

[MS-SIPAPP]:

Session Initiation Protocol (SIP) Application Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
3/31/2010	0.1	Major	Initial Availability
4/30/2010	0.2	Editorial	Revised and edited the technical content
6/7/2010	0.3	Editorial	Revised and edited the technical content
6/29/2010	0.4	Editorial	Changed language and formatting in the technical content.
7/23/2010	0.4	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	1.0	Major	Significantly changed the technical content.
11/15/2010	1.0	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.0	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	1.0	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	1.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	1.1	Minor	Clarified the meaning of the technical content.
4/11/2012	1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	1.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	1.1.1	Editorial	Changed language and formatting in the technical content.
2/11/2013	1.1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	1.2	Minor	Clarified the meaning of the technical content.
11/18/2013	1.3	Minor	Clarified the meaning of the technical content.
2/10/2014	1.3	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	1.4	Minor	Clarified the meaning of the technical content.
7/31/2014	1.5	Minor	Clarified the meaning of the technical content.
10/30/2014	1.5	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2015	2.0	Major	Significantly changed the technical content.
9/4/2015	2.1	Minor	Clarified the meaning of the technical content.
7/15/2016	2.1	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	2.1	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
10/17/2016	2.1	None	No changes to the meaning, language, or formatting of the technical content.
4/27/2018	3.0	Major	Significantly changed the technical content.
7/24/2018	4.0	Major	Significantly changed the technical content.
8/28/2018	5.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	8
1.3	Overview	8
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.7.1	Versioning and Capability Negotiation for ms-call-park	9
1.8	Vendor-Extensible Fields	9
1.9	Standards Assignments.....	9
2	Messages.....	10
2.1	Transport.....	10
2.1.1	Transport for ms-call-park	10
2.2	Message Syntax.....	10
2.2.1	Message Syntax for ms-call-park	10
2.2.1.1	Park Request.....	10
2.2.1.2	Park Response.....	11
2.2.1.3	Unpark Notification.....	12
3	Protocol Details.....	14
3.1	ms-call-park Details	14
3.1.1	Abstract Data Model.....	15
3.1.2	Timers	15
3.1.3	Initialization.....	15
3.1.4	Higher-Layer Triggered Events	15
3.1.5	Message Processing Events and Sequencing Rules	15
3.1.5.1	Initiating a Request to Park a Call at the CPS	15
3.1.5.2	Parkee's Call Successfully Replaced by the CPS.....	16
3.1.5.3	CPS Fails to Replace the Parkee's Call.....	16
3.1.5.4	CPS Receives a BYE from the Parker.....	16
3.1.5.5	CPS Receives a BYE on the Audio Dialog with the Parkee	16
3.1.5.6	CPS Receives an INVITE with Audio SDP	17
3.1.5.7	CPS Transfer of the Retriever to the Parkee Succeeds.....	17
3.1.5.8	CPS Transfer of the Retriever to the Parkee Fails	17
3.1.5.9	CPS Transfer of the Parkee to the Parker Succeeds	18
3.1.5.10	CPS Transfer of the Parkee to the Parker Fails	18
3.1.5.11	CPS Transfer of the Parkee to the Fallback Succeeds	18
3.1.5.12	CPS Transfer of the Parkee to the Fallback Fails.....	18
3.1.6	Timer Events.....	19
3.1.7	Other Local Events.....	19
4	Protocol Examples.....	20
4.1	ms-call-park.....	20
4.1.1	Park a Call	20
4.1.2	Retrieve a Parked Call	24
4.1.3	Failure to Park a Call	30
4.1.4	Failure to Retrieve a Parked Call	30
4.1.5	Auto-Ringback Is Answered by the Parker.....	31
5	Security.....	34
5.1	Security Considerations for Implementers	34
5.2	Index of Security Parameters	34

6	Appendix A: Full XML Schema	35
6.1	ms-call-park XML Schema	35
7	Appendix B: Product Behavior	37
8	Change Tracking	38
9	Index	39

1 Introduction

This Session Initiation Protocol (SIP) Application Protocol document specifies the **ms-call-park** protocol that is used by the client to transfer a remote party of an existing two-party audio call to an inactive state with the purpose of later activation by the same or a different party.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

address-of-record: A **Session Initiation Protocol (SIP)** URI that specifies a domain with a location service that can map the URI to another URI for a user, as described in [\[RFC3261\]](#).

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [\[RFC5234\]](#).

auto-ringback: A process in which a **call park service (CPS)** automatically transfers a parked call from the **parking lot** to the user agent who originally parked the call.

call park service (CPS): A server endpoint that allows a user agent to make a call inactive without terminating that call. The call can then be reactivated by the same user agent, by using the same or a different endpoint, or a different user agent. See also **parking lot**.

Content-Type header: A message header field whose value describes the type of data that is in the body of the message.

fallback URI: A Uniform Resource Identifier (URI), as described in [\[RFC3986\]](#), that specifies the user agent address to which unretrieved calls are transferred.

Globally Routable User Agent URI (GRUU): A URI that identifies a user agent and is globally routable. A URI possesses a GRUU property if it is useable by any user agent client (UAC) that is connected to the Internet, routable to a specific user agent instance, and long-lived.

Internet Protocol version 4 (IPv4): An Internet protocol that has 32-bit source and destination addresses. IPv4 is the predecessor of IPv6.

Internet Protocol version 6 (IPv6): A revised version of the Internet Protocol (IP) designed to address growth on the Internet. Improvements include a 128-bit IP address size, expanded routing capabilities, and support for authentication and privacy.

INVITE: A **Session Initiation Protocol (SIP)** method that is used to invite a user or a service to participate in a session.

orbit: A number that uniquely identifies a parked call and enables a user agent to retrieve that call. The number is assigned automatically by a **call park service (CPS)** and is sent to the user agent who parked the call.

park: A process in which an active call is moved to a **parking lot**, without terminating that call. The call can then be retrieved by the same or another user agent. See also **call park service (CPS)**.

parkee: A user agent whose call is parked by another user agent, by using a **call park service (CPS)**. The parkee's call is not terminated and can be retrieved by the user agent who parked the call or a different user agent.

parker: A user agent who uses a **call park service (CPS)** to park a call. The call can then be retrieved by the same or a different user agent.

parking lot: A collection of one or more orbits that were configured by a **call park service (CPS)**. Each parked call is uniquely identified by the **orbit** that is assigned to it.

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

SIP message: The data that is exchanged between **Session Initiation Protocol (SIP)** elements as part of the protocol. An SIP message is either a request or a response.

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. **TLS** supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). **TLS** is standardized in the IETF TLS working group.

XML: The Extensible Markup Language, as described in [XML1.0].

XML attribute: A name/value pair, separated by an equal sign (=) and included in a tagged element, that modifies features of an element. All XML attribute values are stored as strings enclosed in quotation marks.

XML element: An **XML** structure that typically consists of a start tag, an end tag, and the information between those tags. Elements can have attributes and can contain other elements.

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring **XML schemas**.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/rfc/rfc2119.txt>

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., "SIP: Session Initiation Protocol", RFC 3261, June 2002, <http://www.ietf.org/rfc/rfc3261.txt>

[RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003, <http://www.ietf.org/rfc/rfc3515.txt>

[RFC3840] Rosenberg, J., Schulzrinne, H., and Kyzivat, P., "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004, <http://www.rfc-editor.org/rfc/rfc3840.txt>

[RFC3891] Mahy, R., Biggs, B., and Dean, R., "The Session Initiation Protocol (SIP) "Replaces" Header", RFC 3891, September 2004, <http://www.rfc-editor.org/rfc/rfc3891.txt>

[RFC4566] Handley, M., Jacobson, V., and Perkins, C., "SDP: Session Description Protocol", RFC 4566, July 2006, <http://www.ietf.org/rfc/rfc4566.txt>

1.2.2 Informative References

[MS-OCER] Microsoft Corporation, "[Client Error Reporting Protocol](#)".

[MS-SDPEXT] Microsoft Corporation, "[Session Description Protocol \(SDP\) Version 2.0 Extensions](#)".

[MS-SIPREG] Microsoft Corporation, "[Session Initiation Protocol \(SIP\) Registration Extensions](#)".

[MS-SIPRE] Microsoft Corporation, "[Session Initiation Protocol \(SIP\) Routing Extensions](#)".

[MS-SIP] Microsoft Corporation, "[Session Initiation Protocol Extensions](#)".

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/rfc/rfc5234.txt>

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, <http://www.w3.org/TR/2004/REC-xml-20040204/>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA0] Fallside, D., and Walmsley, P., Eds., "XML Schema Part 0: Primer, Second Edition", W3C Recommendation, October 2004, <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>

1.3 Overview

The **ms-call-park** protocol is used by a SIP client to transfer the remote party of an existing two-party audio call to a **parking lot** on the **call park service (CPS)**. The CPS issues an **orbit** to the client who **parked** the call. This orbit is then communicated via any out-of-band social process to another user or set of users. One of these users can then initiate a SIP call to the orbit and connect to the parked client, removing the parked client from the parking lot.

1.4 Relationship to Other Protocols

The protocols described in this document depend upon Session Initiation Protocol (SIP).

This document defines one or more **XML schemas** that support certain SIP application protocols. For more information about **XML** and **XML schema definition (XSD)**, see [\[XML10\]](#), [\[XMLNS\]](#), and [\[XMLSCHEMA0\]](#).

1.5 Prerequisites/Preconditions

This specification assumes that both client and server support SIP, and that they implement the extensions specified in the following extension specifications as needed:

- Session Initiation Protocol Extensions [\[MS-SIP\]](#)
- Session Initiation Protocol Routing Extensions [\[MS-SIPRE\]](#)
- Session Initiation Protocol Registration Extensions [\[MS-SIPREGE\]](#)

1.6 Applicability Statement

The protocols in this document are applicable when both client and server support SIP and intend to support one or more of the enhancements offered by these protocols.

1.7 Versioning and Capability Negotiation

The versioning and capability negotiation for each individual protocol is described in the following subsections.

1.7.1 Versioning and Capability Negotiation for ms-call-park

The ms-call-park protocol contains version information, whose significance is described in section [2.2.1.1](#), section [2.2.1.2](#), and section [2.2.1.3](#). Subsections under section [3.1.5](#) describe how this versioning information is processed. There is no capability negotiation in the protocol.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Each of the following subsections defines the message transport for one of the protocols within the scope of this document.

2.1.1 Transport for ms-call-park

The **ms-call-park** protocol does not introduce any new underlying transport for the exchange of protocol messages. **ms-call-park XML** protocol messages MUST be carried in the body of **SIP messages**, as specified in [\[RFC3261\]](#) section 18. The **SIP Content-Type header** MUST have a value of "application/ms-call-park+xml" for any body that is an **ms-call-park XML** message. SIP messages are transported over **Transmission Control Protocol (TCP)** or **Transport Layer Security (TLS)** for **Internet Protocol version 4 (IPv4)/Internet Protocol version 6 (IPv6)**[\[1\]](#).

2.2 Message Syntax

Each of the following subsections defines the message syntax for one of the protocols in the scope of this document.

2.2.1 Message Syntax for ms-call-park

The messages for **ms-call-park** are XML-based, and are constrained by an **XML schema**. For more information, see section [6](#). The top-level element for an **ms-call-park** protocol message MUST be a **park-request**, a **park-response**, or an **unpark-notification**. The syntax for each of these is defined in the following subsections, as well as that of additional elements that are not top-level.

When a client or server receives optional **XML elements** or **XML attributes** that it does not understand, and the message is valid according to the XML schema, it MUST ignore those optional elements and attributes that it does not understand.

2.2.1.1 Park Request

A **Park Request** message is sent from a client to the CPS when it wishes to park a call. A park-request element MUST be of type park-request-type and it MUST appear only in the body of a SIP INVITE.

```
<xs:element name="park-request" type="tns:park-request-type"/>

<xs:complexType name="park-request-type">
  <xs:sequence>
    <xs:element name="audio" type="tns:modality-park-request-type"/>
    <xs:any namespace="##other" processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="version" type="xs:string" use="required"/>
  <xs:attribute name="request-id" type="xs:string" use="required"/>
</xs:complexType>
```

park-request Element: The client MUST set the version attribute of the park-request to a string which indicates the major (left of dot) and minor (right of dot) **ms-call-park** protocol versions. The string's value MUST conform to the following ABNF representation, as defined in [\[RFC5234\]](#):

```
1*DIGIT "." 1*DIGIT
```

The **park-request** MUST include a **request-id** attribute, whose value is an arbitrary string that will be used to correlate responses and notifications with their corresponding request in future versions of this protocol.

The park-request element MUST contain a single audio element. It MAY also contain other arbitrary media types in the same request as additional elements under the request.

audio Element: A single audio element MUST appear inside of a **park-request** element. The audio element MUST be of type **modality-park-request-type** and MUST contain one **dialog-info** element and one **target** element. No attributes are defined for the **audio** element.

```
<xs:complexType name="modality-park-request-type">
  <xs:sequence>
    <xs:element name="dialog-info" type="tns:dialog-info-type"/>
    <xs:element name="target" type="xs:anyURI"/>
  </xs:sequence>
</xs:complexType>
```

The purpose of the **audio** element is to encapsulate all the information necessary to locate the audio call and the client that is to be parked.

target Element: A single target element MUST appear inside of an audio element. Its value MUST be a URI. The purpose of this element is to specify how to find the user agent (UA) that is to be parked.

dialog-info Element: A single dialog-info element MUST appear inside of an audio element. A dialog-info element MUST be of type **dialog-info-type** and MUST contain one call-id element, one from-tag element, and one to-tag element. No attributes are defined for the dialog-info element.

```
<xs:complexType name="dialog-info-type">
  <xs:sequence>
    <xs:element name="call-id" type="xs:string"/>
    <xs:element name="from-tag" type="xs:string"/>
    <xs:element name="to-tag" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

The purpose of the **dialog-info** element is to encapsulate the three pieces that uniquely identify the call that is to be parked. These correspond to the entities of the same name as defined in [\[RFC3261\]](#) section 12.

call-id Element: A single **call-id** element MUST appear inside of a **dialog-info** element. A call-id element MUST contain a string value that corresponds to the **SIP call-id** of the call that is to be parked.

from-tag Element: A single from-tag element MUST appear inside of a **dialog-info** element. A **from-tag** element MUST contain a string value that corresponds to the SIP **from-tag** of the call that is to be parked.

to-tag Element: A single **to-tag** element MUST appear inside of a **dialog-info** element. A **to-tag** element MUST contain a string value that corresponds to the SIP **to-tag** of the call that is to be parked.

2.2.1.2 Park Response

A **Park Response** message is sent from the CPS to the client after a park request has been successfully fulfilled. This response message MUST contain a **park-response** element, which MUST be of type **park-response-type**. The response message MUST appear in the body of a SIP 200 OK.

```
<xs:element name="park-response" type="tns:park-response-type"/>
<xs:complexType name="park-response-type">
```

```

<xs:sequence>
  <xs:element name="orbit" type="xs:string" />
  <xs:element name="ms-parked-call" type="xs:string" />
</xs:sequence>
<xs:attribute name="version" type="xs:string" use="required"/>
<xs:attribute name="request-id" type="xs:string" use="required"/>
</xs:complexType>

```

The version and request-id attributes MUST be as defined in section [2.2.1.1](#).

orbit Element: A single orbit element MUST appear in a **Park** response message. Its value MUST be set to a string that a client can call to retrieve the call from the parking lot. The CPS MUST generate only orbits that conform to the following ABNF syntax: **Augmented Backus-Naur Form (ABNF)** syntax, as defined in [\[RFC5234\]](#):

```

ORBIT = (UNPREFIXED-ORBIT / PREFIXED-ORBIT)

UNPREFIXED-ORBIT = 1*9DIGIT

PREFIXED-ORBIT = ("*" / "#") 1*8DIGIT

```

The CPS MAY limit the orbit pool to any subset of that allowed this syntax. The CPS orbit pool SHOULD NOT overlap with any internal enterprise extensions.

ms-parked-call Element: A single **ms-parked-call** element MUST appear inside of a **Park** response message. It is a unique string that identifies a particular parked call. A retriever can use this to guarantee that the call being retrieved is the correct one (because orbits can be re-used over time).

2.2.1.3 Unpark Notification

The **unpark-notification** element is sent by the CPS to the parker to inform it of the ultimate disposition of the audio call that it parked. It MUST be of type **unpark-notification-type**. The response message MUST appear in the body of a **SIP INFO**.

```

<xs:element name="unpark-notification" type="tns:unpark-notification-type"/>

<xs:complexType name="unpark-notification-type">
  <xs:sequence>
    <xs:element name="reason" type="tns:unpark-reason-type"/>
    <xs:element name="target" type="xs:anyURI" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="version" type="xs:string" use="required"/>
  <xs:attribute name="request-id" type="xs:string" use="required"/>
</xs:complexType>

```

The version and request-id attributes MUST be as defined in section [2.2.1.1](#).

The **unpark-notification** element MUST contain a single reason element. If the reason element contains a value of "retrieval" or "fallback", the **unpark-notification** MUST also contain a single target element. If the reason element contains a value of "ringback", the **unpark-notification** MAY contain a single **target** element. If the reason element contains a value of "hang-up" or "drop", the **target** element MUST NOT appear in the **unpark-notification**.

reason Element: A single **reason** element MUST appear inside of an **unpark-notification** element. The **reason** element MUST have a value from a specific set of strings that correspond to the various reasons for which a call can become unparked. The valid values for the reason element are as follows:

- **retrieval:** The parked call was retrieved because a user called the orbit and successfully retrieved the parked call.
- **hang-up:** The parked client disconnected the parked call before it was retrieved.
- **ringback:** The same SIP user who parked the call answered an **auto-ringback** from the CPS.
- **fallback:** No user successfully retrieved the call, all attempts to perform an auto-ringback to the parker went unanswered, and the parked call was successfully sent to the provisioned fallback URI.
- **drop:** No user successfully retrieved the call, all attempts to perform an auto-ringback to the parker went unanswered, the call was not successfully sent to any fallback URI, and as a last resort, the call was disconnected by the CPS.

```
<xs:simpleType name="unpark-reason-type">  
  <xs:restriction base="xs:string">  
    <xs:enumeration value="retrieval"/>  
    <xs:enumeration value="hang-up"/>  
    <xs:enumeration value="ringback"/>  
    <xs:enumeration value="fallback"/>  
    <xs:enumeration value="drop"/>  
  </xs:restriction>  
</xs:simpleType>
```

target Element: A single target element can also appear inside of an **unpark-notification** element. If present, the target element's value MUST be the **address-of-record** of the SIP user that has retrieved the parked call.

3 Protocol Details

3.1 ms-call-park Details

This section describes the details of the ms-call-park protocol. The following diagram illustrates the state transitions for the CPS, the events that precipitate them, and the actions taken at each. The details for each of these are defined in the subsections that follow.

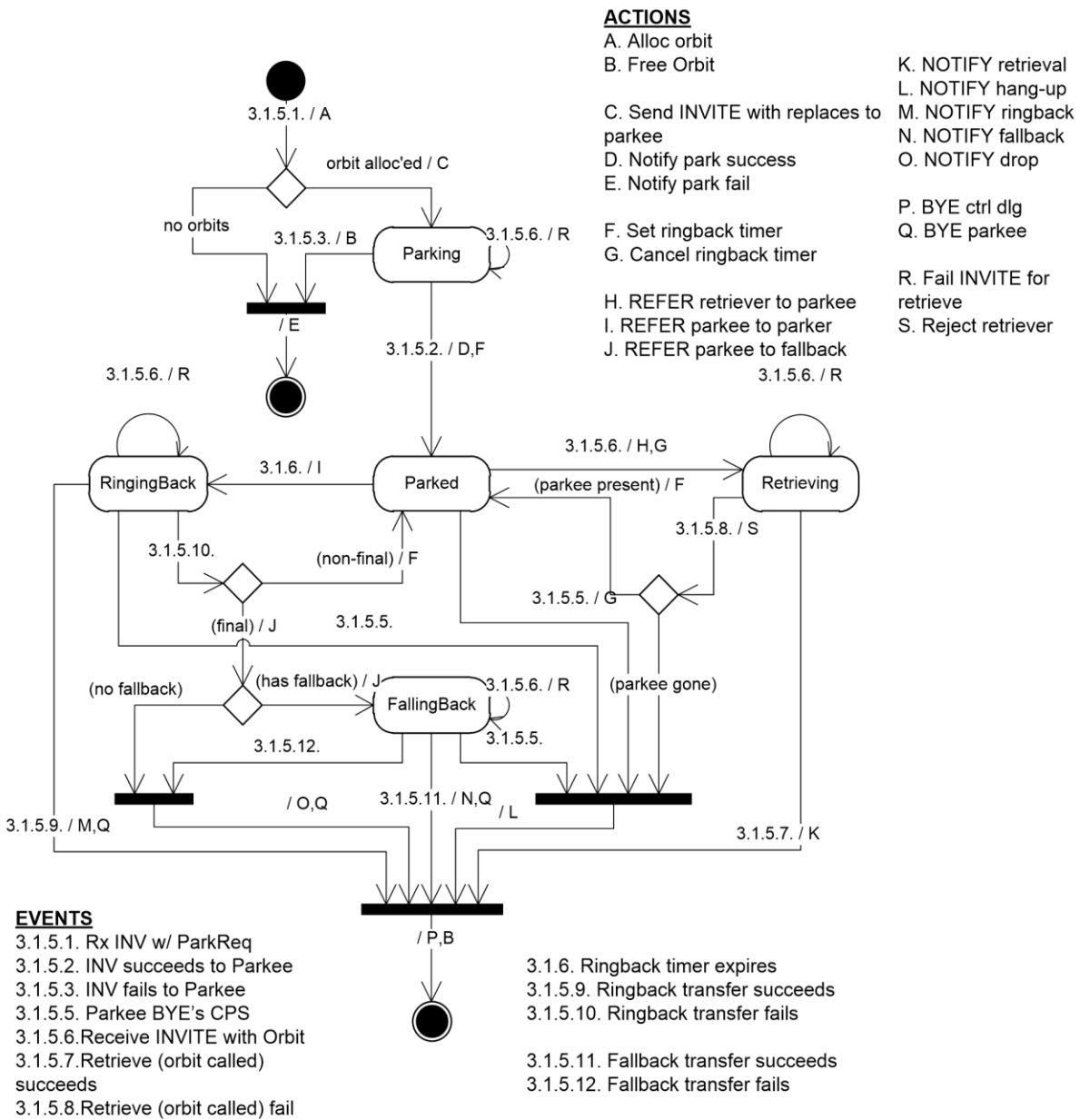


Figure 1: CPS State Transitions

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The CPS MUST maintain a separate instance of the following abstract data model elements for each parked call:

- **Dialog information for the parked call:** This is a requirement of SIP in general, and is not specific to this protocol.
- **request-id:** The string that was supplied by the client when the park action was initiated.
- **Orbit:** The number that can be called to retrieve the parked call.
- **ms-parked-call string:** An additional, opaque string that can be supplied along with the retriever's INVITE to guarantee that the call being retrieved is the same one that was parked, even if a different call is later parked on the same orbit.
- **auto-ringback:** A count for the number of auto-ringback retries. This starts at zero and is incremented after each auto-ringback attempt, but not beyond the configured maximum.

The preceding data MAY be maintained in a persistent store to survive CPS restarts.

3.1.2 Timers

Beyond the timers specified in [\[RFC3261\]](#), ms-call-park requires an **auto-ringback** timer on the CPS. Each parked call MUST maintain its own instance of this timer. The auto-ringback **timer** allows parked calls that have gone unretrieved after a period of time to be transferred back to the parker, to a **fallback URI**, or to be disconnected to prevent resource depletion. The duration for this timer is not defined by this protocol; an implementation is free to choose any appropriate duration, or to make its duration user-configurable.

3.1.3 Initialization

During server restarts, any persistently stored information, as specified in section [3.1.1](#), MAY be read back in order for the **CPS** to correctly restore its state to that prior to the restart.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

Unless otherwise noted, the rules for message processing and error handling specified in [\[RFC3261\]](#) section 13.3 and section 13.3.1.3 and [\[RFC4566\]](#) section 5 MUST be followed. Additional processing of incoming messages is noted in the sections that follow.

3.1.5.1 Initiating a Request to Park a Call at the CPS

When a client attempts to park an existing call at the CPS, it MUST initiate a dialog with the CPS by sending it a **SIP INVITE** message. The **Content-Type** header MUST be set to "application/ms-call-park+xml", and the body of the **INVITE** MUST be a **park-request** element, as defined in section [2.2.1.1](#). The request dialog MUST be initiated to the Call Park Server URI as discovered by in-band provisioning, as defined in [\[MS-SIPREGE\]](#) section 2.2.2.5.1.

The client MUST indicate a version of "1.0" in the **park-request**. The client SHOULD supply an arbitrary string for the **request-id**.

The park-request element MUST have a single **audio** element. The audio element MUST contain a **dialog-info** element with the SIP dialog information of the call that is to be parked. The client MUST set the **target** element to the **Globally Routable User Agent URI (GRUU)** of the party to that call that is to be parked at the CPS. The **park-request** element MAY have other arbitrary elements representing additional media types, but the CPS MUST ignore them.

If the CPS cannot parse or validate the body of an application/ms-call-park+xml **INVITE** according to the **XML schema**, it MUST reject the request with a SIP 415.

If the CPS receives a major version of anything other than "1", it can reject the request by replying with a SIP 488. This rejection SHOULD include an ms-diagnostics header with a value of 35008 as defined in [\[MS-OCER\]](#).

When the server receives a syntactically valid park-request from a client, it MUST attempt to allocate an available **orbit** from its pool. The server can use any type of algorithm to choose an available orbit from the pool, and can attempt to reduce the predictability of the orbit selection or frequent re-use of identical orbits. If no orbit can be allocated, the CPS MUST respond to the **INVITE** with a SIP 500. Such a response SHOULD include an ms-diagnostics of 35002 as defined in [\[MS-OCER\]](#). The CPS MUST also generate an arbitrary **ms-parked-call** string that is unique across all calls parked on a given orbit over time.

If the orbit was successfully allocated, the CPS MUST send a SIP **INVITE** with **Replaces** to the GRUU that was indicated in the target element of the **park-request**. The **Replaces** header is defined in [\[RFC3891\]](#). The CPS MUST identify itself by adding an "isCps" feature tag to its Contact URI. The **INVITE** MUST be audio-only and the SDP of the **INVITE** SHOULD indicate either "hold" or "music-on-hold". See [\[MS-SDPEXT\]](#) section 3.1.5.27 for details of **music-on-hold**.

3.1.5.2 Parkee's Call Successfully Replaced by the CPS

If the **INVITE** with **Replaces** to the parkee successfully establishes a dialog, the CPS MUST respond to the parker's ms-call-park SIP **INVITE** with a SIP 200 OK. The CPS MUST identify itself by adding an "isCps" feature tag to its Contact URI. Feature tags are defined in [\[RFC3840\]](#) section 10. The response from the CPS MUST have a body of type "application/ms-call-park+xml" and be a **park-response** element. Its **orbit** element MUST contain the value of the orbit that was chosen to park the call. The **ms-parked-call** element MUST contain a string chosen by the CPS that uniquely identifies the parked call. The version attribute of the **park-response** element MUST be set to "1.0" and the **request-id** attribute MUST match that of the corresponding **park-request**.

The auto-ringback timer for this parked call MUST be started once park completes.

3.1.5.3 CPS Fails to Replace the Parkee's Call

If the **INVITE** with **Replaces** to the **parkee** fails to establish a dialog, the CPS MUST respond to the parker's ms-call-park SIP **INVITE** with SIP 408. Such a response SHOULD have an **ms-diagnostics** of 35007 as defined in [\[MS-OCER\]](#).

3.1.5.4 CPS Receives a BYE from the Parker

CPS MUST process a **BYE** from the **parker** as specified in [\[RFC3261\]](#) section 15. The CPS MUST NOT take any other action, such as terminate the parked call, simply as a result of receiving a SIP **BYE** on the **ms-call-park** dialog. If this dialog is ended, the CPS MUST NOT send any **unpark-notifications** related to the call that was parked on the terminated dialog.

3.1.5.5 CPS Receives a BYE on the Audio Dialog with the Parkee

CPS MUST process the **BYE** as specified in [\[RFC3261\]](#) section 15. If the CPS is in the process of performing a transfer of the retriever to the parkee or if the parker-CPS dialog no longer exists, the

CPS MUST NOT perform further processing for this event. Otherwise, the following additional steps apply.

The CPS MUST send a SIP **INFO** message on parker-CPS dialog with a **Content-Type** of "**application/ms-call-park+xml**". The body of this message MUST be an **unpark-notification**. The **version** attribute MUST be set to "1.0". The **request-id** attribute MUST be set to that of the original **park-request**. The **reason** element MUST be set to "hang-up", and the **target** element MUST NOT be present.

After sending the preceding notification, the CPS MUST then send a **BYE** on the **parker-CPS** dialog. This **BYE** SHOULD include an **ms-diagnostics** code of 35014 as defined in [\[MS-OCER\]](#). The CPS MUST return the orbit to the pool, and release any resources associated with the parked call.

3.1.5.6 CPS Receives an INVITE with Audio SDP

When a SIP Proxy that supports **ms-call-park** receives an **INVITE** to a number which falls inside its configured orbit pool, it MUST route the **INVITE** to the CPS entity. The CPS MUST attempt to find the parked call that corresponds to the orbit that was called. If no call is parked against this orbit, the CPS MUST reject the **INVITE** with a SIP 404. An **ms-diagnostic** code of 35000, as described in [\[MS-OCER\]](#) section 9.2.2, SHOULD accompany this rejection. If a call is still in the process of being parked at the orbit or is in the process of retrieval by another client, **auto-ringback**, or fallback, the CPS MUST reject the **INVITE** with a SIP 488.

If the orbit has successfully mapped to a parked call after the previous checks, the CPS MUST accept the **INVITE**. The CPS MUST identify itself by adding an "isCps" **feature** tag to its Contact URI. Feature tags are defined in [\[RFC3840\]](#) section 10. After the dialog is established, the CPS MUST cancel the ringback timer and send a **REFER** with **Replaces** to the retriever. The SIP **REFER** method is defined in [\[RFC3515\]](#). The **Refer-To** header MUST be set to the **GRUU** of the parkee for the call being retrieved, and the replaces **dialog-info** MUST be set to that of the **parkee-CPS** dialog.

3.1.5.7 CPS Transfer of the Retriever to the Parkee Succeeds

If the transfer of the retriever to the parkee succeeds, and the **ms-call-park** dialog from the parker to the CPS still exists, the CPS MUST send a SIP **INFO** message on this dialog with a **Content-Type** of "**application/ms-call-park+xml**". The body of this message MUST be an **unpark-notification**. The **version** attribute MUST be set to "1.0". The **request-id** attribute MUST be set to that of the original **park-request**. The reason element MUST be set to "retrieval", and the **target** element MUST be set to the **address-of-record** of the SIP user that retrieved the call.

The CPS MUST then send a **BYE** on the **parker-CPS** dialog. This **BYE** SHOULD include an **ms-diagnostics** code of 35014 as defined in [\[MS-OCER\]](#). The CPS MUST return the orbit to the pool, and release any resources associated with the parked call.

3.1.5.8 CPS Transfer of the Retriever to the Parkee Fails

If the transfer of the retriever to the parkee fails, it MUST terminate the **retriever-CPS** dialog with a SIP **BYE**. If the **parkee-CPS** dialog still exists, the auto-ringback timer is restarted.

If the **parkee-CPS** dialog no longer exists, but the parker-CPS **ms-call-park** dialog does exist, the CPS MUST send a SIP **INFO** message on this latter dialog with a **Content-Type** of "**application/ms-call-park+xml**". The body of this message MUST be an unpark-notification. The version attribute MUST be set to "1.0". The **request-id** attribute MUST be set to that of the original **park-request**. The **reason** element MUST be set to "hang-up", and the **target** element MUST NOT be present.

After sending the preceding notification, the CPS MUST then send a **BYE** on the **parker-CPS** dialog. This **BYE** SHOULD include an **ms-diagnostics** code of 35014 as defined in [\[MS-OCER\]](#). The CPS MUST return the orbit to the pool, and release any resources associated with the parked call.

3.1.5.9 CPS Transfer of the Parkee to the Parker Succeeds

When a transfer of the parkee to the parker (**auto-ringback**) is successful, as described by steps 1 through 10 in the call flow of section [4.1.5](#), and the **ms-call-park** dialog from the parker to the CPS still exists, the CPS MUST send a SIP **INFO** message on this dialog with a **Content-Type** of "application/ms-call-park+xml". The body of this message MUST be an **unpark-notification**. The **version** attribute MUST be set to "1.0". The **request-id** attribute MUST be set to that of the original **park-request**. The reason element MUST be set to "ringback". If the target element is present, it MUST be set to the **address-of-record** of the parker.

The CPS MUST then send a **BYE** on the parker-CPS dialog. This **BYE** SHOULD include an **ms-diagnostics** code of 35014 as defined in [\[MS-OCER\]](#). The CPS MUST return the orbit to the pool, and release any resources associated with the parked call.

3.1.5.10 CPS Transfer of the Parkee to the Parker Fails

When a transfer of the parkee to the parker (auto-ringback) fails because the parkee disconnects the call during the attempted transfer, the event MUST be handled as described in section [3.1.5.5](#). Otherwise, the following processing applies.

The **CPS** MUST increment the ringback attempt count. If the count has not yet reached the maximum, the **auto-ringback** timer MUST be restarted and additional processing of this event MUST NOT be done. Otherwise the following processing applies. The maximum number of auto-ringback attempts is not defined in the **ms-call-park** protocol; an implementation is free to choose any maximum, or to make the maximum user-configurable.

If the maximum ringback count has been reached and there is a configured fallback URI, the CPS MUST initiate a transfer of the parkee to that fallback URI. This MUST be done by sending a SIP **REFER** to the parkee. The **Refer-To** target of the transfer MUST be the configured fallback URI.

If the maximum ringback count has been reached and there is no configured fallback URI, the CPS MUST terminate the call with the parkee by sending a SIP **BYE**. If the control dialog with the parker is still active, the CPS MUST send an **unpark-notification** with a **reason** code of "drop". This notification MUST NOT contain a **target** element. After sending this notification, the CPS MUST then send a **BYE** on the **parker-CPS** dialog. This **BYE** SHOULD include an **ms-diagnostics** code of 35014 as defined in [\[MS-OCER\]](#). The CPS MUST return the orbit to the pool and release any resources associated with the parked call.

3.1.5.11 CPS Transfer of the Parkee to the Fallback Succeeds

When a transfer of the parkee to the fallback URI successfully completes, the **CPS** MUST terminate its audio dialog with the **parkee**. If **CPS-parker ms-call-park** dialog still exists, the CPS MUST send an **unpark-notification** on a SIP **INFO** in the control dialog with the parker. The **reason** code in this **unpark-notification** element MUST be "fallback", and the **target** element SHOULD be set to the fallback URI. After sending this notification, the CPS MUST send a **BYE** on the **parker-CPS** dialog. This **BYE** SHOULD include an **ms-diagnostics** code of 35014 as defined in [\[MS-OCER\]](#). The CPS MUST return the orbit to the pool and release any resources associated with the parked call.

3.1.5.12 CPS Transfer of the Parkee to the Fallback Fails

When a transfer of the parkee to the fallback URI fails because the parkee disconnects the call during the attempted transfer, the event MUST be handled as described in section [3.1.5.5](#). Otherwise, the following processing applies (for example: if the **parkee** has placed the call with **CPS** on hold).

When a transfer of the parkee to the fallback URI fails for any other reason, the CPS MUST terminate the call with the parkee. If the control dialog with the parker is still active, the CPS MUST send an **unpark-notification** with a **reason** code of "drop". This notification MUST NOT contain a **target** element. After sending this notification, the CPS MUST then send a **BYE** on the **parker-CPS** dialog.

This **BYE** SHOULD include an **ms-diagnostics** code of 35014 as defined in [\[MS-OCER\]](#). The CPS MUST return the orbit to the pool and release any resources associated with the parked call.

3.1.6 Timer Events

When the **auto-ringback** timer expires, the **CPS** MUST send a **REFER** to the parkee. The **Refer-To** target of the transfer MUST be the parker's **address-of-record**. The **Refer-To** MUST contain the escaped header "ms-sensitivity=private-no-diversion", so that the triggered **INVITE** will contain this SIP header. This header's syntax and semantics are defined in [\[MS-SIPRE\]](#) section 2.2.10 and 3.9.5.2.1.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following sections provide example of the protocols defined earlier in this document.

4.1 ms-call-park

The following sections contain examples of the ms-call-park protocol. Note the consistent convention throughout the set of examples in these subsections:

- The call park service (CPS) is a logical entity inside a server that implements this protocol.
- Alice, Bob, Carol, and Dave are arbitrary users of a user agent client (UAC) that implements this protocol.
- Alice is the user that parks the call.
- Bob is the user whose call is parked.
- Carol is a user who retrieves Bob from the **CPS**.
- Dave is the user whose SIP URI was previously configured in the CPS as the fallback.

Note that, for clarity, the following are omitted from the diagrams in the following examples (because they have no bearing on the ms-call-park protocol):

- SIP "1xx" provisional responses
- non-final NOTIFY messages for the REFER dialogs
- ICE re-INVITEs

Each of the preceding, however, is generated and processed according to the specifications in [\[RFC3261\]](#).

4.1.1 Park a Call

In this example, Alice and Bob are already in an audio conversation, and Alice decides to park Bob at the CPS, as shown in the following diagram.

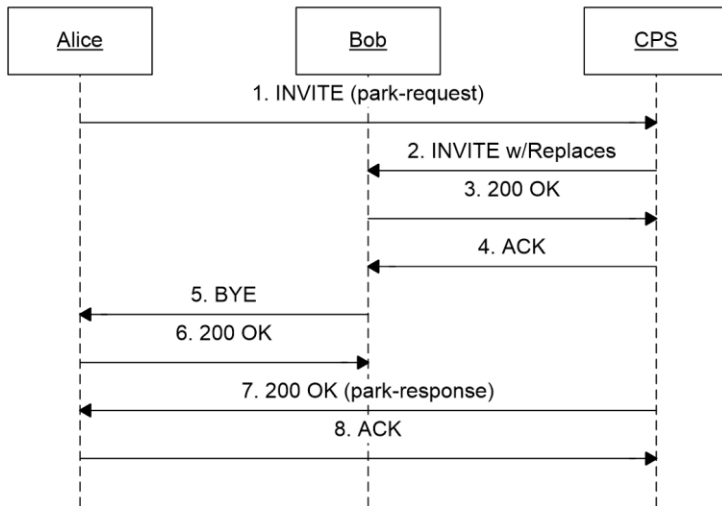


Figure 2: Call parking call flow

The following are the sequence of events:

1. Alice sends a SIP **INVITE** with a park-request element to the **CPS**, specifying the communication medium to be parked (audio), the dialog information for the call, and the **GRUU** of the user that is in the referenced call.

```

INVITE sip:callparkservice.contoso.com@
contoso.com;gruu;opaque=srvr:Microsoft.Rtc.Applications.Cps:4H28w6olDlm9KG3n15tyBQAA SIP/2.0
Via: SIP/2.0/TLS 172.24.32.147:53243
Max-Forwards: 70
From: <sip:alice@contoso.com>;tag=f7f67d48c8;epid=c9a8219cde
To:
<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=srvr:Microsoft.Rtc.Applications.Cps:
4H28w6olDlm9KG3n15tyBQAA>
Call-ID: d80ebcb31ffd4d6bbe4061291478dc00
CSeq: 1 INVITE
Contact: <sip:alice@contoso.com;opaque=user:epid:zNefM3iQbV6i7RUjfAv4LQAA;gruu>
User-Agent: UCCAPI/4.0.7259.501 OC/4.0.7259.501 (Microsoft Communicator 2010 (Beta))
Ms-Conversation-ID: AcqulstFttoMXPODT9SSd7WYs+haWQ==
Supported: timer
Supported: histinfo
Supported: ms-safe-transfer
Supported: ms-sender
Supported: ms-early-media
ms-keep-alive: UAC;hop-hop=yes
Allow: INVITE, BYE, ACK, CANCEL, INFO, UPDATE, REFER, NOTIFY, BENOTIFY, OPTIONS
Proxy-Authorization: NTLM qop="auth", realm="SIP Communications Service", opaque="F1C4F39A",
targetname="callparkservice.contoso.com", crand="6d34b235", cnum="180",
response="010000003231396311f6bb00c225885e"
Content-Type: application/ms-call-park+xml
Content-Length: 357
<?xml version="1.0"?>
<park-request version="1.0" request-id="1">
  <audio>
    <dialog-info>
      <call-id>0fa2623eb27c4fa1b61e5ea6544e8dd1</call-id>
      <from-tag>fdb9094ccb</from-tag>
      <to-tag>5d42553c6f</to-tag>
    </dialog-info>
  
```

```
<target>sip:bob@contoso.com;opaque=user:epid:NbwN8db8nluZbHVdCgJh0AAA;gruu</target>
</audio>
</park-request>
```

2. The CPS sends an **INVITE** with **Replaces** to the parkee. Here, it has SDP and a **Contact** header that indicate that it is streaming music on hold.

```
INVITE sip:172.24.33.241:51780;transport=tls;ms-opaque=1d722d38d1;ms-received-cid=93C000;grid
SIP/2.0
Record-Route:
<sip:callparkservice.contoso.com:5061;transport=tls;opaque=state:T:F:Ci.R93c000:Ieh.wPN9F9YaC
xPXGgsTfcEVACdwFcvCY35DVD11HoRz7Nb-iNuIfe2BYn24cYBCwX4qWl3ELVtwAA;lr;ms-route-
sig=fcpqENIhj3z5BDaZvtNxxXuRIOkm0Rdwq-
jYxwX9sh0DgX4qWl3ELVtwAA>;tag=03DDFB3C955DC1EEFA829A26150FDBAD
Via: SIP/2.0/TLS 172.29.105.97:5061;branch=z9hG4bKF04DE060.A5A5E217A610871A;branched=TRUE;ms-
internal-info="dp3PPLUp1SqNd-3b37waSFX0IMn3CV2L4-kSwlQH2ie34X4qWlsux72QAA"
Authentication-Info: TLS-DSK qop="auth", opaque="6C58FF34", srand="DF4B3AB2", snum="248",
rspauth="4ccbe7bee9e868972257f7d22695308cc22136ad", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Max-Forwards: 69
Content-Length: 3745
Via: SIP/2.0/TLS 172.29.105.97:52493;branch=z9hG4bKacf610b7;ms-received-port=52493;ms-
received-cid=971400
P-Asserted-Identity: "Call Park Service"<sip:RtcApplication-49804d56-9297-4f58-be0a-
0872df68b817@contoso.com>
From: "Call Park Service"<sip:RtcApplication-49804d56-9297-4f58-be0a-
0872df68b817@contoso.com>;epid=9FC09D8ABF;tag=89e629df39
To: <sip:bob@contoso.com;opaque=user:epid:NbwN8db8nluZbHVdCgJh0AAA;gruu>;epid=38e610bbd5
CSeq: 100 INVITE
Call-ID: 3734e209-9e6e-455e-8986-ac1acb135d0d
Contact:
<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=svr:Microsoft.Rtc.Applications.Cps:
4H28w6olDlm9KG3n15tyBQAA>;text;audio;video;image;+sip.rendering="no";isCps
Expires: 600
Priority: Normal
Supported: Replaces
Supported: ms-dialog-route-set-update
Supported: timer
Supported: 100rel
Supported: gruu-10
User-Agent: RTCC/4.0.0.0 Call_Park_Service_1.0
Content-Type: multipart/alternative; boundary=aMHda3MG7oSOMCuhrx23ZCRDS9ibp9Vk
Allow: ACK
replaces: 0fa2623eb27c4fa1b61e5ea6544e8dd1;from-tag=fdb9094ccb;to-tag=5d42553c6f
Ms-Conversation-ID: b3e492f404ea4a7f9ac521f741d61bb3
Session-Expires: 320
Min-SE: 90
Allow: CANCEL,BYE,INVITE,MESSAGE,INFO,SERVICE,OPTIONS,BENOTIFY,NOTIFY,PRACK,UPDATE
History-Info: <sip:bob@contoso.com>;index=1
--aMHda3MG7oSOMCuhrx23ZCRDS9ibp9Vk
Content-Type: application/sdp
Content-ID: 832e213d-1282-4044-89d3-elc21063e1e6
Content-Disposition: session;handling=optional;ms-proxy-2007fallback
v=0
o=- 90 0 IN IP4 172.29.105.97
s=session
c=IN IP4 172.29.105.97
b=CT:100000
t=0 0
m=audio 54594 RTP/SAVP 112 111 0 8 116 4 13 118 97
c=IN IP4 172.29.105.97
a=rtcp:54595
a=candidate:Qz9KjMOW314xWjjAisUuiqvW82Lz+1Ie9eC0G9iS/dw 1 KeJs2nKyybxoNxPEspecJw UDP 0.830
172.29.105.97 54594
a=candidate:Qz9KjMOW314xWjjAisUuiqvW82Lz+1Ie9eC0G9iS/dw 2 KeJs2nKyybxoNxPEspecJw UDP 0.830
172.29.105.97 54595
```

a=candidate: jKBvXbUbFuTZbkOo1GNxHh+vxZ92Fxlrmennv7KHS4w 1 n8w9Nozwe6xhmg/CcTx5wg TCP 0.150
172.29.104.128 53246
a=candidate: jKBvXbUbFuTZbkOo1GNxHh+vxZ92Fxlrmennv7KHS4w 2 n8w9Nozwe6xhmg/CcTx5wg TCP 0.150
172.29.104.128 53246
a=candidate: TBbV97WIGDJh2f0wcO3LgHCVswZT93Ms9mILCCaXJZ0 1 R4nqch2vZr63eY56p8Sa1w UDP 0.450
172.29.104.128 59031
a=candidate: TBbV97WIGDJh2f0wcO3LgHCVswZT93Ms9mILCCaXJZ0 2 R4nqch2vZr63eY56p8Sa1w UDP 0.450
172.29.104.128 53458
a=candidate: 56uDx5Qr3fqMel4PXaL1GNxiTNwFPbDMXV9EIlw2qPM 1 ckrQJz914WKbvjj092koFA TCP 0.250
172.29.105.97 49389
a=candidate: 56uDx5Qr3fqMel4PXaL1GNxiTNwFPbDMXV9EIlw2qPM 2 ckrQJz914WKbvjj092koFA TCP 0.250
172.29.105.97 49389
a=label: main-audio
a=cryptoscale: 1 client AES_CM_128_HMAC_SHA1_80
inline: U5AwqxEVWzlsWSC3ESNuqxZsyjF/8o2EVmpy3ghr|2^31|1:1
a=crypto: 2 AES_CM_128_HMAC_SHA1_80 inline: L+RBO8KufEk/IB+YwG6awdrvCdvZyW9gUYv5wks5|2^31|1:1
a=crypto: 3 AES_CM_128_HMAC_SHA1_80 inline: ijE6xzUzoeubNucTU2Q+Ti/j0EmlXBcd5U0oKA5n|2^31
a=sendonly
a=rtpmap: 112 G7221/16000
a=fmtp: 112 bitrate=24000
a=rtpmap: 111 SIREN/16000
a=fmtp: 111 bitrate=16000
a=rtpmap: 0 PCMU/8000
a=rtpmap: 8 PCMA/8000
a=rtpmap: 116 AAL2-G726-32/8000
a=rtpmap: 4 G723/8000
a=rtpmap: 13 CN/8000
a=rtpmap: 118 CN/16000
a=rtpmap: 97 RED/8000
--aMhda3MG7oSOMCuhrx23ZCRDS9ibp9Vk
Content-Type: application/sdp
Content-ID: 4ce3b538-08e8-47ea-ba52-d1c05b819e68
v=0
o=- 91 0 IN IP4 172.29.105.97
s=session
c=IN IP4 172.29.105.97
b=CT:100000
t=0 0
m=audio 55486 RTP/SAVP 112 111 0 8 116 4 13 118 97
c=IN IP4 172.29.105.97
a=rtcp:55487
a=ice-ufrag: MDhj
a=ice-pwd: tANsCintAKGCI2C+TBKJzZi9
a=candidate: 1 1 UDP 2130706431 172.29.105.97 55486 typ host
a=candidate: 1 2 UDP 2130705918 172.29.105.97 55487 typ host
a=candidate: 2 1 tcp-pass 6555135 172.29.104.128 58409 typ relay raddr 172.29.105.97 rport
52609
a=candidate: 2 2 tcp-pass 6555134 172.29.104.128 58409 typ relay raddr 172.29.105.97 rport
52609
a=candidate: 3 1 UDP 16647679 172.29.104.128 57190 typ relay raddr 172.29.105.97 rport 50314
a=candidate: 3 2 UDP 16647678 172.29.104.128 55772 typ relay raddr 172.29.105.97 rport 50315
a=candidate: 4 1 tcp-act 7076863 172.29.104.128 58409 typ relay raddr 172.29.105.97 rport
52609
a=candidate: 4 2 tcp-act 7076350 172.29.104.128 58409 typ relay raddr 172.29.105.97 rport
52609
a=candidate: 5 1 tcp-act 1684797951 172.29.105.97 52609 typ srflx raddr 172.29.105.97 rport
52609
a=candidate: 5 2 tcp-act 1684797438 172.29.105.97 52609 typ srflx raddr 172.29.105.97 rport
52609
a=label: main-audio
a=cryptoscale: 1 client AES_CM_128_HMAC_SHA1_80
inline: U5AwqxEVWzlsWSC3ESNuqxZsyjF/8o2EVmpy3ghr|2^31|1:1
a=crypto: 2 AES_CM_128_HMAC_SHA1_80 inline: L+RBO8KufEk/IB+YwG6awdrvCdvZyW9gUYv5wks5|2^31|1:1
a=crypto: 3 AES_CM_128_HMAC_SHA1_80 inline: ijE6xzUzoeubNucTU2Q+Ti/j0EmlXBcd5U0oKA5n|2^31
a=sendonly
a=feature: MoH
a=rtpmap: 112 G7221/16000
a=fmtp: 112 bitrate=24000
a=rtpmap: 111 SIREN/16000

```
a=fmtp:111 bitrate=16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:4 G723/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:97 RED/8000
--aMHda3MG7oSOMCuhrx23ZCRDS9ibp9Vk-
```

3. Park response from CPS to parker.

```
SIP/2.0 200 OK
Authentication-Info: NTLM qop="auth", opaque="A3EEB267", srand="A7F38F5A", snum="159",
rspauth="0100000000000000b328734a6251281e", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Via: SIP/2.0/TLS 172.24.32.147:52774;ms-received-port=52774;ms-received-cid=A94F00
FROM: "Alice"<sip:alice@contoso.com>;tag=65c0e6ba25;epid=c9a8219cde
TO: "Call Park
Service"<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=srvr:Microsoft.Rtc.Applicati
ons.Cps:4H28w6olDlm9KG3nl5tyBQAA>;tag=bd5d9ee6e;epid=9FC09D8ABF
CSEQ: 1 INVITE
CALL-ID: 902cb5a0511c49f5a0b3c4c90f9c396a
RECORD-ROUTE:
<sip:callparkservice.contoso.com:5061;transport=tls;opaque=state:F:Ci.Ra94f00;lr;ms-route-
sig=gcsOHixQhEzOTEufH-s3Q1VY2YJJC3uNWKq1GPad2DmTbf3fd43ELVtwAA>
CONTACT:
<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=srvr:Microsoft.Rtc.Applications.Cps:
4H28w6olDlm9KG3nl5tyBQAA>;text;audio;video;image;+sip.rendering="no";isCps
CONTENT-LENGTH: 364
SUPPORTED: ms-dialog-route-set-update
SUPPORTED: gruu-10
SUPPORTED: ms-dialog-route-set-update
SUPPORTED: timer
SUPPORTED: 100rel
CONTENT-TYPE: application/ms-call-park+xml
ALLOW: ACK
SERVER: RTCC/4.0.0.0 Call_Park_Service_1.0
Allow: CANCEL,BYE,INVITE,PRACK,UPDATE
Session-Expires: 320;refresher=uac
Min-SE: 90
<?xml version="1.0"?><park-response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1.0" request-id="1"
xmlns="http://schemas.microsoft.com/rtc/2008/12/callpark">
  <orbit>1504</orbit>
  <ms-parked-call>3e851601-72d9-40d8-88d7-647e6ab8a7d5;from-tag=76ea4cdd28;to-
tag=2950af20b5</ms-parked-call>
</park-response>
```

4.1.2 Retrieve a Parked Call

Now, Carol wants to retrieve Bob's call from the CPS by calling the **orbit** that was verbally given to her by Alice over an intercom system, as shown in the following diagram.

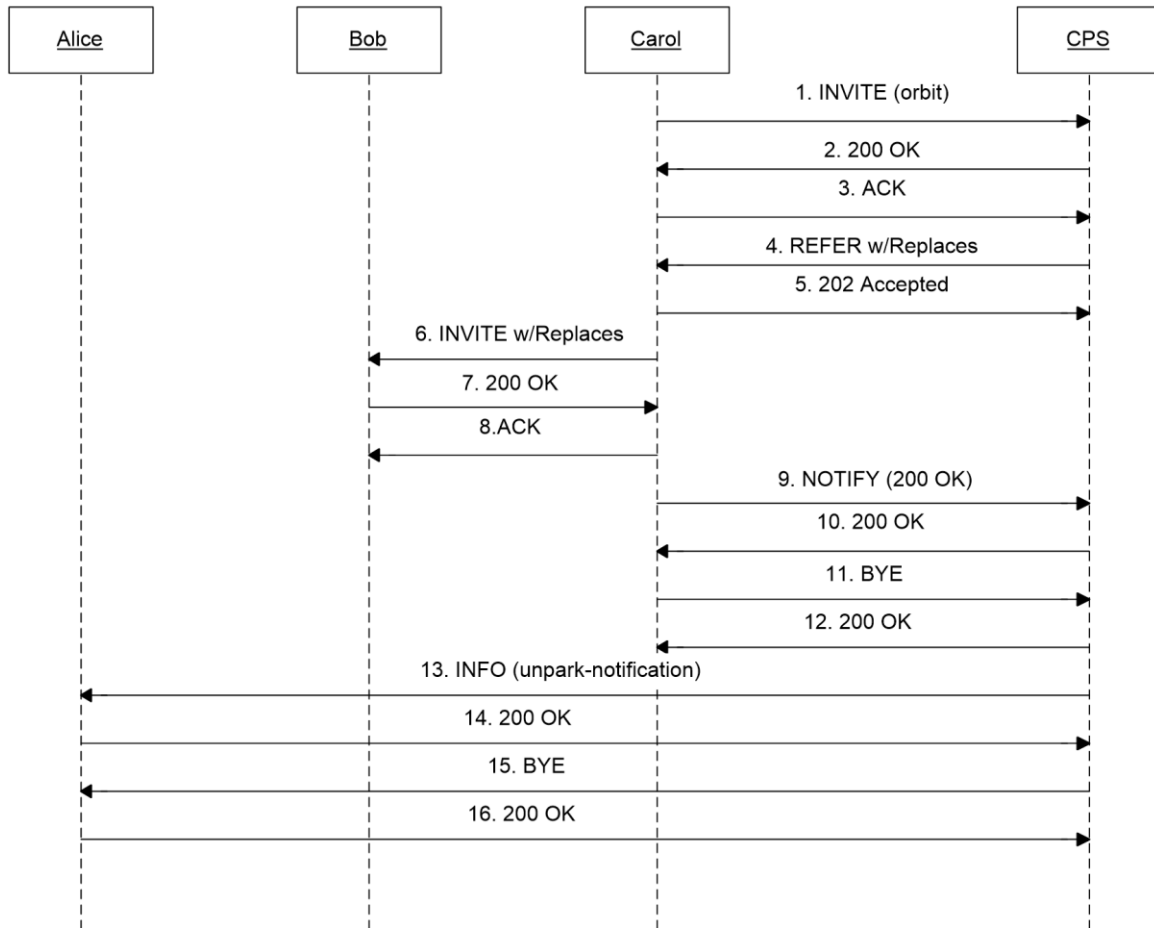


Figure 3: Call retrieval call flow

The following are the sequence of events:

1. **INVITE** from retriever to the CPS (orbit).

```

INVITE sip:1561;phone-context=defaultprofile@contoso.com;user=phone SIP/2.0
Via: SIP/2.0/TLS 172.24.32.147:53243
Max-Forwards: 70
From: <sip:carol@contoso.com>;tag=7c4136f3b6;epid=c9a8219cde
To: <sip:1561;phone-context=defaultprofile@contoso.com;user=phone>
Call-ID: a19e7ff9ec9243bb9416276afe7bc707
CSeq: 1 INVITE
Contact: <sip:carol@contoso.com;opaque=user:epid:zNEfM3iQbV6i7RUjfAv4LQAA;gruu>
User-Agent: UCCAPI/4.0.7259.501 OC/4.0.7259.501 (Microsoft Communicator 2010 (Beta))
Ms-Conversation-ID: Acqult1rcSuM8PsUQRuPSG8+8eSpXQ==
Supported: timer
Supported: histinfo
Supported: ms-safe-transfer
Supported: ms-sender
Supported: ms-early-media
Supported: 100rel
ms-keep-alive: UAC;hop-hop=yes
Allow: INVITE, BYE, ACK, CANCEL, INFO, UPDATE, REFER, NOTIFY, BENOTIFY, OPTIONS
  
```

Accept-Language: en-US
P-Preferred-Identity: <sip:carol@contoso.com>, <tel:+16045009855>
Supported: replaces
Supported: ms-conf-invite
Proxy-Authorization: NTLM qop="auth", realm="SIP Communications Service", opaque="F1C4F39A",
targetname="callparkservice.contoso.com", crand="1a5dd029", cnum="190",
response="010000032313963e2810a50c225885e"
Content-Type: multipart/alternative;boundary="-----_NextPart_000_0112_01CAAE53.CFA8DF70"
Content-Length: 4254
-----_NextPart_000_0112_01CAAE53.CFA8DF70
Content-Type: application/sdp
Content-Transfer-Encoding: 7bit
Content-ID: <b489599777aa430b9da31348efb50512>
Content-Disposition: session; handling=optional; ms-proxy-2007fallback
v=0
o=- 0 0 IN IP4 172.24.32.147
s=session
c=IN IP4 172.24.32.147
b=CT:99980
t=0 0
m=audio 1430 RTP/SAVP 114 9 112 111 0 8 116 115 4 97 13 118 101
a=candidate:7glJym+8xXYE7ln4CPtnW97wVR8b1x13R9zyw0kv6VE 1 6+KIE3erFJRpxVNobDG8uA UDP 0.830
172.24.32.147 1430
a=candidate:7glJym+8xXYE7ln4CPtnW97wVR8b1x13R9zyw0kv6VE 2 6+KIE3erFJRpxVNobDG8uA UDP 0.830
172.24.32.147 1431
a=candidate:G8WU/wakZ6vcWLH6HmGVlqW9TgT/GvwMkgbcyXKmHR4 1 7IxYkxgPD44+SOOReDn6Ug TCP 0.190
172.29.104.128 52522
a=candidate:G8WU/wakZ6vcWLH6HmGVlqW9TgT/GvwMkgbcyXKmHR4 2 7IxYkxgPD44+SOOReDn6Ug TCP 0.190
172.29.104.128 52522
a=candidate:hXdZC/YfrrZ7gB1mBbhOvbCobZ/5uB+HPL2u9LfpPF1Q 1 8TReKb/S6hKU9rTJwOJnUw UDP 0.490
172.29.104.128 57440
a=candidate:hXdZC/YfrrZ7gB1mBbhOvbCobZ/5uB+HPL2u9LfpPF1Q 2 8TReKb/S6hKU9rTJwOJnUw UDP 0.490
172.29.104.128 57209
a=candidate:nc60zTy88o3N6IeFZUQqKxm11Lfe1uzgs9Cb6FotTGU 1 izGbS0hUf71WPrHeNbomoA TCP 0.250
172.24.32.147 12113
a=candidate:nc60zTy88o3N6IeFZUQqKxm11Lfe1uzgs9Cb6FotTGU 2 izGbS0hUf71WPrHeNbomoA TCP 0.250
172.24.32.147 12113
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:R6tc6M3JBw+FSAAv28BjGaY58SA/Sby+3VC5vEg|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:SC+2HWH1PpzL+AFRHPQUOh87XJLuWqGfMyBD2PEX|2^31|1:1
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:KO7nWgLW+1kiIcbtrxHhKs0FySzQvwKzAUIeCxPP|2^31
a=maxptime:200
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=fmtp:9 bitrate=64000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:4 G723/8000
a=rtpmap:97 RED/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:required
-----_NextPart_000_0112_01CAAE53.CFA8DF70
Content-Type: application/sdp
Content-Transfer-Encoding: 7bit
Content-ID: <13540bd27847415a80d642719d10b62e>
Content-Disposition: session; handling=optional
v=0
o=- 0 0 IN IP4 172.24.32.147

```

s=session
c=IN IP4 172.24.32.147
b=CT:99980
t=0 0
m=audio 3464 RTP/SAVP 114 9 112 111 0 8 116 115 4 97 13 118 101
a=ice-ufraq:9PRU
a=ice-pwd:L55ePii649rP6SFRT15b8aSx
a=candidate:1 1 UDP 2130706431 172.24.32.147 3464 typ host
a=candidate:1 2 UDP 2130705918 172.24.32.147 3465 typ host
a=candidate:2 1 TCP-PASS 6556159 172.29.104.128 54033 typ relay raddr 172.24.32.147 rport
12130
a=candidate:2 2 TCP-PASS 6556158 172.29.104.128 54033 typ relay raddr 172.24.32.147 rport
12130
a=candidate:3 1 UDP 16648703 172.29.104.128 51922 typ relay raddr 172.24.32.147 rport 12194
a=candidate:3 2 UDP 16648702 172.29.104.128 57455 typ relay raddr 172.24.32.147 rport 12195
a=candidate:4 1 TCP-ACT 7076863 172.29.104.128 54033 typ relay raddr 172.24.32.147 rport
12130
a=candidate:4 2 TCP-ACT 7076350 172.29.104.128 54033 typ relay raddr 172.24.32.147 rport
12130
a=candidate:5 1 TCP-ACT 1684797951 172.24.32.147 12130 typ srflx raddr 172.24.32.147 rport
12130
a=candidate:5 2 TCP-ACT 1684797438 172.24.32.147 12130 typ srflx raddr 172.24.32.147 rport
12130
a=cryptoscale:1 client AES_CM_128_HMAC_SHA1_80
inline:R6tc6M3JBw+FSAAaV28BjGaY58SA/Sby+3VC5vEg|2^31|1:1
a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:SC+2HWH1PpzL+AFRHPQUOh87XJLuWqGfMyBD2PEX|2^31|1:1
a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:KO7nWgLW+1kiIcbrxHhKs0FySzQvwKzAUIeCxPP|2^31
a=maxptime:200
a=rtpmap:114 x-msrta/16000
a=fmtp:114 bitrate=29000
a=rtpmap:9 G722/8000
a=fmtp:9 bitrate=64000
a=rtpmap:112 G7221/16000
a=fmtp:112 bitrate=24000
a=rtpmap:111 SIREN/16000
a=fmtp:111 bitrate=16000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:116 AAL2-G726-32/8000
a=rtpmap:115 x-msrta/8000
a=fmtp:115 bitrate=11800
a=rtpmap:4 G723/8000
a=rtpmap:97 RED/8000
a=rtpmap:13 CN/8000
a=rtpmap:118 CN/16000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=encryption:required
-----_NextPart_000_0112_01CAAE53.CFA8DF70--

```

2. 200 OK response from the CPS to the retriever.

```

SIP/2.0 200 OK
Authentication-Info: NTLM qop="auth", opaque="F1c4F39A", srand="A9E7FBEB", snum="289",
rspauth="010000000000000027f917b4c225885e", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Via: SIP/2.0/TLS 172.24.32.147:53243;ms-received-port=53243;ms-received-cid=93AD00
FROM: "Carol"<sip:carol@contoso.com>;tag=7c4136f3b6;epid=c9a8219cde
TO: "Call Park Service"<sip:1561;phone-
context=defaultprofile@contoso.com;user=phone>;tag=3840f02e1;epid=9FC09D8ABF
CSEQ: 1 INVITE
CALL-ID: a19e7ff9ec9243bb9416276afe7bc707
RECORD-ROUTE:
<sip:callparkservice.contoso.com:5061;transport=tls;lr>,<sip:callparkservice.contoso.com:5061
;transport=tls;opaque=state:F:ci.R93ad00;lr;ms-route-
sig=fcFKiRGz_gyy36K7L9GluERFkSKBfs9OrDO_37HbOAKs12aC873ELVtWAA>

```

CONTACT:
 <sip:callparkservice.contoso.com@contoso.com;gruu;opaque=svr:Microsoft.Rtc.Applications.Cps:4H28w6o1Dlm9KG3n15tyBQAA>;text;audio;video;image;+sip.rendering="no";isCps
 CONTENT-LENGTH: 1600
 PRIORITY: Normal
 SUPPORTED: Replaces
 SUPPORTED: ms-dialog-route-set-update
 SUPPORTED: gruu-10
 SUPPORTED: ms-dialog-route-set-update
 SUPPORTED: timer
 SUPPORTED: 100rel
 CONTENT-TYPE: application/sdp
 ALLOW: ACK
 P-ASSERTED-IDENTITY: "Call Park Service"<sip:RtcApplication-49804d56-9297-4f58-be0a-0872df68b817@contoso.com>
 SERVER: RTCC/4.0.0.0 Call_Park_Service_1.0
 Content-ID: 87154620-82d8-450d-8ef4-6abbb3d8286c
 Ms-Conversation-ID: 2a3c6c9a81344c02afac4134a4c4ald5
 test: ValueType
 Allow: CANCEL,BYE,INVITE,MESSAGE,INFO,SERVICE,OPTIONS,BENOTIFY,NOTIFY,PRACK,UPDATE
 Session-Expires: 320;refresher=uac
 Min-SE: 90
 v=0
 o=- 92 0 IN IP4 172.29.105.97
 s=session
 c=IN IP4 172.29.105.97
 b=CT:100000
 t=0 0
 m=audio 49880 RTP/SAVP 112 111 0 8 116 4 13 118 97 101
 c=IN IP4 172.29.105.97
 a=rtcp:49881
 a=ice-ufrag:jlXc
 a=ice-pwd:7Jp4lSce/OsXrTRZXv54hlwl
 a=candidate:1 1 UDP 2130706431 172.29.105.97 49880 typ host
 a=candidate:1 2 UDP 2130705918 172.29.105.97 49881 typ host
 a=candidate:2 1 tcp-pass 6555135 172.29.104.128 57627 typ relay raddr 172.29.105.97 rport 49344
 a=candidate:2 2 tcp-pass 6555134 172.29.104.128 57627 typ relay raddr 172.29.105.97 rport 49344
 a=candidate:3 1 UDP 16647679 172.29.104.128 53203 typ relay raddr 172.29.105.97 rport 49394
 a=candidate:3 2 UDP 16647678 172.29.104.128 58574 typ relay raddr 172.29.105.97 rport 49395
 a=candidate:4 1 tcp-act 7076863 172.29.104.128 57627 typ relay raddr 172.29.105.97 rport 49344
 a=candidate:4 2 tcp-act 7076350 172.29.104.128 57627 typ relay raddr 172.29.105.97 rport 49344
 a=candidate:5 1 tcp-act 1684797951 172.29.105.97 49344 typ srflx raddr 172.29.105.97 rport 49344
 a=candidate:5 2 tcp-act 1684797438 172.29.105.97 49344 typ srflx raddr 172.29.105.97 rport 49344
 a=label:main-audio
 a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:yXC7Fbiponz6NoAhfxgWEWRNmBxqBp23KK6kUxfe|2^31|1:1
 a=inactive
 a=rtpmap:112 G7221/16000
 a=fmtp:112 bitrate=24000
 a=rtpmap:111 SIREN/16000
 a=fmtp:111 bitrate=16000
 a=rtpmap:0 PCMU/8000
 a=rtpmap:8 PCMA/8000
 a=rtpmap:116 AAL2-G726-32/8000
 a=rtpmap:4 G723/8000
 a=rtpmap:13 CN/8000
 a=rtpmap:118 CN/16000
 a=rtpmap:97 RED/8000
 a=rtpmap:101 telephone-event/8000
 a=fmtp:101 0-16,36
 a=encryption:rejected

3. REFER with Replaces from the CPS to the retriever.

REFER sip:172.24.32.147:53243;transport=tls;ms-opaque=9ef8a0c1a7;ms-received-cid=93AD00;grid
SIP/2.0
Record-Route:
<sip:callparkservice.contoso.com:5061;transport=tls;opaque=state:F:Ci.R93ad00;lr;ms-route-
sig=fc33YHhnWIrN64ChTBmCxcvUU0eGVhjM2iafkHMCC8oPZATot9h3ELVtwAA>;ms-
rrsig=fcwEOV0eT3SysJWhlv2ihhAzRNg2Cm nucYepdBTAaOZsTot9h3ELVtwAA;tag=03DDFB3C955DC1EEFA829A26
150FDBAD
Via: SIP/2.0/TLS
172.29.105.97:5061;branch=z9hG4bK4763AD94.61DFA213A9538728;branched=FALSE;ms-internal-
info="dpCKv2Ef778Hz2DmLzumqqt2Q0mI2U0VVM5E9MiQY6rSstot9hsux72QAA"
Authentication-Info: NTLM qop="auth", opaque="F1C4F39A", srand="4B931040", snum="290",
rspauth="0100000065003e00a56b0937c225885e", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Max-Forwards: 68
Via: SIP/2.0/TLS
172.29.105.97:49715;branch=z9hG4bKABCD3AAA.C6B4B18CA9538728;branched=FALSE;ms-received-
port=49715;ms-received-cid=93B800
Record-Route:
<sip:callparkservice.contoso.com:5061;transport=tls;lr>;tag=03DDFB3C955DC1EEFA829A26150FDBAD
Via: SIP/2.0/TLS 172.29.105.97:52500;branch=z9hG4bK86f04a78;ms-received-port=52500;ms-
received-cid=971600
FROM: <sip:1561;phone-
context=defaultprofile@contoso.com;user=phone>;epid=9FC09D8ABF;tag=3840f02e1
TO: <sip:carol@contoso.com>;epid=c9a8219cde;tag=7c4136f3b6
CSEQ: 1 REFER
CALL-ID: a19e7ff9ec9243bb9416276afe7bc707
CONTACT:
<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=svr:Microsoft.Rtc.Applications.Cps:
4H28w6olDlm9KG3n15tyBQAA>;text;audio;video;image;+sip.rendering="no";isCps
CONTENT-LENGTH: 0
EXPIRES: 600
REFER-TO:
<sip:bob@contoso.com;opaque=user:epid:NbwN8dbn8nluZbHvDCgJh0AAA;gruu?REPLACES=3734e209-9e6e-
455e-8986-aclacb135d0d%3Bfrom-tag%3D89e629df39%3Bto-tag%3D945de20b6b>
REFERRED-BY: <sip:RtcApplication-49804d56-9297-4f58-be0a-0872df68b817@contoso.com>
SUPPORTED: ms-dialog-route-set-update
SUPPORTED: gruu-10
USER-AGENT: RTCC/4.0.0.0 Call_Park_Service_1.0
P-ASSERTED-IDENTITY: "Call Park Service"<sip:RtcApplication-49804d56-9297-4f58-be0a-
0872df68b817@contoso.com>
REPLACES: 3734e209-9e6e-455e-8986-aclacb135d0d;from-tag=89e629df39;to-tag=945de20b6b

4. Unpark-notification from the CPS to the parker.

INFO sip:172.24.32.147:53243;transport=tls;ms-opaque=9ef8a0c1a7;ms-received-cid=93AD00;grid
SIP/2.0
Via: SIP/2.0/TLS
172.29.105.97:5061;branch=z9hG4bK6C9F3A20.EF80794F7CF82A8A;branched=FALSE;ms-internal-
info="aqmh5h6vyOWEfbB4yyo 5KvxKjV6QD8Cy0DZdBXFOjZ1RPeYDvsux72QAA"
Authentication-Info: NTLM qop="auth", opaque="F1C4F39A", srand="7717EE53", snum="368",
rspauth="01000000000000027da1308c225885e", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Max-Forwards: 69
Via: SIP/2.0/TLS 172.29.105.97:53098;branch=z9hG4bKcbf4261;ms-received-port=53098;ms-
received-cid=988A00
FROM:
<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=svr:Microsoft.Rtc.Applications.Cps:
4H28w6olDlm9KG3n15tyBQAA>;epid=9FC09D8ABF;tag=658f588fa7
TO: <sip:alice@contoso.com>;epid=c9a8219cde;tag=ceb4ba7ad8
CSEQ: 1 INFO
CALL-ID: fc4419ac7c694c57b5d6a9b457f054a9
CONTENT-LENGTH: 352
SUPPORTED: ms-dialog-route-set-update
USER-AGENT: RTCC/4.0.0.0 Call_Park_Service_1.0
CONTENT-TYPE: application/ms-call-park+xml

```
<?xml version="1.0"?><unpark-notification xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1.0" request-id="1"
xmlns="http://schemas.microsoft.com/rtc/2008/12/callpark">
  <reason>retrieval</reason>
  <target>sip:carol@contoso.com</target></unpark-notification>
```

4.1.3 Failure to Park a Call

In this example, Alice and Bob are already in an audio conversation. Alice attempts to park Bob at the **CPS**, but the operation fails because there were no free orbits in the pool. The **INVITE** is sent as illustrated in section 4.1.1. The SIP 500 failure response is as shown in the following code example. The **ms-diagnostics** provides the client with the specific reason that there are no free orbits left in the pool. The following diagram shows the scenario.

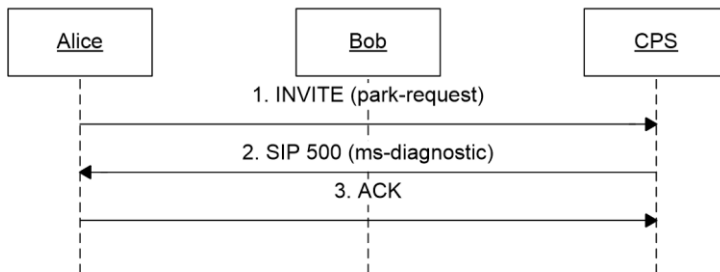


Figure 4: Call park failure call flow

```
SIP/2.0 500 Not Acceptable Here
Authentication-Info: NTLM qop="auth", opaque="F1C4F39A", srand="39318B7B", snum="73",
rspauth="01000000636f72703502ba47c225885e", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Via: SIP/2.0/TLS 172.24.32.147:53243;ms-received-port=53243;ms-received-cid=93AD00
FROM: "Alice"<sip:alice@contoso.com>;tag=3140dd2ac8;epid=c9a8219cde
TO: "Call Park
Service"<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=srvr:Microsoft.Rtc.Applicati
ons.Cps:4H28w6olDlm9KG3n15tyBQAA>;tag=8eb9b1841;epid=9FC09D8ABF
CSEQ: 1 INVITE
CALL-ID: 5ada44a9ad874ceea6a8bba545b72eec
CONTENT-LENGTH: 0
SERVER: RTCC/4.0.0.0 Call_Park_Service_1.0
ms-diagnostics-public: 35002;reason=" No more available orbits"
```

4.1.4 Failure to Retrieve a Parked Call

In this example, Carol dials the **orbit** that she was given by Alice (not depicted). The retrieve operation fails because Bob has already hung up. The **INVITE** is sent as illustrated in section 4.1.2. The failure SIP 404 response is as shown in the following diagram. The **ms-diagnostics** provides the client with the specific reason that no call was found at the orbit that was dialed. The following diagram shows the scenario.

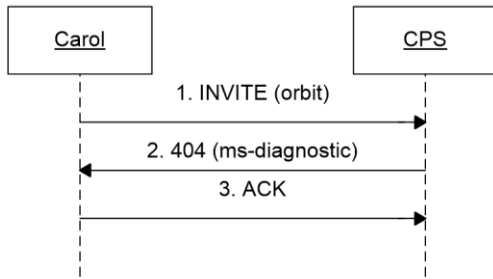


Figure 5: Call retrieve failure call flow

```

SIP/2.0 404 Not Found
Authentication-Info: NTLM qop="auth", opaque="F1C4F39A", srand="4462AD0E", snum="122",
rspauth="010000000000000022721bblc225885e", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Via: SIP/2.0/TLS 172.24.32.147:53243;ms-received-port=53243;ms-received-cid=93AD00
FROM: "Alice"<sip:alice@contoso.com>;tag=8c3c64ab12;epid=c9a8219cde
TO: "Call Park Service"<sip:1610;phone-
context=defaultprofile@contoso.com;user=phone>;tag=6bf9148926;epid=9FC09D8ABF
CSEQ: 1 INVITE
CALL-ID: a00e8b60829649bc8e4803f55202bfbe
CONTENT-LENGTH: 0
PRIORITY: Normal
SUPPORTED: Replaces
P-ASSERTED-IDENTITY: "Call Park Service"<sip:RtcApplication-49804d56-9297-4f58-be0a-
0872df68b817@contoso.com>
SERVER: RTCC/4.0.0.0 Call_Park_Service_1.0
ms-diagnostics-public: 35000;reason="Orbit not found"Ms-Conversation-ID:
2a3c6c9a81344c02afac4134a4c4a1d5
  
```

4.1.5 Auto-Ringback Is Answered by the Parker

In this example, Bob's call has been parked at the server for too long, so the **CPS** initiates an **auto-ringback**, sending it back to the user who parked it (Alice). Alice answers this incoming call. The following diagram shows the scenario.

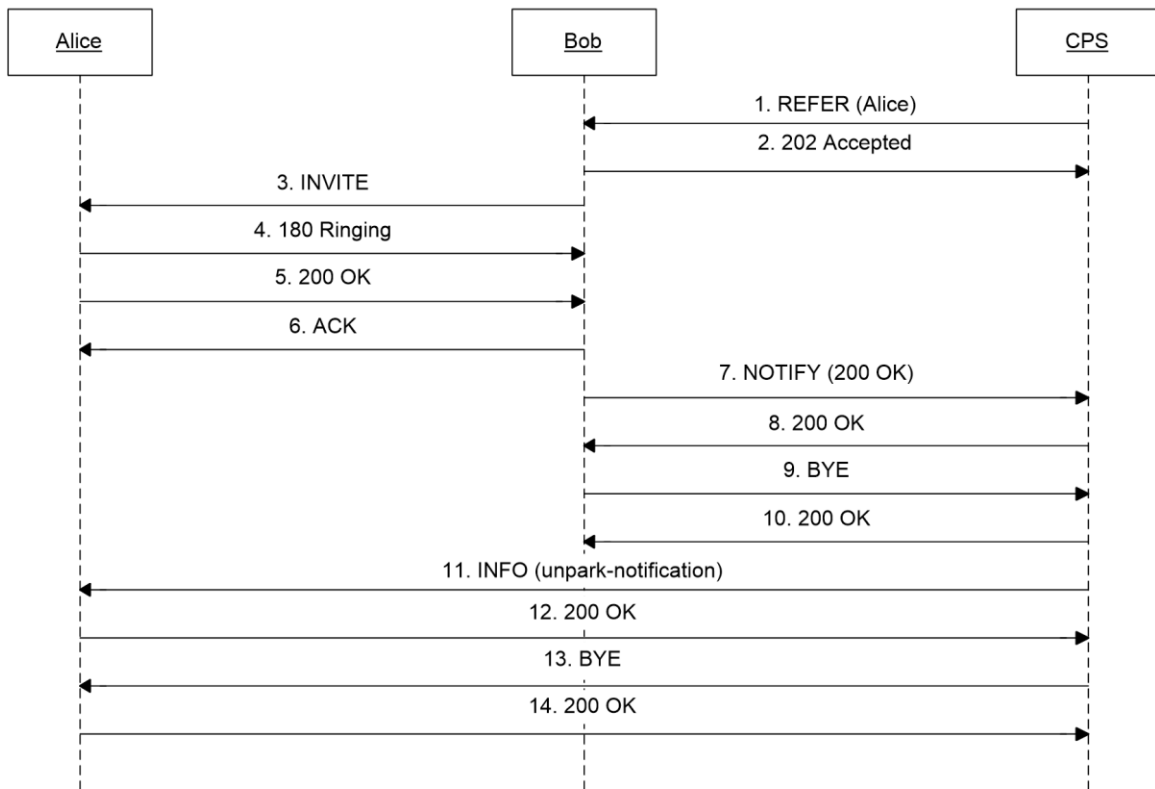


Figure 6: Auto-ringback call flow

The following is the sequence of events.

1. The CPS refers Bob back to the original parker (Alice). The **Refer-To** header contains an escaped **ms-sensitivity** header that will prevent Alice's voicemail from automatically answering this call.

```

REFER sip:172.24.33.241:51780;transport=tls;ms-opaque=1d722d38d1;ms-received-cid=93C000;grid
SIP/2.0
Record-Route:
<sip:callparkservice.contoso.com:5061;transport=tls;opaque=state:T:F:Ci.R93c000:Ieh.wPN9F9YaC
xPXGgsTfcNiXMO1va0pg9kGHtbWO5uwPB8XfNpm_6TLQtNrQu71MippNy3ELVtwAA;lr;ms-route-sig=fcDnk7eC6-
ioeJreYCrFx4VneIRDKeXlDnKyxVQ6UJ46UippNy3ELVtwAA>;tag=03DDFB3C955DC1EEFA829A26150FDBAD
Via: SIP/2.0/TLS
172.29.105.97:5061;branch=z9hG4bKB4652A68.7293A62227ECDD47;branched=FALSE;ms-internal-
info="aqDW1SFf1td8hYZ460ltTFqqPWqp-nQpCukZOY1vh4NzwippNysux72QAA"
Authentication-Info: TLS-DSK qop="auth", opaque="6C58FF34", srand="472D422C", snum="485",
rspauth="5ccd8289853502c316cb46d3474ac238216a49cc", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Max-Forwards: 69
Via: SIP/2.0/TLS 172.29.105.97:53551;branch=z9hG4bK7c84ac40;ms-received-port=53551;ms-
received-cid=99A000
FROM: <sip:RtcApplication-49804d56-9297-4f58-be0a-
0872df68b817@contoso.com>;epid=9FC09D8ABF;tag=a07dbb9f2d
  
```



```

TO:
<sip:bob@contoso.com;opaque=user:epid:NbwN8db8nluZbHVdCgJh0AAA;gruu>;epid=38e610bbd5;tag=63ad
71d932
CSEQ: 116 REFER
CALL-ID: 4b3604e4-7130-4608-86ad-d2e5718b4626
CONTACT:
<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=svr:Microsoft.Rtc.Applications.Cps:
4H28w6olDlm9KG3n15tyBQAA>;text;audio;video;image;+sip.rendering="no";isCps
CONTENT-LENGTH: 0
EXPIRES: 600
REFER-TO: <sip:alice@contoso.com?ms-sensitivity=private-no-diversion>
REFERRED-BY: <sip:RtcApplication-49804d56-9297-4f58-be0a-0872df68b817@contoso.com>;ms-
referee-uri="sip:bob@contoso.com"
SUPPORTED: ms-dialog-route-set-update
SUPPORTED: gruu-10
USER-AGENT: RTCC/4.0.0.0 Call_Park_Service_1.0
P-ASSERTED-IDENTITY: "Call Park Service"<sip:RtcApplication-49804d56-9297-4f58-be0a-
0872df68b817@contoso.com>

```

2. After the auto-ringback has been successfully answered, the CPS sends an **unpark-notification** to the original parker (Alice).

```

INFO sip:172.24.32.147:53243;transport=tls;ms-opaque=9ef8a0c1a7;ms-received-cid=93AD00;grid
SIP/2.0
Via: SIP/2.0/TLS
172.29.105.97:5061;branch=z9hG4bK6CD71B75.EE6B06F2293ACD4C;branched=FALSE;ms-internal-
info="aqdRZF_shkBibefV4cKrf5cznHs5Rmwa30urp2KWm3hRzyBmvusux72QAA"
Authentication-Info: NTLM qop="auth", opaque="F1C4F39A", srnd="8F87CCF5", snum="500",
rspauth="0100000000000000beab4f88c225885e", targetname="callparkservice.contoso.com",
realm="SIP Communications Service", version=4
Max-Forwards: 69
Via: SIP/2.0/TLS 172.29.105.97:53594;branch=z9hG4bK199fe9bf;ms-received-port=53594;ms-
received-cid=99AA00
FROM:
<sip:callparkservice.contoso.com@contoso.com;gruu;opaque=svr:Microsoft.Rtc.Applications.Cps:
4H28w6olDlm9KG3n15tyBQAA>;epid=9FC09D8ABF;tag=79b542798
TO: <sip:alice@contoso.com>;epid=c9a8219cde;tag=972664809d
CSEQ: 1 INFO
CALL-ID: 173c5a778fc54a458a82f44f6acbd276
CONTENT-LENGTH: 275
SUPPORTED: ms-dialog-route-set-update
USER-AGENT: RTCC/4.0.0.0 Call Park Service 1.0
CONTENT-TYPE: application/ms-call-park+xml
<?xml version="1.0"?><unpark-notification xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1.0" request-id="1"
xmlns="http://schemas.microsoft.com/rtc/2008/12/callpark">
  <reason>ringback</reason></unpark-notification>

```

The call flow for the case where a parked call is transferred to the Fallback is similar, but the unpark-notification will show "fallback" instead of "ringback" as the reason. Additionally, there would be a target element in the unpark notification that is set to the address-of-record of the fallback user.

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full XML Schema

The following sections give the **XML schemas** (if applicable) for the protocols defined in this document.

6.1 ms-call-park XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  targetNamespace="http://schemas.microsoft.com/rtc/2008/12/callpark"
  xmlns:tns="http://schemas.microsoft.com/rtc/2008/12/callpark"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  version="1.0">

  <!-- ALLOWED ROOT ELEMENTS -->
  <xs:element name="park-request" type="tns:park-request-type"/>
  <xs:element name="park-response" type="tns:park-response-type"/>
  <xs:element name="unpark-notification" type="tns:unpark-notification-type"/>

  <!-- PARK REQUEST TYPE -->
  <xs:complexType name="park-request-type">
    <xs:sequence>
      <xs:element name="audio" type="tns:modality-park-request-type"/>
      <xs:any namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
    <xs:attribute name="request-id" type="xs:string" use="required"/>
  </xs:complexType>

  <!-- MODALITY PARK REQUEST TYPE -->
  <xs:complexType name="modality-park-request-type">
    <xs:sequence>
      <xs:element name="dialog-info" type="tns:dialog-info-type"/>
      <xs:element name="target" type="xs:anyURI"/>
    </xs:sequence>
  </xs:complexType>

  <!-- DIALOG INFO TYPE -->
  <xs:complexType name="dialog-info-type">
    <xs:sequence>
      <xs:element name="call-id" type="xs:string"/>
      <xs:element name="from-tag" type="xs:string"/>
      <xs:element name="to-tag" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <!-- PARK RESPONSE TYPE -->
  <xs:complexType name="park-response-type">
    <xs:sequence>
      <xs:element name="orbit" type="xs:string" />
      <xs:element name="ms-parked-call" type="xs:string" />
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
    <xs:attribute name="request-id" type="xs:string" use="required"/>
  </xs:complexType>

  <!-- UNPARK NOTIFICATION TYPE -->
  <xs:complexType name="unpark-notification-type">
    <xs:sequence>
      <xs:element name="reason" type="tns:unpark-reason-type"/>
      <xs:element name="target" type="xs:anyURI" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required"/>
    <xs:attribute name="request-id" type="xs:string" use="required"/>
  </xs:complexType>
```

```
<!-- UNPARK REASON TYPE -->
<xs:simpleType name="unpark-reason-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="retrieval"/>
    <xs:enumeration value="hang-up"/>
    <xs:enumeration value="ringback"/>
    <xs:enumeration value="fallback"/>
    <xs:enumeration value="drop"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Lync Server 2010
- Microsoft Lync 2010
- Microsoft Lync Server 2013
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Skype for Business 2016
- Microsoft Skype for Business Server 2015
- Microsoft Skype for Business 2019
- Microsoft Skype for Business Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

[<1> Section 2.1.1](#): Lync 2010, Lync Server 2010: IPv6 is not supported.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
Z Appendix B: Product Behavior	Updated list of supported products.	Major
7 Appendix B: Product Behavior	Updated list of supported products.	major

9 Index

A

[Abstract data model](#) 15
[Applicability](#) 9
Auto-ringback answered by parker
 [example](#) 31

C

[Capability negotiation](#) 9
 [ms-call-park](#) 9
[Change tracking](#) 38

D

[Data model - abstract](#) 15

E

[Examples](#) 20
 [auto-ringback answered by parker](#) 31
 [failure to park a call](#) 30
 [Failure to retrieve a parked call](#) 30
 [park a call](#) 20
 [retrieve a parked call](#) 24

F

Failure to park a call
 [example](#) 30
Failure to retrieve a parked call
 [example](#) 30
[Fields - vendor-extensible](#) 9
[Full XML schema](#) 35

G

[Glossary](#) 6

H

[Higher-layer triggered events](#) 15

I

[Implementer - security considerations](#) 34
[Index of security parameters](#) 34
[Informative references](#) 8
[Initialization](#) 15
[Introduction](#) 6

L

[Local events](#) 19

M

[Message processing](#) 15
 [client request to park a call](#) 15
 [CPS fails to replace parkee call](#) 16

[CPS receives BYE from parker](#) 16
 [CPS receives BYE on audio dialog](#) 16
 [CPS receives INVITE](#) 17
 [CPS replaces parkee call](#) 16
 [CPS transfer of parkee to fallback fails](#) 18
 [CPS transfer of parkee to parker fails](#) 18
 [CPS transfer of retriever fails](#) 17
 [CPS transfers parkee to fallback](#) 18
 [CPS transfers parkee to parker](#) 18
 [CPS transfers retriever](#) 17
 [Message Syntax for ms-call-park message](#) 10
 [park request](#) 10
 [park response](#) 11
 [unpark notification](#) 12

Messages

[Message Syntax for ms-call-park](#) 10
 [park request](#) 10
 [park response](#) 11
 [unpark notification](#) 12
 [transport](#) 10
 [ms-call-park](#) 10

ms-call-park

[abstract data model](#) 15
 [capability negotiation](#) 9
 [examples](#) 20
 [auto-ringback answered by parker](#) 31
 [failure to park a call](#) 30
 [Failure to retrieve a parked call](#) 30
 [park a call](#) 20
 [retrieve a parked call](#) 24
 [higher-layer triggered events](#) 15
 [initialization](#) 15
 [local events](#) 19
 [message syntax](#) 10
 [park request](#) 10
 [park response](#) 11
 [unpark notification](#) 12
 [message transport](#) 10
 [overview](#) 14
 [schema](#) 35
 [timer events](#) 19
 [timers](#) 15
 [versioning](#) 9

N

[Normative references](#) 7

O

[Overview](#) 14
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 34
Park a call
 [example](#) 20
[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 37

R

[References](#) 7
 [informative](#) 8
 [normative](#) 7
[Relationship to other protocols](#) 8
Retrieve a parked call
 [example](#) 24

S

Schemas
 [ms-call-park](#) 35
Security
 [implementer considerations](#) 34
 [parameter index](#) 34
[Sequencing rules](#) 15
 [client request to park a call](#) 15
 [CPS fails to replace parkee call](#) 16
 [CPS receives BYE from parker](#) 16
 [CPS receives BYE on audio dialog](#) 16
 [CPS receives INVITE](#) 17
 [CPS replaces parkee call](#) 16
 [CPS transfer of parkee to fallback fails](#) 18
 [CPS transfer of parkee to parker fails](#) 18
 [CPS transfer of retriever fails](#) 17
 [CPS transfers parkee to fallback](#) 18
 [CPS transfers parkee to parker](#) 18
 [CPS transfers retriever](#) 17
[Standards assignments](#) 9

T

[Timer events](#) 19
[Timers](#) 15
[Tracking changes](#) 38
[Transport](#) 10
 [ms-call-park](#) 10
[Triggered events](#) 15

V

[Vendor-extensible fields](#) 9
[Versioning](#) 9
 [ms-call-park](#) 9

X

[XML schema](#) 35