



The Open Mobile SQM API

VERSION 4.0, JANUARY 2014

Corporate Headquarters
iPass Inc.
3800 Bridge Parkway
Redwood Shores, CA 94065 USA
www.ipass.com
+1 650-232-4100
+1 650-232-0227 fx

Copyright © 2014, iPass Inc. All rights reserved.

Trademarks

iPass, iPassConnect, ExpressConnect, iPassNet, RoamServer, NetServer, iPass Mobile Office, DeviceID, EPM, iSEEL, iPass Alliance, Open Mobile, and the iPass logo are trademarks of iPass Inc.

All other brand or product names are trademarks or registered trademarks of their respective companies.

Warranty

No part of this document may be reproduced, disclosed, electronically distributed, or used without the prior consent of the copyright holder.

Use of the software and documentation is governed by the terms and conditions of the iPass Corporate Remote Access Agreement, or Channel Partner Reseller Agreement.

Information in this guide is subject to change without notice.

Every effort has been made to use fictional companies and locations in this manual. Any actual company names or locations are strictly coincidental and do not constitute endorsement.

TABLE OF CONTENTS

Service Quality Management (SQM)	2
SQM Record Types	2
SQM Field Descriptions	3
XML Documents	15
Overview	15

Service Quality Management (SQM)

With each connection attempt by the Open Mobile client, iPass collects qualitative and quantitative data for the purpose of Service Quality Management (SQM). SQM data are useful for analysts in network operations, quality assessment, technical support, and account billing. IT administrators can obtain reports on SQM records for their end users from the iPass Portal.

SQM Record Types

Multiple types of SQM data are collected regarding each stage of Open Mobile connections, and are transmitted to the iPass Portal. These data types include client, application, connection, and device records. Data consists of individual fields. These records are then made available from the iPass Portal, in the following forms:

- **sqmApplication:** Data regarding VPN records and ROP (run-once packager) records.
- **sqmClient:** Data regarding the Open Mobile client software.
- **sqmConnection:** Data regarding the session, such as start and end times, as well as connection type (Wi-Fi, Mobile Broadband, and others.).
- **sqmConnectionQuality:** Data regarding speed test records.
- **sqmDevice:** Data regarding the software and hardware environment running Open Mobile.
- **sqmMonitor:** Data regarding usage limits and other policies.
- **sqmUsage:** Data regarding bytes sent and received for each session.
- **sqmUserAcceptance:** Data regarding when the user accepted roaming terms

The next chapter describes the fields appearing in these record types.

SQM Field Descriptions

The following fields are used in SQM records. . Each field contains one quality metric, such as a number, a string of text, or a Boolean true/false value pertaining to the client session. For example, the field `clientIpAddress` contains an integer value. Unless otherwise indicated, a field may occur any number of instances in a given record.

`accessPointMacAddress`

Record Type: Connection

Data Type: String (in the format 00:00:00:00:00:00)

Description: The hardware Media Access Control (MAC) address of the network card in the Wi-Fi access point to which the client connected.

`accessProcedure`

Record Type: Connection

Data Type: String

Description: For Broadband connections, the access procedure used to associate to the access point. (This can be correlated with a specific DLL.) Always '802.1x' for 802.1x connections.

`accessTypeDetail`

Record Type: Connection, Device

Data Type: String

Description: the media access type used for the connection. Valid values include the following:

- C=CDMA (iPassConnect v3.30)
- D=1xEV-DO (changed from iPass Connect)
- E=EDGE (changed from iPass Connect)
- G=GSM/GPRS
- H=Home Broadband (iPassConnect v3.2)
- I=ISDN
- L=DSL (iPassConnect v3.36)
- P=PHS
- R=1xRTT
- S=Personal Access Point (iPassConnect v3.2)
- U=UMTS
- W=Wireless Broadband

`authMethod`

Record Type: Connection

Data Type: String

Description: the protocol name used to authenticate a Wi-Fi session. Valid values include the following:

- 8021X_MD5
- 8021X_LEAP
- 8021X_TTLS - PAP
- 8021X_TTLS - CHAP
- 8021X_TTLS - MD5
- 8021X_TTLS - MSCHAP
- 8021X_TTLS - MSCHAPv2

`baudRate`

Record Type: Connection

Data Type: Integer

Description: The modulation rate of the modem connection (in kilobytes per second) for dial-up connections. When a connection fails, the speed is 0.

`cellId`

Record Type: Connection

Data Type: String

Description: The Starting Cell Identifier obtained at the start of the connection.

`clientId`

Record Type: Client, Connection, Connection Quality

Description: A unique identifier, formatted to represent the specific client software installation, in the format [Product Name] [Version Number].

In a Connection record, this field appears exactly once. In a Client record, this field is optional.

`clientIpAddress`

Record Type: Connection

Data Type: Integer

Description: The Internet Protocol (IP) address assigned to the client during a session. For successful dial connections, this appears only in the stop record.

`clientMacAddress`

Record Type: Device

Data Type: String

Description: The hardware Media Access Control (MAC) address of the client, for only a Wi-Fi session.

`companyId`

Record Type: Client, Connection, Connection Quality

Data Type: String

Description: A unique identifier that represents this user's company in all iPass systems and records.

`connectionDiagnostics`

Record Type: Connection

Data Type: String

Description: Captures the error stack trace during the connection process.

`connectionFailureCode`

Record Type: Connection

Data Type: String

Description: An error message that characterizes why a connection was terminated. (This field replaces the iPassConnect element `errorCode`.)

`connectionMode`

Record Type: Connection Quality

Data Type: String

Description: For each session, shows if the user is connected to the Internet through a corporate network (Corporate), a VPN (VPN), or neither (Internet).

`date`

Record Type: Client

Data Type: String

Description: Date of the accompanying records.

`dateTime`

Record Type: Connection

Data Type: String

Description: Date and time of the accompanying records.

`deviceId`

Record Type: Connection, Device

Data Type: String

Description: The name of the device used to connect.

`devicePower`

Record Type: Connection

Data Type: Integer

Description: The power status, in dB, of a Mobile Broadband device.

`deviceType`

Record Type: Device

Data Type: String

Description: Type of device used to connect.

- Broadband values include WIRELESS; ETHERNET
- Dial values include MODEM; VPN; ISDN
- DSL values can report as either PPPoE or ISDN, depending on the DSL driver.

`dialString`

Record Type: Connection

Data Type: String

Description: The telephone number used in a specific dialup session.

`disconnectReasonCode`

Record Type: Connection

Data Type: Integer

Description: A numeric error message that characterizes why a session was terminated. Found only in the STOP record for successful connections.

`downloadServer`

Record Type: Connection Quality

Data Type: anyURI

Description: The server used to download data during a speed test.

`downloadRate`

Record Type: Connection Quality

Data Type: Integer (with a String attribute of unitType with default value: Bps)

Description: The rate of bytes per second downloaded during a speed test.

`endTime`

Record Type: Connection

Data Type: String

Description: Timestamp of the end of a session.

`gisReplyMessage`

Record Type: Connection

Data Type: String

Description: Response the client receives when it makes a GIS connection.

`hostPower`

Record Type: Connection

Data Type: Integer

Description: A number (range 1-100) characterizing the final Host Power supply at the end of connection.

`hwVersion`

Record Type: Connection

Data Type: Integer

Description: the hardware version of the device used to connect.

`index`

Record Type: Connection

Data Type: Integer

Description: An incrementing index for sequencing records within a connection. An Open Mobile connection begins with a Start record of index 0, followed by an arbitrary number of records with index 1, index 2, index 3 and on, ending with an End record.

This field appears exactly once in a connectionType record.

`installId`

Record Type: Client

Data Type: String, formatted in hexadecimal

Description: This is the unique ID generated during client install. The installer ensures this is unique for each device, even for customers that install many end users from the same image.

`lastUpdated`

Record Type: Client

Data Type: dateTime

Description: The time stamp of the most recent software update for a given client component.

`latency`

Record Type: Connection Quality

Data Type: Integer (with a String attribute of unitType with the default value: ms)

Description: The latency time in milliseconds recorded during a speed test.

`latLong`

Record Type: Connection

Data Type: String

Description: The latitude and longitude coordinates of the laptop or device when it connected. This is an element of geoLocation.

`loginString`

Record Type: Connection

Data Type: String

Description: The complete login ID used by Open Mobile to authenticate.

This field occurs 0 or 1 instance in a given record.

`loginUrl`

Record Type: Connection

Data Type: String

Description: URL used when authenticating to a GIS network.

`manufacturer`

Record Type: Device

Data Type: String

Description: The manufacturer name of the hardware device used to connect.

`modemNegotiatedTime`

Record Type: Connection

Data Type: dateTime

Description: The timestamp at which a modem connection became successful or reports failure.

`modemPhoneNumber`

Record Type: Connection

Data Type: String

Description: The telephone number dialed by the client modem. for dialup.

`networkId`

Record Type: Connection

Data Type: String

Description: S network identifier for the network used at start of connection.

- GSM- Network Id
- CDMA- System Id.

`networkName`

Record Type: Connection

Data Type: String

Description: The name of the network used to make a mobile connection.

os

Record Type: Client

Data Type: String

Description: The name of the operating system on which Open Mobile is running.

osVersion

Record Type: Client

Data Type: String

Description: The version number of the operating system running the client hardware.

packetLoss

Record Type: Connection Quality

Data Type: Integer

Description: The percentage of packets lost during a speed test.

popCityCode

Record Type: Connection

Data Type: String

Description: The city code of the access point.

popCountryCode

Record Type: Connection

Data Type: String

Description: The country code of the access point.

popStateCode

Record Type: Connection

Data Type: String

Description: The state, province, parish, or prefecture of the access point, as applicable.

portName

Record Type: Device

Data Type: String

Description: The name of the COM port being used, with a maximum length of 128 characters. This is retrieved from the port entry at the start of a connection.

powerSource

Record Type: Connection

Data Type: String

Description: A number (Range: 0-100) characterizing the power state of the device at the end of a connection.

`pppExtensions`

Record Type: Connection

Data Type: Boolean

Description: A flag. Valid values are 0 and 1. Do not set unless specifically required.

- When this flag is set to 1, RAS disables the PPP LCP extensions. This may be necessary to connect to certain older PPP implementations.
- Clear this flag (set to 0) if it interferes with features such as server callback.

`profileId`

Record Type: Connection, Connection Quality

Data Type: String

Description: User's client software profile ID number.

`rasVersion`

Record Type: Connection

Data Type: String

Description: The release version number of the client's Remote Access Software, for dial access points.

`rtnStatus`

Record Type: Connection

Data Type: String

Description: Shows the response to the RTN (real time network) assessment. Possible values include:

- 0: Unknown
- 1: Successful authorization
- 2: User credentials are incorrect
- 3: RoamServer is inaccessible
- 4: Customer's AAA server returned an error

`roamingIndicator`

Record Type: Connection

Data Type: Boolean

Description: Information regarding the roaming status of a mobile device at the end of a connection. Valid values include Y or N.

`roamingListVersion`

Record Type: Connection

Data Type: String

Description: The Preferred Roaming List (PRL) version for only CDMA hardware.

`scriptFileName`

Record Type: Connection

Data Type: String

Description: The dial-up networking script name when a connection to dial-up pop requires this script.

`securityMode`

Record Type: Connection

Data Type: String

Description: The wireless security method, such as WEP, WPA, and so on. The value is determined by the customer provisioning.

`serialNumber`

Record Type: Device

Data Type: String

Description: The Device Serial Number of the GSM: IMEI or CDMA: ESN device used to connect.

`serverIpAddress`

Record Type: Connection

Data Type: String

Description: The Internet Protocol (IP) address for the Dial up and Access Point or Router IP Address for Broadband.

`serviceName`

Record Type: Connection

Data Type: String

Description: The Service Name for a DSL connection. This is optional depending on the DSL provider.

`sessionId`

Record Type: Connection, Connection Quality

Data Type: Integer

Description: A unique Session ID for SQM-CDR correlation and merging of START and STOP records for successful connections. The format is CUSTOMER_ID/SESSION_ID/USER_ID. The SESSION_ID component is a 10-character descriptor with format as follows:

- character 1: 0-9 representing algorithm used (always 0 for U and Q records).
- character 2: U/S/Q: USID, iSEEL or only SQM.
- characters 3-7: 5 digit Dialer ID, 00000 if not yet obtained from server.
- characters 8-10: Hex counter starting at 001.

`signalStrength`

Record Type: Connection

Data Type: Integer

Description: A number (range: 1–100) which characterizes the signal strength of the Wi-Fi access point, with status taken at the end of a session.

`softwareCompression`

Record Type: Connection

Data Type: Integer

Description: A flag which reflects the compression used in a connection. This flag is set by default.

- When this flag is set, software compression is negotiated on the link, the PPP driver to attempts to negotiate CCP with the server.
- Clearing this flag can reduce the negotiation period when a server does not support a compatible compression protocol.

`source`

Record Type: Connection

Data Type: Enum (with only two acceptable values: “GPS” or “Other”)

Description: The source of the latitude and longitude coordinates recorded in latLong. This is an element of geoLocation.

`ssid`

Record Type: Connection

Data Type: String

Description: The access point SSID used in a specific session.

`stageType`

Record Type: Connection

Data Type: Enum (with three accepted value: “start”, “mid”, and “stop”)

Description: Identifies the beginning, middle, or end position in a specific connection for the record entry it accompanies.

`startTime`

Record Type: Connection

Data Type: dateTime

Description: Identifies the UCT/GMT time of the beginning for a specific session.

`subscriberId`

Record Type: Device

Data Type: String

Description: The numeric IMSI, for only GSM devices.

`subscriberNumber`

Record Type: Device

Data Type: String

Description: The Subscriber Directory Number, or MSISDN, of the hardware device used to connect.

`swVersion`

Record Type: Device

Data Type: String

Description: The version for the firmware contained in the device used to connect.

`testMode`

Record Type: Connection Quality

Data Type: String

Description: This value be “Manual” if the user initiated the speed test, and “Auto” if the speed test happened automatically.

`testStatus`

Record Type: Connection Quality

Data Type: String

Description: This value will value be “Success” if the speed test was succesful, “Failure” if the speed test failed, and “Cancelled” if the speed test was cancelled.

`time`

Record Type: Connection Quality

Data Type: dateTime

Description: Timestamp for when the speed test was initiated.

`timeToConnect`

Record Type: Connection

Data Type: String

Description: Identifies the duration of the connection attempt for a specific session.

`uploadRate`

Record Type: Connection Quality

Data Type: Integer (with a String attribute of unitType with default value: Bps)

Description: The rate of bytes per second uploaded during a speed test.

`uploadServer`

Record Type: Connection Quality

Data Type: anyURI

Description: The server that was used to upload data during a speed test.

`userId`

Record Type: Connection

Data Type: String

Description: This field identifies the user account name for a specific session. For example, jsmith.

`version`

Record Type: Connection

Data Type: String

Description: This field identifies the version of software for a specific client module.

XML Documents

This chapter contains the definitions which iPass uses for SQM record keeping. These files could be used to write a program to parse SQM data collected from the iPass Portal.

Overview

All SQM records are defined and stored in valid XML format, as shown in the following files:

- sqmApplication.xsd
- sqmClient.xsd
- sqmConnection.xsd
- sqmConnectionQuality.xsd
- sqmMonitor.xsd
- sqmUsage.xsd
- sqmUserAcceptance.xsd

sqmApplication.xsd

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="sqmList" type="applicationListType"/>
  <xsd:complexType name="applicationListType" >
    <xsd:sequence>
      <xsd:choice >
        <xsd:element name="vpn" type="vpnApplicationType"/>
        <xsd:element name="rop" type="ropType"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:group name="commonApplicationFields">
    <xsd:sequence>
      <xsd:element name="clientId">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="timestamp" type="xsd:long" />
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:group>
</xsd:schema>
```

```

        <xsd:element name="launchTrigger" type="launchType" minOccurs="1"
maxOccurs="1" />
        <xsd:element name="name" type="xsd:string" />
        <xsd:element name="softwareVersion" type="xsd:string" minOccurs="0"
maxOccurs="1" />
    </xsd:sequence>
</xsd:group>
<xsd:complexType name="ropType">
    <xsd:sequence>
        <xsd:group ref="commonApplicationFields"/>
        <xsd:element name="ropUid" type="xsd:string" minOccurs="1"
maxOccurs="1"/> <!-- ROP ID-->
        <xsd:element name="date" type="xsd:dateTime"/>
        <xsd:element name="status" type="resultType" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="vpnApplicationType">
    <xsd:sequence>
        <xsd:group ref="commonApplicationFields"/>
        <xsd:group ref="commonSessionFields"/>
        <xsd:element name="gateway" type="xsd:string" minOccurs="1"
maxOccurs="1"/> <!-- gateway or url-->
    </xsd:sequence>
</xsd:complexType>
<xsd:group name="commonSessionFields">
    <xsd:sequence>
        <xsd:element name="stage" type="stageType" minOccurs="1"
maxOccurs="1" />
        <xsd:element name="index" type="xsd:integer" minOccurs="1"
maxOccurs="1"/>
        <xsd:element name="startTime" type="xsd:dateTime"/>
        <xsd:element name="endTime" type="xsd:dateTime" />
        <xsd:element name="sessionId" type="xsd:string" /> <!-- unique
session id-->
        <xsd:element name="vpnSessionId" type="xsd:long " minOccurs="0"
maxOccurs="1" />
        <xsd:element name="userId" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="connectionResult" type="resultType"
minOccurs="1" maxOccurs="1"/>

```

```

        <xsd:element name="connectionStatus" type="connectionStatusType"
minOccurs="0" maxOccurs="1"/> <!-- connection status codes -->
        <xsd:element name="disconnectReason" type="disconnectReasonType"
minOccurs="0" maxOccurs="1" /> <!-- disconnect reason codes -->
    </xsd:sequence>
</xsd:group>
<xsd:simpleType name="resultType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="success"/>
        <xsd:enumeration value="fail"/>
        <xsd:enumeration value="unknown"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="stageType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="start"/>
        <xsd:enumeration value="mid"/>
        <xsd:enumeration value="stop"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="launchType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="autolaunch"/>
        <xsd:enumeration value="connectondemand"/>
        <xsd:enumeration value="autoreconnect"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="connectionStatusType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Unknown"/>    <!-- Unknown -->
        <xsd:enumeration value="Connection Success"/>    <!-- Connection
Success-->
        <xsd:enumeration value="Cancelled By User"/>    <!-- Cancelled by
user-->
        <xsd:enumeration value="Dropped By Policy"/>    <!-- Dropped by
policy-->
        <xsd:enumeration value="Authentication Failure"/>    <!--
Authentication failure-->
        <xsd:enumeration value="Invalid Configuration"/>    <!-- Invalid
configuration-->
    </xsd:restriction>
</xsd:simpleType>

```

```

Timeout-->      <xsd:enumeration value="Connection Timeout"/>      <!-- Connection
error-->      <xsd:enumeration value="Internal VPN Error"/>      <!-- Internal VPN
monitoring is disabled-->      <xsd:enumeration value="VPN Monitoring Disabled"/>      <!-- VPN
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name="disconnectReasonType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Unknown"/> <!-- Unknown -->
            <xsd:enumeration value="Disconnected By User"/> <!-- Disconnected
by user-->
            <xsd:enumeration value="Disconnected By Policy"/> <!--
Disconnected by policy-->
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>

```

sqmClient.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="client" type="clientType" />
    <xsd:complexType name="clientType">
        <xsd:sequence>
            <!-- This is the unique id of the client -->
            <xsd:element name="clientId">
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="timestamp" type="xsd:long" />
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>
            <xsd:element name="companyId" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="environment" type="environmentType"
minOccurs="1" maxOccurs="1"/>
            <!-- This is the unique id generated during client install. For
customers that image, the imaging process/protocol will take care of ensuring
that this is unique for each device -->

```

```

        <xsd:element name="installId" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
        <xsd:element name="products" type="productsType" minOccurs="1"
maxOccurs="1" />
        <xsd:element name="profiles" minOccurs="1" maxOccurs="1"
type="profilesType"/>
        <xsd:element name="networkDirectories" minOccurs="0" maxOccurs="1"
type="networkDirectoriesType"/>
        <xsd:element name="hostDevice" type="hostDeviceType" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="clientStartupMode" type="xsd:string"
minOccurs="0" maxOccurs="1"/>

        <!-- We may not need the download rate any more for phone book
        <xsd:element name="phonebookDownloadRate" type="xsd:integer"/>
        -->
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="environmentType">
    <xsd:sequence>
        <xsd:element name="locale" type="xsd:string"/>
        <xsd:element name="os" type="xsd:string"/>
        <xsd:element name="osVersion" type="xsd:string"/>
        <!--TBD other environment -->
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="productsType">
    <xsd:sequence>
        <xsd:element name="product" minOccurs="1" type="productType"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="productType">
    <xsd:sequence>
        <xsd:element name="id" type="xsd:string"/>
        <xsd:group ref="commonVersionFields" />
        <xsd:element name="components" type="componentsType"
minOccurs="0"/>
    </xsd:sequence>

```

```

</xsd:complexType>

<xsd:complexType name="componentsType">
  <xsd:sequence>
    <xsd:element name="component" type="componentType" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="componentType">
  <xsd:sequence>
    <xsd:element name="id" type="xsd:string"/>
    <xsd:group ref="commonVersionFields" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="profilesType">
  <xsd:sequence>
    <xsd:element name="profile" minOccurs="1" type="profileType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="profileType">
  <xsd:sequence >
    <xsd:element name="id" type="xsd:integer"/>
    <xsd:group ref="commonVersionFields" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="networkDirectoriesType">
  <xsd:sequence>
    <xsd:element name="networkDirectory" minOccurs="0"
type="networkDirectoryType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="networkDirectoryType">
  <xsd:sequence >
    <xsd:element name="id" type="xsd:integer"/>

```

```

        <xsd:group ref="commonVersionFields" />
    </xsd:sequence>
</xsd:complexType>

<xsd:group name="commonVersionFields" >
    <xsd:sequence >
        <xsd:element name="version" type="xsd:string"/>
        <xsd:element name="lastUpdated" type="xsd:dateTime"/>
    </xsd:sequence>
</xsd:group>

<xsd:complexType name="hostDeviceType">
<xsd:sequence minOccurs="0">
    <xsd:element name="manufacturer" type="xsd:string"
minOccurs="0"/>
    <xsd:element name="model" type="xsd:string" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>

</xsd:schema>

```

sqmConnection.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="connection" type="connectionType"/>
<xsd:complexType name="connectionType">
    <xsd:sequence>
        <!-- The client / dialer id obtained from the dialerid server
at the time of installation of the client -->
        <xsd:element name="clientId" minOccurs="1" maxOccurs="1">
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:string">
                        <xsd:attribute name="timestamp"
type="xsd:long"/>
                    </xsd:extension>
                </xsd:simpleContent>
                <!-- Must match with the corresponding
timestamp field on the client record.
                This timestamp is used by the server to correlate the
client record. The timestamp is
                just a monotonically increasing number identifying a point
in time when the client record was generated-->
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

```

        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
    <!-- The company ID as seen in the profile.xml. This must be
sent verbatim, unaltered by the client
    The server uses this information to attribute this connection to the specific
customer -->
    <xsd:element name="companyId" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
    <!-- Indicates stage of the connection : start, mid, stop -->
    <xsd:element name="stage" type="stageType" minOccurs="1"
maxOccurs="1"/>
    <!-- Counter for the connection records within a session. For
start records, this is 0.
    For every subsequent connection record in the same session, the
counter is incremented by 1.
    So for instance, if the next connection record is a stop record, the
index is 1 -->
    <xsd:element name="index" type="xsd:integer" minOccurs="1"
maxOccurs="1"/>
    <xsd:choice minOccurs="0" maxOccurs="1">
        <xsd:element name="wifi" type="wifiType"/>
        <xsd:element name="ethernet" type="etherType"/>
        <xsd:element name="mobile" type="mobileType"/>
        <xsd:element name="dsl" type="dslType"/>
        <xsd:element name="dial" type="dialType"/>
        <xsd:element name="home" type="homeType"/>
        <!-- for backward compatibility -->
        <xsd:element name="isdn" type="isdnType"/>
        <!-- for backward compatibility -->
        <xsd:element name="wimax" type="wimaxType"/>
    </xsd:choice>
    <xsd:element name="geoLocation" minOccurs="0">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="latlong"
type="xsd:string"/>
                <xsd:element name="source">
                    <xsd:simpleType>
                        <xsd:restriction
base="xsd:string">

```



```

value="GPS"/>                                <xsd:enumeration
value="Other"/>                               <xsd:enumeration
                                              </xsd:restriction>
                                              </xsd:simpleType>
                                              </xsd:element>
                                              </xsd:sequence>
                                              </xsd:complexType>
                                              </xsd:element>
                                              </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="stageType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="start"/>
    <xsd:enumeration value="mid"/>
    <xsd:enumeration value="stop"/>
    <!-- This stage is sent when the connection attempt is made on
the client -->
    <!-- Currently not used. But intended to capture mid-connection
records -->
    <!-- This stage is sent when the connection is stopped. The
windows client makes
every effort to create this record, say even in situations where
the laptop or when such
power mode change events are propagated by the OS-->
  </xsd:restriction>
</xsd:simpleType>
<xsd:group name="commonConnectionFields">
  <xsd:sequence>
    <!-- The start time of the connection attempt if this is a start
record, the start time of disconnect attempt
if this is a stop record. N/A for mid stage -->
    <xsd:element name="startTime" type="xsd:dateTime"/>
    <!-- The end time of the connection attempt if this is a start
record. The end time of disconnect attempt
if this is a stop record. N/A for mid stage. The session length
is the difference between the end time of the
stop record and the start time of the start record -->
    <xsd:element name="endTime" type="xsd:dateTime"/>
    <!-- Unique session id.

```

The Windows Client uses a specific algorithm using the dialerid, a counter and certain attributes indicating

if this is an ipass network, another attribute indicating if ISEEL is used. NEED EXAMPLES AND FORMAT -->

```
<xsd:element name="sessionId" type="xsd:string"/>
<!--ipass unique session id-->
<!-- In Windows Clients upto 2.1: For initiated connection: the
userid used for authentication on that network: userid@domain
For inherited connection: the userid as specified in the
master account.
If the master account specifies for instance the end user
will be typing in the userid, and
at the point of time when this connection happens, the end
user hasnt provided that userid to
the system yet, then the Windows client obtains the userid
from the OS and sends it in this format:
WINDOWS:<windowsuser>
On an inherited connection, when the user hasnt yet logged
in, the windowsuser is not known.
DEVICE:<machinename> is sent example DEVICE:AVI_LAPTOP
Max length: 255 characters
-->
<xsd:element name="userId" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
<!-- example jsmith@company.com -->
<!-- In Windows Clients starting 2.1: the alternate user ids are
captured in this separate alternate Id element
If the user hasnt yet connected using this client installation
with her or his credentials and a connection
is inherited, either the Windows username if it is possible to
get that from the OS. If not, the device name
is used. The IT admin may have configured the profile to not
collect the Windows user name or the device name,
in which case, the client generates an id that is unique to
each install and sends that.
Either userId or alternateId must be provided.
Max length: 255 characters for each alternateId
-->
<xsd:element name="alternateIds" type="alternateIdsType"
minOccurs="0" maxOccurs="1"/>
```

```

        <!-- The NAI login string used for authentication - only sent if
this connection is made on a WIFI and Dial network
        that is provisioned. This element is not sent if this connection
is made on a personal network or other network
        that is not in one of the directories -->
        <xsd:element name="loginString" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <!-- the complete login id used by ipc to authenticate -->
        <!-- The Profile ID from the Profile.xml -->
        <xsd:element name="profileId" type="xsd:integer"/>
        <!-- Only sent in start records. The only two values are 0
(indicates that this connection attempt succeeded)
        and 1 (indicates that this connection attempt failed and a
connection could not be established) -->
        <xsd:element name="connectionStatus" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
        <!-- Only populated in start records. 0 indicates success for
initiated connections. 14407 indicates success
        for inherited connections In case of failure, the failure code
is sent. -->
        <xsd:element name="connectionStatusCode" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
        <!-- connection failure - was originally errorCode -->
        <xsd:element name="connectReason" minOccurs="0" maxOccurs="1">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="app_action">
                        <xsd:annotation>
                            <xsd:documentation>App
connectivity requirement triggered connection</xsd:documentation>
                        </xsd:annotation>
                    </xsd:enumeration>
                    <xsd:enumeration value="forced">
                        <xsd:annotation>
                            <xsd:documentation>Admin-
defined (directory flagged) (forced) autoconnect</xsd:documentation>
                        </xsd:annotation>
                    </xsd:enumeration>
                    <xsd:enumeration value="auto">
                        <xsd:annotation>
                            <xsd:documentation>Autoconnect (Android) /Auto-
join+autologin(iOS)</xsd:documentation>

```

```

        </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="user">
        <xsd:annotation>
            <xsd:documentation>User
requested connection</xsd:documentation>
        </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="unknown">
        <xsd:annotation>
            <xsd:documentation>Reason for
connection not known</xsd:documentation>
        </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="other">
        <xsd:annotation>
            <xsd:documentation>Client was
reassociated to previously connected network</xsd:documentation>
        </xsd:annotation>
    </xsd:enumeration>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<!-- Only populated in stop records. The reason for the
disconnect attempt is sent -->
    <xsd:element name="disconnectReasonCode" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
    <!-- The IP address of the client. It is the pre-vpn IP address -
->
    <xsd:element name="clientIPAddress" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
    <!-- Only sent in stop records. Indicates total number of bytes
transmitted by the device in this session -->
    <xsd:element name="uploadedBytes" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
    <!-- bytes? -->
    <!-- Only sent in stop records. Indicates total number of bytes
received by the device in this session -->
    <xsd:element name="downloadedBytes" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
    <!-- Only sent in start records. Indicates the name of the VPN
that is specified in the profile -->

```

```

        <xsd:element name="vpn" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <!-- The id of the first directory that had this SSID or network
and resulted in this connection record.
        If the very first directory ....-->
        <xsd:element name="directoryId" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <!-- The MAC Address of the client as obtained from the adapter
used for this connection -->
        <xsd:element name="clientMacAddress" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <!-- Not yet used. Will be used to identify if this connection
was established on a tethered device.
        1 indicates tethered, 0 indicates non-tethered -->
        <xsd:element name="tethered" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="connectionDiagnostics" minOccurs="0"/>
        <xsd:element name="rtnStatus" minOccurs="0"/>
    </xsd:sequence>
</xsd:group>
<xsd:complexType name="wifiType">
    <xsd:sequence>
        <xsd:group ref="commonConnectionFields"/>
        <!-- MAC address of the access point / network -->
        <xsd:element name="accessPointMacAddress" type="xsd:string"/>
        <!-- the SSID of the network being connected to -->
        <xsd:element name="ssid" type="xsd:string"/>
        <!-- loginUrl from GIS tags -->
        <xsd:element name="loginUrl" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <!-- Not used -->
        <xsd:element name="locationDescription" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <!-- Not used -->
        <xsd:element name="locationGroupId" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <!-- Not used -->
        <xsd:element name="locationId" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <!-- PEAP, TKIP, etc TBD -->
        <xsd:element name="authMethod" type="xsd:string" minOccurs="0"
maxOccurs="1"/>

```

```

        <!-- convert to choice example PEAP, TKIP, ,,,-->
        <!-- WEP, WPA, WPA2... TBD-->
        <xsd:element name="securityMode" type="xsd:string" minOccurs="0"
maxOccurs="1"/>

        <!-- maybe a choice? wep, wpa etc -->
        <!-- GI.1, DI.1, TBD -->
        <xsd:element name="accessProcedure" type="xsd:string"
minOccurs="0" maxOccurs="1"/>

        <!-- convert to choice example GIS -->
        <!-- the Network ID of this network as specified in the directory
that was used for making this connection with this network -->
        <xsd:element name="networkId" type="xsd:string" minOccurs="0"
maxOccurs="1"/>

        <!-- new term for popid - maybe optional ... check with the new
directory schema -->
        <!--Received WiFi signal strength contains a value between 0 and
100. A value of 0 implies an actual RSSI signal strength of -100 dbm.

A value of 100 implies an actual RSSI signal strength of -50 dbm.
The range between 1 and 99 is calculated using linear interpolation.-->
        <xsd:element name="signalStrength" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="gisReplyMessage" type="xsd:string"
minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="etherType">
    <xsd:sequence>
        <xsd:group ref="commonConnectionFields"/>
        <!-- GI.1 -->
        <xsd:element name="accessProcedure" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <!-- Not used -->
        <xsd:element name="locationCodeName" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <!-- how is this difft from location id -->
        <!-- make the location info optional - this may not be done as
this is not the most efficient way-->
        <!-- Not used -->
        <xsd:element name="locationDescription" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <!-- Not used -->
        <xsd:element name="locationGroupId" type="xsd:string"
minOccurs="0" maxOccurs="1"/>

```

```

        <!-- Not used -->
        <xsd:element name="locationId" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="mobileType">
    <xsd:sequence>
        <!-- unique id concat of device fields created by client -->
        <xsd:element name="deviceId" type="xsd:string"/>
        <!-- The Network ID as obtained from the Radio -->
        <xsd:element name="networkId" type="xsd:string"/>
        <!-- cell tower id as obtained from the Radio -->
        <xsd:element name="cellId" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <!-- The Network Name as broadcast by the network -->
        <xsd:element name="networkName" type="xsd:string"/>
        <!-- MNC/Network ?? -->
        <!-- Yes if this network is not specified in the home or home
equivalent list of networks in the APN.ini or AdminAPN.xml
>
        No if the network is a home network or home equivalent network --
        <xsd:element name="roamingIndicator" type="xsd:string"/>
        <!-- roaming status ?? -->
        <!-- TBD -->
        <xsd:element name="signalStrength" type="xsd:integer"/>
        <!-- received signal strength ?? -->
        <!-- As reported by the vendor sdk -->
        <xsd:element name="devicePower" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <!-- As reported by the vendor sdk -->
        <xsd:element name="hostPower" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <!-- TBD Battery or AC - as reported by the vendor -->
        <xsd:element name="powerSource" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <!-- As reported by the vendor sdk (CDMA or GSM) -->
        <xsd:element name="networkType" type="xsd:string"/>
        <!-- service type ??? -->
        <!-- Applicable for GSM networks only - the APN name, could be
url or ip address of the access point server -->

```

```

        <xsd:element name="accessPointName" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <!-- only for GSM devices-->
        <!-- Not used -->
        <xsd:element name="country" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <!-- MCC/Country -->
        <!-- Equipment ID as reported by the vendor/TWWA -->
        <xsd:element name="imei" type="xsd:string"/>
        <!-- International Mobile Subscriber ID as reported by the
vendor/TWWA -->
        <xsd:element name="imsi" type="xsd:string"/>
        <xsd:element name="networkRegistration"
type="networkRegistrationType" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="networkMode" type="networkSelectionModeType"
minOccurs="0" maxOccurs="1"/>
        <!-- Network selection mode auto / manual -->
        <xsd:element name="band" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="bearer" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="rssi" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="rscp" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <!-- RSRP for 4G devices-->
        <xsd:element name="rsrp" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="alternateIdsType">
    <xsd:sequence>
        <xsd:element name="alternateId" type="alternateIdType"
minOccurs="1" maxOccurs="2"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="alternateIdType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="identifierType" use="required">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">

```



```

value="SystemUserName"/>
        <xsd:enumeration
            <xsd:enumeration value="DeviceName"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:attribute>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="networkSelectionModeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="auto"/>
        <xsd:enumeration value="manual"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="networkRegistrationType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="circuit"/>
        <xsd:enumeration value="packet"/>
        <xsd:enumeration value="mixed"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="dialType">
    <xsd:sequence>
        <xsd:group ref="commonConnectionFields"/>
        <xsd:element name="rasVersion" type="xsd:string"/>
        <xsd:element name="dialString" type="xsd:string"/>
        <xsd:element name="scriptFileName" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="softwareCompression" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="pppExtensions" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="serverIpAddress" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="baudRate" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="modemPhoneNumber" type="xsd:string"/>
        <xsd:element name="modemNegotiatedTime" type="xsd:dateTime"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="popCountryCode" type="xsd:string"/>

```

```

maxOccurs="1"/>
    <xsd:element name="popStateCode" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="popCity" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="accessTypeDetail" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="dslType">
  <xsd:sequence>
    <xsd:group ref="commonConnectionFields"/>
    <xsd:element name="serviceName" type="xsd:string"/>
    <xsd:element name="softwareCompression" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="pppExtensions" type="xsd:integer"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="serverIpAddress" type="xsd:string"
minOccurs="0" maxOccurs="1"/>
    <xsd:element name="baudRate" type="xsd:integer" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="rasVersion" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="deviceName" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
    <xsd:element name="deviceType" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="homeType">
  <xsd:sequence>
    <xsd:group ref="commonConnectionFields"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="isdntype">
  <xsd:sequence>
    <xsd:group ref="commonConnectionFields"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="wimaxType">
  <xsd:sequence>
    <xsd:group ref="commonConnectionFields"/>
  </xsd:sequence>

```

```

</xsd:complexType>
<xsd:complexType name="placeholder">
  <xsd:sequence>
    <xsd:element name="x8021xRawErrorCode" type="xsd:integer"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

sqmConnectionQuality.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="connectionQuality" type="connectionQualityType"/>

  <xsd:complexType name="connectionQualityType">

    <xsd:sequence>
      <xsd:element name="clientId" minOccurs="1" maxOccurs="1" >
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="timestamp" type="xsd:long" />
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="companyId" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="profileId" type="xsd:int" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="sessionId" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="connectionMode" type="connectionModeType"
minOccurs="1" maxOccurs="1"/>
      <xsd:element name="time" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="testMode" type="testModeType" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="testStatus" type="testStatusType"
minOccurs="1" maxOccurs="1"/>
      <xsd:element name="uploadServer" type="xsd:anyURI" minOccurs="0"
maxOccurs="1"/>

```

```

        <xsd:element name="downloadServer" type="xsd:anyURI"
minOccurs="0" maxOccurs="1"/>
        <xsd:element name="downloadRate" minOccurs="1" maxOccurs="1">
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:int">
                        <xsd:attribute name="unitType" type="xsd:string"
default="Bps"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="uploadRate" minOccurs="1" maxOccurs="1">
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:int">
                        <xsd:attribute name="unitType" type="xsd:string"
default="Bps"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="latency" minOccurs="1" maxOccurs="1">
            <xsd:complexType>
                <xsd:simpleContent>
                    <xsd:extension base="xsd:int">
                        <xsd:attribute name="unitType" type="xsd:string"
default="ms"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
        </xsd:element>
        <xsd:element name="packetLoss" type="xsd:int" minOccurs="1"
maxOccurs="1"/>

        </xsd:sequence>

    </xsd:complexType>

    <xsd:simpleType name="testModeType">

```

```

        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Manual"/>
            <xsd:enumeration value="Auto"/>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="testStatusType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Success"/>
            <xsd:enumeration value="Failed"/>
            <xsd:enumeration value="Cancelled"/>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="connectionModeType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Internet"/>
            <xsd:enumeration value="VPN"/>
            <xsd:enumeration value="Corporate"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>

```

sqmDevice.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="deviceList" type="deviceListType"/>

    <xsd:complexType name="deviceListType" >
        <xsd:sequence>

            <xsd:choice >
                <xsd:element name="mobileDevice" type="mobileDeviceType"/>
                <xsd:element name="ethernetDevice" type="ethernetDeviceType"/>
                <xsd:element name="wifiDevice" type="wifiDeviceType"/>
                <xsd:element name="dialDevice" type="dialDeviceType"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>

```

```

        </xsd:sequence>
    </xsd:complexType>

    <xsd:group name="commonDeviceFields">
        <xsd:sequence>
            <xsd:element name="clientId" minOccurs="1" maxOccurs="1" >
                <xsd:complexType>
                    <xsd:simpleContent>
                        <xsd:extension base="xsd:string">
                            <xsd:attribute name="timestamp" type="xsd:long" />
<!-- unique id for client record -->
                        </xsd:extension>
                    </xsd:simpleContent>
                </xsd:complexType>
            </xsd:element>

            <xsd:element name="deviceId" type="xsd:string" /> <!-- unique id
concat of device fields created by client -->

        </xsd:sequence>
    </xsd:group>

    <xsd:complexType name="mobileDeviceType">
        <xsd:sequence>
            <xsd:group ref="commonDeviceFields" />
            <xsd:element name="portName" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
            <xsd:element name="imei" type="xsd:string" />
            <xsd:element name="manufacturer" type="xsd:string" />
            <xsd:element name="model" type="xsd:string" />
            <xsd:element name="hardwareVersion" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
            <xsd:element name="softwareVersion" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
            <xsd:element name="name" type="xsd:string" />
            <xsd:element name="type" type="xsd:string" />
            <xsd:element name="driverVersion" type="xsd:string" minOccurs="0"
maxOccurs="1"/>

```

```

        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="wifiDeviceType">
        <xsd:sequence>
            <xsd:group ref="commonDeviceFields" />
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="ethernetDeviceType">
        <xsd:sequence>
            <xsd:group ref="commonDeviceFields" />
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="dialDeviceType">
        <xsd:sequence>
            <xsd:group ref="commonDeviceFields" />
        </xsd:sequence>
    </xsd:complexType>

</xsd:schema>

```

sqmMonitor.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="monitor" type="monitorType"/>
    <xsd:complexType name="monitorType">
        <xsd:sequence>
            <xsd:element name="id" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
            <xsd:element name="userId" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
            <xsd:element name="date" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>

```

sqmUsage.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="usage" type="usageType"/>

  <xsd:complexType name="usageType">

    <xsd:sequence>

      <xsd:element name="clientId" minOccurs="1" maxOccurs="1" >
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="timestamp" type="xsd:long" />
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="record" type="recordType" minOccurs="0"
maxOccurs="1" />

      <xsd:element name="sessionId" type="xsd:string" minOccurs="1"
maxOccurs="1"/>

      <xsd:element name="time" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>

      <xsd:element name="totalBytesSent" type="xsd:int" minOccurs="1"
maxOccurs="1"/>

      <xsd:element name="totalBytesRcvd" type="xsd:int" minOccurs="1"
maxOccurs="1"/>

      <xsd:element name="policyBy" type="policyType" minOccurs="1"
maxOccurs="1" />

    </xsd:sequence>

  </xsd:complexType>

  <xsd:simpleType name="policyType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Time Interval"/>
      <xsd:enumeration value="Bytes Used"/>
    </xsd:restriction>
  </xsd:simpleType>

```



```

<xsd:simpleType name="recordType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Initial Threshold Reached"/>
    <xsd:enumeration value="Incremental Threshold Reached"/>
    <xsd:enumeration value="In Session"/>
    <xsd:enumeration value="Usage Limit Reached"/>
    <xsd:enumeration value="High Usage Threshold Reached"/>
    <xsd:enumeration value="Blocked for Usage Limit Reached"/>
  </xsd:restriction>
</xsd:simpleType>

</xsd:schema>

```

sqmUserAcceptance.xsd

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="userAcceptance" type="userAcceptanceType"/>
  <xsd:complexType name="userAcceptanceType">
    <xsd:sequence>
      <xsd:element name="clientId" minOccurs="1" maxOccurs="1" >
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="timestamp" type="xsd:long" />
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="sessionId" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
      <xsd:element name="accepted" type="acceptType" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="date" type="xsd:dateTime" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="matter" type="matterType" minOccurs="1"
maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

```

```
<xsd:simpleType name="matterType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="roaming"/>
    <xsd:enumeration value="softwareupdate"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="acceptType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="yes"/>
    <xsd:enumeration value="no"/>
    <xsd:enumeration value="postponed"/>
  </xsd:restriction>
</xsd:simpleType>

</xsd:schema>
```

Table 1: CommonDeviceFields

Data Element	Data Type	Required	Description
clientID	xsd:string		
deviceID	xsd:string		

Table 2: CommonApplicationFields

Data Element	Data Type	Required	Description
clientId			
launchTrigger	launchType		
name	xsd:string		
softwareVersion	xsd:string		

Table 3: CommonSessionFields

Data Element	Data Type	Required	Description
stage	stageType		
index	xsd:integer		
startTime	xsd:dateTime		
endTime	xsd:dateTime		
sessionId	xsd:string		
vpnSessionId	xsd:long		
userId	xsd:string		
connectionResult	resultType		
connectionStatus	connectionStatusType		
disconnectReason	disconnectReasonType		

sqmApplication

Table 4: applicationListType

Data Element	Data Type	Required?	Description
vpn	vpnApplicationType		See Table 5 for a description of vpnApplicationType.
rop	ropType		See Table 6 for a description of ropType.

Table 5: vpnApplicationType

Data Element	Data Type	Required?	Description
CommonApplicationFields	--		See Table 2 for a description of CommonApplicationFields.
CommonSessionFields	--		See Table 3 for a description of CommonSessionFields.
gateway	xsd:string		

Table 6: ropType

Data Element	Data Type	Required?	Description
CommonApplicationFields	--		See Table 2 for a description of CommonApplicationFields.
ropUid	xsd:string		
date	dateTime		
status	resultType		See Table 7 for a description of resultType

Table 7: resultType

Data Element	Data Type	Required?	Description
xsd:string	success		
xsd:string	fail		
xsd:string	unknown		

Table 8: stageType

Data Element	Data Type	Required?	Description
start	xsd:string		
mid	xsd:string		
stop	xsd:string		