**This paper was published in IEEE COMMUNICATIONS MAGAZINE, NETWORK & SERVICE MANAGEMENT SERIES, December 2011. This is an author copy for personal record only.**

# RESTful Web Services for Service Provisioning in Next Generation Networks: A Survey

Fatna Belqasmi[*1], Chunyan Fu[#2], Roch Glitho[*3]

[*]*Concordia University, Canada*

[1]fbelqasmi@alumni.concordia.ca

[3]glitho@ece.concordia.ca

[#]*Ericsson Canada, Montreal, Canada*

[2] chunyan_fu@hotmail.com

*Abstract*—**Next Generation Networks (NGNs), as envisioned by ITU-T, are packet-based networks, capable of provisioning consistent and ubiquitous services to end-users, independently of the network, the access technology and the devices used. RESTful Web services are now being contemplated as a technology for service provisioning in NGNs. They are emerging as an alternative, which may be more adequate than SOAP-based Web services in some cases. SOAP-based Web services are modular applications that can be discovered and invoked over a network. RESTful Web services, on the other hand, are defined as a network architectural style for distributed hypermedia systems. This paper presents a survey on RESTful Web services for service provisioning in NGNs. It introduces the concept of RESTful Web services and reviews the state-of-the-art of RESTful-based-service provisioning in NGNs. It also provides an evaluation of the overall suitability of RESTful Web services for service provisioning in NGNs, and discusses research directions. RESTful Web services do show significant potential for service provisioning in NGNs. However, open issues such as publication/discovery and mechanisms for the development of complex session-based services need to be solved before its full potential can be realized.**

***Keywords***— RESTful Web services, SOAP-based Web services, Next Generation Networks

## I. INTRODUCTION

Next Generation Networks (NGNs), as envisioned by the International Telecommunication Union (ITU), are packet-based networks, capable of provisioning consistent and ubiquitous services to end-users, independently of the network and the access technology used [1]. The concept of NGNs has emerged in the mid-2000s' to provide a long term vision for telecommunication networks after realizing that the first generation of packet-based telecommunications networks deployed in the early-2000s' did not cater to all the needs introduced by new applications. [2] provides an overview of the ITU-T NGN vision and explains how the 3GPP IP Multimedia System (IMS) is a first step towards this long term vision. IMS is a key component of the third generation telecommunication networks that are currently being deployed. It is also a key component of the emerging fourth generation telecommunications networks. NGNs with varying features have now been deployed by most telecommunications network operators.

Figure 1 depicts a generic NGN that embeds the ITU-T vision. It comprises a transport layer and a service layer. NGNs decouple the service and transport layers as shown in the figure. Furthermore, they provide support for generalized mobility, which enables end-users to communicate and access services, independently of their location, and the access technology and devices they use. In addition, NGNs endow end-users with unrestricted access to different service providers, allowing them to access transport and services provided by different business entities. NGNs support as well the provisioning of a wide range of services, including voice (e.g. telephone service), data (e.g. Web-based services), video (e.g. IP-TV), and combined services (e.g. video telephony).

**This paper was published in IEEE COMMUNICATIONS MAGAZINE, NETWORK & SERVICE MANAGEMENT SERIES, December 2011. This is an author copy for personal record only.**

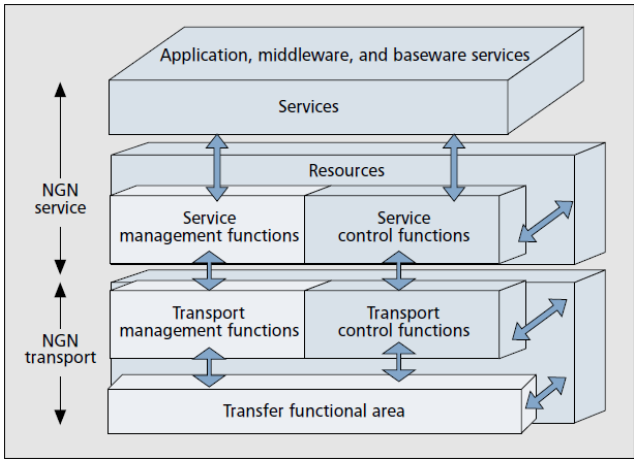


Figure 1: Generic NGN architecture

Much work has already been done on the use of the SOAP-based Web services for service provisioning in telecommunication networks in general, including NGNs [3]. The use of RESTful Web services is now being contemplated. The key reason is that RESTful Web services rely on Web technologies (e.g. HTTP, HTML) that are widely deployed and could be easily re-used. This can only speed up service provisioning in NGNs.

SOAP-based Web services provide a standard means for interoperating between software applications. RESTful Web services are designed following the Representational State Transfer (REST) design style. REST, a technology neutral design style, is defined as a network architectural style for distributed hypermedia systems. Hypermedia systems enable the storage and retrieval of information that may include different media such as text, audio, video, and (hyper)links.

RESTful Web services are being promoted as an alternative that may be more adequate than SOAP-based Web services in some cases. Service provisioning remains a big challenge and RESTful Web services may aid in tackling the challenge. This is a key motive to evaluate the state-of-the-art in RESTful–based service provisioning for NGN, and identify the research directions. It is the goal assigned to this paper.

Section II gives an overview of REST, with conferencing service as illustration. Section III discusses the state-of-the-art of RESTful-based service provisioning in NGNs. Section IV evaluates the overall suitability of RESTful Web services for the purpose and discusses research directions. We conclude in section V.

## II. REST OVERVIEW

In this section, we first introduce SOAP-based Web services seeing that they are very often contrasted with RESTful Web services. The principles of REST are then presented, followed by the description of a RESTful Web service for conferencing service used for illustration purpose. Readers interested in the comparison between SOAP-based Web services and RESTful Web services can consult [4].

### II.1 SOAP-BASED WEB SERVICES IN A NUTSHELL

The SOAP-based Web service architecture [5] defines three entities: service provider, service registry, and service requester (Figure 2). The service provider creates a SOAP-based Web service and publishes the service description in the service registry. The service requester finds the service by querying the service registry, retrieves the service description, and then uses the description to bind to the service implementation and start interacting with it. The service registry aims at the on-line discovery of services. However, it is rarely used today, because most requesters have prior knowledge of existing services, thanks to off-line business agreements.

The communications (operations) among the three Web service entities are based on XML and use the Simple Object Access Protocol (SOAP). SOAP messages are commonly exchanged over HTTP, even though other bindings are possible. The service descriptions are published using the Web Services Description Language (WSDL). WSDL provides information on how to use a Web service, including a description of the service operations and binding information. The most commonly used service registry for SOAP-based Web services is the Universal Description, Discovery and Integration (UDDI) registry. The UDDI specifications define a set of programming interfaces (APIs) for both publication and discovery.

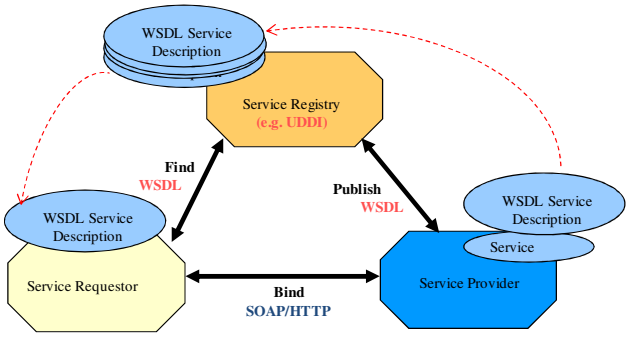


Figure 2: SOAP-based Web services architecture

The operations exposed by a SOAP-based Web service (e.g. createConference, addParticipant, in the case of a SOAP-based Web service for conferencing) are defined by the service provider and each provider can define its own operations (i.e. an operation's name, parameters and behavior). However, SOAP-based Web services can be standardized as a means to increase interoperability; as with Parlay-X multimedia conferencing Web service [6]. The list of exposed operations is then included in the service description.

**This paper was published in IEEE COMMUNICATIONS MAGAZINE, NETWORK & SERVICE MANAGEMENT SERIES, December 2011. This is an author copy for personal record only.**

### II.2 REST PRINCIPLES

REST adopts the client-server architecture of the web. REST does not restrict client-server communication to a particular protocol, but REST is most commonly used with HTTP because HTTP is the primary transfer protocol of the Web. RESTful Web services can be described using the Web Application Description Language (WADL) [7]. A WADL file describes the requests that can legitimately be addressed to a service, including the service's Uniform Resource Identifier (URI) and the data the service expects and serves.

REST relies on three main design principles [8]: addressability, uniform interface, and statelessness. For addressability, REST models the data-sets to operate on as resources, and identifies each resource via a URI. A resource is any form of information that can be named and that is important enough to be referenced (e.g. a document, a row in a database, a search result).

REST resources are accessed via a uniform and standard interface. A uniform interface offers a number of advantages among which are familiarity (i.e. the set of operations a RESTful Web service may expose are known) and interoperability. Statelessness means that each REST request is self-contained with all the information that the server needs to fulfill the request. No client-session data is stored on the server and the server never relies on information from previous requests to answer a new request. The following advantages are usually associated with statelessness: easy application development, good scalability, and easy load balancing.

REST is not an architecture, but a set of design criteria. Resource-Oriented Architecture (ROA) is a RESTful architecture that provides a commonsense set of rules and a step-by-step procedure for designing RESTful Web services following these design criteria. The fundamental mindset of ROA is the concept of resources. Each resource has a name (i.e. a URI) and a representation, and it may be linked to other resources via hyperlinks. A resource representation is what the client receives when it sends a request concerning a resource. The representation can be defined as any useful information about the current state of the resource. An example in the case of conferencing is the list of participants.

REST (and ROA) supports a wide range of representation formats, including plain text, HTML, XML and JavaScript Object Notation (JSON). ROA uses HTTP as the communication protocol. Therefore, the ROA uniform interface consists of HTTP operations, the most commonly used being GET, PUT, POST, and DELETE. We can design a RESTful Web service using ROA in the following steps. We first figure out the data set on which the service will operate, and split it into resources. After that for each resource we proceed as follows.

- First, we name the resource using a URI.
- Second, we identify the subset of the uniform interface that is exposed by the resource.
- Third, we design the representation(s) of the resource as received (in a request) from and sent (in a reply) to the client.
- Fourth, we consider the typical course of events by exploring and defining how the new service behaves and what happens during a successful execution.

For a detailed description of these steps, the reader can consult [8].

### II.3 RESTFUL WEB SERVICE EXAMPLE

The proposed illustrative service provides the same functionalities as the SOAP-based Web service described in Parlay-X Multimedia Conference specification [6]. Conferencing is one of the main services in NGNs.

The Parlay-X conferencing service is technology neutral and allows applications to create and manage a multimedia conference. The underlying model of the Web service is based on three entities: conference, participant and media. The conference is the uniquely-identified context, to which participants can be added and removed. The participant is any party that participates in the conference. The media represents the media stream to support a participant's communication (e.g. audio, video, chat) and the stream direction (i.e. in, out, bidirectional).

In this example, 'conference', 'participant' and 'media' are the data set on which to operate. For sake of simplicity, we focus on conference and participant. The data-set is then split into three resources: 'conference', 'list of participants', and 'participant'. The first resource represents a specific conference. The second lists the participants of the conference, and the last represents individual participants.

The 'conference' resource is named with the URI: http://www.confexample.com/{confId}/, confId being the unique identifier of the conference, the 'list of participants' with: http://www.confexample.com/{confId}/participants/, and the individual participant with URI: http://www.confexample.com/{confId}/participants/{participantURI}/, since every participant is identified by his/her URI.

The three resources can be read, created and deleted at runtime. The first column of Table 1 lists the resources, and the second lists the subset of the uniform interface that is exposed by each resource. The last column gives the representations accepted from the client and those served by the server for each operation.

Table I: Resource description and data representation

| Resource | Exposed subset of the uniform interface | | Data representation | |
|---|---|---|---|---|
| | Operation | HTTP action | Client->Server | Server->Client |
| Conference | Create: establish a conference | POST: http://confexample.com/ | <conference><br><description> discuss project </description><br><maxParticipants>10</maxParticipants><br></conference> | http://www.confexample/conf23@example.com |
| | Read: Get conference status | GET: http://confexample.com/{confId} | None | <status>Active</status> |
| | Delete: end a conference | DELETE: http://confexample.com/{confId} | None | None |
| List of participants/ Participant | Read: Get list of participants | GET: http://confexample.com/{confId}/participants | None | <participants><br><participant><br><uri>alice@ericsson.com</uri><br><status>Connected</status><br></participant><br>….<br></participants> |
| | Create: Add a participant | POST: http://confexample.com/{confId}/participants | <participant><br>alice@ericsson.com<br></participant> | <participant><br><uri>alice@ericsson.com</uri><br><link>http://confexample.com/{confId}/participants/alice@ericsson.com</link><br></participant> |
| | Read: Get a participant status | GET: http://confexample.com/{confId}/participants/{participantURI} | None | <status>Invited</status> |
| | Delete: remove a participant | DELETE: http://confexample.com/{confId}/participants/{participantURI} | None | None |

Figure 3 presents a sample sequence diagram that shows what should happen during a successful execution of the service. The client (i.e. Alice) sends a POST request to the service URI, to request the creation of a new conference. The server creates a new 'conference' resource and sends the resource URI to the client. When the conference is created and the necessary resources reserved, the server sends a 200 OK message. In step 4 of the figure, the client asks for the conference status, which she will get in the 200 OK response. In step 6, the client requests the addition of a new participant. She is first informed that the request is accepted, then she gets a 200 OK when the participant is actually added to the conference.

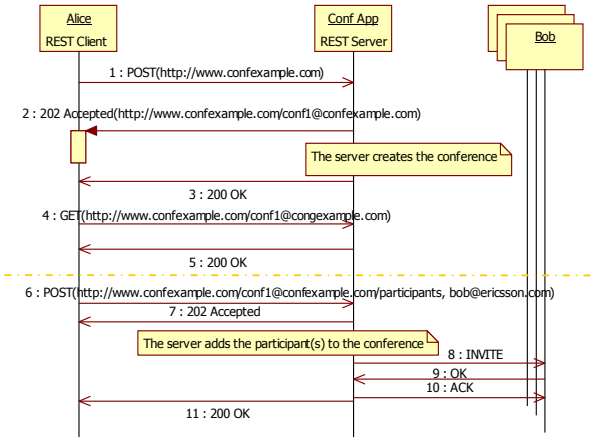


Figure 3: Sample sequence diagram

## III. THE STATE-OF-THE-ART

REST has been widely used outside of NGNs. Some examples are read-only Web applications (e.g. static websites and search engines), Amazon's Simple Storage Service (S3), twitter, and most of Yahoo!'s Web services. The use of REST for service provisioning in NGNs is rather recent and includes both standardization efforts and work done outside standards bodies.

### III.1 STANDARDIZATION EFFORTS

**This paper was published in IEEE COMMUNICATIONS MAGAZINE, NETWORK & SERVICE MANAGEMENT SERIES, December 2011. This is an author copy for personal record only.**

Several bodies are attempting to produce standard specifications for REST-based service provisioning in NGNs. We review here the Open Mobile Alliance (OMA) and the IETF efforts.

The OMA is working on a REST binding (ParlayREST) for Parlay-X Web services. Thus far, the OMA has focused on relatively simple non-session based services. The specifications include Short Messaging, Multi Media Messaging, Payment and Terminal Location Parlay-X Web Services. They have defined the resources and use HTTP as their message transfer protocol. As for resource representation formats, XML and JSON are used for all resources, but other formats may be used for some specific resources.

The ParlayREST specification for Short Messaging Service [9] is used in this paper for the purpose of illustration. It provides support to:

- Send text messages to a terminal and check their delivery status.
- Check, retrieve and delete the incoming messages.
- Create and delete subscriptions for notifications for inbound/outbound messages.

Table II summarizes some of the service resources, their URIs and the operations they accept.

Table II: A subset of ParlayREST SMS resources

| **Resources** | **URL**<br>Base URL:<br>http://{serverRoot}/{apiVersion}/smsmessaging | **HTTP action** |
|---|---|---|
| Outbound SMS message requests | /outbound/{senderAddress}/requests | GET: read pending outbound message requests<br>POST: create new outbound messages request |
| Outbound SMS message request and delivery status | /outbound/{senderAddress}/requests/{requestId} | GET: read a given sent message, along with its delivery status |
| Inbound SMS message subscriptions | /inbound/subscriptions | GET: read all active subscriptions<br>POST: create new message subscription |
| Individual inbound SMS message subscription | /inbound/subscriptions/{subscriptionId} | GET: read individual subscription<br>DELETE: remove subscription and stop corresponding notifications |

Figure 4 presents a sample scenario for sending and receiving a message. In the first part of the figure (i.e. SMS sending), the application sends an 'SMS sending' request to the URI of the 'outbound SMS message requests resource', using the POST operation. The SMS to be sent is included in the request body. The server creates a new resource and sends its URI to the application (including the requestId).

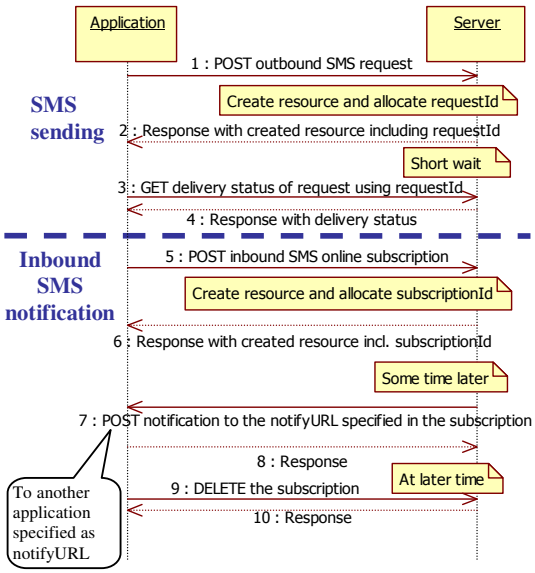


Figure 4: Sample scenario for SMS handling

In step 3, the application checks the delivery status using a GET request sent to the URI of the newly created resource. In the second part of the figure, the receiving application subscribes to the notifications for inbound messages by sending a POST request to the URI of the 'Inbound SMS message subscription'. The server creates a new 'Individual inbound SMS message subscription resource' and transmits its URI to the application. The application may use this URI later to delete or get information about the subscription. When the server receives a SMS destined to the application, it notifies the application whose URI is specified in the subscription request, using a POST request.

The IETF is working on REST-based approach for the Centralized Conferencing Manipulation Protocol (CCMP). CCMP is a stateless, XML-based, client-server protocol for conference control [10]. The CCMP specification includes a general (i.e. non-REST specific) discussion of the protocol, and a discussion of a RESTful approach to the protocol.

The CCMP allows users to create, manipulate (e.g. add/remove participants, add/remove media streams) and delete conference objects. A conference object is a logical representation of a conference instance, representing the current state and capabilities of a conference. The RESTful approach for the CCMP uses HTTP as the transfer protocol for CCMP messages, models the conference objects as resources identified by URIs, and uses XML for data representation.

### III.2 WORK DONE OUTSIDE THE STANDARDS BODIES

Examples of work done outside the standardization bodies are presented in [11] and [12].

[11] discusses three approaches for exposing telecom capabilities (e.g. SMS, presence) with REST. The first approach uses an existing service delivery platform (SDP) as a middle-layer over which the RESTful API is provided (Figure 5). The SDP may belong to the NGN network operator or to a third party. The API can be built as an application inside the SDP that provides the necessary mappings to the actual network elements that provide the capabilities to expose. This option has the advantage of lowering the integration effort of the RESTful API to the network capability. However, the SDP may become an unnecessarily heavy middleware if it is only used to provide the RESTful API.

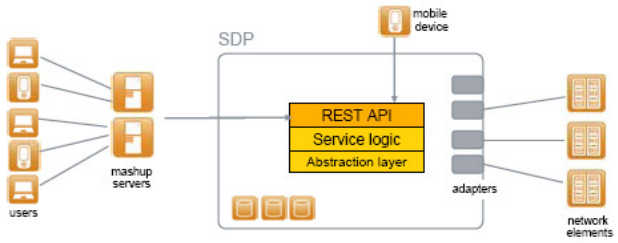


Figure 5: Integration via an SDP

The second approach is to have the RESTful API deployed on a separate system that is integrated to the appropriate network element as-needed. This approach bypasses the SDP overhead, but it requires substantial work on integration. The mapping layer is integrated with the RESTful API, which is directly integrated to a specific network element.

In the third approach, the RESTful interface and the service logic are run as a standalone system, with no integration to the operator network. One example is to provide a RESTful SMS service by integrating to a third-party SMS service provider. This approach allows for the service to be run by any party, but it has the disadvantage of not allowing access to the resources and information residing on the operator network/system (e.g. subscribers' information).

[12] proposes a generic REST approach to expose the session-based capabilities of the 3G IP Multimedia Subsystem (IMS). This approach models the sessions (e.g. multimedia sessions) as resources, and each resource represents a session associated with a specific service. A conferencing session initiated by Alice for instance is named with www.example.com/aliceURI/Conferencing/sessionID. Each resource considers the session's state, list of participants, media description and links to the session's media components.

[12] also proposes an architecture for IMS and Web 2.0 convergence, and discusses two guidelines for exploiting Web 2.0 services and technologies to enrich telecom operator's services. Web 2.0 is a concept that promotes interactive information sharing and collaboration over the Web, as well as Web application consumption by software programs. The first guideline is to incorporate Web 2.0 content (e.g. user-generated video) and events (e.g. contextual information associated with social networks) into telecom services. This can enhance user experience and increase service customization.

The second guideline is IMS services' delivery via web pages. Web 2.0 technologies are used to build on-line applications. The applications use directly the services offered by the operator. An example is a virtual IMS terminal that runs in an end-user's browser. The major benefits here are service ubiquity, the reuse of the major advances achieved by Web 2.0 applications in the field of user interfaces, and a significant simplification of the service development process and deployment.

## IV. SUITABILITY AND RESEARCH DIRECTIONS

We use NGN service provisioning requirements as identified by ITU-T to evaluate the overall suitability of REST. The requirements are presented first. The overall suitability is then discussed in light of the requirements. Research directions are discussed last.

### IV.1 NGN REQUIREMENTS

Some NGN requirements impact all layers, including the service layer, while others impact only specific layers [1]. The main layer independent requirement deals with QoS and security. A mechanism for end-to-end QoS should be defined and security mechanisms should be provided to protect the exchange and the use of sensitive information, including authentication, authorization and encryption. The layer specific requirements are discussed below.

One fundamental requirement of NGNs is the support of a wide range of services, and more specifically, making the creation, deployment and management of all kinds of known and unknown services possible and easy. This aspect includes enabling service providers (or operators) to find and reuse services offered by other providers (operators) to build new services. This requires support for service description, and service publication and discovery.

Still another requirement is to allow for applications to be based on service building blocks and functional entities. This enables the reuse of existing services and allows the building of composed applications.

NGNs also require the support of a wide range of terminals such as telephones, cell phones, PDAs and laptops to access the NGN services, which implies that client applications must be simple and adaptive.

The last requirement is to provide unified characteristics for the same service as perceived by the user. This can be provided via the provisioning of standardized and open interfaces for the provided services.

### IV.2 REST AND NGN REQUIREMENTS

Regarding QoS and security, current RESTful Web services are mostly based on HTTP, and therefore reuse the HTTP best-effort QoS mechanism. In terms of security, the services rely on Web/HTTP security mechanisms, such as Transport Layer Security (TLS), to secure access to RESTful Web services. HTTP defines two authentication and authorization schemes (i.e. simple challenge/response and digest authentication), but there is a standard means to integrate other authentication schemes into HTTP, such as the schemes defined for SOAP-based Web services [8].

Regarding the layer-dependent requirements, RESTful Web services enable a wide range of end-user services because they enable easier development and deployment of these services. The development paradigm is based on the natural way the Web works.

However, the development of complex session-based services may not be so obvious, due in part to the statelessness of REST. This is especially the case for services for which the server needs to maintain some states. One example is floor control-based conferencing where a floor is granted to a requester only if nobody already holds it. It is important to mention that REST statelessness does not mean that the service cannot have a state. The server can store and manage the state of the resources it exposes. For instance, upon reception of a conference creation request, the server creates the conference and maintains its state (e.g. the current list of participants and the participant(s) that hold(s) the floor). The client can ask the server about the new state of the conference at any time but the request is independent of any previous request. The point here is to draw attention to the fact that designing session-based services with REST principles is not straightforward. The design does require much more detailed thought and consideration. However, the design of these services does not necessarily require extensions to REST.

For service publication and discovery, RESTful Web services can be described using WADL, but no appropriate service publication and discovery platform has been defined thus far. RESTful Web services also meet the requirement for building blocks. Elementary building blocks can be composed into more complex Web services through mashups [13].

Mashup is a Web concept where data, presentation or functionality from two or more sources are combined in order to create new services. One example is to get a user location using a location service and display it using a Google map. However, the lack of an automatic service discovery mechanism limits the number of the composed services (we can only compose the services we already know about).

A RESTful Web service can be accessed by a wide range of end-user devices, including laptops, cellphones and PDAs. However, service adaptation to different devices without any changes is not fully achievable. Nevertheless, depending on the particular service, the adaptation level may be controlled by limiting the client complexity (e.g. a simple service may be executed over a Web browser).

In regards to open interfaces, there are ongoing efforts to provide standardized RESTful APIs for telecommunication services (e.g. ParlayREST APIs). However, the fact that the RESTful services may use different data models and resource representation formats may result in interoperability issues. Therefore, the standard should also specify the data models and formats supported by each service.

In summary, RESTful Web services show a strong potential for service provisioning in NGNs. They meet most of the NGN requirements related to service provisioning. However, research remains to be done in certain areas in order to realize its full potential. It is important to stress that extensions to REST may not always be required.

### IV. 3 RESEARCH DIRECTIONS

Research directions related to RESTful Web services include open issues related to REST in general but pertinent to service provisioning in NGNs, and open issues specific to service provisioning in NGNs. A general open issue is service publication and discovery. [14] talks about REST registries where the RESTful Web services are published but it does not give any details about how the registry is designed or how the services are published and discovered. Before a client can start interacting with a RESTful Web service, it needs to know the starting URI of the service and the representation format accepted. The same applies for each of the service resources. Currently, a client can discover such information offline, such as from the service provider web site or by using a Web search engine.

Some potential approaches for starting URI publication and discovery are the use of an enhanced Domain Name System (DNS) or the design of a RESTful Web services registry, along with the publication and discovery interface. Another research direction related to the design of a RESTful Web services registry is to adapt the SOAP-based solutions (e.g. UDDI and WSDL) to the specificities of RESTful Web services.

A key open issue specific to service provisioning is resource-definition for complex session-based services (e.g. conferencing). Indeed, as discussed earlier, the design of such services is not obvious and resource definition is the corner stone. Exposing session-based services with a stateless architectural style requires special attention. Furthermore, besides resource definition, there are other challenges related to the provisioning of these services. An example is the design of enhanced features such as floor control.

Parlay-X, for example, provides a specification for a conferencing SOAP-based Web service. However, to the best of our knowledge, there is no comprehensive RESTful session-based Web service, including conferencing. OMA ParlayREST specifications do not cover session-based services, and the CCMP work is still preliminary.

The conferencing service described in this article is a good starting point for a session-based service. It can be extended to

provide more functionalities such as media manipulation (e.g. add/remove/update a media stream) and floor control, by defining new resources 'media' and 'floor' resources. Potential approaches for notification support include using HTTP 1.1 persistent connections and long-polling, which provide the HTTP server the possibility to push data to clients.

Another open issue is the design of middleware that expose NGN capabilities via RESTful interfaces. This should respond to the requirement of having a common and open RESTful interface to access these capabilities. It will also ease the development of new services based on these capabilities. A Parlay- X gateway, for instance, is a standard way to expose the capabilities via a SOAP-based Web services interface, but to the best of our knowledge, no comprehensive RESTful middleware was proposed in the literature.

A potential approach for designing a such middleware is as follows. First, identify the different mapping alternatives between the RESTful API and the capabilities' interfaces. Second, define a general mapping pattern that can be applied to most (or all) of the capabilities, if any. Third, optimize the middleware performance.

The approaches presented in [11] (and discussed in section III.2) can be used as starting point. The most promising approach can be reused and eventually enhanced to provide a suitable middleware. The middleware should mainly include a mapping functionality, provide an easy to use interface and allow for easy support of additional network capabilities and nodes.

## V. CONCLUSIONS

REST has been widely used outside NGNs. However, several standards bodies are attempting to produce standard specifications for REST-based service provisioning in NGNs (e.g. OMA and IETF). Some work has also been done in the area outside standards bodies.

RESTful Web services meet many NGN service provisioning requirements. They enable easy development and deployment of a wide range of services, support a wide range of terminals (e.g. laptops, cell phones), and allow for service composition through mashups.

However, some issues are still open, such as RESTful Web services publication and discovery, resource definition for session-based services and the provisioning of an adequate middleware. RESTful Web services do indeed show a great potential for service provisioning in NGNs. Nevertheless, the open issues need to be solved before their full potential can be realized.

## REFERENCES

[1] "Next Generation Networks – Frameworks and functional architecture models", ITU-T recommendation Y.2001, December 2004

[2] K. Knightson et al, NGN Architecture: Generic Principles, Functional Architecture, and Implementation, IEEE Communications Magazine, October 2005

[3] D. Griffin and D. Pesch, "A Survey on Web Services in Telecommunications", IEEE Communications Magazine, July 2007, Pages: 28-35

[4] C. Pautasso, O. Zimmermann, and F. Leymann, "RESTful Web Services vs. "Big"Web Services: Making the Right Architectural Decision", In Proceedings of the 17th International World Wide Web Conference, pages 805–814, Beijing, China, April 2008, ACM Press.

[5] E. Newcomer, "Understanding Web Services: XML, WSDL, SOAP, and UDDI", Addison-Wesley, ISBN 0-201-75081-3, May 2002

[6] ETSI TS 129 199-12, Parlay X Web services; Part 12: Multimedia conference, 3GPP TS 29.199-12 version 9.0.0 Release 9, January 2010

[7] W3C Member Submission, "Web Application Description Language", 31 August 2009

[8] L. Richardson and S. Ruby, "RESTful Web Services", O' Reilly & Associates, ISBN 10: 0-596-52926-0, May 2007

[9] Open Mobile Alliance, "RESTful bindings for Parlay X Web Services –Short Messaging", Candidate Version 1.0 – 27 Apr 2010

[10] M. Barnes et al, "Centralized Conferencing Manipulation Protocol", IETF draft, draft-ietf-xcon-ccmp-07, April 26, 2010

[11] S. Mäkeläinen and T. Alakoski, "Fixed-mobile hybrid mashups: experiences and lessons on applying the REST software architecture principles to exposing mobile operator services", Proceedings of ICIN 2008 - the 11th International Conference on Services, Enablers and Architectures Supporting Business Models for a New Open World, NeuStar Secretariat Services, Bordeaux, France, October 20 - 23, 2008

[12] D. Lozano, L.A. Galindo and L. García, "WIMS 2.0: Converging IMS and Web 2.0. Designing REST APIs for the exposure of session-based IMS capabilities", The Second International Conference on Next Generation Mobile Applications, Services, and Technologies (NGMAST), 2008

[13] C. Pautasso, "Composing RESTful services with JOpera", Proc. of the 8th International Conference on Software Composition, volume 5634, July 2009, pp. 142–159

[14] G. Jie1 et all, "Applying Recommender System based Mashup to Web-Telecom Hybrid Service Creation", IEEE Global Communications Conference (GLOBECOM) 2009, p: 1-5

## BIOGRAPHIES

**Fatna Belqasmi** holds a Ph.D. and an M.Sc. degree in electrical and computer engineering from Concordia University, Canada. She is a research associate at Concordia University, Canada. In the past, she worked as a researcher at Ericsson Canada. She was part of the IST Ambient Network project (a research project sponsored by the European Commission within the Sixth Framework Programme -FP6-). She worked as an R&D engineer for Maroc Telecom in Morocco. Her research interests include next generation networks, service engineering, distributed systems, and networking technologies for emerging economies.

**Chunyan Fu** currently works at Tekelec. She worked at Ericssion from 2008 to 2010 as a researcher and a services engineer. From 1997 to 2001, she worked at China Mobile as a system integration engineer. She received her Bachelor degree in Computer Engineering from Nanjing University of Posts and Telecommunications. She received her M.Sc. and Ph.D. degrees in Electrical and Computer Engineering from Concordia University, Canada in 2004 and 2008 respectively. During her M.Sc. and Ph.D. studies, she worked

as an intern at Ericsson Canada. Her research interests include signaling, ad hoc networks, IMS and services in next-generation networks.

**Roch Glitho** [SM] holds a Ph.D. (Tekn. Dr.) in tele-informatics (Royal Institute of Technology, Stockholm, Sweden), and M.Sc. degrees in business economics (University of Grenoble, France), pure mathematics (University Geneva, Switzerland), and computer science (University of Geneva). He is an associate professor of networking and telecommunications at the Concordia Institute of Information Systems Engineering (CIISE), Concordia University, Montreal, Canada where he holds a Canada Research Chair in End-User Service Engineering for Communication Networks. In the past he has worked in industry for almost a quarter of a century and has held several senior technical positions at LM Ericsson in Sweden and Canada (e.g. expert, principal engineer, senior specialist). His industrial experience includes research, international standards setting (e.g. contributions to ITU-T, ETSI, TMF, ANSI, TIA, and 3GPP), product management, project management, systems engineering and software/firmware design. He is a member of several editorial boards including IEEE Network and IEEE Communications Surveys and Tutorials. In the past he has served as IEEE Communications Society distinguished lecturer, Editor-In-Chief of IEEE Communications Magazine and Editor-In-Chief of IEEE Communications Surveys & Tutorials. His research areas include architectures for end-users services, distributed systems, non conventional networking, and networking technologies for emerging economies. In these areas, he has authored more than 100 peer-reviewed papers, more than 30 of which have been published in refereed journals. He also holds 24 patents in the aforementioned areas and has several pending applications.