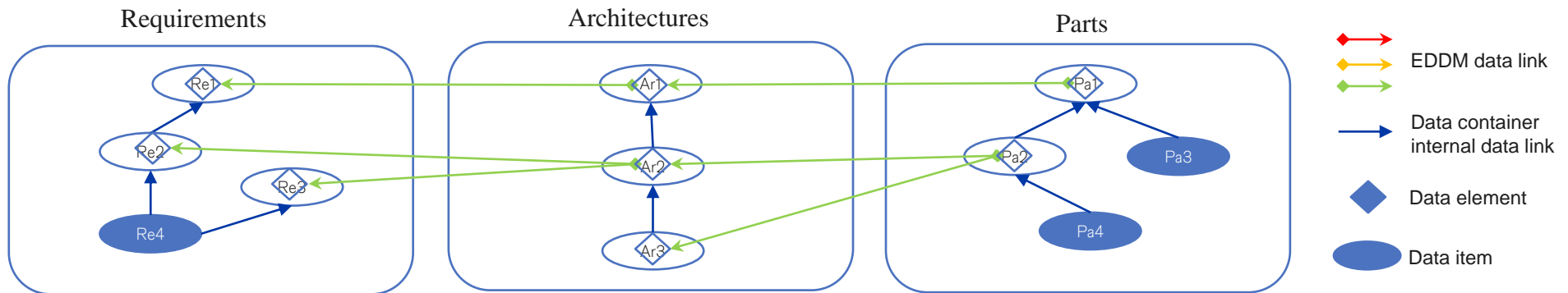
A comprehensive use case – Step 9: Parts release



- Parts release is usually done bottom-up in the dependency hierarchy: Pa4, Pa3, Pa2, Pa1 are released in this sequence. All data container internal links of the Parts data container are frozen as part of the release process
- After the release each parts data item will contain a new data element and the change indicator of all data items will be updated
- All outbound EDDM links from Parts to Architectures are converted from item links to element links. That is done by replacing each item link source and target with the recently created released data elements. Example: the item link Pa1→Ar1 is converted into an element link by using the most recent released Pa1 data element as link source and the most recent released Ar1 data element as link target
- The element links Pa1→Ar1, Pa2→Ar2, Pa2→Ar3 all have the status Valid and the links change indicators are updated

Enterprise data dependency management (EDDM)

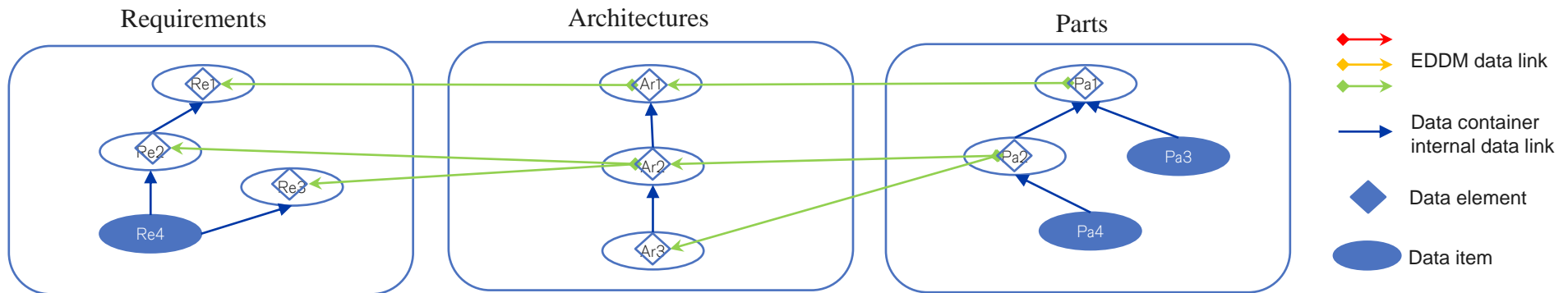
A comprehensive use case – Step 10: EDDM baseline



- An end user starts via an EDDM client the creation of an EDDM baseline
- She selects the element links to be included in the EDDM baseline: the DDG (data dependency graph) of the data set {Pa1, Pa2}: that is $Pa1 \rightarrow Ar1$, $Pa2 \rightarrow Ar2$, $Pa2 \rightarrow Ar3$, $Ar1 \rightarrow Re1$, $Ar2 \rightarrow Re2$, $Ar2 \rightarrow Re3$
- She selects the name „Parts-Architectures-Requirements“ for the baseline
- The new baseline is stored under the chosen name in the EDDM central link repository
- As an optional step the new baseline can be stored in any of the data containers Requirements, Architectures, Parts (please note that the existing element links stay in the respective EDDM servers)

Enterprise data dependency management (EDDM)

A comprehensive use case – Remarks on Step 10: EDDM baseline

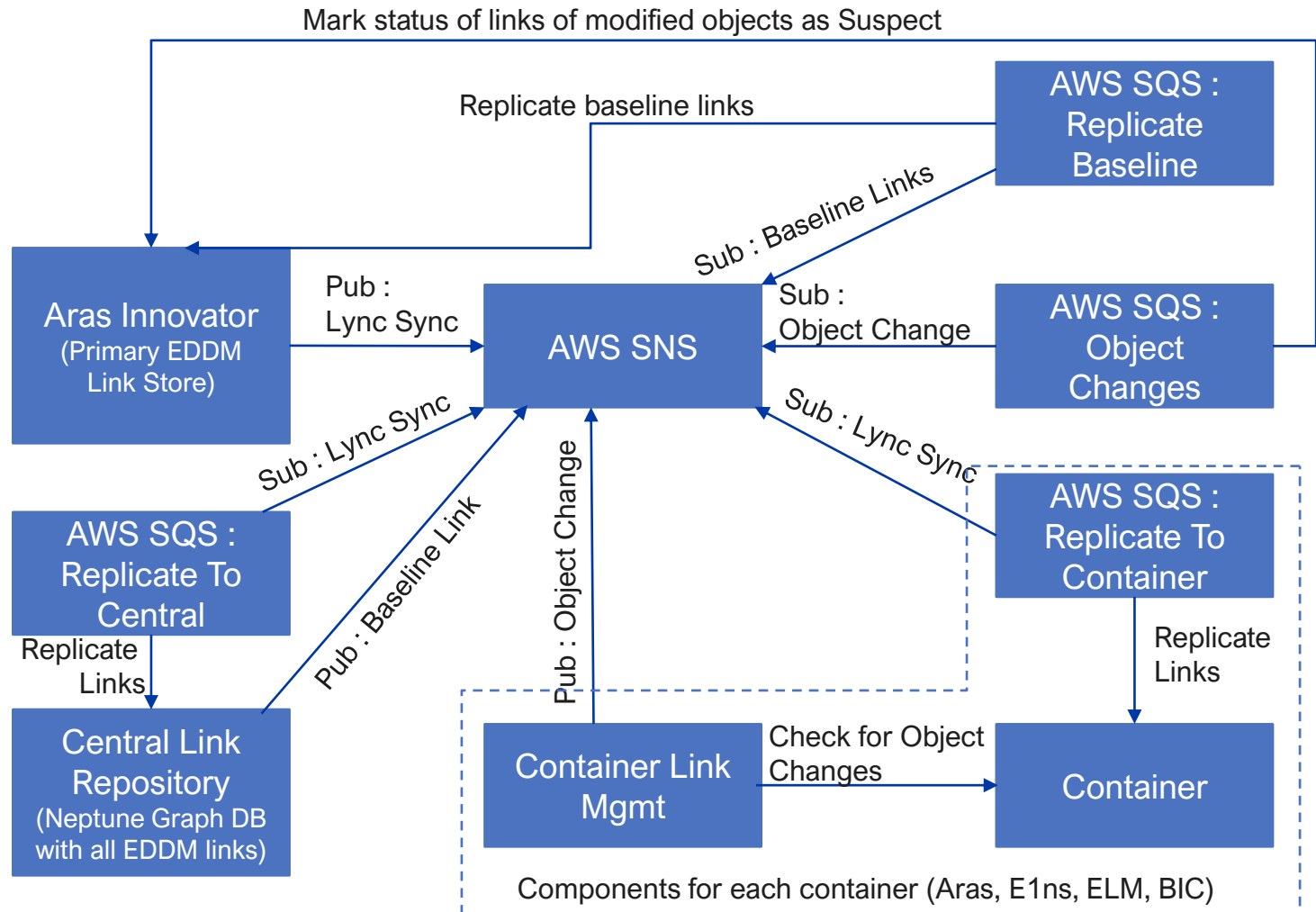


- The crucial step is the selection of all links to be included in the new EDDM baseline. That must be mostly automated by the EDDM platform
- One example for automation is the utilization of the data dependency graph (DDG):
 - The user selects the head of the parts structure (Pa1) which should be used for automated baseline link selection
 - The system traverses the parts structure to find the set of all parts with outbound EDDM element links {Pa1, Pa2}
 - The system calculates the DDG of this data set as follows
 - The system adds the links Pa1→Ar1, Pa2→Ar2, Pa2→Ar3 to the DDG
 - The system determines the targets of all links from the previous step (data elements Ar1, Ar2, Ar3)
 - The system determines all Architectures outbound links that have one of the data elements from the previous step as link source
 - The system adds the links Ar1→Re1, Ar2→Re2, Ar2→Re3 to the DDG
 - Here the algorithm terminates since the Requirements data container has no outbound links with Re1, Re2, Re3 as link source

Enterprise data dependency management (EDDM)

Asynchronous Link Replication and Link Status Maintenance

- Aras Innovator is the primary link store. Whenever an EDDM link is created or modified, publishes a message to AWS SNS
- Queue Processors for Central Link Repository and relevant containers subscribe to the messages and replicate the link changes
- EDDM baseline links are initially created in Central Link Repository and then replicated to Aras



Enterprise data dependency management

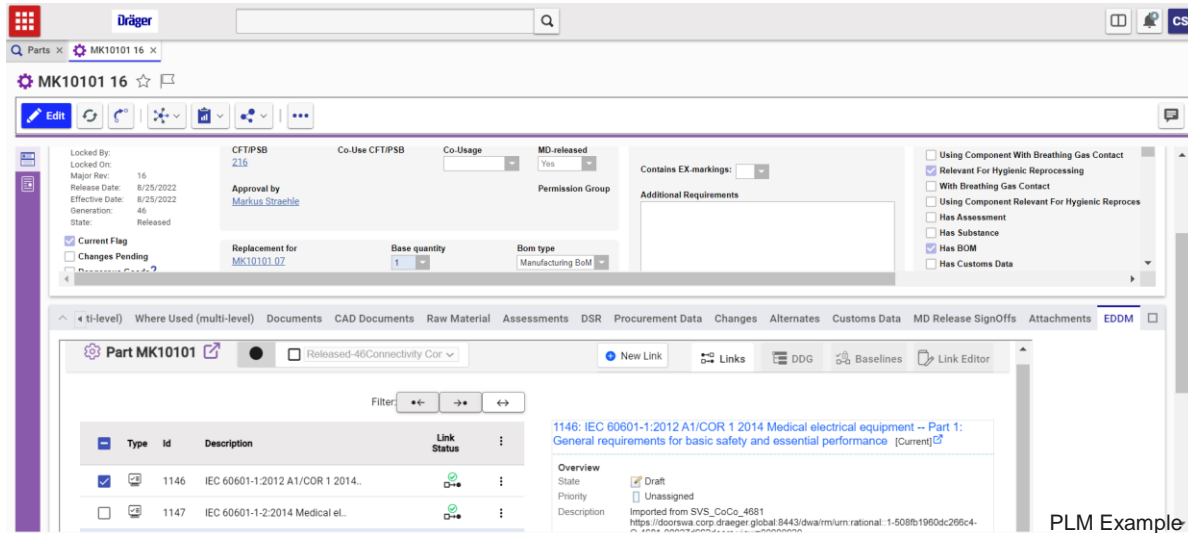
Glossary

Term	Description
Data container	An IT system for the creation/update/deletion and persistence of EDDM artifacts (data elements, data items, data links)
Data element	A self-contained indivisible and immutable unit of information
Data item	A set of data elements and data items (it cannot contain itself)
Data link	A directed relationship between two data elements or data items
DDG	Data dependency graph
EDDM Baseline	A named immutable set of EDDM element links
EDDM Central Link Repository	A EDDM server (it could be included in the EDDM central server) that holds (possibly redundantly) all links and link baselines from all EDDM servers in the EDDM landscape
EDDM Central Server	Enables EDDM services discovery and the translation from/to symbolic EDDM data links into resolvable URL's
EDDM Client	Enables data link create/update/delete
EDDM Server	Manages all EDDM queries for a data container, all data operations on data links for the data container and provides delegated UI's for data elements, items and data links
Element link	A data link where both link source and link target are data elements
Item link	A data link where either link source or link target or both are data items
Resolvable URL	Used to navigate to EDDM link endpoints. Might contain references to changeable IT infrastructure systems
Symbolic EDDM link	Describes an EDDM link without references to changeable IT infrastructure systems

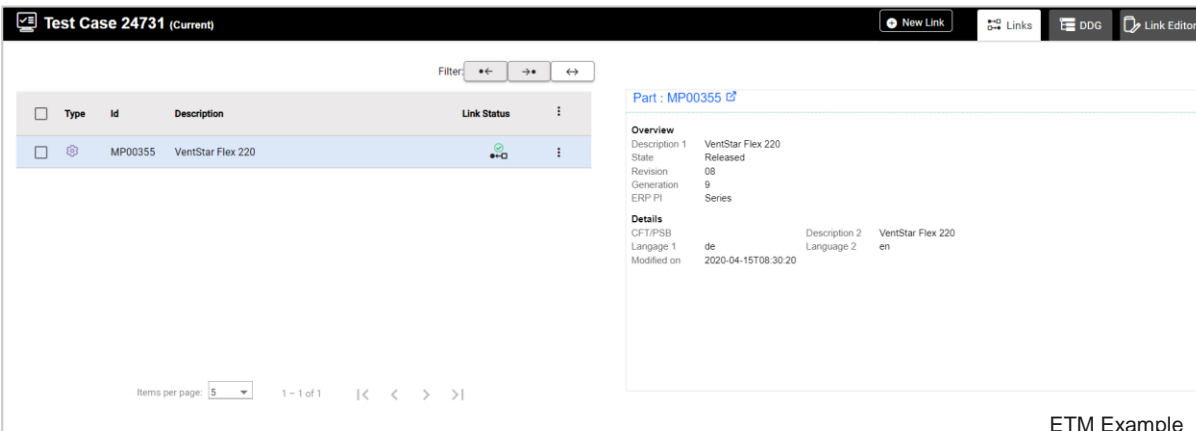
Appendix – EDDM Implementation Screenshots

Enterprise data dependency management (EDDM)

Seamless integration in client UI



PLM Example





ETM Example

- EDDM is integrated in the design of the data container
- + Makes user interface consistent
- + Places users in control of the interface
- + Eases user interaction due to familiar design
- + Enhances user experience due to reduction of cognitive load


Enterprise data dependency management (EDDM)

Bulk operations on link sets


 **Test Case 24731** (Current)




New Link



Links



DDG



Link Editor

Node

Type Part

Id

☐ Item ☐ Element ☒ Both

↔

→•

•←


















Node

Type

Id

☐ Item ☐ Element ☒ Both

Show links with status ☒ Valid ☒ Suspect ☒ Invalid ☒ Dangling Reset Search

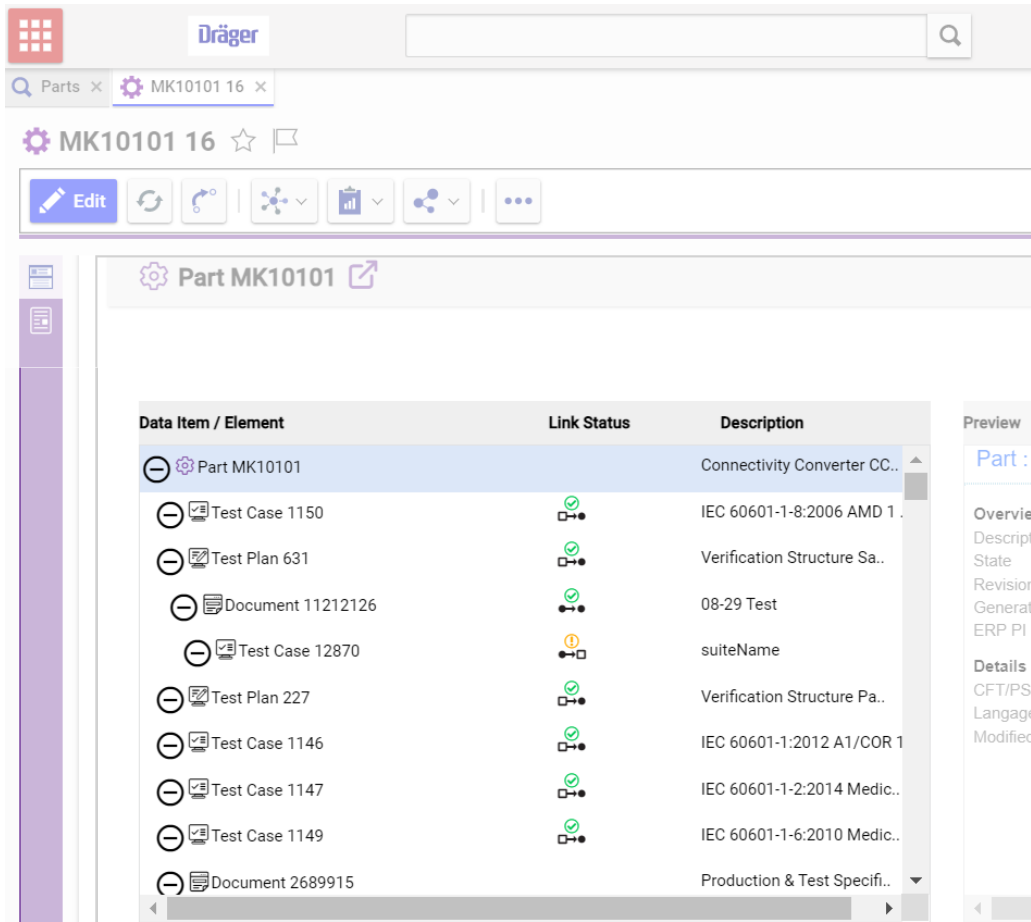
	Type	Id ↑	Description	Link Status	Type	Id	Description	
<input checked="" type="checkbox"/>		MK10101	Connectivity Converter CC300			1147	IEC 60601-1-2:2014 Med electrical e..	<div>Mark Selected As Valid</div>
<input type="checkbox"/>		MK10101	Connectivity Converter CC300			1149	IEC 60601-1-6:2010 Med electrical e..	Mark Selected As InValid
<input checked="" type="checkbox"/>		MK10101	Connectivity Converter CC300			631	Verification Structure Saf List ..	Delete Selected Link
<input checked="" type="checkbox"/>		MK10101	Connectivity Converter CC300			1150	IEC 60601-1-8:2006 AMD (CON ED) 2..	
<input type="checkbox"/>		MK10101	Connectivity Converter CC300			1146	IEC 60601-1:2012 A1/COR 1 2014 Medical ..	

Items per page: 5 1 – 5 of 6 |< < > >|

- EDDM allows performing changes on several links at once, such as deletion and link status change
- + Enables faster changes to link sets
- + Ensures same change to link sets at once

Enterprise data dependency management (EDDM)

Data dependency management



The screenshot displays the Dräger EDDM web application interface. At the top, there is a search bar and a navigation menu with tabs for 'Parts' and 'MK10101 16'. Below the navigation bar, there is a toolbar with icons for 'Edit', 'Refresh', 'Link', 'Add', 'Remove', and 'More'. The main content area shows a list of data items for 'Part MK10101'. The list has three columns: 'Data Item / Element', 'Link Status', and 'Description'. The items are as follows:



Data Item / Element	Link Status	Description
Part MK10101		Connectivity Converter CC..
Test Case 1150	✓	IEC 60601-1-8:2006 AMD 1..
Test Plan 631	✓	Verification Structure Sa..
Document 11212126	✓	08-29 Test
Test Case 12870	⚠	suiteName
Test Plan 227	✓	Verification Structure Pa..
Test Case 1146	✓	IEC 60601-1:2012 A1/COR 1
Test Case 1147	✓	IEC 60601-1-2:2014 Medic..
Test Case 1149	✓	IEC 60601-1-6:2010 Medic..
Document 2689915		Production & Test Specifi..


On the right side of the interface, there is a 'Preview' panel with a sidebar menu containing 'Overview', 'Description', 'State', 'Revision', 'Generatio', and 'ERP PI'. The 'Details' section shows 'CFT/PSB', 'Langage', and 'Modified c'.




- EDDM's data dependency graph (DDG) shows the hierarchy of the links
- + Eases understanding of complex hierarchies
- + Allows efficient analysis of dependencies and impact of changes
















Enterprise data dependency management (EDDM)

Change detection capabilities

 **Part MP00331** 

☐ Released-12VentStar Anaesth 

Filter:   

<input type="checkbox"/>	Type	Id	Description	Link Status	
<input type="checkbox"/>		345	Verification Structure		
<input type="checkbox"/>		386	Software Test		
<input type="checkbox"/>		822	PCS 1.1.2 Verification Structu..		
<input type="checkbox"/>		823	SW Plan v 1.1.2		
<input type="checkbox"/>		824	SW Verification Structure 1.1...		

- EDDM detects automatically different type of changes on the linked items (link status)

EDDM link status:

- ✓ Valid: link is valid and link endpoints have not changed
 - ? Suspect: at least one link endpoint has changed
 - X Invalid: link is invalid (user defined)
 - ∅ Dangling: at least one link endpoint is lost
- + Ensures up-to-date link status
 - + Informs user about changes to any side of the link
 - + Eases recognition of type of change
 - + Allows efficient response to changes