Feasibility Study of a Mobile Payment System on Kasadaka: a sustainable voice service platform

Sharon Afua Grundmann
Supervisor: Victor de Boer

Vrije Universiteit Amsterdam
s.a.grundmann@student.vu.nl

Abstract. The increasing use of mobile phones and the Internet in recent years have led to the rise of web-based services. Many people in rural areas however, are unable to make use of these services due to lack of internet connection and other factors. Kasadaka is a platform that provides web-like voice-based services hosted locally on low resource hardware. As part of a larger investigation into the economic sustainability of the platform, this paper discusses how Kasadaka can be extended to include a mobile payment system. We present the design and partial implementation of this solution and validate it using an existing voice service. We conclude that a mobile payment system on Kasadaka is feasible using SMS and USSD messages and database management.

1 Introduction

During the last few decades, the ICT sector in Africa has evolved rapidly due to increasing use of mobile phones and the Internet. At the end of 2016, there were 420 million unique mobile subscribers in Sub-Saharan Africa making this region account for nearly 10% of the global subscriber base (GSMA, 2017). While this seems promising, there is also a significant divide between rural and urban use of mobile phones and other forms of technology. The field of ICT4D[1] aims to bridge this *digital divide* by using technologies that are already available and have a large user base in the developing world to create information services that are fully adapted to the local context (Bon, 2016).

In this paper, we research how an existing ICT4D project called Kasadaka can be extended to include a mobile payment system to improve its economic sustainability. We make use of Mobile Money, a popular mobile payment service enabling financial inclusion in Sub-Saharan Africa through the use of mobile technology in realizing this goal.

[1] Information and Communications Technology for Development

1.1 Context

Kasadaka Kasadaka[2] is a sustainable platform that provides web-like voice services hosted locally over an available mobile telephone network on low resource hardware, a Raspberry Pi. The operating system it runs on is Raspbian, a Linux-based operating system which supports a wide range of open-source software, making it easy and affordable to use. Kasadaka accesses the available mobile telephone network through a 3G USB modem connected to it. This together with several software applications work to provide the voice service functionality.

Some voice services hosted on Kasadaka include a market information distribution system (RadioMarché), a village citizen journalism platform (Foroba Blon) and a knowledge-based veterinary system (DigiVet). All these services have proven based on past research to be beneficial to the communities they have been deployed in. (Gyan et al., 2013; Lô & Blankendaal, 2016)

Foroba Blon Developed in 2012 by the ICT4D team at the Vrije Universiteit, Foroba Blon is a journalism platform that allows radio stations in rural Mali to better manage the reception and publication of content. A radio station runs Foroba Blon which is accessed by calling the mobile phone number assigned to the hosted Kasadaka or through a web page. End users of Foroba Blon also include amateur journalists who call Kasadaka to leave news reports and local radio listeners who call to contribute to radio shows or announce events, like weddings, taking place within the community. The voice messages are stored as audio files in the Foroba Blon data store, together with meta-information like the date and time of the call, the length of the phone call and the phone number of the caller. In case a listener does not own a phone, a local journalist's phone is used to place the call to Kasadaka. These advertisements are paid for through local journalists who collect payments on behalf of the radio station. This income enables the radio station in sustaining Kasadaka and the Foroba Blon service.

Mobile Money Mobile banking has come in handy in many parts of the world with little or no development, especially in remote and rural areas. Since 2005, mobile financial applications known as "Mobile Money" have emerged in a variety of developing countries. In Sub-Saharan Africa, the situation is no different with the demand for mobile banking and payment permanently increasing (Tchouassi, 2012). The use of Mobile Money services as a means of payment is as a result of its advantages over conventional payment methods like convenience, reliability, speed, flexibility and affordability (Davidson & McCarty, 2011).

In order to make use of Mobile Money, mobile phone users have to create an e-wallet registered to their SIM card at designated agent locations into which electronic cash can be deposited and used for the payment of goods and services. The payment system offers a web interface and a basic interface on mobile phones to complete specific commands together with in-person agent locations who handle the transition from electronic cash to physical money and vice versa.

[2] `https://kasadaka.com/`

In Ghana, the functionality of the mobile money application interface is similar across telecommunication companies. The interface is a basic menu based on the Unstructured Supplementary Service Data (USSD) protocol. Mobile Money users are able to interact with the USSD interface and telecommunication company to send electronic cash from their e-wallet to another and even to non-users of Mobile Money. Additionally, Short Message Service (SMS) messages are sent by the telecommunication company to both sender and receiver of a payment to acknowledge the transaction. The Mobile Money SMS messages contain meta-information about the transaction such as the transaction ID, date and time of transaction, the sender or receiver's name, amount sent or received, reference and the current balance of the e-wallet.

Unstructured Supplementary Service Data (USSD) As mentioned above, the Mobile Money interface is based on USSD. USSD is a technology built into the GSM standard to support the transmission of information over the signaling channels of the GSM network (Téllez & Zeadally, 2012). USSD messages are up to 182 alphanumeric characters long and are sent using a real-time connection created by a USSD session. The connection remains open, allowing a two-way exchange of a sequence of data. This makes USSD a more responsive way of interacting with the telecommunication company instead of SMS.

Short Message Service (SMS) SMS is a service that allows mobile devices and other network-connected devices to exchange short text messages with a maximum length of 160 characters. Unlike USSD which is session-oriented, SMS is a store-and-forward technology. SMS messages are sent to a Short Message Service Center (SMSC) which sends the message to the recipient. If the recipient is not reachable, the SMSC queues the message for a later retry.

1.2 Research Goal

Using the Foroba Blon case as motivation, the main objective of this paper is to research how to implement a mobile payment system on Kasadaka such that it can receive payments from customers for instance, the local radio listeners and also make payments to other stakeholders like the amateur journalists if possible. Since the Mobile Money service previously discussed provides similar functionalities, we formulate the following research questions:

- How do we integrate Mobile Money into Kasadaka to create a mobile payment system?
- How sustainable is this solution?

This paper is structured as follows: section 2 discusses applications related to the context above. Section 3 states the requirements of a mobile payment system on Kasadaka and section 4 explains the challenges involved in meeting these requirements and presents a prototype. We evaluate it in section 5 and discuss the limitations, possible future work and conclude in the next chapters.

2 Related Work

This section first explains the technologies that enable the voice service functionality on Kasadaka. This makes up the foundation on which the mobile payment system must be implemented. Other projects discussed in this section include applications related to mobile payment and the Mobile Money service. Because this field of development is fairly new, most applications available are commercial and web-based.

2.1 Kasadaka voice service development kit (VSDK)

Kasadaka makes use of Asterisk[3], a software implementation of a telephone private branch exchange. In its case, an extension of Asterisk called `chan_dongle` available for GSM and 3G dongles routes incoming calls to a VoiceXML interpreter using a dial plan. In addition to phone call handling, `chan_dongle` supports the sending and receiving of SMS and USSD messages.

When a call is redirected to VoiceXML, a pre-configured URL is passed on to VXI, the VoiceXML interpreter used in Kasadaka, which loads and 'displays' the document to the user. VoiceXML files, which are comparable to HTTP files are generated by an open-source web-framework called Django. Django also has an administrator web interface that allows the management of the data within the application. When an element in the voice application is requested by a user during a phone call, the VoiceXML interpreter (VXI) requests the element through HTTP. Django then retrieves the information about this element from the database, and uses a view to 'render' the element in VoiceXML (Baart, 2017).

2.2 ECOPAY platform

ECOPAY (Mbinkeu, 2012) is a payment application that builds on the Mobile Money service in Cameroon. Built using PHP and Java frameworks, the platform facilitates the payment of goods and services using the users' Mobile Money wallets. In order to pay using ECOPAY, a merchant inserts the customer's mobile phone number and amount into the application and the customer validates the transaction using a PIN. Using a JSON call between the merchant and the client, both accounts are updated and an additional result of the transaction is sent via SMS to the customer. Additionally, the ECOPAY application allows users to withdraw money from their Mobile Money wallets through ATM's using their PIN and a sequence of random numbers determined by the user.

2.3 Hubtel merchant account API

The Hubtel[4] merchant account API is an interface that allows merchant web applications to accept online payments for goods and services into a specified

[3] `https://www.asterisk.org/`
[4] `https://hubtel.com`

Hubtel account. Receiving money through Mobile Money involves the application making a HTTP POST request to the merchant account API to debit a Mobile Money user's wallet. The user confirms or rejects this transaction through a USSD message and the response is forwarded to the application callback URL as an HTTP POST payload. Finally, the customer receives a SMS notification of the transaction. The API supports other transactions like sending money to customers and recalling old payments.

3 Requirements

The applications mentioned in 2.2 and 2.3 rely on an Internet connection in providing the payment functionality. This does not fit the context of Kasadaka and ICT4D. Most areas where Kasadaka is deployed in have little to no Internet connection and even if they did, the users of Kasadaka would not be able to afford it. Such limitations have to be considered when implementing a mobile payment system on Kasadaka. The requirements of the mobile payment system that need to be met are stated below.

Functional Requirements:

- Kasadaka must store transactions received via SMS.
- Kasadaka must make payments via USSD.
- Kasadaka users must be able to view transactions.
- Kasadaka should store SMS messages.
- The payment system could be accessible through VoiceXML.

Non-functional Requirements:

- The system must run without an internet connection.
- The system must run on the existing Kasadaka VSDK platform.

4 Research Challenges

To develop a working prototype of a mobile payment system that meets the requirements in 3, a number of challenges need to be addressed. In this section, we outline the different challenges, namely: configuring Asterisk to handle SMS and USSD messages, extending Kasadaka VSDK's functionalities, parsing transactions and the approach taken in solving them.

4.1 SMS and USSD messaging

The existing Kasadaka platform does not make use of SMS and USSD messages in hosting voice services and therefore, there is no proper handling of incoming and outgoing messages. Implementing this functionality has to do with modifying the behavior of the dialplan[5] of the Asterisk `chan_dongle` extension. Some

[5] in extensions.conf

applications capable of handling SMS and USSD messages include the Asterisk SMS application, Asterisk Gateway Interface (AGI) and Asterisk Manager Interface (AMI).

The SMS application[6] allows the sending and receiving of text messages. Together with the Asterisk Manager Interface or a call file, it can send text messages to SMS capable phones and also accept and store received SMS messages. Although this application supports SMS messaging, it is only available for use in certain countries and does not work over compressed links.

The AGI[7] however, is a suitable alternative to the SMS application. It provides an interface between the Asterisk dialplan and an external program that wants to manipulate a channel in the dialplan. Using the AGI allows programs written in other languages like Python, Java and PHP to control the channel and only return when an action is completed. Initiating a call without being started through dialplan can be done using the Asterisk Manager Interface or call files.

The Asterisk Manager Interface[8] allows a client program to connect to an Asterisk instance and issue commands or read events over a TCP/IP stream. Before issuing commands to Asterisk, a manager session must be established by an authenticated user. Packets can be transmitted in either direction with "action" signifying commands from the client to Asterisk and "event" or "response" from Asterisk to the client.

In the mobile payment prototype, the Asterisk Gateway Interface is used to handle incoming SMS and USSD messages and although outgoing messages could be sent using the AGI together with the AMI, this functionality is not included in the prototype.

4.2 Extending Kasadaka VSDK

In addition to storing voice services in the VSDK application, transactions need to be stored as is required in a mobile payment system. Users must be able to view incoming transactions (deposits) and outgoing transactions (withdrawals).

The web development framework used to build the VSDK application is Django. It follows a Model-View-Controller methodology where a model is a representation of the data used in the application, a view is the presentation layer of the model and a controller controls the flow of information between the model and the view. This design pattern allows applications to be easily modified.

New models were created to support the storage of text messages and transactions which could also be accessed through the Django administrator web interface. In the 'Text Message' model, raw SMS messages together with the caller ID of the sender are stored in the database while the 'Transaction' model contains meta-information found in Mobile Money SMS messages.

[6] `https://wiki.asterisk.org/wiki/display/AST/SMS`
[7] `https://www.voip-info.org/asterisk-agi/`
[8] `https://www.voip-info.org/asterisk-manager-api`

4.3 Transaction parsing

Mobile Money SMS messages received by the Asterisk Gateway Interface need to be stored in the VSDK database. The approach taken by the prototype is to parse such a message and store its contents using a Python script.

Firstly, the caller ID of the sender is checked to ensure that the transaction is legitimate. In a real use case scenario, the caller ID must match that of the telecommunication company providing the Mobile Money service. After the sender of the SMS message is validated, the message body is parsed using regular expression pattern matching to extract the relevant meta-information about the transaction. This includes the transaction ID, date and time of transaction, the sender or receiver's name, the amount sent or received and the reference. An additional label is added indicating whether the transaction is an incoming transaction (deposit) or an outgoing transaction (withdrawal). The data is finally posted to the Django Transaction and Text Message models so that it can later be viewed by a Kasadaka user through the administrator web interface.

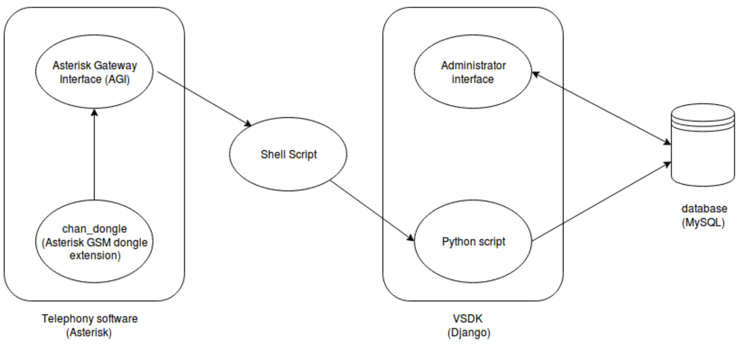

Fig. 1. System architecture of the mobile payment system prototype

When a SMS message is sent, the system executes the following steps (Fig. 1):

1. Asterisk receives the SMS from the GSM dongle and connects it to AGI.
2. AGI runs a shell script that forwards the SMS message together with the caller ID to a Python script.
3. The Python script sets up the VSDK application using Django and imports the Text Message and Transaction models.
4. The Python script saves the SMS message and caller ID as a new Text Message object in the application's database.
5. The Python script checks the caller ID and if it is the appropriate sender, it parses the SMS as a transaction.

6. The Python script extracts the relevant information from the message using pattern matching and saves the data as a new Transaction object in the database.

The source code of the prototype is available on GitHub[9] as well as a video[10] showing how the prototype of the mobile payment system on Kasadaka works.

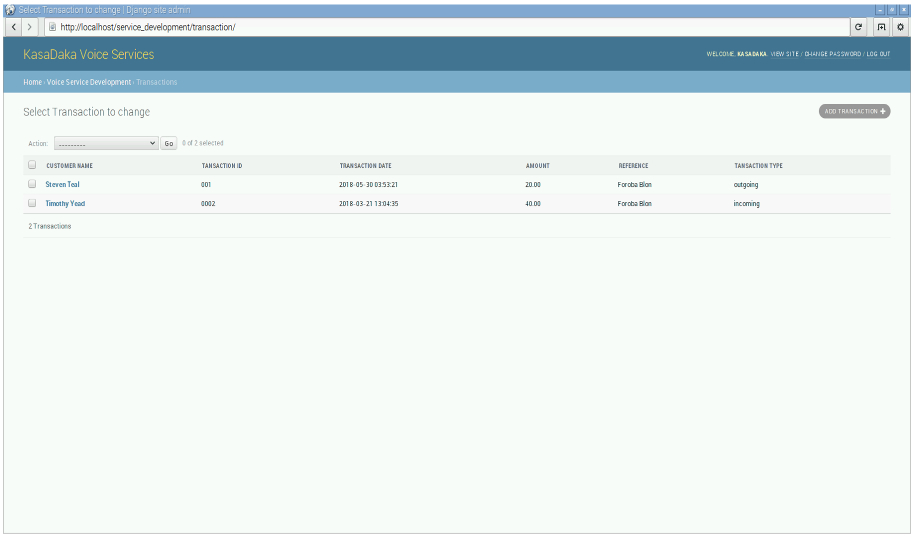

Fig. 2. Screenshot of the Transaction page on Kasadaka VSDK

5 Evaluation

A mobile payment system prototype integrated into Kasadaka was used in an experimental setting conducted in Amsterdam to evaluate its performance. A true evaluation would require the use of Kasadaka in a rural Sub-Saharan setting where it has access to a telecommunication network that offers the Mobile Money service. However, this was not possible within the scope of this research project. Instead, a mobile phone containing a Dutch SIM card was used to send SMS messages to Kasadaka. The format of the text messages resembled Mobile Money SMS messages sent by telecommunication companies in Ghana. This set up mimics the role of the telecommunication company in a real use case scenario.

[9] `https://github.com/shayorshay/KasaDaka-VSDK`
[10] `https://goo.gl/MnVqbi`

5.1 Performance

Usually when a SMS message longer than 160 characters is sent, the telecommunication company splits the message into chunks so that they can be sent over the channel and rendered properly to the recipient. Although the splitting happens, Asterisk is not able to properly render the message resulting in unreadable chunks of data. Because of this, the length of SMS messages used for testing were set within the limit of a standard SMS message and this must be resolved when implementing the prototype for use in the real world.

A major part of the experiment involved sending Kasadaka multiple SMS messages containing payment information to see if all transactions were properly parsed and stored in the database. As it turns out, Kasadaka is able to process and store transactions as long as the format of the text messages matches the required pattern which was the expected behavior. Although this works for testing purposes, it means that before Kasadaka is used in a real use case scenario, the format of the Mobile Money SMS messages must be checked because they may differ from one telecommunication company to another.

5.2 Usability

With regards to usability, two main tests were performed. One had to do with the behavior of Kasadaka when handling a phone call and SMS message simultaneously and the other, the handling of multiple messages at the same time. This is relevant for real use case scenarios because it is expected that users may send multiple payments to Kasadaka while others make use of the voice services.

For this experiment, two mobile phones were used, one to call and interact with Kasadaka and the other to send transaction messages. This test was successful. Kasadaka can handle an incoming SMS and call at the same time which means that there is no need to allocate distinct time frames for users who want to pay Kasadaka and those that want to call the hosted voice services. The other test was also performed by using two mobile phones to send SMS messages to Kasadaka simultaneously. This also worked well, possibly because the shell script can run several instances of the Python script at the same time.

5.3 Reliability

How reliable is the mobile payment prototype? What happens if a payment is made to Kasadaka while its turned off? To answer these questions, several SMS messages were sent to Kasadaka while it was off. When Kasadaka was turned back on, all the SMS messages were parsed and the transactions were stored in the database. The explanation for this is that SMS is a store-and-forward technology as mentioned earlier in this paper. Because of this, Kasadaka can be turned off during the night to save power consumption and all transactions made then can be retrieved when it is turned back on the next morning. The disadvantage of this reliance on the telecommunication company is that transactions cannot be made when there is a phone network failure which occurs quite often in Sub-Saharan Africa.

5.4 Foroba Blon use case

The addition of a mobile payment system to Kasadaka changes the way users interact with it. A new Kasadaka Customer model based on the e3 value model[11] was created to visualize how actors in the Foroba Blon model interact with each other. The new model contains five actors including, a customer in a rural community, radio station, phone company and voice or web service provider as well as a bank. The bank works in conjunction with the phone company to provide the Mobile Money service (MM). This model shows the scenario in which the customer does not rely on a local journalist for phone connection.

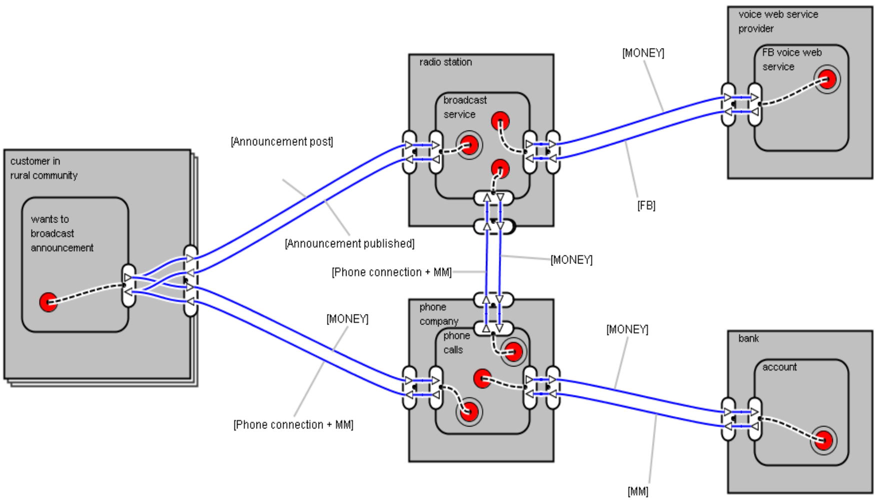

Fig. 3. New Kasadaka Customer Model

Just like in the old Kasadaka Customer Model (Daoudi, 2017) scenario, the customer in the new model scenario is a person living in a rural community in Mali and has the need to announce a message to other people in the area. The customer wants for example to broadcast a message on the radio because there is an upcoming wedding. By reporting this on the local community radio, the message will be reached by people in the surrounding areas. Instead of paying the village reporter for the message to be broadcasted on the radio, the customer pays Kasadaka directly.

The Foroba Blon service would work as follows assuming both the customer and Kasadaka have Mobile Money e-wallets:

[11] `http://e3value.few.vu.nl/`

1. The customer calls the Kasadaka and places an advertisement using Foroba Blon.
2. At the end of the call, Kasadaka informs the user of the estimated cost based on the length of the message.
3. The user transfers this amount to the e-wallet of Kasadaka using the USSD interface on his mobile phone.
4. The telecommunication company of the user acknowledges this transaction by sending SMS messages to both Kasadaka and the customer.
5. When Kasadaka receives this message, it parses and stores the transaction in the database.
6. The radio journalist later accesses the web interface and after seeing that the message has been paid for, broadcasts the advertisement on the radio.

6 Discussion and Future Work

The prototype presented in this paper meets all functional and non-functional requirements stated earlier expect one, that is, Kasadaka cannot make payments via USSD. However, the technologies required to implement this functionality has been presented.

Another problem not tackled in this research is how users access the Mobile Money service on their mobile phones. We assume that a Kasadaka user is capable of making payments using Mobile Money. This is however not a true reflection of the typical Kasadaka user who is an illiterate farmer in a rural Sub-Saharan village who may or may not own a mobile phone. A naive but practical solution to this problem is for this group of people to make use of Mobile Money agents. A better solution suggested as future work in this field of research is to make the USSD interface of Mobile Money service accessible as a voice service or to integrate VoiceXML into the mobile payment system on Kasadaka.

7 Conclusion

In this paper, we introduced the Kasadaka platform and Mobile Money, a service allowing mobile phone users to send and receive money through SMS, USSD and web applications in Sub-Saharan Africa. We investigated a number of challenges surrounding implementing a mobile payment system on Kasadaka and presented a prototype as a solution. The suggested solution is to use the AGI to handle incoming SMS and USSD messages and the AMI together with the AGI to handle outgoing USSD messages. Messages can be parsed using regular expression pattern matching and stored in the Kasadaka database using a Python script. Although this approach to implementing a mobile payment system on Kasadaka works, it has some limitations.

To conclude, it is feasible to implement a mobile payment system on Kasadaka using SMS and USSD messages together with a database management system. Because SMS and USSD messaging is available to all mobile phone users and significantly cheap, the solution presented in this paper fits the context of ICT4D and is sustainable.

References

Baart, A. (2017). Kasadaka: a sustainable voice-service platform. *Master Thesis Vrije Universiteit Amsterdam.*

Bon, A. (2016). Ict4d 3.0: An adaptive, user-centred approach to innovation for development..

Daoudi, J. (2017). Economic sustainability of ict services for the rural poor in sub-sahara africa.

Davidson, N., & McCarty, Y. (2011). Driving customer usage of mobile money for the unbanked. *London, UK: GSM Association (GSMA).*

GSMA. (2017). *The mobile economy: Sub-saharan africa 2017.*

Gyan, N. B., De Boer, V., Bon, A., Van Aart, C., Akkermans, H., Boyera, S., ... Allen, M. (2013). Voice-based web access in rural africa. In *Proceedings of the 5th annual acm web science conference* (pp. 122–131).

Lô, G., & Blankendaal, R. (2016). Digivet: a knowledge-based veterinary system for rural farmers in north-ghana.

Mbinkeu, R. C. N. (2012). Analysis and design of mobile payment platform in african context. In *International conference on e-infrastructure and e-services for developing countries* (pp. 143–152).

Tchouassi, G. (2012). Can mobile phones really work to extend banking services to the unbanked? empirical lessons from selected sub-saharan africa countries. *International Journal of Developing Societies*, *1*(2), 70–81.

Téllez, J., & Zeadally, S. (2012). Mobile payment systems.