Bottom Billion Architecture: An Extensible Software Architecture for ICT Access in the Rural Developing World

Jörg Dörflinger, Tom Gross

***Abstract*— The progress of developing countries towards an information society has entailed a strong demand for Information and Communication Technology (ICT) access solutions that are able to cope with the infrastructural and cultural requirements specific to rural developing areas. Most software architectures developed in former ICTD projects realize ICT access limited to one specific concept like kiosk PCs or Smartphones or SMS applications. However, extensible and reusable architectures supporting multiple concepts are missing. In this paper, we present the Bottom Billion Architecture (BBA), a software architecture that supports various hardware (Desktop PCs, Smartphones, Feature and Non-Feature Phones), applications (native, Java, Mobile Web) and data channels (voice, signalling, data). Implemented in rural South Africa, the BBA was deployed within a real procurement use case, following a Living Lab approach. During a eight month pilot experiment the BBA proved to be an appropriate architecture concept to host an improved ICT enabled procurement process that saves time and money of the participating shops. To test and improve the generalizability of the BBA concept it will be replicated in an agricultural use case in rural Ghana.**

This paper presents our research work which is based on a detailed requirements analysis, following a user centered technical ICTD research methodology. The architecture is described in detail and computational and usability evaluation results are analyzed.

***Index Terms*— Software architecture, ICTD, mobile phones, SMS Server, rural areas, procurement**

I. Introduction

In a globalized world, the economical, social, and political life will be more and more digital, and those without access to ICT will be increasingly excluded. In rural developing areas, access to ICT is a key mechanism for socioeconomic development. The mobile phone is seen as the technology with the greatest impact on development [1] and the increasing mobile phone penetration in developing countries supports this statement. Every day about 1 million people become mobile phone users—85% of these live in the developing world [2]. And in 2009 the mobile cellular penetration in developing countries has reached 67 % [3]. However, in rural areas of developing countries most of these subscriptions are related to low end phones accessing basic telecommunication services like voice and SMS. Mobile broadband penetration in developing countries is still low with only 3 % in 2009. Even if this number will increase in the near future, due to increased international bandwidth and 3G licenses, the only reliable and wide spread available technology today is voice and SMS.

To use the potential of the increasing mobile phone availability in developing world use cases there needs to be an appropriate software architecture behind. To be viable, such an architecture needs to be designed with the specific infrastructural (low, erratic and expensive bandwidth, erratic power supply, low end devices, limited remote support) and cultural (illiteracy, a variety of local languages, low ICT knowledge and different device usage behavior) requirements of rural areas in developing countries in mind. But since ICT is fairly new in developing countries there are only a view, very use case specific, architecture concepts available. What is needed is a scalable architecture that supports the ICT for Development (ICTD) requirements in general and that allows the integration of different technology concepts to be reused in different use case scenarios.

In this paper we present our concept of a software architecture that provides access to ICT via various client technologies to support a variety of rural developing world use case scenarios. Following a user centered research approach the architecture is based on an extensive requirements analysis [4] and built in rapid prototype cycles together with the end users of the Sekhukhune Rural Living Lab (RLL) [5] – a real procurement use case in rural South Africa.

In section 2 we provide an overview of related work. In section 3 we introduce the Sekhukhune RLL and the procurement use case which is the first implementation of the Bottom Billion Architecture (BBA). Section 4 describes the research methodology applied during our research work. Section 5 contains a detailed description of the BBA concepts. In section 6 we describe the implementation of the BBA in the pilot experiment and in section 7 we present the BBA evaluation results. Section 8 describes the replication of the BBA architecture in another use case. In section 9 we conclude the work and provide an outlook to our future work.

Manuscript received April 2, 2010.

J. Dörflinger, SAP Research, CEC Karlsruhe, Vincenz-Priessnitz-Strasse 1, Karlsruhe, 76131, Germany, (joerg.doerflinger@sap.com).

T. Gross, Faculty of Media, Bauhaus-University Weimar, Bauhausstrasse 11, Weimar, 99423, Germany, (tom.gross@medien.uni-weimar.de).

II. Related work

In this section we provide an overview of existing concepts providing ICT access in rural areas of developing countries.

There are two main approaches projects follow to provide access to ICT – using a PC or mobile phone as central technology. There is quite a lot of research work available based on the concept of providing ICT access via a PC in a kiosk setup. One prominent example is the eChoupal project [6]. eChoupals are information centers with PCs connected to the Internet with the aim to connect farmers with global markets. However, the value of the system was not as clear as intended to the farmers which limited its effectiveness [7]. Another project following the PC kiosk concept is the e-Sagu project [8], where PCs have been used to burn CDs, containing crop status of a farmer's field, that are used as communication device. However, evaluation results of kiosk usage behavior have shown, that less than 10% of the kiosk usage was about the intended topics [9]. One project that made a successful shift from a PC based concept towards a mobile phone SMS based system is the Warana Unwired project [10]. It started with the installation of 54 PC kiosks in rural India. The system goals have been defined without detailed on-site requirements analysis. In 2005 an ethnographic study on the Warana project revealed that none of the initial goals have been met because people had completely different demands. With these study results the use case was re-designed towards the real requirements using mobile phones resulting in an appropriate frequently used solution.

Mobile phone based concepts can also be divided into two categories—high and low end mobile phones. One project using high end mobile phones, Smartphones, for example is the CAM project [11]. CAM applications are accessed by capturing barcodes using the mobile phone camera, or entering numeric strings with the keypad. They provide a direct linkage to the formerly paper based processes but make them more efficient using new technology. Another project makes use of Smartphones in combination with GPS information to support rural supply chains [12].

The second category of projects using the mobile phone concept are projects that set the mobile phone requirements at the bottom of the technology level. With the only mobile device requirement of being able to send and receive SMS messages they are able to be used on all available mobile phones on the market. One project already mentioned is the Warana Unwired project, providing agricultural services via basic SMS messages. Another project using SMS messages is the aAqua project [13] which allows for questions to be sent as basic SMS messages. There are many other projects available building on the most common denominator in developing countries – SMS (e.g. txtEagle [14], FrontlineSMS [15], Nokia Life Tools [16]).

The difference between all these projects and what we present in this paper is that we are not limited to one concept or use case scenario but provide an architecture able to support different client access concepts (Low end mobile device, feature phone, Smartphone, PC) and different communication channels (signaling channel, data channel, voice channel) and is build of scalable and reusable software components. The presented architecture concept can be reused in different use case scenarios of the rural developing world and supports a significant feature of modern software architectures—scalability [17]. The first one will be a procurement use case in rural South Africa.

III. Sekhukhune Rural Living Lab

Our research work took place within the Sekhukhune RLL in rural South Africa – part of the Collaboration@Rural (C@R) research project [18]. The BBA was designed, developed, deployed and evaluated in a real procurement use case in the Sekhukhune RLL.

A. Challenge

In rural South Africa there are thousands of small retail enterprises, so called Spaza Shops, selling basic goods to the rural community [19]. One of the biggest inefficiencies of their daily business is the procurement of new stock because of the following reasons: a) Buying stock at the retail store in the city, about 100 km away, requires transportation costs. b) Due to limited liquidity, the shop owners are forced to buy small amounts in short cycles, and thus do not qualify for price reductions. c) An average "procurement trip" typically consumes half of a business day.

B. Solution

The C@R project improved this situation by introducing an ICT enabled business process, where the Spaza Shops get organized in a virtual cooperative and can keep on stock without leaving their shops. This new procurement business process involves the Spaza Shop, the Infopreneur™ [20], and a supplier. Infopreneurs™ are local service providers representing the interface between the informal economy (Spaza Shops) and the formal economy (suppliers). They are responsible to manage (recruit, support) the Spaza Shops and to check and validate incoming stock orders. To improve the bargaining power of the Spaza Shops, they get organized in a virtual cooperative [21], where they also benefit of bulk orders and price reductions. Beside the generic benefit of more transparency, the BBA enables Spaza Shops to place their stock orders via their low end mobile phones without leaving their shops (and thus save the logistics costs), Infopreneurs™ to manage the virtual cooperatives of Spaza Shops and suppliers to get a more transparent and efficient delivery of products to the rural communities.

IV. Research Methodology

The development of a software architecture providing ICT access in the rural developing world belongs to the technical part of Information and Communication Technology for Development (ICTD) research, which refers to ICTD topics specifically relevant for computer scientists and engineers – Technical ICTD. Many previous approaches of providing ICT access in developing countries within the technical ICTD research field failed because of an often applied “copy&paste” approach of 1st economy concepts in use case scenarios with a completely different governmental, cultural and infrastructural context [22]. Successful technical ICTD research needs a shared methodology that involves the end user in all research lifecycle phases spanning design, development, deployment and evaluation [23].

The overall framework in which our research work took place was the Sekhukhune RLL. The Living Lab concept [24] itself is build upon two main principles: a) involve end users as co-creators and b) experimentation in real world settings. To put the focus of our research work on the end user we have applied the scarcely used Mobile HCI research methods of situated use—Field Studies, Case Studies an Action Research [25]. The combination of the user centered Living Lab approach together with the in-situ Mobile HCI research methodologies form the main pillars of our research methodology: user centered design (UCD) and Action Research.

The combination of UCD and Action Research covers all technical ICTD lifecycle phases. The UCD approach was used during design, development and deployment realizing an iterative development process in tight collaboration with the end users. Action research was utilized in the requirements, deployment, and evaluation/monitoring phase. In practice this was realized as follows.

After a requirements analysis case study in June 2007 containing questionnaires and expert interviews of 27 Spaza Shops, 21 regular customers and 3 suppliers, the first prototypes have been created in collaboration with the end users following the participatory design and rapid prototyping approach. Another output of the requirements analysis have been Personas and Scenarios [26] supporting the prototype developers situated thousands of kilometers away from the use case environment. In two workshops with the end users in March and April 2008 the iterative prototypes have been evaluated and requirements collected as input for the next iterative prototyping cycle. In October 2008 a field study was conducted to again refine the iterative prototypes together with the end users. During an end user workshop in February 2009 the system was deployed and even if the users already knew the system due to the former workshop interactions, users have been trained again in how to use the system. In an evaluation workshop in April 2009 feedback from the users was collected, providing input for another refinement iteration of the system. A system evaluation was done end of May 2009 measuring the impact of the ICT system with questionnaires and expert interviews of 21 Spaza Shops, 2 Infopreneurs™ and 3 suppliers. In parallel to the ongoing 8 month pilot experiment (Feb-Sept 2009) system log files have been analyzed and continuously end user interaction via a local expert, the Infopreneur™, have been used to improve the pilot and to collect detailed usability and technology acceptance information.

V. Architecture Concepts

In this section we discuss the purpose and individual building blocks of the BBA.

A. Architecture Purpose

During the requirements analysis phase we’ve identified a set of challenges the BBA needs to cope with [4]. Most of the infrastructural (low, erratic and expensive bandwidth, erratic power supply, low end devices, limited remote support) and cultural (illiteracy, a variety of local languages, low ICT knowledge and different device usage behavior) challenges can be generalized for other use case scenarios in rural areas of developing countries (e.g. [11]). Since the purpose of the BBA is to support use cases in the developing countries context, the utilized approach is quite different compared to high level 1st economy concepts. Advanced concepts, like software as a service (always online) or cloud computing, work fine in an environment with broadband connectivity and high end devices but the deployment of such complex systems on a low end device in a remote area with only erratic GSM network coverage will most probably fail. Other approaches require mobile devices capable to browse XML or HTML, having a local database, file system and a Java runtime installed. But all these requirements limit the applications to a user group still far above the Next Billion [27] market.

One significant difference of our proposed architecture compared to existing approaches is the fact that we optimized the system regarding usability, robustness and performance by the utilization of the simplest set of components and concepts necessary to realize the system on client side but using high end technology at the backend to support extensibility and scalability towards more advanced technologies in future. Following this approach the BBA is able support every mobile device on the market, in every environment with at least basic GSM coverage and can be used by all end users at least able to send a simple SMS message.

In the following sections we will first describe the BBA building blocks before we explain it within the procurement use case of the Sekhukhune RLL.

B. Architecture Building Blocks

The central building blocks of the BBA are the BBAServer, BBADesktopClient, BBAMobileClient and intermediary components (Mobile Service Provider, Web server) (Fig. 1).

1) BBAServer

The central unit of the BBA is the BBAServer. It serves as the common access point and connects the participating clients. The BBAServer is based on the open source J2EE Application Server Apache Geronimo. The BBAServer software architecture is a Service Oriented Architecture (SOA) based on EJB3 Web Services. The utilization of EJB3 enables easy development and deployment of the services and allows deploying the entire backend functionality into more advanced concepts (e.g. the Cloud). The backend MySQL database connection is realized with the J2EE persistence technology Apache OpenJPA. Together with EJB3 the OpenJPA

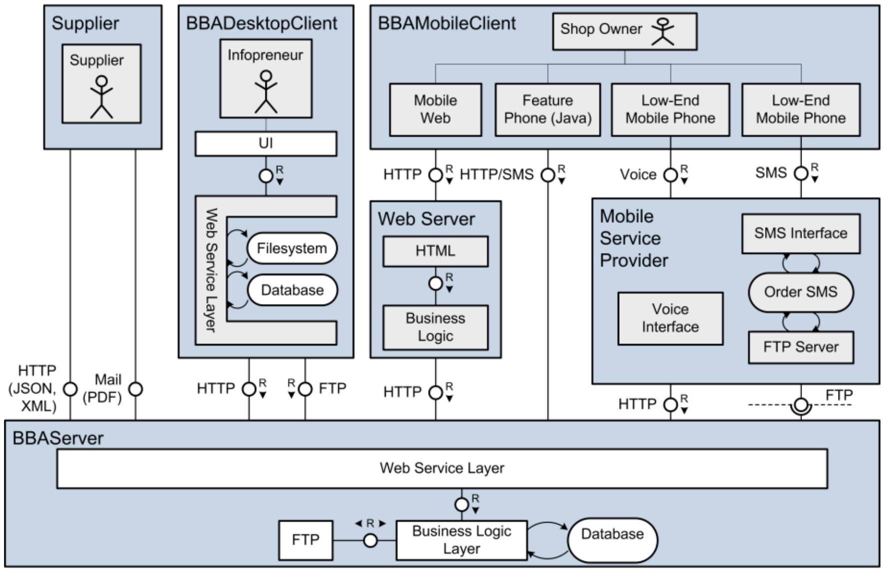

Fig. 1 Bottom Billion Architecture UML Component Block Diagram

persistence enables very efficient database handling and frees the business logic from a specific database vendor. We decided to use Apache products only instead of many diverse products since the corporation of all the different components needs to be smooth and very reliable to run in a real environment. To realize our goal of an open source based solution the Apache license also provides a good legal model.

The BBAServer exposes the business logic functionalities via a web service interface, which decouples the business logic from application server specific requirements. This web service interface also supports the requirement to be flexible in whatever Client device will access the business functionality. The Client device only needs to be able to handle web service calls to use the business functionality. This could be a native application or web application running on a Desktop PC or Laptop with access to the Internet or a feature phone or Smartphone using a mobile web browser via a GPRS network. To support also the low end devices with only SMS capabilities the BBA provides a feature to use a SMS Center (SMSC), which is hosted at a mobile service provider. The SMSC component periodically downloads incoming SMS messages. Those messages then are validated (syntax check, semantic check) and further processed (send SMS answer, trigger order process, …).

To handle the synchronization with the individual clients the BBA makes use of a local FTP server transferring compressed synchronization files. This synchronization process will be described later in more detail.

2) BBADesktopClient

To cope the infrastructural limitations the BBADesktopClient will run offline most of the time. This scenario is supported by the "local server" approach of the BBA. From a backend functionality point of view the BBADesktopClient system is exactly the same that runs on the Server. It only differs in the set of web service interfaces provided to the user. The physical system layout (Apache Geronimo, EJB3, JPA, MySQL) is the same as on the Server. Running a J2EE server on a local machine not necessarily requires a high end server machine since the Apache Geronimo server is highly configurable to only produce a small footprint on the client machine. We will provide evidence on this in the evaluation section. Using the local server approach the BBA enables to run complex applications on a Desktop Client completely offline without functionality drawbacks. Applications behave completely the same in offline and online mode. Having a local server also provides added value due to the fact that if there will be affordable GPRS connectivity in future we can remove the local server and use a real client-server architecture with always online mode without changes in application behavior. In the near future there might be concepts like Village Connection [28] bringing the internet to a remote community. For the time to come the local server could also be used in conjunction with wireless mesh networks to provide a "community network", a kind of a local Internet.

Due to the web service interface to the business functionality also available on the BBADesktopClient the UI is completely decoupled from the business logic. This provides the flexibility to change the UI technology regarding the individual use case requirements.

3) BBAMobileClient

The web service interface and the support of a SMSC enables the BBA to support a variety of mobile clients like low end SMS non-feature phones, Java enabled phones using HTTP or SMS for communication, up to feature phones, Smartphones or Subnotebooks using mobile web browser technology. The voice channel is currently under investigation but will also be integrated using the web service interface. With this multimodal client access architecture concept the BBA is able to support various use cases in rural developing areas. It also supports extensibility and the evolution of a use case scenario over time. A use case starting with basic SMS applications could evolve over time and require more advanced mobile clients like Smartphones. Those technologies can be integrated using the BBA concepts while still supporting the basic SMS applications.

VI. Pilot Experiment

In February 2009 we deployed the BBA in the procurement use case within the Sekhukhune RLL. The pilot duration was 8 month involving 34 Spaza Shops spread across 6 villages, 2 Infopreneurs™, 1 Supplier and an Administrator. Fig. 2 depicts the physical system environment of the Sekhukhune RLL procurement use case. In this example the BBA needs to cope with a broad set of different network connection types, hardware components and end user requirements.

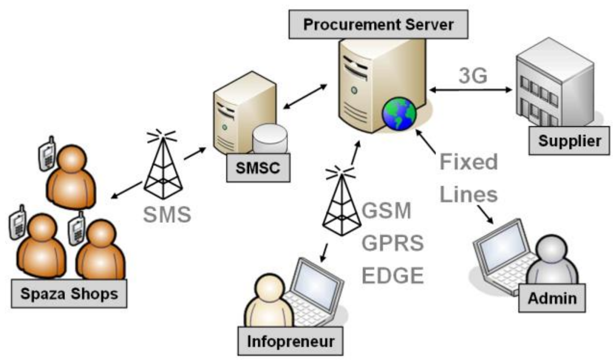

Fig. 2 Procurement Use Case Physical System Components

A. Technology

The technology used for the procurement use case was selected on basis of an extensive requirements analysis. The first implementation of the BBA had the requirement to allow all Spaza Shops to participate. The common denominator of all Spaza Shops are low end non-feature phones with only voice and SMS capabilities in an infrastructure weak environment with only erratic GSM and GPRS coverage. Thus we implemented the first version of the mobile client applications for the procurement scenario based on basic SMS mobile clients. The Infopreneur™ is equipped with a Desktop PC and slow and erratic GPRS coverage. Beside these simple facts we also based our decision on an evaluation of existing infrastructure in the target area. This evaluation analyzed the four key pillars necessary to provide access to ICT – Network, Device, Application and User Interface (UI). It extends the "Technology capabilities" comparison of the W3C Mobile Web for Social Development Roadmap [29] with focus on four attribute categories – Novice IT User, Rural Area, Emerging Economy and ICT4D.

To decide on the most appropriate network we have analyzed different network types (GSM, GPRS, EDGE, UMTS) regarding coverage, speed, reliability, cost, provider limitations, ability to host client server apps, and data transfer capabilities. The result was that GPRS and EDGE are most appropriate but require a device capable to use it. While the Infopreneur™ could use it with a USB GPRS modem, the Spaza Shops will have to rely on basic GSM.

To decide on the most appropriate device we have analyzed different devices (mobile phone, Smartphone, PDA, Mini PC/Laptop, Desktop PC) regarding usability, existing end user knowledge, applicability, robustness, power consumption, antenna power, mobility, price, availability, capability to host thick/thin client applications, and the ability to support future technology features. The result was that a basic mobile phone is most appropriate in the current rural emerging economies environment but with increasing bandwidth and lower prices for data packages the Smartphone will become a more appropriate device to improve the usability. For the Infopreneur™ applications the Desktop PC was the only available device capable to run the more advanced application.

On the application side it was necessary to decide how to develop the procurement applications. Therefore we have analyzed different application types (SMS, USSD, Java, Native, Mobile Web, and Voice Applications) regarding application flow, robustness, usability, input error rates, necessary training, service discoverability, offline capabilities, over the air installation, network requirements, multi-device capabilities, multi-user capabilities, device requirements, language support, illiteracy capabilities, cost predictability, costs, network operator dependencies, applicability for complex business applications, and scalability. The most appropriate application type would be the mobile web, but due to currently limited and expensive bandwidth and low end phones only SMS applications are viable yet for the mobile client. Even with its good usability for the illiterate people and the rising efforts, voice applications implementation is still too complex and there are too many telecommunication provider specific restrictions that block an efficient utilization of this technology. The Infopreneur™ with his Desktop PC will use a very specific concept of the BBA – a local server installation with offline capabilities.

To find the most appropriate UI we have analyzed different UI technologies (Text, Icons/Symbols, Voice) regarding usability, training for literate/illiterate, rate of misunderstandings, usability, cultural adjustability, language independence, and the capability to host complex business applications. The voice UI was most appropriate for mobile clients, but due to high communication costs not applicable in Sekhukhune procurement use case. Also using Icons/Symbols as UI was highly rated but do to the low end phone limitation not applicable. Thus we had to choose the lowest rated UI technology for the mobile clients – Text (SMS). On the Infopreneur™ Desktop PC we are able to use Icons and Symbols to improve the application usability.

The evaluation of the four ICT access requirements provides us with a matrix that could be used to find the most appropriate ICT access solution regarding available infrastructure parameters. Even if the first version makes use of basic SMS we can use the matrix to move to a more advanced technology as soon as better devices become affordable and bandwidth becomes faster. This requirement, to be extensible for more advanced future technologies but still support basic technology, is one key requirement of the BBA and significantly influenced the entire architecture design.

B. Implementation

In this section we describe the implementation of the BBA building blocks in the Sekhukhune RLL procurement use case.

1) Procurement Server

The Procurement Server is the implementation of the BBAServer concept located on a low end server machine in Pretoria (South Africa). It makes use of the local FTP server supporting the synchronization functionality and offers the procurement business functionality via the web service interface. To handle the low end mobile clients it implements the SMS application concept containing a SMSC hosted at a mobile service provider.

During the pilot operation in the Sekhukhune RLL the ability to access the Procurement Application functionality via web services became an important feature for administration and monitoring purposes. Since end users have been new to ICT applications, input error rate was quite high in the beginning. To cope with this, access to the web service interface was used for remote assistance during order placement or completion of an unfinished order.

2) Intermediate (Infopreneur™)

The Infopreneur™ application (see Fig. 3) implements the BBADesktopClient concept on a low end Desktop PC in Kgautswane (rural South Africa, part of the Sekhukhune district) and is used to manage (register, maintain) Spaza Shops and Stock Orders. Due to the expensive, slow and erratic GPRS connectivity, the Infopreneur™ system needs to work offline most of the time and only connect to the Internet to synchronize as less data as possible.

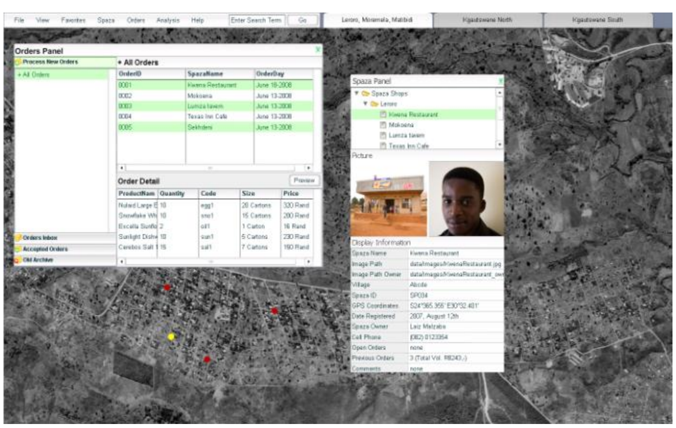

Fig. 3 Infopreneur™ Desktop Client Application

The UI of the Infopreneur™ Application is implemented in Flash. Since the business functionality is decoupled from the UI using web services it is possible to substitute the UI by any other UI technology in future. For the procurement use case we have used Flash since it offers a rich set of usability facilitating functionalities. The Infopreneur™ application UI is a Geographical Information System (GIS) based application combining functionalities to create and manage Spaza Shops as well as processing incoming procurement orders. One significant result of the workshops during the requirements analysis and field studies was that the users immediately have been able to understand and use a GIS interface. Using this highly visual representation of spatially distributed information (e.g. Spaza shop management) supports their requirement for a usability optimized application much more than using a plain text and table information representation. The GIS system also allows for future extensions like logistics route planning, analytics tools and traceability support of products and cash flows. The Infopreneur™ application also contains some specific rural area features like capturing of GPS location of Spaza Shops (important since there are no street names), image of the owner and shop (important for logistics to get the cash on delivery from the responsible person).

3) Administrator

Another implementation of the BBADesktopClient concept is the Administrator. He is responsible to monitor and support the Infopreneur™ tasks. The Administrator in this case was located in Europe, using the same system as the Infopreneur™, but with the difference that he was connected to the internet via a broadband connection. He could assist or act on behalf of the Infopreneur™ if there might be no connectivity in rural South Africa. In rural areas this could easily happen but the procurement orders still need to be processed.

4) Mobile Client (Spaza Shop)

The Mobile Client in the Sekhukhune RLL procurement use case is a Spaza Shop owner using a low end mobile phone to place an order into the procurement system. To support the low end devices the Mobile Client implements the BBA SMS Mobile Client concept using the SMSC. To place an order the Mobile Client sends a structured order SMS message (see Fig. 4) to the dedicated number of the SMSC. The structure of the SMS message is: “username pin products[amount*short code]”. All products with corresponding information like short code and price are available to the Spaza Shop owner via a paper catalogue. The Procurement Server Application periodically retrieves new incoming orders from the SMSC and further processes them (see Fig. 5).

Fig. 4 Order SMS Message

Since the response time of the application is very important, especially for a user group unused to ICT applications, the SMSC immediately provides feedback if the message went through or not. Depending on if the incoming SMS was a new order placement or a confirm/cancel of a placed order the Procurement Server Application sends out further SMS

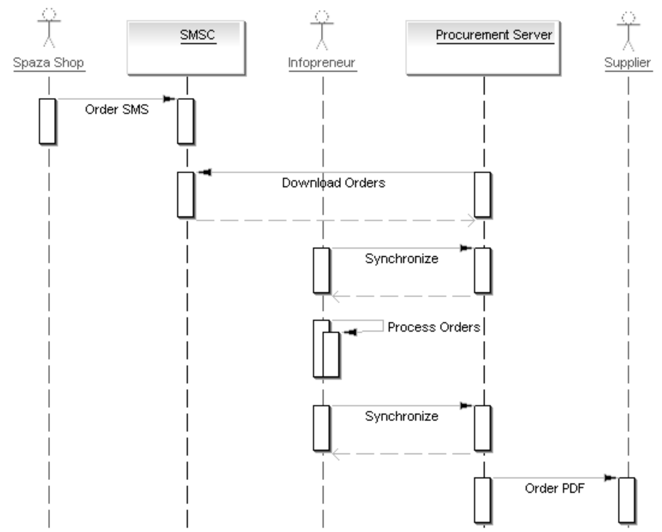

Fig. 5 Simplified Order Process UML Sequence Diagram

messages back to the mobile client within an average response time of 1 minute. The immediate response of the SMSC and a maximum round trip time of about 2 minutes to place an order ensures a high usability level. The system capabilities enable a much faster overall round trip time of about max. 30 seconds to place an order but we have limited the application to only download new orders every minute. This is to reduce the client-server connections and keep the system usage at a minimum level. This makes the procurement application affordable also in areas with high internet costs and minimizes hardware requirements for the server.

5) Supplier

The Supplier is the last step in the order process. He needs to get a document stating the order details. In our first architecture prototype we have implemented the Supplier interface as simple as possible. After the Infopreneur™ has processed and synchronized the orders, the accepted orders will be on the Procurement Server ready to be finalized. These orders now are bundled into one PDF document with the order details and automatically send out to the Supplier. Future implementations will realize the Supplier interaction much more intensive using B2B concepts like ebXML.

C. Runtime Scenario

Due to the infrastructure limitations (slow, erratic, and expansive bandwidth) the BBA supports the applications to run offline most of the time and use the BBA synchronization mechanism to synchronize with the Server to ensure data consistency. In this section we provide an overview of the key technologies supporting these requirements.

1) Occasionally Connected Desktop Client

The Infopreneur™ Desktop Client needs to be able to run in an environment where it will be mostly offline (occasionally online). While limiting the network dependencies of the Desktop Client to a minimum the business process management became much more important. This is because placing a procurement order in an always connected environment is one transaction, but in an environment with only occasionally connectivity the order process is an asynchronous process split into several individual sub processes executed off- and online.

Additionally to the problem of unexpected disconnection during a transaction, which is a problem for every transactional mobile application, the BBA needs to cope with disconnections that last several hours. The Procurement Process (Fig. 5) starts with a structured order SMS placed by the Spaza Shop to the SMSC. Periodically the Procurement Server downloads the incoming SMS messages and processes them (validation, syntax check). The Infopreneur™ connects to the Internet with the Desktop Application approximately twice a day to synchronize with the Procurement Server. After processing the new orders he needs to synchronize again. When the processed orders have been synchronized the Procurement Server automatically creates and sends an order PDF document to the Supplier. Since there is no dedicated timeslot when the Infopreneur™ will synchronize and process the orders it is necessary to keep the order status consistent on client and server side using the BBA synchronization concept.

2) Distributed Business Data

To cope with the mostly offline scenario the BBA realizes the local server concept on the Desktop Client. Thus we do not have a traditional client-server architecture where the entire business data is stored in the central server but have some kind of "thick client" holding its own copy of the relevant business data locally. This distributed business data is subdivided into three categories—Master data (e.g. Product name, price), Transactional order data (e.g. Order items, quantity) and Transactional customer data (e.g. Spaza name, location). The differentiation in these three data categories ensures that the synchronization process only synchronizes the minimum amount of information necessary to keep all nodes in sync.

Master data is only synchronized from the server to the client. This only becomes necessary if new products are introduced or existing product information needs to be changed (e.g. product price).

Since transactional order data contains new orders that need to be transferred from server to the client as well as processed orders that need to be transferred from the client to the server it needs two-way synchronization.

Transactional customer data contains the Spaza Shop information like Spaza name, location and phone number. This information is synchronized in both directions between client and server since the Spaza Shop data could be changed from several nodes (Infopreneur™ and Admin).

3) Synchronization Conflict Management

In a distributed offline scenario like described above the system only is able to ensure consistency during the synchronization process. Only then the system could evaluate and compare client and server side information. Since there are concurrent users (Infopreneurs™ and Administrators) working on the distributed data offline, the BBA provides synchronization functionality using order states and user rights to cope with possible conflicts.

4) Disconnection during Synchronization

The synchronization process is subdivided into several steps. If one of these steps could not be executed e.g. because of connection loss, the error is handled and all previous steps are rolled back. A normal synchronization process (upload from client to server) involves the following steps: (1) Export the information that is not in sync into individual SQL dump files (CVS format). (2) Concatenate the SQL dump files into one CSV file. With the small amount of data using plain text in the CSV file instead of XML or JSON improves the compression by factor 3. (3) ZIP the dump file. (4) Upload the ZIP file via FTP to the server. (5) Adapt the status of the synchronized data in the local and remote database. The download part of the synchronization process follows the same steps but includes a two-way-handshake in the end to confirm the successful synchronization on client and server side.

VII. EVALUATION

As described in the Methodology section, the evaluation combined field studies, workshops, open and expert interviews, questionnaires and system log file analysis. In addition to these well known concepts we've used the help of the Infopreneurs™ to overcome the language and cultural gap between us, as foreign researchers, and the end users. Since he is a trusted person also living in the end user community, the answers he gathered have been much more precise and without distortion due to the fact of people not telling private or "bad" things to foreign researchers (see Gary Marsden's "CCC Workshop Position Paper" in [30]). People felt much more comfortable talking to him instead directly to us. In the following we summarize the evaluation results of the 8 month pilot experiment.

A. Computational Evaluation

Since the BBA needs to run in a remote environment with infrastructure impediments and low end hardware performance and stability are major requirements of an appropriate design.

1) System Performance and Stability

Both, the Desktop Client (Infopreneur™, Administrator) and server machine are low end devices. During the pilot experimentation the Desktop Client machine was a Pentium desktop PC with 512MB RAM and 40 GB HDD. The Server machine also only had 512MB RAM and several other applications running in parallel. The complete BBA Server installation needs about 300 MB HDD space and only 55 MB of memory to run smoothly. The Desktop Client installation, containing the complete Flash GIS application, Backend and Database, needs 400 MB HDD space and average of 82 MB of memory during runtime. Due to this small footprint on the machines it is possible to run the BBA on even slower machines as well while still providing a smooth system interaction. On the Desktop Client a smooth interaction was defined with an average system response time less than a second and a fluid GIS interaction. On Server side a smooth interaction was defined with an overall order placement and synchronization round trip time of less than 1 minute.

With not a single system downtime caused by internal errors over the 8 month pilot phase the BBAServer implementation has proven stability. The BBADesktopClient also performed very well with no internal error system failures but needed some remote administration caused by external errors. The cause for these errors haven't been usability errors but since the machine, on which the application was deployed, was also used for other purposes and general internet access for the local community, errors like accidentally deleted files by users, or blocked ports by virus scanner took place.

2) Synchronization Performance

In an environment with low, erratic and very expensive network connectivity the synchronization performance is a key element to run the BBA. The requirement was to allow the BBADesktopClient, which is connected to a slow GPRS network using a USB GPRS modem, to perform the synchronization in less than one minute.

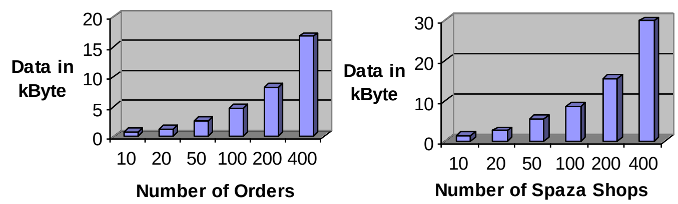

Fig. 7 Synchronization Evaluation

An average synchronization process during the pilot experiment, which only happens 1-2 times per day, contained about 20 orders and 5 Spaza Shops (new registration or master data maintenance), which results in only about 3 kByte to be synchronized (see Fig. 7). Even with a low GPRS connection this amount of data is transferred within seconds. Beside the time consumption another important factor for synchronization performance is the money spent for the data transferred. With an average price of 2.00 ZAR per 1 MB of prepaid data this results in about 0.006 Rand per synchronization. The current performance of the synchronization process also ensures efficient up-scaling of the system. With 400 orders and 400 Spaza Shop datasets per synchronization the system will synchronize about 45 kByte which costs about 0.09 ZAR.

B. Usability Evaluation

To evaluate the usability of the BBA we concentrated on the evaluation of the Desktop and mobile client application that have been built on top of the architecture and thus will rate the appropriateness of the architecture concept itself. The system usability was evaluated using direct observations (screen cam, 2 observing researchers, video recording), questionnaires, workshops and system log file analysis.

1) Technology acceptance

The acceptance of a new technology tells much about its appropriateness and usability. In the early requirements analysis we discovered that 71.4% of all Spaza Shop owners own a cell phone, and the ones that don't have one themselves know somebody with a phone (mostly younger family members). During the evaluation 66.7% said that they have no difficulties using the SMS system. The others said that if they don't know how to place the order they let it be done by their children. Another indicator for technology acceptance is the marketing amongst participants and other community members. 66.7% of the participating end users recommended the system to others which led to new pilot experiment participants towards the end of the 8 months.

2) System Benefits

To evaluate if the system provides benefit to the end users, the system usage is a good factor. 76.2% of the participating end users actively used the SMS ordering system. And 61.9% out of them do it because it offers monetary benefits to them. Another reason is "because it ensures delivery"—formerly trucks passed the shops if the owner was not there at the right time. Overall 90.5% see an added value of using the system.

3) Usability

Desktop application: The usability of the Infopreneur™ Desktop application has been evaluated in two steps. First, in the beginning of the pilot phase during direct observations (50 min sessions with a 1 minutes introduction, screen cam, usability expert observing and logging user behavior) using a structured, scenario based task. And secondly, in the end of the pilot phase using the system log files and continuous feedback gathered from the Infopreneur™. Since the system has been designed together with the end user, the Infopreneur™, the usability results have been very positive and didn't show any major design fault.

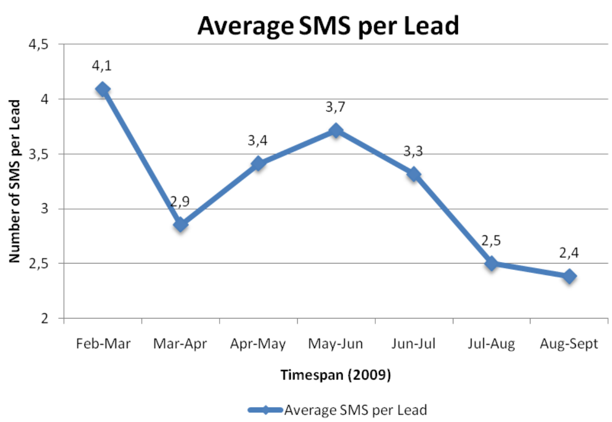

Fig. 8 Mobile SMS ordering system usability evaluation during pilot runtime. 2.0 SMS per Lead would be 0% error rate.

Mobile Application: Beside SMS application usability evaluations during workshops and questionnaires the most relevant evaluation technique was the log files of the server machine. The log files have been used to evaluate the mobile client usage profile. In the beginning of the pilot the users had difficulties using the correct format of the SMS message to order the products. This resulted in a high error rate at the SMS order placement. With additional trainings and field visits (during March/April and end of May) this was improved continuously and reached very low input error rates in the end of the pilot phase (see Fig. 8). A major improvement was realized due to the switch from the former English system language to the local language in July.

VIII. Replication

The implementation of the BBA in the Sekhukhune RLL has shown its applicability to provide the ICT architecture for a procurement use case in rural South Africa. To improve its generalizability the BBA currently is replicated in an agricultural use case in rural Ghana. The BBA will be used to support a use case with several mobile client technologies, Desktop (offline/online) applications and a central server. The methodology used during the replication follows the same approach as in the Sekhukhune RLL—a user centered approach in a real use case [23]. The requirements already have been gathered during three field studies and the first prototypes will be evaluated within end user workshops on site beginning of August 2010. The results gathered during the replication will be used to improve the BBA and strengthen its generic approach being an architecture providing ICT access for use cases in rural areas of developing countries.

IX. Conclusion and Future Work

In this paper we present the concept and a successful implementation of the Bottom Billion Architecture (BBA), a software architecture providing access to ICT in rural areas of developing countries. It is a step forward moving from use case specific point solutions supporting only one specific device, application or data channel towards a generic framework supporting several kinds of hardware (Desktop PCs, Smartphones, Feature and Non-Feature Phones), applications (native, Java, Mobile Web) and data channels (voice, signalling, data). Its main advantage compared with existing architectures for rural developing areas is the flexibility and extensibility to be used in a variety of different use case scenarios instead of being narrowed to one specific use case setup. We have proven the applicability and appropriateness of the BBA during an 8 month pilot experiment in the Sekhukhune RLL. Now we will replicate the BBA in another use case where we will use the evaluation results as input for further research on improved usability.

SMS applications currently are still the most reliable and widespread solution to provide basic mobile services to the under-privileged populations of developing countries and have provided the prove of concept of using the mobile phone to bridge the Digital Divide. However SMS will not allow a large scale, low cost development and deployment of services in future. This goal can only be reached using more advanced technologies like web technology on mobile phones (Mobile Web). But to make the Mobile Web usable and relevant to the Next Billion consumers major adaptations are required. This mostly concerns the interfaces between computer illiterate end users and the applications (mobile browser, Web content and applications) [31]. Of course, to use the Mobile Web technologies in rural areas of developing countries the available infrastructure needs to be improved. But since network coverage and speed are improving very fast, also bandwidth cost will decrease and make better network access affordable for the Next Billion. Also more advanced mobile devices like feature phones or Smartphones will become affordable within the next years. To be able to use and benefit from the improved technology and infrastructure availability, research on this topic needs to start now to have solutions available within the next 3-5 years. Fig. 9 depicts our current Mobile Web prototype of the Sekhukhune RLL Procurement Use Case which is also based on the BBA concepts. Due to the BBA web service interface the Mobile Web prototype is able to be plugged into the existing procurement system.

Fig. 9 Mobile Web Prototype

During the BBA replication in the agricultural use case in Ghana we will concentrate on improvements regarding generalizability of the BBA concept and usability of the individual components. To improve the generalizability we will try to build the individual architecture components in a reusable way. Any improvement of the architecture will be integrated in the Sekhukhune RLL scenario as well to ensure that we don't narrow the scope of the architecture with limitations of the new use case. To improve the usability we will use the input of the Sekhukhune RLL evaluations and concentrate on more advanced client technologies. This will involve more advanced phones (feature phones and Smartphones) to create usability improved UI's while still mostly rely on SMS as data channel.

REFERENCES

[1] "The Real Digital Divide," *The Economist print edition*, pp. 1-3.
[2] ITU, *Measuring the Information Society - The ICT Development Index*, 2009.
[3] ITU, *Measuring the Information Society*, Geneva: International Telecommunication Union, 2010.
[4] J. Doerflinger, C. Friedland, C. Merz, and R. De Louw, "Requirements of a Mobile Procurement Framework for Rural South Africa," *6th International Conference on Mobile Technology, Application & Systems - Mobility 2009*, Nice, France: ACM, 2009, pp. 1-4.
[5] J.V. Rensburg, D. Smit, and A. Veldsman, "System of Innovation and micro enterprises in real world rural SADC: An overview of collaborative SMME incubation in the Rural Living Lab of Sekhukhune," 2007.
[6] R. Kumar, "eChoupals: A study on the financial sustainability of village internet centers in rural Madhya Pradesh," *Information Technologies and International Development*, vol. 2, 2004, pp. 45-74.
[7] K. Toyama and R. Kuriyan, "Review of Research on Rural PC Kiosks," *Microsoft Research India*, 2007.
[8] B.V. Ratnam, P.K. Reddy, and G.S. Reddy, "eSagu 1 : An IT based personalized agricultural extension system prototype – analysis of 51 Farmers ' case studies," *International Journal of Education and Development using Information and Communication Technology*, vol. 2, 2006, pp. 79-94.
[9] R. Veeraraghavan, G. Singh, and K. Toyama, "Kiosk Usage Measurement using a Software Logging Tool," *International Conference on Information & Communication Technologies for Development*, Berkeley, CA: 2006.
[10] R. Veeraraghavan, N. Yasodhar, and K. Toyama, "Warana Unwired: Replacing PCs with Mobile Phones in a Rural Sugarcane Co-operative," *Proceedings of ICTD*, 2007.
[11] T.S. Parikh and E.D. Lazowska, "Designing an Architecture for Delivering Mobile Information Services to the Rural Developing World," *Proceedings of the 15th international conference on world wide web, Edinburgh, Scotland*, New York: ACM Press, 2006, pp. 791-800.
[12] P. Javid and T. Parikh, "Augmenting Rural Supply Chains with a Location-Enhanced Mobile Information System," *International Conference on Information and Communication Technologies and Development - ICTD 2006*, Washington: Ieee, 2006, pp. 110-119.
[13] K. Ramamritham and A. Bahuman, "aAqua: a database-backended multilingual, multimedia community forum," *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, Chicago, IL, USA: ACM, 2006, pp. 784-786.
[14] "txtEagle," *http://txteagle.com/. Last accessed: 23 July 2010*.
[15] "FrontlineSMS," *http://www.frontlinesms.com/. Last accessed: 23 July 2010*.
[16] "Nokia Life Tools," *http://europe.nokia.com/services-and-apps/nokia-life-tools/main. Last accessed: 23 July 2010*.
[17] T. Gross, C. Beckmann, and M. Schirmer, "The PPPSpace: Innovative Concepts for Permanent Capturing, Persistent Storing, and Parallel Processing and Distributing Events," *Eighteenth Euromicro Conference on Parallel, Distributed, and Network-Based Processing - PDP 2010*, Los Alamitos: IEEE Computer Society Press, 2010, pp. 359-366.
[18] "Collaboration@Rural," *http://www.c-rural.net/, Last accessed on 23 July 2010*.
[19] Bureau of Market Research, "Characteristics of Spaza Retailers: Evidence from National Survey. Report No. 305," 2002.
[20] J.V. Rensburg, A. Veldsman, and M. Jenkins, "From Technologists to Social Enterprise Developers: Our Journey as ICT for development practitioners in Southern Africa," *Information Technology for Development*, 2008.
[21] C. Merz, "Incubating Micro Enterprises in Rural South Africa - The Use Case of Virtual Buying Cooperatives," *Münchner Kreis*, 2009, p. 17.
[22] R. Heeks, *The ICT4D 2.0 Manifesto: Where Next for ICTs and International Development?*, Manchester: 2009.
[23] J. Doerflinger and T. Gross, "Technical ICTD – A User Centered Lifecycle," *IFIP International Conference on Wireless Communications and Information Technology in Developing Countries - WCITD 2010*, A. Pont, G. Pujolle, and S. Raghavan, Brisbane: unpublished, 2010, pp. 70-81.
[24] "Mobile Living Labs 09: Methods and Tools for Evaluation in the Wild," *Mobile Living Labs 09 at Mobile HCI 2009 Conference*, Bonn: 2009, p. 48.
[25] J. Kjeldskov and C. Graham, "A review of mobile HCI research methods," *Lecture Notes in Computer Science*, 2003.
[26] C. Putnam, E. Rose, E.J. Johnson, and B. Kolko, "Adapting User-Centered Design Methods to Design for Diverse Populations," *Information Technologies and International Development*, vol. 5, 2009, pp. 51-73.
[27] J. Abraham, D.R. Dean, and A. Subramanian, "Ringing In the Next Billion Mobile Consumers - A Road Map for Accelerating Telecom Growth in India," *Boston Consulting Group*, vol. 1, 2007, p. 37.
[28] N. Nokia Siemens, "Nokia Siemens Networks Village Connection - bringing the benefits of affordable mobile access to rural communities," *Strategies*, 2008.
[29] W3C Interest Group, "W3C Mobile Web for Social Development Roadmap (MW4D)," 2009.
[30] T. Parikh, "Proceedings of the CCC Workshop on Computer Science and Global Development," *Proceedings of the CCC Workshop on Computer Science and Global Development*, Berkeley: 2009, pp. 1-92.
[31] S. Boyera, "White Paper on Mobile Web for Social Development," *W3C Social Development*, 2009, p. 5.