

TAKING ADVANTAGE OF THE MOBICENTS PLATFORM IN THE DESIGN OF THE SINGA ENVIRONMENT

Nomusa Dlodlo¹, JP Tolmay²

^{1, 2}Council for Scientific and Industrial Research,
Meraka Institute, Pretoria

South Africa

Telephone: (+27-12) 841 3190 / 841 4772

Fax: (+27-41) 583 2325

E-mail: ndlodlo@csir.co.za¹, jptolmay@csir.co.za²

Abstract

This paper describes an architecture that was designed to enable access to the scenarios/challenges on the SINGA platform. SINGA is a virtual platform that enables children facing barriers of access to information, services and people to learn the ways and means to acquire the skills to overcome these barriers without relying on adults and officials. The platform imparts skills through exposing the children to challenges that would have been difficult to tackle in real-world situations. Access to SINGA challenges is via “rabbit holes” which are portals into the SINGA world.

This research develops new technologies of accessing the challenges in the SINGA platform through Instant Messaging (IM) and Short Message Service (SMS) messaging. This technology is based on integrating the Mobicents platform into the software development tier landscape. Mobicents is an open source implementation of the JAIN/SLEE specification that enables developers to create, monitor and manage services that integrate voice, data and video. On its own Mobicents is limited to the above functionalities, but once integrated into the software development tier and relevant resource adaptors put in place, it can enable communication with various networks through messaging protocols in the software development tier. The various levels of the software development tier indicate the different levels of software development a developer can operate at. This research came up with two different resource adaptors for the Mobicents architecture – one to support IM and the other to support SMS messaging. Through a simple SMS or IM using a Mxit application, the children can potentially access ‘rabbit holes’ and subsequently challenges on the SINGA platform via this architecture. This is equivalent to allowing the children to program their needs at a MASHUP level of the software development tier without them being aware of the low-level complexities involved.

Keywords

Mobicents, software development tier, instant messaging, SINGA, SMS messaging

1. INTRODUCTION

Africa’s vulnerable children and youths are faced with the problem of in-accessibility to the information, services and people they need to advance in life. It is for these reasons that an initiative called SINGA (short for SINGAZENZELA), which means “we can do it by ourselves” in Zulu, came up [6].

SINGA’s objective is to make it possible for the poorest and most vulnerable of Africa’s children to play a central role in defining and accessing the information, services and people they need to advance in life. SINGA is about providing scenarios in which a child discovers how to “do things” that normally would have been difficult in a real world situation. These digitalised scenarios or challenges enable the child to learn and grow as an individual, to develop an inner resilience, to access peer-based support networks and to navigate successfully through other support systems that will enable him/her to cope with life’s challenges.

This paper generally describes the Mobicents technology [3, 4] integrated into the software development tier and its place in the SINGA environment that enables children to access the platform. The issue of how to gain access the SINGA environment, requires a technology that is accessible to these children at a low cost. 97% of the South African population have access to cell-phones. Instant messaging (IM) through MXIT costs only US\$0.01 per message and an Short Message Service (SMS) would cost US\$ 0.06 per message. On the other hand a laptop would cost about US\$400 as opposed to a cell-phone which can be as low as US\$10. Therefore this research proposes mobile technology as the route to accessing the SINGA platform.

The next section is on the methodology adopted in this research. The next section thereafter gives an overview of the SINGA environment, the software development tier landscape and the architecture that integrates Mobicents into the software development tier. The following section describes the methodology and the architecture of the technology. The next section shows the processes of IM and SMS messaging using resource adaptors to effect communication between the hardware and software layers in the architecture. The following section then discusses the Mobicents architecture and its place in SINGA. At the end is a discussion on the technology and the way forward in terms of research.

2. METHODOLOGY

The research is in the area of software development in a laboratory set-up.

2.1 Purpose of study

The paper develops a technology that will enable the children to access this virtual platform at a reduced cost. The research sees mobile technologies as the answer.

The research question therefore would be:

What mobile technology can be integrated into the SINGA environment to enable low-cost access to the SINGA platform by the vulnerable and needy?

The objective therefore would be:

- To program instant messaging (IM) and Short Message Service (SMS) into the software development tier landscape through the development of resource adaptors for the Mobicents platform.

2.2 Process

This research developed new technologies of accessing the challenges in the SINGA platform through IM and SMS. This technology is based on integrating the Mobicents platform into the software development tier landscape. Mobicents is an open source implementation of the JAIN/SLEE architecture for building communication applications in Java technology. The JAIN/SLEE specification enables developers to create, monitor and manage services that integrate voice, data and video. On its own Mobicents is limited to the above functionalities, but once integrated into the software development tier and relevant resource adaptors put in place, it can enable communication with various networks through messaging protocols in the software development tier. This research came up with two different resource adaptors for the Mobicents architecture – one to support IM and the other to support SMS messaging. The various levels of the software development tier indicate the different levels of software development a developer can operate at.

3. BACKGROUND AND RELATED WORK

The next three sub-sections of the paper describe the SINGA environment, the software development tiers and the Mobicents architecture

3.1 The SINGA environment

The SINGA digital platform (Figure 1) consists of 4 main areas – My World, My Things, My People and My games.



Figure 1: The SINGA platform

My World is the environment where a child can build a unique view of their life, and a map to other resources where they are in control and feel safe. My World is accessed either through a browser or using the cell phone's voice and text message capabilities. My People is an environment where a child can interact with people in the community, family, peers, make new friends and learn and support from other children. A key component of the My People element is the buddy list. The buddy list is the electronic address of all those people the child may wish to contact. Some addresses are provided by default, e.g., the nurse, teacher, doctor, etc. My Things is a safe place where a child can store digital objects such as their birth certificate, letters, photo album, cookbook, memory box, etc. My Things will be securely deposited in a trusted repository – a digital vault. My Games is an environment where a child can learn, play and explore and learn about life in a funny way through challenges.

In Lewis Carroll's "Alice's Adventures in Wonderland" [5], Alice follows a mysterious white rabbit into a "rabbit hole" to enter this wonderland; a world inhabited by many strange characters. This usage has helped make the phrase "rabbit hole" refer more generally to any portal into a different or strange world. The SINGA experience begins with the child discovering a "rabbit hole". "Rabbit holes" are in the form of people, billboards, newspaper adverts, bottle/can labels, radio spots, etc. All "rabbit holes" lead to information which leads to more challenges. SINGA challenges are games, visual cues, physical objects and puzzles intended to attract children through curiosity and mystery to participate in the SINGA experience. Each challenge is designed such that the children can learn something from performing it. These challenges can be solved individually or in groups.

3.2 Software tier development landscape

Figure 2 represents a general view of components that are represented in a software development tier landscape. A software developer can operate at any of the levels indicated in the tier.

The networks and messaging protocols layer harbours protocols such as Extensible Messaging and Presence Protocol (XMPP) [8], Short Message Peer to Peer Protocol (SMPP) [9] and Session Initiation Protocol (SIP) [10] for enabling messaging and communications with the upper layers of the tier. XMPP is an open technology for real communication which powers a wide range of applications including IM, multiparty chat and voice and video calls. SMPP is a telecom industry protocol for exchanging SMS messages between SMS peer entities such as Short Message Service Centres (SMSC). SIP is a signalling protocol used for establishing session in an Internet Protocol (IP) network.

The software components layer is made up of reusable libraries such as Application Programming Interfaces (APIs) [11], Dynamic Link libraries (DLLs) [12] and OCX [12] that make the work of talking to networks and messaging protocols simpler. An API is an interface implemented by a

software program to enable its interaction with other software. The DLL is Microsoft's implementation of the shared library concept in Windows. OCX is an implementation of the shared library concept in Windows with controls containing ActiveX. Developers combine the various reusable libraries into larger Open Source Projects. Open Source projects are typically suites of related software that work together to produce a complete required functionality or standalone functionality.

A service is a means for applications to talk to each other. Examples of services that can appear at the services layer of the tier are Google maps, weather service, telecommunication services, Yahoo news, CNN video service, etc. MASHUP is about combining various services from the services layer to create new customised services or applications for the applications layer. In the SINGA environment it could be a child requesting information on the key required to open a 'rabbit hole' for access to certain information.

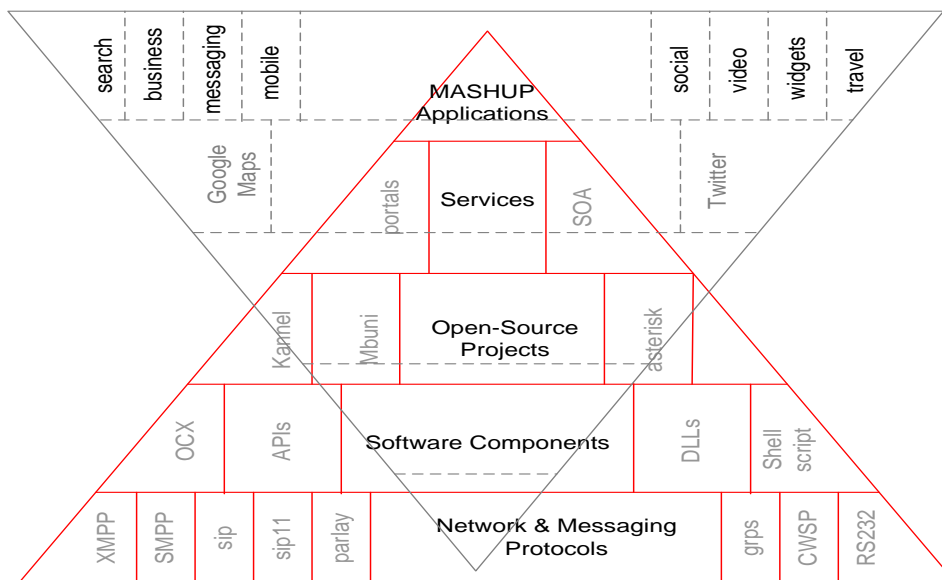


Figure 2: Software development tier landscape

The upwards triangle in the software development tier means that there are many more components at the bottom of the tier than at the top. As you go up the components become less because lower level components are being combined to produce higher level components. The downwards triangle means it is more complex to produce components at the bottom layers than those at the top layers. For example, it would take years to produce a protocol like XMPP and yet it takes only drag and drop to produce a business MASHUP application at the MASHUP level. MASHUP is an application that uses or combines data or functionality for two or more external sources to create a new application or service.

3.3 Mobicents in the software tier development architecture

The Mobicents communication platform is an open source architecture to create, display and manage services and applications integrating voice, video and data across a range of Internet Protocols (IP) and communication networks. This Voice over Internet Protocol (VoIP) platform converges voice, video and data in intelligent applications. It enables the composition of service-building blocks (SBB) such as call control, billing, user provisioning, administration, etc.

Figure 3 represents the structure of Mobicents

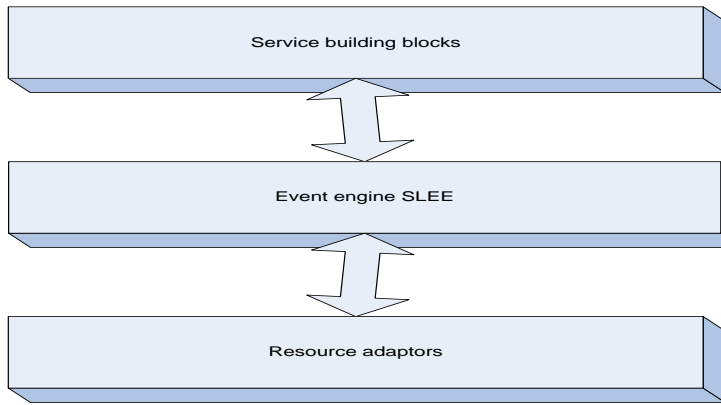


Figure 3: Structure of Mobicents

The Mobicents structure sits in the Open Source project layer of the software development tier. It is made up of service building blocks, the event engine Service Logic Execution Environment for the Java platform (SLEE) and resource adaptors as shown in Figure 4.

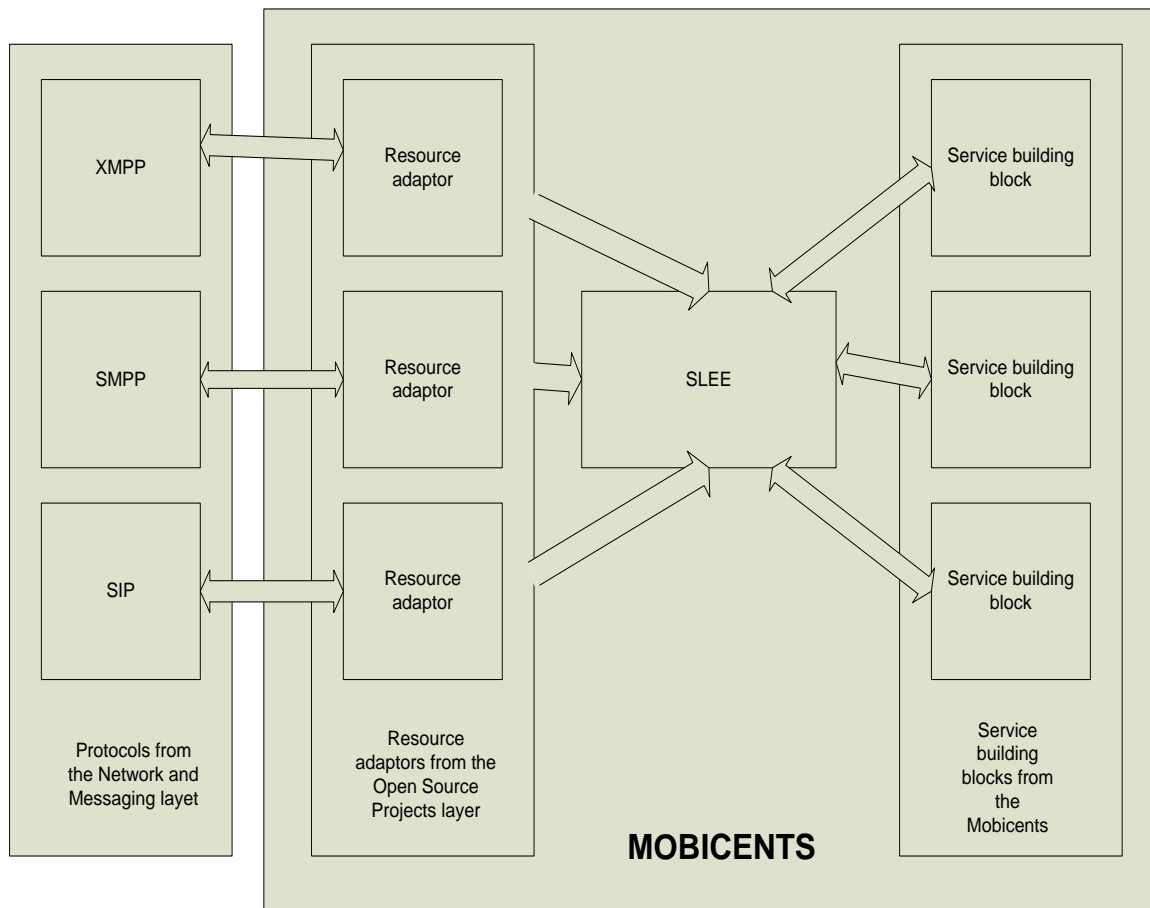


Figure 4: Communication in Mobicents

SLEE is composed of the event-model (resource adaptors) and the component model (service building blocks). The SLEE specification is an event-driven engine that allows popular protocol stacks such as SIP and Web to be plugged in as resource adaptors. The resource adaptors talk to libraries that talk to drivers that in turn talk to the hardware layer. This hardware layer then talks to other communication networks through various networks including the protocols such as XMPP, SMPP, SIP, etc. The resource adaptors translate a protocol into a language that can be understood by Mobicents in the form of events that trigger responses from the SBBs. A resource adaptor can therefore be described as a

system-level software driver that a Java application uses to connect to an enterprise information system (EIS). An EIS is a computing system that enables an organisation to integrate and coordinate their business processes to support the activities of the organisation.

Figure 5 shows the integration of Mobicents into the software-development tier architecture. It also shows the IM component [16] and the SMS messaging component [7] as explained in the next two sections of this paper.

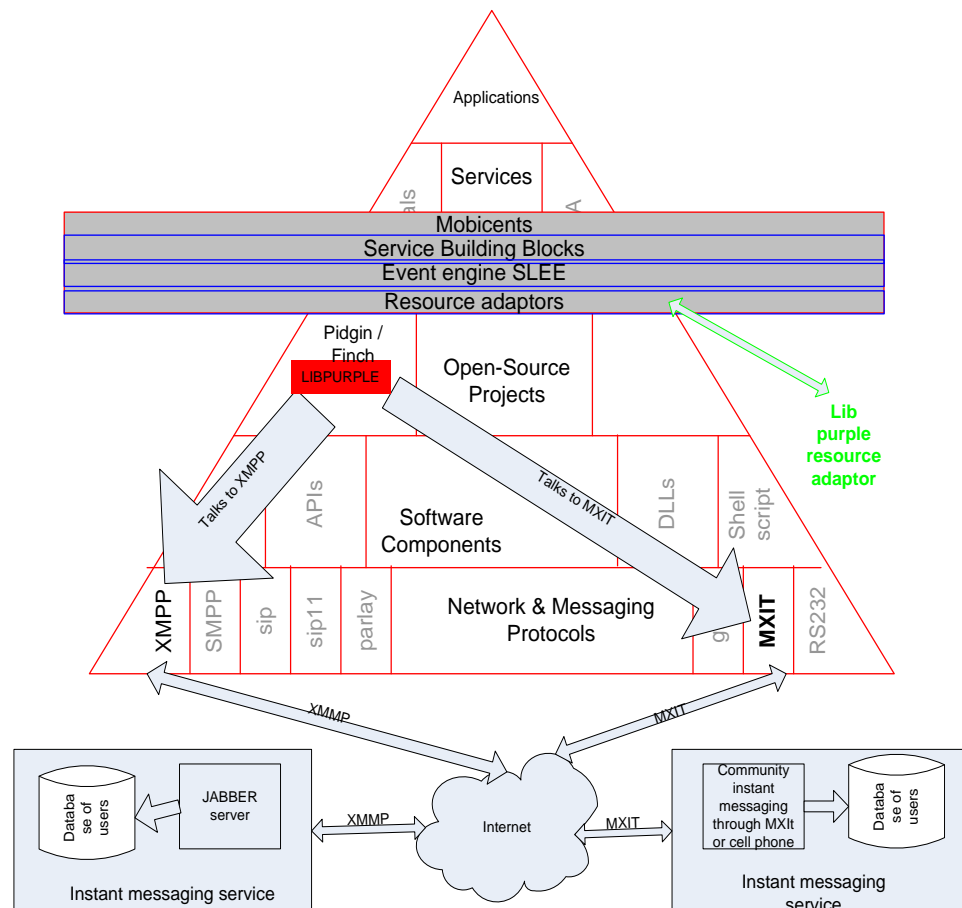


Figure 5: Mobicents in the software development tier

Examples of IM given in Figure 5 show a software library called libpurple that can communicate with MXit [19] and XMPP [8]. Mobicents alone cannot communicate with either of the protocols, for example. Therefore the libpurple resource adaptor developed from the libpurple library by this research is the component that enables communication with MXit, XMPP and the rest of the components in the software tier. Children accessing MXit therefore, are unawarely creating an application at the MASHUP layer to service their request for access to a SINGA challenge. Jabber [25] is an open source project component for IM that uses XMPP protocol. It is a server, has a database and can be used to build IM services. On the other hand MXit is an IM service that communicates with libpurple through the MXit protocol. Therefore libpurple communicates with a wide range of IM services including MXit [19], Google Talk [22], Yahoo messenger [21], MSN messenger [23], QQ [20], XMPP [8], ICQ [24], etc. through various protocols.

4. THE ARCHITECTURE AND ITS FUNCTIONALITY

The research comes up with resource adaptors to support Mobicents functionalities in terms of communicating with protocols that allow access to IM and SMS messaging

4.1 The libpurple resource adaptor and IM

IM is the exchange of text messages through a software application in real-time. The message is immediately displayed. It is analogous to a telephone conversation but uses text-based, not voice-based communication. In the Open Source Project layer of the software development tier is an engine called libpurple which is part of the Pidgin Open Source Project [17]. The libpurple library [18] is used to communicate with various IM services. The libpurple library allowed us to develop a libpurple resource adaptor for the Mobicents platform. This resource adaptor in turn communicates with the services layer through SBBs and subsequently an application can be created in the MASHUP applications layer. Pidgin is the user interface to libpurple and Finch provides a more textual user interface to libpurple.

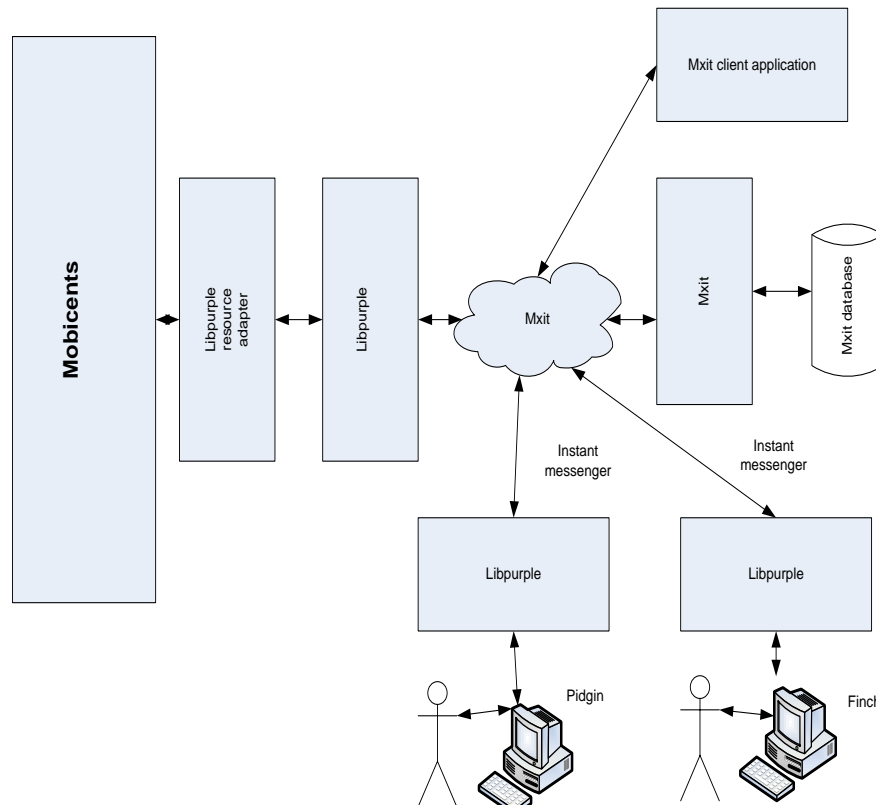


Figure 6: IM using MXit

Pidgin is an easy to use and free chat client that connects to many chat networks such as MXit, Google Talk, Yahoo messenger, MSN messenger, QQ, XMPP, ICQ, etc. MXit is an IM software application that allows one to connect with friends from anywhere inside South Africa using a mobile device. It allows the user to send and receive one-on-one text and multimedia messages to and from other users as well as general chat-rooms. Google talk is a web-based application for IM and VoIP. IM between Google Talk and its clients uses an open protocol, XMPP. Pidgin's libpurple engine makes it easy to access the bottom levels of the software development tier at the network and messaging protocol level.

In Figure 6 pidgin connects to MXit IM service. MXit has large communities of users who run mobile MXit clients on their devices. The libpurple resource adaptor enables communication with MXit services. This allows for reduction in the complexity of work undertaken by Mobicents developers at the service and application layers.

4.2 The SSMI resource adaptor and SMS messaging

SMS is a Global System for Mobile Communications (GSM) [13] -based messaging service that allows for short text messages to be sent from one cell phone to the other or from the web to a cell phone. GSM is a mobile telephony standard enabling international roaming arrangements between mobile phone operators. Once a message is sent, it is received by an SMS centre which must then get it to the appropriate mobile device.

Truteq is a South African company that acts as an aggregator for services provided by the three major cellular networks – Vodacom, MTN and Cell C. Truteq uses Simple Short Message Interface (SSMI) protocol which facilitates rapid development of SMS and Multimedia Messaging Service (MMS) [28] applications. MMS extends text messaging to image, video and audio files to be transmitted with text messages to a cell phone. Truteq enables third party developers to communicate to an SSMI server through means of a proprietary-developed SSMI protocol. This SSMI server is in turn connected to various GSM network operators and short message service centres (SMSC).

When a user sends an SMS to a pre-allocated MSISDN number (cell phone number), Truteq then forwards the message to that particular subscriber’s SSMI client. While a server hosts a service, a client requests a service from the server. The research came up with the SSMI client component based on the SSMI protocol standard. A resource adaptor was then developed using the SSMI client component to reduce the complexity of development for Mobicents developers. The purpose of this aspect of the research was to allow Mobicents to receive and send SMS messages or handle Unstructured Supplementary Service Data (USSD) requests. USSD is a capability built into the GSM standard for support for transmitting information over the signalling channel of GSM networks. USSD is a technology unique to GSM.

Figure 7 shows SMS messaging using the SSMI protocol

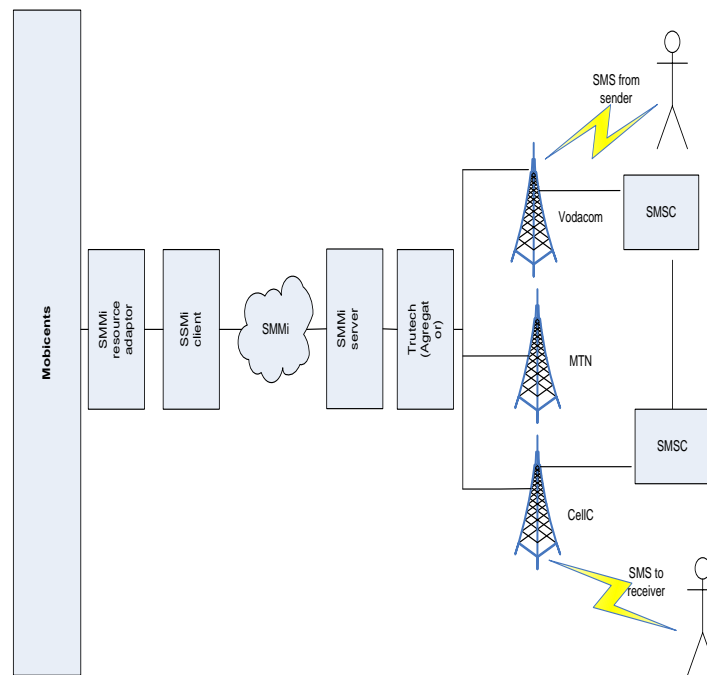


Figure 7: SMS messaging using the SSMI protocol

5. THE MOBICENTS ARCHITECTURE AND ITS USE IN SINGA

The Mobicents architecture assists in developing applications for finding entry points or ‘rabbit holes’ into a SINGA experience using a mobile phone via an SMS message or an IM on MXit.

The previous SINGA architecture was web-based and used an external GSM SMS- handling modem for messaging [1], [2]. A modem provides limited amount of messages per second. On the other hand, the aggregator allows for bulk high volume and low-cost SMS messages. The aggregator means larger volumes of users can be supported at this point. This will allow bulk messages to be sent to and from the SINGA community. The SSMI component is seen as a gateway that connects mobile users in South Africa to the GSM messaging network, e.g. SMS, USSD.

IM is a new way to connect to the SINGA world. The IM allows connecting through MXit. Lots of teenagers are familiar with MXit and the related costs to the user are negligent. The libpurple resource allocator allows us to connect to that group of users who prefer IM communications.

The research achieved the following:

- Developing a resource adaptor for IM
- Developing a resource adaptor for SSMI
- Coming up with new technology to access the SINGA environment

6. DISCUSSIONS

The latest statistics show that cell-phone coverage in South Africa is fast approaching 100%. The majority of South Africans are in possession of a mobile phone or have access to a friend or relative's mobile phone. Therefore the Mobicents-based platform developed in this research has a place in the South African environment. It is also relatively easy to purchase a low-cost prepaid SIM card in South Africa without the need to negotiate a contract with one of the mobile network operators.

Mobicents is open source software obtained under a General Public License (GPL) [27]. The license is designed to make sure that developers have the freedom to distribute copies of free software, that they can get source code and that the software or pieces of it can be used in new programs. The licence stipulates that the software can be used freely, but in case the product is commercialised, payment is to be made to the original developer. That means therefore that the current product will be produced and distributed to the market for social good although a lot of funds have been expended on its development. Mobicents was selected as the platform because it had already been developed by a much larger community over a long period. It has much more support and based on industry-accepted telecommunications standards

The system uses a decentralised approach in the development of the services. Each service is an autonomous application, independent of the next service. When one service fails the others can continue functioning since they are each autonomous.

The previous SINGA platform was web-based and a modem used for messaging. The volume of users that could access the SINGA platform therefore was limited therefore since a modem can only handle low volumes of users. Now, with SMS and IM messaging, more users can access the platform simultaneously

The final output to this research is a software that enables cheap access to the SINGA through mobile technology. This research came up with two different resource adaptors for the Mobicents architecture – one to support IM and the other to support SMS messaging. Through a simple SMS or IM using a MXit application, the children can potentially access ‘rabbit holes’ and subsequently challenges on the SINGA platform via this architecture.. The SINGA platform already exists as a web-based application. The next stage of the research will involve the integration of the resource adaptors into the SINGA platform.

6. CONCLUSION

This paper reports on an architecture that integrates the Mobicents platform into the software development tier. This is to enable access to a digital platform called SINGA. The architecture takes advantage easy access to mobile services in the form of IM and SMS messaging in the community and also the availability of open source software to build the platform from. It a refinement of previous research into SINGA which used the limited capabilities of a modem for messaging.

7. REFERENCES

- [1] Dlodlo, N., Tolmay JP. and Ford, M., “Snap me” as a pointer to rabbit holes in the SINGA environment, IST Africa 2009, Kampala, Uganda, 6-8 May 2009.
- [2] Dlodlo, N., Ford, M., Marques, L., Utilising the living labs approach in the design of a digital platform for SINGA, SAICSIT 2008, Wilderness Beach, 6-8 October 2008.

- [3] Mobicents, www.Mobicents.org/products.html [Accessed: 30 April 2010]
- [4] Mobicents, <http://en.wikipedia.org/wiki/Mobicents> [Accessed: 30 April 2010]
- [5] Carroll, L., Alice in wonderland, Digital Scanning Inc (publishers), ISBN: 10:1582187916, 2007
- [6] SINGAZENZELA, <http://wiki.metalab.co.uk/daisy/metalab/g2/688.html>, [Accessed: 30 April 2010]
- [7] SMS, http://en.wikipedia.org/wiki/short_message_service [Accessed: 30 April 2010]
- [8] XMPP, xmpp.org/about/ [Accessed: 30 April 2010]
- [9] SMPP, http://en.wikipedia.org/wiki/Short_message_peer-to-peer_protocol [Accessed: 30 April 2010]
- [10] SIP, www.sipcentre.com/sip.nsf/html/What+Is+sip+introduction, [Accessed: 30 April 2010]
- [11] AIP, http://en.wikipedia.org/wiki/Application_programming_interface [Accessed: 30 April 2010]
- [12] DLL, http://en.wikipedia.org/wiki/Dynamic-link_library, [Accessed: 30 April 2010]
- [13] GSM, <http://en.wikipedia.org/wiki/GSM>, [Accessed: 30 April 2010]
- [14] MASHUP, <http://en.wikipedia.org/wiki/Mashup>, [Accessed: 30 April 2010]
- [15] JAIN SLEE, http://java.sun.com/products/jain/article_slee_principles.html [Accessed: 30 April 2010]
- [16] IM, http://en.wikipedia.org/wiki/instant_messaging [Accessed: 30 April 2010]
- [17] Pidgin, <http://www.pidgin.im/> [Accessed: 30 April 2010]
- [18] Libpurple, <http://developer.pidgin.im/wiki/WhatIsLibpurple>, [Accessed: 30 April 2010]
- [19] MXit, <http://en.wikipedia.org/wiki/MXit> [Accessed: 30 April 2010]
- [20] QQ, <http://www.qq.co.za> [Accessed: 30 April 2010]
- [21] Yahoo messenger, <http://messenger.yahoo.com>, [Accessed: 30 April 2010]
- [22] Google Talk, http://en.wikipedia.org/wiki/Google_Talk, [Accessed: 30 April 2010]
- [23] MSN Messenger, www.msn.com, [Accessed: 30 April 2010]
- [24] ICQ, www.icq.com, [Accessed: 30 April 2010]
- [25] Jabber, www.jabber.com, [Accessed: 30 April 2010]
- [26] USSD, www.mobilein.com/ussd.htm, [Accessed: 30 April 2010]
- [27] GPL, <http://www.gnu.org/licences/gpl.html>, [Accessed: 30 April 2010]
- [28] MMS, www.webopedia.com/MMS.html, [Accessed: 30 April 2010]