BluStar Presence Server SIMPLE Interface V 3.2.1

SEPTEMBER 2015

TECHNICAL DESCRIPTION

Mitel®

DOCUMENT HISTORY			
Version	Date	Comments	Author
01.01/EN	20/12/12	Initial version (based on BluStar_Presence XML Document 29/07/11)	Bernd Greiner
01.02/EN	19/06/13	CONSOLIDATING PRESENCE INFORMATION	Bernd Greiner
01.03/EN	10/07/13	Add some notices for subscribe, publish	Bernd Greiner
01.04/EN	11/07/13	Added contact header for PUBLISH	Steffen Schmoll
01.05/EN	18/07/13	Chapter “Consolidating presence information” removed Additional information for XML presence document added SBC support for PUBLISH and SUBSCRIBE using Route Header	Steffen Schmoll
01.06/EN	29/08/13	Server Presence Handling added SIP ETag and SIP If Match Handling added	Steffen Schmoll
01.07/EN	10/09/13	Description of AD credential transport added for authenticating users | copyright notice changed to Aastra CH	Schmoll / Stephan
01.08/EN	17/08/13	2.2 / 4 corrected, 2.8 / 2.9 updated / 2.10 added	Schmoll / Stephan
01.09/EN	07/01/15	External Presence handling added	Steffen Schmoll
01.10/EN	27/01/15	Lync Forwarding via PUBLISH	Steffen Schmoll
01.11/EN	03/03/15	Mitel Rebranding	Andre Bitter
01.12/EN	06/07/15	Lync telephoneEvent added	Steffen Schmoll
01.13/EN	10/09/15	Calendar ID updated as element in <aa.xsd>	Steffen Schmoll

Contents

Introduction

1.1. Involved RFCs

RFC 3863: Presence Information Data Format (PIDF)
RFC 4479: A Data Model for Presence
RFC 4480: RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)
RFC 5367: Subscriptions to Request-Contained Resource Lists in SIP

1.2. Abbreviations used

AMC Aastra Mobile Client
AMCC Aastra Mobile Client Controller
BSS BluStar Server
BSC BluStar client (Aastra 8000i, BluStar PC client, BluStar iOS client …)

1.3. Proprietary Aastra XML presence element with basic content

(See aa.xsd)

```
<aa:presence>
  <aa:blustar>
  <!--contains blustar presence information-->
    <aa:status>available</aa:status>
    <aa:note xml:lang="Eng">I'm here</aa:note>
    <aa:timestamp>2013-07-11T11:49:29Z</aa:timestamp>
  </aa:blustar>
  <aa:calendar>
   <!--contains calendar presence information-->
   <aa:id>123456789</aa:id>
   <aa:state>busy</aa:state>
   <aa:subject>Blustar Study</aa:subject>
   <aa:location></aa:location>
   <aa:from>2013-07-11T10:00:00Z</aa:from>
   <aa:end>2013-07-11T12:00:00Z</aa:end>
   <aa:allday/>
   <aa:private/>
   <aa:timestamp>2013-07-11T10:10:00Z </aa:timestamp>
  </aa:calendar>
</aa:presence>
```

.

2. BluStar Server Presence Handling

2.1. Start of a client – new client connection

- Server receives SUBSCRIBE for the own (and in case of PC Presence) additionally for the favorites in a resource list.
- Server connects to LDAP for retrieving an entry based on the address in the <From> Header (or in case of PC Presence based on the entries on the resource list).
  - If the <From> Header or the entries in the resource list looks like: "From: <sip:peter.test@aastra.com:5060>" the Presence Server searches for "peter.test@aastra.com" in the defined "<SIP:>"search column.
  - If the <From> Header or the entries in the resource list looks like: "tel:200;..." the Presence Server searches for "200" in the defined "<Tel:>" search column.
- If no entry can be found the server sends a NOTIFY with "unknown" presence document.
- If an entry was found then the server "subscribes" itself based on available values of device and mail address line state and calendar state. As soon as line state and calendar state is available from the subsystems the server updates the "internal" presence documents and sends additional NOTIFY's with the consolidated presence document to the subscriptions.

2.2. Regular termination of a client

- A) Server receives SUBSCRIBE with "Expires: 0" for all subscriptions.
- B) Server receives PUBLISH with "Expires: 0"
- Server releases all subscriptions for that user.
- Server checks if this was the last client using this subscription of the respective user.
- If it was the last then the server sends NOTIFY ("Signed Out") to all entities having subscribed the presence state of this user.

2.3. Unintended client termination

- Server releases all subscriptions for that user.
- Server checks if this was the last client using this <Contact> Header.
- If yes, server updates BluStar presence state ("Signed Out") for that user.
- If yes, server updates "internal" presence document for that user.
- If yes, server sends NOTIFY with consolidated presence document to all subscribers.

2.4. Receive a status update (PUBLISH) from a client

- See chapter 2.10 for details about the authentication. After the PUBLISH request was accepted by the BluStar Server it sends "200 OK" including the <SIP-ETag> Header (newly created if that's the first PUBLISH, otherwise with the already existing one)
- Server updates "internal" presence document for this user.
- Depending of the provided <SIP-If-Match> Header information the Server ignores the first PUBLISH of a client to prevent a wrong state on client startup (see below for details)
- Server sends NOTIFY with consolidated presence document to all subscribers of the status of the user.

2.5. Receive a line state change from a PBX

- Server updates “internal” presence document for that user.
- Server sends NOTIFY with consolidated presence document to all subscribers of this user’s presence.

2.6. Receive a status update from a Calendar system like MS Exchange

- Server updates “internal” presence document for that user.
- Server sends NOTIFY with consolidated presence document to all subscribers.

2.7. Receive a status update from a presence system like MS Lync

- Server updates “internal” presence document for that user.
- Server sends NOTIFY with consolidated presence document to all subscribers.

2.8. Refresh PUBLISH

- Server receives PUBLISH with “Expires > 0” and Content-Length: 0
- Server sends 200 OK for PUBLISH with existing <SIP-ETag>

2.9. Refresh SUBSCRIBE

- Server receives SUBSCRIBE with “Expires > 0”
- Server sends 200 OK for SUBSCRIBE
- If the BluStar Server identifies an AMCC sending the re-SUBSCRIBE for an AMC it will send presence information (NOTIFY) for the users subscribed by the client to support on / off network conditions of an AMC (which is invisible to the BluStar Server and the AMCC does not contain any cache for presence states to avoid inconsistencies).

2.10. User authentication

The BluStar Server does not have any information about the AD-credentials of the user. This keeps the BluStar Server free from any user-specific configuration.

The user must provide the AD-credentials once to the BluStar client; the AD-credentials will be used to authenticate the user for the respective session.

- The AD-password will be used for presence only
- The AD-password will be stored in encrypted format on the HDD / the file system of the BluStar client
- The BluStar Server can be configured to allow PUBLISH and SUBSCRIBE only if the clients support authentication; this will be a global setting for all clients. The BluStar Server will identify the clients reading the SIP UA string to allow i.e. running BSCpc 3.0.0 and BSCpc 3.0.1 in one installation
  Please note: This setting is likely the default setting after the installation of the BluStar Server.
- BluStar clients must be prepared to perform the authentication for both, SUBSCRIBE and PUBLISH methods.

Description of the authentication procedure:

- User starts the BluStar client
- The BluStar client will try to SUBSCRIBE for its own state
  - Even if the BluStar Server is configured to accept SUBSCRIBE without authentication it will check the UA string to figure out if the client supports authentication; if it does the BluStar Server will force the client to perform the authentication, otherwise it will just accept the SUBSCRIBE.
  - If the BluStar Server is configured to accept SUBSCRIBE with authentication only it will just send “401” for SUBSCRIBE requests without “auth”
  - The BluStar Server will refuse the SUBSCRIBE with “401”. The message will contain “a dynamic” value (56 characters) individually created for every BluStar client in a “X-BSS-auth” tag which is used for encrypting the AD credentials.

- The BluStar client will encrypt the AD-credentials with the same algorithm used for the configuration file encryption and repeat the SUBSCRIBE using the “X-BSS-rauth” tag which carries the encrypted credentials
- The AD-credentials are concatenated to a string before being handed over to the encrypting entity in the form “domain\username:password”, the resulting string will be base-64 encoded and passed to the “X-BSS-rauth” tag of a SUBSCRIBE or PUBLISH message
- The key used for the encryption is transferred from the BluStar Server to the BluStar client only; it is not transferred from the client to the BluStar Server. The BluStar Server stores the individual key sent to the BluStar client in the last “401” message.
- The BluStar client must be prepared to accept key changes at any time triggered by another “401” message (in case the BluStar client supports authentication at all)
- The BluStar Server will decrypt the credentials, hand them over to the AD to validate domain/username and password is correct and will grant the access to the presence service if the answer is positive. If the check leads to a bad result the BluStar Server will send “401” again which will cause i.e. the BSCpc to open a message box asking for the AD-password again.
- The BluStar Server will not store the credentials.

Here is an example:

1st Request without authentication

```
…
SUBSCRIBE sip:10.10.10.10:5060 SIP/2.0
Via: SIP/2.0/UDP 20.20.20.20;branch=z9hG4bK94ac96337c68e6b54;alias;rport=5060
Route: <sip:10.10.10.10:5060;lr;transport=UDP>
Max-Forwards: 70
From: <sip:67251@aastra.com:5060>;tag=1ec207a9de
To: <sip:bill@aastra.com:5060>
Call-ID: 9112b4907352d044
CSeq: 11940 SUBSCRIBE
Contact: <sip:bill@20.20.20.20:5060;transport=UDP>
Event: presence
Expires: 300
Require: recipient-list-subscribe
User-Agent: Aastra BluStar Client/3.0.1.0
Content-Type: application/resource-lists+xml
Content-Length: 337
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@aastra.com" />
    <entry uri="sip:joe@example.org" />
    <entry uri="sip:ted@example.net" />
  </list>
</resource-lists>
```

Response / 401:

```
2013-09-16 16:11:05,334 DEBUG [PipelineThread-7] (BlustarSipCommunicationHandler.java:743) -
handleProcessSubscribeList() -> sendResponse
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 20.20.20.20;branch=z9hG4bK94ac96337c68e6b54;alias;rport=5060
From: <sip:67251@aastra.com:5060>;tag=bab8088c77
To: <sip:bill@aastra.com:5060>
Call-ID: 9112b4907352d044
CSeq: 11940 SUBSCRIBE
Contact: <sip:10.10.10.10:5060>
Expires: 300
X-BSS-auth: 123449940550046629849940559529123449940564666562984994056
Content-Length: 0
```

2nd Request with authentication

```
…
SUBSCRIBE sip:10.10.10.10:5060 SIP/2.0
Via: SIP/2.0/UDP 20.20.20.20;branch=z9hG4bKed70a7e1b9874b30d;alias;rport=5060
Route: <sip:10.10.10.10:5060;lr;transport=UDP>
```

```
Max-Forwards: 70
From: <sip:67251@aastra.com:5060>;tag=4e53d86f7e
To: <sip:bill@aastra.com:5060>
Call-ID: aa6e9dcc8708b62f
CSeq: 29899 SUBSCRIBE
Contact: <sip:bill@20.20.20.20:5060;transport=UDP>
Event: presence
Expires: 300
Require: recipient-list-subscribe
User-Agent: Aastra BluStar Client/3.0.1.0
X-BSS-rauth: VUJDf4D4XJgP/K8mntoWjQaJJYMr3NgESA41234+ctqYal3LLyZS6IoStfgJU1cH25xSkdJryoqeb
7nlme21234dc+w5w0yB1w==
Content-Type: application/resource-lists+xml
Content-Length: 337
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@aastra.com" />
    <entry uri="sip:joe@example.org" />
    <entry uri="sip:ted@example.net" />
  </list>
</resource-lists>
```

2.11. Authentication and operation of the AMCC

To avoid forcing the admin to create an AD-account for the sole purpose of allowing the AMCC to authenticate itself (which may cause issues if this account – by group policy – would require frequent password changes which must be “mirrored” on the AMCC) the credentials used by the AMCC are simple USERNAME / PASSWORD, validated against an account which must be created on the BSS.

The mechanism for authentication from an AMCC to a BSS is the same as described in chapter 2.10 but the AMCC represents all FMC users leveraging presence service from the BSS.

The AMCC sort of hides the AMCs from the BSS at if uses one IP-address to connect to the BSS for all AMCs (it may use on connection / port for all AMC-related packets, too).
The “Contact” used by the AMCC is “fmc@<ip>:<port>”; the FROM header will be different per client allowing the BSS to separate between different AMC-“sessions”.

3. Presence XML document

The following XML presence document is supported:

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
 xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
 xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
 xmlns:aa="http://www.aastra.com/presence/blustar/2011"
 entity="sip:someone@example.com">
  <tuple id="global service">
    <status>
      <basic>open</basic>
    </status
  </tuple>
  <!-- Audio service representing the capability to have voice communication-->
  <tuple id="audio">
    <status>
      <!-- open when audio can be used, closed otherwise-->
      <basic>closed</basic>
    </status
  </tuple>
  <!-- Video service representing the capability to have video communication-->
  <tuple id="video">
    <status>
      <!-- open when video can be used, closed otherwise-->
      <basic>open</basic>
    </status>
  </tuple>
  <!-- Message service representing the capability to have chat communication-->
  <tuple id="message">
    <status>
      <!-- open when chat can be used, closed otherwise-->
      <basic>open</basic>
    </status>
  </tuple>
  <dm:device id="tel:6894">
    <aa:line>
      <!--contains telephony device presence information-->
      <aa:DN>6894</aa:DN>
      <aa:state>idle</aa:state>
    </aa:line>
  </dm:device>
  <dm:device id="tel:7895">
    <aa:line>
      <!--contains telephony device presence information-->
      <aa:DN>7895</aa:DN>
      <aa:state>idle</aa:state>
      <aa:forward type="immediate">6894</aa:forward>
    </aa:line>
  </dm:device>
  <dm:person id="sip:someone@example.com">
    <rpid:activities>
      <rpid:note xml:lang="Eng">In a meeting</rpid:note>
      <!-- global status consolidated from blustar presence, calendar, external presence and line state
information -->
      <rpid:busy/>
    </rpid:activities>
    <aa:presence>
      <aa:blustar>
      <!--contains blustar presence information-->
        <aa:status>available</aa:status>
        <aa:note xml:lang=”ENG” />
         <aa:timestamp>2011-04-04T16:49:29Z</aa:timestamp>
      </aa:blustar>
      <aa:calendar>
        <!--contains calendar presence information-->
        <aa:id> </aa:id>
        <aa:state>busy</aa:state>
        <aa:subject>Blustar Study</aa:subject>
        <aa:location></aa:location>
        <aa:from>2011-04-04T10:00:00Z</aa:from>
        <aa:end>2011-04-04T12:00:00Z</aa:end>
        <aa:allday/>
        <aa:private/>
        <aa:timestamp>2011-04-04T16:49:29Z</aa:timestamp>
      </aa:calendar>
```

```
      <aa:externalUC>
        <!--contains external presence information-->
        <aa:status>available</aa:status>
        <aa:freeText>Working on BluStar 3.2</aa:freeText>
        <aa:fwdDestination>sip:someoneelse@example.com</aa:fwdDestination>
        <aa:telephoneEvent>on-the-phone</aa:telephoneEvent>
        <aa:timestamp>2011-04-04T16:27:14Z</aa:timestamp>
      </aa:externalUC>
    </aa:presence>
    <dm:timestamp>2011-04-04T16:49:29Z</dm:timestamp>
  </dm:person>
</presence>
```

3.1. Devices

```
<dm:device id="tel:7895">
  <aa:line>
    <!--contains telephony device presence information-->
    <aa:DN>7895</aa:DN>
    <aa:state>idle</aa:state>
    <aa:forward type="immediate">6894</aa:forward>
  </aa:line>
</dm:device>
```

The XML presence document contains a <device> XML element for each device associated to the presentity and supervised by the Presence Server.

This <device> XML element contains:

- <deviceID> XML elements
- <line> proprietary XML element that contains:
- <DN> element identifying the directory number of the supervised device
- <state> element giving the line status (idle, ringing, busy, unknown)
- <forward> element indicating if the device is forwarded - when present, it indicates the forward destination and forward type

3.2. Person

The <person> XML element is composed of two parts:

- The standard RPID part contains the consolidated status - so this is what the client uses to define its display etc.
  This part mainly contains RPID XML elements with the consolidated presence status. Possible values for the consolidated presence status:
  <rpid:other>available</rpid:other> corresponds to <aa:status>available</aa:status>
  <rpid:busy/> corresponds to <aa:status>busy</aa:status>
  <rpid:other>unavailable</rpid:other> corresponds to <aa:status>do not disturb</aa:status>
  <rpid:other>signed out</rpid:other> corresponds to <aa:status>signed out</aa:status>
  <rpid:other>unknown</rpid:other> corresponds to <aa:status>unknown</aa:status>
  The text of <rpid:note> contains either the free text published by the user or – if not set by the user – the following text (consolidated from BluStar status, appointments and line state): Available, Busy, Unavailable, Signed out, In a meeting, In a call
- The proprietary AA part contains the source information i.e. the last status received from a BluStar device and the latest calendar / external presence information (if relevant).
  This part contains the <aa:presence> proprietary element.
  The <aa:presence> is the root proprietary XML element under which different pieces of information related to presence are placed:
  - <blustar> XML element containing the BluStar presence: it contains the BluStar status (<status>) and an optional note element containing free text provided by the user (<note>).
    Possible values for <aa:status>: available, busy, do not disturb, signed out, unknown

- <calendar> XML element contains presence information extracted from a calendar system like MS Exchange
- <externalUC> XML element contains external presence information extracted from a presence system like MS Lync

<blustar> presence element

```
<aa:blustar>
  <!--contains blustar presence information-->
  <aa:status>available</aa:status>
  <aa:note xml:lang="ENG" />
   <aa:timestamp>2011-04-04T16:49:29Z</aa:timestamp>
</aa:blustar>
```

This <blustar> XML element contains the presence details of the corresponding BluStar client:

- <status> element giving the BluStar status
- <note> element giving the BluStar free text

<calendar> presence element

```
<aa:calendar>
  <!--contains calendar presence information-->
  <aa:id> </aa:id>
  <aa:state>busy</aa:state>
  <aa:subject>Blustar Study</aa:subject>
  <aa:location></aa:location>
  <aa:from>2011-04-04T10:00:00Z</aa:from>
  <aa:end>2011-04-04T12:00:00Z</aa:end>
  <aa:allday/>
  <aa:private/>
  <aa:timestamp>2011-04-04T16:49:29Z</aa:timestamp>
</aa:calendar>
```

This <blustar> XML element contains the calendar details of a connected calendar system.

<externalUC> presence element

```
<aa:externalUC>
  <!--contains external presence information-->
  <aa:status>available</aa:status>
  <aa:freeText>Working on BluStar 3.2</aa:freeText>
  <aa:fwdDestination>sip:someoneelse@example.com</aa:fwdDestination>
  <aa:telephoneEvent>on-the-phone</aa:telephoneEvent>
  <aa:timestamp>2011-04-04T16:27:14Z</aa:timestamp>
</aa:externalUC>
```

This <blustar> XML element contains the presence details of a connected additional presence system.

4. Examples

4.1. SUBSCRIBE

Request:

```
SUBSCRIBE sip:10.103.230.130:5060 SIP/2.0
Via: SIP/2.0/UDP 10.103.228.151;branch=z9hG4bKae8cac0a52b206371;alias
Max-Forwards: 70
Route: <sip:2.2.2.2:5060;lr>
From: <sip:67251@aastra.com:5060>;tag=1ec207a9de
To: <sip:tom.hanks@aastra.com:5060>;tag=70f4bd96
Call-ID: 4b696543bb91478d
CSeq: 18468 SUBSCRIBE
Contact: <sip:67251@10.103.228.151:5060>
Event: presence
Expires: 300
User-Agent: Aastra BluStar Client/3.0.0.0
Content-Length: 0
```

Response:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.103.228.151;branch=z9hG4bKae8cac0a52b206371;alias
From: <sip:67251@aastra.com:5060>;tag=1ec207a9de
To: <sip:tom.hanks@aastra.com:5060>;tag=70f4bd96
Call-ID: 4b696543bb91478d
CSeq: 18468 SUBSCRIBE
Contact: <sip:10.103.230.130:5060>
Expires: 300
Content-Length: 0
```

Notes:

- On client start up the first subscription has to be the own address
  The Presence Server needs this information to identify the client connection correctly for parallel clients using the same SIP Uri.
- The Route header can be used in a SBC scenario

4.2. SUBSCRIBE CONTAINED LISTS

Request:

```
SUBSCRIBE sip:10.103.230.130:5060 SIP/2.0
Via: SIP/2.0/UDP 10.103.228.151;branch=z9hG4bKae8cac0a52b206371;alias
Max-Forwards: 70
Route: <sip:2.2.2.2:5060;lr>
From: <sip:67251@aastra.com:5060>;tag=1ec207a9de
To: <sip:rls@aastra.com:5060>;tag=70f4bd96
Call-ID: 4b696543bb91478d
CSeq: 18468 SUBSCRIBE
Contact: <sip:67251@10.103.228.151:5060>
Event: presence
Expires: 7200
Require: recipient-list-subscribe
Content-Type: application/resource-lists+xml
Content-Disposition: recipient-list
Content-Length: 337

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <list>
    <entry uri="sip:bill@example.com" />
    <entry uri="sip:joe@example.org" />
    <entry uri="sip:ted@example.net" />
  </list>
 </resource-lists>
```

Response:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.103.228.151;branch=z9hG4bKae8cac0a52b206371;alias
From: <sip:67251@aastra.com:5060>;tag=1ec207a9de
To: <sip:rls@aastra.com:5060>;tag=70f4bd96
Call-ID: 4b696543bb91478d
CSeq: 18468 SUBSCRIBE
Contact: <sip:10.103.230.130:5060>
Expires: 300
Content-Length: 0
```

Notes:

- On client start up the first entry in the subscription list has to be the own address
  The Presence Server needs this information to identify the client connection correctly for parallel clients using the same SIP Uri.
- The Route header can be used in a SBC scenario
- The entries in the subscription can either be in the SIP format sip:peter.test@domain.com or in the TEL format tel:12345

4.3. PUBLISH

Request:

```
PUBLISH sip:tom.hanks@10.103.228.151:5060 SIP/2.0
Authorization: Basic realm="",uri="sip: 10.103.228.151:5060",username="de\ptest:h5l%3k5§k3h"
Accept: application/dialog-info+xml
Via: SIP/2.0/UDP
10.103.228.151;branch=z9hG4bK4083594629d0e7128.3d8571b9502833944.c4e8d546ce288974b;alias
Route: <sip:10.103.230.130:5060;lr>
Max-Forwards: 70
Route: <sip:2.2.2.2:5060;lr>
From: <sip:tom.hanks@aastra.com:5060>;tag=93521f1ee5
To: <sip:tom.hanks@aastra.com:5060>
Call-ID: 2a0864bd7d9593d3
CSeq: 15729 PUBLISH
Contact: <sip:67251@10.103.228.151:5060>
Event: presence
Expires: 3600
SIP-If-Match: 18762
User-Agent: Aastra BluStar Client/3.0.0.0
Content-Type: application/pidf+xml
Content-Length: 864

<?xml version="1.0" encoding="utf-8"?>
<presence xmlns:aa="http://www.aastra.com/presence/blustar/2011"
xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid" xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
entity="sip:bill@example.com" xmlns="urn:ietf:params:xml:ns:pidf">
  <tuple id="audio">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <tuple id="video">
    <status>
      <basic>closed</basic>
    </status>
  </tuple>
  <tuple id="message">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <dm:person id="sip:bill@example.com">
    <aa:presence>
      <aa:blustar>
        <aa:status>busy</aa:status>
        <aa:note xml:lang="Eng">HELLO</aa:note>
     <!--
      NOTICE: when there is no note (or a previous note is to be deleted) it MUST to look like:
      <aa:note xml:lang="Eng" />
     -->
        <aa:timestamp>2012-12-19T16:49:08Z</aa:timestamp>
      </aa:blustar>
    </aa:presence>
  </dm:person>
</presence>
```

Response:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
10.103.228.151;branch=z9hG4bK4083594629d0e7128.3d8571b9502833944.c4e8d546ce288974b;alias
From: <sip:tom.hanks@aastra.com:5060>;tag=93521f1ee5
To: <sip:tom.hanks@aastra.com:5060>
Call-ID: 2a0864bd7d9593d3
CSeq: 15729 PUBLISH
Contact: <sip:10.103.230.130:5060>
Expires: 3600
SIP-ETag: 18762
Content-Length: 0
```

Notes:

- The following values are supported for <aa:status>:
  - available
  - busy
  - do not disturb
  - signed out
- For changing the status text the current status has also to be repeated in the PUBLISH request, even when there is no change of status intended
- The EXPIRES and the CONTACT headers are mandatory
- The LANGUAGE TAG in the BluStar note is mandatory although the value is not used by the Presence Server yet, therefore it can be fixed xml:lang="Eng":
  - e.g. an empty note: <aa:note xml:lang="Eng" />
  - e.g. a non-empty note: <aa:note xml:lang="Eng">HELLO</aa:note>
- When a client terminates it doesn't have to send a publish request with "Signed out", the client may send a publish request with Expires: 0
- The Route header can be used in SBC scenarios

Special PUBLISH for MS LYNC forwarding:

```
PUBLISH sip:tom.hanks@10.103.228.151:5060 SIP/2.0
Authorization: Basic realm="",uri="sip: 10.103.228.151:5060",username="de\ptest:h5l%3k5$k3h"
Accept: application/dialog-info+xml
Via: SIP/2.0/UDP
10.103.228.151;branch=z9hG4bK4083594629d0e7128.3d8571b9502833944.c4e8d546ce288974b;alias
Route: <sip:10.103.230.130:5060;lr>
Max-Forwards: 70
Route: <sip:2.2.2.2:5060;lr>
From: <sip:tom.hanks@aastra.com>
To: <sip:sean.connery@aastra.com>
Call-ID: 2a0864bd7d9593d3
CSeq: 15729 PUBLISH
Contact: <sip:67251@10.103.228.151:5060>
Event: lyncforwarding
Expires: 3600
SIP-If-Match: 18762
User-Agent: Aastra BluStar Client/3.0.0.0
Content-Type: application/pidf+xml
Content-Length: 0
```

4.4. NOTIFY

Presence event based on a BluStar status update (publish)

```
NOTIFY sip:67251@10.103.228.151:5060 SIP/2.0
Via: SIP/2.0/UDP 10.103.230.130,SIP/2.0/UDP 10.103.230.130:5060
CSeq: 7 NOTIFY
To: <sip:67251@aastra.com:5060>;tag=3b3da890af
Call-ID: 8cc08471fb350dcb
Max-Forwards: 70
Contact: <sip:10.103.230.130:5060;transport=udp>,<sip:10.103.230.130:5060>
From: <sip:tom.hanks@aastra.com:5060>
Event: presence
Content-Type: application/pidf+xml
Expires: 300
Subscription-State: active
Content-Length: 969

<?xml version="1.0" encoding="utf-8"?>
<presence entity="sip:tom.hanks@aastra.com" xmlns:aa="http://www.aastra.com/presence/blustar/2011"
xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid" xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns="urn:ietf:params:xml:ns:pidf">
  <tuple id="global service">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <tuple id="audio">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <tuple id="video">
    <status>
      <basic>closed</basic>
    </status>
  </tuple>
  <tuple id="message">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <dm:device id="tel:2719">
    <aa:line>
      <aa:DN>2719</aa:DN>
      <aa:state>idle</aa:state>
    </aa:line>
    <dm:timestamp>2012-12-19T16:30:08Z </dm:timestamp>
  </dm:device>
  <dm:person id="sip:bill@example.com">
    <rpid:activities>
      <rpid:note xml:lang="Eng">HELLO</rpid:note>
      <rpid:busy/>
    </rpid:activities>
    <aa:presence>
      <aa:blustar>
        <aa:status>busy</aa:status>
        <aa:note xml:lang="Eng">HELLO</aa:note>
        <aa:timestamp>2012-12-19T16:49:08Z</aa:timestamp>
      </aa:blustar>
    </aa:presence>
  </dm:person>
</presence>
```

Presence event based on a calendar event

```
NOTIFY sip:67251@10.103.228.151:5060 SIP/2.0
Via: SIP/2.0/UDP 10.103.230.130,SIP/2.0/UDP 10.103.230.130:5060
CSeq: 7 NOTIFY
To: <sip:67251@aastra.com:5060>;tag=3b3da890af
Call-ID: 8cc08471fb350dcb
Max-Forwards: 70
Contact: <sip:10.103.230.130:5060;transport=udp>,<sip:10.103.230.130:5060>
From: <sip:tom.hanks@aastra.com:5060>
Event: presence
Content-Type: application/pidf+xml
Expires: 300
Subscription-State: active
Content-Length: 969

<?xml version="1.0" encoding="utf-8"?>
<presence entity="sip:tom.hanks@aastra.com" xmlns:aa="http://www.aastra.com/presence/blustar/2011"
xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid" xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns="urn:ietf:params:xml:ns:pidf">
  <tuple id="global service">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <tuple id="audio">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <tuple id="video">
    <status>
      <basic>closed</basic>
    </status>
  </tuple>
  <tuple id="message">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <dm:device id="tel:2719">
    <aa:line>
      <aa:DN>2719</aa:DN>
      <aa:state>idle</aa:state>
    </aa:line>
    <dm:timestamp>2012-12-10T13:30:08Z </dm:timestamp>
  </dm:device>
  <dm:person id="sip:bill@example.com">
    <rpid:activities>
      <rpid:note xml:lang="Eng">In a meeting</rpid:note>
      <rpid:busy/>
    </rpid:activities>
    <aa:presence>
      <aa:blustar>
        <aa:status>available</aa:status>
        <aa:note xml:lang="Eng" />
        <aa:timestamp>2012-12-10T13:49:08Z</aa:timestamp>
      </aa:blustar>
      <aa:calendar>
        <aa:id>AQYAAAADYRyhCAAAAAVZvhcAAAAA</aa:id>
        <aa:state>busy</aa:state>
        <aa:subject>test</aa:subject>
        <aa:from>2012-12-10T14:00:00Z</aa:from>
        <aa:end>2012-12-10T16:00:00Z</aa:end>
        <aa:location/>
        <aa:timestamp>2012-12-10T14:00:00Z</aa:timestamp>
      </aa:calendar>
    </aa:presence>
  </dm:person>
</presence>
```

Presence event based on a line state event

```
NOTIFY sip:67251@10.103.228.151:5060 SIP/2.0
Via: SIP/2.0/UDP 10.103.230.130,SIP/2.0/UDP 10.103.230.130:5060
CSeq: 7 NOTIFY
To: <sip:67251@aastra.com:5060>;tag=3b3da890af
Call-ID: 8cc08471fb350dcb
Max-Forwards: 70
Contact: <sip:10.103.230.130:5060;transport=udp>,<sip:10.103.230.130:5060>
From: <sip:tom.hanks@aastra.com:5060>
Event: presence
Content-Type: application/pidf+xml
Expires: 300
Subscription-State: active
Content-Length: 969

<?xml version="1.0" encoding="utf-8"?>
<presence entity="sip:tom.hanks@aastra.com" xmlns:aa="http://www.aastra.com/presence/blustar/2011"
xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid" xmlns:dm="urn:ietf:params:xml:ns:pidf:data-model"
xmlns="urn:ietf:params:xml:ns:pidf">
  <tuple id="global service">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <tuple id="audio">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
  <tuple id="video">
    <status>
      <basic>closed</basic>
    </status>
  </tuple>
  <tuple id="message">
    <status>
      <basic>open</basic>
    </status>
  </tuple>
<dm:device id="tel:2719">
    <aa:line>
      <aa:DN>2719</aa:DN>
      <aa:state>busy</aa:state>
    </aa:line>
    <dm:timestamp>2012-12-10T13:30:08Z </dm:timestamp>
  </dm:device>
  <dm:person id="sip:bill@example.com">
    <rpid:activities>
      <rpid:note xml:lang="Eng">In a call</rpid:note>
      <rpid:busy/>
    </rpid:activities>
    <aa:presence>
      <aa:blustar>
        <aa:status>available</aa:status>
        <aa:note xml:lang="Eng" />
        <aa:timestamp>2012-12-10T13:19:05Z</aa:timestamp>
      </aa:blustar>
    </aa:presence>
  </dm:person>
</presence>
```

5. Appendix

5.1. Aastra.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
    xmlns="http://www.aastra.com/presence/blustar/2011"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
    targetNamespace="http://www.aastra.com/presence/blustar/2011"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

    <xs:include schemaLocation="common-schema.xsd"/>
    <xs:import namespace="urn:ietf:params:xml:ns:pidf:rpid" schemaLocation="rpid.xsd" />

    <xs:simpleType name="blustarStatus">
       <xs:restriction base="xs:string">
          <xs:enumeration value="available"/>
          <xs:enumeration value="busy"/>
          <xs:enumeration value="away"/>
          <xs:enumeration value="do not disturb"/>
          <xs:enumeration value="signed out"/>
          <xs:enumeration value="inactive"/>
          <xs:enumeration value="be right back"/>
          <xs:enumeration value="other"/>
          <xs:enumeration value="unknown"/>
       </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="blustar">
       <xs:sequence>
          <xs:element name="status">
             <xs:annotation>
                <xs:documentation>
                   BluStar client status.
                </xs:documentation>
             </xs:annotation>
             <xs:complexType>
                <xs:simpleContent>
                   <xs:extension base="blustarStatus">
                      <xs:attribute name="display" type="xs:string" use="optional"/>
                   </xs:extension>
                </xs:simpleContent>
             </xs:complexType>
          </xs:element>
          <xs:element name="note" type="Note_t" minOccurs="0"/>
          <xs:element name="timestamp" type="Timestamp_t" minOccurs="0"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attributeGroup ref="fromUntil"/>
       <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="A400">
       <xs:sequence>
          <xs:element ref="rpid:activities"/>
          <xs:element name="timestamp" type="Timestamp_t" minOccurs="0"/>
          <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attributeGroup ref="fromUntil"/>
       <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>

    <xs:complexType name="externalUC">
       <xs:sequence>
          <xs:element name="status">
             <xs:annotation>
                <xs:documentation>
                   UC status.
                </xs:documentation>
             </xs:annotation>
             <xs:simpleType>
                <xs:restriction base="xs:string">
                   <xs:enumeration value="available"/>
                   <xs:enumeration value="busy"/>
                   <xs:enumeration value="away"/>
                   <xs:enumeration value="do not disturb"/>
```

```
                  <xs:enumeration value="signed out"/>
                  <xs:enumeration value="inactive"/>
                  <xs:enumeration value="be right back"/>
                  <xs:enumeration value="unknown"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:element>
         <xs:element name="freeText" type="xs:string" />
         <xs:element name="fwdDestination" type="xs:string" />
         <xs:element name="telephoneEvent" type="xs:string" />
         <xs:element name="timestamp" type="Timestamp_t" minOccurs="0"/>
         <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attributeGroup ref="fromUntil"/>
      <xs:anyAttribute namespace="##any" processContents="lax"/>
   </xs:complexType>

   <xs:complexType name="calendar">
      <xs:sequence>
         <xs:element name="id" type="xs:string"/>
         <xs:element name="state">
            <xs:annotation>
               <xs:documentation>
                  Calendar state.
               </xs:documentation>
            </xs:annotation>
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:enumeration value="free"/>
                  <xs:enumeration value="tentative"/>
                  <xs:enumeration value="busy"/>
                  <xs:enumeration value="out of office"/>
                  <xs:enumeration value="unknown"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:element>
         <xs:element name="subject" type="xs:string"/>
         <xs:element name="location" type="xs:string"/>
         <xs:element name="from" type="Timestamp_t" />
         <xs:element name="end" type="Timestamp_t" />
         <xs:element name="allday" type="empty" minOccurs="0"/>
         <xs:element name="private" type="empty" minOccurs="0"/>
         <xs:element name="reminder-offset" type="xs:integer" minOccurs="0"/>
         <xs:element name="timestamp" type="Timestamp_t" minOccurs="0"/>
         <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>

<xs:anyAttribute namespace="##any" processContents="lax"/>
   </xs:complexType>

   <xs:element name="presence">
      <xs:annotation>
         <xs:documentation>
            Proprietary data collected from BluStar client, A400 and Presence Server.
            In the presence part, only UC and calendar information from Presence Server
            is taken into account. Line information will not appear in this presence tag
         </xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element name="blustar" type="blustar" minOccurs="0" maxOccurs="1"/>
            <xs:element name="A400" type="A400" minOccurs="0" maxOccurs="1"/>
            <xs:element name="externalUC" type="externalUC" minOccurs="0" maxOccurs="1"/>
            <xs:element name="calendar" type="calendar" minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>

   <xs:element name="line">
      <xs:annotation>
         <xs:documentation>
            CTI information collected from Presence Server.
            This XML element will be integrated under device element defined by Data Model
            Therefore, it is not integrated into the presence tag
         </xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
```

```
            <xs:element name="DN" type="xs:string"/>
            <xs:element name="state">
               <xs:annotation>
                  <xs:documentation>
                     Line state.
                  </xs:documentation>
               </xs:annotation>
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:enumeration value="idle"/>
                     <xs:enumeration value="ringing"/>
                     <xs:enumeration value="busy"/>
                     <xs:enumeration value="unknown"/>
                  </xs:restriction>
               </xs:simpleType>
            </xs:element>
            <xs:element name ="forward" minOccurs="0">
               <xs:annotation>
                  <xs:documentation>
                     define the forward element present in line element with its type attribute
                  </xs:documentation>
               </xs:annotation>
               <xs:complexType>
                  <xs:simpleContent>
                     <xs:extension base="xs:string">
                        <xs:attribute name="type" use="required">
                           <xs:simpleType>
                              <xs:restriction base="xs:string">
                                 <xs:enumeration value="immediate"/>
                              </xs:restriction>
                           </xs:simpleType>
                        </xs:attribute>
                     </xs:extension>
                  </xs:simpleContent>
               </xs:complexType>
            </xs:element>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```