

ZDMP: Zero Defects Manufacturing Platform



WP1: Management: Procedures, Metrics, Coordination, and Reporting

EU ID: D012: Technical Management – WP6 Report - Vs: 1.0.0A

ZDMP ID: D1.4.3a

Deliverable Lead and Editor: Tim Dellas, ASC

Contributing Partners: SAG, SIVECO, ASC, ICE

Date: 2020-06

Dissemination: Public

Status: EU Approved

Abstract

The WP6 technical management overview report gives an in-depth view of the degree of completion of the software components developed in WP6. It shows the activities carried out so far, the planning, the main milestones achieved and the roadmap and general progress of those components.

Grant Agreement:
825631



Document Status

Deliverable Lead	Tim Dellas, ASC
Internal Reviewer 1	Francisco Barrena, ITI
Internal Reviewer 2	Philip Usher, ICE
Internal Reviewer 3	Stuart Campbell, ICE
Type	Deliverable
Work Package	WP1: Management: Procedures, Metrics, Coordination, and Reporting
ID	D012: Technical Management – WP6 Report
Due Date	2020-06
Delivery Date	2020-06
Status	EU Approved

Project Partners:

For full details of partners go to www.zdmp.eu/partners



Executive Summary

The main objective of “WP6: ZDMP Platform Building” is to create a base platform for ZDMP, ensuring that all the other components can interlink in a meaningful way to provide an all-over platform for developers to build zApps on and for users to use zApps on. The deliverables of this work package and the WP1 Management work package are divided into software packages and document/reports. In terms of reporting:

- **Process/Status:** This report corresponds to D012 Technical Management: WP6 Report of WP1 Management: Procedures, Metrics, Coordination, and Reporting, and as identified in the DOA, focuses on the process/status of the work accomplished in WP6
- **Software:** All WP6 software deliverables of T6.1-T6.5 (type “OTHER”) are available in the ZDMP public repository with access details and install instructions as indicated in Section 2.1 of this report. Each of these is accompanied by a very brief cover document. The deliverables are software components: SDK: Applications and Service Builder (T6.1), Secure Business Cloud (T6.2, including Storage, Marketplace and Security System Modeler), Human Collaboration Environment (T6.3), Platform Integration and Federation (T6.4) and Inter-Platform Interoperability (T6.5).

This document is structured in the main sections below (one per task/component) in direct relation to the WP6 tasks. Whilst from the reporting perspective each one describes planned activities, the progress, and the next activities for the period, from the software perspective an online public documentation is provided to describe the software functionalities and provides pointers to the location of the software developed so far. In this period, this document reports on the following tasks and specific subtasks, if partners have separated the task in a way, those subcomponents are reported on independent of each other:

- T6.1 - SDK: Applications and Service Builder
 - Application Builder
- T6.2 - Secure Business Cloud
 - Storage
 - Marketplace
 - Security System Modeler
- T6.3 - Human Collaboration Environment
- T6.4 - Platform Integration and Federation
 - Application Runtime
 - Portal
 - Services and Message Bus
- T6.5 - Inter-Platform Interoperability
- T6.6 - General Cross-Task Integration and Improvement

Table of Contents

0	Introduction.....	1
1	Context.....	5
	1.1 Positioning in the Overall ZDMP Architecture	5
	1.2 Purpose of the WP6 Components.....	6
	1.3 Metrics	7
	1.4 Components public documentation.....	7
	1.5 Risks	8
2	T6.1 SDK: Applications and Service Builder.....	9
	2.1 Public documentation.....	9
	2.2 M18 Report	9
	2.2.1 Architecture Implementation	9
	2.2.2 Planned Activities.....	10
	2.2.3 Progress.....	11
	2.2.4 Limitations.....	14
3	T6.2 Marketplace.....	15
	3.1 Public Documentation	15
	3.2 M18 Report	16
	3.2.1 Architecture Implementation	16
	3.2.2 Planned Activities.....	16
	3.2.3 Progress.....	17
	3.2.4 Limitations.....	18
4	T6.2 Storage.....	19
	4.1 Public documentation.....	19
	4.2 M18 Report	19
	4.2.1 Architecture Implementation	19
	4.2.2 Planned Activities.....	20
	4.2.3 Progress.....	20
	4.2.4 Limitations.....	21
5	T6.2 Security Designer	22
	5.1 Public documentation.....	22
	5.2 M18 Report	23
	5.2.1 Architecture Implementation	23
	5.2.2 Planned Activities.....	23
	5.2.3 Progress.....	24
	5.2.4 Limitations.....	25
6	T6.3 Human Collaboration Environment.....	26
	6.1 Public documentation.....	26
	6.2 M18 Report	27
	6.2.1 Architecture Implementation	27
	6.2.2 Planned Activities.....	27
	6.2.3 Progress.....	28
	6.2.4 Limitations.....	29
7	T6.4 Application Runtime	30
	7.1 Public documentation.....	30
	7.2 M18 Report	31
	7.2.1 Architecture Implementation	31
	7.2.2 Planned Activities.....	31
	7.2.3 Progress.....	32

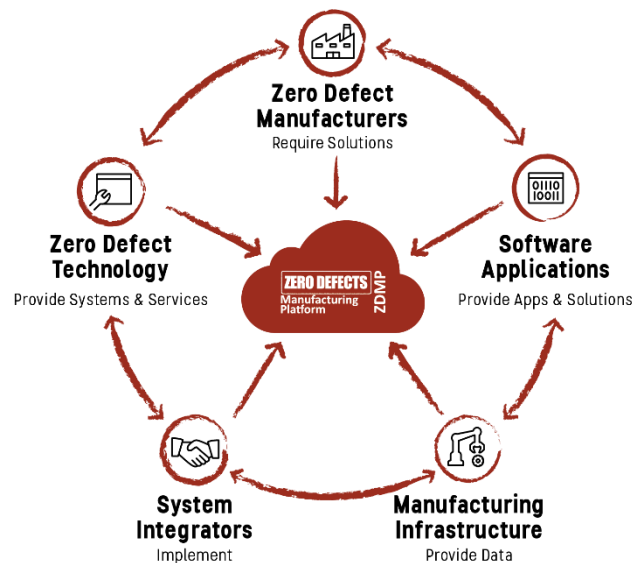
	7.2.4	Limitations.....	33
8	T6.4	Portal.....	34
	8.1	Public documentation.....	34
	8.2	M18 Report	34
	8.2.1	Architecture Implementation	34
	8.2.2	Planned Activities.....	35
	8.2.3	Progress.....	35
	8.2.4	Limitations.....	36
9	T6.4	Services and Message Bus.....	37
	9.1	Public documentation.....	37
	9.2	M18 Report	38
	9.2.1	Architecture Implementation	38
	9.2.2	Planned Activities.....	38
	9.2.3	Progress.....	39
	9.2.4	Limitations.....	40
10	T6.5	Inter-Platform Interoperability	41
	10.1	Public documentation.....	41
	10.2	M18 Report	41
	10.2.1	Architecture Implementation	41
	10.2.2	Planned Activities.....	42
	10.2.3	Progress.....	43
	10.2.4	Limitations.....	44
11	T6.6	General Cross-Task Integration and Improvement	45
	11.1	Public documentation.....	45
12		Risks.....	46
13		Conclusions.....	47
		Annex A: Document History	48
		Annex B: References	49

0 Introduction

0.1 ZDMP Project Overview

ZDMP – Zero Defects Manufacturing Platform – is a project funded by the H2020 Framework Programme of the European Commission under Grant Agreement 825631 and conducted from January 2019 until December 2022. It engages 30 partners (Users, Technology Providers, Consultants and Research Institutes) from 11 countries with a total budget of circa 16.2M€. Further information can be found at www.zdmp.eu.

In the last five years, many industrial production entities in Europe have started strategic work towards a digital transformation into the fourth-industrial revolution termed Industry 4.0. Based on this new paradigm, companies must embrace a new technological infrastructure, which should be easy to implement for their business and easy to implement with other businesses across all their machines, equipment, and systems. The concept of zero-defects in the management of quality is one of the main benefits deriving from the implementation of Industry 4.0, both in the digitalisation of production processes and digitalisation of the product quality.



To remain competitive and keep its leading manufacturing position, European industry is required to produce high quality products at a low cost, in the most efficient way. Today, manufacturing industry is undergoing a substantial transformation due to the proliferation of new digital and ICT solutions, which are applied along the production process chain and are helping to make production more efficient, as in the case of smart factories. The goal of the ZDMP Project is to develop and establish a digital platform for connected smart factories, allowing to achieve excellence in manufacturing through zero-defect processes and zero-defect products.

ZDMP aims at providing such an extendable platform for supporting factories with a high interoperability level, to cope with the concept of connected factories to reach the goal of zero-defect production. In this context, ZDMP will allow end-users to connect their systems (ie shop-floor and Enterprise Resource Planning systems) to benefit from the features of the platform. These benefits include product and production quality assurance amongst others. For this, the platform provides the tools to allow following each step of production, using data acquisition to automatically determine the functioning of each step regarding the quality of the process and product. With this, it is possible to follow production order status and optimize the overall processes regarding time constraints and product quality, the achieving the zero defects.

0.2 Deliverable Purpose and Scope

The purpose of “D012 Technical Management – WP6 Report” is to document the software produced in WP6 and to show the activities carried out so far, and plans, for the WP6 components, and the features available for each one of them.

This document is mainly for internal reporting about scope of software components, mapping to architecture, planned activities, progress, and limitations found. However, the report includes links to public repositories which complement and adds information about guidelines to set up a component, access to the service, installation notes, configurations details, requirements, etc.

Specifically, the DOA states the following regarding this Deliverable:

T1.4	Technical Management: Leadership and Technical Reporting			ITI	M4-36
D012 D013 D014	Technical Management: WP6 Report	R	PU	18, (24), 30, (36), (42) 48	RDI3, 5, 8
On the reporting side this task will deliver 6 monthly overview and per WP reports detailing the activity of T1.4 and T1.5 and WP5-8. Thus, these reports will be the conduit for the necessary paperwork reports related to the WP5-8 software deliverables since within these WPs the only deliverables are software prototypes accompanied by a cover page pointing to T1.4.x.x series of deliverables. Whilst six-monthly only the reports at the formal project periods will be presented to the Funding Authority and the bracketed reports are internal only. Such reports will include installation notes, initial guidance, and configuration details. Note the days to perform this module reporting will come from the RDI tasks themselves.					

0.3 Target Audience

The WP6 report deliverable aims primarily to the ZDMP managers in charge of producing the software components and zApps of the project, and the links to the public repositories are interesting for any party interested in the software produced.

0.4 Deliverable Context

Its relationship to other documents is as follows:

Primary Preceding documents:

- **D021 – Methodology Document:** This deliverable gives the foundations of collaborative software development and points to the different tools that are being used in the project
- **D048 – Requirements Document and Update:** This deliverable contains the end user requirements and those are mapped to the different ZDMP components
- **D051 – Global Architecture Specification and Update:** This deliverable depicts the first version of the architecture
- **D053 – Functional Specification and Update:** This deliverable contains the functional specification of ZDMP components and zApps
- **D055 – Technical Specification and Update:** This deliverable describes the different APIs of the components

Primary Dependant documents:

- **D006 – Technical management: Overview Report:** Describes the software development approach and provides integration guidelines among components

0.5 Document Structure

This deliverable is broken down into the following sections:

- **Section 1: Context:** An introduction to this deliverable including a general overview of WP6 components and their relationship with the overall architecture
- **Section 2: T6.1 SDK: Application and Service Builder:** To help with creating cross platform applications based on Angular, Ionic, Electron, TypeScript and JavaScript and to tie in services from ZDMPs marketplace into services and zApps
- **Section 3: T6.2 Marketplace:** Scope, link to public repository, features, progress, and planned activities for this component
- **Section 4: T6.2 Storage: API gateway for general storage based on Apache Knox as the backbone solution**
- **Section 5: T6.2 Security Design:** Checks the security of the entire setup
- **Section 6: T6.3 Human Collaboration Environment:** Collaborate between developers, manufacturers, and technical systems of ZDMP
- **Section 7: T6.4 Application Runtime:** The Application Run-time provides both the capability to orchestrate and host the individual components of the system using containerization
- **Section 8: T6.4 Portal:** The Portal is a web-entry into ZDMPs technical systems for developers, customers and other parties who wish to interact with ZDMPs services
- **Section 9: T6.4 Services and Message Bus:** Service and Message Bus provides communication between users, components and apps
- **Section 10: T6.5 Inter-Platform Interoperability:** Consists of work towards combining multiple digital platforms together. The gateways and cross-platform functionalities are bundled in this task but managed in the Marketplace and Service and Message Bus subcomponents of T6.2 and T6.4
- **Section 11: T6.6 General Cross-Task Integration and Improvement:** A meta task for the integration between components
- **Section 12: Risks:** Risks associated with the work package
- **Section 13: Conclusion:** Main outcomes
- **Annexes:**
 - **Annex A: Document History**
 - **Annex B: References**

0.6 Document Status

This document is listed in the Description of Action as “public” since it provides information which may be relevant to external parties wanting to test or use the software produced within WP6.

0.7 Document Dependencies

This document is part of an iteration of living deliverables. This is the first version produced in M18. The following versions to be submitted for approval will be delivered in M30 and M48, note consortium-internal updates of this report will be produced in M24, M36 and M42. Each subsequent version will provide updated information mainly, but not only, on the component overview. The additions will be clearly marked in further iterations of the document.

0.8 Glossary and Abbreviations

A definition of common terms related to ZDMP, as well as a list of abbreviations, is available at <http://www.zdmp.eu/glossary>.

0.9 External Annexes and Supporting Documents

- Online documentation for each component is available at: <https://software.zdmp.eu/docs/components>

0.10 Reading Notes

- None.

0.11 Document Updates

- This is the first version of this deliverable

1 Context

ZDMP's main aim is to provide an extendable platform for supporting factories with a high interoperability level, to cope with the concept of connected factories to reach the goal of zero-defect production. The ZDMP approach is modular by design and it is formed by a set of components which provides added-value features.

The features provided by WP6 components represent the basic elements to create a digital manufacturing platform for 3rd party developers to publish zApps on and sell them to customers.

1.1 Positioning in the Overall ZDMP Architecture

Figure 1 shows the global architecture. Each component fits within one tier:

- **Developer Tier (Design-time):** These components aide in the production of containerised applications for zero defect manufacturing - zApps
- **Enterprise Tier (Use-time):** These components assist the run-time
- **Platform Tier (Run-time):** This is where zApps are installed. This component consists of run-time services to provide a base level functionality for ZDMP
- **Edge Tier (Run-time):** This is composed of the Distributed Computing component which allows certain zApps to be run at various locations of the system for performance gains

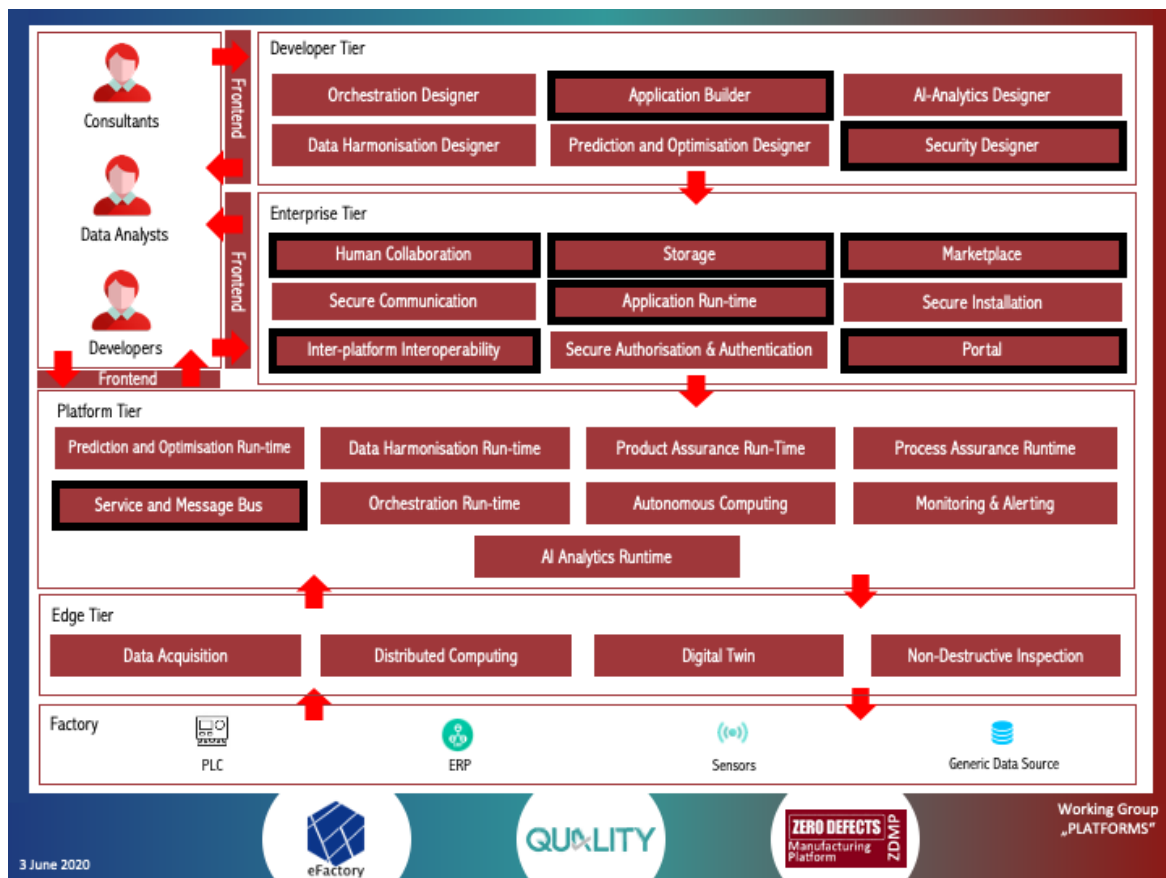


Figure 1: ZDMP Global Architecture, Components of WP6 are framed black

The Figure shows the architectural WP6 components covered by this report. In addition, it gives information about the different general qualities of each one of them.

Components	End-User Application in its own right	Callable Service	Intrinsic in the ZDMP Ecosystem	Standalone from the ZDMP Platform	Available in the Marketplace	Largely Created in Project
Developer Tier						
T6.1: Application Builder	✓			✓	✓	
T6.2: Security Designer	✓			✓	✓	
Enterprise Tier						
T6.2: Marketplace	✓		✓			
T6.2: Storage		✓	✓		✓	
T6.3: Human Collaboration	✓		✓		✓	✓
T6.4: Application Run-time			✓		✓	
T6.4: Portal	✓		✓			
T6.5: Inter-platform Interoperability		✓				
Platform Tier						
T6.4: Service and Message Bus		✓	✓	✓	✓	

Figure 2: WP6 ZDMP Components Classification Table

All components of WP6 share the same qualities:

- **Callable Service:** Typically, applications may be downloaded and put in a user environment; however, services may then be called as they may run solely in someone else's environment – eg in another factory. However, care must be taken since the boundary is blurry – for example, an application is created/downloaded and run as a service for another party then it could fall into both or either camp
- **Intrinsic in the ZDMP Ecosystem:** This implies that the project builds/buys this service/component and it would (always) form part of the platform
- **Available in the Marketplace** Self-explanatory
- **Largely Created in Project:** Self-explanatory

1.2 Purpose of the WP6 Components

It is important to define the purpose of the ZDMP components to set up the scene where these components are being developed:

- **T6.1 SDK: Applications and Service Builder:** To assist in creating cross platform applications based on Angular, Ionic, Electron, TypeScript, JavaScript and to tie in services from ZDMPs marketplace into services and zApps
- **T6.2 Secure Business Cloud:** Consists of a Marketplace to sell the zApps to customers, Storage to save data on client- or ZDMP infrastructure, and the System Security Modeler to check the security of an installation
- **T6.3 Human Collaboration Environment:** Provides collaboration between developers, manufacturers, and technical systems of ZDMP
- **T6.4 Platform Integration and Federation:** Consists of a Platform to provide resources and a framework to run server-side components and zApps, a Portal, which is a web-entry into ZDMPs technical systems for developers, customers and

other parties who wish to interact with ZDMPs services, and a Service and Message Bus to provide communication between users, components and zApps

- **T6.5 Inter-Platform Interoperability:** Functionality for providing connectivity to other platforms. The gateways and cross-platform functionalities are bundled in this task and managed in the Marketplace and Service and Message Bus subcomponents of T6.2 and T6.4

1.3 Metrics

The metrics from the DOA are as follows:

Result	The results of this WP will be an SDK capable of building ZDMP Apps and Services (T6.1) which can then be placed on the ZDMP Business Marketplace (T6.2). T6.3 represents the cooperative part of the platform and the overall platform features will be delivered in T6.4. Inter-platform interoperability with other platforms is delivered in T6.5. A cross WP task for integration is provided in T6.6.	
Metric	<ul style="list-style-type: none"> • 50 API/Service connections available to SDK • Internal/External collaboration chat and video features available 	<ul style="list-style-type: none"> • Platform and Tasks integration require only resources specified (well designed) • Interoperability with key APIs of 2 further platforms

In DOA Amendment 2, it is proposed to augment these since during the course of this activity it became apparent there were no metrics of T6.2 – see relevant rows in the table whose content is self-explanatory. On this basis, the current status is as below, reflecting the early stage of the project and considering the WP6 metrics are more orientated to later stage even if the work has already started and relevant now. Note there are no metrics for T6.6 since this is simply capacity for further general integration work.

Metric	Related Task(s)	Target	M18	M30
50 API/Service connections available to SDK	T6.1	50	3	40
40 ZDMP Assets are uploaded on the Marketplace (PROPOSED)	T6.2	40	0	20
15 Subcall Assets are uploaded on the Marketplace (PROPOSED)	T6.2	15	0	0
Internal/External collaboration chat and video features	T6.3	Fully featured	Partial	Partial
Platform and Tasks integration require only resources specified (well designed)	T6.4	Well designed	To be assessed in M18	Prototype
Interoperability with key APIs of 2 further platforms	T6.5	2	0	1

1.4 Components public documentation

Technical documentation of the components is publicly available in the following URL:

<https://software.zdmp.eu/docs/>

The public documentation includes for each one of the components the following information:

- **Source code:** Location
- **Latest release:** Link to download and numbering version
- **Open API Spec:** Available Open API Specifications for the component

- **General Description:** Description of the use of the component in the project general scope
- **Features:** List of features provided by the component
- **Requirements:** Technical requirements of the component (software and hardware)
- **Installation:** Description of the steps to install the component
- **How to Use:** Instructions of how to set up and use the component

1.5 Risks

For the status of risks related to the work package see Section 12. For project-orientated technology risks in general, please refer to the following document: D006 – Technical management: Overview Report.

2 T6.1 SDK: Applications and Service Builder

2.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/developer-tier/applications-builder/>

ZDMP Documentation Search this site...

Documentation / Components / Developer Tier / Applications Builder

Applications Builder

Application and Service Builder

Resource	Location
Source Code	Link
Latest Release (v0.1.4)	Download
X Open API Spec	Link

General description

The application builder is a HTML / CSS / JS based web frontend that enables developers to build a basic infrastructure of zApps using drag and drop components. These drag and drop components are already linked with APIs provided by other technical components and include calls and bindings to ZDMP services. This enables to accomplish certain basic functionalities that every app needed including notifications, error reporting, sign up forms and dashboards for data visualization, etc.

General description
Features
Requirements
Installation
How to use
Format Cheat-Sheet (remove from final version)
Components
 Alerts
 Another Heading
 This Document
 Pixel Count
 Contact Info
 External Links
 Functional Requirements M18
 Status

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	Source code can be found here: https://zdmp-gitlab.ascora.eu/zdmp_code/developer-tier/t6.1-applications-builder
Latest release:	Available	Vs 1.0.0
Open API Spec:	Available	N/A
General Description:	Available	N/A
Features:	Available	N/A
Requirements:	Available	N/A
Installation:	Available	Using docker Image
How to Use:	In process	Resolving final queries and pending review comments

2.2 M18 Report

2.2.1 Architecture Implementation

The first version of the Application builder allows users to create an application, add multiple pages in the zApp, create a Manifest file, add UI components, and build it. Frontend components which can be added using frontend editor include data charts, input forms, user management elements (registration form, login forms and change password form), tables, images, notifications, audio, and video. The application created via frontend

editors has basic metadata such as colour, language, name, author, and version which can be customized after creation of application.

The Frontend editor also has the option to translate the app into multiple languages, user can add their own language files with the translation which he wants to include in the app. After building the app, typescript, HTML and SCSS files are generated which can be edited to change the design or functionality of build-in components.

The user can also include in-zApp advertisement with the hard-coded data. The basic backend to retrieve information about other zApps is also in-place which can be used after availability of Z- zApps on the marketplace. For tracking the usage of applications telemetry services are being built which provides usage information about application in the form of charts and graphs which is helpful to learn about user experience. In the first version of application builder, dummy data is being used for telemetry service.

In the next version, user will be able to call other ZDMP APIs, including message bus, user management, data analytics and Marketplace. Next version will also enable application builder to package application and upload it to the Marketplace.

The figure below shows the status of the architecture implementation:

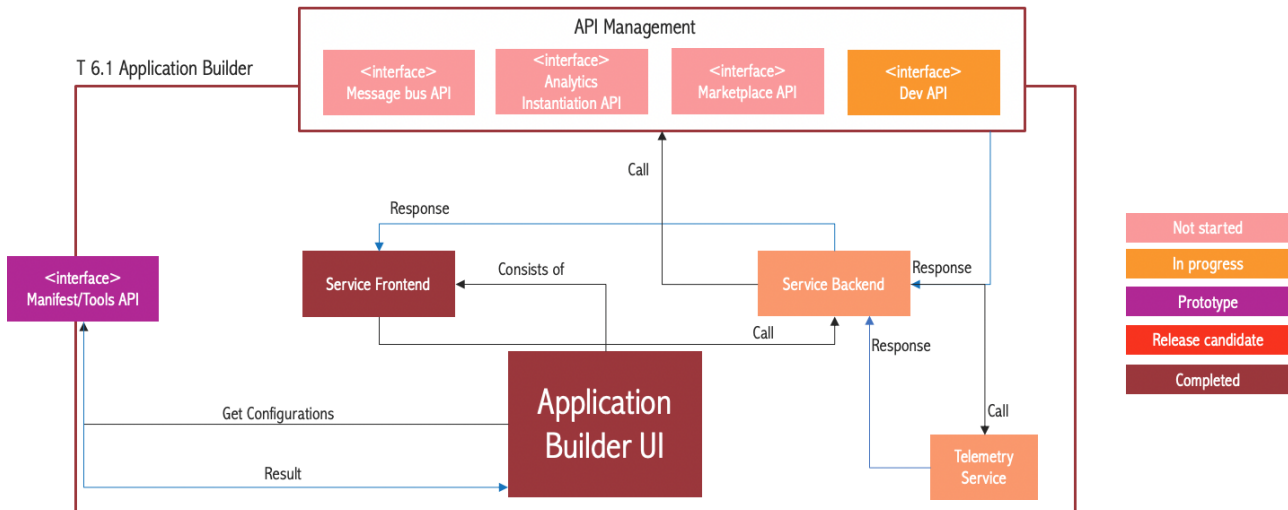


Figure 3: Application Builder Architecture

2.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed] Application Builder:** Create a web frontend based on HTML / SCSS / TS that enables developers to build a basic infrastructure of zApps using drag and drop components. Developers should be able to add following elements and functionalities using application builder:
 - Forms/charts/buttons and other UI elements
 - Error Reporting
 - User Authorisation and Authentication
 - Multi Language Support
 - Application Logging
 - Automatic Updates

- **[Completed] Define manifest model:** Define the necessary fields to be included in application manifest
- **[Completed] Create Application Manifest Form:** Create a manifest form in Application Builder which enables developer to add manifest properties in applications and services
- **[Ongoing] Generate server-side data models:** Server-side data models for calling ZDMP APIs & verdaccio
- **[Ongoing] Telemetry server-side graph building**
- **[Ongoing] Integrate user management services**
- **[Not Started] Integrate monitoring services / data analytics services / alerting services:** Library and server-side API to give us routes
- **[Not Started] List of apps / Services:** A service that identifies the user of a zApp depending on the user and the configuration of their ZDMP instance, this returns a list of services, components and zApps that might be necessary for the client zApp to work, and return this (maybe with any metadata necessary) to the client (so the client can connect to the right services, even if it is only an instance running locally)
- **[Not Started] Encrypted Storage API:** Homogenous encryption
- **[Not Started] SDK logging:** Existing but will be expanded

2.2.3 Progress

The development team was managed based on agile methodologies, conducting sprints of 2 weeks of duration. The following table shows the progress made by the developers per sprint. When applicable, sprint issues are linked to features specified in the functional specifications (D053 – Functional Specification and Update). Note that the “PEnnn” (or similar) refers to the unique identifier assigned to each feature in the functional specification and if there is no PEnnn it implies that the issue is related to a new feature to be included in a future revision of the functional specification or to a non-functional requirement. The documentation is only (generally) up to the last full sprint in May; this point was chosen as the last feasible documentation point to be able to produce and review these documents. The next report will follow on after this point.

Note that in some case spring finish reporting a little earlier (eg mid-May) due to scheduling/public holiday reasons. Of course, they then started up in June again.

Sprint 0 (Feb 02,2020 – Feb 16, 2020)
<ul style="list-style-type: none"> • Define top 10 features. Define top 10 features planned for the M18 milestone.
Sprint 1 (Feb 17, 2020 – March 01, 2020)
<ul style="list-style-type: none"> • Create UI template and behaviour template repositories for Application Builder. Provides UI elements to be used by developers. These elements come with its default properties but are also customisable for its respective purpose. This includes charts and diagrams that can be data-bound to ZDMP services (zAssets), as well as an “About” page with information and informative links. Behaviour elements provide default behaviours for activities, which are related to UI interactions. Those behaviours are easy to integrate and applicable for eg forms, downloads, registration, notifications, etc. Related subtasks: T61A013 Provide a Register form for new Users, T61A014 Provide User Login Form, T61A015 Provide a button to reset a password, T61A016 Provide a form to reset a password, T61A019 Provide UI Elements, T61A020 Show Notification, T61A021 Dismiss Notification, T61A022 Remind Later. • Create dummy data for Client-side implementation of Advertisement Framework. Use dummy data until backend is ready.

- **Project setting for implementation of Advertisement Framework.** Create an angular project to implement advertisement framework.

Sprint 2 (March 02, 2020 – March 16, 2020)

- **Embed editor and syntax highlights into application builder.** Editor can be used to modify generated code.
- **Client-side implementation of vertical banner ads.** Display vertical banner ads in zApps.

Sprint 3 (March 17, 2020 – March 30, 2020)

- **Add multiple languages support in application builder.** The user can choose between several languages and select the intended language to use for this ZDMP Asset. Related subtasks: T61A018 Switch language, T61A017 Translate text.
- **Define project structure in application builder.** Generate typescript, HTML and CSS files after build.
- **API to get metadata of all zApps available for advertisement.** This API will get metadata of all APPs, possibly through secure installation component. Related subtasks: T61A002 Get zApps APIs, T61A001 Get List of Stored Configurations, T61A003 Get ZDMP Asset APIs and Manifests.
- **Create a web backend to provide zApps metadata for advertisement framework.** For each advertisement, there should be the following data / metadata editable in the web backend:
 - Advertised Product ID
 - Product Name (string)
 - Supported Platforms
 - Discount Price
 - Percentage of price discount (calculated)
 - Discount link
 - Valid dates
 - From (datetime)
 - Until (datetime)
 - Advertisement images
 - Advertisement type (Vertical banner, Horizontal banner, custom)
 - Advertisement minimum width
 - Advertisement minimum height
 - Advertisement maximum width
 - Advertisement maximum height
 - Advertisement recommended position
- **Collect metadata for the ZApps that want to be advertised - Advertisement framework server-side implementation.** This metadata will (hopefully) be available in the zApp license manager built by SIVICO in the marketplace. This metadata can hopefully be extracted from there (by calling from their products-API), otherwise a basic Cockpit based thing can be setup and add some example data to it:
 - Name (string)
 - Product Version (double)
 - Version Purpose (choice between shop, website, promotion)
 - Product ID (not database ID, need to crossmatch with marketplace)
 - Supported Platforms (Android, iOS, Windows, Mac, Linux)
 - Primary Price model (choice between free, one-time-purchase, subscription, volume billing)
 - Price (eg 10€)
 - Supported languages
 - Product download URLs for Apple App Store, Google Play Store, External URLs

Sprint 4 (March 31, 2020 – April 13, 2020)

- **Enable compiler and builder to create an app from UI and behaviour elements.** Compile and build app containing all the UI elements (forms, buttons, tables etc) and behaviour elements (registration, logic etc).
- **Client-side implementation of horizontal banner ads.** Display horizontal banner ads in zApps.

Sprint 5 (April 14, 2020 – April 27, 2020)

- **Study online documentation of OpenAPI Generator tool.** Documentation available for the proposed tool for the implementation of the client library generator has been studied (<https://openapi-generator.tech/docs/generators/>)
- **Discuss roadmap for implementation of client library generator with task leader.** Steps for the implementation of a client library generator has been discussed in a telco with the task lead held on Apr. 17th, 2020.
- **Analyse architecture of telemetry service.** Telemetry is providing information from zApps on user-premises into a server-side system, so that developers can analyse how their applications are used and can track usage statistics like advertisement views.
- **Analyse and clean sample telemetry data sent by ASC.** JSON datastore dump containing sample data with a structure similar to the one that will be used in ZDMP was loaded in Microsoft BI, cleaned up and analysed.
- **Test API's configured on STRAPI.** Strapi is an open-source content management system without a frontend and instead a REST API.
- **Add First User to STRAPI ACL.** To add users to the access control list of STRAPI CMS.

Sprint 6 (April 28, 2020 – May 11, 2020)

- **Create a list of manifest form fields.** Write a list of properties to be included in manifest.json. Related subtasks: T61A023 Manifest reading, T61A008 Structure the Build Manifest.
- **Manifest Meta Data format.** Define the user generated and automatically generated manifest properties. Related subtasks: T61A023 Manifest reading, T61A008 Structure the Build Manifest.
- **open-api generator tools installed.** selected tool for client API generator installed on local server. Testing ongoing.
- **Started creating skeletons for telemetry service.** First version based on sample telemetry data.
- **Contact technical partners to inquire about client languages used.** As discussed as part of the roadmap for the implementation of client libraries generator, will contact technical partners to collect information about the programming languages they are planning to use to develop zApps or client modules.

Sprint 7 (May 12, 2020 – May 25, 2020)

- **Setup an instance of Keycloak Server.** This task comes under the purview of User Management. A Keycloak server is used for authentication and authorization in ZDMP. Through this issue expectation is in trying to install a local sever to aid the development effort.
- **Create Application Manifest Form.** Create a manifest form in Application Builder which enables developer to add manifest properties in applications and services. Related subtasks: T61A023 Manifest reading, T61A008 Structure the Build Manifest.
- **How to add users on STRAPI.** The addition of users has to be done on STRAPI CMS. The basic information of the users has to be obtained from the universal ZDMP user management. Need to discuss with the relevant component team to see how to use their API for component's purpose. It might be also required that additional information about the user is also needed.
- **Speech recognition feature.** Speech-to-text & speech commands (for app features) & speech privacy feature for data privacy.
- **Define manifest model.** Define the necessary fields to be included in application manifest. Related subtasks: T61A023 Manifest reading, T61A008 Structure the Build Manifest.

Sprint 8 (May 25, 2020 – June 08, 2020)

- **Integrate the STRAPI with API gateway and gatekeeper.** This task comes under purview of integration with Authorization and authentication component of ZDMP.
- **Verify that User has separate user space and is able to produce advertisement content.** The users should not be able to access another user's content.
- **Create a User on STRAPI through Keycloak instance.** The purpose of this issue is to develop a working prototype for registering a User to STRAPI CMS through Keycloak. This will involve establishing communication between the processes.
- **Analyse integration with GitLab CI pipeline and with SAG's API Gateway.**
- **Download and test on local servers the OpenAPI Generator tool.** The OpenAPI Generator tool (<https://github.com/OpenAPITools/openapi-generator>) for the implementation of the client libraries generator has been downloaded, installed and tested on ALGO's local servers.

- **SDK – Automate generation of client libraries for different languages.** Autogenerate the necessary client libraries for JavaScript / TypeScript and other languages which partners need. Subtasks include:
 - Create the scripts necessary for the different languages
 - Include the scripts in a Gitlab CI pipeline
 - When all the files have been generated, separated by language, the pipeline should store it in an openly accessible location or maybe just all of them in a single Git repository, so people can always reach and use what libraries they need
 - Gather feedback from the parties that need the clients, if they need some other switches turned on or off in the script, or if they can work with the files well
- **Development Portal.** There should be a web page or portal where a developer could access all the different tools available. Even if they have different UIs, there should be a development portal with a brief description and links to those development apps.
- **Packaging scripts.** A tool to create a package and upload it to the marketplace.
- **User management for Advertisement Framework.** Keep the record of users and the apps which they have installed. So that users do not see advertisements for the apps which they already have.
- **SDK Commercial / Advertisement Framework.** An in-app advertisement framework for zApps.
- **Understand how to configure middleware in Strapi.** For establishing communication between Strapi and Keycloak processes, a middleware needs to be created on Strapi.
- **Telemetry-datastore mock-up service:** this service embeds a sample dataset provided by ASC with telemetry sample data; in final version this service will be replaced by a service with similar REST interface extracting real telemetry data from the ZDMP platform.
- **Telemetry-service first version:** working on a first version of REST service espousing endpoints to retrieve a range of possible views on telemetry data.
- **Telemetry-UI:** started designing angular application providing an UI for the visualization of charts and views provided by the telemetry-service.

2.2.4 Limitations

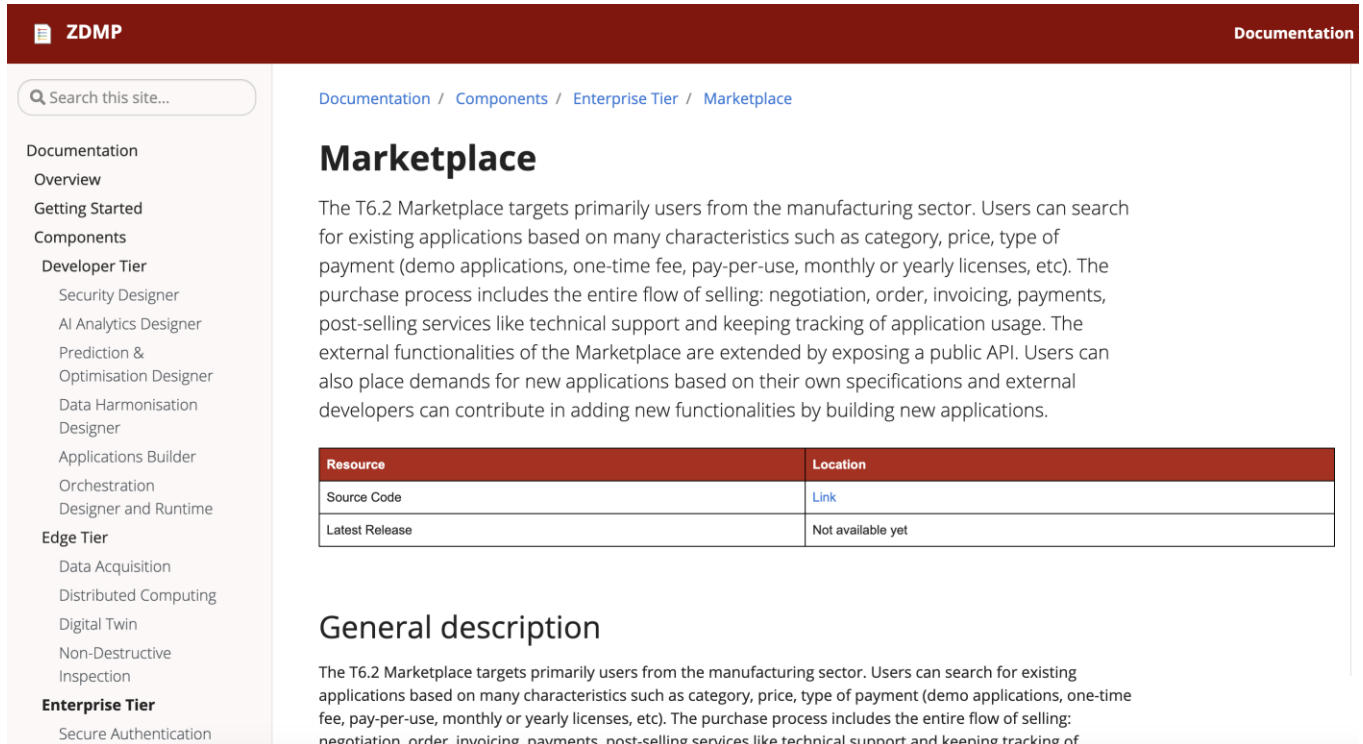
There were no issues found in this period which would limit the final scope of the activities expected.

3 T6.2 Marketplace

The Marketplace is the subcomponent that deals with users purchasing software license from developers, developers publishing and describing that software and generating profits from the software.

3.1 Public Documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/enterprise-tier/marketplace/>



The screenshot shows the ZDMP Documentation website. The header includes the ZDMP logo and a 'Documentation' link. A search bar is present. The left sidebar lists the documentation structure: Overview, Getting Started, Components, Developer Tier (Security Designer, AI Analytics Designer, Prediction & Optimisation Designer, Data Harmonisation Designer, Applications Builder, Orchestration Designer and Runtime), Edge Tier (Data Acquisition, Distributed Computing, Digital Twin, Non-Destructive Inspection), and Enterprise Tier (Secure Authentication). The main content area is titled 'Marketplace' and includes a description of the T6.2 Marketplace targeting users from the manufacturing sector. It details the search capabilities, purchase process, and external functionalities. A table lists resources and their locations.

Resource	Location
Source Code	Link
Latest Release	Not available yet

General description

The T6.2 Marketplace targets primarily users from the manufacturing sector. Users can search for existing applications based on many characteristics such as category, price, type of payment (demo applications, one-time fee, pay-per-use, monthly or yearly licenses, etc). The purchase process includes the entire flow of selling: negotiation, order, invoicing, payments, post-selling services like technical support and keeping tracking of application usage. The external functionalities of the Marketplace are extended by exposing a public API. Users can also place demands for new applications based on their own specifications and external developers can contribute in adding new functionalities by building new applications.

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	Source code can be found here: https://zdmp-gitlab.ascora.eu/zdmp_code/enterprise-tier/t6.2-marketplace
Latest release:	Commercial	Marketplace is one of the few components in ZDMP which is not open source. As the whole source is not available, the component cannot build in a shared continuous integration system, even though the open source parts are checked in. A built artifact though can be access at https://zdmp-marketplace.ascora.de
Open API Spec:	Available	N/A
General Description:	Available	N/A
Features:	Available	N/A
Requirements:	Available	N/A
Installation:	Available	N/A
How to Use:	In process	Resolving final queries and pending review comments

3.2 M18 Report

3.2.1 Architecture Implementation

The current prototype implementation of the Marketplace features a web frontend and backend. The backend provides a basic interface to be able to setup basic zApps with a description, category, icon, and images. The frontend allows discovery of the existing zApps, by list, by category and using a search function. The integration with the Storage and the Secure Installation components is still missing, therefore storing, and downloading components is not yet possible. The shopping cart is already functional, although financially backed contracts with the different payment providers are still necessary to allow actual use of PayPal, VISA, etc.

The main building blocks for the Marketplace component are defined in the final version of the general ZDMP architecture but updated with new subcomponents.

The figure below shows the status of the architecture implementation:

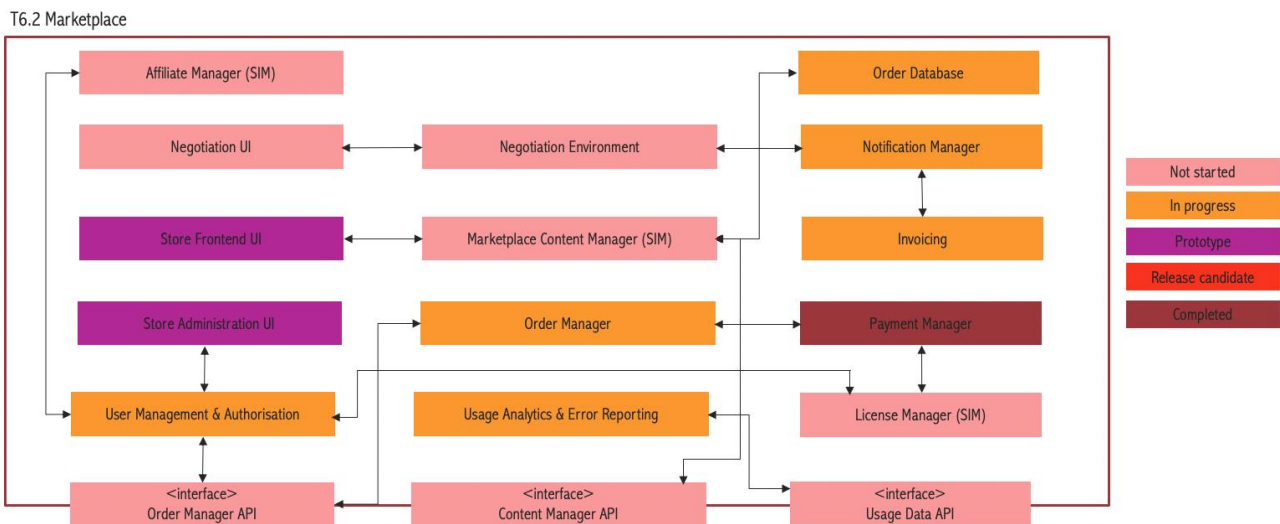


Figure 4: Marketplace Architecture Coverage

3.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed]** Frontend Changes towards ZDMP identity
- **[Completed]** Shopping Cart
- **[Completed]** Rating zApps
- **[Completed]** Research proposal system
- **[On going]** Choose / Explore Categories
- **[On going]** Explore Marketplace
- **[On going]** Marketplace Search
- **[On going]** Filter Marketplace View
- **[On going]** API GET and POST method definition
- **[On going]** Subscription payments (including first prototype)
- **[Not started]** Include new user management from T5.2
- **[Not started]** Notify user group IT admin that new license is available (Scenario task)
- **[Not started]** Bug Reporting Integration into Marketplace / Portal

- **[Not started]** Bug Reporting Setup
- **[Not started]** License Manager & API
- **[Not started]** Product Data Info Manager & UI
- **[Not started]** Affiliate Manager & UI (& related tasks)
- **[Not started]** Pay-Per-Use payments
- **[Not started]** Collaborating with functionally different marketplaces
- **[Not started]** Federation Feature / Issuing license information to other instances / Forwarding requests to other instances
- **[Not started]** Reviewing zApps

3.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 0 (Feb 02,2020 – Feb 16, 2020)
<ul style="list-style-type: none"> • Define top 10 features. Define top 10 features planned for the M18 milestone.
Sprint 1 (Feb 17, 2020 – March 01, 2020)
<ul style="list-style-type: none"> • Frontend Changes towards ZDMP identity. Created a basic frontend that contains a list of dummy products and menu bar to switch between various categories. Related Subtasks: T62A003 Search items.
Sprint 2 (March 02, 2020 – March 16, 2020)
<ul style="list-style-type: none"> • Shopping Cart. User can select the product which he wants to purchase and add them in shopping cart. Related Subtasks: T62A005 Place Order. • Accept payment. Added checkout for the purchased products. Accepted payment methods included credit card, debit card and PayPal. Related Subtasks: T62A005 Place Order.
Sprint 3 (March 17, 2020 – March 30, 2020)
<ul style="list-style-type: none"> • Rating zApp. Buyer can add a star rating and a comment for the products bought by him. Related Subtasks: T62A002 User Negotiation.
Sprint 4 (March 31, 2020 – April 13, 2020)
<ul style="list-style-type: none"> • Add an option for product category. Added an option to add a category label to product. Related Subtasks: T62A003 Search items, T62A004 Upload items. • Filter products by category. User can filter the products by category. Related Subtasks: T62A003 Search items.
Sprint 5 (April 14, 2020 – April 27, 2020)
<ul style="list-style-type: none"> • Add a search bar: Added a search bar on UI to enable use to search for a specific product. Related Subtasks: T62A003 Search items. • Search by product title: Product can be searched by name / title. Related Subtasks: T62A003 Search items. • Search by Category: Products can be searched by category. Related Subtasks: T62A003 Search items.
Sprint 6 (April 28, 2020 – May 11, 2020)
<ul style="list-style-type: none"> • Subscription Option. Added a subscription option to pricing models while listing products on marketplace. Related Subtask: T62A008 Payment, T62A006 Issue invoice.
Sprint 7 (May 12, 2020 – May 25, 2020)

- **Setup an instance of Keycloak Server.** Identify products per user subscription. Related Subtasks: T62A001 Connect to Marketplace.

Sprint 8 (May 25, 2020 – June 08, 2020)

- **Integrate the STRAPI with API gateway and gatekeeper.** Allow user to download subscribed products. Related Subtasks: T62A001 Connect to Marketplace.

3.2.4 Limitations

There were no issues found in this period which would limit the final scope of the activities expected.

4 T6.2 Storage

4.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/enterprise-tier/storage/>

ZDMP Documentation

Search this site...

Documentation
Overview
Getting Started
Components
Developer Tier
Security Designer
AI Analytics Designer
Prediction & Optimisation Designer
Data Harmonisation Designer
Applications Builder
Orchestration Designer and Runtime
Edge Tier
Data Acquisition
Distributed Computing
Digital Twin
Non-Destructive Inspection
Enterprise Tier
Secure Authentication - Authorisation
Secure Communication
Application Runtime
Inter-platform Interoperability

Documentation / Components / Enterprise Tier / Storage

Storage

A short lead description about this content page. It can be **bold** or *italic* and can be split over multiple paragraphs.

Resource	Location
Source Code	Link
Latest Release (v0.1.4)	Download
X Open API Spec	Link

General description

Proposed technical solution

As API gateway for general storage - files from general filesystems (i.e. OS, Hadoop, FTP), databases - Apache Knox was selected as backbone solution, and in addition with Apache Commons-VFS as general library we will develop special gateways based on ZDMP project needs.

Proposed API

Here are first draft proposal for Storage API.

Cleanup directory /user/user

Gen
Prop
Prop
Cle
Reg
REA
Upl
/usr
List
/usr
Rec
file
Rea
Feat
Requ
Insta
How
Form
final
Cc
,
An
T
Pl
Cc
Ex
Func

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	https://zdmp-gitlab.ascora.eu/zdmp_code/enterprise-tier/t6.2-storage/-/tree/master/
Latest release:	Available	Vs 1.0.0
Open API Spec:	Available	N/A
General Description:	Available	N/A
Features:	Available	N/A
Requirements:	Available	N/A
Installation:	Available	Using docker Image
How to Use:	In process	Resolving final queries and pending review comments

4.2 M18 Report

4.2.1 Architecture Implementation

As API gateway for general storage - files from general filesystems, the Storage component enables the saving of data on client- or ZDMP infrastructure. As this includes OS, Hadoop, FTP, databases, Apache Knox was selected as backbone solution because

it can cope with the necessary requirements, and in addition with Apache Commons-VFS as general library special gateways are developed based on ZDMP project needs.

The figure below shows the status of the architecture implementation:

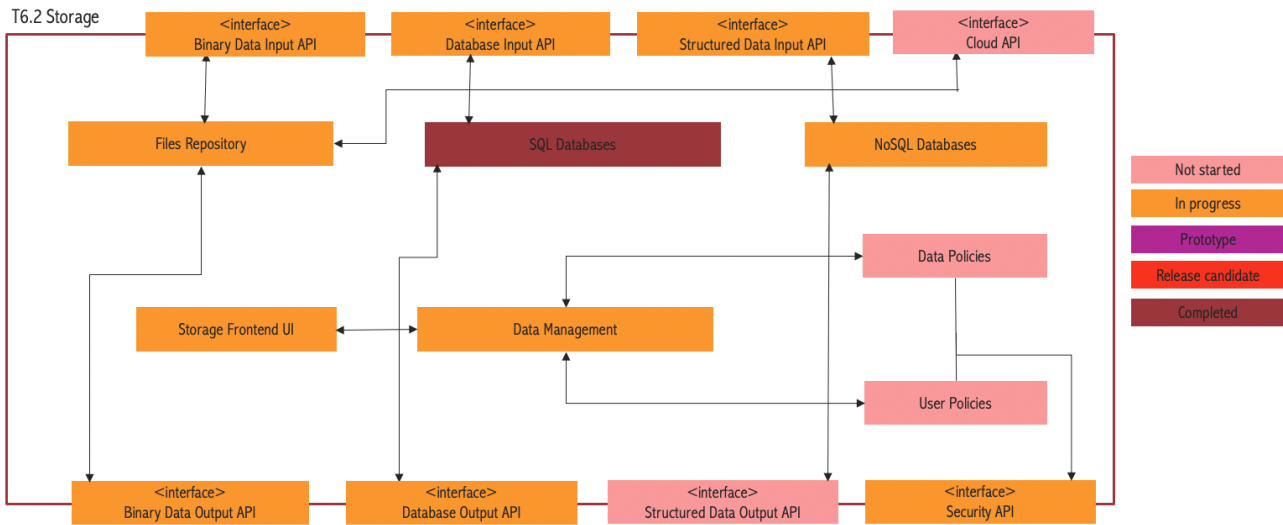


Figure 5: Architecture Coverage for Storage

4.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed]** Databases and tables functionalities for SQL databases
- **[On going]** Implement 3rd party Identity management and connect to security components (from T5.2)
- **[On going]** Configure SQL databases for specific ZDMP components
- **[On going]** Configure storage for ZDMP images
- **[On going]** Configure storage for ZDMP video (massive data)
- **[Not started]** Insert functionalities for SQL databases
- **[Not started]** Update functionalities for SQL databases
- **[Not started]** Delete functionalities for SQL databases
- **[Not started]** Select functionalities for SQL databases
- **[Not started]** Container Registry Management (Docker Images)

4.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 0 (Feb 3, 2020 – Feb 17, 2020)
<ul style="list-style-type: none"> • Review storage requirements • Prepare list for the research OTS tools and libraries
Sprint 1 (Feb 17, 2020 – Mar 2, 2020)
<ul style="list-style-type: none"> • Apache Hadoop research. In context of general files requirements for project • Apache Cassandra and CouchDB research. In context of general databases requirements for project
Sprint 2 (Mar 2, 2020 – Mar 16, 2020)
<ul style="list-style-type: none"> • Apache Hadoop HBase research. In context of general databases requirements for project • Apache Hadoop Distributed File System research. In context of general files requirements for project

Sprint 3 (Mar 16, 2020 – Mar 30, 2020)
<ul style="list-style-type: none"> • Investigate Docker and Docker composer requirements for ZDMP components • Run Docker and Docker composer files for conflict detection (in progress)
Sprint 4 (Mar 30, 2020 – Apr 13, 2020)
<ul style="list-style-type: none"> • Apache Knox research. In context of general API requirements for project • Run Docker and Docker composer files for conflict detection
Sprint 5 (Apr 13, 2020 – Apr 27, 2020)
<ul style="list-style-type: none"> • Apache Commons VFS research. In context of general files requirements for project and possible extensibility. Related subtasks: T62B001 Connect to file repository, T62B007 Send binary data. • PostgreSQL research regarding compatibility with Apache Knox. Related subtasks: T62B002 Connect to SQL database, T62B005 Receive SQL command.
Sprint 6 (Apr 27, 2020 – May 11, 2020)
<ul style="list-style-type: none"> • Apache Knox research for security integration. In context of security of the project. Related subtasks: T62B001 Connect to file repository. • MySQL research regarding compatibility with Apache Knox. Related subtasks: T62B002 Connect to SQL database, T62B005 Receive SQL command, T62B008 Send database records.
Sprint 7 (May 11, 2020 – May 25, 2020)
<ul style="list-style-type: none"> • Investigate ZDMP requirements for Storage components. In order to fulfil common API storage gateway for binary or text files streams, interaction with SQL and NoSQL databases. Apache Know was selected. Related subtasks: T62B001 Connect to file repository, T62B002 Connect to SQL database, T62B003 Connect to NoSQL database, T62B007 Send binary data, T62B005 Receive SQL command. • Research on Apache Knox and Apache Commons VFS capabilities and Dockerisation capabilities. As Knox only support HDFS as native file system supplementary analysis makes a common VFS as candidate for usage in ZDMP internal developed gateway.
Sprint 8 (May 25, 2020 – Jun 5, 2020)
<ul style="list-style-type: none"> • Implement Apache Knox POC. • Implement Apache Know VFS gateway POC. Related subtasks: T62B001 Connect to file repository, T62B002 Connect to SQL database, T62B004 Receive file command, T62B007 Send binary data, T62B005 Receive SQL command.

4.2.4 Limitations

There were no issues found in this period which would limit the final scope of the activities expected.

5 T6.2 Security Designer

5.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component, the description is available at: <https://software.zdmp.eu/docs/components/developer-tier/security-designer/>

The screenshot shows the ZDMP documentation website. The header is dark red with 'ZDMP' on the left, 'Documentation' in the center, and a search bar on the right. A left sidebar contains a navigation menu with categories like 'Documentation', 'Overview', 'Getting Started', 'Components', and 'Developer Tier'. The 'Developer Tier' section is expanded, showing 'Security Designer' as the selected item. The main content area has a breadcrumb trail: 'Documentation / Components / Developer Tier / Security Designer'. The title 'Security Designer' is prominently displayed. Below it, a paragraph describes the tool as a cyber-security risk modelling tool. A table lists resources: 'Source Code' (Not available), 'Latest Release' (Link), and 'X Open API Spec' (Link). To the right, a 'General description' sidebar lists links for 'Features', 'Requirements', 'Installation', 'How to use', 'Functional Requirements', and 'M18 Status'. A red-bordered box contains a detailed description of the Security Designer (SD) tool.

Security Designer

The Security Designer is a cyber-security risk modelling tool. It enables the user to model a socio-technical system, identifies the threats to the system following the web of attack paths, proposes mitigations drawn from a detailed knowledgebase, and computes the risks from the threat likelihood and business impact. The software supports two standards: ISO 27001 and ISO 27005. ISO 27001 is about how organisations should manage information security risks. ISO 27005 is an asset-based risk analysis procedure, which is generally regarded as the best way to find and understand risks.

Resource	Location
Source Code	Not available (proprietary)
Latest Release	Link
X Open API Spec	Link

General description

The Security Designer (SD) enables automated and systematic identification of risks to the assets (both human and technological) contained or connected to the ZDMP platform. SD also allows identifying the knock-on consequences and countermeasures to mitigate these risks. The tool allows collaboration between several stakeholders to develop the system model and the associated risk catalogue.

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Not available	The source code is proprietary, the release is a docker container with a compose file at: https://zdmp-gitlab.ascora.eu/zdmp_code/developer-tier/t6.2-security-designer/container_registry
Latest release:	Available	Vs 2.00
Open API Spec:	Available	Will be updated in M20
General Description:	Available	N/A
Features:	Available	N/A
Requirements:	Available	Will be updated in M20
Installation:	Pending	Will be updated in M20
How to Use:	Pending	Will be updated in M20

5.2 M18 Report

5.2.1 Architecture Implementation

The Security Designer (SD) enables automated and systematic identification of risks to the assets (both human and technological) contained or connected to the ZDMP platform. SD also allows identifying the knock-on consequences of threats and the possible countermeasures that can be employed to mitigate these risks. The tool allows collaboration between several stakeholders to develop the system model.

The first version of the Security Designer includes much of the required core functionality. Improvements to the usability of the user interface have been made (with a view to it being used by less-experienced users) and integration with ZDMP has begun (for example, single sign-on and interfacing with other components to share data).

The figure below shows the status of the architecture implementation:

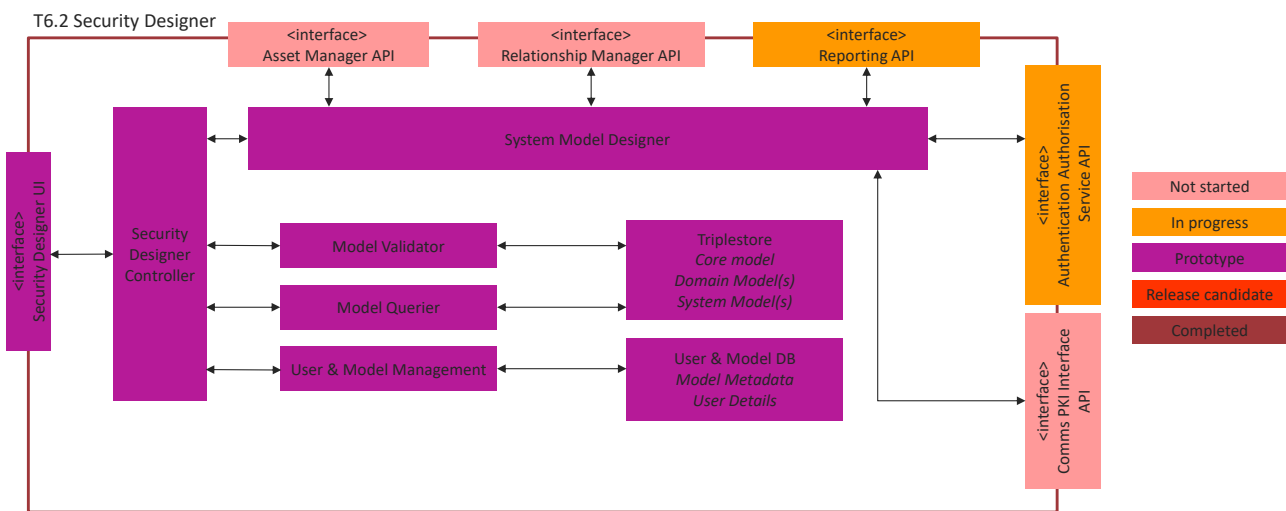


Figure 6: Security Designer Architecture

5.2.2 Planned Activities

A great deal of work has been done on the Security Designer in the period, but most tasks have not yet been completed. There are complex interdependencies between these tasks, and some, such as replacing the existing authentication and authorisation mechanism with KeyCloak and improving the model construction interface, have taken much more effort than anticipated. Nevertheless, there is a capable working prototype.

These were the planned activities for this period with an overview of their status:

- **[Complete] Package Security Designer for deployment on ZDMP platform.** Can generate a Docker image of Security designer.
- **[Ongoing] Provide user login form.** The software has an existing user management system which is being replaced with one backed by KeyCloak in order to be able to be integrated with Enterprise authentication systems and SSO systems.
- **[Ongoing] Provide registration form for new users.** The user fills in the details and submits the Registration form. Once the form has been validated, the user will receive a message "Registration pending confirmation". The existing system is being changed to integrate with KeyCloak.

- **[Ongoing] List models.** A list of existing models is provided so that a user can choose which one to open. The list of models exists but usability is poor. The usability is being improved by making it appear more as a list of files with last-modified date, etc.
- **[Ongoing, 50%] Generate security report.** This function generates a detailed description of the threats and risks to the system. Human- and machine-readable reports will be provided. The format and content the reports are in development.
- **[Ongoing] Risk calculations.** After validating the model that includes potential threats and asset misbehaviours, it is possible to calculate risk levels for them. The risk levels depend on the Trustworthiness Attributes for assets (which determine the likelihood of threats), and the Impact level settings for misbehaviours.
- **[Ongoing] Obtaining information from ZDMP components.** This information is used by security expert for constructing and refining the security model. This information may include the following: Security controls of components and zApps, logical/physical connections between components, deployment of components, authentication/authorization of users, security settings of network topology. The progress on this task depends strongly on the maturity of other components. UOSITI is actively collaborating with partners in order to finalise the integration of Security Designer and other ZDMP components.
- **[Ongoing] Model construction.** Selecting and adding assets to the canvas, adding relationships between assets, deleting assets/relations, and renaming assets. The functionality works but usability is poor. The usability is being improved. This includes a major sub-task on grouping assets.
- **[Ongoing] Threat management.** Threats may be resolved by selecting one or more controls. There are various panels provided to help show the user where to best place controls. The usability needs to be improved.
- **[Ongoing] Develop exemplar system models for common factory use cases.** Developed and already presented several models of industrial use cases involving data analysis and monitoring of machining data. It is planned to develop new models for selected ZDMP use cases.
- **[Ongoing] Build system models from IaC descriptors.** To simplify the model construction process, are analysing the various infrastructure as code (IaC) and configuration management tools to see what information pertinent to the models can be extracted.
- **[Ongoing] General user interface improvements.** The Security Designer is a complex piece of software and providing an understandable interface to users is challenging. The interface is often improved in small ways which do not merit an individual task heading here.

5.2.3 Progress

The software development team at IT Innovation work according to an internal process to deliver features for many projects according to the projects' milestones. The Security Designer code is proprietary and is therefore not present in the ZDMP GitLab. Some issues are raised in the ZDMP GitLab for the purpose of communication with the ZDMP partners, but IT Innovation use an internal GitLab to keep the proprietary code in various feature, bug-fix and release branches, perform detailed issue tracking, record discussions, and raise merge requests. IT Innovation's internal GitLab uses an in-house Continuous Integration system to test the code, record quality metrics, and build Docker images for deployment.

The TM will evaluate at M19 how the development of this component can be better brought under the standard ZDMP process including the better use of ZDMP sprints and their timing.

5.2.4 Limitations

There were no issues found in this period which would limit the final scope of the activities expected.

6 T6.3 Human Collaboration Environment

6.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/enterprise-tier/human-collaboration/>

The screenshot shows the ZDMP documentation website. The header is dark red with 'ZDMP' on the left and 'Documentation' and a search bar on the right. The left sidebar contains a navigation menu with categories like Documentation, Getting Started, Components, Developer Tier, Edge Tier, and Enterprise Tier. The main content area is titled 'Human Collaboration' and includes a description of the mobile app's purpose, a table of resources, a general description, and a list of features.

Human Collaboration

Human Collaboration mobile app purpose is to interconnect users and functionalities in order to create a synergy of efficiency regarding the workflow - communication between users (Video Calls and Tikki System), task scheduling and task completion.

Resource	Location
Source Code	Link
Latest Release	Link
X Open API Spec	Link

General description

The mobile application purpose is to interconnect users and functionalities in order to create a synergy of efficiency regarding the workflow - communication between users (Video Calls and Tikki System), task scheduling and task completion.

Features

- See online users / make video calls between users
- Schedule and assign tasks through Firebase technology for each user.
- User options for task completion/cancellation assigned through Firebase.
- A very simplified and efficient mail client that lets users the possibility to pass each other useful data/informations/notifications that need to remain stored.

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	https://zdmp-gitlab.ascora.eu/zdmp_code/enterprise-tier/t6.3-human-collaboration/-/tree/master/subsystems
Latest release:	Available	Vs 1.0.0
Open API Spec:	Available	N/A
General Description:	Available	N/A
Features:	Available	N/A
Requirements:	Available	Will be updated in M20
Installation:	Available	N/A
How to Use:	In process	Resolving final queries and pending review comments

6.2 M18 Report

6.2.1 Architecture Implementation

The Human Collaboration Environment (HCE) component provides collaboration facilities to enhance teamwork by offering user-user, user-developer, and developer-developer interaction using multiple communication channels enabling diverse ways of collaboration. The HCE also aims to ease the relationship between human users and manufacturing assets by providing holistic information and usage description in digital format. The collaboration environment may also include external developers and users who can request services and use communications platforms such as audio and video streaming, forums, workshops, and solutions to support hackathons.

The figure below shows the status of the architecture implementation:

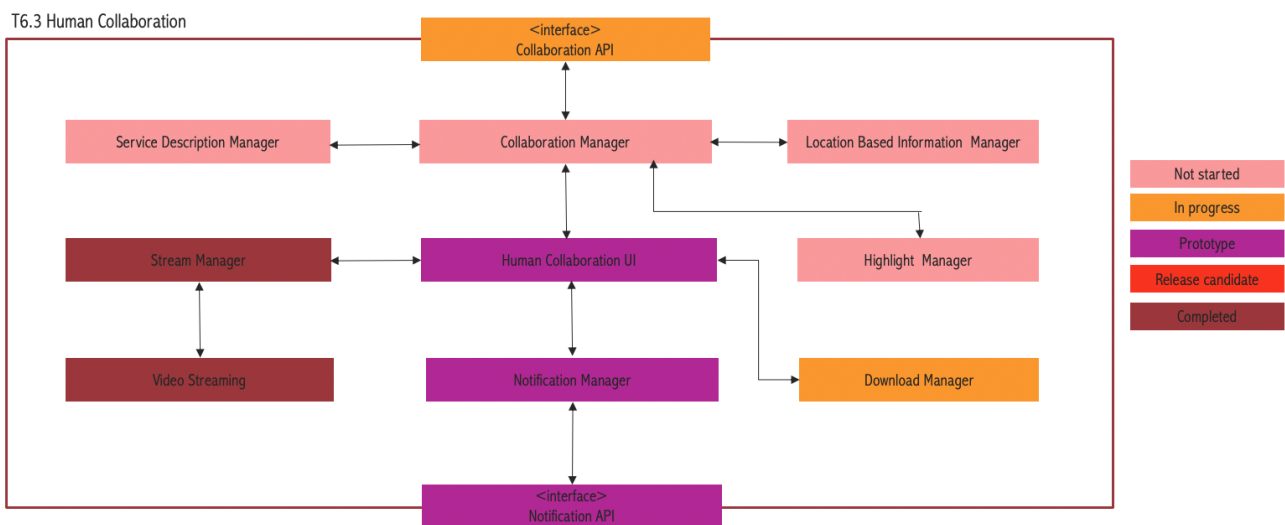


Figure 7: Human Collaboration architecture implementation status

6.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed] Analysis of tools for task implementation:** An analysis of available information to devise a set of tools for the implementation of all the needs arising in terms of human collaboration from the various parts of the project.
- **[Completed] Create a video call:** Using the Human Collaboration mobile, a user can initiate a video call. This feature will be integrated after M18 with the Human Collaboration Web app.
- **[Completed] Join a Video Call:** Using the Human Collaboration mobile, a user can join a video call. This feature will be integrated after M18 with the Human Collaboration Web app.
- **[Ongoing] Notification UI:** This is the user interface of Human Collaboration Mobile App for receiving and sending messages, using Tikki mail.
- **[Ongoing] Setup query and answer site:** One of the tools selected for implementation of human collaboration environment is a stack overflow like query and answer platform, supporting features like searching, reputation mechanisms etc

- **[Ongoing] Setup code collaboration site:** One of the tools selected for an implementation of the human collaboration environment is a code collaboration tool. vf-OS engagement hub was suggested as possible implementation for such a tool, which is based on GitLab ci plus specific plugins.

6.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 1 (Feb 3, 2020 – Feb 17, 2020)

- **Analysis of available material:** all available sources (DOA, other tasks deliverables, plenary discussions, meeting minutes, information concerning open calls etc.) have been examined for the extraction of relevant information for what could be a range of functionality needed to support the goal of establishing a human collaboration environment.
- **Create the Unity project of Human Collaboration Mobile app:** start a Unity based project of Human Collaboration, targeting multiple platforms (mobile as primary target, both Android and iOS, but also can be easily migrated to desktop platforms - Windows, Linux)

Sprint 2 (Feb 17, 2020 – Mar 2, 2020)

- **Human Collaboration Mobile app - Video Streaming Engine:** Develop the possibility for users to interact by using real time video-audio communication, showing online users in a dynamic list, real time almost updated on user entry/exit, based on Google Firebase library. Improve video streaming based on network capabilities. Related subtasks: T63A009 Join video conference.
- **Identification of set of tools:** a set of tools as per the previous action was identified including: a GitLab-based code collaboration platform, a query and answer platform (Stack Overflow-like), a virtual academy supporting portal (for making video lessons and documentation available), a discussion forum, a platform for hackathon support, besides the already video communication tools produced by task leaders. Technical constraints were examined for the working of the set of tools. Related subtasks: T63A002 Search content, T63A004 Send information, T63A005 Receive information.

Sprint 3 (Mar 2, 2020 – Mar 16, 2020)

- **Contact vf-OS for access to information concerning vf-OS engagement hub:** vf-OS engagement hub was proposed during the plenary as a choice for the implementation of a code collaboration tool. Vf-OS engagement hub is based on GitLab with supporting plugin. Information from vf-OS team was collected, including the online documentation,
- **Analysis of vf-OS engagement hub available documentation:** available documentation received from vf-OS team (linking to hub online help file) together with publicly available information on GitHub were analysed. Related subtasks: T63A002 Search content, T63A003 Download content, T63A004 Send information, T63A005 Receive information.

Sprint 4 (Mar 16, 2020 – Mar 30, 2020)

- **Human Collaboration Mobile app - Video Streaming Engine research:** research for a multimedia application on mobile phone platform, boosting the audio and video application over the Internet with Wi-Fi/4G mobile phone to capture video and disseminate, implements the function of video capturing in terminal, decoding and encoding raw data, media playing, real-time streaming media transmission and so on. Related subtasks: T63A009 Join video conference, T63A008 Upload video.
- **Presentation of tool set proposal:** the identified set of tools was discussed with task and work package leader. code collaboration and query and answer tools selected as first two tools to implement besides the video collaboration tool produced by task leader.

Sprint 5 (Mar 30, 2020 – Apr 13, 2020)

- **Human Collaboration Mobile app - Video Call Module:** based on Video Streaming Engine, a real time video-audio communication, showing online users in a dynamic list, real time almost updated on user entry/exit. Related subtasks: T63A009 Join video conference, T63A008 Upload video.
- **Human Collaboration Mobile app - Firebase protocols research:** for deploying functionalities to the mobile application quickly using pre-packaged solutions using configurable.

Sprint 6 (Apr 13, 2020 – Apr 27, 2020)

- **Human Collaboration Mobile app - Task Notification System:** develop a useful task system, almost real time both for users and system administrators, based on Google Firebase library, where a supervisor can push tasks for a specific user, who can mark it as completed or cancelled. Related subtasks: T63A004 Send information, T63A005 Receive information.
- **Analysis of vf-OS engagement hub available documentation:** available documentation received from vf-OS team (links to engagement hub online help file) together with publicly available information on GitHub were analysed.
- **Installation of a local instance of GitLab:** GitLab ci environment setup of ALGO local Linux server, waiting to be transferred to a project deployment platform when will be available. Related subtasks: T63A002 Search content, T63A003 Download content, T63A004 Send information, T63A005 Receive information.
- **Discussions on merging local instance with project GitLab site:** discussions with task and WP leader about possibility of using the same GitLab site already used for internal project activities as basis for the human collaboration code collab tool. not clear the impact and risks this choice could imply on partners work. it was decided to momentarily setup a separate instance to stay on the safe side.

Sprint 7 (Apr 27, 2020 – May 11, 2020)

- **Selection of scoold platform:** scoold was selected as best platform for the implementation of this human collaboration tool. Related subtasks: T63A002 Search content, T63A003 Download content, T63A004 Send information, T63A005 Receive information.
- **Scheduling tasks using Firebase:** A system administrator can create and assign tasks for users with different types of priorities using Google Firebase console. The users will be notified about changes almost in real time thanks to Firebase library capabilities. Related subtasks: T63A004 Send information, T63A005 Receive information.

Sprint 8 (May 11, 2020 – May 25, 2020)

- **Tikki mail implementation:** develop a simple, very responsive mail client for a better, reliable communication between users, based on Google Firebase library. Related subtasks: T63A004 Send information, T63A005 Receive information.
- **Installation of scoold on local servers:** an instance of scoold installed on local servers at ALGO.
- **Analysis of scoold user interface customization:** one of the key requirements in the selection of tools was to be able to customize UI to adapt to common ZDMP style. Related subtasks: T63A002 Search content, T63A003 Download content, T63A004 Send information, T63A005 Receive information.

6.2.4 Limitations

There were no issues found in this period which would limit the final scope of the activities expected.

7 T6.4 Application Runtime

7.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/enterprise-tier/application-runtime/>

The screenshot shows the ZDMP documentation website. The top navigation bar includes the ZDMP logo, a 'Documentation' link, and a search bar. The breadcrumb trail indicates the path: Documentation / Components / Enterprise Tier / Application Runtime. The main heading is 'Application Runtime'. Below it is a table with two columns: 'Resource' and 'Location'. The table lists three resources: 'Source Code' with a 'Link', 'Latest Release (v0.1.4)' with a 'Download', and 'X Open API Spec' with a 'Link'. Below the table is a section titled 'General description' which contains a text box stating: 'The Application Run-time itself provides both the capability to orchestrate and host the individual components of the system using containerisation. It includes a range of sub components to provide baseline functionality of the platform.'

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	Available code at: https://zdmp-gitlab.ascora.eu/zdmp_code/enterprise-tier/t6.4-application-run-time Rancher source code is available on the Rancher website.
Latest release:	Available	Vs 1.0.0
Open API Spec:	In progress	N/A
General Description:	Available	N/A
Features:	Available	Will be updated at a later stage if necessary
Requirements:	Available	Will be updated at a later stage if necessary
Installation:	In progress	Platform infrastructure required for M18
How to Use:	In progress	Documentation in progress, more is planned for >M18

7.2 M18 Report

7.2.1 Architecture Implementation

The Application Runtime represents the platform aspect of ZDMP. It contains a service mesh and API connectivity to monitor and orchestrate deployed apps within the platform. The component is built on open source technologies, Rancher, Kubernetes, Istio, and Docker and can allow the for the orchestration and the underlying communication between a complex mesh of services to be built, monitored, and secured.

The following figure shows the status of the architecture implementation:

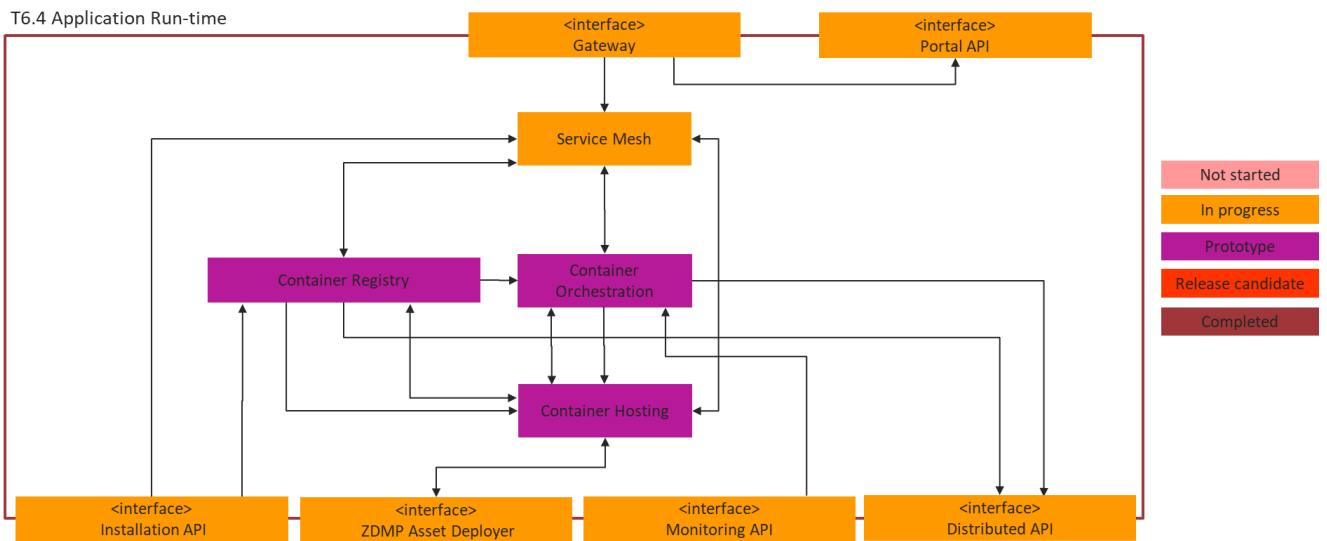


Figure 8: Application Run-time Architecture Coverage

7.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed] Prepare URLs to docker instances for API Gateway:** Definition of URL and port to expose endpoints.
- **[Completed] Custom Helm Charts Questions for User:** Repository structure of custom Helm charts.
- **[Completed] Rancher, integration of deployed apps with helm:** Basic integration of docker registry images, Kubernetes, helm Charts and application runtime.
- **[Ongoing] Application Runtime Integration with other services:** Provide information to link other services.
- **[Ongoing] Basic integration of the Application Runtime with the Services API Management:** Integration of Application Runtime and API gateway.
- **[Ongoing] Application Runtime documentation:** Creation of a document explaining the functionality of the application runtime.
- **[Ongoing] Define required information for manifest and either docker-compose or Kubernetes yaml file:** Design of templates for Kubernetes and Helm Charts for custom installation of apps.
- **[Ongoing] zApp/Component install from Gitlab:** Installation of apps from Gitlab docker registry. Modified to get them from app storage.

- **[Ongoing] Integration Application Runtime and Portal:** Provide APIs/Interface to link application runtime and portal.
- **[Ongoing] Deploy Cluster and Portal:** Deploy Kubernetes cluster and Rancher on the platform infrastructure.

7.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 0 (Jan 20, 2020 – Feb 3, 2020)
<ul style="list-style-type: none"> • Define top 10 features for M18: Definition of the planned features for the M18 milestone
Sprint 1 (Feb 3, 2020 – Feb 17, 2020)
<ul style="list-style-type: none"> • Add Functional Specs to reporting folder: Part of the initial repository setup. Related Subtasks: T64C05 Networking a zApp. • Create repository structure: Part of the initial repository setup. • CI Docker images: Generate CI file to generate and save images in GitHub container registry. Related Subtasks: T64C06 Instantiating a zApp. • CI runner deployment: Generate CI runner locally to execute CI file. Related Subtasks: T64C06 Instantiating a zApp. • GitHub and Docker helm examples: Test examples using GitHub with no authentication. Related Subtasks: T64C01 install a new component or zApp.
Sprint 2 (Feb 17, 2020 – Mar 2, 2020)
<ul style="list-style-type: none"> • CI Gitlab container registry: Generate CI file to generate and save images in GitLab container registry. Related Subtasks: T64C06 Instantiating a zApp. • Create GitLab pages for helm charts: Preparation of helm charts and package using GitLab pages to display the available apps. Related Subtasks: T64C01 install a new component or zApp.
Sprint 3 (Mar 2, 2020 – Mar 16, 2020)
<ul style="list-style-type: none"> • Instructions for High Availability Cluster for Kubernetes: Prepare document to install a High availability cluster for Kubernetes. Related Subtasks: T64C05 Networking a zApp. • Specs for Server: Provide with specifications required to run a Kubernetes cluster. Related Subtasks: T64C05 Networking a zApp.
Sprint 4 (Mar 16, 2020 – Mar 30, 2020)
<ul style="list-style-type: none"> • Ask for an experimentation VM to Samuel and install Rancher: Request a server/PC to install a low resources Kubernetes version for testing on premises. Related Subtasks: T64C05 Networking a zApp.
Sprint 5 (Mar 30, 2020 – Apr 21, 2020)
<ul style="list-style-type: none"> • Versioning Helm Charts: Provide with structure of helm charts versioning. Related Subtasks: T64C03 manage component or zApp settings. • Generation of Secrets to keep GitLab Credentials: Generation of secret file to save credentials to allow download container repository images. Related Subtasks: T64C01 install a new component or zApp. • Monitoring services with ELK Elastic Search Fluent and Kibana: Deploy ELK to monitor services and display logs. Related Subtasks: T64C02 manage installed components and zApps. • Link GitLab Helm charts and container registry: Test and deploy apps linking both, a private Gitlab container repository and private helm chart repository with Kubernetes. Related Subtasks: T64C06 Instantiating a zApp.
Sprint 6 (Apr 21, 2020 – Apr 27, 2020)
<ul style="list-style-type: none"> • Custom Helm Charts, questions for user: Customization of application templates. Related Subtasks: T64C02 manage installed components and zApps.
Sprint 7 (Apr 27, 2020 – May 11, 2020)
<ul style="list-style-type: none"> • Rancher, Integration of deployed apps with helm custom configuration.: Deployment of apps and integration using custom charts. Related Subtasks: T64C03 manage component or zApp settings.

Sprint 8 (May 11, 2020 – May 25, 2020)

- **Define required information for Manifest:** Information required for installation of applications.
Related Subtasks: T64C01 install a new component or zApp.

7.2.4 Limitations

At the moment all tests have been done locally, a platform infrastructure is required for testing and linking with other tasks.

8 T6.4 Portal

8.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/enterprise-tier/portal/>

ZDMP Documentation

Search this site...

Documentation / Components / Enterprise Tier / Portal

Portal

zDMP Portal Main Documentation page

Resource	Location
Source Code	Link
X Open API Spec	Link

General description

The portal is a website for access to the ZDMP functionality including, everything from human collaboration to container management depending on the user's role and permissions. It displays web UIs from the zApps and provides information to the decision makers that require it.

Features

- zDMP Global Dash Button
- User profile with roles
- Organisations/Users management

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	https://zdmp-gitlab.ascora.eu/zdmp_code/enterprise-tier/t6.4-portal
Latest release:	Available	Vs 0.8
Open API Spec:	Pending	Will be updated in M20
General Description:	Available	N/A
Features:	Available	N/A
Requirements:	Available	Will be updated in M20
Installation:	Available	Will be updated in M20
How to Use:	Available	Will be updated in M20

8.2 M18 Report

8.2.1 Architecture Implementation

The Portal component represents the frontend of the ZDMP. It has a front facing UI, backend, roles manager, user interface managers and a marketplace manager. To access all the key facets of ZDMP. It also has an API for programmatic access to the component and to allow interconnectivity between this and other components.

The following figure shows the status of the architecture implementation:

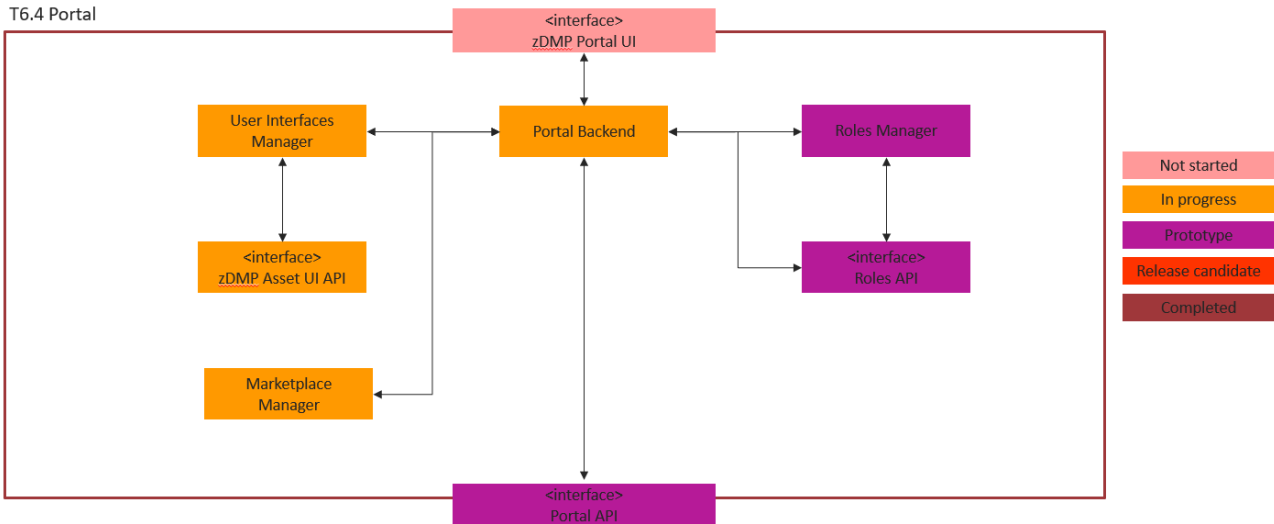


Figure 9: Architecture Coverage of Portal

8.2.2 Planned Activities

These were the planned activities for this period with an overview of their status.

- **[Completed] Login screen integrated with security component** This includes the KeyCloak integration provided by security component.
- **[Completed] Create Organisation** This allows adding new organisations in the system, and includes creating security realms using Security component API
- **[Ongoing] User profile with roles.** This screen allows a user to see/update their profile and their current roles, it works closely with T5.2 Security Component.
- **[Ongoing] Dash button - global ZDMP header web component** Leverages standard web component, to provide a common and reusable ZDMP header for all components
- **[Not Started] Pipelines and repositories configuration screen:** This section allows the user to configure the pipelines and repositories. Are analysing the UI requirements for the T6.4 Application Runtime, so can create the necessary screens for this step.
- **[Not Started] Service discovery configuration screen:** This task is to provide the user a list of services that can be added to a zApp. Since this works closely with both Marketplace and Application Runtime, are analysing the UI requirements to create the necessary screens for these operations.
- **[Not Started] Load balancing configuration screen.** This task allows the main UI for load balancing working closely with Application Runtime. Are analysing the UI requirements so can create the necessary screens for this section.
- **[Not Started] Install an APP frontend action** Allows a user to install an APP, works closely with Marketplace and Application Runtime, are analysing the UI requirements to create the necessary screens for these operations.

8.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 0 (Jan 20, 2020 – Feb 3, 2020)

- **Define top 10 features for M18:** Definition of the planned features for the M18 milestone.

Sprint 1 (Feb 3, 2020 – Feb 17, 2020)
<ul style="list-style-type: none"> • Initial analysis: Create initial repository setup, analysis of requirements and documentation, mockup screens planning and research of existing open source tools like Kubernetes UI, Rancher, Portainer, OpenShift.
Sprint 2 (Feb 17, 2020 – Mar 2, 2020)
<ul style="list-style-type: none"> • Add Functional Specs to reporting folder: review functional specifications document and add them to the reporting folder. • Lookup frontend template: research on open source frontend UI templates or other frameworks to be used as the template base of the UI. Related Subtasks: T64D06 Register new organisation, T64D07 Select organisation to login, T64D01 Manage users.
Sprint 3 (Mar 2, 2020 – Mar 16, 2020)
<ul style="list-style-type: none"> • Prototypical Web-UI with links to different ZDMP UIs: implement a proof of concept global dash button leveraging web components standard, review official documentation and ensure browser compatibility, package, and test publicly.
Sprint 4 (Mar 16, 2020 – Mar 30, 2020)
<ul style="list-style-type: none"> • Researching inter-linking with security components (T5.2): research integration with KeyCloak, implement authentication and local realm creation, review integration with security component T5.2 and create a new requirements plan. Related Subtasks: T64D01 Manage users.
Sprint 5 (Mar 30, 2020 – Apr 21, 2020)
<ul style="list-style-type: none"> • Outline basic structure of the portal: create new NodeJS application that is the portal API and interacts directly with T5.2 security component. Implement authentication bridge, create realm/organisation and query for multiple organisations. Related Subtasks: T64D06 Register new organisation, T64D07 Select organisation to login.
Sprint 6 (Apr 21, 2020 – Apr 27, 2020)
<ul style="list-style-type: none"> • Run Rancher UI from source code against Rancher API docker: research running locally Rancher UI from official source code and work against local Rancher API docker. Analyse extensibility options to ensure it is a valid option as the interaction with the application runtime and/or Kubernetes.
Sprint 7 (Apr 27, 2020 – May 11, 2020)
<ul style="list-style-type: none"> • Create basic dash button: extend the current proof of concept of the dash button and make it work with the Portal API. Ensure it follows the web components standard and is embeddable by common browsers and does not cause side issues with 3rd party frameworks.
Sprint 8 (May 11, 2020 – May 25, 2020)
<ul style="list-style-type: none"> • Embed dash button as header: implement a test application with embedded dash button and test its basic functionality. • Create organisation screen with mockup service: complete create organisation screen and test it calls correctly Portal API to send collected information. Related Subtasks: T64D06 Register new organisation, T64D07 Select organisation to login. • Create Angular skeleton app with routing: create the main Portal frontend application using Angular, setup routes and complete integration with Portal API. Related Subtasks: T64D06 Register new organisation, T64D07 Select organisation to login, T64D01 Manage users.

8.2.4 Limitations

Several not started activities had strong dependencies on other components which were not ready:

9 T6.4 Services and Message Bus

9.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/platform-tier/service-and-message-bus/>

ZDMP

Search this site...

Documentation / Components / Platform Tier / Service & Message Bus

Service & Message Bus

Resource	Location
Source Code	Link
Latest Release (v0.0.1)	Download
X Open API Spec	Link

General description [↗](#)

The T6.4 Services and Message Bus represents the communication layer of the ZDMP Platform. It provides the following main functions:

1. It acts as the central endpoint to all ZDMP APIs and allows the convenient management of the APIs through a WebUI. Thereby it automatically updates an API's service endpoint when installing a component on the platform and automatically handles user authentication and authorisation for the APIs. An illustration of the functionality is shown below.

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	N/A	The Services API Management is based on closed-source software from SAG thus no source code provided. The Message Bus is based on RabbitMQ and its source code is available on the RabbitMQ website.
Latest release:	Available	N/A
Open API Spec:	N/A	Currently there is no REST interface required to interact with the component. The Services API Management can be accessed via Web-UI, the Message Bus via standard software libraries
General Description:	Available	N/A
Features:	Available	Will be updated at a later stage if necessary
Requirements:	Available	Will be updated at a later stage if necessary
Installation:	Available	Docker compose file is planned for M18
How to Use:	Available	Documentation for the Services API Management subcomponent is already available. The Message Bus documentation is planned for >M18

9.2 M18 Report

9.2.1 Architecture Implementation

The first version of the Service and Message Bus component comprises prototypical implementations of the Services API Management, Message Bus, and Integration Server subcomponents.

- **Services API Management:** The prototype allows to expose and manage ZDMP component APIs. In combination with the API mock-up functionality, this enables the ZDMP partners to integrate their component APIs into the platform and simulate their behaviour for testing purposes
- **The Message Bus:** This prototype features the initial message-based communication between ZDMP components through the AMQP and MQTT protocols
- **Integration Server:** The Integration Server prototype provides an alternative way to REST APIs for the integration of ZDMP components

After M18, the Service and Message Bus subcomponents will be further integrated with the T5.2 Secure Authentication and Authorisation, T5.2 Secure Communication, and T6.4 Application Run-time components. This includes, for example, the extension of the Service and Message Bus API to support the management of users and access rules by the T5.2 Secure Authentication and Authorisation component.

The following figure shows the status of the architecture implementation:

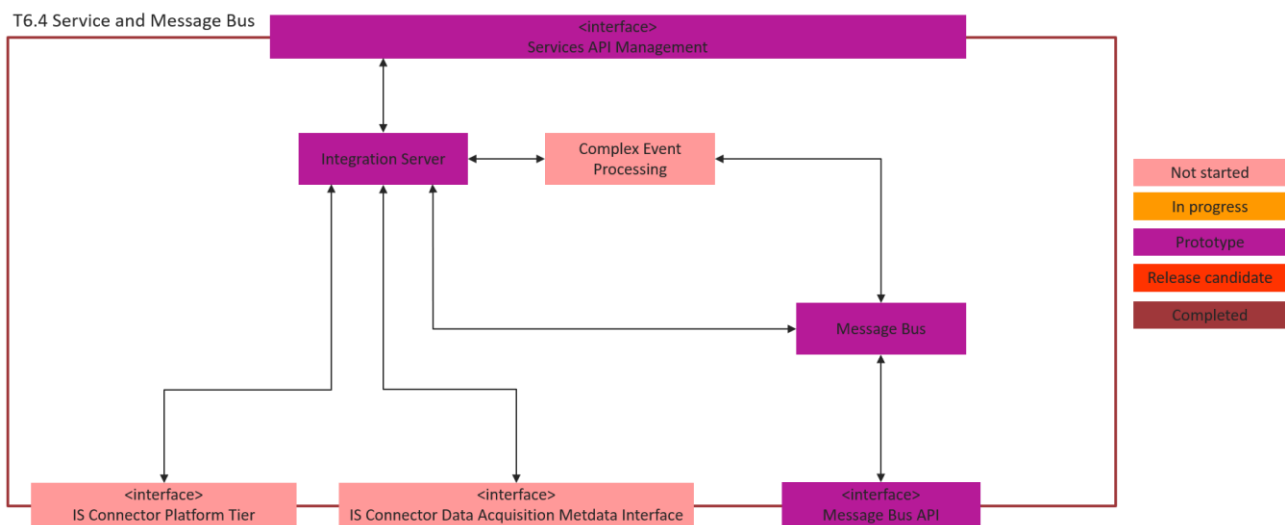


Figure 10: Architecture Coverage of Services and Message Bus

9.2.2 Planned Activities

These were the planned activities for this period:

- **[Completed] API Management UI for developers:** Provide developers with a user interface to expose their APIs via the Services API Management subcomponent
- **[Completed] Naming convention for API endpoints:** Definition of a naming convention for the API endpoints provided by the Services API Management subcomponent

- **[Completed] Services API Management tutorial for developers:** Creation of a user tutorial for ZDMP Developers explaining the functionality of the Services API Management
- **[Completed] Selection and setup of a message broker:** Selection and setup of a message broker for the implementation of the Message Bus component
- **[Completed] Naming convention for messaging channels/topics:** Definition of an initial naming convention for messaging channels/topics for the publish/subscribe functionality provided by the Message Bus
- **[Ongoing] Basic integration of the Services API Management with T5.2 Secure Authentication and Authorisation:** Basic implementation of the interface between T6.4 Services API Management and T5.2 Secure Authentication and Authorisation. This interface enables the T6.4 Services API Management subcomponent to integrate the Authentication and Authorisation mechanisms provided by T5.2 into the API Request workflow
- **[Ongoing] Basic integration of the Services API Management with the Application Runtime):** Basic implementation of the interface between the T6.4 Services API Management component and the T6.4 Application Run-time. This interface enables the T6.4 Application Run-time to update the native services endpoints of the APIs exposed via the T6.4 Services API Management subcomponent.

9.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 0 (Jan 20, 2020 – Feb 3, 2020)
<ul style="list-style-type: none"> • Define top 10 features for M18: Definition of the planned features for the M18 milestone. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls, T64A04 API logging, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs, T64B002 Message Bus Topic Management, T64B003 Publishing messages on topics and subscribing to topics.
Sprint 1 (Feb 3, 2020 – Feb 17, 2020)
<ul style="list-style-type: none"> • Add Functional Specs to reporting folder: Part of the initial repository setup. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls, T64A04 API logging, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs, T64B001 Message Bus access tokens, T64B002 Message Bus Topic Management, T64B003 Publishing messages on topics and subscribing to topics, T64C001 Optional integration of Enterprise Tier, Platform Tier, and Edge Tier components. • Create repository structure: Part of the initial repository setup. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls, T64A04 API logging, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs, T64B001 Message Bus access tokens, T64B002 Message Bus Topic Management, T64B003 Publishing messages on topics and subscribing to topics, T64C001 Optional integration of Enterprise Tier, Platform Tier, and Edge Tier components
Sprint 2 (Feb 17, 2020 – Mar 2, 2020)
<ul style="list-style-type: none"> • Test Dockerised API Gateway instance: Install and test the webMethods API Gateway docker container. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls, T64A04 API logging, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs
Sprint 3 (Mar 2, 2020 – Mar 16, 2020)
<ul style="list-style-type: none"> • Research on webMethods API Gateway for implementing the Services API Management component: Getting familiar with the functionality and configuration of webMethods API Gateway, which is being used to implement the Services API Management. Related Subtasks: T64A001

Management of APIs, T64A002 API access tokens, T64A003 API function calls, T64A04 API logging, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs

Sprint 4 (Mar 16, 2020 – Mar 30, 2020)

- **Setup message broker:** Set up and configuration of RabbitMQ on SAG development server via docker. Related Subtasks: T64B002 Message Bus Topic Management, T64B003 Publishing messages on topics and subscribing to topics
- **Services API Management tutorial:** Preparation of a tutorial for ZDMP Developers to explain them the functionality of the Services API Management. Related Subtasks: T64A001 Management of APIs, T64A003 API function calls, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs.
- **Preparation API policies:** Definition of global API policies that will be applied to all ZDMP component APIs stored in the Services API Management component. This includes, eg, routing- and security-related policies. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls

Sprint 5 (Mar 30, 2020 – Apr 21, 2020)

- **Prepare default API access rule:** Definition of a default access rule for APIs to provide an initial protection against unauthorised API access. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls.
- **Creation of API Groups:** Creation of API Groups to structure the different ZDMP APIs in the Services API Management by specific components. Related Subtasks: T64A001 Management of APIs, T64A003 API function calls
- **Naming convention for ZDMP APIs:** Definition of a naming convention for ZDMP component APIs. These names will be reflected in the different API endpoint URLs. Related Subtasks: T64A001 Management of APIs, T64A003 API function calls, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs.
- **Create user accounts for ZDMP Developers:** Create user accounts for ZDMP Developers with appropriate rights to create and manage ZDMP component APIs via the Services API Management. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls.

Sprint 6 (Apr 21, 2020 – Apr 27, 2020)

- **Scheduled Services API Management backups:** Implementation of scheduler to periodically run backups of the Services API Management configuration and the ZDMP APIs stored in the Services API Management. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls, T64A04 API logging, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs.
- **API Management Configuration Backup mechanism:** Implementation of a backup mechanism to back up the Services API Management configuration. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A04 API logging.
- **API Backup mechanism:** Implementation of a backup mechanism to back up the ZDMP APIs stored in the Services API Management. Related Subtasks: T64A001 Management of APIs, T64A002 API access tokens, T64A003 API function calls, T64A04 API logging, T64A005 API documentation, T64A006 (De-)Activation of APIs, T64A007 Versioning of APIs, T64A008 Mocking of APIs.

Sprint 7 (Apr 27, 2020 – May 11, 2020)

- **Test Message Bus WebSocket Connection:** Initial tests of the WebSocket connection provided by the Message Bus. Related Subtasks: T64B002 Message Bus Topic Management, T64B003 Publishing messages on topics and subscribing to topics.
- **Enable external access to Keycloak Admin Console:** Configure Nginx as a reverse proxy for T5.2 Authentication/Authorisation. Related Subtasks: T64A002 API access tokens
- **Install T5.2 Secure Authentication/Authorisation on SAG VM:** Local installation of the T5.2 Secure Authentication/Authorisation for integration tests. Related Subtasks: T64A002 API access tokens.

9.2.4 Limitations

There were no issues found in this period which would limit the final scope of the activities expected.

10 T6.5 Inter-Platform Interoperability

10.1 Public documentation

Public documentation is available as indicated in Section 1.3. Regarding this component the description is available at: <https://software.zdmp.eu/docs/components/enterprise-tier/inter-platform-interoperability/>

ZDMP Documentation

Search this site...

Documentation / Components / Enterprise Tier / Inter-platform Interoperability

Inter-platform Interoperability

Resource	Location
Source Code	Not yet available
Latest Release	Not yet available
X Open API Spec	Not yet available

General description

The Inter-platform Interoperability component's aims is to link the ZDMP platform with other external platforms. For example, it is supposed to support the ability to sell ZDMP services to other platforms and purchase other components from some those. It also includes a layer to link data sources from different platforms. Finally, it integrates security between the platforms to ensure appropriate access levels across the platforms. It includes a connection to other instances of ZDMP.

This component expects to link to the platform of the EU funded project, eFactory9, the ADAMOS commercial IIoT platform, and with partner UOS-ITI's and their platform SDAIM. Further platform linkages will be investigated as the project progresses. The component aims to manage three main concepts: Data, Marketplace, and Security. The eFactory project (a project about lot-size-one manufacturing in the same call as ZDMP) also aims to integrate four platforms, vf-OS, DIGICOR, NIMBLE and COMPOSITION. This component follows the design of eFactory and creates a plugin for each platform to be integrated so creating an amplifier effect.

CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Pending	Will be available >M18*
Latest release:	Pending	Will be available >M18*
Open API Spec:	Pending	Will be available >M18*
General Description:	Available	N/A
Features:	Pending	Will be available >M18*
Requirements:	Pending	Will be available >M18*
Installation:	Pending	Will be available >M18*
How to Use:	Pending	Will be available >M18*

* The T6.5 Inter-Platform Interoperability component strongly depends on other ZDMP components, which are not yet fully implemented. For that reason, there is no release version, source code, API specification, and guidelines available yet. Similarly, the feature set and requirements of the component are not yet defined.

10.2 M18 Report

10.2.1 Architecture Implementation

The T6.5 Inter-Platform Interoperability component strongly depends on the functionality provided by other ZDMP components and their interfaces. This includes, for example, the components developed in the Tasks:

- T5.1 Data Acquisition and IIoT
- T5.2 Robust Industrial Network Support
- T5.3 Data Harmonisation and Interoperability
- T6.1 SDK: Applications and Service Builder
- T6.2 Secure Business Cloud
- T6.4 Platform Integration and Federation

These components are not yet fully implemented, and their initial functionality and interfaces may change after first integration tests. For these reasons, prototypical implementations of T6.5 subcomponents are not yet available and the work on some subcomponents has not been started.

The following figure shows the status of the architecture implementation:

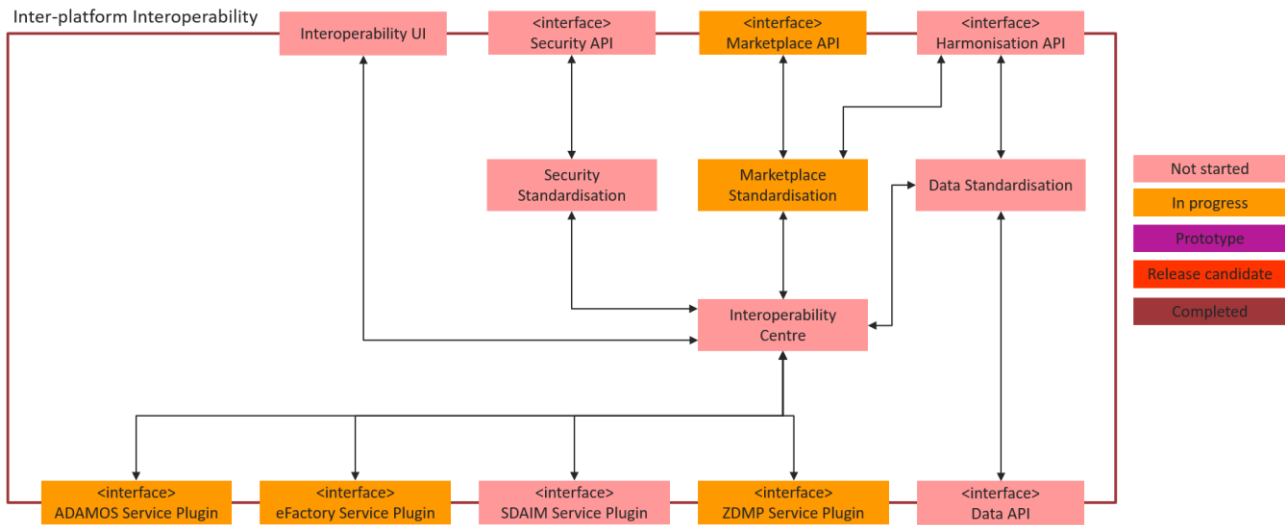


Figure 11: Architecture coverage of Inter-Platform Interoperability

Notably, some features of the T6.5 Inter-Platform Interoperability component have already been considered in the development of other ZDMP components such as the T6.2 Marketplace and the T6.4 Service and Message Bus. This includes, for example, the marketplace interoperability to eFactory and the integration of external platforms and multiple ZDMP instances through REST APIs and MQTT/AMQP messaging.

After M18, and as soon as the required ZDMP components and interfaces become available, the different standardisation subcomponents and the service plugins for the interconnection of ZDMP with eFactory, ADAMOS, SDAIM, and other ZDMP instances will be further developed.

10.2.2 Planned Activities

The planned activities for this period are listed below. Note that all activities are marked as ongoing because the T6.5 Inter-Platform Interoperability component strongly depends on other ZDMP components, which are currently under development and not yet available:

- **[Ongoing] Planning for interlinking multiple ZDMP instances:** Planning of the next steps for interlinking two or more ZDMP instances. This includes topics such as the interconnection of multiple Services and Message Bus components to provide API access to ZDMP resources running on different hosts
- **[Ongoing] Planning for interlinking ZDMP with ADAMOS:** Planning of the next steps for interlinking the ADAMOS IIoT platform with ZDMP. This includes topics

such as the interconnection of the Cumulocity instance used in the T5.1 Data Acquisition component with the Cumulocity backend of ADAMOS

- **[Ongoing] Planning for interlinking the ZDMP marketplace with the eFactory marketplace:** Planning of the next steps for interlinking ZDMP's marketplace with the eFactory marketplace. An interlink enables the cross-platform marketing of zApps. This milestone is strongly related to the work done in Task T6.2 Marketplace
- **[Ongoing] Planning for interlinking ZDMP with SDAIM:** Planning of the next steps for interlinking ZDMP with UOS-ITI's SDAIM framework for real-time data analytics
- **[Ongoing] Merging of permissions between two platforms:** Implement a module to map the ZDMP's security model with the model used by external platforms. Suggest design improvements from T5.2 to take a federated approach
- **[Ongoing] Merging of data pipelines:** Implement a module to map the ZDMP's data model with the model used by external platforms. Make sure that the T6.4 Services and Message Bus and the T6.4 Application Run-time considers a federated approach.
- **[Ongoing] Merging of application flow:** Implement a module to map the ZDMP's application flow with the model used by external platforms. Make sure the T6.4 Services API Management can manage federated architecture and connect to another version of itself on another running instance of ZDMP

10.2.3 Progress

See Section 2.2.3 for comments on the methodology.

As can be seen in the table below, the final “sprint” is more of a slow ‘jog’ lasting 3 months. As mentioned above, this task is completely dependent on the results of many others (all in their first phase) so most of the actions are in a kind-of “holding” pattern until more results are seen. Thus, there is no specific feature resolved during this time because of this, and all issues are still ongoing. Since the sprint reports only contain completed issues, all intermediate sprints would be empty. Since this task is so closely coupled with other tasks during this people some related issues were being worked on via them such as ZDMP-to-ZDMP, and ZDMP-to- eFactory interlinking via T6.2 and T6.4 where relevant preparatory work is being performed.

Sprint 0 (Jan 20, 2020 – Feb 3, 2020)

- **Define top 10 features for M18:** Definition of the planned features for the M18 milestone. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65B001 ZDMP and eFactory Marketplace Integration, T65C001 Exchange of device- and sub-device Information, T65C002 Exchange of supported operations per device, T65C003 Exchange of historical data of devices, T65C004 Exchange of end user/owner information.

Sprint 1 (Feb 3, 2020 – Feb 17, 2020)

- **Add Functional Specs to reporting folder:** Part of the initial repository setup. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65B001 ZDMP and eFactory Marketplace Integration, T65C001 Exchange of device- and sub-device Information, T65C002 Exchange of supported operations per device, T65C003 Exchange of historical data of devices, T65C004 Exchange of end user/owner information.
- **Create repository structure:** Part of the initial repository setup. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65B001 ZDMP and eFactory Marketplace Integration, T65C001 Exchange of device- and sub-device Information, T65C002 Exchange of supported operations per device, T65C003 Exchange of historical data of devices, T65C004 Exchange of end user/owner information.

Sprint 2 - 7 (Feb 17, 2020 – May 11, 2020)

- **Planning for interlinking multiple ZDMP instances:** Planning of the next steps for interlinking two or more ZDMP instances. This includes topics such as the interconnection of multiple Services and Message Bus components to provide API access to ZDMP resources running on different hosts. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms.
- **Planning for interlinking ZDMP with ADAMOS:** Planning of the next steps for interlinking the ADAMOS IIoT platform with ZDMP. This includes topics such as the interconnection of the Cumulocity instance used in the T5.1 Data Acquisition component with the Cumulocity backend of ADAMOS. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65C001 Exchange of device- and sub-device Information, T65C002 Exchange of supported operations per device, T65C003 Exchange of historical data of devices, T65C004 Exchange of end user/owner information.
- **Planning for interlinking the ZDMP marketplace with the eFactory marketplace:** Planning of the next steps for interlinking ZDMP's marketplace with the eFactory marketplace. An interlink enables the cross-platform marketing of zApps. This milestone is strongly related to the work done in Task T6.2 Marketplace. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65B001 ZDMP and eFactory Marketplace Integration.
- **Planning for interlinking ZDMP with SDAIM:** Planning of the next steps for interlinking ZDMP with UOS-ITI's SDAIM framework for real-time data analytics. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms.
- **Merging of permissions between two platforms:** Implement a module to map the ZDMP's security model with the model used by external platforms. Suggest design improvements from T5.2 to take a federated approach. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65B001 ZDMP and eFactory Marketplace Integration, T65C004 Exchange of end user/owner information.
- **Merging of data pipelines:** Implement a module to map the ZDMP's data model with the model used by external platforms. Make sure that the T6.4 Services and Message Bus and the T6.4 Application Run-time considers a federated approach. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65B001 ZDMP and eFactory Marketplace Integration, T65C001 Exchange of device- and sub-device Information, T65C002 Exchange of supported operations per device, T65C003 Exchange of historical data of devices, T65C004 Exchange of end user/owner information.
- **Merging of application flow:** Implement a module to map the ZDMP's application flow with the model used by external platforms. Make sure the T6.4 Services API Management can manage federated architecture and connect to another version of itself on another running instance of ZDMP. Related Subtasks: T65A001 Interconnection of ZDMP with external platforms, T65B001 ZDMP and eFactory Marketplace Integration, T65C001 Exchange of device- and sub-device Information, T65C002 Exchange of supported operations per device, T65C003 Exchange of historical data of devices, T65C004 Exchange of end user/owner information.

10.2.4 Limitations

There were no issues found in this period which would limit the final scope of the activities expected.

11 T6.6 General Cross-Task Integration and Improvement

11.1 Public documentation

This is not a standard component software task, but a meta task for the integration between components. Therefore, there is also no deliverables, status, or progress reports, besides a plain text explanation of current integration issues. At the current state of the project, integration between components is still low. The most urgent integrations happening currently are related to:

- T5.2: Integration of authentication and authorization mechanisms to provide user management for the platform, where IKER is currently active in providing the necessary systems and APIs
- T6.2: Integration of storage for the diverse tasks that need data storage is currently hampered by a slow start of SIVECO / SIMAVI, where currently still discussions about the concept for the storage tasks are ongoing as the concept is still not clear to the consortium
- T6.4: integration of software artifacts at Platform level is currently hampered by not having a central platform to run it on, but this issue is known and being dealt with by the project coordinator
- T6.4 / T6.5: Message Bus and Service Bus are set up as commercial systems by Software AG, and are reported to be testable

12 Risks

The risk status associated with this WP and underlying task are as follows:

Risk	WP Likelihood % Impact 🌟	Critical risks for implementation - Proposed risk-mitigation measures -	Current Status
Failure in Functions (ZD production and products)	WP: WP8 %: Low 🌟: High	This is considered as a core issue in the project. This is marked low occurrence due to the previous research that has been performed and the success of the dependent projects; Nonetheless, having multiple pilots in the project will allow early detection of the problem if it occurs and WP11 will work as early testing environment to avoid such issues.	Pilots concrete implementation through ZD platform has not started yet; thus, no change to risk level Risk future change: % Low: No change 🌟 High: No change
Time for development underestimated	WP: WP8 %: Medium 🌟: High	Pilots and research & development of components will be performed in parallel in a hybrid-agile approach. Pilots can be re-scheduled and re-scoped to mitigate against the delayed delivery of software. Changes in the scope might be motivated by lack of specific essential components to conduct the work or the risk of technologies not being mature enough for the purpose of the pilots. All technical providers have many years of development experience and timely delivery. Most partners selected have a long history of EU projects.	Pilots concrete implementation through ZD platform has not started yet; thus, no change to risk level Risk future change: % Medium: No change 🌟 High: No change
Uncomplete Datasets from use cases for ML/AI training	Tasks: T8.2/3 %: Medium 🌟: Medium	This is a well-known problem regarding most standard ML/AI techniques and WP9/10 need to be ongoing to deepen the analysis of the use cases data. Nonetheless, this should be performed within the second year, what is more sub-calls could fulfil possible gaps at this regard.	WP9/10 activity has just recently started; thus, no change to risk level Risk future change: % Medium: No change 🌟 Medium: No change
Computing load and related requirements	Tasks: T8.2/3/4 %: Medium 🌟: High	Machine Learning models, especially for the training process, can be highly resource consuming. Furthermore, in T8.2/T8.4 continuous training mechanisms are foreseen. Resources should follow expected requirements, at the same time training processes must be scheduled or run on specific cluster outside premises. Platform services like Autonomous and Distributed Computing should help.	Implementation needs to be more mature and datasets available to better assess the risk; thus, no change to risk level Risk future change: % Medium: No change 🌟 High: No change
Too high latency on results of T8.3	Tasks: T8.3 %: Medium 🌟: High	Considering that T8.3 is partly based on machine vision techniques, could be that micro services approach and platform layers increase latencies so that they become not compatible with real application needs. In that case will review the module using edge computing approach and develop only a microservice connector to platform for configuration and result publication coming from edge-based solution.	Basic Platform instantiation is needed to assess the risk; thus, no change to risk level Risk future change: % Medium: No change 🌟 High: No change

13 Conclusions

This document reports on the different WP6 tasks' activities and components development providing the current status at M18.

It first illustrates ZDMP WP6 work in the scope of ZDMP global architecture and describes what tasks and components compose it. Its main aim is to create a basic platform to execute the other ZDMP components and the zApps created by the third-party developers, as well as supporting this creation of zApps.

It is composed of the following tasks/components:

- T6.1 - SDK: Applications and Service Builder
- T6.2 - Secure Business Cloud:
 - Storage
 - Marketplace
 - Security System Modeler
- T6.3 - Human Collaboration Environment
- T6.4 - Platform Integration and Federation:
 - Platform
 - Services and Message Bus
- T6.5 - Inter-Platform Interoperability
- T6.6 - General Cross-Task Integration and Improvement

All these tasks are being monitored and evaluated against the requirements presented in D048 Requirement Document and Update, and Software development approach is based on agile sprints following D021 Methodology Document.

Every section of the document describes the achievements of each task/component and is being complemented by the software reports found in the public repository.

The main challenges in this first iteration was to find a mutual understanding of how the components are supposed to work, as after the specification phase there were still unanswered concept questions. Especially the SIVECO lead-components were missing a clear vision. Also, some other components are not in an optimal shape, such as the Marketplace or the Storage, where not too much work has been performed. However, the consortium put in place the following measures to overcome these challenges:

- SIVECO / SIMAVI is now closer looked at in terms of reporting
- Ascora is changing its internal organizational structure which enables developers to invest more resources in a more agile way on Marketplace

Annex A: Document History

Document History	
Versions	<ul style="list-style-type: none">• V0.0.1<ul style="list-style-type: none">• First tentative TOC and initial content in section 0• V0.0.2<ul style="list-style-type: none">• Draft content• V0.0.3-4<ul style="list-style-type: none">• New drafts after coordinator comments• V0.2.3<ul style="list-style-type: none">• Final version for review• V0.2.2<ul style="list-style-type: none">• Version reviewed• V1.0.0<ul style="list-style-type: none">• Final version for review• V1.0.0A<ul style="list-style-type: none">• Final version for review
Contributions	<ul style="list-style-type: none">• ASC:<ul style="list-style-type: none">• Tim Dellas• Nargis Tahara• ITI:<ul style="list-style-type: none">• Francisco Barrena• Santiago Cáceres• ICE:<ul style="list-style-type: none">• Stuart Campbell• SAG:<ul style="list-style-type: none">• Martin Hess• SIMAVI:<ul style="list-style-type: none">• Mircea Vasilie• UOS:<ul style="list-style-type: none">• Stephen Phillips

Annex B: References

None



www.zdmp.eu