VOIce-based Community-cEntric mobile Services for social development

Grant Agreement Number 269954

First iteration of High Level Architecture

Deliverable No D1.5

David Blaisonneau (FT)
Filipe Cabral Pinto (PTIN)
Stéphane Boyera (WF)
Morgan Richomme (FT)
Chantal Stroek (TNO)

March 2012

SEVENTH FRAMEWORK PROGRAMME
THEME ICT-2009.9.1 – International cooperation

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:	*18 February 2011*

Document	
Deliverable number:	D1.5
Deliverable title	First iteration of High Level Architecture
Contractual Date of Delivery:	09 2011
Actual Date of Delivery:	03 2012
Editor (s):	
Author (s):	Morgan Richomme [FT] David Blaisonneau [FT]
Reviewer (s):	FT, PTIN, TNO, WebF, CSR4
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:	1.1
Draft/Final:	Final
Total number of pages (including cover):	48
Keywords:	High level architecture

CHANGE LOG

Reason for change	Revision	Author	Date
First draft	0.2	David Blaisonneau	29/07/2011
Corrections	0.3	Filipe Cabral Pinto	03/08/2011
Add summary. Correction on WP5 requirements. Correction and adds in 'Technical needs description'	0.4	David Blaisonneau	02/09/2011
Add TTS for WP4 use cases	0.5	David Blaisonneau	12/09/2011
Add Crowd sourcing for WP3	0.6	David Blaisonneau	15/09/2011
First review	0.7	Etienne Barnard	20/09/2011
Add Performance and ASR openeness recommandations. Add conclusion. Final review	1.0	David Blaisonneau	29/09/2011
Full rework. Add new chapter on sustainability and business model.	1.1	M.Richomme	31/03/2012

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.

SEVENTH FRAMEWORK PROGRAMME

THEME ICT-2009.9.1 – International cooperation

Table of contents

Table of figures

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 the partners of the VOICES projects.

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 consists in describing the platform. It will detail the high level architecture and justify the choice of the Emerginov platform as the service development and testing framework for the VOICES project. The full technical description is available in the D2.1[2].

The third chapter explains how the platform will evolve, 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 business models from a service creation point of view and from the service hosting point of view.

Definitions and acronyms

API	Application Programming Interface
BoP	Base of the Pyramid
BSD	Berkeley Software Distribution
ASR	Automatic Speech recognition
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
HTTP	Hypertext Transfer Protocol
ICT	Information and Communications Technologies
IT	Information technology
IP	Internet protocol

IVR	Interactive Voice Responder
LAMP	Linux Apache MySQL PHP
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
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

1 GENERAL DESCRIPTION

1.1 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 in support of various business models. The delivered platform can be matched with services which are tested in the agri pilot in Mali (WP4) and the health pilot in Senegal (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. A visualization of the VOICES platform-service structure as such is depicted in figure 1.

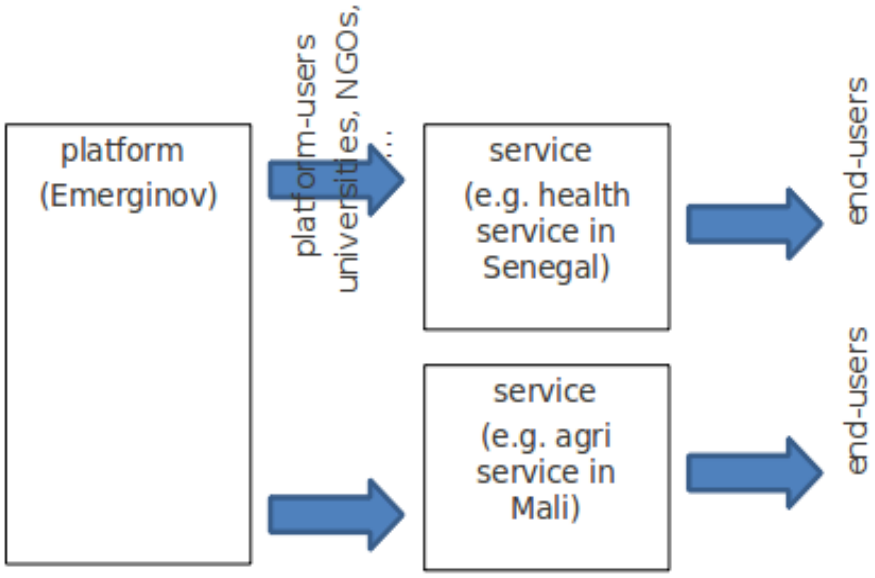


Figure 1: VOICES platform-service structure

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.

1.2 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 7.

1.2.1 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.

1.2.2 Voices tools

1.2.2.1 IVR

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

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

1.2.2.2 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.

1.2.2.3 TTS

A Text To Speech (TTS) function shall be available for services hosted by the platform. This function must propose different languages to fit the use cases needs. This function shall be compliant with Speech Synthesis Mark-up Language (SSML) and Pronunciation Lexicon Specification (PLS)

1.2.3 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.

1.2.3.1 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.

1.2.3.2 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).

1.2.3.3 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 will be evaluated on the basis of the amount of digital content to be managed.

1.3 User characteristics

1.3.1 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 hoster, 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 redo)
- a single point of management of code and contents
- a coherent management through open source licence for code and creative commons 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 and as such 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 populating, 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.

1.3.2 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-bases services.

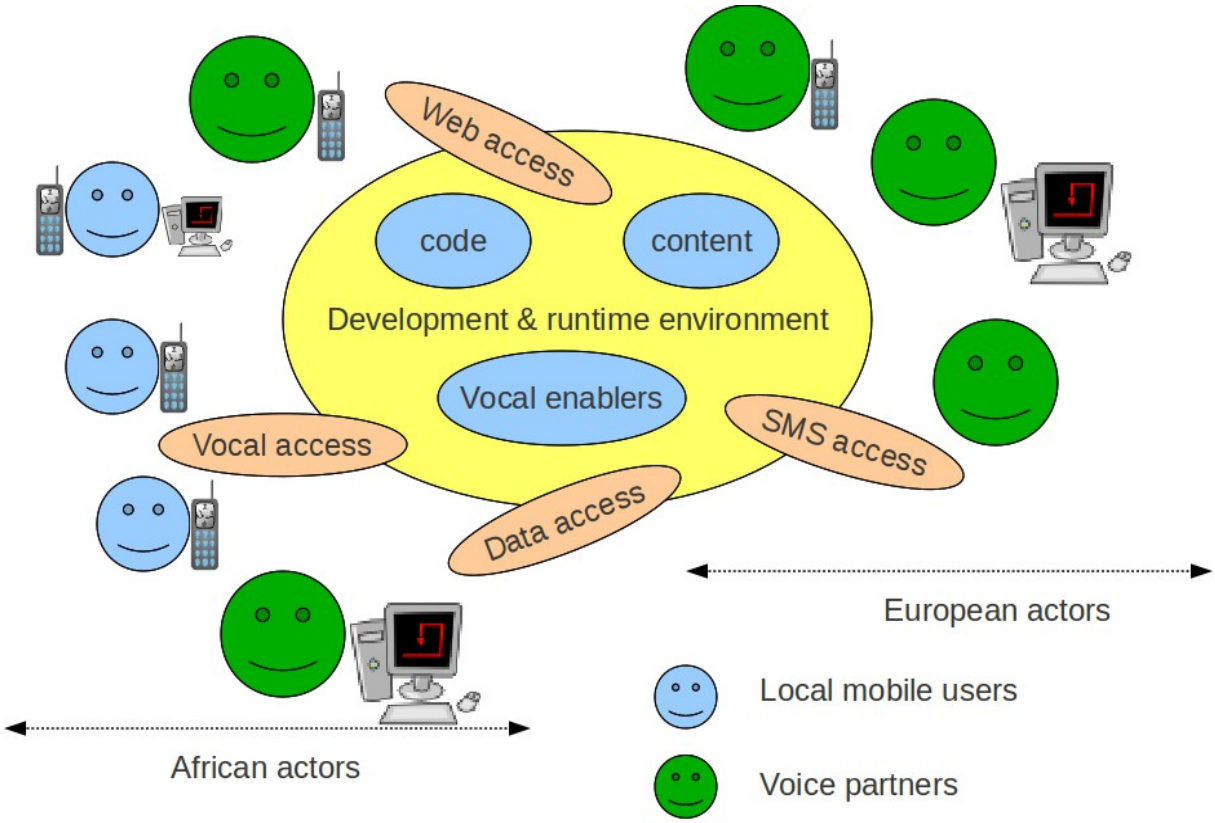


Figure 2: Actors in VOICES

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.

1.3.3 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.

1.4 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 lexicon for vocal recognition are expensive. Home made solutions has been used.

1.5 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 off last miles 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 is 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.

2 PLATFORM DESCRIPTION

2.1 High level architecture

The architecture of the VOICES platform should be like this:

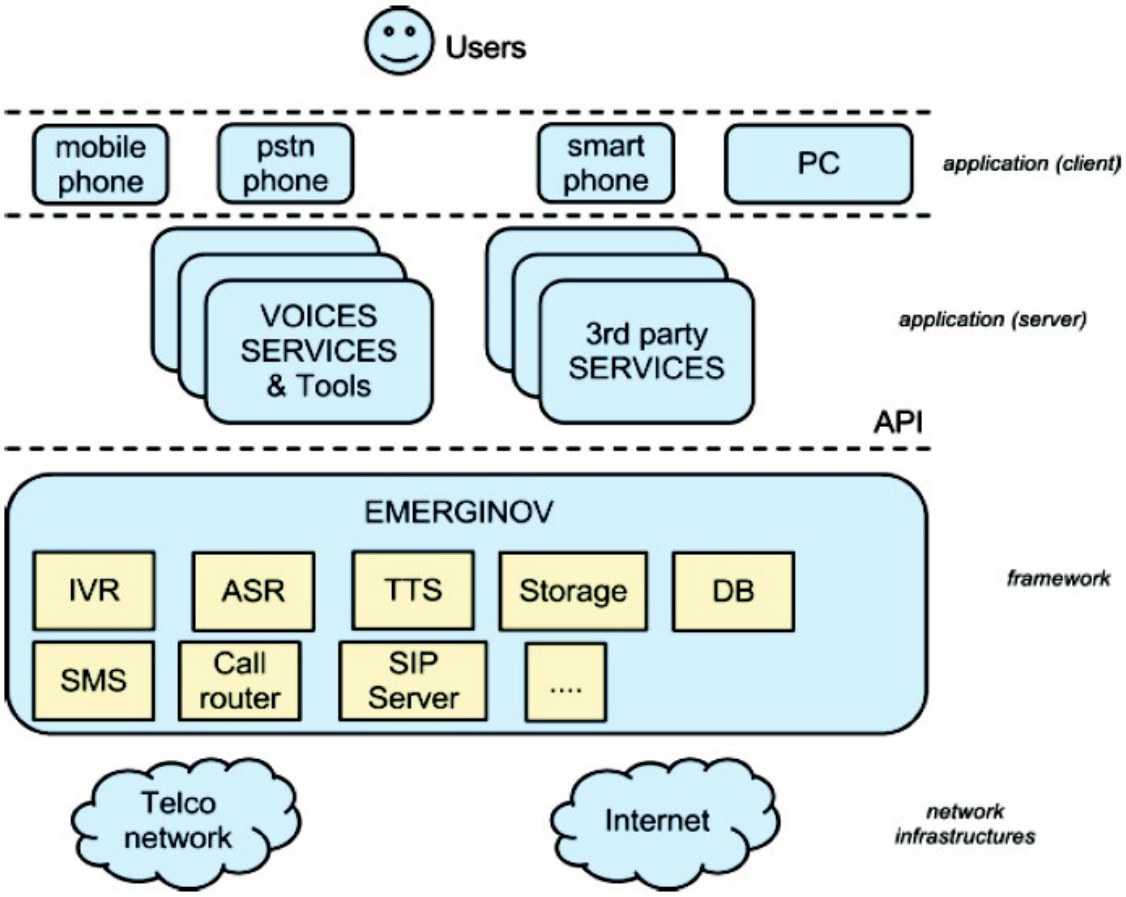


Figure 3: High level architecture

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

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

2.1.1 Flexibility

The VOICES platform must be as flexible as possible. It practically 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 of Facebook APIs for instance through internet or another telecommunication channel (use of a modem instead of a Telecommunication gateway).

2.1.2 Reachability

The VOICES platform shall provide 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.

2.1.3 Accessibility

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

2.1.4 Openness and Re-usability

To be used after VOICES project or by other persons, the VOICES platform shall be 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.

2.1.5 Long-term sustainability

The architecture must 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

2.1.6 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)

2.2 The Emerginov Choice

It has been 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.

Cloud solutions providing Voices enabler do not focus on Africa, thanks to the Emerginov solution we were able to offer a local framework to quickly develop services on top of African operator network. Please note that the platform has not been “developed”. The platform consists of smartly integrated open source components. The developments 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.

2.2.1 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:

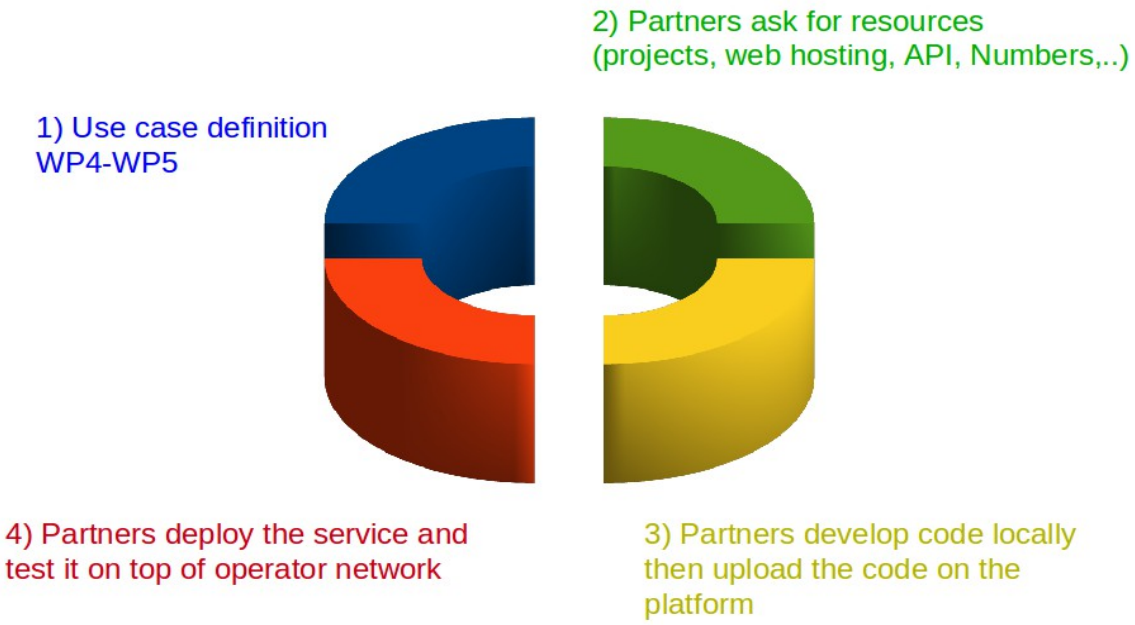


Figure 4: Development cycle

The different steps can be managed through a web interface. The developers can thus ask for project creation, project joining, resources allocation, 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.

2.2.2 Sustainable service development

The platform must be sustainable which entails that the different services shall be operable anywhere any time. 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, any time. 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, the installation procedures will be provided.

2.2.3 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 lead 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 lead to a design that could be exported locally. From the beginning, it was planned to install the solution closed 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.

2.3 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 accurate 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 closed 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 argument for sustainability.

2.3.1 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 vitualization 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
- developments 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 commiters (coders contributing to the solution code)
- activity of the mailing lists
- availability of documentation

2.3.2 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 days 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	Astersk, freeswitch, Mobicents	Asterisk	http://www.asterisk.org/
Vocal recognition	Julius, CMU Sphinx	Julius	http://julius.sourceforge.jp/en_index.php
Text-To-Speech	Festival, Flite, MARY,..	Festival	http://www.cstr.ed.ac.uk/projects/festival/
Code management	SVN, Redmine, Maven	SVN	http://subversion.apache.org/

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

2.3.3 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.

2.3.4 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.

2.3.4.1 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

2.3.4.2 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 follow:

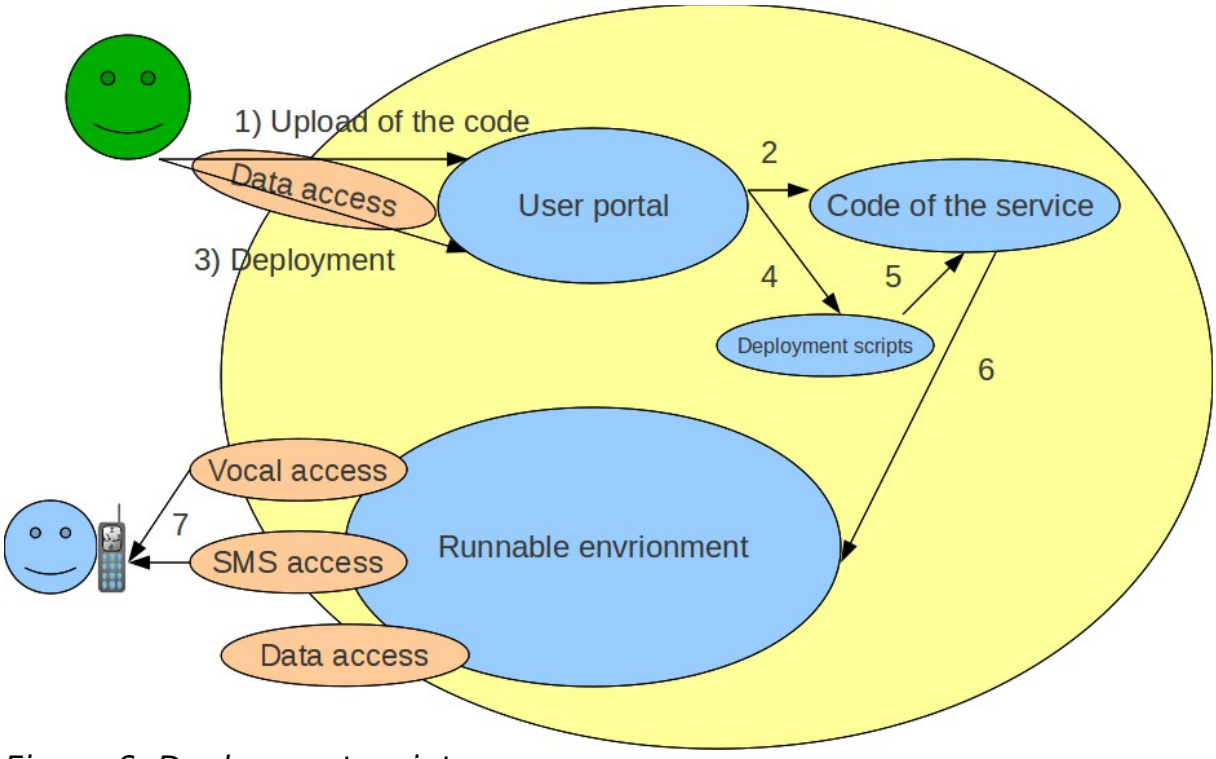


Figure 6: Deployment scripts

2.4 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.

3 PLATFORM EVOLUTION

The platform - it 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.

3.1 Roadmap

The roadmap can be summarized as follow:

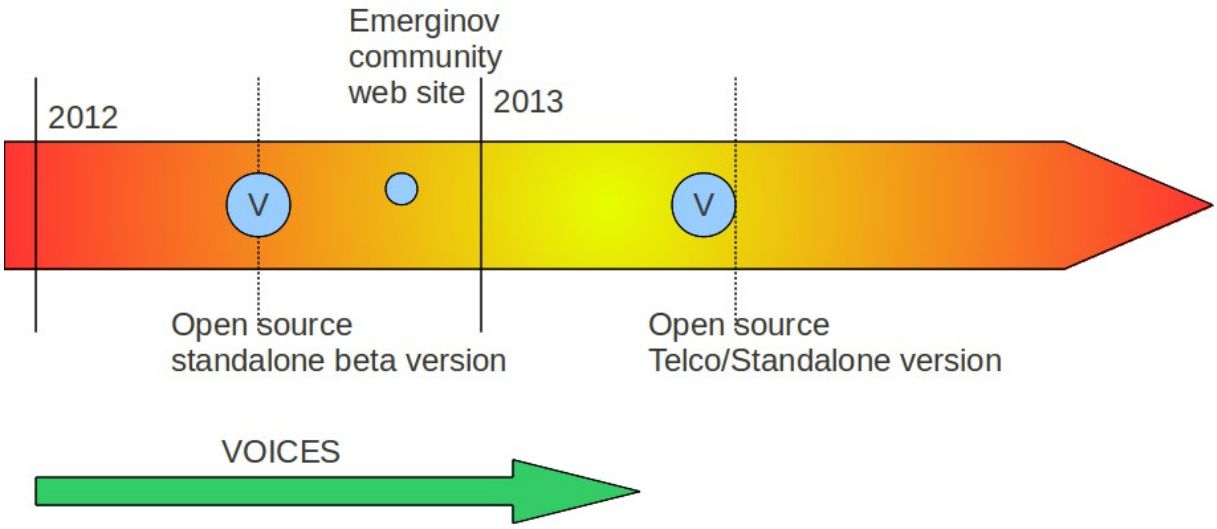


Figure 7: Emerginov community roadmap

The main goals is the publication of an easy-to-install solution. In a first step it will be a standalone version then a Telco version. The documentation, the procedures and scripts shall be available to allow beta tests with partners before the end of 2012.

3.2 towards a 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. In fact 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 will have its own portal and a code repository.

The portal will provide

- news
- documents
- access to source code
- training
- forum
- mailing lists
- Referencing of the applications: in deed 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 ambition is to create a forge of business applications under free licence in order to ease massive reuse.

A first version of the portal shall be online before end 2012.

The code repository shall be defined before end of 2012.

3.2.1 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.

3.2.2 Telco advanced version

This version is closed 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.

This version will also integrate supervision tools.

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

The publication of this version is planned for 2013.

3.3 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.

3.4 Costs

The infrastructure costs can be summarized as follow:

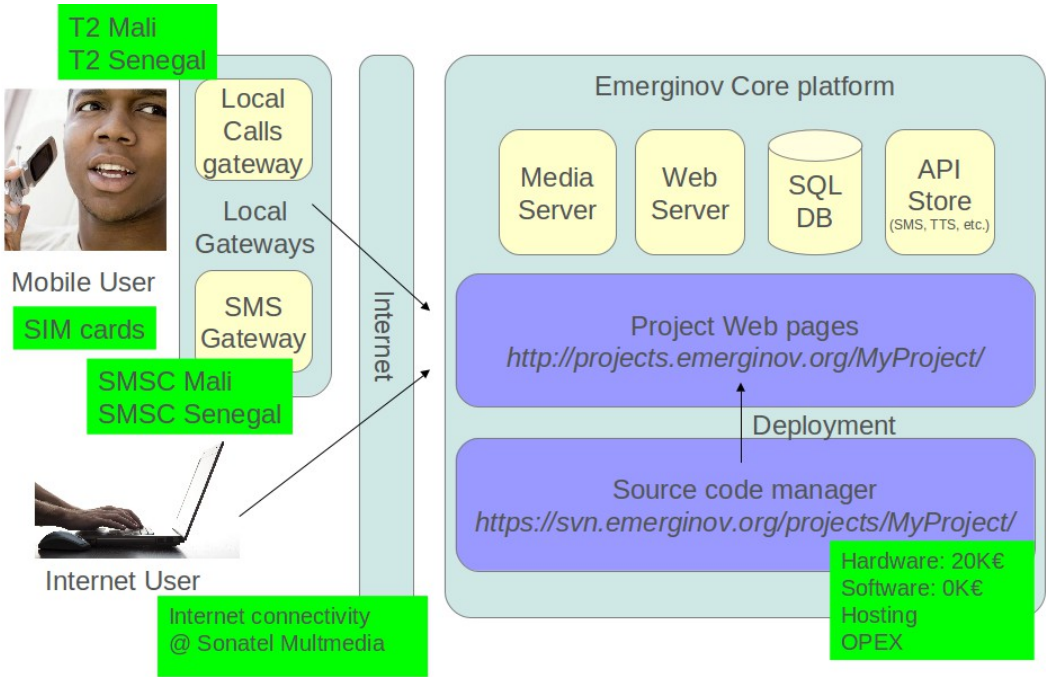


Figure 8: Cost distribution overview (advanced version)

3.4.1 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€

3.4.2 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 €HT Monthly fees: 316.30HT (for 20 B canal Max) ref: OBS offer
Senegal	Build: 700.000 FCFA Monthly: depends on the number of canal but 100.000 FCFA minimum / month Ref: Sonatel
Mali	For 30 simultaneous canals and 50 numbers (extensible) Build: 975.000 FCFA Monthly fee (without traffic): 200.000 FCFA Ref: Orange Mali

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.

3.4.3 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 an other 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.

3.5 Installation

Installation will be possible trough 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 considered 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.

3.6 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.

3.7 Feature extensibility

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

3.7.1 Extension of the existing features

3.7.1.1 Natural extension of the components

Most of the upgrade will be possible thanks to the upgrade of the existing components. Each component follows it 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 communities.

We may expect that the Emerginov community could redirect to the accurate community for efficient support.

3.7.1.2 The API nursery

The second element, the nursery, can be used to control the use of your own APIs. It can also be use 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 to expose non open source element within a full open source environment.

3.7.2 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

3.7.2.1 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 answered 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 Mobicents[1] 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 (UUSD 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/

4 BUSINESS MODEL

THIS SECTION WILL BE DETAILED IN THE SECOND ITERATION OF THE DOCUMENT.

5 CONCLUSIONS

This report has described the high level architecture of the platform selected for the Voices project. The technical needs of each Voices use case have been described. The complete technical description of the platform can be found in D2.1[2].

This first iteration focused on the Emerginov platform, even if the infrastructure is the core of the WP2. The goal was to provide the global view of the service development , motivate the choices done in the VOICES project and detail how this infrastructure will allow a sustainable digital development of services beyond the project.

In the second iteration, a focus on business model will be done.

6 REFERENCES

[1] VOICES D1.1, use cases and requirements document

[2] VOICES D2.1 Technical Requirements

[3] VOICES D 3.1 Report on start 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, Basgeet, 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. ran Reijswoud, C. Toppi, Alternative routes in the digital world, Open Source Software in Africa, http://opensource.mit.edu/papers/reijswoudtopi.pdf

7 APPENDIX A: PILOTS

7.1 WP4 – m-Health

7.1.1 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:

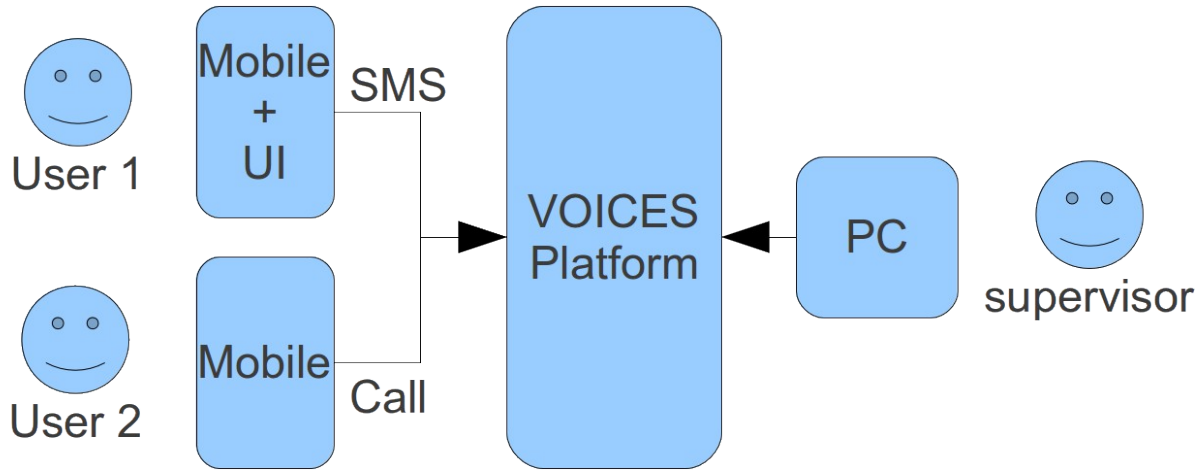


Figure 9: WP4 - Use case 1

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

7.1.2 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.

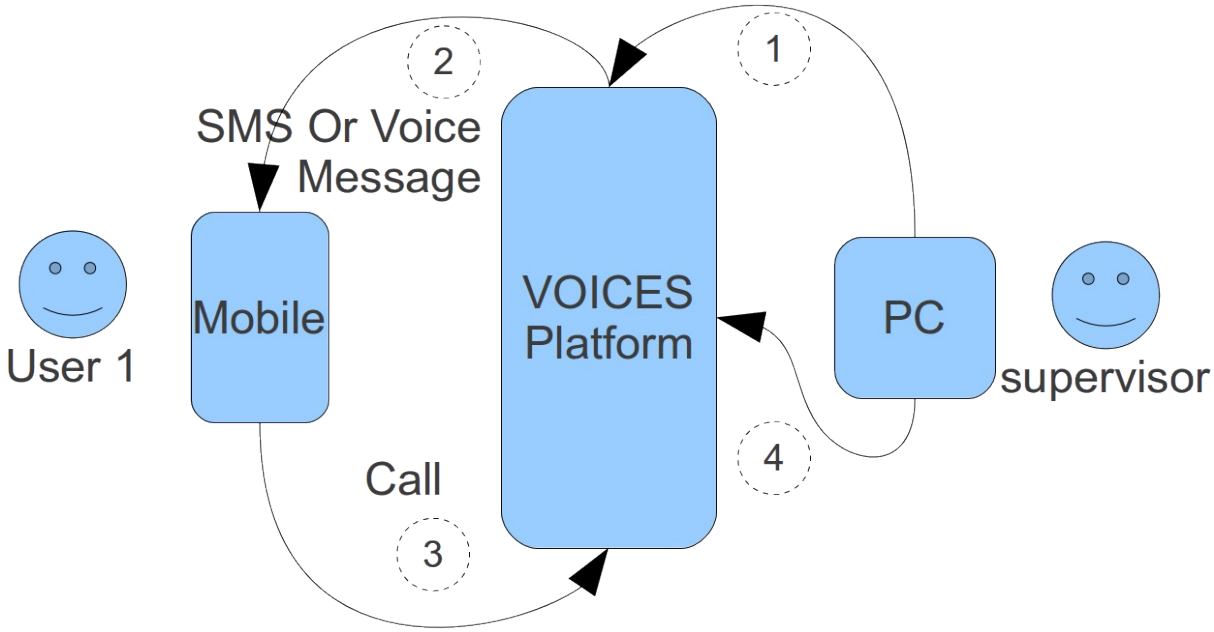


Figure 10: WP4 - Use case 2

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

7.1.3 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.

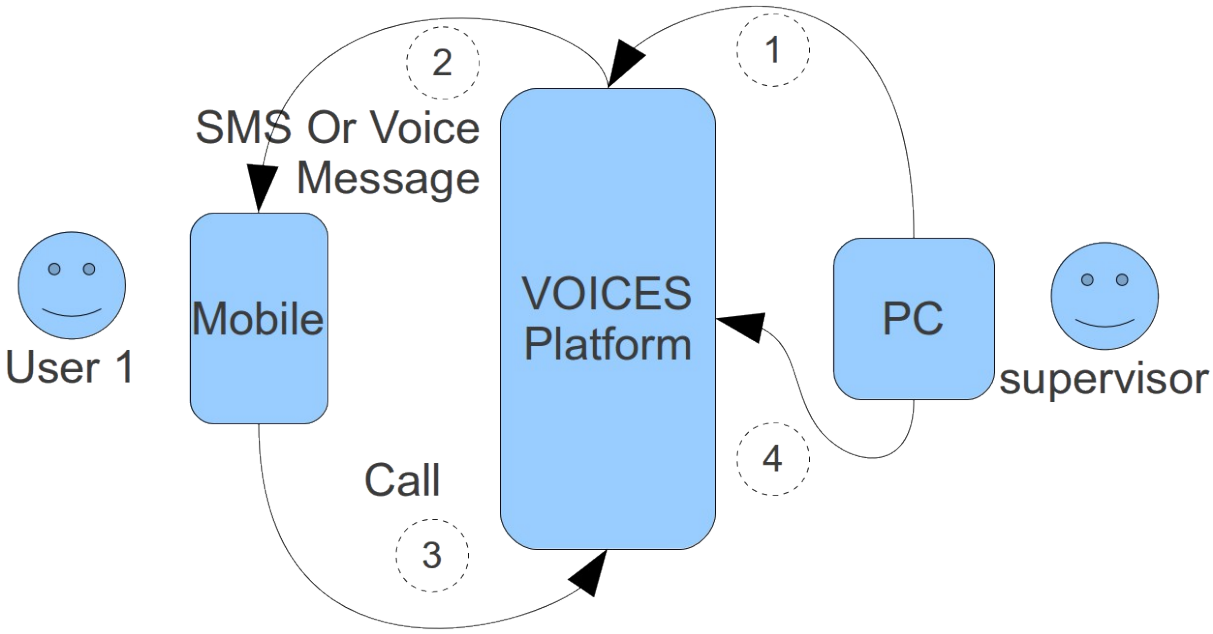


Figure 11: WP4 - Use case 3

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

7.2 WP5 – m-Agri

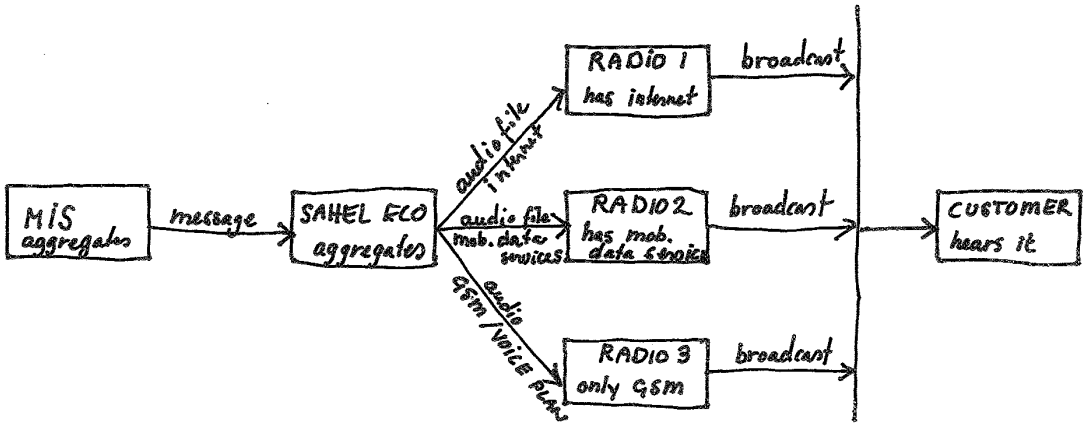
7.2.1 Use case 1 – Market information system

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 12: 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

7.2.2 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 asked by pilot developers)
- Audio file storage

Needs of local number for countries: Mali

8 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 recording, 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.)

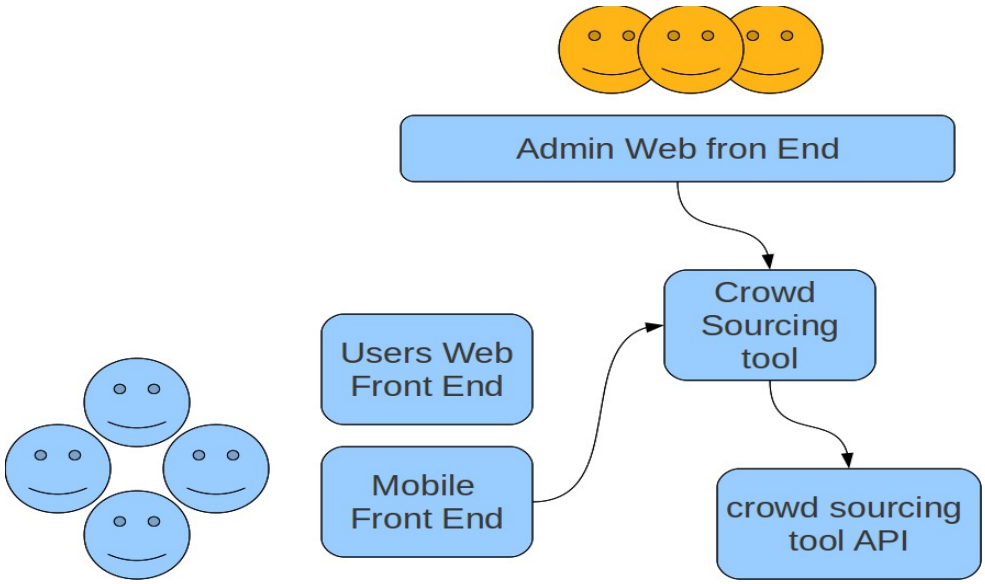


Figure 13: WP3 crowd sourcing tool

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 asked by pilot developers)
- Crowd sourcing software
- Optional:
  - Platform to mobile SMS

Needs of local number for countries: Mali

9 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