

Project Identification

Project number	No. 611421
Duration	1 st Dec 2013 – 30 th Nov 2016
Coordinator	Andreas Stainer-Hochgatterer
Coordinator Organisation	AIT Austrian Institute of Technology GmbH, Austria
Website	www.miraculous-life.eu



Miraculous-Life

Miraculous-Life for Elderly Independent Living

Document Identification

Deliverable ID:	D5.3b Specification of Miraculous-Life system integration
Release number/date	V14 / 02.07.2015
Checked and released by	Emanuel Sandner / AIT
Work Status	Finished
Review Status	Accepted

Key Information from "Description of Work"

Deliverable Description	Describes the assembly and integration of the different components developed in WP2 to WP4 into the complete prototype system.
Dissemination Level	PU=Public
Deliverable Type	R = Report
Original due date	Project Month 16 / 30. April 2015

Authorship & Reviewer Information

Editor	Sten Hanke, Emanuel Sandner / AIT
Partners contributing	UCY, Zoobe, UniGe, Noldus
Reviewed by	Carsten Stockl�w / Fh-IGD

Release History

<i>Release Number</i>	<i>Date</i>	<i>Author(s)</i>	<i>Release description /changes made</i>
V01	16.06.2014	SH/AIT	First version of Del Template, initial structure
V02	05.08.2014	SH/AIT	Added KnowledgeBase information
V03	19.09.2014	ES/AIT	Added system requirements
V04	30.09.2014	ES/AIT	Modified requirements Added Installation setup
V05	01.10.2014	ES/AIT SH/AIT	Added Integration setup Added system integration Added system deployment Document ready for review
V06	02.10.2014	ES/AIT	Final modifications Document ready for review
V07	03.10.2014	ES/AIT CS/Fh-IGD	Incorporating reviewer comments Finalizing deliverable
V08	23.04.2015	SH/AIT	Applied review comments in document structure
V09	27.05.2015	SH/AIT	Added Asterisk Server integration setup Added Asterisk endpoint and client setup
V10	01.06.2015	SH/AIT	Added Development chapter
V11	15.06.2015	ES/AIT	Added Miraculous-Life integration plan Updated various chapters according to the current project development
V12	17.06.2015	ES/AIT	Updated System integration components figure by adding the Asterisk Server component and added description Updated Executive Summary Deliverable ready for review
V13	29.06.2015	CS/Fh-IGD	Deliverable review
V14	02.07.2015	ES/AIT	Incorporating reviewer comments Finalizing deliverable

Miraculous-Life Consortium

Miraculous-Life (Contract No. 611421) is a project within the 7th Framework Programme. The consortium members are:

Partner 1	<i>AIT AUSTRIAN INSTITUTE OF TECHNOLOGY GMBH (AIT, Project Coordinator, AT)</i>
Contact person:	Andreas Stainer-Hochgatterer
Email:	Andreas.Stainer-Hochgatterer@ait.ac.at
Partner 2:	<i>UNIVERSITY OF GENEVA (UniGe, CH)</i>
Contact person:	Maher Ben Moussa
Email:	maher.benmoussa@unige.ch
Partner 3:	<i>UNIVERSITY OF CYPRUS (UCY, CY)</i>
Contact person:	George Samaras
Email:	cssamara@cs.ucy.ac.cy
Partner 4	<i>ORBIS MEDISCH EN ZORGCONCERN (ORBIS, NL)</i>
Contact person:	Cindy Wings
Email:	c.wings@orbisconcern.nl
Partner 5	<i>FRAUNHOFER IGD (Fh-IGD, DE)</i>
Contact person:	Carsten Stockl�w
Email:	carsten.stockloew@igd.fraunhofer.de
Partner 6	<i>Noldus Information Technology BV (Noldus, NL)</i>
Contact person:	Ben Loke
Email:	b.loke@noldus.nl
Partner 7	<i>CITARD SERVICES LTD (Citard, CY)</i>
Contact person:	Eleni Christodoulou
Email:	eleni_christodoulou@cytanet.com.cy
Partner 8	<i>ZOOBE MESSAGE ENTERTAINMENT GMBH (Zoobe, DE)</i>
Contact person:	Sascha Fagel
Email:	fagel@zoobe.com
Partner 9	<i>MAISON DE RETRAITE DU PETIT-SACONNEX (MRPS, CH)</i>
Contact person:	Donato Cereghetti
Email:	donato.cereghetti@hotmail.com

Table of Contents

<i>Release History</i>	<i>II</i>
<i>Miraculous-Life Consortium</i>	<i>III</i>
<i>Table of Contents</i>	<i>IV</i>
<i>Table of Figures</i>	<i>VII</i>
<i>List of Tables</i>	<i>VIII</i>
<i>Abbreviations</i>	<i>IX</i>
<i>Executive Summary</i>	<i>1</i>
<i>1 About this Document</i>	<i>2</i>
1.1 Role of the deliverable	2
1.2 Relationship to other Miraculous-Life deliverables	2
<i>2 Changes to previous deliverable version</i>	<i>3</i>
<i>3 Overall Miraculous-Life integration plan</i>	<i>4</i>
<i>4 Integration Setup</i>	<i>6</i>
4.1 Introduction to this chapter	6
4.2 Integration setup	6
<i>5 Requirements</i>	<i>8</i>
5.1 Introduction to this chapter	8
5.2 Hardware requirements	8
5.2.1 Server	8
5.2.2 Workstation	8
5.2.3 Tablet	9
5.2.4 Sensors	9
5.3 Software requirements	10
5.3.1 Server	10
5.3.2 Workstation	11
5.3.3 Tablet	11
5.4 Networking requirements	12
5.5 Summary of the chapter	13
<i>6 Installation setup</i>	<i>14</i>
6.1 Introduction to this chapter	14
6.2 Miraculous-Life server setup	14
6.2.1 Operating System	14
6.2.2 TeamViewer	14
6.2.3 Remote Desktop Connection	14

6.2.4	Microsoft SQL Server	15
6.2.5	Erlang	22
6.2.6	RabbitMQ Server	22
6.2.7	Java SE SDK	25
6.3	Miraculous-Life workstation setup	25
6.3.1	Operating System	25
6.3.2	Java SE SDK	25
6.3.3	Mongoose Webserver	26
6.3.4	Microsoft speech platform SDK	26
6.3.5	Microsoft speech platform runtime	26
6.3.6	Microsoft speech platform languages	26
6.4	Miraculous-Life tablet setup	26
6.5	Asterisk (PBX) Server setup	27
6.5.1	Installation of AsteriskNow	27
6.5.2	Asterisk configuration files	27
6.5.3	Checking out the sound files	28
6.6	Asterisk endpoint and client setup	28
6.6.1	Asterisk server related configuration files in the Miraculous-Life project	29
6.7	Summary of the chapter	29
7	<i>Miraculous-Life system integration</i>	30
7.1	Introduction to this chapter	30
7.2	Apache Karaf OSGi runtime	30
7.2.1	General	30
7.2.2	Hardware Requirements	30
7.2.3	Usage information	30
7.2.4	Useful commands	31
7.2.5	Installing OSGi bundles	31
7.3	Noldus Communication Framework	31
7.4	Dialog Manager	31
7.4.1	Software Requirements	32
7.4.2	Installation	32
7.5	Home daily activity and safety services	32
7.5.1	Hardware Requirements	32
7.5.2	Software Requirements	32
7.5.3	Installation	33
7.6	KnowledgeBase manager	33

7.7	Speech Recognition Engine setup	33
7.7.1	Hardware Requirements	33
7.7.2	Software Requirements	34
7.8	Zoobe Server setup	34
7.9	User Interface	34
7.9.1	General	34
7.9.2	Integration description	35
7.10	Summary of the chapter	36
8	<i>Miraculous-Life System deployment</i>	37
8.1	Introduction to this chapter	37
8.2	All in one deployment server	37
8.3	All in one deployment workstation	37
9	<i>Development</i>	39
9.1	Environmental setup	39
9.1.1	Java JDK 7	39
9.1.2	Apache Maven	39
9.1.3	IDEs	39
9.1.4	Version Control	39
9.1.5	Source	40
9.1.6	Android SDK	40
9.2	Building the Miraculous-Life project	41
9.3	Importing into Eclipse	41
9.3.1	Importing Miraculous-Life Maven project	41
9.3.2	Importing Android project	43

Table of Figures

Figure 1: System integration components	6
Figure 2: Adding a new user at the RabbitMQ server	23
Figure 3: Adding a virtual host at the RabbitMQ server	24
Figure 4: Adding user and permissions to the virtual hosts	24
Figure 5: After adding the user and virtual hosts at the RabbitMQ server	25
Figure 6: Miraculous-Life Android App start screen	35
Figure 7: User Interface showing the "home screen" with the Avatar as main interaction element. Available services can be seen on the right hand side.	36
Figure 8: Eclipse Import dialog	41
Figure 9: Eclipse Import dialog – selecting projects	42
Figure 10: Eclipse package explorer	42
Figure 11: Eclipse import dialog – selecting Miraculous-Life Android project	43
Figure 12: Adding Crosswalk library project to Miraculous-Life Android project	44

List of Tables

Table 1: This table shows the relationship to other deliverables	2
Table 2: Deliverable changes to previous deliverable version	3
Table 3: Detailed Miraculous-Life integration plan and planning	4
Table 4: Minimum system requirements for the Miraculous-Life Server components	8
Table 5: Minimum system requirements for the Miraculous-Life workstation component	8
Table 6: Tablet specification	9
Table 7: Kinect for Windows v2 PC hardware requirements	9
Table 8: Miraculous-Life Server software components	10
Table 9: Miraculous-Life workstation software components	11
Table 10: Miraculous-Life tablet software components	11
Table 11: Miraculous-Life network configuration	12

Abbreviations

<i>Abbrev.</i>	<i>Description</i>
AAL	Ambient Assisted Living
LAN	Local Area Network
SDK	Software Development Kit
APK	Android Application Package
NCF	Noldus Communication Framework
UI	User Interface
HTML	Hyper Text Markup Language
PC	Personal Computer
IP	Internet Protocol

Executive Summary

This deliverable describes the system integration of the different components and elements, which are introduced in D5.1, developed during the project into a complete system. In the first version, the installation process of the components relevant for the pre-trials, at which the current setup will be evaluated, is described.

Detailed information about the hardware and software requirements as well as specific installation and configuration procedures are provided by the responsible partners. The overall system requirements are further derived from these specifications.

The system integration chapter describes how the different modules and components are integrated to a complete and running system. Finally, a description of how to deploy the whole system on the server, the workstation and on the tablet is provided in the deployment chapter.

In the second version of this deliverable a detailed integration plan towards the next trial phase is provided. Furthermore, a chapter about the Asterisk Server setup has been added as well as a chapter about setting up a proper development environment for the Miraculous-Life system. Based on the ongoing development, some changes were applied to distinct chapters like hardware and software requirements. Chapter 2 gives an overview about the changes to the previous deliverable version.

1 About this Document

1.1 Role of the deliverable

This deliverable covers the integration of all developed components and modules that are described in the overall system architecture in D5.1 into a complete and running system.

1.2 Relationship to other Miraculous-Life deliverables

The deliverable is related to the following Miraculous-Life deliverables:

Table 1: This table shows the relationship to other deliverables

<i>Deliv:</i>	<i>Relation</i>
D5.1	Specification of overall system architecture and security and privacy infrastructure. D5.3 describes the assembly and integration of the different components and modules presented in D5.1.

2 Changes to previous deliverable version

Table 2 provides an overview of the content changes of this deliverable version to version D5.3a.

Table 2: Deliverable changes to previous deliverable version

Deliverable changes	
Action	Title
Add	Chapter 3 Overall Miraculous-Life integration plan
Add	Chapter 5.2.4 Sensors
Add	Chapter 5.2.4.1 Kinect for Windows v2
Add	Section 6.5 Asterisk (PBX) Server setup
Add	Section 6.6 Asterisk endpoint and client setup
Add	Chapter 9 Development
Update	Figure 1: System integration components: added the Asterisk Server component
Update	Chapter 4 Integration Setup: added information about the Asterisk Server
Update	Section 5.2.1 Hardware requirements Server: more RAM and better CPU
Update	Section 5.2.2 Hardware requirements Workstation: more RAM and better CPU
Update	Section 5.3.1 Software requirements Server: added Windows Server 2012 R2 Essentials
Update	Section 5.3.2 Software requirements Workstation: Microsoft speech SDK and runtime 64Bit
Update	Section 5.3.2 Software requirements Tablet: Android v4.4, Miraculous-Life App v0.6
Update	Section 5.4 Networking requirements: added Tablet – Asterisk Server configuration to Table 11
Update	Section 6.2.6 RabbitMQ Server: detailed installation and setup description
Update	Section 6.2.7 Java SE SDK: detailed configuration description
Update	Section 6.4 Miraculous-Life tablet setupJava SE SDK: added Android App deploy description
Update	Section 7.9.1 User Interface General: new UI images

3 Overall Miraculous-Life integration plan

Table 3 shows the detailed planning for the 3rd pre-trials, which starts right after the 2nd pre-trials are finished.

Table 3: Detailed Miraculous-Life integration plan and planning

Pre-trial	Start (in PM)	Duration (in calendar dates)	What	Extra information
	M17 (April)	20 th – 21 st April	New prototype of ML installed at both end users locations	
2nd	M17 (April)	22 nd – 23 rd April	Expert based evaluation second pre-trial	Technical partners will receive direct feedback from this to update the system before the controlled user based evaluation.
	M18 (May)	11 th May – 22 nd May	user based evaluation second pre-trial	End-users need 2 weeks to plan all elderly and caregivers. Open issues can be directly reported in the issue tracker but the complete analyses will take at least one month.
	M19 (June)	30 th June	First 3 rd pre-trial version of KB	KB script is available in SVN for execution at AIT server (and other)
	M19 (June)	30 th June	Deadline for services for 3 rd pre-trials ¹	Services a tested and available in the SVN (for the services specified see table below). Please send it one week before to be integrated in the service manager.
	M20 (July)	15 th July	Feedback from AIT and UniGe if something missing / updates regarding KB	
	M20 (July)	15 th July	Deadline for service components	Components which are necessary to realize the services with dependencies are delivered (Fh-IGD and Noldus)
	M20 (July)	30 th July	Deadline for Videos	WP6 will provide the videos
	M20 (July)	31 st July	Emotion Fusion components	According to the interface specified by WP3
	M20 (July)	1 st July – 30 th September	Service testing and Dialogue Integration	Three months of service tests and Dialogue Manager integration. Constant updates of service APIs as well as KB

¹ Services without dependencies (see table service planning)

	M21 (August)	2 nd August	Deadline for services for 3rd pre-trials ²	Services are tested and available in the SVN (for the services specified see table below). Please send it one week before to be integrated in the service manager.
	M21 (August)	15 th August	Feedback from AIT and UniGe if something missing / updates regarding KB	
	M22 (September)	17 th September – 7 th October	Closed Beta: Pre-testing by all technical partners (especially Noldus and UniGe)	All technical partners test the system intensively. Especially the technical partners UniGe and Noldus who are responsible for MRPS and Orbis. Installations in UniGe as well as Noldus in this time are required.
	M23 (October)	8 th October – 14 th October	Issues to be solved based on issue tracker – Closed Beta Bug Fixing	
	M23 (October)	15 th October – 28 th October	Open Beta: Pre-testing of experts at Orbis and MRPS	
	M24 (November)	29 th October – 8 th November	Issues to be solved based on issue tracker – Open Beta Bug Fixing	
3rd	M24 (November)	9 th November – 16 th November	expert based evaluation of third pre-trial	
	M24 (November)	17 th November – 30 th November	user based evaluation third pre-trial	The final outcomes and analyses will be shown in D1.4b and D6.4b

² Services with dependencies (see table service planning)

4 Integration Setup

4.1 Introduction to this chapter

This chapter describes the integration setup of the Miraculous-Life system.

4.2 Integration setup

Figure 1 shows the integration setup of the Miraculous-Life system. The integration setup is optimized for different requirements of the setup (residential houses, private living flats etc.). The system can be easily setup and deployed for a various numbers of clients.

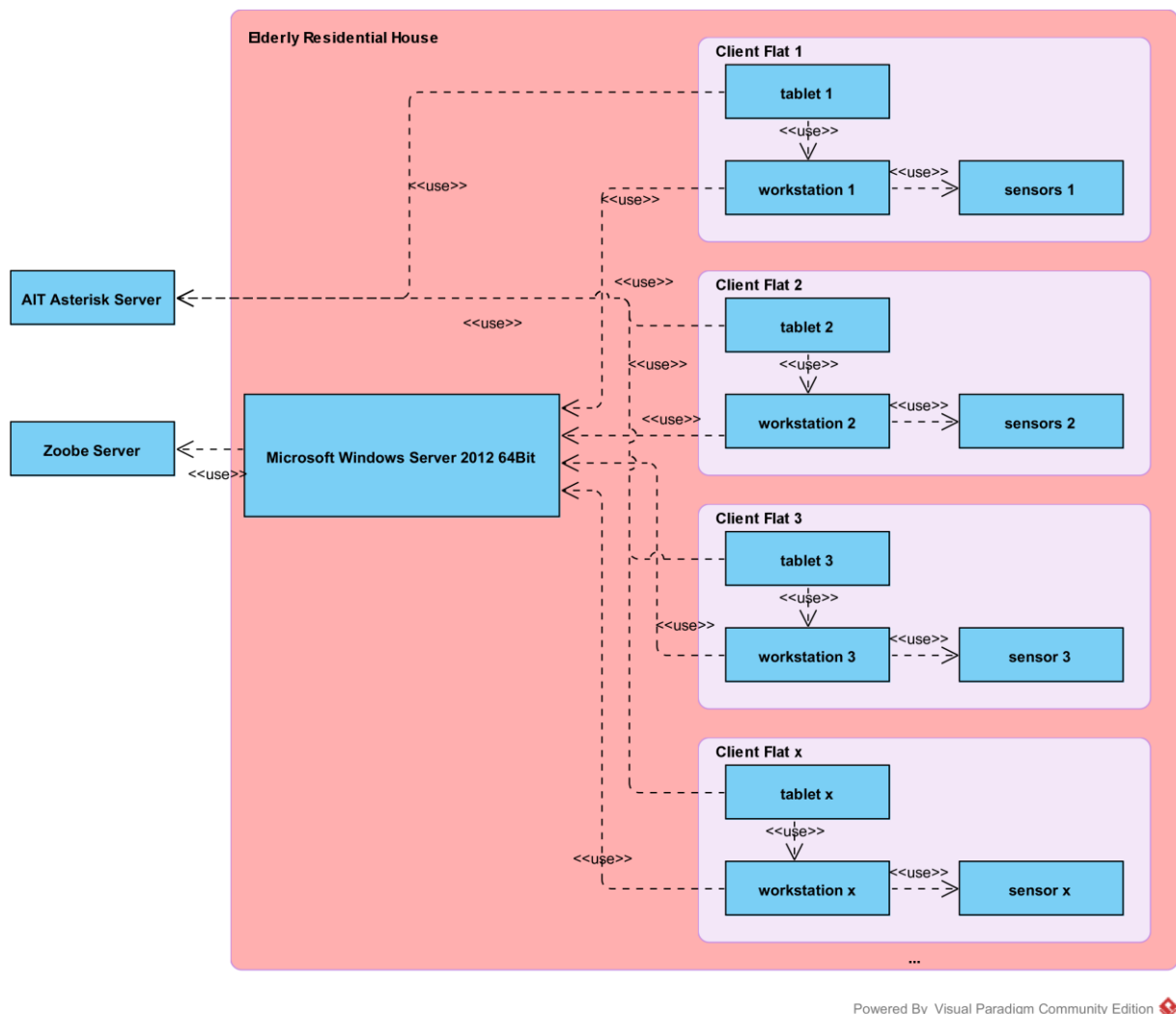


Figure 1: System integration components

The Microsoft Windows Server 2012 is a computer at a central place in the residential house or any other setup. Important is that it is a local server which runs all the (web) services and contains the knowledge base. Therefore, it is assured that all private data are stored locally. Another advantage is that this server provides the services to many clients. Hence, this server is the place where services can be updated for all clients at once and the performance of the system can be optimized in a central location. The server will either run a Microsoft Windows Desktop or Server edition as operating system.

Additionally, an Asterisk Server is used which provides the phone call and SMS functionality. This telephone Server can call ordinary phone numbers as well as internal

client numbers. The clients are connected to the Asterisk Server using the SIP protocol. The whole calling functionality on the client side is already integrated into the HTML5 interface, and is therefore not a native functionality of the tablet's operating system. For the 3rd pre-trials, one Asterisk Server will be used to handle the call functionality on both trial sites. Depending on the performance and outcome of multi-user tests, the Miraculous-Life system will either run with one Asterisk Server accessible by all installations or with locally installed Asterisk Server instances.

Different client flats are connected to the Windows Server. In each of them, a small local PC (i.e. a workstation) is running, to which the sensors installed in the flats are connected.

The user is accessing the services and the avatar via a tablet. Since the avatar is rendered as an HTML5 interfaces, the choice and the operating system of the tablets is very flexible. In the current deployment we use tablets running Android version 4.4.

In the current setup, the only server outside the local environment and network is the server provided by Zoobe which performs the avatar generation and rendering as well as the emotion recognition based on speech.

5 Requirements

5.1 Introduction to this chapter

This chapter gives an overview about the hardware and software requirements for the system integration setup.

5.2 Hardware requirements

For a general description of the used hardware components, please refer to Deliverable 5.1, chapter 3.2.

5.2.1 Server

The Miraculous-Life Server must maintain the database as well as the home daily activity and safety services. Note that some services will be split up based on their usage in the system. For example, a component from a safety service might need direct access to the Kinect sensor attached to the workstation. In this case, this safety service components is located on the workstation and communicates over the local network with the more general services located on the server. The Server is a core component of the whole system and has to be accessible up to 100% of the time. During this time, the server may not run full with RAM and hard disk space. Hence, we recommend the system specifications (minimum requirements) listed in Table 4.

Table 4: Minimum system requirements for the Miraculous-Life Server components

Minimum Server System requirements	
CPU	Intel i5 quad core or Intel Xeon processor
RAM	12GB
HDD (Data)	2TB
SSD (System)	240GB
Network	(W-)LAN Internet access

Although these specifications are already chosen higher than needed for the second prototype, they may be adapted during the project runtime, based on the further developments and requirements.

5.2.2 Workstation

Table 5 shows the current minimum system specification for the workstation based on the known components so far.

Table 5: Minimum system requirements for the Miraculous-Life workstation component

Minimum workstation system requirements	
CPU	Intel i5-4200U 2x1.6GHz
RAM	12GB

SSD	240GB
Network	(W-)LAN Internet access
I/O	USB3.0

5.2.3 Tablet

Table 6 shows the recommended tablet specification based on our internal testing process.

Table 6: Tablet specification

Tablet system requirements	
CPU	Dual Core
RAM	2GB
Memory	16/32GB
Network	W-LAN
Features	Webcam (front) 2MP Integrated Microphone

5.2.4 Sensors

5.2.4.1 Kinect for Windows v2

Table 7 shows the hardware requirements of a PC in order to support the usage of the Kinect sensor. These requirements are taken from Microsoft's official hardware requirements webpage:

https://www.microsoft.com/en-us/kinectforwindows/purchase/sensor_setup.aspx

Table 7: Kinect for Windows v2 PC hardware requirements

Kinect for Windows v2 PC hardware requirements	
CPU	Physical dual-core 3.1 GHz (2 logical cores per physical) or faster processor, 64 Bit
USB	USB 3.0 controller dedicated to the Kinect for Windows v2 sensor
RAM	4GB
Graphics card	Support for DirectX 11
OS	Windows 8 or 8.1, or Windows Embedded 8

5.3 Software requirements

For a general description of the used software components, please refer to Deliverable 5.1, chapter 3.3.

5.3.1 Server

Table 8 shows the software components used by the Miraculous-Life Server.

Table 8: Miraculous-Life Server software components

Server software		
Name	Version	Information
Microsoft Windows	2012 R2 Essentials Server, 64 Bit	Operating System Windows Server will be used.
Microsoft SQL Server	2008 R2 Developer Edition	Database
TeamViewer	≥ 9	Remote access for maintenance
Remote Desktop Connection (RDP; optional)	Integrated Windows remote desktop connection	Alternative and/or additional way for remote maintenance
Erlang	≥ OTP 17.3	Needed for RabbitMQ Server
RabbitMQ Server	≥ 3.3.5	Messaging Server for inter-component communication via NCF
RabbitMQ management plugin	-	Management plugin for easy setup and maintenance of RabbitMQ via a web interface
Java SE SDK	≥ 7	Java runtime to run the Miraculous-Life Karaf framework

The software requirements for the Server may be adapted during the project development, which will be reflected in upcoming versions of this deliverable.

5.3.2 Workstation

Table 9 shows the software components used by the Miraculous-Life workstation.

Table 9: Miraculous-Life workstation software components

Workstation software		
Name	Version	Information
Microsoft Windows	8.1, 64 Bit	Operating System
Java SE SDK	≥ 7	Java runtime to run the Miraculous-Life Karaf framework
Mongoose WebServer	≥ 5.4.1 free edition	WebServer that provides the user interface logic
Microsoft speech platform SDK	≥ 11, 64Bit	Part of the speech recognition engine used in Miraculous-Life http://www.microsoft.com/en-us/download/details.aspx?id=27226
Microsoft speech platform runtime	≥ 11, 64Bit	Part of the speech recognition engine used in Miraculous-Life https://www.microsoft.com/en-us/download/details.aspx?id=27225
Microsoft speech platform languages	≥ 11	Part of the speech recognition engine used in Miraculous-Life http://www.microsoft.com/en-us/download/details.aspx?id=27224 Supported languages up until now: Dutch English French

5.3.3 Tablet

Table 10 shows the software components used by the Miraculous-Life tablet.

Table 10: Miraculous-Life tablet software components

Tablet software		
Name	Version	Information
Android	≥ 4.4	Tablet operating system
Miraculous-Life App	≥ 0.6	Miraculous-Life Android App

5.4 Networking requirements

The Miraculous-Life server and the workstation need to have a working internet connection. Furthermore, the tablet(s) and the associated workstation have to be in the same LAN. Table 11 shows the necessary network port configuration in order for the system to work properly.

Table 11: Miraculous-Life network configuration

Network configuration			
From – To	Protocol	Port	Information
Workstation – Server	TCP	5672	RabbitMQ
Workstation – Server	TCP	15672	RabbitMQ Management
Server – Internet	TCP	443	HTTPS For connecting to Zoobe Server
Tablet – Workstation	TCP	8080	WebServer connection (HTTP); displays the UI (HTML)
Tablet – Workstation	TCP	51010	WebSocket connection (WS); handles the communication and UI content update
Tablet – Asterisk Server	UDP	5060	Asterisk SIP connection

The ports 8080 (WebServer) and 51010 (WebSocket) can be adjusted to personal needs by the following configuration files:

- Port 8080:**
 Server side: Modify the *mongoose.conf* file (which can be found next to the mongoose executable) and set a custom value for “*listening_port*”. Note that this port must not be used by another program, otherwise the webserver won’t start.

 Client side: Use the aforementioned port when starting the Miraculous-Life Android App in the start screen.
- Port 51010:**
 Server side: Modify the config file *at.ac.ait.hbs.homer.core.event.connector.websocket.cfg* located in the *framework/etc/* folder and set a custom value for “*server.port*”. Note that this port must not be used by another program;
 Client side: Modify the JavaScript file *main.appframework.js* located in the *UI/js/* folder and set the same port value for “*websocketPort*”. Note that in order for this change to take effect, the Miraculous-Life App should be restarted.

5.5 Summary of the chapter

The shown requirements in this chapter are good estimations, but might still change to some extent as this project is still under development. Any upcoming changes on one of the three areas (software, hardware, network) will be incorporated in future releases of this deliverable.

6 Installation setup

6.1 Introduction to this chapter

This chapter describes the installation process of the Miraculous-Life software components for each device, which are the server, workstation and the tablet. As it is out of scope of this document to give a detailed description of, for example, setting up an operating system or installing ordinary software, only the more crucial installation and configuration parts will be covered here. In case that there are no additional information written in this document, the reader is advised to follow the installation instructions provided by the corresponding software provider.

In general, it is recommended that the server and workstation is setup by an appropriate person, which already has experience setting up computer systems and networks.

6.2 Miraculous-Life server setup

It is suggested that the user who is accessing the server remotely has administrator privileges in order to install, configure and uninstall certain programs.

6.2.1 Operating System

After installing the operating system by following the official guideline, additional configuration steps should be taken in order to have the Server properly configured. This information will be given in the next version of this deliverable.

6.2.2 TeamViewer

TeamViewer is used for accessing and maintaining the server from a remote PC. The following options should be set in order to ensure a flawless remote interaction. Open the TeamViewer options via *Extras->Options*. Each topic on the left hand side of the options window is covered by one list element.

- General
 - Check “*Start TeamViewer with Windows*” to install TeamViewer as a service, which enables you to access and login to the remote machine without the need of a logged in user.
- Security
 - Set a strong password for the remote access connection
 - Set the *Windows logon* setting to “*Allowed for administrators only*”
- Advanced → Show advanced settings
 - *Access control*: “*Full Access*”
 - Under the *More* heading, check the “*Disable TeamViewer shutdown*” option, to prevent users to shut down the TeamViewer instance manually.

6.2.3 Remote Desktop Connection

Using the built-in remote control functionality from Microsoft Windows would be an alternative to the TeamViewer installation. Depending on the Windows version, the configuration might vary. Please refer to the official Microsoft user guide for enabling this functionality.

6.2.4 Microsoft SQL Server

This section describes the installation prerequisites and guides of the KnowledgeBase database. More specifically, it includes the system hardware and software requirements followed by the installation process. The installation process includes a step by step guide of the Microsoft SQL Server 2008 R2 Database Server, followed by the installation of the Miraculous KnowledgeBase database (MiraculousKB) and the initialization of the access user.

6.2.4.1 Hardware requirements

The hardware requirements for the KnowledgeBase services are presented in this section. Those requirements sets are selected to provide a sufficient performance of the software. The KnowledgeBase installation has as prerequisite the Windows SQL Server 2008 R2 thus we present the minimum requirements for both standard versions (i.e. 32 and 64-bit).

Hardware:

The following table shows the minimum system requirements for SQL Server 2008 R2 Standard (32-bit):

Component	Requirement
Processor	Processor type: <ul style="list-style-type: none">• Pentium III-compatible processor or faster Processor speed: <ul style="list-style-type: none">• Minimum: 1.0 GHz• Recommended: 2.0 GHz or faster
Memory	RAM: <ul style="list-style-type: none">• Minimum: 1 GB• Recommended: 4 GB or more• Maximum: 64 GB
Hard Disk Space Requirements	At least 3.6 GB of available disk space on the system drive
Drive	A CD or DVD drive, as appropriate, is required for installation from disc.
Display	SQL Server 2008 R2 graphical tools require Super VGA or higher resolution: at least 800x600 pixel resolution.

The following table shows system requirements for SQL Server 2008 R2 Standard (64-bit) x64:

Component	Requirement
Processor	Processor type: <ul style="list-style-type: none"> • Minimum: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support Processor speed: <ul style="list-style-type: none"> • Minimum: 1.4 GHz • Recommended: 2.0 GHz or faster
Memory	RAM: <ul style="list-style-type: none"> • Minimum: 1 GB • Recommended: 4 GB or more • Maximum: 64 GB
Hard Disk Space Requirements	At least 3.6 GB of available disk space on the system drive
Drive	A CD or DVD drive, as appropriate, is required for installation from disc.
Display	SQL Server 2008 R2 graphical tools require Super VGA or higher resolution: at least 800x600 pixel resolution.

6.2.4.2 Software requirements

This section describes the platform requirements to install the KnowledgeBase. Following the example of the hardware requirements, both versions of SQL Server are presented.

The following table shows the minimum system requirements for SQL Server 2008 R2 Standard (32-bit):

Component	Requirement
Operating system	Windows XP Professional SP3 Windows Server 2003 (not the standard edition) Windows Vista SP2 (not the standard edition) Windows Server 2008, R2 Windows Server 2012, R2 Windows 7 (not the standard edition) Windows 8/8.1 32-bit, 64-bit
Framework	SQL Server Setup installs the following software components required by the product: <ul style="list-style-type: none"> • NET Framework 3.5 SP1 • SQL Server Native Client

	<ul style="list-style-type: none"> • SQL Server Setup support files
Software	SQL Server Setup requires Microsoft Windows Installer 4.5 or a later version. After installing required components, SQL Server Setup will verify that the computer where SQL Server 2008 R2 will be installed also meets all the other requirements for a successful installation.
Network Software	Supported operating systems have built-in network software. Stand-alone named and default instances support the following network protocols: <ul style="list-style-type: none"> • Shared memory • Named Pipes • TCP/IP • VIA
Internet Software	Microsoft Internet Explorer 6 SP1 or a later version is required for all installations of SQL Server 2008 R2. Internet Explorer 6 SP1 or a later version is required for Microsoft Management Console (MMC), SQL Server Management Studio, Business Intelligence Development Studio, the Report Designer component of Reporting Services, and HTML Help.

The following table shows system requirements for SQL Server 2008 R2 Standard (64-bit) x64:

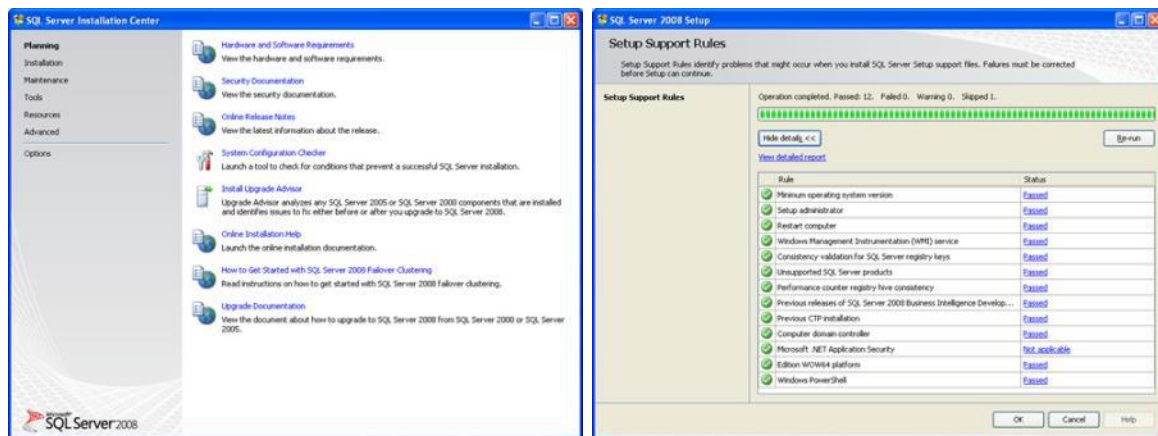
Component	Requirement
Operating system	Windows XP Professional SP2 x64 Windows Server 2003 (not the standard edition) Windows Vista SP2 (not the standard edition) Windows Server 2008 64-bit, R2 Windows Server 2012 64-bit, R2 Windows 7 64-bit (not the standard edition) Windows 8/8.1 64-bit
Framework	SQL Server Setup installs the following software components required by the product: <ul style="list-style-type: none"> • NET Framework 3.5 SP1 • SQL Server Native Client • SQL Server Setup support files
Software	SQL Server Setup requires Microsoft Windows Installer 4.5 or a later version. After installing required components, SQL Server Setup will verify that the computer where SQL Server 2008 R2 will be installed also meets all the other requirements for a successful installation.
Network	Supported operating systems have built-in network software.

Software	<p>Stand-alone named and default instances support the following network protocols:</p> <ul style="list-style-type: none"> • Shared memory • Named Pipes • TCP/IP • VIA
Internet Software	<p>Microsoft Internet Explorer 6 SP1 or a later version is required for all installations of SQL Server 2008 R2. Internet Explorer 6 SP1 or a later version is required for Microsoft Management Console (MMC), SQL Server Management Studio, Business Intelligence Development Studio, the Report Designer component of Reporting Services, and HTML Help.</p>

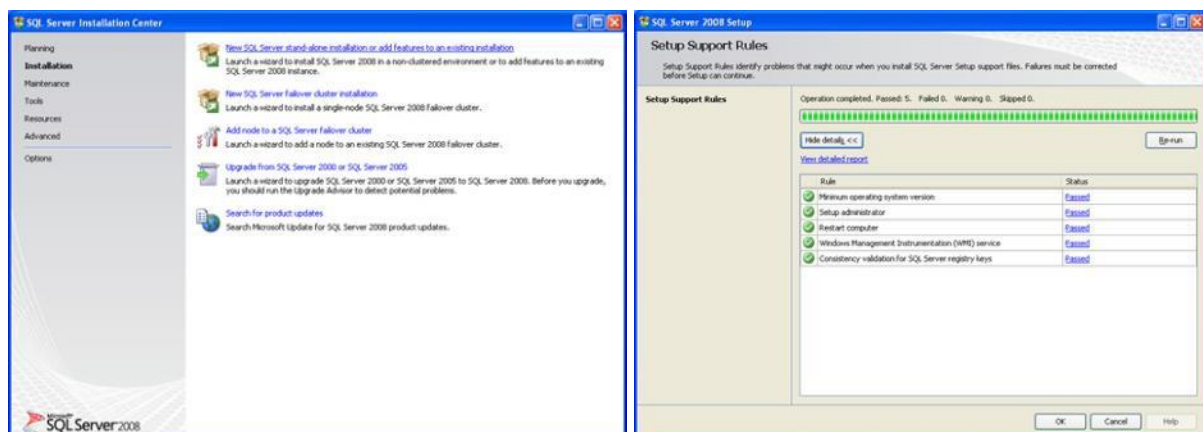
6.2.4.3 Installation of the SQL Server 2008 R2 Database Server

In the following pages we describe the installation procedure for the SQL Server 2008 R2 database server. We accompany the description of each subsequent step with figures.

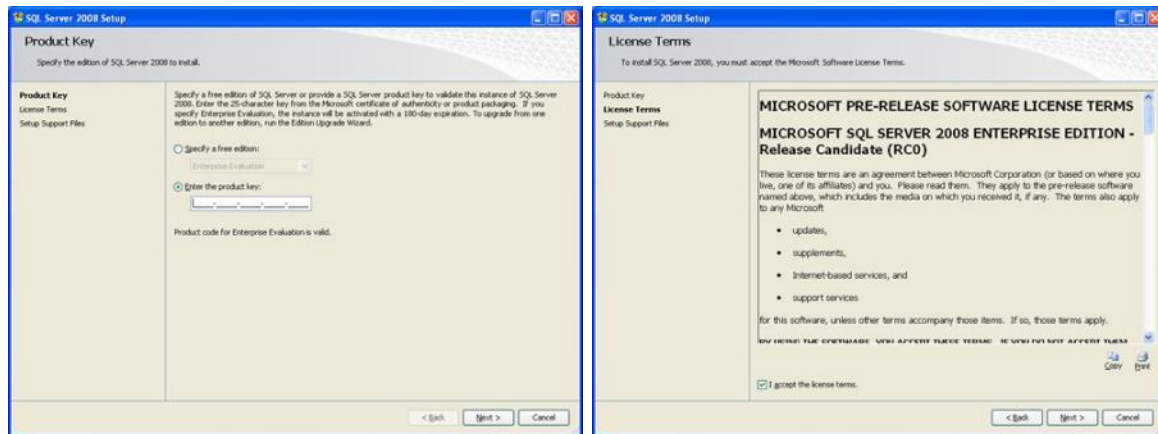
By inserting the Installation CD of the SQL Server 2008 R2, the setup process will start. In order for the installation to proceed we select the Hardware and Software Requirements option.



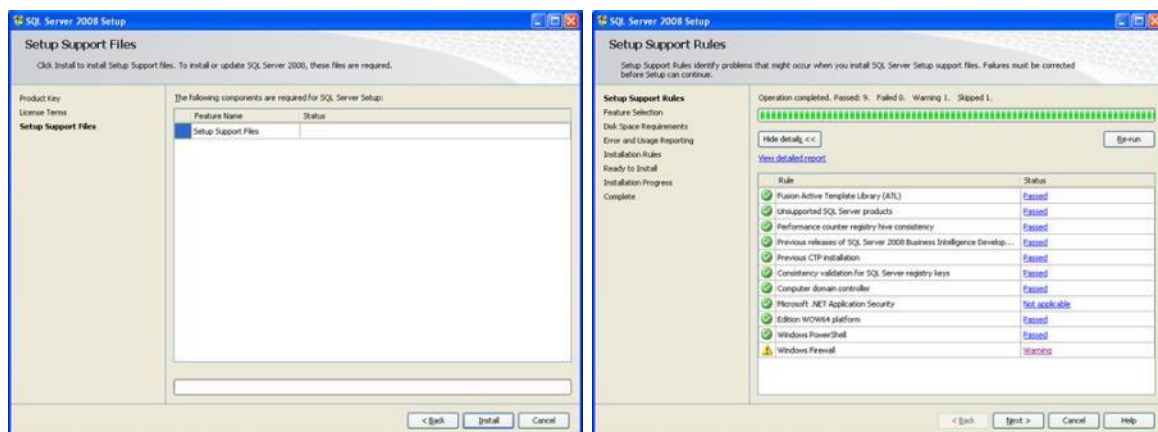
When this process finishes and passes all checks, we select the option for creating a New SQL Server stand-alone installation.



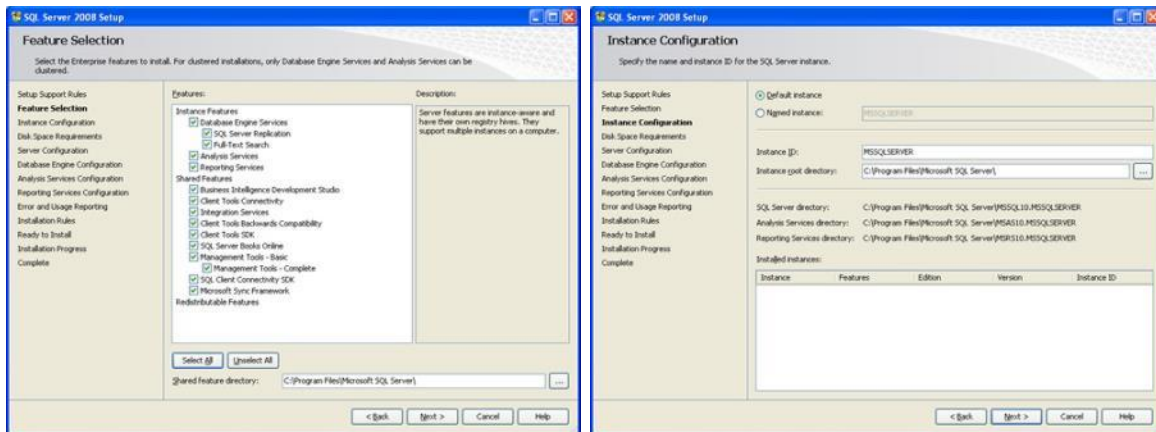
When the second pass of checks finishes, the installation will ask the user to enter a valid product key. After entering the Product key, the system will ask the user to accept the license terms. In order to do that, the user checks the box “I accept the license terms” and presses the “Next” button.



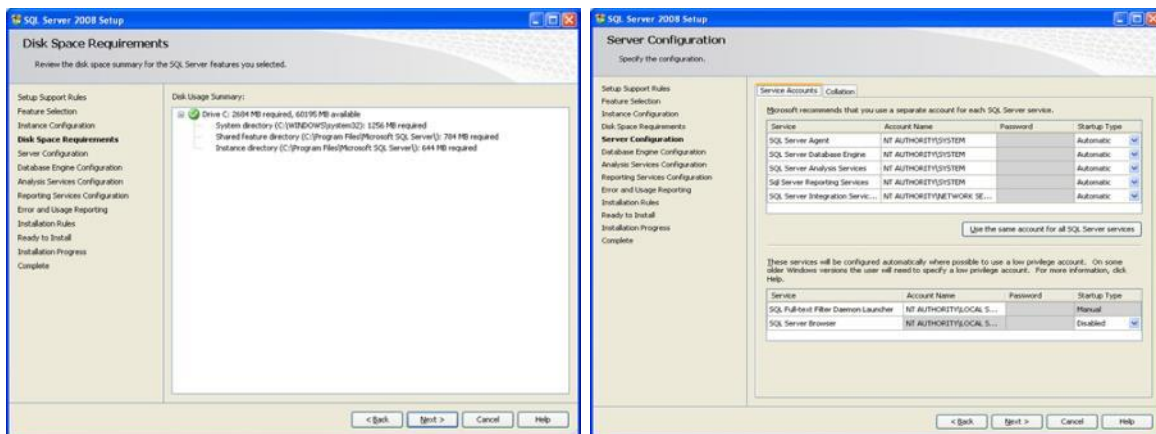
The installation will continue with the copy procedure of the Setup Support Files by pressing the “Install” button.



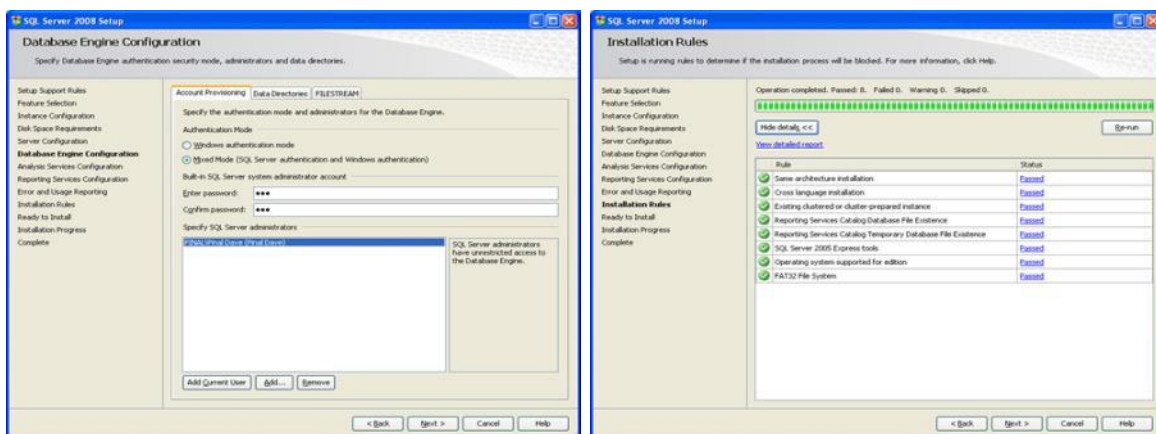
Next, the installation will ask the user to confirm the installation components that the user wants to install. We suggest skipping the selections of Analysis and Reporting Services if they are not required. The installation will ask to enter the name of the Instance of the SQL Server. This practically is the name of the server and it has to be unique. The default is MSSQLSERVER.

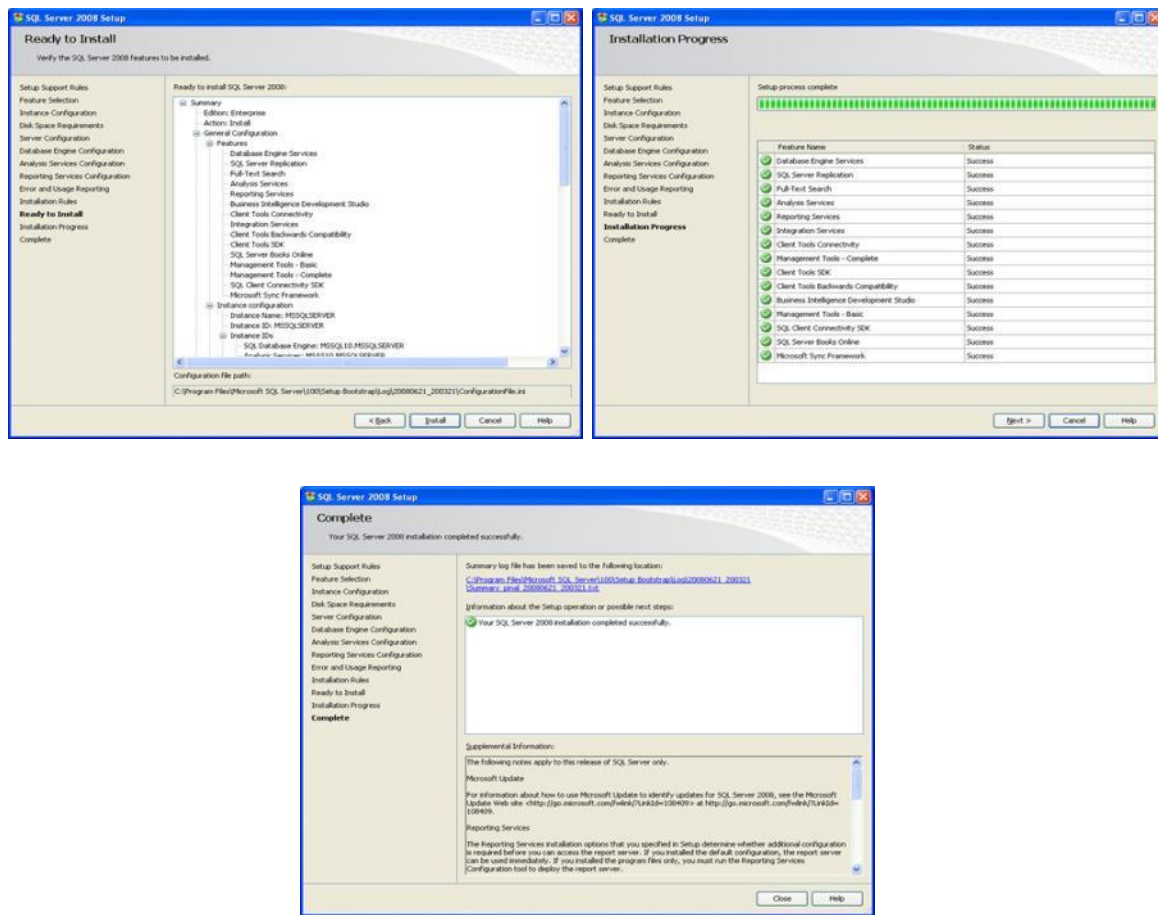


During the next step, the installation will display the Disk Space Requirements according to the previous selections. By pressing the “Next” button, the System Configuration menu will appear. We suggest keeping the default options.



One of the most important steps is the Database Engine Configuration. During this step, you will select the **Mixed Mode** as Authentication Mode and provide a strong password. After this step the installation will start, reporting periodically the status and asking for confirmations. If any errors are found, the installation will provide support for each error.





6.2.4.4 Installation of the Miraculous KnowledgeBase Database

In order to attach the Miraculous database please perform the following steps:

Please note that the mentioned database files below are not yet publicly available due to the ongoing development process. As soon as they are available, the download information will be updated here.

1. Download the database files (MiraculousKB.mdf) and (MiraculousKB_log.ldf) and copy them to the SQL Server Data folder. By default this folder is located in the SQL Server installation folder under the "MSSQL\DATA\" path.
2. Open the SQL Server Management Studio Object Explorer, connect to the instance of the Microsoft SQL Server Database Engine, and then expand that instance.
3. Right-click **Databases** and click **Attach**.
4. In the **Attach Databases** dialog box, to specify the database to be attached, click **Add**; and in the **Locate Database Files** dialog box, select the disk drive where the database resides and expand the directory tree to find and select the .mdf file of the database; for example:
C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA\MicaculousKB.mdf
5. Optionally, to specify a different name for the database to attach as, enter the name in the **Attach as** column of the **Attach Databases** dialog box.
6. Optionally, change the owner of the database by selecting a different entry in the **Owner** column.
7. When you are ready to attach the database, click **OK**.

6.2.4.5 Initialization of the Miraculous KnowledgeBase Database

In order to provide secure access through its API, the final step of the KnowledgeBase installation procedure is to create user that has access only to see the data of views (not the base tables) and to execute stored procedures.

The following script can be used to create the user automatically.

Note that for the example we have named this user as ml_user.

```
USE [MiraculousKB]
GO
CREATE USER [ml_user] FOR LOGIN [ml_user] WITH DEFAULT_SCHEMA=[dbo]
GO
DECLARE @SQL NVARCHAR(MAX)
DECLARE @NAME NVARCHAR(MAX)
DECLARE VIEW_CURSOR CURSOR FOR
SELECT name
FROM sys.views
WHERE NOT name='sysdiagrams'
OPEN VIEW_CURSOR
FETCH NEXT FROM VIEW_CURSOR INTO @NAME
WHILE @@FETCH_STATUS=0
BEGIN
    SET @SQL = 'GRANT SELECT ON dbo.' + @NAME + ' TO [ml_user]'
    EXECUTE sp_executesql @SQL
    FETCH NEXT FROM VIEW_CURSOR INTO @NAME
END
CLOSE VIEW_CURSOR
DEALLOCATE VIEW_CURSOR
DECLARE PROC_CURSOR CURSOR FOR
SELECT name
FROM sys.procedures
OPEN PROC_CURSOR
FETCH NEXT FROM PROC_CURSOR INTO @NAME
WHILE @@FETCH_STATUS=0
BEGIN
    SET @SQL = 'GRANT EXECUTE ON dbo.' + @NAME + ' TO [ml_user]'
    EXECUTE sp_executesql @SQL
    FETCH NEXT FROM PROC_CURSOR INTO @NAME
END
CLOSE PROC_CURSOR
DEALLOCATE PROC_CURSOR
```

6.2.5 Erlang

No additional configuration steps are necessary after installing Erlang by following the official guideline.

6.2.6 RabbitMQ Server

No additional configuration is necessary during the installation process according to the official guideline. After successful installation, a user miraculouslife (with password miraculouslife) and two virtual hosts must be created for the RabbitMQ server. This can be done by using the management plugin for RabbitMQ. To do so, open the command line (cmd) and navigate to the folder where rabbitmq-plugins.bat file is located (by default it is in "C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.1\sbin"). Then execute this command:

```
rabbitmq-plugins enable rabbitmq_management
```

If it is executed successfully, messages saying that some of the plugins are enabled/activated will be displayed. Then, open the following address on the browser: localhost:15672. If it does not show anything, try the troubleshooting below.

Troubleshooting:

If the localhost:15672 does not open, try to restart the rabbitmq-server:

```
rabbitmqctl stop
rabbitmq-server start
```

Then try to open the address again on the browser.

Once the address opens successfully, login to the web interface (username: `guest`, password: `guest`). Click on **Admin** tab and select **Users** from the menu on the right side. Under the “**Add user**” section, add a new user as shown in figure below:

Username: `miraculouslife`

Password: `miraculouslife`

Tags: `admin` (administrator)

The screenshot shows the 'Users' management page in the RabbitMQ web interface. At the top, there's a section for 'All users' with a filter and a 'Regex (?)' checkbox. Below this is a table listing existing users. The table has columns: Name, Tags, Can access virtual hosts, and Has password. One user, 'guest', is listed with 'administrator' tags, access to all virtual hosts ('/'), and a password ('•').

Below the table is a section for 'Add a user'. It contains several input fields:

- Username:** A text input field containing 'miraculouslife'.
- Password:** Two password input fields, both containing 'miraculouslife'. The second field is labeled '(confirm)'.
- Tags:** A text input field containing 'administrator'.

 There are red asterisks (*) next to the Username, Password, and Tags fields, indicating they are required. Below the Tags field, there's a list of tag options: 'Set', 'Admin', 'Monitoring', 'Policymaker', 'Management', and 'None'.

At the bottom of the 'Add a user' section is a blue 'Add user' button. Below the button, there are links for 'HTTP API' and 'Command Line'.

Figure 2: Adding a new user at the RabbitMQ server

Then click on “**Virtual Hosts**” from the menu on the right side. And add two virtual hosts: 1 and 2.

Virtual Hosts

▼ All virtual hosts

Filter: ☐ Regex (?)

Overview		Messages			Network		Message rates		+/-
Name	Users (?)	Ready	Unacked	Total	From client	To client	publish	deliver / get	
/	guest, miraculouslife	0	0	0	0B/s	0B/s	0.00/s	0.00/s	
1	miraculouslife	?	?	?					
2	miraculouslife	?	?	?					

▼ Add a new virtual host

Name: *

[HTTP API](#) | [Command Line](#)

Figure 3: Adding a virtual host at the RabbitMQ server

When the virtual hosts are created, click on each one of them and add permissions and the user miraculouslife as shown in figure below:

▼ Permissions

Current permissions

User	Configure regexp	Write regexp	Read regexp
miraculouslife	*	*	*

Set permission

User:

Configure regexp:

Write regexp:

Read regexp:

Figure 4: Adding user and permissions to the virtual hosts

Then go back to the “**Users**” option of the menu on the right side, and it should display the results as shown in the following figure:

Users			
▼ All users			
Filter:	<input type="text"/>	<input type="checkbox"/> Regex (?)	
Name	Tags	Can access virtual hosts	Has password
guest	administrator	/	•
miraculouslife	administrator	/, 1, 2	•

Figure 5: After adding the user and virtual hosts at the RabbitMQ server

The following configuration files can be used to change settings of RabbitMQ and NCF:

- `deploy/dialogue_manager/ASR/ASRManager2.properties`
- `deploy/framework/etc/eu.miraculouslife.dialoguemanager.cfg`
- `deploy/framework/etc/eu.miraculouslife.homer.mlsystem.cfg`

Note that the *guest* login is only available when the server is accessed locally! Since RabbitMQ is only used in a local area network environment, the RabbitMQ server must not be accessible from the internet. Therefore it is suggested to block the ports 15672 and 5672 in your firewall for the internet connection.

6.2.7 Java SE SDK

After successful installation of the JDK, the JAVA_HOME environment variable must be set up. Go to My Computer, right click and choose Properties, select Advanced Settings on the left side. Under Advanced tab, go to Environment variables. Add a new variable to the System variables:

Name: `JAVA_HOME`

Path: Path to your JDK (for example `C:\Program Files\Java\jdk1.7.0_79`)

Then edit the Path variable – add `%JAVA_HOME%\bin` to the value of the variable. Please note that the variables are separated by semicolon, therefore add semicolon before adding this path. Also make sure that there is no other Java path added to the Path variable.

To make sure that JDK 7 was installed correctly, please check the java version by running the `java -version` command in the command line tool (cmd). The proper output for this Java version would be:

```
java version "1.7.0_79"
```

```
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
```

6.3 Miraculous-Life workstation setup

6.3.1 Operating System

No additional configuration is needed after installing the operating system by following the official guideline.

6.3.2 Java SE SDK

No additional configuration is needed after installing the Java SDK by following the official guideline and setting up environment variables as described in 6.2.7.

6.3.3 Mongoose Webserver

The webserver does not have to be installed in the system, since it's a single executable file. It is already configured as needed within the Miraculous-Life *deploy* structure and also starts automatically when the Miraculous-Life system is started.

6.3.4 Microsoft speech platform SDK

No additional configuration is needed after installing the Microsoft speech platform SDK by following the official guideline.

6.3.5 Microsoft speech platform runtime

No additional configuration is needed after installing the Microsoft speech platform runtime by following the official guideline.

6.3.6 Microsoft speech platform languages

No additional configuration is needed after installing the languages packs for the Microsoft speech platform by following the official guideline.

6.4 Miraculous-Life tablet setup

The setup for the tablet is rather straight forward, since only the developed Miraculous-Life app, provided as an apk file, has to be installed on the device. The latest version of the Android app is located in directory:

- `deploy\AndroidApp`

Using APK installer

There are many apps and programs for both Windows and Android OS that can install apk files on Android device. Any of them can be used to install the Miraculous-Life Android app.

Using Android SDK

If Android SDK is installed on the PC, the apk files can be installed on Android devices by using the command:

```
adb install NameOfApp.apk
```

from the directory in which the apk file is located. For example, to install an Android app called `MiraculousLifeWebView_0.6.apk` located in `deploy\AndroidApp` folder, navigate to that folder in command line and execute:

```
adb install MiraculousLifeWebView_0.6.apk
```

6.5 Asterisk (PBX) Server setup

Asterisk is a software implementation of a telephone private branch exchange (PBX); it allows attached telephones to make calls to one another. In this project, it is used to make calls to caregivers from the elderly's device in case of emergency situations.

6.5.1 Installation of AsteriskNow

For this project, AsteriskNow distribution is used. Download 32-bit version of AsteriskNow from: <http://www.asterisk.org/downloads/asterisknow>

Create a new virtual machine on VMWare Player using the AsteriskNow image file and follow the installation instructions. Set the network connection as bridged. After the installation is done, login to AsteriskNow using the username and password that was set during the installation.

For installing AsteriskNow on a separate machine, just use the AsteriskNow image file and follow the instructions.

6.5.2 Asterisk configuration files

All the configuration files of asterisk are located in directory: `/etc/asterisk`. After the fresh installation, the default configurations are stored in that folder. Follow the following instructions to sync the configuration files with the latest versions from the AIT SVN repository (<https://dev.arcsmed.at/svn/miraculous-life/trunk/etc.asterisk>).

- 1) Backup the `/etc/asterisk` folder to `/etc/asteriskbackup`

```
cp -r /etc/asterisk /etc/asterisk-backup
```

- 2) Remove all the contents of the folder asterisk:

```
rm -rf /etc/asterisk/*
```

- 3) Check the contents of the directory `/etc/asterisk`

```
cd /etc/asterisk  
dir
```

It should be empty! Otherwise, make sure that the steps above are followed correctly

- 4) Checkout the latest asterisk configuration files:

```
svn checkout https://dev.arcsmed.at/svn/miraculous-life/trunk/etc.asterisk .
```

NOTE: Don't forget the dot at the end of the line, it is to checkout the contents of the `../etc.asterisk` folder, and not the folder itself.

Make sure that the configuration files are indeed downloaded.

5) Start the asterisk with the following command:

```
asterisk -cvvvvv
```

If the asterisk is already running, then restart it using the following commands:

```
asterisk -rx "core restart now"
```

```
asterisk -rvvvvv
```

Don't forget to update the contents of the `etc/asterisk` folder with the SVN repository when there are new changes.

6.5.3 Checking out the sound files

The found files in asterisk are located by default in: `/var/lib/asterisk/sounds/en`. These sound files are used to playback messages to the endpoints.

First move to that folder and then checkout the sound files from the repository into new folder called `mlife`:

```
cd /var/lib/asterisk/sounds/en
mkdir mlife
cd mlife
svn checkout https://dev.arcsmed.at/svn/miraculous-life/trunk/asterisk.soundfiles .
```

NOTE: don't forget the dot at the end of the line

Then start (or restart) asterisk to make the changes effective.

6.6 Asterisk endpoint and client setup

Asterisk "endpoints" are the accounts that are able to call other accounts. At the moment, there is one endpoint for the elderly (x001) and two endpoints for the caregivers (x004, x005) for each testing site (MRPS and Orbis). The endpoint for the elderly does not need to be registered, it is registered automatically when the UI is started. To register endpoints for caregivers:

1. Install Zoiper on an Android tablet or smartphone
2. Open Zoiper and go to "**Config**" tab
3. Click on "**Accounts**" and then on "**Add account**"
4. Select "**Yes**" when asked: "Do you already have an account?"
5. Then select "**Manual configuration**"
6. Select "**SIP**" from the list

7. x001 users are reserved for elderly devices and should not be registered on Zoiper devices. Please contact AIT in case you need to register users at x004 and x005 endpoint.

x001 users must be registered in the UI app. Check the following configurations for correct setup:

- `deploy/UI/js/sip.js`: make sure that correct username is assigned for `asteriskAccountEndpoint`, check the table below.
- `deploy/framework/etc/eu.miraculouslife.emergencycall.service.cfg`: make sure that correct username is assigned for `callerNum`
- Please contact AIT in case you need specific login credentials

6.6.1 Asterisk server related configuration files in the Miraculous-Life project

Make sure to check the following configuration files and adapt them based on your local installation:

1) Asterisk Java Manager Interface settings

`deploy\framework\etc\eu.miraculouslife.services.asteriskinterface.cfg`

`directCall`: true if call is initiated from the device. If false, the call is originated from the asterisk server.

`simulateCall`: true if call simulation is used, false if real call should be made via the asterisk server. However, if the asterisk server is not functioning well during the demonstration, this can be set to true in order to simulate the call.

Note that `simulateCall` will NOT call any caregivers! This function is intended to test and demonstrate the calling functionality on the UI without the need of an Asterisk Server.

2) Emergency call settings

`deploy\framework\etc\eu.miraculouslife.emergencycall.service.cfg`

6.7 Summary of the chapter

The described installation and configuration procedures for the software packages are used for the second prototype and may be adapted in future prototypes. Furthermore, some software packages may be removed or added according to further requirements and developments of the system. These changes will be included in upcoming releases of this deliverable.

7 Miraculous-Life system integration

7.1 Introduction to this chapter

This chapter describes how the different system components are integrated into the whole system. Several components, as described in deliverable 5.1, are developed in Java as OSGi enabled bundles, which can be deployed in a Karaf runtime. A detailed description on how to build the Miraculous-Life system from source using Apache Maven and adding or deploying new bundles to the framework is provided in Chapter 9. For this version, a description is given how the developed bundles are deployed in the Miraculous-Life Karaf runtime, and how they are interacting with other, non-Java based components.

7.2 Apache Karaf OSGi runtime

7.2.1 General

The Apache Karaf runtime, which can be obtained from the official homepage: <http://karaf.apache.org/index/community/download.html> , builds the basis for the HOMER development, since the HOMER environment is deployed in a Karaf runtime. Additionally, other components, like the “*dialogue manager*” or the “*Home daily activity and safety services*” are also deployed in the same Karaf runtime.

For the second prototype, *Apache Karaf version 2.3.8* is used.

7.2.2 Hardware Requirements

The Karaf runtime has the following hardware requirements:

- 5 MB of free disk space for the OSGi core framework (e.g. Apache Felix Karaf)
- (Optional) 3 MB of free disk space for the OSGi core framework source distributions or SVN checkout.

7.2.3 Usage information

The directory layout of an Apache Karaf environment is as follows:

- /bin: control scripts to start, stop, login, ...
- /etc: configuration files
- /data: working directory
 - /cache: OSGi framework bundle cache
 - /generated-bundles: temporary folder used by the deployers
 - /log: log files
- /deploy: hot deploy directory
- /lib: contains the bootstrap libraries
 - /lib/ext: directory for JRE extensions
 - /lib/endorsed: directory for endorsed libraries
- /system: OSGi bundles repository, laid out as a Maven 2 repository

The data folder contains all the working and temporary files for Miraculous-Life.

The Karaf console offers a variety of commands and functions, like interactivity, tab-completion and interactive help.

To see a list of the available commands in the console press the <tab> key at the prompt.

To display the help for a command type either `<command> _help` or `help <command>`

For example, if you type in the console the command `list` and press enter, you will get a list of available bundles and their id and status.

As we now know the bundle id, we can proceed with various other commands which expect the bundle id as a parameter. For example, we can stop and then start different bundles during runtime.

7.2.4 Useful commands

Here is a list with some useful commands which are also very handy when developing or debugging bundles:

- `help` → Display list of all available commands
- `bundle:list` → List all bundles
- `bundle:start` → Start a bundle
- `bundle:stop` → Stop a bundle
- `bundle:refresh` → Refresh a bundle
- `bundle:update` → Update a bundle
- `bundle:install` → Install a bundle from the file system
- `bundle:uninstall` → Uninstall a bundle
- `system:shutdown -f` → Shutdown the whole framework
- `log:display` → Display the log
- `log:tail` → Monitor the log
- `log:get` → Get log level
- `log:set` → Set log level

Using Apache Maven simplifies the process of developing and adding new components / bundles to an existing Karaf framework. A more detailed description is given in Chapter 9.

7.2.5 Installing OSGi bundles

An OSGi bundle can be deployed in two ways:

- Manual Installation: via console commands or by copying the bundle into the “deploy” folder of the Karaf framework.
- Automatic Installation: Can be done during framework start-up as a “feature”. The integration of these bundles into the whole system is then handled by the OSGi framework.

7.3 Noldus Communication Framework

The NCF is used for inter-component communication, especially between non-Java software modules, as can be seen in the *overall system architecture* diagram in deliverable 5.1. The NCF is provided as a Java bundle, which is already integrated in the Miraculous-Life Karaf framework and can therefore be used by any deployed bundle. In order to enable a working communication between two software components, the RabbitMQ Server has to be running and setup according to this document.

7.4 Dialog Manager

This section describes how to install the Dialogue Management software module, focusing on software requirement as well as on the configurations of the component.

7.4.1 Software Requirements

The software and platforms required to run and develop the Dialogue Manager are:

Operating Systems:

- Windows: Windows XP SP2 and above
- Unix-based (any Unix platform that supports Java): Ubuntu Linux, Powerdog Linux, MacOS, AIX, HP-UX, Solaris.

Environment:

- OSGi core framework
- Java Developer Kit (JDK) 1.7.x
- Development suggestion: Eclipse IDE for Java EE Developers with OSGi support.

7.4.2 Installation

The Dialogue Manager is provided as an OSGi bundle and is installed according to section 7.2.5.

Furthermore, the dialogue manager performs interactions with the users based on prescript dialogue scripts. These dialogue scripts need to be located in “/framework/etc/” in the Miraculous-Life system deployment folder, see chapter 8 for more details.

7.5 Home daily activity and safety services

This section describes how to install the *home daily activity and safety services*, which can be seen in deliverable 5.1, Figure 3, focusing on hardware and software requirements as well as on the installation of the component. Up until now, all sub service classes, like the *safety service*, *education & leisure service* or the *guidance service* column are provided in the same bundle. For upcoming system versions, those services will be split it in separate bundles which is more logical and also facilitates the maintenance process.

7.5.1 Hardware Requirements

The hardware requirements for the services are presented in this section. Those requirements are divided into two sets: (1) for hosting and (2) for using these services. Both requirement sets are selected to provide a sufficient performance of the software.

Hardware:

- 5 MB of free disk space for the OSGi core framework (e.g., Apache Felix Karaf)
- (Optional) 3 MB of free disk space for the OSGi core framework source distributions or SVN checkout.

Note for Maven: During the build process, Maven will automatically download all dependencies for each bundle used the system.

7.5.2 Software Requirements

This section describes the software/platforms required to run and develop the *home daily activity and safety services*.

Operating Systems:

- Windows: Windows XP SP2 and above
- Unix-based (any Unix platform that supports Java): Ubuntu Linux, Powerdog Linux, MacOS, AIX, HP-UX, Solaris.

Environment:

- OSGi core framework
- Java Developer Kit (JDK) 1.7.x. Configuration after installation of JDK is described in section 6.2.7.
- Development suggestion: Eclipse IDE for Java EE Developers with OSGi support

7.5.3 Installation

The services are provided as an OSGi enabled Java bundle, namely the ServiceManager (ML_ServiceManager) and is installed according to section 7.2.5.

7.6 KnowledgeBase manager

The KnowledgeBase manager (ML_KnowledgeBaseManager) is an OSGi bundle, which provides general and specific services to interact with the Miraculous-Life database (i.e. the “KnowledgeBase”). These services are available to other bundles through the Karaf framework, which enables for example the services mentioned in chapter 7.5 to insert, select, updates or deletes datasets from the database. The installation of the KnowledgeBase manager is the same as for the services, as described in chapter 7.5.3.

7.7 Speech Recognition Engine setup

With the installation of the speech recognition software components described in chapter 0 and with the RabbitMQ server running, the speech recognition engine can be started by executing the *ASRManager.exe* file, located in the *deployment folder* at “*dialogue_manager\ASR*”. By default, it requires a *dictionary*, which is provided by the *dialogue manager* during runtime, to detect spoken commands.

7.7.1 Hardware Requirements

The hardware requirements for the Speech Recognition software module are based on the hardware requirements of Microsoft Speech Engine and which are:

Recommended Hardware (by Microsoft):

- Dual 2GHz CPU
- 1 GB RAM
- 40+ GB SATA hard drive
- Ethernet network adapter
- DX9-compatible video card, with 3.0 pixel shader and vertex shader support (when available)
- Two USB 2.0 host controllers, which are required when you use DVD emulation and performance analysis tools simultaneously

- Analog, digital (USB) or integrated microphone

7.7.2 Software Requirements

The software and platforms required to run and develop the Speech Recognition Module are:

Operating Systems:

- Windows Vista and above

Environment:

- Microsoft Speech Platform - Software Development Kit (SDK) (Version 11)
- Microsoft Speech Platform - Runtime (Version 11)
- Microsoft Speech Platform - Runtime Languages (Version 11)
- RabbitMQ (Version 3.x)

For better performance of the speech recognition, it is advised to configure the Microphone in Windows Speech settings. This can be found under

Configuration Screen -> Speech Recognition -> Set up Microphone

7.8 Zoobe Server setup

The Zoobe server is not setup independently, as it is part of the dialog management framework. To establish a connection to the Zoobe server, the secure websocket protocol (wss) is used. Please refer to deliverable 3.2a for more information about the integration of Zoobe server inside the dialog management framework or refer to deliverable 3.4a report for more information about the Zoobe client-side interface specification and installation.

7.9 User Interface

7.9.1 General

The user interface is provided by handheld tablets. Figure 6 shows the start screen of the app, which provides the user with an input field for the IP address of the webserver (on the workstation) and the associated port number.

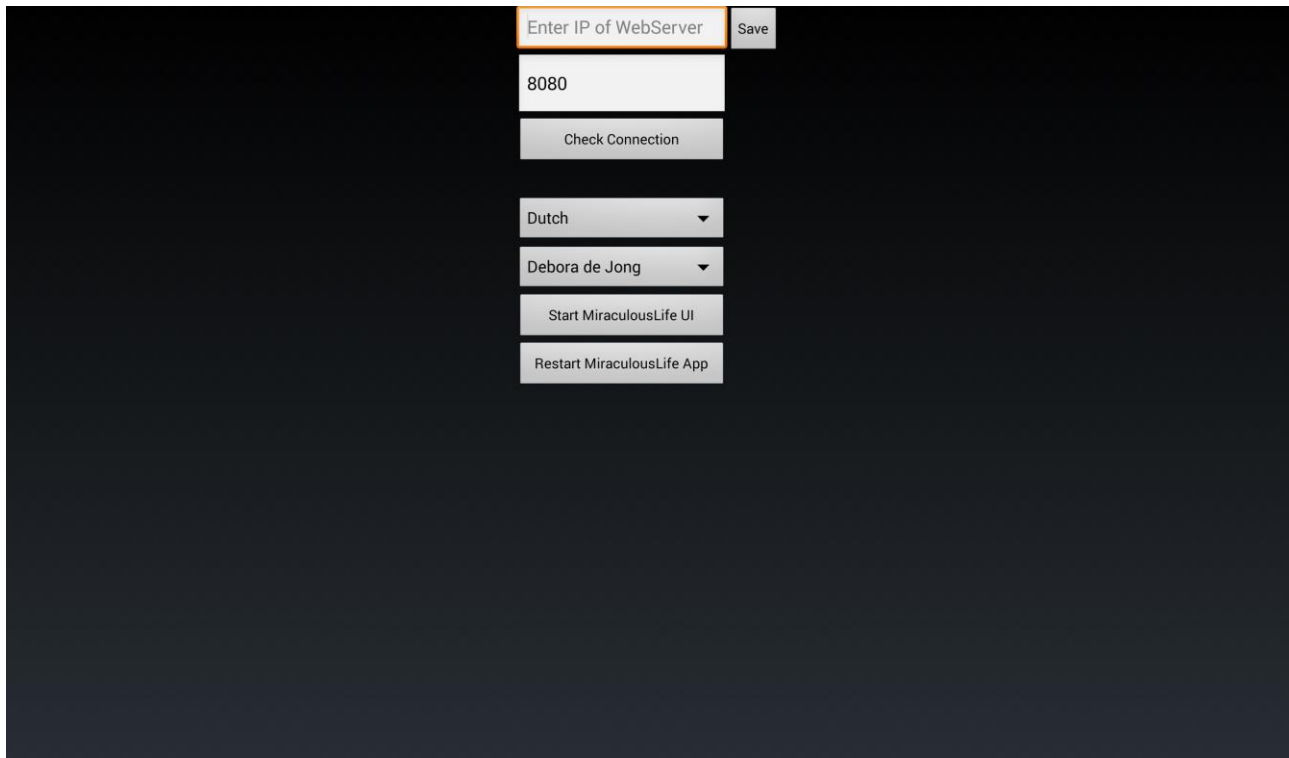


Figure 6: Miraculous-Life Android App start screen

The “*Check Connection*” button can be used, to see if a connection to the webserver can be established with the entered IP and port values. Next, the system language as well a user must be chosen from predefined selection lists. Finally, the Miraculous-Life interface can be started by pressing the “*Start MiraculousLife UI*” button.

7.9.2 Integration description

The user interface is developed as a HTML5 website, which can therefore be accessed by any web browser that supports HTML5 and JavaScript. The app loads the website in a *webview* in order to have better control over the user interaction possibilities. The dynamic user interaction with the interface is realized by using JavaScript to display new content and new Avatar scenes. The communication between the user interface and the Miraculous-Life system logic is established by connecting to a WebSocket server using the websocket protocol. New content derived from the system is submitted via this connection to the user interface and presented to the user through JavaScript.

Figure 7 demonstrates the start or home screen of the user interface, showing the available services on the right hand side and the Avatar on the left hand side. The services can be accessed by saying the displayed word or sentence or by touching the element manually. If the spoken command is recognized or the user provided manual input, the selection is highlighted in a different color.

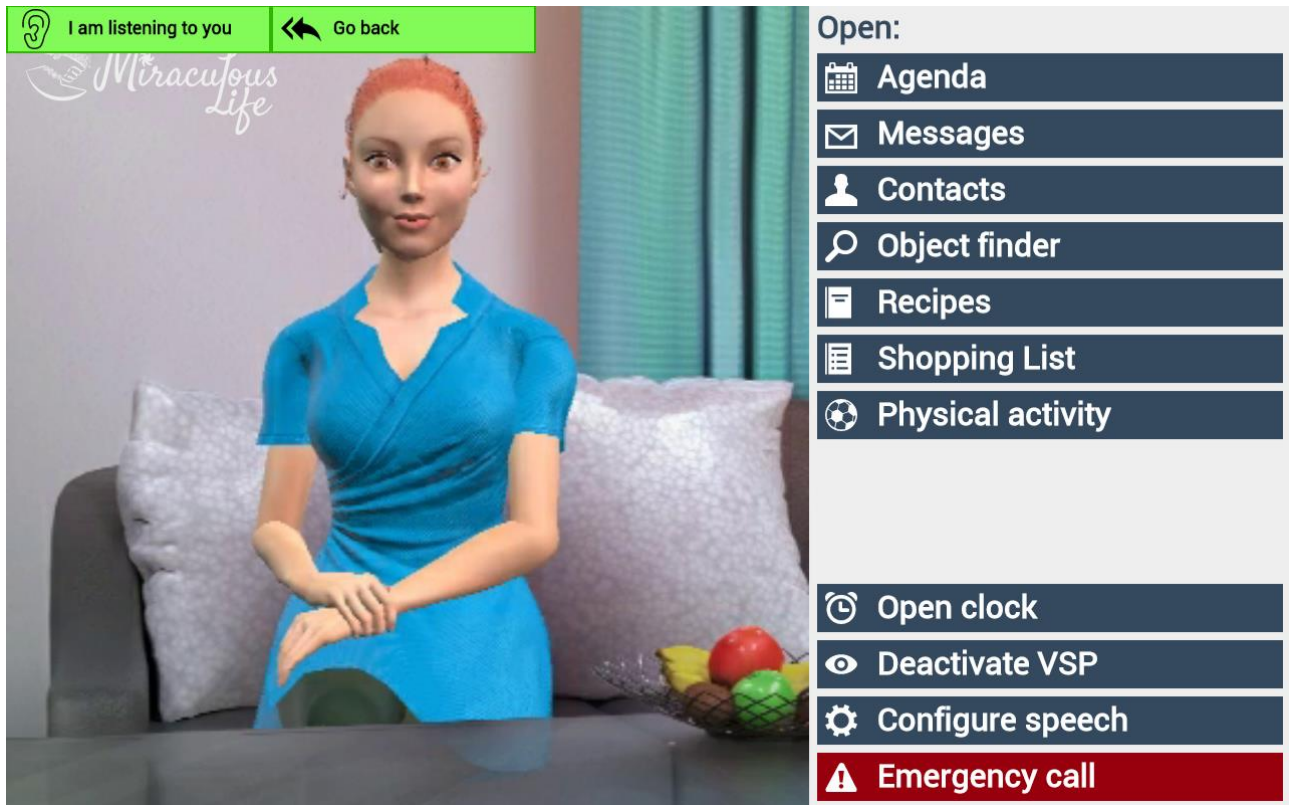


Figure 7: User Interface showing the "home screen" with the Avatar as main interaction element. Available services can be seen on the right hand side.

7.10 Summary of the chapter

Up until now, there are a few components still missing in the second version of this deliverable. When further details about those elements are available, they will be included in this chapter. Although the basic infrastructure, like the Karaf framework and the NCF communication stays the same, some hardware and software requirements might change and need to be adapted if it is necessary due to the development process.

8 Miraculous-Life System deployment

8.1 Introduction to this chapter

This chapter describes how the Miraculous-Life system is deployed on the server and the workstation. The deployment process includes the Karaf framework, the ASRManager, the webserver, the android app and the user interface. The system for the server and for the workstation is provided as an “all in one” zip archive.

8.2 All in one deployment server

After unpacking the all in one zip archive, the following directory layout is apparent:

- `/:`
root folder, containing the start script and one stop script
 - `ML_ServerStartScript.bat`: execute (double click) this file to start the server system.
 - `ML_ServerStopScript.bat`
- `/framework/:`
this folder contains the Karaf framework with the server system bundles already included

The Start script is used to start the Miraculous-Life server system.

In order to exit the Miraculous-Life server system, simply execute the `ML_ServerStopScript.bat` file, located next to the Start script. After that, any remaining open windows related to Miraculous-Life can be closed.

8.3 All in one deployment workstation

After unpacking the all in one zip archive, the following directory layout is apparent:

- `/:`
root folder, containing different Start scripts for different languages and one stop script
 - `ML_StartScript_<lang-code>.bat`: execute (double click) this file to start the system with the corresponding lang-code language.
 - `ML_StartScript_<lang-code>_ASRv2.bat`: provides the usage of a different speech recognition engine.
 - `ML_StartScript_<lang-code>_With_Reminder.bat`: starts the system in order to test it with the activity reminder functionality
 - `<lang-code>` at the moment:
 - EN (English)
 - FR (French)
 - NL (Dutch)
 - Note that only one of the aforementioned start scripts and therefore Miraculous-Life Karaf instances may be running at a time!
 - `ML_StopScript.bat`

- /AndroidApp/:
this folder contains the android app apk file that should be used. Note that the android app is currently provided within the workstation zip archive, but will also be available separately in later stages of the project.
- /dialogue_manager/ASR/:
this folder contains the ASRManager (speech recognition engine)
- /framework/:
this folder contains the Karaf framework with the system bundles already included
- /UI/:
this folder contains the user interface website structure, with one index.html file and a further directory structure for css files, images, videos (avatar) and JavaScript)
 - /UI/css/: contains the css style sheets
 - /UI/images/: contains the images
 - /UI/js/: contains the JavaScript files
 - /UI/videos/: contains the avatar videos. New videos get downloaded and stored automatically in this folder
- /WebServer/Windows/:
this folder contains the Mongoose webserver and its configuration

The Start scripts are used to start the Miraculous-Life system with the associated language, which is up until now Dutch (NL), English (EN) or French (FR).

Note that in order for the ASRManager, i.e. the speech recognition engine, to work properly, a microphone has to be attached to and detected by the workstation.

After confirming that a microphone is available, simply start the ML system by executing one of the aforementioned StartScript bat files. Several different windows will pop up, which can be safely ignored, but they must not be closed! There is no concrete “system ready” notification yet, except for a browser window, which will be shown after about 25 seconds. Since the user interface is provided as a JavaScript controlled website, it is possible to interact with the system via any browser that supports HTML5, CSS and JavaScript.

When the system is ready, start the Miraculous-Life App on the tablet, enter the connection data and start the user interface.

In order to exit the Miraculous-Life system, simply close the App on the tablet and execute the ML_StopScript.bat file, located next to the Start scripts. After that, any remaining open windows related to Miraculous-Life can be closed.

9 Development

This section is intended to give an overview about the steps that have to be taken to setup a proper development environment.

9.1 Environmental setup

9.1.1 Java JDK 7

No additional configuration is needed after installing the Java SDK by following the official guideline and setting up environment variables as described in 6.2.7.

9.1.2 Apache Maven

- Download the latest Apache Maven binaries 3.0.5, 3.1.1 or greater (<http://maven.apache.org/download.cgi>)
- Follow the installation instructions provided at http://maven.apache.org/download.cgi#Installation_Instructions

9.1.3 IDEs

Eclipse IDE is used for the development of the Miraculous-Life system. However, any other IDE that supports Java and Apache Maven can be used for the development.

Eclipse

- Download the latest Eclipse Standard 4.3.x installer <http://www.eclipse.org/downloads/>
- Execute the installer and follow the installation instructions

Maven Integration for Eclipse plugin (M2E)

- M2E is used to support Maven projects in Eclipse IDE.
- Follow the instructions at <http://www.eclipse.org/m2e/>

9.1.4 Version Control

For Windows platform, TortoiseSVN can be used to sync the source code with the repository. For other platforms, install an appropriate SVN-client.

TortoiseSVN

- Download the latest TortoiseSVN installer: <http://www.tortoisesvn.net/downloads.html>
- Execute the installer and follow the installation instructions

Issue Tracking

- The project team uses the Trac issue management system available at <https://dev.arcsmed.at/trac/miraculous-life/> for keeping track of bugs.

9.1.5 Source

- The source code of the Miraculous-Life system is available at <https://dev.arcsmed.at/svn/miraculous-life>

Browsing

- To browse the code via a web interface you can use <https://dev.arcsmed.at/trac/miraculous-life/browser>

Checkout

- To checkout the Subversion repository to your local workstation, get the Miraculous-Life system source code from <https://dev.arcsmed.at/svn/miraculous-life>

9.1.6 Android SDK

Android SDK for Windows platform:

Download the SDK tools installer in <https://developer.android.com/sdk/index.html#Other> and follow the instructions.

Android SDK for other platforms:

Please follow the instructions on <https://developer.android.com/sdk/installing/index.html>.

Eclipse plugin

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT). This plugin provides a powerful, integrated environment in which to develop Android apps.

To add the ADT plugin to Eclipse:

1. Start Eclipse, then select **Help > Install New Software**.
2. Click **Add**, in the top-right corner.
3. In the Add Repository dialog that appears, enter "ADT Plugin" for the *Name* and the following URL for the *Location*:
 - <https://dl-ssl.google.com/android/eclipse/>

Note: The Android Developer Tools update site requires a secure connection. Make sure the update site URL you enter starts with HTTPS.

4. Click **OK**.
5. In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.
6. In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
7. Read and accept the license agreements, then click **Finish**.
8. If you get a security warning saying that the authenticity or validity of the software can't be established, click **OK**.
9. When the installation completes, restart Eclipse and configure the ADT plugin (set location of the Android SDK directory that was recently installed).

9.2 Building the Miraculous-Life project

Apache Maven is used for building the project. For more information about Apache Maven, please refer to the Apache Maven Homepage (<https://maven.apache.org/>). After checking out the source code of the project, navigate to the directory where the sources are extracted and execute the following command:

```
mvn clean install
```

Or run the `build.bat` file that is located in the root directory of the project source code.

Running the build

Once the build is done successfully, run the script: `ML_StartScript_EN.bat` for starting the English version of the Miraculous-Life system. For Dutch or French versions, run `ML_StartScript_NL.bat` Or `ML_StartScript_FR.bat` script files respectively. In order to exit the Miraculous-Life server system, simply execute the `ML_ServerStopScript.bat` file.

9.3 Importing into Eclipse

9.3.1 Importing Miraculous-Life Maven project

Miraculous-Life project can be easily imported into Eclipse using m2e plugin. For installation details for this plugin please see section 9.1.3.

When the m2e plugin is enabled in Eclipse, go to **File > Import...** and select **Maven > Existing Maven Projects** in the opened window and confirm clicking on **Next >**.

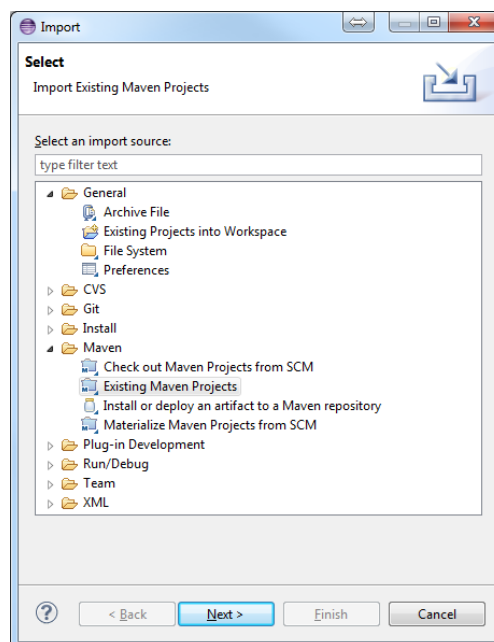


Figure 8: Eclipse Import dialog

In the next import dialog enter the Root Directory of the project (the folder where the pom.xml resides), select the loaded project and click on Finish.

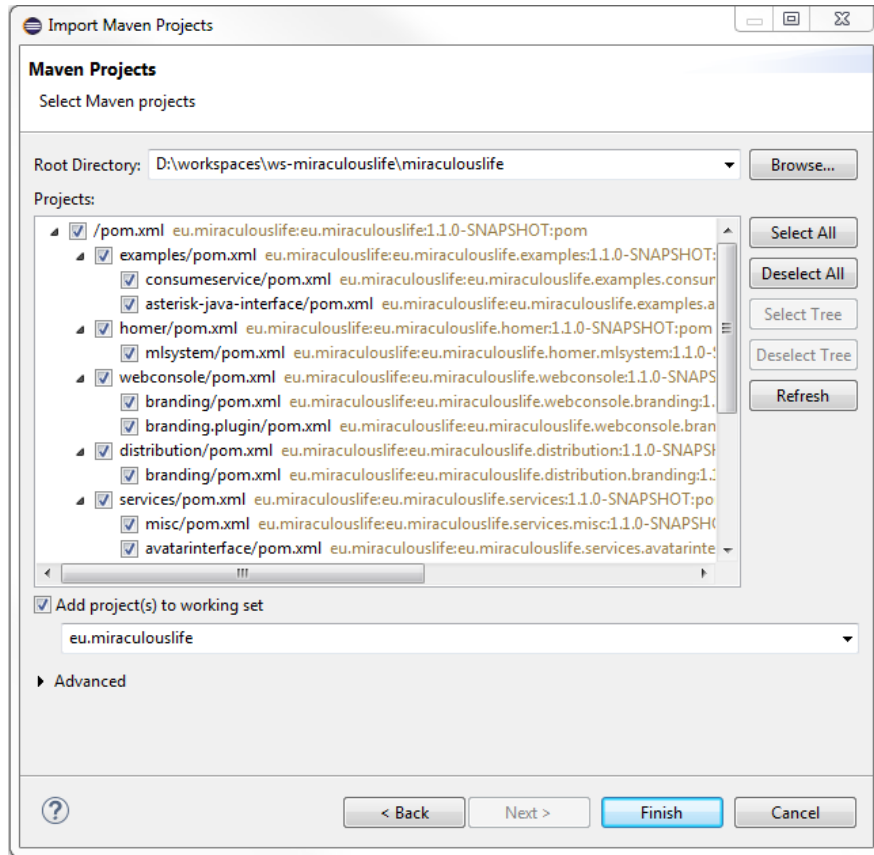


Figure 9: Eclipse Import dialog – selecting projects

After the successful import the example project should show up in the Package Explorer like this:

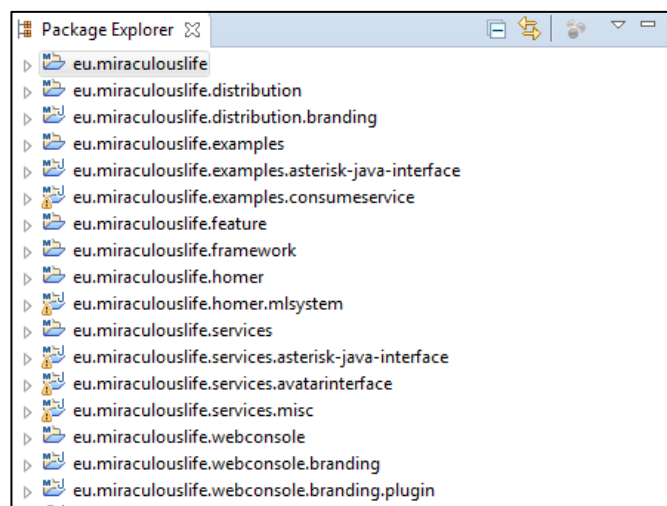


Figure 10: Eclipse package explorer

9.3.2 Importing Android project

After the Eclipse ADT plugin is successfully installed, go to **File > Import...** and select **Android > Existing Android Code Into Workspace** in the opened window and confirm clicking on **Next >**. In the next import dialog enter the directory of the Android projects (RootDirectory \AndroidApp \) the loaded project and click on **Finish**.

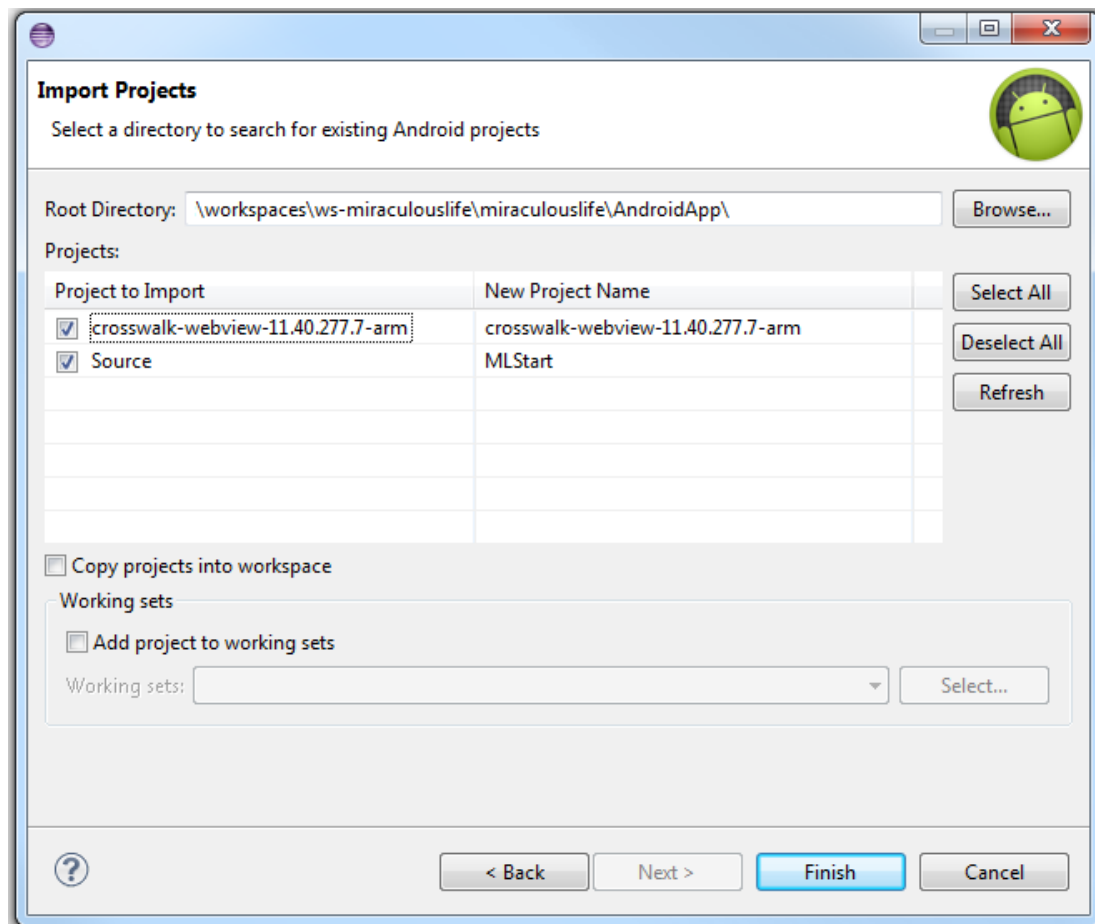


Figure 11: Eclipse import dialog – selecting Miraculous-Life Android project

There are two projects to import:

- Miraculous-Life Android project
- Crosswalk library project

The Crosswalk library project should be automatically added as a library to the Miraculous-Life Android project, otherwise right click on Miraculous-Life Android project, click on **Properties**, go to **Android** and check if the Crosswalk library project is added as a library project.

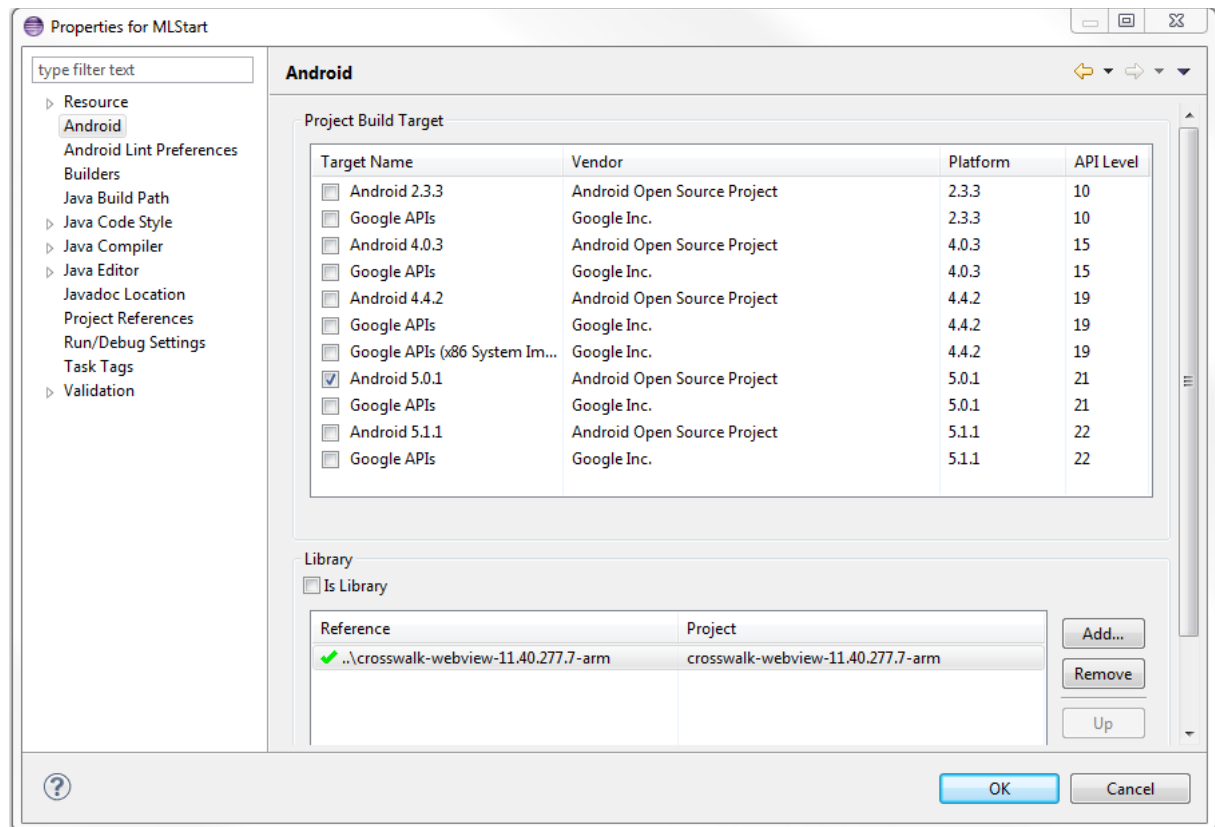


Figure 12: Adding Crosswalk library project to Miraculous-Life Android project

The Android project is ready for development now.

Building and deploying the app

ADT plugin in Eclipse handles the building and deploying of the Miraculous-Life Android project. To run and deploy the project, right click on the project and click on **Run as > Android Application**.