Towards An Interoperable Mobile Wallet Service

Pradipta De†, Kuntal Dey*, Vinod Mankar*, Sougata Mukherjea*
pradipta.de@sunykorea.ac.kr, {kuntadey,vinomank,smukherj}@in.ibm.com
*IBM Research, New Delhi, India
†CEWIT Korea and SUNY Korea

***Abstract*—Advancement in mobile technology, coupled with a market wide growth in mobile subscriber base, is encouraging financial, telecom and even third party operators to offer mobile payment services. In presence of a plethora of such services, an imminent problem is lack of interoperability across them. In this work, we address the challenge of interoperability by designing a wallet service built upon a concept of token. A token encapsulates a payment instruction, which can be of different types, such as instant, dated or installment payment. A token is represented by a Info-Matrix code, viz. Quick-Response (QR) code, thereby bringing in additional advantages in terms of user interactions with the wallet service. We present the lifecycle management of a token, beginning with token generation on user request, transfer of token, and acknowledged receipt and encashment of a token. This work presents an architecture for interoperable wallet service, along with the implementation of a system that demonstrates the use of QR-codes as standardized tokens for transactions.**

I. Introduction

Phenomenal advances in mobile technology is leading to a renewed interest in mobile enabled financial services. Several companies worldwide have launched mobile payment service. At present there are 124 mobile payment systems across the globe [7]. With third party players, like Google and Paypal, entering the mobile wallet market, wallet-as-a-service will gain more prominence. However, as none of the payment systems interoperate among them, it introduces a new inconvenience for the user.

Current mobile money deployments act as 'walled gardens', where a user cannot transfer money to another user or a merchant who is not registered with the same provider. In countries, like Tanzania, where there are competing mobile payment offerings from four different telecom vendors, this leads to hindrance in executing financial transactions seamlessly. From service providers' viewpoint it may appear that introducing interoperable wallets may negatively impact their sales. However, in an interoperable setting, users have lower incentive to subscribe to different providers, thereby reducing possible customer churn, and also potentially increasing transaction volume.

Designing an interoperable money exchange ecosystem allowing users to use wallets and interoperate seamlessly with traditional money exchange presents the following set of challenges.

- **Payment artifact:** Interoperability mandates a payment artifact for money exchange representation that needs to capture details about the transactions and the parties involved. The representation needs to be amenable to electronic as well as traditional systems in terms of creation/procurement, transfer and encashment. One can think of cash notes (dollar bills) and cheques as examples of artifacts.
- **Standardization:** There needs to exist a standardization for payment instruction exchange and the process of settlement. Further, a state of each transaction needs to be maintained by all the interoperating parties. and tese states need to be consistent.
- **Scalability:** Interoperability requires designing an ecosystem where new financial players (e.g. wallets) can easily enter the process and start operating irrespective of the other parties. It needs to scale linearly so that a new financial player does not require registering with any other (existing) financial player to start operating, by complying with the platform protocol.
- **Modality & User Control:** Users need to be able to pay with all the primary money exchange modes, which includes instant exchange equivalent to cash transactions, dated exchanges equivalent to cheque payments and multi-transaction exchanges such as installment settlements. In addition to it, receiver must have the flexibility of withdrawing within a range of permissible dates, such as encashing a dated cheque on a preferred date within a specified time range rather than immediately after the cheque becomes valid.
- **Usability:** Users need to be able to use the ecosystem irrespective of the device type, including smart devices with advanced capabilities and basic devices with only plain-text format support, and offline also. Ease-of-use (e.g. minimal keypress), reliablity (e.g. consistent view of money across front-end and backend), robustness (e.g. loss or damage of device should not invalidate money) and human error tolerance (e.g. typing errors) are desirable.
- **Security:** Payment solutions are unusable in absence of security - so it is important to ensure that the ecosystem is secure and fraud-proof.

In this paper, we present an architecture for interoperable wallet services that overcome the above challenges. The design involves the sender and the receiver of the money as front end parties. Each front end party will be associated with a backend wallet service capable of sending and receiving payments from other backend wallet services. The design captures the payment instructions as tokens moving across the

various platform components involved in the system.

We then implement DigiToken, a front-end application based upon this architecture, that addresses all the money exchange modalities. A token is used in this system to represent transactions objectively across all the players participating in the money exchange process. The token is modeled as a machine-readable pictorial info-matrix code - specifically, Quick-Response (QR) code. We design a scalable backend in a manner that is capable of handling all the payment modalities and works for smart as well as basic front-end devices. This enables basic phones and other devices with feature restrictions also to participate in this money exchange ecosystem and operate on each of the transaction modalities. We enable backend tracking of token lifecycle.

The system allows the sender to create and send the tokens at different times to different receivers, associating one or more validity dates and monetary values to each token. The system requires an explicit acceptance from the receiver before transferring the token to the receiver's account. Apart from instant exchange scenarios, where an acceptance of token by the receiver is the only signal to encash, we implement the encashment to be initiated on the receiver's explicit expression of intent. Using third-party security providers like Verisign, password/PIN protection and private-public key encryption can be considered for providing security.

The rest of the paper is organized as follows. In Section II different payment transaction types are presented, followed by an architecture for interoperability. The implementation details of the QR-code based tokens are presented in Section III. In Section IV, we illustrate the DigiToken application which uses the QR-coded tokens for enabling an interoperable mobile wallet system, as well as, show the pros and cons of using QR-coded tokens as the medium of exchange. We explore the state of current literature in Section V. We conclude in Section VI.

II. Transaction Types & System Architecture

In this section, different forms of payments are introduced. We also present an architecture and workflow to enable interoperability across wallet services.

A. Types of Payments

Typical wallet transactions, viewed from a user's perspectives, can be of three types. A user can make a payment to a merchant, as at a Point-of-Sales (PoS), or to another user, similar to a day-to-day cash transaction. These transactions are *instant transactions*, where the transfer must happen immediately. The second notion of transfer is captured by a dated cheque transaction, where a user marks the date of transaction. This is referred to as *dated transaction*, where the transfer is initiated on or after the specified date. The third form of transfer is *installment payments*, where on specific dates a transfer is initiated and it repeats for a pre-defined number of times. A wallet service should enable all 3 forms of transfer: instant, dated and installment.

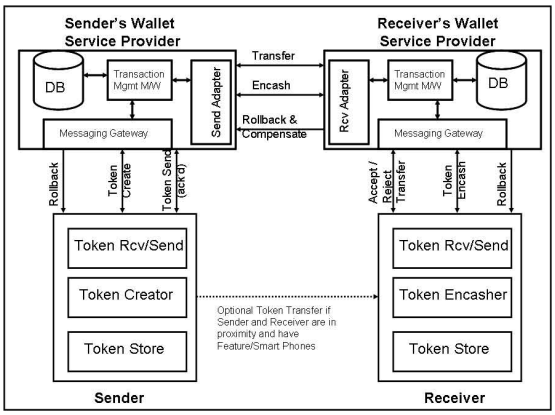

Fig. 1. Architecture for interoperability in mobile payment transactions

B. System Architecture

Wallet service has a front-end hosted on the end device, and the backend, corresponding to a front-end, is hosted by a service provider with whom the user is registered, as shown in Fig. 1. The front-end of the wallet service acts as the control point for three operations in the wallet service workflow: Token Generation, Token Transfer, and Token Encashment.

Token generation is the process where a user requests for a digital token which can act as the instrument of payment, and can be easily exchanged. The sender initiates the process of token generation by sending the required information, like value, receiver name, validity dates to the backend. From a mobile phone the request is sent to a pre-defined short-id provided by the Mobile Network Operator. Once a token is generated by the backend, a token identifier is sent to the user and can be used subsequently to trigger transfers. Alternatively, the sender can receive the pictorial QR-coded token, which can be directly scanned by receiver from sender to trigger a transfer.

Token transfer is the process to initiate the transfer of a token generated earlier to another user. When a user wants to transfer the token, the sender sends a message to a short id with the token id to instruct transfer of the token. If the receiver is registered with the same wallet service provider as sender, then a message is sent to the receiver to accept the token transferred to him. Otherwise, if the receiver is registered with a different service provider, then sender's service provider communicates with the receiver's service provider to notify the user of the token. When the receiver acknowledges the token transfer, the sender's service provider sends a copy of the token to receiver's service provider to maintain the subsequent token states.

Token encashment is the step when an available token is encashed. In case of instant token types, the encashment step is initiated as soon as the token is accepted in the token transfer process. For other token types, like dated or installment, as soon as token encashment date is reached, it is available to the receiver to encash. A receiver can view all the tokens ready for encashment, and trigger encashment, leading to actual clearance of funds from sender's wallet to receiver's wallet. The encashment step is broken into acknowledgment of token,

and encashment to avoid unwarranted transfer of funds.

C. Error Handling and Transaction Rollbacks

We model the overall process as long running transactions and treat the three steps, namely token generation, transfer, and encashment as nested inner transactions. Each transaction rollback event involves a compensating transaction, except when the sender fails to receive the token generated, and when a dated or installment token is not encashed within the specified date range.

In our system, disruption in message delivery is one of the key sources of errors in the workflow, and necessitates rollbacks. The scenarios are explained below:

i. After token generation, the message from backend to the sender failed, thereby sender not receiving the token id. In this case, the generated token is invalidated. Since the user fails to receive the token, he is expected to repeat the request for generation of the token.

ii. On transfer request by sender, the token could not be delivered to the receiver. The token is marked cancelled, and the sender notified of the token cancellation. The sender has the option to re-instantiate the token, if he wants to try the transfer again to the receiver.

iii. On receiving a token, the receiver may reject a token, if the token is wrongly delivered, or the token information, like the amount, is incorrect. The token is again invalidated, and the sender notified. The sender has the option of editing the token, and re-instantiating it.

iv. During token encashment, the clearance from the sender's wallet may fail if there is insufficient balance. In this case, the token is marked as *not encashed*. The receiver is notified of the failure, and the sender is notified of the payment failure. The token is invalidated, and must be regenerated. This can be handled like a dishonored check.

III. Implementation Details

We implemented a payment system that uses QR-code tokens as the basic unit of transaction. The front-end application, called DigiToken, enables a user to transact using tokens. In this section, we present the implementation a QR-code based token, and the methods to enable interoperability using the token. We describe in detail the token structure, followed by different states of the token lifecycle. We also describe the implementation of token generation, token transfer, and token encashment phase. Since we assume that one of the primary mechanisms for exchanging tokens between the backend and front-end is SMS, we also describe how a QR-code based token can be exchanged, when necessary, over the SMS channel.

A. Token Structure

A token is an entity that carries the assurance of payment. A token is designed to contain several fields. Each token is associated with a unique identifier when it is generated. The unique identifier is a combination of service provider id, and a unique sequence number. For example, if the provider id is 110, and the unique sequence number is 12345, then the token id will be 11012345. This unique id is used to reference the token across service providers. Different types of payment modes are encoded using a type field, which denotes instant, dated or installment payment. Each type requires some specific fields to define it unambiguously. Two other necessary fields are, sender id and receiver id. The sender and receiver's service provider id, or bank account details can be optionally present in the token. Finally, the amount to be transferred is also maintained in the token. The fields are summarized in Table I.

In case of dated payment, the date from when the payment instrument is valid is also maintained in the token, along with the validity period. For installment payment, along with the first payment date, the token also maintains the number of payments, and payment value at each payment cycle.

B. Token State Transition

In order to maintain the lifecycle of the token, the token status is maintained at the backend. Several states are introduced which denotes the current state of the token. Important states of a token are as follows:

- *TokenGenerated*: This denotes the generation of the token, with the details sent by the sender. The token is marked with a unique identifier, the details are embedded in a QR-code, and the token is stored in the database of the sender's service provider.
- *TokenDeliveredToSender*: This denotes the successful delivery of a token identifier, or the QR-code representing the token, to the sender.
- *TokenAcceptedByReceiver*: This denotes the successful delivery and acknowledgment of receipt of the token id by the receiver when the sender has issued the transfer request to the backend. The sender's backend must send the token to the receiver's backend so that the token states can be maintained in a distributed manner. In the instant payment scenario, the encashment process starts immediately without explicit request from the receiver. If the encashment succeeds, the token state moves to TokenEncashmentSucceeded.
- *TokenRejectedByReceiver*: The receiver may reject a token for various reasons, like the amount is incorrect. In that case, the token is moved to TokenRejectedByReceiver state on the receiver's backend. The state is communicated by the backend to sender's backend as well.
- *TokenEncashmentRequested*: When the receiver requests to encash a token in his store, the specific token is moved to this state. For dated or installment payment, explicit encash request is expected, therefore, we introduce this state.
- *TokenPartEncashmentRequested*: In the installment payment scenario, the backend must keep track of partial encashments of the installments. This state captures the receiver's intent to encash the installment payment.
- *TokenPartiallyEncashed*: In installment payment scenario, this state denotes successful completion of an

Token Field	Explanation
Token Id	Unique token identifier across providers
Token Type	Denotes whether payment is instant, deferred or installment
Token Generation Date	Denotes the timestamp when the token was created
Sender Id	Sender identifier
Receiver Id	Receiver Identifier
Amount	Transferable amount
Token Status	Denotes the current status of the token
Sender Account	Service provider or bank of sender
Receiver Account	Service provider or bank of receiver

TABLE I
THE FIELDS COMMON TO ALL TOKENS

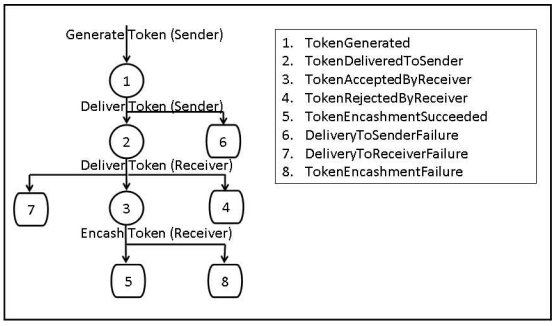

Fig. 2. State transition diagram representing token lifecycle for instant payment.

installment payment. We also maintain the payment status of each installment separately in the backend databases.

- *TokenEncashmentSucceeded*: After the wallet balances are adjusted and the actual transfers are completed, then the token moves to this state. This denotes that this token has completed its lifecycle, and can be archived.
- *DeliveryToSenderFailure*: A token moves to this state when a token is generated, but the token id cannot be delivered to the sender. This denotes a failure state, and the token can be moved to invalid or non-usable state.
- *DeliveryToReceiverFailure*: A token delivery to receiver may also fail, moving the token to this state. This is also a failure state, and may lead to token invalidation.
- *TokenEncashmentFailure*: When a wallet clearance fails, viz. due to insufficient balance on the sender's wallet, then the token is moved to this state. This is another failure state.

Fig. 2 shows the lifecycle of a token for an instant payment scenario. A sender requests a token to be generated by (a) using DigiToken app on a smartphone to request a new token, or (b) send an SMS to a pre-defined SMS short code. Once a token is generated, the token id is delivered to the sender, indicating that a token is ready for transfer. During transfer the token id is checked by the sender's service provider, and receiver's service provider is contacted to notify the receiver. In instant payment, the encashment step is triggered as soon as the receiver accepts a token. For dated and installment mode, the encashment is explicitly requested by the receiver. In the fund clearance process, the service provider of the receiver determines the service provider of the sender by extracting the unique provider identification number from the token id. A clearance request is raised to the senders service provider by the receiver's service provider indicating the token id. Once the tokens are identified, and the token validity established, then the actual clearance is performed, where the wallet balance of the sender is deducted, and the same amount is added to the wallet balance of the receiver.

C. Delivery of QR-coded token over SMS

Delivering QR-coded token, instead of just a token id, allows additional flexibility to the sender to transfer a token directly to receiver. A receiver scans the token from the sender's device. The token is parsed on the receiver's device to match the receiver id in the token. The receiver then sends a message to the backend acknowledging token receipt.

The transfer of token from the backend to an end device can be performed over MMS. In most countries MMS costs more than SMS. Any telecom provider supports SMS. The QR-coded token is encapsulated over multiple SMS messages. The SMS messages are marked with sequence number, and a type code that can be deciphered at the device end and the SMS messages are merged to regenerate the QR-coded token.

A QR-code is typically a black-n-white pictorial representation. Therefore, it is possible to deliver the complete pixel information as a bitmap. However, the number of bytes to send will be high, leading to large number of SMSes. An optimized delivery mechanism is possible due to the structure of QR-codes. QR-codes comprise of modules, where modules are black or white. The number of modules present depends on the content length, and the level of error correction. A higher version QR-code, which encapsulates more content, has more modules. Typically, the number of modules in a QR-code can be computed as $modules = 4v + 17$, where v is the QR-code version number which can vary from 1 to 40 as per standards. If the module information is transmitted, then the QR-code can be reconstructed. In our implementation, we decipher the bit matrix representing the QR-code module information, and send the bit matrix to the sender from the backend.

IV. DISCUSSION

DigiToken is demonstrated as a Android based smartphone app. In Fig. 3 we show execution steps for an instant payment. Let us assume that the sender and receiver maintains the wallet on two different service providers. Fig. 3-(a) shows the screen pulled up by sender in order to begin a payment.

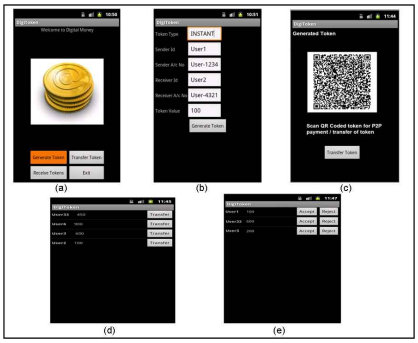

Fig. 3. DigiToken application screens showing different phases.

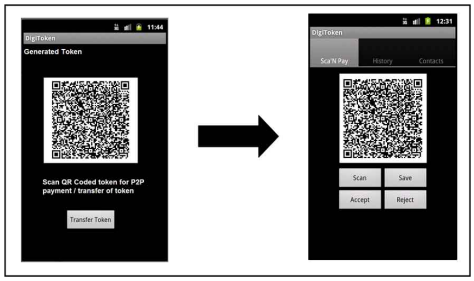

Fig. 4. DigiToken usage in a direct sender to receiver transfer scenario.

Generate token option leads to screen shown in Fig. 3-(b), which requires the different fields necessary for generating a token. Once the values are submitted, a token is generated and delivered to the user. Fig. 3-(c) shows the QR-coded token. Sender has the option to review all tokens ready for transfer, as shown in Fig. 3-(d). Once the token transfer is initiated, the backend notifies the receiver. Receiver can take a look at all tokens available to him, as shown in Fig. 3-(e). Once he accepts a token, the encashment process is initiated in case of instant transfer mode.

An alternative transfer mode is demonstrated in Fig. 4. In this mode, the sender displays the token to be transferred to the receiver. Receiver scans the token using DigiToken app on his device, and the transfer process is complete. Receiver can now accept the token, and the backend is updated.

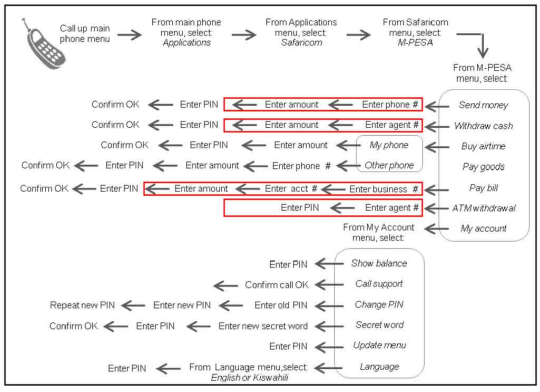

Fig. 5. Workflow in MPesa [15], and keypresses that can be eliminated by use of QR-coded tokens

Fig. 5 shows the workflow of transactions in MPesa Mobile Money system. Observe that at each step several key presses are involved. The use of QR-coded tokens can eliminate the need to type in several information, which can be stored in a token. We have shown the steps that can be simplified through use of tokens, represented as QR-codes, marked as red blocks in Fig. 5.

A. Feasibility of DigiToken

Token content can be of different lengths, depending on the size of each fields and the type of the token (instant, dated or installment). In addition, QR-code encodes alphabets and digits differently. There are also different error correction levels for encoding a content into a QR-code. We study the size of modules corresponding to different token content length and error correction levels. The size of modules dictates the number of SMSes, thereby the cost incurred, in the DigiToken setup.

Table II records the module length corresponding to different token types, and their error correction levels. Instant tokens has the least number of fields compared to Dated and Installment payment types. We have also varied the token field sizes, and reported the number of modules required to accommodate a small and large content in a token.

Since with increasing information content, number of modules in a QR-code increases, we calculated the number of SMSes when number of modules grows from 33 to 93. We chose 33 modules since it corresponds to the smallest token content. At 33 modules, 1 SMS is required to transmit the module information, whereas at 93, SMSes required are 8. If MMS is used to transmit the QR-code, then only 1 MMS is required. Based on mobile phone packages in different countries, we observe that in some countries SMS-based QR-code transmission technique, as described in Section III-C, would be more cost efficient. In UK, SMS is free, whereas MMS has fixed price; in India the cost of MMS is about 5 times the cost of SMS; in Australia, it is twice; and in USA, it is almost comparable. In Singapore, the cost of SMS and MMS is same, thus making MMS the favored way to transfer, whereas in Kenya, the cost of MMS is significantly higher than SMS, making the SMS based transmission a suitable option.

B. Security of DigiToken

We note that existing techniques can be effectively used for securing transactions. Trusted third-party services, like Verisign [11], that can sign each token before issuing to user can be used. The digital signature will be verified during token encashment, and a validation failure will trigger transaction rollback.

In our system theft of device will not invalidate any account or money. In case of theft of receiver's device, the receiver can easily download the existing tokens on a replacement device. If the thief attempts to encash the accepted tokens, then the money will go to the receiver's backend wallet rather than any of the thief's financial account. Password/PIN-authenticated token generation request will easily protect from potential implications of theft of sender's device.

Token Type	Content Length		QR-Code Error Correction Level	#Modules	#SMS
	#Characters	#Digits			
Instant	161	54	low	53	3
	173	82	high	57	3
	161	54	low	77	6
	173	82	high	85	7
Dated	186	70	low	57	3
	198	98	high	61	4
	186	70	low	85	7
	198	98	high	89	8
Installment	210	70	low	61	4
	222	98	high	61	4
	210	70	low	85	7
	222	98	high	93	8

TABLE II
EVALUATION OF QR-CODE SIZES WITH RESPECT TO DIFFERENT TOKEN TYPES

V. RELATED WORK

Several third party service providers and startups have launched mobile wallet services. Google Wallet uses QR-code to store a registered user's credit card information [5]. At a NFC enabled PoS, the QR-code storing the user details is transferred from a NFC-enabled phone to the merchant terminal. Google also supports a wallet entity where user can add money. Paypal Wallet is designed in a similar fashion, except that they do not explicitly depend on NFC-enabled phones, and provide other means of payments [9]. Similar use of NFC-enabled phones for transactions, where the wallet is maintained either at the backend or on the device, is explored by Balan et al. [12]. Other notable ventures in the mobile payment space are by American Express, with their serveAmex product [10], and a joint venture among AT&T Mobility, T-Mobile USA and Verizon Wireless to launch ISIS mobile payment platform [6]. These efforts are targeted mostly at instant payments, and do not address dated or installment payments. BitPay, which is an application built on top of a virtual currency platform, called BitCoin [1], uses QR-codes in their application. Bitpay uses QR-code to encode entities, like payment invoice, to aid in electronic peer-to-peer exchange of information [2]. Bitpay does not present a notion of token, which is treated as a first class transaction instrument in our system.

Among MNO-led mobile money initiatives in developing regions, MPesa is the most well-known. MPesa, managed by Vodafone in Kenya, allows small money transfers between ordinary mobile phones [16]. Similar approach is adopted by G-Cash in Phillipines [4], and Eko in India [3]. Some of these MNO-driven initiatives bring a degree of off-network interoperability by introducing agent kiosks, from where any user can encash her money into paper money.

In some countries, government intervention has introduced some degree of interoperability. In India, the Mobile Payment Foundation of India [8] is developing a model for interoperability. As part of this work, Kumar et al. has proposed architectural choices for interoperability [13], [14]. However, their model is specific to highly regulated financial environment in India, where every transaction is processed by a bank.

VI. CONCLUSION

Mobile payment systems are in rise across the globe with currently 124 mobile payment services operational worldwide. However, these services do not interoperate among themselves. We have presented an approach to address interoperability among mobile payment service providers. DigiToken application is based on a uniform format of a digital token that can be exchanged across providers. We have used an info matrix code, QR-codes, for encoding the information in a token. QR-codes are a standardized pictorial encoding of information, thereby making it machine readable, but not easily interpretable by humans. This along with a trusted third-party digial signature allows a degree of security, and at the same time it makes the token easily interoperable. Digitoken application demonstrates basic functionalities of the concept using QR-code.

REFERENCES

[1] "Bitcoin." [Online]. Available: http://www.bitcoin.org
[2] "Bitpay." [Online]. Available: https://bit-pay.com/aboutMobile.html
[3] "Eko." [Online]. Available: http://eko.co.in/index.php
[4] "G-cash." [Online]. Available: http://gcash.globe.com.ph/
[5] "Google wallet." [Online]. Available: http://www.google.com/wallet/
[6] "Isis." [Online]. Available: http://www.paywithisis.com/
[7] "Mobile money live." [Online]. Available: http://www.wirelessintelligence.com/mobile-money/
[8] "Mobile payment forum of india." [Online]. Available: http://www.mpf.org.in/
[9] "Paypal wallet." [Online]. Available: http://mashable.com/2011/11/02/paypal-wallet/
[10] "Serve." [Online]. Available: http://www.serve.com/
[11] "Verisign." [Online]. Available: http://www.verisign.com/
[12] R. K. Balan, N. Ramasubbu, K. Prakobphol, N. Christin, and J. Hong, "mferio: the design and evaluation of a peer-to-peer mobile payment system," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09, 2009.
[13] P. Chandrahas, D. Kumar, R. Karthik, T. A. Gonsalves, A. Jhunjhunwala, and G. Raina, "Some design considerations for a mobile payment architecture," in *Proceedings of the Seventeenth National Conference on Communications*, ser. NCC, 2011.
[14] D. Kumar, T. A. Gonsalves, A. Jhunjhunwala, and G. Raina, "Mobile payment architectures of india," in *Proceedings of the 16th National Conference on Communications*, ser. NCC, 2010.
[15] I. Mas and O. Morawczynski, "Designing mobile money services lessons from m-pesa," *Innovations: Technology, Governance, Globalization*, vol. 4, no. 2, pp. 77–91, April 2009.
[16] I. Mas and D. Radcliffe, "Mobile payments go viral: M-pesa in kenya," *Capco Institutes Journal of Financial Transformation*, vol. 32, Aug 2011.