CloudProxy:1 Service

For UPnP Version 2.0

Status: Standardized DCP (SDCP)

Date: December 31, 2015

Document Version: 1.0

Service Template Version: 2.00

Authors [a]	**Company**
Clarke Stevens	Cablelabs
Wouter van der Beek (Chair)	Cisco
Bich Nguyen	GoPro
Keith Miller	BKM Systems Group, LLC

[a] The UPnP forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.

CONTENTS

1 SCOPE

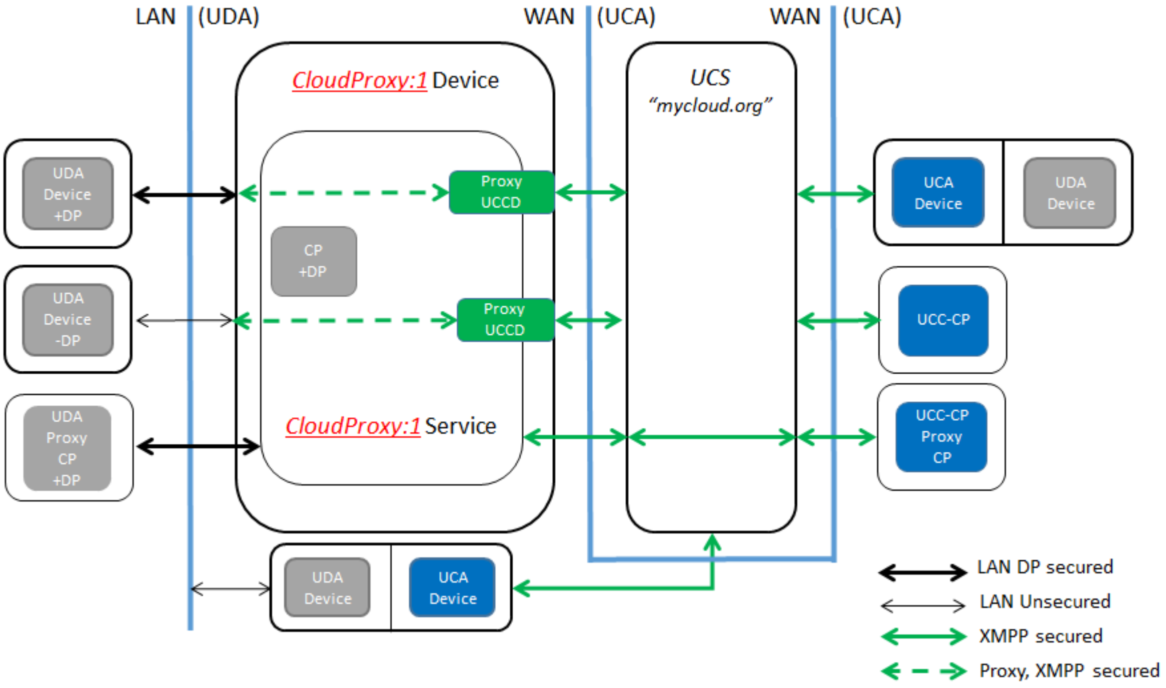
1.1 INTRODUCTION

This document defines the service CloudProxy:1, which identifies Version 1 of the service named CloudProxy:1. This Publicly Available Specification is applicable to Standardized DCPs of the UPnP Forum which include this service.

The CloudProxy service provides a technical means for legacy devices supporting only a UDA or local network interface to connect to the *Cloud*, that is to UCA UCCDs and UCC-CPs via an intermediate device hosting the CloudProxy:1 service. For version 1, the intermediate device shall be a CloudProxy:1 device [PROXY]. This is illustrated in Figure 1-1 where two legacy devices, one with DeviceProtection and one without, are "proxied" to the *Cloud* via the CloudProxy:1 service on the CloudProxy:1 device. The CloudProxy service can, via an embedded control point, discovery UDA devices, establish DeviceProtection secured *User* and control point *Roles*, subscribe to device events and forward UCC-CP received actions to local devices. The Proxy UCCD of the local device is a virtual device created by the CloudProxy service based on the DDD and SCPDs on the local device obtained with the embedded control point and connected to a *Cloud Account* configured via the CloudProxy control point; this *Cloud Account* could be pre-configured. It is also possible that the UCS is implemented on the same device hosting the CloudProxy service.

When *Cloud Accounts* are configured for the CloudProxy service, DeviceProtection *Roles* are created (and deleted) to mirror those *Cloud Account* access privileges for the CloudProxy control point. For local devices supporting DeviceProtection, all communication links are encrypted[1]. The CloudProxy device itself may be exposed on the UCA interface by the CloudProxy service but by default it is turned off.

Figure 1-1 CloudProxy Service Overview



[1] Note that the protection is not end-to-end.

This service definition is compliant with the UPnP Device Architecture, version 2.0.

2 NORMATIVE REFERENCES

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[UDA] UPnP Device Architecture, version 2.0, UPnP Forum, February 20, 2015. Available at: http://upnp.org/specs/arch/UPnPDA10_20000613.pdf. Latest version available at: http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v2.0.pdf.

[ISO_8601] ISO 8601 Data elements and interchange formats – Information interchange -- Representation of dates and times, International Standards Organization, December 21, 2000. Available at: http://www.iso.org (ISO 8601:2004).

[CDS4] UPnP Content Directory Service, version 4.0, UPnP Forum, June 30, 2015. Available at http://www.upnp.org/specs/av/ContentDirectory-av-v4-Service-201150630.pdf. Latest version available at: http://www.upnp.org/specs/av/ContentDirectory-av-v4-Service.pdf.

[RFC_2119] IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, 1997. Available at: http://www.faqs.org/rfcs/rfc2119.html.

[HTTP1.1] HyperText Transport Protocol – HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, June 1999. Available at: http://www.ietf.org/rfc/rfc2616.txt.

[RFC_3339] IETF RFC 3339, Date and Time on the Internet: Timestamps, G. Klyne, Clearswift Corporation, C. Newman, Sun Microsystems, July 2002. Available at: http://www.ietf.org/rfc/rfc3339.txt.

[RFC_6122] IETF RFC 6122, Extensible Messeging and Presence Protocol, P. Saint-Andre, Cisco, March, 2011. Available at: http://www.ietf.org/rfc/rfc6122.txt.

[XML 1.0] Extensible Markup Language (XML) 1.0 (Third Edition), François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004. Available at: http://www.w3.org/TR/2004/REC-xml-20040204.

[XSD 2.0] XML Schema Part 2: Data Types, Second Edition, Paul V. Biron, Ashok Malhotra, W3C Recommendation, 28 October 2004. Available at: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028.

[PROXY] UPnP CloudProxy:1 Device, UPnP Forum July 1, 2013. Available at: http://www.upnp.org/specs/cloud/UPnP-cloud-CloudProxy-v1-Device-20151231.pdf. Latest version available at: http://www.upnp.org/specs/cloud/UPnP-cloud-CloudProxy-v1-Device.pdf.

[DP] UPnP DeviceProtection:1 Service, UPnP Forum, February 24, 2011. Available at: http://www.upnp.org/specs/gw/UPnP-gw-DeviceProtection-v1-Service-20110224.pdf. Latest version available at: http://www.upnp.org/specs/gw/UPnP-gw-DeviceProtection-v1-Service.pdf.

[CPROXY] UPnP CloudProxy:1 Service, UPnP Forum, December 31, 2015. Available at: http://www.upnp.org/specs/cloud/UPnP-cloud-CloudProxy-v1-Service-20151231.pdf. Latest version available at: http://www.upnp.org/specs/cloud/UPnP-cloud-CloudProxy-v1-Service.pdf.

[CTS] UPnP CloudTransport:1 Service, UPnP Forum, December 31, 2015. Available at: http://www.upnp.org/specs/cloud/UPnP-cloud-CloudTransport-v1-Service-20151231.pdf.

Latest version available at: http://www.upnp.org/specs/cloud/UPnP-cloud-CloudTransport-v1-Service.pdf.

[XSD_CPU] XML Schema UPnP CloudProxy Update IDs, UPnP Forum, December 31, 2015. Available at: http://www.upnp.org/schemas/cloud/cloudproxyupdate-v1-20151231.xsd. Latest version available at: http://www.upnp.org/schemas/cloud/cloudproxyupdate.xsd.

[XSD_CPDL] XML Schema UPnP CloudProxy Device List, UPnP Forum, December 31, 2015. Available at: http://www.upnp.org/schemas/cloud/devicelist-v1-20151231.xsd. Latest version available at: http://www.upnp.org/schemas/cloud/devicelist.xsd.

[XSD_CPPL] XML Schema UPnP CloudProxy Proxy List, UPnP Forum, December 31, 2015. Available at: http://www.upnp.org/schemas/cloud/proxylist-v1-20151231.xsd. Latest version available at: http://www.upnp.org/schemas/cloud/proxylist.xsd.

[XSD_CPUL] XML Schema UPnP CloudProxy UCS List, UPnP Forum, December 31, 2015. Available at: http://www.upnp.org/schemas/cloud/ucslist-v1-20151231.xsd. Latest version available at: http://www.upnp.org/schemas/cloud/ucslist.xsd.

3 TERMS, DEFINITIONS AND ABBREVIATIONS

For the purposes of this document, the terms, definitions and abbreviations given in [CDS4], [DP], [UDA], [CTS] and [PROXY] apply.

3.1 NON-RESTRICTABLE

A category of action that, when the DeviceProtection [DP] service is implemented on the device, cannot be blocked according to the presence or absence of a specific *Role* attached to a *Control Point Identity* or *User Identity*. See [CDS4] for further explanation of *Role*, *Control Point Identity* and *User Identity*.

3.2 RESTRICTABLE

A category of actions that, when the DeviceProtection [DP] service is implemented on the device, can be blocked according to the presence or absence of a specific *Role* attached to a *Control Point Identity* or *User Identity*.

4 SECURITY FEATURE

4.1 DEVICE PROTECTION

The CloudProxy Service shall be implemented on a device supporting the DeviceProtection service [DP] and the DeviceProtection service shall support the following roles:

- *Basic* – A control point with the *Basic* role can read or write specific CloudProxy tables which permit this access.
- *Public* – A control point with the *Public* role can read or write specific CloudProxy tables which permit this access.
- *Admin* – A control point with the *Admin* role can configure and manage the *UCSlist*.
- *VendorJID* – A *Cloud Account* (aka a UCS Account, UCA Account, or XMPP Account) specific *Role* established when a *<UCSList>::<UCS>::<UserName>* state variable element is added, it has the form of "User@Cloud". This *Role* shall be required to interact further with the corresponding *Cloud Account*.

And may support additional *Roles*, such as:

- *UPnPVendor* - A control point with a *Role* defined by any UPnP DCP such as *AVSuperReader* in [CDS4].

Privileges obtained by UCC-CPs during a session via *UserLogin()* actions shall remain with the UCC-CP until one of the following events happens: 1) a presence "**`unavailable`**" stanza is received by the CloudProxy service from the UCC-CP, 2) the UCC-CP invokes a *UserLogout()* action, 3) an implementation defined time-out period is reached.

4.2 RESTRICTABLE AND NON-RESTRICTABLE ACTIONS

The CloudProxy service actions defined in this specification have the *Restrictable*, *Non-Restrictable* assignments as indicated in Table 4-1.

Table 4-1: Assignment of Restrictable/Non-Restrictable Roles

Action Name	Restrictable/Non-Restrictable to Indicated Role[1]				
	Public	*Basic*	*Admin*	*UPnPVendor*	*VendorJID*
AddUCSAccount	YES	YES	NO	YES	NO
AddProxyDevice	YES	YES	YES[2]	YES	YES[2]
DeleteUCSAccount	YES	YES	YES[2]	YES	YES[2]
DeleteProxyDevice	YES	YES	YES[2]	YES	YES[2]
GetDeviceList	NO[3]	NO[3]	NO[3]	NO[3]	NO[3]
GetProxyList	YES	YES	NO	YES	NO
GetUCSList	YES	YES	NO	YES	NO

5 NOTATIONS AND CONVENTIONS

5.1 NOTATION

- Strings that are to be taken literally are enclosed in "`double quotes`".
- Words that are emphasized are printed in *italic*.
- Keywords that are defined by the UPnP Working Committee are printed using the *forum* character style.
- Keywords that are defined by the UPnP Device Architecture are printed using the **arch** character style.

[1] A YES value in the table indicates that the action shall be *Restrictable*. Since the *Security Feature* is required all values in this table are applicable for the CloudProxy service for all control points. In general a control point with only the *Role* indicated shall not have *Action level access* and shall receive an error code 606 (see UPnP Device Architecture [UDA]) in response to the action invocation. However, since control points with a *VendorJID* *Role* by default have an *Admin* *Role* as well the restriction will generate a CloudProxy service specific error code.

A NO value in the table indicates that the action shall be *Non-Restrictable*, meaning that, even though the *Security Feature* is supported, all control points shall have *Action level access* when invoking the action and shall not receive an error code 606 based on the *Security Feature*.

[2] The control point shall have the specific *VendorJID* *Role* needed for the action, otherwise the action will return a CloudProxy service specific error code.

[3] The devices available on the local network can be discovered by normal UPnP SSDP and therefore no need to restrict.

- Keywords that are defined specific to the UPnP Device Architecture Annex C are printed using **UCA** character style.
- Keywords that are defined specific to XMPP are printed using **XMPP** character style.
- A double colon delimiter, "::", signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument, parentProperty::childProperty.

5.2 DATA TYPES

This specification uses data type definitions from two different sources. The UPnP Device Architecture defined data types are used to define state variable and action argument data types UPnP Device Architecture, version 2.0 [UDA]. The XML Schema namespace is used to define property data types XML Schema Part 2: Data Types, Second Edition [XSD 2.0].

For UPnP Device Architecture defined Boolean data types, it is strongly recommended to use the value "**0**" for false, and the value "**1**" for true. The values "**true**", "**yes**", "**false**", or "**no**" may also be used but are not recommended. The values "**yes**" and "**no**" are deprecated and shall not be sent out by devices but shall be accepted on input.

For XML Schema defined Boolean data types, it is strongly recommended to use the value "*0*" for false, and the value "*1*" for true. The values "*true*", "*yes*", "*false*", or "*no*" may also be used but are not reommended. The values "*yes*" and "*no*" are deprecated and shall not be sent out by devices but shall be accepted on input.

5.3 VENDOR-DEFINED EXTENSIONS

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation shall follow the naming conventions and XML rules as specified in UPnP Device Architecture, version 2.0 [UDA], Clause 2.5, "Description: Non-standard vendor extensions".

6 SERVICE MODELLING DEFINITIONS

6.1 SERVICE TYPE

The following URN identifies a service that is compliant with this specification:

urn:schemas-upnp-org:service: CloudProxy:1

CloudProxy service is used herein to refer to this service type.

6.2 CLOUDPROXY:1 SERVICE ARCHITECTURE

The CloudProxy service provides the functionality to discovery UDA only type devices on a local network, and represent those devices on a UCA interface as UCCDs.

The service defined herein provides the following functionality:

- Discovers, via an embedded control point, local (UDA) devices, decomposes them into individual devices if they contain embedded devices (see section C.5.5 of [UDA])and maintains a list (*DeviceList* state variable) of these available devices for proxying.
- Maintains a list (*UCSList* state variable) of UCS (XMPP servers) either provided by a CloudProxy control point or through pre-configuration.
- Adds and deletes individual *Cloud Account*s to and from the UCSs available in the *UCSList*.

- Creates and deletes DeviceProtection *Role*s mirroring the addition and deletion of *Cloud Account*s. The naming of these Roles is based on the **Bare JID** of the Cloud Account.
- Provides observable state information (*DeviceList* state variable) indicating the security of the connection on the LAN side.
- Adds and deletes individual devices in the *DeviceList* to and from the *ProxyList* using an available *Cloud Account*. This includes, connecting the local device, reading the devices DDD and SCPDs, connecting as a UCA conformant XMPP client to the UCS via the indicated *Cloud Account*, and acting as a UCCD.

The CloudProxy:1 service shall only be deployed on a CloudProxy device [PROXY].

The CloudProxy device implementing the CloudProxy service shall:

- Implement a control point for discovering and interacting with devices on each local UDA interface that the device wants to proxy from.
- Be capable of acting as a UCC-CP to establish *Cloud Account* access.
- Be capable of maintaining an XMPP client connection (a UCCD) for each local device it is proxying.
- Be capable of translating local UDA UPnP SSDP Notify and ByeBye messages to XMPP **available** and **unavailable** messages for proxied devices.
- Be capable of accepting commands from UCC-CPs and converting them to SOAP actions on a UDA interface.
- Be capable of responding to UCA specific stanzas requesting DDD and SCPD documents from proxied devices.
- Be capable of decomposing devices with embedded devices into separate UCCD devices.
- Be capable of augmenting the services available on a proxied device with a CloudTransport service [CTS].

6.3 KEY CONCEPTS

Cloud Account - an XMPP account established on an XMPP server supporting UDA 2.0 Cloud Annex, that is, a UCS account.

6.3.1 CloudProxy Events

The CloudProxy service can notify its clients of state changes via the evented *CloudProxyUpdate* state variables. The *CloudProxyUpdate* state variable indicates if there is a change in any of the list type state variables - *DeviceList*, *ProxyList*, *UCSList* - and which one changed.

6.4 STATE VARIABLES

Note: For first-time reader, it may be more insightful to read the theory of operations first and then the action definitions before reading the state variable definitions.

6.4.1 State Variable Overview

Table 6-1 — State Variables

Variable Name	R/A [a]	Data Type	Allowed Value	Default Value	Eng. Units
CloudProxyUpdate	*R*	**string**	See 6.4.2		
DeviceList	*R*	**string**	See 6.4.3		
ProxyList	*R*	**string**	See 6.4.4		
UCSList	*R*	**string**	See 6.4.5		
A_ARG_TYPE_DevId	*R*	**string**	See 6.4.6		
A_ARG_TYPE_BareJID	*R*	**string**	See 6.4.7		
A_ARG_TYPE_FullJID	*R*	**string**	See 6.4.8		
A_ARG_TYPE_Port	*R*	**ui2**	See 6.4.9	"5622"	
A_ARG_TYPE_String	*R*	**string**	See 6.4.10		
Non-standard state variables implemented by a UPnP vendor go here	*X*	*TBD*	*TBD*	*TBD*	*TBD*

NOTES:

[a] For a device this column indicates whether the state variable shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

[b] CSV stands for Comma-Separated Value list. The type between brackets denotes the UPnP data type used for the elements inside the list. The CSV list concept is defined more formally in the ContentDirectory service template.

[c] See referenced subclause for conditions under which the implementation of this state variable is required.

6.4.2 State Variable *CloudProxyUpdate*

This required state variable indicates when a change has occurred in any of the list type state variables, that is, *DeviceList*, *ProxyList* or *UCSList*.

The *CloudProxyUpdate* state variable is a string that contains a *CloudProxyUpdate XML Document*. The following example shows a generalized “template” for the *CloudProxyUpdate XML Document*. The example shows the elements that need to be filled out by individual implementations in the *vendor* character style. The Schema associated with the *CloudProxyUpdate* state variable can be found at [XSD_CPU].

```
<?xml version="1.0" encoding="UTF-8"?>
<CloudProxyUpdate xmlns="urn:schemas-upnp-org:cloud:cloudproxyupdate"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:cloudproxyupdate
      http://www.upnp.org/schemas/cloud/cloudproxyupdate.xsd">
   <ProxyUpdateId> unsigned 8 byte integer value </ProxyUpdateId>
   <DeviceUpdateId unsigned 8 byte integer value </DeviceUpdateId>
   <UCSUpdateId> unsigned 8 byte integer value</UCSUpdateId>
</CloudProxyUpdate>
```

<?xml>

Allowed. Case sensitive.

<CloudProxyUpdate>

Required. <XML>. Shall include a namespace declaration for the CloudProxy CloudProxyUpdate XML Schema (“urn:schemas-upnp-org:cloud:cloudproxyupdate”). This is the base wrapper element for the state variable. Shall include one of each of the following elements:

<DeviceUpdateId>

Required. Type is **xsd:ui8**. Shall appear one time. Indicates a change has occurred in *DeviceList* state variable. The initial value shall be "0". It shall increase monotonically with each change of the *DeviceList* state variable. The value of incrementation shall be "1". It shall be persisted across reboots of the CloudProxy device. If the value reaches rollover, that is, a value greater than "18446744073709551615" then it shall reset to a value of "1".

<ProxyUpdateId>

Required. Type is **xsd:ui8**. Shall appear one time. Indicates a change has occurred in *ProxyList* state variable. The initial value shall be "0". It shall increase monotonically with each change of the ProxyList state variable. The value of incrementation shall be "1". It shall be persisted across reboots of the CloudProxy device. If the value reaches rollover, that is, a value greater than "18446744073709551615" then it shall reset to a value of "1".

<UCSUpdateId>

Required. Type is **xsd:ui8**. Shall appear once time. Indicates a change has occurred in *UCSList* state variable. The initial value shall be "0". It shall increase monotonically with each change of the UCSList state variable. The value of incrementation shall be "1". It shall be persisted across reboots of the CloudProxy device. If the value reaches rollover, that is, a value greater than "18446744073709551615" then it shall reset to a value of "1".

The *CloudProxyUpdate* state variable is subject to event moderation.

6.4.3 State Variable *DeviceList*

This required state variable contains a list of devices on the local (UDA) interface(s) that can be proxied to the *Cloud* via UCS(s) and associated *Cloud Account*s described in the *UCSList* state variable. The *DeviceList* state variable can be configured[1] to always include at least one device that being the CloudProxy device hosting the CloudProxy service.

The *DeviceList* state variable is a string that contains a *DeviceList XML Document*. The following example shows a generalized "template" for the *DeviceList XML Document*. The example shows the elements that need to be filled out by individual implementations in the *vendor* character style. The Schema associated with the *DeviceList* state variable can be found at [XSD_CPDL].

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceList xmlns="urn:schemas-upnp-org:cloud:devlist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:devlist
      http://www.upnp.org/schemas/cloud/devlist.xsd">
   <Device DevId="id unique to this service">
      <DeviceType>device type according to UDA</DeviceType>
      <DeviceProt>1|0</DeviceProt>
      <DeviceProtStatus>1|0</DeviceProtStatus>
   </Device>
</DeviceList>
```

<?xml>

Allowed. Case sensitive.

<DeviceList>

Required. Shall include a namespace declaration for the CloudProxy DeviceList XML Schema. ("urn:schemas-upnp-org:cloud:devicelist"). This is the base wrapper element for the state variable. Shall include zero[2] or more of the following element:

<Device>

Conditionally Required. <XML>. Identifies a UPnP device that has been discovered on a network that the CloudProxy service has access to or the CloudProxy device itself. If the discovered device has embedded devices then the CloudProxy service shall represent each device as a separate device as described in Annex C of [UDA].

[1] The CloudProxy device shall use an out-of-band mechanism to decide whether to allow itself to be proxied. This is the only allowed instance of a CloudProxy service on any potential proxied devices. The CloudProxy service shall remove the CloudProxy service from any other devices it creates a proxy for.

[2] Depending on the configuration there may always be at least one device available since the CloudProxy device hosting the CloudProxy service is always known.

@DevId

Required. Type is **xsd:string**. Shall occur once per *<Device>* element. Shall not be an Empty string. Shall be unique within the CloudProxy service to identify a discovered device. Should be persisted across the discovered device entering and leaving the network(s) accessible by the CloudProxy service.

<DeviceType>

Required. Type is **xsd:string**. Shall not be an Empty string. Shall occur once per *<Device>* element. Shall have a value corresponding to the discovered devices *DeviceType*; either a value of urn:**schemas-upnp-org**:**device**:*deviceType*:*ver*
or urn:*domain-name*:**device**:*deviceType*:*ver* depending on whether the deviceType is UPnP standardized device or vendor defined respectively.

<DeviceProt>

Required. Type is **xsd:boolean**. Shall occur once per *<Device>* element. Shall have a value of "*1*" if the discovered device supports the DeviceProtection service. Shall have a value of "*0*" otherwise.

<DeviceProtStatus>

Required. Type is **xsd:boolean**. Shall occur once per *<Device>* element. Shall have a value of "*1*" if there is an active, secure connection between the local device and the CloudProxy service embedded control point. Shall have a value of "*0*" otherwise.

The *DeviceList* state variable is not evented.

6.4.4 State Variable *ProxyList*

This required state variable contains a list of local devices that have been proxied to the Cloud. For each unique UCCD advertised on a UCA interface one *<Proxy>* element shall be included. By default the CloudProxy device, when configured to allow itself to be proxied, shall not be proxied unless directed to by an authorized CloudProxy control point.

The *ProxyList* state variable is a string that contains a *ProxyList XML Document*. The following example shows a generalized “template” for the *ProxyList XML Document*. The example shows the elements that need to be filled out by individual implementations in the *vendor* character style. The Schema associated with the *ProxyList* state variable can be found at [XSD_CPPL].

```
<?xml version="1.0" encoding="UTF-8"?>
<ProxyList xmlns="urn:schemas-upnp-org:cloud:proxylist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:proxylist
      http://www.upnp.org/schemas/cloud/proxylist.xsd">
   <Proxy ProxyId="id unique to this service">
      <DeviceJID>localpart@domainpart/resourcepart of proxied device</DeviceJID>
      <Device DevId="id unique to this service">
         <DeviceType>device type according to UDA</DeviceType>
         <connected>1|0</connected>
         <DeviceProt>1|0</DeviceProt>
         <DeviceProtStatus>1|0</DeviceProtStatus>
      </Device>
   </Proxy>
</ProxyList>
```

<?xml>

Allowed. Case sensitive.

<ProxyList>

Required. Shall include a namespace declaration for the CloudProxy ProxyList XML Schema (“urn:schemas-upnp-org:cloud:proxylist”). This is the base wrapper element for the state variable. Shall include zero or more of the following elements:

<Proxy>

Allowed. <XML>. Indicates that a UDA device has been proxied to a *Cloud Account*. There shall be one occurence of this element per device connection to a *Cloud Account*. For example, if the UDA device is proxied to two different *Cloud Accounts* then there will be two occurrences of this element for that device.

@ProxyId

Required. Type is **xsd:string**. Shall occur once per *<Proxy>* element. Shall not be an Empty string. Shall be unique within the CloudProxy service per proxied device per *Cloud Account*. Should be persisted across the discovered device entering and leaving the network(s) accessable by the CloudProxy service.

<DeviceJID>

Required. Type is **xsd:string**. Shall occur once per *<Proxy>* element. Indicates the UCA side interface of the proxied device . Shall contain the **Full JID** of the proxied device.

<connected>

Required. Type is **xsd:boolean**. Shall occur once per *<Proxy>* element. Shall have a value of "*0*" if the discovered device is not currently connected to a UCS. Shall have a value of "*1*" if the discovered device is currently connected to the UCS. This element indicates essentially the status of the link between the UCS and the proxied device . If the link is disrupted, either on the UCA or UDA interface, a value of "*0*" would be indicated until the connection is re-established. Detection of link status is implementation dependent.

<Device>

Required. <XML>. Shall occur once per *<Proxy>* element. Each occurence shall correspond to a specific local device present in the *DeviceList* state variable that has been proxied to a *Cloud Account* .

@DevId

Required. Type is **xsd:string**. Shall occur once per *<Device>* element. Shall not be an Empty string. Shall match the value of the *@DevId* of the *<Device>* element corresponding to the specific local device present in the *DeviceList* state variable that has been proxied to a *Cloud Account*.

<DeviceType>

Required. Type is **xsd:string**. Shall occur once per *<Device>* element. Shall not be an Empty string. Shall match the value of the *<DeviceType>* element of the *<Device>* element corresponding to the specific local device present in the *DeviceList* state variable that has been proxied to a *Cloud Account* .

<DeviceProt>

Required. Type is **xsd:boolean**. Shall occur once per *<Device>* element. Shall have a value of "*1*" if the discovered device supports the DeviceProtection service. Shall have a value of "*0*" otherwise. Shall match the value of the *<DeviceProt>* element of the *<Device>* element corresponding to the specific local device present in the *DeviceList* state variable that has been proxied to a *Cloud Account*.

<DeviceProtStatus>

Required. Type is **xsd:boolean**. Shall occur once per *<Device>* element. Shall have a value of "*1*" if there is an active, secure connection between the local device and the CloudProxy service embedded control point. Shall have a value of "*0*" otherwise. Shall match the value of the *<DeviceProt>* element of the *<Device>* element corresponding to the specific local device present in the *DeviceList* state variable that has been proxied to a *Cloud Account*.

The *ProxyList* state variable is not evented.

6.4.5 State Variable *UCSList*

This required state variable indicates UCS(s) and related *Cloud Account*s that are available for proxying local devices to the *Cloud*.

The *UCSList* state variable is a string that contains a *UCSList XML Document*. The following example shows a generalized “template” for the *UCSList XML Document*. The example shows the elements that need to be filled out by individual implementations in the *vendor* character style. The Schema associated with the *UCSList* state variable can be found at [XSD_CPUL].

```
<?xml version="1.0" encoding="UTF-8"?>
<UCSList xmlns="urn:schemas-upnp-org:cloud:ucslist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:ucslist
      http://www.upnp.org/schemas/cloud/ucslist.xsd">
   <UCS UCSId="domainpart of the UCS Cloud Account of UCS JID">
      <UserName>localpart of the UCS Cloud Account of UCS JID</UserName>
   </UCS>
</UCSList>
```

<?xml>

Allowed. Case sensitive.

<UCSList>

Required. <XML>. Shall include a namespace declaration for the CloudProxy UCSList XML Schema (“urn:schemas-upnp-org:cloud:ucslist”). This is the base wrapper element for the state variable. Shall include zero or more of the following elements:

<UCS>

Allowed. <XML>. Indicates a UCS reachable by the CloudProxy service.

@UCSId

Required. Type is **xsd:string**. Shall occur once per *<UCS>* element. Identifies an UCS available to the CloudProxy service. Shall be the **`domainpart`** of the UCS. Shall not include a **`namepart`** or **`resourcepart`** of the *Cloud Account* **`JID`**.

<UserName>

Conditionally Required. Type is **xsd:string**. Shall occur zero or more times per *<UCS>* element. Identifies an UCS User available to the CloudProxy service. Shall be the **`namepart`** of the Cloud Account. Shall not include a **`domainpart`** or **`resourcepart`** of the *Cloud Account* **`JID`**. For example, a UCS may have been identified to the CloudProxy service but not connected to a *Cloud Account*; in this case, the *<UCS>* element would have no *<UserName>* elements. On the other hand, two separate *Cloud Accounts* could be established on the same UCS; in this case the *<UCS>* element would have two *<UserName>* elements. A *<UCS>* element shall not have more than one *<UserName>* element with the same value.

The *UCSList* state variable is not evented.

6.4.6 State Variable *A_ARG_TYPE_DevId*

This required state variable provides type information to identify a local device to be proxied to the Cloud or removed from the proxied state. The data type is **string**.

6.4.7 State Variable *A_ARG_TYPE_BareJID*

This required state variable provides type information to identify a **`Bare JID`** of the form **`domainpart`** or **`localpart@domainpart`** of a XMPP **`JID`**. The data type is **string**. It has a general Augmented Backus-Naur Form (ABNF) of:

```
domainpart =  IP-literal / IPv4address / ifqdn
```

or

```
[localpart"@"]domainpart
```

See [RFC_6122] section 2.1 for specific details.

6.4.8 State Variable *A_ARG_TYPE_FullJID*

This required state variable provides type information to identify a **`Full JID`** **`localpart@domainpart/resourcepart`** of a XMPP **`JID`**. The data type is **string**. It has a general Augmented Backus-Naur Form (ABNF) of:

```
[localpart"@"]domainpart["/"resourcepart]
```

See [RFC_6122] section 2.1 for specific details.

6.4.9 State Variable *A_ARG_TYPE_Port*

This required state variable indicates a port number to connect the UCCD or UCC-CP XMPP client to that is not the IANA assigned "xmpp-client" port number of 5622. The data type is **ui2**.

6.4.10 State Variable *A_ARG_TYPE_String*

This required state variable is use of type **string**.

6.5 EVENTING AND MODERATION

Table 6-2 — Eventing and Moderation

Variable Name	Evented	Moderation	
		Moderated[a]	Criteria
CloudProxyUpdate	*YES*	*YES*	Min 0.2 seconds between events
DeviceList	*NO*		
ProxyList	*NO*		
UCSLIst	*NO*		

[a] *YES* = The state variable shall be moderated with the criteria

6.5.1 Eventing of *CloudProxyUpdate*

The *CloudProxyUpdate* state variable is evented and moderated according to the GENA eventing mechanism as defined by the UPnP Device Architecture, version 2 [UDA]. When multiple modifications occur within the same moderation period, as determined by the implementation, only the last state of the *CloudProxyUpdate* state variable, needs to be evented. For example, when a *Cloud Account* is deleted as a result of a *DeleteUCSAccount()* both the *UCSList* and *ProxyList* are likely to change, the CloudProxy service would accumulate the state changes in the *CloudProxyUpdate* state variable before issuing the event.

6.6 ACTIONS

Table 6-3 — Actions

Name	Device R/A [a]	Control Point R/A [b]
AddUCSAccount	R	R [c]
AddProxyDevice	R	R
DeleteUCSAccount	R	R
DeleteProxyDevice	R	R
GetDeviceList	R	R
GetProxyList	R	R
GetUCSList()	R	R

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = REQUIRED, *O* = ALLOWED, *CR* = CONDITIONALLY REQUIRED, *CA* = CONDITIONALLY ALLOWED, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *O-D*).

[b] For a control point this column indicates whether a control point shall be capable of invoking this action, where *R* = REQUIRED, *A* = ALLOWED, *CR* = CONDITIONALLY REQUIRED, *CA* = CONDITIONALLY ALLOWED, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *O-D*).

[c] A CloudProxy control point shall not invoke this action over an unencrypted connection.

6.6.1 *AddUCSAccount()*

This required action directs the CloudProxy service to connect to a specific UCS or a specific *Cloud Account*. It returns a **`Bare JID`** of a Cloud Account when a UCS *Cloud Account* connection is successful, or a **`localpart JID`** when the UCS is reachable.

6.6.1.1 Arguments

Table 6-4 — Arguments for *AddUCSAccount()*

Argument	Direction	relatedStateVariable
User@Cloud	*IN*	*A_ARG_TYPE_BareJID*
Port	*IN*	*A_ARG_TYPE_Port*
UCSJID	*OUT*	*A_ARG_TYPE_BareJID*
Password	*OUT*	*A_ARG_TYPE_String*

6.6.1.2 Argument *User@Cloud*

This input argument is a string that identifies a UCS domainpart or a *Cloud Account* **localpart**@**domainpart** of a **JID** as indicated in 6.4.7 and Annex C of [UDA]. It is of type *A_ARG_TYPE_BareJID*. This value is case sensitive.

6.6.1.3 Argument *Port*

This input argument is a 2 byte unsigned integer or **ui2** value indicating the port that the CloudProxy service should connect its XMPP clients to. It is of type *A_ARG_TYPE_Port*.

6.6.1.4 Argument *UCSJID*

This output argument is a string that is of type *A_ARG_TYPE_BareJID* and:

1. if the CloudProxy service is able to connect (log in) to the UCS *Cloud Account* indicated by a **localpart**@**domainpart** *User@Cloud* shall have a form of "user@cloud" equivalent to the *User@Cloud* input argument,

or,

2. if the CloudProxy service is able to reach the UCS indicated by a **domainpart** only *User@Cloud* input argument shall have a form of an "@cloud" or "IP-literal" equivalent to the@**domainpart** of the *User@Cloud* input argument.

6.6.1.5 Argument *Password*

This output argument is a UTF-8 encoded string of the generated *User Identity* password created by a successful *Cloud Account* activation by the invocation of this action or the default "" (Empty string) value when only the identification of a UCS is performed. The string shall only contain characters [a-z, A-Z], digits [0-9], and special characters "!", "@", "#", "$".

6.6.1.6 Service Requirements

This action is a *Restrictable* action.

6.6.1.7 Control Point Requirements When Calling the Action

The CloudProxy control point shall have the *Admin* Role to successfully invoke this action.

6.6.1.8 Dependency on Device State

None.

6.6.1.9 Effect on Device State

If the CloudProxy service reaches the UCS identified by the **domainpart** of the *User@Cloud* input argument and that UCS is not already identified by a *<UCS>* element in the *UCSList* state variable then a *<UCS>* element shall be added to the *UCSList* state variable and its *@UCSId* attribute value set to the **domainpart JID**.

If the CloudProxy service reaches the UCS identified by the **domainpart** of the *User@Cloud* input argument and is able to log in to the associated *Cloud Account* and that UCS is not already identified by a *<UCS>* element in the *UCSList* state variable then a *<UCS>* element shall be added to the *UCSList* state variable and its *@UCSId* attribute value set to the **domainpart JID** and a *<UserName>* element shall be added to the *<UCS>* element and its value shall be the **localpart** of the **JID**.

If the CloudProxy service reaches the UCS identified by the **domainpart** of the *User@Cloud* input argument and is able to log in to the associated *Cloud Account* and that UCS is already identified by a *<UCS>* element in the *UCSList* state variable then a *<UserName>* element shall be added to the *<UCS>* element and its value shall be the **localpart** of the **JID**.

In all cases above the *CloudProxyUpdate* *<UCSUpdateId>* element shall be incremented by "1" and the *CloudProxyUpdate* state variable evented.

If one the three outcomes described above do not occur when invoking the *AddUCSAccount()* action then one of the error codes defined in Table 6-5 shall be returned.

If for some reason the CloudProxy service loses connectivity to a *Cloud Account* and cannot reconnect within a reasonable period of time, determined by the implementation, then it may remove the *<UserName>* element from the related *<UCS>* element or the *<UCS>* element altogether depending on if the *Cloud Account* can no longer be logged into or the UCS is no longer reachable respectively.

Note that when logging in to a *Cloud Account* the CloudProxy service shall act as a UCC-CP.

To avoid sending the *Cloud Account* password over an unsecured link the action shall cause the CloudProxy service to prompt the user with an out-of-band means for entering the password. In most cases, this would be a prompt on the device supporting the CloudProxy service to input the password via a user interface. If the password is not input within 30 seconds the action shall fail with error code 714. Repeated failures of the action should be handled similar to log in repeated authentication failures in [DP].

Once a specific *Cloud Account* log in on the target UCS has been successful, the CloudProxy service shall internally co-ordinate an update to the *ACL* state variable of the co-implemented DeviceProtection service. That is, it shall:

- add an *<ACL>*::*<Identities>*::*<User>*::*<Name>* element with a value identical to the value of the *User@Cloud* input argument if it does not already exist,
- add an *<ACL>*::*<Identities>*::*<User>*::*<RoleList>* element and a *<ACL>*::*<Roles>*::*<Role>*::*<Name>* element with a value identical to the value of the *User@Cloud* input argument if it does not already exist (the *VendorJID*),
- internally escalate the control point invoking the action to the *User Identity* above if not already escalated,
- Save an implementation determined encrypted[1] value of the verified *Cloud Account* password for future XMPP log ins associated with the created *User Identity* if not already stored,
- Assign a new *Password* for future DeviceProtection *UserLogin()* actions for the new or already created *User Identify*, the assigned password shall be of sufficient strength to avoid common password attacks. A minimum length of 10 characters is recommended.

[1] The encryption shall be of strength greater than or equal to the minimum encryption level required for DeviceProtection passwords.

- Calculate and store a STORED value as described in [DP]. The value of *Password* shall not be the same as the `password` for the Cloud Account login.

- Return the *Password* value to the CloudProxy control point for use in future log ins. The value of *Password* shall be reset upon each successful invocation of the action. To avoid the need to constantly update the *Password* for the *User Identify* associated with a specific *user@cloud* the associated *Role* may be added to the *RoleList* of the recognized CloudProxy control point invoking the action.

6.6.1.10 Errors

Table 6-5 — Error Codes for *AddUCSAccount()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Not authorized	The control point does not have the *Admin* *Role* needed to invoke this action.
702	JID not parsable	The *User@Cloud* input argument is not a valid **`Bare JID`**
703	UCS not reachable	The domainpart of the *User@Cloud* input argument timed out or was not reachable.
704	Maximum UCSs reached.	The maximum number of UCSs the service can maintain has been reached.
705	Maximum Cloud Accounts reached	The maximum number of UCS *Cloud Account*s the service can maintain has been reached.
712	Bad port	The UCS does not have an XMPP service at the port indicated.
713	Login failed	The CloudProxy service could not log in to the XMPP service.
714	Password timeout	The password was not entered in time for the action to be successful.

6.6.2 *AddProxyDevice()*

This required action controls which local devices are proxied to a *Cloud Account*.

6.6.2.1 Arguments

Table 6-6 — Arguments for *AddProxyDevice()*

Argument	Direction	relatedStateVariable
DevId	*IN*	*A_ARG_TYPE_DevId*
User@Cloud	*IN*	*A_ARG_TYPE_BareJID*
DeviceJID	*OUT*	*A_ARG_TYPE_FullJID*

6.6.2.2 Argument *DevId*

This input argument is a string that identifies a specific device present in the *DeviceList* state variable that is to be proxied to a *Cloud Account*. To successfully invoke this action the value of this input argument shall match a value of the *@DevId* attribute of a *<Device>* element of the *DeviceList* state variable. It is of type *A_ARG_TYPE_DevID*.

6.6.2.3 Argument *User@Cloud*

This input argument is a string corresponding to a **`Bare JID`** that identifies the *Cloud Account* that the local device is to be proxied to. Its **`localpart`** shall match a *<UCSList>*::*<UCS>*::*<UserName>* element of a *<UCS>* element matching its **`domainpart`** currently in the *UCSList* state variable. It is of type *A_ARG_TYPE_BareJID*. This value is case sensitive.

6.6.2.4 Argument *DeviceJID*

This output argument is a string corresponding to the UCCD identity (see Annex C of [UDA]) of a successfully proxied local device. It is a **`Full JID`**, that is, of form **`localpart`**@**`domainpart`**/**`resourcepart`**. It is of type *A_ARG_TYPE_FullJID*.

6.6.2.5 Service Requirements

This action is a *Restrictable* action.

The CloudProxy service behavior when the *ADDProxyDevice()* action is successfully invoked is described next. UCCD creation is described for the local device indicated by *DevId*.

6.6.2.5.1 Addressing or "**`presence`**"

1. The CloudProxy service shall bind to the UCS as a UCCD as described in section C.5.4 of [UDA] where the **`resourcepart`** is constructed from the local device **DeviceType** and **UDN**.

2. The CloudProxy service shall then send a **`<presence>`** "available[1]" stanza as described in section C.6 of [UDA2.0] to the selected UCS. If the local device does not have a **configId** then the CloudProxy service shall create and maintain an associated **configIdCloud** value when constructing the **`<presence>`** stanza **<uc>** element.

6.6.2.5.2 Eventing or "**`PubSub`**"

1. The CloudProxy service shall then determine if the local device has any evented state variables. If it does, then the UCCD shall check to see if there is an existing **`PubSub collection`** for that device as described in section C.7.1.1 of [UDA]. If the **`collection`** does not exist then the UCCD will need to construct its **`PubSub collection`** as described in section C.7.2 of [UDA].

2. Once the **`PubSub collection`** exists, then the CloudProxy service will only need to update the **`PubSub collection`** when events occur on the local device. This will require that the embedded local control point of the CloudProxy service subscribe to all evented state variables of the local device. Then, when it receives a local event, it shall make that event available via an internal interface to its UCCD interface and publish that event as described in section C.7.3 of [UDA].

6.6.2.5.3 Description

1. If the local device includes a **`<iconList>`** element in its DDD then the CloudProxy service shall also publish its icon information for the UCCD as described in section C.6.6.3 of [UDA].

2. When the UCCD interface receives an **`<iq>`** stanza from a UCC-CP with a **<query>** element then the CloudProxy service shall construct a response, either an error message or a DDD/SCPD response, according to section C.6.6 of of [UDA]. This may be based on a cached and pre-processed version of the DDD/SCPD or constructed in real-time by the CloudProxy service first fetching the DDD/SCPD using its local embedded control point and then converting it to a UCCD compliant version and responding to the UCC-CP.

3. For each local device not containing a CloudTransport service [CTS] a *CloudTransport:1* service shall be included in the DDD and a CloudTransport:1 SCPD generated so that any UCC-CP

[1] The CloudProxy service may also send a <**`presence`**> "unavailable" stanza if the local device is offline but the CloudProxy service has previously cached the DDD and SCPD information needed to construct the **`<presense>`** stanza and the UCCD **`PubSub collection`**.

interacting with the UCCD interface of the proxied local device will see a CloudTransport service available.

6.6.2.5.4 Control

1. When the UCCD interface receives an **<iq>** stanza from a UCC-CP containing a SOAP request it shall forward the SOAP element to the targeted local device in a properly formatted HTTP message as described in section 3 of [UDA]. Upon receiving the response from the local device the CloudProxy service shall format the response, either an error message or SOAP response into an return **<iq>** stanza for the initial **<iq>** request.

6.6.2.5.5 Timeouts

1. If, after a sufficient amount of time (suggested 30 seconds), the CloupProxy service has not received a response from the local device it should send an **<iq>** stanza response with an XMPP **modify** **@type** <**error>** element with a **<text>** element as follows:

```
 <iq
   id="vendor defined value"
   from="localpart@domainpart/resourcepart of UCCD conforming to
         section C of UDA 2.0"
   to="UCCP requesting response to UCCD interface of proxied device"
   type="error"
   <error type="modify">
      <undefined-condition xmlns="urn:ietf:params:xml:ns:xmpp=stanzas"/>
      <text xml:lang="en" xmlns="urn:ietf:params:xml:ns:xmpp=stanzas">
         "Proxied device is not responding"
      </text>
   </error>
</iq>
```

2. If the CloudProxy service receives the response from the proxied local device after it has sent the above error, it shall discard the response.

6.6.2.5.6 Maintenance of proxied device presence

The embedded control point shall monitor the status of all local proxied devices for the following changes in status and provide the described response on the UCCD interface, as well as, update the specific CloudProxy state variable(s) as follows:

1. If the local proxied device issues an **sspd:byebye** message then the CloudProxy service shall:

   - Send an "unavailable" **<presence>** stanza for all UCCD interfaces proxied for this local device, that is, this stanza will be sent once for each *ProxyList* state variable *<Proxy>*:: *<Device>* ::*@DevId* attribute corresponds to the local device.
   - Update all *ProxyList* state variable *<connected>* elements associated with the local device to a value of "*0*".
   - Increment the *CloudProxyUpdate*::*<ProxyUpdateId>* value by "1" and event it.

2. If the local proxied device issues an **sspd:update** message then the CloudProxy service shall:

   - Determine if the local devices configuration has changed by inspecting its **BOOTID.UPNP.ORG**, **CONFIGID.UPNP.ORG** (if available) and DDD/SCPD XML documents (if necessary) and following the procedures in section C.6.3 and C.7 of [UDA] to update the UCCDs **PubSub** **collection** if it has changed.

3. If the CloudProxy service at some point determines that the local proxied device has left the network[1] without issuing an **ssdp::byebye**, by perhaps pinging the local device occasionally with its embedded control point then it shall respond as described in step 1 in this section.

6.6.2.5.7 Other requirements

The UCCD interface exposed for a local device shall conform to all requirements of a UCCD in [UDA] Annex C.

6.6.2.6 Control Point Requirements When Calling the Action

The CloudProxy control point shall have, in order to successfully invoke the*AddProxyDevice()* action, the *Admin* *Role* and the specific *vendorJID* *Role* corresponding to the "User@Cloud" value matching the *User@Cloud* input argument.

6.6.2.7 Dependency on Device State

The action may not be successful if the target local device is already proxied to the target *Cloud Account* or the CloudProxy service has run out of resources to proxy additional devices.

6.6.2.8 Effect on Device State

If the *AddProxyDevice()* action is successful then the action shall return a *DeviceJID* with a value corresponding the proxied UCCD's **Full JID** as described in Annex C of the [UDA]. otherwise it shall return one of the error codes in Table 6-7.

Upon successful invocation of the *AddProxyDevice()* action the *ProxyList* state variable shall be updated with an additional *<Proxy>* element entry whose:

- *<Device>* element is identical to the *UCSList* state variable *<Device>* element indicated in the *DevId* input argument but has the added *Role* associated with the *vendorJID* identified by the *User@Cloud* input argument value.
- *<DeviceJID>* element has an identical value to the *DeviceJID* output argument.
- *<connected>* element has a value of "*1*".

The *CloudProxyUpdate* *<ProxyUpdateId>* element shall be incremented by "1" and the *CloudProxyUpdate* state variable evented.

6.6.2.9 Errors

Table 6-7 — Error Codes for *AddProxyDevice()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Not authorized	The control point does not have the *Admin* role needed to invoke this action.
702	JID not parsable	The *User@Cloud* input argument is not a valid **Bare JID**.
706	Device ID invalid	The *DevId* input argument does not match an *@DevId*.
706	Not authorized Account	The control point does not have the *vendorJID* *Role* needed to invoke the action.

[1] If the local device remains offline for an implementation determined period of time the CloudProxy service may remove the device from the *ProxyList* and *DeviceList* state variables to free up resources or as a general cleanup of CloudProxy state. The behavior when removing a local device from the *ProxyList* and *DeviceList* are described in section 6.6.4.6 and 6.2 respectively. It is suggested that the time threshold for removal from the *DeviceList* state variable be greater than that for the *ProxyList* state variable.

ErrorCode	errorDescription	Description
707	Not enough connections	The CloudProxy services cannot add another UCCD connection.
708	The target device is proxied.	The target device has already been proxied to the requested *Cloud Account*.

6.6.3 *DeleteUCSAccount()*

This required action removes a UCS or *Cloud Account* from the CloudProxy service. It can also result in the removal of proxied devices as well.

6.6.3.1 Arguments

Table 6-8 — Arguments for *DeleteUCSAccount()*

Argument	Direction	relatedStateVariable
BareJID	*IN*	*A_ARG_TYPE_BareJID*

6.6.3.2 Argument *BareJID*

This input argument indicates either a UCS or a *Cloud Account* that is to be removed from the CloudProxy service depending on if the input argument is a **`domainpart`** only (for example "`mycloud.org`") or **`localpart`**`@`**`domainpart`** (for example "`jeffrey@mycloud.org`") entry respectively. It is of type *A_ARG_TYPE_BareJID*. This value is case sensitive.

6.6.3.3 Service Requirements

This action is a *Restrictable* action.

6.6.3.4 Control Point Requirements When Calling The Action

The CloudProxy control point shall have, in order to successfully invoke the *DeleteUCSAccount()* action, the *Admin Role* and the specific *`vendorJID`* *Role* corresponding to the "`User@Cloud`" value that matches the *BareJID* input argument. If the target is a *<UCS>* without any *<UserName>* elements then only the *Admin* Role is required.

6.6.3.5 Dependency on Device State

The outcome of the *DeleteUCAccount()* action is dependent on the state of the *UCSList* and *ProxyList* state variables, the *Roles* assigned to the CloudProxy control point, and the requested deletions. Table 6-9 lists the possible outcomes dependent on the input parameter and the current state of the *UCSList* state variable.

Table 6-9 — Outcomes for *DeleteUCSAccount()*

BareJID	State of *UCSList* at invocation	State of *UCSList* at response	State of *ProxyList* at response	ErrorCode (if any)
Is **domainpart** only	The *UCSList* state variable has a *<UCS>* element with *@UCSId* attribute matching the **domainpart** only *BareJID* input argument but no *Cloud Accounts* assigned to the UCS.	The targeted *<UCS>* element is removed from the *UCSList* state variable and increment the *CloudProxyUpdate*:: *<UCSUpdateId>* element and event the state variable.	No change.	None.
Is **domainpart** only	The *UCSList* state variable has a *<UCS>* element with *@UCSId* attribute matching the **domainpart** only *BareJID* input argument with one or more *Cloud Accounts* assigned to the UCS.	No change.	No change.	711
Is **localpart**@ **domainpart**	The *UCSList* state variable has a *<UCS>* element with *@UCSId* attribute matching the **domainpart** of the *BareJID* input argument and an associated *<UserName>* element matching the **localpart** of the *BareJID* input argument.	The *<UserName>* element associated with the *<UCS>* element shall be removed from the *UCSList* state variable, otherwise and increment the *CloudProxyUpdate*:: *<UCSUpdateId>* element and event the state variable. All occurrences of the *User Identity* and *VendorJID Role* associated with the target *Cloud Account* shall also be removed from the co-supported DeviceProtection *<ACL>* state variable as well as *Cloud Account* and DeviceProtection *Password* collateral.	All local devices proxied with the target *Cloud Account* are removed from the ProxyList as described in 6.6.4 and increment the *CloudProxyUpdate*:: *<ProxyUpdateId>* element and event the state variable.	None.

These outcomes assume the CloudProxy control point has a *Role* matching the *BareJID* input argument when it has a **localpart**@d**omainpart** format and the targeted UCS is not a permanent UCS bundled with the CloudProxy service.

6.6.3.6 Effect on Device State

See Table 6-9 rows for outcome of *UCSList* and *ProxyList* state variables.

6.6.3.7 Errors

Table 6-10 — Error Codes for *DeleteUCSAccount()*

ErrorCode	errorDescription	Description

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Not authorized	The control point does not have the *Admin* role needed to invoke this action.
702	JID not parsable	The *BareJID* input argument is not a valid **Bare JID**.
709	UCS ID invalid	The **domainpart** of the *BareJID* input argument does not match an *@UCSId*.
710	UCS is permanent	The UCS is permanently provided by the CloudProxy service.
711	Other Cloud Accounts	There are other *Cloud Account*s associated with the UCS, they need to be removed before the UCS can be removed.

6.6.4 *DeleteProxyDevice()*

This required action removes an already proxied device from the proxied state.

6.6.4.1 Arguments

Table 6-11 — Arguments for *DeleteProxyDevice()*

Argument	Direction	relatedStateVariable
DeviceJID	*IN*	*A_ARG_TYPE_FullJID*

6.6.4.2 Argument *DeviceJID*

This input argument identifies a specific device in the ProxyList state variable that is to be removed from the proxy state, that is, its UCCD is disconnected from its associated *Cloud Account*. It is of type *A_ARG_TYPE_FullJID*.

6.6.4.3 Service Requirements

This action is a *Restrictable* action.

6.6.4.4 Control Point Requirements When Calling The Action

The CloudProxy control point shall have, in order to successfully invoke the *DeleteProxyDevice()* action, the *Admin Role* and the specific *vendorJID Role* corresponding to the "User@Cloud" value that matches the **localpart**@**domainpart** *of the DeviceJID* input argument.

6.6.4.5 Dependency on Device State

To be successfully invoked the *DeviceJID* input argument shall match a *ProxyList* state variable *<Proxy>*::*<DeviceJID>* element value exactly.

6.6.4.6 Effect on Device State

If the *DeviceJID* input argument matches a *ProxyList* state variable *<Proxy>*::*<DeviceJID>* element as described in 6.6.4.5 then the CloudProxy service shall close the connection **<stream>** as described in section C.12 of [UDA].

Prior to closing the **<stream>** the CloudProxy service may remove the PubSub collection associated with the proxied device, however it is preferable for the CloudProxy service to internally cache known UCCD **PubSub collection**(s) and instead clean up the **PubSub** collection(s) associated with a specific *Cloud Account* during the execution of *UCSDeleteAccount()* action, since the deletion of the proxy may only be temporary.

Once the **stream** has been closed, the *ProxyList* state variable shall be updated by removing the *<Proxy>* element identified by the *DeviceJID* input argument and the *CloudProxyUpdate*::*<ProxyUpdateID>* element are incremented by "1" and the *CloudProxyUpdate* state variable evented.

Otherwise, the action shall return the appropriate error code indicated in Table 6-12.

6.6.4.7 Errors

Table 6-12 — Error Codes for *DeleteProxyDevice()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
715	Device JID invalid	The target UCCD does not match a *DeviceJID* in the *ProxyList* state variable.

6.6.5 *GetDeviceList()*

This required action is used to retrieve the *DeviceList* state variable. This status is returned in the *DeviceList* output argument which contains a properly escaped *DeviceList XML Document*.

6.6.5.1 Arguments

Table 6-13 — Arguments for *GetDeviceList()*

Argument	Direction	relatedStateVariable
DeviceList	*OUT*	*DeviceList*

6.6.5.2 Argument *DeviceList*

This output argument is a string that contains the *DeviceList XML Document*.

6.6.5.3 Service Requirements

None.

6.6.5.4 Control Point Requirements When Calling the Action

This action is a *Non-Restrictable* action and can be invoked by any control point.

6.6.5.5 Dependency on Device State

The return value is dependent on the value of the *DeviceList* state variable.

6.6.5.6 Effect on Device State

None.

6.6.5.7 Errors

Table 6-14 — Error Codes for *GetDeviceList()*

ErrorCode	errorDescription	Description

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
7XX		Reserved for future extensions

6.6.6 *GetProxyList()*

This required action is used to retrieve the *ProxyList* state variable. This status is returned in the *ProxyList* output argument which contains a properly escaped *ProxyList XML Document*.

6.6.6.1 Arguments

Table 6-15 — Arguments for *GetProxyList()*

Argument	Direction	relatedStateVariable
ProxyList	*OUT*	*ProxyList*

6.6.6.2 Argument *ProxyList*

This output argument is a string that contains the *ProxyList XML Document*.

6.6.6.3 Service Requirements

This action is a *Restrictable* action.

6.6.6.4 Control Point Requirements When Calling the Action

The CloudProxy control point shall have the *Admin* Role to successfully invoke this action.

6.6.6.5 Dependency on Device State

The return value is dependent on the value of the *ProxyList* state variable.

6.6.6.6 Effect on Device State

None.

6.6.6.7 Errors

Table 6-16 — Error Codes for *GetProxyList()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
7XX		Reserved for future extensions.

6.6.7 *GetUCSList()*

This required action is used to retrieve the *UCSList* state variable. This status is returned in the *UCSList* output argument which contains a properly escaped *UCSList XML Document*.

6.6.7.1 Arguments

Table 6-17 — Arguments for *GetUCSList()*

Argument	Direction	relatedStateVariable
UCSList	*OUT*	*UCSList*

6.6.7.2 Argument *UCSList*

This output argument is a string that contains the *UCSList XML Document*.

6.6.7.3 Service Requirements

This action is a *Restrictable* action.

6.6.7.4 Control Point Requirements When Calling The Action

The CloudProxy control point shall have the *Admin* Role to successfully invoke this action.

6.6.7.5 Dependency on Device State

The return value is dependent on the value of the *UCSList* state variable.

6.6.7.6 Effect on Device State

None.

6.6.7.7 Errors

Table 6-18 — Error Codes for *GetUCSList()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
7XX		Reserved for future expansion.

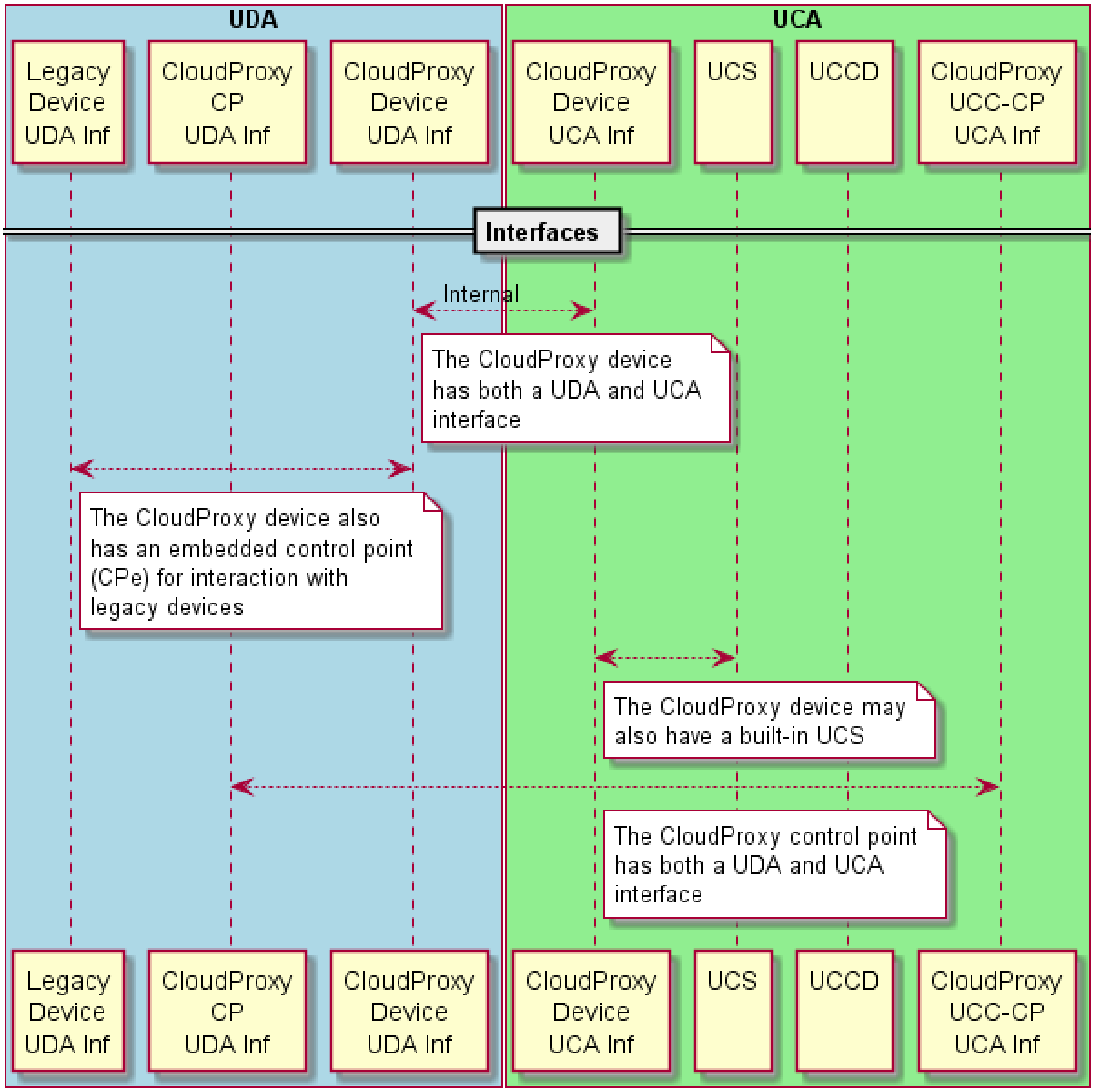
7 THEORY OF OPERATIONS (INFORMATIVE)

This section walks through general interactions within a scenario composed of a CloudProxy device, a CloudProxy control point, multiple legacy devices, i.e. UDA only, a UCA UCS, UCCD and UCC-CP.

7.1 GENERAL CONFIGURATION

A CloudProxy device will have both a UDA and UCA interface, as well as, a UDA side embedded control point. It may also have a built-in UCS. A typical sequence for enabling *Cloud* side access to legacy devices is: 1) establish the UCS and *Cloud Account* (`Bare JID`) to connect devices to, 2) authorize the CloudProxy device to connect to the *Cloud Account*, 3) establish DeviceProtection *Roles* between the CloudProxy device and the legacy devices if they support DeviceProtection, 4) connect the legacy device to the *Cloud Account*, 5) test UCA side connection and escalate access privileges. The typical scenario is illustration in Figure 7.1.

Figure 7-1 CloudProxy Configuration


UDA
UCA
Legacy Device UDA Inf
CloudProxy CP UDA Inf
CloudProxy Device UDA Inf
CloudProxy Device UCA Inf
UCS
UCCD
CloudProxy UCC-CP UCA Inf
Interfaces
Internal
The CloudProxy device has both a UDA and UCA interface
The CloudProxy device also has an embedded control point (CPe) for interaction with legacy devices
The CloudProxy device may also have a built-in UCS
The CloudProxy control point has both a UDA and UCA interface

7.2 CONNECTING TO A UCS

A CloudProxy Device can have a built-in UCS. In this case, the *UCSList* state variable will indicate at least one UCS is available for proxying local devices. The CloudProxy service does not include actions for establishing a *Cloud Account* on a UCS (see section C.5 of [UDA] for additional details) but does include actions for connecting to an already established *Cloud Account*.

In the first example, the initial state of the *CloudProxyUpdate*, *DeviceList*, *ProxyList*, and *UCSList* state variables are as follows:

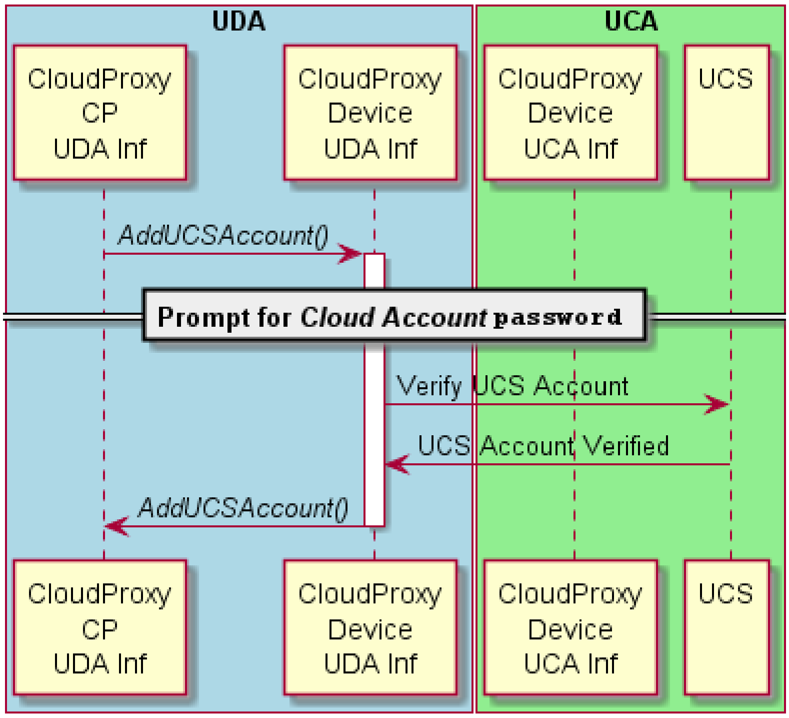
```
<?xml version="1.0" encoding="UTF-8"?>
<CloudProxyUpdate xmlns="urn:schemas-upnp-org:cloud:cloudproxyupdate"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:cloudproxyupdate
      http://www.upnp.org/schemas/cloud/ cloudproxyupdate.xsd">
   <ProxyUpdateId>0</ProxyUpdateId>
   <DeviceUpdateId>2</DeviceUpdateId>
   <UCSUpdateId>0</UCSUpdateId>
</CloudProxyUpdate>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceList xmlns="urn:schemas-upnp-org:cloud:devlist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:devlist
      http://www.upnp.org/schemas/cloud/devlist.xsd">
      <Device DevId="dev00">
         <connected>0</connected>
         <DeviceType>urn:schemas-upnp-org:device:CloudProxy:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>1</DeviceProtStatus>
      </Device>
      <Device DevId="dev01">
         <connected>0</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:1</DeviceType>
         <DeviceProt>0</DeviceProt>
         <DeviceProtStatus>0</DeviceProtStatus>
      </Device>
      <Device DevId="dev02">
         <connected>0</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:4</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>0</DeviceProtStatus>
      </Device>
</DeviceList>

<?xml version="1.0" encoding="UTF-8"?>
<ProxyList xmlns="urn:schemas-upnp-org:cloud:proxylist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:proxylist
      http://www.upnp.org/schemas/cloud/proxylist.xsd">
</ProxyList>

<?xml version="1.0" encoding="UTF-8"?>
<UCSList xmlns="urn:schemas-upnp-org:cloud:ucslist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:ucslist
      http://www.upnp.org/schemas/cloud/ucslist.xsd">
</UCSList>
```

Indicating that there are no UCSs currently available to the CloudProxy service but that there are 3 devices - a MediaServer:1 with no DeviceProtection service, a MediaServer:4 with DeviceProtection service, and the CloudProxy:1 device itself - for potential proxying. Note that a CloudProxy:1 service will not proxy another CloudProxy service or CloudProxy device. Also, note that in this case the CloudProxy service has been configured to let itself be proxied to the *Cloud*.

The CloudProxy control point invokes an *AddUCSAccount()* action with a known *User@Cloud* as illustrated in Figure 7-2.

Figure 7-2 CloudProxy UCS Setup


UDA
UCA
CloudProxy CP UDA Inf
CloudProxy Device UDA Inf
CloudProxy Device UCA Inf
UCS
AddUCSAccount()
Prompt for Cloud Account password
Verify UCS Account
UCS Account Verified
AddUCSAccount()
CloudProxy CP UDA Inf
CloudProxy Device UDA Inf
CloudProxy Device UCA Inf
UCS

The action request and response are as follows:

```
Request (SOAP):
AddUSCAccount(
   jeffrey@mycloud.org
)
```

Note that the CloudProxy control point needs to be logged in with the *Admin* *Role* to successfully invoke a *AddUCSAccout()* action. Within the next 30 seconds the CloudProxy service prompts for and receives the correct *Cloud Account* password and successfully logs in to the UCS. The CloudProxy service then returns the **Bare JID** and new *User Identity Password* as follows:

```
Response (SOAP):
AddUCSAccount(
   jeffrey@mycloud.org,
   oC84Dl#66$
)
```

The *UCDList* and *CloudProxyUpdate* state variables are subsequently updated and the *CloudProxyUpdate* state variable evented as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<UCSList xmlns="urn:schemas-upnp-org:cloud:ucslist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:ucslist
      http://www.upnp.org/schemas/cloud/ucslist.xsd">
   <UCS UCSId="mycloud.org">
      <UCSName>jeffrey</UCSName>
   </UCS>
</UCSList>
```

and

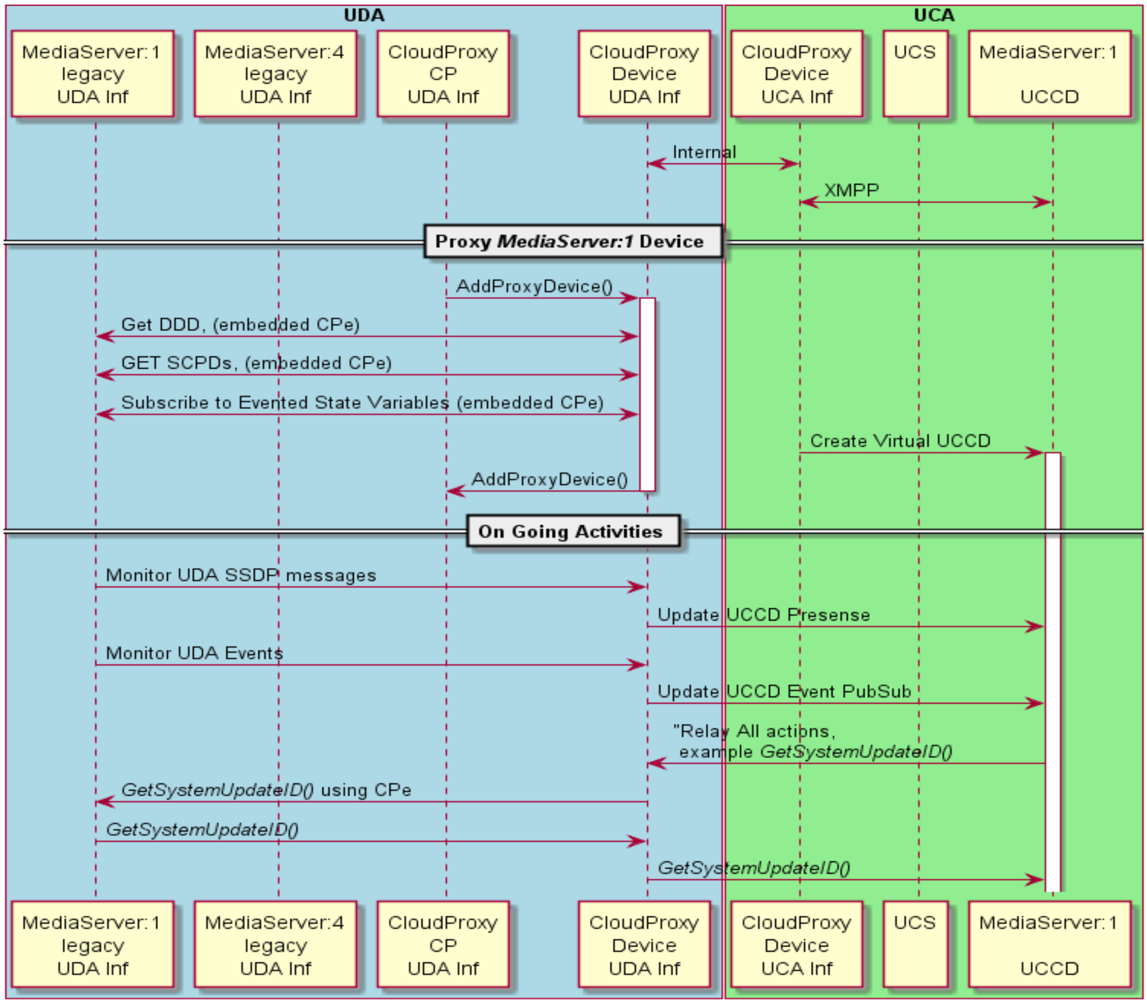
```
<?xml version="1.0" encoding="UTF-8"?>
<CloudProxyUpdate xmlns="urn:schemas-upnp-org:cloud:cloudproxyupdate"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:cloudproxyupdate
      http://www.upnp.org/schemas/cloud/ cloudproxyupdate.xsd">
   <ProxyUpdateId>0</ProxyUpdateId>
   <DeviceUpdateId>2</DeviceUpdateId>
   <UCSUpdateId>1</UCSUpdateId>
</CloudProxyUpdate>
```

Also, internally the CloudProxy service interacts with the co-located DeviceProtection service and updates the *ACL* state variable so that a *vendorJID* *Role* and *User Identity* with a value corresponding to the value of *User@Cloud* argument ("jeffrey@mycloud.org") are added. The invoking control point is also given the newly created *User Identity* for this session. The CloudProxy control point can store the *password* for future access to this *Role* or *User* Identity.

7.3 PROXYING A LOCAL DEVICE

To proxy a device to the *Cloud* an *AddProxyDevice()* action is invoked by the CloudProxy control point as illustrated in Figure 7-3. The input arguments correspond to an *@DevId* attribute of the *DeviceList* state variable and a *<UCS>*::*<UserName>* pair of the *UCSList* state variable corresponding to the local device to be proxied and the *Cloud Account* the local device is to be proxied to respectively. If the action is invoked successfully, then the response returns the **Full JID** of the proxied device in the *DeviceJID* output argument.

Figure 7-3 Proxy a Local Device


UDA
UCA
MediaServer:1 legacy UDA Inf
MediaServer:4 legacy UDA Inf
CloudProxy CP UDA Inf
CloudProxy Device UDA Inf
CloudProxy Device UCA Inf
UCS
MediaServer:1 UCCD
Internal
XMPP
Proxy *MediaServer:1* Device
AddProxyDevice()
Get DDD, (embedded CPe)
GET SCPDs, (embedded CPe)
Subscribe to Evented State Variables (embedded CPe)
Create Virtual UCCD
AddProxyDevice()
On Going Activities
Monitor UDA SSDP messages
Update UCCD Presense
Monitor UDA Events
Update UCCD Event PubSub
"Relay All actions, example *GetSystemUpdateID()*
GetSystemUpdateID() using CPe
GetSystemUpdateID()
GetSystemUpdateID()

In this example the MediaServer:1 device is proxied. The action request and response are as follows:

Request (SOAP):
```
AddProxyDevice(
   "jeffrey@mycloud.org","dev01"
)
```

and

Response (SOAP):
```
AddProxyDevice(
   "jeffrey@mycloud.org/urn:schemas-upnp-org:device:MediaServer:4:uuid:
         e70e9d0e-d9eb-4748-b163-636a323e7950"
)
```

The *ProxyList* and *CloudProxyUpdate* state variables are subsequently updated and the *CloudProxyUpdate* state variable evented as follows:

```
<ProxyList xmlns="urn:schemas-upnp-org:cloud:proxylist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:proxylist
      http://www.upnp.org/schemas/cloud/proxylist.xsd">
   <Proxy ProxId="pro01">
      <DeviceJID>
```

```
         jeffrey@mycloud.org/urn:schemas-upnp-
         org:device:MediaServer:4:uuid:e70e9d0e-d9eb-4748-b163-636a323e7950
      </DeviceJID>
      <Device DevId="dev01">
         <connected>1</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>0</DeviceProtStatus>
      </Device>
   </Proxy>
</ProxyList>
```

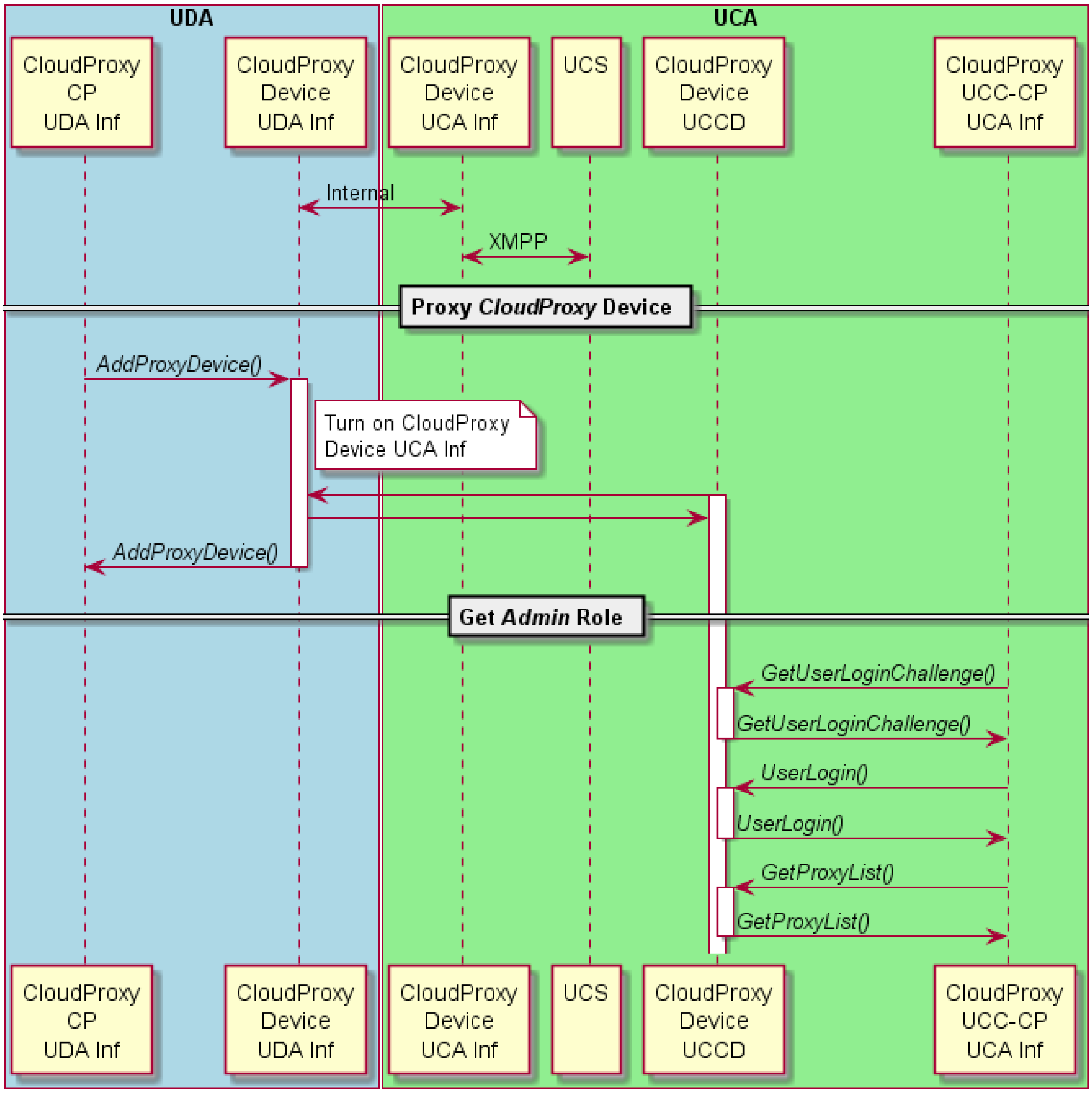
and

```
<?xml version="1.0" encoding="UTF-8"?>
<CloudProxyUpdate xmlns="urn:schemas-upnp-org:cloud:cloudproxyupdate"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:cloudproxyupdate
      http://www.upnp.org/schemas/cloud/ cloudproxyupdate.xsd">
   <ProxyUpdateId>1</ProxyUpdateId>
   <DeviceUpdateId>2</DeviceUpdateId>
   <UCSUpdateId>1</UCSUpdateId>
</CloudProxyUpdate>
```

7.4 PROXYING A CLOUDPROXY DEVICE

Similarly the CloudProxy device itself can be proxied to the *Cloud* (if configured to allow this) using the *AddProxyDevice()* with the available *@DevId* for the CloudProxy device itself.

Figure 7-4 Proxying the CloudProxy Device



Request (SOAP):

```
AddProxyDevice(
   "Jeffrey@mycloud.org","dev00"
)
```

and

Response (SOAP):

```
AddProxyDevice(
   "jeffrey@mycloud.org/urn:schemas-upnp-org:device:CloudProxy:1:uuid:
   88509d0e-e8f5-80ca-4123-225886a50ee7"
)
```

Similarly, the *ProxyList* and *CloudProxyUpdate* state variables are subsequently updated and the *CloudProxyUpdate* state variable evented as follows:

```
<ProxyList xmlns="urn:schemas-upnp-org:cloud:proxylist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:proxylist
      http://www.upnp.org/schemas/cloud/proxylist.xsd">
   <Proxy ProxId="pro00">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-org:device:CloudProxy:1:uuid:
         88509d0e-e8f5-80ca-4123-225886a50ee7
      </DeviceJID>
      <Device DevId="dev00">
         <connected>1</connected>
         <DeviceType>urn:schemas-upnp-org:device:CloudProxy:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>1</DeviceProtStatus>
      </Device>
   </Proxy>
   <Proxy ProxId="pro01">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-
         org:device:MediaServer:4:uuid:e70e9d0e-d9eb-4748-b163-636a323e7950
      </DeviceJID>
      <Device DevId="dev01">
         <connected>1</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>0</DeviceProtStatus>
      </Device>
   </Proxy>
</ProxyList>
```

and

```
<?xml version="1.0" encoding="UTF-8"?>
<CloudProxyUpdate xmlns="urn:schemas-upnp-org:cloud:cloudproxyupdate"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:cloudproxyupdate
      http://www.upnp.org/schemas/cloud/ cloudproxyupdate.xsd">
   <ProxyUpdateId>2</ProxyUpdateId>
   <DeviceUpdateId>2</DeviceUpdateId>
   <UCSUpdateId>1</UCSUpdateId>
</CloudProxyUpdate>
```

A CloudProxy control point will also be a UCC-CP and as a check can log in to the UCA interface for the `jeffrey@mycloud.org` account and verify the presense of the CloudProxy device. As well as the services already present on the legacy devices, a CloudTransport service [CTS], [PROXY] will be made available on the UCA interface for each proxied device assuming it is not available on the UDA interface of the legacy device, courtesy of CloudProxy device. In the example above, the CloudProxy UCC-CP first gets *Admin* privileges by an "`Administration`" *User Identity* log in with the actions *GetUserLoginChallenge()* and *UserLogin()* using the credentials and log in information from its UDA control point and then issues a *GetProxyList()* action to check the status of all devices currently proxied using the CloudProxy service as follows:

`Request UCC-CP->CloudProxy(XMPP/SOAP):`

```
<iq
   to="jeffrey@mycloud.org/urn:schemas-upnp-org:device:CloudProxy:1:
      uuid:88509d0e-e8f5-80ca-4123-225886a50ee7"
   id="cp-1-soap-action-1"
   type="set">
   <s:Envelope
      xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
      s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <s:Header mustUnderstand="1">
         <uc xmlns="urn:schemas-upnp-org:cloud-1-0"
```

```
               serviceId="RenderingControl"/>
      </s:Header>
      <s:Body>
         <u:GetProxyList xmlns:u="urn:schemas-upnp-org:
            service:CloudProxy:1">
         </u:GetProxyList>
      </s:Body>
   </s:Envelope>
</iq>
```

and

Response (XMPP/SOAP):

```
<iq
   to="jeffrey@mycloud.org/urn:schemas-upnp-org:ControlPoint:1:
       ad93e8f5-634b-4123-80ca-225886a5c0e8"
   id="cp-1-soap-action-1"
   type="result">
   <s:Envelope
      xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
      s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <s:Header mustUnderstand="1">
         <uc xmlns="urn:schemas-upnp-org:cloud-1-0"
             serviceId="CloudProxy"/>
      </s:Header>
      <s:Body>
         <u:GetProxyList xmlns:u="urn:schemas-upnp-org:
            service:CloudProxy:1">
            <ProxyList>
               ;&ltProxyList xmlns=;&quoturn:schemas-upnp-org:cloud:
               proxylist;&quotxmlns:xsi=;&quothttp://www.w3.org/2001
               /XMLSchema-instance;&quotxsi:schemaLocation=;&quoturn
               :schemas-upnp-org:cloud:proxylist http://www.upnp.org
               /schemas/cloud/proxylist.xsd;&quot;&gt;&ltProxy ProxI
               d=;&quotpro00;&quot;&gt;&ltDeviceJID;&gtjeffrey@myclo
               ud.org/urn:schemas-upnp-org:device:CloudProxy:1:uuid:
               88509d0e-e8f5-80ca-4123-225886a50ee7;&lt/DeviceJID;&g
               t;&ltDevice DevId=;&quotdev00;&quot;&gt;&ltconnected;
               &gtno;&lt/connected;&gt;&ltDeviceType;&gturn:schemas-
               upnp-org:device:CloudProxy:1;&lt/DeviceType;&gt;&ltDe
               viceProt;&gt1;&lt/DeviceProt;&gt;&ltDeviceProtStatus;
               &gton;&lt/DeviceProtStatus;&gt;&gt;&lt/Device;&gt;&lt
               /Proxy;&gt;&ltProxy ProxId=;&quotpro01;&quot;&gt;&ltD
               eviceJID;&gtjeffrey@mycloud.org/urn:schemas-upnp-org:
               device:MediaServer:4:e70e9d0e-d9eb-4748-b163-636a323e
               7950;&lt/DeviceJID;&gt;&ltDevice DevId=;&quotdev01;&q
               uot;&gt;&ltconnected;&gtno;&lt/connected;&gt;&ltDevic
               eType;&gturn:schemas-upnp-org:device:MediaServer:1;&l
               t/DeviceType;&gt;&ltDeviceProt;&gt0;&lt/DeviceProt;&g
               t;&ltDeviceProtStatus;&gt0;&lt/DeviceProtStatus;&gt;&
               lt/Device;&gt;&lt/Proxy;&gt;&lt/ProxyList;&gt
            </ProxyList>
         </u:GetProxyList>
      </s:Body>
   </s:Envelope>
</iq>
```

7.5 USING DEVICEPROTECTION ROLES

DeviceProtection requires that a paired relationship be defined between a control point and a device supporting the DeviceProtection:1 service and a secured connection established between the authenticated endpoints. For a device on the UDA side this relationship will be between the embedded control point of the CloudProxy device and the legacy device. To use device protection *Roles* from the *Cloud* on a proxied device the UCC-CP should initiate a *User* log in first to a *User Identity* that possesses the desired *R*ole and then begin using the *Role*. The activation action will be either towards the DeviceProtection service in the proxied device, if the *Role* is desired on the proxied device, or the DeviceProtection service on the CloudProxy device if the *Role* is desired for the CloudProxy service itself.

Continuing the previous example, the UCC-CP wants to add a proxied device to the "jeffrey@mycloud.org" *Cloud Account*. Similarly to its previous activation of the Administration User *Identity*, it invokes an "Jeffrey@mycloud.org" *User Identity* log in with the actions *GetUserLoginChallenge()* and *UserLogin()* using the credentials and log in information it used to establish the account with *AddUCSAccount()* as a CloudProxy control point. Once the "jeffrey@mycloud.org" log in is successful, the *AddProxyDevice()* action is invoked as follows:

```
Request (SOAP):
AddProxyDevice(
   "ucsuser01","dev02"
)
```

and

```
Response (SOAP):
AddProxyDevice(
   "jeffrey@mycloud.org/urn:schemas-upnp-org:device:MediaServer:1:uuid:
   9d0e8850-80ca-e8f5-4123-a50ee7225886"
)
```

Similarly, the *ProxyList,* and *CloudProxyUpdate* state variables are subsequently updated and the *CloudProxyUpdate* state variable evented as follows:

```
<ProxyList xmlns="urn:schemas-upnp-org:cloud:proxylist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:proxylist
      http://www.upnp.org/schemas/cloud/proxylist.xsd">
   <Proxy ProxId="pro00">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-org:device:CloudProxy:1:uuid:
         88509d0e-e8f5-80ca-4123-225886a50ee7
      </DeviceJID>
      <Device DevId="dev00">
         <connected>no</connected>
         <DeviceType>urn:schemas-upnp-org:device:CloudProxy:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>1</DeviceProtStatus>
      </Device>
   </Proxy>
   <Proxy ProxId="pro01">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-
         org:device:MediaServer:4:uuid:e70e9d0e-d9eb-4748-b163-636a323e7950
      </DeviceJID>
      <Device DevId="dev01">
         <connected>1</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>0</DeviceProtStatus>
      </Device>
   </Proxy>
   <Proxy ProxId="pro02">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-
         org:device:MediaServer:1:uuid:9d0e8850-80ca-e8f5-4123-a50ee7225886
      </DeviceJID>
      <Device DevId="dev02">
         <connected>1</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:1</DeviceType>
         <DeviceProt>0</DeviceProt>
         <DeviceProtStatus>0</DeviceProtStatus>
      </Device>
   </Proxy>
</ProxyList>
```

and

```
<?xml version="1.0" encoding="UTF-8"?>
<CloudProxyUpdate xmlns="urn:schemas-upnp-org:cloud:cloudproxyupdate"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:cloudproxyupdate
      http://www.upnp.org/schemas/cloud/ cloudproxyupdate.xsd">
   <ProxyUpdateId>2</ProxyUpdateId>
   <DeviceUpdateId>2</DeviceUpdateId>
   <UCSUpdateId>1</UCSUpdateId>
</CloudProxyUpdate>
```

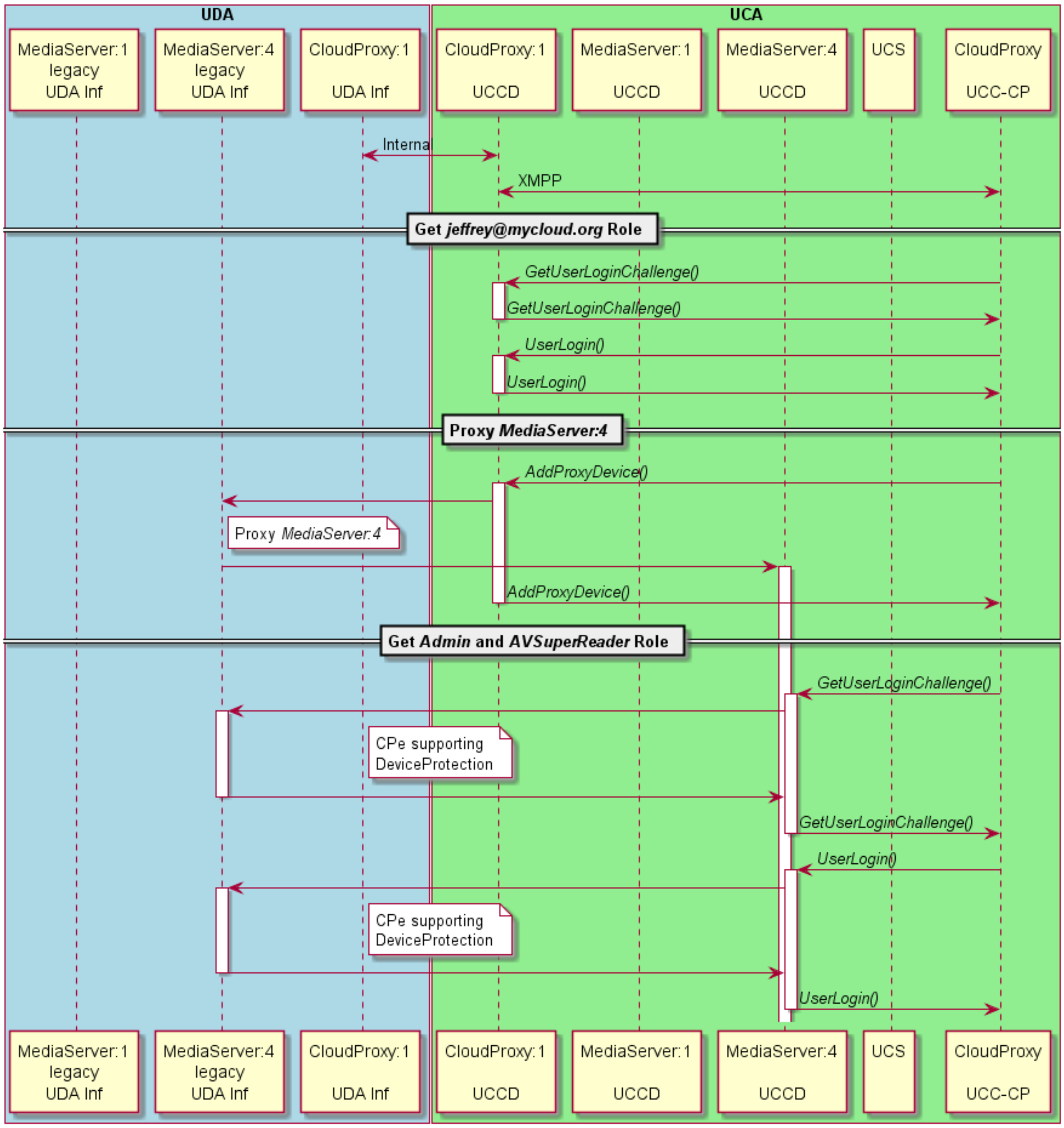
Finally, the UCC-CP issues a log in to the MediaServer:4 device for the `Administration` *User Identity* which includes both the *Admin* and *AVSuperReader Roles*. The log in is relayed through the embedded control point. Upon success, the *ProxyList* (to show the change in status of the *<DeviceProtStatus>* element to "1") and *CloudUpdateProxy* state variables are updated and the *CloudProxyUpdate* state variable evented as follows:

```
<ProxyList xmlns="urn:schemas-upnp-org:cloud:proxylist"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:proxylist
      http://www.upnp.org/schemas/cloud/proxylist.xsd">
   <Proxy ProxId="pro00">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-org:device:CloudProxy:1:uuid:
         88509d0e-e8f5-80ca-4123-225886a50ee7
      </DeviceJID>
      <Device DevId="dev00">
         <connected>no</connected>
         <DeviceType>urn:schemas-upnp-org:device:CloudProxy:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>1</DeviceProtStatus>
      </Device>
   </Proxy>
   <Proxy ProxId="pro01">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-
         org:device:MediaServer:4:uuid:e70e9d0e-d9eb-4748-b163-636a323e7950
      </DeviceJID>
      <Device DevId="dev01">
         <connected>1</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:1</DeviceType>
         <DeviceProt>1</DeviceProt>
         <DeviceProtStatus>1</DeviceProtStatus>
      </Device>
   </Proxy>
   <Proxy ProxId="pro02">
      <DeviceJID>
         jeffrey@mycloud.org/urn:schemas-upnp-
         org:device:MediaServer:1:uuid:9d0e8850-80ca-e8f5-4123-a50ee7225886
      </DeviceJID>
      <Device DevId="dev02">
         <connected>1</connected>
         <DeviceType>urn:schemas-upnp-org:device:MediaServer:1</DeviceType>
         <DeviceProt>0</DeviceProt>
         <DeviceProtStatus>0</DeviceProtStatus>
      </Device>
   </Proxy>
</ProxyList>
```

and

```
<?xml version="1.0" encoding="UTF-8"?>
<CloudProxyUpdate xmlns="urn:schemas-upnp-org:cloud:cloudproxyupdate"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:cloud:cloudproxyupdate
      http://www.upnp.org/schemas/cloud/ cloudproxyupdate.xsd">
   <ProxyUpdateId>3</ProxyUpdateId>
   <DeviceUpdateId>2</DeviceUpdateId>
   <UCSUpdateId>1</UCSUpdateId>
</CloudProxyUpdate>
```

Figure 7-5 Proxying with DeviceProtection



8 XML SERVICE DESCRIPTION

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
    <specVersion>
        <major>1</major>
        <minor>0</minor>
    </specVersion>
    <actionList>
        <action>
            <name>AddUCSAccount</name>
            <argumentList>
                <argument>
                    <name>User@Cloud</name>
                    <direction>in</direction>
```

```
         <relatedStateVariable>
            A_ARG_TYPE_BareJID
         </relatedStateVariable>
      </argument>
      <argument>
         <name>Port</name>
         <direction>in</direction>
         <relatedStateVariable>
            A_ARG_TYPE_Port
         </relatedStateVariable>
      </argument>
      <argument>
         <name>UCSJID</name>
         <direction>out</direction>
         <relatedStateVariable>
            A_ARG_TYPE_BareJID
         </relatedStateVariable>
      </argument>
      <argument>
         <name>Password</name>
         <direction>out</direction>
         <relatedStateVariable>
            A_ARG_TYPE_String
         </relatedStateVariable>
      </argument>
   </argumentList>
</action>
<action>
   <name>AddProxyDevice</name>
   <argumentList>
      <argument>
         <name>DeviceId</name>
         <direction>in</direction>
         <relatedStateVariable>
            A_ARG_TYPE_DeviceId
         </relatedStateVariable>
      </argument>
      <argument>
         <name>User@Cloud</name>
         <direction>in</direction>
         <relatedStateVariable>
            A_ARG_TYPE_BareJID
         </relatedStateVariable>
      </argument>
      <argument>
         <name>DeviceJID</name>
         <direction>out</direction>
         <relatedStateVariable>
            A_ARG_TYPE_FullJID
         </relatedStateVariable>
      </argument>
   </argumentList>
</action>
<action>
   <name>DeleteUCSAccount</name>
   <argumentList>
      <argument>
         <name>BareJID</name>
         <direction>in</direction>
         <relatedStateVariable>
            A_ARG_TYPE_BareJID
         </relatedStateVariable>
      </argument>
   </argumentList>
</action>
<action>
   <name>DeleteProxyDevice</name>
   <argumentList>
```

```
            <argument>
               <name>DeviceJID</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_FullJID
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>GetDeviceList</name>
         <argumentList>
            <argument>
               <name>DeviceList</name>
               <direction>out</direction>
               <relatedStateVariable>
                  DeviceList
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>GetProxyList</name>
         <argumentList>
            <argument>
               <name>ProxyList</name>
               <direction>out</direction>
               <relatedStateVariable>
                  ProxyList
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>GetUCSList</name>
         <argumentList>
            <argument>
               <name>UCSList</name>
               <direction>out</direction>
               <relatedStateVariable>
                  UCSList
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
   </actionList>
   <serviceStateTable>
      <stateVariable sendEvents="yes">
         <name>CloudProxyUpdate</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>DeviceList</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>ProxyList</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>UCSList</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_DevId</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
```

```
            <name>A_ARG_TYPE_BareJID</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_FullJID</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_String</name>
            <dataType>string</dataType>
        </stateVariable>
        <stateVariable sendEvents="no">
            <name>A_ARG_TYPE_Port</name>
            <dataType>ui2</dataType>
        </stateVariable>
    </serviceStateTable>
</scpd>
```