

DELIVERABLE REPORT

D3.3.3

“Iterative System Integration”

collaborative project

MASELTOV

Mobile Assistance for Social Inclusion and Empowerment of Immigrants with Persuasive Learning Technologies and Social Network Services

Grant Agreement No. 288587 / ICT for Inclusion

project co-funded by the
European Commission

Information Society and Media Directorate-General
Information and Communication Technologies
Seventh Framework Programme (2007-2013)

Due date of deliverable:	30 September ,2014
Actual submission date:	8 October, 2014
Start date of project:	Jan 1, 2012
Duration:	39 months

















Work package	Work Package 3 – SYSTEM SPECIFICATION & INTEGRATION
Task	Task 3.3 – Iterative System Integration
Lead contractor for this deliverable	JR
Editor	Stefan Ladstätter (JR)
Authors	Patrick Luley (JR), Stefan Ladstätter (JR), Anna Weber (JR), Mirjana Artukovic (FLU), Ian Dunwell (COV), Lukas Neumann (CTU), Mark Gaved (OU), Nicoletta Bersia (TI), Charlie Pearson (PP), Sofoklis Efremidis(AIT)
Quality reviewer	Sofoklis Efremidis(AIT), Lucas Paletta (JR)

Project co-funded by the European Commission within the Seventh Framework Programme (2007–2013)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

VERSION HISTORY

version	date	author	reason for modification	status
001	16.09.2014	Stefan Ladstätter	First draft version	Internal
002	18.09.2014	Stefan Ladstätter	Added 3 rd party input	Internal
003	18.09.2014	Stefan Ladstätter	Added corrections	Internal
004	22.09.2014	Stefan Ladstätter	Second iteration	Internal
005	23.09.2014	Stefan Ladstätter	Input from FLU, formatting	Internal
006	25.09.2014	Stefan Ladstätter	Version for review	Internal
007	30.09.2014	Stefan Ladstätter	Final version	Internal

© MASELTOV - for details see MASELTOV Consortium Agreement

MASELTOV partner			organisation name	country code
01	JR		JOANNEUM RESEARCH FORSCHUNGSGESELLSCHAFT MBH	AT
02	CUR		CURE CENTRUM FUR DIE UNTERSUCHUNG UND REALISIERUNG ENDBENUTZER- ORIENTIERTER INTERAKTIVER SYSTEME	AT
03	AIT		RESEARCH AND EDUCATION LABORATORY IN INFORMATION TECHNOLOGIES	EL
04	UOC		FUNDACIO PER A LA UNIVERSITAT OBERTA DE CATALUNYA	ES
05	OU		THE OPEN UNIVERSITY	UK
06	COV		COVENTRY UNIVERSITY	UK
07	CTU		CESKE VYSOKÉ UCENI TECHNICKE V PRAZE	CZ
08	FHJ		FH JOANNEUM GESELLSCHAFT M.B.H.	AT
09	TI		TELECOM ITALIA S.p.A	IT
10	FLU		FLUIDTIME DATA SERVICES GMBH	AT
11	BUS		BUSUU ONLINE S.L	ES
12	FUN		FUNDACION DESARROLLO SOSTENIDO	ES
13	DAN		VEREIN DANAIDA	AT
14	MRC		THE MIGRANTS' RESOURCE CENTRE	UK
15	PP		PEARSON PUBLISHING	UK
16	ATE		AUSTRIAN INSTITUTE OF TECHNOLOGY	AT

CONTENT

Version History	2
1. Executive Summary	6
2. System architecture (JR)	7
3. Startup Procedure (JR)	8
3.1 First start of MApp	8
3.1.1 Registration	8
3.1.2 Anonymous Registration	10
3.1.3 Login	10
3.2 Normal Startup	12
3.3 Startup with User logged out	12
3.4 Start after MApp installation with existing User	12
4. Seamless integration (JR)	13
4.1 Recommendation Interlinkage (AIT)	17
4.1.1 The User Location Recommendation	17
4.1.2 Recommendations to the Language Learning component	19
4.1.3 Recommendations that open web resources in a browser	21
4.1.4 Recommendations related to coins	22
4.2 Inter-linkage between Language Learning and the Forum (PP)	23
4.3 Inter-linkage between the Translation Tool and the Forum (CTU)	24
4.4 Interlinkages from Forum to other existing Social Networks (TI)	26
4.4.1 Facebook interlinkage	26
4.4.1.1 Development details	27
4.4.2 Twitter interlinkage	27
4.4.2.1 Development details	28
4.5 Navigation Service – Situated Navigation Assistance (FLU)	29
4.6 POI Service AND Navigation Service (FLU)	29
4.7 Info and POI Service (FLU)	31
4.8 Integrating Services through a Common Currency (COV)	32
4.8.1 Currency as a Motivator	32
4.8.2 Designing a Currency System for MASELTOV	33
4.8.3 Technical Implementation	35
5. Notifications (JR)	36
6. Data Exchange (AIT)	38

6.1	Usage Events	39
6.2	Progress Events	40
6.3	Interesting Thing Events	41
7.	MApp Workflow control (AIT)	43
7.1	Login/Logout Operations in MApp	43
7.2	Querying Login Status	45
7.3	Broadcasters	45
7.3.1	Broadcasters from User Profile.....	46
7.3.2	Login Broadcaster	46
7.3.3	Logout Broadcaster	46
7.3.4	Settings Changed Broadcaster.....	46
7.3.5	User Preferences changed Broadcaster	46
8.	Final Integration Status (FLU)	46
8.1	Technical solution for software integration	46
8.2	Current status of integration	48
8.3	Integration plan	53
9.	3 rd Party Plugin concept (JR)	54
9.1	Querying Data	54
9.1.1	Is User Logged In?	55
9.1.2	Query all User Data	55
9.2	Sending Events	56
9.3	Sending Content	57
10.	Summary and outlook	57
11.	References	58

1. EXECUTIVE SUMMARY

This deliverable describes the work carried out in the scope of task 3.3 “Iterative System Integration” of work package 3 “SYSTEM SPECIFICATION & INTEGRATION”. The finalized work within task 3.3 was planning and controlling of the integration of software components from different partners within one application towards a seamless user experience.

Especially in the case of the project MASELTOV, where several services were developed by different diversely located project partners, an intensive and efficient management of the overall integration process has played a crucial role within the project to ensure the quality of the final application. Therefore, comprehensive integration planning and controlling was needed continuously to ensure a seamlessly integrated MASELTOV application at the end of the project.

Deliverables of Task 3.3 were used within the consortium as a reference with important technical background information and guidelines for software integration which are relevant for all technical partners who are providing parts of the software. Therefore and with respect to completeness of this reference the successive deliverables always include technical information from the preceding one.

This document extends on its predecessor D3.3.2 with the necessary adjustments made after experiences of the first field trials. The modules covered in this report represent their final development stage:

- System architecture (Chapter 2.)
- Startup procedure (Chapter 3.)
- Seamless integration and interlinkages concept (Chapter 4.)
- Notifications (Chapter 5.)
- Data exchange concept for communicating events and issuing recommendations (Chapter 6.)
- Outline of MApp workflow (Chapter 7.)
- Roadmap and status of final integration (Chapter 8.)
- How to integrate 3rd party applications (Chapter 9.)

2. SYSTEM ARCHITECTURE (JR)

A major result of deliverable D3.1.2 was the system architecture of MASELTOV which is depicted in Figure 1 and gives an overview of all software components of the MASELTOV system. Those components were integrated into one seamless MASELTOV application. Detailed descriptions of all software components can be found in deliverable D3.2.2.

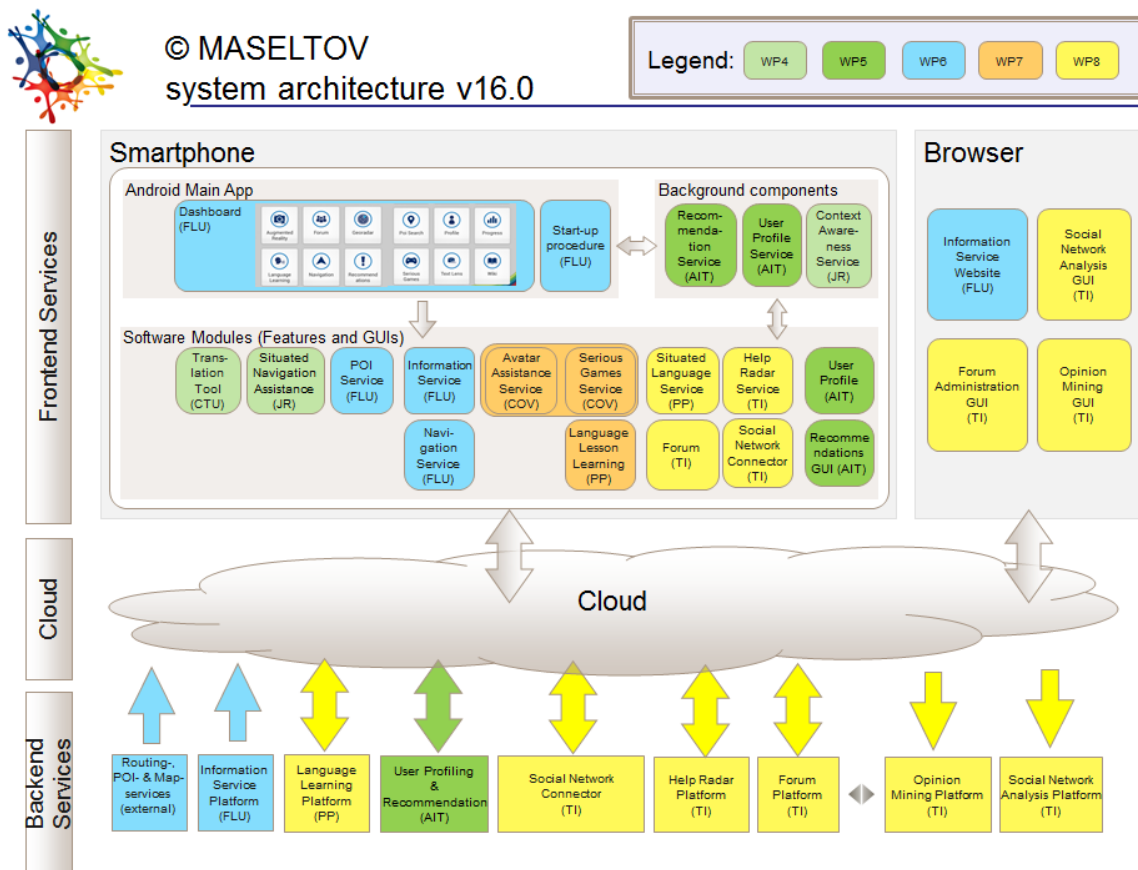


Figure 1. Schematic picture of the MASELTOV mobile application.

The complex nature of the MASELTOV application is demonstrated by the high number of software components provided by multiple partners and the fact that they have to be deployed in different physical scopes like on the server or on the client side. Especially on the client side a huge amount of software modules had to be seamlessly integrated into just one mobile application which has proved to be challenging.

Furthermore each integrated software module had to provide and use interfaces for/of each other module in order to communicate and take advantage of each other in an integrated workflow which showed the added value by combining all the dimensions of MASELTOV services. A detailed description of all services and tools can be found in D6.1.2.

3. STARTUP PROCEDURE (JR)

This section describes the startup procedure of the MASELTOV application in view of four different scenarios: (1) First start without a registered user, (2) Normal startup, (3) Startup with User logged out and (4) Start after fresh install and registered user.

3.1 FIRST START OF MAPP

At the first start of MApp the language selection screen appears which allows the user to select the preferred language for the application (depicted in Figure 3). As soon as the user selects a language, the application switches to the login screen (Figure 4).

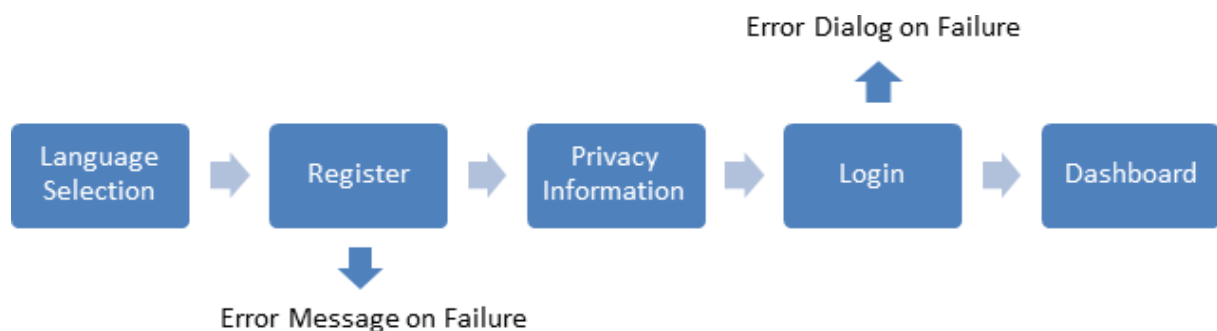


Figure 2: First Startup Procedure

3.1.1 REGISTRATION

For users that do not already have an account, the client application presents them an option to register (Figure 5). Here a valid email address, password, username as well as the city have to be entered. To be able to proceed, the user has to accept the privacy information (Figure 6). If the registration fails, an error message is displayed (e.g. when entering an invalid email address format, too short password or a user name that is not allowed or not available).

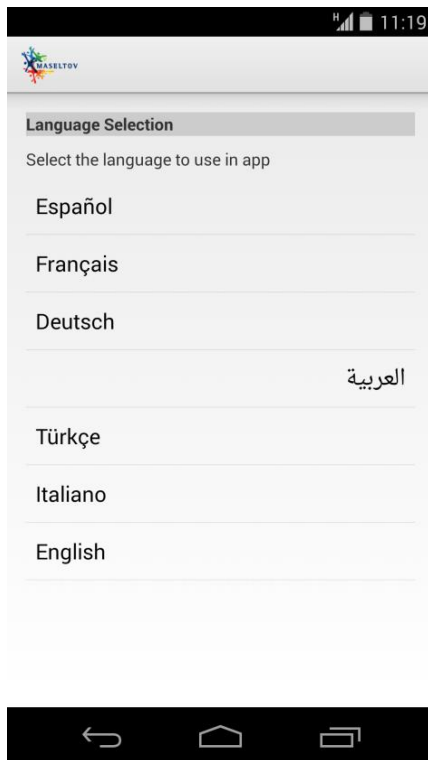


Figure 3: Language Selection

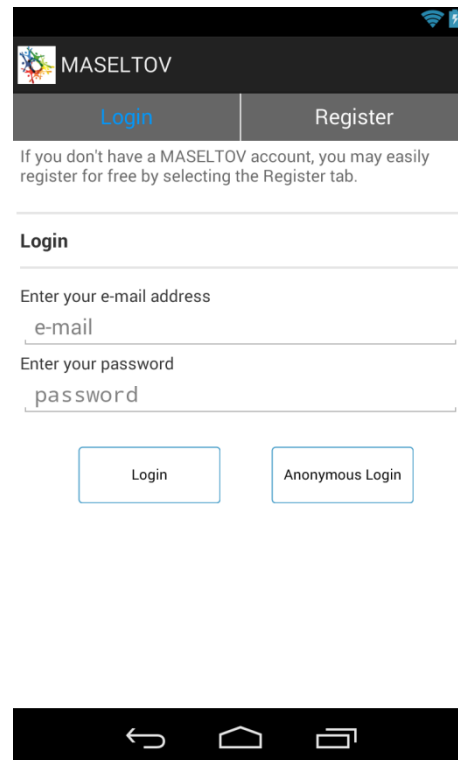


Figure 4: Login Screen

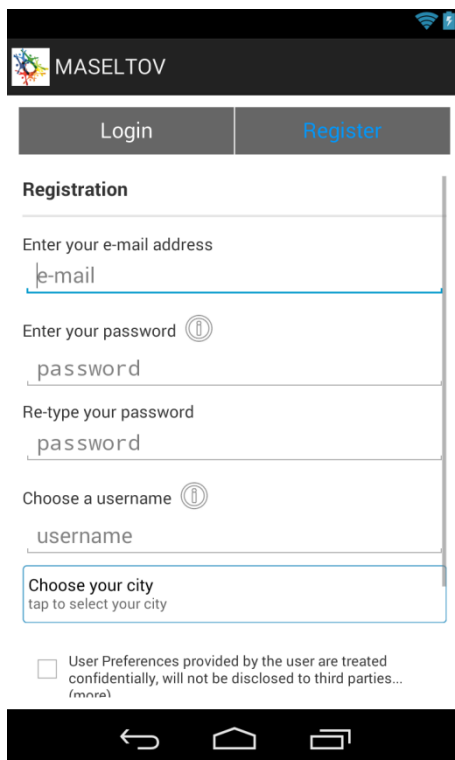


Figure 5: Registration

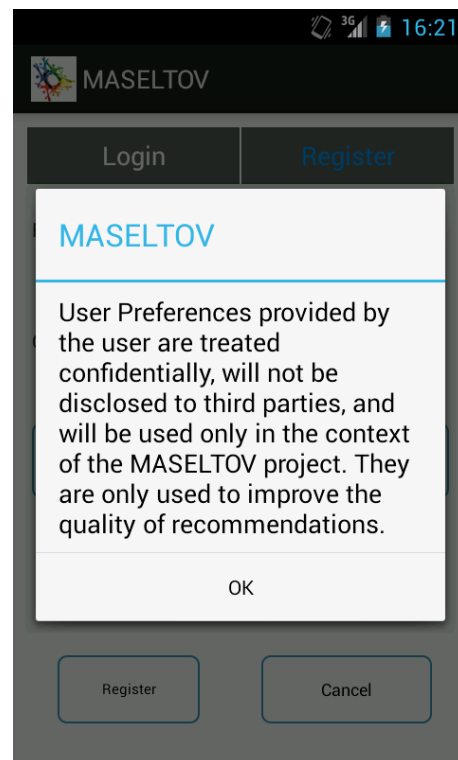


Figure 6: Privacy Information

3.1.2 ANONYMOUS REGISTRATION

Also an anonymous login is supported by the platform allowing users to login without registering an email address (Figure 7). Instead the device id is used by the client software for creating the user identity. There can only be one anonymous account per device.

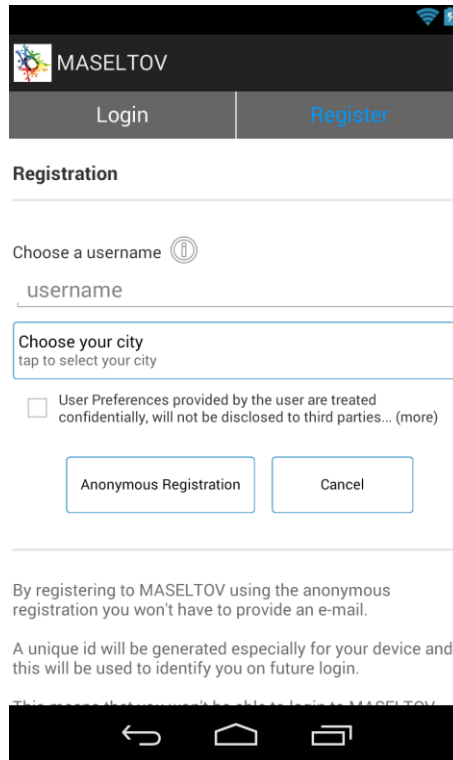


Figure 7: Anonymous Registration

3.1.3 LOGIN

After a successful registration the application returns to the login screen (Figure 8). The user is asked to fill out email and password in order to log in. If the login procedure fails, an error dialog is shown (Figure 9). In case of an anonymous registration the user has to choose the option anonymous login, leaving the login form empty. Afterwards the application switches to its main screen (Figure 10).

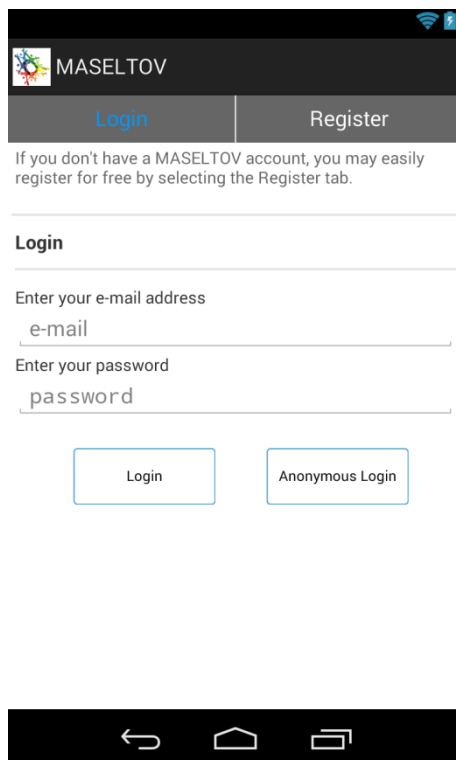


Figure 8: Login/Registration Screen

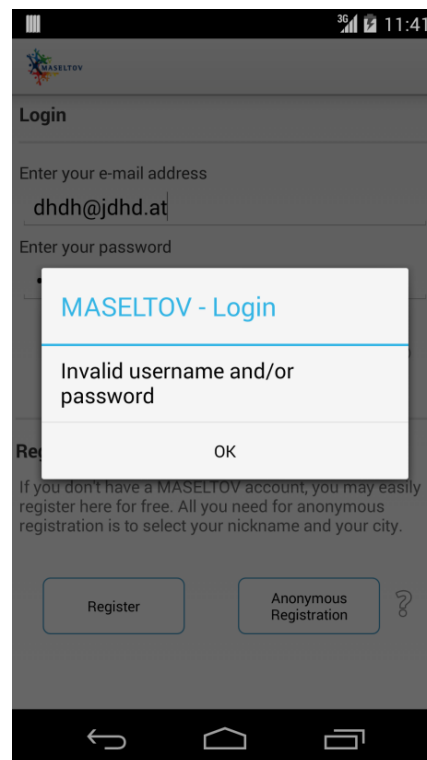


Figure 9: Login Error Dialog

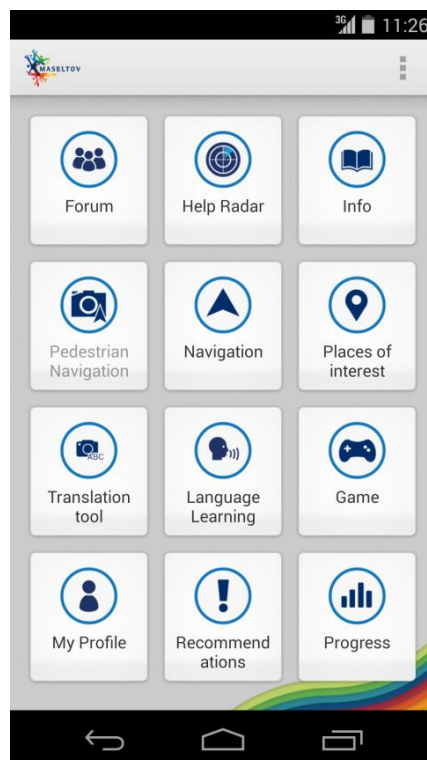


Figure 10: Main Screen

3.2 NORMAL STARTUP

On the next startup, MApp immediately switches to the dashboard activity (Figure 10) assuming the user is logged in.

3.3 STARTUP WITH USER LOGGED OUT

The following image (Figure 11) shows the startup procedure for an already existing but currently logged out user on the device. The language information is already stored on the device and the language selection screen is therefore not shown in the startup process.

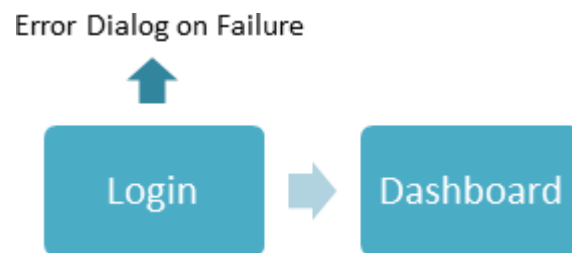


Figure 11: Startup User logged out

3.4 START AFTER MAPP INSTALLATION WITH EXISTING USER

After newly installing MApp the language selection screen appears. The following figure shows the startup with an existing user on a newly installed MApp (Figure 12). This procedure will also be applied if the application data is erased from the device.

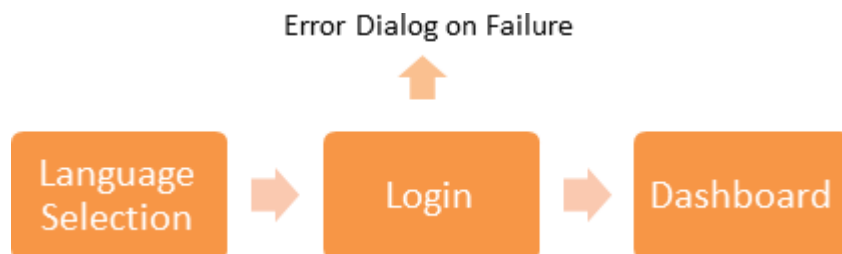


Figure 12: Start after Installation with existing User

4. SEAMLESS INTEGRATION (JR)

Seamless integration of MASELTOV tools and services is accomplished by providing useful linkage between different services to take advantage from each other and to generate a highly valuable over-all service.

This chapter shows how tools and services are taking advantage of each other (inter-linkage) in order to provide a highly valuable integrated MASELTOV service. Beside direct linkage of services or tools the application provides a concept of providing useful linkage through recommendations. This means that the system provides automatically generated recommendations based on the individual progress or discovered preferences of users which are linked to tools, services or contents which might be of high interest to users. In addition to this, the various services incentivize their usage by issuing virtual coins which can be redeemed in the serious game. Details of this integration through a common currency can be found in subsection 4.8.

So users of MApp are offered three different possibilities to consume services or tools.

1. Start services or tools manually on demand using the dashboard
2. Be pointed from one service/tool to another by the system (useful inter-linkage)
3. Tools or services are recommended to the users based on their current situation and context.

The following table shows the current status of planned linkages between services or tools (corresponding to point 2 in the list above), modules which are sending coins and also linkages which are provided by different kinds of recommendations.

Table 1: Seamless integration by smart combination of MApp services.

MASELTOV - SMART CONNECTIONS BETWEEN MApp services		
Linkage of software components to take advantage from each other and to add value.		
Software module	Linkage description and added value to MASELTOV services	Link to which component
Direct linkage between Software Modules		
Navigation service (FLU)	The user is given the possibility to use the AR-navigation tool with calculated routes from the navigation service (FLU).	Situated navigation assistance service (JR)
POI service (FLU)	The address from POI can be used as start/destination point for the navigation service.	Navigation service (FLU)
Information service (FLU)	Mentioned Addresses in articles can be linked to POI service and used in consequence as start/destination point for the navigation service.	POI service (FLU)
Maseltov forum (TI)	User post can be shared with external social network send data to Social network connector	Social network connector (TI)
Language lesson learning (PP)	Direct link of app to the MASELTOV forum	MASELTOV Forum (TI)
Translation Tool (CTU)	Post Text recognitions/translations which could not be understood by users directly to the Maseltov Forum to ask for explanations	MASELTOV Forum (TI)

Coin based integration with the serious game via user profile		
Situated navigation assistance service - AR Navigation (JR)	send 1 coin per minute usage	User Profile (AIT)
Translation Tool (CTU)	send 1 coin per minute usage	User Profile (AIT)
Language Learning (PP)	1 coin for completing an activity 1 coin for posting an ok score on a test 2 coins for posting a good score on a test 3 coins for posting an excellent score on a test 2 bonus coins for completing a lesson	User Profile (AIT)
Maseltov forum (TI)	Sending 5 coins whenever user sends a post / reply	User Profile (AIT)
Help Radar (TI)	Sending 50 coins whenever user rates an assistance	User Profile (AIT)
Route Search (FLU)	1 coin when the user searches for a new route	User Profile (AIT)
POI Search (FLU)	1 coin when the user searches for a new route	User Profile (AIT)
Recommendations - Situational linkage		

Recommendation service (AIT)	Recommendation on Nearby POIs of Interest	POI service (FLU)
Recommendation service (AIT)	Recommendation on relevant Language Learning Lessons	Language lesson learning (PP)
Recommendation service (AIT)	Recommendation on emergency vocab	Language lesson learning (PP)
Recommendation service (AIT)	Recommendation on relevant content in the Info Service	Information Service (FLU)
Recommendation service (AIT)	Recommendation on person to contact for help/advice	Help Radar (TI)
Recommendation service (AIT)	Recommendation to use the Game (i.e. redeem coins) when users earns more than 20 coins in total (we might need to amend the number 20 if we see that this number is too high)	Game (COV)
Recommendation service (AIT)	recommendation of relevant content via URLs in the context of (i) Language Learning, (ii)..., (iii)...	Internet Web Sites

From a technical point of view the inter-linkage between services had to be implemented by interfaces and by communicating data and commands over those interfaces. The following figure (Figure 13) shows how this was established taking into account the different physical scopes in which software components of MASELTOV were deployed in.

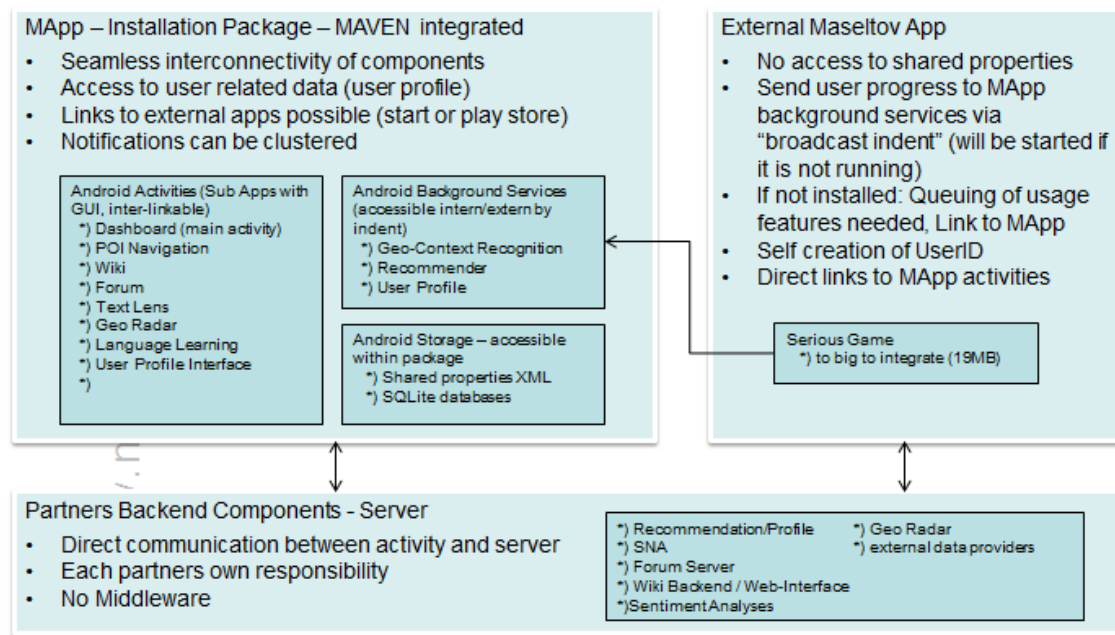


Figure 13: Technical inter-linkage concept diagram.

In the following subsections, the interlinkages between the MApp components are described.

4.1 RECOMMENDATION INTERLINKAGE (AIT)

The User Profile makes available a Content Provider to other MApp components, which allows retrieval of the user's data and preferences and, also retrieval and logging of events. Events are produced by other MApp components and are forwarded as JSON objects to the backend server through the User Profile Content Provider (UPCP). When events reach the backend server they are stored in the database and, in addition, a Java application is triggered to add them into the rules management system (DROOLS), which is the engine on which the recommender is based. Every event as inserted into the rule management system may trigger one or more of the rules that have been implemented. The following paragraphs contain a list of all recommendations that have been implemented, the preconditions under which they are triggered by events from the various MApp components, the recommendation they produce, and how they can direct the user to another MApp component or application on the client side.

4.1.1 THE USER LOCATION RECOMMENDATION

While a user is logged into the MApp, a service runs that monitors and records the user's position (geolocation). The service records the position every five minutes and only if the user has moved at least 500 meters from the last recorded position. The numbers are configurable constants of the geolocation service but it was found that their specific values provide for a

good balance between capturing the user context, maintaining low network traffic, and minimizing battery consumption. Every time the service decides that a new location must be reported, a new event is created and is sent to the backend server through UCP.

After the event that carries the user's location is logged, the recommender queries a service (provided by JR) to retrieve all POIs around that specific location. In the sequel it analyses these POIs and tries to match them with the *hobbies* entries that are specified by the user in Preferences tab of the User Profile.

In case the user has no selected hobbies in his/her profile the recommendation shown in Table 2 will be produced for the user. The user may then tap on the recommendation (in the Recommendation MApp component) and the User Profile will open in order to select hobbies. The rule for this case is shown in Table 3.

Table 2: Recommendation for user who have no Hobbies selected in their profile.

"Select your hobbies and interests from User Profile / User Preferences and we can recommend you related Points of Interest"

Table 3: DROOLS rule for user who has no hobbies.

```
rule "User Preferences Hobbies Is Empty"
  when
    $e: Event(
      $source : source == "User.Location",
      $r: getUser().userHasHobbies( )
    )
  then
    $e.createUserHobbiesRecommendations( $r );
  end
```

In case the user has already selected some hobbies, the recommender tries to match (keyword based) the points of interest around the user (in a 500 meter radius). As soon as the recommender matches a POI that might be of user's interest, a recommendation as shown in Table 4 will be created. The corresponding rule is shown in

Table 5.

Table 4: Recommendation for user who is detected near a POI matching his/her hobbies.

"The British Museum is near you at Great Russell Street, London, United Kingdom"

Table 5: DROOLS rule for the User Location Recommendation

```
rule "User Preferences and POIs"
  when
    $e: Event(
      $source : source == "User.Location",
      $r:
        getUser().checkUserPreferencesWithPOIs( getPoi() )
    )
  then
    $e.createUserLocationRecommendations( $r );
  end
```

The user may then tap on the produced recommendation and will be transferred to MApp Navigation component.

4.1.2 RECOMMENDATIONS TO THE LANGUAGE LEARNING COMPONENT

A number of events trigger the recommender system to produce a recommendation that directs the user to a specific lesson in the Language Learning component. The Translation Tool component typically produces events that contain the detected text. If the recommender matches any of the words detected with a keyword that is related with one of the available lessons then a recommendation like the one shown in Table 6 will be produced. The rule for this case is shown in Table 7.

Table 6: Recommendation for a user who read a sign about measles using the Translation Tool component

"You recently read a sign about measles. How about taking a course in Health"

Table 7: DROOLS rule for Translation Tool recommendations.

```
rule "TranslationTool Recommendation"
  when
    $e: Event(
      $source : source == "TranslationTool",
      $keyword : getInfoValue("keyword")
    )
  then
    $e.createUserTranslationToolRecommendations(
      $e.getTranslationTool() );
  end
```

The MASELTOV Context service is able to detect when users have entered a POI. In such case an event is sent to the UCP and the recommender tries to match the POI's metadata with the user's hobbies. If a match is found then a recommendation as shown in Table 8 is produced for the user. The corresponding rule is shown in Table 9.

Table 8: Recommendation for a user who was detected to stay within a hospital for more than 5 minutes.

"You visited XYZ Hospital. Would you like to take lesson about Health Care? "

Table 9: DROOLS rule for a user entering a POI recommendation.

```
rule "User Enters a POI Recommendation"
    when
        $e: Event(
            $source : source ==
            "MaseltovContext.PlaceEntry",
            $longitude : getInfoValue("longitude"),
            $latitude : getInfoValue("latitude")
        )
    then
        $e.createUserEnteredAPOIRecommendations( $longitude,
        $latitude );
    end
```

As users browse and seek information with the Info component, events are produced carrying keywords from the title of the read text. The event is used by the recommender to match the carried keywords with a set of predefined ones leading the user to an appropriate Language Learning lesson in case a match is found. Hence, in case the user reads a text about First Aid in the Info component a recommendation as shown in Table 10 will be produced. The rule for this case is shown in Table 11.

Table 10: Recommendation for a user who read a text about First Aid in Info component.

"Looking for first aid? Try a Learning Course about Health Care "

Table 11: DROOLS rule for Info reading recommendation.

```
rule "Info Article Title Recommendation"
    when
        $e: Event(
            $source : source ==
            "fluinfo.ArticleTitle",
            $keyword : getInfoValue("articleTitle")
        )
    then
        // Rule body logic
    end
```

```

        then
            $e.createUserWikiSearchRecommendations (
                $e.getWikiSearch() );
        end
    
```

Each of the produced recommendations when selected by the user will open the Language Learning component on a preselected lesson (according to what the recommendation is about). This is done by matching the keyword carried by the event with a set of keywords related to each available lesson. A list of lessons and their related keywords is provided by Pearson Publishing, which implements the Language Learning component. The recommendation produced holds the lesson id that will then be passed to Language Learning component for directing the user to the specific lesson screen. Table 12 shows the URI the recommendation component uses in order to open a specific Language Lesson screen.

Table 12: URI for starting a specific language lesson activity in Language Learning.

```

content://uk.co.pearsonpublishing.maseltov.recommend/<lesson_id>
where <lesson_id> the id of a lesson to call e.g.,
content://uk.co.pearsonpublishing.maseltov.recommend/mas_health
    
```

4.1.3 RECOMMENDATIONS THAT OPEN WEB RESOURCES IN A BROWSER

A number of events may trigger a different set of recommendations that contain a web link that can be opened within a browser and will provide the user the ability to navigate on related web pages online.

When a user achieves a certain level of language skills, an event is produced containing the information about the level s/he reached. A number of learning resources can be recommended to the user. Different online resources may be appropriate for people who are in a certain language skills level. Table 13 shows a recommendation that can be produced for a user who achieved level 2 in English reading. The rule for this case is shown in Table 14.

Table 13: Recommendation for a user who achieved a new language skill level.

```

"Well done for your improvement in English reading
(you are now in level 2)! You can now read
http://www.newsnow.co.uk/h/Current+Affairs/Immigration"
    
```

Table 14: DROOLS rule for a new achieved language skills level.

```

rule "Learning Level Recommendation"
when
    $e: Event( $userLanguage:getUser().getLanguage(),
               $source : source ==
               "User.LanguageLevel",
               $ll:
               getUser().setLearningLevel(getInfoValue("language"),
               getInfoValue("course"),
    
```

```

        getInfoValue("level") )
    )
    then
        $e.getUser().setLanguageLearningLevel($l1.getLanguage(),
                                                $l1.getCourse(),
                                                $l1.getLevel());

        $e.createUserLearningLevelRecommendations(
            $l1.getLanguageText(
                $l1.getLanguage(),
                $userLanguage ),

            $l1.getCourseText( $l1.getCourse(), $userLanguage ),

            $l1.getLevel(),

            $l1.getLinkAndText($userLanguage)
        );
    end

```

The Info component produces events as users seek information and browse text. The generated event contains the text's title. The recommender tries to match words from the title with a set of predefined keywords that may point the user to additional online resources. For example, when the user reads a text about dentists in the Info component, a recommendation as shown in Table 15 is produced. The rule that produces this recommendation is shown in Table 11.

Table 15: Recommendation to a web page for a user who read an article in Info component.

"Looking for dentist? Have a look at <http://www.nhs.uk/Service-Search/Dentist/LocationSearch/3>"

All recommendations presented above are active, in the sense when tapped by the user the default browser application opens the referenced web page. The Translation Tool component produces an event that contains the detected text. If the recommender matches any of the detected words with a keyword that is related with one of the available web resources, then a recommendation like the one shown in Table 6 is produced. The rule for this case is shown in Table 7.

Table 16: Recommendation with a web resource, for a user who read a sign about measles using the Translation Tool component.

"You recently read a sign about measles. You may read further details at www.who.int/topics/measles/en"

4.1.4 RECOMMENDATIONS RELATED TO COINS

When users use MApp components, they may gain virtual coins. Every operation may reward them with a different number of coins. Every time one or more coins are gained an event is

produced and the recommender produces a recommendation. When the user has more than 20 coins, like the one shown in Table 17, the system urges the user to tap the recommendation and visit the Game in order to find a way to ‘spend’ them. The rule for this case is shown in Table 18.

Table 17: Recommendation for a user who gained more than 20 coins.

"You earned 2 coins. You have 23 coins in total. Tap here to see how to use them!!"

Table 18: DROOLS rule for redeem coins recommendation.

```
rule "Add Coins Recommendation"
  when
    $e: Event(
      $source : source == "addCoins",
      $keyword : getInfoValue("coins"),
      $coins : getUserCoins()>20
    )
  then
    $e.createUserCoinsRecommendations( $keyword, $coins );
  end
```

The application when the user taps this recommendation will open the Serious Game.

4.2 INTER-LINKAGE BETWEEN LANGUAGE LEARNING AND THE FORUM (PP)

The Language Learning content contains links to the forum as HTML anchor elements of the form:

```
<a href="masforum:/// {Language name}">
```

When the user taps the link, the Client's `shouldOverrideUrlLoading` method is called. This detects URLs of the correct form for the forum, and launches the forum activity in the standard Android way, using an Intent.

The Intent object has action `com.tilab.SOCIALAPP` and carries string extras:

- `extApp` (Language Learning)
- `language` (the language name from the desires URL (forum topic), e.g. English)

Sample code:

```
Intent intent = new Intent("com.tilab.SOCIALAPP");
intent.putExtra("extApp", "Language Learning");
intent.putExtra("language", {Language name});
view.getContext().startActivity(intent);
```

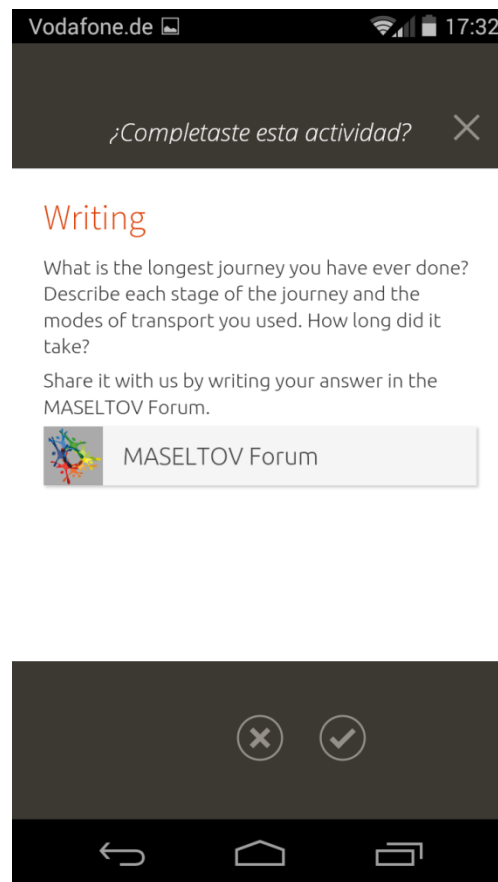


Figure 14: Language learning activity showing a link to the forum

4.3 INTER-LINKAGE BETWEEN THE TRANSLATION TOOL AND THE FORUM (CTU)

In cases when the Machine Translation fails, the Translation Tool application provides a simple way to share detected text and image with other android applications.

The inter-linkage is realized in the standard android way: applications are using Intent APIs and the ActionProvider object.

On successful text detection, the new Intent object is registered (with intent action `android.content.Intent.ACTION_SEND`). The Intent object carries:

- the detected text (field `android.content.Intent.EXTRA_TEXT`)
- the image (`android.content.Intent.EXTRA_STREAM`).

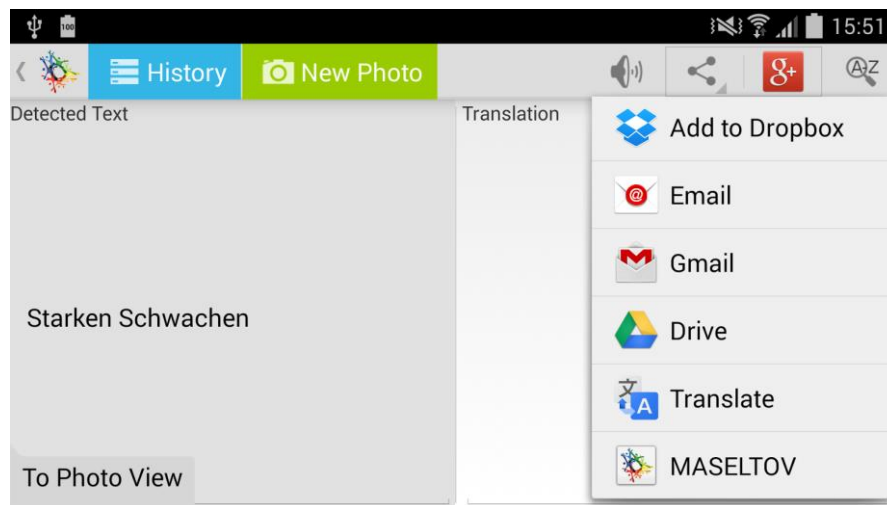


Table 19: Translation Tool, Share Detection

On the Translation Tool GUI, the standard “Share Icon” is displayed in the Action Bar area.

It is possible to share the text with all applications installed on the device supporting the “ACTION_SEND” capability, e.g. MASELTOV Forum application, E-mail client or favourite social network app.

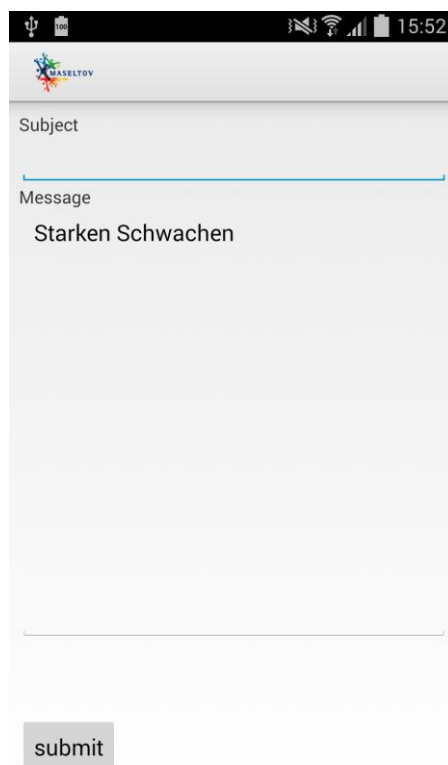


Figure 15: Retrieving intent with detected text

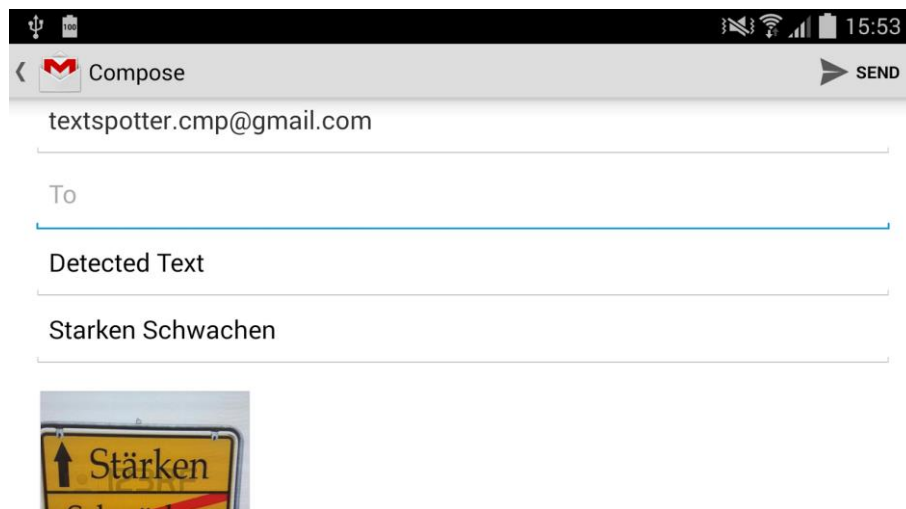


Figure 16: E-mail client app.: Retrieving intent with detected text

4.4 INTERLINKAGES FROM FORUM TO OTHER EXISTING SOCIAL NETWORKS (TI)

As immigrants are highly connected with existing standard Social Networks (SNs) (Facebook, Twitter, etc.), the MASELTOV Forum is interlinked with some SNs in order to motivate the communication between an existing SN (and its large community) and the MASELTOV local community.

MASELTOV users that find useful information during their experience in the Forum, can share the content in some social network. The interlinkages from the MASELTOV Forum to Facebook and Twitter have been implemented.

The user is allowed to share on Facebook or Twitter using the URL of the Forum post and not the full text of the post. This is a mandatory choice because Twitter allows the user to share only 140 characters per tweet and posts are usually longer. The shared post can be read at the specific URL by means of a Forum web browser (the access is read-only).

4.4.1 FACEBOOK INTERLINKAGE

The first time that the user tries to share content on Facebook a pop-up is launched. In this window Facebook asks the user for Facebook credentials in order to authorize the MApp to share content on the user's Facebook timeline. The user's Facebook credentials are read only by Facebook login interface and are not stored in the MASELTOV platform.

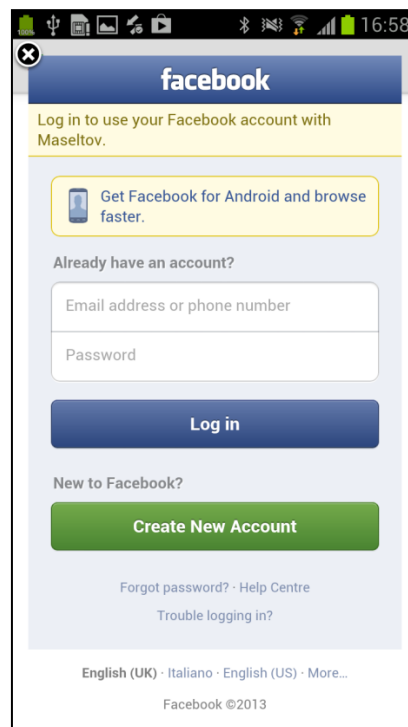


Figure 17: Facebook Interlinkage

No further authorization will be required as long as the user doesn't remove the MApp from the list of allowed applications in the Facebook panel.

4.4.1.1 DEVELOPMENT DETAILS

The procedure initially opens a new session with Facebook using the standard callback. The Facebook callback takes charge of the user login.

The session is opened and a request for publish permission is made. Then the new post is created, inserting the desired information in a Facebook *GraphObject*, and published.

Data exchanged between the Forum and the Facebook site are:

- the title of the thread that contains the post to be shared
- the URL of the post that can be successively read by means of a web browser (the access is read-only).

Then the app shows a dialog, which informs the user that the operation has been completed successfully. In order to implement this communication the Facebook SDK library is included in the Forum's apklib. Further implementation information (Android manifest file and pom.xml file) is available in D.8.1.2 "Social Network Analysis".

4.4.2 TWITTER INTERLINKAGE

The first time the user tries to share content on Twitter, a web browser is launched. In this window Twitter asks the user for the Twitter credentials in order to authorize MApp to share that content amongst her/his own tweets.

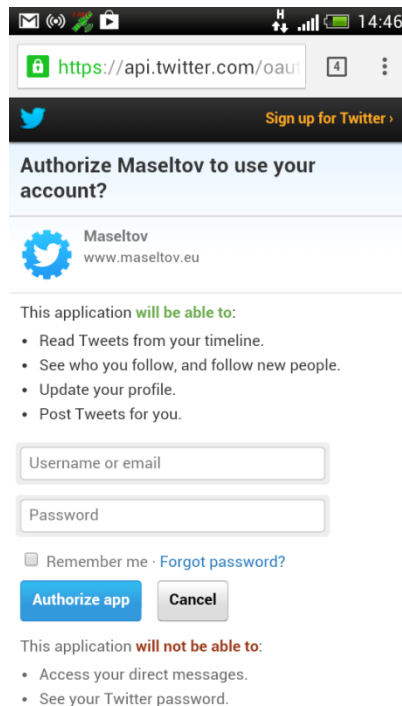


Figure 18: Twitter Interlinkage

User's Twitter credentials are read only by Twitter login interface and not stored in MASELTOV. Given the user doesn't remove MApp from the list of allowed applications from her/his Twitter panel no further authorization will be required.

4.4.2.1 DEVELOPMENT DETAILS

The procedure initially uses a standard request and asks for a new access token in order to authenticate the user. The Twitter callback takes charge of the user login. The session is opened and a request for publish permission is made. Then the new tweet is created and published.

Data exchanged between the Forum and Twitter are:

- the title of the thread that contains the post to be shared
- the URL of the post that can be successively read by means of a web browser (the access is read-only).

Then the app shows a dialog, which informs the user that the operation has been completed successfully. In order to implement this communication the Twitter library is included in the Forum's apklib. Further implementation information (Android manifest file and pom.xml file) is available in D.8.1.2 "Social Network Analysis".

4.5 NAVIGATION SERVICE – SITUATED NAVIGATION ASSISTANCE (FLU)

Description:

To help users follow walking segments within their routes, MApp offers in the route view the option to switch from *Route Service* to the *Situated Navigation Assistance*.

Technical Implementation:

As soon as the route component identifies a walking segment within the route, an icon appears beside each walking segment, as depicted in Figure 19: Situated Navigation Assistance, having tapped on that icon, the view automatically switches to the *Situated Navigation Assistance* view. The *Route Service* sends the start point as well as the end of the walking segment. The two points are represented as a Lat/Lon coordinate pairs.

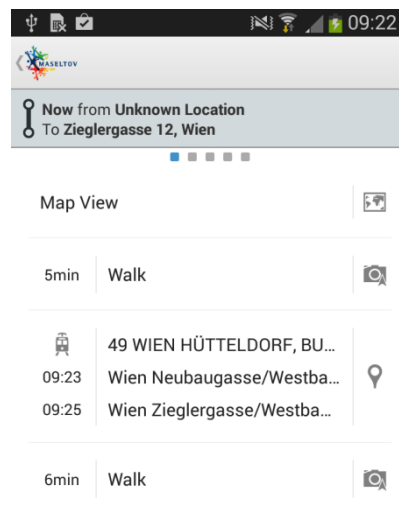


Figure 19: Situated Navigation Assistance

4.6 POI SERVICE AND NAVIGATION SERVICE (FLU)

Description:

When the user starts the *POI service*, she receives the POIs in the surrounding area of her current GPS position; she can filter what categories of POIs she wants to receive by activating or deactivating POI categories as depicted in Figure 20: POI Categories Filter.

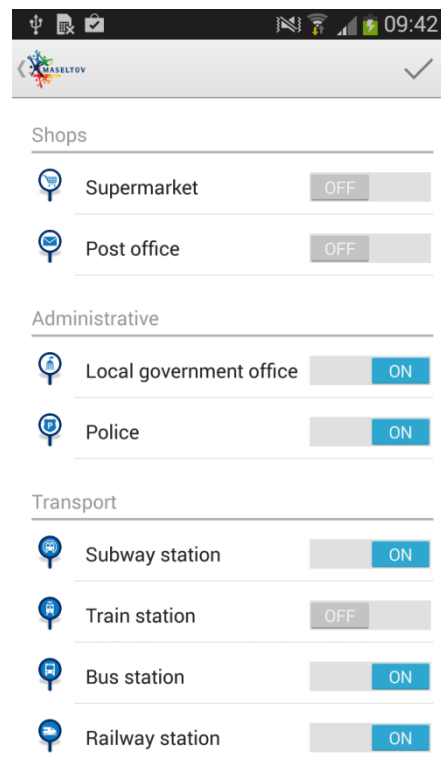


Figure 20: POI Categories Filter

Technical Implementation:

When the *POI Service* starts, the current position as Lat/Lon and a list of selected POIs categories are used to retrieve a list of POIs in the surrounding area. The POIs within the retrieved list are then shown on the map. By tapping on the pin of a point on the map view, a menu appears and offers the user the ability to use the selected point as a start or a destination of her trip as depicted in Figure 20: POI Categories Filter. Figure 21: Select one point as a "Start" or "Destination". Having selected "Start" or "Destination", the *Routing Service* starts and the selected point of interest is then set in the route view. The *POI Service* sends the selected point of interest to the *Routing Service*.

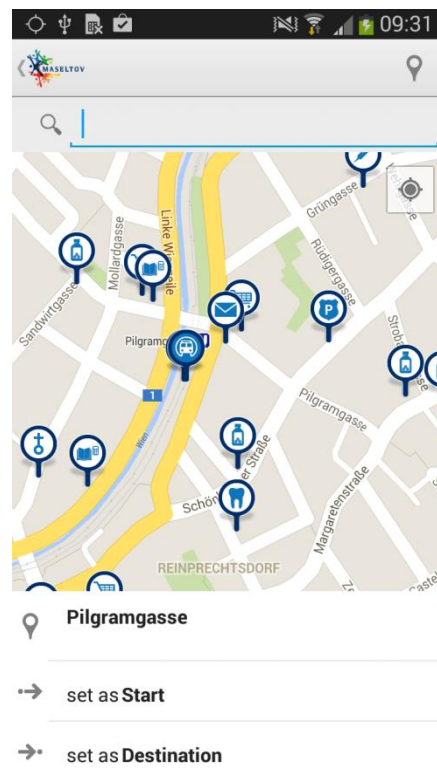


Figure 21: Select one point as a "Start" or "Destination"

4.7 INFO AND POI SERVICE (FLU)

Description:

MApp users can view the addresses included within the pages of the *Info Service*. By tapping on an address, the view of MApp switches from *Info Service* to *POI Service*, a pin of the chosen address is shown on the map, tapping on that pin, a menu appears and offers the user the ability to use the selected point as a start or a destination of her trip. Having selected "Start" or "Destination", the *Route Service* starts and the selected point of interest is then set in the route view.

Technical Implementation:

The addresses are viewed as hyperlinks on the pages of the *Info Service* as depicted in Figure 22: Addresses are viewed as Hyperlinks; MApp defines its own data protocol. The Android system recognizes that data protocol, and starts the *POI Service* whenever the user taps on the address hyperlink in an *Info Service* page.

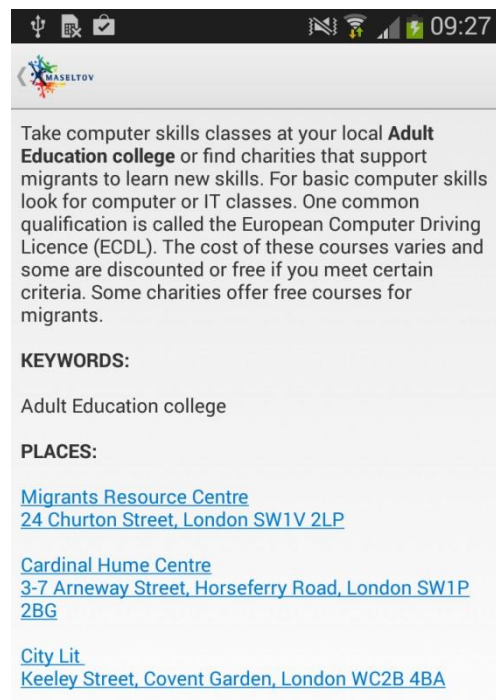


Figure 22: Addresses are viewed as Hyperlinks

4.8 INTEGRATING SERVICES THROUGH A COMMON CURRENCY (COV)

This section describes the rationale and approach taken to provide an additional layer of integration for the tools in the MApp, using the serious game as a means to generate value for currency which is "earned" through the use of other services. The currency can then be spent in the game on a range of outfits and other rewards for the player character. This allows for user transitions between the game and other MApp services to be incentivised, towards the goal of scaffolding a transition from using a single MApp component to understanding and interacting with the full suite of tools and services. In the first of the following three subsections, we outline the existing evidence supporting the use of virtual currency as a motivator, drawing parallels to other techniques such as gamification. The second subsection describes the specific design created for MASELTOV, with its implementation and technical aspects discussed in the final subsection.

4.8.1 CURRENCY AS A MOTIVATOR

In recent years, the notion of "gamification" as a means to adapt existing processes and systems to provide a more motivating and entertaining environment for the end-user has been extensively explored (Cechanowicz, Gutwin, Brownell, & Goodfellow, 2013; Kappen & Nacke, 2013). In some cases, this is achieved simply through the addition of elements such as scores, leaderboards, or trophies to reward intended behaviour (Barata et al., 2013). Some evidence exists to suggest this can be effective in improving educational or behavioural outcomes, though a counter-argument has also developed which considers these simplistic approaches to be naive and lack an understanding of how an underlying game is defined (Mekler et al., 2013). For example, Mekler et al. (2013) note that many common elements

can have negative as well as positive impacts on intrinsic motivation: a leaderboard, for example, can motivate able participants whilst demotivating those who struggle to attain high rankings. Similarly, trophies and achievements can promote playing to achieve them in the most efficient means possible, which can lead to serious objectives being circumvented or ignored.

More comprehensive frameworks suggest considering the wider context in which gamification is applied (Deterding, 2012; Rojas, Kapralos, & Dubrowski, 2013). Ultimately, this understanding of broader context, as well as an understanding of how a game might function within it, is fundamental in the application of game elements to drive motivation and behaviour. In the MASELTOV context, therefore, we reflect on this context and related mechanisms for learning, for example the Incidental Learning Framework (D7.1.2). We also consider the ethical framework within the project and how interactions between users are handled. This firstly leads us to consider that competitive elements, such as the leaderboard example, may serve to demotivate immigrants who rank poorly. Whilst careful implementation may ameliorate or eliminate this risk, it suggests other mechanisms might hold precedence. Achievements and trophies also hold a similar concern, as their use as a motivational tool depends on the ability to generate a perception of value amongst users (Montola et al., 2009), for which social mechanisms or tangible rewards are beneficial. In the case this social mechanism is competition amongst peers, for example to gain achievements which other users struggle to attain, or gain more achievements than other users, the concern regarding competition remains.

Currency, by comparison, has the dual benefit of an immediately relatable real-world analogue for non-gamers, and a system which can focus on an individual's perception of reward and value rather than a social dimension. The first benefit is useful for the case of immigrants who may enter the MASELTOV platform for other services, and have a limited understanding or knowledge of games and gamification. The second benefit allows competitive social mechanisms to be forgone in favour of an individualised approach. To realise this effectively, however, requires that a perception of value be generated for the currency: it must be relatable to rewards to drive behaviour. Substantial evidence exists to suggest that for online games, players are willing to purchase virtual currency which can be spent on in-game items for real-world money (Debeauvais, Nardi, Lopes, Yee, & Ducheneaut, 2012). However, rather than seek to use currency as a means by which to get immigrants to pay for services, the free-to-use model at the core of MASELTOV allows this behaviour to be explored in terms of the investment of time by the immigrant in using different MApp services, rather than paying a real-world fee.

As per the proverb, "time is money", this explores the notion that virtual currency, provided sufficient reward incentive exists, can be used to encourage the user to invest time. The design approach detailed in the next section, therefore, defines how both perceived value is generated through the serious game (D7.4.2), and how other services reward currency in response to use.

4.8.2 DESIGNING A CURRENCY SYSTEM FOR MASELTOV

As alluded to in the previous section, the two key research questions underpinning the design of the currency system are:

1. How can perception of value for currency be generated within the MApp?
2. How can this perception of value be used to drive positive behaviours?

With respect to (1) above, the challenge lies in rewarding currency through the ability to spend it, without compromising accessibility for tools and content across the platform. For example, if a language lesson requires currency to be accessed, then whilst this may incentivise the earning of currency, it also restricts access to learning content. Similarly, requiring a level of currency be earned before using tools such as AR Navigation or the Translation Tool would restrict their use. Whilst generating a perception of value in currency is desirable in terms of its ability to encourage behaviours, it is fundamentally a mechanism rather than an end-product: the immigrant gains the benefits of the MApp through the use of tools and services rather than the accumulation of virtual currency. Fortunately, through the inclusion of a serious game, the MApp platform has a clear service through which currency can be rewarded without compromising access to services. Noting existing research suggests in-game cosmetic items can drive purchases and incentives as well as functional upgrades (Debeauvais, et al., 2012), a game can provide a useful medium to provide such rewards. Consequently, effort was invested within T7.4 to create a cosmetic upgrade path for the player character, as illustrated in Figure 23: Purchasing character upgrades in-game using coins.



Figure 23: Purchasing character upgrades in-game using coins

This required the development of various outfits ("skins") and hairstyles/hats. These were then implemented in an in-game store allowing the player to spend coins on the upgrades. Initial prototypes for the game (D7.4.1) allowed the player to earn coins in-game through searching and collecting; these were removed with a view towards incentivising the use of other services. Hence, the only way the player can earn coins and subsequent upgrades is to leave the game and use other tools. This leads to point (2) above: how can the perception of value be used to drive behaviours. This required the development of a strategy by which other services reward currency. Tools and services were selected based on their suitability for this goal. In particular, the language lessons provide a particularly good synergy, due to their ability to assess progress through tests as well as monitor usage. Hence, for the language lessons a system was implemented that rewards both coins for completing activities, regardless of outcome, and bonus coins for posting test scores ranging from "ok" (1 coin) to "excellent" (3 coins). As lessons can be repeated, an objective is to avoid negative,

demotivational effects from low-scores by allowing the learner to repeat the task, and with a constant, though slower, currency gain for doing so regardless of outcome.

Help Radar requires significant time investment to undertake and rate an assistance. As such, the coin gain for doing so is correspondingly higher (50 coins). In fact, undertaking and rating 10 assistances would likely reward the user with sufficient coins to "buy out" the in-game store. Field trials will examine its efficacy in this context: understanding whether the currency is seen as a primary motivator or a beneficial side-effect of undertaking assistances is important to assess. It is also possible for the system to be circumvented, for example with an immigrant and collaborator repeating simple and trivial assistances. However, the goal of the currency system at the research stage is not to provide a robust, watertight system, rather to explore its effect on motivation. As such, if it is observed to drive such behaviours to work-around the system, this in itself would be a promising result in terms of the ability of a currency system to drive behaviours. Subsequent exploitation could then refine the system to fix loopholes established through field trial.

For the Forum, 5 coins are rewarded for creating a new thread or replying. Again this system is not robust (a user, for example, could post short or nonsensical messages); however, allowing this also enables observation of whether the currency system can influence behaviour. This relates to observations regarding behaviours on forums which, by nature of common bulletin board software, tend to expose post count to other users (Ghosh & Kleinberg, 2013). As such an intrinsic, though not explicit, gamification of forums can occur, with users competing for post counts to gain prestige. Both AR Navigation and Translation Tool also reward users, at a rate of 1 coin per minute of usage.

4.8.3 TECHNICAL IMPLEMENTATION

The currency value is held as an integer in the user profile, with simple methods implemented to query, add, and remove coins. This provides a lightweight and compatible way for services to interact around the value, without requiring complex implementation. It also allows for future services and tools to quickly implement the currency value without needing to develop an additional interface. The game also scaffolds the user's introduction to the currency through interactive dialogue. This serves as a bridge between the fictional narrative of the game and the real-world service use. Figure 24: Example of an in-game dialogue script explaining the coin system immersively illustrates a section of XML underlying a branching dialogue with the information officer in the opening scene of the game.

```
<main>Couldn't you help me out and donate a few spare coins?</main>
- <choice0>
  <main>Ah, there are hardly any coins in this world. Using these other "mApp" tools is your only option.</main>
  - <choice0>
    <main>Looks like I'll need to try them out - thanks!</main>
    - <choice0>
      <main>NEW_ROOT Dialog.choice4</main>
    </choice0>
  </choice0>
- <choice1>
  <main>Alright, alright. I'll need some more advice, then...</main>
  - <choice0>
    <main>NEW_ROOT Dialog</main>
  </choice0>
```

Figure 24: Example of an in-game dialogue script explaining the coin system immersively

The goal here is to link between the service use external to the game, and the internal narrative which in turn guides the user in the in-game store, contributing to the perceived value of the currency and items. The in game store itself provides an immersive and integrated section of the game allowing the user to browse items, shown in Figure 25. Figure 25: The player character in the in-game shopping area. Note the coin total displayed on screen. The coin total is also permanently displayed in-game at the top right of the viewable area.



Figure 25: The player character in the in-game shopping area. Note the coin total displayed on screen.

5. NOTIFICATIONS (JR)

This chapter shows the current collection of messages which are sent to users from MApp components. Those messages are directly sent to the Notification Bar by the use of the corresponding Android API. All messages will be grouped under just one root entry in the notification bar called i.e. “MaseltoV: x pending messages”. Grouping of Messages is provided natively by the Android API available in Android version 4.1 and above.

The following table gives an overview of all notifications used within MApp and the linkage to certain MApp services.

Table 20: List of user notifications sent by MApp services.

Software module	Description of the notification - Intention of the notification - wished reaction of users	Text of the notification shown to the user	when is notification sent - trigger?
Help Radar (TI)	New message has arrived	Message from <nickname>	This message is sent when user starts a chat session with volunteer
Help Radar (TI)	Help radar checks if GPS is turned on. If not, a notification is sent. User should turn GPS on	GPS signal required	At device boot
MASELTOV forum (TI)	New personal message has arrived	New Personal Message	This message is sent when a new forum personal message is sent to the user
Recommendation service (AIT)	Recommendations are sent to the notifications area	"You have # pending recommendations"	When one or more recommendation rules fire

6. DATA EXCHANGE (AIT)

Events are sent from MApp components in order to send data that could be useful for the recommender system in order to produce personalised recommendations for the user.

Every time a MApp component needs to send an event to User Profile Content Provider (UPCP) creates a JSON object and sends it to the URI shown in Table 21.

`content://com.ait.userprofile.AITUserProfileProvider/insert/event`

Table 21: URI for sending events to User Profile Content Provider.

The JSON object must contain three elements; *timestamp*, *source* and *info*.

Timestamp is provided in format YYYYMMDDHHIISS, source is a plain text that is used to identify the MApp component and the reason of sending the event, and info is a JSON object that contains any key value pairs of information that describes the event. Table 22 and Table 23 show the JSON objects of a rather simple and a more complicated event used in MApp.

```
{
  "timestamp":20140905133020,
  "source":"User.Location",
  "info":
    {
      "lat": "37.943664",
      "long": "23.869982"
    }
}
```

Table 22: JSON Object for User Location event.

```
{
  "timestamp":20140905133020,
  "source":"MaseltovContext.ModeOfTransportation"
  "info":
    {
      "id": "dceedde4",
      "device": "8ff6eb7c-ff3c-40c2",
      "module": "TransportationMode",
      "timestamp": "1353417528",
      "event_timestamps": [1353417528, 1353417248, 13534175125],
      "activity_types": [1,2,3],
      "confidences":[100, 35, 65]
    }
}
```

Table 23: JSON Object for MaseltovContext.ModeOfTransportation event.

As shown in Figure 26: Workflow of events, the UPCP stores all events in a local database (SQLite). When network is available the events stored in local database are transmitted to the backend server using the API and upon successful transmission they are removed from the local database. When events are stored in the backend server's database (MySQL) the recommender is triggered in order to start processing the event. If the event triggers a recommender rule and produces a recommendation, the latter is shown in the recommendations list in Recommendations component.

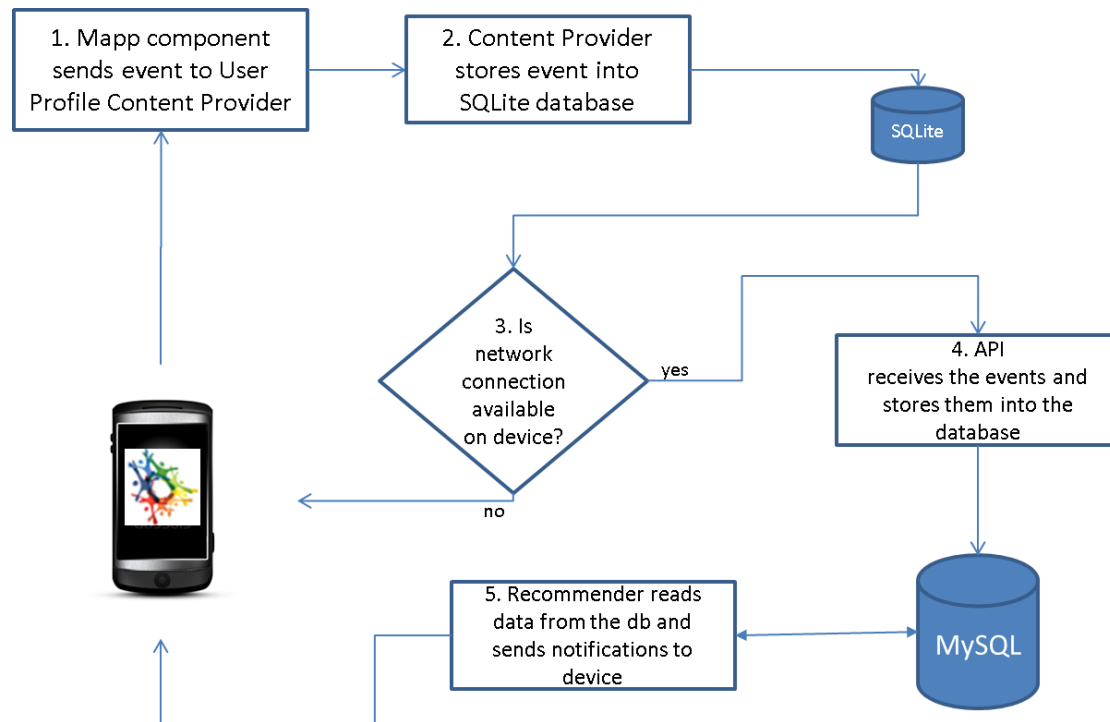


Figure 26: Workflow of events.

Events received are grouped into three categories; usage events, progress events and interesting things.

6.1 USAGE EVENTS

These events determine the usage of the components. We use these events to produce the statistics shown in the components usage pie chart under the User Profile → Statistics in MApp. Each MApp component produces such type of event every time their components goes on pause or closes. Table 24 shows the events sent from each MApp component. Duration is sent in seconds.

MApp Component Name	Responsible Partner	Source	Values of <key,value> pairs
Language learning	PP	LanguageLearning	("duration", <duration>)
Translation Tool	CTU	Translation Tool	("duration", <duration>)
User Profile	AIT	User Profile	("duration", <duration>)
Recommendations	AIT	Recommendations	("duration", <duration>)
Info	FLU	fluinfo	("duration", <duration>)
Augmented reality	JR	maseltov-arnav	("duration", <duration>)
Help Radar	TI	Help Radar	("duration", <duration>)
Navigation	FLU	flunav	("duration", <duration>)
Places of Interest	FLU	flupoi	("duration", <duration>)
Serious Game	COV	SeriousGame	("duration", <duration>)

Table 24: Usage Events sent to user Profile Content Provider by each MApp Component.

6.2 PROGRESS EVENTS

These events record the progress of certain user's activities in MApp components. The Language Learning component sends the events as shown in Table 25 every time either when

- The user completes an activity in a language lesson, or
- The user completes a test at the end of a language lesson, or
- The user rates a statement about their language lesson within a value of 1-5.

MApp Component Name	Responsible Partner	Source	Values of <key,value> pairs
Language learning	PP	LanguageLearning.ActivityCompleted	("publication", <string: publication code>), ("lesson", <string: lesson id>), ("task", <string: task id>), ("progress", <number:

			percentage>)
Language learning	PP	LanguageLearning.LessonTestScore	("publication", <string: publication code>), ("lesson", <string: lesson id>), ("score", <number: percentage>)
Language learning	PP	LanguageLearning.LessonSelfEvaluation	("publication", <string: publication code>), ("lesson", <string: lesson id>), ("statement", <string: statement id>), ("rating", <number: integer 1..5>)

Table 25: Progress Events sent to user Profile Content Provider by each MApp Component.

6.3 INTERESTING THING EVENTS

These events mainly carry contextual information like the current user's location, information about the activities of the user, user's social communications etc.

MApp Component Name	Responsible Partner	Source	Values of <key,value> pairs
User Profile	AIT	GPS tracking	("Longitude", <long>), ("Latitude", <lat>)
Translation Tool	CTU	Translation Tool	("detectedText", <user corrected text in the image>)
Maseltov Context	JR	MaseltovContext. ModeOfTransportation	("type", <String>), ("confidence", <int>)
Maseltov Context	JR	MaseltovContext. ActivitySummary	("total_distance", <long>), ("distance_walking", <long>), ("distance_driving", <long>), ("distance_biking", <long>),

			("distance_unknown", <long>), ("time_moving", <long>), ("time_still", <long>), ("time_walking", <long>), ("time_driving", <long>), ("time_biking", <long>), ("time_unknown", <long>)
Maseltov Context	JR	MaseltovContext. Interests	("interests", <String keyword list>), ("weights", <Integer weight list>)
Maseltov Context	JR	MaseltovContext. PlaceEntry	("longitude", <double>), ("latitude", <double>)
Maseltov Context	JR	MaseltovContext. PlaceExit	("longitude", <double>), ("latitude", <double>)
Maseltov Context	JR	MaseltovContext. PlaceHistory	("coordinates", <Double coordinates list>), ("type", <String types list>), ("ts_entry", <Long ts_entry list>), ("ts_exit", <Long ts_exit list>), ("osm_id", <Integer id list>)
Maseltov Context	JR	MaseltovContext. PlacesOfInterest	("coordinates", <Double coordinates list>), ("type", <String>), ("visit_duration", <long duration list>), ("visit_count", <Integer visit_count list>), ("osm_id", <Integer id list>)
Maseltov Context	JR	MaseltovContext. SocialInteraction	(Only for illustration purposes in JSON) "contact_data": [{ "contact_id": "int", "calls" : ["call" : { "ts": "long", "duration" : "int", "type" : "String"}}, ...], "messages" : ["message" : {"ts": "long", "type" : "String"}, ...] {...}]
Help Radar	TI	Help Radar. signupVolunteer	("username", <String>), ("knowledges", <String>), ("languages", <String>), ("status", <String>)
Help Radar	TI	Georadar. removeVolunteer	("volunteerUsername", <String>)
Help Radar	TI	Georadar. statusVolunteer	("volunteerUsername", <String>), ("status", <String>)

Help Radar	TI	Help Radar. contactVolunteer	("username",<String>), ("volunteerUsername",<String>), ("reqKnowledges",<String>), ("reqLanguage",<String>)
Help Radar	TI	Help Radar. ratingAssistance	("username",<String>), ("volunteerUsername",<String>), ("rating",<String>)
Forum	TI	Social Forum.sendPost	("username",<String>)
Forum	TI	Social Forum.sendReply	("username",<String>)
Places of Interest	FLU	flupoi.PoiSearch	("keyword",<String>)
Info	FLU	fluinfo.CategoryTitle	("categoryTitle",<String>)
Info	FLU	fluinfo.ArticleTitle	("articleTitle",<String>)
Navigation	FLU	flunav.RouteStart	{"pointAs":"start", "title":<title>, "latlng":<latitude>:<longitude>: WGS84:", "type":<type>"}
Navigation	FLU	flunav.RouteEnd	{"pointAs":"end", "title":<title>, "latlng":<latitude>:<longitude>: WGS84:", "type":<type>"}

7. MAPP WORKFLOW CONTROL (AIT)

7.1 LOGIN/LOGOUT OPERATIONS IN MAPP

In order to use the MApp the user must login with a valid MASELTOV account. The first screen the user sees when starting the MApp is a register/login screen. Additionally, anonymous use of the MASELTOV services is also supported by registering and logging in from a specific device without the need to use their e-mail address. When registering, users are asked to provide a minimal set of mandatory information for the purpose of identifying them and allowing also the correct functioning of some MApp applications. Thus, users are asked to enter an e-mail address, a password, a username of their choice and the user's city. The e-mail serves as a user identifier for the platform, even though it may refer to a non-existent service provider. For the MASELTOV platform the e-mail is treated as a string identifier and actually reveals no information about the identity of the user. The username is a user chosen identifier that allows him/her to be referenced in the MASELTOV forum, and again, it bears no relation to the actual user identity. The e-mail and username fields (both mandatory) are necessary to register a user and allow him/her to hold an account to the MASELTOV platform. Finally, the name of the user's home city is mandatory, and is used for initializing other MApp applications like the POI. The workflow diagram of Figure 27: Inter-relationship Diagram shows the interconnections

between the User Profile activities and the other MApp applications. After a number of activities are invoked the User Profile will return to the activity that first called it.

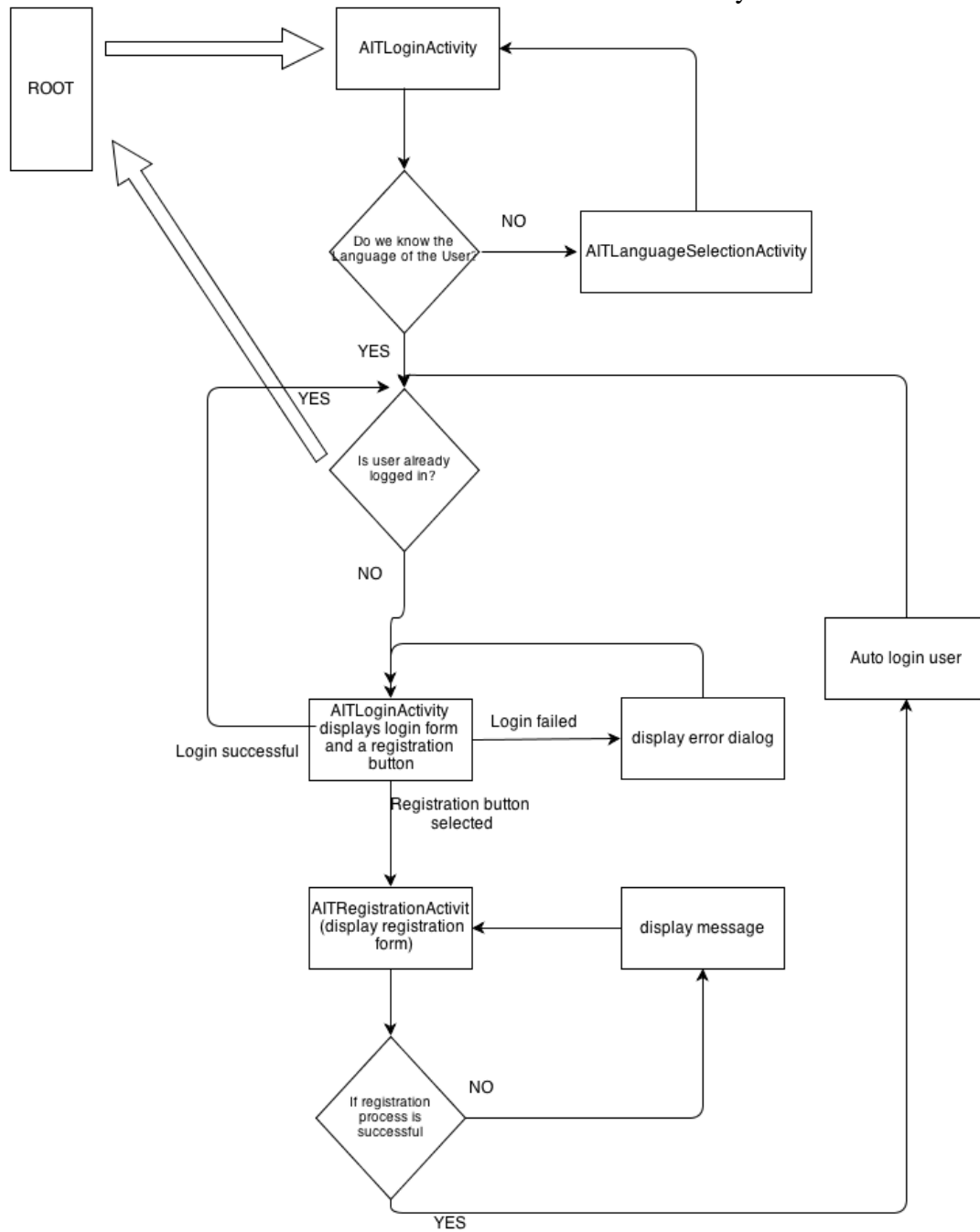


Figure 27: Inter-relationship Diagram

Login is performed using Lamport hashes. The server maintains for each user a pair $(k, h^k(p))$ where h is a hash function and p is the user selected password. After a successful login k is decremented by 1 and the new pair $(k-1, h^{k-1}(p))$ replaces the previous one. When k reaches 0 then a new password must be entered by the user.

Using this mechanism all API communications are completed using a different transmitted “password” for each user every time the user performs a new login. As this token decreases,

at some point (for example when the n number from 100 reaches 1) the User Profile will ask the user to change his/her password and the token will be updated back to its starting point.

The user has the option to logout from the MApp Dashboard top right menu. As soon as the user logout from the application the login screen will appear for the next login.

7.2 QUERYING LOGIN STATUS

It is very important for each MApp component to know if there is a logged in user or not. Since this information is stored in User Profile, there is a functionality built so that MApp components can query the status of the user. MApp components use User Profile Content Provider in order to query the login status of the user. Table 26 shows the URI used for this purpose.

Table 26: URI for requesting login status of the user from the User Profile Content Provider.

<code>content://com.ait.userprofile.AITUserProfileProvider/isUserLoggedIn</code>
--

If a user is currently logged in then the UPCP return the user's id and the his/her username.

7.3 BROADCASTERS

Broadcasters are used in order to alert MApp components that a status or a value in User Profile has changed. This change may have to do with updating a value like the city of the user or by getting a notice that the user logged out and all services related with MApp must stop. Following is a description of each broadcaster initiated by User Profile.

Android's Context.sendBroadcast method is used to broadcast the action. Action is set by creating an Intent (passed to sendBroadcast method). Table 27 shows a sample of how User Profile sends the login broadcast.

Table 27: Sample of how User Profile sends a Login Broadcaster.

<pre>Intent i = new Intent("com.ait.userprofile.USER_LOGGEDIN"); context.sendBroadcast(i);</pre>
--

7.3.1 BROADCASTERS FROM USER PROFILE

The User Profile provides all the information about the user and is also responsible for login and logout operations. There is a number of operations that need to be immediately addressed to the MApp components. This is possible through a number of broadcast messages transmitted by the User Profile.

7.3.2 LOGIN BROADCASTER

This is broadcasted every time the user is successfully logged in into the MApp. This broadcaster will start background services in the start-up procedure. The Intent's action for this broadcaster is `com.ait.userprofile.USER.LOGGEDIN`.

7.3.3 LOGOUT BROADCASTER

This is broadcasted every time the user selects to logout from the option in the MApp Dashboard. This broadcaster intends to inform MApp components to terminate any services running. The Intent's action for this broadcaster is `com.ait.userprofile.USER.LOGOUT`.

7.3.4 SETTINGS CHANGED BROADCASTER

This is broadcasted every time one of the settings under the User Profile is changed from on to off or vice versa. This broadcaster is to inform other MApp components to start or stop collecting data depending on the user's choice. The Intent's action for this broadcaster is `com.ait.userprofile.SETTINGS_CHANGED`.

7.3.5 USER PREFERENCES CHANGED BROADCASTER

This is broadcasted every time the user changes the value of one or more fields in User Profile. This broadcaster is to inform MApp components that a value of the user's preferences is changed and they might have to refresh their local values. The Intent's action for this broadcaster is `com.ait.userprofile.USER_PREFERENCES_CHANGED`.

8. FINAL INTEGRATION STATUS (FLU)

As already mentioned in the previous chapters, the MASELTOV services will be developed by different partners in a very autonomous way. To end up with an integrated MASELTOV application a process is needed that fits the specific requirements.

8.1 TECHNICAL SOLUTION FOR SOFTWARE INTEGRATION

Within this chapter the software integration process will be described on the architectural level. Figure 28: Integration process. depicts the final integration process. Afterwards, the required steps for the integration will be described including the used tools.

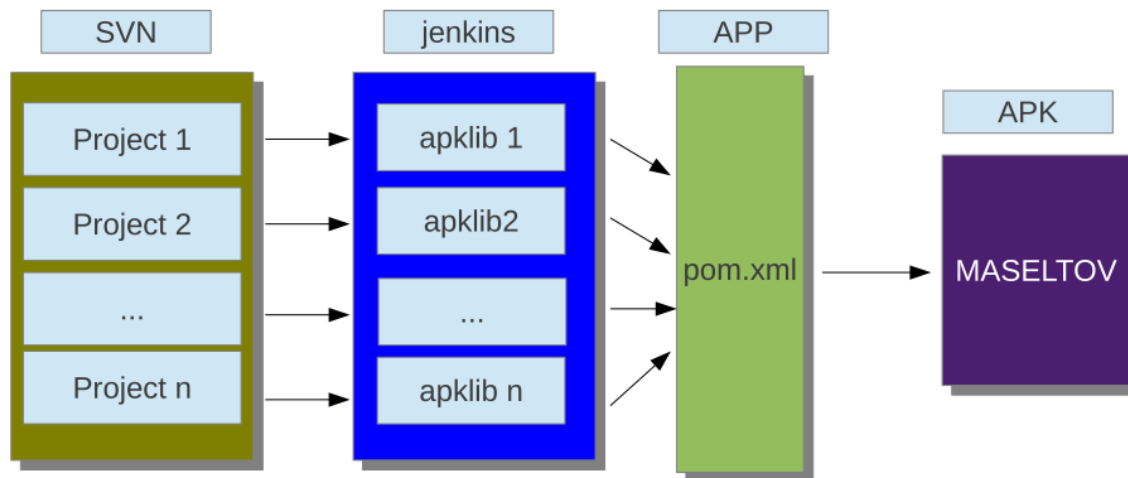


Figure 28: Integration process.

The integration process is based on source-code integration process which is enabled through subversion (SVN), Maven, a build and packaging framework and Jenkins, an open source continuous integration tool. The fundamentals of these tools will be discussed in the next chapter.

In the first step, a SVN repository is needed for each partner and each project must contain a maven pom file where the dependencies are defined. There exists two possibilities how to set up the development environment:

- Either the project partners use the server infrastructure from Fluidtime or
- They set up their own SVN and Maven Repository and link it to Fluidtime services.

For the first solution Fluidtime will use its internal maven repository to produce the apklib files from the modules developed by project partners.

For the second solution Fluidtime takes the apklib file which was already created by the project partner itself.

In the last step, Fluidtime will use a skeleton project where all services of MASELTOV are combined and compile the MASELTOV.apk file

Following the input of technical partners, with respect to the chosen development environment, the deduced final development set-up is depicted below.

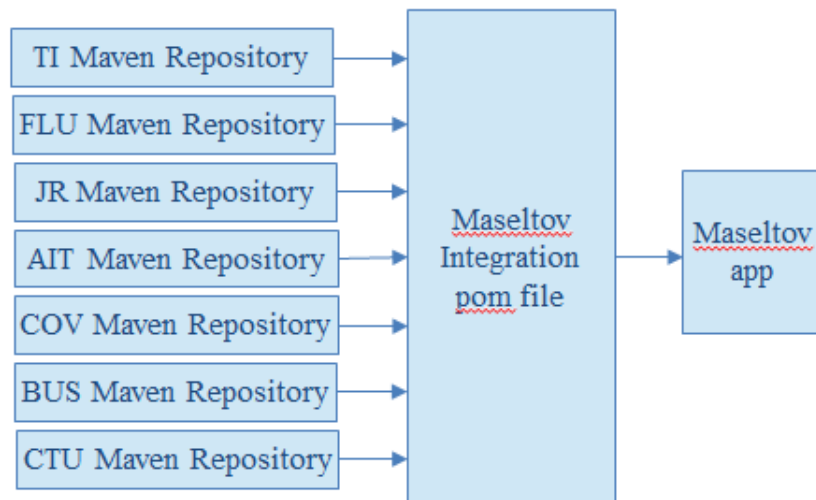


Figure 29: Set-up for the development environment.

Every partner has to set-up its own development environment and Fluidtime will link to their Maven repositories where the pom file is placed.

8.2 CURRENT STATUS OF INTEGRATION

Figure 30 represents the actual status for the integration of the MApp application. Currently, fully integrated partners are JR, TI, AIT and FLU.

Due to technical limitations and the high demand of mobile phone resources (memory, CPU usage, ...) it was decided, that the components of COV – Translation Tool - and CTU – Game - it was decided that their components will be developed as stand-alone modules which will be seamlessly linked to MApp. Additionally, the new partner Pearson Publishing was integrated as a stand-alone component.

Beside linkage to the dashboard those tools are linked by recommendations and also deliver user related data to the user-profile component. The storage demand of the serious game for example is more than 30MB, which would blow up MApp to an overall size which would lead to serious problems with regard to deployment and user acceptance.

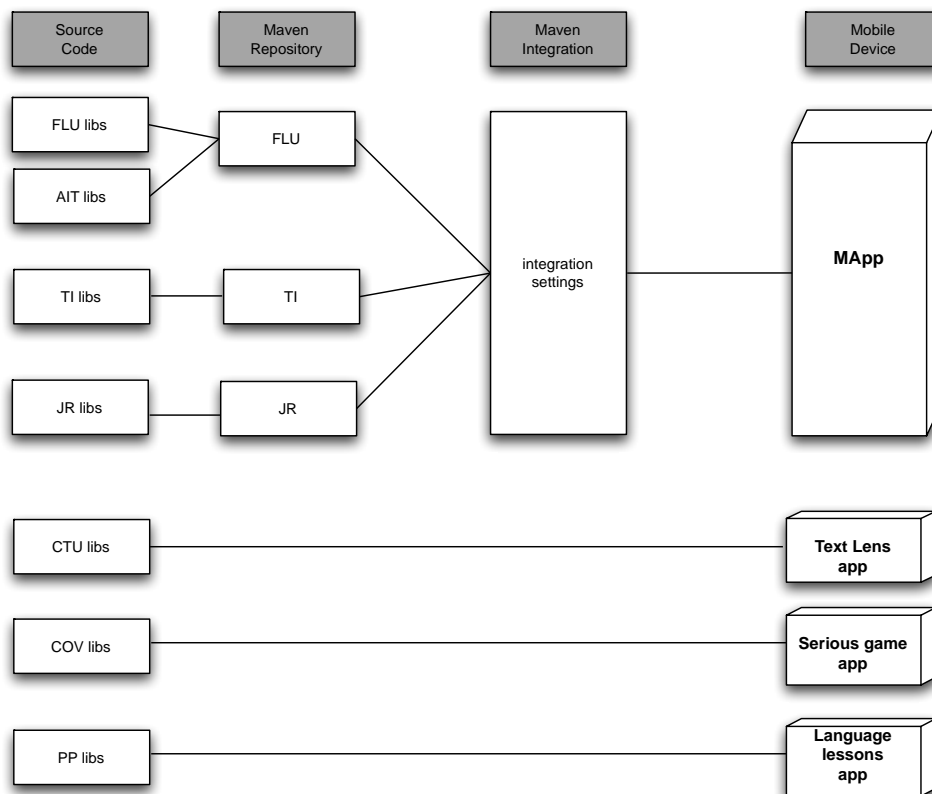


Figure 30: Set-up for the development environment.

Since all partners were in the development phase, several changes in their application were implemented. The fourth integration run (took place in September) was not successful since development partners used different libraries. Based on this, Fluidtime and all development partners agreed on the library versions and dependencies shown in the following table.

Table 28: Android library versions and dependencies.

Group ID	Artifact ID	Type	Version	Scope	Pom dependency
Android					
com.google.android	android	jar	4.0.3	Provided	
	android-support-v4	jar	r18		<dependency><groupId>android.support</groupId><artifactId>compatibility-v4</artifactId><version>18</version></dependency>
	gcm-server	jar	r3	Provided	<dependency><groupId>com.google.android</groupId><artifactId>gcm-server</artifactId><version>r3</version></dependency>

	com.google.android.gms	google-play-services	jar	v12		<code><dependency><groupId>com.google.android.gms</groupId><artifactId>google-play-services</artifactId><version>12</version><type>apklib</type></dependency></code> <code><dependency><groupId>com.google.android.gms</groupId><artifactId>google-play-services</artifactId><version>12</version><type>jar</type></dependency></code>
		google-play-services	apklib	v12		
	com.google.code.gson	gson	jar	2.2.2	Compile	<code><dependency></code> <code><groupId>com.google.code.gson</groupId></code> <code><artifactId>gson</artifactId></code> <code><version>2.2.2</version></code> <code><scope>compile</scope></code> <code></dependency></code>
	com.loopj.android.http	android-async-http	jar	1.4.2		<code><dependency></code> <code><groupId>com.loopj.android.http</groupId></code> <code><artifactId>android-async-http</artifactId></code> <code><version>1.4.2</version></code> <code></dependency></code>
Fluidtime	com.fluidtime.android.maseltov	maseltov_route	apklib		Compile	
		maseltov_maps	apklib		Compile	
		maseltov_lib	jar		Compile	
		maseltov_json	jar		Compile	
JR	at.joanneum.maseltov	maseltov-context	apklib	0.0.1-SNAPSHOT	Compile	<code><dependency></code> <code><groupId>at.joanneum</groupId></code> <code><artifactId>maseltov-context</artifactId></code> <code><version>0.0.1-SNAPSHOT</version></code> <code><type>apklib</type></code> <code></dependency></code>
		maseltov-arnav	apklib	0.0.1-SNAPSHOT	Compile	<code><dependency></code> <code><groupId>at.joanneum</groupId></code> <code><artifactId>maseltov-arnav</artifactId></code> <code><version>0.0.1-SNAPSHOT</version></code> <code><type>apklib</type></code> <code></dependency></code>

		maseltov-poi	jar	0.0.1-SNAPSHOT	Compile	<dependency> <groupId>at.joanneum.mase ltov</groupId> <artifactId>maseltov- poi</artifactId> <version>0.0.1- SNAPSHOT</version> </dependency>
TI	com.tilab	georadar	ap kli b	0.0.1-SNAPSHOT	Compile	<dependency> <groupId>com.tilab</groupId> <artifactId>georadar</artifactId> <version>0.0.1-SNAPSHOT</version> <type>pom</type> </dependency>
		socialapp	ap kli b	0.0.2-SNAPSHOT	Compile	<dependency> <groupId>com.tilab</groupId> <artifactId>socialapp</artifactId> <version>0.0.2-SNAPSHOT</version> <type>pom</type> </dependency>
AIT	com.maseltov	userprofile	jar	1.0-SNAPSHOT	Provided	<dependencies> <dependency> <groupId>com.google.android</groupId> <artifactId>android</artifactId> <version>4.1.1.4</version> <scope>provided</scope> </dependency> </dependencies>
Apache	org.apache.httpcomponents	httpclient	jar	4.2.5	Compile	
		httpmime	jar	4.2.5	Compile	
Maven - Android Plugin	com.jayway.maven.plugins.android.generation2	android-maven-plugin		3.6.0		
Twitter	org.twitter4j	twitter4j-core	jar	3.0.3		<dependency> <groupId>org.twitter4j</groupId> <artifactId>twitter4j-core</artifactId>

					<version>3.0.3</version> </dependency>
Facebook	com.github.avianey	facebook-android-api	apklib	3.5	<dependency> <groupId>com.github.avianey</groupId> <artifactId>facebook-android-api</artifactId> <version>3.5</version> <type>apklib</type> </dependency>

All development partners agreed to use the Android version 4.1.1.4 which will be also available in the Maven repository.

During the development phase it was decided to set-up a Google Doc for the integration runs. The idea behind this is that every partner informs if they delivered their component and what are the new features. Consequently, the integration run can be executed on time and every partner is informed about the new features within the new version of MApp.

The next table represents the actual status of the integrated modules (September 2014):

MApp Service / Responsible Partner	Version Number	Implemented features/bugs in comparison to the previous version	Maven Repository (y/n)	.apk File (y/n)
userprofile / AIT	1.0.14	- Login screen changed. Login and Register tabs added on the top of the screen. - Disable Account option added in User Profile -> About You		
recommendations / AIT				
TextLens	0.9.6	Added events for evaluation ((GUINavigation, "NewPhoto" "ShareDetection" "PhraseTool" "Dictionary")) Updated event in coins integration: "source", "TextLens.addCoins"	n	y
maseltov-arnav	0.0.5-SNAPSHOT	Coins are labeled after source now	y	
maseltov-context	0.0.4-SNAPSHOT	Updated to latest POI tool, adjusted query radi.	y	
maseltov-poi	0.0.4-SNAPSHOT	POIs are now guaranteed to have an address. Query with poi.getMetaData(MetaDataPoint.KEY_ADDRESS)	y	
Help Radar / TI	0.1.0-	- changed source field in coin events;	y	y

	SNAPSHO T	- fixed marker gender-based color on map; - changed strings in select field of knowledge;		
	0.1.0- SNAPSHO T	- changed source field in coin events; - added threadpool for asynhttp requests; - changed URL behaviour in posts; - fixed annoying notifications behaviour; - removed action button in notifications; - fixed 'follow post'; - fixed new private message activity stack;	y	y
SocialApp / TI Language Learning	1.4	Coins are labelled after source	n	y
maseitov- navigation		- Interlinkage between info and POI/navigation	y	n

8.3 INTEGRATION PLAN

In order to avoid the problems with the integration, the following two steps are important:

- All partners use the same Android versions and libraries (see previous chapter for details)
- Adaption of the Maven repository: in order to avoid integration problems at the early stage, FLU made adaptations for the MAVEN repository (see Figure 31). The idea behind is to have a test project, where all modules are tested and then the actual integration will take place.

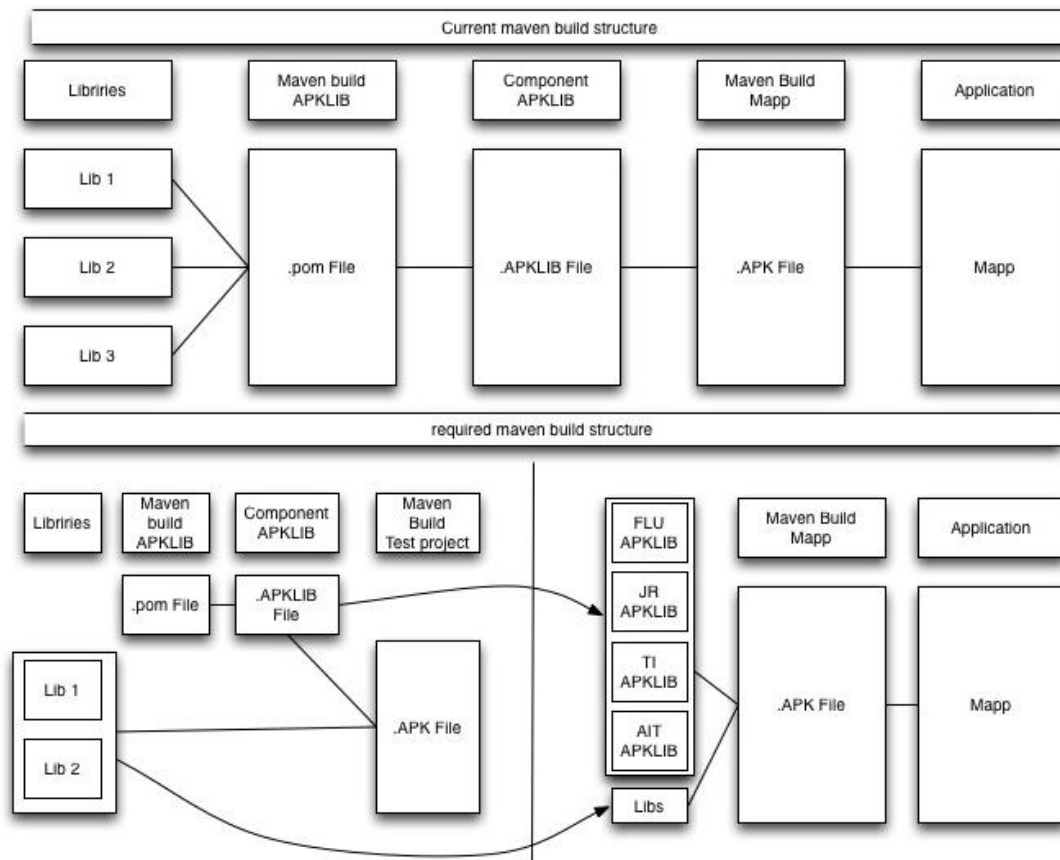


Figure 31: Adaption Maven repository.

The new concept was tested successfully middle of November 2013. After the successful test, the integration run has been based on the adapted structure. During the development phase the integration run was conducted on a monthly basis in order to have the current development status of partners integrated in MApp permanently. Since the final field trials are planned for middle of October, the integration run will be executed on weekly basis in order to have stable, final MApp release for the trials available.

9. 3RD PARTY PLUGIN CONCEPT (JR)

The MApp offers the possibility for third party tools to integrate with the user profile. Language Learning, Translation Tool and Serious Game are examples of plugins which extend the functionality of MApp considerably. Integrating with MApp allows for querying data from the user profile as well as sending context events and content to other components. In the following, the necessary steps to adapt an application to communicate with MApp are described.

9.1 QUERYING DATA

The concept for querying (as well as communicating own events) is realized through the User Profile component by an Android content provider. This commonly accessible interface manages user registration and authentication as well. The basic methods necessary for retrieving user related data are:

9.1.1 IS USER LOGGED IN?

In order to be able to query for user data, it has to be made sure that the user has logged in first.

Purpose	Returns the list of user's id and user's username for the currently logged user
Type	query
URI	content://com.ait.userprofile.AITUserProfileProvider/isUserLoggedIn
Returns	1. A Cursor with one row of 'id', 'username' which represent the user's id, and the user's username for the logged in user 2. <i>null</i> if the user is not logged in

```
//check if the user is logged in
Uri uri =
Uri.parse("content://com.ait.userprofile.AITUserProfileProvider/isUserLoggedIn");

String[] projection = new String[] { "id", "username" };

Cursor userPreferences = getContentResolver().query(uri, projection, null, null, null);
if (userPreferences!=null){
    if (userPreferences.getCount()>0){
        int indexId = userPreferences.getColumnIndex("id");
        int indexName = userPreferences.getColumnIndex("username");
        userPreferences.moveToFirst();
        do { //actually there is only one row in cursor
            //you can get now and use the user's id and user's username
            String id = userPreferences.getString(indexId);
            String name = userPreferences.getString(indexName);
        } while (userPreferences.moveToNext());
    }
}
else {
    //user is not logged in;
}
```

9.1.2 QUERY ALL USER DATA

After having registered with the MApp service, the user has the possibility to specify additional data such as home town, years lived in the country, hobbies, etc. These data could be of significance for a third party application and can be queried as such:

Purpose	Returns a list of (id, name, value) tuples of all fields for the currently logged user
Type	query
URI	content://com.ait.userprofile.AITUserProfileProvider/allFields
Returns	1. A Cursor with a list of 'id', 'name', 'value', 'userEditable' which represent <ul style="list-style-type: none"> • 'id': the field's id, • 'name': the field's name, • 'value': user's selected value. This is either the actual value (for

	<p>integers and free text) or a JSONObject that always has a 'value' which is what you use to display, and an 'id' to use if you need to update the user's selected 'value' in his profile,</p> <ul style="list-style-type: none"> • 'userEditable': zero (0) if the field is not user editable and one (1) if the field is user editable. <p>2. <i>null</i> if the user is not logged in; you must call the AITUserProfileProvider as described in sections earlier in this document</p> <p>3. <i>IllegalArgumentException</i> if the URI is not valid</p>
--	--

```
//get all fields in available in user profile
Uri uri = Uri.parse("content:// com.ait.userprofile.AITUserProfileProvider/allFields");

String[] projection = new String[] {"id", "name", "value"};
Cursor userPreferences = getContentResolver().query(uri, projection, null, null, null);
if (userPreferences!=null){
    if (userPreferences.getCount()>0){
        int indexId = userPreferences.getColumnIndex("id");
        int indexName = userPreferences.getColumnIndex("name");
        int indexValue = userPreferences.getColumnIndex("value");

        userPreferences.moveToFirst();
        do {
            //variables that contain the data
            //retrieved and can be used for your purpose
            //id, name, value

            String id = userPreferences.getString(indexId);
            String name = userPreferences.getString(indexName);
            String value = userPreferences.getString(indexValue);
        } while (userPreferences.moveToNext());
    }
    else {
        //user is not logged in; do what you have to do
    }
}
```

9.2 SENDING EVENTS

The MApp service reacts to the user's context by issuing tailored recommendations. Depending on his or her interests, the recommendation engine would inform the user if a museum is nearby or recommend taking language lessons for a specific subject after having used the translation tool, for instance. In order to integrate with the recommendation engine, it is necessary to send events generated by the third party module. The data structure for sending events is structured as follows:

```
String source;
GregorianCalendar timestamp;
HashMap<String, Object> info;
```

Here, source is the corresponding module which generated the event, timestamp represents the time it occurred and the HashMap is a key/value store containing the payload. An example event reacting to a change in the user's current mode of transport would have the following data structure:


```
MaseltovContext. ModeOfTransportation  
("type", <String>), ("confidence", <int>)
```

Here, type denotes the current activity (walking, sitting, cycling, driving) and confidence the confidence level from 0 – 100.

9.3 SENDING CONTENT

The available MApp modules also allow for direct communication through dedicated interfaces. E.g.: In order to post content to the forum, the standard Android intent action (android.content.Intent.ACTION_SEND) can be invoked. Specifications on how to communicate with the various components are outlined in Section 4.

10. SUMMARY AND OUTLOOK

This deliverable is the third and final iteration of D3.3 and gives a detailed overview on the final status of system integration within the MASELTOV project.

The major goal of Task 3.3 Iterative System Integration was to conceptualize and control the seamless integration of all MASELTOV services and tools supplementing each other within one application. Additionally, it is clearly shown which services of MASELTOV are available in different cities and languages. Technical concepts for inter-component communication and concrete interface specification are defined as prerequisite for seamless integration.

Beside specifications this document summarizes the software integration process and shows the technical guidelines for developers which are in charge of the actual system integration.

Clear definitions of the technical integration plan and integration concepts, their technical requirements and responsibilities were defined for the approach of a decentralized development process, which was necessary in the scope of an international project like MASELTOV. Despite the concept of decentralized development it has been a major goal of MASELTOV to form one integrated service with only one user-interface to provide a seamless user experience.

This last deliverable of Task 3.3 has integrated lessons learned during the final implementation phase and includes all technical adoptions made to solve integration issues. At this point in the project MASELTOV all software components have been successfully integrated in the final demonstrator application (MApp).

11. REFERENCES

- Barata, G., Gama, S., Jorge, J., Gon, D., #231, & alves. (2013). *Improving participation and learning with gamification*. Paper presented at the Proceedings of the First International Conference on Gameful Design, Research, and Applications, Toronto, Ontario, Canada.
- Cechanowicz, J., Gutwin, C., Brownell, B., & Goodfellow, L. (2013). *Effects of gamification on participation and data quality in a real-world market research domain*. Paper presented at the Proceedings of the First International Conference on Gameful Design, Research, and Applications, Toronto, Ontario, Canada.
- Debeauvais, T., Nardi, B. A., Lopes, C. V., Yee, N., & Ducheneaut, N. (2012). *10,000 gold for 20 dollars: an exploratory study of World of Warcraft gold buyers*. Paper presented at the Proceedings of the International Conference on the Foundations of Digital Games, Raleigh, North Carolina.
- Deterding, S. (2012). Gamification: designing for motivation. *interactions*, 19(4), 14-17. doi: 10.1145/2212877.2212883
- Ghosh, A., & Kleinberg, J. (2013). *Incentivizing participation in online forums for education*. Paper presented at the Proceedings of the fourteenth ACM conference on Electronic commerce, Philadelphia, Pennsylvania, USA.
- Kappen, D. L., & Nacke, L. E. (2013). *The kaleidoscope of effective gamification: deconstructing gamification in business applications*. Paper presented at the Proceedings of the First International Conference on Gameful Design, Research, and Applications, Toronto, Ontario, Canada.
- Mekler, E. D., Br, F., #252, hlmann, Opwis, K., & Tuch, A. N. (2013). *Do points, levels and leaderboards harm intrinsic motivation?: an empirical analysis of common gamification elements*. Paper presented at the Proceedings of the First International Conference on Gameful Design, Research, and Applications, Toronto, Ontario, Canada.
- Montola, M., Nummenmaa, T., Andr, #233, Lucero, s., Boberg, M., & Korhonen, H. (2009). *Applying game achievement systems to enhance user experience in a photo sharing service*. Paper presented at the Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era, Tampere, Finland.
- Rojas, D., Kapralos, B., & Dubrowski, A. (2013). *The missing piece in the gamification puzzle*. Paper presented at the Proceedings of the First International Conference on Gameful Design, Research, and Applications, Toronto, Ontario, Canada.