OMA API Program Q&A

What is the OMA API Program?
The OMA API program is a comprehensive framework supporting all of OMA's API-related activities. The OMA API Program maintains an inventory of all the API specifications produced and published by OMA, allowing you to view all the APIs, and sort through the list by API name, functional category, protocol binding, etc. In addition, the OMA API program ensures API governance, maintains the API roadmap and API best practices. Further useful information can be found by consulting the API News column.

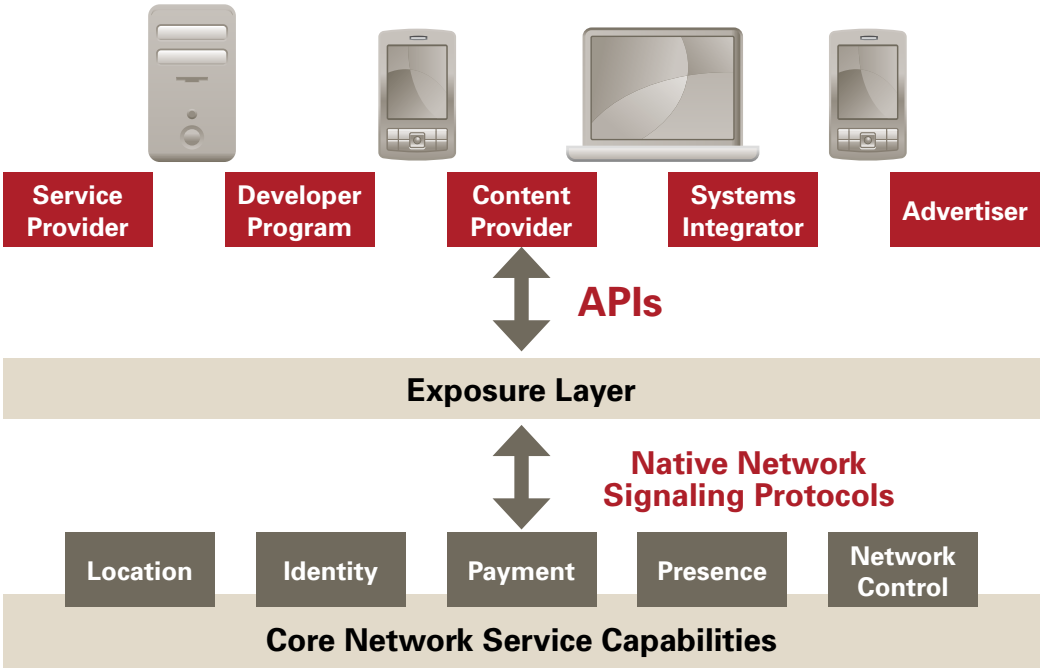
What is an API?
API stands for Application Programming Interface. This is an interface exposed to application developers, focused on a specific development community (e.g. Web 2.0). An API acts as an abstraction layer that encapsulates application level protocols and associated data formats, hiding complexity that is not needed by the API users.

Why standardize APIs?
As the number of APIs that perform the same functionality proliferate, fragmentation occurs. This limits developer access to subscribers, and operator and service providers' choices of development platforms and communities. The OMA API Program, through standardization, solves this problem.

What core network assets does the OMA API program allow operators to expose?
Any service capability, resource or asset in the operator's core network or on the device, can be made available through Application Programming Interfaces. These are the assets that developers need to build new apps and services on top of existing systems. Examples include Location, Messaging and Payment.

What specific OMA APIs are currently being deployed by operators?
Replace with: Besides the three Canadian operators who have commercially deployed the OneAPI profile of OMA RESTful Network APIs, OMA is not aware of many other public announcements of commercial rollout. However, in discussion with some of OMA's operator members, there is anecdotal confirmation of commercial deployments or pending rollouts, but no hard confirmations to date.

What is the primary benefit to developers who will create applications and services using the OMA API program?
The OMA set of APIs increases the portability of applications and services in order to reach the subscriber base of operators and service providers that deploy OMA APIs.



What is the relationship between the OMA API Program and GSMA OneAPI?

OMA and GSMA collaborate on the promotion of Network APIs, in a complementary fashion. GSMA OneAPI focuses on the commercial and regulatory aspects of deploying a third party portal that provides access to core network assets and capabilities. As such, GSMA describes commercial strategies, payment reconciliation, business models, and developer outreach and submits market requirements to OMA. OMA creates and publishes the technical API specifications and supporting files with formal operation definitions and protocol bindings, as well as test collateral and conformance requirements. OMA has also profiled several of its API specifications for GSMA OneAPI.

Does OMA specify Network APIs or Device APIs?

Applications may execute on a device or on a network server. Each has different characteristics (e.g. different run-time environment) and may be targeted at different developer communities. OMA specifies both Network APIs and Device APIs and distinguishes them as follows:

- Network APIs are exposed by a resource residing in the Network (e.g. a Server), explicitly NOT residing in or running on a Device
- Device APIs are exposed by a resource residing in or running on a Device (e.g. a client of an Enabler running on a mobile device)

Can I profile an OMA API?

An API Profile is a subset of supported operations and/or parameters, selected from an OMA API or API Protocol Binding.

For a full list of OMA's API inventory, click here.

What is a Protocol Binding?

APIs are specified in terms of abstract operations that can be invoked on a client or a server. A protocol binding is the formal set of rules to represent these abstract operations as messages of a specific communication protocol across the network. OMA API specifications can contain one or more protocol bindings, or can only contain abstract API definitions.

What is an Abstract API?

An API specification, which does not include a specific protocol binding for its operations, is referred to as an Abstract API.

What protocol bindings does the OMA API Program support?

OMA APIs presently provides the following bindings (this list is not exhaustive):

- *RESTful APIs:* These APIs use an HTTP protocol binding using the REST architectural style. The API operations are represented by HTTP methods (GET, POST, PUT, and DELETE) to allow the client to query the current state of a resource, or to change it
- *SOAP Web Service APIs:* These APIs include a SOAP binding which represents the API operations and their input parameters as an XML document within an HTTP POST method.
- *LDAP APIs:* These APIs represent their operations as LDAP AddRequest, SearchRequest, ModifyRequest, and DelRequest messages.

Can I contribute requirements or ideas for new or existing APIs for the OMA API Program?

Any OMA member company can propose a Work Item for the creation of a new API or for profiling or updating an existing one. As per OMA processes, every Work Item requires four supporting companies and subsequent review and approval by the

OMA API Development Process

Work Items and Requirements	OMA Specifies	API Specifications Deployed	
From OMA member companies From other industry bodies From application wholesalers From development platforms/ environments	Expose functionality of • Generic network services • OMA client or network service enablers Based on specific bindings or protocols Packaged according to programming language or style Grouped for specific development environments Modular functions within packages and groups	On operator networks Across wholesale application platforms Within existing client/server environments In development environments	**Developers have new standardized access to greater number of core services**

Technical Plenary. If you are not an OMA member yet, please consider Joining OMA.

How do I submit technical comments to one of the OMA APIs?
Technical comments and feedback should be addressed to OMA through the Technical Comments Form.

For most operators, there is still confusion and questions about the need for standardized APIs. What kind of uptake and support has OMA seen for its API program?
OMA has strong support among operator members for its API program. Operators have indicated that the repeated engagement on the level of technical interfaces is a problem. In September 2011, more than 30 OMA member companies endorsed the OMA API Program including 11 operators from around the globe.

What specific OMA APIs are you seeing deployed by operators?
Among the developer community, payment apps are very popular for obvious reasons. Among the operator community traditional telco apps such as location and SMS are popular. There is a focus on increased content and media driven requirements as well as feature enhancements to image, video and file share (GSMA RCSe).

Are there any other operators (in addition to Rogers Communications, Telus, and Bell Mobility) who have implemented GSMA OneAPI – a profile of OMA RESTful Network APIs?
Besides the three Canadian operators who have commercially deployed the OneAPI profile of OMA RESTful Network APIs, we are not aware of many other public announcements of commercial rollout. However, in discussion with some of our operator members, we do have anecdotal confirmation of commercial deployments or pending rollouts, but no hard confirmations to date.

How is the data model defined between SOAP vs. RESTful APIs—do you see a pattern with regards to "popularity" between SOAP vs. RESTful APIs?
The high-level functionality is the same, and both SOAP/WSDL as well as REST APIs use the same, or similar, abstractions of the resources and capabilities residing in the networks or on devices. However, both types of API are used by different sets of developers. The more web-savvy developers of web/internet (HTML) applications predominantly use the REST APIs, which are very appropriate for client-server communications. SOAP/WSDL APIs are more popular in business-to-business settings, and server-to-server communications. SOAP/WSDL Web Services provide more elaborate transactional support and are well suited for business solutions such as finance and banking.

A recent Oracle press release announced that they were embedding OneAPI to expose networks to developers. Do you know of any other vendors who are doing the same?
Most vendors have implemented parts or all of the OneAPI profile of the OMA RESTful Network APIs, as part of their SDP or Application Layer portfolio.

Some operators are introducing proprietary add-ons to standard APIs in order to differentiate their offering. Doesn't that defy the purpose of standardization?
Modifications and proprietary extensions will happen. OMA expects this. However, many of these operators and their vendors actively participate in the standards development process. Within OMA, we use clear and open change management procedures to manage extensions, modifications and error corrections to existing and published standards. Part of these proprietary extensions make their way into the next version of the API specifications. Our procedures and best practices for backwards compatibility allow us to manage this change process. Besides the changes that make it back into the standards, there will be proprietary extensions to differentiate various offers. As long as these extensions concern a delta on top of a standardized API, the extended interface still uses the same data model, design patterns, name space, authorization framework, etc. Therefore, much of the infrastructure and much of the application developer expertise can still be used. This is not a significant issue—80% standard-20% delta is definitely manageable.

Can you talk about proprietary operator solutions vs. standards and the perception of time-to-market issues?

- Standards are important for many reasons: change control is a powerful aspect of APIs—it is important to know that, once you have implemented your application, your API will work and that there is a mechanism for backwards compatibility—not the case with Facebook which uses a proprietary solution which may change at any time without prior notification.
- Regarding time-to-market, standards do take time, but not as long as people may think. For APIs, the time to publication is 6 months on average, which represents a considerable improvement of other standards development cycles.

For a full list of OMA's API inventory, click here.