Ref. Ares(2022)2370439 - 30/03/2022

AI-augmented automation supporting modelling, coding, testing, monitoring and continuous development in Cyber-Physical Systems

D5.1 AIDOaRt Integration Approach

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007350. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Sweden, Austria, Czech Republic, Finland, France, Italy, Spain. This document reflects only the author's view and that the Commission is not responsible for any use that may be made of the information it contains.

Contract number:	101007350
Project acronym:	AIDOaRt
Project title:	AI-augmented automation supporting modelling, coding, testing, monitoring and continuous development in Cyber-Physical Systems
DELIVERY DATE	March 31, 2022
Author(s):	Gerald Stieglbauer
Partners contributed:	Per Strandberg (WESTMO), Raul Santos de la Cámara (HIB), Marc Oriol (UOC), Hugo Bruneliere (IMTA), Luca Berardinelli (JKU), Romina Eramo (UNIVAQ), Bilal Said (SOFT), Ganesh Chandramouli (BT), Claudia Keinrath (AVL)
Date:	March 30, 2022
Version	V1
Revision:	09
Abstract:	This report describes the overall integration approach taking into account various integration aspects such as Use Case Providers and Solution Providers specific point of view as well as integration aspects related to commonalities and synergies, state-of-the-art and relevant literature, relation to AIDOaRt architecture, enhancing existing Use Case environments, relation to involved data sets and interfaces between tools, model exchange formats and deployment peculiarities in order to provide end-users with an integrated experience. The core of the report is a so-called AIDOaRt integration mediator pattern alongside a dedicated process, which supports the definition of vertical and horizontal integration relations between use case and solution providers.
Status:	Type: R, Dissemination Level: PU

DOCUMENT REVISION LOG

VERSION	REVISION	DATE	DESCRIPTION	AUTHOR
V1	01	28. Jan	Initial Draft, ToC and expected content description	Gerald Stieglbauer
V1	02	13. Feb	Initial texts in most of the sections to be discussed with AIDOaRt partners	Gerald Stieglbauer, reviewed by WP leaders
V1	03	20. Feb	Add Section about Integration on Data Level	Gerald Stieglbauer, reviewed by Marc Oriol
V1	04	22. Feb	Add Section about Integration on Use Case Environment and AIDOaRt Architecture Integration Level	Gerald Stieglbauer
V1	05	25. Feb	Add Section about Integration Roadmap and Conclusion	Gerald Stieglbauer
V1	06	5. Mar	Add Section 2.2, finish version for the first iteration of AIDOaRt-internal review	Gerald Stieglbauer
V1	07	19 Mar	Incorporate feedback of first AIDOaRt-internal review	Gerald Stieglbauer, Per Strandberg (main reviewer), Marc Oriol, Hugo Bruneliere, Luca Berardinelli, Romina Eramo, Bilal Said
V1	08	20 Mar	Add some citations, finish version for second iteration of AIDOaRt-internal review	Gerald Stieglbauer
V1	09	30 Mar	Incorporate feedback of second AIDOaRt-internal review	Gerald Stieglbauer, Raul Santos de la Cámara

TABLE OF CONTENTS

DOCUMENT REVISION LOG 3

TABLE OF CONTENTS 4

1. Introduction 6
1.1. Scope, Motivation 6
1.2. Document Structure 7
1.3. Relations to Other Deliverables 9
1.4. Terms Used in the Document 10

2. Integration Goals and Approaches 12
2.1. Integration Goals 12
2.2. Use Case Provider's vs. Solution Provider's Perspective on Integration Requirements 14
2.2.1. Vertical vs. horizontal integration aspects in AIDOaRt 14
2.2.2. Potential integration constraints related to use case and solution providers 15
2.3. Integration Approach: The AIDOaRt Integration Mediator Pattern 20
2.3.1. Definition of the AIDOaRt Integration Mediator Pattern 20
2.3.2. Definition of Relation and Integration Status 23
2.3.3. Definition of the AIDOaRt Mediator Integration Pattern Process 25
2.3.4. Additional Conclusions on Gap Analysis and Integration Status 26
2.4. Summary 27

3. Integration Aspects 29
3.1. Integration Aspects on Generic Requirements Level 29
3.2. Integration Aspects on State-of-the-Art Level 33
3.3. Integration Aspects on AIDOaRt Architecture Component Level 38
3.4. Integration Aspects on Data Engineering Level 39
3.5. Integration Aspects on Use Case Environment Extensions 42
3.6. Integration Aspects on AIDOaRt Architecture Component Interface Level 45
3.7. Potential further Integration Aspects on Core Tool Set and AI-Augmented Tool Set 48
3.8. Summary 48

4. Integration Roadmaps 49

5. Conclusions 54

Bibliography 55

1. Introduction

This report describes the general integration approach of AIDOaRt. AIDOaRt focuses on AI-enabled methods, tools, and use cases for different DevOps stages. For this reason, integration needs to be discussed and evaluated from more than one perspective. Accordingly, the term integration is interpreted in the context of AIDOaRt in a broader sense than classical definitions that focus, for example, on tool coupling or system integration[1]. AIDOaRt is organized by two groups of partners, the use case providers and the solution providers. Sometimes competing perspectives are brought in by specific requirements from use case providers on the one hand and solution providers on the other hand. For example, while solution providers want to maximize their impact and the universal applicability of their tools and methods, use case providers are interested in ensuring that the method and tools can be easily integrated into their existing environment and therefore want to minimize the required customizations. In certain cases, integration also involves agreement on the methods used, manifested in specific algorithms or frameworks, which can lead to large differences in the required integration efforts. As AIDOaRt is focusing on AI-augmented methods, data aspects play an important role for integration as well, as it has to be consumed and interpreted by various components of an AI-augmented DevOps approach.

1.1. Scope, Motivation

The scope and motivation of this report are to systematically collect and describe these various integration aspects in a unified project-wide form. To some extent, this will also summarize some integration-related activities happening in various other work packages in a single place. These aspects include commonalities and synergies, state-of-the-art and relevant literature, relations to a common AIDOaRt architecture, demands on enhancing existing use case environments, relation to involved data sets and interfaces between tools, model exchange formats, and deployment peculiarities to provide end-users an integrated experience

The document also describes how these different aspects of integration will be captured, monitored and evaluated during the remainder of the AIDOaRt project and how this will strengthen the collaboration during the project and the joint project outcomes at the end of the project.

The document as such will not yet provide results regarding the captured, monitored and evaluated integration aspects of the various Use Cases and Solutions. However, the unified integration approach presented in this document will be applied during Task T5.1 and will lead to the deliverables D5.2, D5.3, D5.4, which will contain the proposed integration approaches and evaluation of the use case and solution related components. If required, these follow-up deliverables will present an updated version of the integration approach described in this document, as the practical application of the approach is intended to gain more insights about the various integration aspects and beyond.

[1] See for instance: https://en.wikipedia.org/wiki/System_integration

1.2. Document Structure

This report is organized as follows: After a brief introduction by the current chapter (including relationships to other deliverables and terms used in the document), an overview of the various integration aspects is provided in Chapter 2, setting them in the context of the particular use case and the solution provider's perspective. The core of the chapter is Section 2.3, which proposes a generic pattern to support various integration activities within the AIDOaRt project. This generic pattern is referred to as the AIDOaRt integration mediator pattern and is accompanied by a specific AIDOaRt integration mediator process. Both the pattern and the process are then applied to six different integration aspects, which are described in detail in Chapter 3. Each description of an integration aspect is supported by (generic) examples to outline its intention. Chapter 4 then derives a concrete integration roadmap from the approaches described in Chapters 2 and 3. Finally, the document closes with some conclusions in Chapter 5.

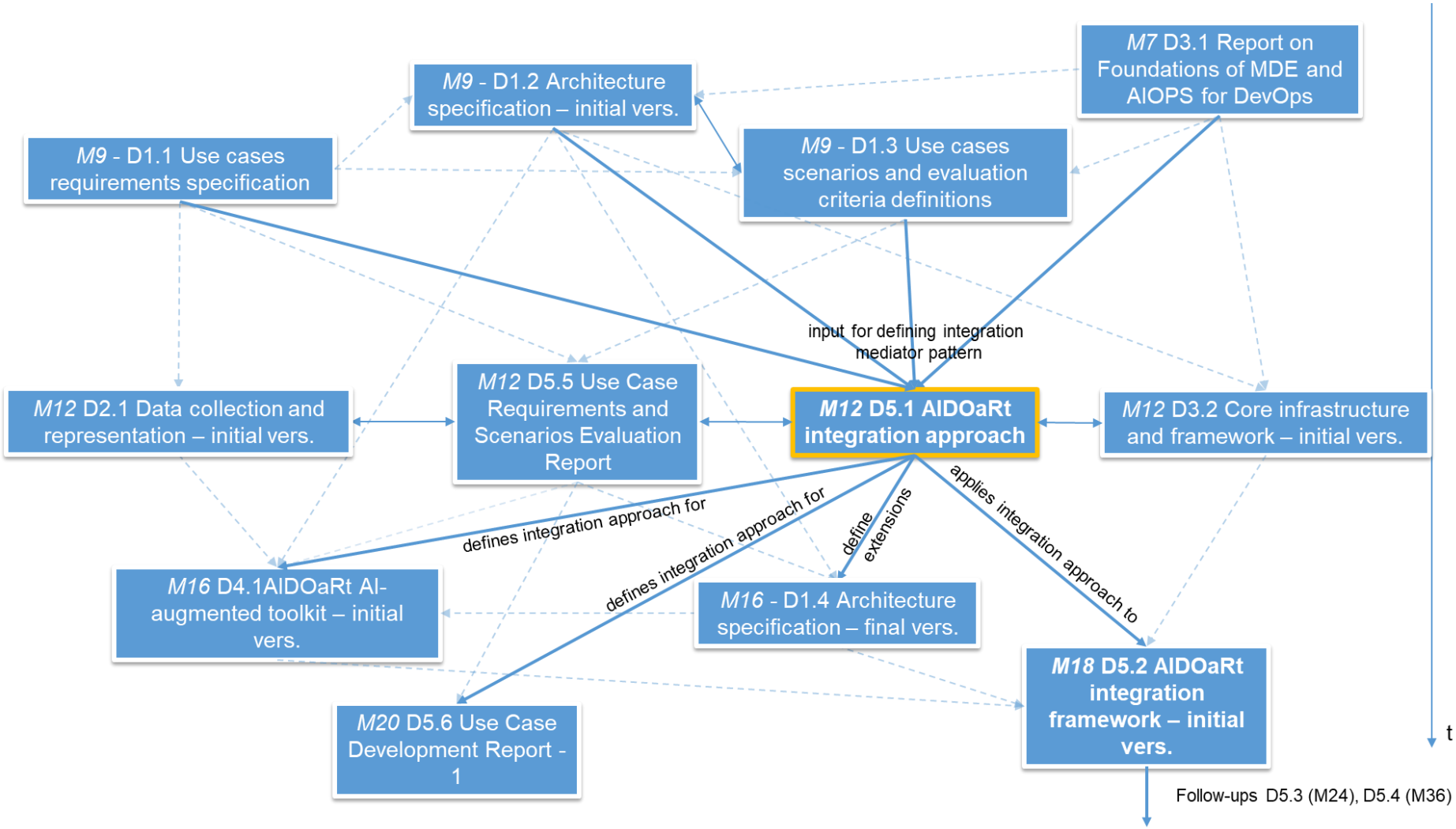

Figure 1 Related Deliverables

1.3. Relations to Other Deliverables

Figure 1 gives an overview of the related deliverables and how this deliverable D5.1 is embedded, taking into account the planned schedule. D5.2 is marked as bold as well since it will be the direct follow-up deliverable of D5.1. Relations between other deliverables than D5.1 are indicated by transparent lines, as these relations may be interesting in general but are not further discussed in this report. The notation Mxx denotes the deliverable submission date (e.g., M12 denotes month 12 of the AIDOaRt project, i.e., March 31, 2022)

Firstly, contribution D5.1 is based on some activities described in previous contributions produced by WP1 and WP3 work packages in the first phase of AIDOaRt. These activities led to the idea of generalizing some of the activities there in the form of a generic AIDOaRt integration mediator pattern, which is outlined in this deliverable in the following chapters. For example, in D1.1 (Use Case Requirements Specification), generic requirements were designed to generalize specific integration aspects between use cases and solution components. Similarly, in D3.1 (Report on Foundation of MDE and AIOPS for DevOps), a systematic mapping study (SMS) was performed, and a list of relevant SMS papers was created that could play a similar mediating role as the generic requirements.

Secondly, D5.1 will be produced in close collaboration with the parallel work planned in M12. D2.1 (Data collection and representation - initial vers.) and D3.2 (Core infrastructure and framework) provide further input that was similarly considered as generic requirements and SMS papers to define further integration aspects. These are mostly related to the corresponding AIDOaRt architecture component and its extensions. In this context, the results from the previous deliverable D1.2 (Architecture specification - initial vers.) and discussions around the upcoming deliverable D1.4 (Architecture specification - final vers.) were taken into account.

Thirdly, D5.1 is an input source for a number of other upcoming deliverables. Most important, of course, is deliverable D5.2 (AIDOaRt integration framework – initial vers.), which directly follows D5.1. D5.2 applies the integration approach presented in this report to the AIDOaRt integration framework. Furthermore, the AIDOaRt integration mediation pattern outlined in this document implies several extensions to the AIDOaRt architecture metamodel, which are presented in D1.4. In addition, another WP5 deliverable, namely D5.6 (Use Case Development Report - 1), depends on D5.1, as integration approaches should strongly influence the development and implementation of use cases. In addition, D4.1 (AIDOaRt AI-augmented toolkit - initial vers.) represents a set of scientific deliverables that should provide appropriate solution components for the use cases and are therefore also relevant for several integration aspects.

Finally, the AIDOaRt integration approach outlined in this document will be continuously developed through its immediate follow-up contributions. This is, as already mentioned, contribution D5.2, which will be refined and updated by its follow-up contributions, namely contributions D5.3 (M24) and D5.4 (M36).

1.4. Terms Used in the Document

In the following, the most important terms used in this document are listed and shortly defined:

- **Use Case Provider** – Partner, which provide a *Use Case Scenario* related to a dedicated end product or service in form of a *Case Study*
- **Case Study** - is about the intended end product in the form of a demonstrator and/or prototype of a system
- **Case Story** - describes what the product should offer the end user as concrete product features
- **Use Case Scenario** - describes *which* requested/required AI-augmented technical solution/approach is intended to be applied within a concrete Use Case to realize the related Case Story
- **Use Case Requirements** – describes *what must be fulfilled* to enable the Use Case Scenario in the context of the intended AI-augmented technical solutions/approaches manifested by Component Capabilities (see below)
- **Use Case Data Requirements** - are derived requirements for each Use Case Scenario but is not focussing on the AI-augmented technical solutions/approaches as such, but on the involved data sets with which these solution/approaches must deal with in the context of AI-augmented topics. In this sense, Use Case Data Requirements are driven not only by the Use Case Providers, but also by the Solution Providers, since their tools have to deal with the data related to the Use Cases.
- **Solution Provider** – Partner, which provides a corresponding solution for one or more *Use Case Providers* by a dedicated method, tool or algorithm manifested in concrete *Component Capabilities*.
- **Solution Component (Capabilities)** - are concrete tool features, methods or other solutions provided/developed by the Solution Providers. They should fulfil Use Case (Data) Requirements.
- **Generic Requirements** – based on the set of Use Case Requirements, Use Case-independent Generic Requirements have been derived and potentially assigned to dedicated Solutions and Component Capabilities
- **AIDOaRt Integration Approach** – The overall integration approach presented by this document, which is continuously evolving during T5.1 activities and is reported by a series of deliverables (D5.1-4).
- **AIDOaRt Integration Mediator Pattern** – A generic pattern-based approach, which is intended to detect potential and actual integration relationships between use case and solution providers
- **AIDOaRt Integration Mediator Instance** – A concrete instance of the AIDOaRt Integration Mediator Pattern usually refers to a dedicated integration aspect and can be referred by use case and solution providers to derive potential indirect relationships.
- **Integration Aspect** – A concrete topic for an AIDOaRt Integration Mediator Instance, which associates a (potential) relationship between use case and solution providers (or use case

and solution aspect) with a specific content (e.g., generic requirement, applied state-of-the-art, involved AIDOaRt architecture component, etc.).

- **AIDOaRt Integration Mediator Process** – A dedicated process associated with the AIDOaRt Integration Mediator Pattern, which evaluates potential horizontal and vertical relationships between use case and solution providers to be confirmed or rejected. It comes along with a gap analysis for the rejected ones and an integration status for the accepted ones.
- **AIDOaRt Integration Status** – Refers to the confirmed (actual) relationships (horizontal and vertical) between use case and solution providers. It further refers to concrete involved components related to the corresponding integration aspect (e.g., relevant papers, AIDOaRt architecture elements or toolsets, etc.)
- **Integration Strategy** – concrete report how specific components and elements referred by the AIDOaRt integration status will be integrated – this will be part of one of the upcoming deliverables D5.2-4, but is not part of this deliverable.
- **Vertical Integration Relations** - Integration relations relevant between use case and solution providers.
- **Horizontal Integration Relations** – Integration relations relevant either between one or more use case providers or between one or more solution providers.
- **AIDOaRt Architecture** – Overall architecture of the AIDOaRt project as elaborated by task T1.3 and reported by the deliverables D1.2 and D1.4
- **AIDOaRt Architecture Component** – Concrete component defined by the AIDOaRt architecture reflecting a certain aspect of the AIDOaRt project
- **AIDOaRt Architecture Component Interface** – Generic interface description of a concrete AIDOaRt architecture component.
- **AIDOaRt Modelio model-based approach** – Model-based approach to define a project-wide single source of truth of the AIDOaRt architecture, use cases, solutions and the relationships among them.
- **Systematic Mapping Study (SMS)** – Extensive literature study performed by T3.1 to find AIDOaRt relevant literature as basis for the aimed AIDOaRt goals and research activities

2. Integration Goals and Approaches

This chapter presents different integration objectives and approaches pursued by the activities of Task T5.1 in particular and by the AIDOaRt project in general. For this reason, **Section 2.1** first looks at the general integration goals and explains what is meant by the term integration in the context of the AIDOaRt project. **Section 2.2** then looks at different aspects of integration from the perspective of a use case and a solution provider. This is necessary because the AIDOaRt project is structured by corresponding partner groups (in some cases a use case provider is also a solution provider). Under this aspect, vertical (integration aspects between use case providers and solution providers) and horizontal integration aspects (integration aspects between multiple solution providers or multiple use case providers) are defined and described in Section 2.2.1. Section 2.2.2 lists potential conflicts of interest between use case providers and solution providers that affect different integration aspects in the form of constraints. Subsequently, a generic integration approach is defined in **Section 2.3**.

It consists of a so-called AIDOaRt integration mediation pattern (Section 2.3.1), which is intended to reveal and capture vertical and horizontal integration relationships between use case and solution providers. The pattern is manifested by concrete instances that focus on a specific integration aspect, which is initially also presented in this section. The integration aspects are then the basis for potential integration relationships between use case providers and solution providers and for monitoring their integration status (Section 2.3.2). This monitoring requires a corresponding AIDOaRt integration mediator process, which is outlined by Section 2.3.3. This process proposes a series of microtasks, which includes besides monitoring the integration status as well a gap analysis in case of orphan use cases or solution components. In this sense, Section 2.3 forms the basis for a detailed description of the individual integration aspects in Chapter 3 on the one hand and for an integration roadmap in Chapter 4 on the other. Specifically, the integration roadmap applies the integration aspects from Chapter 3 to the integration process defined in Section 2.3 in order to derive concrete microtasks and align them with the planned schedule of deliverables for task T5.1 (represented by deliverables D5.1-4). Finally, it closes with a short summary and conclusion (**Section 2.4**)

2.1. Integration Goals

The predecessor project MegaM@Rt2[2] followed an integration approach based on classical software system integration, tool coupling and tool interoperability [1]. There, an overview of various integration principles can be found, which have been relevant for this project, including technology, data format, location and temporal dependencies. Dependencies as such can be classified from *tightly-coupled* to a *loosely-coupled* systems. Relevant parameters for classifying coupling strength

[2] MegaM@Rt[2] project consisted of six national clusters and 27 partners with nine industrial case studies and was funded in total with 16.5M€. It was about a scalable model-based framework, for continuous development and runtime validation of complex systems (DevOps). Many partners of this project are also partners of AIDOaRt (e.g., both projects are led by Märlardalen University). AIDOaRt can thus be seen as a follow-up project and is extending the MegaM@Rt[2] project goals by AI-augmented approaches for model-based engineering and DevOps, or AIOps for short.

can be found in [2]. Technically, MegaM@Rt2 focus was on the Eclipse modelling framework[3] environment and the integration approaches included tool packaging and delivery methods and an interoperable Eclipse interface.

If methods of integrations are considered, vertical and horizontal integrations are considered. While vertical integrations are rather interpreted as silos with limited re-use potential, horizontal integrations are considered with higher re-use and sharing of (core) functionality for different applications. Classical concepts can be found for instances in [3].

Although these classic SW engineering aspects of integration are not left out, the term integration is interpreted in a broader sense in this report. The overall goal of AIDOaRt is to provide a harmonized platform consisting of solution components that successfully implement multiple use cases in the context of case studies. Since AIDOaRt focuses on AI-augmented DevOps Engineering, or AIOps for short, all these case studies are related to this project scope (including AI, MDE and/or DevOps), and are intended to extend the current state of the art or state of practice accordingly. The goal of a collaborative project like AIDOaRt is that a solution component is universally applicable to different use cases, while a dedicated use case integrates different solution components into the targeted AIOps-specific extension defined by the associated case study.

To be successful, integration must not occur only at the interface level of the solution components but also at the level of the methodology offered by the different solution components. For example, if a use case specifies a particular goal, namely solving a particular problem using AI, integrating any learning method or algorithm at the software integration level will not necessarily lead to success. Consequently, the integration also needs to be analyzed at a higher level, whether the AI method is able to implement the requirements of the use case or how it needs to be adapted to be successfully integrated. A similar discussion needs to be held about the role of data. On the one hand, data interfaces, data formats and standards are important and necessary for successful integration, but on the other hand, there is no guarantee that integration will succeed just by considering this aspect. Successful integration can only be guaranteed if the data as such fulfil certain requirements, which are also determined by the applied learning method of the solution component involved and not only by the case study as such. This also brings into play different perspectives and interests of use case providers and solution providers. In the given example, the use case provider might tend to provide the solution provider with a minimum of required data (as data is costly) and expect the solution provider to achieve maximum output. The opposite expectation might be the case if the solution provider expects a perfect or overcomplete data set from the use case provider that can be used to incrementally improve the learning of the applied solution or method. The next section therefore introduces the idea of an explicit negotiation phase between the parties involved, supported by a generic process described in Section 2.3, in which such discrepancies between expectations should be made explicit at an early stage of the project, followed by appropriate mitigation actions. It also introduces another view on vertical and horizontal integration.

[3] https://www.eclipse.org/modeling/emf/

Beyond classical approaches such as mentioned above, vertical integration aspects in this report includes integration relations between use case and solution providers, while horizontal integration means either integration relations between various solution providers or integration relations between various use case providers.

2.2. Use Case Provider's vs. Solution Provider's Perspective on Integration Requirements

As outlined in the previous section, the perspectives of a use case provider and a solution provider may differ on different integration aspects. This section attempts to compare the two different perspectives and the possible impact on integration aspects. Where applicable, the impact on so-called vertical and horizontal integration aspects will be discussed.

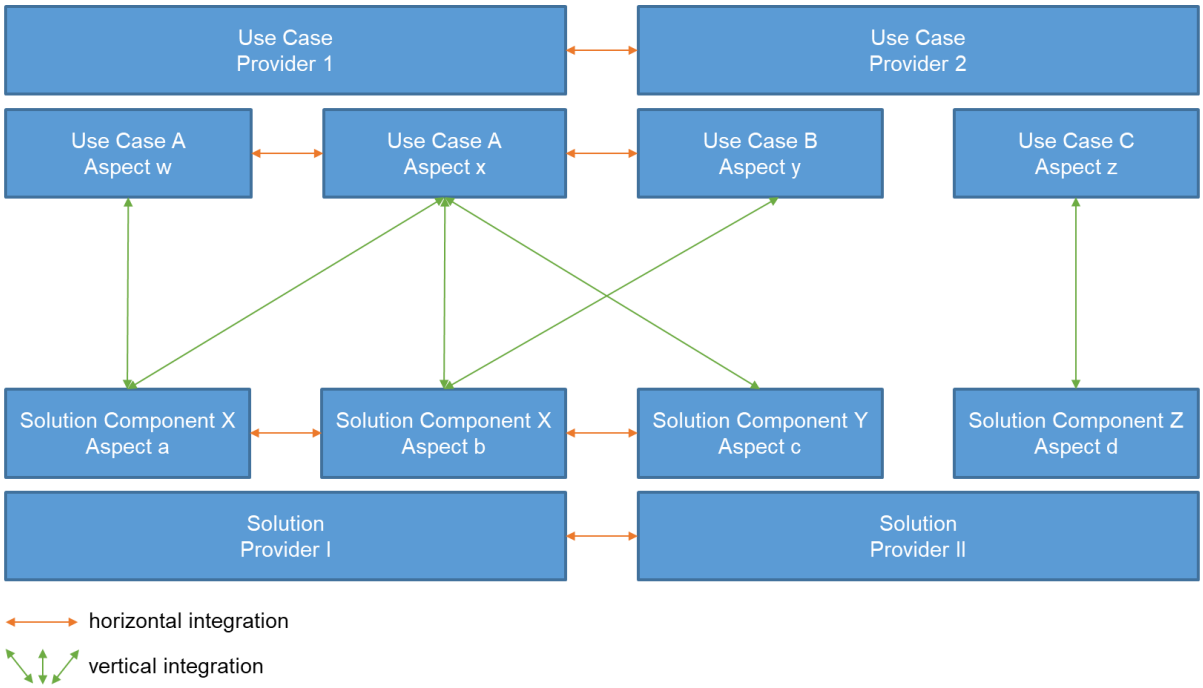

Figure 2 Vertical vs. horizontal integration aspects in AIDOaRt

2.2.1. Vertical vs. horizontal integration aspects in AIDOaRt

Figure 2 outlines a plausible interpretation of *vertical* and *horizontal integration* between use case and solution providers. In the context of AIDOaRt, integration aspects between use case providers and solution providers are referred to as *vertical integration*. Vertical integrations are not limited to a single use case aspect and a single solution component aspect as outlined in this figure, but rather can represent a complex network of cross-relationships between different use cases from different use case providers and multiple solution components from more than one solution provider. How vertical integration is ultimately achieved depends on the sum of relations between use case and solution providers.

For example, if solution component X of solution provider I is to be integrated into use case A and use case B of use case provider 1 and use case provider 2 respectively, solution provider I has to consider the specific requirements of use case provider 1 and use case provider 2. To minimize the effort of integrating solution component X (aspect b) and two use cases, harmonized methods, interfaces or requirements for the solution component are proposed by solution provider I to use case providers 1 and 2. This potentially also leads to *horizontal integration* between the two Use Case Providers, e.g., a common interface layer adapted to the respective Use Cases. Similar horizontal integration within different solution components of the same or different solution providers may be required depending on their planned vertical integrations. For example, as solution component X (aspect b) and solution component Y (aspect c) are planned to be vertically integrated into use case A (aspect x), horizontal integration between solution components may be required to meet certain cross-solution component requirements of use case A.

2.2.2. Potential integration constraints related to use case and solution providers

The following constraints, as well as possible diverging perspectives of use case and solution providers on horizontal and vertical integration, potentially influence the chosen integration approach or aspect. In some cases, this can lead to a kind of conflict zone between the interests of use case and solution providers. In D1.1 (Use cases requirements specification), the need for some kind of negotiation phase between these two parties alongside different integration aspects was pointed out. While in D1.1 this negotiation phase was mainly related to data aspects, in this work it is also extended to other integration aspects and is one of the drivers for the idea to introduce an AIDOaRt integration mediation pattern that can be applied during this negotiation phase.

Use case environment constraints: Typically, most of the case studies in AIDOaRt are based on existing DevOps environments that are now to be extended, improved or further automated with various AIOps-relevant approaches. This means that the solutions usually cannot be implemented straightforwardly in a use-case-independent manner, but that corresponding integration solutions to existing environments must be offered. In other words, an existing environment usually cannot be changed on a large scale due to various dependencies and follow-up costs. Therefore, use case providers will typically seek to select and favor solution components that require minimal changes to their environment or even require solution providers to provide suitable adapters for integration. This tends to shift the effort of integration from the use case providers to the solution providers. The increased integration effort then reduces efforts for a use case independent implementation on the side of the solution providers, so that potential vertical integrations may have to be cancelled at the expense of individual integrations. A reduction in vertical integrations then also results in a reduction in horizontal integrations and thus a general reduction in project-wide collaborations. Monitoring this development and, if necessary, counteracting it at project level will be one purpose of the integration approach presented in this document.

Resource constraints: Both use case providers and solution providers agree on the common goal of working together to implement new and innovative solutions. However, from a use case provider's point of view, additional pressure arises from a competitive market situation, customer requirements or management policies. In practice, this pressure is amplified by limited resources and lack of time.

Accordingly, the criteria for selecting a particular solution component are not only driven by its degree of innovation but mainly as a combination of degree of innovation and resource constraints, often referred to by the slogan low-hanging-fruits first. This is of course related to the above-mentioned environment constraints in the sense that preference is given to solution components that require no or few environment extensions to be integrable. However, it also limits the willingness to integrate horizontally with other use case providers (e.g., the implementation of a common middleware) if the effort for this integration is estimated to be higher than the benefit of the integrated solution component. In particular, this may be contrary to the interest of the solution provider, where an individual one-to-one integration for each use case partner is beyond its capabilities due to resource constraints. Or in other words, solution providers want to maximize the applicability and thus the impact of their solution component by applying it to a range of use cases. Balancing this at the project level is another overarching goal of the approach presented in this document.

Contradictory requirements: Requirements can also be contradictory at both levels in terms of vertical and horizontal integration aspects. An example of a vertical requirements conflict has already been mentioned in the previous section at the level of data science, while here the requirements for data quality and size for AI-supported approaches are in conflict between use case and solution providers. Similarly, horizontal requirement conflicts can also influence the integration of solution components: Despite the goal of solution providers to maximize the impact of their solution components, conflicting requirements from use case providers may make it (technically) impossible to provide a universally applicable solution component. For example, performance requirements exclude certain algorithms from the solution or a certain underlying technology (e.g., Java vs. C++ environment) limits the possibility of integration. To avoid later inadequacies, these limitations need to be made visible at an early stage of negotiations between use case and solution provider.

Constraints due to standards: In certain cases, the introduction of standards supports issues related to various aspects of vertical and horizontal integration (e.g., interoperability standards such as OSLC or data standards such as ASAM ODS). This is especially the case when a standard is already established. However, the opposite can also be the case if the process of establishing a standard is rather in an early phase, as the establishment of a standard usually exceeds the lifetime of a research project like AIDOaRt (of course, AIDOaRt could provide valuable input for potential standards). However, there may also be restrictions if, for example, a use case provider has to comply with certain standards (e.g., for legal reasons). In this case, solution providers do not have much room for negotiation and have to consider the relevant standard for their integration approach. This can cause additional effort, especially in a multi-domain project like AIDOaRt, as the mandatory standards can differ from domain to domain. Consequently, it depends on a case-by-case basis whether standards promote vertical integration and facilitate horizontal integration or not.

Further aspects (especially related to data requirements): This list of possible constraints may evolve during the actual application of the integration approach presented by this report (e.g., use of open source and potential licensing issues). Especially concerning data requirements further constraints are expected. A detailed list of potential issues can be already found in Section 2.2.3 of the deliverable D1.1, which will be thus not repeated by this deliverable but further developed by

corresponding activities within WP2. Related data-related aspects especially related to AI-augmented approaches are limited trust, security issues or other human factor aspects that usually come along with such approaches.

The integration approach, presented in general in the next section and specifically in the next chapter, supports the assessment of potential and actual horizontal and vertical integration within the AIDOaRt project. On this basis, concrete integration approaches can be agreed with the use case and solution providers and constraints such as those mentioned above can be taken into account at an early stage of the project.

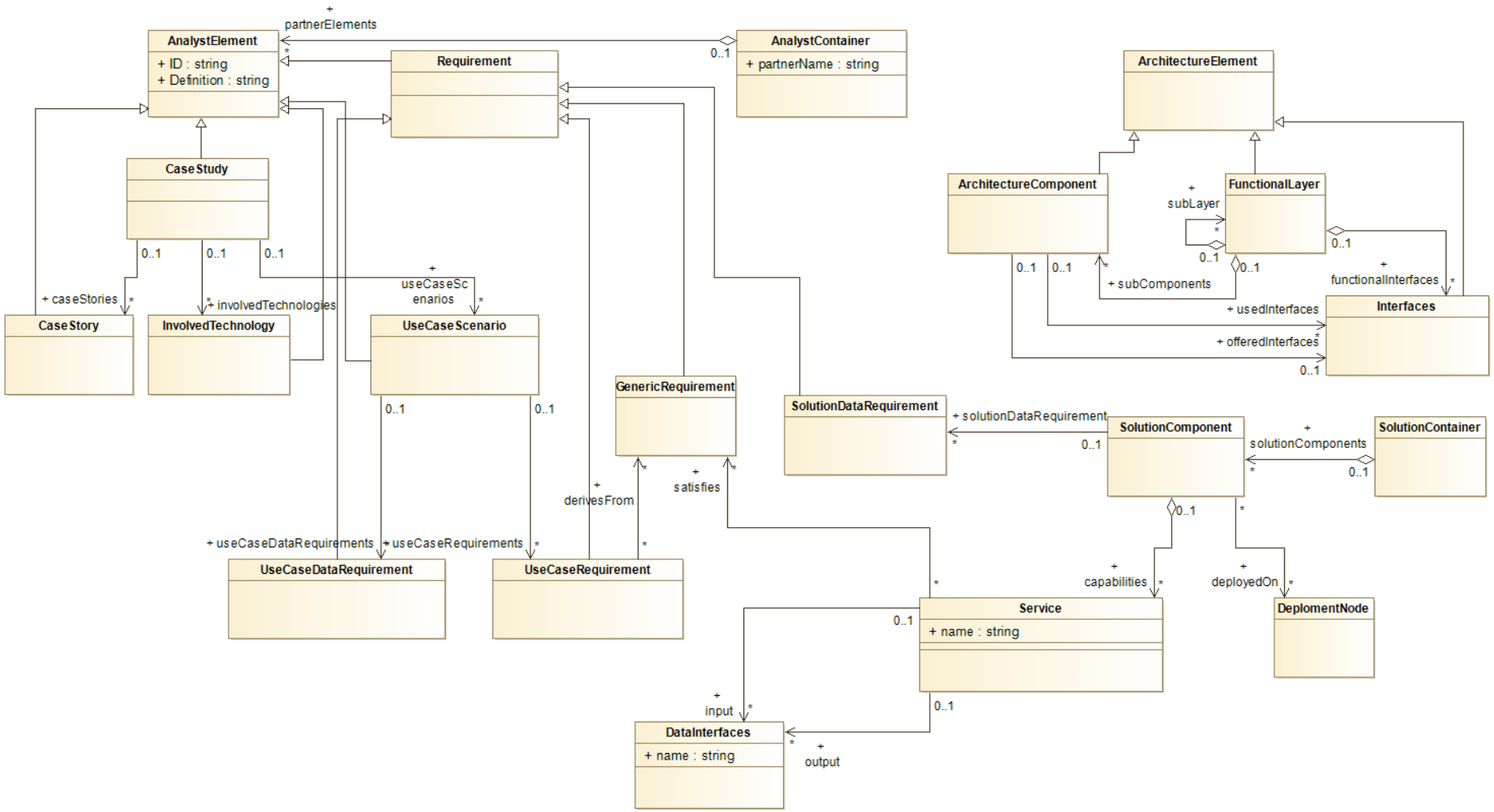

Figure 3 AIDOaRt Modelio metamodel (as of March 2022)

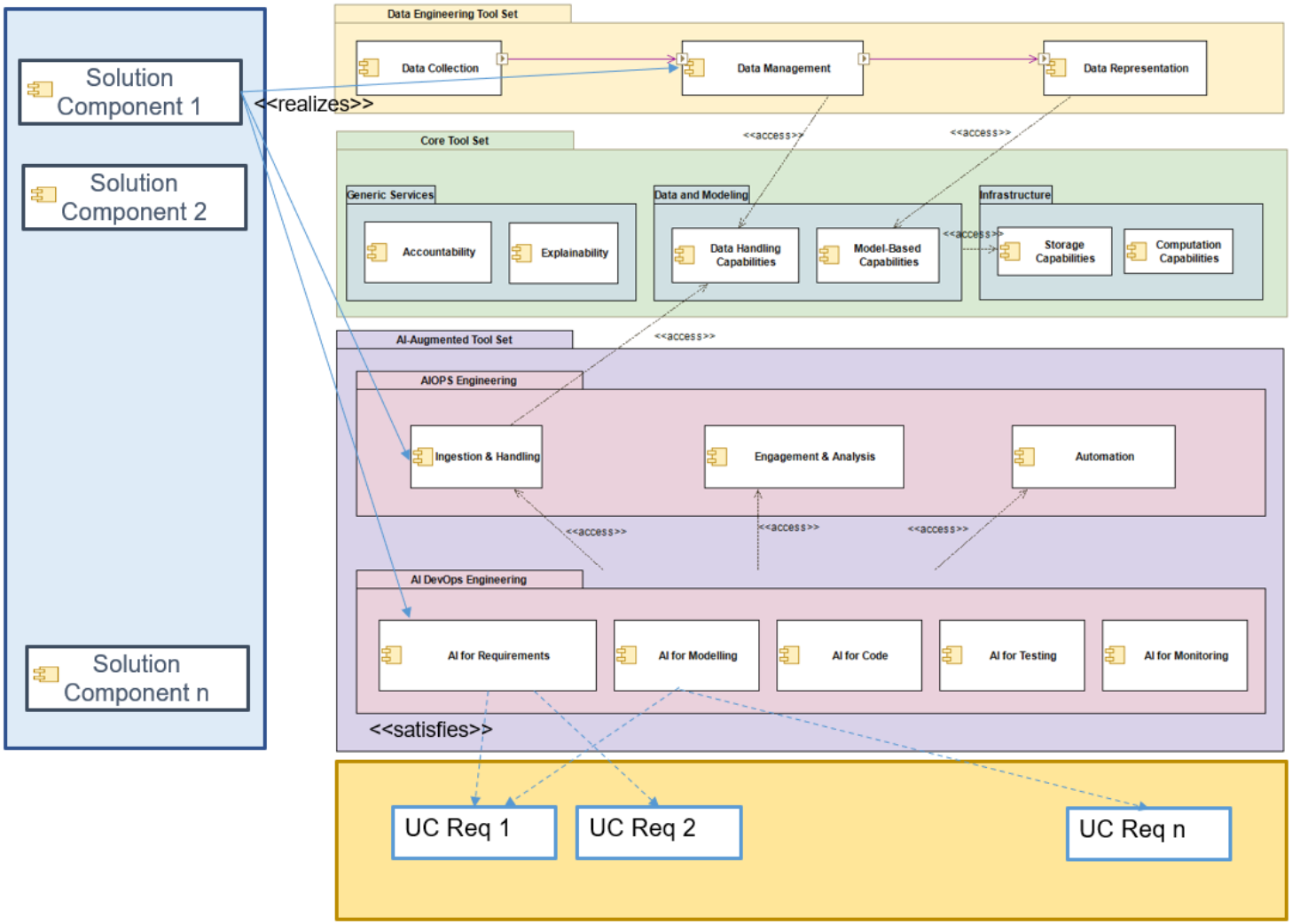

Figure 4 AIDOaRt Architecture (as of March 2022) and some example relations to use case requirements and solution components

2.3. Integration Approach: The AIDOaRt Integration Mediator Pattern

As outlined in the previous section, a systematic evaluation of the horizontal and vertical integration relationships and aspects between use case and solution providers is required. Furthermore, this systematic evaluation should consider certain constraints. This led to the idea of proposing a mediation mechanism between use case and solution providers to get a project-wide overview of potential and actual integration relationships.

In this section, a generic AIDOaRt integration mediator pattern is therefore presented as the central AIDOaRt integration approach. This generic approach will then be applied in Chapter 3 to the different integration aspects derived and resulting from several previous, ongoing or planned activities in WP1-5. These different integration aspects and the applied overall AIDOaRt integration approach will then describe the intended methods for capturing, monitoring, documenting and evaluating the integration status of the different use cases and solution components in the upcoming deliverables D5.2, D5.3 and D5.4

2.3.1. Definition of the AIDOaRt Integration Mediator Pattern

In AIDOaRt, Modelio[4] is used to capture various project-related aspects, activities and results. The corresponding AIDOaRt metamodel (see Figure 3[5]) is composed of components such as use case requirements, solution components and the generic AIDOaRt architecture (see Figure 4 and, for more details, D1.2). Corresponding relationships allow for an appropriate interpretation of each component for a given context, e.g., for which use cases a certain solution component could be a potential solution. During the project duration, the model will iteratively evolve and increasingly complete the overall picture of the AIDOaRt partners' activities, collaboration and joint results. In addition, through the use of document generators, the model will form the basis and the single source of truth for the content of many deliverables (including the upcoming deliverables D5.2, D5.3 and D5.4).

In this way, Modelio will be the central tool for monitoring and recording the integration status of the different components involved. If needed, the corresponding AIDOaRt module in Modelio will be extended to implement the general AIDOaRt integration approach and the AIDOaRt integration mediator pattern as described in this document. In addition, the AIDOaRt model will support ongoing discussions between partners by deriving concrete conclusions from a particular snapshot of the model. In this way, the Modelio-based approach will also support the negotiation phase and the integration process between use case and solution providers.

---

[4] The modelling tool Modelio is used in AIDOaRt project-wide to capture various aspects about the project structure, concepts, use cases, solution components and their relations (https://www.modelio.org/)

[5] The version shown in the figure is based on the terminology introduced by D1.1 (see this deliverable for more information). As it is still under development by the T1.2 task, it is still under development (the version used for this document is from March 2022). A final version with all details and a complete description is planned for the upcoming deliverable D1.4.

Figure 5 visually illustrates the intention of this AIDOaRt integration mediation model.

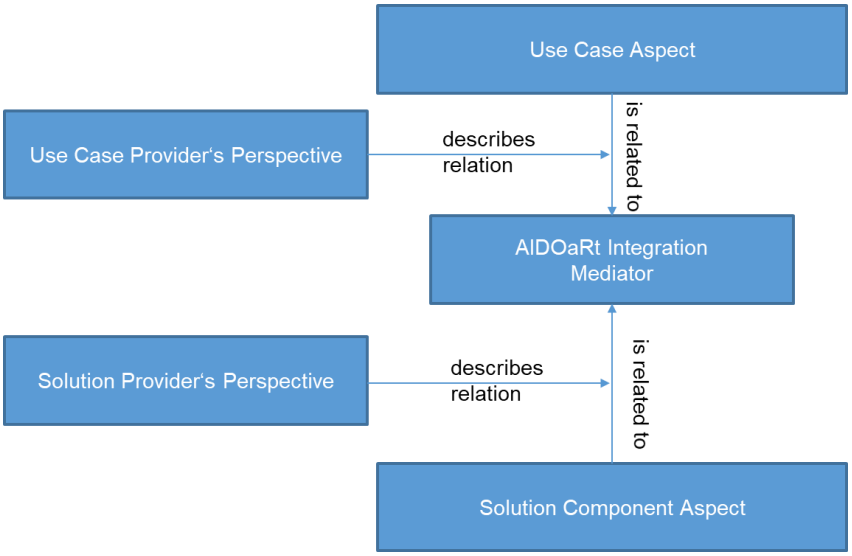

Figure 5 The AIDOaRt integration mediator pattern

The actual core of the model is an AIDOaRt integration mediation component. This component is usually represented by a corresponding AIDOaRt metamodel component (or set of components) used by the AIDOaRt Modelio model-based approach. The AIDOaRt integration mediator component is thus a placeholder for a specific integration aspect (e.g., a generic requirement derived from several use case requirements from different use case providers). Due to its high-level description, it supports a common understanding among all project participants for a specific integration aspect without the need to already agree on all details for possible collaborations and integration activities. Consequently, relevant use case aspects such as use case scenarios or use case requirements can be directly related to such a mediator component. In this mapping, the use case provider can bring in its specific perspective and constraints, as suggested in Section 2.2. The same applies from the perspective of the solution providers, who similarly map their solution components to this mediator component. As both the use case provider and the solution provider additionally describe their specific views of the relationship, the collected mappings will support the negotiation phase between the use case provider and the solution provider, which should eventually lead from a list of potential vertical and horizontal integration relationships to a concrete relationship reflecting the overall integration activities within the AIDOaRt project.

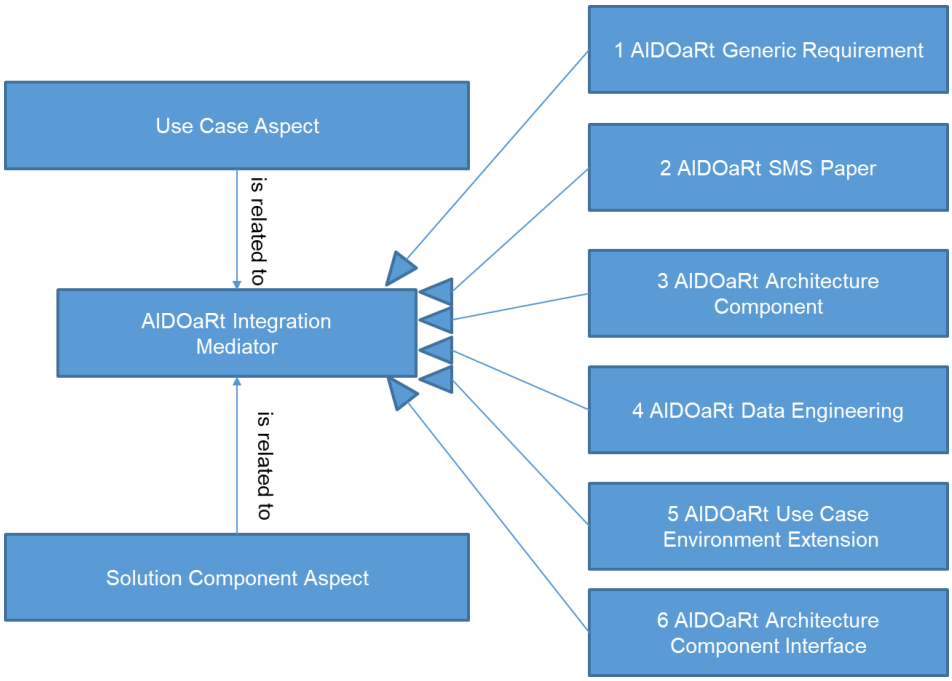

Figure 6 Concrete Integration Mediators

AIDOaRt integration mediator instances and integration aspects: This document now attempts to define concrete *AIDOaRt integration mediator instances* (see Figure 6) that relate to a number of *integration aspects*, which are outlined below and described in detail in Chapter 3:

- *AIDOaRt Generic Requirement:* Generic requirements are derived by WP1 (T1.1) from several use cases, in a specific phase of requirements analysis (see D1.1). Generic requirements are an important integration aspect as they define commonalities between different approaches and thus should drive both vertical and horizontal integration aspects.
- *AIDOaRt SMS Papers*: The systematic literature review conducted in WP3 (T3.1, reported by D3.1) is another candidate for supporting integration based on common methodologies that are considered state-of-the-art. Similar to the general requirements, the state-of-the-art aspects describe commonalities between solution providers (horizontal integration) and support the creation of a common understanding between use case and solution providers (vertical integration).
- *AIDOaRt Architectural Component:* The AIDOaRt architecture (defined by WP1, T1.3, reported by D1.2 in its first version) structures the different components that capture a variety of aspects needed for a complete AIOps-enabled environment. They are intended to divide an integration aspect into specific topics
- *AIDOaRt Data Engineering:* Since data engineering plays a central role for integration in the context of AI-based methods, special attention is given to this integration aspect. It refines to a certain extent the AIDOaRt Architectural Component aspect.
- *AIDOaRt Use Case Environment Extension:* For the integration of (existing) use case environments and (novel, innovative) solution components, appropriate environment extensions (e.g., adapters, data converters, remote access, etc.) are needed in the

integration process. This aspect is particularly important in the negotiation phase between use case and solution providers, as described in Section 2.2.

- *AIDOaRt Architecture Component Interface:* In order to finally couple and integrate the use case environment with solution components, corresponding (functionally and data harmonized) interface descriptions are necessary.

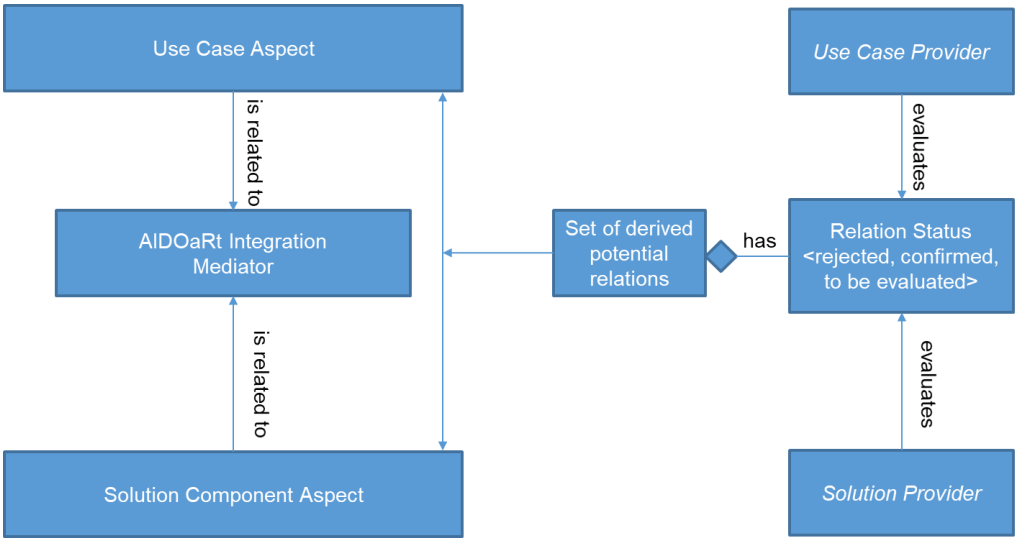

Figure 7 Deriving and evaluating the set of potential relations

2.3.2. Definition of Relation and Integration Status

Relation Status: Based on each defined direct relationship from a use case aspect to a concrete AIDOaRt integration mediator instance (provided by the use case provider) and a solution component aspect to a concrete AIDOaRt integration mediator instance (provided by the solution provider), a set of potential relationships between the use case aspects and the solution component aspects can be automatically derived (see Figure 7). Since the potential relationships are already a subset of all theoretically possible relationships between a use case and a solution provider, they can be interpreted as a first filter for promising candidates for vertical integrations. Practically speaking, such a filtered list of relationships in a large project like AIDOaRt should increase the efficiency of collaboration alignment and may reveal potential relationships that would otherwise not be discovered. However, each of the potential relationships needs to be further evaluated in terms of its validity and practicality (taking into account possible constraints) by the use case provider on the one hand and the solution provider on the other. As shown in Figure 7, each potential relationship should therefore have a relationship status that can be rejected (either the use case provider and/or the solution provider does not plan to integrate based on the corresponding integration mediator component), confirmed (the use case provider and the solution provider consider integrating) or still needs to be evaluated (further investigation is required, possibly at a later stage of the project).

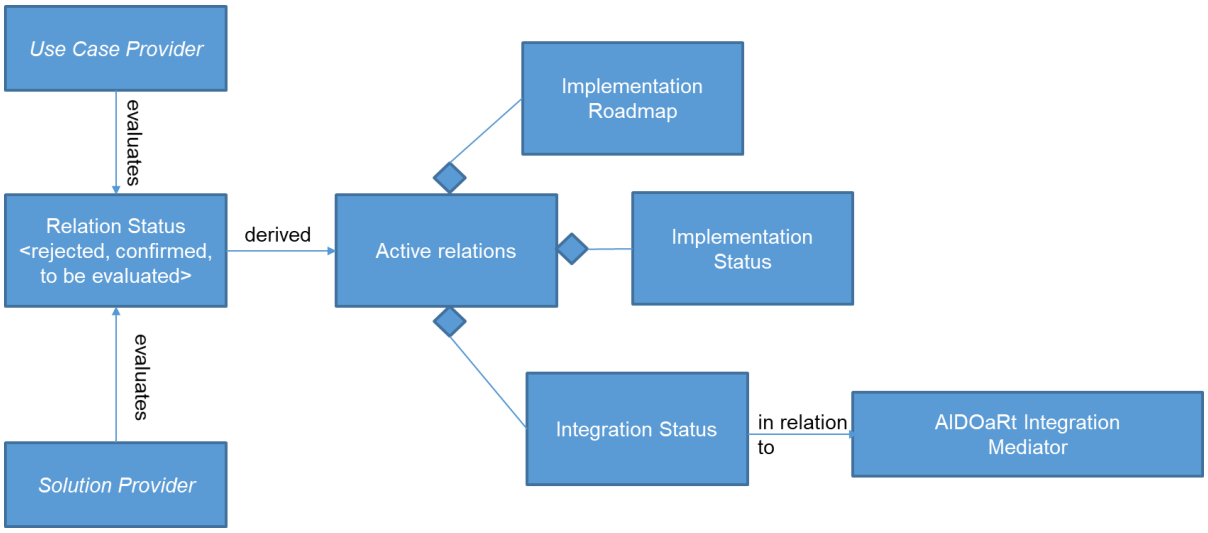

Figure 8 Capturing and monitoring active relations

This assessment is to be considered as ongoing and will be continuously updated during the first phases of the AIDOaRt project. Thus, the AIDOaRt Modelio model is used here as a living document that captures the current (vertical and horizontal) integration relationships of the different use cases and solution component aspects. As mentioned before, from the evaluation of the potential relationships, concrete active relationships between use case and solution providers can be derived. As previously mentioned, concrete active relationships between use case and solution providers can now be derived from the evaluation of the potential relationships.

Integration Status: As shown in Figure 8, each of the active relations is then described firstly in terms of its implementation roadmap and status of the components involved, but more importantly in terms of its *integration status* in relation to a concrete AIDOaRt integration mediator instance representing a specific integration aspect. Each integration status thus reports how the integration was achieved in relation to the involved AIDOaRt Integration Mediator component. For example, the report could reference a specific SMS paper describing a method that was applied and extended in the integration of use case A with solution component B, or if the referenced mediator element is a data standard, it could describe how solution component C was adapted to adhere to the standard and thus can be applied to use case D, whose use case environment depends on that standard.

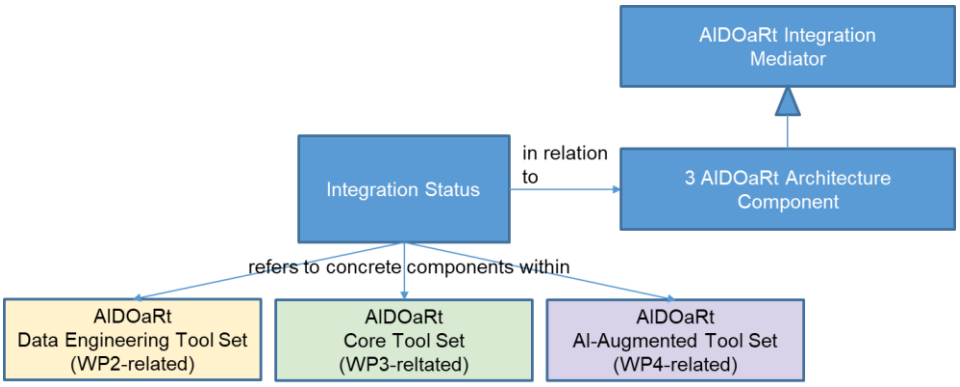

Figure 9 Example of possible references indicated by the integration status

Figure 9 illustrates another example for the AIDOaRt Architecture Component aspect. Here, concrete toolset components could be referred by the integration status. These toolsets are defined on a meta-level by the AIDOaRt architecture in Task T1.3 but will be more concretized by the activities of the various scientific work packages WP2, WP3 and WP4. Again, the AIDOaRt Modelio approach will support capturing these components and the relations relevant to monitor the corresponding integration status of them taking both horizontal and vertical integration strategies into account.

2.3.3. Definition of the AIDOaRt Mediator Integration Pattern Process

As mentioned above, this generic approach of defining relationships between such integration mediator component instances and use case components or solution components, followed by an ongoing assessment of the derived potential relationships between use case and solution providers and the associated integration activities, can be mapped to a specific process shown in Figure 10.

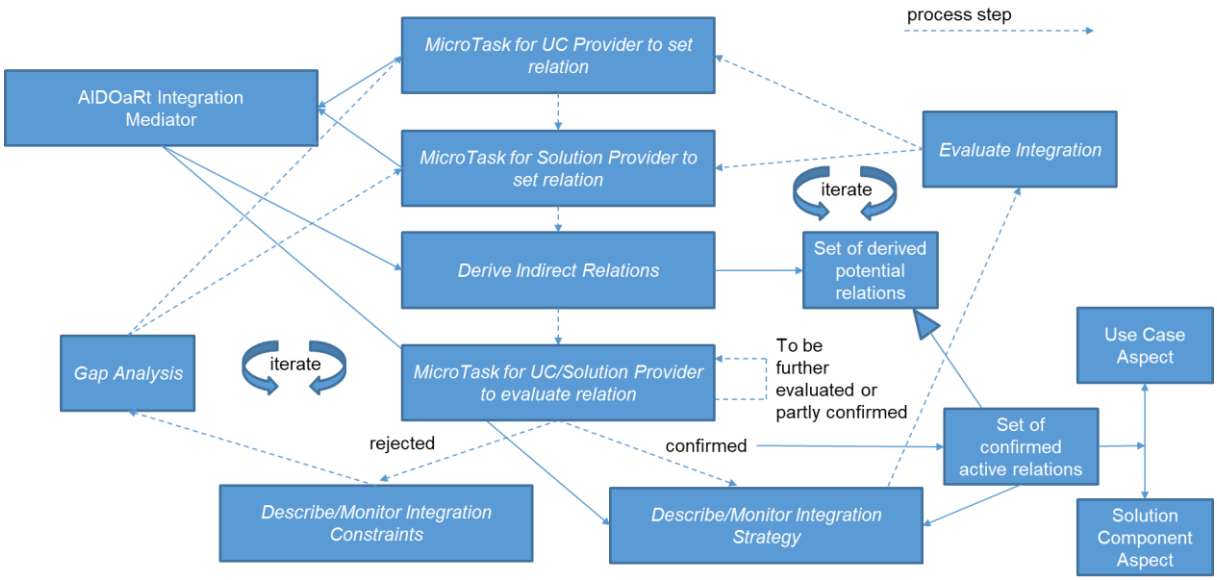

Figure 10 Mediator integration pattern process

As introduced by WP1 (see e.g., D1.1 for more details), so-called microtasks are regularly defined in the form of concrete instructions for both use cases and solution providers. They help to manage AIDOaRt activities, break down larger goals into smaller parts and track their implementation. In many cases, these microtasks consist of concrete Modelio modelling assignments to document a specific topic defined by the microtask. In addition to adding modelling elements and relationships, further descriptions are usually requested (either in text form or as Word snippets). A model enriched in this way is then the basis for a document generator, which has already been used for several AIDOaRt deliverables.

In the case of the AIDOaRt integration mediator sample process, corresponding micro-tasks are first defined for both use case providers and solution providers. A micro-task related to the use case provider requires adding and describing the corresponding relationship between a use case aspect (e.g., a use case requirement) and a concrete mediator element (e.g., a generic requirement), while a microtask related to the solution provider similarly requires a relationship and description between

an aspect of a solution component (e.g., a solution capability) and the same concrete mediator element (i.e., the generic requirement).

After completing these first two microtasks, the potential relationships between Use Case Provider and Solution Provider per concrete integration mediator instance are intended to be automatically derived by appropriate Modelio queries on the overall AIDOaRt Modelio model. Based on the list of potential relations further microtasks are derived and assigned, which request the evaluation of each potential relation by every individual involved AIDOaRt partner. In case the evaluation cannot be concluded at the moment the microtask has been announced, it will be scheduled for other microtasks at a later project stage.

Rejection of relations: In case, the potential relationship is rejected, the respective partner has to argue why. The collected rejections and descriptions (e.g., related to integration constraints such as described in Section 2.2.2) will lead to a gap analysis during T5.1, which evaluates if the reason for rejection has to do with integration constraints that may need to be overcome by advanced integration approaches beyond the current state of the art and should thus come in the focus of corresponding AIDOaRt research activities. Another issue that may occur is that Use Case Provider and Solution Provider may disagree about a specific potential integration relationship. These disagreements can be collected systematically and may lead to further discussions, potential collaborations or interesting research questions. Finally, it can be also the case that components remain with insufficient links (e.g., a dedicated solution capability is not assignable to any concrete use case requirement). One conclusion may be to remove this component then from the AIDOaRt project or to think about extensions (e.g., define another use case, if especially interesting from a research perspective), which again enables the integration of the respective components.

Confirmation of relations: If the potential relationship is confirmed, a set of active relationships can be derived. Based on this list of active relationships, periodic microtasks can be derived requesting a corresponding integration report from the partners involved on a specific aspect of integration. This report could include an *integration status* (see Section 2.3.2) and an *integration strategy* composed of the integration methods and techniques used, and the advancements applied beyond the state-of-the-art, as well as the establishment of relationships with the involved components of the AIDOaRt architecture in Modelio (and provided by the scientific work packages WP2-4). The collected reports will then be presented in the upcoming deliverables D5.2, D5.3 and D5.4 and can be largely generated by taking appropriate snapshots of the current AIDOaRt Modelio model.

2.3.4. Additional Conclusions on Gap Analysis and Integration Status

Some conclusions on *gap analysis* and *integration strategies*: As described in Chapter 4, the mediator pattern approach including the proposed process will be applied during the upcoming T5.1 activities (e.g., in the form of microtasks as proposed by Section 2.3) leading to follow-up deliverables, starting with deliverable D5.2 (M18). During this application, the described integration approach will be evaluated, improved and extended in parallel. This includes in particular the *gap analysis* and the *integration status*.

Gap analysis: Specific topics for gap analysis have been identified in this chapter but are not limited if further analysis becomes necessary. This also includes further integration constraints related to possible limitations between use case providers and solution providers. Special focus will be set on the analysis of orphaned use case or solution component aspects: If the application of the AIDOaRt integration mediator pattern followed by rejection of some potential vertical relations leads to unlinked components (e.g., a solution component aspect, which is not related to any use case aspect), the following three questions are raised in general: (1) is the orphaned component something, which is out of the scope of the project and should be thus removed, or (2) is the orphaned component very relevant to the AIDOaRt aims but the project set-up is incomplete and is required to be enhanced (e.g. by another use case), or (3) – as kind of special case of (2) – is the orphaned component an indicator for something beyond the current state-of-the-art and should be thus get more attention within the project alongside corresponding project enhancements (e.g. further solution components are required). If considered to be beyond the scope of the project, a backlog of unsolved issues may be a valuable input for any follow-up project and thus also a useful result.

Integration status: The integration status refers to the confirmed (actual) relationships (horizontal and vertical) between use case and solution providers. It further refers to concrete involved components related to the corresponding integration aspect (e.g., relevant papers, AIDOaRt architecture elements or toolsets, etc.). A detailed list of referred components and elements will be derived from the ongoing integration activities and the partner feedback gathered during the next integration iteration. Special attention will then of course be paid to the results of the WP2, WP3 and WP3 research work packages. As outlined in the AIDOaRt architecture, they will focus in particular on the *Design Engineering Tool Set*, *Core Tool Set* and *AI-Augmented Tool Set*. Consequently, an integration status will be accompanied by concrete references to specific solutions from these toolsets and how they will be integrated (integration strategy). For each area mentioned, horizontal integration in particular will play an important role for the chosen integration strategy in order to increase the impact of the solutions offered within the AIDOaRt project, but also beyond. Where applicable, the AIDOaRt Modelio modelling approach will support this by linking confirmed integration relationships to these elements of the AIDOaRt architecture. A more detailed definition of what comprises a gap analysis and an integration status (including an integration strategy) will therefore be presented as part of the follow-up deliverable D5.2.

2.4. Summary

Based on different overall integration goals from the perspective of the AIDOaRt project, this chapter first discusses different *integration issues and constraints* from the perspective of the *use case and solution providers*. *Vertical and horizontal integration strategies* are presented as well as a generic *AIDOaRt integration mediator pattern* that supports the integration relationship between the parties involved. The pattern was then concretized by various instances in the form of *integration aspects*, which are explained in more detail in the next Chapter 3. Furthermore, the mediator pattern is accompanied by a specific iterative process in the form of microtasks to be completed by the AIDOaRt partners and is supported by further evaluations such as a *gap analysis* and the definition of an *integration status* (including the announcement of in integration strategy planned for the

upcoming follow-up deliverables D5.2-4). Both gap analysis and integration strategies have been briefly outlined by this document but are subject of further refinements reported by the follow-up deliverables D5.1-4. On this basis, it is not excluded that other aspects of integration will be discovered as additional instances of the AIDOaRt integration mediator pattern - if so, this will be reported accordingly.

3. Integration Aspects

In this chapter, different integration aspects are presented in the context of concrete instances of the generic AIDOaRt integration mediator pattern (or in case of Section 3.7 further potential instances). For each integration aspect, the corresponding instance is thus presented and described, and its application is explained using an abstract example. This example also serves as a connecting element between the individual sections and is used to discuss the mutual dependencies between the integration aspects and to show how vertical and horizontal integration relationships can be derived through the given approach by applying a specific process as presented in the previous chapter.

3.1. Integration Aspects on Generic Requirements Level

In WP1, especially through the activities in Task T1.1 (Use Cases Requirements), several use case provider case studies were analyzed, resulting in a set of use case scenarios and use case (data) requirements. Based on this work, specific working groups analyzed these use case requirements depending on the different DevOps phases (requirements development, modelling, testing, coding and monitoring). Each working group developed a set of high-level generic requirements per DevOps phase, which were initially derived from the existing use case requirements. These initial generic requirements were then refined and supplemented by several well-defined micro-tasks in close cooperation with the respective use case provider. Appropriate links were made between the use case requirements and the generic requirements. In some cases, the generic requirements were even defined by several abstraction layers. The relationships between the generic requirements and the use case requirements even allowed the definition of so-called common requirements, i.e., generic requirements that have an above-average number of relationships to specific use case requirements. This would mean that the implementation of such common generic requirements has a high potential for commonalities and synergies between the provided case studies, suggesting, in particular, a number of horizontal integration aspects. Further details on the specific (common) generic requirements and the process of the requirements analysis phase can be found in Deliverable D1.1.

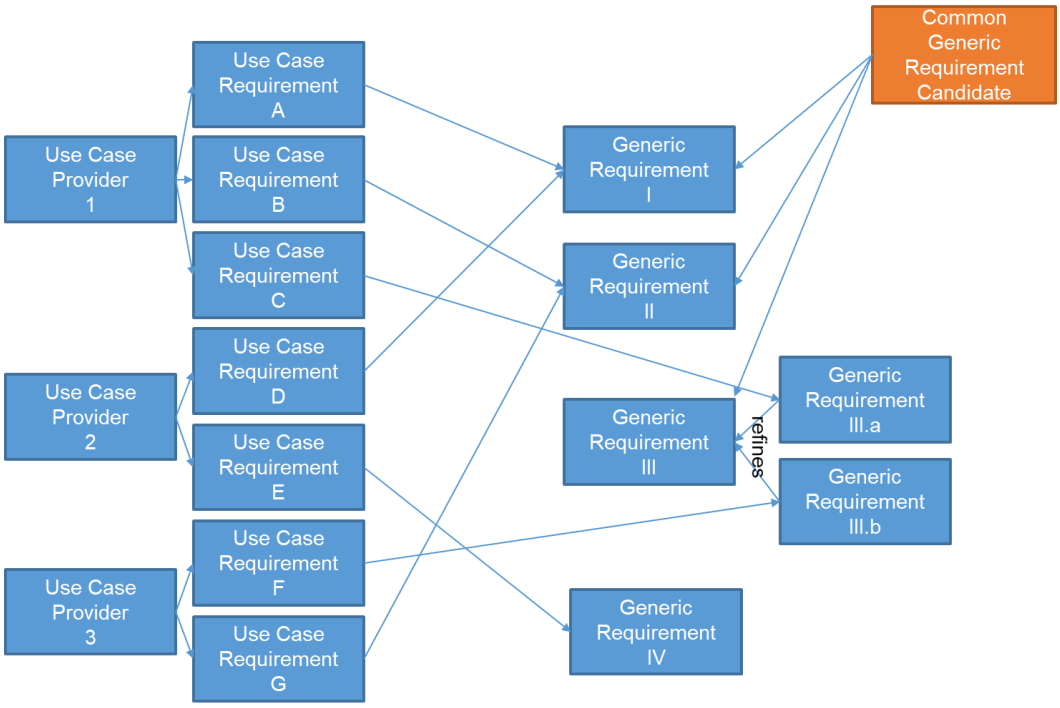

Figure 11 Example about use case requirement and generic requirements

Example: To analyze the impact on the integration aspects, let's take a closer look at a theoretical example. Figure 11 outlines six use case requirements (A-G) provided by three use case providers (1-3). During the analysis phase, four generic requirements (I-IV) are derived from these use case requirements in several iterations between use case providers and specific workgroup leaders (the assignment of a generic requirement to a specific DevOps phase is not considered in the example). In the case of generic requirement III, some specificities of the respective use case requirement led to the definition of two additional generic requirements (III.a and III.b). In this example, generic requirements I-III now have a link count of 2, while generic requirement IV has a link count of 1.[6] Consequently, generic requirements I-III are above average in their link count and are therefore selected as common generic requirement candidates.

Common generic requirements now have particular integration implications. Since each candidate is relevant to more than one use case provider, potential solution components that fulfil the corresponding common generic requirement have a greater impact than solutions that implement a generic requirement with a below-average number of links. However, this impact only occurs if the corresponding component implementation is universally integrable. In other words, solution components that address common generic requirements impose higher requirements on integration aspects than others, lead to different horizontal integration activities, and may face specific integration constraints as listed in Section 2.2.

[6] In AIDOaRt and according to D1.1, 10 Use Case providers are defining 30 generic requirements linked by 78 use case requirements. At this stage, the average link count was four, leading to 16 common generic requirement candidates.

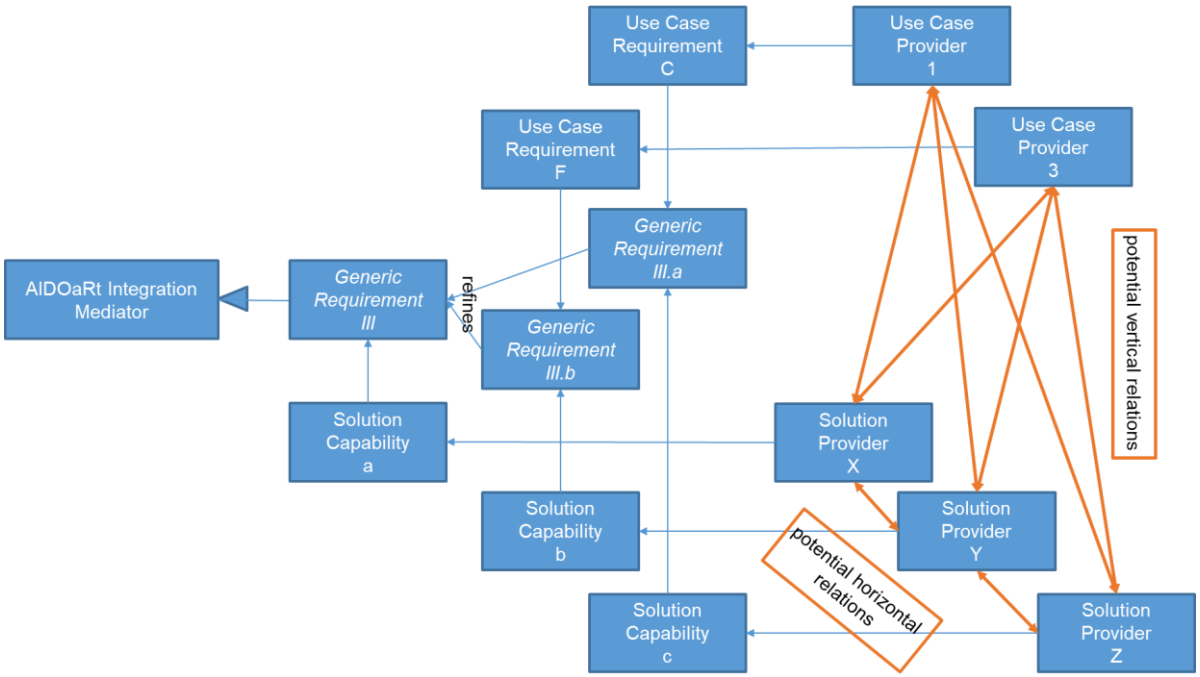

Figure 12 Example about deriving potential relations between use case and solution providers

In WP1, therefore, the solution providers were asked by a corresponding microtask to relate their solution component capabilities to the iteratively defined generic requirements (see D1.3 for more details on the outcome). In the abstract example presented in this section, three solution providers (X-Z) offer three solution components (a-c), as shown in Figure 12: While solution component a is linked to the higher-level generic requirement III, solution components b and c relate to generic requirement IIIa and IIIb respectively. This now means that each solution provider X, Y and Z can potentially offer a solution that can be integrated for more than one use case provider. Furthermore, since each solution competence can potentially be helpful for each use case provider, they may even need to interact and integrate with each other. This is now referred to as horizontal integration (solution capabilities a-c are integrated against each other) and vertical integration aspects (solution capabilities a-c are integrated into use cases). In this sense, different integration requirements are considered (e.g., what kind of common interfaces, data formats, methods, state-of-the-art techniques, etc.) need to be applied and aligned. However, these requirements may become less important if the solution capabilities cover a different aspect of the use case, e.g., solution capability a refers to the data acquisition process, while solution capability b is about the explainability of learned data (data acquisition and explainability are two examples from the list of AIDOaRt architecture components). Therefore, further evaluation of potential relations as suggested in the AIDOaRt mediator pattern process is inevitable.

The attentive reader will have noticed that these integration requirements are indeed the subject of the following sections, where all the aspects mentioned are mapped to the corresponding integration facilitator elements. However, before diving deeper into these areas, it is worth exploring in more detail the potential relationships between use case providers and solution providers as shown in Figure 12. In the event that it turns out that not all potential relationships are relevant in

practice, the overall reconciliation on the necessary integration requirements in the following sections can possibly be simplified from the outset.

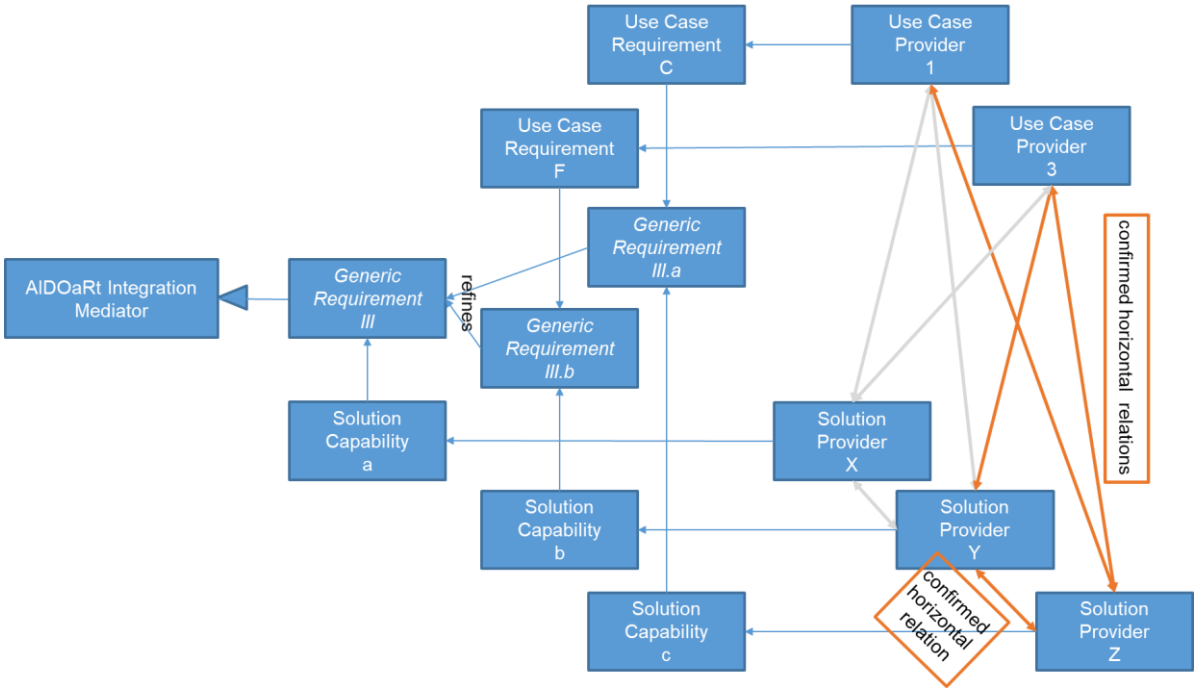

Figure 13 Potential relations after applying the AIDOaRt integration pattern process

This evaluation should be carried out using the evaluation process proposed in Section 2.3. Figure 13 shows a possible outcome of this process: Each automatically derived potential (vertical) relationship is further evaluated by the use case and solution provider. In this example, the partners involved conclude that the solution capability a is not relevant for the use cases of use case providers 1 and 3. In this case, a gap analysis should discuss whether solution capability a is thus no longer relevant for AIDOaRt and thus no integration is required. However, it may also turn out that the topic can be relevant to the AIDOaRt project even without a dedicated use case and should therefore remain. In this case, perhaps an abstract, non-industrial use case should demonstrate the usefulness of the solution capability. It may even turn out during the project that with such a demonstrator there is applicability to other use cases not yet mapped to generic requirement III. In this case, the corresponding link will be established in the AIDOaRt Modelio model and further investigations will be initiated in a later iteration phase of the AIDOaRt project.

This outcome also has an effect on the horizontal relationships between solution providers X, Y and Z. Since solution provider X is not involved any more into horizontal relationships, the conclusion may be valid that horizontal integration between all mentioned solution providers is not required any more. However, independently of established vertical relationships, horizontal relationships may still make sense from a perspective, which goes beyond the use case provider's scope involved in the AIDOaRt project. This could again be analyzed and discussed by the gap analysis of the AIDOaRt integration mediator pattern process.

Another consequence of the removal of solution capability a in the context of use case providers 1 and 3 would be that there is no longer a directly linked generic requirement III. However, the impact seems to be limited as the refined generic requirements III.a and III.b are still covered and thus still cover the overarching generic requirement III.

Furthermore, it has been shown in the example that solution capability b only seems to be relevant for use case provider 3, which potentially facilitates the integration requirements in the case of solution provider Y. This judgement could be made in the case that the SMS study implies such a treatment, which would provide another link between this section and Section 3.2.

Finally, the solution capability c is considered relevant for both providers of use cases 1 and 3, so that further horizontal integration aspects should be considered, especially those presented in the following sections of this chapter.

3.2. Integration Aspects on State-of-the-Art Level

A systematic literature mapping study (SMS) was conducted in task T3.1 (Foundation for MDE and AIOPS in DevOps) of work package WP3. In this context, 1062 publications were screened and checked for their relevance to the AIDOaRt project. Among them, 133 papers have been selected and considered relevant for AIDOaRt. The screening was based on various criteria, such as belonging to more than one AIDOaRt pillar (i.e.: Model-Driven Engineering (MBE), Artificial Intelligence (AI) or DevOps), the relation to specific DevOps phases (in AIDOaRt: requirements engineering, modelling, coding, testing and monitoring). In addition, the peer-reviewed papers were categorized, for example, according to the respective application domain (e.g., automotive, maritime, etc.) or marked as domain-independent. For further details, please refer to the content of D3.1.

For horizontal or vertical integration activities, the agreement should be reached on which state-of-the-art methods (such as algorithms, learning methods, data analysis techniques, etc.) and tools/frameworks the integration is based on. Reference to the relevant literature should help to narrow down the integration decision and act as a catalyst for discussion between the partners involved in a particular integration activity.

Once the relevant literature for each integration is captured, this will support the documentation of the integration approach for the upcoming deliverables such as D5.2, D5.3 and D5.4. There, the final technical implementation of the approach presented in the remaining section is also mentioned, as the SMS was conducted using a spreadsheet-based approach, while many other elements relevant to this section are represented by the AIDOaRt Modelio model-based approach, and will therefore most likely be a pragmatic combination of both.

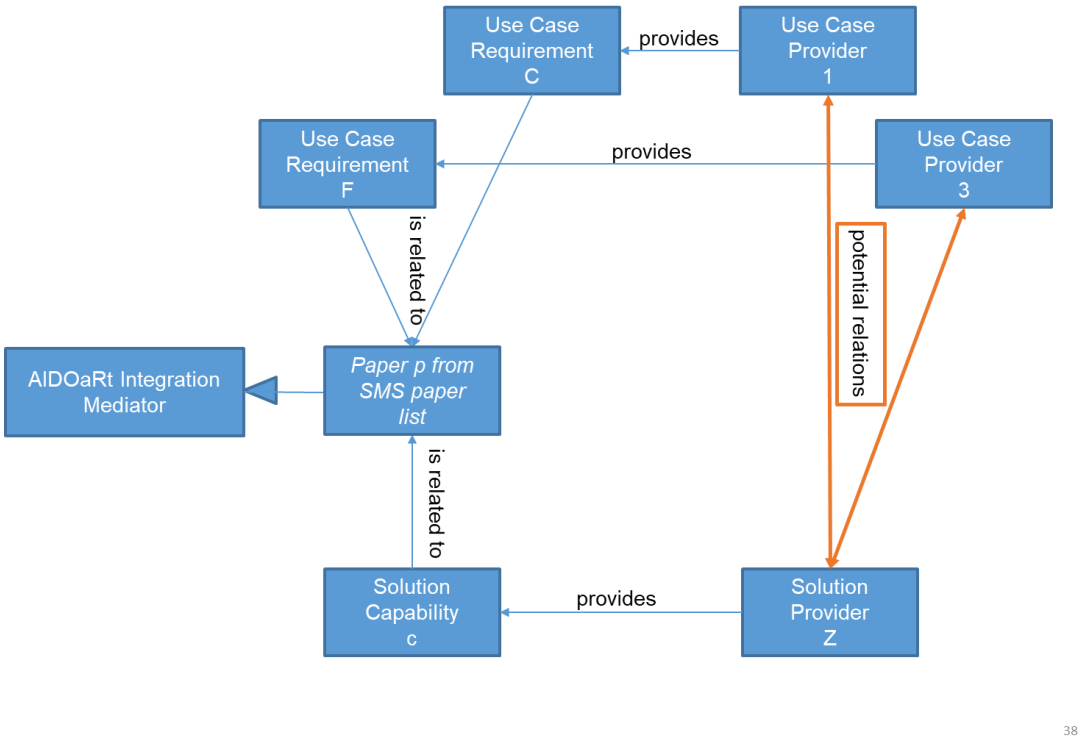

Figure 14 Applying papers from the SMS paper list as an integration mediator element

Example: Figure 14 revisits the example from Section 3.1 to illustrate reasonable vertical integration (i.e., between a solution capability and use case requirements) using a selected paper from the SMS paper list. Using a similar process in the form of well-defined microtasks, use case providers and solution providers are asked to relate their use case requirements and solution capabilities to the relevant papers from the SMS paper list. Once this task is completed as in the given example (use case requirement F and C match paper p, the same applies to solution capability c), potential integration relationships between use case providers and solution providers can be derived. To reduce the number of potential relationships at this stage, the results of the microtasks described in Section 3.1 can be used as a kind of relationship pre-filter: For example, if solution provider X confirms that solution capability a is also related to paper p, then the potential relationships between use case providers and solution providers are expanded to include solution provider X. This would reduce the number of potential horizontal relationships at this stage. In fact, this would include a potential horizontal integration between solution provider X and Z. However, since the previous evaluation of the potential relationships between use case providers 1 and 3 on the one hand and solution provider X on the other has already been rejected by the analysis of generic requirement III, the list of potential relationships in the example given in Figure 13 can be reduced to the two relations indicated with bold arrows in contrasting color.

This now reduces the required discussion between use case providers and solution providers to review or reject the potential vertical and horizontal integration relationships. On the other hand, additional potential relationships introduced by a paper from the SRL paper list can also be useful information for the gap analysis mentioned in Section 2.3 and illustrated there by Figure 10. The

practical application of the process in the next steps of the AIDOaRt project will show what is actually helpful and manageable in this respect.

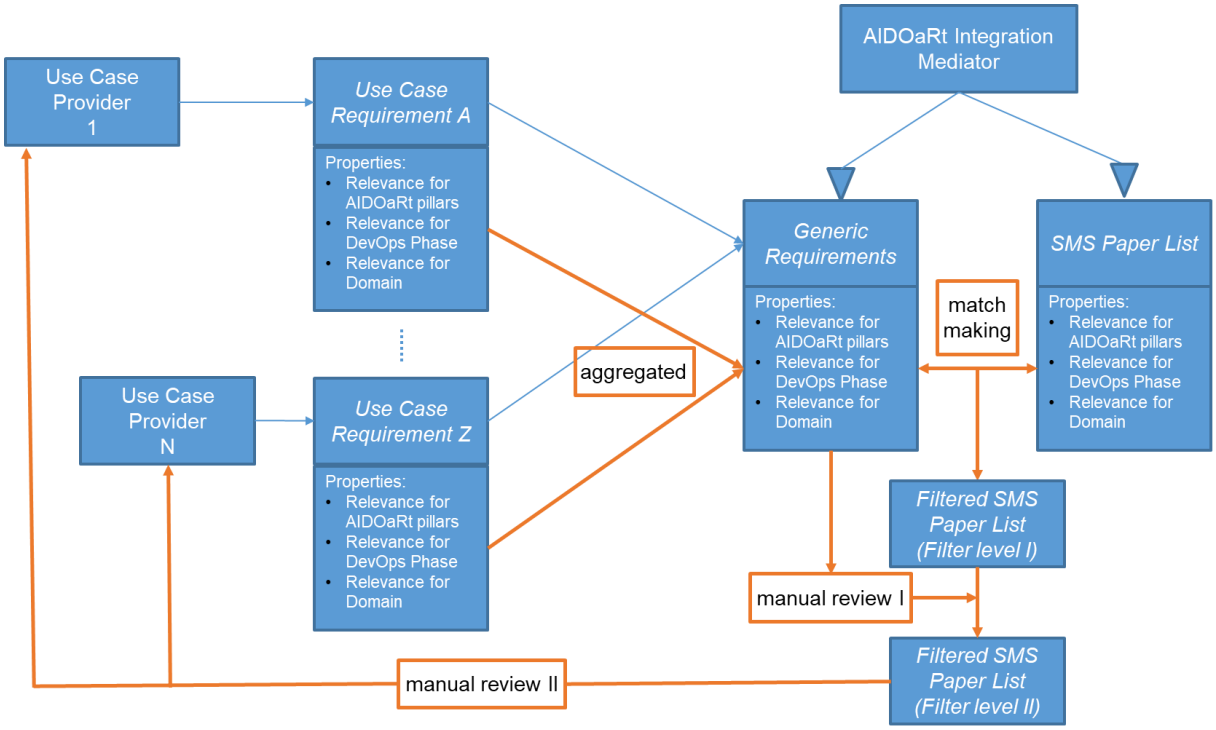

Figure 15 Process to filter paper list

As the list of reviewed papers in the SMS led to 133 AIDOaRt relevant papers, the effort for both use case providers and solution providers to assign each paper to a use case requirement and a solution capability, respectively, would be high. In order to reduce the required effort and to facilitate the mapping, a process to filter the paper list for the use case provider is proposed in this section and is illustrated by Figure 15 (a similar process is planned to be developed for solution providers): As already mentioned, every paper in the SMS paper list has dedicated properties the describe

- their relevance for the three AIDOaRt pillars: AI, MDE and DevOps
- their relevance for a dedicated AIDOaRt DevOps phases: Requirement Engineering, Modelling, Coding, Testing and Monitoring
- and their relevance for a dedicated application domain such as automotive or maritime.

Similar information can be found in the definitions of use case requirements (AIDOaRt pillars) or related components such as use case scenarios (AIDOaRt DevOps phases) and is defined in D1.1 and D1.3 (cf. also in the AIDOaRt Modelio model used to produce these documents). In combination with the modelled relationships between use case requirements and generic requirements, this information provided by the use case requirements can now be automatically aggregated to each generic requirement. Based on these aggregated generic requirement properties, a matchmaking process is planned to be applied between the properties of the generic requirement and the properties of the SMS paper list, resulting in a first set of automatically filtered SMS papers (filter level I).

In T1.1, five specific working groups were set up to derive generic requirements from the given list of use case requirements. Each working group was specialized in reviewing the requirements in the context of a specific AIDOaRt DevOps phase. Since the filtered paper list also provides the information of the associated DevOps phase, the list could be divided into five groups and could then be passed to these working groups accordingly. Each working group then in turn conducts a brief review (e.g., by screening the paper abstracts supported by the SMS team) of the individual paper groups to see whether the paper also meets the generic requirements in terms of content. This should lead to a further pre-filtered paper list (filter level II). This paper list is then forwarded to each use case provider by applying another filter (filter level III, not shown in Figure 15): Since each paper is associated with a generic requirement and each generic requirement is associated with a use case requirement, only those papers are forwarded to a particular use case provider whose use case requirements have a reference to a corresponding generic requirement with which a paper is associated. Further filters (filter level IV, not shown in Figure 15) related to relevant DevOps phases, AIDOaRt pillars or application domains can be applied individually per use case partner. This should result in a reasonable number of contributions that should eventually be reviewed per use case partner to relate each use case requirement to the appropriate state-of-the-art literature.

Again, orphan use case requirements could provide helpful feedback for a gap analysis. This analysis could, for example, lead to the conclusion that the SMS paper list is either incomplete and needs to be updated with further papers, or that the intended requirement requires methods, which have not yet been identified nor addressed by the scientific community and thus go beyond the state-of-the-art being researched within AIDOaRt. This analysis could also lead to a kind of innovation level aimed by the AIDOaRt project by implementing novel solutions for the given use cases.

This section presented a method to match a filtered paper list to the appropriate use case requirements by applying different matchmaking approaches. A similar process could then be applied to solution providers and their solution components. Once they have also established relationships with the relevant papers, potential relationships can be derived as described at the beginning of this section. Whether such filtering and matchmaking approaches lead to meaningful insights will be the subject of corresponding microtasks in T5.1. D5.2 will report on the outcomes of the concrete application of such an approach and on its possible extensions for solution providers.

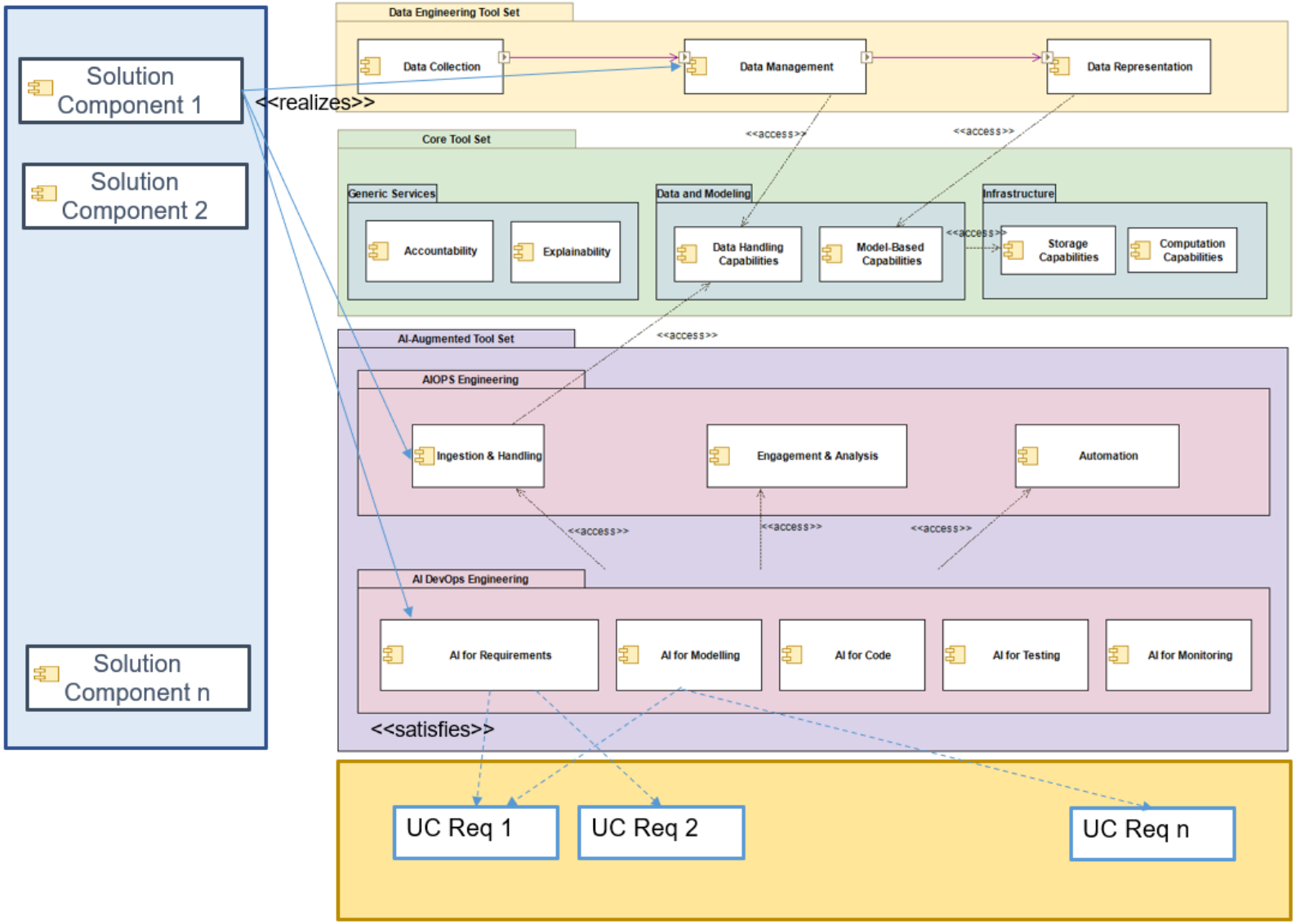

Figure 16 AIDOaRt architecture components and its relation to use case requirements and solution components

3.3. Integration Aspects on AIDOaRt Architecture Component Level

The definition of the AIDOaRt architecture is a joint effort of several AIDOaRt work packages (in particular WP1, WP2, WP3 and WP4). It is an inherent part of the AIDOaRt Modelio modelling approach and the associated metamodel and is documented in detail in deliverables D1.2 (first version) and D1.4 (final version).

Figure 17 illustrates the different AIDOaRt architecture components grouped in several sections (D1.2 explains their meaning in detail). Corresponding microtasks in the context of T1.3 asked both use case and solution providers to relate the use case (data) requirements or solution components to these architecture components. In a more abstract sense, this again corresponds to the generic AIDOaRt integration mediation pattern as shown in Figure 17. In this sense, however, the architecture component, e.g., data management, takes on the role of an integration mediator element between use case and solution provider.

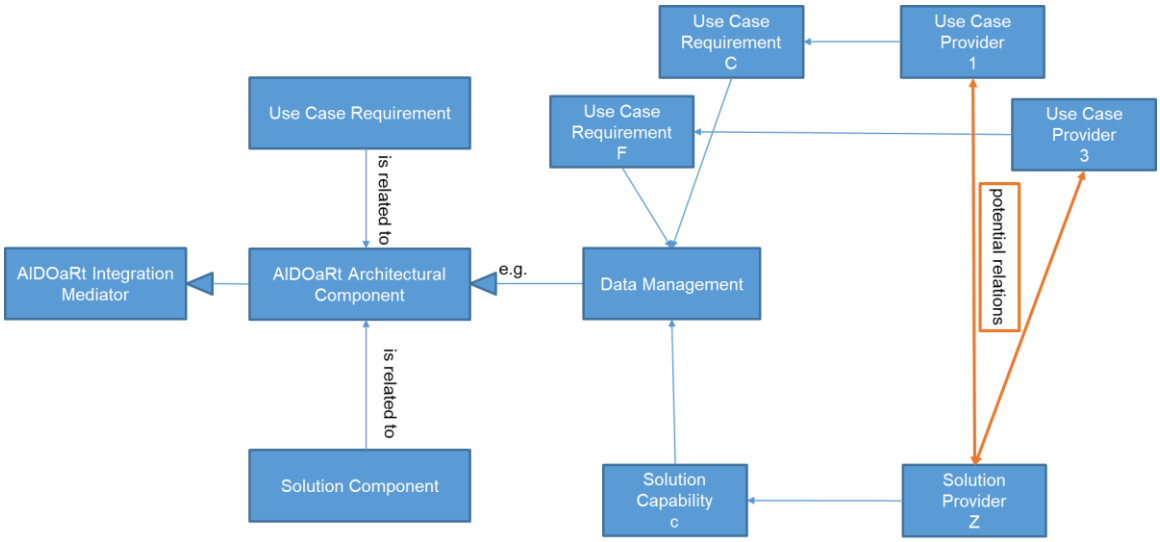

Figure 17 Potential integration relations between use case provider and solution provider based on architectural components

Example: In the given example (reused from the previous sections of the current chapter of this document), both the use case requirements C and F and the solution capability c are linked to the architecture component Data Management. From an integration perspective, this means that the potential relationships between the use case provider and the solution provider, as shown in Figure 17, represent a required alignment in terms of the applied data management techniques, methods and tools. In the case of data management, data format and standard aspects can play an important role for integration, but also the required data pre-processing and so forth. With regard to vertical and horizontal integration, synergies, harmonization and reuse aspects should be discussed in order to validate or reject the list of potential relationships proposed in Section 2.3. Validation and rejection should also contribute to the gap analysis in case e.g., use case providers ask for corresponding solution capabilities, but on closer inspection the potential relationships cannot be met in practice. To close the gap, appropriate solution components would then need to be recruited outside of existing AIDOaRt solution providers (e.g., by integrating open-source solutions).

3.4. Integration Aspects on Data Engineering Level

Since AIDOaRt focuses on AI-augmented DevOps Engineering, or AIOps for short, data plays a central role in various integration aspects. The AIDOaRt architecture takes this into account by defining a separate area called Data Engineering Tool Set (see Figure 18).

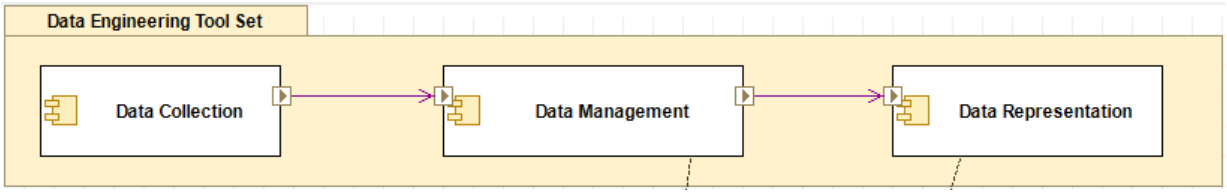

Figure 18 The Data Engineering Tool Set section in the AIDOaRt architecture

The Data Engineering Tool Set area contains three relevant elements, namely *data collection*, *data management* and *data representation*. All three elements are relevant for further integration aspects between use cases and solutions (e.g., when a use case has different requirements for data management capabilities that are provided by a solution component and should therefore be integrated). Therefore, corresponding relationships between use case requirements and solution capabilities, as discussed in Section 3.3, are further refined in terms of their data engineering integration aspects.

However, there is one element that requires a certain amount of attention in terms of integration and is therefore the focus of this section: *data representation*. Data representation drives the concrete integration of the use case environments and the solution components to be integrated on a deeper level compared to the integration aspects of the previous sections (it can be classified more on the level of component interfaces). Considering the representation of data is essential for successful integration and can become a showstopper if the matchmaking between use case environments and solution components fails. Since constraints on both use case providers and solution providers can cause such failure (e.g., depending on a certain standard required by some regulatory requirements), early identification of such potential failures is necessary and alternative integration strategies, resources and efforts need to be planned.

Therefore, the first activities to start a joint discussion were already initiated by WP1 and task T1.1 by defining data requirements in addition to the ordinary use case requirements. Each data requirement was related to a specific use case scenario in order to put it in a specific context that would be easily understood by potential solution providers (see deliverable D1.1 for details). All these activities were carried out in close cooperation with WP2 via cross-work package workshops.

At the time of writing, WP2 will be refining this work, leading to D2.1 (Data collection and representation – initial vers.). As these activities are being carried out in parallel and the final results of D2.1 are expected at the time of submission of this document, not every aspect of these results and the integration activities derived from them can be included in this document. However, as it is planned to refine the integration concept presented in this document through the upcoming

integration tasks, an updated version of this section is expected to be presented in the follow-up deliverables D5.2, D5.3 and D5.4.

Independent of these time constraints for the project, the integration concept will be based on a refinement of the data representation element of the AIDOaRt architecture on the one hand and the definition of the data requirements on the other. For this, the data requirements will first be extended by further attributes based on the results of the WP2 workshop than those originally defined by WP1. Subsequently, appropriate microtasks will establish corresponding links between the extended data requirements and an updated version of the data representation element of the overall AIDOaRt architecture. This is to implement the AIDOaRt integration mediator pattern presented in this document, in particular for data aspects shown in Figure 19.

During the WP1 activities, data requirements were collected for both use cases and solutions. In the case of use cases, they were related to the descriptive use scenarios and in the case of solutions, they were related to the current solution component. These relationships were captured using the AIDOaRt Modelio modelling approach. At the current state of the model, the relationships between use case and solution data requirements are only implicitly defined, e.g., via the relationships between use case requirements and solution capabilities explained in Section 3.1. Similar to generic requirements, the data representation element can now generalize different data requirements by introducing additional metadata of the involved models/metamodels that will facilitate the following steps: (1) generalization of models and (2) matching these models with solution data requirements. All these aspects are currently being discussed in WP2 (and will be reported one of the upcoming deliverables D2.1 or D2.2), but include questions such as: does the data represent a domain model (e.g., images, video, numeric sensor values in time series, etc.) or a meta-model (e.g., compliant with a domain specific language, higher-level programming language) or what types of standards are involved, etc.

Corresponding microtasks ensure that both use case data requirements and solution data requirements are mapped to these types of data representation aspects, as shown on the left side of Figure 19. Similar to what was described in the previous chapter, implicit relationships between use case and solution providers can be derived from these links, which need to be further evaluated through a specific evaluation process as proposed in Section 2.3.3. However, as the approach presented in this section is intended as a refinement of the integration mediator pattern presented in Section 2.3.1, the relationship information from other utilizations of the integration facilitator pattern should be taken into account to further reduce the list of potential relationships between use case and solution providers from the outset (thus optimizing the required integration effort).

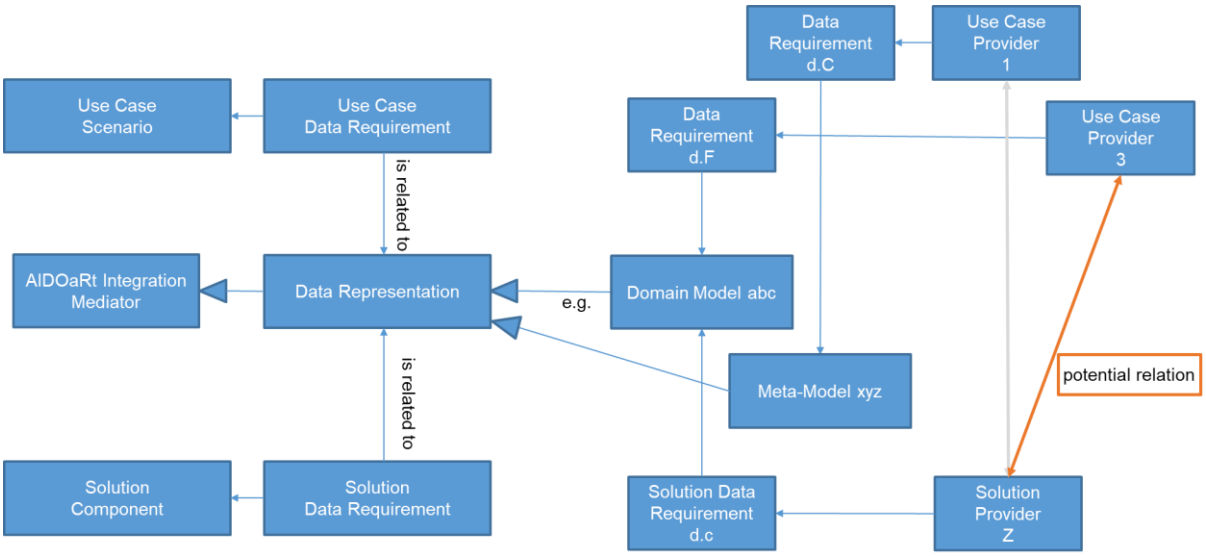

Figure 19 Data Representation as integration mediator element on data level integration

Example: The right part of Figure 19 thus continues the example from the previous sections, which state a vertical relationship between the use case providers 1 and 3 on the one side and solution provider Z on the other. In the example of this section, additional data requirements d.C and d.F are introduced (e.g., which are defining certain data aspects of the original use case requirements C and F). In this example, the data requirement d.F refers to the domain model abc, which is also fulfilled by the solution data requirement d.c and thus can be provided by the associated solution component of the solution provider Z. This strengthens the relationship between the use case provider 3 and the solution provider Z, and the integration activities are based on the use and/or extension/improvement of the domain model abc during their collaboration.

In contrast, the solution's data request is not related to the metamodel xyz, which is thus not initially provided by the corresponding solution component. This now weakens the relationship between use case provider 1 and solution provider Z, since the data requirement d.C is not covered by involved solution component of solution provider Z. For this situation, the integration mediator pattern process model provides a gap analysis to clarify whether the data requirements on both sides are either conflicting or can be resolved by investing additional resources (possibly on both sides) to close this gap (e.g., solution provider Z implements meta-model xyz as an additional feature of its respective solution component).

This situation, as D1.1 and Section 2.2 summaries, is assessed as a negotiation phase between use case and solution providers expected in AIDOaRt at data rather than feature level. The associated gap analysis and possible strategies to close the gaps are therefore intended to add value beyond the state-of-the-art in the context of AIOps-based approaches. In other words, the gap analysis should focus more on data level aspects than on the integration mediator pattern components presented so far. This means, however, that the related integration activities (designed to practically close the discovered gaps) will potentially represent a large part (if not the majority) of the overall integration activities. The forthcoming findings D5.2, D5.3 and especially D5.4 will return to this assertion with further analysis and conclusions.

3.5. Integration Aspects on Use Case Environment Extensions

The application of the generic AIDOaRt integration mediation pattern and the associated process described in the previous sections resulted in concrete collaborative relationships for both horizontal integration (i.e., between use case providers or between solution providers) and vertical relationship (i.e., between use case and solution providers). Depending on the context of the relationship (e.g., specific (generic) requirement, selection or extension of a state-of-the-art methodology, a specific AIDOaRt architectural element or a data representation), concrete integration activities will be derived and harmonized. These activities will be described in detail in the upcoming deliverables D5.2 (M18), D5.3 (M24) and D5.4 (M36) in different versions depending on the progress of the project.

While the previous sections mainly focus on integration discussions of advanced AIOps features and their harmonization to extend their impact and universal applicability, the following aspect has only been a side story so far: how does the given (legacy) use case environment need to be extended to be AIOps-ready and practically integrate novel AI-enhanced approaches into existing DevOps chains. In other words, the necessary integration requirements have so far tended to be defined top-down (i.e., from the use case provider to the solution provider). Use case providers defined their use case requirements and requested matching components from solution providers.

This section focuses instead on bottom-up integration aspects, some of which have already been addressed in Section 3.4 on data-level integration and discussed in Section 2.2 in the context of possible integration constraints depending on the view of the use case or the solution provider. There, a so-called negotiation phase between use case and solution providers was mentioned in the context of data requirements. The solution providers are here more in a position to drive the integration aspects, as without the fulfilment of certain data requirements defined by the solution provider by the use case provider, even the best implementation of a solution component is likely to fail (e.g., certain data quality requirements such as sufficient corner case representation in learning datasets).

Similarly, existing use case environments and architectures may need to be extended, adapted or modified to enable the integration of the required solution components, regardless of their flexibility or universal applicability from a methodological perspective. Existing use case environments and architectures have already been documented in D1.3 and should also serve as a basis for further improvements and their evaluation. Corresponding evaluation criteria and their relationship to use case-specific and project-wide KPIs are defined in D5.5 and are not the subject of this document. Instead, this section focuses exclusively on the use case environment enhancements required for successful integration and on coordinating possible synergies and collaboration in this area. Similar to how we talked about horizontal integration between solution providers in the previous section to harmonies and maximiser the impact of solution components, this section also assesses the potential of horizontal integration between use case providers: Depending on existing use case environments, the necessary extensions (e.g., data adapters, data access infrastructure, etc.) could be shared or developed by use case providers.

Example: The example given in the previous example was based on three use case providers and three solution providers. After applying and combining the AIDOaRt integration mediator and process for the given aspects, only the relationship between use case provider 3 and solution provider Z remains. This is also due to the relatively small number of use case and solution providers in the given example (in AIDOaRt we have 10 use case providers and 23 solution providers). In order to continue the discussion on horizontal and vertical integration aspects related to the required extensions of the use case environment, more use cases (4, 5 and 6) and solution providers (Zx, Zy, Zz) are added to the given examples, as shown in Figure 20.

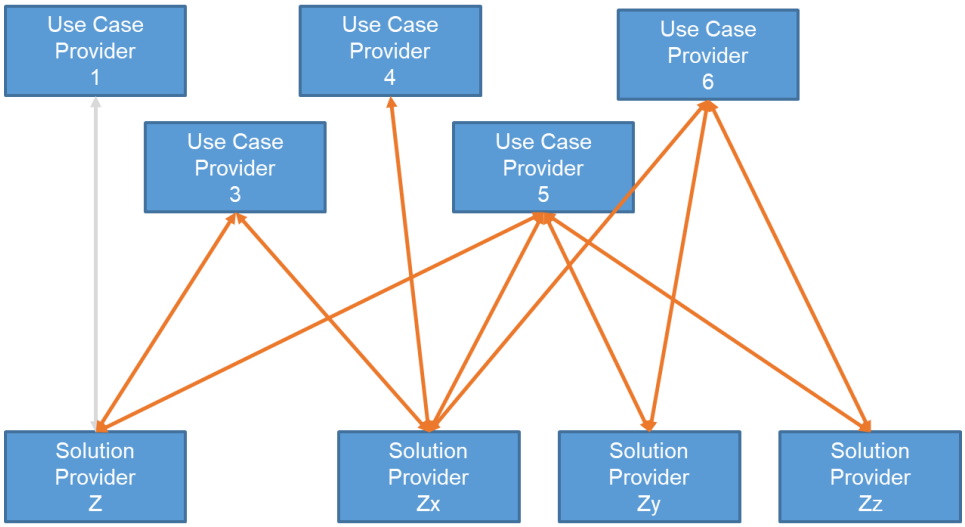

Figure 20 Extended example with more involved use case and solution providers (after the application of the various integration approaches of the previous sections)

These relationships reflect the concrete collaboration between use case providers and solution providers after applying the different integration approaches presented in the previous sections of this chapter (for example, they show that solution provider Zx has quite a large impact on a joint integration of its solution for a number of use case providers). Let us assume that these relationships lead to a set of required use case environment extensions to enable successful integration of a particular solution. These use case specific environment extensions A-G are shown in Figure 21.

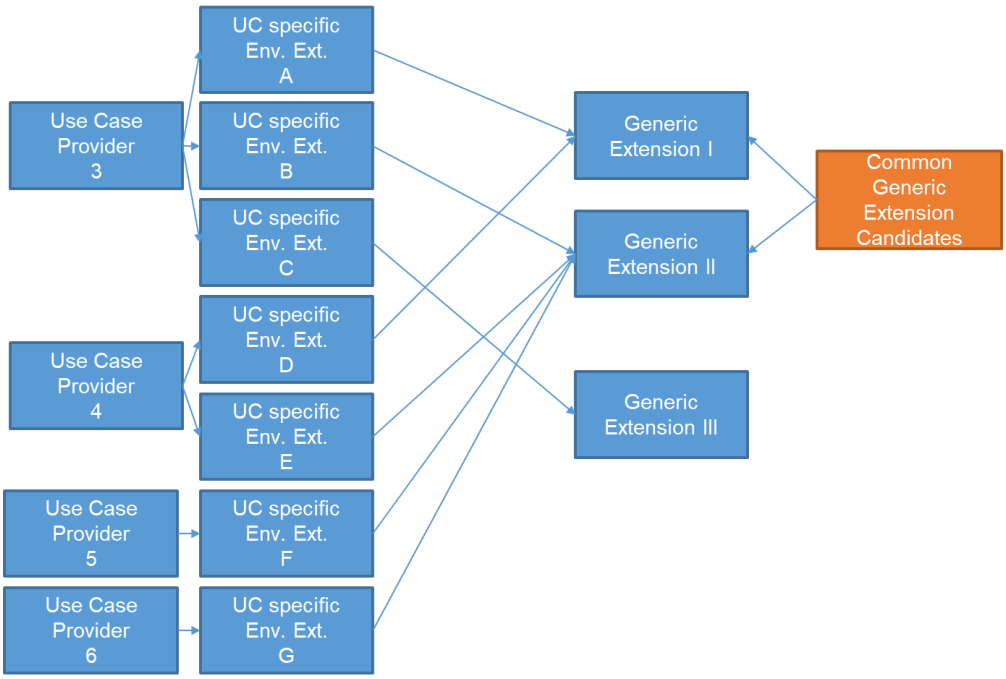

Figure 21 Required use case specific environment extensions to integrate solution components

Similar to the use case requirements, these environment extensions are now analyzed and generalized according to their goal and the required components. In this example, three generic extensions are derived, with two of the three generic extensions being required by more than one use case provider: The fact that generic extension II is only needed by use case provider 3 does not mean that the required vertical integration between use case provider 3 and the respective solution provider is abandoned. It may even mean that in the case of use case provider 3, both solution provider Z and Zx are related to the same extension, which would mean further horizontal integration between the involved solution providers in order to be compatible with the corresponding extension, but this is not the focus of this section.

However, since generic extensions I and II are linked by different use case providers, these extensions are considered as common generic extensions (similar to common generic requirements). As shown in Figure 22, the introduction of these generic requirements does not require any new (potential) vertical relationships between use case and solution providers. However, the existing vertical relationships naturally determine the content of the generic extensions. This content assigned to the generic requirements implies that a harmonized integration approach across use case providers can enable various synergies and effort reductions. Based on the given relationships between extensions and use case providers, potential horizontal integration relationships can be derived as shown in Figure 23. Following the process proposed in Section 2.3, these potential relationships need to be further evaluated for possible benefits and synergies. If synergies are agreed between use case providers, a joint implementation with shared resources for the integration of a dedicated solution component is implied. This joint integration will be appropriately documented in one of the upcoming deliverables D5.2, D5.3 and D5.4 to demonstrate its universal applicability.

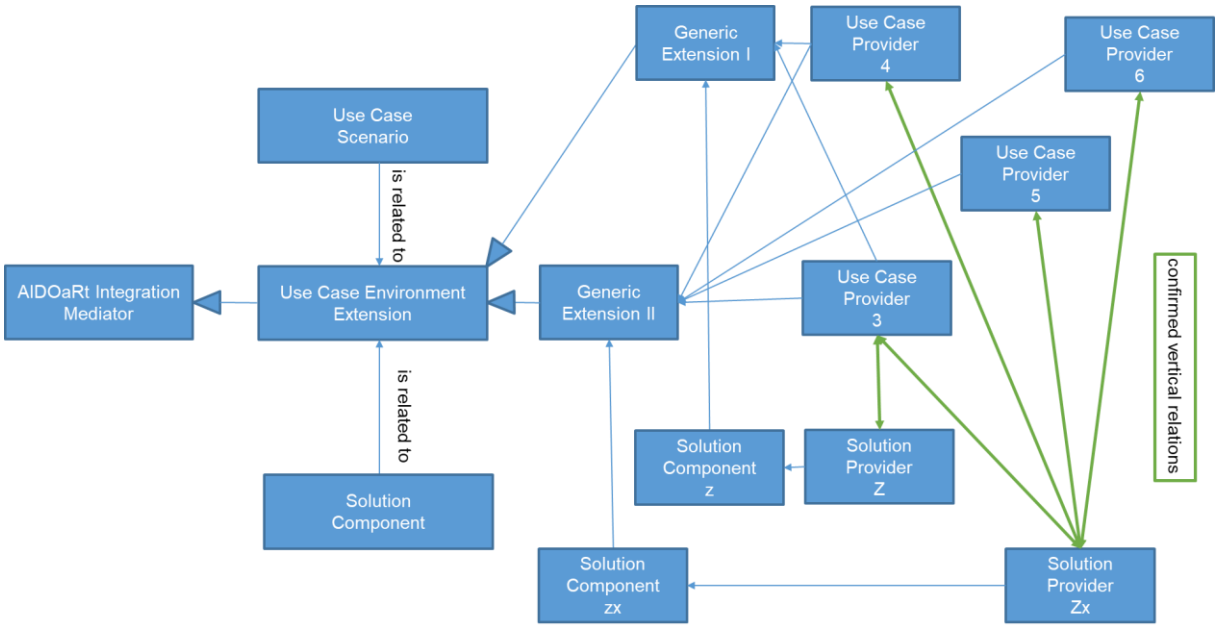

Figure 22 Confirmed vertical relationships for generic extension integration

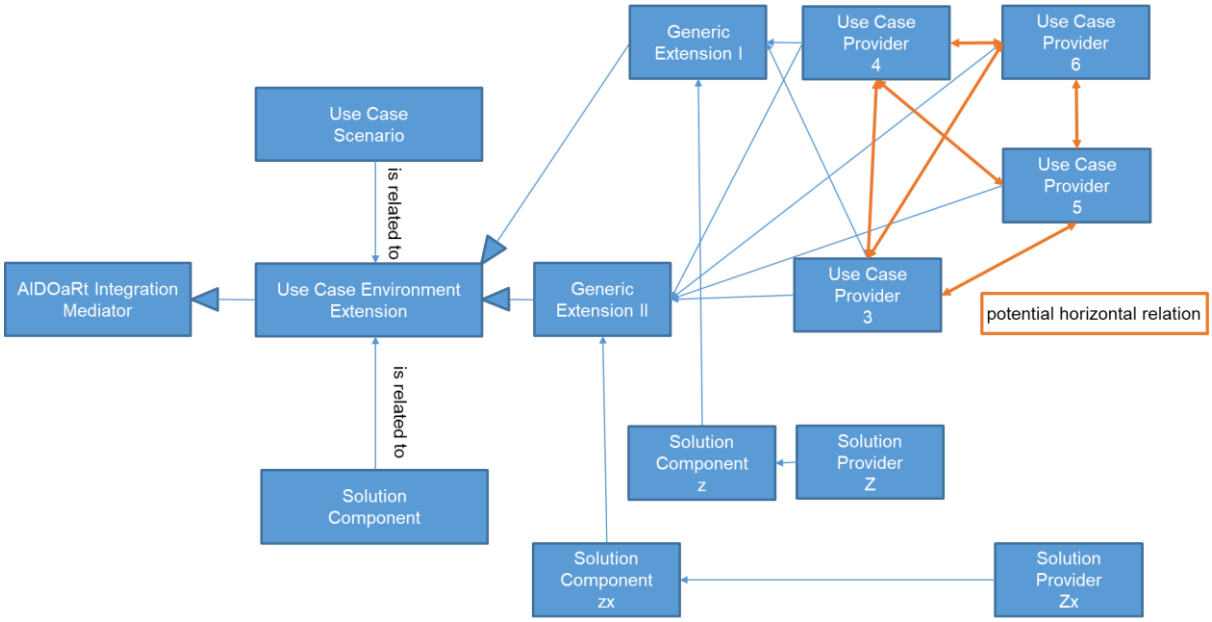

Figure 23 Potential horizontal relations between use case providers

3.6. Integration Aspects on AIDOaRt Architecture Component Interface Level

The integration approaches and activities presented in the previous sections of this chapter are intended to lead to a detailed analysis of which solution and use case components are to be integrated and which aspects are to be considered. Both horizontal and vertical integration are

intended to maximize relevance, reusability and standardization and to minimize the effort required to integrate the respective components by exploiting synergies.

The topic of this section, integration aspects at the component interface level, is intended as a further refinement step that completes the collected knowledge from the activities described in the previous sections and the applied process (in the form of microtasks, assessments and gap analyses). The AIDOaRt architecture assigns a high-level interface element to each architectural component. Figure 24 shows an example of the AIDOaRt architecture component *Data Management* associated with the interfaces IF-FILTERING-HARMONIZATION, IF-DATA-TRANSFORMATION, IF-DATA-FILTERING-AGGREGATION. Further details and complete lists and descriptions for all components and interfaces of the AIDOaRt architecture can be found in the AIDOaRt report D1.2 (and the upcoming report D1.4).

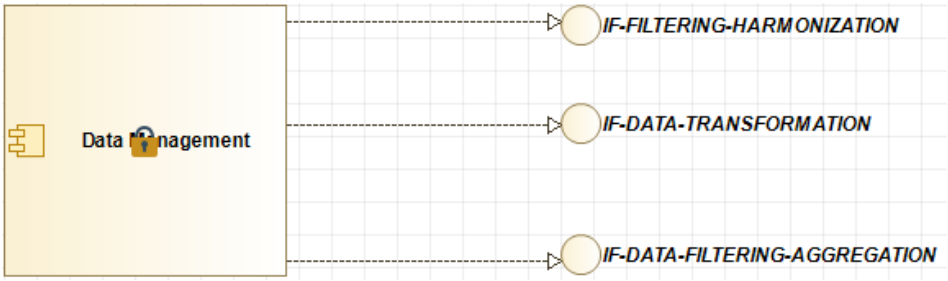

Figure 24 High-level interfaces defined for Engagement & Analysis

On the one hand, these interfaces refine the corresponding AIDOaRt architecture component and, on the other hand, generalize required interfaces for both the corresponding use case and solution components. In this way, these AIDOaRt architecture component interfaces are considered as the sixth manifestation of the AIDOaRt integration mediator pattern. Or in other words: In their role as refinements of the AIDOaRt architecture components, the architecture component interfaces thus refine in particular the integration aspects presented in Section 3.3.

At the time of writing, the concrete AIDOaRt Modelio metamodel elements between which the architecture component interfaces mediate have not yet been finally defined and are the subject of the forthcoming document D1.4. However, possible candidates on the use case provider side include use case (data) requirements, (formalized) use case scenario steps (see D1.3) and the components they reference, or the use case environment extensions defined in the previous section. Possible candidates on the solution side are the already established data interfaces, solution data requirements or solution services or a refined version of them. In this section, we refer to these candidates simply as use case interface aspect and solution component interface aspect, as shown in the left side of Figure 25 (these placeholder terms will be replaced with concrete terms in subsequent papers D5.2-4).

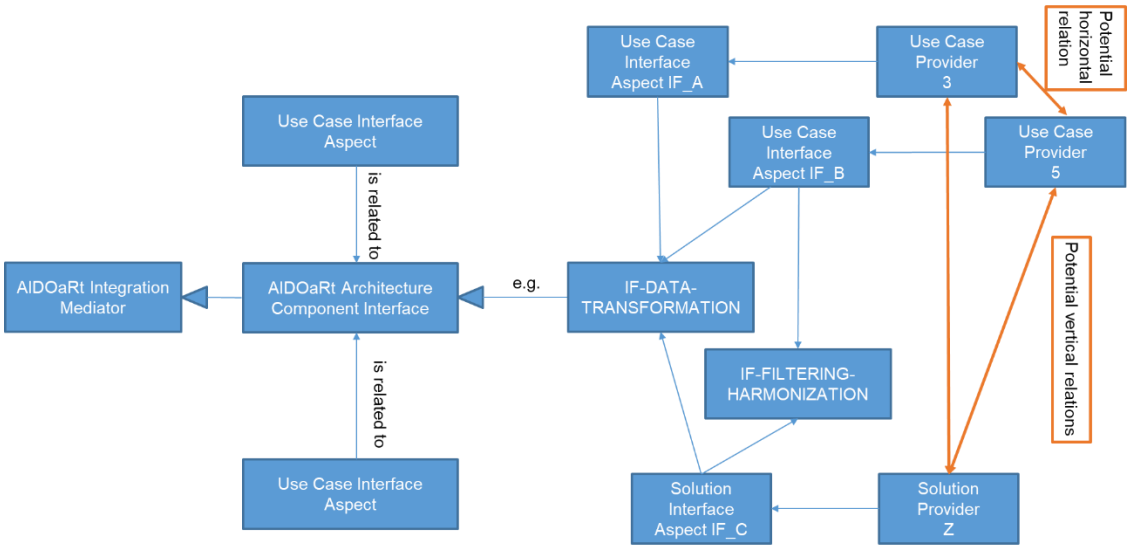

Figure 25 AIDOaRt Architecture Component Interface as integration mediator element

Example: The right-hand side of the figure shows another example that refines and extends the example presented in Section 3.3. There, the AIDOaRt architecture component Data Management was used as an intermediary between the use case requirements of the two use case providers 1 and 3 and the solution provider Z. Since it was found in Section 3.4 that use case provider 1 and solution provider Z are not compatible due to data representation restrictions, the example mentions use case provider 5 (which is also connected to solution provider Z and was introduced in Section 3.5) instead. If it is assumed that both use case providers 3 and 5 have use case requirements that relate to data management. As shown in Figure 24, the data management component is refined by three interfaces, two of which (IF-DATA-TRANSFORMATION and IF-FILTERING-HARMONIZATION) are part of the example shown in Figure 25. Specifically, the use case interface aspect IF_A from use case provider 3 relates only to IF-DATA-TRANSFORMATION, while the use case interface aspect IF_B from use case provider 5 relates to both IF-DATA_TRANSFORMATION and IF-FILTERING-HARMONIZATION. Note that these relationships are established by the process described in Section 2.3 (e.g., using microtasks). Through the same process, another microtask results in corresponding relationships between the given interface components and the solution interface aspect IF_C provided by the solution provider Z. Based on these relationships, the overall potential relationships as previously defined through the use of the data management component as an integration mediator component are confirmed.

However, when it comes to the concrete collaboration and integration between solution provider Z and use case providers 3 and 5 respectively, different integration aspects need to be considered depending on which interface the collaboration is based on. In this sense, the example refines the situation described in Section 3.3: use case provider 3 collaborates with solution provider Z via the IF-DATA-TRANSFORMATION interface, while use case providers 3 and 5 collaborate with solution provider Z via IF-FILTERING-HARMONIZATION. For the latter, this means that a possible horizontal integration makes sense for use case providers 3 and 5 to harmonize interfaces and exploit synergies when applying the corresponding solution component from solution provider Z.

Of course, further constellations may occur once the given process is applied. If more than one solution provider is connected to the same AIDOaRt architecture component interface but to the same use case providers, horizontal integration activities between solution providers also come into focus. All these conclusions can be derived from the different integration aspects of the previous sections and lead to corresponding queries and analyses of the AIDOaRt Modelio model. This analysis may also reveal contradictions or further gaps, which then need to be further evaluated through a corresponding gap analysis as proposed in Section 2.3.

3.7. Potential further Integration Aspects on Core Tool Set and AI-Augmented Tool Set

As proposed in Section 2.3.2, the *integration status* related to the AIDOaRt architecture component level integration aspects (see Section 3.3) should include links to the concrete toolsets of the AIDOaRt architecture: *Data Engineering Tool Set* (WP2 related), *Core Tool Set* (WP3 related) and *AI-Augmented Tool Set* (WP4 related). Specific to the Data Engineering Tool Set, Section 3.4 defines another instance of the AIDOaRt integration mediation pattern related to data engineering (more specifically on data representation) that refines the expected outcome of Section 3.3. This is because current activities within WP2 already show that appropriate mediated discussions between Use Case Provider and Solution Provider are required to agree on data requirements, representations and standards. From the current perspective of T5.1, it is not yet clear whether WP3 and WP4 will require further application of the AIDOaRt integration mediator process (leading to further AIDOaRt integration mediator model instances) or whether the reference to concrete toolset components, as intended by integration status monitoring, will be sufficient. This could be especially the case, since integration aspects on the interface level of the involved solution components is already considered by a corresponding AIDOaRt integration mediator pattern instance presented in Section 3.6.

As it is expected that the AIDOaRt integration approach will be refined during the project lifetime based on lessons learnt from the application of the approach, relevant updates of further integration aspects may be part of one of the upcoming follow-up deliverables D5.2-4.

3.8. Summary

In this section, six concrete integration aspects have been outlined that are linked to corresponding instances of the AIDOaRt integration mediator pattern. Their application to abstract examples is intended to demonstrate how potential and confirmed vertical and horizontal integration relationships between use case owners and use case providers can be established through the application of the AIDOaRt integration mediator pattern process as defined in Section 2.3.3. The last section furthermore holds out the prospect that the various integration aspects will be further developed in the course of the project depending on the insights gained. In the next chapter, various aspects presented in this section are mapped onto the iterative integration roadmap as planned by the AIDOaRt project. Due to the iterative nature of the roadmap (represented by a series of successive deliverables), the aforementioned extensions could thus also be introduced iteratively.

4. Integration Roadmaps

This chapter outlines in detail the envisaged integration roadmap for AIDOaRt. The integration is basically divided into three iterations. During each iteration, the integration roadmap is evaluated and improved for the next iteration. Figure 26 shows an overview of these three iterations (with submission dates M18, M24 and M36). The first integration iteration is started by M12 with the integration roadmap defined in this chapter. At the end of each iteration, a corresponding document (e.g., D5.2-4) will present the results of the integration activities of that particular iteration.

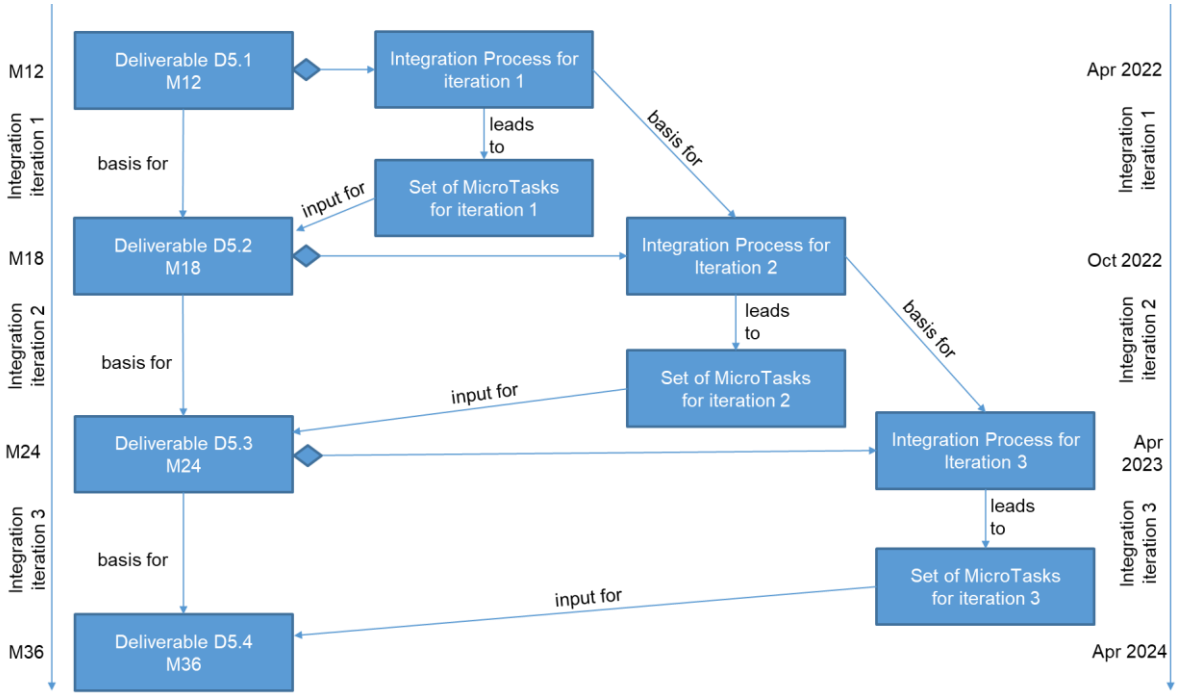

Figure 26 Overall Integration Roadmap

Starting with D5.1, the follow-up documents D5.2 and D5.3 will provide an updated description of an integration process for each iteration (in this document, this process is described in Section 2.3), with each integration process description based on the description of the previous iteration and including adaptations and improvements from the experience gained during each iteration phase. From each integration process, a dedicated set of corresponding microtasks is derived (for use case providers, for solution providers and, if necessary, for dedicated additional working groups). These microtasks are in many cases represented by corresponding Modelio modelling tasks and provide concrete integration descriptions that will lead to generated content for deliverables D5.2-4 per integration iteration phase.

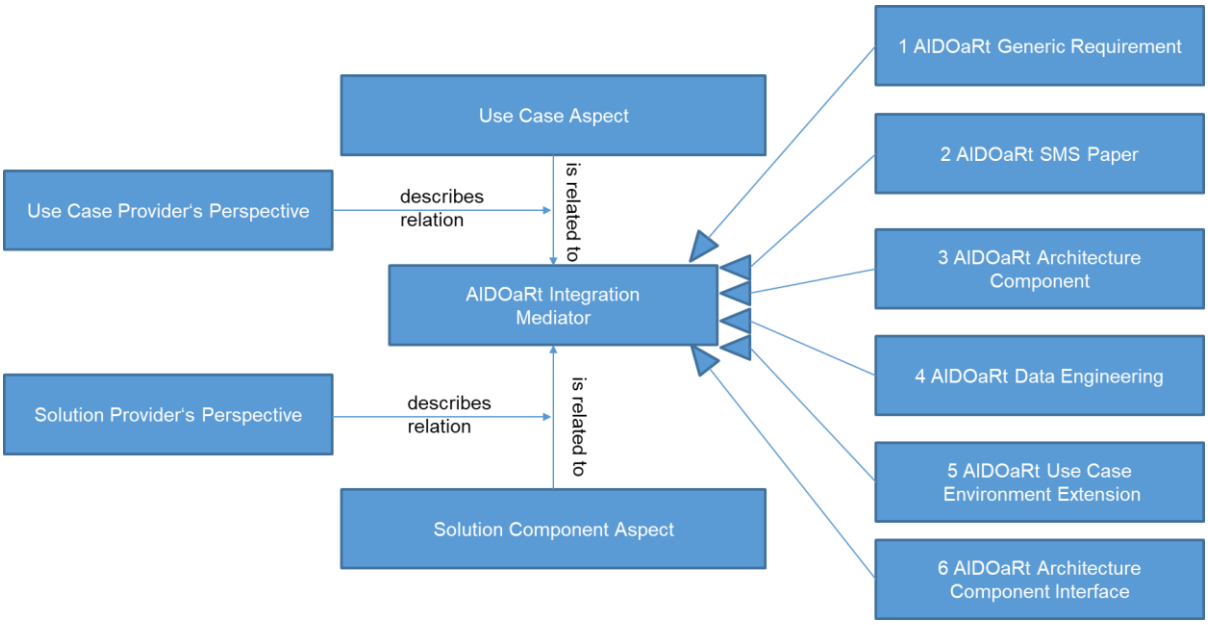

Figure 27 Overview about AIDOaRt integration mediators

Figure 27 illustrates once again the different AIDOaRt integration facilitators described in detail in Chapter 3. Each confirmed relationship is associated with a corresponding description of the integration strategies used, including both the use case and solution providers' perspectives.

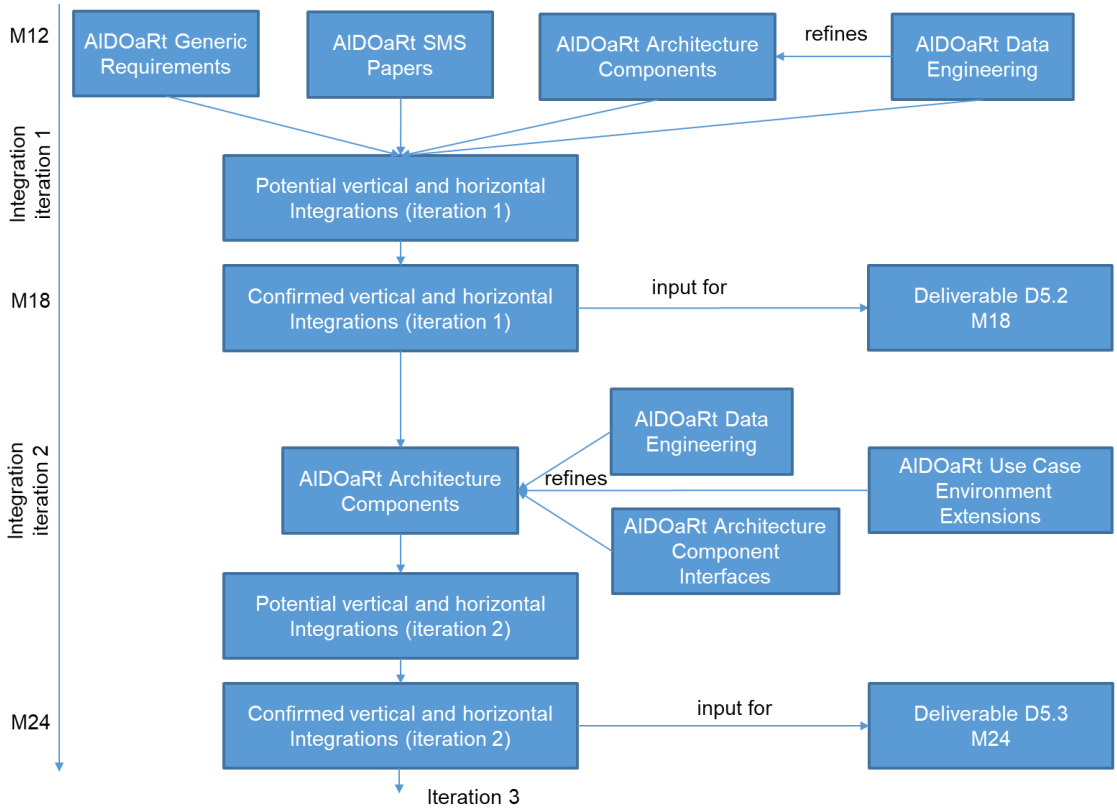

Figure 28 Roadmap for the application of the AIDOaRt integration mediators

With respect to these integration mediators, Figure 28 outlines the planned roadmap for their application in the first two of three integration iterations (the third iteration is planned by D5.2, which will also include an updated roadmap for the second iteration).

Iteration 1: For the upcoming first iteration, the application of the Generic Requirements, the SMS papers and the Architecture Component Mediators is planned. As described in Section 3.4, integration aspects for data engineering are considered as a refinement for the architecture component mediator and the results generated by the application of the integration process. Since data aspects play an essential role in AIDOaRt (represented by a separate WP2 work page), this refinement will already take place in the first integration iteration. The application of the integration process will then lead to the definition and description of a number of potential vertical and horizontal integrations between use case and solution components. After appropriate evaluations, the confirmed integrations will provide appropriate inputs (by applying model to text generators) to the deliverable D5.2.

Iteration 2: In the second iteration, the architecture component mediator will be further refined by an updated version of the data engineering mediator and by the use case environment extensions and component interface mediators. Similar to data engineering, the latter two integration mediators are intended to refine the architecture component mediator by providing more details about the targeted and applied integration strategies. It may turn out that, depending on the results of the first iteration phase, an updated version of the generic requirements and SMS paper mediators may also be required, but from today's perspective these are not planned. Similar, further iteration aspects as currently planned may be applied to refine the architecture component mediator as outlined by Section 3.7. In any case, the application of the integration process will result in an updated set of potential and vertical integration relationships, which will be confirmed or discarded in a further step, eventually leading to outcome D5.3, which is the basis for the third and final iteration leading to D5.4.

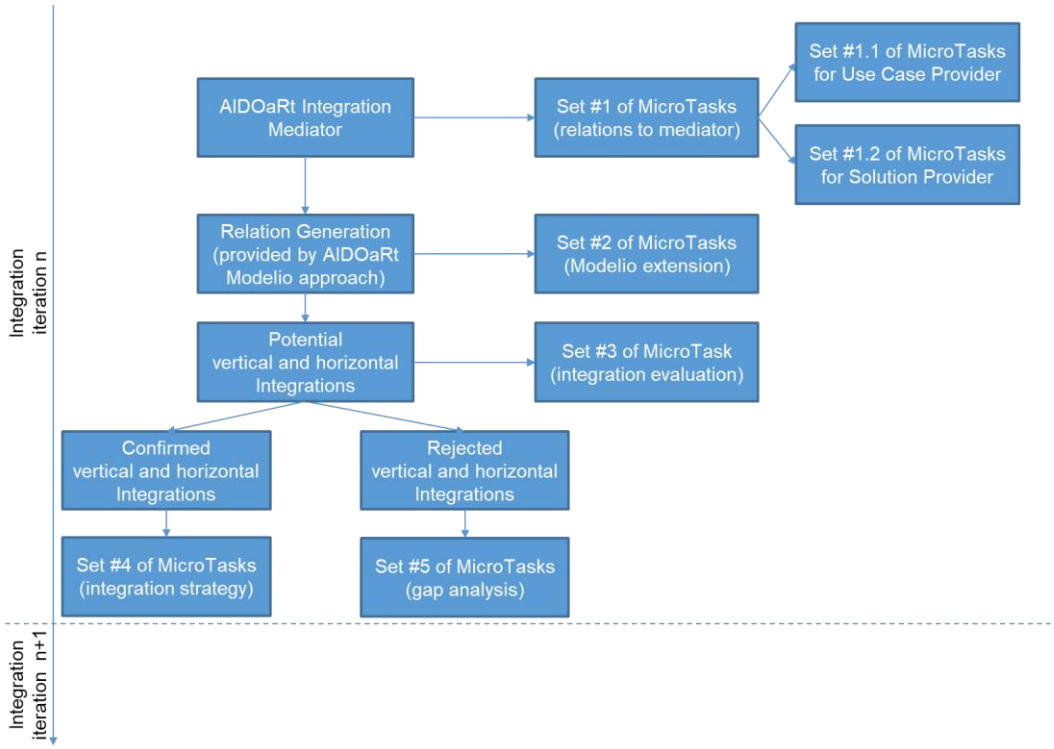

Figure 29 Definition of dedicated sets of microtasks

For planning each iteration period of integration, a specific set of microtasks is derived by applying the AIDOaRt Mediator integration process. These microtasks will be carefully planned for each iteration process and submitted to all AIDOaRt partners upfront to plan resources and efforts for the different integration tasks in advance. Fulfilment and tracking of the micro-task activities will also allow progress and potential problems in component integration to be tracked at a detailed level.

Figure 29 shows five sets of these microtasks defined per AIDOaRt integration mediator. Set #1 is about mapping the respective integration aspect to the respective integration mediator. It is divided into Set #1.1, which focuses on use case aspects, while Set #1.2 deals with solution provider aspects. Set #2 of the microtasks then addresses the Modelio extensions required to derive potential vertical and horizontal relationships based on the input from microtask Set #1. Microtask Set #3 then deals with the evaluation of the potential vertical and horizontal integrations and is intended as an intensive phase of collaboration between the AIDOaRt partners. The outcome of these microtasks is then confirmed and rejected vertical and horizontal integrations. The confirmed ones then lead to microtask #4 on the definition of a common *integration strategy* between the partners involved in a confirmed relationship, while for the rejected ones a set of microtasks (set #5) is defined to perform a gap analysis. This gap analysis is carried out by specific working groups and is an important input for the next iteration phase of the integration. An enhanced description of the performed gap analysis and the agreed integration strategies will be presented in the follow-up deliverables D5.2, D5.3 and D5.4.

Summary: The application of the AIDOaRt integration mediator pattern and process (Section 2.3) and its various pattern instances in the form of integration aspects (Chapter 3) to the planned AIDOaRt

schedule for T5.1 resulted in an iterative integration roadmap. Two out of three iterations are already defined by this chapter, while the second will be refined after the first one is completed. The refinements include in particular the refinement of the integration aspects as such. However, for the first iteration, a concrete set of microtasks can already be derived and communicated to the AIDOaRt partners. This set will help to plan the planned activities for T5.1, which should lead to the next T5.1-related deliverable D5.2. This planning strategy also serves to mitigate emerging integration risks and helps to keep the overall AIDOaRt project timeline.

5. Conclusions

This deliverable describes the generic AIDOaRt integration approach and strategies in detail by dividing the strategies into six integration aspects (which may be even enhanced based on the insights gained during their application). Based on these aspects, a generic AIDOaRt integration mediation pattern was derived and a generic AIDOaRt integration mediation process was defined. The pattern and process were then applied to each integration aspect represented by a dedicated integration mediator pattern instance. This was discussed in terms of several AIDOaRt partner-neutral or abstract examples explaining the meaning of possible vertical and horizontal integration approaches. During this discussion, several interdependencies of the integration aspects were identified, e.g., that some integration aspects can be interpreted as a refinement of another aspect or how the aspect relates to the overall AIDOaRt architecture.

Based on this work, a specific integration roadmap for the various AIDOaRt integration mediator pattern instances have been derived, which has been aligned with the originally planned timeline for related follow-up deliverables. This roadmap provides the definition of a corresponding set of integration-related microtasks in advance. These microtasks support, on the one hand, the detailed planning of critical integration activities between partners and, on the other hand, enable the tracking and monitoring of integration activities at project level. This monitoring could also manifest itself in an integration progress dashboard and is also an important input for the activities in T5.3, which assesses the implementation of use cases for which successful integration is considered a necessary prerequisite.

Bibliography

[1] "D5.1: MegaM@RT Integration Approach," 30 11 2018. [Online]. Available: https://megamart2-ecsel.eu/wp-content/uploads/2018/12/D5.1-MegaM@Rt2-Integration-Approach.pdf.

[2] R. I. Beekun and W. H. Glick, "Organization Structure from a Loose Coupling Perspective: A Multidimensional Approach," *Decision Sciences,* vol. 32, no. 2, pp. 227-250, 2007.

[3] B. Gold-Bernstein and W. A. Ruh, Enterprise Integration: The Essential Guide to Integration Solutions, Addison Wesley, 2005.