



VOIce-based Community-cEntric mobile Services for social development

Grant Agreement Number 269954

Deliverable No 1.7
High Level Architecture – cycle 2

PROJECT DELIVERABLE REPORT

Project	
Grant Agreement number	2699542
Project acronym:	VOICES
Project title:	VOIce-based Community-cEntric mobile Services for social development
Funding Scheme:	Collaborative Project
Date of latest version of Annex I against which the assessment will be made:	
Document	
Deliverable number:	D1.7
Deliverable title	Final Report High Level Architecture
Contractual Date of Delivery:	06 2013
Actual Date of Delivery:	06 2013
Editor(s):	Bineke Posthumus (TNO)
Author(s):	David Blaisonneau (FT), Morgan Richomme (FT), Bineke Posthumus (TNO), Sander van der Aa (TNO)
Contributor(s):	Filipe Cabral Pinto (PTIN), Stephane Boyera (WF)
Reviewer(s):	PTIN
Work package no.:	WP1
Work package title:	Sustainable architecture
Work package leader:	TNO
Work package participants:	WebF, FT, VUA, CRS4, ESMT, SE, PTIN
Distribution:	
Version/Revision:	1.8
Draft/Final:	Final
Total number of pages (including cover):	58
Keywords:	High level sustainable architecture, open source, business models

DISCLAIMER

This document contains description of the VOICES project work and findings.

The authors of this document have taken any and all available measures in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any responsibility for actions that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the VOICES consortium and can in no way be taken to reflect the views of the European Union. The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 27 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (<http://europa.eu.int/>)



VOICES is a project funded in part by the European Union.

CHANGE LOG

Version	Date	Author	Description of Changes
0.2	29/07/2011	David Blaisonneau	First draft
0.3	03/08/2011	Filipe Cabral Pinto	Corrections
0.4	02/09/2011	David Blaisonneau	Add summary. Correction on WP5 requirements. Correction and adds in 'Technical needs description'
0.5	12/09/2011	David Blaisonneau	Add TTS for WP4 use cases
0.6	15/09/2011	David Blaisonneau	Add Crowd sourcing for WP3
0.7	20/09/2011	Etienne Barnard	First review
1.0	29/09/2011	David Blaisonneau	Add Performance and ASR openness recommendations. Add conclusion. Final review
1.1	31/03/2012	Morgan Richomme	Full rework. Add new chapter on sustainability and business model.
1.2	15/06/2012	Sander van der Aa	Add business models
1.3	29/06/2012	Etienne Barnard & Maurizio Agelli	Final review
1.4	01/05/2013	Bineke Posthumus	Update to final report (add open business models)
1.5	07/05/2013	Morgan Richomme	Update TTS and ASR section based on new microservices developed by WP3
1.6	22/05/2013	Bineke Posthumus	Final Adjustments
1.7	27/05/2013	Filipe Cabral Pinto	Final issues
1.8	31/05/2013	Bineke Posthumus	Last edits overall report and specifically business models section

TABLE OF CONTENTS

PROJECT DELIVERABLE REPORT	2
Disclaimer	3
Change Log	4
Table of contents	5
Table of figure	7
1 Summary	8
2 General description	10
2.1 Services stakes of the VOICES project	10
2.2 Platform functions	11
2.2.1 Multi-channel accessibility	11
2.2.2 Voice tools	12
2.2.3 Service hosting and execution	12
2.3 User characteristics	13
2.4 Constraints	15
2.5 External dependencies	16
3 Platform description	17
3.1 High level architecture	17
3.1.1 Flexibility	17
3.1.2 Reachability	18
3.1.3 Accessibility	18
3.1.4 Openness and Re-usability	18
3.1.5 Long-term sustainability	18
3.1.6 Details of the service feature from other WP	18
3.2 The Emerginov Choice	19
3.3 Rationalization of the service life cycle	19
3.4 Sustainable service development	20
3.5 Specificity linked to the context of emerging countries	21
3.6 The Choice of open source components	21
3.6.1 Selection criteria	22
3.6.2 Description of the components	23
3.6.3 Licensing	24
3.6.4 A story of integration	25
3.7 Platform capacities	26
4 Platform evolution	27
4.1 Roadmap	27
4.2 Towards community management	28
4.3 Standalone version	28
4.4 Telco advanced version	29
4.5 Software update	29
4.6 Costs	29

4.7	Installation	32
4.8	Maintenance	32
4.9	Feature extensibility	33
4.10	New components	33
5	Business model	35
5.1	Introduction	35
5.2	Business models	35
5.3	Open source Business models	36
5.4	Business model for the Emerginov-platform	43
5.5	General conclusions	48
6	Conclusions	49
	References	50
	Appendix A: Pilots	51
	Appendix B: Voice tools	57
	Appendix C: Technical requirements list	58

TABLE OF FIGURE

Figure 1: VOICES platform-service structure	10
Figure 2: Actors in VOICES	14
Figure 3: High level architecture	17
Figure 4: Development Cycle	20
Figure 5: Open source solutions	23
Figure 6: Deployment scripts	26
Figure 7: Emerginov community roadmap	27
Figure 8: Cost distribution overview (advanced version)	30
Figure 9: Business model canvas	36
Figure 10: Integration and support – Business model	38
Figure 11: Premium Plugins & Extensions – Business model	39
Figure 12: Freemium & Premium – Business model	40
Figure 13: Software free, Hardware Paid – Business model	41
Figure 14: Advertisement – Business model	42
Figure 15: Emerginov platform – business model	43
Figure 16: Visualisation of open source beneficiaries and corresponding values	44
Figure 17: Emerginov customers	46
Figure 18: Emerginov customers and open source business models	47
Figure 19: WP4 – Use case 1	52
Figure 20: WP4 – Use case 2	53
Figure 21: WP4 – Use case 3	54
Figure 22: WP5 – Use case 1	55
Figure 23: WP3 crowd sourcing tool	57

1 SUMMARY

Purpose

The report presents the VOICES high level architecture based on requirements elaborated in the task 1.1-Use cases and requirements. This architecture is built to be flexible and dedicated to emerging regions. The intended audience are all interested stakeholders for voice-based services.

Overview

This document will firstly provide a general description of the project. It will indicate the required associated high level features and identify the users of the solution.

The second chapter describes the platform. It details the high level architecture and justifies the choice of the Emerginov platform as the service development and testing framework for the VOICES project. The full technical description is available in deliverable D2.1 [2].

The third chapter explains how the platform evolved, details the costs related to the use, the maintenance and the exploitation of the service. This chapter deals also with the possibility of feature evolution and draws a high level roadmap.

The last chapter is dedicated to the possible business models of the Emerginov platform.

Definitions and acronyms

API	Application Programming Interface
ASR	Automatic Speech recognition
BoP	Base of the Pyramid
BSD	Berkeley Software Distribution
CMS	Content Management System
CSR	Corporate Social Responsibility
CTIC	Information Technology and Communication Council
DAM	Digital Asset Management
DB	Database
GNU GPL	GNU is Not Unix General Public License
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HTK	Hidden Markov Model Toolkit
HTTP	Hypertext Transfer Protocol
ICT	Information and Communications Technologies
IT	Information technology

IP	Internet protocol
IVR	Interactive Voice Responder
LAMP	Linux Apache MySQL PHP
LGPL	Lesser General Public License
MIS	Management Information System
MIT	Massachusetts Institute of Technology
MMS	Multimedia Messaging Service
NGO	Non-Governmental Organization
PaaS	Platform as a Service
PHP	Personal Home Page
PLS	Pronunciation Lexicon Specification
PSTN	Public Switched Telecommunication Network
RNL	Réseau National des Laboratoires
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SME	Small & Medium sized Enterprise
SMS	Short Message Service
SOHO	Small Office Home Office
SSML	Speech Synthesis Mark-up Language
TTS	text To Speech
USB	universal Serial Bus
USSD	Unstructured Supplementary Service Data
VM	Virtual machine
VoiceXML	Voice eXtended Mark-up Language
VoIP	Voice over IP
WP	Work Package

2 GENERAL DESCRIPTION

2.1 Services stakes of the VOICES project

The VOICES project aims to leverage the potential of voice based services in African countries. Leveraging of content as well as lifting access barriers are two central challenges. Within the eight Working Packages (WPs) the development of viable and sustainable voice-based services is central. The separate WPs deal with: required technologies (WP2 and WP3), pilots (WP4 and WP5), a training lab (WP6) dissemination, exploitation and community building (WP7) and the management of the overall project (WP8). WP1 aims to ensure coherency and sustainability of the VOICES project. This is achieved through regular monitoring of the requirements derived from the various WPs, the development of suitable business models as well as the design of a flexible architecture to guide the overall project.

Notions central to the VOICES project are ‘accessibility’ and ‘sustainability’ of services. To achieve this, all the services and products developed should be replicable to a certain extent. This entails that the generic ICT architecture developed in the VOICES project should be usable in different situations, with different technologies and with support of various business models. The delivered platform can be matched with services which are tested in the health pilot in Senegal (WP4) and the agri pilot in Mali (WP5). Combining the generic architecture with the services that are tested in the pilot studies, results in a full-fledged voice-based service chain that incorporates the technological basis as well as the non-technical value of context appropriate services. Figure 1 is a visualization of the VOICES platform-service structure.

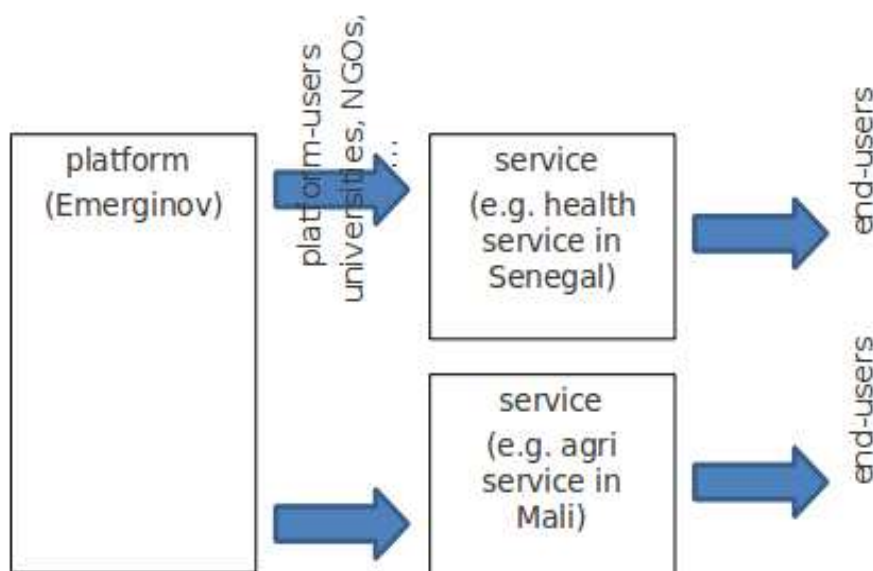


Figure 1: VOICES platform-service structure

The separate building blocks can be exchanged for technologies or services with similar functionality. As a result the full structure can be used in different contexts. This directly links to the BoP specific necessity of services being scalable.

2.2 Platform functions

From a service point of view the VOICES platform must include the following features:

- Multi-channel accessibility (SMS, voice)
- Voice tools
- Service hosting and execution

One of the main challenges lies in providing sustainable services within 3 years while several partners with different backgrounds are involved. It means that we must support the pilots but also have in mind that the services shall be runnable beyond the project. Therefore the infrastructure supporting the services must be reproducible out of the project context.

In order to ensure sustainability beyond the project deadline as well as support from local entrepreneurs, a Mobile Training Lab is set up within WP6. Within this training lab, local (Senegalese) entrepreneurs are trained to develop sustainable voice services using the Emerginov platform. In order to do so, the entrepreneurs are trained in programming, but also in business skills. The sustainability of the training centre itself is ensured by developing a viable business model that makes the centre able to be self-sufficient (or at least, not depending on any resource coming from the Voices project).

The detailed functions per pilot are described in chapter 0.

2.2.1 Multi-channel accessibility

Data or services can be available through different media channels (in both ways, from or to the platform):

- ⤴ fixed phone call
- ⤴ mobile call
- ⤴ SMS
- ⤴ Internet
- ⤴ Radio

The channels must furthermore be available in these countries:

- ⤴ Senegal
- ⤴ Mali

The vocal channels shall be available in Bambara (Mali), Wolof (Senegal) and French.

2.2.2 Voice tools

IVR

An Interactive Voice Responder function must be available for each service.

The IVR function can be scripted in several ways: VoiceXML, PHP Scripts or Asterisk language. The use of VoiceXML is recommended.

ASR

An Automatic Speech Recognition function shall be available for services hosted by the platform. The language packages shall be updated during the project.

The openness recommendation applies here for both the ASR engine and its compliance with open integration of new languages. For this purpose, de-facto open standard - such as HTK acoustic models - shall be used.

Please note that the WP3 developed an ASR solution for Bambara called VOICES_asr. This solution has been successfully integrated as a micro-service in Emerginov. The solution can be found at http://projects.emerginov.org/VOICES_asr/ and the source code can be retrieved at https://svn.emerginov.org/listing.php?repname=VOICES_asr.

This project required the installation of the open source solution Julius on the virtual machine Dev even if it was already present on the virtual machine Media as well as specific authorization to allow the service to invoke the binary. As any Emerginov micro service, this solution has been published under a free license.

TTS

A Text To Speech (TTS) function shall be available for services hosted by the platform. This function must support different languages to fit the use cases needs. Compliance with standards such as Speech Synthesis Mark-up Language (SSML) and Pronunciation Lexicon Specification (PLS) will be considered, with appropriate standards being chosen based on likely developer requirements.

Please note that the WP3 developed a specific application for TTS in Bambara and Bomu called Voices_snftts. The application has been developed as a micro service (idem Pilot WP4 and WP5) and can be tested at http://projects.emerginov.org/voices_snftts/. The source code can be retrieved at: https://svn.emerginov.org/listing.php?repname=voices_snftts. This micro service requires some specific authorization to allow the web page to invoke a binary. This new TTS function can be used by any web application. As any Emerginov micro service, this solution has been published under a free license.

2.2.3 Service hosting and execution

We must clearly distinguish between the platform and the service/hosting of the platform.

The platform is an asset of the Voices project and shall be operable outside of an 'Orange environment' to ensure the sustainability of the project (see next chapter).

Orange will provide hosting until at least the end of the Voices project; however there are no guarantees regarding the hosting and the availability of this hosting after the project has ended as there are no guarantees on bug fix on any of the applications that could be developed on top of it.

Service execution

The services must be executable on the platform. The service development language will depend on the components used, but open and well known development languages such as PHP are recommended.

File Hosting

All files needed for the services or created by them shall be hostable by the platform and sharable between the components (in the limit of a fair usage).

Database

A database must be available for each service. A relational database management system such as MySQL is recommended. The usage of a DAM (Digital Asset Management) platform is evaluated on the basis of the amount of digital content to be managed.

2.3 User characteristics

Ecosystem of users

Depending on the version of the platform we can imagine several potential users.

Version	Potential users
Standalone	Universities, NGO, research entities
Advanced	Operators, web hosts, IT and business incubator, institutional groups, NGO with IT division, development agencies

The platform could be, for the different users

- a simplification of the Telcoweb architecture
- an autonomous solution not linked to industrial products
- a way to rationalise the creation of micro projects
- a solution to keep the history of the different projects and expose them (reuse rather than redo)
- a single point of management of code and contents
- a coherent management through open source licences for code and creative commons licences for content

For the platform to be used successfully, it should anticipate certain needs that are distinctive to emerging countries. A poor or basic infrastructure, for example, may create the need for simple systems that nevertheless constitute a useful platform. Power outages should for example not hamper nor harm the platform and its services. Related to this is the ease of maintenance; for users this could be an important factor as failing services may be worse than no service at all. Another important feature of emerging countries is the diversity in population, at times the large diversity that exists tends to be overlooked. It is because of this diversity that the flexibility of the VOICES architecture is such a strong feature, a feature which needs to be safeguarded.

Users within VOICES

Within the context of the VOICES project two services are developed and tested in the pilots in Senegal and Mali (WP4 and WP5). The first facilitates knowledge exchange in the health sector while the second facilitates contact with farmers in the agri sector. In both of these pilots different partners are involved showing the wide applicability of the voice-based services.

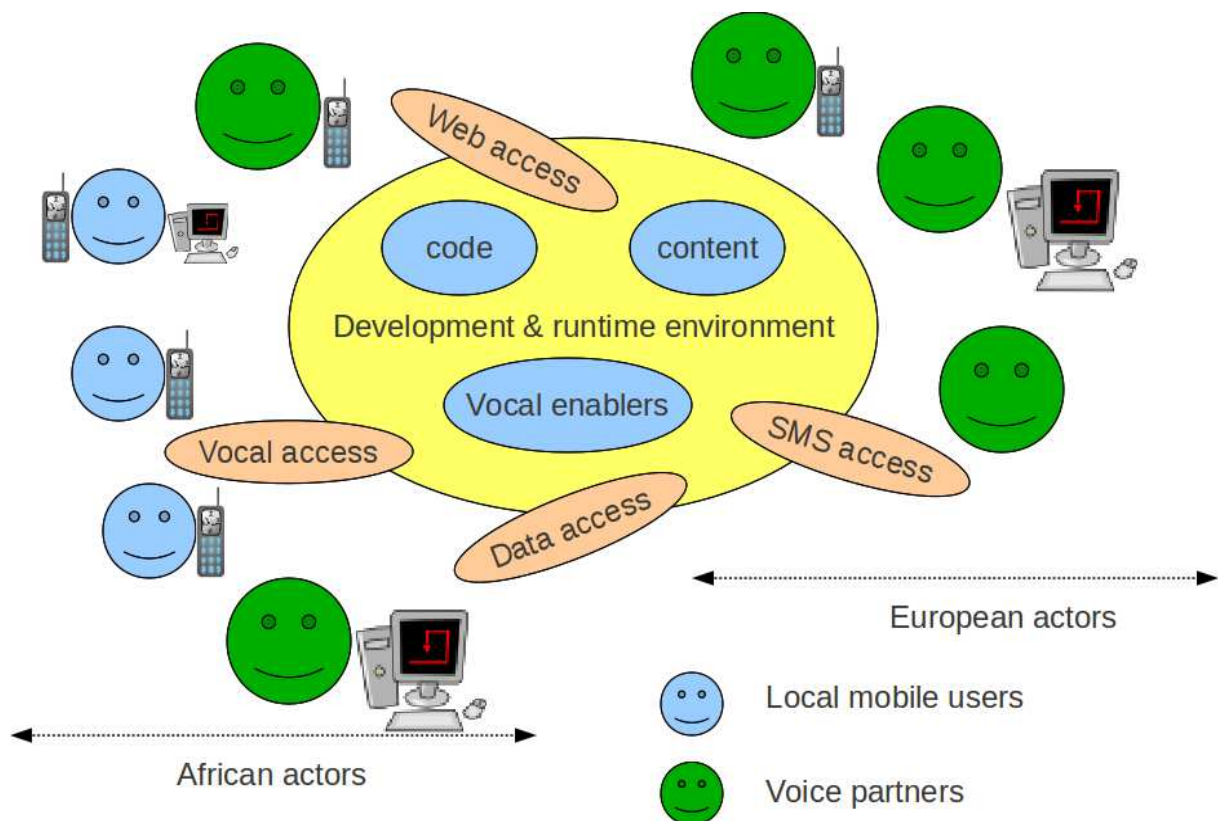


Figure 2: Actors in VOICES

The health pilot in Senegal links peripheral biomedical laboratories with regional and national laboratories. Specifications are described in the deliverable D4.1. The user of the platform is the Réseau National des Laboratoires (RNL) which uses it to reach doctors and

laboratory workers at district level. The platform can be of use in a number of ways. It can function as a network to gather information about the spread of contagious diseases, to analyze the professionals' knowledge or to send out monthly bulletin updates. Each of these services can be built on the platform. In this pilot the user of the platform and the target group of the services built on the platform belong to the same organization namely the RNL. This may complicate the viability of the services' business models as the provider and the user of the service overlap.

In Mali the platform is used by Sahel Eco, a regional NGO. It uses the platform to automate a number of already existing services. The platform is for example used for services to store details of the target group or to send out messages to a large group of farmers. The Radio Marché service has also been automated. Through this service the NGO gives producers the possibility to broadcast their offers on local community radio which increases their chance of finding a buyer. The TTS technology has been an important feature of the automation of this service and has enhanced the reliability of the Radio Marché service. The service does not cover its own costs though, which entails that it is not sustainable in its current form. For this to change the implementation of a suitable business model is required. Introducing payments for the end-consumer of the service (the farmer) could be part of such a business model but among these end-users the willingness to pay may be too low. Another question is how the service can be scaled up as a radio broadcast only allows for a limited number of messages. Such considerations affect the future development of the platform use and the services built on top of it. How the user, in this case Sahel Eco, envisages the future use of the platform is an important factor in the further development of the platform.

Other potential users

One functional aspect of the high level architecture is to provide open access to the sources and documentation of services. This will create the opportunity for platform users to share their services and learn from others. The two VOICES pilots show how the platform can be used in different sectors and by different actors. Other areas in which the platform can serve as a basis for the development of voice-based services are universities, institutional entities or SOHO (small office/home office). For universities the platform can be used to facilitate administrative contacts with students and faculty or services can be built to share knowledge. In a similar fashion institutional entities can use the platform for both internal and external services. The flexibility of the platform allows to arrange procedures within the institutions but also to reach out to the wider public. For SOHOs the accessibility of the platform could be the biggest benefit as they may not be able to afford large investments in complicated ICT services.

2.4 Constraints

One of the main challenges consists in providing sustainable services within 3 years. The platform shall be easy to use for the developers of the pilots and easy to install to ensure the sustainability beyond the project.

Another constraint deals with vocal services. The industrial enablers for vernacular languages are limited. The creation of a voice for the TTS function or the acoustic models and lexicons for speech recognition is expensive. Home-made solutions have been used.

2.5 External dependencies

The architecture has been designed to limit external dependencies as much as possible.

The choice of the platform and more generally of open source solutions lead to a form of independence towards industrial solutions (e.g. Apache web server versus Microsoft ISS) and privilege open standards versus proprietary protocols (e.g. use of SIP instead of Skype).

It is however not possible to get rid of last-mile dependencies. The main external dependencies are therefore linked to the local operators providing the Telecommunication terminations: Orange Mali in Mali and Sonatel in Senegal. Please note that these companies are part of the Orange group but are legally totally independent from Orange Labs. It can explain the delay in the delivery of the Telecom access even if the core platform has been available since May 2011.

We may also mention a dependency towards the terminals. Some applications have been designed for Android devices but all the device aspects are out of scope of the back end platform.

3 PLATFORM DESCRIPTION

3.1 High level architecture

The architecture of the VOICES platform is presented in **Figure 3**:

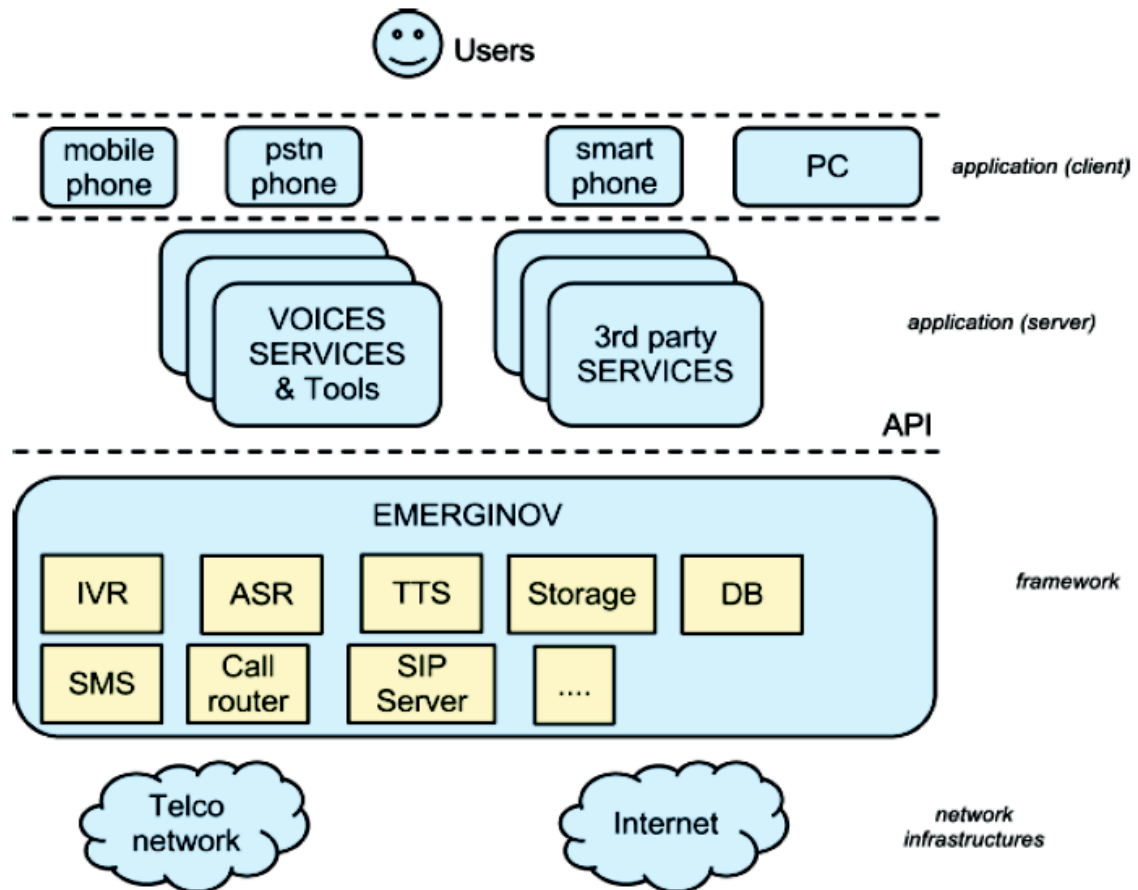


Figure 3: High level architecture

This high level architecture designed by Orange will be deployed in Orange premises in France, Senegal and Mali.

This architecture is flexible, reachable, accessible, open, re-usable. It shall guarantee the sustainability of the services developed on top of it.

3.1.1 Flexibility

The VOICES platform entails that each component can be replaced by another component with the same functionalities. This implies that all components must target open and documented interfaces. This also implies that the services developers can use third party services such as Google or Facebook APIs for instance through internet or another telecommunication channel (use of a modem instead of a Telecommunication gateway).

3.1.2 Reachability

The VOICES platform provides multiple accesses to reach the services hosted on it. In the VOICES context, the targeted public is mainly composed of mobile phones users, radio listeners and most of them share internet access when it is available.

3.1.3 Accessibility

The VOICES platform provides components to enhanced service accessibility to reach illiterate people and people who only speak local (marginal) languages.

3.1.4 Openness and Re-usability

To be used after VOICES project or by other persons, the VOICES platform is built, as much as possible, on open components, free software, open interfaces and open data structures. The VOICES platform shall also provide an access to all services sources and documentation.

3.1.5 Long-term sustainability

The architecture does not prevent the local adoption and exploitation of the VOICES tools and methods beyond the project lifetime.

The complete technical architecture of the platform is described in D2.1 [2].

We may just indicate the 2 main components:

- core platform: the integration of all the applicative component on a virtualized environment
- gateways: the Telecom entry point allowing vocal and SMS services

3.1.6 Details of the service feature from other WP

As mentioned in the first chapter, the service features required from the different WP are similar:

- SMS accessibility
- Vocal accessibility
- Internet accessibility
- Web hosting
- Database
- Interactive vocal services
- Text To Speech
- Vocal recognition

The features are detailed by WP in the appendix section

We may also mention a need to organize the developments

- code management
- deployments mechanism (from the code to the lived network)

3.2 The Emerginov Choice

It was decided to use the Emerginov platform to support the different services developed within the VOICES project.

The architecture will be described in chapter 4 and more precisely in D2.1 [2]. This chapter aims to clarify the reasons behind the choices made in the design this platform.

This platform has been designed by Orange Labs to allow rapid prototyping in emerging countries since beginning 2010 [6]. The three main goals were:

- To provide an IP multimedia infrastructure for emerging countries
- To animate an ecosystem of co-innovation partners
- To build a library of business applications under free licenses

The idea of the platform is to allow local development of services by local people for local people. It was imagined to be a bridge between local mobile contexts and the world wide web.

Most of the web functions or access to APIs were available in richer IT environment, but the general approach – to combine Telecom and Web technology - was innovative, applicable and fitting in the African context.

Thanks to the Emerginov solution we are able to offer a local framework to quickly develop services on top of the networks of African operators, despite the fact that cloud solutions are not focused on Africa. Please note that the platform has not been “developed from scratch”. The platform consists of smartly integrated open source components. The development thus hinges in a sense on the ‘glue’ that is required to integrate the different components. The idea was to use mature components as much as possible. The open source approach was also chosen as a way to simplify the service environment creation and a way to involve local actors on both the service creation side as on the back end side.

Developers were able to find support because the enablers they used were open source.

People in charge of the platform were also independent from industrial vendors, no licences were required. They had to source some operations such as installation, supervision or maintenance. At the end they were able to fully manage the whole solution.

3.3 Rationalization of the service life cycle

In collaborative projects, we have to deal with the history and the background of each partner, especially on the development side. The infrastructure, as an open ‘PHP Platform As A Service’ (Paas), provides a simple and open framework that allows any partner to contribute in a collaborative way without needing to go through big cultural changes.

The life cycle of all the services of all the pilots may be summarized as follows:

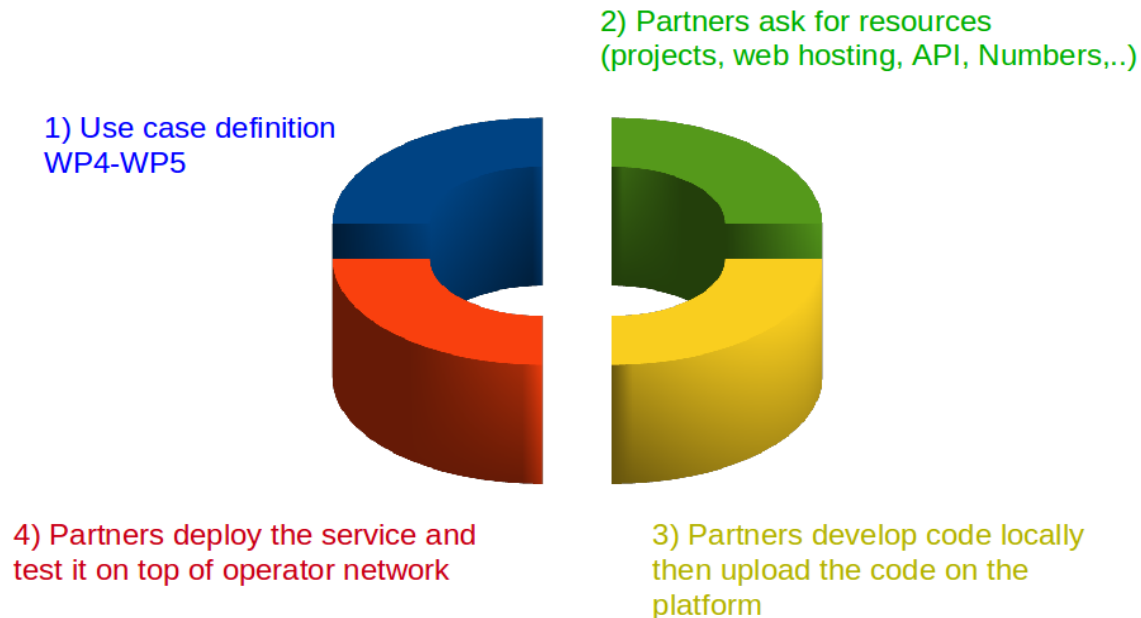


Figure 4: Development Cycle

The different steps can be managed through a web interface. The developers can thus ask for project creation, project participation, resource allocation and deployment of the services on the public networks (internet and PSTN). Most of the operations have been automated and can be used through the user web pages.

3.4 Sustainable service development

The platform must be sustainable which entails that the different services shall be operable anywhere anytime. The services must be operable beyond the life of the VOICES projects. This means that the code must be reachable and the runtime environment must be reproducible.

Regarding the platform, by construction, the platform is open source and any components can be installed anywhere, anytime. The minor non open source elements have been modified/changed to be fully open source. And, if in the case of Voices we used proprietary APIs, we provided a degraded mode based on open source components. The whole platform with the 'glue' developed by Orange Labs will also be published under free licence. This procedure will be detailed in the next sections. It means concretely than that after the project has ended, people could install their own runnable environment. The platform and the installation procedures will be provided.

3.5 Specificity linked to the context of emerging countries

The architecture originally took into account the context characteristics of emerging countries:

- High mobile penetration rate
- Low literacy rate
- Unstable data connectivity

The low literacy rate led to the integration of Voices tools in order to provide vocal services. The features were initially limited but have been extended within the context of VOICES as explained in D2.1 [2].

The lack of sufficient data connectivity for local access and/or internet connection outside the capitals leads to a design that could be exported locally. From the beginning, it was planned to install the solution close to the users. The work in rural areas was not compatible with nice cloud solutions due to lack of bandwidth. With respect to access, only GSM was furthermore reliable enough to provide service in remote areas. Data access was possible, in GPRS, Edge or 3G but not everywhere. Moreover the (data) costs for end users were not negligible for field tests. Even in 3G covered areas, it has been noticed that data consumption remains weak. Several explanations can be suggested:

- no cultural aspiration for data services
- no value added services (no African services)
- low literacy rate
- cost of the data

Moreover most of the users have prepaid GSM account where data is an option and feature phones that do not all support 3G.

Modem based solutions were also possible but such solutions would entail that the scalability is reduced. The Emerginov solution allows field trials with vocal capabilities up to 20 simultaneous calls, which will be too complex to manage with modems. Moreover it is directly connected to the production network of the operator and should in theory provide a higher quality of service in term of sound and availability. Finally the complexity of modem management (vocal, SMS, hybrid, 1 or several SIMs, power supply) is in the end quiet big compared to a network overlay configuration.

3.6 The Choice of open source components

It must be clearly stated that the platform is a smart integration of free software/ open source components. We reused massively open source components. The developments were limited.

The main work has consisted in the selection of these mature components and the 'glue' to connect them.

By nature all the open source components are available under the term of their own licence (BSD, MIT, Apache, GNU GPL,...). The glue and the specific parts developed by Orange shall also be published under free licence.

Thus the full open source version, including the installation guide, shall allow the re-creation of a runnable platform out of Orange runnable environment.

Please note that the design of the platform has been influenced by the result of field trials. The cloud solutions are not acceptable yet due to a lack of bandwidth especially in rural areas and the lack of stability of the internet connections. The idea was to be able to deploy a micro service incubator close to the target users to reduce network issues at the access level but also at internet interconnection level. Another main goal consisted in involving local people on the maintenance/support of the solution and on the developments.

Therefore, the choice of an open source platform to be deployed locally is one of the main arguments for sustainability.

3.6.1 Selection criteria

As mentioned in D2.1 [2], 90% of Emerginov is based on free software/open source components. At the beginning of the VOICES project, the non-open source components were located in the Orange API shop called nursery and in the virtualization layer based on XEN Citrix, an evolution of the full open source XEN solution. XEN Citrix has been released under a free license during the project and Orange Labs has worked on the API shop in order to replace the non-open source elements by open source ones.

It means that Orange did not develop any of the components from scratch, but used existing mature components to provide the support platform. The major tasks were:

- selection of the components
- compliance with requested features
- maturity
- facility for integration
- security
- integration of the components
- development of missing mechanisms to simplify the developer path.

The maturity of an open source component is not easy to evaluate, we considered the following criteria:

- date of first release
- date of last stable release
- number of lines
- number of code committers (coders contributing to the solution code)

- activity of the mailing lists
- availability of documentation

3.6.2 Description of the components

The open source ecosystem is very wide. The following picture depicts some of the most famous solutions. But new projects are created every day and projects are also dying every day. A Darwinian selection leads to mature products.



Figure 5: Open source solutions

The selection based on the criteria described in the previous chapter lead to the selection of the following main components:

Feature	Existing components	Selected Component	Link
Virtualisation	XEN, KVM, OpenVZ	XEN	http://www.xen.org
Operating System	Debian, Ubuntu, CentOS,	Debian	http://www.debian.org/
SMS gateway	Kannel, playSMS, frontlineSMS	Kannel	http://www.kannel.org/
Voice gateway	Asterisk, Freeswitch	Asterisk	http://www.asterisk.org/
Web server	Apache, Nginx, Tornado	Apache	http://apache.org/
Database	MySQL, MariaDB, Drizzle, PostgreSQL	MySQL	http://www.mysql.org
Interactive vocal server	Asterisk, freeswitch, Mobicents	Asterisk	http://www.asterisk.org/
Vocal recognition	Julius, CMU Sphinx	Julius	http://julius.sourceforge.jp/en_index.php Specific VOICES development (WP3)
Text-To-Speech	Festival, Flite, MARY,..	Festival	http://www.cstr.ed.ac.uk/projects/festival/ Specific VOICES development (WP3)
Code management	SVN, Redmine, Maven	SVN	http://subversion.apache.org/

See D2.1 [2] for further details, including the version and the license type.

3.6.3 Licensing

Emerginov, as a platform based on the integration of lots of sub components, will not have one single license that would cover all the licenses of all its components.

The different components have their own free license (Apache, GNU PGL, LGPL, MIT, BSD,...) but as they are independent there is no compatibility issues, we just must keep the existing licenses and comply with the different license terms.

As mentioned we integrated the components but we did not develop them, therefore there is also no issue regarding the redistribution of modifications.

The “glue” developed by Orange Labs (see next sections) will be also published under the term of a free license so the whole solution will be covered by free licenses.

The services developed on top of the solution must select a free license prior to any deployment on the production network. By default, we suggest Apache 2.0 or GNU GPL v2. If for any reason (compatibility, regulation) another free license is required, it can be managed case by case.

3.6.4 A story of integration

The open source components described in the previous section are not natively integrated together. You can install any of them independently.

The Emerginov platform provides a global framework where all these elements are put together.

Portals

First of all, 2 portals have been created:

- a user portal
- an admin portal

These 2 portals provide a single point of access for the user or the administrator of the platform.

On the user portal, you will find links to

- the user self-care page, allowing the creation of projects and the access to the resources.
- The developer wiki with lots of sample codes
- the integrated social network
- the digital library
- the list of components

Scripts

Orange Labs also developed scripts to ensure the links between the different components.

Scripts are used to transfer the code from a repository area to the runtime area. The operations can be performed by clicking on the user self-care. Mechanisms of auto deployment have been added to simplify the update of the service.

The mechanism of deployment can be summarized as visualised in figure 6:

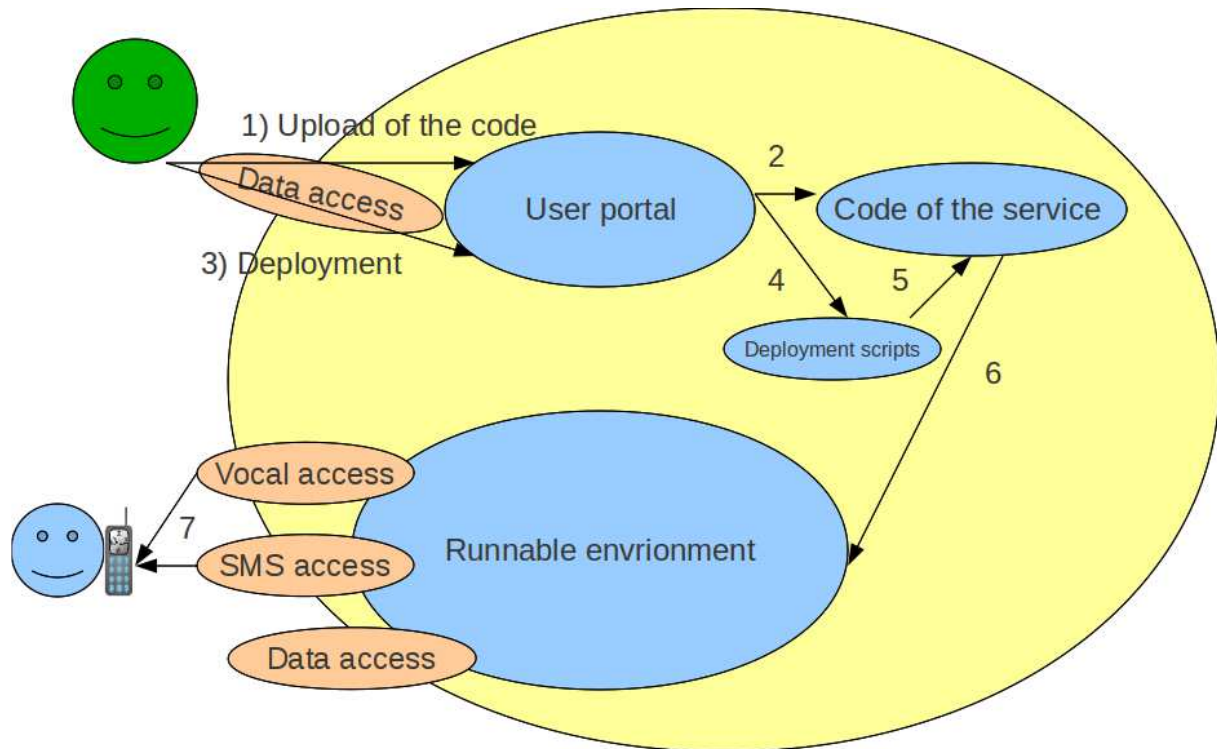


Figure 6: Deployment scripts

3.7 Platform capacities

The platform shall have the following capacities:

- 1 local phone number per use case
- 1 SMS short code per country (with as many keywords as use case)
- SIP lines (for development purpose)
- Manage concurrent calls (> 5)
- Telecom capacities may be limited by quotas (to avoid unwanted usage, not to limit the normal usage of the service)
- Web hosting capabilities may not be limited (in regard to a fair usage)

The dimensioning elements are the gateways with up to 30 simultaneous calls per gateway on vocal side and possible rate shaping on SMS side to avoid SMS spamming.

4 PLATFORM EVOLUTION

The platform - which means the whole framework allowing the development of services - must be published before the end of the project in order to allow the instantiation of runnable environment supporting the projects created within the project.

The services are deployed under a free license (prerequisite prior to deployment) and can therefore be directly imported/exported in any framework. The term platform deals only with the framework in this chapter.

Concretely it is not because all the components are open source that you can instantiate easily the whole platform. The added value of Orange labs consisted in integrating all these elements together by developing the glue between the different components. At the end the installation of the whole platform shall be as simple as the installation of one single component. That is why the whole platform shall be published in order to allow an easy installation of this complex integration of components.

4.1 Roadmap

The roadmap can be summarized as follow:

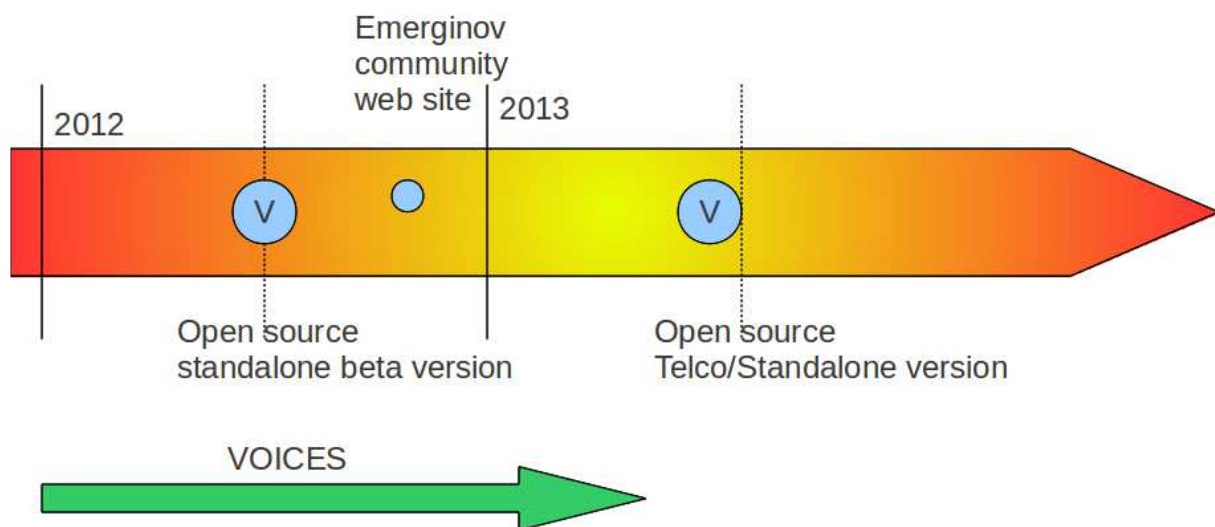


Figure 7: Emerginov community roadmap

The main goal was the publication of an easy-to-install solution. In a first phase Emerginov was a standalone version; currently it is a Telco version. The documentation, the procedures and scripts are available to allow beta tests with partners.

4.2 Towards community management

The creation of a community to support the open source version of the platform is expected. As the code produced is limited, the main goal of the community will consist in providing the platform through the code source and/or packaged version. It will also consist in providing the installation and configuration guide as well as the entry point for any of the used components.

In fact the goal is not to rewrite the existing documents but to clearly identify them. The user guides of all the components are written by the communities. It is useless to rewrite a user guide for MySQL or Apache for instance. However, the link to all these documents shall be clearly indicated.

The community has its own portal and a code repository.

The portal provides;

- news
- documents
- access to source code
- training
- forum
- mailing lists
- Referencing of the applications: indeed any application developed on the platform must select a licence priori to deployment. Then, thanks to the licence, anyone can get the code, modify and distribute it. One of the ambitions is to create a forge of business applications under free licence in order to ease massive reuse.

4.3 Standalone version

This version will be compact and limited to the core of the solution. It shall be runnable on a single PC.

This version will not include

- the nursery (tool to expose the internal API and control the use of these APIs)
- supervision tools

The API will be called directly without filtering or account management.

It shall be easy to install, easy to manage.

The performance will be limited and the scalability reduced.

It corresponds to the USB key we already delivered to our partners based on Ubuntu operating system. We will add the administration function and implement the service deployment mechanisms. The goal is to provide this version for the first semester 2012.

4.4 Telco advanced version

This version is close to the one we used for the different pilots of the VOICES project. Several virtual machines will be required. Some options will be possible including the nursery.

It will be “telco” grade in term of performance and security but it will be more complex to install and manage.

As an illustration, the standalone version is hosted “all-in-one” on a single machine. It means that in case of severe disk crash, there is a risk to lose content and applications. The advanced version includes several automatic backup mechanisms (backup of the virtual machines, backup of the databases, backup of the code repository). From a security perspective a proxy virtual machine is dedicated to the interface with public networks allowing a better protection than an all-in-one solution. It also includes a supervision tool allowing to anticipate capabilities problem (disk space, connectivity issues, ..).

Concretely a university or a NGO could deal with a standalone version assuming that regular manual backups are performed. An advanced solution would be more suitable for operators, institutional entities with IT capabilities.

4.5 Software update

The Emerginov open source version (standalone or Telco) will package lots of open source components. Each solution has its own version and its roadmap. New versions of the components will be released regularly.

It is not possible to guarantee the upgrade of all the components but any administrator can decide whether an upgrade can be done or not. The migration procedures are usually well described and could be applied to any of the components.

The script and portals will be available through a repository. Therefore an upgrade of these components should be possible via a command line.

4.6 Costs

The infrastructure costs can be summarized as follow:

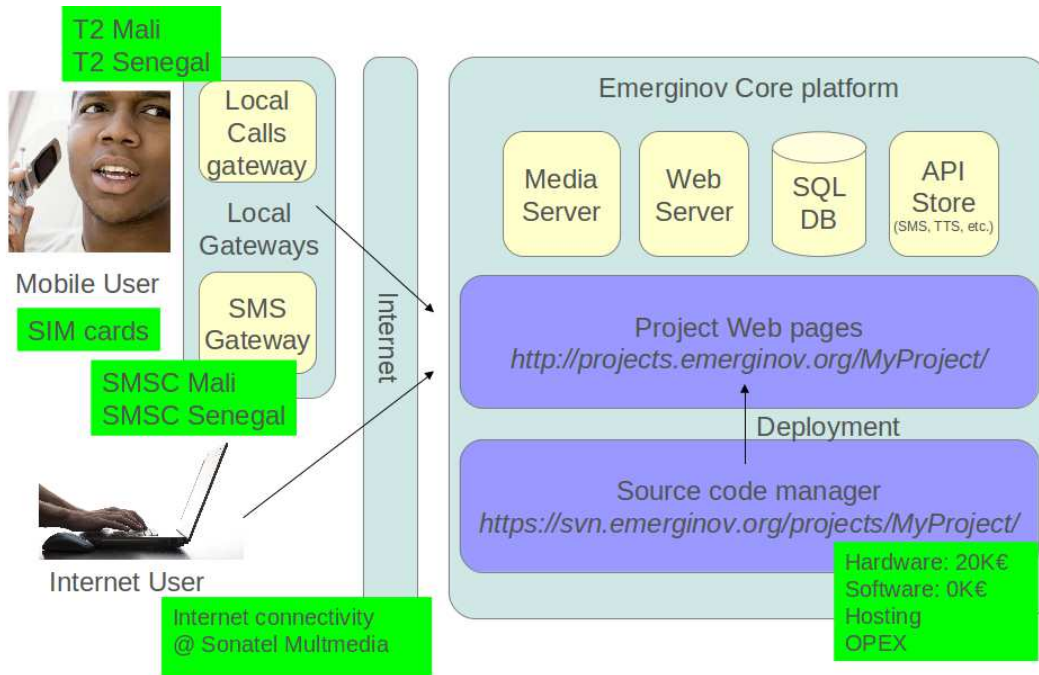


Figure 8: Cost distribution overview (advanced version)

Core platform

The costs are indicative but shall give a rough idea to anyone willing to install its own instance.

Version	Standalone	Advanced
Hardware	500€	20 K€
Software	0€	0€

Gateway

The cost of a single Gateway (1 T2 access + 2 network cards) is about 1K€.

The gateway is well adapted to the advanced version. It is thus possible to install an advanced version with N gateways in N different countries. In the case of the advanced version, the performance bottleneck will be the gateways. It is thus possible to scale by adding capabilities on the gateway side (new gateways or card switch 2,4, 8 T2 connections meaning (~ 40, 80 or 160 simultaneous calls)).

The primary access required for a Telco grade solution can be rent.

The table hereafter gives indicative information of the cost for renting

Country	T2 renting (per month)
France	Build: 640.29 € not including taxes. Monthly fees: 316.30 not including taxes (for 20 B channels Max) ref: OBS offer (2012)
Senegal	Build: 700.000 FCFA not including taxes Monthly: depends on the number of channels but 100.000 FCFA minimum / month Ref: Sonatel (2012)
Mali	For 30 simultaneous channels and 50 numbers (extensible) Build: 975.000 FCFA not including taxes Monthly fee (without traffic): 200.000 FCFA not including taxes Ref: Orange Mali (2012)

A gateway is however not mandatory, in fact it is possible to rely on web/SIP brokers or on modems.

Web/SIP brokers

The platform integrates a SIP proxy, so it is theoretically possible to use a SIP broker to manage voice calls.

However the existing brokers do not commercialize offers in Africa yet. The regulation is often not very clear and the business opportunities are limited.

However we can assume that new destinations will be soon available and could answer to the needs.

Modems

It is possible to connect simple modem as Telecom access points. These modems include one or several SIM cards and can be used for vocal or SMS accesses.

Regarding vocal access, any SIP/PSTN gateway can be used. The SIP adaptation can be done thanks to OpenSIPS. A list of gateway is available on <http://www.voip-info.org/wiki/view/VOIP+GSM+Gateways>.

We can mention:

- Wavecom Fastrack GSM Modem (~ 180€)
- 2N voiceBlue Lite (~ 1K€)

- Dinstar gateways
- Elgato Gateways
- Gempro gateways

Regarding SMS, we can recommend the Huawei E220 (~ 15€). It has been successfully tested as pseudo SMSC with the SMS Kannel gateway.

Other costs

We may mention here the costs of APIs. In fact even if most of the APIs can be freely called in experimentation context, some advanced API may not be free to use.

For example the Text-To-Speech API, we can use the free festival API with free Mbrola voices (no commercial use) but if we want to have specific voices, we may choose another provider (Nuance, Voxygen, Orange Partner...). The cost depends on the volume of the traffic and the specificity of the voices.

Hosting costs (power supply, air conditioning, Space, DNS, Firewalling) are out of scope
The OPEX costs (support and maintenance) are also out of scope of this document.

4.7 Installation

Installation will be possible through a repository. The installation of an Emerginov open source platform should be at the end as easy as: `apt-get install Emerginov` for the standalone version.

As the Telco version is distributed on several machines, the installation will be slightly more complex. We plan to automate as much as possible using open source tools (puppet). For the Telco version, we may also consider the possibility to have installation options (e.g. API shop).

Virtual images could also be released to simplify the installation configuration.

A mechanism of remote configuration based on puppet is also planned.

4.8 Maintenance

The maintenance is under the responsibility of the platform administrators. Support shall be available through the portal or the mailing lists of the community.

4.9 Feature extensibility

The platform has been designed to be extensible. Existing features may be extended; new ones can be created through API or directly on the machine.

Natural extension of the components

Most extensions will be achieved through upgrades of the existing components. Each component follows its roadmap and new functions are included. The upgrade of the component will automatically enrich the platform.

It is for example possible to add new voices based on Mbrola on Festival Text-To-Speech, it is even possible to create your own synthetic voices if you follow the guidelines of these may expect that the Emerginov community could redirect to the appropriate communities communities.

We dedicated to the underlying components for efficient support.

The API nursery

The second element, the nursery, can be used to control the use of your own APIs. It can also be used to make more visible external APIs you want to re-expose to simplify the use or generalize it. Please note that external APIs can be used anyway directly from the developers. The nursery allows exposing non open source element within a full open source environment.

4.10 New components

The separation of the signalling (HTTP or SIP) and the service logics has been taken into account on the conception of the platform.

We also integrated in the telco version a component to expose API. This component called nursery can expose internal but also external API if required.

2 elements can be used to add components on the current architecture:

- the proxy HTTP for web services
- the proxy SIP for vocal services

The proxies

Whatever the version, any user can call any external API. It means that from the developer code it is possible to invoke any API. There is no limitation and the richness of the APIs shall already answer to lots of feature requests.

The HTTP proxy is also used to give only one single HTTP entry point. Any LAMP (Linux Apache MySQL, PHP) application can be installed on the content VM and leads to a customization of your platform (agriculture/health, focus on content, ...). You shall

distinguish the “generic” application that can be used by all the developers (e.g. tracker ([Mantis](#)) , video portal ([Clipbucket](#)), any CMS...) and the service application (e.g. [openEMR](#) for health project, [LimeSurvey](#) for survey,...).

The SIP proxy allows also some flexibility on the call routing. By default, the calls are routed through the proxy to the Asterisk media server but it is possible, through configuration, to configure any SIP enable application server. It is for example possible to plug a Mobicents¹ application server. This rich convergent SIP/HTTP application server provides interesting features based on JSlee technology. Thanks to this application server, it shall be possible to manage USSD, MMS. Please note that Mobicents, thanks to the JSlee technology, offers a nice way to connect the web and mobile worlds, however the application server shall be connected to the operator equipment at the end (USSD gateway, MMSC).

New APIs

Any API can be used and therefore extend the possibility of the platform. As an example, the Google API on speech recognition has been easily integrated. It can be used directly from the applications or be exposed through the nursery for a control of the usage, statistics,...

1 <http://www.mobicents.org/>

5 BUSINESS MODEL

5.1 Introduction

In this chapter we will concentrate on the possible business models of the Emerginov platform. The platform offers an open source solution to different customer groups to develop and apply voice services. In order to be able to offer the Emerginov platform and/or the services on top of the platform after the VOICES project, one should think about a sustainable business model.

This section is organized as follows. Firstly chapter 5.2 will provide a general introduction on business models; consequently chapter 5.3 we will provide an overview of open source business models. In chapter 5.4 the Emerginov business model is discussed. We will end this chapter with a conclusion.

5.2 Business models

Many definitions of the concept of business models exist. Different definitions emphasize different aspects, such as the architecture for a product or service, a description of roles and relations of a company, the way to do business, how a company goes to market, how value is added, how to make a business viable, etc. The definition used in this report captures the main elements of the definitions mentioned above. It defines a business model as:

A description of how a company or a set of companies intend to create and capture value with a product or service. A business model defines the architecture of the product or service, the roles and relations of the company, its customers, partners and suppliers, and the physical, virtual and financial flows between them.

In this report we make use of the business model canvas to visualize the business models (Figure 9). By making use of these canvasses, the available information is structured and presented clearly. As shown in Figure 9, the customer is on the right half side of the model, as well as the channels and relationships that are required for a product/service to reach the customer. Based on what a customer is willing to pay for a service or product, a company can create revenue streams. The business model canvas shows that the three blocks at the right (i.e. customer segment, customer relationships and channels) together result in a revenue stream (which is in its turn a derivative of these three blocks). The revenue streams describe how the business model generates money for the organisation. On the left side, we see the elements that are necessary to make, produce or offer a product or a service. These blocks include the key partners, the key activities and the key resources that are necessary to actually make or offer a product or service for the focal company. The cost structure represents all the costs incurred by key partnerships, activities and resources. The business model canvas' central part contains the most important blocks; value that actually is delivered to the customer (i.e. the value proposition).

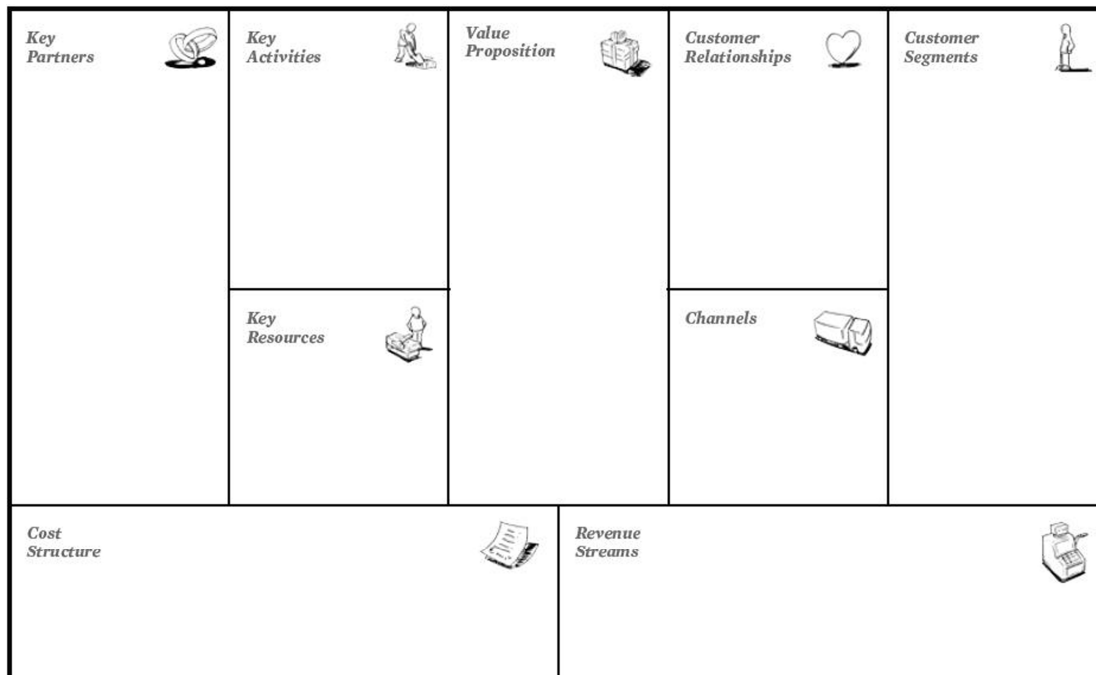


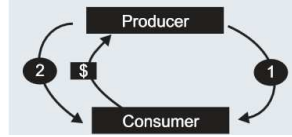
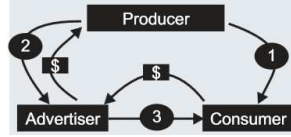
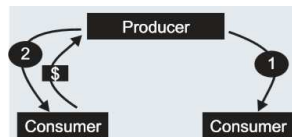
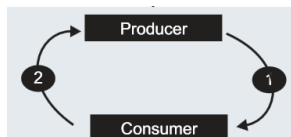
Figure 9: Business model canvas²

5.3 Open source Business models

Open source projects are somewhat different from traditional commercial projects. For example, contributions to open source projects can be numerous and from heterogeneous sources. Furthermore legal issues are managed by a simple but strong license and community management is different. Because of these differences, new business models are used to generate revenues and profits. In the past ten years, many business model variations are developed by different commercial parties. In general there are two ways to earn money from open sources: direct and indirect. The indirect options are e.g.: books, implementation, seminars, support and so on. The direct ways consist of: advertisement in manuals and websites and payment for professional version and so on.

² Osterwalder A, Pigneur Y. (2010), Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Published by John Wiley & Sons, Inc., Hoboken. New Jersey. 2010.

Chris Anderson³ described four basic business models that have a free component.

	
<p>The producer of the open source gives the consumer product 1 for free in the hope this will then entice the consumer to subsequently purchase product 2.</p>	<p>The producer provides product 1 free-of-charge to consumers. Advertisers pay to be included in product 1 in the hope they will be able to sell product 3 to the consumer.</p>
	
<p>A free version (product 1) is made available to anyone who wants it in the hope some users will then choose to upgrade to the paid premium version (product 2) which has more features.</p>	<p>In some markets, no money changes hands at all. People gift or give things to each other without any ulterior motives or expectation of future payment whatsoever.</p>

Five open (source) business models that are widely used are described in this section. Each business model description contains three aspects. First the concept behind the specific business model will be discussed, followed by the most important components of the business model (illustrated by using the Business Model Canvas) and finally an example of a company that use this business model will be shared.

Integration & Support

In this business model the software or product is offered for free to the users. In most cases the product that is offered is very complex. This means that the free software can be used easily for simple environments, but for larger organizations with a complex infrastructure or complex processes it is not so easy to adopt the free product.

In the integration and support business model, the company that offers the open source provides this integration and support for a fee. Their business model is to create high-end software that is too complex for large organizations to integrate without help and support. And who can deliver the support better than the company that created (or has a significant role in the development)? The free software can be seen as a way of marketing.

The business model starts with a dual value proposition. This is the combination of free software (which is available for free) and the support and integration (which is paid and generates the revenues for the company). The customer relationship for the paid integration and support is personal assistance. To deliver the total product, the company focuses on the

³ Anderson, Chris (2009) Free: The Future of a Radical Price

development of high-end software and the knowledge and capacity to support their clients. The open source community is also valuable as they help improve the software.

Key partners	Key activities	Value proposition	Customer relationship	Customer segments
	- software development - providing support - providing integration	- Free software - Paid intergation - Paid support	- personal assistance	
	Key resources		Channels	
	- open source community - personnel - marketing		- community	
Costs		Revenues		
- software development - community		- support fee - integration project		

Figure 10: Integration and support – Business model

Example: Red Hat Linux⁴

Red Hat was founded in 1994 by Bob Young and Marc Ewing with the goal of providing products and services based on open source software. Today, Red Hat is the largest and most recognized company in the open source movement with more than 500 employees worldwide.

The main product Red Hat sells is Red Hat Linux, a software package based on the Linux operating system. The current version, Red Hat Linux 7, is made up of over 800 individual packages, almost all of which have been developed as open source projects. In the spirit of open source development, the full 540MB Red Hat Linux operating system can be downloaded directly from the Red Hat website. As a more convenient solution, Red Hat also offers the Red Hat Linux 7 Standard Edition, which comes packaged with documentation, an installation guide, and 60 days web-based support for about \$30 dollars. Red Hat also sells higher priced versions of the product (the highest is \$2500) targeted at specific platforms or uses, such as servers or workstations.

But the product is only half of what Red Hat markets. Because Linux and the systems it runs are so complex, Red Hat can sell the product itself relatively cheaply and then charge the buyers for the technical support and training required to maintain the system. The levels of technical support vary greatly, from the limited telephone or web-based support that comes with (commercial) package, to 24x7 emergency technical support for large corporations. Red Hat also provides courses to train professionals and developers in operating and using Linux to its fullest extent.

⁴ <http://www-cs-faculty.stanford.edu/~eroberts/cs201/projects/open-source/econ.htm>

Premium Plugins & Extensions

In this business model the basic software is offered for free. This means that users have access to the basic functionalities, but for pro users this is often not enough. Pro-users can buy the functionality to customize the software by buying plugins and extensions for the basic software to meet their needs.

In the premium plugins & extensions business model, the revenues are created by selling the plugins and extensions. In this models there are variations; the developer can choose to keep the source closed and create the extensions and plugins themselves or keep the source open and let third parties develop them and share revenues.

Key partners	Key activities	Value proposition	Customer relationship	Customer segments
	- develop basic software - develop plugins	<u>1: Users of the system</u> - free basic software - paid extensions - paid plugins	- self service	1: users of the system 2: third party developers
	Key resources - developers - support third parties	<u>2: Third party developers</u> - developers kit	Channels - community	
Costs		Revenues		
		- paid extensions and plugins		

Figure 11: Premium Plugins & Extensions – Business model

Example: Joomla

Joomla is a free and open source content management system (CMS) for publishing content on the World Wide Web and intranets. Joomla is written in PHP and stores data in a MySQL or MS SQL database and includes basic features such as page caching, RSS feeds, printable versions of pages, news flashes, blogs, polls, search, and support for language internationalization.

As of March 2012, Joomla has been downloaded over 30 million times. Over 6,000 free and commercial extensions are available from the official Joomla! Extension Directory, and more are available from other sources. It is estimated to be the second most used CMS on the Internet after WordPress.

Freemium & Premium

Freemium or shareware (also termed trialware or demoware) is proprietary software that is provided to users without payment on a trial basis and is often limited by any combination of functionality, availability (it may be functional for a limited time period only), or convenience (the software may present a dialog at startup or during usage, reminding the user to purchase it). Freemium and shareware are often offered as a download from an Internet website. The rationale behind freemium or shareware is to give buyers the

opportunity to use the program and judge its usefulness before purchasing a license for the full version of the software. It is usually offered either with certain features only available after the license is purchased, or as a full version but for a limited trial period of time.

Key partners	Key activities	Value proposition	Customer relationship	Customer segments
	- software development	- free version software		- software users
		- paid full version of software		
	Key resources		Channels	
	- developers		- free repositories	
			- app stores	
Costs		Revenues		
		- paid version of software		

Figure 12: Freemium & Premium – Business model

Example: Open Office

Both StarOffice and OpenOffice offer an office suite of tools compatible with Microsoft Office. Both are based on the same code base. There are a couple differences to TMO and what the customer should know before selecting one or the other. The key differences are in the branding, packaging and support.

StarOffice is pitched more to those who want the indemnification against suit by Microsoft: governments, OEMs and to those who want the ease of having at one go the application plus the most useful extensions and for those who wish to buy, again at the outset, the support, services, training that Sun sells. But support, services, training are also available for OpenOffice.org, and many prefer it over StarOffice because an officiating body has insisted that the license of the application be open. StarOffice is built with open-source software (same code as OpenOffice.org) but the license of the actual application is closed. In one sense, that puts OpenOffice.org a little further out the edge due to community contributions. StarOffice has the advantage of being the more thoroughly qualified product.

Software free, Hardware Paid

The Cloud (or SaaS) model has no physical need for indirect distribution since it is not distributed physically and is deployed almost instantaneously. The first wave of SaaS companies built their own economic model without including partner remuneration in their pricing structure, (except when there were certain existing affiliations). It has not been easy for traditional software publishers to enter into the SaaS model. Firstly, because the SaaS model does not bring them the same income structure, secondly, because continuing to work with a distribution network was decreasing their profit margins and was damaging to the competitiveness of their product pricing. Today a landscape is taking shape with SaaS

and managed service players who combine the indirect sales model with their own existing business model, and those who seek to redefine their role within the 3.0 IT economy.

Unlike traditional software which is conventionally sold as a perpetual license with an up-front cost (and an optional ongoing support fee), SaaS providers generally price applications using a subscription fee, most commonly a monthly fee or an annual fee. Consequently, the initial setup cost for SaaS is typically lower than the equivalent enterprise software. SaaS vendors typically price their

product based on the hardware that is used to deliver the service. The amount of hardware that is necessary can be based on e.g. usage parameters, such as the number of users using the service. The more commercial model focuses on the customers' data, as it resides with the SaaS vendor. In that case the providers often charge per transaction, event, or other unit of value.

A key driver of SaaS growth is SaaS vendors' ability to provide a price that is competitive with on-premises software. This is consistent with the traditional rationale for outsourcing IT systems, which involves applying economies of scale to application operation, i.e., an outside service provider may be able to offer better, cheaper, more reliable applications.

Key partners	Key activities	Value proposition	Customer relationship	Customer segments
	- maintenance service	- free software	- personal assistance	- customers
	- software development			
		- paid hosting		
		- paid hardware		
	Key resources		Channels	
	- software			
	- hosting			
Costs		Revenues		
		- hosting subscription		
		- hardware		

Figure 13: Software free, Hardware Paid – Business model

Example: Hostnet

Hostnet is a Dutch hosting company, that has built a large datacenter. They work with euNetworks and maintain data centers in Dublin, Frankfurt and Amsterdam. The datacenters are high-end and they offer services to large companies, SME's and consumers. They offer webhosting to their clients in different packages. For each of those packages you pay for the storage hardware you use: for example 2, 5 or 7.5 GB. But besides of the storage you get extras, such as software to create websites, security services and so on.

Advertisement

This is the most common free business model there is. It is the basis of almost all media, especially magazines, newspapers and free-to-air television. Advertisers buy advertising from the publisher (product 2) so they can then sell what they have to offer to the consumers who read the magazines or watch the TV shows.

“Newspaper and magazine publishers don’t charge readers anything close to the actual cost of creating, printing and distributing their products. They’re not selling papers and magazines to readers, they’re selling readers to advertisers. It’s a three-way market. In a sense, the Web represents the extension of the media business model to industries of all sorts. Media companies make money around free content in dozens of ways.”

Key partners	Key activities	Value proposition	Customer relationship	Customer segments
	- collect user information - sell user information	- free web apps - marketing information		- users of apps - advertisers
	Key resources		Channels	
	- apps - user data			
Costs		Revenues		
		- pay with user information		
		- pay for user profiles		

Figure 14: Advertisement – Business model

Example: Google Apps

Google Apps is a service from Google providing independently customizable versions of several Google products under a custom domain name. It features several Web applications with similar functionality to traditional office suites, including Gmail, Google Groups, Google Calendar, Talk, Docs and Sites. Google Apps are free for personal and educational use. Google collect all kinds of user data that they use to create profiles used for advertisements.

Twilio

Twilio is a cloud communications (IaaS) company based in San Francisco, California. Twilio allows software developers to programmatically make and receive phone calls and send and receive text messages using its web service APIs. Twilio's services are accessed over HTTP and are billed based on usage.

The company launched Twilio Voice, an API to make and receive phone calls completely hosted in the cloud. Twilio's text messaging API was released in February 2010, and SMS shortcodes were released in public beta in July 2011. Twilio uses geographically distributed datacenters to connect with carriers around the globe. Calls are optimally routed through the platform to reduce latency and provide a high quality experience for the users.

Twilio supports the development of open-source software and regularly makes contributions to the open source community. In June 2010 Twilio launched OpenVBX, an open source product that lets business users configure phone numbers to receive and route phone calls. Later Twilio engineer Kyle Conroy released Stashboard, an open source status dashboard written in the Python programming language that any API or software service can use to display whether their service is functioning properly or not. Twilio also sponsors Localtunnel, created by now ex-Twilio engineer Jeff Lindsay, which enables software developers to expose their local development environment to the public internet from behind a NAT.

5.4 Business model for the Emerginov-platform

In general the term platform is used for the technical infrastructure on which entrepreneurs can create voice-services. Within the VOICES project, the Emerginov platform (developed by Orange) was used by different partners, such as ESMT to develop code for the African context in order to apply the Emerginov APIs and Orange to offer the actual vocal and SMS services and hosting of the platform. As also mentioned in Chapter **Fout! Verwijzingsbron niet gevonden.**, the VOICES partners may not be able to continue to work on the activities concerning the Emerginov platform after the VOICES project is finished. This means that one needs to think of a suitable business model for the Emerginov platform and the operationalization of the platform. Nevertheless, the difficulty in creating a business model for the platform is that the business case for the platform itself is not as clear as for the separate services. The platform basically enables a lot of business models (for individual services), while the platform and the runtime environment ask for more extraordinary business models. In this chapter we will aim to provide an insight in the business model possibilities of the Emerginov platform by discussing the elements of the business model canvas.

Key partners	Key activities	Value proposition	Customer relationship	Customer segments
<ul style="list-style-type: none"> Platform stakeholders CSR donors 	<ul style="list-style-type: none"> Maintenance and evolution of open source platform Building and managing open-source community Providing training Providing support 	<ul style="list-style-type: none"> Platform for runtime environment Website Open-source community 	<ul style="list-style-type: none"> Word of mouth Advertising Online 	<ul style="list-style-type: none"> Community of platform stakeholders
	Key resources		Channels	
	<ul style="list-style-type: none"> Core Emerginov platform (hardware; linux; opex) Gateway (hardware; link to operator) 		<ul style="list-style-type: none"> Online Local publicity Via sms 	
Costs		Revenues		
<ul style="list-style-type: none"> Hardware Link to operator Personnel 		<ul style="list-style-type: none"> Fee / service-module Telecom traffic / data traffic Remuneration for successful service-module 		

Figure 15: Emerginov platform – business model

Value proposition

The central value proposition for the Emerginov platform is the core platform (runtime implementation of the solution) and the solution (code you can download). On top of the platform voice services can be created and tested. The platform itself only generates indirect benefits via the runtime environment and the voice services that are created. This provides an incentive for those parties being enabled to generate business by the platform to maintain the platform. Given the open source nature of the platform, this gives rise to the existence of an Emerginov open source community.

As indicated in Figure 16, two main beneficiaries of the open source community can be distinguished. In the first place we have the user community who, as the word says, have an interest in the use of the environment. These are parties as universities, paying or nonpaying customers, such as NGOs, entrepreneurs etc. With regard to Emerginov, the value being offered is in the first place an agile and flexible infrastructure on which various value adding voice solutions can be created. The combination of the availability of an open source platform with free APIs (SMS, Text-To-Speech, Call control, GSM groups etc.) and low cost gateways provide a basis for services suitable for mobile phone end-user requiring low cost and an offering in vernacular languages. These services can range from information micro-services to e-learning and mobile voice-services. Secondly we have the vendor community, who has an interest in offering their (open source) products and services. Vendors could be providers of hardware, software, maintenance etc. The value being offered to these parties is that a connection to a user community can be made, enabling a service offering. In addition, the platform makes it possible for vendors to easily share technology.

In an open source community, there is also an overlap between both beneficiary groups, i.e. a vendor could be a user of the Emerginov platform and the user could be a vendor. In this sense, the overall value for the vendor and user community is based on the proposition to function as an incubator, a source for idea spotting and to stimulate the local ecosystem. In addition the platform also functions as a prototyping platform.

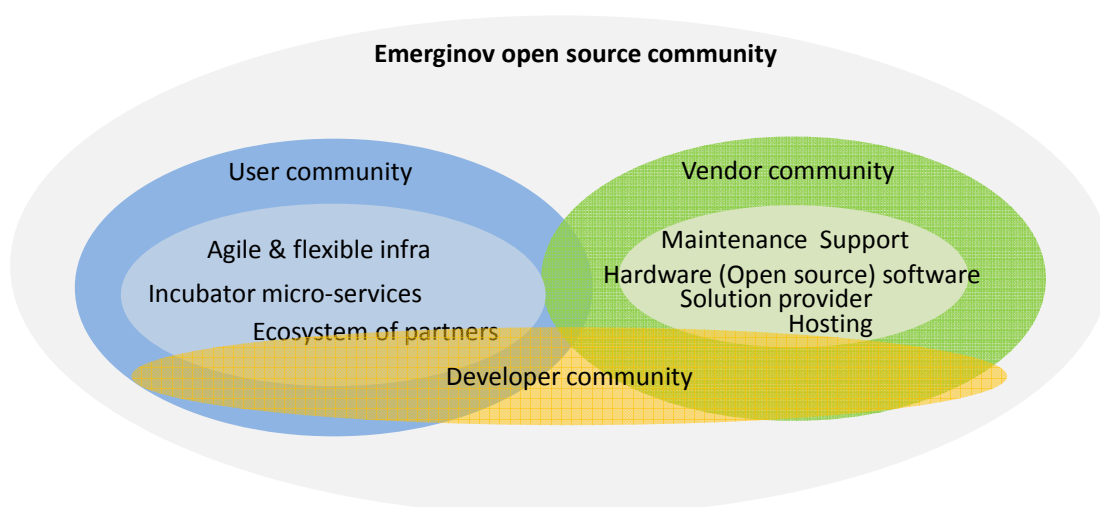


Figure 16: Visualisation of open source beneficiaries and corresponding values

Customer, channels and relationships

The main customer for this platform is a community of stakeholders (users and vendors). This community of stakeholders is enabled to create value by the nature of the existence of the platform. This value can be monetary or in other forms, such as knowledge or goodwill. Without the Emerginov platform, these parties will not be able to generate any impact and/or revenue. This creates an incentive for these parties to invest in the platform. This is done via various channels including online and local publicity, via schools, polytechnics and universities, local campaigns, twitter, world open-source meetings and a network of actors from previous projects. The customer relationship is maintained via word-of-mouth communication, advertising and online communication.

To illustrate this, we look at use of the Emerginov platform during the VOICES project. In the project, we can distinguish three customer groups of the Emerginov platform, as also illustrated in Figure 16. Firstly, we see the platform as a basis for research institutes to develop and test knowledge. For example, CSIR could use the application of the Emerginov platform in a living lab situation (WP4 and WP5) to develop and test speech technologies. Secondly, Emerginov could be used by large scale organisation to test and develop the basis for a large scale service. Orange Labs, for example, developed prototypes of m-Health services in WP5 on top of the Emerginov platform, which can be commercialised by the (local) French Telecom organisation on a more scalable platform to other health organisations in various African countries. Thirdly, we can BoP centric customers (entrepreneurs or NGOs) that wish to use Emerginov as a basis for small scale services, both internally as externally. In the VOICES project, a good example is Sahel Eco, who used the developed mAgro services to better reach their target group, agricultural producers.

In order to meet customer demand, interaction between the vendor and user community is required. For example, in the third case, Emerginov as a basis for small scale services, a (vendor) party is required to develop the Emerginov basics into a tangible service, offer the required hardware (e.g. a pc to run the service), arrange connectivity aspects (internet), hosting and maintenance & support. Depending on the customer, more or less interaction between vendor and user is required. The execution of the before mentioned activities could be done (partly) within and (partly) outside the Emerginov platform.

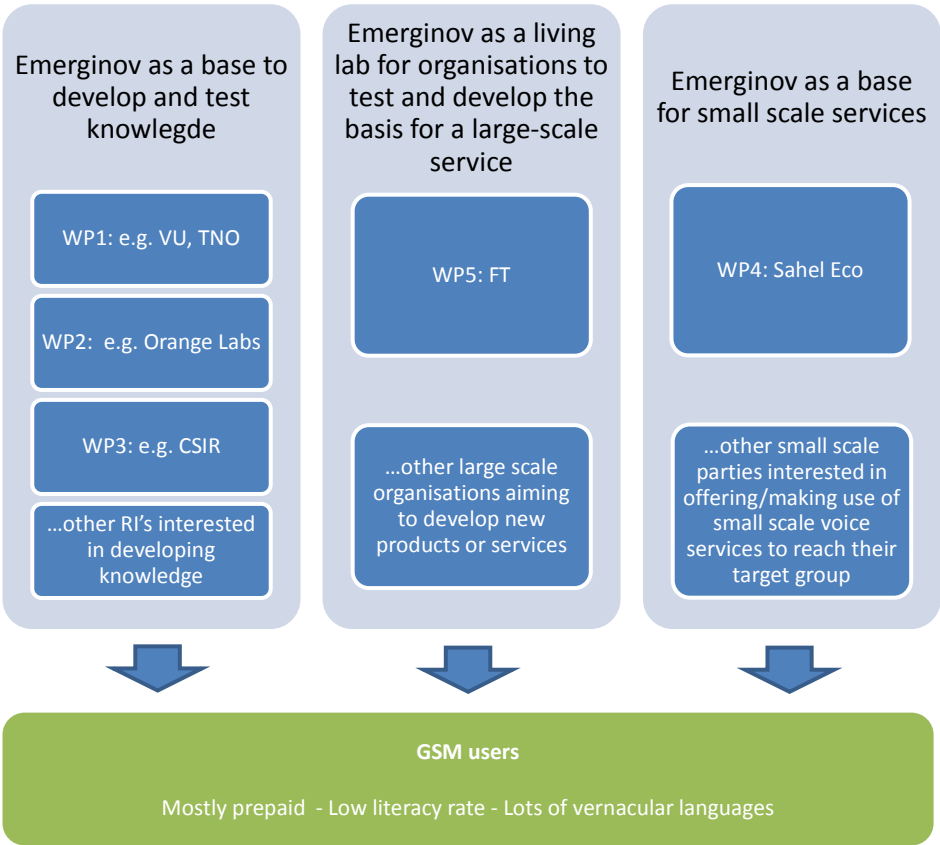


Figure 17: Emerginov customers

Resources, Activities and Partners

The main resources of the platform are the core of the Emerginov platform (consisting of hardware, software and application programming tools), the local gateway (consisting of hardware, software and a link to a telecom operator) and the speech technologies. Nevertheless, in order to be able to operationalize these resources, a strong community and the existence of co-creation is required. It is important that parties reaping benefits from the prototyping on the platform and runtime environment also maintain the platform and runtime environment itself. This can be done via a community of beneficiaries of the platform, such as the Emerginov community.

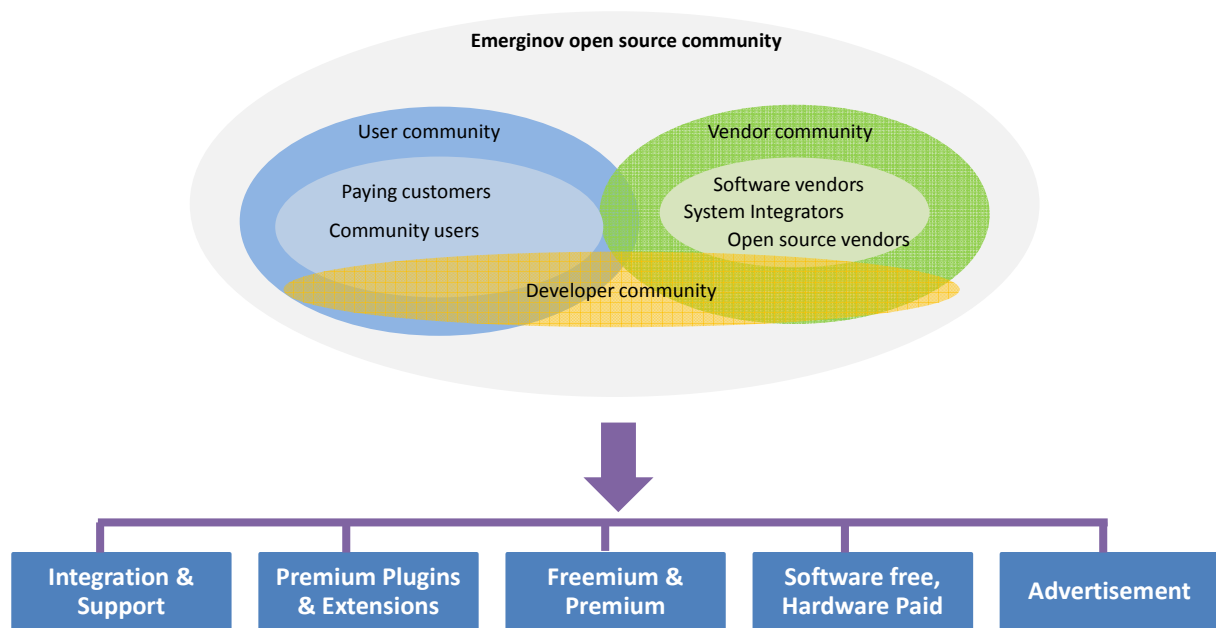


Figure 18: Emerginov customers and open source business models

The key activities very much depend on the members of the community and the business model of these members. Orange for example, may use the Emerginov platform to offer the developed voice services for free (and act as a user in the Emerginov community), but generate revenues by telecom traffic and data traffic (Business Model: Software free, Hardware Paid). On the other hand, Orange could also act as a service provider in developing customized services for professional users (Business Model: Integration & Support) and hereby generate a service fee. In addition, a research institute (community user) could 'team' with a software vendor to develop a premium software model (Freemium & Premium model).

In general, the Emerginov community could focus on convincing advertisers to promote their products or services on the Emerginov website. This could cover the cost of operating the website. In this sense it would be an important activity to attract advertisers.

Partners necessary to make the platform a success are the platform stakeholders, most likely being schools, universities, polytechnics, various vendors and entrepreneur platforms. In addition CSR donors can be necessary for some initial support and community management.

Cost structure and revenue streams

The main costs incurred in this model are the costs for hardware, the link to a telecom operator and some personnel costs for maintenance and support. As also mentioned in the key activities, various models can be developed, depending on the users of the Emerginov platform. As a result, also the revenue model will be dependent on the way the Emerginov platform will develop. Possible revenue models are a fee per license, a fee per training, a fee per service module, a fee per amount of voice data used and an agreed fee for generated

telecom or data traffic. Furthermore, a fixed remuneration can be asked when a prototype is moved from the platform to a production environment.

5.5 General conclusions

This section has provided some insights into the possible business models of the open source platform Emerginov. Open source projects are different from commercial projects. Therefore, new business models have to be used. The emergence of a co-creating community is of crucial importance to create a sustainable open source business model. This is because the co-creating community will then depend on the existence of the platform, which creates an incentive to develop and maintain the platform. Throughout the chapter, several open source revenue models were discussed. The coming period, the stakeholders connected to the Emerginov platform will be working on the most suitable revenue model.

6 CONCLUSIONS

The VOICES project aims to leverage the potential of voice based services in African countries. Notions central to the VOICES project are ‘accessibility’ and ‘sustainability’ of services. To achieve this, all the services and products developed should be replicable to a certain extent. This entails that the generic ICT architecture developed in the VOICES project should be flexible, reachable, accessible, open, re-usable and guarantee the sustainability of the services developed on top of it. The Emerginov platform, developed by Orange, was chosen as a local framework to quickly develop services on top of the networks of African operators. The idea of the platform is to allow local development of services by local people for local people and to be a bridge between local mobile contexts and the World Wide Web. The platform consists of smartly integrated open source components. The development thus hinges in a sense on the ‘glue’ that is required to integrate the different components. In order to be able to involve local actors on both the service creation side as on the back end side a community was developed. The Emerginov platform consists of an own community portal and a code repository. The community portal provides access to the code repository, a forum for co-innovation and discussion, referencing of the applications and news, amongst others.

After the VOICES project, all partners may not be able to continue to work on the activities concerning the Emerginov platform. The continuation of the Emerginov platform would be a valuable asset for various potential users. Therefore, it is important that after the VOICES project a suitable business model for the Emerginov platform will evolve. This report has provided some insights into the possible business models of the open source platform Emerginov. The overall business model of the Emerginov Platform can be explained along the business model canvas. The central value proposition for the Emerginov platform is the core platform (runtime implementation of the solution) and the solution (code you can download). Two main beneficiaries of the open source community can be distinguished. In the first place we have the user community who, as the word says, have an interest in the use of the environment. Secondly we have the vendor community, who has an interest in offering their (open source) products and services. When these two come together, voice services could be created for different stakeholders, such as research institutes, companies and NGO. The main resources of the platform are the core of the Emerginov platform (consisting of hardware, software and application programming tools), the local gateway (consisting of hardware, software and a link to a telecom operator) and the speech technologies. Nevertheless, in order to be able to operationalize these resources, a strong community and the existence of co-creation is required. The main costs for the platform are the costs for hardware, the link to a telecom operator and personnel costs for maintenance and support. In the report, several revenue models are discussed.

The coming period, the stakeholders connected to the Emerginov platform will be working on the most suitable revenue model. Orange has decided to continue hosting the platform this and maybe coming year, which provides an opportunity for the development of a sustainable business model.

REFERENCES

- [1] VOICES D1.1, use cases and requirements document
- [2] VOICES D2.1 Technical Requirements
- [3] VOICES D 3.1 Report on state of the art and development methodology
- [4] VOICES D4.1 m-Health Pilot Content
- [5] VOICES D5.1 m-Agro Knowledge Sharing Pilot Content
- [6] Richomme, Suddul, Bageet, Soobul, AFRICOMM (2010), “Emerginov: How free software can boost local innovation, a win-win deal between operator and local innovation partners” AFRICOMM 2010
- [7] Esther Duflo, Le développement humain, Lutter contre la pauvreté (I). Seuil (2010)
- [8] V. van Reijswoud, C. Toppi, Alternative routes in the digital world, Open Source Software in Africa, <http://opensource.mit.edu/papers/reijswoudtopi.pdf>

APPENDIX A: PILOTS

WP4 – m-Health

Use case 1 – Analysis result collection

This use case is a daily data collection from users without an internet access. This collection can be done by two ways:

- ⤴ using a graphical user interface on a mobile phone, then this application send pre-formatted SMS to the platform.
- ⤴ using call from a mobile to an Interactive Voice Responder.

These data are collected, analysed and stored on the platform. The results must be consultable by a supervisor on an internet interface.

Fields collected can have those formats:

- ⤴ dates
- ⤴ IDs
- ⤴ sets of possibilities
- ⤴ free fields (string or audio format)

The supervisor interface shall provide:

- ⤴ A history of all declarations sortable by date and or lab ID
- ⤴ A possibility to listen an audio field and write the field as a string
- ⤴ An exportable monthly report
- ⤴ A map with reported data
- ⤴ A function to send SMS or voices messages to users that forget to declare the lasts cases.
- ⤴ An interface to add/modify/remove labs, users...

This is a diagram of this use case:

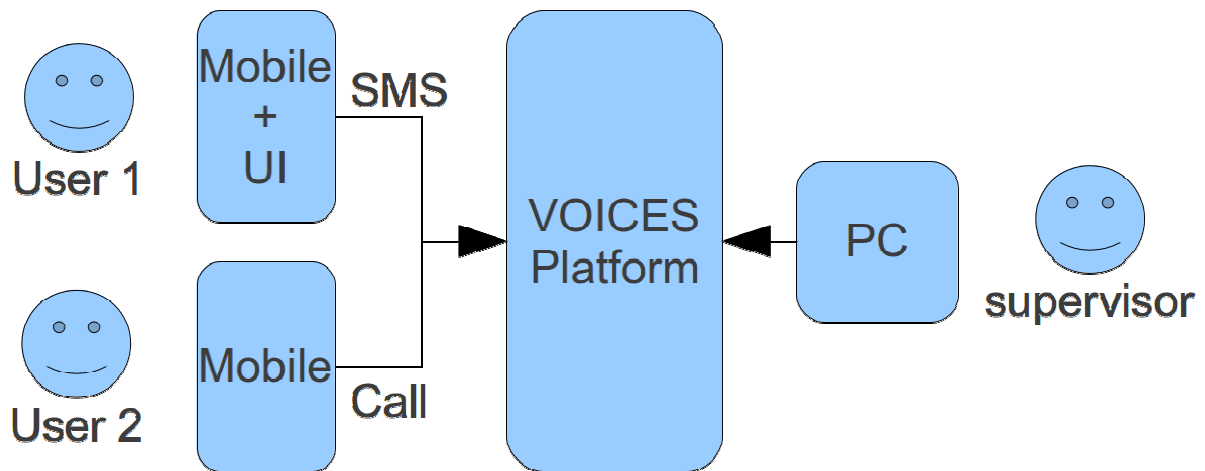


Figure 19: WP4 – Use case 1

The technical needs for this use case are:

- ✦ mobile application
- ✦ mobile/fix originated calls to an interactive voice responder
- ✦ mobile to platform SMS
- ✦ platform to mobile SMS
- ✦ web interface
- ✦ database
- ✦ French TTS engine

Needs of local number for countries: Senegal

Use case 2 - Quiz

This use case is an anonymous vocal quiz for mobile users. A supervisor creates the quiz and then launches a massive SMS/call campaign to ask users to do the quiz.

This service runs in four steps:

1. The supervisor creates the quiz on an internet interface by writing the questions and the answers in text fields. The platform or the interface transforms those texts to audio phrases and the supervisor can listen and correct the spell.
2. When the supervisor validates the quiz, a mass notification campaign is launched. For an SMS notification, the SMS asks the user to call a specific number. For a call notification, the user can answer the quiz directly or call back latter.
3. The user answers to the quiz, and after it, he can have more information if he wished.
4. The supervisor can see statistics of the quiz.

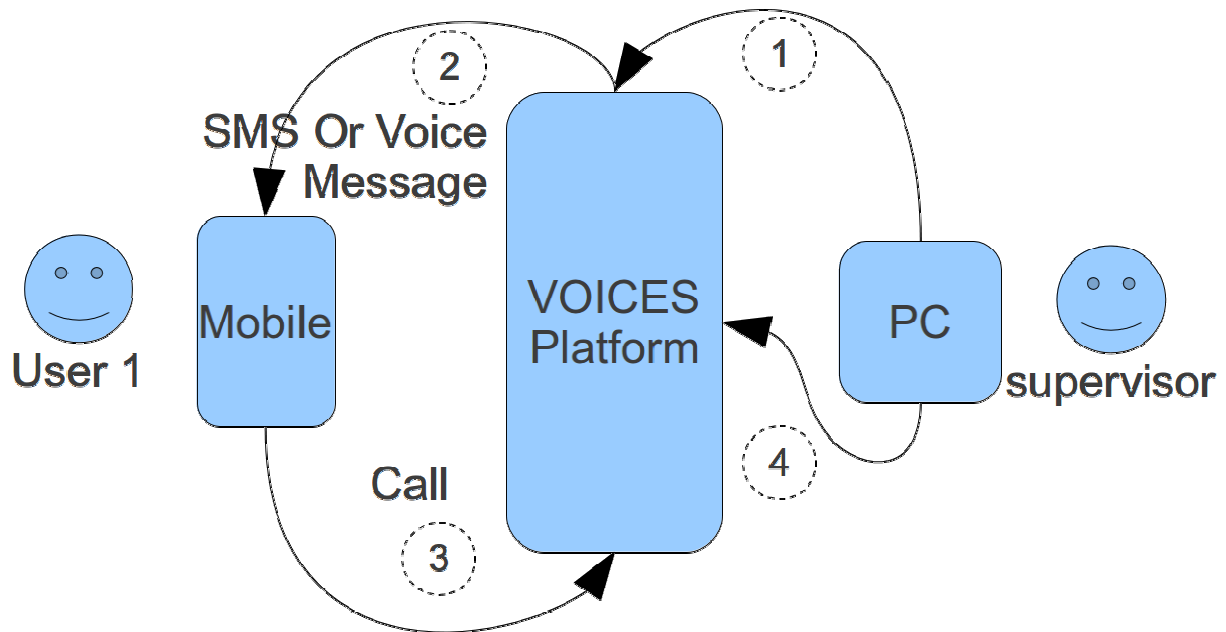


Figure 20: WP4 – Use case 2

The technical needs for this use case are:

- ⤴ web interface
- ⤴ database
- ⤴ platform to mobile SMS
- ⤴ platform to phone calls
- ⤴ phone to platform calls
- ⤴ interactive voice responder
- ⤴ French TTS engine

Needs of local number for countries: Senegal

Use case 3 – Information letter

This use case is a mass information campaign for mobile users. A supervisor creates the information letter then launches a massive SMS/call campaign to ask users to consult it.

This service runs in four steps:

1. The supervisor creates the information letter on an internet interface by writing the rubrics in text fields. The platform or the interface transforms those texts to audio phrases and the supervisor can listen and correct the spell.
2. When the supervisor validates the letter, a mass notification campaign is launched. For an SMS notification, the SMS asks the user to call a specific number. For a call notification, the user can listen directly or call back latter.
3. The user listen the letter and navigate through rubrics using DTMF or voice orders or consult old information letters

4. The supervisor can see statistics about the letter consultation.

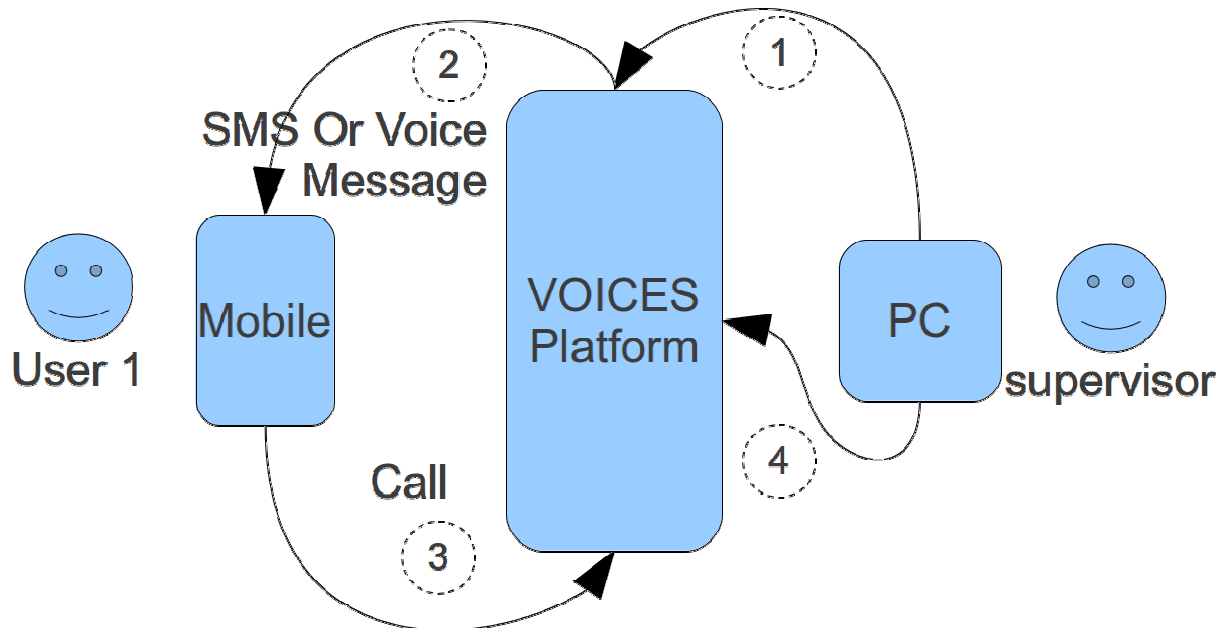


Figure 21: WP4 – Use case 3

The technical needs for this use case are:

- ⤴ web interface
- ⤴ database
- ⤴ platform to mobile SMS
- ⤴ platform to phone calls
- ⤴ phone to platform calls
- ⤴ interactive voice responder
- ⤴ French TTS engine

Needs of local number for countries: Senegal

WP5 – m-Agri

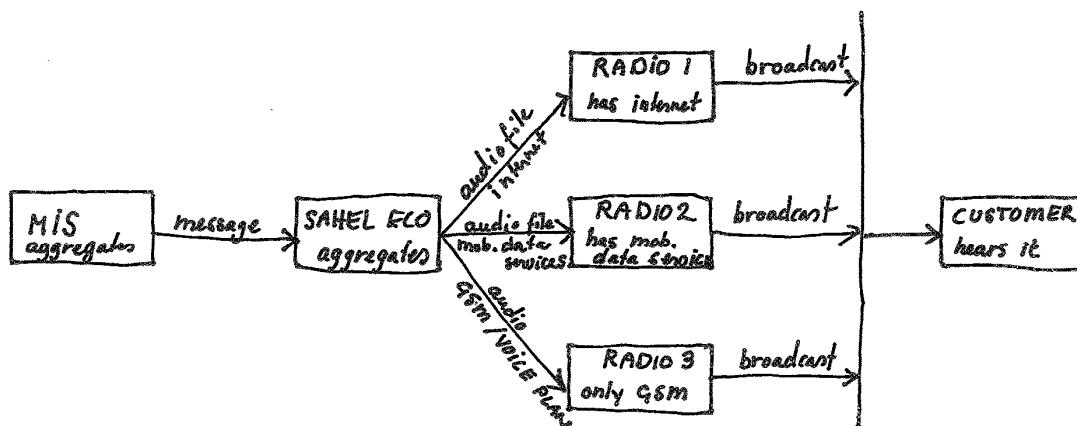
Use case 1 – Market information system

Extract from W4RA_UC_V4.0 document:

This use case will focus on the MIS communiqué transfer from Sahel Eco to Radio.

This can be done through voice and phone, depending on the options available at the radio end. Specifically this can/will be:

- ✧ Downloading the MIS communiqué AUDIO file and Broadcast:- Computer + Internet is available
- ✧ Downloading the MIS communiqué AUDIO file directly to the phone and Broadcast:- Mobile Data Services available
- ✧ Broadcast directly from the mobile phone at the Community Radio: GSM/Some voice plan with the Radio Station is available



Sheabutter & honey usecase

Figure 22: WP5 – Use case 1

The technical needs for this use case are:

- ✧ web interface
- ✧ database
- ✧ Text To Speech function
- ✧ interactive voice responder (VoiceXML is asked by pilot developers)
- ✧ massive SMS campaign

Needs of local number for countries: Mali

Use case 2 - M-Event Organizer for Re-greening events

This use case is an event notification and tracking system including the option of co-opting new members in the system.

This service runs in three steps:

1. Sahel Eco enter the event details on a web form
2. Sahel Eco broadcast the information via a voice broadcast message
 1. A subscribed user can answer the call directly
 2. A subscribed user can call latter to listen the message
 3. A non-subscribed user can call to a given number to access the information triggering its automatically registration
3. The user can accept or reject the invitation, and leave a message

The technical needs for this use case are:

- ⤴ web interface
- ⤴ database
- ⤴ Text To Speech function
- ⤴ platform to phone calls
- ⤴ phone to platform calls
- ⤴ Interactive Voice Responder (VoiceXML is requested by pilot developers)
- ⤴ Audio file storage

Needs of local number for countries: Mali

APPENDIX B: VOICE TOOLS

Deliverable D3.1 provides a high-level overview of the approaches employed in current speech technology systems and discusses some of the resources needed to develop such systems in new languages. Those resources are mainly corpora of words, sentences and recordings, extended with acoustic, phonemic, syntactic and grammatical annotations.

To minimize the expertise and costs needed to develop such resources for the VOICES project and beyond, a crowd sourcing approach needs to be set up. This requires integrating to the project platform some new software components: a crowd sourcing framework and some dedicated interfaces to help workers to perform the tasks (i.e., entering sentences, annotating sentences, recordings, etc.)

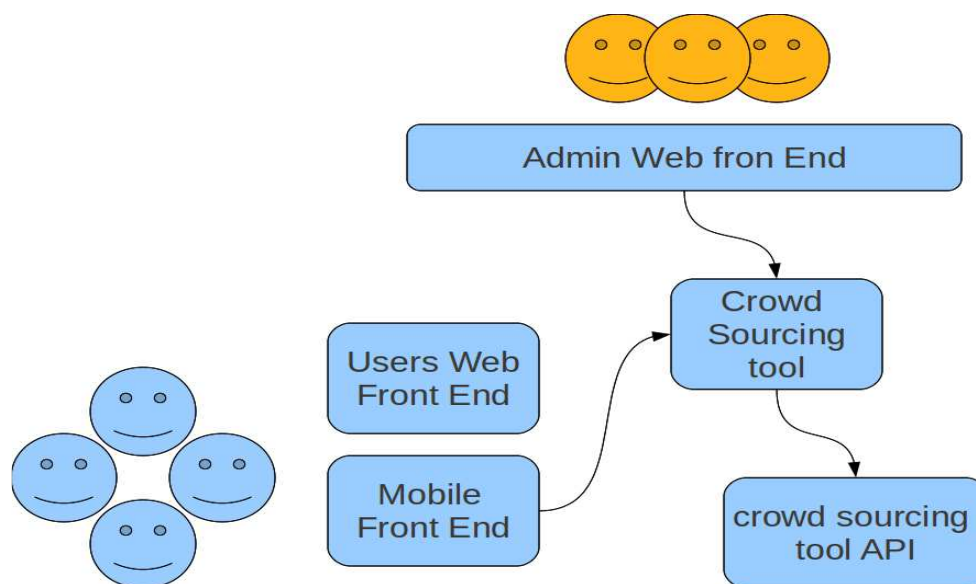


Figure 23: WP3 crowd sourcing tool

The technical needs are:

- ⤴ web interface
- ⤴ database or DAM (Digital Asset Management)
- ⤴ Text To Speech function
- ⤴ phone to platform calls
- ⤴ Interactive Voice Responder (VoiceXML is requested by pilot developers)
- ⤴ Crowd sourcing software
- ⤴ Optional:
 - Platform to mobile SMS

Needs of local number for countries: Mali

APPENDIX C: TECHNICAL REQUIREMENTS LIST

Multi channel accessibility	Analogical and mobile calls - Senegal	R.1.1
	SMS - Senegal	R.1.2
	Internet - Senegal	R.1.3
	Analogical and mobile calls - Mali	R.1.4
	SMS - Mali	R.1.5
	Internet - Mali	R.1.6
Service hosting	Service execution	R.2.1
	File hosting	R.2.2
	Database	R.2.3
	DAM	R.2.4
	Crowd sourcing tool	R.2.5
Voices tools	IVR	R.3.1
	ASR	R.3.2
	TTS	R.3.3
Developers needs		R.4.1
Administration needs		R.5.1
Mobile Applications		R.6.1
Platform capacities		R.7.1