Project Title:
Supported Hearing in Elderly Citizens

Project acronym:
SHiEC

Contract No:
AAL-2013-6-065

Deliverable Reference:
D4.1. Design and architecture of prototype defined.

Date:
30.04.2015

Title:

Design and Architecture

October 2014 – May 2015

Authors: Stefan Lievens, Filiep Vanpoucke

Contributors: Ann De Bolster (Cochlear Technology Centre), Paul Govaerts (Otoconsult), Elke Huysmans (VUmc), Cas Smits (VUmc)

Type of Deliverable: Restricted

SHiEC Platform

Architecture Specification

8043B

Contents

1 INTRODUCTION

The SHiEC project (Supporting Hearing in Elderly Citizens) description contains a final technical work package "*WP4 Web service development, system integration and user field trials"* that specifies in broad terms a prototype of a web portal must be built with the explicit goal to enable user evaluations. This prototype constitutes an eHealth system for senior users of a cochlear implant (CI) that supports them in a user-friendly way with various facilities.

1) Personalized information on their CI. By logging in, the user is known and the device information can be retrieved. The system knows e.g. the device generation and makes that information available. It can also show the personal warranty status.
2) Overview of the usage data (data logging) of the users with additional counselling
3) Speech performance assessment through audio testing (digit triplet test and speech audiometry) and questionnaires
4) Hearing exercises (rehabilitation)

The need for these functionalities also emerged during customer workshops with OPCI, the Dutch user society of CI users.

The functionalities themselves are being developed in the context of work packages WP1 (counselling and data logging), WP2 (formal assessment, auditory training) and WP3 (performance monitoring at home).

The SHiEC platform is essentially a research device to understand how to build an eHealth platform connecting a clinician and a CI end user, what value-adding services to offer through such a platform and how to design appropriate user interfaces. Its intention is to enable early evaluations with relatively small groups of patients, and learn from the experience. Although the technologies used are scalable, at the end of the project, further work will be needed to build a product quality version.

2 PURPOSE

This document describes the system architecture of the SHiEC platform. Reading this document should provide you with a clear picture of which software components will be part of the solution, as well as how they all work together in the various supported use cases.

3 SCOPE

The content of this document is limited to server and software schematics, as well as descriptions of each of the introduced components. Additional attention is given to the interfaces between these components.

As this is an architecture description, no detailed design specification is included.

At the time of writing, two options are kept open as far as clinician and recipient app are concerned: either an actual application is built that would include online communication functionality, or the application is kept on the server side, which means a regular web browser can be used to access the planned functionality. The rest of this document solely talks about clinician app and recipient app as their functionality will be the same, but be aware that the final implementation may still happen on the server side.

In terms of system environments, neither the surgical nor the research environments are covered as they are not applicable to the solution.

4 OVERVIEW

This system architecture document provides an overview of SHiEC platform, its subsystems and their configurations in key environments of use while focusing on interfaces, functions and performance elements. Figure 1 shows key subsystems and interactions in a general context.

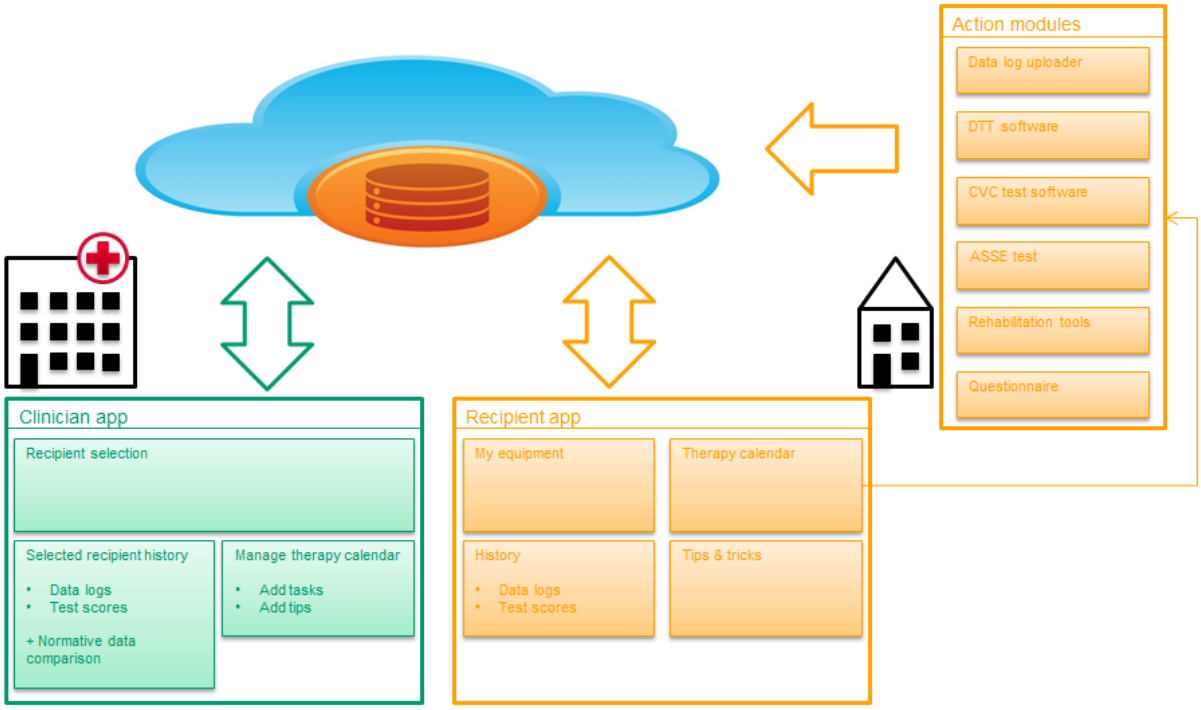

Figure 1: Overview of key system components

As can be seen from the figure, the interactions between the clinician and the recipient will be guided by a therapy calendar. This calendar in essence defines the tasks that the recipient ought to perform from the home environment. Examples may be

- Read a document of watch a video explaining how to change the battery (week prior to device activation)
- Complete a rehabilitation exercise (month 1 after activation)

- Perform a speech test from home (month 4 after activation)

The system supports the following functionalities

1) Data logging complemented with counselling tips

Data logging refers to the capability of a modern sound processor to monitor how long it being used, what the sound environment is and which maps (user settings) the user is selecting. This information about the actual use of the device by the hearing impaired person is of great help to the clinician in guiding the therapy. The figure below shows the current user interface in the professional software (Custom Sound). The SHiEC portal will allow the exchange of this usage information from the home environment.

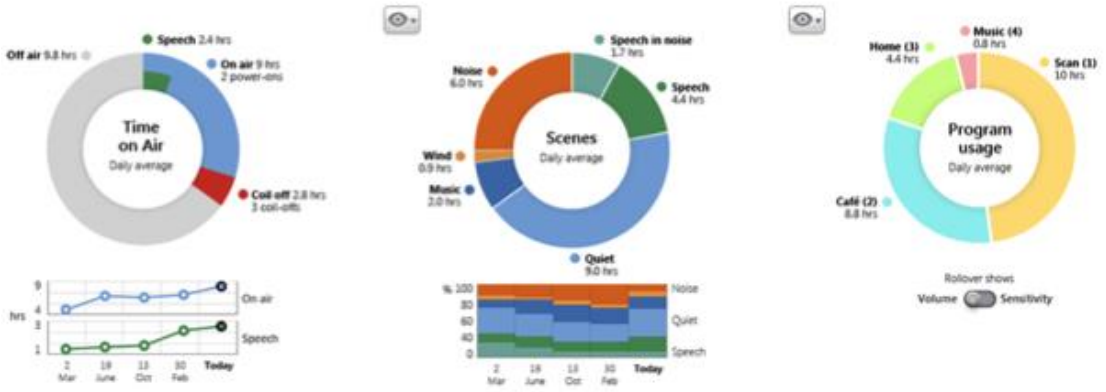

Figure 2: Example of data logging in the Nucleus 6 generation of the cochlear implant

2) Digit Triplet Test

In the Digit Triplet Test [1] (DTT) a random sequence of three digits is played at comfortable level while a stationary white noise of variable level is presented in the background. This test determines the maximum amount of noise a user can tolerate before his hearing performance drops to 50%. This test is ideal for home monitoring of speech performance as it is known to correlate well with general hearing performance of CI users in background noise. The task is intuitive to the participant, the vocabulary is well known and a simple user interface can be shown (keypad). An example of the screen is shown in the figure below.

[1] Smits C, Theo Goverts S, Festen JM. J Acoust Soc Am. 2013 Mar;133(3):1693-706. The digits-in-noise test: assessing auditory speech recognition abilities in noise.

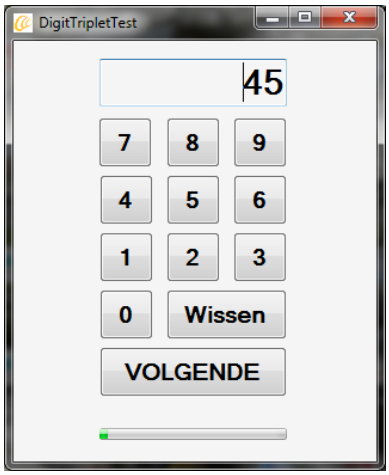

Figure 3: Example of a digit triplet test screen for the end user

3) CVC Test

A CVC (Consonant-Vowel-Consonant) test present lists of monosyllabic words (e.g. pen, bath, cat, …) and measures the amount of phonemes that are correctly recognized. This is the standard test that is offered in the CI clinic to determine how well a user can understand speech in quiet. The test uses calibrated and phonetically well balanced word lists. This material is less suited for home use, but it is supported in the platform as a comparison point to DTT (golden standard).

4) A§E[2] (Auditory Speech Sound Evaluation) test

This is the test suite that one of the project partners, Otoconsult, is using to determine optimality of the CI device fitting. The test suite contains phoneme discrimination, loudness scaling and speech audiometry tasks. An example of the test screens is shown in the figure below.

Figure 4: A§E test screens

[2] A§E test: see http://otoconsult.com/products/asse/

5) Rehabilitation

As the sound of an electrical implant is in many ways very different from the sound a normal hearing person perceives, the brain has to learn how to listen with the new sound. This process can be facilitated with dedicated hearing exercises.

6) Questionnaires

Hearing outcomes are broader than measures how well a person understands speech. It has to do with how the person functions in everyday circumstances, how well he can communicate, how well he can socially connect and what quality of life he reaches. There are standard questionnaires available to measure these outcomes such as the Speech, Spatial and Qualities of Hearing Scale questionnaire[3] (SSQ).

4.1 System Components

This section briefly describes key system components (Figure 5). The list of use cases is provided for each of the components.

In essence, the SHiEC project allows a clinician to follow up on the progress of his recipients. Examples are to monitor the use of the implant through data logging as well as monitor hearing performance through remote test sessions. These tests can be scheduled through an online therapy calendar, which may also be used to pass on counselling tips. The recipient data logs and test results are uploaded to an online data storage component, which holds all information required for these use cases.

Both the clinician app and recipient app are applications that will run on a Windows tablet, the online data storage component is an Apache CouchDB[4] running on the amazon web services (AWS) cloud solution.

At the configure test step, the clinician schedules tasks to the data storage component, which can be fetched by the recipient app.

Execution of a test takes place on a tablet computer, which runs a local application software from which individual tasks can be performed. The results are sent to the data storage component.

Finally, viewing results can be done from both the recipient app as well as the clinician app. The data log information and result information are stored online, and can be visualized appropriately towards clinician and recipient.

[3] SSQ questionnaire: http://www.ihr.mrc.ac.uk/products/display/ssq
[4] For more info on CouchDB, see http://couchdb.apache.org/

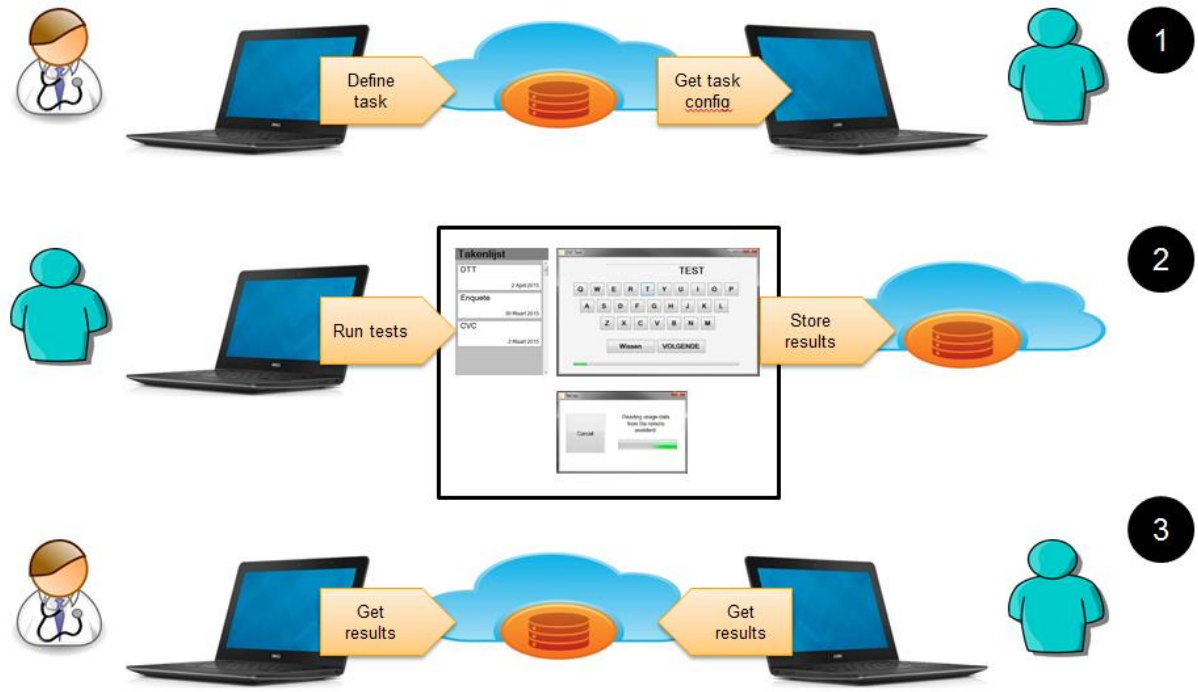

Figure 5: How the apps send around data

4.1.1 Clinician App

The clinician app is a piece of software that provides control of recipients' therapy calendars to the clinician. In addition, the data shared by the recipient can be visualized, enabling the user to follow up individual recipients based on data logs and test scores.

The use cases for the clinician app are:

1. Login/Logout: for security and privacy of the people involved, all interactions between clinician app and online data storage shall be protected.
2. Get recipient list: fetch a list of all recipients connected to the logged in clinician.
3. Creation of a new recipient: set up a recipient therapy calendar based on the recipient's profile (information stored in his CDX[5] file) and a template.
4. Individual recipient selection: choose a recipient to work with.
5. Retrieving data logs from online data storage: fetching a collection of data log entries selected by recipient and time window.
6. Retrieving test scores from online data storage: getting a collection of test scores selected by recipient and time window.
7. Fetching normative data from online data storage: getting relevant normative data to allow visual comparison with individual recipient data, both data logs and test scores.

[5] CDX: Cochlear data exchange format

8. Visualize recipient therapy calendar entries: get an overview of what a recipient has on his calendar.
9. Add an item on the therapy calendar of a selected recipient: this item can be a test task or a counselling tip. The definition of the item on the therapy calendar includes the applicable configuration parameters. Any change on the therapy calendar is sent to the online data storage.
10. Remove an item from the therapy calendar: allows correcting errors and rescheduling of tasks and tips. Changes are sent to the online data storage.

4.1.2 Recipient App

The goal of the recipient app is to get the relevant information from the online data storage component to the recipient in a secure way.

The recipient app has following use cases:

1. Login/Logout: as with the clinician, any operation is limited to a logged in user.
2. Retrieve data logs: based on the logged in recipient, the history of data logs can be fetched for a given time window.
3. Retrieve personal test scores: the logged in recipient can fetch personal test scores over a specified time window.
4. Fetch therapy calendar items: when logged in, the recipient app will automatically populate the items in the therapy calendar.

4.1.3 Online Data Storage

This component stores the information that is to be shared between clinician and recipient and makes it accessible to both parties. It also handles the data security aspects by requiring personal logins for clinicians and recipients.

The online data storage cloud component stores following information:

1. Persons: Both clinicians and recipients.
2. Organizations: Typically clinics, with clinicians and recipients linked to them.
3. Devices: Information about the different devices such as implants, sound processors, accessories, etc.
4. Therapy calendars: This data structure is essentially a list of tasks that the clinician can define for a given recipient with a given time stamp. From the recipient side, items on the list will be actionable.
5. Data logs: When recipients upload their data logs, the records are obviously stored so clinician and recipient apps can visualize them. A service in the cloud will process the data log information periodically in order to create updated normative data.

6. Normative data log info: For each of the information categories found in the data logs, normative data will be specified as collections of average, minimum, maximum, and standard deviation. Normative data records exist with a given segmentation in terms of recipient age and experience categories. Normative data shall be visualized in the clinician app only.
7. Test scores: For every completed test task (DTT, CVC, A§E, Rehabilitation tools, Questionnaires) a record specifying the outcome will be stored in the cloud as well. Both the clinician and recipient apps will download them and visualize them appropriately. In addition, a service in the cloud can periodically process all available test scores and refresh the normative data.
8. Normative test scores: As with the data logs, the test scores are also processed into a set of normative data, including average, minimum, maximum, standard deviation for different segments of the population. Normative data is only used in the clinician app.

4.1.4 Action modules

The action modules are a suite of computer programs that share the feature of being able to upload data (logs or test scores) to the online data storage.

In the scope of SHiEC, the datalog uploader, questionnaire, DTT, CVC test, and A§E test software modules will be considered.

1. Datalog uploader: This program connects to a CR230 remote that holds sound processor data log snapshots in its memory, and uploads them to the online data storage.
2. Questionnaire module: In order to support the use case of validating if a recipient has read certain information, there is a module that sends status information into the online data storage component about the completion of information-type therapy calendar tasks.
3. DTT: The Digit Triplet Test program is an audiometric test in which recipients are asked to enter a sequence of digit triplets being presented to them in a noise environment. The results are uploaded to the online data storage.
4. CVC test: The consonant-vowel-consonant test software presents auditory word lists to a recipient. He has to type in the response (word recognized) using a keyboard. Scoring is done automatically and the results are uploaded to the online data storage.
5. A§E test software: The A§E test software is a third party program that performs various acoustic tests. The results are uploaded to the online data storage.

4.2 System Interfaces

This section briefly describes key system interfaces, see Figure 1. All of the interfaces support the connection between no more than 2 components. All interfaces technically work bidirectional for the logging in functionality; the arrows in Figure 1 merely indicate the flow of data logs, test scores and therapy calendar items.

4.2.1 Clinician App – Online storage

The logical connection between clinician app and online data storage has to support the use cases of the clinician app (see 4.1.1). This is a bidirectional interface, where clinicians get information from the online data storage and send data to the online repository as well.

Clinician app	Interface	Online storage
Log in	→ Login credentials ← Response OK/Failed	Check if username exists and if yes password correct, send response back
Creation of new recipient	→ Logged in clinician, create new recipient request, CDX file	Process the CDX and set up default therapy calendar items
Fetching recipient list	→ Logged in clinician, Request recipient list ← List of recipients	Query database and send back the list
Fetching recipient data logs	→ Logged in clinician, selected recipient, request data logs ← Data log history	Query database for logs and normative data, package info and send back the result
Fetching recipient test scores	→ Logged in clinician, selected recipient, request scores ← Test scores	Query database for scores and normative data, package results and send back results
Create new therapy calendar item	→ Logged in clinician, selected recipient, new task details	Create new task record

4.2.2 Recipient App – Online storage

The interface between recipient app and the online data storage component handles the bidirectional communication around use cases of the recipient app (see 4.1.2).

Recipient app	Interface	Online storage
Log in	→ Login credentials ← Response OK/Failed	Check if username exists and if yes password correct, send response back

Get recipient data logs	→ Logged in recipient, request data logs ← Data log history	Query database for logs, package info and send back the result
Get recipient test scores	→ Logged in recipient, request scores ← Test scores	Query database for scores, package and send back results
Get therapy calendar items	→ Logged in recipient, request therapy calendar ← List of tasks	Query database for tasks, package and send back results

4.2.3 Action modules – Online storage

Client applications mainly send information to the online storage component. It is not the intention to create client applications that fetch information from the online data storage, as information for the recipient is presented through the recipient app.

Module	Interface	Online storage
Data log uploader module	→ Logged in recipient, data log record	Store the incoming data
Questionnaire module	→ Logged in recipient, questionnaire result record	Store the incoming data, mark task as completed
DTT module	→ Logged in recipient, DTT result record	Store the incoming data, mark task as completed
CVC module	→ Logged in recipient, CVC result record	Store the incoming data, mark task as completed
A§E module	→ Logged in recipient, A§E result record	Store the incoming data, mark task as completed

5 DATA MODEL

This chapter specifies the different data records and which information they need to hold. The overall structure looks like this:

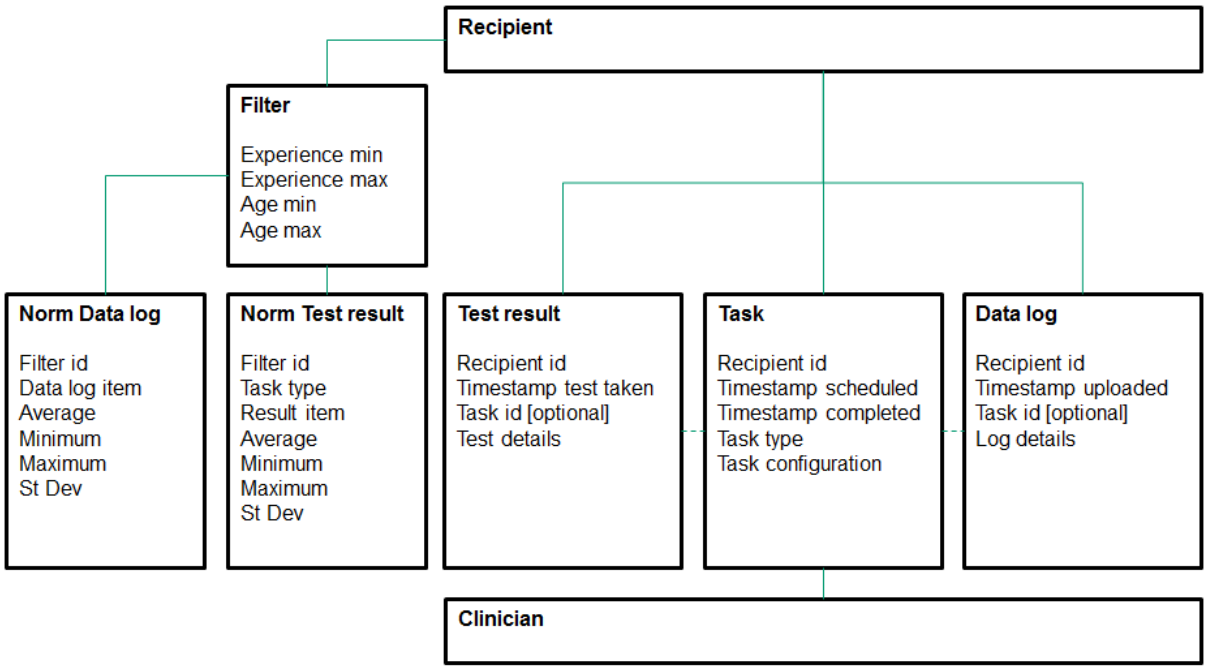

Figure 6: The data model

The blocks on the left (Filter, Norm data log, and Norm test result) are used to keep segmented normative data, supporting use case 7 of the clinician app (showing relevant normative data).

As specified in the use cases of the clinician app (section 4.1.1), it is the clinician who defines which tasks shall appear when the recipient uses his app. All tasks belonging to the same recipient form the therapy calendar. Several different types of tasks are identified:

Figure 7: Task types

Depending on the task type, the task configuration will be different: the request for sending data logs is just that, whereas a quiz (or questionnaire) will need to specify the questions as well as multiple choice answer sets if applicable. The different test modules (DTT, CVC, and A§E) each come with their own configuration parameters, and rehabilitation tools will need yet another set of configuration parameters.

All incoming data is stored in the Data log and Test result tables. The data log details are xml formatted, the contents is specified in “Nucleus 6 feature design – Data logging”, Windchill document number 391622.

6 System Environments of Use and Configurations

The following subsections are structured by system configurations. Every subsystem is described by way of its interfacial, functional and performance elements as applicable.

Within the scope of this project, the surgical and research environments are not applicable, the programming environment has relations to the clinician app, the normal use environment is limited to performing speech tests, and the diagnostics and troubleshooting environment has links throughout more or less the entire system.

6.1 System Configuration in the Programming Environment

Figure 8 shows the applicable system configuration in the programming environment.

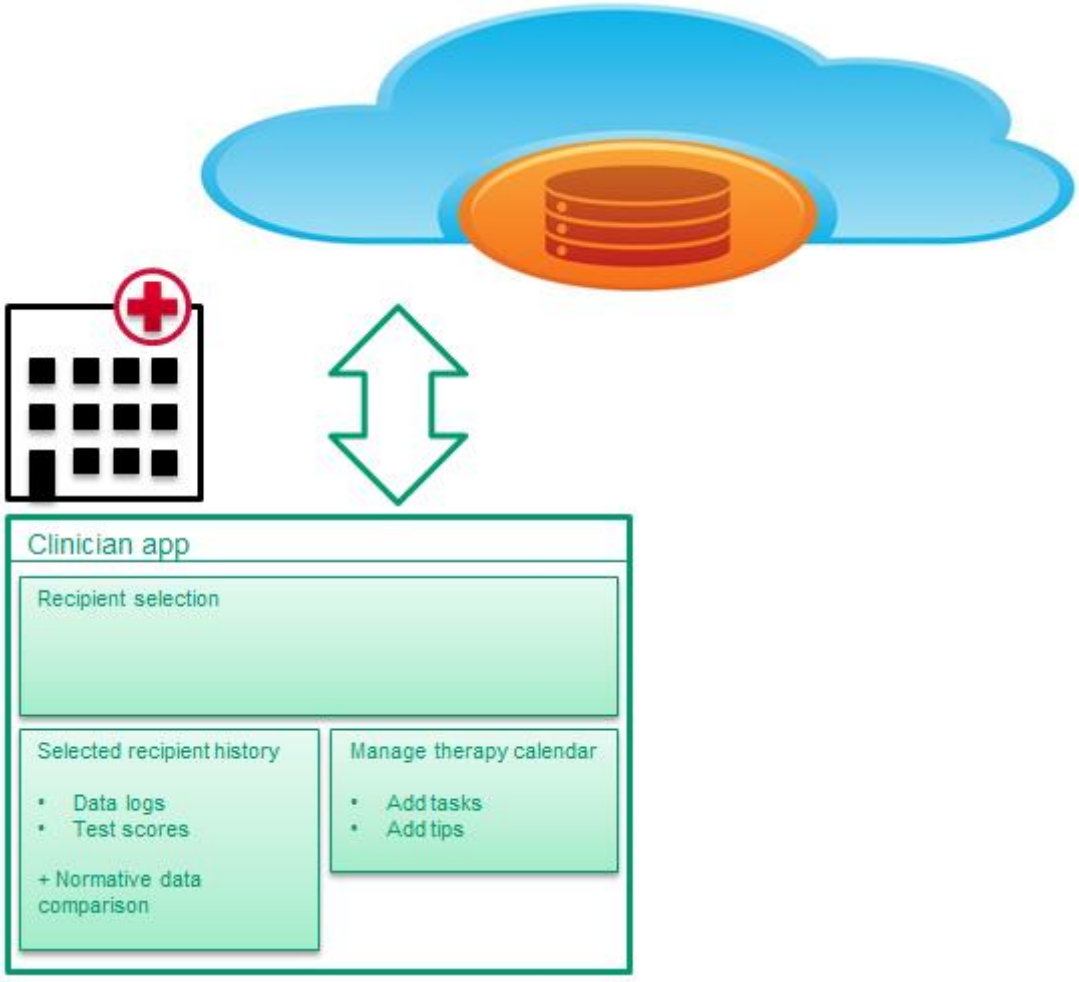

Figure 8: Programming Environment Configuration

The functionality used in the programming environment is limited to the actions taken by the clinician. Results and logs can be visualized and appropriate actions can be managed in the therapy calendar for the recipient.

6.1.1 Clinician app in programming environment setting

The only subsystem used in the programming environment is the clinician app. The detailed description of how the clinician app behaves and which functionalities are available can be found in section 4.1.1.

During programming, a clinician will mainly use the clinician app to investigate the recipient history, i.e. his recent data log and test score information. Depending on his expertise, certain items may be added and/or modified on the recipient's therapy calendar.

6.1.1.1 Interfaces

In terms of interfaces, only the clinician app-online storage link is used (see section 4.2.1).

6.1.1.2 Functions

The functions that are used in a programming environment setting are:

1. Login/logout of a clinician

Input	Function	Output
Username and password of clinician	Check if the credentials are correct	Clinician is allowed access to his recipient list

2. Selection of a recipient

Input	Function	Output
Clinician selecting a recipient	The therapy calendar, data log, and test scores linked to that recipient are collected	The clinician app shows the recipient history

3. Fetching data from the online data storage

Input	Function	Output
The selected recipient identification is sent to the data storage component	All related data objects are queried and packaged for transfer back to the clinician app	The data is received and visualized in the clinician app

4. Modifying therapy calendar items

Input	Function	Output
Clinician selects to add a new item on the calendar	Configuration of the task is determined based on the input of the clinician	A new therapy calendar item is sent to the online data storage, making sure the recipient app will find it the next time the recipient logs into the system

6.1.1.3 Performance

In terms of performance requirements, the clinician app shall be responsive enough to facilitate the clinician fetching data at speeds that don’t cause meaningful delays in the

normal care giving process. It is therefore important to limit the data sent over the communication link to the relevant bits.

6.2 System Configuration in the Normal Use Environment

As the normal use environment is limited to the phase where recipients are stimulated, its system configuration is limited to the conduction of the various speech tests.

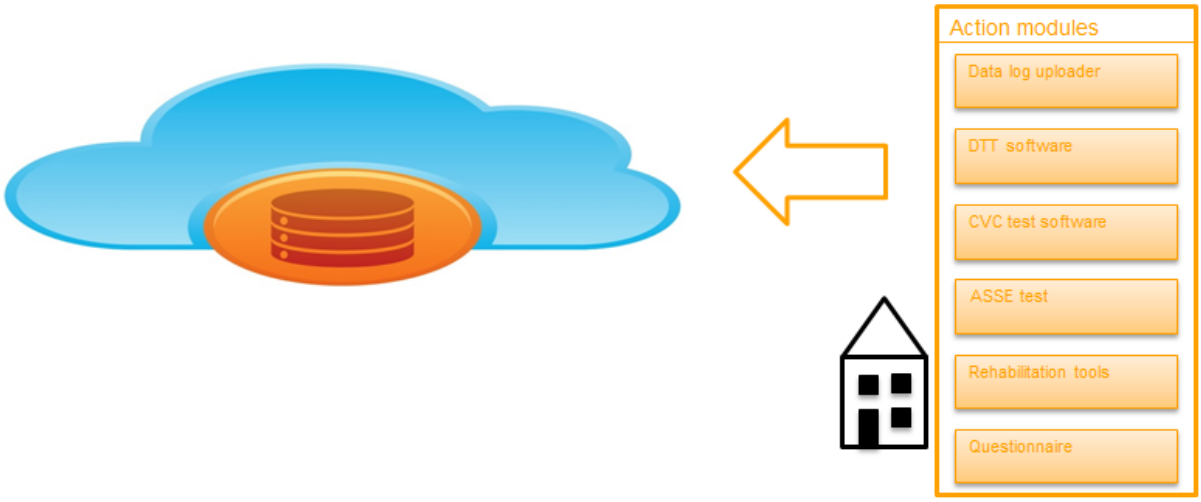

Figure 9: Normal Use Environment Configuration

6.2.1 The test modules in the normal use environment

Figure 9 zooms in on the relevant part of the system overview in the context of normal use. Note however that the data log uploader and questionnaire components are not part of this environment; they are used in the diagnostics and troubleshooting environment.

6.2.1.1 Interfaces

The only interface in the normal use environment is specified in section 4.2.3. It is used to send the results obtained while testing to the online data storage.

6.2.1.2 Functions

The functional specification of each of the components will naturally be different (a digit triplet test has a different workflow than an A§E test e.g.). However, in the context of the SHiEC project, all of the tests share the functionality to send the results of a test into the online data storage component. This happens like this:

Input	Function	Output
The raw data is packaged in the defined xml structure	The module sends the data to the online storage component where it is stored and processed. Processing includes adding task completion timestamps if applicable.	The results are now available to both clinician and recipient from their respective apps

6.2.1.3 Performance

The user experience shall be smooth throughout the execution of the test. When finished (i.e. when the results are to be sent into the online data storage), the results shall be uploaded using the available internet connection while progress is indicated.

6.3 System Configuration in the Diagnostics and Troubleshooting Environment

The diagnostics and troubleshooting environment really is a collaborative effort between most of the system components, with main focus on the exchange of information that may lead to improved understanding of how the recipient is doing.

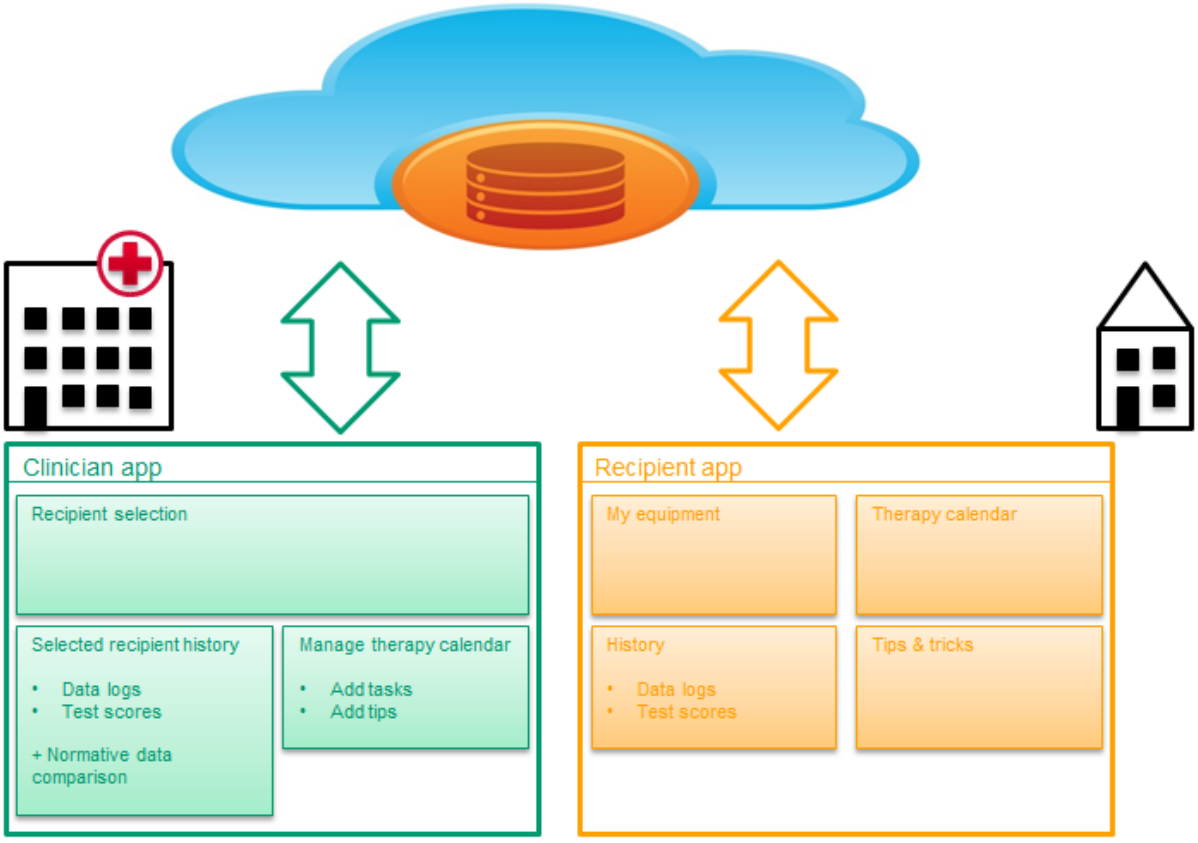

Figure 10: Diagnostics and Troubleshooting Environment

6.3.1 The clinician app in the diagnostics and troubleshooting environment

When talking about diagnostics and troubleshooting, the clinician's tasks are to investigate the available data and possibly modify the therapy calendar of the recipient if deemed necessary.

6.3.1.1 Interfaces

As Figure 10 shows, the interface between online data storage and clinician app is used in this environment. Its usage is specified in section 4.2.1.

6.3.1.2 Functions

The functions used in the clinician app in the diagnostics and troubleshooting environment are essentially the same as the ones used in the programming environment. The functions for logging in, fetching data, and modifying therapy calendar items are specified in section 6.1.1.2.

6.3.1.3 Performance

In terms of performance, the clinician app shall deliver a smooth user interaction. As the online communication speed is an unknown and may vary over time, progress shall be indicated whenever data is transmitted between the different components.

6.3.2 The recipient app in the diagnostics and troubleshooting environment

Troubleshooting from a recipient app perspective is limited to reading counselling tips that have been added as tasks to the therapy calendar.

6.3.2.1 Interfaces

Figure 10 shows the recipient app interfaces to the online data storage component in this environment. The specification of the interface can be found in section 4.2.2.

6.3.2.2 Functions

The use case of a recipient in the context of the diagnostics and troubleshooting environment is all about accessing the relevant information stored in the online data storage component. Following scenarios are applicable:

1. Retrieve own data logs

Input	Function	Output
Recipient logs in Note: own data logs are fetched at login time	App requests data log information from online storage component. The appropriate log details are packaged and sent back.	Recipient sees parsed data log information

2. Retrieve personal test scores

Input	Function	Output
Recipient logs in Note: own test scores are fetched at login time	App requests test score information of the logged in recipient. The chosen scores are sent back to the recipient app.	The scores are visualized to the recipient

3. Investigate therapy calendar items

Input	Function	Output
Recipient logs in Note: therapy calendar items (tasks) are fetched at login time	App requests task list for the recipient who logs in. All uncompleted tasks are packaged and sent back	The recipient gets an overview of actionable tasks in his therapy calendar.

6.3.2.3 Performance

In terms of performance, the recipient app shall deliver a smooth user interaction. As the online communication speed is an unknown and may vary over time, progress shall be indicated whenever data is transmitted between the different components.

7 REFERENCES

7.1 Internal References

Document Title	Number
Cochlear Glossary of Terms	REF 11068
PIP Glossary	E14007AG
Nucleus 6 feature design – Data logging	391622

7.2 External References

Document Title	Number

8 TERMS AND DEFINITIONS

A list of the standard definitions used in PIP projects is contained in the PIP Glossary (E14007AG).

A list of standard definitions used at Cochlear is contained in Cochlear Glossary of Terms (REF11068).

The following table lists definitions that are specific to this document.

Term	Definition
Surgical	System is used during or immediately following surgery, eg, for intra-operative system diagnostics.

Term	Definition
Programming	System is used by clinicians, recipients or carers to program MAPs.
Normal use	System is used by a recipient to listen to their daily sound environment.
Diagnostics and Troubleshooting	System is used by clinicians, recipients or carers to diagnose or troubleshoot.
Research	System is used to conduct research.

9 CHANGE HISTORY

Version	Date	Change
1.0	30 April 2015	Initial version