Burton IT1 Research
G00218061

The Evolving Intersection of Mobile Computing and Authentication

Published: 22 December 2011

Analyst(s): Mark Diodati

Behind the hype of IT consumerization lies a tectonic shift in enterprise computing. Corporate IT organizations must address authentication from mobile devices. Authentication methods and technologies surveyed include X.509 certificate, OAuth 2.0, Near Field Communication (NFC), and software one-time password (OTP) devices.

Table of Contents

List of Tables

List of Figures

Summary of Findings

Bottom Line: As the consumerization of IT continues, enterprises are pushed to support the user's computing platform of choice. Increasingly, the platforms of choice are the tablet and the smartphone. Consumerization forces the enterprise to create new identity management (IdM) functions — especially authentication — for mobile devices. In all likelihood, the enterprise will need new infrastructure components like mobile device management (MDM) software to pull it off. Authentication mechanisms include X.509 certificates, OAuth, software-based one-time passwords (OTPs), and ultimately Near Field Communication (NFC)-based technologies.

Context: Gone are the days when the enterprise can expect to leverage a single user platform — the Windows PC or Mac connected to Microsoft Active Directory (AD). Mobile device computing is effectively decoupled from the Windows domain model, so alternative authentication and single sign-on (SSO) strategies must be considered. Additionally, the ability of the mobile device operating systems to consume a single set of credentials for multiple applications varies significantly by vendor.

Take-Aways:

- X.509 certificate:
  - May provide the best device integration, but will likely require additional infrastructure (e.g., MDM)
  - MDM:
    - Provides certificate management capabilities for remote devices
  - Native platform capabilities:
    - Maturity of operating systems vary, with BlackBerry and iOS having more enterprise capabilities than Android
    - Certificate consumption and trusted CA list update:
      - The ability to consume a single credential for multiple applications on the device is best on BlackBerry and iOS
    - Android is the least mature
  - Certificate authentication for email:
    - Implementations exist that can eliminate the storage of the Active Directory credential on the phone
- OAuth 2.0:

- Emerging as a popular protocol for session management of users accessing resources via rich mobile applications (RMAs)
- Rich mobile versus mobile Web applications (MWAs):
  - MWAs function well with traditional enterprise authentication and session management
  - Rich mobile applications may break existing infrastructure like authentication and Web access management (WAM)
- The importance of identity management:
  - OAuth leverages long-lived session tokens for usability; prompt de-provisioning of users is essential
- OAuth as authorization framework:
  - Currently, OAuth's strength is supporting authentication and session management functions for rich mobile applications
  - Over time, additional identity and access management (IAM) technologies (e.g., User-Managed Access [UMA] and Extensible Access Control Markup Language [XACML] will enhance OAuth's authorization capabilities)

- Near Field Communication:
  - Currently, NFC-enabled devices are a rare breed, but this will change in the next few years
  - Point of sale (POS) payment systems will drive innovation and create opportunities for NFC use for authentication, including access to physical resources
  - Technology:
    - Mobile device support:
      - **Android:** Best support due to Samsung Galaxy platform and introduction of products by other hardware vendors
      - **BlackBerry:** Introduced NFC support in a few phones in 2011
      - **iOS:** Not yet supported
      - **Tap to Pay:** a) Google Wallet and b) ISIS
  - Software OTP Client
- Strengths:
  - Device maturity: Mobile devices have support for a common certificate and private key storage
  - Authentication
  - Infrastructure

    - OAuth
    - NFC
    - Software OTP
- Weaknesses:
    - Enterprise readiness of operating systems
    - Smartphone and tablet limitations
    - NFC is coming, but . . . don't hold your breath
    - Mobile application duality
- Recommendations:
    - Be specific
    - Use X.509 certificates and mobile device management
    - Evaluate application development architectures
    - Use OTP devices for non-employees
    - Look, but don't leap onto NFC

Conclusion: As the consumerization of IT continues, enterprises are pushed to support the user's computing platform of choice. Increasingly, the platforms of choice are the tablet and the smartphone. Consumerization forces the enterprise to create new identity management functions — especially authentication — for mobile devices. In all likelihood, the enterprise will need new infrastructure components like software to pull it off. Authentication mechanisms include X.509 certificates, software-based OTPs, OAuth 2.0, and ultimately NFC-based technologies.

Analysis

Much hype surrounds the "consumerization of IT," but behind the hype lies a tectonic shift in enterprise computing. Employees are demanding a bring your own device (BYOD) policy to enable them to be productive. The advent of the tablet form factor (and its relatively generous screen size) makes this productivity possible. Consumerization is also gaining momentum due to cloud computing because it reduces the dependency on the traditional user platform — the PC (or Mac) that is bound to the enterprise's Active Directory network operating system. The implication of consumerization is that corporate IT organizations are pressured to grant access to corporate resources (or at least a subset of resources) via mobile devices.

Authentication

Corporate IT organizations must address how users authenticate when operating smartphones and tablets. Gartner defines authentication as "the procedure through which a user provides sufficient

credentials to satisfy requirements for access to resources." Implicit within this definition is the concept of proof; the organization must have confidence that the entity at the other end of the transaction is a legitimate user. Authentication is an important pre-condition for authorization because entitlement management is worthless until the user's identity has been vetted.

Shifting to the Mobile Computing Platform

Traditionally, the discussion of mobile devices and identity has focused upon the use of the mobile device as an authenticator for accessing resources via another computing platform. For example, users may enter a one-time password that is generated on a mobile device for access to resources via their PC or Mac.

Because smartphones and tablets have become viable computing platforms for end users, this assessment focuses on user access to corporate resources via the mobile device, including four authentication methods:

- X.509 certificate
- Software one-time password (OTP) client
- OAuth 2.0
- Near Field Communication

X.509 Certificate

The ability to leverage certificates is governed by two factors: the maturity of the mobile device operating system and the robustness of the MDM solution. When discussing certificate management and mobile devices, two specific capabilities are required for functionality:

- Certificate procurement
- Trusted CA list update

Certificate Procurement

When leveraging PKI authentication, the user must possess an X.509 certificate and associated private key. In the case of mobile computing, the certificate and private key must exist on the mobile device (unless the device supports a contactless smart card — a rare occurrence). Organizations generally have two methods for certificate procurement:

- **Distribution (Push):** Install the certificate and associated private key to the device. Certificate distribution represents a "push" modality and usually provides the best usability and scalability.
- **Enrollment (Pull):** This procurement method represents a "pull" modality; the user must enroll for a certificate via contact with the organization's certificate authority. The enrollment can be somewhat automated via certificate profiles or policy distribution.

Regardless of push or pull modality, scalable certificate management is best achieved via MDM capability.

Distribution (Push)

Distribution is generally the most usable and scalable method to place the certificate on the device. Distribution is achieved via one of two methods:

- **Over the air:** Certificates and private keys are distributed remotely over Wi-Fi or cellular network, typically via third-party MDM solutions.
- **MicroSD Smart Card:** Organizations can distribute microSD devices with certificate and private key. This method requires that the mobile device have a microSD slot.

MicroSD Smart Card

ActivIdentity and CertGate provide smart card capabilities via the microSD container, which can provide public-key infrastructure (PKI) services for BlackBerry devices that have a microSD slot. A traditional smart card (sans the plastic substrate) is housed within the microSD form factor. As with traditional smart cards, the organization can personalize the card with a certificate and private key, then distribute the microSD container to the users, who then install the device into the microSD slot. In the future, the microSD Smart Card will function as a secure element within NFC-enabled devices.

Enrollment (Pull)

With enrollment, the mobile device initiates the enrollment of the certificate by directly contacting the organization's certificate authority. Two enrollment methods exist:

- **Automatic:** The MDM solution may distribute the necessary enrollment information (e.g., Simple Certificate Enrollment Protocol [SCEP] parameters) to enable enrollment directly from the CA. This method is exemplified by the solutions from Apple and BlackBerry. The BlackBerry Enterprise Server (BES) provides a more complete solution with better usability; its approach compares well with certificate distribution methods provided by the third-party MDM vendors.
- **Manual:** This method is the least desirable approach because it requires interaction with the user to complete the enrollment process and therefore introduces opportunities for human error.

Trusted CA List Update

The mobile device's trusted certificate authority list must be updated to ensure that the mobile device can connect to on-premises resources with certificates that are not tied to a public CA. For example, when implementing a VPN solution in conjunction with certificates, the user and therefore the mobile device must have a certificate and associated private key. Typically, the VPN server also has a certificate key pair. If the mobile device's CA trust list cannot be updated, in most cases organizations will have difficulty implementing the VPN solution because the VPN server certificate will not be trusted by the mobile device. The update of the certificate trust list varies by operating system platform. For example, iOS and BlackBerry operating systems enable the update of the certificate trust list. However, Android does not currently support this capability (unless you "root"

the device, a process similar to iOS "jailbreaking"). The result is that third-party VPN solutions (e.g., Juniper and Cisco) — each with its own certificate store and trusted CA list — must be leveraged for Android.

Certificate Consumption Capabilities

The three major mobile device platforms (i.e., BlackBerry, iOS, and Android) provide a native certificate store that enables the use of certificates and associated private keys via the operating system cryptographic API. However, the maturity of the operating system and its applications that consume certificates varies by platform. BlackBerry has the best overall integration with certificates and PKI, followed closely behind by Apple's iOS. Android remains the most immature operating system for PKI but will likely mature to support more enterprise use cases.

Table 1 summarizes the ability of the Android, BlackBerry, and iOS operating systems to leverage certificates from the centralized operating system certificate store.

Table 1. Summary of Operating System Certificate Consumption Capabilities

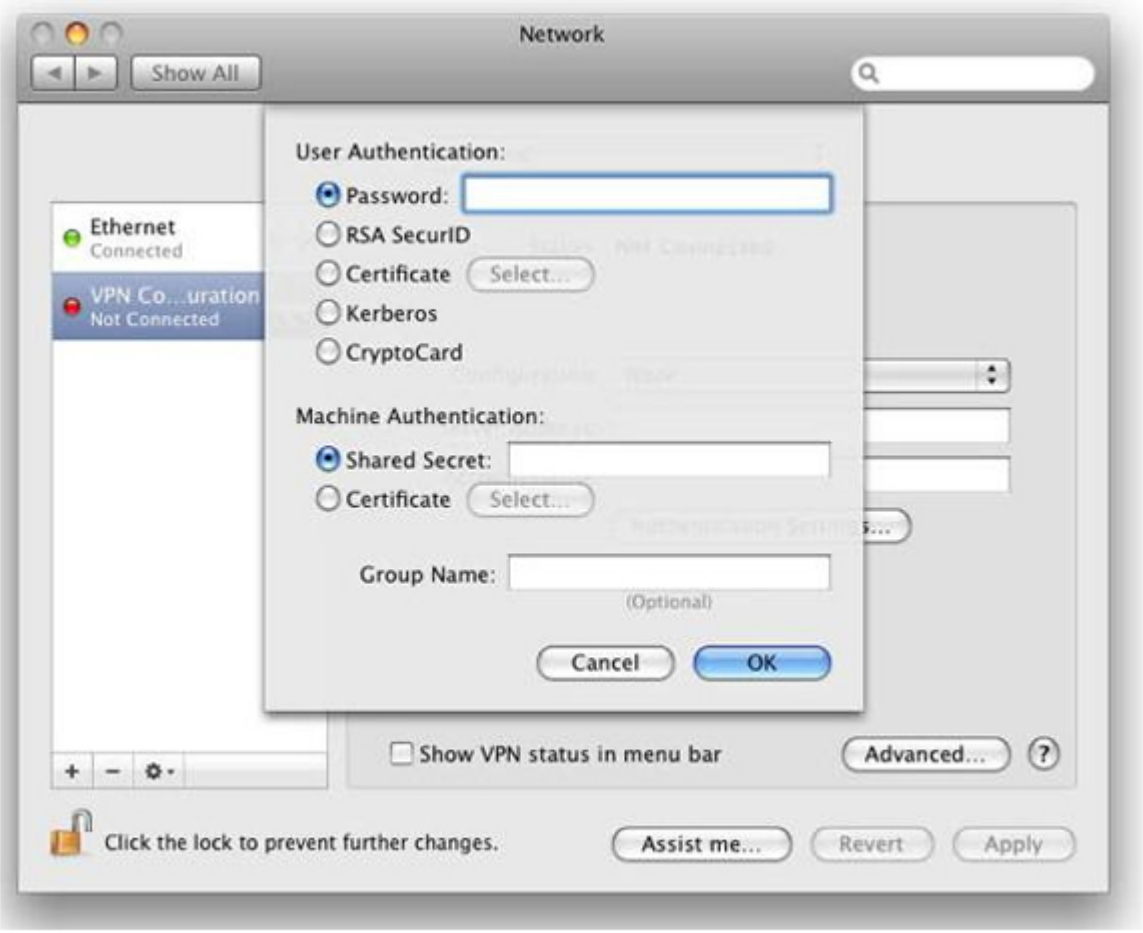
	Wi-Fi	Email authentication[2]	IPsec	Web browser	S/MIME
Android	Yes	No	No	Yes	No
BlackBerry	Yes	Yes	Yes	Yes	Yes[1]
iOS	Yes	Yes	Yes	Yes	Yes

1 Not installed by default, but available via download from BlackBerry and can be pushed to mobile devices by BES.
2 The use of certificate-based authentication without requiring additional password authentication.

Source: Gartner (December 2011)

The iOS (see Figure 1) and BlackBerry VPN clients (see Figure 2) have broad interoperability with commercial VPN products and support many authentication types. Android's client is the least functional; most enterprises deploy a VPN client from the vendor (e.g., Juniper or Cisco). Certificates are stored in the VPN client's certificate store (not the device's centralized store).

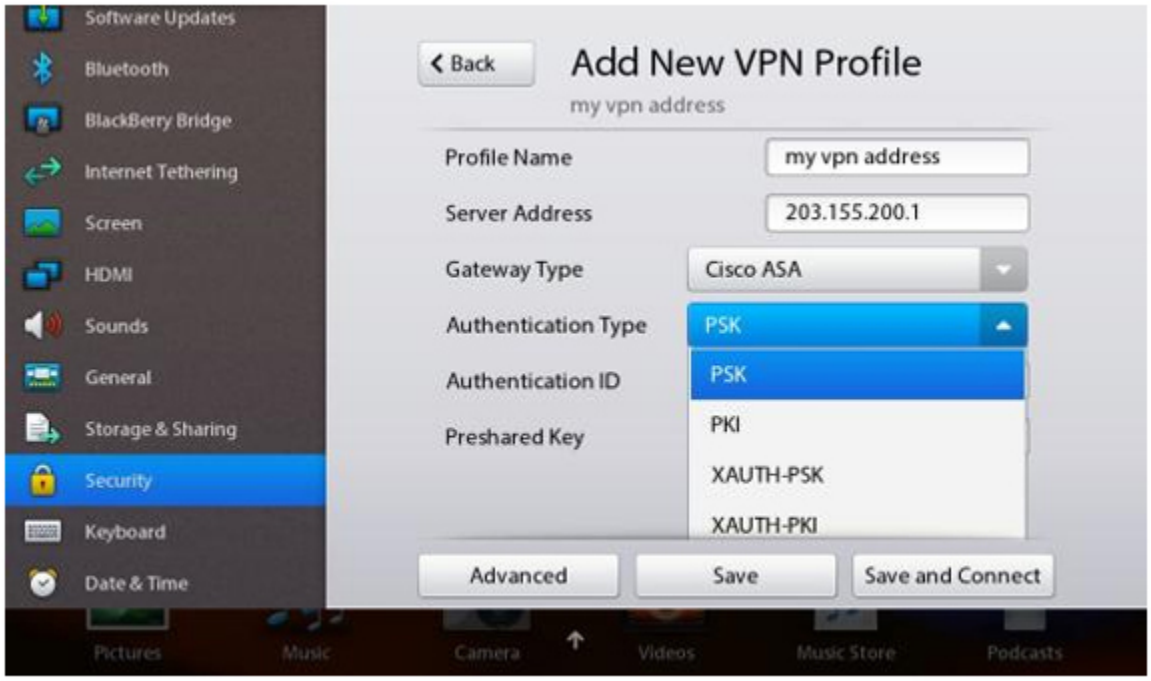
Figure 1. iOS VPN Client Configuration



Source: Apple

Source: Apple (December 2011)

Figure 2. BlackBerry VPN Configuration



Source: aussecurity (December 2011)

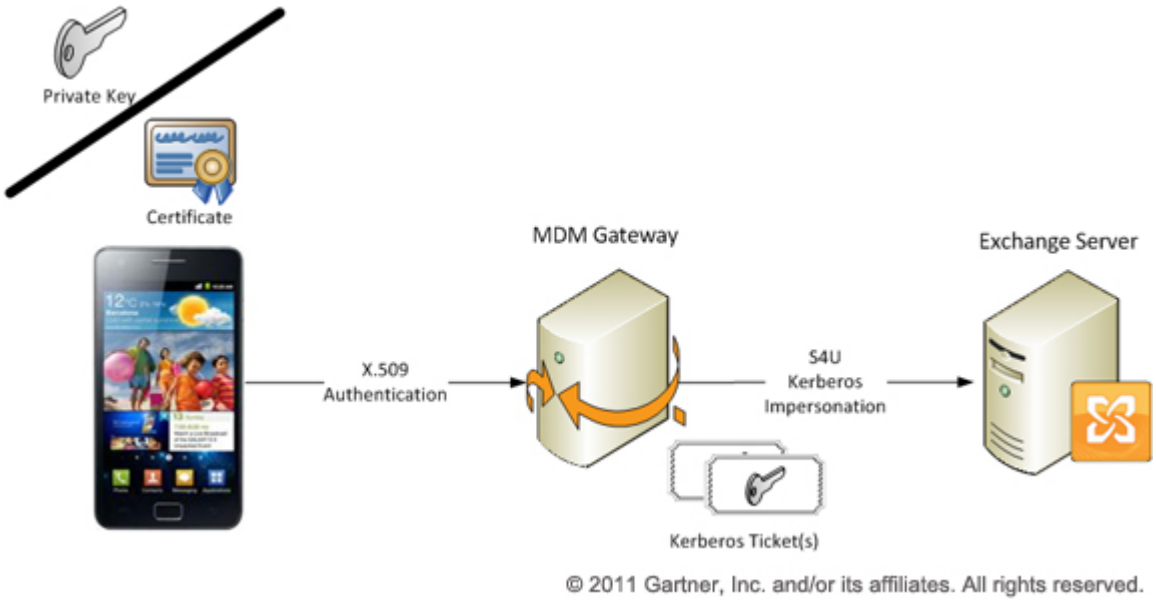
Certificate Authentication for Email

A growing concern among organizations is the embedded storage of the user's Active Directory password within the email client on the device. This is a difficult problem to solve because Active Directory typically requires the user's password to enable access to the user's exchange mailbox. If the mobile device is compromised, the Active Directory password can be retrieved and used for access to other sensitive resources. Several components are required to eliminate password storage within the email client. First, the email client must support certificate-based authentication. Additionally, the user's certificate must be published to Active Directory.

Kerberos Impersonation

The McAfee Enterprise Mobility Management (EMM) solution can also assist with eliminating the use of the Active Directory password for email authentication for iOS and BlackBerry. The solution includes a gateway that sits between Exchange and the mobile device. The gateway and mobile device enter into a mutually authenticated Secure Sockets Layer (SSL) session, then the Gateway transitions the certificate-based authentication to the user's Windows identity via the S4U Kerberos extensions. This functionality is very similar to using a federation solution that enables the transition of federated credentials to a Windows identity (see Figure 3). The user certificate on the mobile device need not be trusted by Active Directory. For more information on the S4U Kerberos extensions, see "Road Map: Replacing Passwords with Smart Card Authentication."

Figure 3. MDM Solution to Eliminate Passwords



Source: Gartner (December 2011)

Mobile Device Management

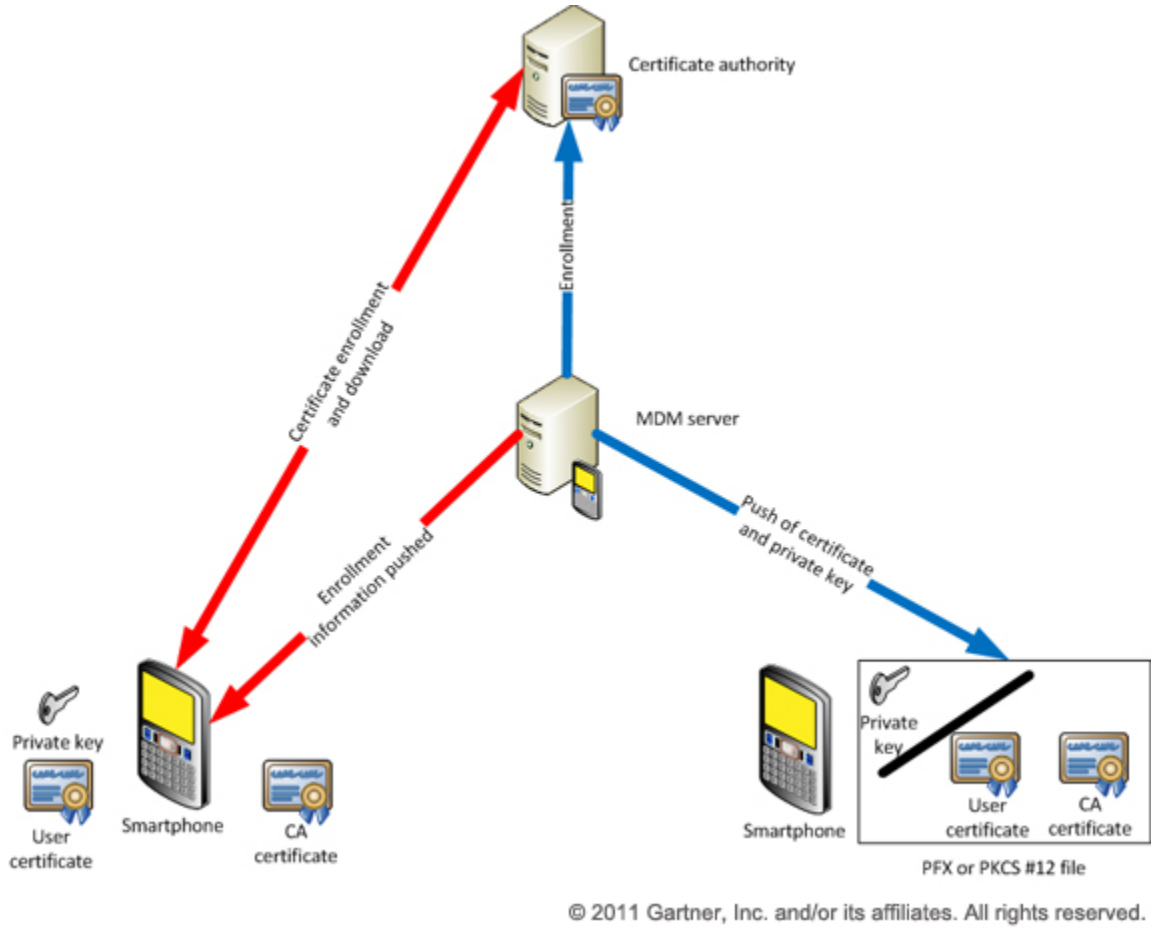
While theoretically possible, the manual management of certificates and trusted CA lists by users is not a desirable practice due to scalability and usability concerns. Therefore, a mobile device management (MDM) capability is an essential component for "over the air" certificate management (see Figure 4).

While this assessment examines MDM certificate management capabilities, MDM products provide a broad set of policy management capabilities and the ability to remotely wipe the device when the device is lost or stolen. For more information, see "Evaluation Criteria for Smartphone Mobile Device Management."

Third-Party MDM Solutions

Third-party MDM solutions (e.g., products from AirWatch, McAfee, MobileIron, Sybase, and Symantec) are generally the best choice when supporting a heterogeneous population of mobile devices (see Figure 4). In most cases, the MDM deployment process begins by the user downloading and installing the MDM application onto the device, unless the organization has personalized the device prior to its distribution. Most third-party MDM solutions have the ability to wipe the mobile device "over the air" after it is reported lost or stolen.

Figure 4. MDM and Certificate



Source: Gartner (December 2011)

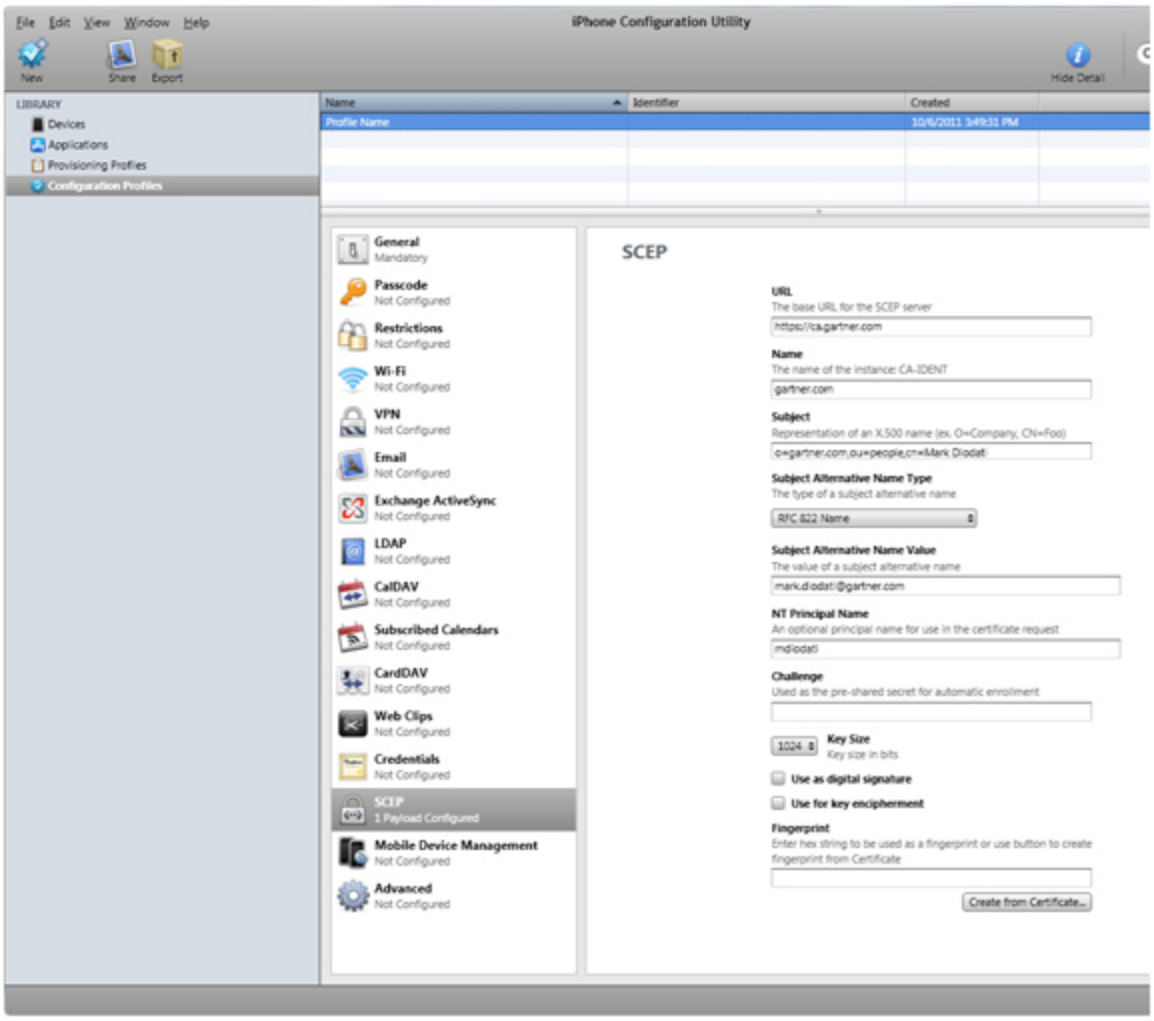
The SecureAuth Identity Enforcement Platform product is not an MDM solution, but it provides certificate management capabilities for mobile devices, including the Cisco and Juniper VPN client certificate stores. In addition, it couples this capability with a server-side implementation that can transition the certificate-based authentication to other token formats, includes Windows Kerberos and Security Assertion Markup Language (SAML).

Apple iPhone Configuration Utility

Apple does not provide an out-of-the box MDM solution. Additionally, Apple does not provide a remote wipe capability; remote wipe is typically achieved using ActiveSync (which is part of the Microsoft Exchange Server functionality). However, Apple's iPhone Configuration Utility provides some infrastructure to assist with certificate enrollment by creating Simple Certificate Enrollment Protocol (SCEP) enrollment profiles that can be distributed via email, network file share, Web server, and so on (see Figure 5). Traditionally, SCEP been used for the enrollment of certificate for network devices (e.g., routers), but it also works well with tablets and smartphones. Configuration profiles can also be published to third-party MDM software for distribution to mobile devices.

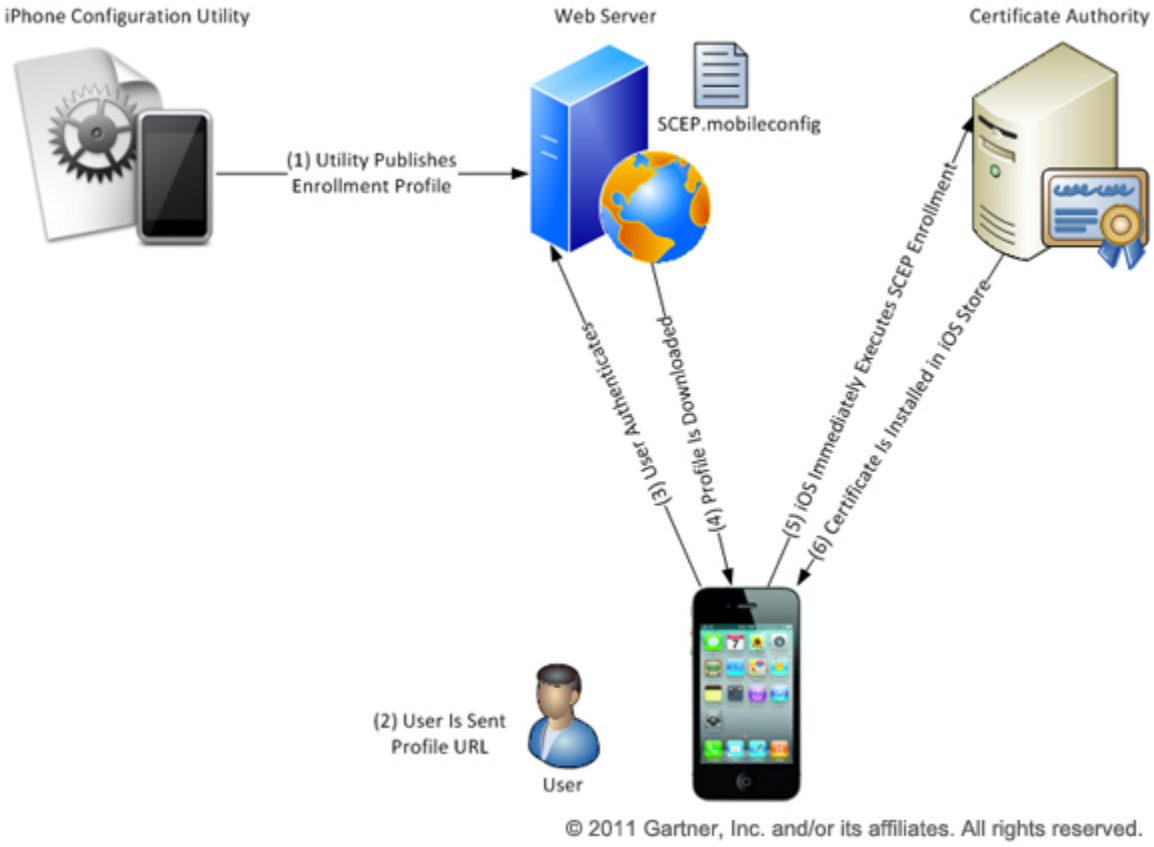
Figure 6 shows the distribution of iOS certificate profiles via a Web server. The organization creates and configures the Web server. Once the profile is downloaded via Safari, iOS executes the certificate enrollment using the information contained in the profile.

Figure 5. iPhone Configuration Utility SCEP Configuration



Source: Gartner (December 2011)

Figure 6. iOS Certificate Enrollment via Profile and Web Server (Simplified)



Source: Gartner (December 2011)

BlackBerry Enterprise Server

Of the three principal mobile device operating system vendors, only one vendor provides an MDM solution: Research In Motion (RIM). While some of the MDM vendors support the BlackBerry platform, most organizations with heterogeneous mobile devices deploy third-party MDM software for non-BlackBerry platforms. BlackBerry devices are typically managed by the BlackBerry Enterprise Server (BES).

While it does not directly distribute certificates and associated private keys to mobile devices, BES supports the distribution of enrollment URLs for both Public Key Cryptography Standard (PKCS) 10 and PKCS 7 browser-based enrollment, as well as SCEP. Because BES also provides a policy framework, BlackBerry mobile devices can be configured to auto-enroll for a certificate when the user first opens the device. BES does not have the ability to update the mobile devices trusted certificate list, which is required when issuing certificates from a private root certificate. Instead, the CA certificates are installed by the user via the BlackBerry Desktop Manager, an application installed on the user's PC.

Software OTP Client

One-time password (OTP) authentication systems have been the default strong authentication method within enterprises for at least 15 years. Traditionally, the form factor has been the hardware OTP device. This device possesses the OTP secret (i.e., symmetric key) and optional clock necessary to calculate passcodes. In recent years, the software-based OTP has become more popular as organizations seek to reduce cost and improve usability; the mobile device provides the storage and display (see Figure 7) and reduces the number of devices that the user must carry. For additional information on OTP systems and software-based OTP devices, see "On the Verge: Strong Authentication as a Service" and "Road Map: Replacing Passwords with OTP Authentication."

Figure 7. OTP on Android, BlackBerry, and iOS Mobile Devices

Source: RSA and Symantec (December 2011)

OAuth 2.0

OAuth originated as a protocol to help mitigate risks associated with the "password anti-pattern," which is a condition where a primary application wishes to access the resources in a secondary application. Traditionally, this condition has resulted in the primary application knowing the user's password for the secondary application, which is unpalatable from a security perspective. For example, consider a user who wishes to share Flickr photos with Facebook. In the past, the user revealed the Flickr password to Facebook. With OAuth, the sharing of passwords between the two applications is no longer necessary because the first application issues an access token that the second application uses to connect to the first application. Along the way to solving the password anti-pattern, OAuth is gaining popularity for rich mobile applications (RMAs) on mobile devices, including those applications without a secondary application. For more information on OAuth, see "Enterprise Use Cases for Open Identity: OpenID and OAuth."

Rich Mobile vs. Mobile Web Applications

Two categories of mobile device applications exist. The first is the mobile Web application (MWA). Its technology and protocols are reasonably well understood because the MWA leverages a Web browser for communication with resources. The result is that the typical identity and access management tricks of the trade, including federation, OTP devices, and WAM cookies function well. The protocol for accessing resources is fairly traditional; the browser interacts with resources on behalf of the user using the typical HTTP request/response mechanisms. But compatibility comes with a price; MWAs lack the usability that comes with access to the mobile device hardware (e.g., camera). For more information on WMAs, see "Mobile Web Applications." For more information on WAM, see the "Web Access Management Market 2007: Expanding Boundaries."

As users demanded a richer user experience (i.e., native "look and feel" and hardware device access), developers began leveraging the mobile device's native software development kits (SDKs) to build rich mobile applications (RMAs). The result is reflected in the adoption of RMAs versus MWAs. RMAs represent 65% of deployed mobile device applications. But MWAs are catching up. Over time, the adoption of HTML 5 may enable MWAs to have comparable usability as compared to RMAs while providing compatibility with existing Web architectures. OAuth 2.0 has emerged as a popular protocol for RMAs.

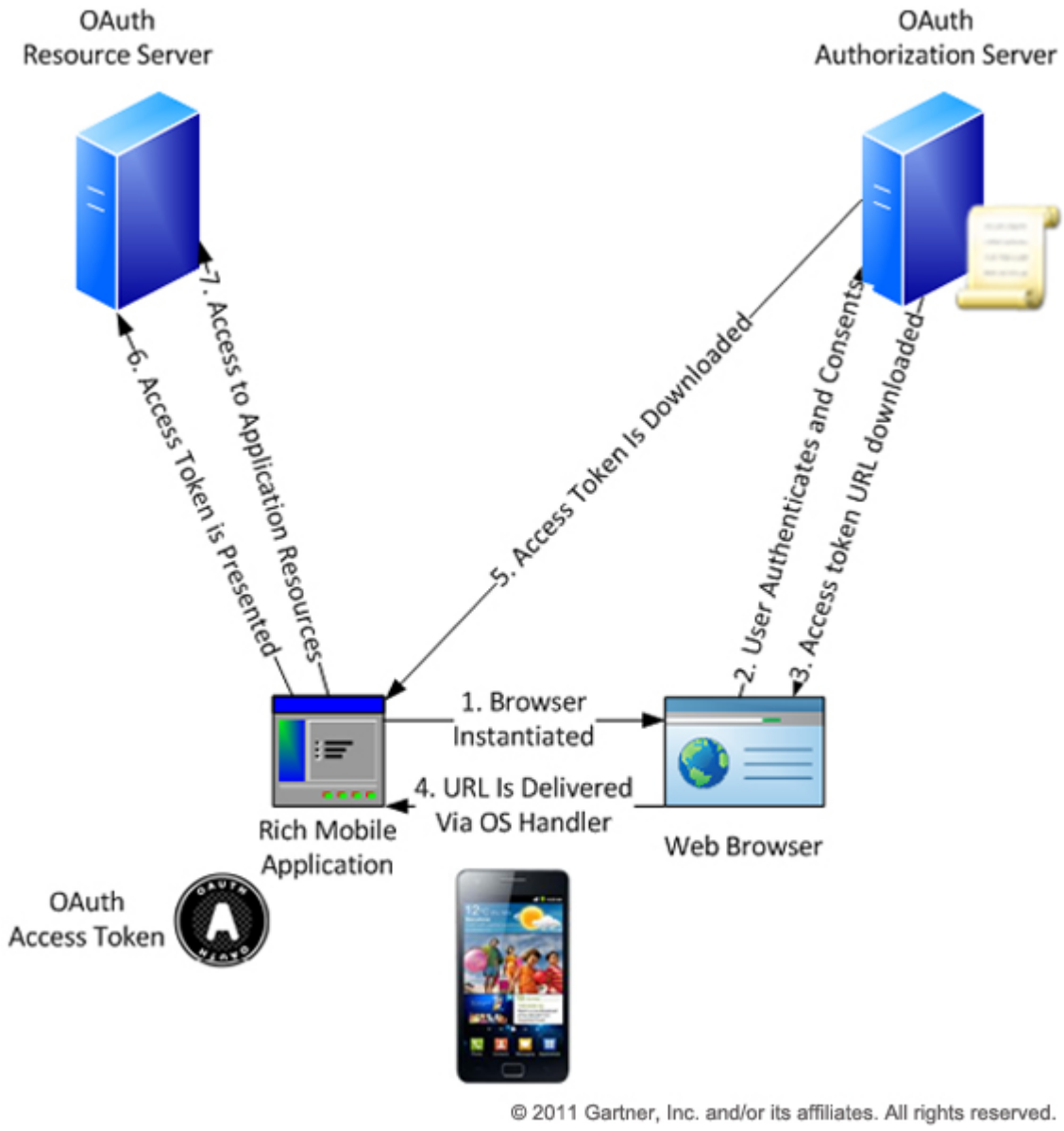
Components and Process

Five components are typically leveraged in OAuth (see Figure 8):

- Authorization server
- Resource server
- Mobile device Web browser
- Rich mobile application
- OAuth tokens

The authorization server and the resource server are traditionally associated with the same application owner and are therefore tightly coupled.

Figure 8. OAuth 2.0 Initial Authentication



Source: Gartner (December 2011)

User Authentication

When OAuth 2.0 is used with a rich mobile application (RMA), the user typically taps on the application icon to start the session. When the RMA requires access to resources, it instantiates the Web browser session (Step 1). The user authenticates to the authorization server (Step 2). The authentication method may be a password, X.509 certificate, or other method; the method is effectively decoupled from the OAuth protocol. The user also expressly consents to the primary application accessing resources managed by the secondary application.

Delivery and Use of Tokens

After the user has successfully authenticated, the authorization server delivers the URL location of the OAuth tokens back to the Web browser (Step 3). The browser recognizes the URL as outside its functionality and passes it to the RMA via an operating system handler (Step 4). The RMA downloads the access token from the URL (Step 5). The RMA presents the access token to the resource server (Step 6) for access (Step 7).

Although most user authentication occurs via the device's Web browser, some rich mobile applications will directly facilitate the initial authentication via browser emulation. This approach may be preferable from a usability perspective, but is not preferred from a security perspective because the user's password for one application may be shared with another application (a move back to the password anti-pattern).

The application downloads two tokens: the access token and the refresh token (only the access token is shown in Figure 7). The former has a shorter lifetime. The rich mobile application submits the access token to the resource server in order to gain access. The refresh token is used by the application to get a new access token when the access token expires.

The Importance of Identity Management

Relative to other token types like Kerberos tickets, WAM cookies, or SAML assertions, OAuth tokens have a much longer life and are valid for many user sessions. Although the access and resource tokens provide enhanced usability because the user need not re-authenticate for every session, these tokens introduce security considerations not normally associated with other token types. These tokens are not easily revocable, so it's critical that user identity management (IdM; e.g., de-provisioning) be performed in a timely fashion so that the resource server will not grant access to resources when presented with an issued access token.

Authentication vs. Single Sign-On

OAuth 2.0 is not a user authentication method. OAuth relies upon externalized user authentication methods. Its most interesting feature remains single sign-on (SSO) and session management for RMAs. Because the OAuth 2.0 refresh tokens are long-lived, the RMA need not force the user to re-authenticate for each session.

OAuth as Authorization Framework

OAuth 2.0's sweet spot is its affinity to support session management and SSO for RMAs. Although there is much hype regarding OAuth 2.0's authorization capabilities, the reality is that authorization is limited without additional emerging identity technologies (e.g., User-Managed Access [UMA] and Extensible Access Control Markup Language [XACML]). The former technology provides the necessary decoupling of the authorization and resource servers, which today are a closed loop within one security domain. The latter provides back-end storage and management of policy information.

In the near-term, more help is on the way to enable OAuth to better support authorization. First, the Internet Engineering Task Force (IETF) is currently working on a transparent OAuth token type that leverages the Java Web token (JWT — pronounced "jot") format. The current OAuth token is opaque and can be interpreted only by the authorization server. The JWT (which is digitally signed) can be read and therefore leveraged to express authorization to other components beside the resource server.

Near Field Communication

Near Field Communication (NFC) is a mechanism to support authentication and payment systems, via RFID, within a distance of less than 10 centimeters.

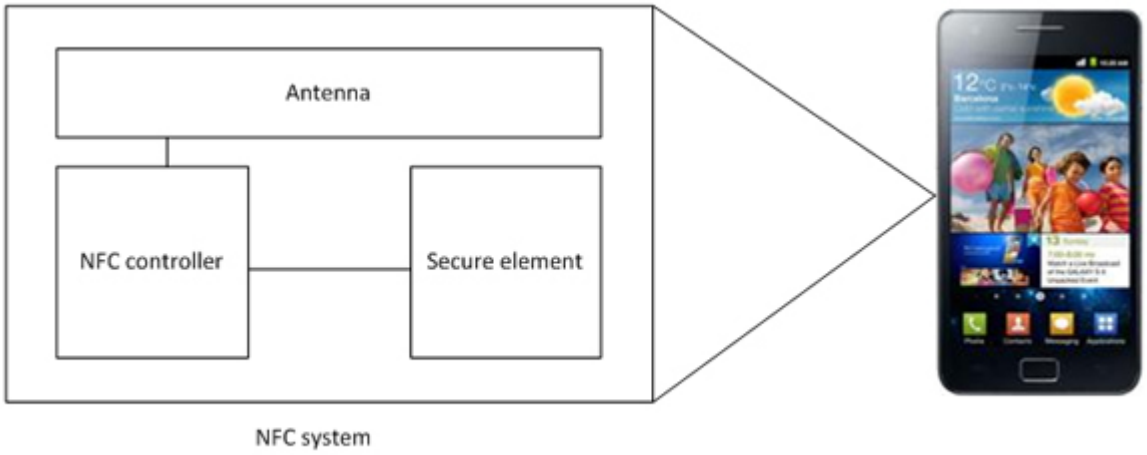
NFC-enabled mobile devices are currently hard to find, but more devices are coming to market. Over time, the percentage of NFC-enabled smartphones will increase, and NFC will therefore become a viable authentication capability. Gartner estimates that 50% of the smartphone population in 2015 will be NFC-enabled. Current NFC support varies across the three primary mobile operating systems. The adoption of NFC for payments will drive up the percentage of NFC-enabled mobile devices in the hands of end-users. One logical use case for NFC after mobile payment systems is enterprise authentication.

Technology

When integrated with a mobile device, NFC typically has three components (see Figure 9). The first component is the NFC controller, which enables the mobile device to communicate over a close distance with external components like point-of-sale terminals, physical access control systems, and PCs. The NFC controller interfaces with a minimum of two components: the secure element and the antenna.

The secure element is essentially a cryptographic smart card. The secure element provides a storage mechanism for sensitive information (e.g., credentials for payment and enterprise authentication). The secure element is usually embedded within the NFC component of the mobile device (i.e., system on a chip [SoC]). In lieu of a SoC implementation, the NFC controller can leverage the mobile phone's subscriber information module (SIM) card and communicate directly with the SIM card via the Single Wire Protocol (SWP), which bypasses the mobile device's operating system for improved security. The secure element can be a microSD smart card. In the case of SoC, the secure element is fused to the NFC controller.

Figure 9. NFC System With Mobile Device



Source: Gartner (December 2011)

One optional NFC component is the tag, which is a passive RFID chip with a small amount of memory — typically 96 bytes to two kilobytes (see Figure 10). The tag can be embedded in posters or attached to surfaces via its self-adhesive surface. The tag receives power via electromagnetic coupling from the nearby NFC reader. Coupling enables the NFC reader to power the tag and perform read and write operations. NFC tags can be personalized to provide commands to mobile devices (e.g., turning Bluetooth on when entering an automobile or disabling Wi-Fi when leaving home[1]). For more information on contactless technologies, see "Let's Get Logical: The Convergence of Physical Access Control and Identity Systems."

Figure 10. NFC Tag, Actual Size 43mm x 43mm

Source: http://www.toptunniste.fi

Source: Identified by ToP Tunniste: ToP Tunniste (December 2011)

Mobile Device Support

NFC-enabled devices are a rarity. Support varies across the three primary mobile operating systems and the mobile device vendors.

Android

The Android 2.3 operating system (aka Gingerbread, released in late 2010) introduced support for NFC technology. Android runs on mobile devices from many vendors; NFC support is hampered by the lack hardware of support. The availability of NFC-enabled mobile devices in the U.S. is limited to a few Samsung Galaxy Android-based mobile devices that shipped coincident with the Gingerbread release; since then, Samsung has introduced other models. Other mobile device manufacturers will be releasing NFC-enabled smartphones starting in 4Q11.

BlackBerry

In August of 2011, Research In Motion (RIM) introduced the BlackBerry bold 9990 and 9930 smartphones, which offer NFC capabilities. The two smartphones also run the new BlackBerry 7 operating system, which adds NFC support. Prior to the release of the two smartphones, RIM provided dormant NFC policy management capabilities within the BlackBerry Enterprise Server v5.0 SP3. The first policy setting is coarse-grained and enables or disables NFC conductivity on the smartphone. The second policy setting controls NFC access from specific applications on the smartphone.

Apple

Currently, NFC support for the Apple iPhone and iPad is nonexistent. While Apple has yet to release the functionality for the next-generation iPhone, Apple is rumored to have purchased NFC chips from its chip manufacturer, and has filed patents specifically for payment applications. One possible outcome is that Apple will enter the market with its own payment system to compete with Google Wallet and ISIS. The payment system might leverage a new iOS wallet application.

Tap to Pay

The primary near-term use case associated with NFC and smartphones is mobile payment. NFC-enabled devices will become more popular as a means of payment at point of sale (POS) terminals in retail environments. The user convenience is obvious; the user simply taps the POS terminal, which initiates the payment transaction. The information that passes from the smartphone to the POS terminal includes user and credit card data.

Google Wallet

In the spring of 2011, Google announced its payment system (Google Wallet). Google Wallet couples a mobile device application with NFC contactless/secure storage. When it launches, Google Wallet will support two credit cards: the Citi MasterCard and the Google Prepaid Card (see Figure 11). The mobile application will be initially provided by one carrier only — Sprint. Currently, no other carriers have been announced.

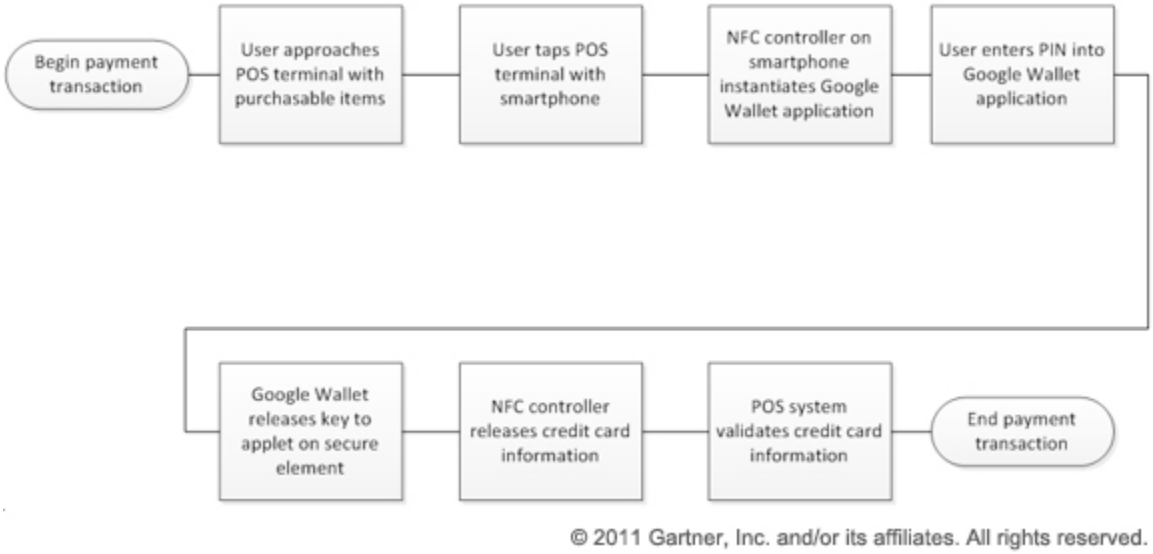
Figure 11. Google Wallet Application and First Data POS Terminal

Source: http://www.linuxfordevices.com
and http://www.amazon.com, respectively

Source: Gartner (December 2011); images: Google Wallet faces security challenges, say analysts (http://www.linuxfordevices.com/c/a/News/Google-Wallet-analysis/) and FIRST DATA FD-20 CONTACTLESS PAYMENT CARD READER FD20 UNIT (http://www.amazon.com/FIRST-DATA-CONTACTLESS-PAYMENT-READER/dp/B005E0CC5W?tag=food1d1-20), respectively

Retail organizations interested in accepting Google Wallet will partner with First Data for payment processing and new POS contactless terminals. As part of the personalization process, users must enter their credit card information into the Google Wallet application, which will write the information to the NFC secure element. The wallet will interface with Citi to validate its MasterCard credentials. Additionally, users may use Google Wallet with Google's own prepaid card. The prepaid card can be funded from any major credit card.

Figure 12. Google Wallet Payment Process



Source: Gartner (December 2011)

Google Wallet is not simply a payment system. It leverages the smartphone's geo-location capabilities to provide location-aware services, including coupons, and will transmit the buyer's loyalty card information as part of the payment transaction.

ISIS

In late 2010, AT&T introduced ISIS, a network for electronic payments. In the summer of 2011, ISIS received an additional $100 million in funding from Verizon and T-Mobile. Discover, Visa, MasterCard and American Express have also partnered with ISIS. ISIS is very much a competitor to Google Wallet; as of 4Q11, it has more partnerships with carriers and credit card issuers. It is noteworthy that MasterCard has announced partnerships with both Google Wallet and ISIS systems, essentially betting on both horses in the race.

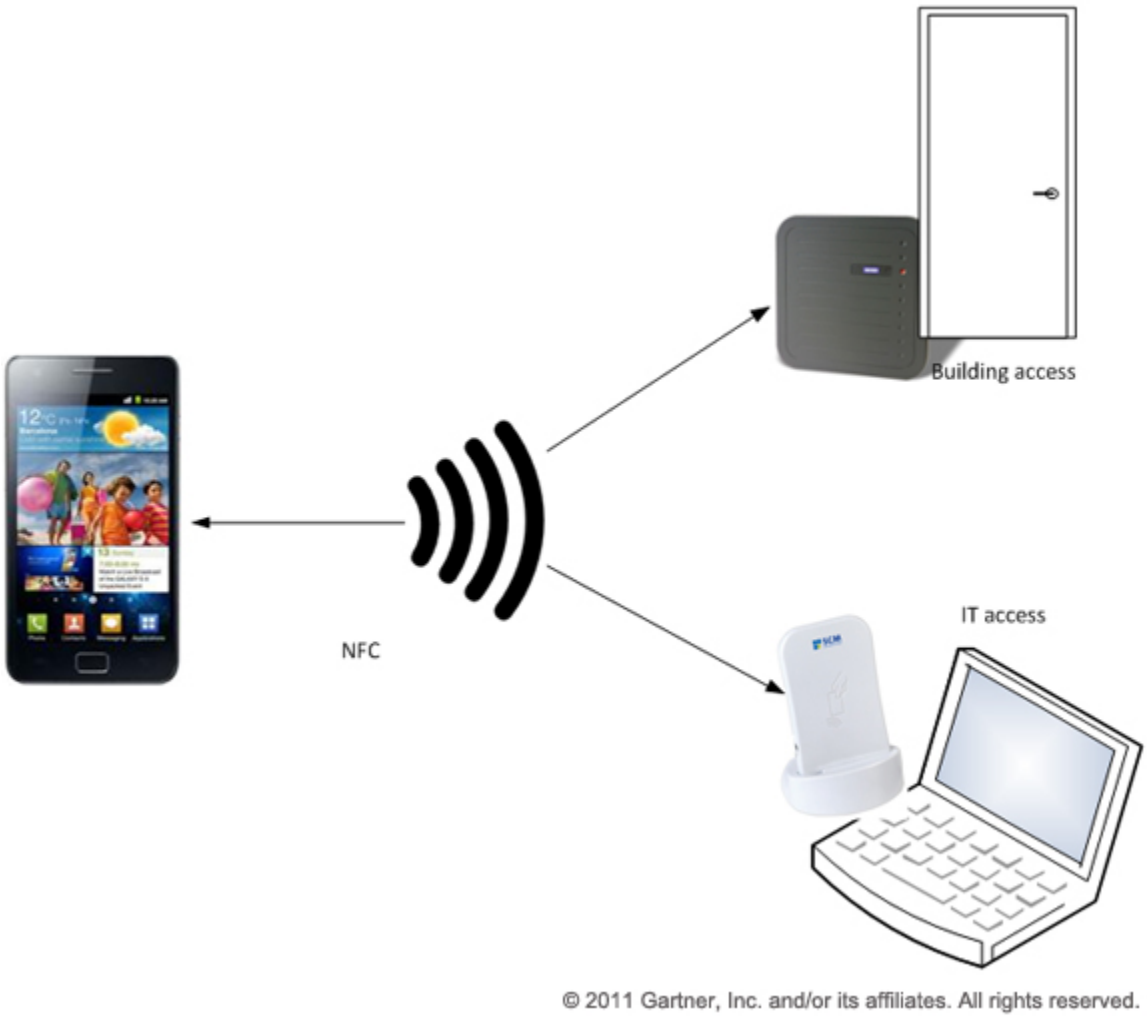
Both Google and ISIS will require that the merchant install contactless terminals at their point of sale (POS) stations. ISIS will require (at least in the short term) a different contactless POS terminal than Google Wallet. The ISIS wallet application will be available for mobile phones from AT&T, T-Mobile, and Verizon.

Device as Smart Card

While this specific use case is not associated with the use of the mobile device as a computing platform, it merits discussion because it will likely be become popular as NFC matures. X.509 certificates remain an essential component for identity-based smart card deployments. Contactless smart cards (e.g., the Personal Identity Verification [PIV] card) do not require physical connectivity to facilitate X.509-based authentication. Rather, the card can be used in close proximity to a reader. NFC utilizes the same contactless protocols (i.e., ISO 14443 family) as the PIV smart card. Over time, Gartner expects that the NFC standards body will expand the NFC tagging mechanism to support contactless smart card authentication via certificate.

The use of a mobile device as a contactless smart card is conceptually the opposite of the use of a contactless smart card for applications on the mobile device (see Figure 13). The NFC controller mimics a contactless smart card for access to physical and logical resources. For example, the mobile device emulates a smart card to enable certificate-based authentication to Windows and other public key-aware applications (e.g., browsers) on the PC. Additionally, this use case also would enable the user to authenticate to a tablet device using the smartphone as a smart card emulator.

Figure 13. NFC Phone as Contactless Smart Card Emulator



Source: Gartner (December 2011)

In the fall of 2011, RIM and HID Global announced a partnership. The BlackBerry NFC-enabled devices can store HID iClass credentials for access to physical environments. The iClass cards are not smart cards, but the RIM-HID Global integration is an important step.

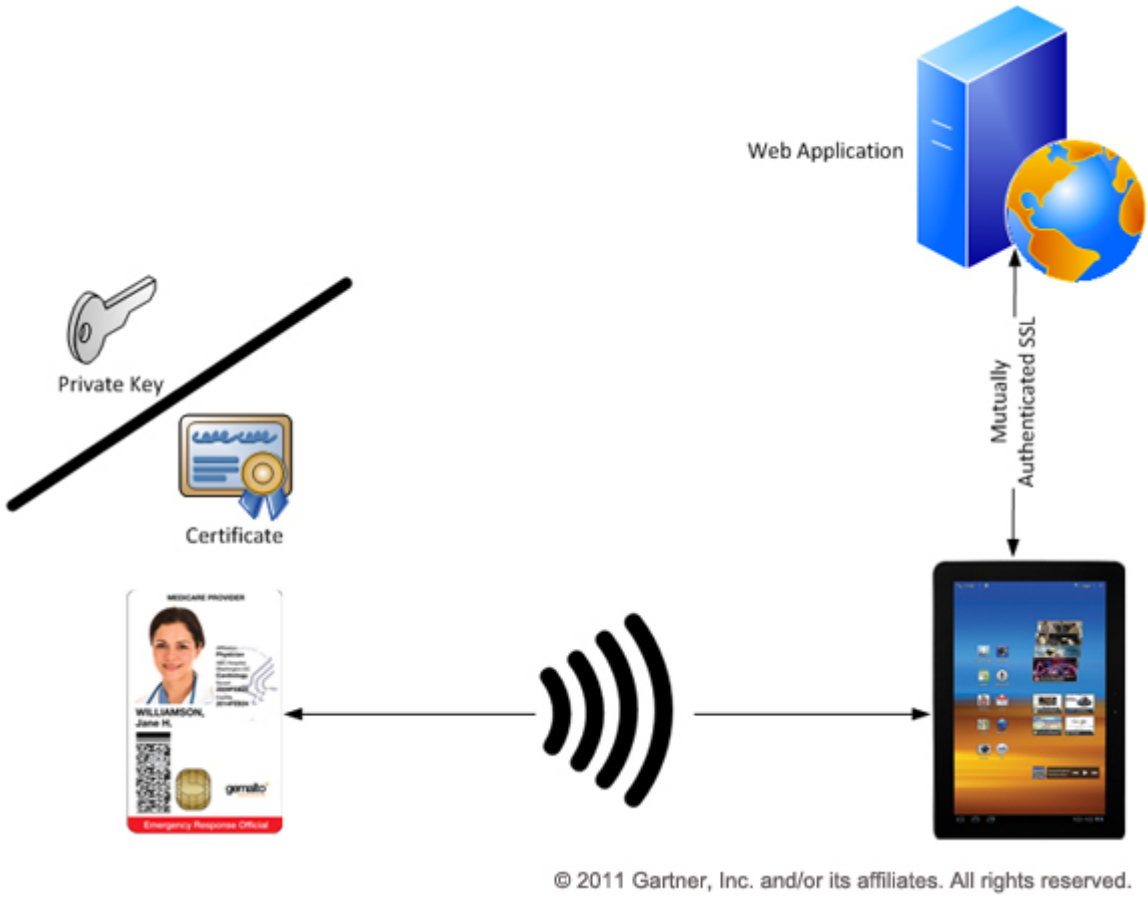
Device as Smart Card Reader

As NFC provides the contactless interface, it stands to reason that contactless smart cards could be used to authenticate to applications via the mobile device. The smart card provides the credentials for mutually authenticated SSL sessions, client authentication to email servers, VPN tunnels, and Wi-Fi authentication.

Currently, significant technological hurdles preclude the use of a contactless smart card in conjunction with a mobile device. Mobile device operating systems like iOS, Android, and

BlackBerry lack the cryptographic sophistication of their workstation analogues. For example, the Windows operating system — via its cryptographic APIs — can facilitate private key-signing functions with hardware-based devices, particularly when the private key is not resident in the operating system. Current mobile devices don't support the decoupling of the private key storage from its usage. Finally, organizations are concerned about the smart card PIN traveling "in the clear" via the NFC interface.

Figure 14. Contactless Smart Card Authentication Through Tablet via NFC



Source: Gartner (December 2011)

BlackBerry

Although it does not speak directly with the contactless smart card via NFC, RIM provides a solution for using contactless smart cards in conjunction with a BlackBerry mobile device (see Figure 15). The solution has been available for about five years and recently received U.S. government approval for use in conjunction with common access card (CAC) and PIV smart cards. Some organizations have a mixed opinion about the BlackBerry smart card solution and are looking to replace it with a microSD Smart Card option.

A smart card reader driver must be installed on the BlackBerry device. The smart card reader is worn on a lanyard around the user's neck. The reader does not obfuscate the graphical components on the smart card, which enables visual authentication of the user. The reader proxies the communication between the smart card's contact interface and the BlackBerry device's Bluetooth interface using AES 256 bit encryption.

The BlackBerry solution provides two specific capabilities. The first capability enables applications on the mobile device to leverage certificate-based functionality on the smart card. For example, the certificate and associated private key stored on the smart card can be used for mutually authenticated SSL sessions by the browser, and emails can be encrypted via Secure/Multipurpose Internet Mail Extensions (S/MIME). The second capability enables the locking of the mobile device when the smart card is out of range, which provides additional security controls against casual "snooping" and unauthorized access before the organization can wipe the smartphone.

Figure 15. BlackBerry Contactless Smart Card Solution

Source: BlackBerry

Source: BlackBerry (December 2011)

Device as OTP Display

NFC supports other kinds of authentication mechanisms. For example, idOnDemand has introduced a proof-of-concept OTP system that leverages a contactless smart card. Before issuance to the user, the idOnDemand smart card management system (CMS) installs an applet on the smart card. The applet contains an OTP symmetric key bound specifically to the user, as well as NFC tags for authentication to applications. The tags contain the necessary information to access a specific website. One of the tags is generic, and displays the OTP passcode for copy and paste into other applications.

The idOnDemand OTP system does not provide an authentication authority. Rather, the solution leverages Initiative for Open Authentication (OATH)-based authentication authorities (e.g., Gemalto's cloud-based OTP service). The OTP seeds (i.e., secrets) and associated user identity information are exported from the idOnDemand CMS to the OATH authentication authority. The smart card can also be personalized with X.509 certificates. Depending on the usage of this specific smart card, the organization may not need a contactless smart card chip.

Strengths

Mobile devices can support a variety of authentication methods and are made scalable by an MDM solution. Further, the emergence of NFC-enabled devices will provide additional authentication form factors:

- **Device maturity:** Smartphones and tablets are sophisticated computing devices and can provide access to most enterprise resources, particularly when coupled with virtual desktop technology for non-Web applications.
- **Authentication:** The three major mobile computing platforms (i.e., Android, iOS, and BlackBerry) also possess native authentication capabilities, particularly via X.509 certificates. The ability of native mobile device applications to consume certificates varies by operating system. BlackBerry and iOS provide the best interoperability between the device's certificate store and native applications (e.g., email, Wi-Fi, and VPN).
- **Infrastructure:** MDM solutions can provide the necessary certificate management capabilities to enable smartphone and tablet access to corporate resources. Mileage will vary depending upon the specific MDM solution and mobile device operating system.
- **OAuth 2.0:** OAuth 2.0 has emerged as the protocol between mobile computing and MWAs. OAuth 2.0's access and refresh tokens are ideally suited for MWAs because they are not impacted by the DNS restrictions for HTTP cookies. Many large service providers support OAuth 2.0, including Google, Facebook, and salesforce.com. Additionally, identity management vendors are beginning to support the protocol — a trend that is likely to continue.
- **NFC:** While only a small percentage of NFC-enabled smartphones exist, retail organizations, payment processing companies, banks, and major players like Google, Verizon, AT&T, and Sprint are betting on NFC for payments. Other major retailers have formally stated support for the platform. Because contactless payments at point-of-sale (POS) environments will drive the increased availability of NFC-enabled smartphones, in the long term, enterprises will be able to leverage NFC for authentication to physical and "logical" resources.
- **Software OTP:** Although occasionally perceived as "old school" technology, one-time password (OTP) authentication systems function well with mobile devices and browser-based applications. Mobile OTP software may be the best approach for consumer- or partner-based applications.

Weaknesses

Although mobile devices are very present in the enterprise, the maturity of the infrastructure required to support mobile authentication varies significantly:

- **Enterprise maturity of operating systems:** Smartphones and tablets are sophisticated computing platforms. However, when discussing enterprise maturity (i.e., policy management, applications, and authentication capabilities) significant variability exists among the three major mobile operating systems. BlackBerry provides a complete ecosystem for its devices via BlackBerry Enterprise Server. As expected, the ecosystem is homogeneous and supports only BlackBerry smartphones and tablets. BlackBerry devices rate the highest of the three operating system types. In recent years, iOS devices (i.e., iPhone and iPad) have emerged as sufficiently mature for the enterprise, but they require a third-party MDM solution or significant customization to extend the iPhone Configuration Utility. Android's young age and consumer orientation make it the least ready.
- **Smartphone and tablet limitations:** Smartphones and tablets are relatively new entrants to enterprise computing. Although the major operating systems support a variety of authentication types, they lack the capabilities that are provided by the Microsoft Active Directory and Windows 7 ecosystem. MDM products can help bridge the gap for policy and credential management, but don't expect the same capabilities as provided by the Active Directory environment.
- **NFC is coming, but . . . don't hold your breath:** Gartner estimates that only 50% of mobile devices will be NFC ready by 2015. The near ubiquity of NFC-enabled devices is an essential consideration for the use of employee-owned smartphones for contactless authentication. Additionally, the dependency for NFC in the enterprise will be the adoption of contactless payment systems (which will increase the number of NFC-enabled phones and contactless infrastructure). NFC payment systems will require the deployment of contactless terminals in the POS environment, which is a significant upgrade.
- **Mobile application duality:** Gartner has already seen customers struggle with the introduction of rich mobile applications, which leverage operating system APIs for usability and eschew Web browsers and therefore break compatibility with WAM systems (e.g., CA SiteMinder or Tivoli Access Manager). The result is that organizations must prepare for two distinct access management infrastructures for access to Web-based resources. Over time, the maturity and adoption of HTML 5 (and therefore mobile Web architectures) may provide a comparable level of usability as rich mobile applications while providing built-in compatibility with existing Web architectures.

Recommendations

Gartner recommends that enterprises have a specific strategy for each supported mobile device platform, continually monitor the progress of NFC, implement X.509-based certificate authentication for enterprise applications, and consider using software-based one-time password devices for consumer or partner access.

Be Specific

Because the three major mobile operating systems differ significantly in functionality and ecosystems, the enterprise must be ready to apply different strategies for the support of each platform. For example, an all-BlackBerry environment can be readily supported by BlackBerry Enterprise Server, RIM's MDM platform. However, BlackBerry support in third-party MDM systems will vary significantly, with many vendors offering no support. The iOS-based smartphones and tablets have good endpoint policy management and authentication capabilities, but lack the necessary centralized MDM capability. Therefore, plan to use a third-party MDM product with iOS. The Android operating system is the least enterprise ready of the three platforms and therefore requires more handholding than the other two. For example, the Android operating system lacks the ability to distribute certificates and update the trusted CA root list. Additionally, it lacks a native VPN client that integrates with the existing certificate management infrastructure on the phone. Some of the MDM vendors have road maps for the creation of their own specific VPN client for Android, and certainly Android clients exist from the VPN vendors.

Enterprises should embrace the IT consumerization of computing platforms, but that does not mean they must support all possible platforms. Enterprises should carefully evaluate the security and identity management capabilities of each proposed mobile platform and additionally determine the quantity of specific platforms for committing to supporting the platform.

Use X.509 Certificates and MDM

Although software-based OTP devices that reside on smartphones and tablets can provide authentication to enterprise resources, X.509 certificates will likely be the best option for enterprises that want native application support and enhanced usability. Certificates alone won't do the job; MDM functionality will be required for the scalable, "over the air" management of certificates and the associated user identity life cycle.

Use OTP Devices for Non-employees

Software-based OTP devices from mobile platforms can provide enhanced security and simpler deployment (compared with X.509 certificates). The deployment of OTP software to the mobile device may be achieved via self-service (in the case of consumers) or via some MDM platforms (specifically the BlackBerry Enterprise Server). From a usability perspective, software-based OTP systems don't readily integrate into native applications, but can be usable via a cut-and-paste process or integrated into the application via the OTP vendor's SDK.

Those organizations that have extensively deployed OTP systems may find that it is easier to stick with software-based OTPs for mobile devices than to deploy X.509 certificates (with their corresponding requirement for MDM software).

Plan for Different Mobile Device Application Architectures

Two specific mobile device methodologies currently exist: mobile Web and rich mobile. The former architecture is based upon using a standard Web browser on a mobile device but at the expense of usability (including access to the device's hardware). The latter architecture leverages the native

APIs on the mobile device for maximum usability, as well as a (typically) rest-based API for communication with the Web server. MWAs are largely compatible with existing Web security infrastructure (e.g., WAM federation and password-based authentication). Rich mobile applications provide a much better user experience but are generally incompatible with traditional Web security methods. Those organizations that wish to deploy rich mobile applications should plan for architectural differences to ensure that their identity and security mechanisms can secure these applications. In many cases, two separate infrastructures will be required to support both architectural styles of mobile applications.

Banish the Password

The storage of the user's Active Directory password on the mobile device represents a security exposure because the password can generally be used for access to other enterprise resources in addition to email. When possible, organizations should use other authentication mechanisms that can eliminate the password storage. In addition to password elimination, all organizations should ensure that their remote device-wipe processes are efficient, usable, and speedy.

Look, but Don't Leap on NFC

NFC-enabled smartphones promise increased security and enhanced usability. But as discussed in the Near Field Communication section, NFC-enabled devices and the supporting contactless readers for the enterprise are at least three years away. Enterprises should carefully monitor the status of available NFC-enabled smartphones and supporting infrastructure, including contactless readers for PCs and physical access control systems — but do no more. Once NFC has reached its "tipping point," enterprises should evaluate its applicability.

Recommended Reading

Some documents may not be available as part of your current Gartner subscription.

Paul DeBeasi. "Evaluation Criteria for Smartphone Mobile Device Management." *Gartner*. 20 Jul 2010.

Bob Blakley. "Enterprise Use Cases for Open Identity: OpenID and OAuth." *Gartner*. 29 Sep 2011.

Mark Diodati. "Let's Get Logical: The Convergence of Physical Access Control and Identity Systems." *Gartner*. 22 Feb 2008.

Mark Diodati. "Road Map: Replacing Passwords with Smart Card Authentication." G*artner*. 7 Jun 2011.

Mark Diodati. "Authentication." *Gartner*. 24 Feb 2011.

Kirk Knoernschild. "Mobile Web Applications." *Gartner*. 7 Sep 2011.

Mark Diodati. "Road Map: Replacing Passwords with OTP Authentication." *Gartner*. 3 Nov 2010.

Mark Diodati. "On the Verge: Strong Authentication as a Service." *Gartner*. 15 Jun 2010.

Mark Diodati. "More, More, More: The Challenge of Extended Enterprise Authentication Mobility." *Gartner*. 25 Sep 2009.

Mark Diodati. "Web Access Management Market 2007: Expanding Boundaries." *Gartner*. 30 May 2007.

Acronym Key and Glossary Terms

AD	Active Directory
BES	BlackBerry Enterprise Server
BYOD	bring your own device
EMM	enterprise mobility management
IAM	identity and access management
IdM	identity management
IETF	Internet Engineering Task Force
JWT	Java Web token
MDM	mobile device management
MWA	mobile Web application
NFC	Near Field Communication
OATH	Initiative for Open Authentication
OTP	one-time password
PIV	Personal Identity Verification
PKCS	Public Key Cryptography Standard
PKI	public-key infrastructure
POS	point of sale
RMA	rich mobile application
S/MIME	Secure/Multipurpose Internet Mail Extensions
SAML	Security Assertion Markup Language
SCEP	Simple Certificate Enrollment Protocol
SDKs	software development kits
SIM	subscriber information module

SoC	system on a chip
SSL	Secure Sockets Layer
SSO	single sign-on
SWP	Single Wire Protocol
UMA	User-Managed Access
WAM	Web access management
XACML	Extensible Access Control Markup Language

Notes

[1] Sarah Perez. "Weekend Project: Make Your Own NFC Tags." *ReadWriteWeb*. 8 Apr 2011. Accessed online 29 Nov 2011.

Regional Headquarters

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
USA
+1 203 964 0096

Japan Headquarters
Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

European Headquarters
Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Latin America Headquarters
Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509

Asia/Pacific Headquarters
Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600