

# Oracle Mobile Security

## A Technical Overview

ORACLE WHITE PAPER | MAY 2015





## Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



## Table of Contents

Disclaimer	1
Executive Overview	1
Oracle Mobile Security – Address Employee and Consumer use cases	2
Oracle Mobile Security Suite – Enterprise Mobility Management Solution	3
Assembling the Blocks – Core Solution Components	4
Oracle Mobile Security Container	5
Oracle Mobile Security App Containerization Tool	6
Oracle Mobile Security Access Server (MSAS)	6
Oracle Mobile Security Manager	7
Oracle Access Management Mobile and Social	7
OAuth Support	8
Extending Enterprise Security to Mobile Apps	9
Oracle Mobile and Social Client SDKs	16
Conclusion	17
Appendix: The New Mobile Computing Paradigm	18
Mobile App Development Models	18
Oracle Platform Security Services	19
REST	19
JSON	19
JSON Web Token	19
OpenID and OpenID Connect	19
OAuth	20



SAML	22
WS-Security and SOAP	22




## Executive Overview

Mobile computing gradually allows us to make the elusive “anytime, anywhere access” mantra a reality. More and more employees use their own mobile device in the workplace, a phenomenon known as “Bring Your Own Device” (BYOD), resulting in employees using the same device for personal and business purposes.

Many companies recognize the importance of personal mobile devices for business use: users can access corporate resources from their own mobile devices at their convenience to improve productivity, and companies can enable access to corporate resources through native mobile apps to improve user experience. However, introducing mobile devices in the enterprise presents additional security challenges. User-owned mobile devices contain personal information and have special privacy considerations. Mobile devices lack enterprise security controls but they need to blend seamlessly into the corporate computing landscape in order to preserve security without disrupting the workflow of the enterprise. Typically, applications running natively on mobile devices should integrate with the enterprise-wide identity governance and access control infrastructure for security and compliance reasons. While encouraging BYOD model, many organizations still benefit from issuing corporate owned devices for their employees, enabling personal or corporate only use.

Oracle provides a complete, modular mobile solution that includes (1) out-of-the-box mobile apps, (2) a developer's platform for designing and deploying mobile apps, and (3) integrated mobile app security. Each module can be used separately, based on customer requirements.

- **Oracle Mobile Apps:** In order to give the mobile workforce easy access to enterprise information, Oracle offers a wide range of out-of-the-box mobile apps including Oracle E-Business Suite, PeopleSoft, Siebel, and JD Edwards. In addition, Oracle Fusion App provides secure mobile access to Oracle Cloud Applications such as human capital management (HCM) or talent management.
- **Oracle Mobile Platform:** Enterprises need to easily create and deliver engaging user experiences on a secure platform, for any application, any device, and around any data. Oracle Mobile Platform leverages the Oracle Fusion Middleware infrastructure to provide a common technology framework, Oracle Mobile Application Framework (MAF), enabling developers to build and extend enterprise applications for Apple's iOS and Google's Android from a single code base leveraging enterprise data and backend services via the cloud, and consuming Oracle Mobile Security services.

- 
- Oracle Mobile Security: Organizations must effectively separate and protect enterprise apps and data on any market-leading mobile device, and manage the end-to-end lifecycle of user identities (extranet consumers or intranet employees) across all enterprise resources, within and beyond the corporate network, and into the cloud. Oracle Mobile Security helps organizations strengthen security, simplify compliance, and capture business opportunities around mobile and social access, for both BYOD and corporate owned devices.

This paper focuses on Oracle Mobile Security, a mobile device and mobile apps security solution including two main components:

- An employee-centric, comprehensive Enterprise Mobility Management (EMM) solution addressing a wide range of Mobile Device Management (MDM), Mobile Application Management (MAM), Mobile Content Management (MCM) and Mobile Identity policies by seamlessly tying to existing user identities and leveraging advanced features of the enterprise's backend identity management infrastructure for mobile access. Security policies, adhering to corporate needs, can be defined to enforce a complete device lock down (typically for corporate owned devices) and/or to separate personal apps from secure, "containerized" corporate, "off-the-shelf" apps and data (for BYOD cases).
- A consumer-centric mobile and social service that provides a software development kit (SDK) allowing corporate developers to secure custom enterprise apps for Apple's iOS and Google's Android devices, bridging the gap between mobile devices, social networks, and the enterprise's backend identity management infrastructure.

This document is designed for security architects, line-of-business managers, and Information Technology (IT) staff. To avoid disrupting the reading flow, we provide an appendix at the end of the document briefly describing the main technologies leveraged by Oracle Mobile Security.

## Oracle Mobile Security – Address Employee and Consumer use cases

Oracle Mobile Security addresses both mobile computing and social networks security requirements in order to allow organizations to fully benefit from these disruptive technologies while managing risk. As shown in Figure 1, Oracle Mobile Security includes security components leveraging two environments: Oracle Mobile Security Suite, and Oracle Access Management Mobile and Social.

- *Oracle Mobile Security Suite* is a standalone environment that provides a company's employees with secure access to business resources and centralized control of enterprise information on employee/corporate-owned mobile devices. Oracle Mobile Security Suite is a full EMM suite that provides the right level of control for your particular use case combining both MDM and MAM services to fit your

needs. Both allow a clear isolation between personal and enterprise information and apps. Access to enterprise information and apps requires user authentication using familiar credentials with various levels of security (username/password, digital certificates, or one-time passwords). Single sign-on (SSO) across intranet resources supports standard Kerberos and Microsoft Windows NT LAN Manager (NTLM) mechanisms as well as the Oracle Access Management platform's authentication and SSO capabilities. Enterprise data is protected using strong FIPS-approved cryptographic algorithms. Policies can be enforced to enforce device restrictions and/or to disallow offline storage, remotely lock/wipe enterprise information and apps, and set time-fence and geo-fence restrictions. Secure app features protect against the risk of leaking enterprise information through copy/paste, email, printing, and messaging. All communication to intranet resources goes through an authenticated Transport Layer Security (TLS) / Secure Socket Layer (SSL) tunnel ("AppTunnel") that can only be used by vetted (or "containerized") apps.

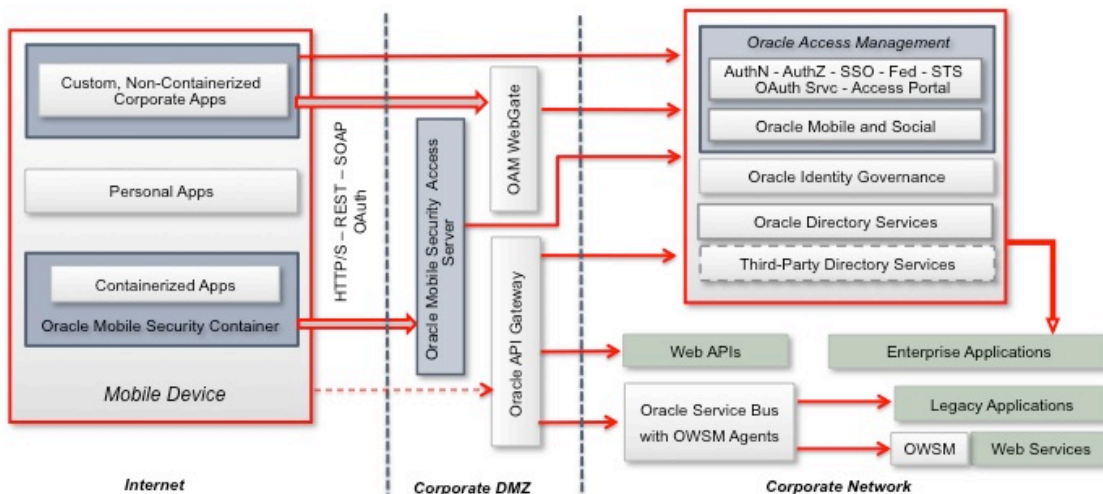


Figure 1: Oracle Mobile Security leveraging Oracle Access Management

- Oracle Access Management Mobile and Social is designed to secure custom enterprise mobile apps accessed by mobile users over the Internet. As an integral part of the Oracle Access Management platform, Oracle Mobile and Social leverages the enterprise's existing backend identity management infrastructure to provide single sign-on between browser-based and native mobile apps, strong authentication, device fingerprinting, and device-context-based authorization. Oracle Mobile and Social also provides client software development kits (SDK) used by developers to weave security into native mobile apps for tight integration with identity management. In addition, Oracle Mobile and Social enables enterprises to securely leverage social identities for personalization and federated sign-on to help organizations grow their business through social networks.

## Oracle Mobile Security Suite – Enterprise Mobility Management Solution

Oracle Mobile Security Suite (OMSS) provides a comprehensive Enterprise Mobility Management (EMM) solution that can address a mix of both BYOD and corporate owned models without compromising security, user experience or privacy. This best in class EMM solution enables organizations to easily move to a "Mobile First" strategy as part of the complete Oracle Mobile Platform, to build feature rich, cross platform, integrated apps and leverage an advanced and industry leading Oracle Identity and Access Management solution for securing corporate access.

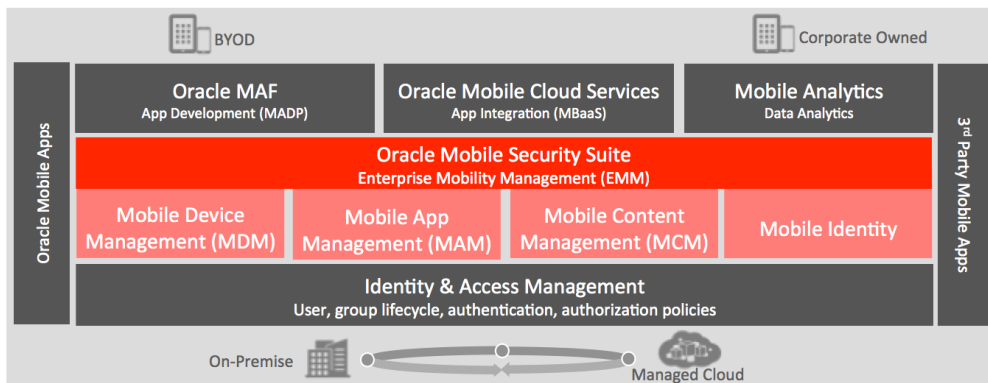


Figure 2: Oracle Mobile Security Suite – A platform for value added mobile services

OMSS secures corporate owned devices by enforcing device restrictions that conform to corporate security policies and providing remote controls to manage the device. Device management includes self-service device enrollment with enterprise authentication, automated provisioning and removal of corporate profiles, settings apps and certificates. Device restriction policies ensure non-compliant devices are detected continuously and are remediated to protect corporate data. Device controls are performed over-the-air allowing remote selective or full wipe, device lock/unlock as well as reset device passcode. It also provides a data rich device and app inventory that can be used to build a wide variety of reports using included BI Publisher.

For employee owned devices, it provides a container app that ensures security by isolating personal from corporate data and apps. The container can provide authentication, single sign-on across apps and a wide variety of data leakage prevention policies for any mobile app without requiring developers to understand and code for such complex security constructs and ever changing corporate policies. An app level SSL tunnel from the container ensures secure and user friendly access to corporate resources without needing a device level VPN. Corporate IT can rest easy as they still have control on this corporate container on the users device without affecting or intruding into end users personal data.

### Assembling the Blocks – Core Solution Components

Oracle Mobile Security Suite includes several components that work together to protect corporate applications and enable a secure enterprise workspace that meets enterprise security requirements without compromising the user experience.



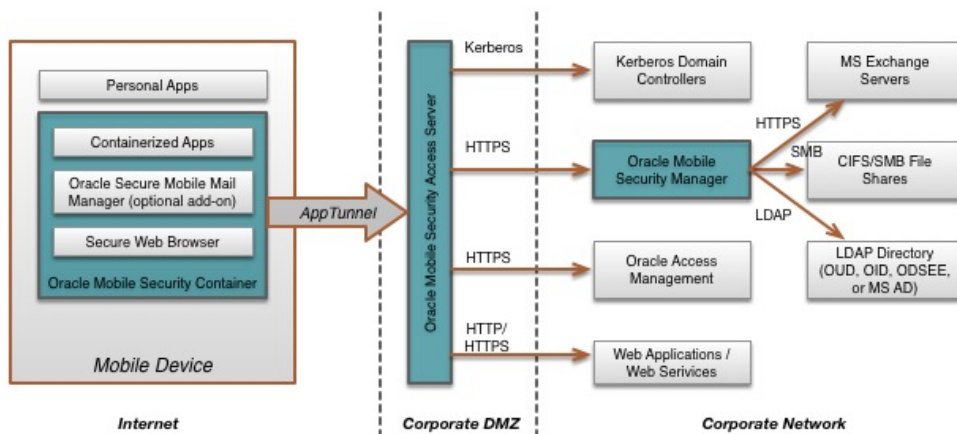


Figure 3. Oracle Mobile Security Suite components

The Oracle Mobile Security Suite components are distributed across the corporate DMZ and the enterprise intranet (or corporate network), and the Oracle Mobile Security Container is installed on the mobile device, as shown in Figure 3.


### Oracle Mobile Security Container

The Oracle Mobile Security Container is designed to hold “containerized” apps, i.e., apps that have been securely linked to their specific container. The Oracle Mobile Security Container includes a secure browser, file manager, and document editor, an optional Secure Mobile Mail Manager provided separately (See Figure 3). The Secure Mobile Mail Manager includes personal information management (PIM) apps such as a mail client, calendar, contacts, tasks, and notes synchronizing with corporate mail servers via the Microsoft Exchange ActiveSync (EAS) protocol. Many Oracle apps such as Oracle Business Intelligence (BI), Oracle Fusion Tap, Oracle Social Network, Oracle Enterprise Manager Cloud Control, Oracle WebCenter Spaces, etc., and a broad range of third-party enterprise applications can be containerized with the Oracle Mobile Security Container (Note: Oracle does not redistribute containerized third-party enterprise applications). All data at rest inside containerized apps on the mobile device is encrypted (encrypted data storage includes database, file store, cache, and user preferences). Data in transit through Oracle’s AppTunnel is encrypted using TLS/SSL with FIPS-approved algorithms.

When a web browser or other client program makes an unauthenticated request to the Oracle Mobile Security Access Server (described below), the Oracle Mobile Security Access Server responds with a redirect to the appropriate Oracle Mobile Security Container. Oracle Mobile Security Containers use a key hierarchy to protect data. All keys are derived from user credentials that are never stored. The key hierarchy involves multiple keys to support different sensitivity of data. For example, a unique key is used for the user’s authentication certificate, which is allowed to be open for a very short period of time. A different key is used for the browser cache, which must remain decrypted for an entire session. The main Oracle Mobile Security Container distributes and manages keys for the complete set of apps in the user’s secure enterprise workspace.

The secure container has three distinctive benefits over current mobile VPN solutions:

- **Device Trust vs. Gateway:** Oracle Mobile Security Suite extends your network’s Kerberos authentication trust directly to the user’s device instead of stopping at a gateway server sitting in the DMZ. Oracle Mobile Security Suite is significantly more efficient and secure than implementing “constrained delegation” offered



by VPN providers; a constrained delegation solution is not only less secure but also more cumbersome to set up and maintain.


- *Secure Container Password vs. Device Password:* The tradeoff between usability and security is magnified when dealing with consumer devices and BYOD programs. Corporate IT requires strong passwords to protect corporate data on BYOD devices. Conversely, users want simple passwords—or preferably no device password at all—so they can easily access social networks and other consumer applications. Requiring a device password is frustrating for users, as they are constantly using the device for non-enterprise purposes that don't require enterprise authentication. Oracle's solution provides the necessary balance between security and usability when dealing with BYOD programs by requiring a password only to access corporate applications.
- *Secure Container AppTunnel vs. Device-Level VPN:* Device-level VPNs provide a trusted, secure tunnel between a user's device and the enterprise's network. However, device-level VPN solutions are more appropriate for corporate-owned and secured endpoint devices such as laptops than for consumer mobile devices. Once a mobile-device VPN tunnel is open to your network, any app on that device has access to this secure tunnel. This is a huge security hole and a pathway to danger. With Oracle's solution, the connection from the mobile device to the enterprise intranet exists only between the secure container and enterprise servers.

### Oracle Mobile Security App Containerization Tool

The Oracle Mobile Security App Containerization Tool is a toolset for customers to inject security functionality for native, 3rd party or custom apps with zero code changes and to sign them with their enterprise distribution certificates. These containerized apps can then be deployed in the OMSS app catalog, distributed to the Secure Workspace based on policies and configured to share content, authentication, encryption keys, policies and participate in SSO between apps without ever needing to cache the password in the device. Once an app is containerized you now have control over authentication, network tunneling, data encryption, DLP controls and policy controls.

### Oracle Mobile Security Access Server (MSAS)

The Oracle Mobile Security Access Server is typically deployed in the corporate DMZ and multiple server instances can be deployed behind a load balancer for high availability and scalability. MSAS provides tunneled connections between the server and containerized apps. MSAS brokers authentication (strong authentication leverages HTTPS connections to Oracle Access Manager or Kerberos connections to Kerberos Domain Controllers as shown in Figure 3), authorizes, audits, and enables single sign-on for, and proxies requests to, their destination (resources in the corporate intranet). The Oracle Mobile Security Access Server acts as the terminating end-point of the tunneled connections initiated by the Oracle Mobile Security Container and containerized apps. The Oracle API Gateway (described later in this document) and Oracle Web Services Manager (OWSM) add security, threat protection, and throttling policies to an organization's REST API infrastructure (REST is described in the Appendix at the end of this document). Single sign-on (SSO) is supported through OAuth, OAM tokens, Kerberos, and NTLM. SAML (described in the Appendix) is supported through OAM or Kerberos integration with SAML identity providers such as Oracle, CA, or Ping Identity. The Oracle Mobile Security Access Server is integrated with the Oracle Access Management (OAM) platform and supports the retrieval of OAM and OAuth tokens for SSO to backend resources protected by OAM, OAG, and OWSM (OAuth is described in detail in the Appendix). MSAS also supports "virtual smart card" authentication by performing PKI authentication to Microsoft Active Directory protected by a PIN. Digital certificates are provisioned inside the Oracle Mobile Security Container app and only accessed after successful PIN validation.



MSAS integration with Oracle Access Management allows for context aware, risk based, step-up authentication. The Oracle Mobile Security Access Server is currently certified on Oracle Linux and Red Hat Enterprise Linux.

### Oracle Mobile Security Manager

The Oracle Mobile Security Manager is a Weblogic managed server running on either Oracle Linux or Red Hat Enterprise Linux. The Oracle Mobile Security Manager integrates with LDAP servers to provision users, assign and manage policies for Mobile Device Management and for accessing the Oracle Mobile Security Container, manage the app catalog, control the remote lock or wipe of the device and secure workspace apps (wiping the container removes all data and configuration for workspace apps), and set access control policies for the Oracle Mobile Security Container. Policies are assigned to users by associating policy templates with users and user groups. Available policy controls include Device Restrictions, Authentication (authentication frequency, failed attempt threshold, PIN strength for PKI); Catalog (apps, URLs, file shares); Container/Apps (compromised platform, location services, offline status, inactivity duration, data leak prevention (DLP)); Time Access (lock if outside time window); Geo Access (lock if outside geo-fence (e.g., city, state, country)); Devices (whitelist specific device models, specify minimum OS level); Browser (disable address bar, disable download); File Browser (allow/disallow, disable download, specify file server URL); PIM (mail server URL); Provisioning (invite template, PKI details). If a user is in multiple groups and has multiple policies, policy combinations are resolved following specific rules. The Oracle Mobile Security Manager (MSM) maintains the EMM policies, which are then associated to one or more user groups in the directory. MSM does not perform any user or group management but leverages these identities and groups directly (no synchronization) from the directory store. MSM uses Apple Push Notification Services (APNS) and Google Cloud Notification (CGN) over HTTPS to send notifications to devices. The Oracle Mobile Security Manager also exposes a WebDAV front-end to internal CIFS/SMB-enabled File Systems or Microsoft SharePoint servers, and enables browsing intranet file shares from the client.

### Oracle Access Management Mobile and Social

With more and more organizations establishing a presence on social networks, IT departments need support for social identities, which rely on more lightweight security standards than enterprise identities but are better adapted to the requirements of social networks. For example, some websites may require users to provide access tokens obtained from Facebook or Google in order to be authenticated to their services.

Oracle Access Management Mobile and Social includes a server that interfaces with existing backend identity management infrastructures. The Oracle Mobile and Social server acts as an intermediary between supported mobile client apps and backend identity services. This approach decouples the client apps from the backend infrastructure so that you can modify your backend infrastructure without having to update your mobile client programs. Oracle Mobile and Social includes the following functionality:

- Delegated authorization leveraging the OAuth standard (see more detail on OAuth support in the following section).
- Mobile Services connecting browser-based (HTML5) and native mobile apps to the enterprise identity management infrastructure, typically the Oracle Access Management platform.
- Internet Identity Services providing functionality that lets Oracle Mobile and Social be used as the relying party when interacting with popular, cloud-based identity authentication and authorization services, such as Google, Yahoo, Facebook, Twitter, or LinkedIn. By deploying Oracle Mobile and Social, you provide the user with multiple login options without the need to implement access functionality for each identity provider individually.

- User Profile Services providing a REST interface for LDAP create, read, update, and delete (CRUD) operations (customers use the same REST interface to build graphical user interfaces for apps), user self-service functions such as self-registration, profile maintenance, password management, and account deletion (see an explanation of REST in the Appendix at the back of this document). User Profile Services are also available as an OAuth resource.
- Access Management Integration Services for leveraging Oracle Access Management (OAM) through a runtime REST interface provided by an agent software development kit.

Customers can install the Mobile and Social service as a standalone component when not using Oracle Access Management 11g R2, or when using older versions of Oracle Access Manager.

Standalone mode is used in conjunction with the Oracle API Gateway (OAG) functionality (described later in this document), and User Profile services protected by JSON Web Tokens (JWT). (See the Appendix for a definition and explanation of JSON and JWT).

### OAuth Support

Oracle Mobile and Social leverages the Oracle Access Management platform's OAuth 2.0 service. The mobile client app must be configured in the OAuth 2.0 service to use Mobile and Social (see the interaction between the client app and the Access Management platform's OAuth service in Figure 4 below). Oracle Mobile and Social uses platform-specific application notification mechanisms to identify the clients, viz., Apple Push Notification Service (APNS) and Google Cloud Messaging (GCM). The user has an option to give his authorization to register each mobile OAuth app installation instance on a mobile device. Upon registration, the app is issued a client assertion token. The mobile client submits the client assertion token as an input parameter to use all of the OAuth access service endpoints. Upon mobile client registration, the client performs the OAuth access service interaction to obtain the access token used to access protected resources (Figure 4).

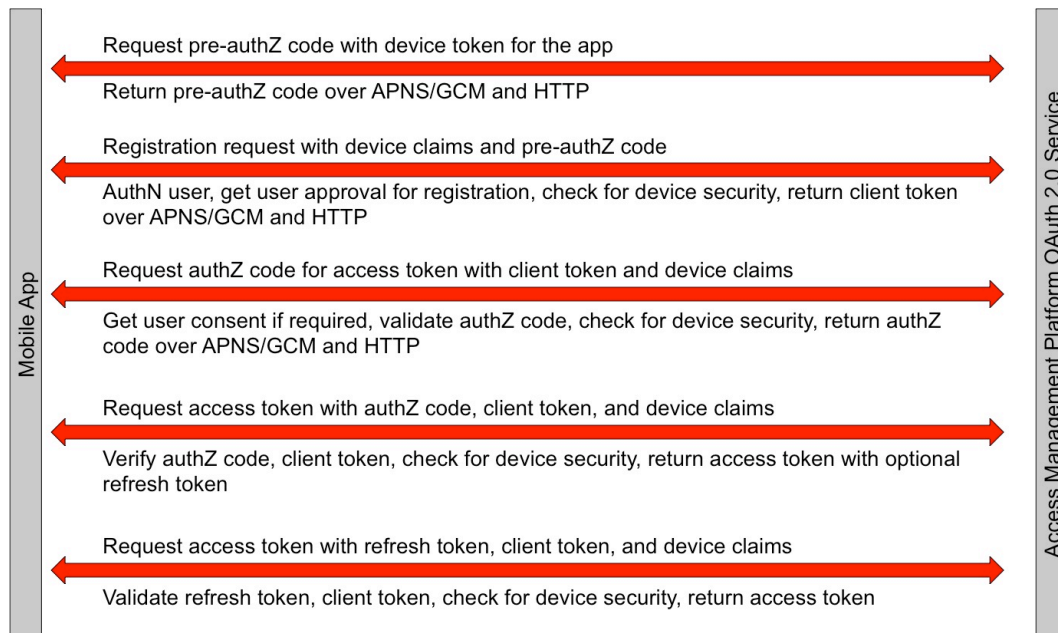


Figure 4: Interaction between mobile app and OAuth access service

**Note:** User authentication and user consent happen via a browser; the approval for device registration is optional; user consent for the authorization code is not required if the scope doesn't require user consent or the scope is not requested in the authorization request; the access token is presented to the resource server for access to a resource by the application. (Refer to the Appendix for details on the OAuth 2.0 process flow.)

## Extending Enterprise Security to Mobile Apps

By extending Oracle's unique identity and access management platform approach, customers can securely bring advanced mobile computing into the enterprise. In a typical reference architecture (as shown in Figure 5), Oracle Mobile and Social leverages multiple components of Oracle's identity and access management, including:

- Oracle Access Management (OAM) for web application authentication, authorization, and single sign-on.
- OAM's Adaptive Access component (OAAM) for mobile device fingerprinting and registration, risk-based authentication factoring in the mobile device context, and fraud detection.
- Oracle API Gateway (OAG) for first line of defense supporting multi-protocol and multi-format web services and web application programming interfaces (APIs), security gateway to cloud services, data redaction (in conjunction with Oracle Entitlements Server), identity propagation, and access to legacy applications.
- Oracle Entitlements Server (OES) for fine-grained authorization policies and access to mobile apps based on the mobile device context.
- Oracle Directory Services for direct access of mobile applications to LDAP-based user directories such as Oracle Internet Directory (OID) or Oracle Unified Directory (OUD).

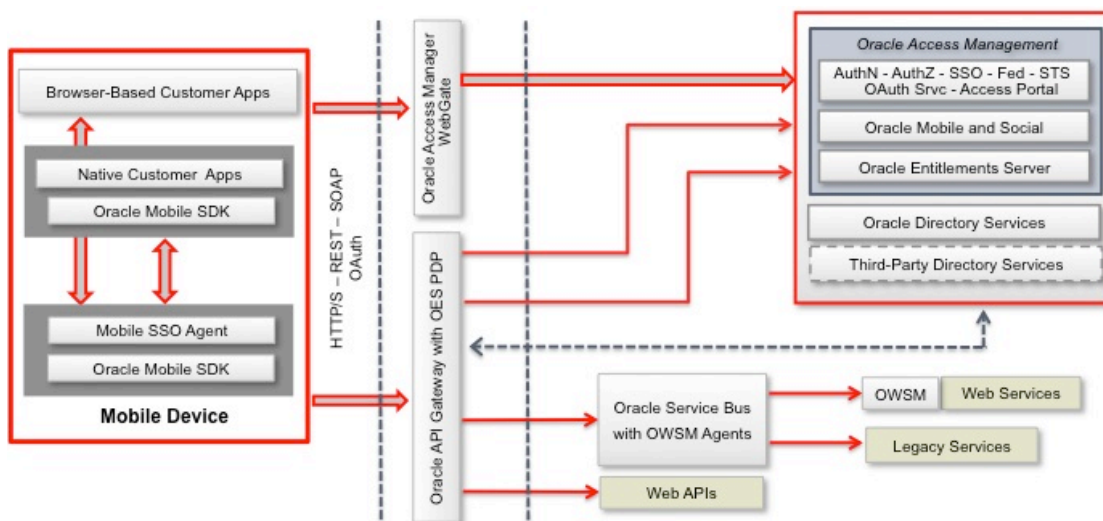


Figure 5: Extending the Oracle Access Management platform to mobile apps

## Authenticating Mobile Users

Oracle Mobile and Social provides authentication services that let you extend an existing authentication infrastructure to new mobile apps as well as the non-mobile applications already in place (strong authentication is provided by OAM's strong authentication services when required).



Mobile services support the following common token types:

- User Token granting the token bearer with the permissions associated with the person who has been authenticated.
- Access Token granting access to a specific protected resource, such as an enterprise web application.
- Client Token granting access to a non-mobile device, such as a web application or server application.

Mobile services also use client registration handles, which are similar to client tokens. A client registration handle represents a mobile client app running on a mobile device, such as a device running Apple iOS or Android. Because mobile devices and non-mobile devices present different security challenges, mobile authentication and non-mobile authentication are managed separately by Oracle Mobile and Social.

You can configure Mobile services and Social Identity services to work together. For example, you can use Social Identity services to let users authenticate with Google, Facebook, or Twitter, and you can use Mobile services to provide local authentication functionality, or generate a user token by accepting a user identity assertion from a social Identity Provider.

### **Authorizing Mobile Users**

Authorization is handled by Oracle Entitlements Server (OES), a policy-based, fine-grained authorization server designed to externalize authorization from applications and make decisions regarding the access of an authenticated party to protected applications.

OES provides organizations with fine-grained control over mobile users and apps:

- The REST APIs a given client or app can invoke.
- The business transactions a given user or mobile client can submit.
- What data a given user is able to access, and what the user can do with that information.


Authorization policies are defined in accordance with the Attribute Based Access Control (ABAC) and Extensible Access Control Markup Language (XACML) standards, allowing organizations to define policies that include environmental, device, resource, user, and transaction attributes and values. For example, a mobile user should only be able to submit a given type of transactions if the device is trusted and if the transaction amount is less than \$1,000.

OES and the Oracle Access Management platform provide a unique end-to-end solution that enables context-aware security policy management based on Identity Context, a service that is part of Oracle's identity and access management offering.

Identity Context is made up of attributes known to the multiple identity and access management components involved in a transaction. Identity Context attributes include user profile (typically stored in a user directory), application and enterprise roles, authentication type (weak, strong), device status (known, managed, trusted), device context such as location and configuration information, federation (partners' attributes), and risk assessment (pattern analysis), as well as network and other devices' information. Identity Context is shared across Oracle's identity and access management components, and Identity Context attributes, especially device context for mobile access, are made available to the security components designed to make access decisions, in particular OES for authorization.

Oracle's mobile access management solution including Oracle Mobile and Social, Identity Context, Oracle Entitlements Server, and Oracle API Gateway allows organizations to enforce fine-grained access control without making any changes to their existing backend systems and applications. Access control is enforced in the Oracle





API Gateway layer through which all REST traffic between mobile apps and an organization's backend systems is routed.

### **Integrating with User Directories**

User Profile services let web, mobile, and desktop applications perform a variety of directory lookup and update tasks. User Profile services make it possible to build an app that lets users in your organization access user profiles from mobile devices through REST calls to Oracle Mobile and Social (See Figure 7 later in this document).

### **Providing Single Sign-On Across Mobile Apps**

Mobile single sign-on (SSO) allows a user to run multiple custom mobile apps on the same device without having to provide credentials for each app. Both native and browser-based apps can participate in mobile SSO.

For mobile SSO to work, a customer app installed on the mobile device needs to be designated as a mobile SSO agent. With Oracle Access Management Platform 11gR2, mobile SSO agents are supported for Apple iOS and Android devices. Users build and brand their own mobile SSO agent apps using the Mobile and Social Client SDK for iOS and Android (described later in this document).

The mobile SSO agent app serves as a proxy between the remote Mobile and Social server and the other apps on the device that need to authenticate with the backend identity services. The agent can either be a dedicated agent (that is, an app that serves no other purpose), or the agent can be a business app that also provides SSO agent functionality (see Figure 5).


Mobile SSO agents and mobile SSO client apps using Oracle's client SDKs are configured on the Oracle Mobile and Social server. Typically, the client app sends the device registration request, the app registration request, and the user token request to the SSO agent, and the SSO agent makes the necessary acquisitions on behalf of the client app (device registration is handled by Oracle's client SDK). The app uses the client SDK authentication API. After authentication, the client app can then request any access tokens it needs because it has the registration handle and user token necessary to do so.

Mobile app developers benefit from using the mobile SSO agent because it handles device registration and advanced authentication schemes (including multi-factor and one-time password authentication), which means that this functionality does not have to be built into each mobile app. The mobile SSO agent also centralizes the task of collecting local device attributes to be passed to the server for risk-based authentication and Identity Context. When the mobile SSO agent is present, user credentials are never exposed to the mobile business app.

A browser-based business app can be configured to use a mobile SSO agent for authentication. When the browser-based request is intercepted by an OAM WebGate, the WebGate defers to the OAM server, which detects the mobile browser and sees that the authentication scheme is set to Mobile and Social. OAM calls the SSO service on the Oracle Mobile and Social server, which then redirects to the mobile SSO agent on the user's device. The SSO agent then requests an access token for the resource (on behalf of the business app) and redirects the browser to the URL of the business app with the access token included in the HTTP header.

Similarly, REST web service calls from native or browser-based apps are intercepted by Oracle API Gateway, which brokers SSO for these REST service invocations by validating the tokens against Oracle Mobile and Social, before a request to target backend services is granted.

Native and browser-based apps can be opened on the device without asking the user to provide credentials. A business app will fail if configured at the server for SSO, with the SSO app missing in the device. A business app can only directly collect and send credentials if the server-side is configured to allow that.



The mobile SSO agent can additionally time-out idle sessions, manage global logout for all apps, and help in selective device wipeouts. The SSO agent one-way encrypts and locally stores user passwords.

### **Using Mobile Services with OAM**

Enterprise resources may be secured today by a web access solution such as OAM, or they may be SOAP- or REST-based APIs and web services protected by Oracle Web Services Manager (OWSM) and Oracle API Gateway (OAG) as shown in Figure 5.

Oracle Mobile and Social supports multiple types of resources by offering two token types to secure the path between mobile apps and resources: OAM tokens (HTTP cookies) and JWTs (see the Appendix for a definition of JWT).

The Mobile and Social client SDK (described later in this document) handles authentication programmatically after the SDK collects user credentials using the credential collection user interface. The SDK then uses the Mobile and Social REST interfaces to authenticate the user with the token service configured for the app.

OAM-generated tokens are delivered as JSON payload by the Mobile and Social service (see the Appendix for a definition of JSON). The application developer extracts the received token and incorporates it into the resource request. When presented to an OAM interceptor (WebGate or AccessGate) by a mobile app, these tokens are validated by the OAM policy server, and they allow access to any type of resource protected by OAM without the OAM interceptor requesting a browser redirection for authentication.

JWTs are generated by the Oracle Platform Security Services framework (OPSS). (See the Appendix for an explanation of OPSS.) JWTs are issued and validated by the Oracle Mobile and Social server. These tokens are OAuth-compliant, and they can be enforced by a solution that can accept JWTs (e.g., Oracle API Gateway) and validate those tokens against the Mobile and Social service.

### **Protecting Mobile Apps and Web APIs with OAG**


Organizations build mobile apps to enable anywhere, anytime access to information stored in databases, content management systems, and, in some cases, mainframes on the corporate network. The information users should be able to access, and the various types of business transactions that users should be able to submit from mobile devices have in the past often been available to users through applications hosted within the corporate network, accessible through devices issued and managed by the organization. As such, corporate systems often have little, if any, security and compliance controls built in and instead rely on some degree of implicit trust.

Security and access control becomes a critical requirement now that organizations need to expose their internal systems to devices running outside the corporate network, accessed by internal and external users, from unknown locations, and over potentially unsecure networks. As a consequence, organizations must be able to control and audit what kind of business transactions can be submitted, what information leaves the corporate network, under what circumstances.

Mobile apps typically access corporate information through lightweight REST-based APIs as mobile devices lack support for more involved application, web services, and SOA-based infrastructures using the SOAP, Java Message Service (JMS), Message Queue (MQ), or even File Transfer Protocol (FTP) technologies that existing corporate systems often rely on.

Oracle's complete Access Management solution is designed to address these challenges. With Oracle API Gateway (OAG), organizations can expose internal systems and corporate data as fully secure REST-based APIs (using JSON payloads) without the need for any coding; this is achieved by virtualizing the existing backend SOAP or JMS services as REST APIs through OAG. Existing transport protocols and security tokens required for





authentication, identity propagation, and user claims (attribute assertions) can also be automatically transformed to address modern mobile requirements without changing existing backend systems. For example, an organization may only want to accept REST-based JWT tokens issued by the Oracle Access Management platform; once authenticated the tokens can be converted to SAML or any other type of token required by the SOAP-based backend systems.

OAG adds many capabilities to an organization's REST API infrastructure: API access, transactions, and the data requested/returned can be monitored and audited. Requests from mobile clients (or business partners and cloud applications) can be validated to ensure they are properly formed, are free from any malicious content and threats such as SQL injection attacks, denial-of-service attacks (even based on message payload content), viruses, and a large number of other XML, cryptographic, and other types of threats. Throttling policies can be defined to ensure that certain types of clients – perhaps based on different mobile apps or the user's subscription level (gold, silver, bronze) – can only perform a given number of transactions over a specified time interval, allow an organization to charge for API usage if desired, and ensure that rogue clients do not overload the system with excessive or malicious requests.

As a mobile and cloud access gateway, OAG provides the following control features:

- Validation of HTTP parameters, REST query and POST parameters, XML and JSON schemas.
- Protection against denial-of-service (DoS), SQL injection, and cross-site scripting attacks.
- Creation and exposure of virtual APIs tailored to mobile consumers and mobile apps formats.
- Throttling, rate limiting, and quota controls over web API traffic.
- Full OAuth 2.0 support, acting as an authorization server and resource server for both 2- and 3-legged scenarios.
- Context- and content-based routing.
- Data redaction and fine-grained access control over mobile business transactions leveraging OES.
- Reporting and analytics for tracking and metering API usage.
- Auditing and logging web API usage for each mobile client.
- Mapping of XML to JSON for consumption by mobile devices.
- Response caching for common web API requests, and response aggregation.
- Brokering of calls to cloud services, and centralized cloud connectivity.

Oracle API Gateway is part of Oracle's complete mobile access management solution and integrates with Oracle Access Management for authentication, validation of user tokens, fraud detection, and Identity Context propagation; Oracle Entitlements Server for authorization and audit of REST API access, transactions, and selective data redaction of the response payload; and Oracle Directory Services for user lookup and enrichment of the message payload.

### **Validating Device Identities**

Oracle Mobile and Social enforces new layers of authentication by requiring both device and app registration. Each app communicating with Oracle Mobile and Social downloads configuration parameters, and obtains an app registration handle that is required for all subsequent requests from that app. When SSO is configured, the app registration for SSO serves as a device registration, and that device registration handle is required for all subsequent

requests. Device registration is also subject to the policies and risk assessments available in the Adaptive Access component (OAAM) of the Oracle Access Management platform. These policies can trigger step-up challenges such as knowledge-based authentication (KBA) or one-time passwords (OTP) delivered via email or SMS text.

OAAM policies can be implemented for first-time access, so new device registrations require KBA, or more sensitive applications can require OTP (Figure 6). Policies can also be defined for specific users, allowing users with lower levels of access in with a username and password, but requiring an OTP for users with more privileged access (step-up authentication).

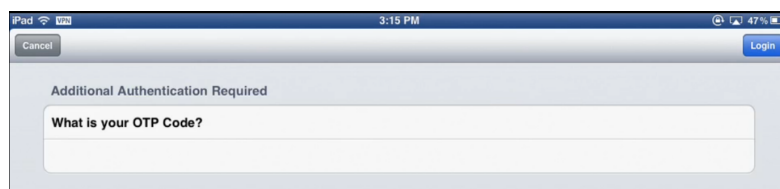


Figure 6: OAAM challenge for One-Time Password

For organizations wanting stricter control over the devices accessing their systems, devices can be required to be pre-registered in OAAM before their applications can authenticate or obtain tokens. Each request to the Mobile and Social service can be required to provide additional device Identity Context data such as operating system platform and version, Oracle Mobile and Social SDK version, “jailbreak” status, VPN status, telephone number, Media Access Control (MAC) address, and location data. This data can be used for device fingerprinting and data reporting, as well as fine-grained authorization to determine access rights (see *Authorizing Mobile Users* section above).

Through OAAM services, device Identity Context data can be stored and used to create a more comprehensive fingerprint that can be compared to previously stored fingerprints or device attributes, and policies can use the device data to determine risk and respond accordingly. On its own, Oracle Mobile and Social contains basic capabilities to detect changes in the device from request to request by using one of the attributes (such as a MAC address) to identify a unique device.

### Addressing Device Loss and Theft

Smart phone loss and theft create a high security risk for users and companies, particularly when these devices are used to access corporate resources. Oracle Mobile and Social working in conjunction with OAAM addresses this risk by providing a way to mark a device lost or stolen, and then implement specific policies that are enforced when a stolen device tries to access enterprise applications.

Since device Identity Context data is delivered to OAAM each time a device attempts to communicate with the Mobile and Social server, OAAM has the ability to challenge a user if the device has been reported lost, or deny any access from a device if the device has been reported stolen.

Additionally, if the device attempts to communicate with the Mobile and Social server after being reported lost or stolen, Oracle Mobile and Social can reply with an instruction to the device to wipe out all authentication tokens and handles stored in it (that instruction can also be leveraged by mobile apps as a trigger to clear cached data). The wipe instruction executes in one of two ways, based on the policies configured by the customer:

If OAAM is invoked and the device is marked as stolen, OAAM can deliver an error status of “blocked” with a sub-code of “wipe” which is executed if an application attempts to authenticate against Mobile and Social;

If Oracle Mobile and Social is configured with jailbreak detection on, and the policy is set to wipe when the device is jail broken, the wipeout happens as part of the re-authentication process.

Finally, if anomalies are detected in the access pattern, Oracle API Gateway can be configured to completely shut down access to REST APIs and corporate resources.

### Exposing Directory Data Through User Profile Services

LDAP directory services are used for many functions, including user self-service, company white pages, or help-desk user account maintenance. Oracle Mobile and Social makes directory services available to mobile devices, without a need for building LDAP clients (you still need a mobile client app such as a white-list app).



Figure 7: Looking up directory services from an iPad app

Oracle Mobile and Social provides REST interfaces to Oracle Directory Services as well as third-party directory services such as Microsoft Active Directory. Oracle Mobile and Social's User Profile services gives users and administrators access to configured directories for many common functions, and provides additional outward-facing security layered on the directory's own security.

User Profile services include the ability to search, view, create, update and delete users, groups and relationships (such as a user's manager), subject to both directory permissions and an optional layer of Mobile and Social permissions. These services are protected by either an OAM token or a JWT, and they can also require device and app registration.

User Profile services are essential for user self-service functions such as self-registration, profile maintenance, password management, and account deletion. Corporate or community white pages are another common application using User Profile services. A user can look up other users (see Figure 7), navigate up and down the management chain, and copy a contact into their mobile device's contacts app. Users can also update attributes on their own record, such as mobile phone number or home office address.

Mobile directory administrative tools can also be created. An authorized administrator can use a native mobile app to create users, set passwords, delete accounts, create and update groups, or change manager relationships.

## Logging and Auditing Mobile Transactions

Oracle Mobile and Social offers two types of auditing: auditing as provided by OAAM, and Mobile and Social audit events and diagnostic logs.

Oracle Mobile and Social uses post-authentication policies and challenge policies from OAAM. The audit events corresponding to those policies and their evaluation are supported through the integration of OAAM with the Mobile and Social server.

The Mobile and Social server logs each REST transaction to its own log. Diagnostic logs use Oracle Diagnostic Logging (ODL), and Mobile and Social logs go to the standard log files controlled by Oracle Fusion Middleware (log levels are configured through Oracle Fusion Middleware configuration settings).

## Oracle Mobile and Social Client SDKs


Oracle Mobile and Social acts as a proxy between a mobile user seeking to access protected enterprise resources and the backend identity and access management services that protect these resources (typically, the Oracle Access Management platform).

Oracle Mobile and Social provides client libraries that allow developers to add feature-rich authentication, authorization, and identity capabilities to registered mobile applications. On the backend, the Mobile and Social server's pluggable architecture lets system administrators add, modify, and remove identity and access management services without having to update software installed by the user.

Oracle Mobile and Social provides separate client software development kits (SDKs) for Apple's iOS, Google's Android, and generic Java for desktop applications. These client SDKs are designed to build identity security features into your mobile apps and enable you to use your existing identity infrastructure for authentication, authorization, and directory-access services. Current SDKs will support integration with the Oracle Mobile Security Container described in the first part of this document.

Client SDKs allow developers to get native mobile apps to interact with Oracle Mobile and Social through REST calls. The table below summarizes the functionality provided by the iOS and Android SDKs as well as the Java SDK.

Functionality	iOS / Android	Java
Build a mobile app that can acquire a client registration handle, user, and access tokens through the Mobile and Social server.	X	
Build a desktop application that can acquire client, user, and access tokens through the Mobile and Social server.		X
Interact with a user directory server and implement User Profile services.	X	X
Create a mobile single sign-on app.	X	



Oracle Mobile and Social client SDK for Apple iOS can be used in the Xcode development environment on a Mac, the Android SDK can be used in the Eclipse environment (supported on multiple operating systems).

## Conclusion

Mobile security is no longer just managing the device. It is about enabling secure access to corporate data and ensuring a consistent experience for a single digital identity across multiple access points. Security policies for mobile should be consistent with other access points defined and enforced in traditional enterprise centric IAM solutions. Leveraging a centralized IAM platform for mobile security ensures the control is restored back to IT when a corporate rolls out its mobility program. Having the knowledge of each user's privileges along with the information of their devices will provide IT complete visibility on user access that can significantly reduce audit exposure. But more importantly, incorporating the capabilities of a robust IAM platform opens new avenues for organizations to scale out their business to engage with customers, partners and the extended community. By leveraging Oracle's industry leading IAM infrastructure, Oracle Mobile Security provides exactly this kind of integrated and consolidated approach to securing the device, applications and access to corporate resources.

For more information, please visit Oracle's website at <http://www.oracle.com/identity>.

## Appendix: The New Mobile Computing Paradigm

With the advent of more powerful and functional tablets and smart phone offerings, mobile computing is fast becoming the “new normal.” However, because of the (still) limited capabilities of these wireless devices (reduced processing power and network bandwidth), new, more lightweight development and deployment techniques need to be used. This appendix briefly summarizes the new technologies and industry standards that characterize mobile and social computing.

### Mobile App Development Models

There are three ways to develop mobile apps for heterogeneous smart phones and tablets: web apps, native apps, and hybrid apps.


Mobile Web Apps	Native Mobile Apps	Hybrid Mobile Apps
<ul style="list-style-type: none"><li>▪ Online application accessed through mobile device browser</li><li>▪ Browser governs access to local storage and device services (camera, GPS, etc.)</li><li>▪ Highly reusable code</li><li>▪ Highly portable</li></ul>	<ul style="list-style-type: none"><li>▪ Application installed &amp; runs on device</li><li>▪ Optimized for specific mobile platform and form factor</li><li>▪ Direct access to local storage and device services</li><li>▪ Code reuse can be complex</li><li>▪ Portability requires work</li></ul>	<ul style="list-style-type: none"><li>▪ Application installed &amp; runs on device with HTML5 UI</li><li>▪ Optimized for specific mobile platform &amp; form factor</li><li>▪ Direct access to local storage and device services</li><li>▪ Code reuse simplified</li><li>▪ Portability simplified</li></ul>

Web apps run on a server and are rendered in the device’s browser. Web apps development requires HTML5, Cascading Style Sheets (CSS) and JavaScript. This mode of development targets the many web developers already familiar with these technologies, but it does not allow developers to fully leverage the specific features of the mobile devices, such as gestures or voice recognition.

Native apps are specific to each mobile device platform, for example iOS SDK for Apple’s iPhone and iPad, or Google Android SDK for Android. This approach requires developers to be familiar with the device platform SDK and development environment (Objective-C / Cocoa and Java, respectively). The benefit of this approach is that developers can leverage all the physical characteristics of the device through application programming interfaces (APIs). This is currently the most popular (and end-user friendly) form of mobile device app development.

Hybrid apps compile to each device’s native platform from a common source code. For example, PhoneGap is an HTML5 app platform that allows one to author native apps with web technologies targeting various mobile device platforms such as iOS and Android.

*Note:* Oracle Application Development Framework (ADF) is a multi-channel development environment that supports both web app and hybrid app development (leveraging PhoneGap). Multi-channel development means that Oracle ADF allows you to develop clients for various platforms (e.g., laptops, or mobile devices with the Oracle ADF Mobile extension), as well as server-side applications or web services over heterogeneous networks, all from a single code



line and homogeneous development environment. Oracle ADF Mobile provides support for development of business logic in Java for any device platform, and HTML5, CSS, JavaScript support provides cross-device user experience.

Oracle ADF Mobile authenticates against a server, such as the Oracle Access Management platform. (Oracle ADF Mobile can also authenticate against any basic authentication server). The Oracle Mobile and Social SDK (described earlier in this document) handles the authentication for both remote authentication servers and authentication against the credential store on the device. If the login succeeds, Oracle ADF Mobile receives an Oracle Access Management token that it reuses for each login connection (SSO). The Oracle Mobile and Social SDK saves the credentials in the device's credential store.

### Oracle Platform Security Services

Oracle Platform Security Services (OPSS) is Oracle Fusion Middleware's foundation security layer. OPSS is a Java framework that provides security as a service through open application programming interfaces. OPSS is used by developers to weave security into applications, separately from the applications (security information is stored as metadata in XML files). OPSS services (such as authentication, authorization, or session management) are consumed by the various Oracle Fusion Middleware components (such as Oracle WebCenter, Oracle Integration, and of course, Oracle Identity Management).

### REST

Representational State Transfer (REST) is a lightweight alternative to SOAP (defined below) for designing and accessing web services. REST is based on the HyperText Transfer Protocol (HTTP). REST uses clients to initiate requests and servers to process client requests and return responses (clients ask for a specific representation through HTTP context negotiation). While SOAP exposes operations that represent logic, REST exposes resources that represent data. REST allows for different resource representations: text, XML, or JSON (defined below).

### JSON

JavaScript Object Notation (JSON) is a lightweight data-interchange format based on a subset of the JavaScript language. In terms of functionality and use, JSON is comparable to XML (although the JSON data model is much simpler than XML, and arguably less powerful). JSON's popularity is growing and some very well known social network companies now use JSON to the exclusion of XML.

### JSON Web Token

A JSON Web Token (JWT, often pronounced "jot") represents a set of claims encoded as a signed and/or encrypted JSON object and transferred between two parties (examples of JWT claims include "expiration time" or "JWT identifier"). The Oracle Security Developer Tools (OSDT) package (part of Oracle Platform Security Services described above) includes Oracle JSON Web Token, a full Java solution that provides extensive support for generating and consuming JWTs.

### OpenID and OpenID Connect

OpenID (<http://openid.net/>) is an authentication standard that any web site can leverage without having to develop its own authentication system. As a user, the OpenID standard allows you to log in to multiple OpenID-enabled sites with a single openID (the name of the token (openID) is the same as the name of the standard (OpenID)). OpenID is not as widely used as it used to be.

OpenID Connect 1.0 (<http://openid.net/connect/>) is designed to provide a REST-based authentication layer on top of OAuth 2.0 (defined below).

## OAuth

OAuth (Open Authorization, hosted at <http://oauth.net>) is an IETF standard (<http://datatracker.ietf.org/wg/oauth/charter/>) that was originally designed to allow a User to transparently share his private data stored on one site (Service Provider) with another site (Consumer). For example, an OAuth-enabled photo-sharing site (Service Provider) can allow an individual (User) to use an OAuth-enabled printing web site (Consumer) to print the individual's photos without allowing the printing site to know about the individual's identity.

With the advent of OAuth 2.0, the original consumer-centric delegated authorization use case now extends to the enterprise and the Cloud. OAuth 2.0 enables a third-party application to obtain access on its own behalf (two-legged process flow) or obtain limited access to an HTTP service on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service (three-legged process flow described below).

### Parties and Components:

- *Service Provider*: A web application that allows access to its resources via OAuth.
- *User*: An individual that has an account with an OAuth-enabled Service Provider.
- *Consumer*: Web application that uses OAuth to access a Service Provider on behalf of a User.
- *Protected Resources*: Data controlled by the Service Provider, which the Consumer can access through authentication.
- *Request Token*: A value used by the Consumer to obtain authorization from the User, and exchanged for an Access Token.
- *Access Token*: A value used by the Consumer to gain access to the Protected Resources on behalf of the User, instead of using the User's Service Provider credentials (three-legged scenario).
- *Token Secret*: A secret used by the Consumer to establish ownership of a given Token (three-legged use cases).

OAuth 2.0 can be deployed in two different scenarios: Two-legged process flow without end-user involvement and three-legged process flow optionally requesting user consent.

### Two-Legged Process Flow

In a two-legged process flow, four entities are involved:

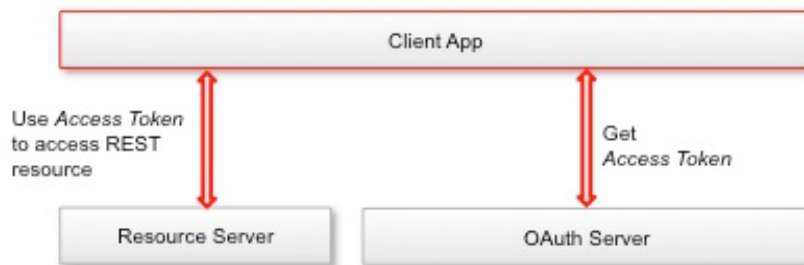
- Service provider or Resource server providing access to a service or a resource.
- Authorization Server, the authorization authority and token issuer (the Authorization Server is referred to as Token Service in Oracle Mobile and Social – the Token Service issues access tokens and their refreshed tokens to the client after obtaining authorization); Oracle Mobile and Social implement this functionality as an extension to the Oracle Access Management platform.
- Resource owner, a user who owns the protected resources. The resource server hosting the protected resources accepts and responds to protected resource requests using access tokens.
- Client, an entity that needs to access the service or resource based on the owner's consent (authorization). The term client does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices).



Two possibilities

- No user involved: the client accesses the resource on its own behalf.
- The user is involved: the authorization grant is implicit in the resource policy and the authorization is in the form of an assertion obtained out of band.

Bearer (access) tokens are issued to clients by an authorization server with the approval of the resource owner. The client uses the bearer token to access the protected resources hosted by the resource server (bearer tokens are used over HTTP 1.1 using the Secure Socket Layer (SSL) to access protected resources).



### Three-Legged Process Flow

The three-legged process flow is similar to the two-legged process flow except that the “third leg” is the act of obtaining the authorization grant through the OAuth 2.0 protocol. The Authorization Server can seek explicit consent from the user (consent orchestration). For example, a user may want to allow an Internet site (e.g., [www.ustream.tv](http://www.ustream.tv)) to access his Twitter account.



Oracle Mobile and Social supports both two-legged and three-legged OAuth 2.0 scenarios.


### Authentication With OAuth

OAuth 2.0 does not directly support authentication because OAuth 2.0 is currently missing an “authenticator”, for example a SAML assertion or a JWT. Oracle and Microsoft have submitted a draft to the IETF adding authentication to OAuth (<https://tools.ietf.org/id/draft-hunt-oauth-v2-user-a4c-01.text>).

Other ways to support OAuth authentication include:

OpenId Connect 1.0 (described above, mainly used by Google).

Abstract extensions to the OAuth specification such as (1) the SAML 2.0 Bearer Assertion Profile for OAuth 2.0 where a client application presents a SAML assertion to the OAuth authorization server in exchange for an OAuth



access token or (2) a JWT Bearer Token used to request an access token without a direct user approval step at the authorization server.

Proprietary authenticators used by consumer facing companies such as amazon.com and Facebook.

## SAML

The Security Assertion Markup Language (SAML) is a standard framework for sharing security information on the Internet through encrypted and digitally signed XML documents. The SAML framework includes 4 parts:

*Assertions:* How you define authentication information and attribute statements in XML snippets.

*Protocols:* How you ask (SAML Request) and get (SAML Response) the assertions you need.

*Bindings:* How SAML Protocols ride on industry-standard transport (e.g., HTTP) and messaging frameworks (e.g., SOAP, defined below).

*Profiles:* How SAML Protocols and Bindings combine to support specific use cases (e.g., browser profile, artifact profile, etc.).

In the context of WS-Security (explained below), only SAML assertions are used (the WS-Security framework provides the protocol and bindings).

## WS-Security and SOAP

WS-Security is an XML framework that specifies SOAP security extensions. (Originally known as Simple Object Access Protocol, SOAP provides an XML envelope that defines how messages must be structured and exchanged in XML-based web services interactions.) WS-Security also includes profiles that specify how to insert different types of binary and XML security tokens in WS-Security headers for authentication and authorization purposes (SAML assertions are the most common example of security tokens used with WS-Security).



CONNECT WITH US



[blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)

[facebook.com/oracle](https://facebook.com/oracle)

[twitter.com/oracle](https://twitter.com/oracle)

[oracle.com](http://oracle.com)

**Oracle Corporation, World Headquarters**

500 Oracle Parkway  
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**

Phone: +1.650.506.7000  
Fax: +1.650.506.7200

**Hardware and Software, Engineered to Work Together**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0515

Oracle Mobile Security – A Technical Overview  
May 2015



Oracle is committed to developing practices and products that help protect the environment