Network Working Group	C. Boulton	
Internet-Draft	NS-Technologies	
Intended status: Standards Track	L. Miniero	
Expires: September 5, 2009	University of Napoli	
	March 04, 2009	

TOC

Media Resource Brokering
draft-boulton-mediactrl-mrb-04

Status of this Memo

Copyright Notice

Abstract

The MediaCtrl work group in the IETF is currently proposing an architecture for controlling media services. The Session Initiation Protocol (SIP) will be used as the signalling protocol which provides many inherent capabilities for message routing. In addition to such signalling properties, a need exists for intelligent, application level

media service selection based on non-static signalling properties. This is especially true when considered in conjunction with deployment architectures that include 1:M and M:M combinations of Application Servers and Media Servers.

Table of Contents

---

1. Introduction

TOC

The topic of Media Resources has been in discussion for a number of years with varying proprietary solutions being used today. It is clear that, as we move towards a consistent architecture and protocol for Media Server Control, a standard mechanism is required for accurate media resource location.
As IP based multimedia infrastructures mature, the complexity and demands from deployments increase. Such complexity will result in a wide variety of capabilities from a range of vendors that should all be interoperable using the architecture and protocols produced by the MediaCtrl work group. It should be possible for a controlling entity to be assisted in Media Server selection so that the most appropriate resource is selected for a particular operation. The importance increases when you introduce a flexible level of deployment scenarios, as specified in the MediaCtrl Requirements (Dolly, M. and R. Even, “Media Server Control Protocol Requirements,” February 2008.)

[I-D.ietf-mediactrl-requirements] and MediaCtrl Architecture (Melanchuk, T., “An Architectural Framework for Media Server Control,” November 2008.) [I-D.ietf-mediactrl-architecture] documents. These documents make statements like "it should be possible to have a many-to-many relationship between Application Servers and Media Servers that use this protocol". This leads to the following deployment architectures being possible when considering media resources. The simplest deployment view is illustrated in Figure 1 (Basic Architecture).

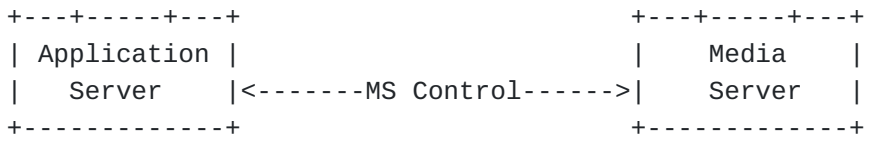

Figure 1: Basic Architecture

This simply involves a single Application Server and Media Server. Expanding on this view, it is also possible for an Application Server to be controlling multiple (greater that 1) Media Servers. This deployment view is illustrated in Figure 2 (Basic Architecture). Typically, such architectures are associated with application logic that requires high demand media services. It is more than possible that each media server possesses a different media capability set. Media servers may offer different media services as specified in the Mediactrl architecture document. A Media server may have similar media functionality but may have different capacity or media codec support.

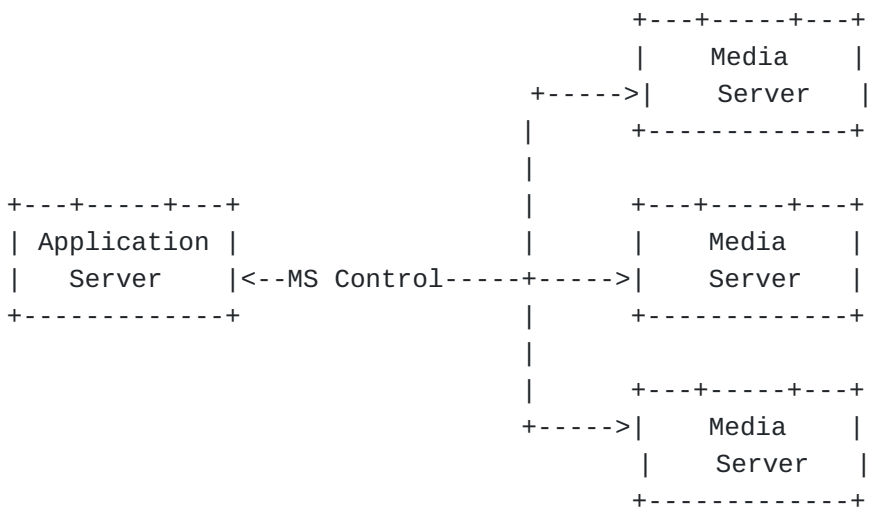

Figure 2: Basic Architecture

Figure 3 (Basic Architecture) conveys the opposite view to that in Figure 2 (Basic Architecture). In this model there are a number of (greater than 1) application servers controlling a single media server. Typically, such architectures are associated with application logic that requires low demand media services.

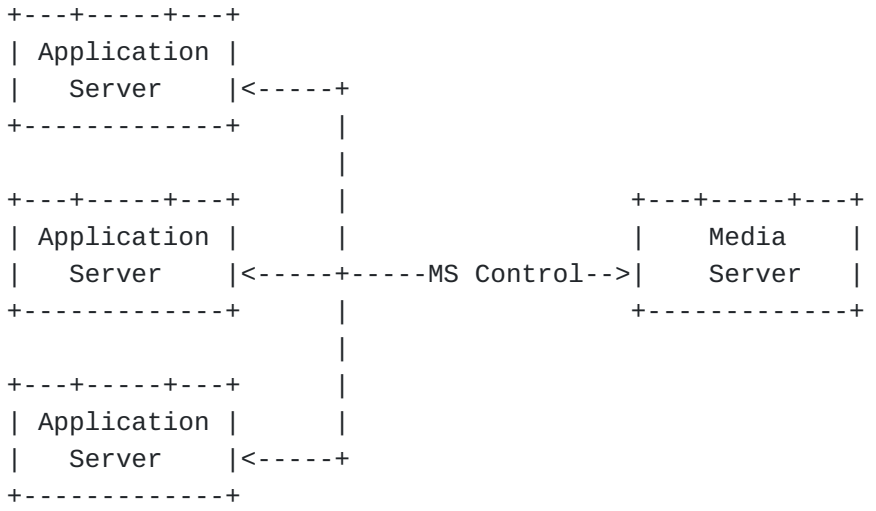

Figure 3: Basic Architecture

The final deployment view is the most complex. In this model (M:M) there exists any number of Application Servers and any number of Media Servers. It is again possible in this model that media servers might not be homogenous and have different capability sets.

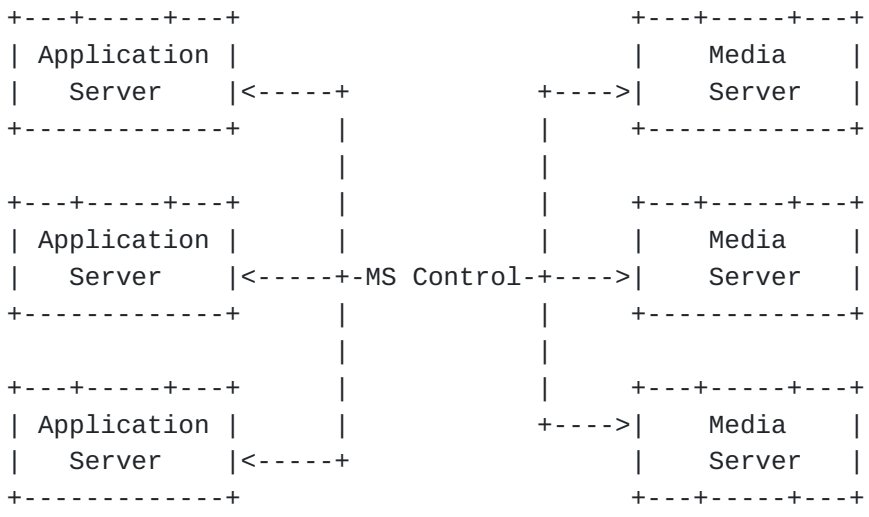

Figure 4: Basic Architecture

This document will take a look at the specific problem areas related to such deployment architectures. It is recognised that the solutions proposed in this document should be equally adaptable to all of the previously described deployment models. It is also recognised that the solution is far more relevant to some of the previously discussed deployment models and can almost be viewed as redundant on others.

2. Conventions and Terminology

TOC

In this document, BCP 14/RFC 2119 (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.) [RFC2119] defines the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL". In addition, BCP 15 indicates requirement levels for compliant implementations.

This document inherits terminology proposed in the MediaCtrl Architecture (Melanchuk, T., "An Architectural Framework for Media Server Control," November 2008.) [I-D.ietf-mediactrl-architecture] and Media Control Channel Framework (Boulton, C., Melanchuk, T., and S. McGlashan, "Media Control Channel Framework," October 2009.) [I-D.ietf-mediactrl-sip-control-framework] documents. In addition, the following terms are defined for use in this document and for use in the context of the MediaCtrl Work group in the IETF:

> **Media Resource Broker (MRB):** A logical entity that is responsible for both collection of appropriate published Media Server (MS) information and supplying of appropriate MS information to consuming entities.
>
> **Query MRB:** An instantiation of an MRB (See previous definition) that provides an interface for an Application Server to retrieve the location of an appropriate Media Server. The result returned to the Application Server can be influenced by information contained in the query request.
>
> **In-line MRB:** An instantiation of an MRB (See definition) that directly receives requests on the signalling path. The decision making process is totally delegated to the MRB.

---

3. Problem Discussion

TOC

It is clear from Section 1 (Introduction) that the MediaCtrl group will be producing a solution that must service a wide variety of deployment architectures. These range from the simplest 1:1 relationship between Media Servers and Application Servers to potentially linearly scaling 1:M, M:1 and M:M deployments.
This still does not seem like a major issue for the proposed solution until you add a number of additional factors into the equation that increase complexity. As Media Servers evolve it must be taken into consideration that, where many can exist in a deployment, they may not have been produced by the same vendor and may not have the same capability set. It should be possible for an Application Server that exists in a deployment to select a Media Service based on a common, appropriate capability set. In conjunction with capabilities, it is also important to take available resources into consideration. The ability to select an appropriate Media Service function is an extremely useful feature but becomes even more powerful when considered in conjunction with available resources for servicing a request.
In conclusion, the intention is to create a tool set that allows MediaCtrl deployments to effectively utilize the available media resources. It should be noted that in the simplest deployments where

only a single media server exists, an MRB function is probably not required. Only a single capability set exists and resource unavailability can be handled using the appropriate underlying signalling e.g. SIP response. This document does not prohibit such uses of an MRB, it simply provides the tools for various entities to interact where appropriate. It is also worth noting that the tools provided in this document aim to provide a 'best effort' view of media resources at the time of request for initial Media Server routing decisions. Any dramatic change in media capabilities after a request has taken place should be handled by the underlying protocol.

---

4. Deployment Scenario Options

TOC

On researching Media Resource Brokering it became clear that a couple of high level models exist. The general principles of "in-line" and "query" MRB concepts are discussed in the rest of this section.

4.1. Query MRB

TOC

The "Query" model for MRB interactions provides the ability for a client of media services (for example an Application Server) to "ask" an MRB for an appropriate Media Server, as illustrated in Figure 5 (Query MRB).

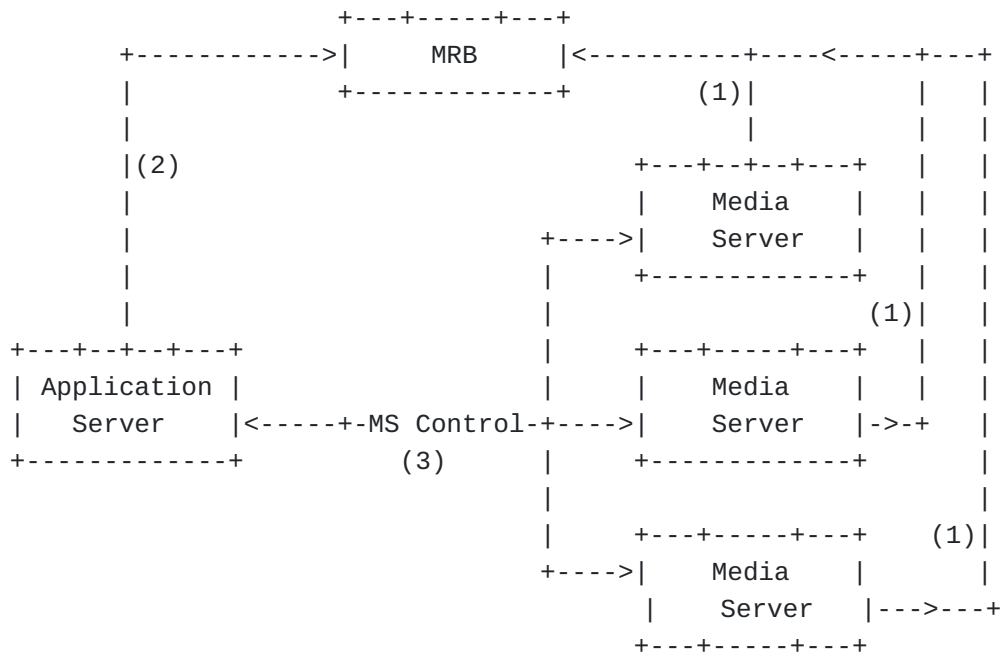

Figure 5: Query MRB

In this deployment, the Media Servers use the "Media Server Resource Publishing Interface", as discussed in Section 5.1 (Media Server Resource Publishing Interface), to convey capability sets as well as resource information. This is depicted by (1) in Figure 5 (Query MRB). It is then the MRB's responsibility to accumulate all appropriate information relating to media services in the logical deployment cluster. The Application Server (or other media services client) is then able to query the MRB for an appropriate resource (as identified by (2) in Figure 5 (Query MRB)). Such a query would carry specific information related to the Media Service required and enable the MRB to provide an increased accuracy in its response. This particular interface is discussed in "Media Resource Consumer Interface" in Section 5.2 (Media Service Resource Consumer Interface). The Application Server is then able to direct control commands (for example create conference) and Media Dialogs to the appropriate Media Server, as shown by (3) in Figure 5 (Query MRB).

4.1.1. Hybrid Query MRB

TOC

As mentioned previously, it is the intention that a tool kit is provided for MRB functionality within a MediaCtrl architecture. It is

expected that in specific deployment scenarios the role of the MRB might be co-hosted as a hybrid logical entity with an Application Server, as shown in Figure 6 (Hybrid Query MRB - AS Hosted).

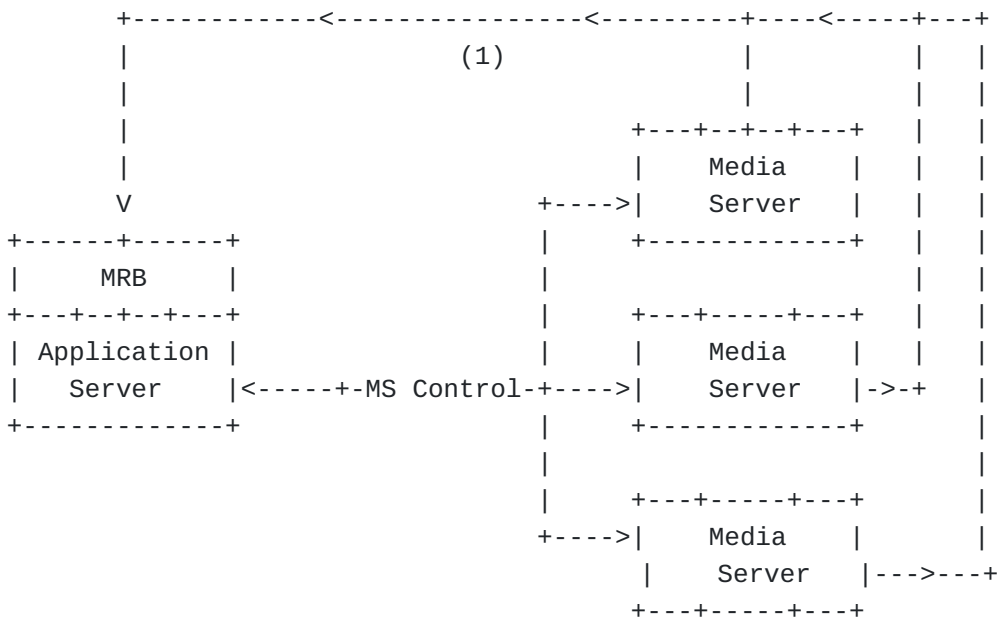

Figure 6: Hybrid Query MRB - AS Hosted

This diagram is identical to that in Figure 5 (Query MRB) with the exception that the MRB is now hosted on the Application Server. The "Media Server Publishing Interface" is still being used to accumulate resource information at the MRB but as it is co-hosted on the Application Server, the "Media Server Consumer Interface" has collapsed. It might still exist within the Application Server/MRB interaction but this is an implementation issue. This type of deployment suits a single Application Server environment but it should be noted that a "Media Server Consumer Interface" could then be offered from the hybrid if required.
In a similar manner, the Media Server could also act as a hybrid for the deployment cluster, as illustrated in Figure 7 (Hybrid Query MRB - MS Hosted).

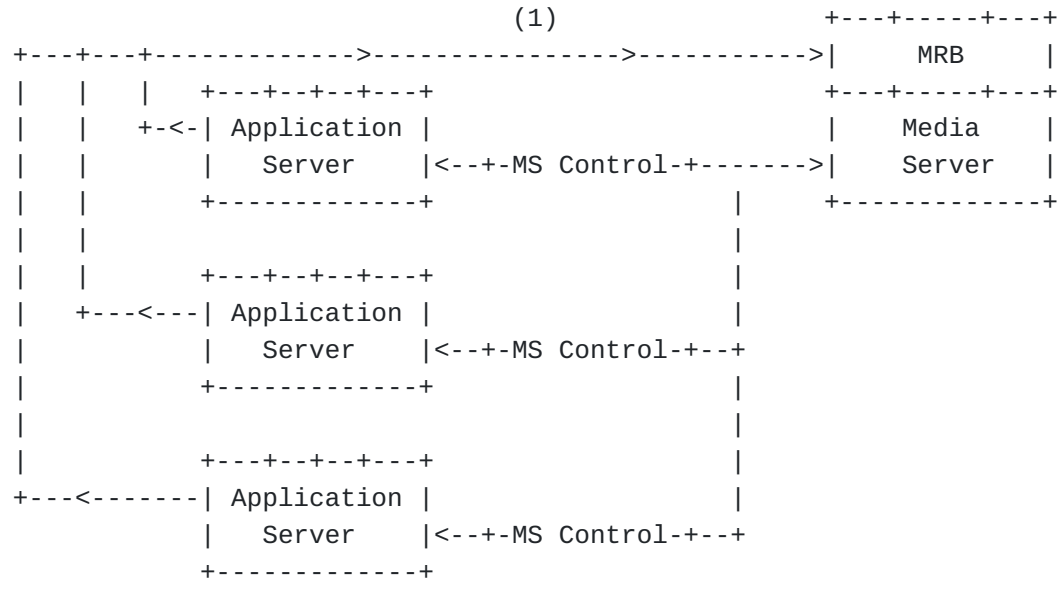

Figure 7: Hybrid Query MRB - MS Hosted

This time the MRB has collapsed and is co-hosted by the Media Server. The "Media Server Consumer Interface" is still available to the Application Servers (1) to query Media Server resources. This time the "Media Server Publishing Interface" has collapsed onto the Media Server. It might still exist within the Media Server/MRB interaction but this is an implementation issue. This type of deployment suits a single Media Server environment but it should be noted that a "Media Server Publishing Interface" could then be offered from the hybrid if required.

> [MRB-01] Is this second hybrid topology really useful? The only use case for it seems to be for AS[i] to send a request, receive a response, and decide whether or not to place a CFW request at all (since only one MS is available). Such a use case makes sense, but it may be a bit overkill to make use of the MRB for that. Or should we envisage the possibility of having for instance the AS[i] construct a "lighter" CFW request if the MS is too loaded, and a "normal" CFW request otherwise?

TOC

4.2. In-Line MRB

The "In-line" MRB is architecturally different from the "Query" model that was discussed in the previous section. The Concept of a "Media Server Consumer Interface" disappears. The client of the MRB simply uses the signalling to offload the decision making process - this applies to both media server Control and Media Dialogs. This type of deployment is illustrated in Figure 8 (In-line MRB).

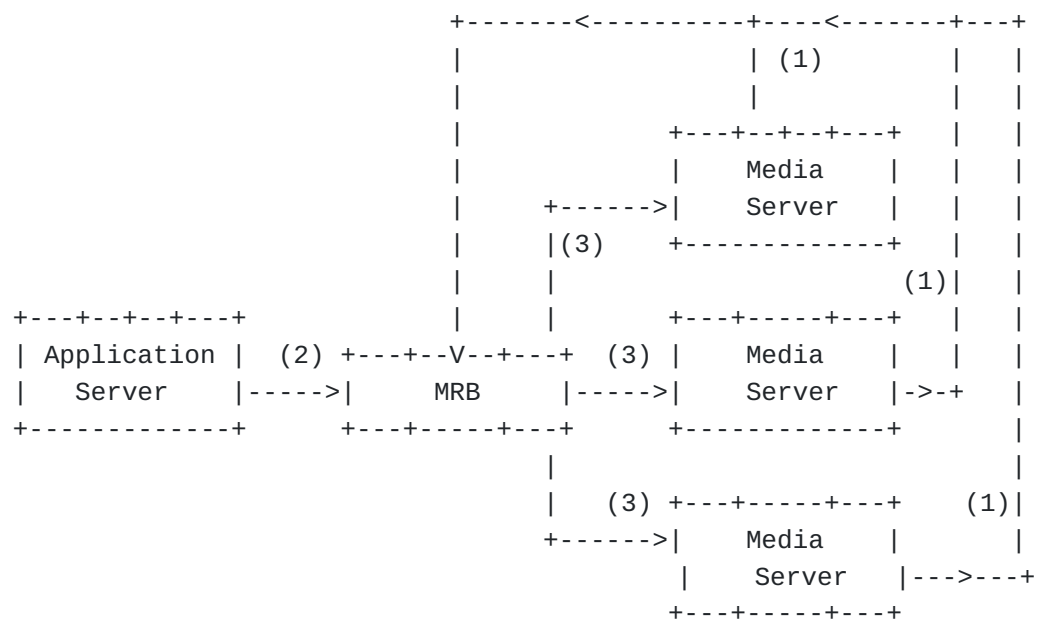

Figure 8: In-line MRB

The Media Servers still use the 'Media Server Publishing Interface' to convey capabilities and resources to the MRB - as illustrated by (1). The media server Control and Media dialogs are blindly sent to the MRB (2) which then selects an appropriate Media Server (3). The result of such an architecture is that the decision is left entirely to the MRB and the Application Server has no input into the selection process. This is the opposite to the "Query" model which provided information that would help influence the Media Server decision making process on the application server. As a by-product of this decision shift, a lot more emphasis is placed on the intelligence of the MRB to interpret the required capabilities of the request. It will actually have to inspect both the SIP signalling and the media server control protocol PDUs for the purpose of Media Server selection. This includes, for example, looking for explicit capabilities in the signalling and session details

such as media types, codecs and bandwidth requirements. Ultimately the decision making and policy enforcement is removed from the Application Server and shifted to the MRB logical entity.

> [MRB-02] How much intelligence should the MRB have in this case? In fact, it's not just a matter of capabilities, but of package-specific decisions as well. For instance, let's say there's a conference on MS[3]. If AS[2] has created the conference and attaching new participants to it, it definitely wouldn't want the MRB to arbitrarily join its users on different MS (with having its requests failing anyway, since a non-existing conference would be invoked). Should we envisage, as a parameter, something like a "context" or an "application session"? Considering no AS would be able to decide to invoke a specific MS, such a parameter would be the only (or at least a possibly working) way of letting an AS tell the MRB "this request is part of this specific application logic context, and so using a different MS would make it all collapse".

---

5. Interface Definition

TOC

As discussed in previous sections in this document, the intention is to provide a toolkit for a variety of deployment architectures where media resource brokering can take place. As a result, two main interfaces are required to support the differing requirements. The two interfaces are described in the remainder of this section and have been named the 'Media Server Resource Publishing' and Media Server Resource Consumer' interfaces. These two interfaces have extremely differing responsibilities and usages which is reflected in the choice of solutions.
It is beyond the scope of this document to define exactly how to construct an MRB. This includes interpreting the data for the Media Service Consumer interface supplied by the Media Server Publishing interface. It is, however, important that the two interfaces are complimentary so that development of appropriate MRB functionality is supported.

5.1. Media Server Resource Publishing Interface

TOC

The Media Server Resource Publishing interface is responsible for providing an MRB with appropriate Media Server resource information. It is generally accepted that this interface provides both general and specific details related to Media Server resources. This information needs to be conveyed using an industry standard mechanism to provide

increased levels of adoption and interoperability. A Control Package for the Media Control Channel Framework will be specified to fulfill this interface requirement. It provides the perfect establishment and monitoring mechanism to enable a Media Server to report appropriate statistics to an MRB.
[EDITORS NOTE: The use of the Media Control Channel Framework is still up for debate. This should be revisited and discussed appropriately. It is fair to say that Media Servers will already support the base Media Control Channel Framework and so adding this extra auditing facility provides nice synergy and reuse.]
EDITORS NOTE: Need to map resources to a control package and define appropriately. The following information has been taken from feedback from the community. Please comment on existing entries and any other that you feel should be added to the list. Note that some of the publishing topics would naturally be included in the 'AS Request to MRB' section that follows. At this stage it is only included in one place for further discussion:

> [MRB-03] Should we talk of a new Control Package and/or of additional requirements to the auditing mechanism for each package? Some of the addressable resources are generically MS-related (e.g. how many G711 sessions you have available, since it affects the SIP negotiations with new UACs), but some look like information that only packages would be able to provide.

*Active RTP sessions (including codec information). For example, 10 G711 RTP sessions, 3 H.264 sessions.

* -[MRB-04] This may not be required, since the purpose of the MRB is to check for available resources rather than occupied resources. Nevertheless, such details might be useful for complementary functionality as debugging and monitoring inside the MRB. What are your feelings about it?

*Active Mixers. For example F4: (2 G711, 3 G729), (second mixer and the codecs), (third mixer), ...).

*Non Active sessions - so sessions available on this MS (based on codecs supported). For example, 80 G711 RTP session,120 G729 sessions,30 H.264 sessions.

* -[MRB-05] Should this be an AND or an OR? (e.g. an AS takes all the 80 G711 RTP sessions, are the G729 sessions still available as well or were the specified resources shared?)

*MS Uptime.

*Codecs/media supported (could just be bundled with above 'Non Active Sessions'.

*In addition to the generic media processing related information, there are definitely cases where the AS will want to specify application-level criteria, which will be application-specific, and difficult to enumerate in advance. So I'm thinking we need a way to express arbitrary application specific criteria in addition to the generic media processing criteria. For example, the AS may need an MS which is capable of prompting and performing speech recognition in Swahili. Or, an MS which has the capability to invoke some application-specific functionality.

*File formats supported for announcement. E.g.: MP3, WAW etc... May be this information is enough to determine announcement format supported i.e. audio or video.

*Maximum duration for an announcement. Media servers can have restrictions on memory to play the announcements for very long durations.

*Variable announcements. Where the substitution variable can be time, date, cost etc.

*DTMF detection and generation support.

*Types of mixing (conference supported) audio, video.

*Supported tone types in the Media Server. Different countries may have different characteristics for the same tone. So the tone characteristics can be configured in the media server or can be downloaded. Capability to play the tone in both directions may be required for conferencing applications. E.g. playing a tone when a new participant joins in the conference. The tone needs to be played towards the existing participants and also towards the new participant.

* -[MRB-06] All these features are something that probably fit better in auditing, rather than in here (and some are actually already there). What do you think?

*Audio RTSP streaming. Audio conferencing. Audio record. Audio transcoding.

*ASR/TTS usage. ASR grammar complexity. Language complexity.

*Speaker verification/recognition.

*Music recognition.

*Audio transformation (mask voice, raise tone, add echo, effects etc.)

*VoiceXML dialogs and their complexity.

*Encryption of audio/video media streams.

*Video transcoding.

*Dynamic or static video frame rate, bit rate or picture size adaptation per multimedia stream.

*Video record.

*Video RTSP streaming.

*Media insertion (audio, video, text, picture, logo, avatar or background/ambiance) in a multimedia stream.

*Video mixing.

*Video broadcasting.

*Face/shape/image detection/removal.

* -[MRB-07] Some of these features are not available in any package at the moment, but considering that additional packages might be written in the feature, it's probably ok to leave them there. What are your feelings about it?

> [MRB-08] There's another additional information that might be useful, something that might actually fit in the codec-related information. When we say MS[i] supports codec X, what kind of support are we talking about? Are both encoding and decoding supported? Is it passthrough only (e.g. I understand it but I won't transcode)? Can the MS encapsulate an X encoded stream according to the proper RFC? Such details would likely provide valuable information considering that it would affect how a conference mix, a prompt, a recording etc. would work inside the MS. Any comments about it?

5.2. Media Service Resource Consumer Interface

TOC

The Media Server Consumer interface provides the ability for clients of an MRB, such as Application Servers, to request an appropriate Media Server to satisfy specific criteria. The interface allows a client to pass detailed meta-information to the MRB to help select an appropriate Media Server. The MRB is then able to make and informed decision and provide the client with an appropriate media server resource.

It appears the most appropriate interface for such a 'query' style interface is in fact a RESTful type HTTP usage. Using HTTP and XML combined reduces complexity and encourages use of common tools that are widely available in the industry today. The following subsections explain the main operations required to request and then receive information from an MRB. The following description will describe the use of HTTP RFC 2616 (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.) [RFC2616] and HTTPS RFC 2818 (Rescorla, E., "HTTP Over TLS," May 2000.) [RFC2818] as transport for a query for media resource and the appropriate response. Examples of the interface can be seen in section [ref examples section].

> [MRB-09] Are we all ok with this or should we consider/list alternatives?

5.2.1. Media Service Resource Request

TOC

The media resource query is carried in the body of an HTTP/HTTPS POST request. The MIME type contained in the HTTP/HTTPS request/reponse should be 'application/mrb+xml'. This value MUST be reflected in the appropriate HTTP headers like 'Content-Type' and 'Accept'. The body of the POST request MUST only contain the 'mediaResourceRequest' element as defined in Section 6 (Media Service Resource Consumer Interface XML Schema). The 'mediaResourceRequest' element is the primary container of information related to a media resource request and has the following child elements which specify the request parameters:

5.2.1.1. element

TOC

The <mediaResourceRequest> element provides a container for clients wishing to query an external MRB entity. The <mediaResourceRequest> element has the following child elements that are used to provide appropriate contextual information relating to the request: [Editors Note: Convert groups input into appropriate XML schema.]

*RTP requirements - including media/codec type, codec priority.

*Conference requirements - number of users.

5.2.2. Media Service Resource Response

TOC

The use of HTTP/HTTPS for carrying the media service resource information has no impact on the protocol. If protocol level operations and errors occur then they should be signalled as specified in HTTP RFC 2616 (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," June 1999.) [RFC2616] and HTTPS RFC 2818 (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.) [RFC2119]. A successful response to a HTTP POST request containing the 'mediaResourceRequest' MUST be responded to with a 200 OK HTTP/HTTPS response message. This signifies that the request was received, was valid and could be responded to appropriately. If the receiving MRB wishes to generate information for the requesting entity it MUST include a 'mediaResourceResponse' element in the 200 OK HTTP/HTTPS response (as discussed later in this section). An MRB can alternatively return an application level error by including a 'mediaResourceError' element in the 200 OK HTTP/HTTPS response (as discussed later in this section).

> [MRB-10] As it was discussed when presenting CCMP, the use of 200 to convey responses (whether the resource has been found or not) and of error codes to handle HTTP-related errors is in contrast with a pure RESTful approach. Is it ok to proceed anyway? Or should we "lighten" the REST proposal just say something like "it's XML on HTTP", clarifying that we're not claiming to be pure RESTers?

5.2.2.1. element

TOC

The <mediaResourceResponse> element provides a container for the MRB to generate a response to a previous query. The <mediaResourceResponse> element has the following child elements that are used to provide appropriate contextual information relating to the request: [Editors Note: Convert groups input into appropriate XML schema.]

- list of appropriate media server resources (include individual capabilities).

TOC

5.2.2.2. element

The <mediaResourceError> element provides a container for the MRB to generate an error response to a previous query. The <mediaResourceError> has element the following child elements that are used to provide appropriate contextual information relating to the request: [Editors Note: Convert groups input into appropriate XML schema.]

*list of appropriate error response codes.

---

6. Media Service Resource Consumer Interface XML Schema

TOC

This section gives the XML Schema Definition [W3C.REC-xmlschema-1-20041028], [W3C.REC-xmlschema-2-20041028] of the "application/held+xml" format.

```
<?xml version="1.0"?>

<xsd:schema
  targetNamespace="urn:ietf:params:xml:ns:mediactrl:mrb"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:mrb="urn:ietf:params:xml:ns:mediactrl:mrb"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

<xsd:import namespace="http://www.w3.org/XML/1998/namespace"
           schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xsd:element name="mrb-message" type="mrb-message-type" />

 <xsd:complexType name="mrb-message-type">
  <xsd:sequence>
  <xsd:choice>
   <xsd:element name="mediaResourceRequest"
             type="mrb:mediaResourceRequestType"/>
   <xsd:element name="mediaResourceResponse"
             type="mrb:mediaResourceResponseType"/>
   <xsd:element name="mediaResourceError"
             type="mrb:mediaResourceErrorType"/>
   <xsd:any namespace="##other" minOccurs="0"
             maxOccurs="unbounded" processContents="lax" />

  </xsd:choice>
 </xsd:sequence>
 <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="mediaResourceRequestType">
  <xsd:complexContent>
      <xsd:sequence>
       <xsd:any namespace="##other" minOccurs="0"
             maxOccurs="unbounded" processContents="lax" />
      </xsd:sequence>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="mediaResourceResponseType">
  <xsd:complexContent>
      <xsd:sequence>
         <xsd:any namespace="##other" minOccurs="0"
             maxOccurs="unbounded" processContents="lax" />
      </xsd:sequence>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
```

```
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="mediaResourceErrorType">
  <xsd:complexContent>
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:any namespace="##other" processContents="strict"/>
      </xsd:choice>
      <xsd:attribute name="status" type="status.datatype"
            use="required"/>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexContent>
</xsd:complexType>

 <!-- DATATYPES -->

<xsd:simpleType name="status.datatype">
  <xsd:restriction base="xsd:NMTOKEN">
   <xsd:pattern value="[0-9][0-9][0-9]"/>
  </xsd:restriction>
 </xsd:simpleType>
      </xsd:complexContent>
</xsd:complexType>

</xsd:schema>
```

Figure 9

7. Acknowledgments

TOC

The authors would like to thank

8. Security Considerations

TOC

Security Considerations to be included in later versions of this document.

TOC

9. References

9.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2578]	McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)," STD 58, RFC 2578, April 1999 (TXT).
[RFC2579]	McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2," STD 58, RFC 2579, April 1999 (TXT).
[RFC2580]	McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2," STD 58, RFC 2580, April 1999 (TXT).
[RFC2616]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," RFC 2616, June 1999 (TXT, PS, PDF, HTML, XML).
[RFC2818]	Rescorla, E., "HTTP Over TLS," RFC 2818, May 2000 (TXT).
[RFC3410]	Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework," RFC 3410, December 2002 (TXT).
[W3C.CR-wsdl20-20051215]	Chinnici, R., Moreau, J., Ryman, A., and S. Weerawarana, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language," W3C CR CR-wsdl20-20051215, December 2005.
[W3C.REC-soap12-part1-20030624]	Mendelsohn, N., Gudgin, M., Moreau, J., Nielsen, H., and M. Hadley, "SOAP Version 1.2 Part 1: Messaging Framework," World Wide Web Consortium FirstEdition REC-soap12-part1-20030624, June 2003 (HTML).
[W3C.REC-soap12-part2-20030624]	Nielsen, H., Mendelsohn, N., Hadley, M., Moreau, J., and M. Gudgin, "SOAP Version 1.2 Part 2: Adjuncts," World Wide Web Consortium FirstEdition REC-soap12-part2-20030624, June 2003 (HTML).

9.2. Informative References

TOC

[I-D.ietf-mediactrl-architecture]	Melanchuk, T., "An Architectural Framework for Media Server Control," draft-ietf-mediactrl-architecture-04 (work in progress), November 2008 (TXT).
[I-D.ietf-mediactrl-requirements]	Dolly, M. and R. Even, "Media Server Control Protocol Requirements," draft-ietf-mediactrl-requirements-04 (work in progress), February 2008 (TXT).
[I-D.ietf-mediactrl-sip-control-framework]	Boulton, C., Melanchuk, T., and S. McGlashan, "Media Control Channel Framework," draft-ietf-mediactrl-sip-control-framework-11 (work in progress), October 2009 (TXT).

Authors' Addresses

TOC

	Chris Boulton
	NS-Technologies
Email:	chris@ns-technologies.com
	Lorenzo Miniero
	University of Napoli
Email:	lorenzo.miniero@unina.it