



PROJECT IDENTIFICATION

PROJECT NUMBER	AAL-2016-089
DURATION	1 st March 2017 – 29 th February 2020
COORDINATOR	Markus Garschall
COORDINATOR ORGANIZATION	AIT Austrian Institute of Technology GmbH
WEBSITE	www.success-aal.eu



DOCUMENT IDENTIFICATION

DELIVERABLE ID	D4.1 Functional specification and integrated architecture support
RELEASE NUMBER / DATE	v1.0 / 08/10/2017
CHECKED AND RELEASED BY	Markus Garschall (AIT)

KEY INFORMATION FROM 'DESCRIPTION OF WORK'

DELIVERABLE DESCRIPTION	Report on the functional specification and architecture of SUCCESS. This deliverable summarizes the first version of architecture and specification definition based on the output produced so far by the Use Case Definitions
DISSEMINATION LEVEL	Public
DELIVERABLE TYPE	Report
ORIGINAL DUE DATE	M05 / 31/06/2017

AUTHORSHIP & REVIEWER INFORMATION

EDITOR	Ntalaperas Dimitrios (SIL)
PARTNERS CONTRIBUTING	EXT, AIT, UCY, CIT
REVIEWED BY	Jakob Hatzl (EXT)

ABBREVIATIONS

ABBREVIATIONS	DESCRIPTION
AAL	Active and Assisted Living
AAL CMU	AAL Central Management Unit
PwD	Person with Dementia

TABLE OF CONTENTS

1	ABOUT THIS DOCUMENT	5
1.1	ROLE OF THE DELIVERABLE	5
1.2	RELATIONSHIP TO OTHER SUCCESS DELIVERABLES	5
2	INTRODUCTION	6
3	SYSTEM ARCHITECTURE	6
4	COMPONENTS	7
4.1	OUTPUT PLATFORM	8
4.1.1	FUNCTIONAL AND NON FUNCTIONAL VIEW	8
4.1.2	PROCESS VIEW	8
4.1.3	LOGICAL VIEW	9
4.1.4	INTERFACES	10
4.2	PROFILER	10
4.2.1	FUNCTIONAL AND NON FUNCTIONAL VIEW	11
4.2.2	PROCESS VIEW	11
4.2.3	LOGICAL VIEW	12
4.2.4	INTERFACES	13
4.3	REWARDER	14
4.3.1	FUNCTIONAL AND NON FUNCTIONAL VIEW	14
4.3.2	PROCESS VIEW	15
4.3.3	LOGICAL VIEW	16
4.3.4	INTERFACES	16
4.4	ADVISER	17
4.4.1	FUNCTIONAL AND NON FUNCTIONAL VIEW	18
4.4.2	PROCESS VIEW	18
4.4.3	LOGICAL VIEW	20
4.4.4	INTERFACES	20
4.5	CONTENT PROVIDER	21
4.5.1	FUNCTIONAL AND NON FUNCTIONAL VIEW	22
4.5.2	PROCESS VIEW	23
4.5.3	LOGICAL VIEW	24
4.5.4	INTERFACES	24
5	FURTHER WORK	26

EXECUTIVE SUMMARY

The present report documents the set of functional requirements and the integrated architecture of the SUCCESS platform as these are defined until M05 based on the output produced so far from *T2.2: Definition of uses cases and scenarios*. Based on these needs, the architecture was designed to include a total of five modules, each responsible for a wide set of functionalities ranging from providing content, offering advice, providing positive feedback to the user and displaying all the content and dialogues to user interfaces. For each of these core modules the list of functional and non-functional requirements have been identified, data flow and component diagrams have been provided and the interfaces between them were defined. Though, as the use cases are defined in even more detail, several aspects of the specifications and architecture may change, the modular design of the architecture provides much room for adaptation.

1 ABOUT THIS DOCUMENT

1.1 ROLE OF THE DELIVERABLE

Specification and implementation of the SUCCESS Platform follows an agile and iterative process whereas, in conjunction with the definition of use cases, the architecture is defined in incremental steps with the aim of converging to a version that fully supports the desired functionality, as this is defined by the use cases.

The present deliverable documents the first iteration of the functional specification and architecture specification of SUCCESS. Since the Use Cases are being defined in parallel, it mostly describes the high-level definition of components and the general functional requirements of the system. The final version of the document, scheduled for M17 will document the full architecture which will incorporate all details of the Use Cases.

1.2 RELATIONSHIP TO OTHER SUCCESS DELIVERABLES

The deliverable is related to the following SUCCESS deliverables:

DELIVERABLE	RELATION
D2.2 Use Cases, Scenarios, Service and Interaction Design Concept	D4.1 relies heavily on D2.2 since the details of the architecture and the functional requirements should be defined as to fully cover the whole range of the Use Cases.
D4.2 Security and privacy infrastructure specification	D4.2 specifies the security and privacy infrastructure for the SUCCESS system based on the architecture defined in the deliverable D4.1 at hand.

2 INTRODUCTION

The aim of SUCCESS is to provide an innovative training application on the user's mobile device to support and accompany formal and informal caregivers, and the public, to appropriately and effectively interact with persons with dementia, based on evidence-based communication and intervention strategies for dementia. The present report presents the first version of the functional specification and the integrated architecture of the SUCCESS platform. The architecture of SUCCESS is designed in a modular way, with each component being loosely coupled and thus providing flexibility in future implementation and deployment. More specifically, while fulfilling all the global and component specific requirements, each component will also consist a standalone module, and will thus be able to be deployed either as an application or a micro service, and, for development purposes, either in a host, virtual machine or in a container using docker.

Designing the architecture and defining the specifications will also follow an agile approach; a range of functionalities that the SUCCESS platform must fulfil should be defined and the architecture will be designed towards fulfilling those specifications. In parallel with implementation, the next iteration of the design will narrow the internal specifications of the components and their corresponding interfaces. The first prototype implementation will therefore fulfil the criteria of the final specification, and we will consequent software releases. The work produced in the context of *D2.2 Use Cases, Scenarios, Service and Interaction Design Concept* will provide the guidelines for the final version of the requirements definition and system architecture.

3 SYSTEM ARCHITECTURE

In broad terms, the SUCCESS Platform should be able to execute the following tasks:

- Allow users to register and personalize their application settings and personal profile information
- Provide guidance to users regarding a specific situation, both in terms of offering advice per the criteria set by the user and in terms of offering training to the user
- Provide training to the users so they can learn about communication strategies when dealing with persons with dementia
- Offering suggestions and information to users on how to provide for a meaningful life to persons with dementia
- Advising users on how to offer emotional support to persons with dementia.
- Provide the means to users for evaluating the system.

Figure 1 depicts the high level architecture diagram of the SUCCESS Platform. The Platform consist of a number of loosely coupled modules:

- The Output Platform displays the visual representation of the current state of the application to the user

- The Profiler handles the user data and can retrieve any relevant information to queries involving questionnaire parameters
- The Rewarder provides feedback to the user based on performance measures and on her/his actions
- The Adviser is responsible of generating advise based on the current node of a dialogue initiated between the user and the application that can be audibly and visibly represented by an avatar.
- The Content Provider is the general back-end component that manages the contents that are presented to the user (e.g. nodes of a dialogue)

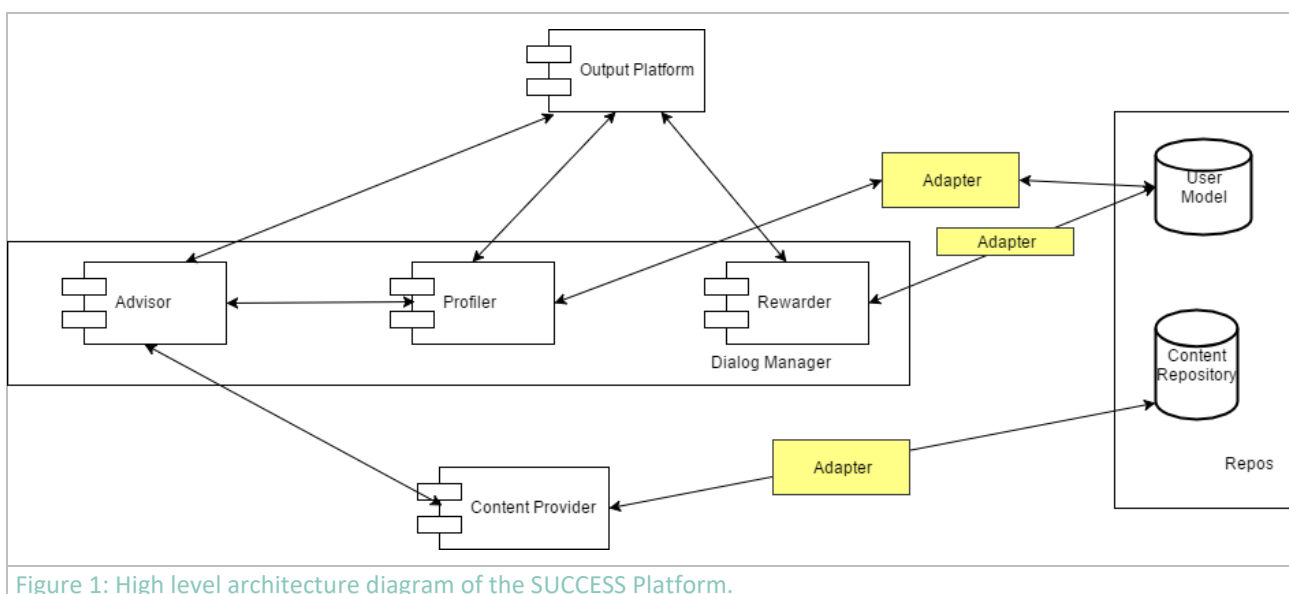


Figure 1: High level architecture diagram of the SUCCESS Platform.

4 COMPONENTS

The aim of the present Section is to present the specification of each of the components defined in Section 6. For each component a list of functional and non-functional, if any, requirements will be given, as well as the level 0 (level 1 in terms of the general architecture) data flow and component diagrams. Finally, a list of defined interfaces will also be presented.

The name of each requirement will be uniquely identified by three components: a) The type of requirement (FUNC being functional while REC being non-functional), b) a short letter description of the component and c) an incremental integer. Each requirement can be either mandatory (whereby it the described functionality will be annotated with the verb “MUST”) or desired (whereby “SHOULD” will be used).

For the Interfaces, two type of interfaces are identified: a) User interfaces between component and users and b) Communication interfaces between components. The complete specification of the interfaces will depend on the final form of the Use Cases, since the data flow and data format between components must follow the sequences defined in the Use Cases. When possible details of the specified Interfaces regarding the type and the data format are given; otherwise the full specification will be given in the next version of D4.1.

4.1 OUTPUT PLATFORM

The output platform is one of the three components realising the Representation and Interaction part of the SUCCESS solution. More specifically, the output platform represents the main user frontend with which the users interact directly. For this, market-based mobile devices like tablet PCs and Smartphones will be used.

The main purpose of the output platform is the rendering of the user interface that will allow the user to interact with the system and use the services provided by SUCCESS. It also integrates the natural communication module and the affective avatar and it coordinates the dataflow between the user and the e-learning backend platform.

4.1.1 FUNCTIONAL AND NON FUNCTIONAL VIEW

Table 1 depicts the list of functional and non-functional requirements for the Output Platform. Briefly, the Output Platform should integrate with the various visual modules, like the affective avatar, and coordinating the dataflow originating in the backend as to represent it visually in a coherent way.

Requirement	Description
FUNC_OUT_1	MUST integrate the natural communication module
FUNC_OUT_2	MUST integrate the affective avatar
FUNC_OUT_3	MUST coordinate the dataflow between the user and the e-learning backend platform
REC_OUT_1	SHOULD provide the means to present information to the end user in a unified manner in order to sustain uniformity.

Table 1: Requirements List for the Output Platform

4.1.2 PROCESS VIEW

Figure 2 depicts the level 0 Data Flow Diagram of the Output Platform. External entities can interact by the various internal modules of the Output Platform (denoted by the generic name “Process” in the Figure) and have access to the Data Store in this manner. Users can login via the Login Process and from there on have access to the various panels of the Output Platform.

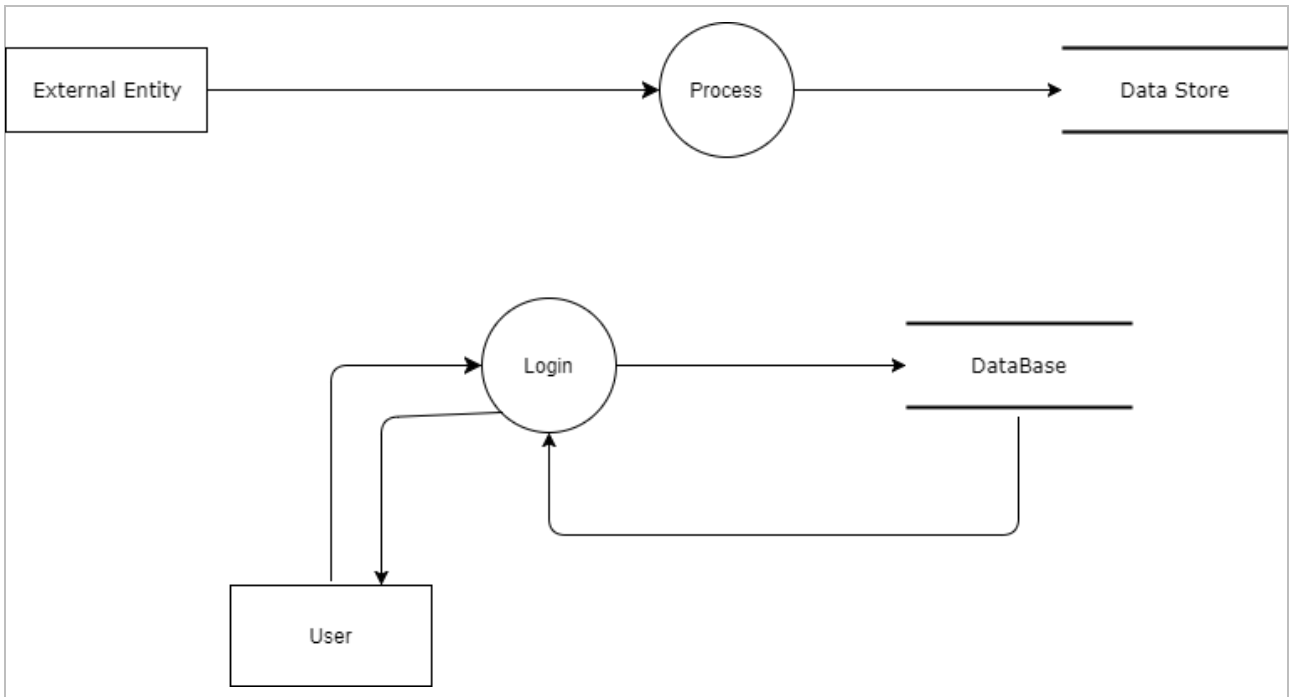


Figure 2: Level 0 Data Flow Diagram for the Output Platform

4.1.3 LOGICAL VIEW

Figure 3 depicts the level 0 Component Diagram of the Output Platform. Typical of the Model-View-Controller architecture the Login Form shares an interface with the Login Backend which is interfacing with the DataBase Adapter for accessing data. The DataBase Adapter exposes generic interfaces for other modules of the Component. These other modules have a business logic which will be fully concretized after the finalization of the Use Cases and are therefore omitted in the present version of the deliverable.

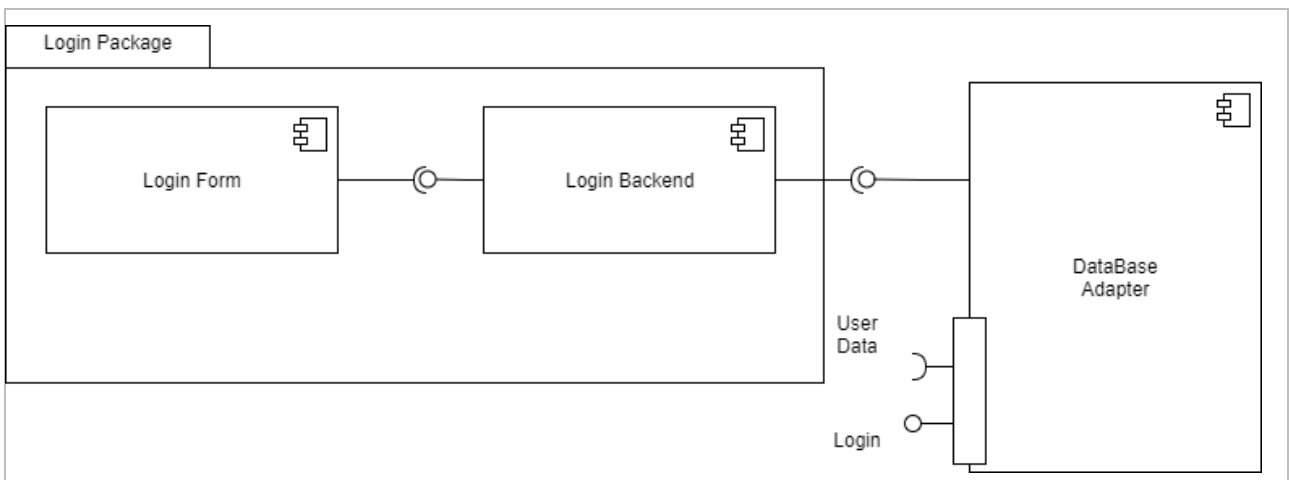


Figure 3: Level 0 Component Diagram for the Output Platform

4.1.4 INTERFACES

4.1.4.1 USER INTERFACES

Table 1 depicts the User Interfaces of the Output Platform. The panels cover the range of activities defined thus far in the Use Cases.

Interface	Description
Login Screen	The Login Screen will provide a graphical interface to the user, which can be used to login to the platform.
Communication Strategies Screen	This screen will provide innovative training with the avatar for evidence-based communication and intervention strategies
Situation Guidance Screen	This screen will provide general methods and guide users to effectively respond to specific situations
Emotional Support Screen	This screen will provide strategies for emotional support as to maintain on the carers' own identity
Care Activities Screen	This screen will help carers to create meaningful activities, thus, maintaining a sense of purpose at the individual's level of ability.
Stigmatisation Avoidance Screen	This screen will assist the users to learn how to cope with behaviours and address their own feelings and avoid stigmatisation
Gamification Screens	Various screens will present games to provide an interesting and innovative way of teaching and guidance

Table 2: User Interfaces of the Output Platform

4.1.4.2 COMMUNICATION INTERFACES

Communication interfaces of the Output Platform to the various other components requires the definition of the succession of steps of the various Use Cases as well as the definition, at least in the most general sense, of the format of data that is to be exchanged. As this is a work in progress, the Communication Interfaces of the Output Platform will be defined in the next version of the present deliverable.

4.2 PROFILER

The Profiler component is one of the three components realising the dialogue management in the SUCCESS solution.

The main purpose of the Profiler is to administer data about the user in a user model. The Profiler does this through continuously recording of data during system usage, as well as through initiating active inquiry.

The main tasks are:

- Acquisition of user-relevant data, through: active inquiry, usage patterns, Adviser input etc.
- Administering the internal user model
- Realizing a Questionnaire like dialogue model to produce the initial user model/profile, and managing subsequent updates to it.

4.2.1 FUNCTIONAL AND NON FUNCTIONAL VIEW

Table 3 depicts the list of requirements of the Profiler. In general, basic functionality involving data handling and involving the Profile is mandatory, while a list of more refined functionalities like combination and reasoning on the data are optional requirements.

Requirement	Description
FUNC_PROF_1	MUST be able to read and update the user profile.
FUNC_PROF_2	MUST be able to interpret questionnaire like dialogue model to fetch user profile data.
FUNC_PROF_3	SHOULD generate a questionnaire like dialogue model from a user profile.
FUNC_PROF_4	SHOULD be able to include usage metrics into the user profile data.
FUNC_PROF_5	SHOULD be able to include adviser component data into the user profile data.
FUNC_PROF_6	SHOULD be able to combine user profile data to deduce additional profile data.

Table 3: Requirements List for the Profiler

4.2.2 PROCESS VIEW

Figure 4 depicts the level 0 Data Flow Diagram for the Profiler. The user Profile is accessed by various processes that can extract data like *Usage Metrics*, *Questionnaire Inquiry* and *Questionnaire generation*. These processes provide data to the output platform either directly or indirectly via other modules like the Adviser.

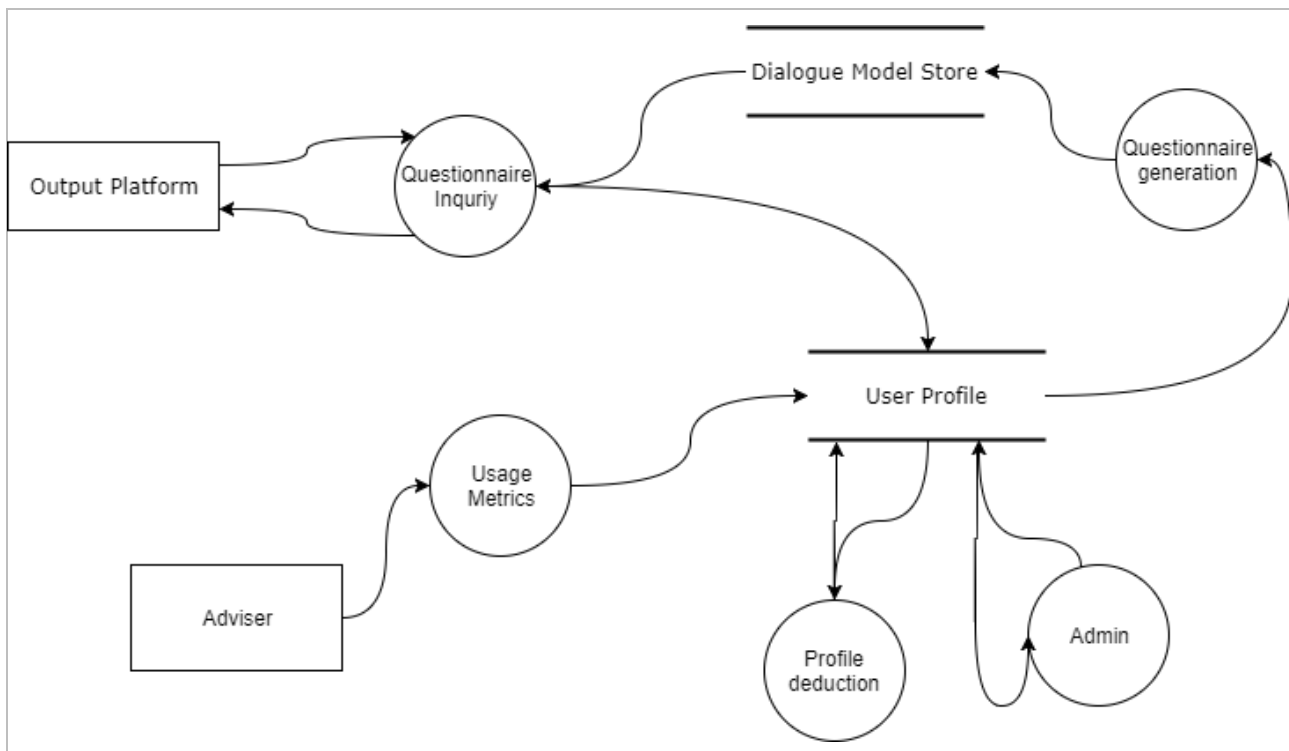


Figure 4: Level 0 Data Flow Diagram for the Profiler

4.2.3 LOGICAL VIEW

The level 0 Component Diagram of the Profiler can be seen at Figure 5. The core modules of the Profiler that implement the business logic for the basic functionality are contained in a single package, while modules that implement the interfaces reside in separate packages.

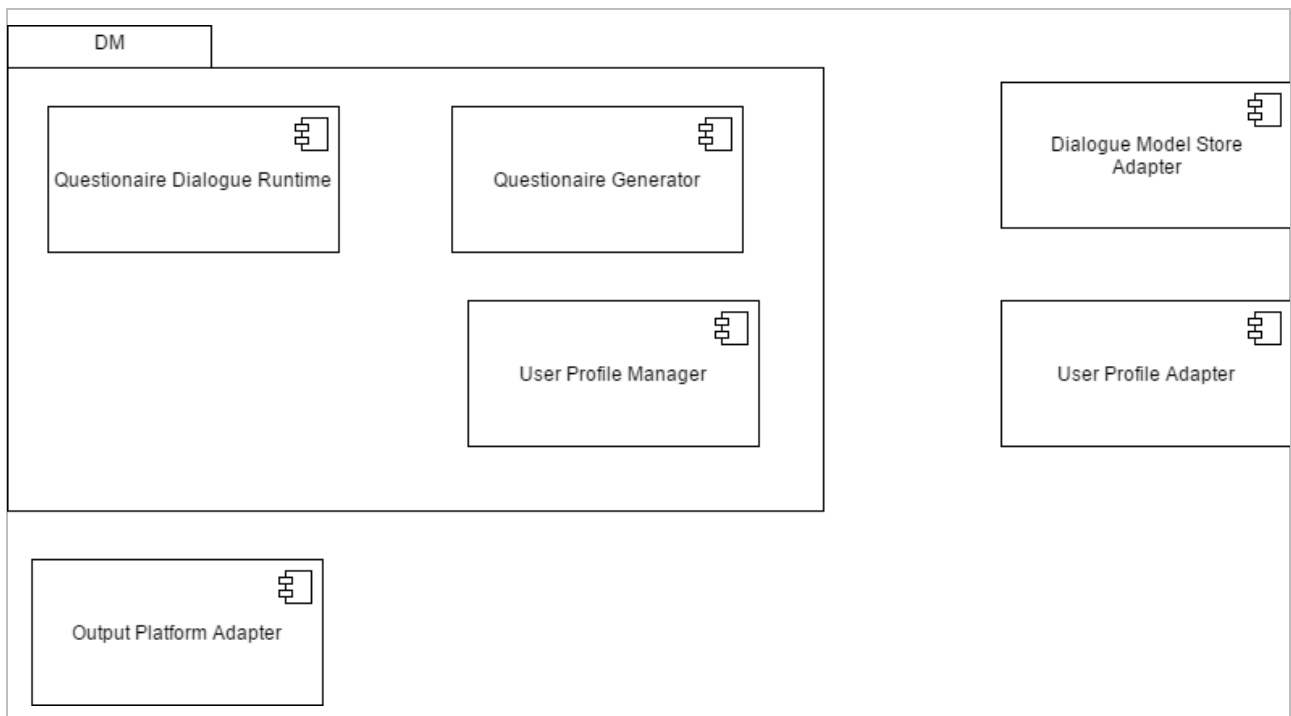


Figure 5: Level 0 Component Diagram for the Profiler

4.2.4 INTERFACES

4.2.4.1 USER INTERFACES

The Profiler interacts with the User when questionnaires need to be filled by the user. This is depicted in Table 4.

Interface	Description
Questionnaire Screen	The Login Screen will provide a graphical interface to the user, which can be used to login to the platform.

Table 4: User Interfaces of the Profiler

4.2.4.2 COMMUNICATION INTERFACES

The Interfaces of the Profiler with other modules are depicted in Table 5. The Profiler needs to communicate the output of the questionnaires to the Output Platform and provide the usage metrics

needed by the Adviser. In the data level, the Profiler needs to interface with the User Model in order to be able to retrieve information about the user.

Interface	Description	Interfacing Component
Questionnaire	Send output request for questionnaires	Output Platform
Profile	Read user profile data to generate questionnaire dialog models	User Model (Adapter)
Metrics	Receive usage metrics	Adviser

Table 5: Communication Interfaces of the Profiler

4.3 REWARDER

The rewarder component is one of the three components realising the dialogue management in the SUCCESS solution.

The main purpose of the rewarder component is to choose or generate feedback to the user. It does so by choosing appropriate feedback for performance measures from the system (i.e. derived from Profiler data). The rewarder utilises adequate and meaningful content to increase motivation and engagement with the trainings.

4.3.1 FUNCTIONAL AND NON FUNCTIONAL VIEW

In general, the Rewarder needs to measure performance indicators and based on these generate feedback to the user. Table 6 depicts the full list of requirements for the Rewarder.

Requirement	Description
FUNC_REWA_1	MUST measure performance indicators from interaction data.
FUNC_REWA_2	SHOULD be able to incorporate performance metrics.
FUNC_REWA_3	MUST be able to create content queries to fetch content.
FUNC_REWA_4	SHOULD be able to rate motivational content according to relevancy (when, what).

FUNC_REWA_5	MUST be able to generate dialogue models to give feedback to the user of the system.
-------------	--

Table 6: Requirements List for the Rewarder

4.3.2 PROCESS VIEW

The level 0 Data Flow Diagram for the Rewarder is shown in Figure 6. Based on output from the Content Provider the *Content Classification* process classifies the content and associates the relevant information with the User Profile. *Feedback* also receives input from the Content Provider and data from the User Profile and generates output which can be displayed by the Output Platform. The *Performance measure* process needs input concerning the actions of the User which is provided by the Output Platform; it communicates with the Profiler and the User Profile to associate the measured performance with the profile of the user. Finally, *Performance Metrics*, can be defined by obtaining data from the Profiler

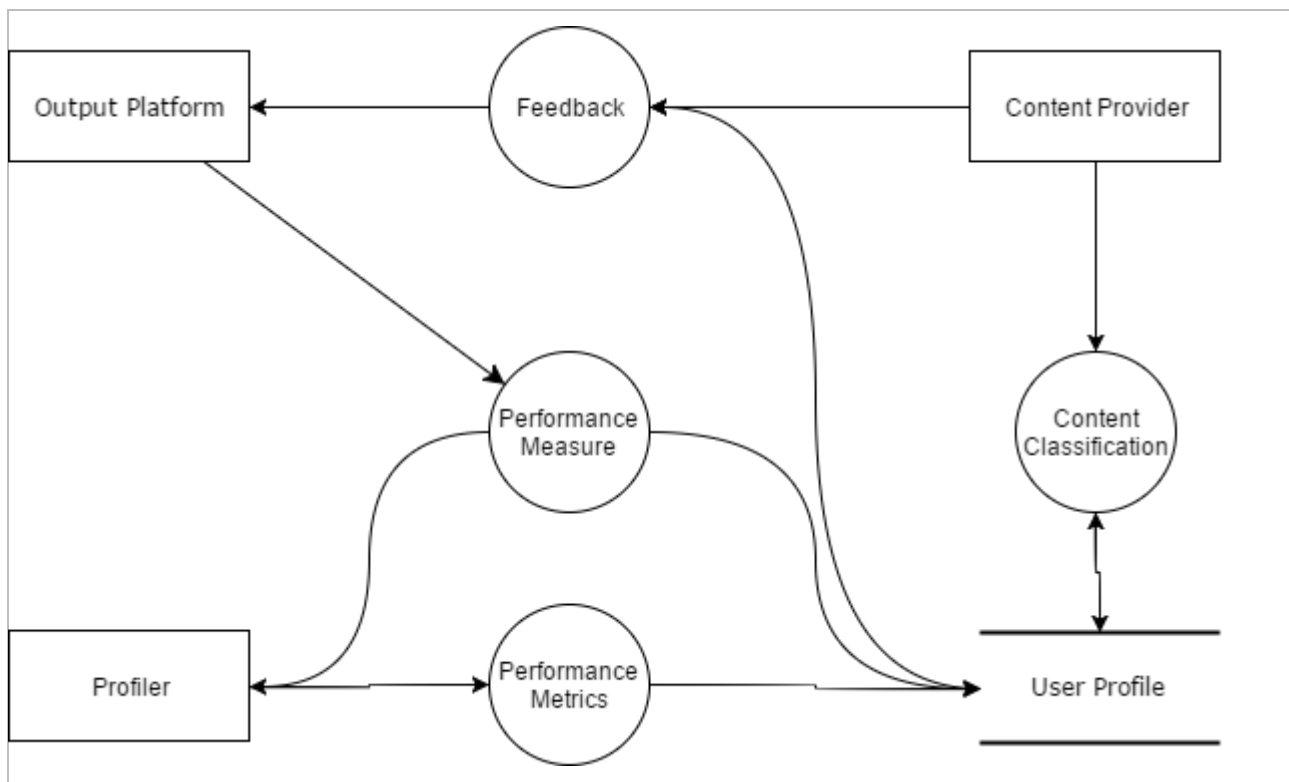


Figure 6: Level 0 Data Flow Diagram for the Rewarder

4.3.3 LOGICAL VIEW

The component diagram of the Rewarder can be seen in Figure 7. The main modules of the Rewarder are the Performance Recorder, the Content Classifier and the Feedback Generator. They need to be able to interface with the adapters of the modules that they share data with, as these can also be seen in the Data Flow Diagram presented in the previous section.

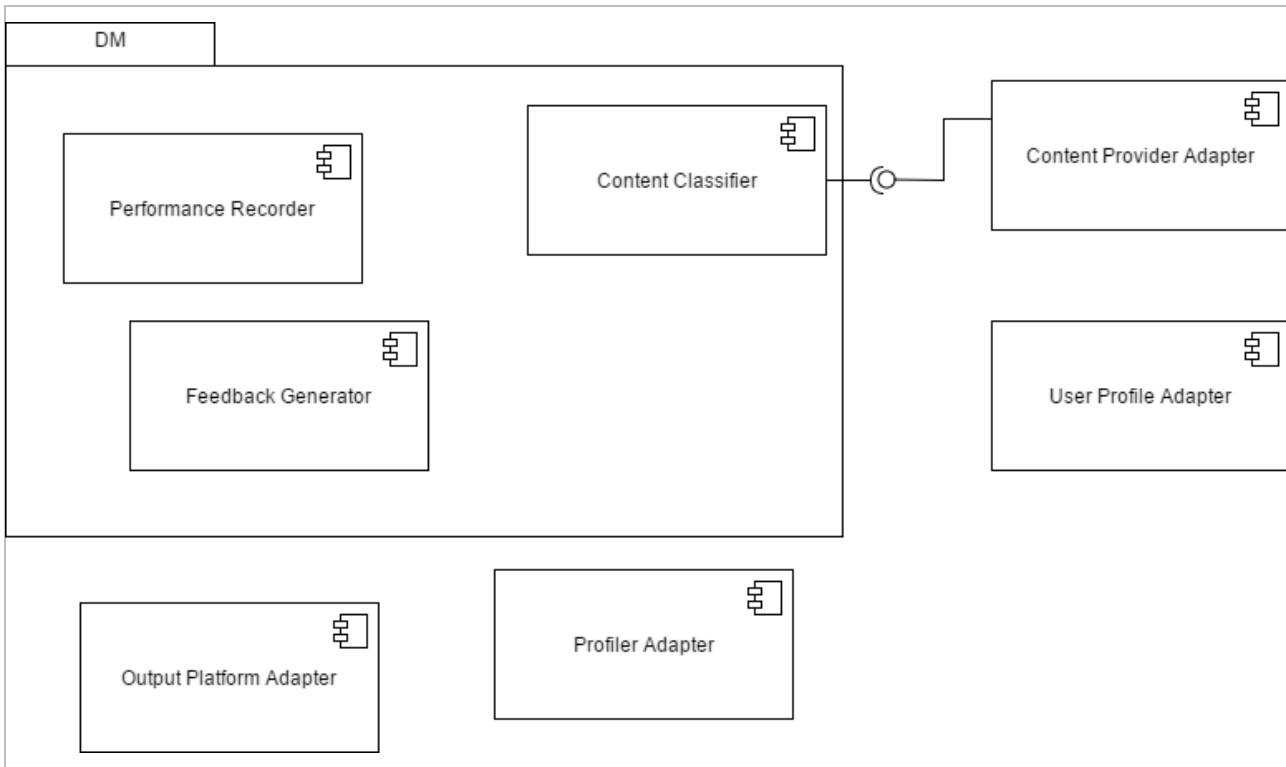


Figure 7: Level 0 Component Diagram for the Rewarder

4.3.4 INTERFACES

4.3.4.1 USER INTERFACES

Table 7 shows the sole user interface that the Rewarder has. It is used to provide the feedback generated to the user.

Interface	Description
Feedback Screen	Should be shown to provider user with motivating feedback.

Table 7: User Interfaces of the Rewarder

4.3.4.2 COMMUNICATION INTERFACES

Table 8 lists the interfaces the Rewarder shares with the other components of the SUCCESS Architecture. Also depicted in the Data Flow Diagram, the interfacing components are the Output Platform, the Content Provider and the Profiler.

Description	Interfacing Component
Send Output Requests for Feedback	Output platform
Receive User Interaction data	Content Provider
Receive feedback content nodes	
Send Performance Measures	Profiler
Receive Performance Metrics	

Table 8: Communication Interfaces of the Profiler

4.4 ADVISER

The adviser component is one of the three components realising the dialogue management in the SUCCESS solution.

The main purpose of the advisor is to produce an advising entity that is visually and audibly represented by the avatar component. A dialogue like interaction is generated from content nodes that contain dialogue models and the structure of the Meta model.

The main tasks are:

- Converting Meta Model based data structures for content navigation, into a model for dialogue like interaction.
- Interpretation of Avatar content (dialogue models) at run time taking some context information into account; generating real time interaction instructions for the Output platform's avatar component.
- Acquisition of interaction data, for profiling including usage and performance metrics

4.4.1 FUNCTIONAL AND NON FUNCTIONAL VIEW

The Adviser needs to recognize the state of the dialogue as this is encoded in the current succession of dialogue nodes, associate this state with information present in the User Profile and generate advise based on this combination. Table 9 lists the individual requirements that the Adviser needs to fulfil.

Requirement	Description	Comments
FUNC_ADVI_1	MUST be able to transform the Meta Model into a dialogue model for content navigation.	
FUNC_ADVI_2	SHOULD be able to interpret a dialogue model that models content navigation.	
FUNC_ADVI_3	MUST be able to interpret dialogue models conforming to the specification.	
FUNC_ADVI_4	MUST be able to create Output Platform requests from dialogue model interpretation state.	
FUNC_ADVI_5	MUST be able to interpret user input from the Output Platform.	
FUNC_ADVI_6	MUST be able to read and update the user profile.	
FUNC_ADVI_7	MUST be able to create content queries for the content provider, suitable for the given interpretation context.	
FUNC_ADVI_8	MUST provide the Profiler component with usage metrics based on input data from the output platform.	

Table 9: Requirements List for the Adviser

4.4.2 PROCESS VIEW

The main processes of the Adviser can be summarized as:

- Dialogue Execution: Obtains information from the Model describing the dialogue, the Content Provider, the User Model and the current state of the Output Platform and produces advise that is sent to the Output Platform for rendering. Also, gives feedback to the Usage Metrics Process
- Usage Metrics: After an advice is generated the Usage Metrics sends information of the interaction that has just taken place to the Profiler.
- Generate Content Navigation: Receives data from the Content Provider that are to be transformed to data structures that model the dialogue

The flow of information can be seen in Figure 8.

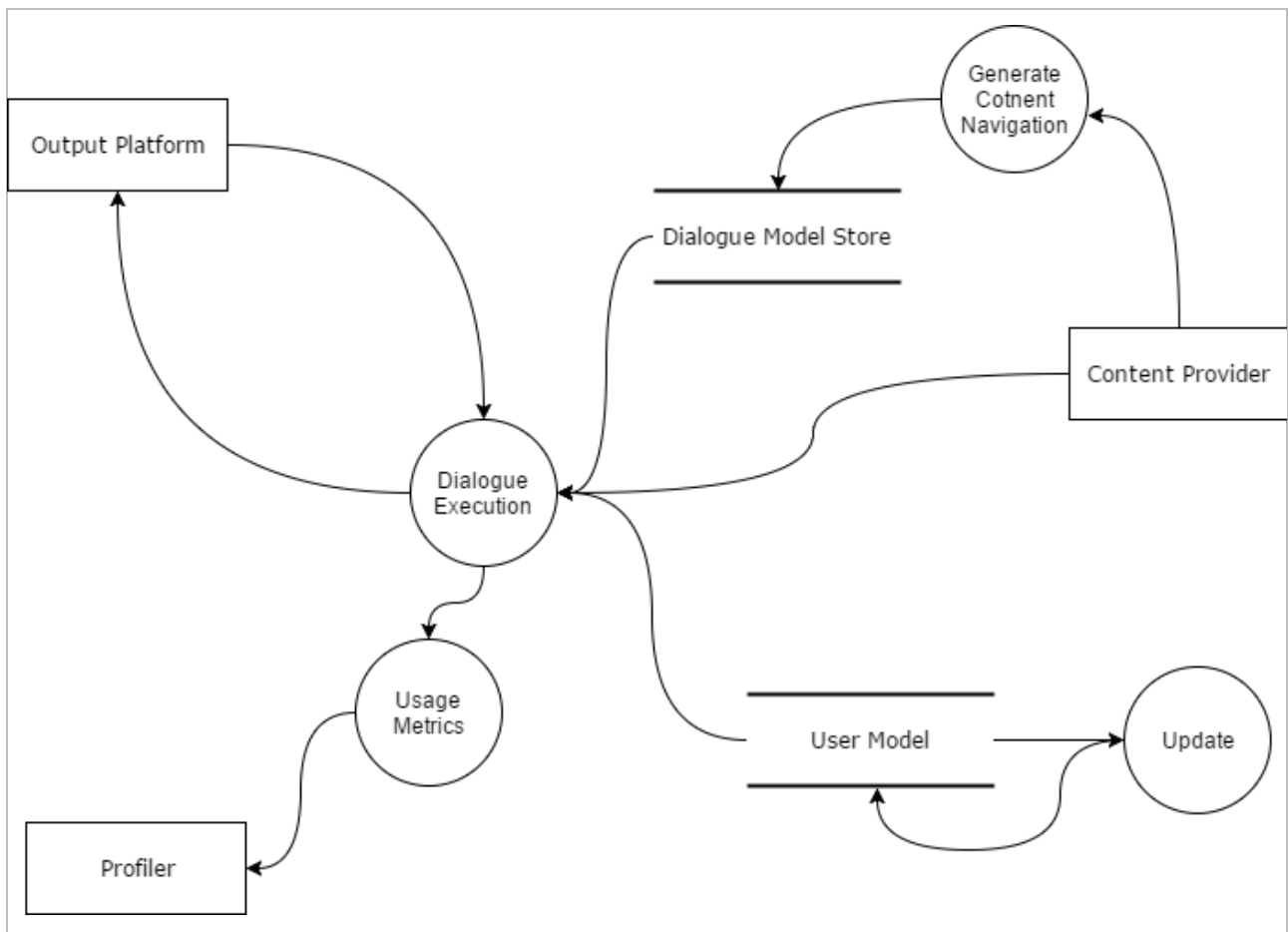


Figure 8: Level 0 Data Flow Diagram for the Adviser

4.4.3 LOGICAL VIEW

The components that implement the required functionality of the Adviser are depicted in Figure 9. A set of interfaces to the adapters of the other modules of the SUCCESS Platform will be implemented which the modules can use to exchange the required information.

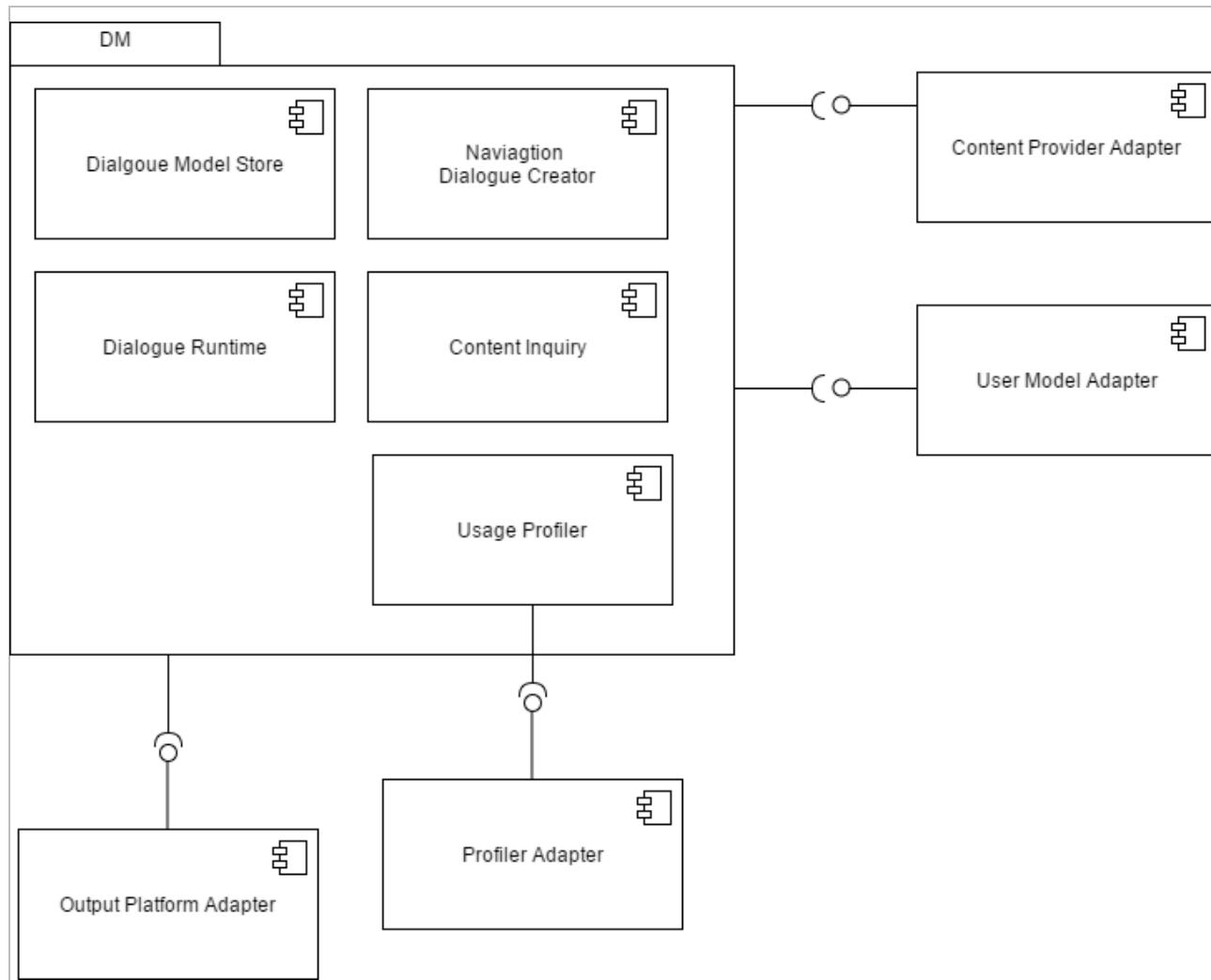


Figure 9: Level 0 Component Diagram for the Adviser

4.4.4 INTERFACES

4.4.4.1 USER INTERFACES

Table 10 shows the User Interfaces for the Adviser. The adviser needs to offer a UI for various content types and be able to display avatars according to the dialogue execution state.

Interface	Description
Content Output	UI for various content types (avatar, audio, video, text, ...)
Avatar Screen	UI for displaying avatars according to the dialogue execution state

Table 10: User Interfaces of the Adviser

4.4.4.2 COMMUNICATION INTERFACES

Table 11 lists the interfaces of the Adviser with the other modules of the SUCCESS Platform. Though subject to change to the final version of the architecture, most interfaces are REST and exchange information in JSON format. The only exception is the Content Node Interface which transmits the Content Nodes in binary representation.

Interface	Description	Interfacing Component	Type	Format
UI I/O Interface	Send Output Requests, Receive Input Responses	Output Platform	REST	JSON
Meta Model Interface	Fetching Meta Model	Content Provider	REST	JSON
Content Node Interface	Fetching Content Nodes	Content Provider	REST	HTTP, Binary
Profile Interface	Sending Usage metrics	Profiler	REST	JSON

Table 11: Communication Interfaces of the Content Provider

4.5 CONTENT PROVIDER

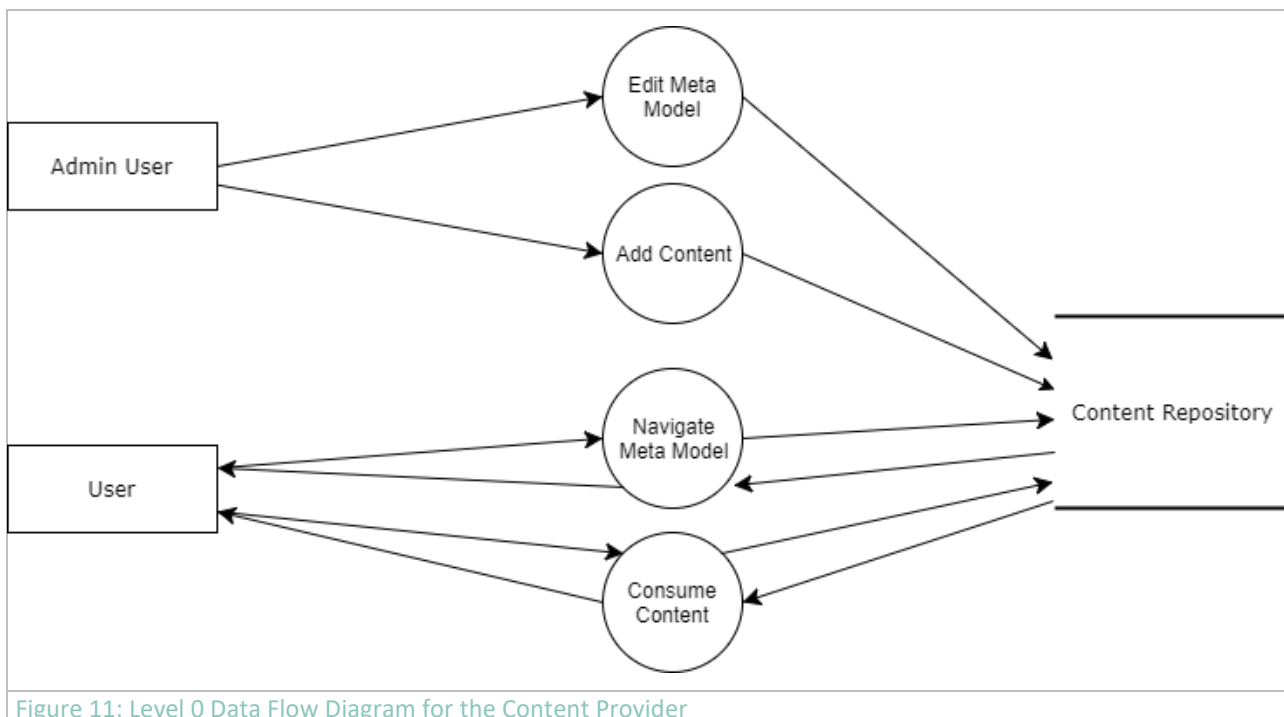
The content provider is the backend component that is responsible for managing and providing the contents that are presented to the Users. The contents are based on the Meta Model (see Figure 10), that is the internal model for structuring the different aspects of dementia covered in the application. The content is structured in different levels:

Requirement	Description
FUNC_CONT_1	SHOULD provide the Admin with the ability to add a layer to the meta model.
FUNC_CONT_2	SHOULD provide the Admin with the ability to add a structure node on any layer in the meta model
FUNC_CONT_3	MUST provide the Admin with the ability to add a content node on the content level in the meta model.
FUNC_CONT_4	MUST provide the Admin with the ability to add binary content to a content node.
FUNC_CONT_5	MUST be able to provide the Meta model as a structure for consummation.
FUNC_CONT_6	MUST be able to provide binary content for a specific content node for consummation.
REC_CONT_1	MUST provide the ability to specify different languages for each node in the structure or content level

Table 12: Requirements List for the Content Provider

4.5.2 PROCESS VIEW

Figure 11 lists the Data Flow Diagram of the Content Provider. In general, information flows between Users and the Content Repository. Admin Users can add content and edit the Meta Model. Registered Users can navigate the Meta Model and use the content.



4.5.3 LOGICAL VIEW

The Content Provider will be implemented by providing two service modules for managing and consuming content. These modules will be interfacing with the Content Repository. Following the Service Controller pattern, the business logic of the services is separated from the public controllers. The behaviour of the business logic service modules will be controlled by two respective publicly available controller modules. Figure 12 depicts the organization of the components of the Content Provider.

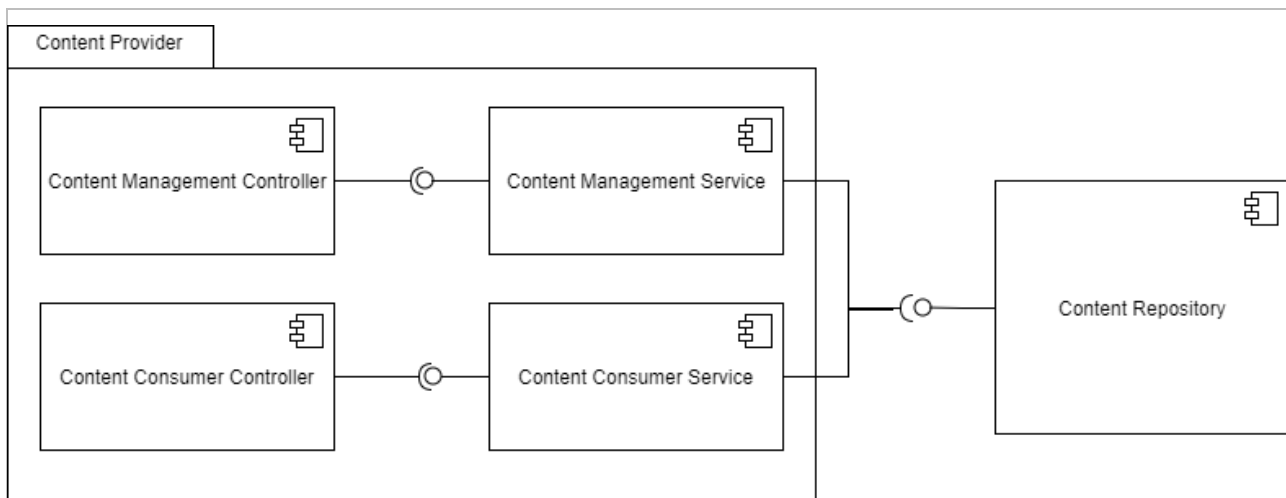


Figure 12: Level 0 Component Diagram for the Content Provider

4.5.4 INTERFACES

4.5.4.1 USER INTERFACES

As can be seen in Table 13, the Content Provider offers two user interfaces for Administrator users to be able to manage the Meta Model and the content of the content nodes.

Interface	Description
Meta Model Management Interface	Administrative User Interface for managing the content tree structure i.e. the Meta Model structure (may also be scripted in the first version)
Content Node Management Interface	Administrative User Interface for managing the binary content of specific Content Nodes

Table 13: User Interfaces of the Content Provider

4.5.4.2 COMMUNICATION INTERFACES

Table 14 lists the interfaces the Content Provider shares with the other components of the SUCCESS Platform. The Meta Model API which will be used to transverse the Content Tree will be shared with three components (Rewarder, Adviser and Output Platform) and the data will be in JSON format over REST. The actual contents of content nodes will be transmitted to the Adviser in binary format. The exact format of the interfaces is subject to change in the final version of the SUCCESS Architecture.

Interface	Description	Interfacing Component	Type (REST, queue, socket)	Format (e.g. json)
Meta Model API	This interface will allow interaction with the Content Tree (traversing the tree and node structure)	Rewarder Adviser Output Platform	REST	JSON
Content API	This interface will provide binary content from specific content nodes	Adviser	REST	Binary

Table 14: Communication Interfaces of the Content Provider

5 FURTHER WORK

The SUCCESS Architecture should offer the functionality to fully encompass all the Use Cases defined. Task *T2.2 Definition of use cases and scenarios* is ongoing and performed in parallel with task *T4.1 Architecture and interface specification*. Though the output produced in the context of *T2.2* has been incorporated in the current version of *D4.1*, the final version of the SUCCESS Architecture should account for the detailed Use Case definition. Future work consist of collaborating with the partners responsible for the Use Case Definition and monitoring the work being done in the context of *T2.2*. The modular design of the SUCCESS Architecture will allow the incremental modification and adaptation of the Architecture as the Use Cases are being refined. This process will culminate in the second version of *D2.2* containing the full Functional Specification and Architecture definition of the SUCCESS Platform.