LeArning and robuSt decision SupporT systems for agile mANufacTuring environments

Project Acronym:

ASSISTANT

Grant agreement no: 101000165

Deliverable no. and title	**D2.2 - Intermediate architecture document**	
Work package	WP 2	Ethic and Human Centric Toolbox
Task	T 2.1	Human Centric Architectural Design
Subtasks involved	Task 2.2	
Lead contractor	Institut Mines-Telecom (IMT) Alexandre Dolgui, alexandre.dolgui@imt-atlantique.fr	
Deliverable responsible	Europa Universität Viadrina Jan-Hendrik Passoth, passoth@europa-uni.de	
Version number	V2.0	
Date	30/06/2022	
Status	Deliverable	
Dissemination level	Public (PU)	

Authors

Participant no.	Part. short name	Author name	Chapter(s)
	JB	Johan Buchholz	All
	BL	Benedict Lang	All
	JHP	Jan-Hendrik Passoth	All
	VX	Evangelos Xanthakis	3,4
	EV	Eduardo Vyhmeister	4

Document History

Version	Date	Author name	Reason
v0.1	04.12.2020	Félicien Barhebwa-Mushamuka	Initial Template
V0.2	01.06.2021	Benedict Lang	Initial draft of structure and first contents
V0.9	06.08.2021	Benedict Lang	First version of the document to be circulated for feedback
V1.0	**30.08.2021**	**Benedict Lang**	**Deliverable D2.1; integrated feedback; editing for better readability**
V1.1	20.12.2021	Johan Buchholz	Initial state for D2.2, editing for better readability
V1.2	04.05.2022	Johan Buchholz Benedict Lang	Adjust structure of the document and include methodology. Literature review is updated.
V1.3	20.05.2022	Johan Buchholz Benedict Lang Vangelis Xanthakis	Update of technical architecture, update chapter 2
V1.4	30.05.2022	Johan Buchholz Benedict Lang	Update of content in all chapters
V1.5	31.05.2022	Johan Buchholz Benedict Lang Vangelis Xanthakis	First version of D2.2 for internal review, to be circulated for feedback
V1.6	27.06.2022	Johan Buchholz Benedict Lang Vangelis Xanthakis	Integration of internal reviewer feedback (from FLM, BiTi)
V2.0	**30.06.2022**	**Johan Buchholz Benedict Lang Jan-Hendrik Passoth**	**Deliverable D2.2; integrated feedback; editing for better readability**

Publishable Executive Summary

This deliverable represents the current state of the human centric architecture for the ASSISTANT project in month 20. It combines approaches from responsible research and innovation with the high-level technical architecture that was provided as an input for this document. This deliverable is the second in a series of three. This intermediate architecture document will be updated in the final stage of the ASSISTANT project.

Within work package 2, there are different approaches that try to ensure the responsible or ethical development of artificial intelligence within the digital twins. This document mostly focuses on the ex-ante approach which raises issues that need to be considered during the process and which are supposed to shape the design process. Ex-post architectural considerations are included in other deliverables within work package 2.

In order to develop a human centric architecture, we combine theoretical approaches with the technical architecture and make suggestions how to integrate the ART-principles into the further design and development process of ASSISTANT. As this is the second deliverable in a series of three, we will exemplify the steps of the process, demonstrate preliminary results and outline how we proceed in the future to deepen our analysis and to contribute to the responsible development of AI-components in ASSISTANT. In order to guide the reader, reading notes are provided in the first chapter.

In the second chapter, the challenge to practically translate abstract values and concepts into concrete development practices is identified. Facing many guidelines and frameworks for responsible technology development, the additional question arises, which one to choose. Additionally, the question of how the specific context of implementation can be considered is identified as a challenge. In order to address these challenges, a literature review of approaches that aim at ensuring that technologies are shaped and used in a responsible way is provided. Therefore, responsible research and innovation, ethics by design, human-centric design are discussed. Finally, the promising ART principles (aiming at accountability, responsibility and transparency) are introduced, as well as the idea to translate abstract values into norms and finally into functionalities of a system.

The third chapter provides a methodology how the approaches discussed in the literature review can be used in concrete development and implementation projects. Requirements engineering, stakeholder identification as well as an analysis of their values are important steps, before the value interpretations are aggregated in a structured procedure. A central result of this methodology is a workshop concept in order to map values, norms, and functionalities. Additionally, questions for a qualitative evaluation of the results of the methodological approach are suggested.

The fourth chapter introduces the preliminary results concerning the ASSISTANT human centric architecture. We give an overview on the different components, the stakeholders, as well as the interventions to apply the developed methodology in the project. The human centric architecture describes general principles, such as accountability, responsibility, and transparency and the related norms within ASSISTANT, before discussing approaches that ensure these principles: Data governance, interoperability, privacy exemplify how the values associated can be ensured through a concrete approach. Additionally, component-related specifications are discussed, highlighting the specificities in the context of process planning and real time actuation. At this point of the living architecture document, most identified values have been translated into norms. In additional workshops, the norms will be linked to concrete functionalities, providing a translation between abstract values and concrete functionalities of the ASSISTANT architecture (and vice versa). The chapter therefore documents the current state of the ASSISTANT human centric architecture.

The last chapter describes the following steps and highlights the learnings in the process of developing both, the methodology, and the result of its application: the human centric architecture. Further steps include the detailed analysis of components, as well as attempts to visualize the translation process of values into norms and functionalities. These steps are considered as important in order to highlight how concrete functionalities of ASSISTANT are linked to high-level values, both if societal values change or additional components are developed (or removed).

Table of contents

List of figures

List of tables

1. About this document

1.1 Content and aim of this document

This document provides an overview of the steps taken within the ASSISTANT project to ensure a responsible and trustworthy architecture for ASSISTANT system and components. The endeavour is based on abstract frameworks in the area of “ethics-by-design” and suggests how they can be applied within the concrete case of AI in industrial manufacturing. Not only does it raise questions or aspects that might become ethical issues it also provides suggestions on how to approach these different challenges.

This is a living document insofar as its development is an ongoing, iterative and explorative process (see also section 1.2). At this point in time, our approach is based on a literature review, the requirements (see deliverables D3.1, D4.1, D5.1, D6.1, and D7.1), on the initial technical architecture of the digital twins to be used within the ASSISTANT project (as developed within T2.2), the methodology that was developed in order to design a human centric architecture (D2.1), and some first empirical insights, e.g. reflections of the stakeholders in workshops.

This document is neither a checklist nor a to-do list. We as the authors are convinced that to design ‘ethical AI systems’, it is not enough to use off-the-shelf methods and ticking off boxes. Instead, it is necessary to deliberate, to discuss and to co-create among stakeholders. Therefore, this document invites stakeholders to the table to foster these debates.

First, the document contains a **review of the literature** and positions our work. We introduce different approaches towards responsible design of artificial intelligence that range from ethics to responsible research and innovation (RRI) and introduce the ART-principles that are the base for the methodological approach for the human centric architecture. ART stands for accountability, responsibility and transparency and provides a lens to look at the interaction of the system and the human that is using the system (chapter 2).

After providing the theoretical background, this document contains a chapter introducing a **methodology** developed to design a human centric conceptual architecture. This includes the description of necessary steps, such as requirements engineering, the identification of relevant stakeholders and approaches to identify, aggregate and map values in a structured procedure. Additionally, a workshop concept was developed that concretizes the theoretical reflections from the previous chapter and allows to actively discuss the values, norms, and functionalities that will be integrated (chapter 3).

The following chapter describes the **ASSISTANT human centric architecture** and uses the theoretical background, as well as the methodology from the previous chapters to describe the stakeholders within ASSISTANT, as well as the preliminary results concerning the architecture. General principles concerning the ASSISTANT human centric architecture are introduced and discussed, including accountability, transparency, and responsibility (ART-principles). Reflections from the development of the technical architecture resulted in considerations concerning data governance, and interoperability. An additional section describes context-specific considerations concerning the components of ASSISTANT (chapter 4).

With this document, we aim at 1) providing a blueprint for other organizations and projects that want to implement their technical applications in an ethically responsible way. By being transparent about how we approach the task of creating an ethical-by-design artificial intelligence application, we invite feedback, critique and debates about further improvement. 2) We document the current state of the ASSISTANT human centric architecture. We attempt to make decisions within the consortium explicit and allow informed readers to challenge, adjust and modify them according to their values, norms, and functionalities.

As other deliverables within the ASSISTANT project cover the full list of requirements and therefore also an in-depth description of each component that together constitute the overall system, we decided to not replicate this information but instead want to invite you as the reader to refer to Deliverables D3.1, D4.1, D5.1, D6.1 and D7.1, as well as the assessment of all components with the trustworthy guidelines in task T2.2.

With this document, we aim to contribute to the knowledge of how abstract frameworks and principles can be translated into concrete adaptations of technical developments. In doing so, we make decisions and considerations within the development phase of the ASSISTANT solutions transparent.

1.2 About the living document – overview of changes compared to deliverable D2.1

This deliverable (D2.2) describes the state of the living document of the human centric architecture. It is the second deliverable in a row of three and therefore documents preliminary results. This explains why certain sections of the human centric architecture document are more detailed than others. The document will be updated in the further process of the project and the final deliverable (D2.4) will include the final version of the human centric architecture.

In order to guide the reader, we will give a brief overview on the changes and continuities compared to the previous deliverable (D2.1). First of all, additional feedback was integrated and the structure of the living human centric architecture document was adjusted. It now consists of an introduction into the topic and existing literature (chapter 2), an overview on the methodology in order to derive a human centric architecture (chapter 3), and the documentation of the (preliminary) results (chapter 4)

- Chapter 2 includes an extended and updated version of the literature review. In section "2.2.3 Accountability, Responsibility, and Transparency as central principles", the steps to translate abstract values into norms are added and elaborated. In this chapter we identify the need to make the approach to translate these values into norms, and functionalities concrete.
- Chapter 3 is restructured and includes the newly added methodology in order to develop a human centric architecture document. While section 3.1 on requirements engineering remains as in D2.1, the identification of relevant stakeholders (3.20), their values (3.3), the aggregation of value interpretations (3.4), as well as following approach in order to map values, norms, and functionalities was developed within this deliverable (D2.2). Additionally, we introduce a workshop format (3.5), as well as the remaining/further steps in order to develop a human centric architecture (3.6). Finally, we suggest criteria for an evaluation of the approach (3.7). Therefore, the chapter contributes a toolbox of methodologies that can be used in AI development projects in order to link values, norms, and functionalities in a structured way.
- Chapter 4 is restructured and now reflects general considerations, as well as reflections concerning the components of the ASSISTANT system. Minor changes are integrated into the first part of section 4.1 on digital twins for manufacturing, while the identification of relevant stakeholder is added in this deliverable. The interventions within ASSISTANT are newly documented (4.2), before the basic human centric architectural principles are introduced (4.3): While content on the values

accountability, responsibility, transparency, and data governance is newly added (4.3.1 – 4.3.4, subsections on interoperability, security and encryption, role based access control, central authentication system, data encryption, privacy (**Error! Reference source not found.** – 4.3.11) include only minor updates and adjustments compared to D2.1. This chapter documents the considerations and steps in ASSISTANT during the application of the developed methodology from the previous chapter and link values, norms, and functionalities both, on a general level, as well as related to the components.

- Chapter 5 is updated and restructured but does not include new sections at this point in time. However, it gives an updated overview on the next steps, the learnings of the process, as well as limits and reflections of the approach chosen.

Additionally, we highlight the aim of each chapter and the changes compared to the earlier deliverable D2.1 in the beginning of each chapter in a summary box:

In these boxes you will find information on the content that you can expect in the following chapter, as well as a very short summary of the changes compared to D2.1.

1.3 Definition of Success criteria

At the beginning of a process, it is necessary to be clear about what the process is supposed to achieve. We try to be transparent in that sense through the definition of the following success criteria:

Our expectation towards a successful approach within the ASSISTANT project is defined by the following criteria:

1. Our approach has to enable explicit deliberations about values. It is our assumption that the development of technical systems is always influenced and shaped by the values and perceptions of the engineers and the developing teams. Values and how they are embedded in technology are a core topic of Science and Technology Studies.
2. Our approach has to offer the potential to improve processes in which AI systems are produced. It is our assumption that in order to improve technologies, it is necessary to improve the processes in which they are developed.
3. Our approach must enable reflections about potential biases and different perspectives of the people involved. We assume that in order to create a human centric architecture, multiple stakeholders need to have a seat at the table.
4. Our approach has to be connected to broader discussions in the field.
5. Our approach has to be concrete and offer tangible instructions for the actors working in the ASSISTANT project.
6. Our approach has to consider specificities of the manufacturing sector in connection to artificial intelligence.
7. Our approach has to function as a blueprint for others that want to adopt our work for their projects.

2. Towards a human centric architecture

In this chapter, we discuss why it is important to reflect the values and norms that are integrated in the development process of a software system. As technology is deeply embedded in society and organizations, it is therefore necessary to develop a human centric architecture that makes these values and norms visible. We explore different approaches towards the integration of ethical, societal or organizational values into the design process and suggest following a design for values approach, aiming at the integration of Accountability, Responsibility, and Transparency (ART-principles) in a structured procedure.

Compared to D2.1 this section has been updated with recent literature and has been restructured. Additionally, the discussion of accountability, transparency, and responsibility now elaborates necessary steps to translate values into norms, and functionalities (see section 2.2.3).

Developing a human centric architecture within the ASSISTANT project is part of a broader effort to ensure that the technologies produced within the project are designed in a responsible way. In the first section of this chapter, we will refer to debates on the need to reflect questions of responsibility in AI development. In the second section, we are going to discuss existing approaches towards the development of a human centric architecture.

2.1 Human centric architecture: term, challenges, and importance

The term *human centric architecture* requires some explanation: Therefore – and as literature does not provide a unified definition of the term, we will approach it from different directions. Architecture in relation to technical systems serves as a metaphor for a blueprint of the structural design and central components. Such an architecture can be considered as a design principle, describing relevant aspects of the system to be developed. With regard to software architectures, this has been formalized further as "protocols for communication, synchronization, and data access; assignment of functionality to design elements; physical distribution; composition of design elements; scaling and performance; and selection among design alternatives" (Garlan and Shaw 1993) are described. A human centric architecture is not limited to software and can take into account other structural elements shaping the system (e.g. massive noise on the shopfloor that limit further design choices for interaction to a software system, as it excludes voice recognition), furthermore it stresses the values and norms that are inevitably transferred into the development of a system. Designing a *human centric*[1] system means to include the perspectives, values, and norms of the people who might be affected by it. Therefore, a *human centric architecture* systematically links the values and norms of the people affected to the structures and properties of a system. As a consequence, the human centric architecture needs to be developed in close collaboration and alignment with the technical architecture, including iterative circles of development. ASSISTANT involves the development of different digital twins using artificial intelligence (AI), a main focus of the ASSISTANT human centric architecture refers to the properties of the software systems. A human centric architectural design in the context of ASSISTANT therefore makes sure that "AI components (…) consider ethical aspects, human centric, trustworthy AI, and other responsible AI approaches during all the software production phases"(ASSISTANT proposal 2020: 15). It is

[1] Human centric design must not be mixed up with human-centred design, which is defined in the ISO-norm 9421-210 and refers to design processes that include the users in order to develop products based on their requirements. It is therefore closely related to the value of the product for the user, i.e. the product will be used and will contribute to the company's profit.

closely linked to the technical architecture and informs and affects the design decisions concerning the technical architecture developed in T2.2. As such as process includes iterations, the human centric architecture will be developed in a living document that accompanies this process.

To develop a human centric architecture in relation with AI, we will review existing approaches and the challenges related to them. Current debates exemplify the need for AI systems to reflect issues of responsibility, such as questions of fairness, transparency and accountability in order to prevent harm to society and people (Dignum 2019a; Floridi and Cowls 2019; Jobin, Ienca, and Vayena 2019; Winfield et al. 2021). One approach to reflect questions of responsibility is the use of a broad variety of guidelines that were developed by companies, governments and NGOs.[2] However, the question remains how such abstract frameworks and terms can be translated into practice (Aler Tubella et al. 2019; Hagendorff 2020b). In other words: Knowing about these frameworks does not mean that the central principles are reflected in the development of the software architecture or in lines of code of a certain system.

Under the umbrella term “ethical AI” the boundaries as well as the expected or desired behaviour of AI systems are discussed. While we acknowledge these general discourses, the concrete development of AI systems usually happens in projects. However, ethical AI approaches aim at providing universal guidelines and do not address the constraints of concrete projects, e.g. project plans, requirements, or interests of the project partners that have to be aligned. Recent literature on the responsible development of AI points out that the integration of guidelines requires “adequate implementation strategies” (Jobin et al. 2019), while others highlight the variety of interpretations and values in different socio-cultural contexts that require reflection of the designers in order to make informed decisions about trust into a given system (Dignum 2019a). To overcome the pros resulting from generalized and abstract guidelines and ethical AI, we suggest focusing on the process of technology development. Arguments suggesting moving away from checklists and general frameworks – that claim to provide on holistic sets of questions and criteria – reflect the abstract and generalized form of frameworks and are therefore convincing. However, when it comes to the development of AI, very concrete decisions and tangible interventions must be made in order to ensure that the AI acts in a responsible way. This applies even more to complex settings such as development projects, in which different actors have to align their expectations and interests.

In manufacturing contexts, organizations have developed procedures and processes to get things done, while making sure that basic standards are considered and preventing harm from the people affected. However, the change of technologies in these manufacturing processes are highly contested. As work contexts are subject to regulation, additionally they are a matter of responsibility concerning the people working in them. The idea of creating a digital twin as a digital replication of the production context, including an optimization according to specific criteria and then a translation of this optimized model back into physical manufacturing, opens questions: E.g., in the context of ASSISTANT this raises the questions, which assumptions, selections, and decisions are taken in the steps of translation between physical manufacturing and digital twin (and vice versa)? To address these questions, it is necessary to reflect on potential biases, blind spots, and the values that should be incorporated into the technical, digital tools.

To address the challenges of 1) selection of a framework/guideline among a variety of them, 2) putting frameworks into practice, and 3) reflecting the contexts of implementation, we will review existing approaches from literature that address questions of human centric,

[2] E.g. the NGO Algorithmwatch provides an overview of more than 170 frameworks (AlgorithmWatch 2019); earlier Jobin et. Al. (2019) identified 84 documents.

trustworthy, and responsible AI. This step is necessary in order to situate the approach chosen and developed within ASSISTANT.

2.2 Theoretical Framework

This chapter gives an overview of existing approaches towards human centric architectural design in the context of AI. We provide this overview to be transparent about the assumptions and schools of thought that underlie this document. We will contextualize specific approaches among existing frameworks on human centric and ethical AI to then develop our own approach.

As we connect different streams of literature in the following section, Figure 1 provides an overview of what will be covered. On the one hand we have a strand of literature discussing Ethics, on the other hand we have the term Responsible Research and Innovation that was initially coined by the Horizon 2020 framework. From top to bottom, we see a concretization of the two streams, that is leading from normative to applied ethics and that is framing the ART-principles as a part of human centric design. Even though the graphic suggests a solely structured order, the different approaches are not that easily distinguished but somehow overlap, and develop next to as well as influence each other.

Figure 1 overview of theoretical framework

In recent years, debates on ethical, trustworthy, or responsible AI have been fostered by political actors, researchers, and civil society. A growing number of guidelines to ensure 'ethical' AI demonstrates the relevance of the issue.[3] However, these rather abstract approaches are hardly translated into concrete results in the development process, as they are not "put into practice" (Hagendorff 2020b). The development of a human centric architecture document for the concrete context of manufacturing within ASSISTANT is embedded in these broader discussions and debates on ethics, responsible research and innovation (RRI) and other frameworks. We consider responsibility to be the key aspect that links different discourses and approaches towards a development process, which reflects the assumptions and needs of different stakeholders[4].

Debates on ethics are usually connected to different **theoretical fields of ethics**, such as normative or applied ethics. While the former focuses on the identification of moral standards

[3] Hagendorff (2020a) discusses a selection of these guidelines, a non-exhaustive overview is provided by Algorithm Watch (2021).
[4] An analysis of the stakeholders in the context of ASSISTANT will be provided in section 4.1.

that can be used to differentiate between wrong and right behaviour, the latter approaches specify controversial issues. Following one of these approaches within the development process of the ASSISTANT architecture would include the definition of standards and boundaries that should not be ignored or crossed. Therefore, such an approach might include abstract recommendations that have to be considered when it comes to the specific context of the ASSISTANT AI systems. While the strength of such approaches lies in being explicit about certain boundaries and values, it is hard to reflect and understand the local situation through *ex ante* definitions of norms and standards. Therefore, we also extend our perspective towards responsible research and innovation and other approaches. These are to be adapted to each situation in which they are implemented.

2.2.1 Responsible Research and Innovation and ethics by design

One approach to consider and anticipate the consequences of a certain technology in society next to normative ex-ante ethics is the field of **responsible research and innovation (RRI)**. Shaped by contributions from Science and Technology Studies (STS), this approach has been established and prominent in recent projects funded by the EU. It includes continuous reflection on different questions during the research process, involves actors from the research context as well as civil society, such as third sector organizations, to align the development process and the outcomes with expectations of society. One aim of this approach is to make the assumptions that are embedded in development processes visible and transparent. This transparency supports the development of teams to ensure that they are on track regarding their responsibilities. It also allows for users and stakeholders of technologies to criticize them.

Within the area of responsible research and innovation, an approach has emerged that is called **ethics by design**. This approach includes a set of best practices that focus on including ethical or responsible deliberations already in the design process. These best practices include organizational aspects – for example, the establishment of an ethics board as well as suggestions for the actual process (Leidner and Plachouras 2017). This aims at an integration of “ethical decision routines in AI systems” (Hagendorff 2020b), where we understand ethics as the values of stakeholders that are integrated explicitly into the decision algorithms. Reflecting these values in the design process would therefore ensure “ethics by design”. This document is supposed to contribute to an ethics-by-design approach in that it not only presents the different technical components but also points towards the integration of values as part of the design process.

Such a process can start with **frameworks**. One example of such a framework is the assessment list for trustworthy AI provided by the “High-Level Expert Group on Artificial Intelligence” (2019). The authors point out that AI should be lawful, ethical, and robust to be trustworthy30/06/2022 13:49:00. The framework comes with a range of questions, grouped by issues such as human agency and oversight, technical robustness and safety, privacy and data governance, transparency, diversity, non-discrimination and fairness, societal and environmental well-being, as well as accountability. As part of an assessment list, these topics provide important foci in the reflection of AI applications.

The strengths of such frameworks lie in reflecting abstract dimensions that should be considered during AI development and use. They are supposed to ensure that socio-technical systems do not contradict specific societal needs and issues, e.g., that AI does not exclude people based on dimensions such as race, gender, education, and others.

However, different frameworks by governments, businesses and from the third sector exist (Algorithm Watch 2021) and highlight slightly different aspects. And even when the decision towards the use of a framework is taken, the translation of the mentioned issues into the context of development is a highly complex task that requires additional decisions. Therefore, abstract frameworks can only provide starting points for a discussion on trustworthy and human centric AI, while the actual implementation process requires the consideration of the very local contexts, the associated assumptions and decisions that are taken for granted.

2.2.2 Responsible human centric AI

Principles of RRI are reflected in, **human centric design, which provides** sets of methodologies and strategies to make these abstract frameworks tangible in concrete development projects: it is informed by approaches in design thinking and uses methods from the social sciences such as interviews, group discussions, ethnographic approaches, but also many more.[5] Human centric design can thus shape a development process according to the needs of humans, as the formulation and reflection of these needs is a core element of this process. However, the use of these methods alone does not guarantee that ethical aspects are considered, as they may not be relevant to the actors involved in the process. Therefore, we combine elements of the mentioned approaches to ensure that the human centric architecture is ethical-by-design.

More concretely, human centric design – as we understand it – is an approach to narrow down the scope of values that need to be addressed within design processes. It places the focus on the interaction between humans and technologies. This is specifically helpful in contexts of the development of artificial intelligence systems because a lot of the moral dilemmas that emerge from them are entangled with the non-humanness of the AI. This interaction between humans and technologies is radically changed through artificial intelligence applications, which is a good reason to focus on these interactions.

Creating a **responsible, human centric AI** is a complex task that needs to reflect these different streams of literature and thought. We suggest including elements of these different discourses through the notion of responsible AI and the use of a specific framework, which is open for taking up core-concepts from ethical debates, responsible research and innovation, and abstract frameworks. Also, there should be openness to the requirements and needs within the specific context of AI systems in industry. Concrete work in that sense must be eclectic and must put together different components adopted for the specific situation.

2.2.3 Accountability, Responsibility, and Transparency as central principles

One concrete approach towards responsible AI based on Human Centric Design encompasses the ART principles (Dignum 2019a, 2019b).[6] Dignum's suggestion spells out the abstract idea of the responsible creation of AI through a set of values: Accountability, Responsibility, and Transparency. This is helpful, as the values allow to raise questions regarding the process of development and the system to be developed. Therefore, we consider it an adequate starting point for our endeavour to make assumptions explicit during the process. The ART-principles allow us to address issues of accountability, responsibility, and transparency within ASSISTANT.

The underlying assumption is that processes of software and technology development are full of decisions that the "designers, developers and other stakeholders" have to make, "many of them of an ethical nature" (Dignum 2019a:48). Dignum therefore also highlights the difference between an ethical process of development and the AI system being capable of making its decisions in an ethical way. Whether an AI system can decide ethically is dependent on the

[5] several handbooks, such as IDEO (2015); LUMA Institute (2012) provide an overview on these methods.
[6] As part of the project HumanE AI – which was funded within the Horizon 2020 programme – Dignum authored a report that introduces a methodology to develop what she calls Responsible AI systems. This includes mainly two thoughts. At first, the ART-principles are introduced as a requirement towards AI systems. Secondly, they introduce the methodology of Design for Values that aims at making values and the process of embedding them into software design projects visible and transparent.

values that are embedded within it and therefore a result of the development process. Additionally, it is dependent on negotiation processes within societies that define what is ethical and what is not. As we share this understanding in relation to values in design processes and the ideas are already present in the text above, we will not further elaborate on them but will continue with the introduction of the ART principles.

The ART principles for Responsible AI can be summarized as follows: They include three aspects that need to be reflected when pursuing responsible systems. Accountability, Responsibility and Transparency have a close connection to each other and seem to be to some extent overlapping. However, they provide a different focus and complement each other.

- "***Accountability** refers to the requirement for the system to be able to explain and justify its decisions to users and other relevant actors*" (Dignum 2019a). This means that the system needs to be able to be held accountable in relation to humans that interact with it and are affected by it. Therefore, decisions need to be explainable *after* they have been taken.[7]
- "***Responsibility** refers to the role of people themselves in their relation to AI systems*" (Dignum 2019a). Responsibility is different from accountability in that it focuses on the people involved and is not related to the content of the decision: It links to questions of liability, on the one hand, but also to who is capable of behaving morally. Questions of responsibility could be: Who delegates which decisions to the system and how are decisions supervised? The responsibility dimension encourages reflections about the role of different persons within the process of decision-making and system development.
- "***Transparency** indicates the capability to describe, inspect and reproduce the mechanisms through which AI systems make decisions [...]*" (Dignum 2019a). It is therefore a precondition to determine responsibilities and to hold the responsible people accountable. Transparency increases trust, as people do not only have to trust but can ground their faith on a sophisticated understanding of how algorithms work. Making the algorithms transparent allows stakeholders to criticize what is going on. Transparency is different from accountability in that it is not necessarily linked to one specific situation that is evaluated ex-post but includes a more general need for openness.

As stated above, the three principles are closely connected to each other, interdependent and intertwined. But still, they address different specific foci. What Dignum (2019a) generally suggests is to concretize abstract values into more concrete norms that can then again be translated into concrete functionalities. For example, one can define "openness" as a value which is to be translated into norms that could mean "access for stakeholders" or "being adoptive to stakeholders' feedback". In the implementation phase, where actual code is produced, this results in concrete forms, buttons or dashboards that enable control or insight.

The translation of values into norms follows an interpretation of these values (Dignum 2019a), e.g., in a specific context. In chapter 3 we suggest how the abstract values can be interpreted in the context of manufacturing and digital twins. This step defines what an abstract value means and therefore expresses an interpretation or a norm to be followed in the development and use of the system. Formal definitions that describe a relation in the form "X counts-as Y" (2019a) also inform the translation of norms into concrete functionalities. This architecture

[7] Throughout the course of the document (section 4.3 and appendix 7.2) you can find concrete examples of how these values translate into a concrete task within the project. To give you a glimpse already, one of the examples how **accountability** was interpreted is that *ASSISTANT solutions allow the documentation of every action performed*. Also, for Responsibility and Accountability, you can concrete examples in the later sections.

document represents the interpretations and concretisations within the ASSISTANT human centric architecture. As the translation of values into norms can be done in different ways, we suggest the following methodology, which bases on Dignum (2019a:64) and specifies the proposed steps:

1. The identification of relevant stakeholders: This step involves the identification of relevant stakeholders in the context of the technical system. In development phases, it can be necessary to include potential users.
2. Documentation of values of the stakeholders identified in step 1: In this step, it is the main challenge and goal to elicit values and other requirements of the stakeholders.
3. Aggregation of value interpretations in a structured procedure: For this step, fitting methodologies must be provided to map, integrate and select the values relevant for the system that is developed.
4. Mapping links between values and norms: In this step, values and norms are linked to each other. Therefore, values are interpreted in specific ways in relation to the context in which they are used. Additionally, norms can be concretized and linked to functionalities of the system.

Following these steps allows us to make the assumptions, values, and norms that were reflected in the development process transparent. Linking values, norms and functionalities explicitly allows to react, when

a) values change over time (e.g., values are disregarded, or new values should be reflected and
b) functionalities of the system are replaced in innovation cycles.

In both cases it is possible to identify the linked norms, values, and functionalities. Then
In concrete situations, there are also different perspectives that must be aligned. This means that these ethical concerns (e.g., values, norms and functionalities) and domain requirements (e.g., functional and non-functional) influence the actual process of developing the AI. They have an impact on the motives and roles, the goals that are to be achieved and finally the actual plans & actions. Both domain requirements and ethical considerations can be structured hierarchically to match more high-level or more specific aspects of the design process.

Important to note is that – like other system development methodologies – the process is designed to be, on the one hand, iterative and, on the other hand, goes beyond the first going live of the application. It requires management throughout the entire lifecycle of the AI application. Obviously, the creation of responsible AI – following the suggestions of Dignum (2019a) – does not replace legal compliance and the reflection of the regulatory context in which the system is to be embedded.

3. Developing a human centric architecture for ASSISTANT

In this chapter, we translate the insights from the previous chapter (literature review) into concrete approaches and methods for developing a human centric architecture. These methods are closely linked to the approaches mentioned in the literature review and can therefore be considered as a concretization of the existing literature, aiming at the development of a human centric architecture.

In comparison to D2.1, we concretised the process towards the final human centric architecture in two ways. First, we concretised parts of the process (see following paragraphs). Secondly, we sketched out additional parts of the process. While section 3.1 remained almost unchanged, our approach for identifying stakeholders and values was added. Additionally, we elaborated the process of identifying (3.3), aggregating (3.4) and mapping values, norms, and functionalities through a workshop format (3.5). We discuss the further steps (3.6) and suggest criteria for evaluation of the success of the approach (3.7).

Within the ASSISTANT project and its work package 2, there are different tasks that are supposed to ensure the responsible and ethical creation of AI. This human centric architecture document is part of an ex-ante approach that points out potential issues, explicates values and discussions of the project in the first place and therefore focuses on the process. The KPIs and evaluation criteria that are defined in another task are supposed to help in examining the success of the project. Additionally, the human centric architecture focuses on the interaction of the complete system with its users and the interaction of the different components with each other.
A visualization of the two approaches is shown in Figure 2. The figure shows differences in the approaches in relation to the levels they address, the frameworks they use, the object of their attendance, and the responsibility of the organization within the ASSISTANT project: We aim at developing a methodology that goes beyond abstract frameworks and addresses concrete values and norms in the development process of ASSISTANT. We focus on the overall system and the interaction of different components.

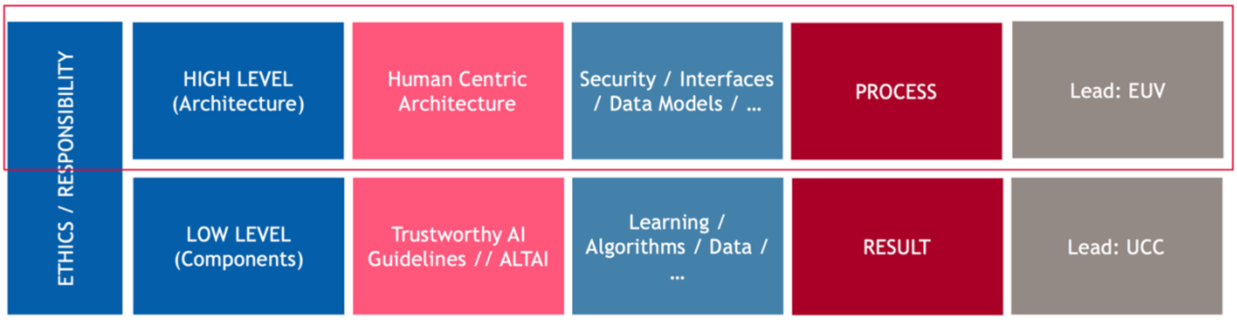

Figure 2 Visualization of the different approaches for responsible design within ASSISTANT

The two approaches not only complement each other in relation to time but also in relation to the objects they address. While the human centric architecture focuses on a higher level and looks at the different components and their interaction, the trustworthy AI guidelines will address what happens inside the components and focus more on the concrete algorithms and machine learning mechanisms. Throughout the further course of the project, it will be necessary to further integrate these two approaches to gain a more in-depth analysis.

These activities in ASSISTANT have two main goals: First, making sure that the development process of ASSISTANT, as well as its result considers ethical aspects, human centric, trustworthy and responsible AI principles, as we. Second, to develop methodologies for such an endeavour. This chapter mainly follows the second goal in the sense that it is depicting our considerations and approaches. It invites others to criticise, adapt and improve our work when translating it to their contexts. Results of our work can be found in chapter 4.

The following figure provides an overview of the timeline of our approach. After developing the theoretical framework for this project, and after requirements were collected in all work packages of the project, we aim at identifying and translating values into norms as guiding principles. Then, to deepen the integration of technical aspects with human centric approaches, we will analyse the different components of the ASSISTANT system and their interaction to derive and implement concrete functionalities from the values and norms identified before. All the steps are feeding into this living architecture document. It is therefore both, the documentation of our approach as well as the documentation of the human centric architecture as the result of this approach (see chapter 4).

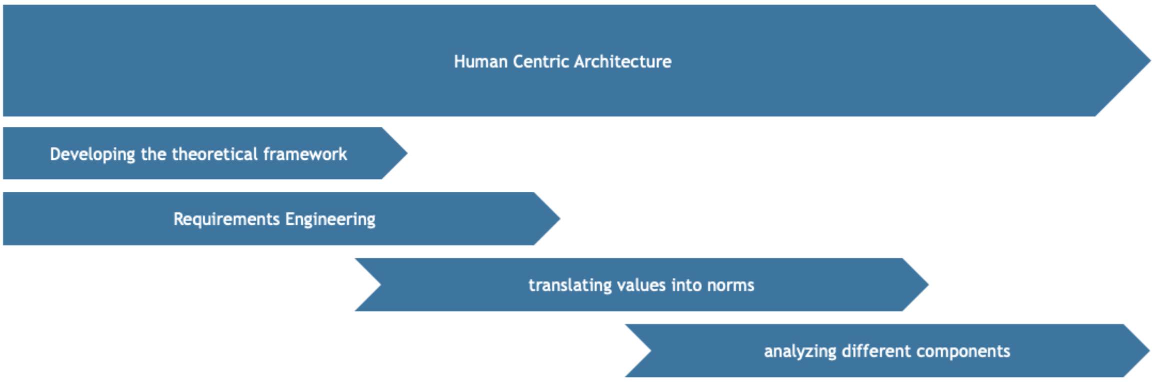

Figure 3 process for developing human centric architecture

As shown above, the whole process is informed and influenced by the ART-principles. They, however, are not final but rather a starting point and a lens through which to develop the concrete methodology for ASSISTANT. We are strongly convinced that there is no such thing as an off-the-shelf method. This means that we will not implement the ART-principles 1:1 but instead use them as a starting point to avoid reinventing the wheel. The ART-principles do not only inform our approach as three values to be reflected in designing responsible systems, but it is rather their embedding in responsible design that is influencing all the different steps of this methodology. The following subchapters will go through the different steps that are depicted in Figure 3.

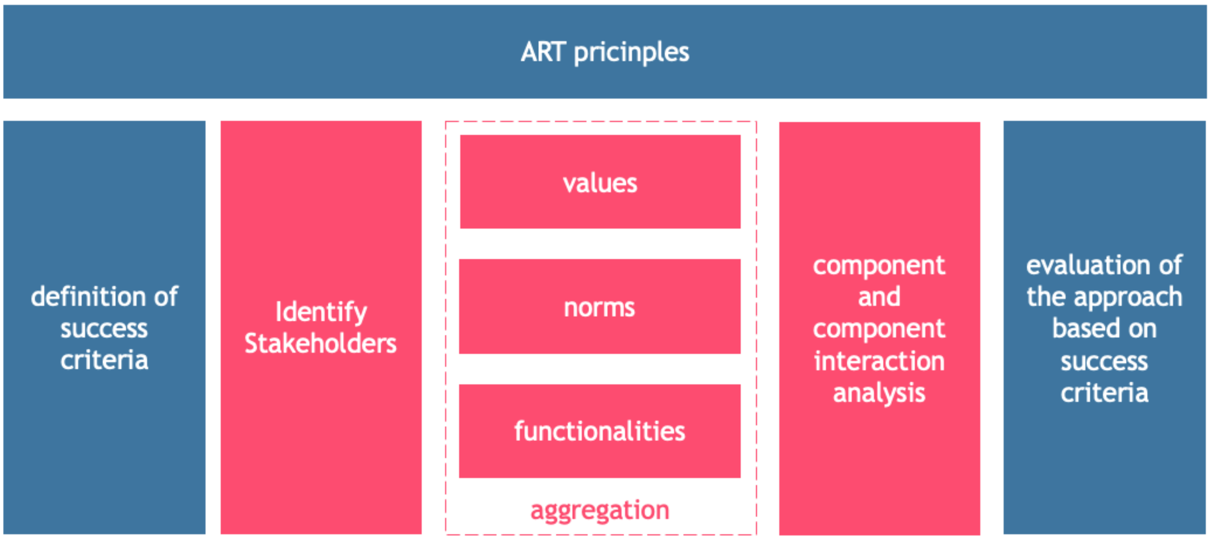

Figure 4 embeddedness of the ART principles in ASSISTANT

We want to explicitly state that throughout the whole process of the development of a human centric architecture, there is no right or wrong. As we have discussed in the literature review, the goal of the human centric architecture is not to include as many values as possible but to make visible which values, norms and decisions are present within the specific project context. Questions of saturation and completeness are therefore not answerable in a general fashion.

Additionally, the procedure that we proposed is therefore in line with the criteria that are suggested by Aldewereld et al. (2015:834): (1) We explicitly defined “global aims” for this architecture document and explained how it is supposed to help us in building a human centric architecture. (2) We plan to facilitate discussions about which stakeholders to involve in the project and we (3) are going to make decisions explicit in the revisions of this document.

Considering the development of a human centric architecture as a process means that the theoretical framework can be extended and adjusted in further revisions of this living document. This means that we will inspect and include more approaches and frameworks from the field of the relevant disciplines, if necessary. However, it is more important to decide on a concrete approach than considering everything that is out there. Our aim within this project is not to provide a holistic overview of existing approaches but rather to contribute to the experiences in bridging the gap between abstract frameworks and concrete developments. This is even more important, as the field is very dynamic and new developments are emerging rapidly.

3.1 Requirements engineering and technical design methodology

The first step towards a human centric architecture is a process of requirements engineering to understand what it is that is to be built. In order to gather the requirements within the project, an integrated requirements engineering methodology is proposed that is described in the following.

Figure 5 depicts this methodology, which was derived from the usual requirements engineering process. The foundations of the methodology are formed by the actual processes of “End User/Customer” which are described and analysed as current practices.

The current practices ("as is" scenario) represent a typical image of today's manufacturing processes and contain several challenges. These challenges, also called points of improvement, constitute the major drivers of the project. Use cases will be derived which constitute an alternative business process. Within the methodology, two main types of requirements are distinguished: the end-user requirements and the IT requirements.

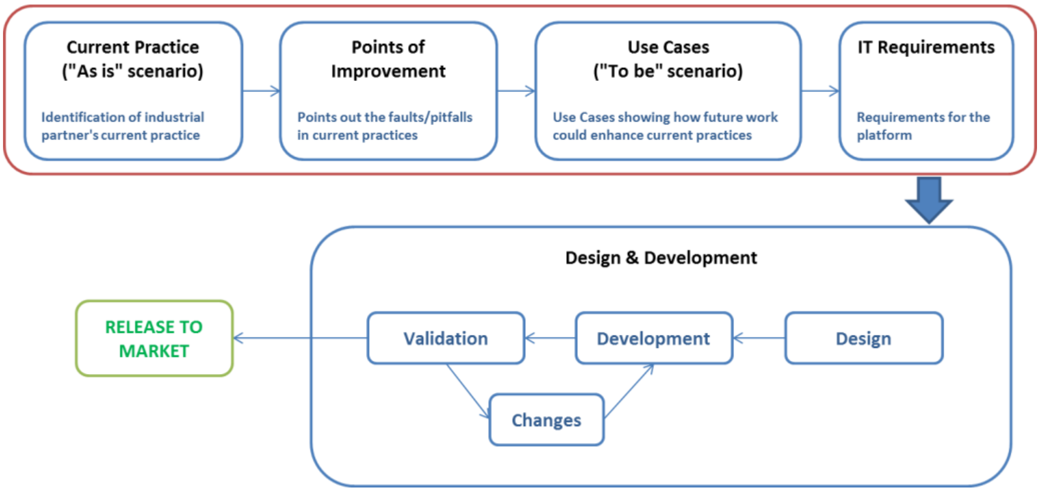

Figure 5 Requirements engineering approach

The end-user requirements are output of process step "points of improvement" and are valid for the entire project. End-user requirements are requirements purely from the end-user point of view. They represent the main business drivers of the project that need to be satisfied by the project outcomes indicating the main pain points in industrial practice at the moment.

The **IT requirements** are formed based on the use cases ("to be" scenario) and target the development by specifying software & hardware functionalities that should be delivered by the solution to support the "to be" scenario.

"As is" scenario: This process should describe the pilot and each pilot's current practices. Text format is appropriate for such a description. This is described in deliverable D7.1.

Points of Improvement: The points of improvement describe the flaws of the "as is" scenario and describe how the current practices could be improved. It is recommended that the output of this process includes a list of "point of improvements"/flaws in bullets.

Along with the improvements, KPIs should also be described that could measure the impact of the improvements achieved by the ASSISTANT project. The cardinality could be many-to-many since grouping of improvements with a single KPI or a single improvement to affect multiple KPIs is allowed. This section is also covered in subsections of chapter two in D7.1.

3.1.1 Component Use Case Definitions

Within the ASSISTANT project, the term *use case* is originally used for the concrete application of the digital twins with the partners from Atlas Copco, Siemens Energy and PSA. The term *component use case* in the following refers to the meaning that is common within requirements engineering, that is, a "behaviourally related sequence of transactions performed by an actor" (Dano, Briand, and Barbier 1997).

For developing the use cases, the requirements engineering process facilitates a formalized description of each use case that includes the following fields:

Name	Each use case should have a unique name that clearly describes the main goal of the use case. Typically, the name is a verb phrase, for example: Withdraw Cash. The reader should be able to determine the goal of a use case simply by observing its name.
Brief Description	The brief description of the use case should reflect its purpose.
Involved components	This is the list of the ASSISTANT components that interact to achieve the use case goal in the flow of events.
Pre-Conditions	A precondition (assumption) is the state of the system and its surroundings that is required before the use case can be started.
Basic Flow of Events	A use case describes the interactions between the ASSISTANT component(s) and the factory system in the form of a dialogue, structured as follows: • The component 1 <<does something>> • The component 2 <<does something in response>> • The system <<does something else>>
Alternative flows	
Subflows	
Key-Scenarios	
Post-Conditions	
Special requirements	
Relevant pilot case(s)	

Table 1 Template for describing use cases

In appendix "7.6 ASSISTANT Component Use Case Definitions", there is a list of such templates filled in for each component of the ASSISTANT component landscape.

3.1.2 Requirements Catalogue

The system requirements are defined based on the use cases and the pilot cases. The system requirements could be defined using the table presented below. The requirements may be connected to one or more use cases or to none of the use cases. The requirements describe both functionality as a response of the system to some user action (that is presented in the use cases chapter) as well as functionality that may not be "visible" to the user but is expected by the system in order to be able to respond to user needs (non-functional requirements). Table **2** shows the template for the catalogue that is used to gather the system requirements.

ID	Overall Description	Specific requirements						
		Performan ce	Logical Database	Hardware Constraints	Standards Compliance	Priority (Low/ Medium/ High)	Module that implements this requirement	Relevant pilot case(s)

Table 2 Template for requirements catalogue

In appendix “7.6 ASSISTANT Component Use Case Definitions”, there is a list of such templates filled in for each component of the ASSISTANT component landscape.

3.2 Identifying relevant stakeholders in the context of ASSISTANT

As discussions about values and norms are central part of our work, it is always necessary to reflect who is having a seat at the table and who can bring in their perspectives. This is even more important within projects like ASSISTANT that may have severe impacts on some stakeholders. Different stakeholders will always be affected differently.

A structured procedure to identify stakeholders starts with a deep understanding of the domain and the thematic area. In order to understand impacts of amendments of processes requires an understanding of who is part of these processes in the first place. While this is not always possible to understand from the outside and with generic knowledge, we propose to speak with experts in the field and to also get into the field oneself, to collect impressions and insights.

Stakeholders can be identified mainly on two levels: First, stakeholders directly involved in the project and second, societal actors that have a relation to the project. Within the project, stakeholders can be identified by being mentioned explicitly or implicitly in various documents: requirements documents, project proposals and use case descriptions. The broader group of stakeholders can be found in publications of governmental as well as non-governmental organisations that are active in the domain. In our case, we look at discourses on manufacturing in general and production optimization and the usage of artificial intelligence and digtal twins in particular. In that area, for example contributions from interest groups like unions can be found as well as statements from inquiry commissions of the parliament or similar structures.

At this stage of the process, it is not yet necessary to study all the recommendations and positions in detail, but instead to search for affected groups of people in these different documents, reports and publications and list them to come back to them later in the project.

We want to explicitly highlight the importance of this process step. If we miss out on some stakeholders, this will have an impact on all the following steps, as they are all dependent on the facilitation of exchange between the different groups. If one stakeholder group is not represented, their values, norms and perspectives will not make it into the system.

As there will always be restrictions in the identification and even more in the gathering of all stakeholders, we argue for transparency to remedy these restrictions. Rather than aiming for everyone to have a seat at the table, we believe it is more important to be explicit and transparent about who is having a seat at the table and why.

3.3 Identifying values based on the stakeholders

After the identification of stakeholders, we start with identifying values that are important to them. These are represented in very different ways, which leads to multiple potential approaches, how to extract and observe them. For example, one can start with frameworks for responsible AI from different stakeholder groups (e.g. supranational actors such as the European Commission, but also Unions, etc.). The goal of this step is to monitor existing values and norms and to also identify potential conflicts within the specific context. It is important to look at both: implicit and explicit values.

Some of the stakeholders – specifically those who are identified on more abstract levels, may remain without direct involvement in the project – publish statements and policy papers that can be skimmed for positions that should be reflected in the project. In this step, it is necessary to make sure that the statements and positions are applicable for the specific domain and context we are in. General recommendations about artificial intelligence might have limited relevance for the specific situation in manufacturing, while health-care applications for example might lead some stakeholders to reflect different values additionally.

For those groups that do not have published documents or statements, there are different ways of finding out about their values. One way is to send around questionnaires that people can answer independently of place or time. In these questionnaires, however, we recommend to not ask question like “what are your values?” but to rather think of questions that are closer to home, as most people involved in engineering projects – at least in our experience – do not necessarily often speak about values explicitly. Instead, questions like “what is important for you? What concerns you?” and alike are easier to answer and still allow for deriving values from the answers. I it is necessary to provide assistance for filling out the forms whenever necessary, so stakeholders can ask questions and understand what is to be achieved with the overall process.

Interpreting these questionnaires requires specific attention to interpretative flexibility regarding the content. Especially as values are rather abstract, it is important to reach out to the participants, whenever in doubt whether some conclusions are valid or not.

Another way of identifying values is using interviews. In these interviews, questions that might be like those in the questionnaire can be asked directly to the stakeholders. In comparison to the questionnaires, the interview allows for clarifying answers and their interpretation through the interviewer. Within those interviews, it is necessary to explain the interviewees explicitly and detailed how their answers will proceed to ensure that they feel comfortable speaking about their values and norms.

If possible, one can also facilitate a workshop format in which different stakeholders exchange their perspectives. The first part of the workshop format we developed for the in-person-meeting within the ASSISTANT consortium explicitly aimed at the gathering of the values of the stakeholders. In these workshops it is necessary to ensure fair moderation and to make sure that no actor dominates the discussion too much. In a first round – the gathering of values – different understandings can also be tolerated, as the aggregation and mapping as well as the translation into norms follows in later steps.

3.4 Aggregation of value interpretations in a structured procedure

The values are then aggregated and discussed in moderated environments, in which the relevant stakeholders can group and (in case of conflicting values) prioritize the values and norms through discussing them and through exchanging perspectives. The results are documented both for further work with them as well as for questions that might arise later about which values were selected and discussed.

In literature as well as in different practical examples, we can observe that the understanding of values is not stable and uniform but always subject to interpretation. While this is not generally problematic, it is important to ensure common understanding of the values that are to be agreed on among project partners. The negotiations about the interpretations best happen in on-site workshop formats, rather than in written or other distributed formats.

During that process of aggregation, connections between different values can be established. Also, conflicts between different values can be solved in a way that suits the different perspectives. For example, transparency is a prerequisite for other values but also transparency might contradict privacy in some cases.

Here, again, fair moderation is necessary to ensure that the different positions can be depicted properly and evenly and that solutions to conflicts are shared and accepted among the stakeholders. If certain stakeholders cannot participate in these aggregation workshops, it is important to integrate their feedback on the agreements and discussions. For the workshop format, it is important to foster interaction and discussion instead of just putting different perspectives next to each other without establishing links and conflicts between them.

The documentation of the results of the workshops – as well as the discussion process – is of high importance. You can find the results and concrete examples in section 4.3 and 4.4, while a more detailed documentation of the workshop results can be found in appendix 7.2.1f. This allows stakeholders who are joining the project at later stages can catch up on the discussions and agreements. It is even more important for those stakeholders too far outside of the project to be part of the discussions. Transparent documentation of the discussions and values counteracts black-boxes and ensures that also others can criticise decisions that were taken in the project democratically. For ASSISTANT, you can find this documentation in chapter 4 and in the appendix.

In the intermediate version of the human centric architecture, we focussed on the implementation and facilitation of a workshop format that identifies and translate certain values. By translating values into norms, we were able to acquire more detailed framing in which the technical components are to be developed.

3.5 Workshop concept for mapping values, norms and functionalities

The overall goal of the design for values approach is to explicitly integrate values in the design of systems. As discussed in the previous chapter, we are aiming at linking values and norms onto concrete functionalities that are implemented as components within ASSISTANT. The links can then be used for further development and application steps both, when values change (e.g., in other environments or over time) or components and norms change. This allows to identify the parts of the ASSISTANT system that are connected to specific values.

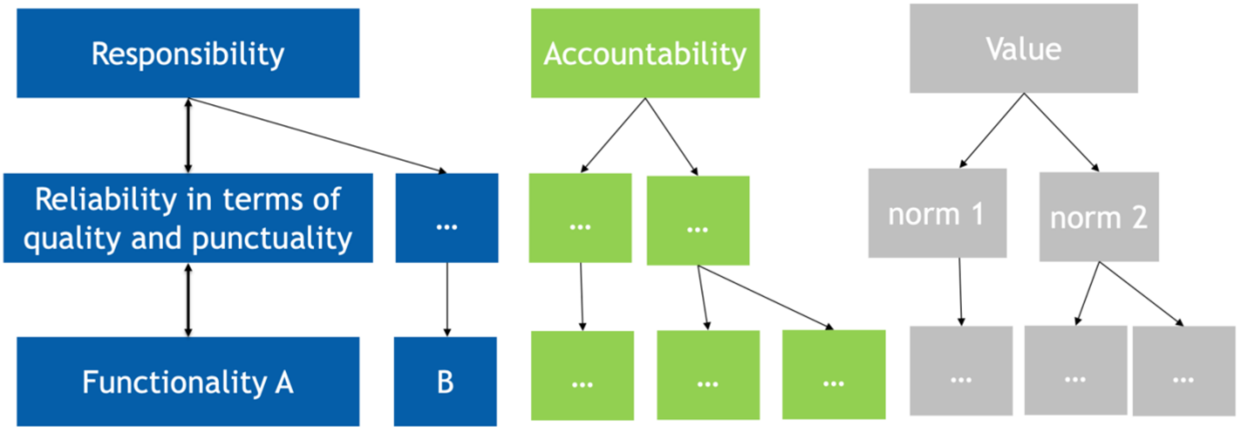

Figure 6 Mapping Values, Norms and Functionalities (following Dignum (2019a); including our own example)

For deriving the concrete functionalities, we again suggest a concrete workshop with the relevant stakeholders to brainstorm and discuss how the implementation can happen (therefore, no example for functionalities is provided in Figure 6). Functionalities do not necessarily have to be technical functions but can also be assistive tools like documentations, organisational measures and alike.

As this is only the intermediate version of the human centric architecture, we have not yet fully developed methods for how to achieve this translation but have gathered first experiences in the workshop described in the following.

To create a human centric architecture, the considerations of the involved stakeholders concerning design decisions have to be made transparent. One way to do so, is to facilitate exchange opportunities, in which relevant stakeholders discuss questions concerning the underlying values, norms, and the concrete functionalities of the AI systems. Open conversations foster an exchange between the involved project partners and stakeholders and can be used in to informed decisions on controversial issues. We therefore decided to create a format that on the one hand allows open deliberation and on the other hand concretizes the approach to translate values into norms and functionalities (see 2.2.3). The workshop content is informed by available material on requirements of the very technical system to be developed.[8] In the following, we will describe the suggested workshop format and our approach to develop it.

The process that we developed is of course not independent from the experiences that we made in the project, including the restrictions concerning on-site meetings due to the pandemic. For the general approach, we suggest using mainly workshops as a format that also happen in-person and on-site. However, to be equipped for remote work, it is helpful to think about a potential virtual version of the workshop format, while developing it. Also, online workshops can simulate in-person experience to some extent, allowing for improving the designed workshop format for the next on-site meeting through testing it online. The workshop concept allows to be applied for both, on-site and online formats.

The workshop concept aims at linking values to concrete functionalities through norms. The workshops are supposed to include relevant internal and external stakeholders and document the discussions on values, norms, and functionalities that will be reflected within ASSISTANT. The workshops consist of three parts:

[8] In ASSISTANT, this refers to D3.1, D4.1, D5.1, D6.1, and D7.1.

1) in an introduction we discuss, why it is important to reflect questions of values and norms in order to build responsible results. Additionally, the approach to interpret values and concretize norms is introduced to the participants.

2) In-depth discussions with the participants allow to document the values, norms, and functionalities that matter to them and go into an exchange on the meaning, possible scenarios and problems related to these interpretations.

3) the results are documented and the results are presented to the other participants. This step is necessary to document why decisions were taken in a specific way.

The results of the workshops provide transparency on which values and norms were discussed, which decisions were taken and finally, how values and norms are linked to functionalities. This approach allows to document the reflections of the involved stakeholders, allowing others to challenge assumptions and to adapt the developed solution, whenever this is necessary. The detailed workshop schedule is part of the methodology developed within ASSISTANT and can be found in the appendix (section 7.2). Intermediate results regarding values, norms and functionalities can be found in chapter 4.

3.6 Further steps until final deliverable: Norms and functionalities

In order to analyse and adjust the developing technical architecture within ASSISTANT, we are going to examine the components and their interplay through the lens of the identified values and norms and the design for values approach. Consequently, we will identify both components and combinations of them that point to potential issues for further discussion. In another step, we will document these issues, group them and discuss selected issues with the developers and, where applicable, with the people affected.

The most important task concerning the living human centric architecture document is the consolidation of the different values and norms and to then monitor whether the concrete functionalities fulfil these norms within ASSISTANT. This is ensured in close collaboration with T2.2, which focusses on the conceptual architecture and can be considered a “driver of the for the developments (WP3-WP6)” (ASSISTANT proposal, p. 16).

The technical architecture here plays a crucial role since it needs to translate these functionalities into guidelines and rules that should be adopted by each ASSISTANT component. The technical architecture aims at grouping these guidelines into component groups (i.e., functional guidelines for user interfaces, functional guidelines for AI components etc.). In a second step it’s role will be the monitoring and the evaluation of their deployment and the recommendation of countermeasures in case of deviations. Apart from the regulation of the functional guidelines the architecture itself needs to be further elaborated in order to make sure that these recommendations are supported seamlessly.

Practically the translation of values into norms and functionalities involves the project partners from the technical workpackages (WP3-6) in collaborative formats, such as workshops, interviews, and other exchange formats (see section 4.2). This approach ensures the entanglement of human centric aspects and the technical architecture within the proposed framework of the project.

3.7 Evaluation

At the end of the project we suggest a qualitative evaluation of the process. While metrics and KPIs in the context of a procedural approach can hardly be defined in quantitative measures, qualitative measures can provide a starting point for measuring the success of the approach. Additionally, a qualitative evaluation can highlight the steps taken in order to develop a human centric architecture (while it is not possible to measure the effect of interventions, as the object for comparison is altered in the procedural approach). We therefore suggest the following reflection questions:

- Are all components of the ASSISTANT system linked to norms?
- Are all norms of the ASSISTANT system linked to values?
- Which actors and stakeholders were involved in the process of mapping values, norms, and functionalities?
- Which design decisions were specified or revised in the course of value and norms discussions in workshops with the relevant stakeholders?
    - Have the requirements been (re-)defined in iterative, reflective stages? At which stages?
    - How have conflicts and conflicting interests been solved and integrated?
- What endeavours have been undertaken to ensure dissemination of knowledge in the area of responsible research and innovation within the project team?
- In which ways has the development process been documented to learn for future projects?
- How has the project team exchanged with other project teams, working in the same thematic areas?

Answering these questions will allow the reader to form an opinion on the development process and whether and how questions of responsibility have been addressed in the process of development. Additionally, these criteria give insights into how exchange on these important aspects was organized.

4. ASSISTANT human centric architecture

This chapter contains the ASSISTANT human centric architecture. It includes different parts: First, we describe the different digital twins and their different tasks (section 4.1). We then present the interventions that we as the team of T2.1 have made (4.2). The following section contains general values and norms of the ASSISTANT human centric architecture (4.3). Finally, we include a preliminary analysis in relation to values, norms and functionalities for specific components (4.4). Doing so, this chapter is the empirical result of the work following the methodology and streams in literature described in the previous chapters.

While in section 4.3 we integrated some parts from the initial human centric architecture D2.1 (in detail: interoperability, security and encryption, privacy and requirements), the component-related human centric architecture (4.4) is mostly new as it is the result of the work conducted between the initial and the intermediate human centric architecture.
So far, two components of ASSISTANT are integrated in the architecture in more detail than others: process planning and real-time actuation. The reasons for choosing these parts of the ASSISTANT solutions (and not production planning, scheduling, or the data fabric) relate to practical concerns: On the one hand, a recent deliverable on the state process planning was submitted and real-time actuation was a major topic during the first on-site visit of the consortium. Addressing both aspects allowed to test the suggested methodology to translate values into norms and finally into functionalities, as well as fitting into the current state of the project. However, the other parts of ASSISTANT will be included in the following versions of the human centric architecture living document (and the final D2.4).

This intermediate version of the human centric architecture sketches out the preliminary structure, as well as some first results concerning the architecture. It thereby includes parts of the Service Oriented Architecture (SOA, developed by T2.2), in which data exchange and data usage in each ASSISTANT module is documented. The approach to interpret values into norms and finally link them to concrete functionalities integrates the SOA at the level of components and modules.

However, before going into detail, the context of ASSISTANT as system for manufacturing using digital twins is discussed, including a mapping of relevant stakeholders. The following section bases on the methodology discussed in chapter 3 and gives a brief overview on the interventions that were conducted within ASSISTANT (until now). In Section 4.3, the basic principles for the human centric architecture will be introduced. We thereby integrate the reflections of the project participants on accountability, responsibility, and transparency (ART-principles). Additionally, these values are not exclusive, but other values were mentioned in the process (e.g. interoperability, security). We will discuss these values and highlight the norms and the approach, how they are put into practice in ASSISTANT. Section 4.4 highlights specific values and norms regarding some of the components to be developed within ASSISTANT. This approach allows us to discuss general values and norms linked to ASSISTANT, as well as the ones linked to specific components.

4.1 Digital twins for manufacturing and relevant stakeholders

ASSISTANT solutions are developed within the context of manufacturing and integrate the perspectives of the developers, use cases and other considerations. In this section, we will give a brief overview on the ASSISTANT digital twins for manufacturing, as well as the relevant stakeholders for the development of the human centric architecture.

The use cases of ASSISTANT are extensively documented and described in the deliverable 7.1. Therefore, we do not want to provide a full description of the manufacturing context that this project is conducted in. However, we want to mention this context here again, to ensure that the considerations are connected to general discussions and developments that are going on in manufacturing, production automation, production optimization and alike.

Figure 7 provides an overview of the different components that constitute the ASSISTANT architecture. There are three different digital twins that contribute in different ways to the optimization of the manufacturing process (process planning, production planning and scheduling, and execution). They focus on different functional parts of the overall production process and the desired optimizations. All digital twins share a common data fabric that provides services in relation to data storage services, data control and data analysis. Concrete exchange of data between the digital twins and the data fabric can be found in Figure 8, which depicts the data flow.

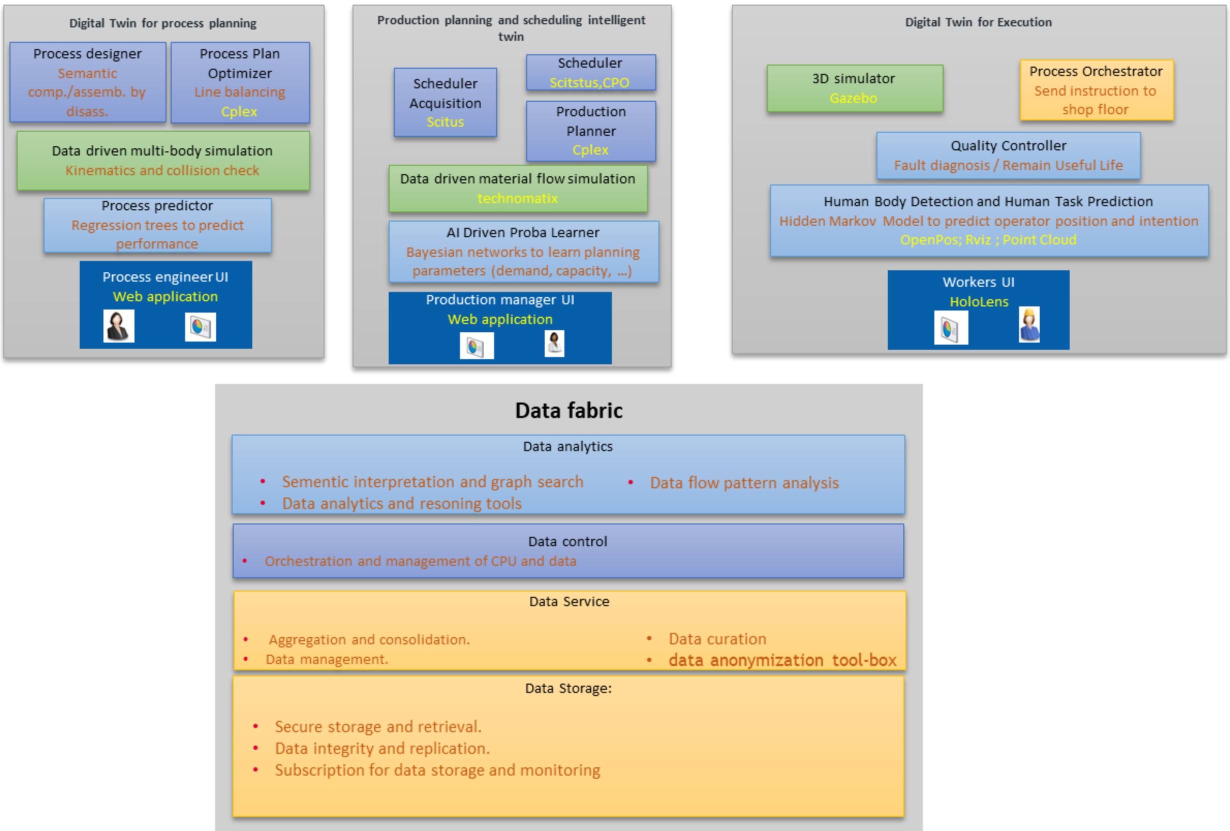

Figure 7 Component Landscape and digital twins

'Digital Twin for process planning' focuses on the design of the production process. The "process designer" and "process plan optimizer" are responsible for generating the different process graphs. Then, for different production plans and technical changes, the "process predictor" enables the forecast of various KPIs regarding cost, time, and quality. These components are joint through the "process engineer". Process engineer provides a user interface that supports users in generating efficient and effective decisions regarding the process design.

‘Production planning and scheduling digital intelligent twin’ aims at providing tools for production planning and scheduling based on AI prediction and simulation. The “Scheduler Acquisition” is responsible for generating a constraint model from a set of tables with schedule-related data. “Scheduler” then optimizes the constraint, model while “Production Planner” computes the production plan (quantity to produce per period, quantity to order, and capacity adjustment with overtime). Again, these functionalities are provided to the production manager via a user interface (“Production Manager UI”).

The responsibility of ‘Digital Twin for Execution’ is to utilize the output of the two aforementioned component groups (namely process plans and production schedule) in order to successfully drive production in the shop-floor. “Process Orchestrator” works as the interface of the digital twin with the process and production planning, while “3D Simulator” monitors the real-time behaviour of the system along with “Human Body Detection and Human Task Prediction” and guides the system to accomplish the task at hand uninterrupted. The “Quality Control” module offers a near real-time system performance in order to quickly identify faults and propose countermeasures. Again, these functionalities are offered to the end users via a graphical user interface (“Workers UI”).

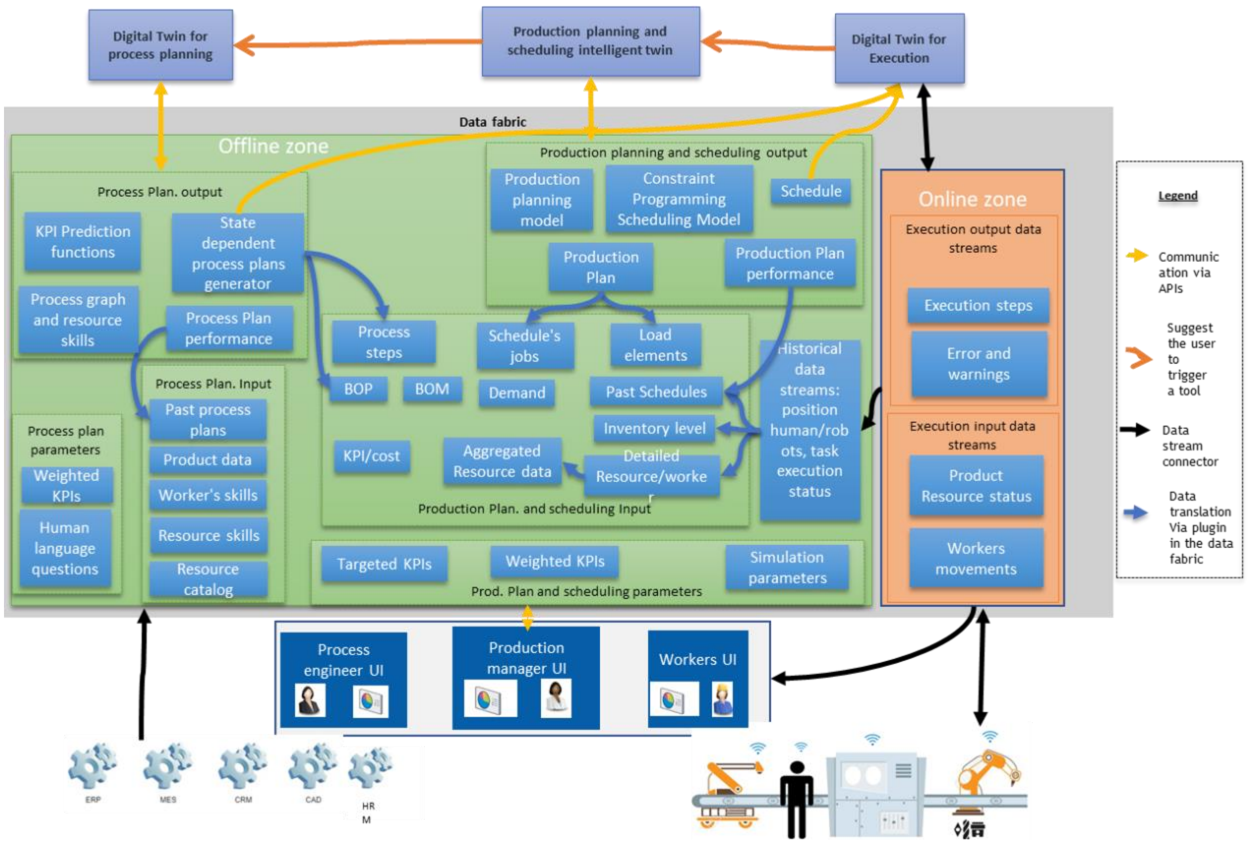

Figure 8 ASSISTANT Data Flow

Above is the initial ASSISTANT domain model along with the envisaged data flow. The initial input (“Process Plan Input”) consists of the product and production system data (Resources, Workers, Skills etc.) utilized to produce the “Bill Of Processes/Materials” (BOP/BOM) for a specific product. This data set (BOP/BOM) along with the generated scheduling constraints (resource workload, inventory capacities etc.) and KPIs serving as scheduling criteria are utilized for generating the production schedule by the “Production planning and scheduling intelligent twin”. The generated production schedule is then fed into the “Digital Twin for

Execution”, which generates and stores real-time status data related to the processes and the resources. Apart from the domain model and data flow explained briefly above, the diagram also depicts communication actions through components APIs. Such an action consists of a trigger event, i.e. an end-user presses a button on the relevant UI component which triggers an operation on the data (i.e. generation of production schedule). Similarly, there are some communications triggered by the system, which prompts the end-user to perform another operation. The most common case is the rescheduling of the production and/or choosing another production process based on shop-floor status (i.e. machine malfunction) and/or quality component measurement (i.e. bad quality).

In the context of the ASSISTANT system, different stakeholders can be identified. We suggest to analyse relevant stakeholders in the context of development, use, and other stakeholders. With this approach, we acknowledge that the ASSISTANT solutions are – as socio-technical systems - embedded in society. The differentiation of the stakeholders allows to include them in different stages in the development process.

Stakeholders in the context of the development of ASSISTANT are first of all the partners of the consortium. This group introduces their professional perspectives, viewpoints and also values through the decisions in the development process. However, it would not be enough to only include the developing partners in the development of the human centric architectural design: The designers, creators of manufacturing lines and integrators of solutions are relevant partners in this context as well. Additionally, the group of people affected by the ASSISTANT solutions should be included in several steps of the process. This group includes e.g. production planners, budget planners, shop floor planners, order execution managers, workers on the shop floor and people interacting with the ASSISTANT systems. In the process of the development process, the list of potential stakeholders from this group is updated.

Additionally, the ASSISTANT system addresses challenges in manufacturing. Therefore, the human centric architecture aims at integrating the perspectives of relevant stakeholders within manufacturing contexts. This is done on the one hand through policy documents and on the other hand through direct involvement of these stakeholders, whenever possible. This allows to include the perspectives of intertrade-organizations, works councils, unions, think-tanks, and NGOs. In a first analysis of existing policy documents, we identified 24 potential documents that can be considered in the further process of developing the human centric architecture.

4.2 Interventions within ASSISTANT

4.2.1 Questionnaire

Questionnaires are a suitable method to collect feedback by the involved partners and work on specific questions. In the process of the development of the human centric architecture, questionnaires were used to collect information for the start of discussions. E.g. In the beginning of the project, we developed a questionnaire that we sent out to the project partners. This questionnaire already integrated the ART-principles and asked the partners about their perspectives on the different components within the ASSISTANT systems. The questionnaire can be found in the appendix of this deliverable. Additionally, guiding questions were developed to learn about the perspectives of relevant stakeholders within the use cases. These questions were tested in collaboration with the Siemens Energy use case. The results will influence further discussions on process planning and the connected values, norms, and functionalities of ASSISTANT.

4.2.2 Focused (online) ethnography

To understand the development process of ASSISTANT, the authors attended project meetings, asked questions and identified potential issues for the development of the human centric architecture. This approach allows to understand the debates and issues discussed in the context of different components of the ASSISTANT system. Additionally, it allows active contribution to the discussions in the development process of the project. As the authors were not involved in all discussions, the approach resembles a focused ethnography (Knoblauch 2001).

4.2.3 Workshops

Based on the workshop concept (section 3.5) described above, we were using a consortium meeting to actually facilitate discussions about two main parts of the ASSISTANT project. Together with colleagues we explored which values are relevant for process planning and for real-time actuation. We chose these two parts because of practical reasons: It was possible to discuss process planning, as the involved experts within the consortium were present. Real-time actuation was chosen because the field visit was connected to a guided tour in the shop floor of Atlas Copco that is one of the use-cases.

The results will be discussed in the following sections. A more detailed workshop documentation can be found in section 7.2.1 and 7.2.2.[9] As described in the workshop concept, the workshop consisted of an introduction into the topic, as well as an overview of the approach. Subsequently the group gathered around discussion tables on process planning and real-time actuation.

4.3 Basic human centric architectural principles

Within ASSISTANT, there are two architecture documents that both continuously evolve. While the technical architecture is produced within task 2.2, the human centric architecture is the result of task 2.1, represented by this deliverable. As both documents evolve, the aim is to deepen the integration of both in the course of the project. The endeavour of bringing the two of them more and more together is supposed to actually bridge the gap between theoretical concepts and the actual responsible development of the digital twins.

The role of the overall architecture of ASSISTANT is to provide a framework for individual sub-architectures to collaborate on a common cause. The key for this approach is to introduce interoperability on the different sub-architectures.

The entire technical architecture can be found in a report from task 2.2 of the ASSISTANT project. While it is not public for everyone, the engineers and project members have access to it and can use it for their technical implementations. Within this document, there is merely a small subset that was extracted from the input from task 2.2. For the following revision of this architecture document, it is planned to include further details, if necessary. Thus, the contribution document serves as a repository for informing this architecture document.

Within this chapter, you will find general considerations and basic principles that are important for the human centric architecture for ASSISTANT. After describing the three principles that we

[9] The workshop concept describes the discussions about process planning and real-time actuation and introduce the roles of the participants, before describing the values and norms that were discussed, as well as the links between them.

have derived from the work of Dignum (see literature for more), we will describe the insights that we derived from the empirical work within ASSISTANT. This includes aspects of interoperability, security, data flows and alike. While this subchapter contains the values for the design on an overall level, the following subchapter will then go into more detail for different subcomponents and the different digital twins.

For being able to grasp values and norms easier in the following sections, we will use **bold font** for **values** and *italic font* for *norms*.

4.3.1 Accountability

Even though the technical architecture itself and the various documentations/deliverables accompanying the different components give an adequate description of the system and its expected behaviour, they are not intended for the users of the ASSISTANT solution as a target group, especially as they vary in experience, skills and background. Therefore, the value of **Accountability**[10] is to be implemented as concrete functionalities within the system. As a norm, *ASSISTANT aims at addressing* **accountability** *through self-documentation of actions and results through its human interface*. Due to the service-oriented architecture, where different components interact to collaboratively produce a decision, *it needs to be clear which component contributes what, and each component itself needs to work towards explainability to establish accountability to ensure outcomes and decisions are comprehensible*.

This is to be achieved through concrete functionalities, that

- The value **accountability** in the context of ASSISTANT will be translated into the following norms:
    - *ASSISTANT solutions allow the documentation of every action performed.*
    - *concrete information about the roles of different users of the system is provided.*
    - *Not overloading the user of unneeded information (i.e., display a summary message with a link towards more details).*
    - *Each "action" message should be accompanied with a list of data to be accounted for*
    - *A description of the expected behaviour of the action should be provided (i.e., "Based on the current resources status/state and the set of selected business KPIs the best production schedule can be calculated").*
    - *Clear decision trees and visualizations of decision processes that are then delegated to algorithms in the further course of the project have to be established.*
    - *contact information of accountable organizations and people involved in the development and at work are accessible to stakeholders.*

If the functionality is framed as interaction between the different components of the service-oriented architecture and if therefore the focus is shifted towards interoperability, it is necessary to mention at this point that not all questions of accountability can be delegated to the interaction. Also, each component itself contributes functionality for which it needs to be

[10] **Accountability** is the first principle of the framework. It was discussed in section 2.2.3 and corresponds to the understanding of a system action/reaction. To be able to criticize decisions that the system provides to the user, it is necessary to understand what happens within the system. Depending on different grades of automation, it is necessary to come up with differently detailed descriptions of the processes that are taking place within the system. If no human oversight is planned, this requirement is even more important.

held accountable. Therefore, not only the interaction and interoperability but also the functionality of the components itself need to be designed explicitly.

4.3.2 Responsibility

Responsibility as the second principle governs the decision-making process. It is mainly considered as responsibility concerning the humans interacting with the system. Additionally, environmental responsibility is discussed in the context of the data fabric. However, it is important to mention that the system consists of multiple services that interact with each other. For the user, the interaction with the system is an interaction process with the system in general - not with the sum of its individual components. Therefore it is necessary to reflect responsibility both, in a general way, but also how the different components reflect responsibility, while interacting with each other.

In the first version of the human centric architecture document, we suggested to develop a responsibility map that covers responsibilities of each stakeholder. This map is supposed to have two functions: On the one hand, it makes the responsibilities for the developers explicit and, on the other hand, makes the responsibilities of the stakeholders visible for the end users. The visualization of responsibilities is now integrated into the approach to translate values into norms, and functionalities.

- **Responsibility** is reflected through a norm formulated early in the project stage:
    - *The ASSISTANT solution by its definition has an embedded supervising mechanism coming from the pilot requirement that only humans should make decisions and the ASSISTANT solution should be considered as a decision support system.*
    - Addressing these requirements *each tool category (process schedule, production plan, production schedule, process execution etc.) should always include a human for decision making.* All the aforementioned modules have a human-in-the-loop principle implemented by respective graphical interfaces that supports the human through KPIs on decision making.

This understanding of **responsibility** facilitated discussions in the early stage of the project about where to locate functions. When we speak about multiple services that interact with each other, the definition of how to exactly cut the services is a matter of discussion. Depending on where a certain task – for example, ensuring data quality – is located, responsibility is also re-distributed.

- The SOA and the discussions of the involved project partners document how the value **responsibility** is connected to the norm that
    - *only humans should make decisions and the ASSISTANT solution should be considered as a decision support system.* The concrete functionalities that make sure, the norm is reflected, will be discussed in a further revision of this living document and in the sections referring to the components of the system (4.4).

4.3.3 Transparency

Transparency refers to the capability of describing, inspecting and reproducing how the AI systems suggests decisions (e.g. Dignum 2019a), or in other words: the way ASSISTANT systems suggest certain result. It therefore corresponds to the understanding of a system action/reaction. Even though the architecture itself and the various documentations/deliverables accompanying the different components gives an adequate description of the system and its expected behaviour, they are not intended to be elaborated by the users of the platform which vary in experience, skills and background. The main way of

ASSISTANT to address **transparency** is by self-documentation of actions and results through its human interface.

- The value **transparency** can be translated into norms in the context of manufacturing, e.g. that
  - o *the reasons why a decision and outcome occurred can be retraced*
  - o *it is clear to the users of the system which data points are connected in which way and how this affects the decisions and outcomes.*
  - o *the design decisions of the system are documented*
  - o *the places at which data is stored is known to users.*
  - o Concerning the ASSISTANT graphical user interfaces (GUI), these norms are reflected, as the *ASSISTANT GUI should*
    - ▪ *allow the documentation of every action performed*
    - ▪ *indicate all relevant and necessary information and provide information on further clarifications (i.e., display a summary message with a link towards more details).*
    - ▪ *each "action" message should be accompanied with a list of data to be accounted*
    - ▪ *A description of the expected behaviour of the action should be provided (i.e., "Based on the current resources status/state and the set of selected business KPIs the best production schedule can be calculated").*

The concrete functionalities that make sure, the norm is reflected, will be discussed in a further revision of this living document and in the sections referring to the components of the system (section 4.4).

4.3.4 Data Governance for ensuring ART-principles

The first general principle that is more concrete than the ART principles, described above, is supposed to ensure transparency and accountability through a data governance. The concrete descriptions of data flows and data managements offers interested stakeholders the opportunity to understand the underlying considerations for the functionalities. This information can also be used by from a HCI perspective to be included into concrete GUIs for the users. This chapter contains the introduction of the methodology for the description of data flows. First templates, filled out for specific parts of ASSISTANT, are attached in the appendix of this document (see section 7.8). for the development of the final human centric architecture, an updated version of the data flow descriptions will follow.

Following a structured approach templates have been created to help information collection and presentation. Mainly two different templates have been created addressing the two main aspects of data detailing process, the "Data Flows" and the "Data Management". "Data Flows" correspond mainly to the creator and the consumer of data while "Data Management" template details how the data are handled (i.e., creation rate, units etc.). In the following section the templates are being presented alongside with description of each template field.

In the appendix you can find the collected data up to this writing of the deliverable. The collected information is grouped per main ASISTANT modules which mainly correspond to ASSISTANT work package structure (WP3, WP4, WP5 and WP6). The level of detail per module varies as modules are in different development stages. This process of information detailing continues until the developments of each work packages are stabilized and the integration reaches a mature level.

4.3.4.1 Data Flows Template

MODULE NAME						
Input	**From**	**Where**	**What**	**When**	**How**	**Status**
Data flow 1	Module providing the input	HTTP (WEB communication over REST), FTP	JSON (over HTTP), XML/CSV	Periodic (every X fixed time), on demand (pull), or when available (push)	Server, client	concept, defined, approved, implemented, tested
Output	**To**	**Where**	**What**	**When**	**How**	**Status**
Data flow 2	Module receiving the output					

Table 3 Data Flow Template

Legend:

Input: a batch of data provided as a single package.

From, To: name of the module providing/receiving the input

Where: HTTP (WEB communication over REST), FTP

What: JSON (over HTTP), XML/CSV

When: Periodic (every X fixed time), on demand (pull), or when available (push)

How: Specify here if the module will implement the **server** part of the communication or if it will connect as a **client** to the server provided by the interacting module

Status

- **concept:** only shown in presentations but never discussed with other modules developers
- **defined (by provider):** where, what when and how have been defined and shared (eg. JSON or XML file structure defined in examples) by the producing module
- **approved (by receiver):** where, what when and how have been defined by the providing module and approved by the receiving module
- **implemented by provider:** relevant software code on the providing module has been implemented (after definition and approval)
- **implemented by receiver:** relevant software code on the receiving module has been implemented (after definition and approval)
- **tested:** complete data flow from the providing to the receiving module has been tested (after implementation by provider and receiver

4.3.4.2 Data Management Template

Module input												
Data	Producer	Input method	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency	Read/write
Name of the data	Who produces this data: it could be either a human operator, a report, or another module	How the data is entered the ASSISTANT systems: it could a GUI, a file, a database **resource**	The data format in which the data will be used (integer, double, signed, unsigned, ...)	Units of measure	Briefly explains why the data is needed	The component of the module that needs this data	It can be the local **consumer** database or nothing, if the data is volatile (**not stored by the consumer**)	How the data input is notified to the consumer (**DB query or REST service**)	How much of this data is needed. It can be a precise number or a reference to the decomposition tree (e.g. one for each hardware component)	How often the data is written **by the producer**	How often the data is read **by the consumer**	Details if the data can be changed by the module or if is left unchanged.

Table 4 Data Management Template I

Module internal results											
Data	Producer	Computed from	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency
Name of the data	Who produces this data: it could be either a human operator, a report, or another module	The data that contribute to the calculation of this data	The data format in which the data will be used (integer, double, signed, unsigned, ...)	Units of measure	Briefly explains why the data is needed	The component of the module that needs this data	It can be the module database or nothing, if the data is volatile	How the data input is notified to the consumer	One for each hardware component	How often the data is written	How often the data is read

Table 5 Data Management Template II

Module Output												
Data	Producer	Output method	Computed from	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency
Name of the data	Who produces this data: it could be either a human operator, a report, or another module	How the data will be made available to the other modules (either by authorizing access to the module database or by direct information transfer via web services)	The data that contribute to the calculation of this data	The data format in which the data will be used (integer, double, signed, unsigned, ...)	Units of measure	Briefly explains why the data is needed	The component of the module that needs this data	It can be the local **consumer** database or nothing, if the data is volatile (**not stored by the consumer**)	How the data input is notified to the consumer	One for each hardware component	How often the data is written **by the producer**	How often the data is read **by the consumer**

Table 6 Data Management Template III

4.3.5 Interoperability as negotiation of responsibility

Interoperability is defined as how easily a system can share information and exchange data with other systems. Interoperability in a "system of systems" is defined by the standard interfaces each system offers and the standard data representation (data format). As the ASSISTANT architecture is based on SOA (Service Oriented Architecture), it is composed by different services that cooperate to accomplish a common task, thus rendering ASSISTANT as a "system of systems".

It is in the nature of a distributed service-oriented architecture that different components are developed by different people. While interoperability is mostly framed as a technical issue in the architecture, it is important to also reflect how it is established. **In order to make sure that the overall system is developed responsibly, it is important to negotiate about the distribution of responsibility for each specific component**. Decisions in a distributed system are more difficult to take in this more complex environment as different actors play roles that are also part of different organizational entities. Fair distribution of responsibility as a value requires certain norms to be considered:

- Responsibilities must be clearly defined among the project partners. This means that interfaces need detailed specifications both technically and in relation to the boundaries of contentwise tasks.
- As interfaces negotiate responsibilities it is important that negotiations about interfaces happen in a fair environment with all necessary stakeholders at the table.
- the fair exchange of arguments shall ensure that the implementation is chosen that works best altogether for the entire system and especially for its users.

What we can see here is that the norms we try to identify are not only important in relation to the product that is built in the end but also for the process in which they are developed.

In the following sections, a few technical details are discussed for ensuring the interoperability from a technical perspective. Interoperability requirements are considered a non-functional requirement, but its role is important in contributing to efficient development and the integration of different tasks. ASSISTANT's interoperability will consist of three main types of interoperability: the information interoperability, the technical interoperability and, finally, the presentation interoperability. Information interoperability defines how information is to be shared among the different stakeholders and is described on section 4.3.5.1. Technical interoperability defines how technical services are shared and connected to each other; this aspect is described in section 4.3.5.2. Finally, presentation interoperability defines a common look-and-feel approach through a common portal-like solution which guides the user to the underlying functionality of the set of services. This kind of interoperability is discussed in section 4.3.5.3.

4.3.5.1 Information Interoperability

Data representation is the main focus of information interoperability. The main requirements imposed on the ASSISTANT solution regarding data representation are the clear, shared expectations regarding the contents, context and meaning of that data. Even though there are standard formats for different data domains (i.e. STEP ISO for 3D representation), we will not focus in this section on the most efficient data representation per domain (which will probably result in a multitude of data representation formats) but on a "global" data representation format capable of supporting ASSISTANT data-related requirements. In the ASSISTANT architecture, we can easily distinguish a major component related to the data itself, the Data

Fabric. This component would play the role of ASSISTANT data storage, thus data also from different domains should be represented in a common format.

Within ASSISTANT, domain modelling workshops are taking place to ensure that the information that needs to be contained in the domain models is agreed upon within the project team. As this workshops are currently ongoing, results and documentation of these workshops will be part of the final human centric architecture.

4.3.5.2 Technical Interoperability

Technical interoperability is the ability of two or more components/applications to accept data from each other and perform a given task in an appropriate way without the need of additional intervention. The following paragraphs provide general information about the standard communication protocol proposed to exchange data among the ASSISTANT components.

This technical interoperability is achieved through common file formats as well as communication protocols that each component commits. Through an agreement on common technical standards, it is ensured that each component as a separate individual entity can fulfil their negotiated responsibility for the overall system.

Data formats can generally be separated into two categories, schema based and schemaless based. Schema based formats have the advantage of being able to be considered valid or not according to a predefined data structure (schema), while the schemaless can only be evaluated as well as formed (syntactically correct), but no rules can be applied regarding the structure of these data. Schemaless data, on the other side, provide greater flexibility since they can accommodate any kind of data and can be expanded with less effort than the schema based.

To ensure the information interoperability, ASSISTANT will use standard formats for modern SOA. As file formats, XML, JSON and CSV will be used:

- In computing, **Extensible Markup Language (XML)** is a mark-up language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
- In computing, **JSON (JavaScript Object Notation)** is an open-standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is the most common data format used for asynchronous browser/server communication, largely replacing XML.
- **CSV (comma separated values)** format is a common data exchange format that is widely supported by consumer, business, and scientific applications. Among its most common uses is moving tabular data between programs that natively operat e on incompatible (often proprietary and/or undocumented) formats.

To ensure the information interoperability, ASSISTANT will use standard protocols for modern SOA. As protocols, HTTP and REST will be used:

- The **Hypertext Transfer Protocol (HTTP)** is an application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.
- **Representational state transfer (REST)** are Web services providing interoperability between computer systems on the Internet. Using HTTP, as is most common, the kind of operations available include those predefined by the HTTP verbs GET, POST, PUT, DELETE and so on.

4.3.5.3 Visualization Interoperability

The definition of templates for the screens was the initial step for the front-end developments. The templates were designed as a guideline for the implementation of the user interfaces. The existence of some common templates had the goal of achieving a uniformity of the user interface both in terms of appearance and in terms of behaviour. The basic defined templates are presented below.

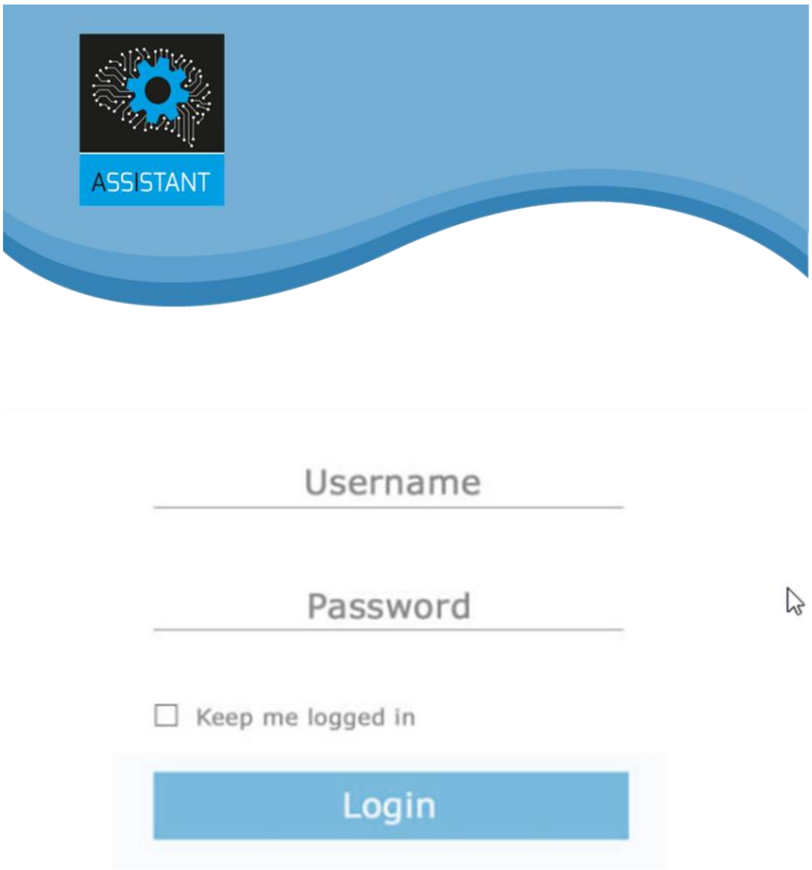

Figure 9 Loginscreen

The log-in screen (Figure 6) template was designed to be used as guideline of the page(s) that will be employed for user log-in to the applications. The welcome/home screen (Figure 7) is the screen that the user “sees” as soon as he logs in the application. It consists of three parts: header, content, and footer. The header contains the logo of the project, the logout button and the menus for navigating inside the application. The footer contains the logos of the developer(s) and the project logo. The content of the welcome screen is a dashboard style containing various action buttons, information, etc. The “Various Screens” template (Figure 8) is the template for the implementation of the other screens of the user interface. It also consists of the header, the footer, and the content section. The content section is divided into two parts: the left part contains the context menu (if applicable), while the right part contains the actual content of the screen.

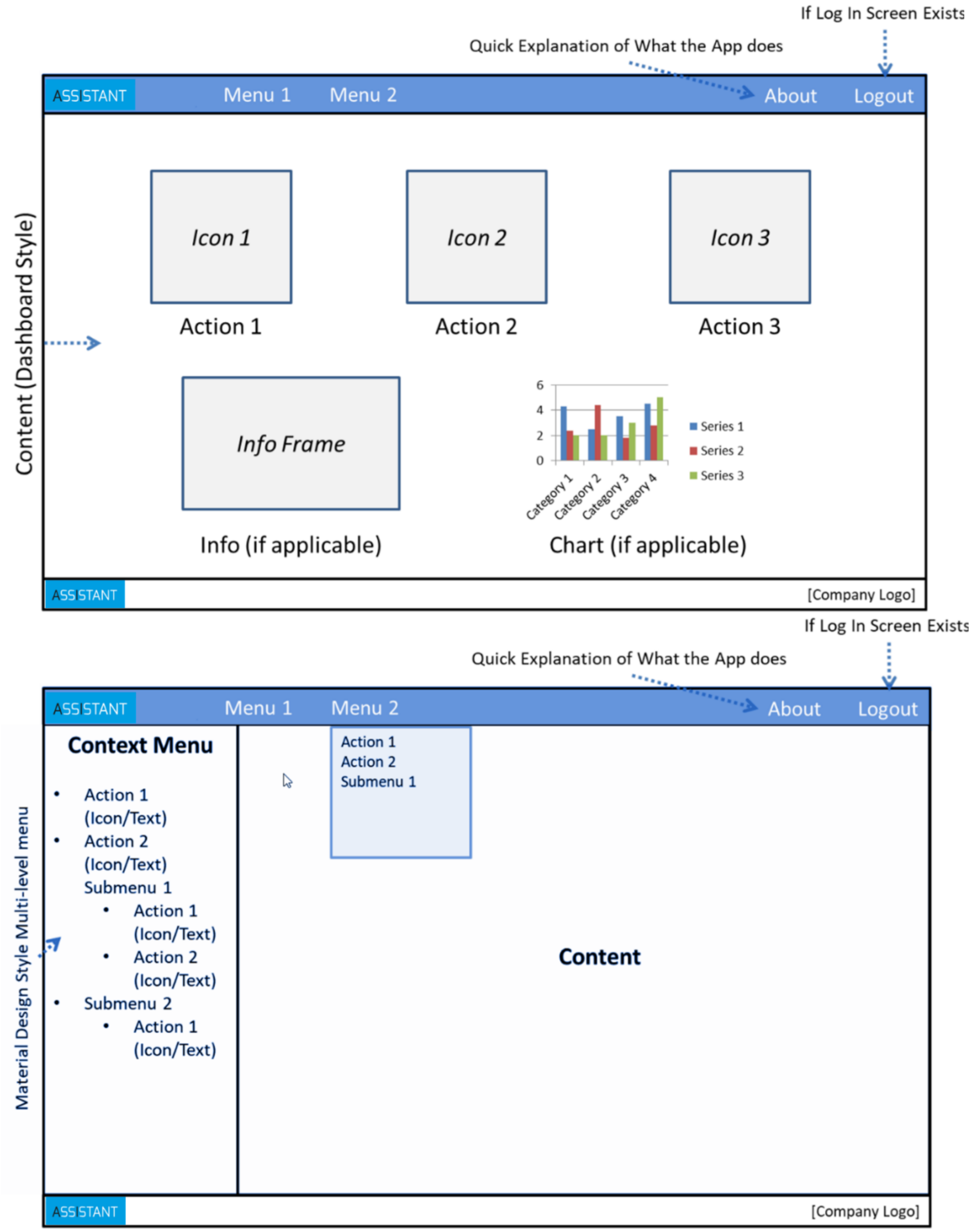

Figure 10 Various Screens

4.3.6 Security and Encryption

IT security is a set of strategies that prevents unauthorized access to organizational assets such as data. Data security strategies are divided mainly into two categories of strategies that will prevent unauthorized access to data (Access Control System) and data encryption strategies that even if data access fails, data cannot be read. ASSISTANT will implement both of these strategies. For the first category, an RBAC (Role Based Access Control) mechanism based on a central authentication system (CAS) will be developed, while data transition channels will enforce data encryption with a Transport Layer Security (TLS). The access control will be centrally implemented, but each client must make sure that access to security areas and data are properly handled.

4.3.7 Role Based Access Control (RBAC)

Role-based access control (RBAC) is an access-control mechanism defined around roles and privileges. The components of RBAC are permissions, subject and roles. Combined, these three elements form the security policy rules (i.e. role “Administrator” has permission “Creation” on subject “Users”). These policies are to be stored in a database that will implement the RBAC system. The policies will be enforced on service usage and/or data retrieval. Essential for such a system to work is the successful identification of the user and its role. This is called user authentication and is discussed in the next sections.

4.3.8 Central Authentication System (CAS)

The Central Authentication Service (CAS) is a single sign-on protocol for the web which is responsible of identifying and authenticating a user of the system to be the one that the user claims to be (by the use of a username and password). Its purpose is to permit a user to access multiple applications while providing their credentials (such as username and password) only once. It also allows web applications to authenticate users without gaining access to a user's security credentials, such as a password. The name CAS also refers to a software package that implements this protocol. The ASSISTANT platform foresees the use of different modules/applications by different users/companies. To allow one single access point to each user, according to the contract stipulated, i.e. which modules are available to this specific user, as part of the “user friendliness” the CAS service allows only one single access, avoiding multiple insertions of credentials. By doing this, also the security part is being handled in a centralized way by one service.

The CAS server and clients are the two physical components of the CAS system architecture depicted in Figure 9. They communicate by means of various protocols.

CAS Server
The CAS server is Java servlet built on the Spring Framework whose primary responsibility is to authenticate users and grant access to CAS-enabled services, commonly called CAS clients, by issuing and validating tickets. An SSO session is created when the server issues a ticket-granting ticket (TGT) to the user upon successful login. A service ticket (ST) is issued to a service at the user’s request via browser redirects using the TGT as a token. The ST is subsequently validated at the CAS server via back-channel communication. These interactions are described in great detail in the CAS Protocol document.

CAS Clients
The term “CAS client” has two distinct meanings in its common use. A CAS client is any CAS-enabled application that can communicate with the server via a supported protocol. A CAS client is also a software package that can be integrated with various software platforms and applications in order to communicate with the CAS server via some authentication protocol (e.g. CAS, SAML, OAuth). CAS clients supporting a number of software platforms and products have been developed.

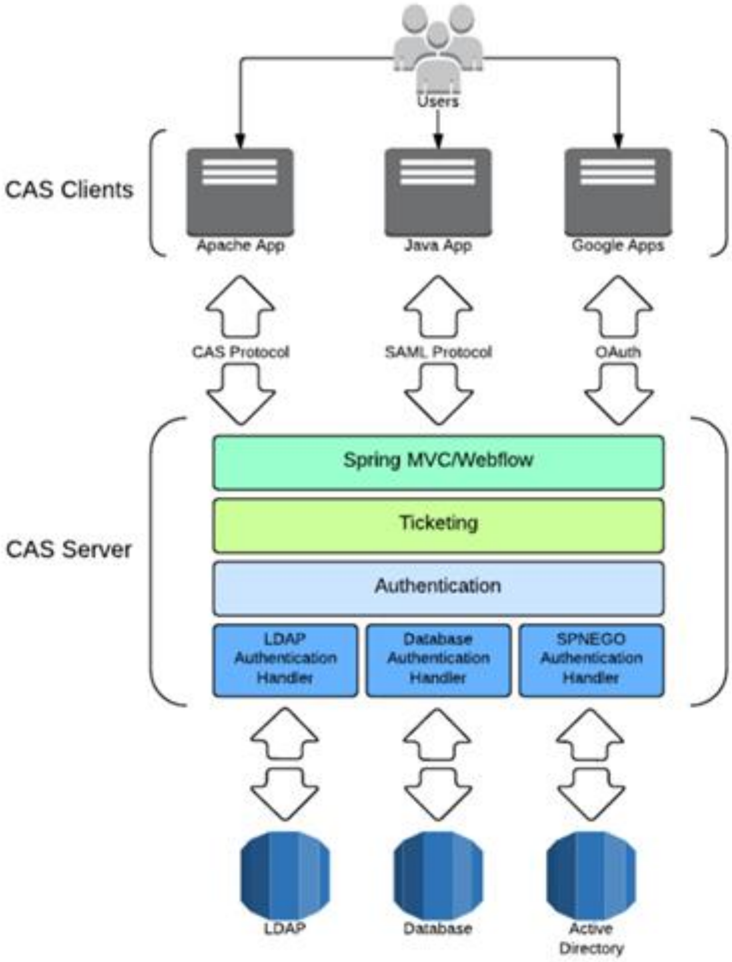

Figure 11 CAS Server Architecture components

4.3.9 Data Encryption

Exchange of data between components will be encrypted via industry standard technologies which make use of symmetric/asymmetric cryptography, such as TLS 1.2. Specific certificates for TLS 1.2 will be created and maintained to enforce data encryption on the communication level.

On the other hand, data storage should encrypt or otherwise obfuscate sensitive data as an extra security measurement.

4.3.10 Privacy

Transparency requires the accessibility of information. Privacy requires that not all data that might be stored is accessible. Additionally, storing logfiles that make processes within the system potentially traceable comes with costs. They are often very technical, they need processing and storing, and they require certain skills to be understood and to be analysed.

- A strategy for understanding how a concrete decision was taken by the system needs to be developed. The output of this strategy could be a visualization that is not only traceable in cases of conflicts but also presented to the user.

4.3.11 Iterative processes and requirements open for adjustment

Requirements are by nature gathered in the early phase of projects, and only sometimes continuously monitored, refined and fed back into projects. The process of Design for Values, however, requires time to evolve. Therefore, results of that process, that need to be fed into the development of the components, can only evolve at a point where the requirements are already collected and documented. Requirements often serve as a contract between the users, who require functionalities, and the developers, who implement functionalities. It is therefore relevant what is put in the requirements documents. In order to develop a system that is ethical by design, it will be necessary to include the insights from the human centric architecture in the requirements. Therefore, the following norms can be derived:

- Requirements that are collected in earlier phases of the project need to remain flexible. This way, insights from the human centric architecture and the discussion it facilitates can be included as requirements that are equally to be fulfilled as the early functional requirements.
- It is necessary to rethink methodologies both of requirements engineering and for further projects that acknowledge the lack of synchronicity between the moment when responsibility aspects would be necessary in the project and the moment when we are able to implement them. This is especially true, as the approach we chose is not only supposed to raise issues but also intervene in the development process.

4.4 Component-related human centric architectural principles

The general considerations from the previous section concerning values, principles, and norms related to ASSISTANT are a central part of the human centric architecture. In this section, we will discuss context-specific considerations, results and reflections concerning the different components of ASSISTANT. As this document is a living-document, not all sections are filled with content, as the following sections are specifications and additions to the general considerations. Additionally, not all components have been discussed in detail so far, as this is part of the ongoing development process.[11] However, concerning process planning, real-time actuation and the data fabric, some first results will be presented. These results stem from discussions and workshops, which are documented in the annex, sections 7.2.1ff).

4.4.1 Process planning

The process planning component comes with several specifications of the discussed values:

- The value **transparency** will be considered through the following norm:
  - *Transparency in process planning means that all affected users know how the system calculates suggestions and which data is used.*
- The value **responsibility** was discussed in the context of non-discrimination and will be considered through the norms:
  - *non-discrimination in the context of process-planning means that human/worker assignment does not take place due to skills.*
  - Additionally, *workers and machines should only be assigned to task for which they are qualified.*

[11] In the following project months, the discussions of all technical components will be continued in close collaboration with T2.2. The living document will be updated regularly and allows therefore to adjust the architecture in the process.

These values and norms are the result of workshop discussions, in which the component was analysed by the participants of the project (the detailed documentation can be found in section 7.2.1). These preliminary results will be amended with the outcomes of further workshops and discussions concerning the process-planning component, including a translation of the defined norms into functionalities.

Additionally, the participants of the deliberations highlighted that technical robustness is a major topic, especially against uncertainty (e.g. stochastic fuzzyness, but also resource and product changes). This discussion was linked to the question whether and how the system is capable to be applied to different product variants, and whether the modularity allows tuning to different product variants.

An additional open topic is how to include the preferences of the people affected: Some participants mentioned that it might be difficult to determine the preferences of the workers concerning the task variability. We discussed possible approaches: a) Asking the workers how they would like to structure their work, b) interviewing process planners who know the boundary conditions and most likely preferences in their context. This topic must be addressed in the further development of ASSISTANT.

4.4.2 Production planning

(the documentation of production planning specific considerations follows in the final version of the human centric architecture)

4.4.3 Scheduling

(the documentation of scheduling specific considerations follows in the final version of the human centric architecture)

4.4.4 Real-time actuation

Concerning real-time actuation, the general values and norms need to be amended and concretized in the following way:

- The value **Transparency** was discussed intensively, as there were different understandings and links to other values (e.g. security, interactivity) and norms. It should be translated in the norm
    - **Transparency** in the context of real-time actuation is a precondition to create a system that is *interactive and allow the users to bring in feedback.*
    - Additionally, *decisions of the system should be **transparent** and certain data should be presented together with the decision.*
- The values **safety** and **security** were initially mentioned as the same concept, but have now been translated into the following norms
    - **Safety** *for the operator should always be the number 1 priority.*
    - **Safety** *in the context of real-time actuation means to increase ergonomics for the worker.*
    - **Security** *in the context of real-time actuation requires that personal data is stored safely.*

- The values **stability** and **robustness** are essential for optimization, but also the well-being of the operators[12]
    - *Therefore,* ***stability*** *in the context of real-time actuation means that decisions and proposals should be stable and farsighted and not be changed (too) frequently.*
    - ***Robustness*** *means that decisions and behaviour should be continuous and consistent.*
- Additionally, **Accessibility** was discussed in the sense of usability, and therefore relates to responsibility in terms of who is able to use the component. The value will be translated into norms that reflect the *explicitness of information modelling*, in order to allow users to understand what is going and to interact meaningful with the component. Additionally,
    - **Accessibility** *in the context of real-time actuation means that also non-experts should be able to interact with the system.* This norm addresses the idea that the system should be both *easy to use* as well as flexible in the sense that it should be *capable of performing different tasks*. It was acknowledged, however, that even though this general flexibility is to be achieved, that the system should *reflect limits to where the tool can be used*.
- The value **quality** was addressed as well. As the overall goal of the system is aspired to increase productivity, quality refers to both, products, as well as the quality of data and predictions of the machine learning components. Therefore, the following norm was defined:
    - **Quality** in the context of real-time actuation addresses the *reusability and maintainability of data, information and knowledge*.
- **Equality** as a value related to responsibility was translated into the following norm:
    - *all workers should be treated equally,* and *no single worker should be preferred in some way.*

The participants of the discussions until now were project participants, including developers – not necessarily directly involved in developing technology for RTA but also in general – infrastructure providers, an industry partner as well as general project managers responsible for integrating different work threads. Additionally, actors responsible for the ethical development of the project had a seat at the table. The discussion was moderated by a representative from WP2.

In general, the participants suggested values that are closely related to optimization and quality improvements for the industry partner but also values that are relevant for the well-being of the operators. So for example, transparency as a value was identified that matters both for the workers and the quality as far as a worker who is knowledgeable about his role and his contribution to the general production process is more productive.

In the next steps, the values will be validated and the translation of values to norms will amended through adequate functionalities in close collaboration with T2.2.

4.4.5 Data Fabric

(the documentation of production planning specific considerations follows in the final version of the human centric architecture)

[12] This supports the commitment of the project partners to improve quality through the operators wellbeing, as *nervousness of the worker through constant changes is to be avoided.* This commitment was a starting point in the documented workshop.

5. Further process, limits, and reflections

In this chapter, we describe the further process towards the final human centric architecture document. We will reflect and announce next steps, including upcoming updates of the human centric architecture, discuss the lessons learnt from the process, the integration with the other approach in WP2, and limits of the approach.

Compared to D2.1 no new section was added, however the content of the chapter was updated according to the current state of the living document, highlighting the learnings from the process of development of the intermediate human centric architecture.

This document will be updated, maintained, and revised in the course of the project. Specifically, updated versions of the human centric architecture are planned to be circulated every three months. This allows to update, integrate and reflect the values, norms and functionalities in close collaboration with T2.2 on the technical architecture. Additionally, we aim at providing a - more or less - complete list of values and norms six months before the final human centric architecture will be documented in D2.4. This gives time for adjustments, iterative cycles, and the reflection of the implementation. The final version of the document provides detailed information about the concrete implementation of the components, as well as in-depth discussions of selected functionalities.

The ART-principles – as discussed above – are the starting point for the development of this human centric architecture. In the initial version of this document, reflections have been shared that shed light on the potential issues that might arise and how to cope with them. It is important to note that the ART-principles function as a starting point. During the concretization and implementation process it is necessary to put them into practice. Therefore, this version of the document suggests a methodology and starts applying it to ASSISTANT. First results show that the ART principles are accompanied through other values, - e.g. safety, a value connected to responsibility, while the detailed relationship can be explored in further revisions of the human centric architecture document.

So far, the ART-principles have only been applied to the overall architecture, the technical integration - and therefore on a high level. Additionally, the workshop concept and the initial reflections concerning process planning and real-time actualization have been added in order to concretize values into norms for the specific contexts of application. For the next revision, it is planned to take a closer look at the individual components that constitute the digital twins and reflect on these components from the perspectives of accountability, responsibility, and transparency. Additionally, the requirements for the respective component need to be considered: some of them might change, some of them might be extended or adapted to the new insights reflected in the course of the process.

The process of creating a human centric architecture is a learning process. We would like to highlight the following observations from the process:

1) The development of a methodology and the concretization of the approach to translate values into norms and functionalities goes beyond the current state of the art. The workshop concept allows to discuss values, norms, and functionalities, while documenting the decisions of the involved people at the same time. It makes the decisions and approach transparent and gives an overview on how abstract values are addressed in the context of ASSISTANT.
2) A central observation is that discussions and workshops about sensitive topics such as values can hardly be held effectively in an online environment: On-site workshops support the integration of all participants through moderation. Additionally, characteristics of physical meetings – including short comments, direct reactions, non-

verbal communication and overlapping statements - can foster a creative exchange that can hardly be achieved in online environments. Therefore, we recommend to plan additional physical workshops for the discussion of values, norms, and functionalities of the ASSISTANT solution and its components.

3) While requirements give a first idea on issues and topics that need to be reflected in the implementation, many of them have to be discussed in detail. The approach to use the requirement documents as a preparation for workshops allowed to use the results from the requirements documents as starting points in order to discuss them further.
4) We noticed that the translation of values into norms and functionalities requires suitable visualization approaches, both in the workshop settings, but also in the documentation of the results. Therefore we suggest to explore and experiment with visualizations that allow to trace how abstract values are linked to concrete functionalities in the further revisions of the human centric architecture living document. All stakeholders will benefit from suitable visualization methods, e.g. as it should make the links between functionalities and the underlying values (and vice versa) explicit.

Within the ASSISTANT project, there has been an amendment regarding the timeline of the different inputs. Before the amendment, we planned to base our reflection on a questionnaire that was supposed to replace in-person workshops, which could not take place due to the pandemic. You can find the questionnaire in the appendix. After the amendment, however, we were able to base our reflections directly on the inputs from the other work packages and tasks. Additionally, we integrate input collected at workshops with the participants of the technical (WP3-6) and industrial use cases.

These reflections on the human centric architecture mostly focus on potential issues that could arise during the further course of the project and how to cope with them. However, we can also notice that even though we have discussed responsible development and the ethical assessment on a quite abstract level in project meetings, we have experienced very positive reactions from the project partners. This is mentioned here because presenting only potential risk might create an imbalanced perception from a normative perspective. In the discussions that we have been involved so far in the project, we already influenced the mindsets. It is fair to state that we therefore may already have positively impacted decisions that were taken in the design of the architecture and in the writing of D3.1, D4.1, D5.1, D6.1, and D7.1. Therefore, it is crucial to create such a reflection space within projects such as ASSISTANT, in order to allow the deliberations described in this document.

5.1 Integrating different approaches from work package 2: Assessment based on trustworthy guidelines

Within work package 2 of the ASSISTANT project, there are two different approaches towards the responsible development of the digital twin (see Figure 2 Visualization of the different approaches for responsible design within ASSISTANT). While this document mostly focuses on the development of the architecture, there is also a task that defines and evaluates assessment criteria. At this point in time, a preliminary list of assessment criteria already exists and is included in this document.

While the methodology for the development of a human centric architecture is explicitly not supposed to define a check-list that needs to be ticked off, the aspects that are raised within the assessment criteria might still inform the development process of the digital twins and can also serve as a starting point for considerations about issues of responsibility.

While the architecture tries to integrate an ex-ante approach, the criteria perform an ex-post approach. Therefore, different frameworks are employed to solve the two different tasks. For the development of the criteria and the assessment, the concrete tools of Trustworthy Guidelines (High-Level Expert Group on AI, 2019) and ALTAI – Assessment List for Trustworthy AI – are used, while the human centric architecture mobilizes the ART-principles as described above.

ALTAI aims to provide a basic evaluation process for Trustworthy AI self-evaluation. Organizations can draw elements relevant to the particular AI system from ALTAI or add elements to it as they see fit, taking into consideration the sector they operate in. It helps organizations to understand what Trustworthy AI is, in particular what risks an AI system might generate. It raises awareness of the potential impact of AI on society, the environment, consumers, workers and citizens (in particular children and people belonging to marginalized groups). It promotes involvement of all relevant stakeholders (within as well as outside of an organization). It helps gain insight on whether meaningful and appropriate solutions or processes to accomplish adherence to the requirements are already in place (through internal guidelines, governance processes etc.) or need to be put in place (High-Level Expert Group on AI, 2020).

ALTAI is supposed to help in fostering responsible and sustainable AI innovation in Europe. It seeks to make ethics a core pillar for developing a unique approach to AI, one that aims to benefit, empower and protect both individual human flourishing and the common good of society.

ALTAI and the trustworthy guidelines share the understanding of the importance of including values in design processes and of making these values explicit. It therefore makes sense to ensure an exchange between the two processes within the project. This will be established through the ethical management plan that is developed in task 2.3.

An initial component analysis based on these guidelines was performed within other tasks of WP2. The initial results are placed in the appendix of this document. The considerations inform and shape the further development of individual components and therefore also might have an impact on the overall architecture, which is the reason to include them in the appendix of this version of the human centric architecture document.

While for the developer, it might be easier to adopt concrete instructions instead of questions for the developing process, the assessment is still phrased in the form of questions, as this is the format that is required for the evaluation process. However, it might be an option to translate these questions into concrete instructions for the developers in the upcoming revision of the human centric architecture.

5.2 Limits and Reflections

This section refers to the limits of the human centric architecture document. Therefore, considerations on the role of this document within the project, as well as the current state of the living document are pointed out.

Additionally, we highlight a possible blind spot of the document as it is created within and applied to a specific context: by authors with a background in social sciences and science and technology studies within a consortium on manufacturing using AI. Integrating different perspectives is therefore crucial and will be fostered through the dissemination of the approach in different contexts.

While we try to be transparent about our approach, we also want to acknowledge that the process we chose is not without limits. By basing the development of the human centric architecture on the ART-principles, we have chosen one specific approach where we could have also decided for others. We have provided reasons for this decision. However, this means that we do not include all the dimensions for reflection that are out in the field. We will keep that in mind and add other frameworks and approaches as they seem necessary and seem to fit.

Another limit to the development of the architecture document is that, so far, the project team had a few on-site workshops, but other relevant stakeholders such as workers or unions were not yet included in the development process of the document. We hope that in the further course of the project, it will be possible to facilitate additional workshops and discussions in person. Especially in the very normative context of responsibility and ethics, meetings in person are very helpful, as non-verbal communication and ideation takes place in a different way than online.

6. References

Aldewereld, Huib, Virginia Dignum, and Yao-hua Tan. 2015. 'Design for Values in Software Development'. Pp. 831–45 in *Handbook of Ethics, Values, and Technological Design*, edited by J. van den Hoven, P. E. Vermaas, and I. van de Poel. Dordrecht: Springer Netherlands.

Aler Tubella, Andrea, Andreas Theodorou, Frank Dignum, and Virginia Dignum. 2019. 'Governance by Glass-Box: Implementing Transparent Moral Bounds for AI Behaviour'. Pp. 5787–93 in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. Macao, China: International Joint Conferences on Artificial Intelligence Organization.

Algorithm Watch. 2021. 'AI Guidelines Global Inventory'. *AI Guidelines Global Inventory*. Retrieved 30 June 2022 (https://inventory.algorithmwatch.org).

Dano, B., H. Briand, and F. Barbier. 1997. 'A Use Case Driven Requirements Engineering Process'. *Requirements Engineering* 2(2):79–91. doi: 10.1007/BF02813027.

Dignum, Virginia. 2019a. *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*. Cham: Springer International Publishing.

Dignum, Virginia. 2019b. *Toward AI Systems That Augment and Empower Humans by Understanding Us, Our Society and the World Around Us*. *Deliverable*. 1. Umeå Universitet.

Floridi, Luciano, and Josh Cowls. 2019. 'A Unified Framework of Five Principles for AI in Society'. *Harvard Data Science Review*. doi: 10.1162/99608f92.8cd550d1.

Garlan, David, and Mary Shaw. 1993. 'An Introduction to Software Architecture'. Pp. 1–39 in *Series on Software Engineering and Knowledge Engineering*. Vol. 2. WORLD SCIENTIFIC.

Hagendorff, Thilo. 2020a. 'The Ethics of AI Ethics: An Evaluation of Guidelines'. *Minds and Machines* 30(1):99–120. doi: 10.1007/s11023-020-09517-8.

Hagendorff, Thilo. 2020b. 'The Missing Link in Putting AI Ethics into Practice'.

High-Level Expert Group on AI. 2019. *Ethics Guidelines for Trustworthy AI*. *Report*. Brussels: European Commission.

IDEO, ed. 2015. *The Field Guide to Human-Centered Design: Design Kit*. 1st. ed. San Francisco, Calif: IDEO.

Jobin, Anna, Marcello Ienca, and Effy Vayena. 2019. 'The Global Landscape of AI Ethics Guidelines'. *Nature Machine Intelligence* 1(9):389–99. doi: 10.1038/s42256-019-0088-2.

Knoblauch, Hubert. 2001. 'Fokussierte Ethnographie'. *Sozialer Sinn* 2(1). doi: 10.1515/sosi-2001-0105.

Leidner, Jochen L., and Vassilis Plachouras. 2017. 'Ethical by Design: Ethics Best Practices for Natural Language Processing'. Pp. 30–40 in *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*. Valencia, Spain: Association for Computational Linguistics.

LUMA Institute, ed. 2012. *Innovating for People: Handbook of Human-Centered Design Methods*. First edition. Pittsburgh,PA: LUMA Institute.

Winfield, Alan F. T., Serena Booth, Louise A. Dennis, Takashi Egawa, Helen Hastie, Naomi Jacobs, Roderick I. Muttram, Joanna I. Olszewska, Fahimeh Rajabiyazdi, Andreas Theodorou, Mark A. Underwood, Robert H. Wortham, and Eleanor Watson. 2021. 'IEEE P7001: A Proposed Standard on Transparency'. *Frontiers in Robotics and AI* 8:665729. doi: 10.3389/frobt.2021.665729.

7. Appendix

Table of contents of the appendix

7.1 Abbreviations

Table 5 Abbreviations

Abbreviation	Meaning
ASSISTANT	LeArning and robuSt decision SupporT systems for agile mANufacTuring environments
ART-principles	Principles of Accountability, Responsibility, Transparency, referring to (2019a)
CSV	comma separated values
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
REST	Representational State Transfer
RTA	Real time actuation
SOA	Service Oriented Architecture
T[number]	Task [number] within ASSISTANT
WP[number]	Workpackage [number] within ASSISTANT
XML	Extensible Markup Language

7.2 Workshop concept for linking values, norms, and functionalities

The following table contains the concrete planning for the workshop format that is supposed to come out with a mapping of values, norms and functionalities. After a general introduction, the main work is done at the different world café discussion tables.

Time	Duration	Aim	Content	Method	Material
00:05	00:05	Arrival and time buffer			
00:20	00:15	Contextualization: Clarify why it is important to anticipate the societal and organizational impact of technologies and give a brief overview on suitable approaches	Example how technologies shape society and vice versa (Winner?) Discussions on responsible AI Approach: Design for values	presentation	power point
00:40	00:20	Summary: What do we know from policy documents and project documents (requirements) concerning relevant values within the ASSISTANT project?	- Overview on values from policy documents (Lit review from global AI inventory) - Overview on results from project documents	presentation	
01:25	00:45	- Discuss and document perspectives of the participants on relevant values and norms within the ASSISTANT system and parts (process planning, production planning, scheduling, real time actuation, data fabric) - Link values with corresponding norms (and vice versa)	Participants discuss relevant values and norms within the domain of each part of the ASSISTANT system	World café, including a host at each table / part of the system including EUV, possibility to change tables after 20 minutes	3 iPads hat document the results from the discussions, allow to exchange results within workshop group and serve as external displays in the workshop setting
01:45	00:20	Presentation of results from World Café	Results from previous step	4 minutes per table, presentation of the main results	
01:55	00:10	- short break -			
02:25	00:30	Aggregation of results and prioritization of identified values, including documentation of the reasons why the decisions were taken	The participants discuss the results and their prioritization. Different methods for weighing values are applied. Tensions between conflicting values as well as the following decisions are documented.		

02:40	00:20	Reflection: What were the decisions of today? Which parts of the architecture have been changed? Are all relevant perspectives included (and how might missing perspectives be included)?	Each participant will answer the questions and share them with the group.	Short round, answer to guiding questions.	Documentation on Flipchart
02:45	00:05	Wrap up and outlook			

7.2.1 Workshop documentation on process planning (May 2022)

Concerning process planning, a first workshop with stakeholders from the consortium took place in May 2022. The context and the results of the workshop will be described in the following paragraphs:

Workshop results

The participants of the process planning part of the workshop brought different perspectives from the ASSISTANT consortium into discussion: The Participants are involved in developing the digital twin for process planning, in contributing specific parts (such as chat bot and simulation parts), and industrial perspectives (e.g. validating and making buy or make decisions, checking simulation input data, receiving process plans; data mining and providing example data sets). Additionally, a perspective from model acquisition from schedule data and a moderator (as a facilitator for linking values, norms, and functionalities) were part of the discussion.

The group therefore represented different perspectives from the ASSISTANT consortium, including participants who are directly involved in the development of process planning, as well as an "outside" perspective on process planning. However, it was not possible to integrate external stakeholders (such as workers, workers council representatives, political representatives) into the discussion.

Concerning transparency, the discussion focussed on the technical robustness of the digital twin system, especially against uncertainty (e.g. stochastic fuzziness, but also resource and product changes). This discussion was linked to the question whether and how the system is capable to be applied to different product variants, and whether the modularity allows tuning to different product variants. The participants of the discussion agreed that it should always be possible to understand, how the digital twin for process planning calculates a certain suggestion. The value **transparency** was therefore translated into a context-specific norm for production planning: "Transparency in process planning means that all affected users know how the system calculates suggestions, which data is used".

The other value discussed extensively was related to the question of (non-)discrimination: It was discussed whether it should be possible to use skill data for process planning. Concerns in this context were that if process planning considers such data, the tasks for the workers can be monotonous (e.g. if the used data states that person A takes 2 minutes to complete task B, while person C takes 3 minutes to complete the same task, the optimization might result in attributing the task always to person A). This is problematic, as the attempt to optimize for efficiency in terms of working time would lead to a possible change of the tasks performed at the workplace. Other participants mentioned that it might be difficult to determine the preferences of the workers. We discussed possible approaches: a) Asking the workers how they would like to structure their work, b) interviewing process planners who know the boundary conditions and most likely preferences in their context. This topic has to be addressed in the further development of ASSISTANT. In relation to **non-discrimination** the workshop participants formulated the norm that *non-discrimination in the context of process-planning means that human/worker assignment does not take place due to skills*. Additionally, *workers and machines should only be assigned to task for which they are qualified.*

7.2.2 Workshop documentation on real-time actuation (May 2022)

Concerning real-time actuation, a first workshop with stakeholders from the consortium took place in May 2022. The context and the results of the workshop will be described in the following paragraphs:

Workshop results

Participants of this discussion were developers – not necessarily directly involved in developing technology for RTA but also in general – infrastructure providers, an industry partner as well as general project managers responsible for integrating different work threads. Additionally, actors responsible for the ethical development of the project had a seat at the table. The discussion was moderated by a representative from work package 2.

A lot of input was gathered through the digital whiteboard where people could add post-its containing values or norms that matter to them in relation to RTA. Generally, the participants discussed about the concrete distinction between values and norms. An agreement was made that within the discussion, values are rather abstract and norms are more concrete and have a connection to the context of RTA in ASSISTANT. Also, the different values and norms that were discussed were identified to have interrelations and connections that are not always easy to grasp.

In general, the participants suggested values that are closely related to optimization and quality improvements for the industry partner but also values that are relevant for the well-being of the operators. So for example, transparency as a value was identified that matters both for the workers and the quality as far as a worker who is knowledgeable about his role and his contribution to the general production process is more productive.

Stability and **robustness** as values were mentioned in that spirit. In order for the relationship of the worker and the system, the participants agreed that *decisions and proposals should be stable and farsighted and not be changed (too) frequently*. Along those lines, the *decisions and behaviour should be continuous and consistent*. In the sense of the initial comment about quality improvement through worker well-being this again is visible here: *Nervousness of the worker through constant changes is to be avoided* to support these two dimensions.

Safety and **Security** were initially mentioned together but were then distinguished: Safety is mostly important in the sense of **Safety for the operator** while Security is understood in the sense of **Cybersecurity.** This suggests that the general values safety and security can be specified as more concrete values that are not necessarily a norm directly. Going into more depth, participants agreed that for safety, the *operators' safety should always be the number 1 priority* and the system should *increase ergonomics for the worker*. For Cybersecurity, however, it was said that *personal data should be stored safely*.

Transparency was one of the most discussed values within the group. This was due to the different understandings and the links to other values or norms. On the one hand, transparency was discussed in the context of security but also in relation to interactivity. The system is supposed to be *interactive and allow the users to bring in feedback*. Transparency was identified as a precondition therefore. Additionally, *decisions of the system should be transparent and certain data should be presented together with the decision*. As there were different understandings of transparency, the group agreed that the different definitions have to be reflected.

Accessibility in the sense of usability was also mentioned as value. The discussion evolved around the *explicitness of information modelling* to allow users to understand what is going on so they are able to interact meaningful with the system. At the same time, also *non-experts should be able to interact with the system*. Connected with accessibility was the idea that the system should be both *easy to use* as well as flexible in the sense that it should be *capable of performing different tasks*. It was acknowledged, however, that even though this general flexibility is to be achieved, that the system should *reflect limits to where the tool can be used*.

Another discussion thread was evolving around values of **quality** and **efficiency**. The overall goal of the system was aspired to increase productivity. Quality was both reflecting the quality of the products that are to be built as well as data quality and therefore the quality of predictions of the machine learning components. Connected norms were the *reusability and maintainability of data, information and knowledge*.

The norm that *all workers should be treated equally* and *no single worker should be preferred in some way* were a result of the discussion of the value **equality**.

7.3 Questionnaire for Initial Reflections

Question 1: General Information About The WP/UC

Before starting to set up the initial structure for the "Ethical-by-design" architecture for ASSISTANT, we would like to get a better understanding of the proposed solutions and the engineering challenges in the work packages and use cases.

1.1 Digital twin solutions

How do you envision digital twin solutions for process planning, production planning or reconfigurable manufacturing?

Can you give us one or more examples of how the problems you plan to solve with the implemented digital twin solution are usually solved in process planning, production planning or reconfigurable manufacturing?

1.2 Existing digital twin solutions in your field

How does your envisioned solution differ from other digital twin solutions in your field? Can you give us one or more examples of existing solutions for digital twin solutions for process planning, production planning or reconfigurable manufacturing or for the use of a data fabric that you are aware of?

1.3 Data handling, instrumentation and integration

Solutions developed in ASSISTANT are supposed to be integrated and orchestrated using a shared data fabric. Can you give us one or more examples of how this integration and orchestration has been handled in previous or similar projects or do you know of good examples of such a shared data fabric?

Please indicate, which type of information and data you are going to process with AI components and your estimation about the sensitivity of that data.

1.4 Data and ML/AI methods and components

Do you already have preferences for certain AI/ML tools and methods or are you planning to reuse and/or adapt previously developed methods in ASSISTANT? Can you give us one or more examples of the use of these methods in process planning, production planning or reconfigurable manufacturing?

Can you give us a rough overview of the type of data you are using with these methods?

Please name the components that you are developing that have a relation to AI

Question 2: Areas affected and initial ethical reflections

To get us started with identifying the aspects of your work packages or use case that need reflection from a Ethics perspective, we would like to ask you to provide us with any thoughts that you have – if any – that could be relevant for your tasks within the following dimensions. We are looking for the components that need to be further examined and discussed in the ongoing process of the project, to ensure that they are designed and developed in an ethically responsible way.

The three dimensions stem from the Human Centered AI framework. You can read more about the framework here.

2.1 Accountability

Accountability refers to the requirement for the system to be able to explain and justify its decisions to users and other relevant actors. To ensure accountability, decisions should be derivable from, and explained by, the decision-making mechanisms used. It also requires that the moral values and societal norms that inform the purpose of the system as well as their operational interpretations have been elicited in an open way involving all stakeholders.

2.2 Responsibility

Responsibility refers to the role of people themselves in their relation to AI systems. As the chain of responsibility grows, means are needed to link the AI systems' decisions to their input data and to the actions of stakeholders involved in the system's decision. Responsibility is not just about making rules to govern intelligent machines; it is about the whole socio-technical system in which the system operates, and which encompasses people, machines and institutions.

2.3 Transparency

Transparency indicates the capability to describe, inspect and reproduce the mechanisms through which AI systems make decisions and learn to adapt to their environment, and the provenance and dynamics of the data that is used and created by the system. Moreover, trust in the system will improve if we can ensure openness of affairs in all that is related to the system. As such, transparency is also about being explicit and open about choices and decisions concerning data sources and development processes and stakeholders. Stakeholders should also be involved in decisions about all models that use human data or affect human beings or can have other morally significant impact.

Question 3: Further Resources / Examples

It is important for us, to take your experiences from various disciplines into account and to learn from (best) practices. If you know any resources from previous projects, use-cases or from your experience, we would like to ask you to send them to us. If you want to describe context for the documents, please add the descriptions here.

The resources can include either architecture documents that reflect on ethical questions or articles from your disciplines that present approaches and experiences towards ethical engineering.

The resources can but do not necessarily have to be related to the *Human Centric Approach*.

Document	Filename	Comments

7.4 Component evaluation dimensions based on Trustworthy Guidelines

The following tables within this section provide an overview of an initial cross-analysis performed by partners in the project based on the Trustworthy guidelines. In addition, these tables show the initial identification of potential risks that need consideration.

The nomenclature used in these tables refers to the trustworthy requirements of (1) Human Agency and oversight (Trust1), (2) Technical Robustness and Safety (Trust2), (3) Privacy and Data Governance (Trust3), (4) Transparency (Trust4), (5) Diversity, Non-Discrimination and Fairness (Trust5), (6) Societal and Environmental Well-Being (Trust6), (7) Accountability (Trust7).

Component Definition	Trust1	*Trust2*	Trust3*	Trust4*	Trust5*	Trust6*	Trust7*
Process Manager UI	x		x	x			x
Process Designer		x		x		x	x
Process Predictor		x		x			x
Process Optimiser		x		x		x	x

Table 6 Trustworthy Guidelines: requirements for Process Planning

Component Definition	Trust1*	Trust2*	Trust3*	Trust4*	Trust5*	Trust6*	Trust7*
Simulation			x		x	x	x
Production Planner		x		x			x
Model Acquisition for Scheduling		x		x			x
Scheduler Optimisation		x		x		x	x
Production Manager UI	x		x	x			x

Table 7 Trustworthy Guidelines: requirements for Production Planning

Component Definition	Trust1*	Trust2*	Trust3*	Trust4*	Trust5*	Trust6*	Trust7*
Streamhandler	x		x				x
Execution Control and Reconfiguration		x		x		x	x
Digital Twin for Execution		x					x
Human Body Detection and Human Task Prediction		x	x	x			x
Human Side Interfaces	x		x				

Table 8 Trustworthy Guidelines: Real-Time Control and its components

Component Definition	Trust1*	Trust2*	Trust3*	Trust4*	Trust5*	Trust6*	Trust7*
Data Fabric		x	Optional*	Optional*		x	Optional*

Table 9 Trustworthy Guidelines: data fabric

*The Data fabric will require to be considered under transparency and accountability if any process of data modification by an AI component is embedded within it. Furthermore, privacy principles should be evaluated if sensitive information is kept within the data fabric or if any information can be linked to private information. In the opposite case, the data fabric should only focus on technical robustness and safety perspectives.

7.5 Component evaluation questions based on ALTAI Framework

Process Planning Feedback
General: 1. If personal data will be manipulated, are these processes aligned to a standard (IEEE, ISO)?. 2. What other stages could be used for data input/output (apart from the UI and data fabric) that could cause security concerns?. 3. What components would process or describe transparency results (i.e. users will be able to check explainability, open communication, and set/see traceability throughout the UI interface)?. 4. Have been established the ***responsibilities*** of the AI components (WP3-WP6) and the interactions involved (i.e. AI results) in architectural design ASSISTANT during its deployment and development? 5. Have been defined a methodology in which users can provide feedback (and tagging) from biased or risk information? 6. Are process/risk KPIs expected to be estimated online and provided to the users? 7. Are process/risks KPIs (if implemented) be formated with tagged information and kept easily accessible and secure (e.g. data fabric)? Process Manager UI: 1. Are end-users made adequately aware that a decision, content, advice, or outcome result from an algorithmic decision (especially important if these decisions, content, advice or outcomes are indirectly provided to end-users – e.g. shop floor workers)? 2. Are the end-users informed that they are interacting with an AI system? 3. Did you put in place procedures to avoid that AI system end-users over-rely on the AI system? 4. Would it require users specific training on how to exercise oversight? If so, what protocols would be used for users to fulfil this training? 5. Have mechanisms been established to deal with privacy or data protection through the different communication channels (UI and data fabric)? 6. Would any personal information be required to manipulate the process manager UI (this implies if GDPR conditions have been set at each stage in case of required personal data)?. 7. Is the user/stakeholders informed of the accuracy of the results? Process Predictor: 1. Has the considerations of tagging – monitoring –documenting the system accuracy (e,g +how good are the predictions based on training /validation sets)? 2. It has been considered saving prediction conditions to avoid reprocessing and, therefore, using energy-consuming optimisation processes? Process Predictor and Process Optimiser: 1. What are the expected results and measures are taken if the AI component fails in its execution? 2. Can adversary results produce considerable damaging consequences (safety, economy, security) to your components or other components of ASSISTANT? 3. What implicates would have a failing component in other components that are dependent on it, including external ones (e.g. if process predictor fails, how will the process optimiser effect)? 4. How is it secured that data used for development, training, and estimations fulfil quality requirements established for each component involved?

5. Has placed the consideration of tagging and documenting the outputs (and its data used for generating those results) and keep them for accountability purposes
6. Does exist a complete fall-back plan in case of irreversible situations for running the system?
7. Are considered to save optimisation results to avoid re-running conditions that are not necessary to be re-evaluated? .

Table 10 Considerations for the Process Planning

Process Planning Feedback
General: 1. If personal data will be manipulated, are these processes aligned to a standard (IEEE, ISO, GDPR)? 2. What other stages could be used for data input/output (apart from the UI and data fabric) that could cause security concerns?. 3. What are the components that would process or describe transparency results (i.e. users will be able to check explainability, open communication, and set/see traceability throughout the UI interface)?. In other words, what component will give the users an explanation of what was developed by the global component and explain how the final result was obtained.? 4. Have been considered generating an individual component for developing transparency that will include analyses in terms of quality and, at the same time, forward this set of cases/results to the data fabric (as an additional tag and result)? 5. Have been established the responsibilities (and under what circumstances) of the components, users, and developers in case they could produce any harm or considerable economical impact (i.e. the system fails)? 6. Have been considered who will be responsible for a failure condition on data transfer. For example, who will be responsible for the receiver of the information or the sender (this analysis should consider users responsibility too).? 7. Have been considered to save simulation/optimisation processes and tag them to avoid reprocessing conditions and avoid using energy-intensive tasks (i.e. tag results in callback results if they already exist with the same sets)? 8. Have been established the ***responsibilities*** of the AI components (WP3-WP6) and the interactions involved (i.e. AI results) in architectural design ASSISTANT during its deployment and development? 9. Have been placed procedures to avoid that end-users over-rely on the AI system? 10. Would it require users specific training on how to exercise oversight? If so, what protocols would be used for users to fulfil this training? Production Manager UI: 1. Are end-users made adequately aware that a decision, content, advice, or outcome result from an algorithmic decision? 2. Are the end-users informed that they are interacting with an AI system? 3. Was it place procedures to avoid that end-users AI system over-rely on the AI system? 4. If humans would participate in the decision making (and affecting WP5 or other processes), would it be required specific training on how to exercise oversight? 5. Have been established mechanisms to deal with privacy or data protection of data through the UI?. For example, would any personal information be required to manipulate the production manager UI for each development, testing, and deployment?. 6. Are the users informed of the system accuracy (from the components that predict or pass through a training process/parameter process)?

Simulation:

1. Have been considered that If simulations are describing/manipulating the same content of production planner and production scheduler information/data, the same considerations applied to them regarding data managing should be applied for the simulation.?
2. Have been considered that If training data are generated by simulation, each AI component results should be tagged with this consideration?
3. Has it been considered that if data generated by simulation will be used, the system users are aware of these considerations?
4. Can simulations be used as a source of explainability for optimal and non-optimal solutions?

Production Planner

1. What are the expected results and measures are taken if the AI component fails in its execution?
2. What implicates would have a component fail in another component that is dependant on it (if process predictor fails, how will be affected the process optimiser)?
3. How is it secured that data used for development, training, and estimations fulfilled the requirements of each component involved?
4. What would be the impact if the overall system crash (i.e. what alternatives are recommended to be used on these cases - Fall-back plan)?
5. Are production planner users (e.g. shop-floor users) aware that an algorithm has generated the production plan?
6. Can users override the main result and modify the plan (would this affect the ERP system directly or managed indirectly)?

Model Acquisition for Scheduling

1. Is it consider to put in place a methodology (algorithm, validation process, etc.) to establish that the models created are not accurate (e.g. place a tolerance and tag the accuracy of the models), especially if users wrongly set tables in the system?.
2. Is the previously defined validation error consider always to be reported to make sure the user does not over-rely on the system?.
3. Is it possible at any stage to modify the values of the tables from the users and, therefore, affect data integrity?
4. Have been considered the impact of allowing to use inaccurate models by the users?

Scheduler Optimisation

1. Have been considered that if reprocessing (run optimisation again) is not under conditions that would change results (e.g. stochastic optimisation), and the algorithms used for optimisation are exact (i.e. non-metaheuristics), the reported solutions would be the same as previous ones (i.e. avoid re-running unnecessary processes)?.
2. Have been analysed under conditions in data will make the optimisation process unable to be run?.
3. Are the users aware of the conditions in data that will make the optimisation process unable to be run?.

Table 11 Considerations for the Process Planning

Real-time control and actuation
General & stream handler: 1. Since the stream handler manages all types of information related to control, is there information that requires GDPR considerations at any stage?. 2. Are these processes (1) aligned to a standard (IEEE, ISO, GDPR)? 3. Does the responsibility and accountability to access such information will be lay on WP5 components or others?. 4. Does the digital twin of execution specifies anonymity for its component (clarification should be made for each component to secure privacy and data governance - i.e. make the WP5 component independent of such type of information)?. 5. Are defined how frequency or tolerances of error acceptable for defining training requirements, which will lead to review technical robustness and safety of the system? 6. Have been established the responsibilities of the AI components (WP3-WP6) and the interactions involved (i.e. AI results) in architectural design ASSISTANT during its deployment and development? Execution Controller: 1. Since The process Orchestrator uses information fed back from WP3 and WP5 through the data fabric, is there any instance in which the input information is validated? 2. Have been defined and specified conditions in which AI-based control systems should be overridden?. Digital Twin of Execution: 1. Has it been considered metrics that will be used to measure and evaluate the system performance and, at the same time, provide the user with dynamic information to check system reliability?. 2. Under what circumstances the digital twin should not be used, and how would the system recognise these conditions? 3. Are the user provided with the information defined in (2)? 4. Until what point is the digital twin model accountable in case of an error that would produce loss or harm? 5. Are determinates the risks involved in the planner to perform or estimate incorrect trajectories?. 6. Are implemented andy metric that defines the accuracy of the representation or models to the actual scenario?. If so, are these given to the users? 7. Were there procedures to avoid end-users over-rely on the AI system been considered in the architecture (consider numeral one and other methods)? Human Body Detection and Human Task Prediction: 1. Can the AI be combined with other sources of information to recognise specific users and at the same time, is the user fully aware of the use of this information? For example, does it involve only human recognition for safety and processing information?. 2. Has it been defined and provided to the user the information that will be handled (together with its purpose)?.

3. Has it been considered that If wearable connected through human body detection should focus on body detection for safety considerations or human-machine behaviour analyses only, or other purpose specified and informed to the user?.
4. Has it been considered anonymity to avoid track users outside the duties involved on the workstation?.
5. Is there any potential form of attack to which the AI system could be vulnerable and in the long term produce harm (these include data poisoning, model evasion, model inversion, or misuse by the user)?.
6. Has it been considered the use of risk metrics and risk levels specific for the use cases?.

Smart Human Interfaces:

1. Can wearables information be linked to a user to be tracked and check their behaviour on the shop-floor (i.e. other than safety considerations or that clearly specified to the user)?
2. Are users provided with information regarding the possible threats to the AI system before their use (design faults, technical faults, environmental threats)?.

Table 12 Considerations for the Real-Time Control

Data Fabric
General: 1. Is there any process involved in data curation imposed over the data fabric? If so, are those processes AI-based or methodologies that can easily be linked to well-known methodologies? 2. Is there any standard that will be followed for data security and data managing? 3. What are the main processes for backup plans for collected/created information in case of error with the system that causes loss of information (from an architectural point of view)?. 4. Have been established the ***responsibilities*** of the AI assets (WP3-WP6) and their interactions involved (i.e. AI results)? What responsibilities are involved in WP6 in regards to their results and managing?

Table 13 Considerations for the data fabric

7.6 ASSISTANT Component Use Case Definitions

As already mentioned above in the document, this appendix also contains two sections from the technical architecture document from task 2.2. We provide that content because of two reasons: First, the content is relevant to understanding the project and the concrete application that will be build within ASSISTANT. Secondly, this allows us to also document progress of the integration of the technical and the human centric architecture along the way, as both of them are living documents. The publication of sections of the technical architecture in task 2.2 freezes the current status and allows us to reflect on future changes. You can find those sections in the next pages.

7.6.1 Process Planning

7.6.1.1 Process Manager UI

Name	Process Manager
Brief Description	The process manager as a user interface supports users in generating efficient and effective decisions by applying the 6.
Involved components	The data fabric, the process designer, the process predictor and the process optimizer are involved components that interact with the process designer.
Pre-Conditions	Product and production system are required as input.
Basic Flow of Events	It is a user-specific interface, which, depending on the user's role, enables the control of the digital twin and provides visualisation of process planning artefacts like the resulting process plans, requirements, and skills. In addition, a chatbot supports, for example, process planners by answering their questions. //bullets for process planner, first time building the plan – Login – execute designer and optimizer (see below) – visualize input (data fabric), output of modules stored – manual changes
Alternative flows	– instead of executing he can just look at exisitng ones and can compare
Subflows	-
Key-Scenarios	– when needed: changes, new products, variants, changes to productions, manual changes of the production system o haven szenarios or changes.
Post-Conditions	Store generated visualizations in the data fabric.
Special requirements	-
Relevant pilot case(s)	AC and PSA

7.6.1.2 Process Designer

Name	Process Designer
Brief Description	The process designer developed in T 3.3 takes the three-dimensional files of the product and of the production system with all its resources to build all possible process graphs.
Involved components	The data fabric and the process manager are involved components that interact with the process designer.
Pre-Conditions	The three-dimensional files of the product to be produced and the production system used as well as their characteristics are present in the data fabric.
Basic Flow of Events	The first process designer first analyzes the product and the production system. A precedence graph will result after product analysis. Using the analysis information, the designer compares the resulting product and process requirements with the skills of the production system to assign all possible resources to each assembly, logisics, or monitoring process within the so-called process graph.
Alternative flows	Not applicable
Subflows	-
Key-Scenarios	The user selects the input data as precondition and starts process designing via the process manager. Then the process designer itself automatically develops the task and production skill model as well as the process graphs. Those results can be viewed via the process manager.
Post-Conditions	The production skill model, the task model and the resulting process graph must be stored inside the data fabric.
Special requirements	-
Relevant pilot case(s)	AC and PSA

7.6.1.3 Process Predictor

Name	Process Predictor
Brief Description	For different production plans and technical changes, the *process predictor* enables the forecast of various KPIs regarding cost, time, and quality.
Involved components	The data fabric, process designer and the process manager are involved components that interact with the process designer.
Pre-Conditions	Trained decision tree, change or process plans are available.
Basic Flow of Events	For three-dimensional technical changes, the designer is executed first. Developed process plans, as well as resulting process plans from three-

	dimensional changes, are then the input of the predictor. For each process in the process plan, a decision tree trained with historical data then predicts the KPIs depending on the assigned resources, tasks, and parts. In the end, the KPI prediction of a process plan is added up using all forecasts on the process step level. Similarly, the impact of textual changes can be predicted. By using natural language processing, the predictor identifies change features of a documented product or production change and a decision tree predicts the KPI categories based on the impact of similar historical changes.
Alternative flows	-
Subflows	-
Key-Scenarios	The user selects the input data as precondition and starts process prediction via the process manager. Then the process designer itself automatically develops the task and production skill model as well as the process graphs. Those results can be viewed via the process manager.
Post-Conditions	Store predicted KPIs
Special requirements	-
Relevant pilot case(s)	AC and PSA

7.6.1.4 Process Optimizer

Name	
Brief Description	
Involved components	
Pre-Conditions	
Basic Flow of Events	
Alternative flows	-
Subflows	-
Key-Scenarios	
Post-Conditions	
Special requirements	-
Relevant pilot case(s)	AC and PSA

7.6.2 Production Planning and scheduling

7.6.2.1 Simulation

Name	Validate decision
Brief Description	Validate decision within the degrees of freedom of a certain use case scenario.
Involved components	Production Planner, Production Scheduler (Model Acquisition & Optimization), Production Manager UI, Data Fabric
Pre-Conditions	All required data is provided in the Data Fabric.
Basic Flow of Events	All alternative flows from below make sense, there is no basic flow.
Alternative flows	– Manual set up and execution: Production Manager UI —> sets up executes —> Simulation – Final validation of production plan: Production Planner —> provides decisions to be finally validated by —> Simulation – Iterative feedback from Simulation to Production Planner: Production Planner —> provides different choices all to be validated by —> Simulation – Generation of training data for model acquisition: Simulation —> is set up and executed several times to provide training data for —> Production Scheduler (Model Acquisition) – Final validation of production schedule : Production Scheduler (Optimization) —> provides decisions to be finally validated by —> Simulation – Iterative feedback from Simulation to Production Scheduler : Production Scheduler (Optimization) —> provides different choices all to be validated by —> Simulation
Subflows	
Key-Scenarios	– Validate an already given schedule regarding OTD, cost, etc. (SE Scenario Schedule Validation) – Validate different choices for release dates, shift models, prioritization logics, etc., regarding OTD, cost, etc. (SE Scenario Schedule Optimization) – Validate different choices for make-or-buy split regarding OTD, cost, etc. (SE Scenario Make-or-Buy Proposal)
Post-Conditions	All data calculated by the simulation needs to be stored in the Data Fabric.
Special requirements	-
Relevant pilot case(s)	SE, AC

7.6.2.2 Production Planner

Name	Capacity adjustment and requirement planning
Brief Description	Tools that automatically computes a production plan (quantity to produce per period, quantity to order, and capacity adjustment with overtime)
Involved components	Data fabric, domain model, scheduler, simulation
Pre-Conditions	Required data is available, the simulator is on and it create the required output.
Basic Flow of Events	(1) User connect to production manager UI. (2) Production planning interface send a request to domain model to update data (demand, machine and workers available) (3) Domain model get the data from the correct system (ERP, worker management, MES) (4) The user enter the targeted KPIs (minimize expected costs/ensure a service level of 95%/…) (5) Production planner find a production plan: 5.a Solver find production quantities and extra capacity required to meet the KPI targets 5.b The simulation validates the plan , in case of negative update capacity computation and resolve. 6. The production manager UI displays the plan, and the output of the latest simulation run.
Alternative flows	The loop in (5) stops after a predetermined iteration limit if no feasible plan is found.
Subflows	-
Key-Scenarios	The shopfloor manager wants to adjust it production capacity and place orders to suppliers based on the latest information on customer demand
Post-Conditions	All data calculated by the production planner needs to be stored in the Data Fabric.
Special requirements	-
Relevant pilot case(s)	SE, AC

7.6.2.3 Model Acquisition for scheduling

Name	
Brief Description	The component goal is to acquire a constraint model from a set of table with schedule data
Involved components	data fabric, scheduler optimization
Pre-Conditions	Data must be prepared in accordance with Task 4.3

Basic Flow of Events	1. The user manually launches Model Acquisition via Production Manager UI 2. The user selects one or more data tables from Data Fabric, that he wants to process 3. The user can specify, which columns will be outputs (i.e. can they be calculated from entries of other columns in data tables from the same or different rows) during the model acquisition 4. The user starts the process of Model Acquisition, which will result in a set of equations. 5. The user reviews the equations and, if needed, returns to the step 3, to select new set of output columns. When the user is satisfied with the result, the user proceeds to the step 6. 6. The user selects all or part of the equations and stores them into Data Fabric
Alternative flows	-
Subflows	-
Key-Scenarios	
Post-Conditions	All data created by the Model Acquisition needs to be stored in the Data Fabric.
Special requirements	
Relevant pilot case(s)	SE, AC

7.6.2.4 Scheduler's optimization

Name	
Brief Description	The component goal is to optimize a constraint model to create new schedule tables
Involved components	data fabric, simulation, production planner, model acquisition
Pre-Conditions	Data must be prepared in accordance with Task 4.3
Basic Flow of Events	1. The user manually launches Scheduler's Optimization via Production Manager UI 2. The user selects one of the models stored in Data Fabric 3. The user selects an optimization criteria 4. The user starts the process of schedule optimization. 5. The user reviews the results and, if needed, returns to the step 3, to select new optimization criteria. When the user is satisfied with the result, the user proceeds to the step 6. 6. The user stores the selected schedule table or tables into Data Fabric
Alternative flows	In the step 6, the user can use the selected schedule table for the simulation

Subflows	Step 5 should allow for comparisons between two or more schedule tables, that are results from different constraint models or the same model but with different optimization criteria
Key-Scenarios	
Post-Conditions	All data created by the Model Acquisition needs to be stored in the Data Fabric. Schedule table must be created in accordance with Task 4.3
Special requirements	
Relevant pilot case(s)	SE, AC

7.6.2.5 Production Manager UI

Name	
Brief Description	The user interface of the production manager will present the front part of the work-package 4.
Involved components	Production planning tool, model acquisition (scheduling) tool and simulation tool
Pre-Conditions	
Basic Flow of Events	(1) user authentication (2) User input data (keyboard, files), and store in the data fabric (3) Trigger the tool (simulation/planner/scheduler), the tool store the result in the data fabric. (4) Read the result from the data fabric and display.
Alternative flows	-
Subflows	-
Key-Scenarios	The production manager will be able to launch the simulation, planning and production scheduling tools (model acquisition) as well as to visualize their outputs. In this interface, it will also be possible to view the different KPIs with the possibility of adjusting them.
Post-Conditions	
Special requirements	
Relevant pilot case(s)	SE, AC

7.6.3 Real-time control and actuation

7.6.3.1 Streamhandler

Name	Streamhandler
Brief Description	A publish/subscribe infrastructure based on Apache Kafka
Involved components	Components requiring real time access to shopfloor data, Components that produce high volume/velocity shopfloor data.
Pre-Conditions	A topic has been setup for producers to produce messages and consumers to consume the arriving messages
Basic Flow of Events	Consumer registers to the predefined topic Producer sends a message to the predefined topic Consumer is notified and provided the new message Consumer works on the message and notifies infrastructure of the work completion.
Alternative flows	-
Subflows	-
Key-Scenarios	Gathering shopfloor data
Post-Conditions	
Special requirements	
Relevant pilot case(s)	AC, STELLANTIS

7.6.3.2 Process Orchestrator

Name	
Brief Description	This component is responsible for feeding the digital twin for the task at hand.
Involved components	Digital Twin of Execution, Process Designer, Production Planner, Data Fabric, Quality Controller
Pre-Conditions	
Basic Flow of Events	1. Retrieves the product/process/resource assignment from Production planner 2. Retrieves the production process to be executed in the production line from Process Planner 3. The Quality control module will monitor production and provide feedback to the Process Orchestrator regarding the process in a closed-loop manner 4. Triggers the production digital twin to perform the related process

Alternative flows	-
Subflows	-
Key-Scenarios	AC, STELLANTIS
Post-Conditions	
Special requirements	
Relevant pilot case(s)	

7.6.3.3 Quality Controller

Name	
Brief Description	This component monitors the production and provides feedback on the Process Orchestrator regarding the process in a closed-loop manner.
Involved components	Process Orchestrator, Digital Twin of Execution
Pre-Conditions	Quality controller must always be aware of production's current situation
Basic Flow of Events	-
Alternative flows	-
Subflows	-
Key-Scenarios	
Post-Conditions	
Special requirements	-
Relevant pilot case(s)	AC, STELLANTIS

7.6.3.4 Digital Twin of Execution

Name	
Brief Description	The DTE is responsible for providing information regarding the current state of the production.
Involved components	Process Orchestrator, StreamHandler, Data Fabric, Quality Controller
Pre-Conditions	
Basic Flow of Events	Process orchestrator: ● Triggers the cell's DTE to perform relative process.

	• Connection with WP4: retrieves the product/process/resource assignment. • Connection with WP3: retrieves the production process to be executed in the production line. Streamhandler: • Collection of data from shopfloor and store them in Data Fabric. • Real time monitoring Data Fabric: • Direct communication to the data storage. • Integration with tools through the exposure of domain models. Quality Controller: • Provision of feedback to the Process Orchestrator
Alternative flows	1. Human Body Detection (HBD) gives input to the DTE and the Human Task Prediction (HTP) about the position of the operator. 2. HTP informs the DTE about the operator's tasks that are executed. 3. DTE informs Process Orchestrator. 4. DTE informs the operator through human side interfaces
Subflows	Gather human side information (HBD, HTP): • Gathering of sensor data • Data reasoning • Extraction of human state (position, current task execution) Provide information to human: • Process Orchestrator provides information to DTE about task execution • Transfer information to human side interfaces • Visualize current state Gather Robot information: • Robot controller sends Robot Status to DTE • Execution feedback is transferred to Execution Controller Provide Robot information: • Execution Controller sends task • Task splits into *resource, action, and part* • Task resolves to machine command • Command is sent for execution
Key-Scenarios	
Post-Conditions	
Special requirements	
Relevant pilot case(s)	STELLANTIS, AC

7.6.4 Secure and intelligence data fabric

7.6.4.1 Data Fabric

Name	Data Fabric
Brief Description	The ASSISTANT data fabric is a data management system that provides a unified interface to data access and storage, and abstracts the resource management details of data provisioning
Involved components	The data fabric is implemented in a layered architecture realized as a set of distributed services. The data fabric builds on widely available and technology neutral tools such as JSON-based REST services and integrates with (but does not depend on) other components of the ASSISTANT architecture.
Pre-Conditions	Data fabric services are deployed on infrastructure resources, components have access to the data fabric services via networks and APIs. Clients of data fabric services (human end-users and software tools) are authenticated using a shared security infrastructure.
Basic Flow of Events	1) clients are authenticated in the security infrastructure 2) clients initiate and drive interactions with data fabric services via the ASSISTANT domain models and / or the data fabric APIs and service interfaces 3) data is stored and (optionally) processed in the data fabric services, potentially resulting in multiple new data sets 4) metadata for all new data is generated and published by the data fabric services to facilitate usage of data 5) clients make use of data fabric search and query services in conjunction with metadata to organize, identify, access, retrieve, and use data outside the data fabric (likely through the domain models)
Alternative flows	similar to the basic flow, but time series data is routed to the system via the Intrasoft StreamHandler
Subflows	-
Key-Scenarios	A (digital twin) tool defines a domain model for its data and uses this and the data fabric to abstract data management and provisioning
Post-Conditions	Data is available and persistently stored / archived in the data fabric
Special requirements	-
Relevant pilot case(s)	SE, AC, PSA

7.7 ASSISTANT Component Interface Requirements

7.7.1 Process Planning

7.7.1.1 Process Designer

7.7.1.1.1 Overall description

The *process designer* developed in T 3.3 takes the three-dimensional files of the product and of the production system with all its resources to build all possible process graphs. These process graphs consider all ways to produce the product with the given production system. To do so, submodules first analyze the product and the production system. A precedence graph will result after product information. Using the analysis information, the designer compares the resulting product and process requirements with the skills of the production system to assign all possible resources to each production or monitoring process within the so-called process graph. The digital twin for process planning must identify the processes necessary to produce a part, predict the process parameter of each process, select the optimal process plans by comparing different resource allocations and provide a suitable interaction platform for the various roles in production (e.g., process planners, production planners, developers, and operators).

7.7.1.1.2 External interface requirements

User interfaces	not applicable
Hardware interfaces	not applicable
Software interfaces	Data Fabric, Process Manager
Communications interfaces	The process manager triggers the transfer JTS, JSON, XML

7.7.1.1.3 Performance requirements

The first outputs of the process designer are assembly, logistics, and monitoring processes, and dependent product and process requirements as well as capabilities of the production system, which are described in the so-called process graph. For this to be represented at different levels of granularity, the underlying **processes must be represented at a granular leve**l (cf. Defliverable 3.1 - 5.1.3). That includes representations of process levels considering sequences of activities needed to produce a product, operational levels, task levels, and functional levels. During process design, process planners generate and collect all necessary skills, processes, and their sequence or application to produce a product. The processes, requirements and skills are stored inside a process graph. The **process designer must generate reliable process graphs** realistically mapping this step. The mapping includes a process plan prediction. Cost, time, and quality parameter predictions must be enabled for each step in the process plan. These include, for example, product, production and reconfiguration costs. Those predictions have to depend on the detailed threefold structure: the process to be executed, the product to be produced, and the resource to be used. Additionally the mapping includes a change prediction. Cost, time, and quality parameters should be predicted for textual and geometrical changes. Those predictions must depend on change characteristics. Specific performance indicators on computing time, required storage space, capacities, response times, real-time capability, and others are to be detailed in the further course of the project.

7.7.1.1.4 Logical database requirements

Regarding the logical requirements for information to be stored in and provided by databases, the following can be said: The type of data and information used by several modules in WP3 are among others CAD files of the produced products, CAD files of the used production systems, and MES data. This data is generated and uploaded into the database by different users and has to be accessible to several parties accordingly. The frequency of use can vary from several requests a day to only a few requests per year. Data entities and their relationships heavily depend on the existing data structure in the respective companies and the specific ways CAD and MES data is stored. Integrity constraints are limited to the requirement that the CAD file used as an input for the process designer are up to date and correct, in terms of being a physically correct model of the real product/process to be analysed. As of now, no specific data retention requirements could be identified for the process planner.

7.7.1.2 Process Predictor

7.7.1.2.1 Overall description

For different production plans and technical changes, the *process predictor* enables the forecast of various KPIs of the magical triangle, including cost, time, and quality. For three-dimensional technical changes, the designer is executed first. Developed process plans, as well as resulting process plans from three-dimensional changes, are then the input of the predictor. For each process in the process plan, a decision tree trained with historical data then predicts the KPIs depending on the assigned resources, tasks, and parts. In the end, the KPI prediction of a process plan is added up using all forecasts on the process step level. Similarly, the impact of textual changes can be predicted. By using natural language processing, the predictor identifies change features of a documented product or production change and a decision tree predicts the KPI categories based on the impact of similar historical changes.

7.7.1.2.2 External interface requirements

User interfaces	Not applicable
Hardware interfaces	Not applicable
Software interfaces	Process Designer, Data Fabric, and Process Manager
Communications interfaces	JTS and JSON

7.7.1.2.3 Performance requirements

The process optimizer must enable predictions of various parameters of processes since they influence the optimal process plan choice. This includes KPIs regarding cost, time, and quality, that should be predicted for each process step. Those are the base to select an optimal process plan. The representation of the process parameter prediction can be specified as follows. The predictor must map process plan predictions. Cost, time, and quality parameter predictions must be enabled for each step in the process plan. These include, for example, product, production, and reconfiguration costs. Those predictions have to depend on the detailed threefold structure: the process to be executed, the product to be produced, and the resource to be used. Furthermore, the DTPP must map change predictions. Cost, time, and quality parameters should be predicted for textual and geometrical changes depending on change characteristics. Specific performance indicators on computing time, required storage space, capacities, response times, real-time capability, and others are to be detailed in the further course of the project.

7.7.1.2.4 Logical database requirements

Regarding the logical database requirements of the predictor, the same requirements as for the process designer (cf. 3.1.1.4) account. Additionally, it can be emphasized that especially JTS, JSON files, and equivalents will serve as main input and output data.

7.7.1.3 Process Optimizer

7.7.1.3.1 Overall description

The *process optimizer* selects the optimal process planning by evaluating the value of an objective function. The submodules generate possible process plans, add secondary tasks, evaluate the objective function using the predictor and validate the process plans via simulations. During Optimization, the submodules are iteratively executed to derive the process plan with the highest objective value.

7.7.1.3.2 External interface requirements

User interfaces	Not applicable
Hardware interfaces	Not applicable
Software interfaces	Process Predictor, Data Fabric, and Process Manager
Communications interfaces	JSON and XML

7.7.1.3.3 Performance requirements

The process optimizer needs to support selecting the optimal process plan by choosing a process plan with optimal parameters. To enable a realistic representation of process optimization, the following requirements must be met. Firstly, the optimizer must **map the users KPI value within the optimization**. The optimal process plan must satisfy the user in terms of boundary clarified by him/her for KPIs. If a user sets a KPI acceptance range/limit, the optimal process plan should respect the proposed limit from the user. The tool should inform the user if no process plan matches his expectations. Secondly**, the optimizer must facilitate the automated generation of possible process plans**. The results of the optimizers must therfore allow for automatically suggesting a (scalable) process plan The optimizer must include approaches to automatically suggest possible resource assignments to the processes and must include resulting transportation processes. The suggested plan must respect the constraints/targets set by the user. Thirdly, **the optimizer's results must facilitate suggesting robust process plans** and include the best alternative process plans for probable machine breakdowns and operator illness. That is process plans that remain valid when the parameters change. This must be enabled by predicting the KPIs. The tool for automatic process plan generation must account for all flexibility of the line. This adjustment allows spreading the load along the line to react to possible parameter variation like processing time or machine failure, and to maintain performance. Specific performance indicators on computing time, required storage space, capacities, response times, real-time capability, and others are also for this module to be detailed in the further course of the project.

7.7.1.3.4 Logical database requirements

Regarding the logical database requirements of the predictor, the same requirements as for the process designer (cf. 3.1.1.4) account. Additionally, it can be emphasized that especially JTS, JSON files, and equivalents will serve as main input and output data.

7.7.1.4 Process Manager

7.7.1.4.1 Overall description

The *process manager* supports users in generating efficient and effective decisions by applying the DTPP. It is a user-specific interface, which, depending on the user's role, enables the control of the digital twin and provides visualisation of process planning artefacts like the resulting process plans, requirements, and skills. In addition, a chatbot supports, for example, process planners by answering their questions.

7.7.1.4.2 External interface requirements

User interfaces	Mouse click or textual user intend within the process manager
Hardware interfaces	Not applicable
Software interfaces	– data fabric: itself here retrieval of production and product data – process designer: provided by process predictor, to call module and provide location of data – process predictor: to call module and provide location of data – process optimizer: to call module and provide location of data
Communications interfaces	Python or C#-based communication interfaces

7.7.1.4.3 Performance requirements

The process manager has to support users in controlling the process planning. The applicable requirements in this regard are as follows. Firstly, **the tool must enable the timely execution of process planning** allowing the execution of all process planning phases described in Deliverable 3.1 (WP3). Secondly, **the tool must support the user by analysing the process planning itself**. The analysis includes the visual and textual evaluation or description of the inputs, outputs, and the individual process planning steps. Thirdly, **the tool must allow user-**

specific permission rights. Constructors, process planners, production planners and operators should have different rights to execute and analyse the process planning. Accordingly, the tool has to automatically check and grant or deny access to the analysis or execution. Specific performance indicators on computing time, required storage space, capacities, response times, real-time capability, and others are again to be detailed in the further course of the project.

7.7.1.4.4 Logical database requirements

Regarding the logical database requirements of the predictor, the same requirements as for the process designer (cf. 3.1.1.4) account. Additionally, it can be emphasized that especially Windows Forms, APIs and JSON files (or equivalent) will serve as main input and output data formats.

7.7.2 Production Planning and scheduling

7.7.2.1 Simulation

7.7.2.1.1 Overall description

Simulation allows to validate decisions within the degrees of freedom of a certain use case scenario, e.g.

- Validate an already given schedule regarding OTD, cost, etc. (SE Scenario Schedule Validation)
- Validate different choices for release dates, shift models, prioritization logics, etc., regarding OTD, cost, etc. (SE Scenario Schedule Optimization)
- Validate different choice for make-or-buy split regarding OTD, cost, etc. (SE Scenario Make-or-Buy Proposal)

To this purpose, this component will calculate a material flow simulation with all the required data input and providing all calculated data output. A major benefit is the detailed view on the production flow that allows to agree on the "best" decisions from a business target perspective. The time and effort to create simulation runs are relevant objectives. The goal is to reduce this effort significantly by automating this process and by finding the most relevant simulation experiments.

7.7.2.1.2 External interface requirements

User interfaces	– User can set up (load and edit) and execute manual simulation runs. – User can view simulation results. – User can compare different simulation runs.
Hardware interfaces	-
Software interfaces	Production planner Model Acquisition and Optimization for scheduler Data Fabric External tools: – Simulation tools (e.g. Tecnomatix Plant Simulation)

	– python libraries (e.g. for creating lightweight simulations to be integrated in an easier way than commercial tools)
Communications interfaces	Interface between simulation and production planner : – Production planner triggers simulation, signaling the simulation's input data is available in the data fabric – Simulator triggers the production planner that the simulation's output data is available in data fabric. Communicate by read/write data from the data fabric. Communication with Production Manager UI (for manual triggering) and scheduler are similar to production planner

7.7.2.1.3 Performance requirements

Multi-user is out-of-scope. Multi-core (4 cores) should be considered.
Amount of information to be handled and response duration in time depend strongly on the industrial use case and will be specified later.

7.7.2.1.4 Logical database requirements

Cannot be (fore)seen yet. Need an integrated view from all components.
Data entities and their relationships will be specified in Task 4.3.
Regular data management (read/write/update).

7.7.2.2 Production Planner

7.7.2.2.1 Overall description

Tool for production planning, including capacity and requirement planning. Given input data from the domain model, and KPI targets inputted by the user through the production planner interface, the tool automatically suggest decisions related with the adjustment (shift length) of the production capacity, subcontracting, and orders to place to suppliers. See D4.1 for more information.

7.7.2.2.2 External interface requirements

User interfaces	Production planner will not provide any graphical interface, but it will interact with the production planner UI. The production planner UI allows: – The user to control the outputted plan by entering targeted KPI values, or by giving a priority to the objectives. – The user to visualize/compare different production plans.

	– The tool should show the impact of uncertain date (worst case scenario, average,….)
Hardware interfaces	-
Software interfaces	**<u>Interface with the material flow simulation:</u>** The objective is to validate/evaluate the production plan. The precise information transferred between the two software may change during the project, but a first version is given below. The information toward the simulation includes the production load per process step and per period, the extra work required, the quantity of outsources production. The message return by the simulation gives the end date of each production load. **<u>Interface with Production Manager UI:</u>** Production planner -> production manager UI: send the production plan, and this include the matrix of: – Production quantity for each item in each period – Inventory Level for each item in each period – Number of overtime required per resource – Purchase quantity for each component/subcontracted item. **<u>Interface with Production manager UI :</u>** – Target values for KPIs – ordering of the KPIs **<u>Interface with the domain model</u>** to get the input data and store the output data.
Communications interfaces	**<u>Communication with data fabric:</u>** Communicate by REST services provided by the datafabric to get each input data required for production planning (resources with capacity, Flexible BOM, capacity consumption per operation on each resource, costs, targeted KPIs values, ordering of the KPIs, ...) Communicate by REST services provided by the datafabric to store each output data (production quantity per period, inventory level per item and period, quantity ordered to suppliers/subcontractors) **<u>Communication with production manager UI:</u>** Production planner will provide a rest service for the production manager to request a production run. (1) Production Manager UI store the targeted KPI values and the ordering of the KPIs to the data fabric. (2) Production Manager requests a run from production planner. (3) Production planner computes the production plan and it stores it posts the results in the data fabric. (4) Production planner respond to the request to inform computation are done. **<u>Communication with simulation:</u>**

	Simulation provides a rest service for the production manager to request a simulation run. (1) Production planner post the production quantity per period in the data fabric, as well as the parameters of the simulation run. (2) Production planner requests a simulation run. (3) Simulation get the simulation's input data from the data fabric (3) Simulation runs and post the simulation output in the data fabric. (4) Simulation responds to the request to inform computation are done.

7.7.2.2.3 Performance requirements

The tool should be able to handle 1/10 simultaneous users; The tool will consume a heavy load on the processor (ideally, 100% for several hours) as well as memory (several GO in RAM). It might handle some mega octet of data during communication with other softwares, and some logs maybe quiet heavy (close to 1 G0). The response time will be lower than few hours with updates every few minutes.

7.7.2.2.4 Logical database requirements

Regular data management (read/write/update).

7.7.2.3 Model Acquisition for scheduling

7.7.2.3.1 Overall description

Model Acquisition is aimed to obtain a constraint model from a set of tables with schedule data. Adherence to a precise structure of data is not required. The Model Acquisition will find relations between tables and functional relations beween columns within tables to use it to generate a constraint model. User can select output columns, i.e. the columns which values are the result of formula that takes inputs from some other column or columns.
The user can later review the obtained model and select or deselect constraints, change output columns and send the model to the scheduler's optimization, the simulation or the production planner.

7.7.2.3.2 External interface requirements

User interfaces	Graphic UI with the support of M/KB. User can select tables and input/output columns User can review and select acquired constraints User can run different acquired models and compare their results

Hardware interfaces	-
Software interfaces	Data Fabric, scheduler's optimization
Communications interfaces	Python or Java

7.7.2.3.3 Performance requirements

Single-user access. A computer with multi-core CPU and large amount of RAM is advised. Exact requirements could be specified later, when the scope of an industrial case is known

7.7.2.3.4 Logical database requirements

A requirement for an access to a relational database, specified in accordance with the Task 4.3

7.7.2.4 Scheduler / Optimization of scheduling

7.7.2.4.1 Overall description

Optimization of scheduling is aimed to take any model obtained by the Model acquisition and create a schedule table from it. The user can select one of the criteria and launch this tool. After the optmization process, the newly created schedule table can be tested in the Simulation or stored directly in the Data Fabric

7.7.2.4.2 External interface requirements

User interfaces	Graphic UI with the support of M/KB. User can run different acquired models and compare their results
Hardware interfaces	-
Software interfaces	Data Fabric, model acquisition, simulation
Communications interfaces	Python or Java

7.7.2.4.3 Performance requirements

Single-user access. A computer with multi-core CPU and large amount of RAM is advised. Exact requirements could be specified later, when the scope of an industrial case is known

7.7.2.4.4 Logical database requirements

A requirement for an access to a relational database, specified in accordance with the Task 4.3

7.7.2.5 Production Manager UI

7.7.2.5.1 Overall description

The production manager User Interface will give the user the possibility to communicate with the modules developed in WP4 while hiding the complexity of these modules. The following outputs from WP4 modules will be managed by the production manager UI:

Production planning

- Production plan and purchasing plan.
- Planned inventory/back-order level.
- Planned resource/capacity consumption.
- Target KPIs values
- Impact of uncertainty on production plan

Scheduling (Model acquisition)

- Model and schedule visualization
- Optimization cost and main KPIs
- Gantt chart (schedule) and visualisation of resource utilisation

Simulation

- Input load data (customer orders and due date)
- Initial factory state/condition
- Input planning/scheduling decisions
- Input simulation parameters
- Output production flow
- Main KPIs
- Output comparison of different simulation runs

Three levels architecture is needed:

1. display level: Display outputs, KPIs
2. Control level: Read data from (3) and transfer it to (1). This level will also send additional constraints to involved modules based on KPIs adjustement
3. Data level

7.7.2.5.2 External interface requirements

User interfaces	Thin client (web application)
Hardware interfaces	Not applicable
Software interfaces	Production planning, model acquisition (scheduling), simulation software modules (. exec if possible) -> WP4 backend
Communications interfaces	Json and Java script modules Communicate through REST Service to get the data from the data fabric (transform if needed) Use the REST service provided by production planner to trigger a production planning run. The production planning service will require authentication

7.7.2.5.3 Performance requirements

The following performances are expected:

- Adjustable KPIS: The user can specify the characteristic of the solution to create, i.e., providing upper or lower bound for the KPI, assigning weights to the optimization objectives, enter additional constraint on the plan/schedule, etc.
- Plot tables with several rows and columns based on the planning horizon and scheduling horizon.
- Plot textual form of the acquired model with link back to the data
- UML diagram in the model (if aplicable)

7.7.2.5.4 Logical database requirements

Not applicable, inputs to the production manager UI will be provided by Json format

7.7.3 Real-time control and actuation

7.7.3.1 Streamhandler

7.7.3.1.1 Overall description

Streamhandler is a high-performance distributed streaming platform for handling real-time data based on Apache Kafka.

7.7.3.1.2 External interface requirements

User interfaces	-
Hardware interfaces	-
Software interfaces	Kafka API
Communications interfaces	Kafka Consumer/Producer/Connectors. Connector bridges for various data sources.

7.7.3.1.3 Performance requirements

Up to 5 seconds from producing a message until message reaches consumer.

7.7.3.1.4 Logical database requirements

None

7.7.3.2 Process Orchestrator

7.7.3.2.1 Overall description

This component will serve as the entry point of the WP5 developments since it will be responsible for triggering the cell's digital twin to perform the related process. It will also be indirectly connected to the task scheduler/planner by retrieving the product/process/resource assignment and the Process Planner by retrieving the production process to be executed in the production line. The Quality control module will monitor production and provide feedback to the Process Orchestrator regarding the process in a closed-loop manner. Once a defect from the Quality Control module is identified or a problem during a process execution occurs, the Execution Controller in an online fashion will communicate the error to the Process Planner for evaluating and providing alternative process plans as well as to the Production Scheduler for adapting the schedule accordingly (i.e., continue with other products and/or resources).

7.7.3.2.2 External interface requirements

User interfaces	
Hardware interfaces	
Software interfaces	REST API
Communications interfaces	REST API

7.7.3.2.3 Performance requirements

7.7.3.2.4 Logical database requirements

7.7.3.3 Quality Controller

7.7.3.3.1 Overall description

The combination of historical and streamed statistical analyses and machine learning methods will use the information to evaluate and predict both the state of the process and the state of the products.

7.7.3.3.2 External interface requirements

User interfaces	-
Hardware interfaces	-
Software interfaces	Process orchestrator: ● Quality controller takes as input information regarding the state of the process and the state of the products. DTE: ● Exchange of data regarding the status of the production
Communications interfaces	

7.7.3.3.3 Performance requirements

Quality control is a component that is continuously running during the production and provides information regarding the state of the process and the products, which leads to the need of the real-time response time.

7.7.3.3.4 Logical database requirements

Sensor data, event data

7.7.3.4 Digital Twin of Execution

7.7.3.4.1 Overall description

The DTE will represent the whole workstation. It must interact directly with the real world and more specifically it takes as input information about the workstation area layout, the resources and the different parts that exist in the real world. The proposed DTE is hardware agnostic and could integrate seamlessly multiple robots and sensors.

7.7.3.4.2 External interface requirements

User interfaces	AR application, Web application
Hardware interfaces	Robots (ROS), Sensors (ROS), AR headset (Web socket), Android devices (Web socket)
Software interfaces	• Process Orchestrator TBD message format and content • StreamHandler TBD message format and content • Data Fabric TBD message format and content • Quality Controller TBD message format and content
Communications interfaces	ROS, bridges, REST, bridges for MQTT, bridges for OPCUA, PROFITNET

7.7.3.4.3 Performance requirements

UI response time: ranges between 1 to 3 seconds
HTP response time: real time
HBD response time: around 0.05 seconds

7.7.3.4.4 Logical database requirements

Sensor data, event data (streamhandler)

7.7.4 Secure and intelligence data fabric

7.7.4.1 Data Fabric

7.7.4.1.1 Overall description

7.7.4.1.2 External interface requirements

User interfaces	The data fabric does not provide end-user oriented GUIs but does integrate with other project data visualization tools
Hardware interfaces	The data fabric services are deployed as virtualized components (virtual machines and / or containers) and does not make use of platform-specific hardware features or interfaces
Software interfaces	The data fabric provides APIs for all services and integrates with the ASSISTANT domain model tools for abstraction
Communications interfaces	Data fabric services are accessible over networks using standardized service technologies, e.g., JSON-based REST services

7.7.4.1.3 Performance requirements

The data fabric services must operate within reasonable parameters for application-specific data acess and query response time, and requires suitable amounts of storage (memory and disk / persistent storage) and network capacity for caching, access, and transfer of data. The data fabric is designed for multi-tenant environments and concurrent access to the system services. As the data fabric services only provides limited capabilities for data processing within the data fabric itself, not hard requirements are placed on compute capacity.

7.7.4.1.4 Logical database requirements

The data fabric is designed to make use of and abstract multiple types of data stores, including databases. The fabric itself requires platform capabilities for storing and efficiently associating (text-based) metadata to store data. Typically the metadata will be several orders of magnitude smaller than the payload data (at least for large data sets) and as such the data fabric needs capabilities but not a lot of capacity for this.

7.8 Data Flows

In this chapter all the data flows between modules will be characterized.

7.8.1 Process Planning

7.8.1.1 Process Planner

MODULE NAME Data Fabric						
Input	**From**	**Where**	**What**	**When**	**How**	**Status**
CAD file of product and production system	Data fabric	?	CAD (step/object)	on demand (pull)	Server	concept
Product, process and resource information	Data fabric	?	JSON	on demand (pull)	Server	concept
Ontology	Data Fabric	?	turtle	on demand (pull)	Server	concept
Output	**To**	**Where**	**What**	**When**	**How**	**Status**
Process graphs (with and without resource allocation) and product graph	Data fabric	?	turtle (Ontology) JSON (Data fabric)	when available (push)	?	concept

7.8.1.2 Predictor

MODULE NAME Data Fabric						
Input	**From**	**Where**	**What**	**When**	**How**	**Status**
Process graphs (historical and to be evaluated)	Data fabric	?	JSON/turtle	on demand (pull)	?	concept
Historical KPIs	Data fabric		JSON/csv	on demand (pull)	?	concept
Output	**To**	**Where**	**What**	**When**	**How**	**Status**
Predicted KPI for unknown process graph	Data fabric		JSON/turtle/csv	when available (push)	?	concept

7.8.1.3 Process Optimizer

MODULE NAME Data Fabric						
Input	**From**	**Where**	**What**	**When**	**How**	**Status**
Predicted KPIs and process plans	Data fabric		JSON/turtle/csv	on demand (pull)	?	concept
Output	**To**	**Where**	**What**	**When**	**How**	**Status**
“Best” process plan for robust production planning	Data fabric		JSON/turtle	when available (push)	?	concept

7.8.1.4 Process Engineer UI

MODULE NAME Data Fabric						
Input	From	Where	What	When	How	Status
User						
Output	To	Where	What	When	How	Status
Vizualizatio n						

7.8.2 Production Planning and scheduling

7.8.2.1 Simulation

MODULE NAME Simulation						
Input	From	Where	What	When	How	Status
Production master data (products, processes, resources)	Data Fabric	?	?	on demand (pull)	?	concept
Production scenario data (order lists, resource availabilities, stock inventory)	Data Fabric	?	?	on demand (pull)	?	concept
Production plan	From Production Planner via Data Fabric	?	?	on demand (pull)	?	concept
Production schedule (initial, optimized)	From (Model Acquisition &) Optimisation	?	?	on demand (pull)	?	concept

Output	To	Where	What	When	How	Status
Production schedule (simulated)	Data Fabric	?	?	on demand (pull)	?	concept

7.8.2.2 Production Planner

MODULE NAME Production Planner						
Input	From	Where	What	When	How	Status
Production plan	Production plan->simulation	HTTP	JSON	when available	client	concept
Output	To	Where	What	When	How	Status
Feasibility: production capacity and lead time per lot	Simulation -> production planner	HTTP	JSON	when available	server	concept

7.8.2.3 Production Manager UI

MODULE NAME Production Manager UI						
Input	From	Where	What	When	How	Status
Production planning output data	From production planning module	HTTP	JSON	After runnning production planning module	Thin client	Concept
Model acquisition (scheduling) output data	Model acquisition (scheduling) module	HTTP	JSON	After running scheduling module	Thin client	Concept
Simulation	Simulation module	HTTP	JSON	After running simulation	Thin client	Concept

Output	To	Where	What	When	How	Status
Production planning output data	Thin client user interface	HTTP	-	After runnning production planning module and reading output data from this module	Thin client	Concept
Model acquisition (scheduling) output data	Thin client user interface	HTTP	-	After running scheduling module and reading output data from this module	Thin client	Concept
Simulation	Thin client user interface	HTTP	-	After running simulation and reading output data from this module	Thin client	Concept

7.8.3 Real-time control and actuation

7.8.3.1 Execution Controller

DTE						
Input	From	Where	What	When	How	Status
Facility Data	Data Fabric	Rest Api	Available resources	Upon initialization of execution	Client	Concept
Production Schedule	Data Fabric	Rest Api	The assignments of the resources	Upon initialization of execution	Client	Concept

Output	To	Where	What	When	How	Status
Visualization	End user	Web application	Information regarding task execution	Near-real time	Server	Concept

1.5.4.2 Digital Twin of Execution

DTE						
Input	From	Where	What	When	How	Status
Sensor Data	Hardware Devices (Sensors, Robots)	ROS network	Sensor data (over TCP/IP)	Periodic (according to sensor freq)	Client	Tested on simulation
Output	To	Where	What	When	How	Status
Digital status of the work cell	Robot controller, HTP, Execution Controller	ROS network, HTTP	ROS messages, JSON (over HTTP)	when available (push)	Server	implemented

7.8.3.2 AI for Fenceless Human-Robot collaboration

HBD						
Input	From	Where	What	When	How	Status
Sensor Data	Hardware Devices (Sensors, Robots)	ROS network	Sensor data (over TCP/IP)	Periodic (according to sensor freq)	Subscriber	Tested
Output	To	Where	What	When	How	Status

Human Body points, Human hand points	DTE	ROS network	ROS messages	periodic	Publisher	Tested

HTP						
Input	From	Where	What	When	How	Status
Human hand points, Detected Objects	DTE	ROS network	ROS messages	periodic	Subscriber	Tested
Output	To	Where	What	When	How	Status
Human current task	DTE, Execution Controller	ROS network, HTTP	ROS messages, JSON (over HTTP)	when available (push)	Publisher and Server	concept

Robot behavior controller						
Input	From	Where	What	When	How	Status
Task to be executed	Execution Controller	HTTP	JSON	on demand (pull)	Server	concept
Output	To	Where	What	When	How	Status
Task completion	Execution Controller, DTE	HTTP, ROS network	JSON	when available (push)	Server	Concept

7.8.4 Secure and intelligence data fabric

7.8.4.1 Data Fabric

MODULE NAME Data Fabric						
Input	From	Where	What	When	How	Status

data fabric services	ASSISTANT tools	REST/HTTPS	JSON, binary data, domain models	periodic, on demand	server	defined
Output	**To**	**Where**	**What**	**When**	**How**	**Status**
data fabric services	ASSISTANT tools	REST/HTTPS	JSON, binary data, domain models	on demand	server	defined

7.9 Data Management

7.9.1 Process Planning

[will be added in further revision of the living document]

7.9.1.1 Process Planner

7.9.1.2 Predictor

7.9.1.3 Process Optimizer

7.9.1.4 Process Engineer UI

7.9.2 Production Planning and scheduling

7.9.2.1 Simulation

[will be added in further revision of the living document]

Production Planner

7.9.2.1.1 Input

Module input												
Data	Producer	Input method	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency	Read/ write
Set of resources with capacity	-ERP /user/ MES/IOT (RFID)	Rest request to data fabric	Set of objects, with integer, strings	minutes	building the optimization model	The optimization component	Data fabric, and external tool accessible through domain model.	read the data through a REST service provided by the data fabric whenever computation starts	few octets	every months/year	The tool is supposed to be used every week. During usage, domain model may be queried a few times.	read
Inventory levels for end-items and components	-ERP MES/IOT (RFID)	Rest request to data fabric	Set of objects, with integer	units of item	building the optimization model	The optimization component	Data fabric, and external tool accessible through domain model.	read the data through a REST service provided by the data fabric whenever computation starts	few octets	every months/year	The tool is supposed to be used every week. During usage, domain model may be queried a few times.	read
-Flexible bill of material	-ERP /user	Rest request to data fabric	integers	units of item	building the optimization model	The optimization component	Data fabric, and external tool accessible through domain model.	read the data through a REST service provided by the data fabric whenever	few octets	every months/year	The tool is supposed to be used every week. During usage, domain model may	read

								computation starts			be queried a few times.	
Demand per period and per end-item (e.g., per week)	AI-Based probability learner	Rest request to data fabric	Object with double, integer, strings to describe the probability distributions	unit of item	building the optimization model	The optimization component	Data fabric	read the data through a REST service provided by the data fabric whenever computation starts	few octets	every day/week	A few times whenever the tool is used	read
Production yield per operation	AI-Based probability learner	Rest request to data fabric	Object with double, integer, strings to describe the probability distributions	%	building the optimization model	The optimization component	Data fabric	read the data through a REST service provided by the data fabric whenever computation starts	few octets	every day/week	A few times whenever the tool is used	read
Delivery lead times	AI-Based probability learner	Rest request to data fabric	Object with double, integer, strings to describe the probability distributions	period (week)	building the optimization model	The optimization component	Data fabric	read the data through a REST service provided by the data fabric whenever computation starts	few octets	every day/week	A few times whenever the tool is used	read
Capacity per resource and per period	AI-Based probability learner	Rest request to data fabric	Object with double, integer, strings to describe	minutes	building the optimization model	The optimization component	Data fabric	read the data through a REST service provided by the data fabric whenever	few octets	every second/minutes/hours	every second/minutes/hours	read

			the probability distributions e					computation starts				
Resource consumption per operation	Simulationer	Rest request to data fabric	integers	minutes	learning the optimization model	learning component	Data fabric	read the data through a REST service provided after simulation response	few octets	every second/minutes/hours	every second/minutes/hours	read
Production lead times per lot	Simulation	Rest request to data fabric	integers	period (week)	learning the optimization model	learning component	Data fabric	read the data through a REST service provided by the data fabric after simulation response	few octets	every second/minutes/hours	every second/minutes/hours	read

7.9.2.1.2 Output

Module Output												
Data	Producer	Output method	Computed from	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency
Production plan (quantity per period over the planning horizon)	Production planner	Post to a Rest service	optimization	doubles	units of items	For the production scheduler/simulation to know the production load	Scheduler, Simulation,	Data fabric	no communication to scheduler (send an	few octets	Every weeks	Every day

							Production manager UI		error message to inform the user he must create a production plan). Send a request to simulation respond to a request made by production manager UI			
Extra capacity required per period and per resource	Production planner	Post to a Rest service	optimization	doubles	minutes	For the production manager to adjust the capacity	Simulation, Production manager UI	Data fabric	Send a request to simulation respond to a request made by production manager UI	few octets	Every weeks	Every day
Quantities to ordered to suppliers/subcontractors	Production planner	Post to a Rest service	optimization	doubles	units of items	For the production manager to place orders to suppliers	Simulation, Production manager UI	Data fabric	Send a request to simulation respond to a request made by production manager UI	few octets	Every weeks	Every day

7.9.2.2 Model Acquisition for scheduling & Optimisation

7.9.2.2.1 Input

Module input												
Data	Producer	Input method	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency	Read/write
A table or a set of tables with schedule data and signature denoting relevant input-output attributes for the model to be generated	ERP /user/	Manual / semi-automatic creation of a file or files containing schedule tables and signatures (i.e. selection of relevant tables and columns) should be done by the caller The signature file, that describes which attributes must be considered as an input / output for the model acquisition	*.xls *.xlsx *.odt JSON	All data within tables must be presented as integers	Data tables and signature files will be converted to the format recognized by SICStus. Then the Model acquisition module will extract a constraint model from the presented schedule data that can later be used for further optimization of the schedule	Model Acquisition module	Either data fabric, or locally on the user's machine.	Since Model Acquisition module requires additional licensed software to run, as of now it will be run on machines of IMT-Atlantique by sending table files directly to them.	All the data provided by the data tables and signatures may be used by the Model Acquisition module; otherwise it could lead to incomplete or imprecise constraint model.	Decided by the producer	Whenever the Model Acquisition module is launched.	Left unchanged

		, must be handled by the user. A system for automatic recognising of which attributes could be used as inputs/outputs is outside of project's scope										

7.9.2.2.2 Output

Module Output												
Data	Producer	Output method	Computed from	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency
A constraint model created from the schedule tables and signatures	The user who launched the Model Acquisition module (see Comm protocol for 6.1.2.3.1)	The Model Acquisition module produces one or more files containing information about extracted	Data tables and signature files will be converted to the format recognized by SICStus.	One or several of formats below: -*.pl, PROLOG file -*.tex file -Math	non applicable	The extracted constraint model could be used later for Optimization for Scheduling module.	Optimization for Scheduling module.	Locally on on user's machine.	The user later sends these files to the machine that handles Optimization for Scheduling module. if both modules are on the same	Decided by the producer	Once, after the user launched the Model Acquisition module.	Once per launch of Optimization for Scheduling module.

		constraint model.		ML format - Latex report					computer, the user provides file path to the Optimization for Scheduling module.			

7.9.2.3 Optimization for scheduling

7.9.2.3.1 Input

Module input												
Data	**Producer**	**Input method**	**Data format**	**Data Units**	**Purpose**	**Consumer**	**Data storage**	**Comm protocol**	**Quantity and/or reference**	**Write Frequency**	**Read Frequency**	**Read/write**
A constraint model describing a schedule	User, after running Model Acquisition model.	The user runs a program to that converts a *.pl PROLOG file containing a constraint model created by Model Acquisition module	*.pl	Unapplicable	Run the model, in combination with a selected objective function, on a solver to produce an optimized schedule	the user can use the new schedule for the Production Planner	Either Data Fabric or locally on the user's machine	The user must send files to the computer that has MiniZinc or another solver via MiniZinc interface, or	The model must be loaded completely	Decided by the producer	Whenever Optimization for Scheduling module is launched. After it launched, the model file is read once to be converted in the format recognized by MiniZinc	Left unchanged

		into MiniZinc file.						he can run the solver himself			or another solver via MiniZinc interface.	
Objective function (KPI) for constraint model	The criteria function is extracted from from a signature file sent. For now, the objective function may handled as a sum or maximum of all entries for an attribute column within one of the input tables	The user runs a program to that converts a JSON containing an objective function into MiniZinc file, or use	JSON file	Unapplicable	Run a selected objective function , in combination with the selected model, on a solver to produce an optimized schedule	the user can use the new schedule for the Production Planner	Either Data Fabric or locally on the user's machine	The user must send files to the computer that has MiniZinc and/or another solver, or he can run the solver himself	Information about the objective function could be handled in a separate file (in this case it'll be loaded completely) or be a part of an input signature file for Model Acquisition module (in this case only small part of the file is used)	Decided by the producer	Whenever Optimization for Scheduling module is launched. After it launched, the model file is read once to be converted in the format recognized by MiniZinc or another solvervia MiniZinc interface.	Left unchanged

7.9.2.3.2 Output

Module Output

Data	Producer	Output method	Computed from	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency
An optimized schedule	The user who launched the Optimization for Scheduling module (see Comm protocol for 6.1.2.4.1)	The system produces one or more files. MiniZinc will save results in its own format that must be converted into JSON	A file with a constraint model and a preselected file with an objective function. Both files a run through MiniZinc or another solver via MiniZinc interface	JSON	not applicable	An optimized schedule to be used later in the Production Planner module	Production Planner module	Data Fabric or locally	Could be sent to Data Fabric or to a user who handles Production Planner module	Decided by the producer	Once, after the user launched the Optimization for Scheduling module.	Decided by Production Planner module.

7.9.2.4 Production Manager UI

7.9.2.4.1 Input

Module input												
Data	Producer	Input method	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency	Read/write
Production planning	**production planning mudule**	Request to production	JSON		Displaying important information	Production manager	Data fabric	HTTP via		Week/Month	Week	read and write (new constraint

output data		planning module			related to output of production planning module			REST rules				based on KPIs adjustement)
Model acquisition (scheduling)	**Model acquisition (scheduling) module**	Request to model acquisition (scheduling) module	JSON		Displaying main information related to the outputs of model acquisition (scheduling) module	production manager	Data fabric	HTTP via REST rules		Hours/days	Hours	Read
Simulation	**simulation module**	Request to the simulation module	JSON		Displaying customers orders and due date, initial condition of the factory,calendar with release dates per order, shift model and machine group priorities, time horizon and number of montecarlo simulations	Production manager	data fabric, domain model	HTTP via REST rules		Hours/weeks/months	Hours	Read

7.9.2.4.2 Output

Module Output												
Data	Producer	Output method	Computed from	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency
Production planning output data	production planning mudule	Thin client	production planning mudule	Tables, chart		Displaying important information related to	Production manager	Data fabric	HTTP via		weeks/Month	Week

						output of production planning module			REST rules			
Model acquisition (scheduling)	Model acquisition (scheduling) module	Thin client	Model acquisition (scheduling) module	UML diagram, text form of the acquired model, gantt chart, other chart		Displaying main information related to the outputs of model acquisition (scheduling) module	production manager	Data fabric	HTTP via REST rules		Hours/days	Hours
Simulation	Simulation module	Thin client	Simulation module	Tables, calendars , chart		Displaying customers orders and due date, initial condition of the factory,calendar with release dates per order, shift model and machine group priorities, time horizon and number of montecarlo simulation	Production manager	Data fabric	HTTP via REST rules		Hours/days	Hours

7.9.3 Real-time control and actuation

[will be added in further revision of the living document]

7.9.3.1 Execution Controller

7.9.3.2 Digital Twin

7.9.3.3 AI for Fenceless Human-Robot collaboration

7.9.4 Secure and intelligence data fabric

7.9.4.1 Data Fabric

7.9.4.1.1 Input

Module input												
Data	Producer	Input method	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency	Read/write
Payload and metadata	ASSISTANT tools and twins	APIs and service network interfaces	structured (e.g., JSON, XML), unstructured (binary), and time series data	Text documents, binary files, domain models	Facilitate data management and provisioning for other ASSISTANT tools	ASSISTANT tools	Files, databases	JSON-based REST	Dependent on clients and scenarios	Dependents on clients and scenarios, new data sets are expected to be produced once and iteratively refined	Dependent on clients and scenarios, reads (and queries) are expected to outnumber write accesses for most types of data	All data assumed to be persistently stored in a read-only format, updates are made using a copy-on-write semantic

7.9.4.1.2 Output

Module Output												
Data	Consumer	Output method	Computed from	Data format	Data Units	Purpose	Consumer	Data storage	Comm protocol	Quantity and/or reference	Write Frequency	Read Frequency
Payload and metadata	ASSISTANT tools and twins	APIs and service network interfaces	Stored data and / or data provided by clients in requests	structured (e.g., JSON, XML), unstructured (binary), and time series data	Text documents, binary files, domain models	Facilitate data management and provisioning for other ASSISTANT tools	ASSISTANT tools	Files, databases	JSON-based REST	Dependent on clients and scenarios	Dependent on clients and scenarios, reads (and queries) are expected to outnumber write accesses for most types of data	All data assumed to be persistently stored in a read-only format, updates are made using a copy-on-write semantic