Introducing RCD Host Tokenization

Shifting the focus to where it should be

Rob Stringer

ABSTRACT

Introduces the term Host Token and provides a fresh review and comparison of several mobile payment technologies under this lens. Technologies included are: Softcard, HCE, Paydiant, Apple Pay, and Cortex's own RCD Host Token

Change is the norm for Payments in 2014

In the past year there have been some major movements in the mobile payments space.

- The mandated migration to EMV in October 2015,
- EMVCo tokenization specification announcement in March, and
- Apple Pay joining the fray in September.

These happenings, along with the general unrest and noise in the industry provide tremendous opportunity for those supporting mobile wallet initiatives.

Shift in Liability

Liability will shift to acquirers and merchants who only accept payment via magnetic stripe in October 2015. As a result, acquirers and their merchants are scrambling to upgrade their POS terminals away from mag stripe to a more secure way to pay. This massive POS upgrade cycle is an opportunity for mobile payments to make a mark against plastic-based form factors, as merchants can upgrade with NFC (Near Field Communication) or QR codes embedded in their terminals, alongside the EMV chip card capabilities mandated by the networks. A major obstacle for this migration from card-based systems to mobile is: which acceptance technologies do you embrace, QR codes, NFC or some other technology? Apple Pay (working with the networks) has backed NFC for now, and MCX will back QR codes. Why should a

retailer have to choose? Truth is, they don't have to. There is a solution that works with both NFC and QR codes and runs on iOS and Android devices. As merchants consider their capital expenditures in 2015-2016, they now have a mobile payment option that insulates them from making an outmoded POS technology decision, future-proofing their mobile payment expenditures today.

EMVCo Tokenization Specification as a new standard

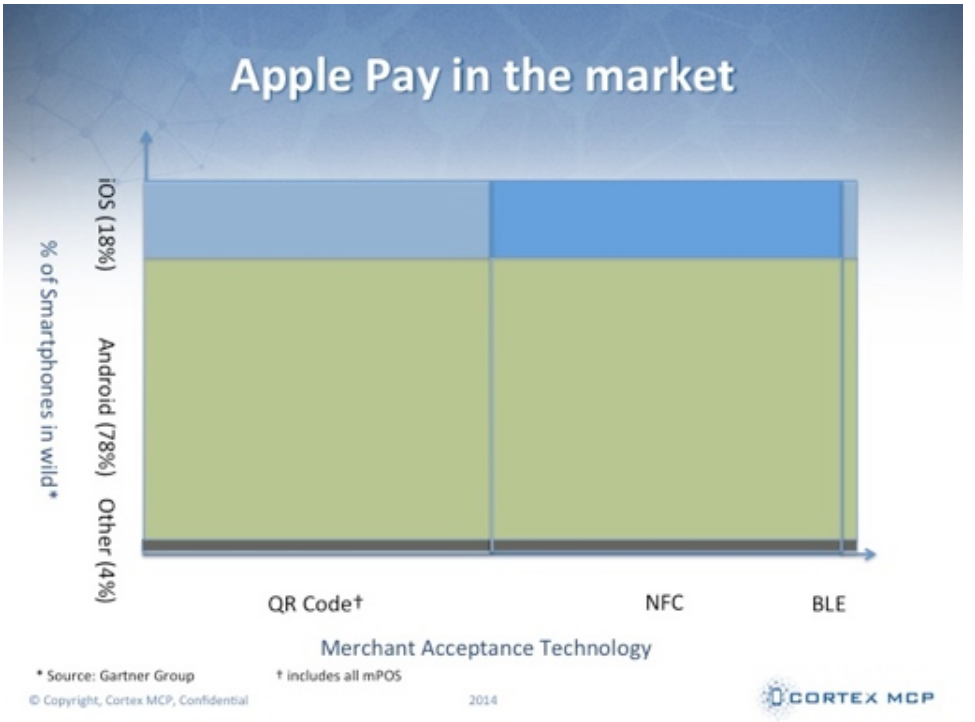
EMVCo – the same organization that came up with the chip card standard – has also delivered a tokenization specification for Visa, MasterCard, and the other credit networks. In the last six months this specification has garnered attention behind the scenes, but it wasn't until Apple announced Apple Pay in September 2014 that a significant player in the market validated this tokenization model. Apple has chosen to use each network as the Token Service Provider for their respective cards, and has established agreements with some of the big card issuers to load their cards onto Apple Pay. Now other players are rushing to create a tokenization model that works within the EMVCo specifications so they can argue for rates similar to, or better than, the ones Apple negotiated with the issuers and networks.

While the EMVCo specs are the new tokenization standard for the credit rails, what's to say a token couldn't be pass on another technology that's already embedded in many POS terminals worldwide? The move to tokenization may also create some issues among merchant loyalty programs, as some use the customer's card number (PAN) as the unique identifier in their

systems, so the move away from the PAN and to a token will cause their loyalty programs to no longer work as designed. This may hinder roll out of Apple Pay and other tokenization plans, since the PAN was used for more than just payments. That said, tokenization is here to stay. The only questions are when the migration will occur, what geographies will see it first, and in which market will the dominant retailer come on-board so the rest will follow?

Apple Pay's Impact

Apple's foray into mobile payments has put the spotlight on the ecosystem as a whole. While many in popular media saw Apple's entrance into mobile payments as the final straw that will finally



bring it mainstream, those in the industry are taking a more measured view. Apple did not reinvent anything in the space. They have chosen (so far) to embrace the incumbent network model and its nascent tokenization platform; however there are still many hills to climb before it's considered a success. Apple's unwillingness to open their NFC controller to outside development is a great limiting factor. There must be a token service provider that offers Android wallet providers (Apple Pay issuing partners among them) the same security and similar user experience as provided by Apple without requiring special hardware or a cloud connection to work. This token service provider would also need to support Apple devices leveraging a QR code transmission method. What kind of solution could compete with a SE without requiring a cloud connection to work?

What is the RCD Host Token™?

RCD Host Tokenization embeds a user generated PIN into the token itself instead of sending it as a separate block as is done with traditional PIN debit transactions today. The RCD Host Token is designed to run on the credit rails leveraging the EMVCo tokenization specifications, but is flexible in its formatting to secure multiple tender types and run in different rails. It's a model (PIN-based) with which consumers are comfortable while maintaining offline capabilities and PCI-grade security. The RCD Host Token sends one data packet consisting of an encrypted token coupled with other mobile device authenticating measures. This token and device combo is authenticated and validated server-side, through the POS connection, at the Token Service Provider. It is then either accepted or rejected pre-transaction to reduce the potential for fraud.

In this model, the token is always stored in device memory in an incomplete state, less the user generated PIN. This reduces the need to invest in the secure element (SE) – although host tokenization can pull the data from a SE if that is already in play. The incomplete token is only completed and usable when the user inputs their own unique PIN, ensuring two factor authentication. As a result the token is available for use anytime, anywhere, regardless of data connectivity with no special hardware required. The RCD Host Token can be formatted to comply with the existing EMVCo specification formats (field 2), and it is arguable that this type of token should garner the highest token

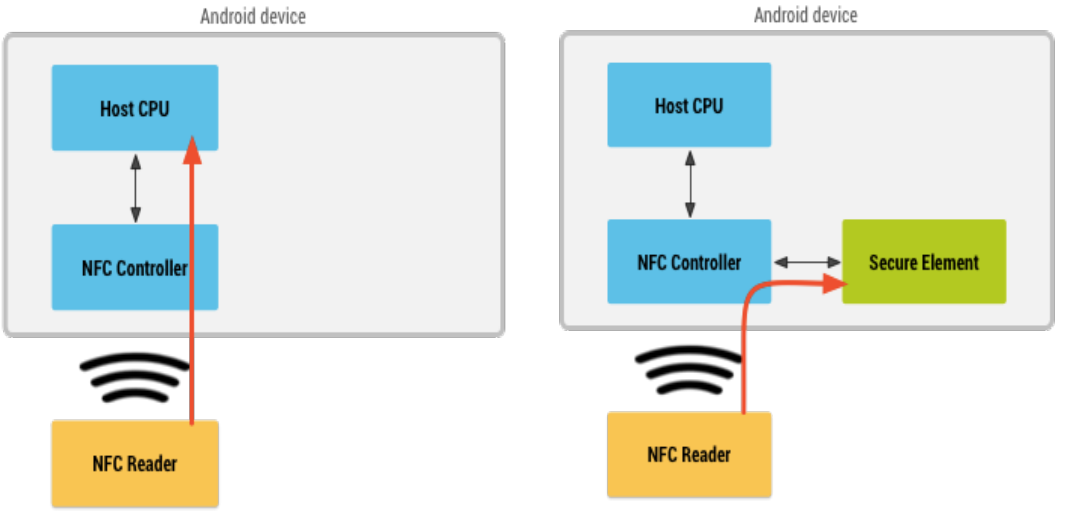
assurance levels from the networks. It relieves the banks and other wallet players of having to invest in chip technology or have an incomplete (i.e. no standard off-line use case) solution. This is potentially a big issue in the Android market as these phones no longer come with a built-in SE. Instead it seems like they're dependent on SIM-based SEs meaning that they'll be controlled by the wireless providers / Softcard. Even if there is agreement shipping out all sorts of new SIMs this introduces a lot of friction. With the Cortex RCD Host Token a single app could support every smartphone out there. This is a simple solution that could and should become the industry standard across face-to-face and on-line transactions.

The RCD Host Token was built to work with the existing infrastructure, and was conceived prior to the EMVCo tokenization specs were released in March 2014. Thus, it can be implemented by existing Token Service Providers (the networks) or new players in the space to provide a solution that works on multiple POS infrastructures, mobile operating systems, and while the phone is offline without degradation of the EMVCo token validation level. Consumers deserve the seamless security, and merchants deserve the uptime and reliability the RCD host token system brings.

As merchants and their payment processing partners invest in non-mag stripe POS technologies, they should consider a platform that is the flexible, secure, and low maintenance. The RCD Host Token is all of the above, and it is available now for pilots and production.

Defining what HCE really is (and isn't)

What is HCE? According to Wikipedia, "Host Card Emulation (HCE) is the presentation of a virtual and exact representation of a smart card using only software." According to Wikipedia, this definition of the term was coined by SimplyTapp to support their cloud-based product line. Google, a more authoritative source, uses the term more loosely. Defining HCE as "the ability to access the NFC controller for payments from a host CPU, instead of only the SE" (see below).



ref: http://developer.android.com/guide/topics/connectivity/nfc/hce.html

Most current mobile wallet implementation models of a HCE system assume that the "Host" is in the cloud, and that the mobile

application can pull the data from the cloud host to the NFC controller on the device. The Google Wallet model of old and current Isis Softcard model, store the track data from the card on the mobile device by using a SE. Card network PCI rules state that one is not allowed to store track data in device memory on the phone, so it has been argued that to have a secure, Card Present, transaction without a SE on the device, you need a SE "in the cloud" to act the part of the SE on the device. HCE proponents claim this kind of model, but if in fact the application is calling a true card emulation (we are talking about host card emulation) then the track data of the card, the data that previously had been allowed to reside on the SE on the device, has to be stored in its entirety in the cloud but then how is it going to be transmitted to the mobile device and then to the POS? Will there be software that sits between the SE in the cloud and the NFC controller to encrypt and decrypt the payload for use? If that's the case, then hackers have a way to steal the data on the phone, as it's going to be in the clear at some point on the device and not in a SE or with a TSM controlling it. The application cannot touch or encrypt the payload and decrypt it on the device, so to store the payment data in a "SE in the cloud" it must get transmitted to the device in the clear.

HCE proponents will argue they aren't sending the card data, they are sending a token so the payment data is never in the clear. To that we say, if you send a token instead of a card, shouldn't we change the name host card emulation? It's not Host Card Emulation anymore; it's Host Tokenization. Host card emulation as a term was created before the implications of sending card data

in the clear were fully understood, and it is misleading and unduly limiting.

Host Token and Host Tokenization

Cortex MCP has coined the generic terms Host Token and Host Tokenization to refer to "any token that is stored on a secure host for use in a transaction on a mobile device." It does not limit the usage to only NFC transactions, but includes QR code, Bluetooth Low Energy (BLE), and future data exchange methods not yet invented. It does not limit where the Host is; the host could be in the cloud, on a SE, or in device memory. The implications of using Host Tokenization vs. Host Card Emulation when talking about mobile payment transactions are important ones. While HCE is focused completely on NFC as a transmission technology and is seen as the cloud alternative to the SE, Host Tokenization puts the focus on the token, and allows the host to reside in a SE, the cloud, or, potentially, in device memory as in the RCD Host Token.

SE systems like Apple Pay use SE Host Tokenization for NFC, where the SE describes where the host token is stored for use in the transaction. SimplyTapp and Paydiant use a cloud-based Host Token to complete their transactions. SimplyTapp couples that with NFC and Paydiant with QR codes. Isis Softcard does not use host tokenization at this point, as they embed the track data on the SE, not a token.

Method	Token	Host	Technology
~~Isis~~ Softcard	No	SE	NFC
HCE	Yes	Cloud	NFC
Paydiant	Yes	Cloud	QR
Apple Pay	Yes	SE	NFC
Cortex RCD	Yes	Device Memory	NFC/QR

Host Tokenization broadens the conversation on what is possible for mobile transactions so the real focus can be the real problem, which is how to migrate from an inherently insecure technology (mag stripe) to a more secure mobile standard. HCE is a term whose relevance is limited in today's tokenized world.

Shedding a New Light Reveals Hidden Features

Changing terminology sheds new light on existing technologies and allows the viewer to concentrate on a feature that might not have stood out before.

If the payments industry starts talking about Host Tokenization instead of HCE the players can open themselves up to implementations that go beyond NFC and bring tokenization to the fore (where it should be for mobile proponents) enabling more collaboration and setting up standards that will help the entire ecosystem. HCE is merely a type of Host Token. In this chapter we will delve into each type of Host Token to compare and contract their strengths and weaknesses. We will let the readers come up with their own opportunities and threats for a full SWOT analysis.

Let's take them in order.

Softcard/Google Wallet 1.0

We all must give thanks to Google for launching their first version of the Google Wallet back in 2011. Without that ill-fated attempt to take over the payments world, the ecosystem wouldn't be in the shape it's in today. Maybe no existing player in the "mobile wallet wars" owes as much to Google as does Softcard (née Isis). Softcard was created as the mobile network operators defense against Google for trying to use them as "dumb pipes" and a distribution channel for its SE-based wallet. It seemed like a good idea at the time, when there was no talk of tokenization, and the

only way to qualify for card present rates was to use the complicated and expensive TSM SE model. The landscape has changed, and what was once strength is now a weakness.

Strengths

Softcard has a gigantic lead in its ability to distribute its product to the masses. Softcard's owners also each have an existing fiduciary relationship with their customers that they could leverage under the right circumstances to create value for both parties. Softcard has also spend time and money establishing relationships within the payments processing industry to grow their established base and have invested in getting NFC terminals working in key launch sectors and geographies. This is the type of third party infrastructure investment that may be necessary to get merchants and processors on the NFC bandwagon.

Weaknesses

Softcard threw their lot in early with a hardware play. It started with the SE and now relies on consumers buying a device that has the SE embedded or to distribute a SIM-based SE in order to run their wallet service (Android). Yes they have third parties that have developed sleeves for some popular phones that don't come standard with a SE that Softcard can use, but to scale, relying on a consumer to buy a new piece of hardware is a very limiting factor. Another weakness is that Softcard's model was designed prior to the EMVCo tokenization specs came out, and Softcard didn't adapt to the rise of tokens to help them out of the track data-on-the-SE hole. Finally, Softcard was dealt a crushing blow when it had to rename itself, eroding whatever brand it had established among

consumers. The fact that Apple Pay has effectively shut out Softcard from ever being used on all iOS products in a user friendly way is also a hindrance for Softcard.

HCE

HCE was initially created as an alternative to the SE model, with both focused exclusively on NFC. Prior to the EMVCo tokenization specifications coming out, HCE was truly host card emulation, with companies trying to send track data in the clear from a cloud-based host to the mobile device to use by the NFC controller. In theory, this works, however, in practice it became apparent that companies couldn't send track data in the clear. So the model moved to storing tokens in the cloud and sending the token to the device instead of the vulnerable track data.

Strengths

HCE is a term and model that has gathered a lot of steam in the past 9 months, and rightly so. It is a viable alternative to the SE model (popularized by Google Wallet 1.0) that preceded it. It now has solid brand recognition, albeit misguided. HCE also runs on NFC, which is a transmission technology that has had new life breathed into it by the launch of Apple Pay (more on that later). NFC is de rigueur in Europe with contactless EMV, but it hasn't quite caught on in the US. According to Javelin Research, Global NFC ready POS terminals will be just over 50% of total POS by 2017. NFC is a known quantity in the existing payments infrastructure, tried and tested. It is a technology that the

networks support in their data formatting, and no changes would have to be made at a network level to roll out a NFC solution.

Weaknesses

A weakness in this model is that it assumes the consumer's mobile device has a cloud connection at the time of purchase. If the consumer's mobile device is online, the model works (assuming no latency in the connection): it's secure enough, and transactions can go through. Offline mode, or times when the consumer's device cannot get cell coverage, are use cases that do not fit neatly in the HCE model. Workarounds have been developed to create a "one-time token" that resides on the device for a limited amount of money and limited timeframe. This isn't the ideal scenario, and the merchant gets penalized for something out of their control when one of these offline tokens is used, since it's not as secure or verifiable since it sits in device memory in the clear and can be scraped or stolen fairly easily if the phone is in the hands of a hacker. An additional weakness of the HCE model is its limitation, by definition, to the NFC transmission technology. While NFC is an important contactless credential transmission method and one that has been tested with legacy POS terminals and systems, it is not the only one. To build out an entire industry based on a technology that has rightful competition may be short sighted.

Paydiant

Paydiant has been around longer than most people think. Founded in 2010, Paydiant designed a mobile payment system leveraging tokens and QR codes way before the EMVCo tokenization specifications came out. Its system is flexible in that

either the consumer or the POS can show the QR code that would be scanned by the other party, thus enabling merchants without a QR code scanner to use their product. But is this really a strength or is it a weakness for scalable rollout?

Strengths

Paydiant has inked some good deals with some major names in the industry, and has a strong brand among the small and medium retail segments. Paydiant has first mover advantage in the QR token space, as they have been "talking tokens" for four years now, while most in the physical retail space have started to use the work token in 2014. The flexibility of Paydiant's system in terms of which device (POS or mobile) presents the QR code allows them to get into some merchants that don't want to invest in a QR code reader.

Weaknesses

Paydiant's system requires that both the POS and the consumer's device are both connected online in order for their token to be presented and validated. Similar to HCE, there are workarounds where the mobile device can have a one-time token provisioned and stored on it in case it's off line, but this again is not ideal. There is still risk of theft if a hacker has the phone. One good thing (but also a weakness in the overall scheme of things) is that since Paydiant only offers card not present rates to its SMB customers there is no network interchange penalty for using an off-line token vs. one that is provisioned at the time of sale (like HCE). Paydiant's reliance on QR codes as transmission methods could also be seen as limiting their ability to scale. Finally, implementing

the Paydiant system requires significant investment by the merchant in upgrading their POS systems, either to accurately read or present the QR code.

Apple Pay

Apple Pay has not yet launched (updates will be provided as more information is gathered post-launch) but it has garnered the lion's share of attention over the last month since it was announced. Many pundits have delved into its deepest depths (through Apple's partners) so we won't reiterate them in this paper (but we did in our blog). Apple is always a force to be reckoned with in any market it goes into, but will it have enough strength to outweigh the weaknesses in its 1.0 payment offering or will it go by the wayside like Google 1.0? Apple, like Google, is relying on a SE, but, unlike Google, Apple is loading a token on the SE and has a payment processing environment that is much more open than it was when Google launched its first version of Google Wallet. Finally, Apple controls the hardware and distribution in ways that Google could only dream of when it launched.

Strengths

Brand, brand, brand. Apple will get people to put their qualifying cards in the Apple Pay passbook wallet just because they are Apple fans. The number of cards loaded is a key metric Softcard and others have touted as they've grown, and Apple will surely have many cards loaded by the time of launch. Apple has already established that it has a strong start with partner retailers willing to put their brand alongside Apple's at launch to accept Apple Pay. Apple's biggest strength must be that they are the first wallet to

embrace the EMVCo tokenization specifications and benefit from the newly formed token assurance level to get an interchange rate between that of card not present and card present. By using the network's specifications and token service providers (TSPs), Apple is supporting the incumbent payment infrastructure that will reduce friction as it encounters bumps in the road. Apple has the cash on hand to invest in supporting technology rollouts (NFC POS terminals) and training – similar to what Softcard has done in its markets, should it deem important enough to do so. Apple has also locked down their NFC controller but <could> open it up to third party developers once they find a way to monetize the risk of payment processing.

Weaknesses

Apple, for all its brand awareness, technical prowess, and strong cash position, is still only one player in the ecosystem. It takes a coordinated effort to make any changes in payments, and Apple is not known for playing well with others. Apple handsets only account for 18 percent of worldwide smartphone sales. Apple's strong brand could also signal a weakness as it extends via Apple Pay to physical retail experiences it does not control. What happens when Apple consumers and Apple Pay users have a bad experience trying to shop at an Apple Pay partner retailer? Will they blame the partner, Apple, or both? Apple has formed some additional partnerships with the big credit card issuers, but what happens when their competitive juices start flowing? Which issuer will get the coveted "top of wallet" status in Apple Pay? How long before those big issuers start creating their own TSPs to

not be beholden to the networks for yet another number? They'll still run on the same rails, just use a different TSP than the network's. Speaking of the Issuers, what about their Android ambitions? It might be safe to assume that the big issuers will want their brand on a wallet that all of their customers can use across devices and operating systems. Apple's refusal to open up their NFC controller to third party app developers will hinder that ambition. Apple will surely start out strong, but Apple did not and will not reinvent payments overnight, and one may question if it there is appetite to overcome all obstacles for a full-scale rollout.

Cortex RCD

The Cortex RCD Host Token is a payments platform that was created several years ago, but has only recently garnered attention as the industry's tokenization lexicon has matured. Cortex's solution was built to work with any tender type, yet takes advantage of the EMVCo tokenization specs to allow TSPs to have a secure solution that runs on all mobile operating systems and transmission methods, regardless of device connectivity. The RCD Host Token is stored on the phone in an incomplete state, less the user defined PIN. At time of payment, it is reconstituted by embedding a user generated PIN into the partial token encrypted in device memory to create the completed RCD Host Token. That completed token is then transmitted, along with additional device specific signatures, via the merchant POS to the back end for validation and processing following industry standards.

Strengths

The RCD Host Token is the only token platform that is truly agnostic. It can be used with NFC-based wallets or QR code-based formats. It was designed to flexible in its formatting to comply with the "rules of the rails," be them EMVCo tokenization specs, ACH, or even future standards such as BitCoin. Unlike cloud-based host tokens, it is not a requirement that the consumer's mobile device be online to generate and transact with the RCD Host Token, enabling additional use cases cloud-based approaches fail to address. The real strength of the RCD Host Token is the user control component that is embedded in the token itself. By requiring a user generated PIN to complete the RCD Host Token, the token is by-definition a two-factor authenticated token.

Weaknesses

Cortex is a relatively unknown brand in the industry. The on-device, PIN completed, RCD host token method is not a model that is used by others in the industry, so has not received the same level of validation in-market by the networks as cloud-based tokenization platforms or SE models.

Checking in, BLE and other interesting side shows

In 2010 at the National Retail Federation show in New York City, mobile was just starting to get its due in merchant circles. Merchants were looking for the Minority Report kind of interaction with their potential customers and thought mobile could be the answer. The biggest stumbling block was they didn't control the consumer's mobile device, or have access to proximity data to know when or where their customers were. Many questioned how they could get the consumer to switch from their mobile network data stream to the in-store Wi-Fi (where they'd have control of the data). All that has changed, as the retailer no longer needs control at the network level if they have the right settings set in their mobile app (and it's loaded on the consumer's device). Privacy experts cried foul at some of the early "push" notifications coming from proximity marketing since it wasn't explicit in the terms of the application, so marketers (and app developers) got savvier. Next, "check-ins" were the rage.

Square was one of the first to make a splash with their "pay with your face" check in technology. A consumer would store a card-on-file or bank account with Square, upload a picture of themselves, and proactively "check in" at a location where they were going to shop. When they checked in, their previously uploaded image would appear on the custom POS (Square Register) so the clerk could check them out. This failed to make

much of a dent, as it required significant staff training and low volume throughput to work effectively.

The concept of check in's has not yet died though. Before there was talk of tokenization there were beacons, another way to automatically "check in" and pay. First PayPal came out with Beacon, followed closely by… iBeacon from Apple. These BLE devices got plugged into an outlet at the merchant's store, integrated with their POS, and were supposed to usher in a new era of customer engagement in-store. Customers who had beacon applications enabled on their devices would automatically be "checked in" after a quick verification step. One benefit of Beacon technology is that through BLE a device can get access to the cloud even if their mobile network data connection is down. For this to happen, the merchant has to have a beacon installed and the consumer has to: a) have a BLE compatible phone; b) have downloaded the merchant application and enabled beacon in it; and c) be close enough to the Beacon to get a clear transmission signal. Beacons could be the answer to some cloud-based tokenization initiatives, however Beacon technology is not yet mainstream, nor have the privacy experts quieted on this front. iBeacon is designed so that all newer iOS devices can act as beacons of their own. In fact, some experts were expecting Apple to launch a BLE solution, and they may still. But for now, BLE is a dream for marketers more than it is a reality for payments professionals.

Finally, there are prepaid/loyalty applications like Starbucks or LevelUp that are trying to be more than just niche players in the

payments space. The Starbucks app is a shining example of success. However few retailers have the demographic or the willingness to invest in their mobile app and staff the way Starbucks did to achieve success. Nevertheless, Starbucks has proven beyond a reasonable doubt that value of mobile payments in that they're seeing a 50% increase in spend for customers who switch to paying with their mobile app from their branded prepaid cards. LevelUp has tried to reinvent itself in multiple ways, but faces the challenge mentioned above in that staff training required to implement its custom payment system smoothly is not under their control. This limits their ability to see success on a scale basis.

Method	Token	Host	Tech	Strengths	Weaknesses
Softcard	No	SE	NFC	CP Rates, Distribution, Customer Relationships, Known Tech, offline use	Non-Token, Expensive, Complex
HCE	Yes	Cloud	QR	Known Tech, Brand/Buzz, Model Vetted	NFC Only, Cloud Connection Necessary
Paydiant	Yes	Cloud	NFC	Established in verticals, key partnerships, Mature Token	Cloud Connection Necessary, QR code only, CNP rates
Apple Pay	Yes	SE	NFC	Brand recognition, TSP rates, Known Tech, Model accepted, offline use, compliant with EMVCo	Limited distribution, walled garden, NFC only
RCD	Yes	Device Memory	NFC/ QR	Built in 2 factor authentication, off-line use, flexible tech, compliant with EMVCo	Model not yet validated

A case for the RCD Host Token

Each technology has its strengths and weaknesses. The market is far from settled, and, as time goes on, the true measure of each will become apparent. We believe offline is a bigger issue than most think at this time. We also believe the only way to solve it is to either have a SE or an RCD host token. Only one of those options is software-only and easily deployable.

This kind of graph is slightly misleading as we are not comparing apples to Apples (pun not intended). There are cases where one player could use another's technology to augment their existing solution, and there are some cases where they two systems are mutually exclusive. Competitive forces will determine what kind of co-opetition exists as the industry settles.

We believe that merchants will demand the flexibility to install the transmission method they choose (NFC or QR code) and that the system they ultimately choose should work with both NFC and QR codes. Any branded wallet will also have to be usable for both iOS and Android over time; otherwise there will too much fragmentation. The only company and technology that has all of these traits: offline capability, support for NFC and QR code, and the ability to run on all operating systems is the Cortex MCP RCD host token.

Cortex's platform is white-label ready and purpose-built to enable mobile wallet functionality in partner applications. It provides a uniform mobile payment experience to customers across all

devices and is ready-to-go right now. Given the recent market changes highlighted in this paper, the time to push for adoption and try new methods is now, less the risk of ceding this space to these significant new entrants. The RCD host token should be considered by any organization seeking to establish a mobile payment relationship with their customers. This includes those organizations that are planning a new solution or those who have already invested in one of the existing methods and searching for a complement to get to the next level.

Please visit http://www.cortexmcp.com to learn more about the Cortex Mobile Commerce Platform and RCD Host Token.

About the Author

ROB STRINGER, VP MARKETING & PRODUCT DEVELOPMENT

Rob thrives on bringing new business models into existing infrastructures. With over 12 years of experience identifying, developing, and executing innovative corporate growth strategies, Rob has a proven track record of bringing disruptive products to market successfully. A graduate of the Olin School of Business at Babson College, Rob is a Red Sox fan, and still cannot believe that Dave Roberts stole the bag.