



EUQoS

QoS impact in AAA

End-to-End QoS Workshop
Paris 21-22 June 2005

Authors:

E. Angori, G. Martufi, M. Curcetti (Datamat)
{enrico.angori, giuseppe.martufi, matteo.curcetti} @datamat.it

- Introduction
- AAA inside EuQoS Architecture
- AAA QoS-related data exchange
- Which QoS data
- AAA Architecture Overview

The role of AAA in EuQoS project is:

*“[...] to manage user access to network resource (Authentication), to grant services and QoS level to the requesting user (Authorization) and to collect accounting data (Accounting).”**

** EuQoS Description of Work*

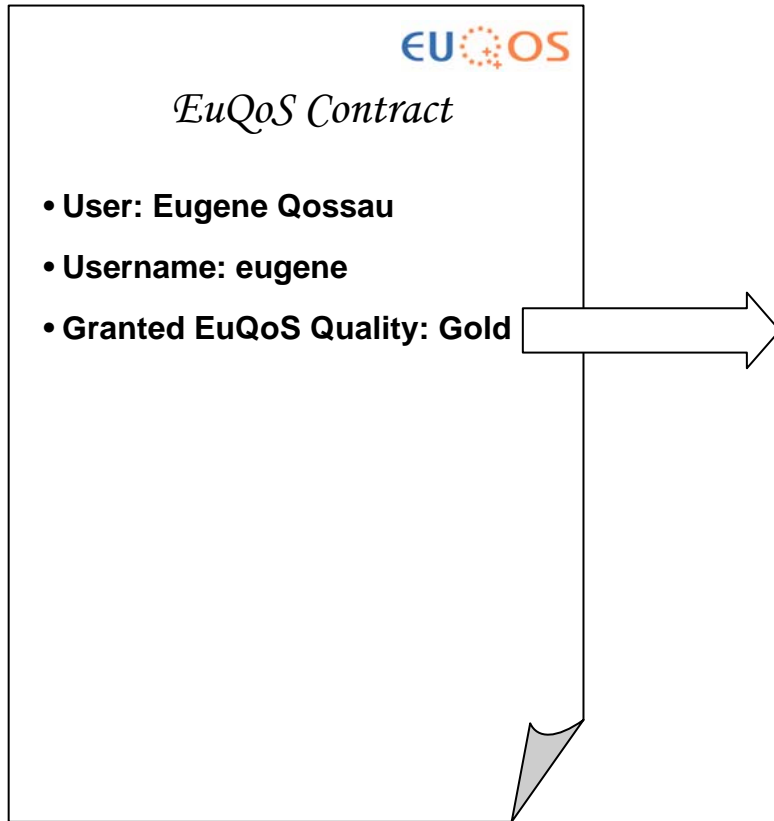



EuQoS Contract

- **User: Eugene Qossau**
- **Username: eugene**
- **Granted EuQoS Quality: Gold**

Every EuQoS User has a “Contract”, an agreement granting EuQoS services levels

The Contract-related Constraints List



EUOS

EuQoS Contract

- **User: Eugene Qossau**
- **Username: eugene**
- **Granted EuQoS Quality: Gold**

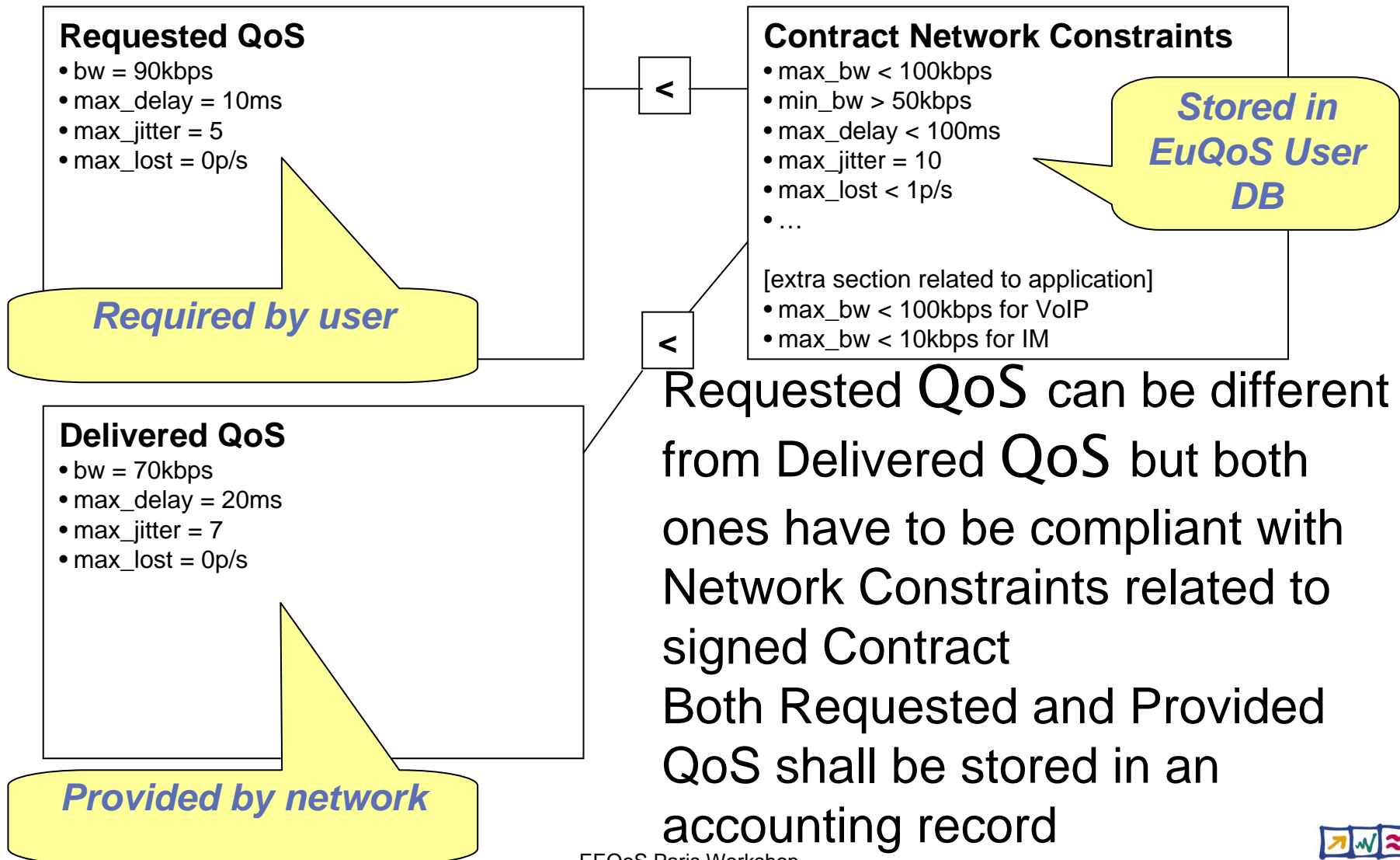
Contract Network Constraints

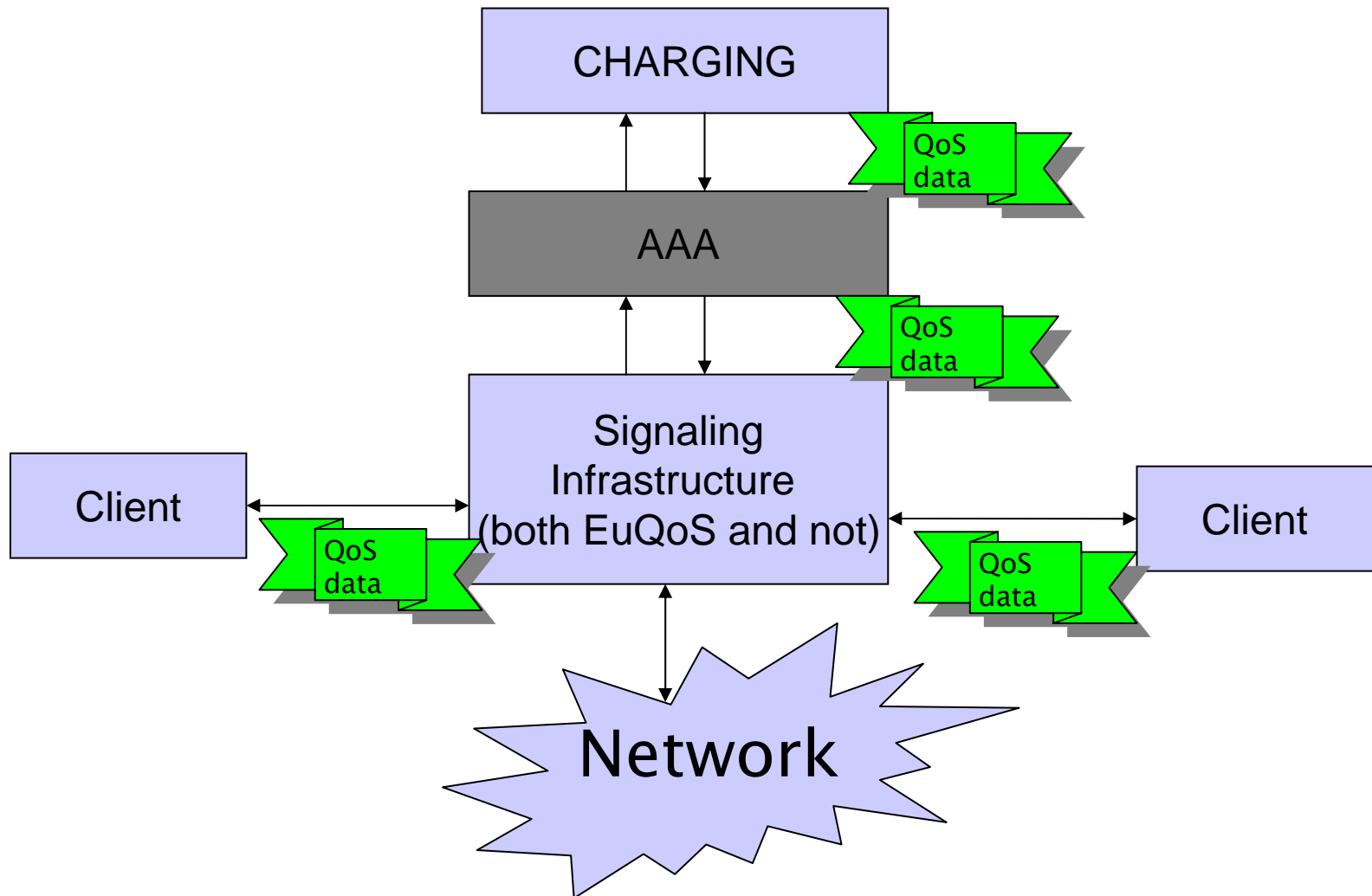
- max_bw < 100kbps
- min_bw > 50kbps
- max_delay < 100ms
- max_jitter = 10
- max_lost < 1p/s

[some extra sections related to application]

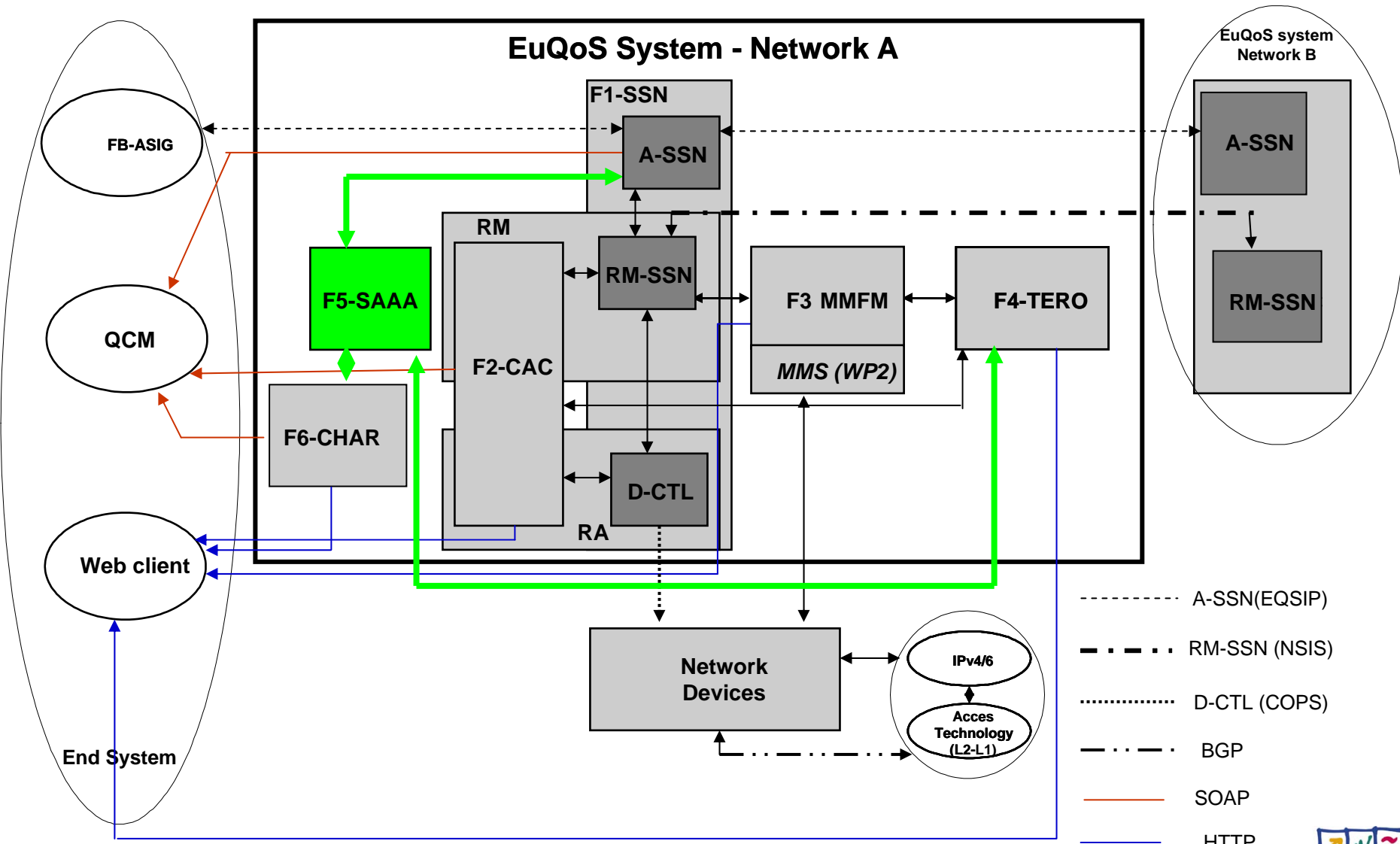
- max_bw < 100kbps for VoIP
- max_bw < 10kbps for IM

Each EuQoS Profile is related to a Network Constraint List (and some info related to applications)





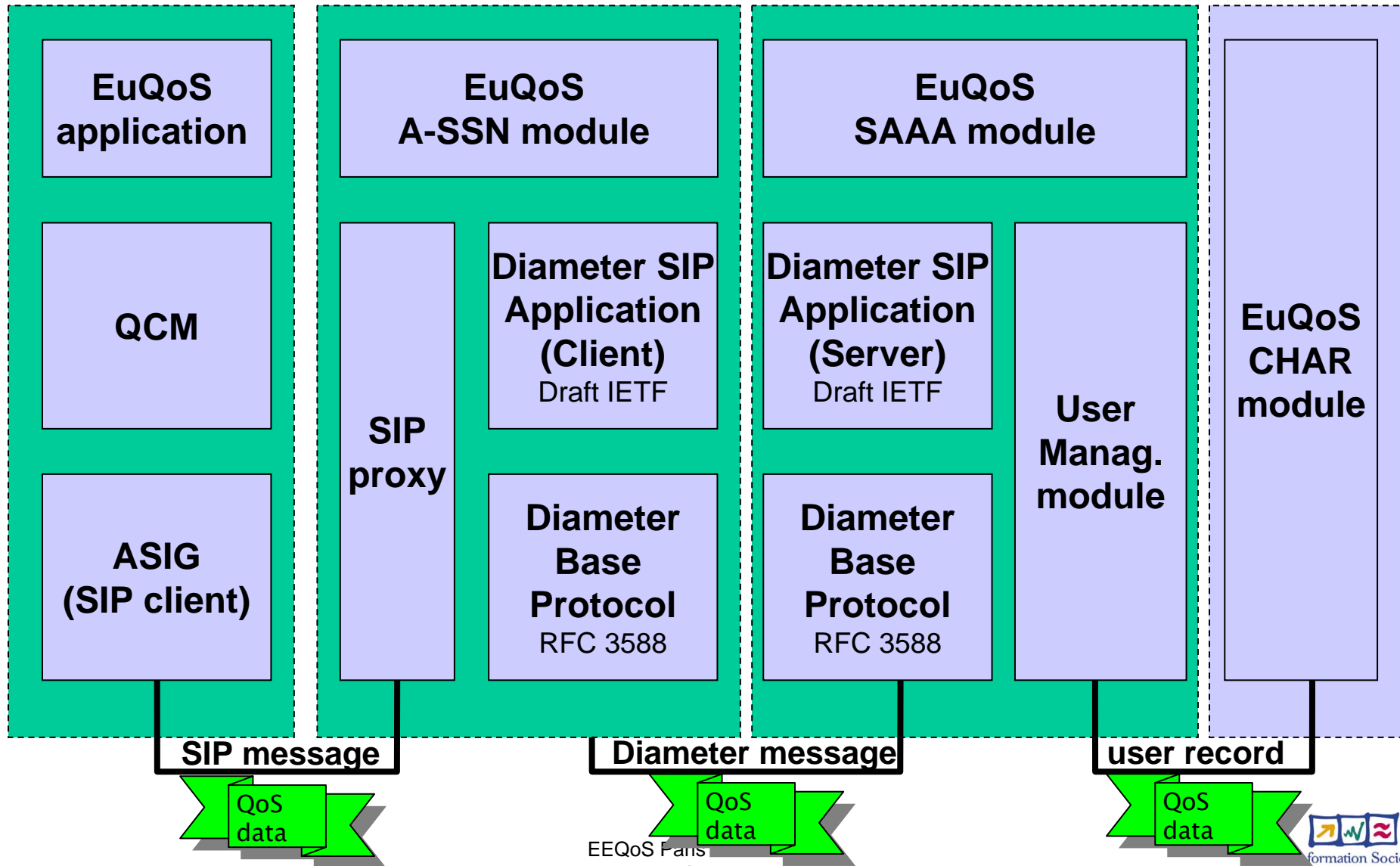
AAA inside EuQoS Architecture



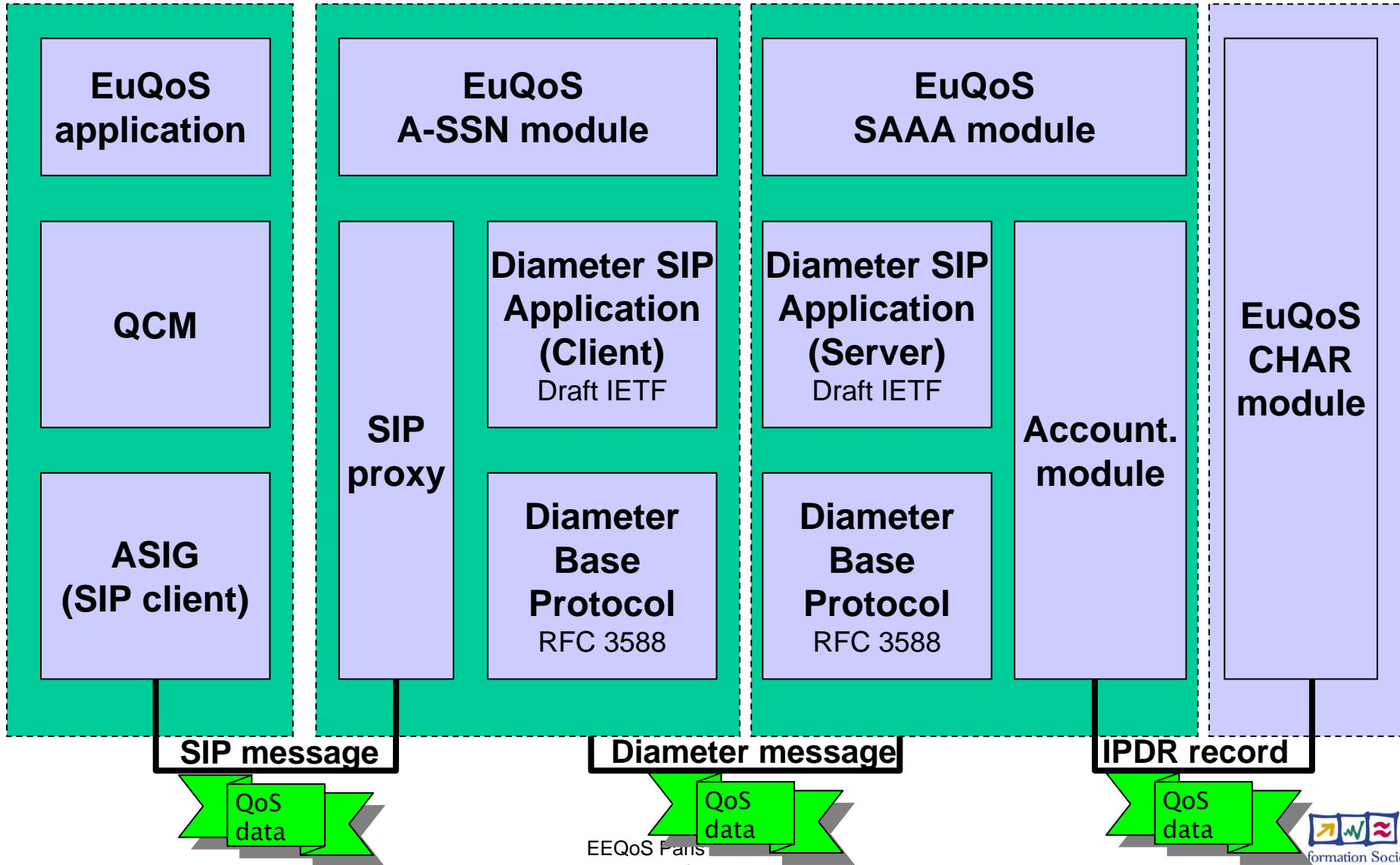
AAA exchanges data with:

- A-SSN (SIP Proxy)
 - to let supplicants(ASIG) and authenticator(AAA) exchange credentials and user-subscribed QoS level
 - to provide grant messages in order to enable user access to EuQoS system with required QoS level
- CHAR (Charging module)
 - to exchange user provisioning, de-provisioning (including subscribed QoS level)
 - to provide accounting IPDR record (including QoS data)
- TERO
 - to provide it with user statistics (no QoS data)

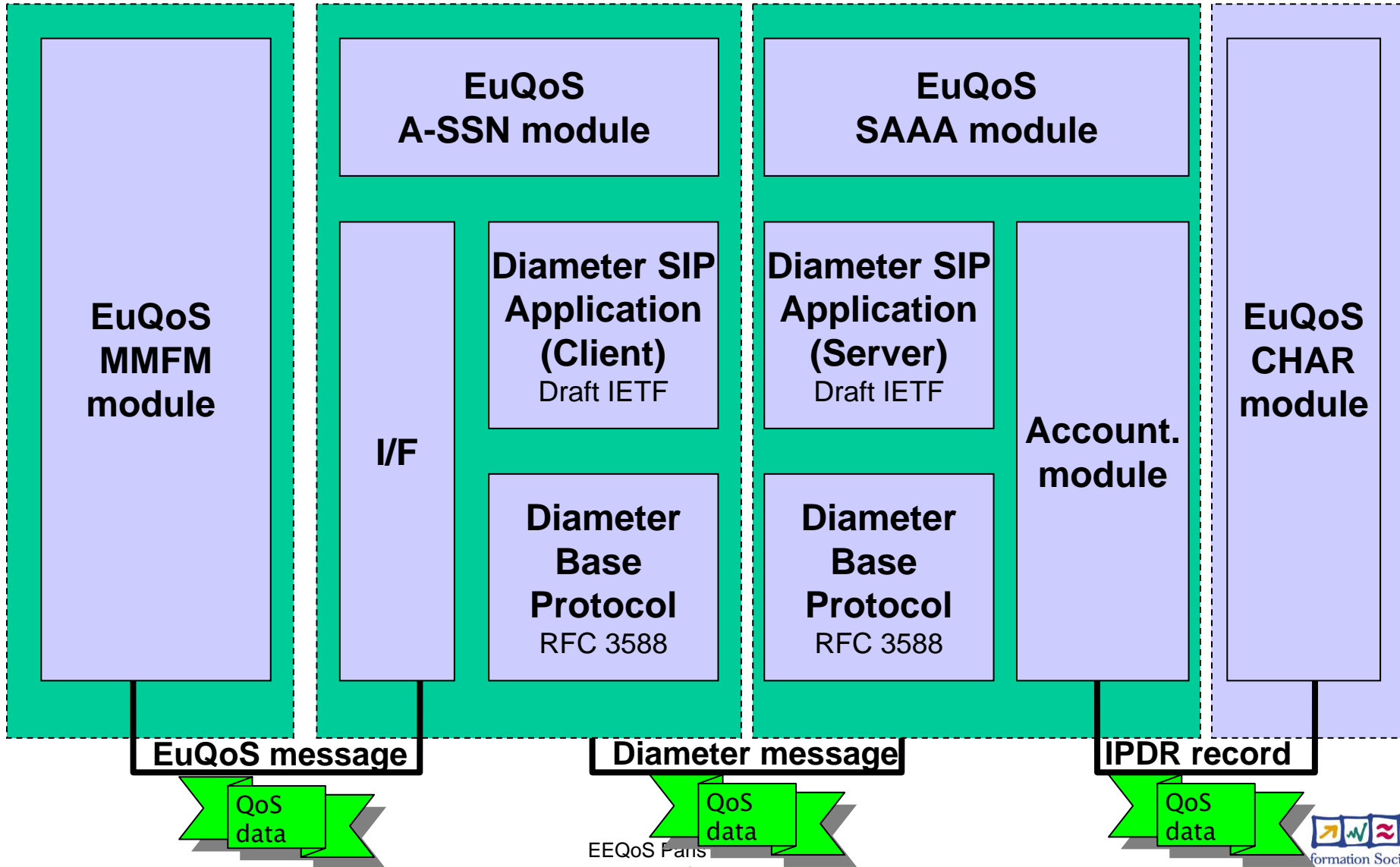
QoS-related provisioning data exchange and logon QoS configuration



QoS-related accounting data exchange - user-triggered events



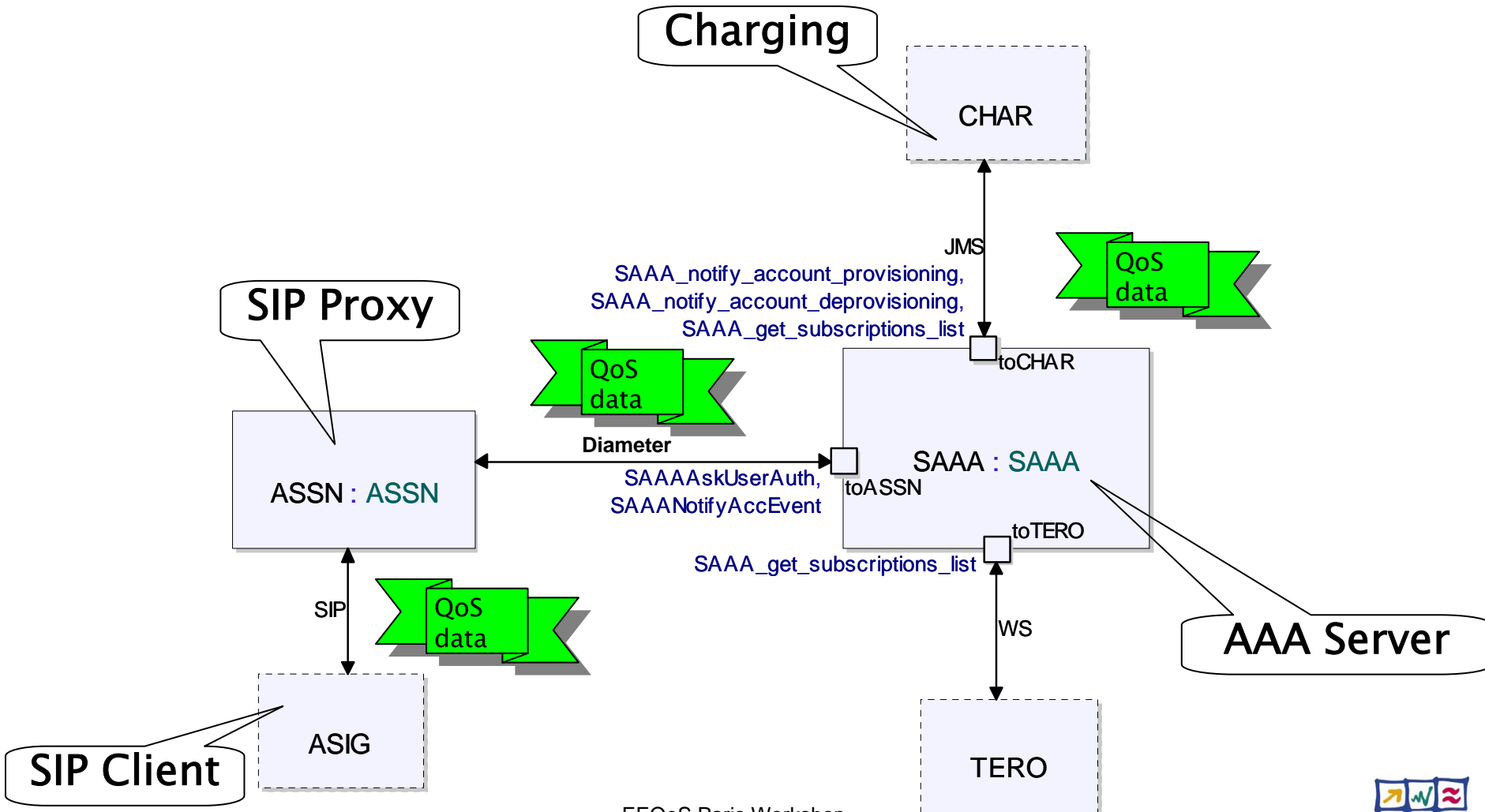
QoS-related accounting data exchange - network-triggered events



AAA Interactions (UML2 composite structure diagram)

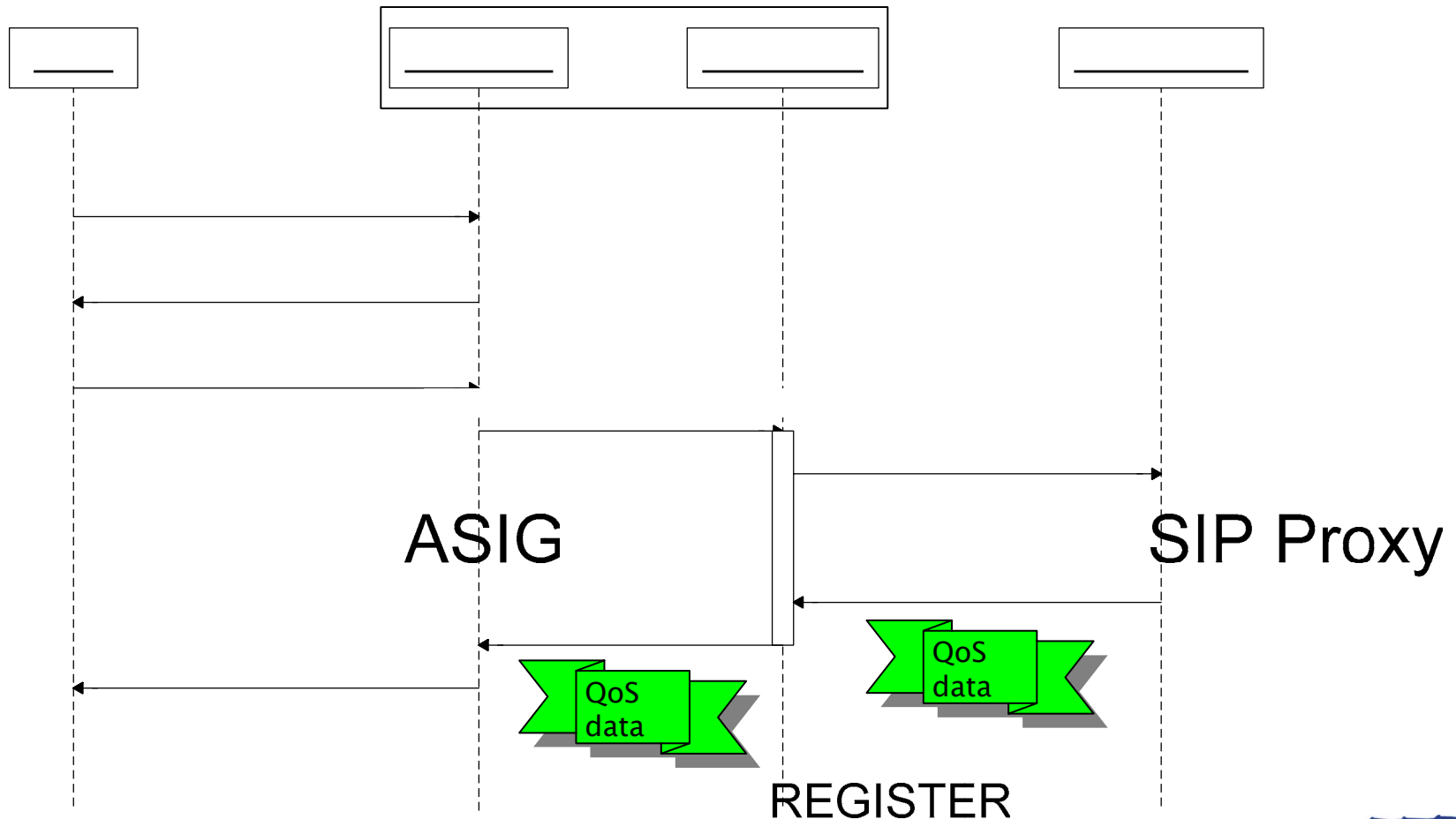
SAAAInteraction

collaboration euqos {1/1}

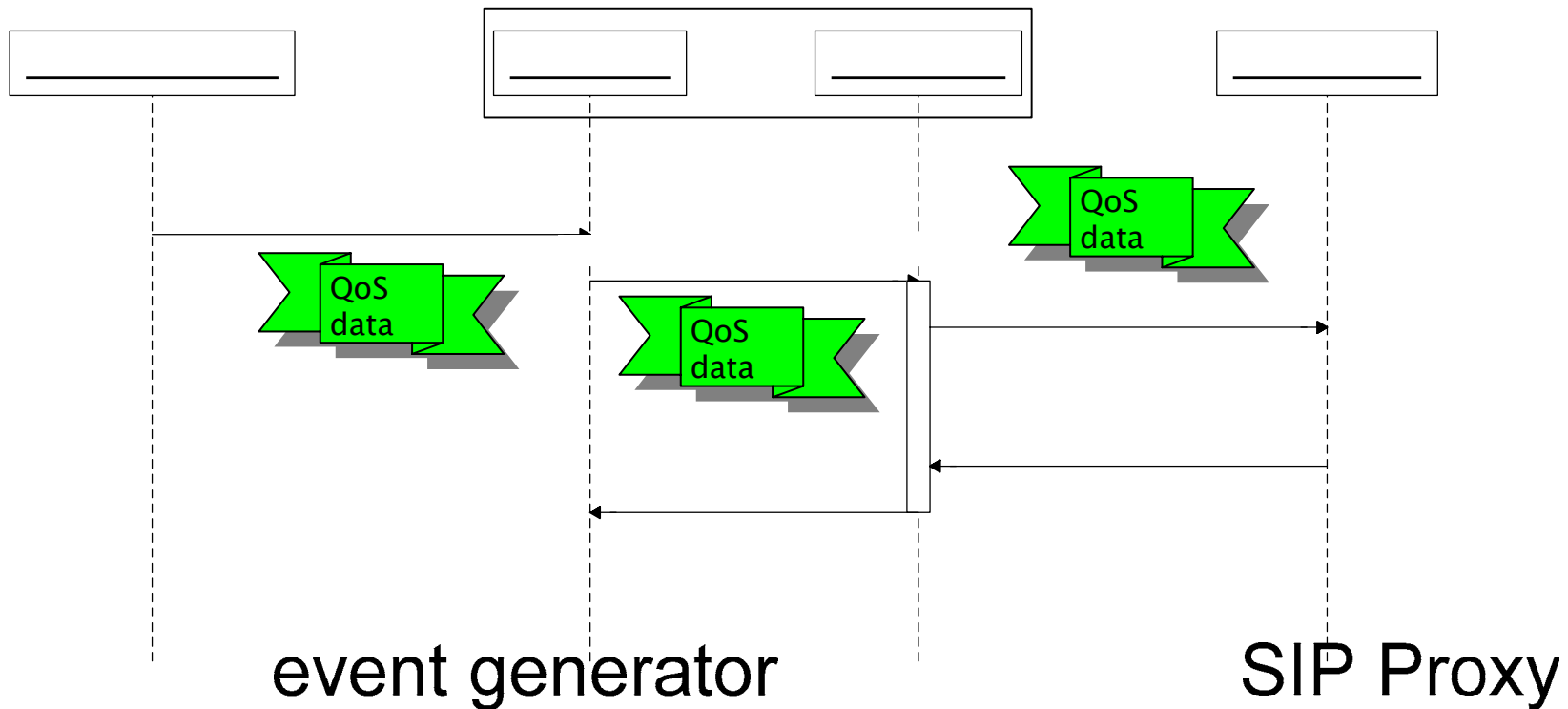


- EQSIP sequence diagrams (including SDP+ messages)
- Diameter AVP-based messages
- IPDR XDR-based records

Authentication/Authorization



Accounting



- SDP QoS extensions includes session QoS requirements
- SDP contains QoS requirements for each media stream that forms a session, one codec associated with each media stream
- QoS requirements can be expressed in terms of different set of QoS data:
 - traffic information
 - peak bit rate (kbit/s)
 - sustainable bit rate (kbit/s)
 - sensitivity information
 - maximum end-to-end delay (ms)
 - maximum end-to-end jitter (ms)
 - maximum packet loss ratio (ms)

Example 1

m=audio 49172 RTP/AVP 0 3

a=rtpmap:0 PCMU/8000

Peak bit rate

a=qos-ti: 29 29

Sustainable bit rate

a=qos-si:si0 80 10 2E-2

e2e max delay

max packet loss

a=qos-si:si1 120 20 2E-2

e2e max jitter

a=rtpmap:3 GSM/8000

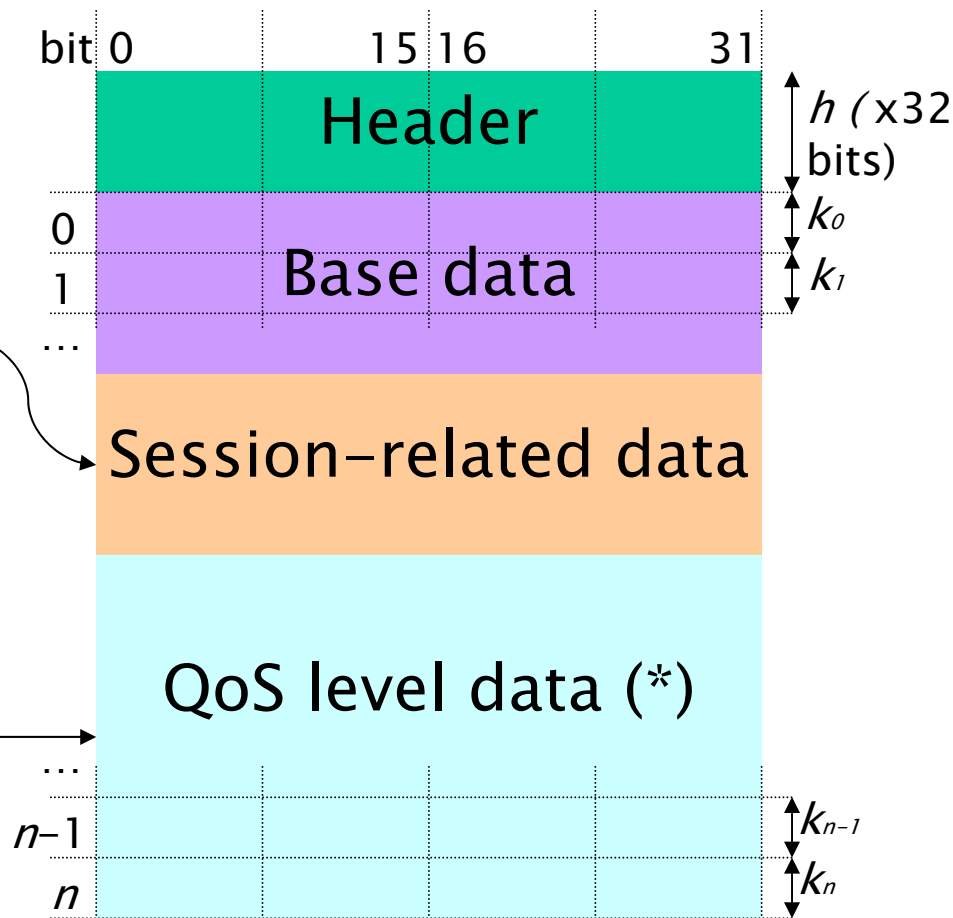
a=qos-ti: 13 13

a=qos-si:si0 60 20 2E-4

Principal AVPs (Attribute-Value Pairs) contained inside Diameter messages:

- User ID
 - Session ID
 - Application data
 - ...
-
- Granted QoS level data
 - QoS type
 - Related CoS
 - Required resources
 - ...

Diameter message



(*) these new AVPs shall be submitted to IANA for standardization

4. IPDR XDR Example (ipdr.org)

```
<?xml version="1.0" encoding="UTF-8"?>
<IPDRDoc xmlns="http://www.ipdr.org/namespaces/ipdr" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance
xsi:schemaLocation="http://www.ipdr.org/namespaces/ipdr
C:\home\eclipse_workspace\ipdr\EuQoS.xsd" docId="11111111-1111-1111-1111-111111111111">
  <IPDR xsi:type="IPDR-EuQoS-Type">
    <subscriberID>euqos_subscriber</subscriberID>
    <hostname>euqos_client</hostname>
    <ipAddress>10.150.5.161</ipAddress>
    <startTime>2005-03-08T13:20:00Z</startTime>
    <endTime>2005-03-08T17:20:00Z</endTime>
    <timestamp>2005-03-08T17:20:30Z</timestamp>
    <sentBytes>10000</sentBytes>
    <receivedBytes>10001</receivedBytes>
    <duration>60</duration>
    <disconnectReason>Normal</disconnectReason>
    <QoSRequested.profileName>Gold</QoSRequested.profileName>
    <QoSRequested.maxGuaranteedBw>200</QoSRequested.maxGuaranteedBw>
    <QoSRequested.maxDelay>50</QoSRequested.maxDelay>
    <QoSRequested.maxJitter>0</QoSRequested.maxJitter>
    <QoSRequested.networkService>BE</QoSRequested.networkService>
    <QoSDelivered.profileName>Silver</QoSDelivered.profileName>
    <QoSDelivered.maxGuaranteedBw>100</QoSDelivered.maxGuaranteedBw>
    <QoSDelivered.maxDelay>40</QoSDelivered.maxDelay>
    <QoSDelivered.maxJitter>0</QoSDelivered.maxJitter>
    <QoSDelivered.networkService>BE</QoSDelivered.networkService>
  </IPDR>
</IPDRDoc>
```

4. IPDR XDR Schema (1 / 3)

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.ipdr.org/namespaces/ipdr" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ipdr="http://www.ipdr.org/namespaces/ipdr"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="0.2">
  <include schemaLocation="IPDRDoc3.5.1.xsd"/>
  <element name="accountingRecordType">
    <simpleType>
      <restriction base="string">
        <enumeration value="Start"/>
        <enumeration value="Stop"/>
        <enumeration value="Interim"/>
      </restriction>
    </simpleType>
  </element>
  <element name="EuQoSSessionID" type="string">
</element>
  <element name="userName" type="string">
</element>
  <element name="hostname" type="string">
</element>
  <element name="callingIPAddress" type="ipdr:ipV4Addr">
</element>
  <element name="calledIPAddress" type="ipdr:ipV4Addr">
</element>
  <element name="callingID" type="string"/>
  <element name="calledID" type="string"/>
  <element name="applicationType" type="string">
</element>
  <element name="authenticatorIPAddress" type="ipdr:ipV4Addr">
</element>
  <element name="startTime" type="ipdr:dateTimeMsec">
</element>
  <element name="endTime" type="ipdr:dateTimeMsec">
</element>
  <element name="timestamp" type="ipdr:dateTimeMsec">
</element>
  <element name="sentBytes" type="int">
</element>
  <element name="receivedBytes" type="int" />
</element>
  <element name="duration" type="int">
</element>
  <element name="disconnectReason">
    <simpleType>
      <restriction base="string">
        <enumeration value="Normal"/>
        <enumeration value="Client Failure"/>
        <enumeration value="Server Failure"/>
        <enumeration value="Connection Failure"/>
        <enumeration value="Other Failure"/>
      </restriction>
    </simpleType>
  </element>
</schema>
```

4. IPDR XDR Schema (2 / 3)

```
<!--QoS requested-->
  <element name="QoSRequested.profileName" type="string">
  </element>
  <element name="QoSRequested.maxGuaranteedBw" type="int">
  </element>
  <element name="QoSRequested.maxDelay" type="int">
  </element>
  <element name="QoSRequested.maxJitter" type="int"/>
  <element name="QoSRequested.networkService">
    <simpleType>
      <restriction base="string">
        <enumeration value="RT"/>
        <enumeration value="BE"/>
        <enumeration value="NC"/>
      </restriction>
    </simpleType>
  </element>
  <!-- QoS Delivered -->
  <element name="QoSDelivered.profileName" type="string">
  </element>
  <element name="QoSDelivered.maxGuaranteedBw" type="int">
  </element>
  <element name="QoSDelivered.maxDelay" type="int">
  </element>
  <element name="QoSDelivered.maxJitter" type="int"/>
  <element name="QoSDelivered.networkService">
    <simpleType>
      <restriction base="string">
        <enumeration value="RT"/>
        <enumeration value="BE"/>
        <enumeration value="NC"/>
      </restriction>
    </simpleType>
  </element>
  <!-- Start of Record structure -->
  <complexType name="IPDR-EuQos-Type">
    <complexContent>
      <extension base="ipdr:IPDRType">
        <sequence>
          <element ref="ipdr:accountingRecordType"/>
          <element ref="ipdr:EuQoSSessionID"/>
          <element ref="ipdr:userName"/>
          <element ref="ipdr:hostname" minOccurs="0"/>
          <element ref="ipdr:callingIPAddress" minOccurs="0"/>
          <element ref="ipdr:calledIPAddress" minOccurs="0"/>
          <element ref="ipdr:callingID"/>
          <element ref="ipdr:calledID"/>
          <element ref="ipdr:applicationType"/>
          <element ref="ipdr:authenticatorIPAddress"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

```

4. IPDR XDR Schema (3 / 3)

```
<element ref="ipdr:authenticatorIPAddress"/>
<element ref="ipdr:startTime" minOccurs="0"/>
<element ref="ipdr:endTime" minOccurs="0"/>
<element ref="ipdr:timestamp"/>
<element ref="ipdr:sentBytes" minOccurs="0"/>
<element ref="ipdr:receivedBytes" minOccurs="0"/>
<element ref="ipdr:duration" minOccurs="0"/>
<element ref="ipdr:disconnectReason" minOccurs="0"/>
<element ref="ipdr:QoSRequested.profileName" minOccurs="0"/>
<element ref="ipdr:QoSRequested.maxGuaranteedBw" minOccurs="0"/>
<element ref="ipdr:QoSRequested.maxDelay" minOccurs="0"/>
<element ref="ipdr:QoSRequested.maxJitter" minOccurs="0"/>
<element ref="ipdr:QoSRequested.networkService" minOccurs="0"/>
<element ref="ipdr:QoSDelivered.profileName" minOccurs="0"/>
<element ref="ipdr:QoSDelivered.maxGuaranteedBw" minOccurs="0"/>
<element ref="ipdr:QoSDelivered.maxDelay" minOccurs="0"/>
<element ref="ipdr:QoSDelivered.maxJitter" minOccurs="0"/>
<element ref="ipdr:QoSDelivered.networkService" minOccurs="0"/>
```

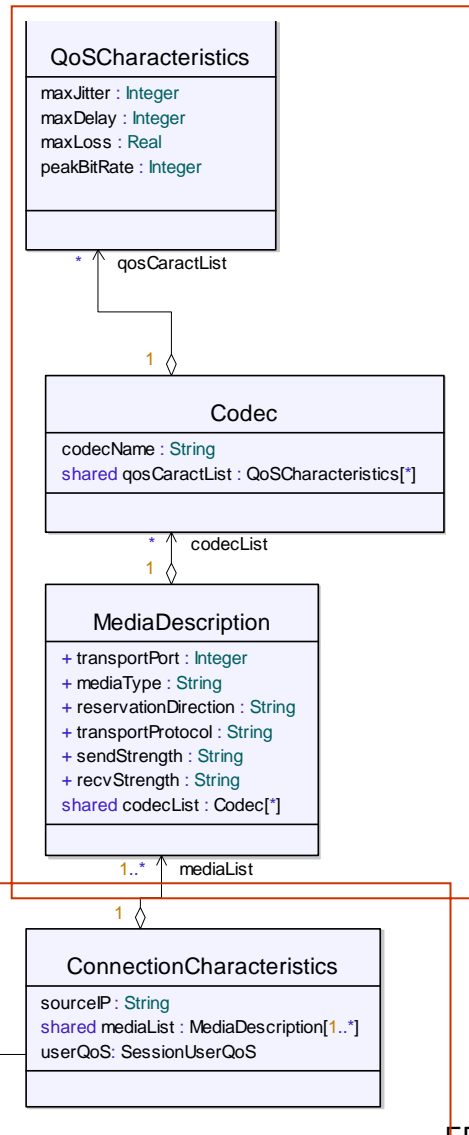
</sequence>

</extension>

</complexContent>

</complexType>

</schema>



QoS Data set managed by EuQoS

- ConnectionCharacteristics

It links logical QoS name

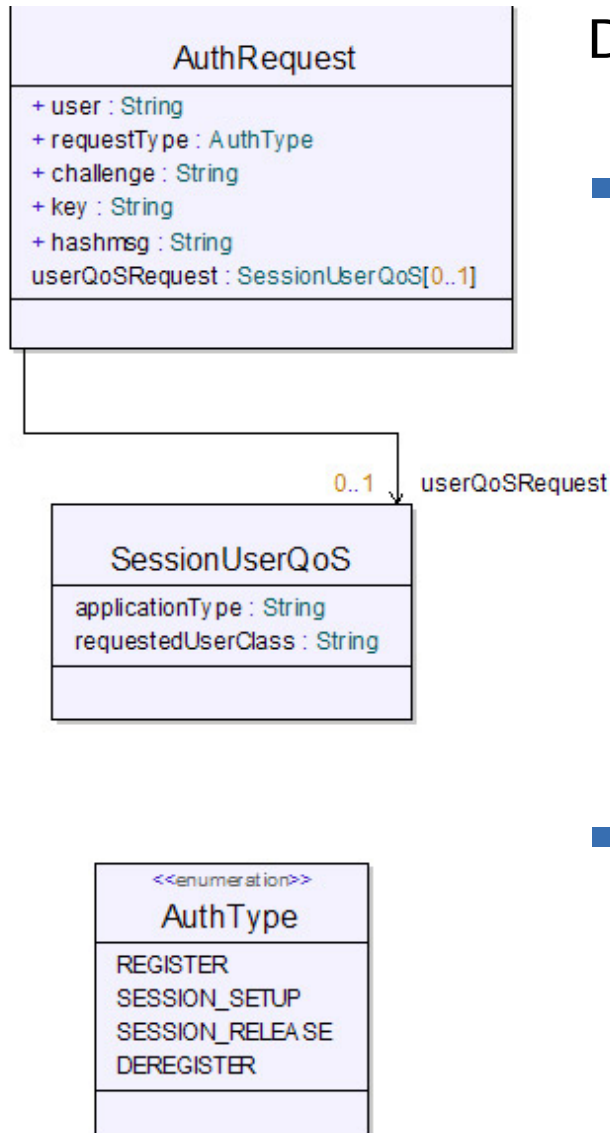
- SessionUserQoS structure

To physical characteristics

- MediaDescription (media characteristics)

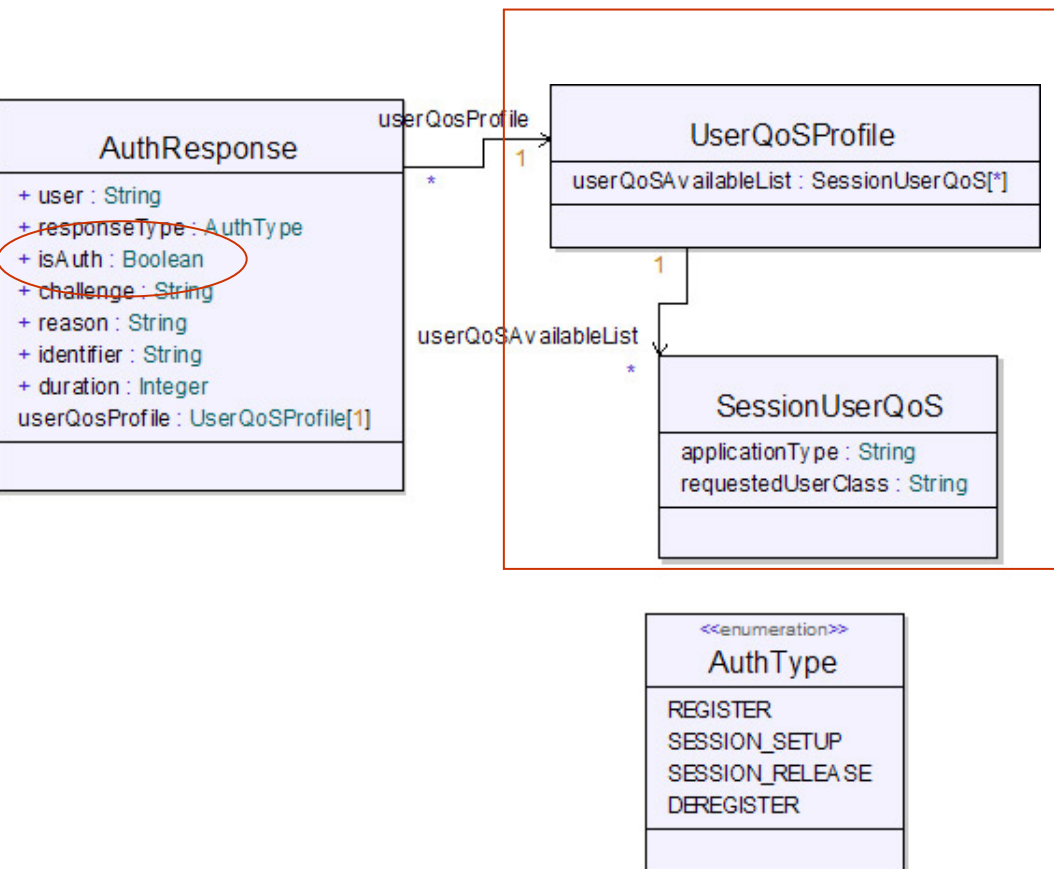
- Codec

- QoSCharacteristics (network parameters)



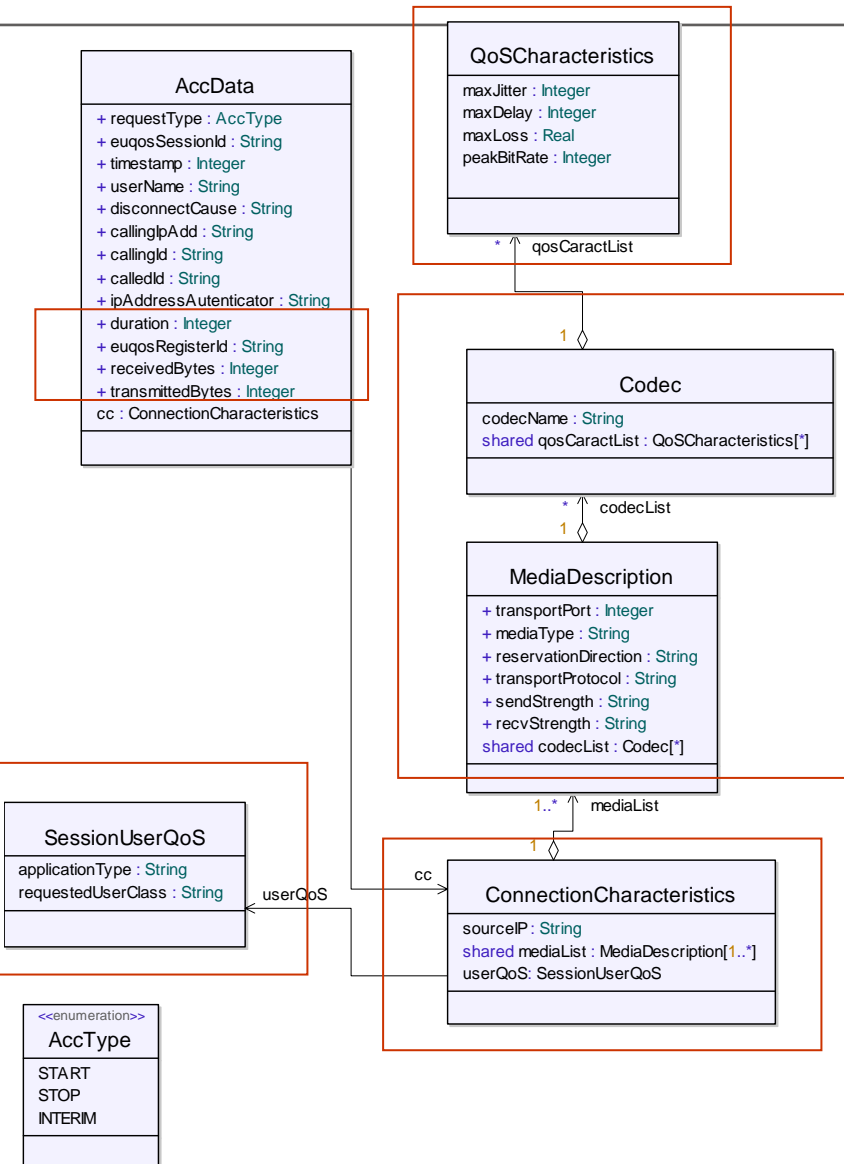
Data sent by A-SSN to SAAA for authentication/authorization requests

- **AuthRequest:**
 - user: the EuQoS username the user wants to register/start a new session.
 - requestType: REGISTER, SESSION_SETUP, SESSION_RELEASE, or DEREGISTER.
 - challenge: string generated from A-SSN for security purposes
 - key: a string produced by a proper algorithm.
 - hashmsg: a string produced by a proper function on to the original user message for security purposes.
 - userQoSRequest:
- **SessionUserQoS:**
 - applicationType: describes the type of application.
 - requestedUserClass: logical name of the class requested for the session.



SAAA answers with:

- **AuthResponse:**
 - `isAuth`: a boolean value indicating if the Authentication/Authorization succeed.
- **UserQoSProfile:**
 - The list of application types and user classes available for the user by his subscription.



Main accounting information:

- duration, receivedBytes, transmitted bytes.
- connection characteristics: a set of fields describing:
 - the QoS used
 - the media used (audio/video, codec, ..)
 - the related network resources (max jitter, max delay, ...)

| Message | From | To | Description |
|------------------------------------|-------|------|--|
| SAAA_notify_account_provisioning | CHAR | SAAA | Notify/Update user data (including QoS data) |
| SAAA_notify_account_deprovisioning | CHAR | SAAA | Remove user data |
| SAAA_get_accounting_records | CHAR | SAAA | Extracts accounting events |
| SAAAAskUserAuth | A-SSN | SAAA | Notify user logon request |
| SAAANotifyAccEvent | A-SSN | SAAA | Notify accounting event (both user-required and network related) |

SAAA is composed by:

- AAA Server
 - performs authentication, authorization, accounting (based on QoS data)
- Diameter Core
 - accomplishes DIAMETER message exchange (compliant to RFC3588) by implementing new AVPs
- User Repository Manager
 - in charge of storing and managing user information (including QoS data)
- Accounting Module
 - collects accounting information, stores QoS data in IPDR records and sends them to CHAR

Distributed applications

Java Diameter Client

Java Diameter Server

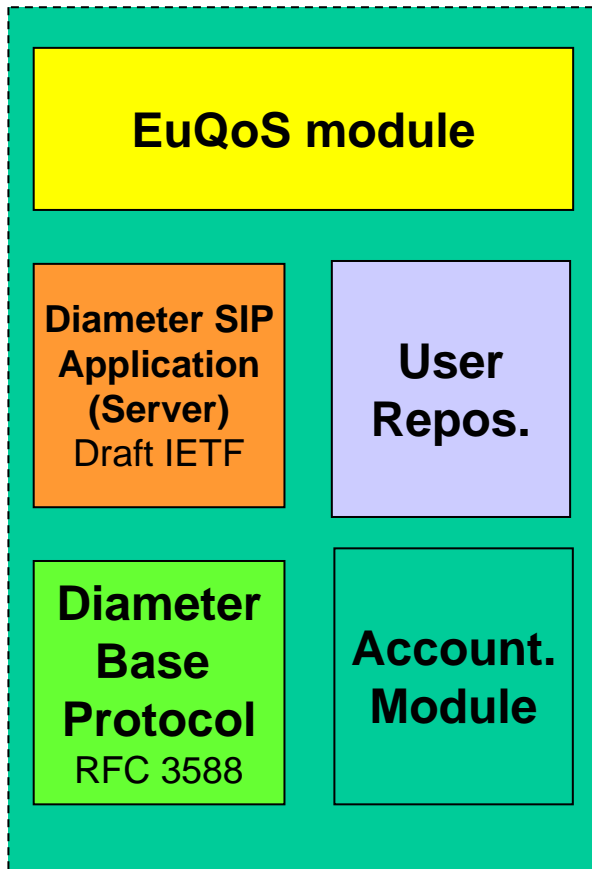
Java Diameter API

Diameter base protocol impl.

IP Network

Diameter Core

SAAA



• Diameter base protocol

Provides basic services for user session authentication and accounting

Defines and delivers AVPs (attribute value pairs) for basic applications

Is extensible through addition of new AVPs and commands

Provides roaming support

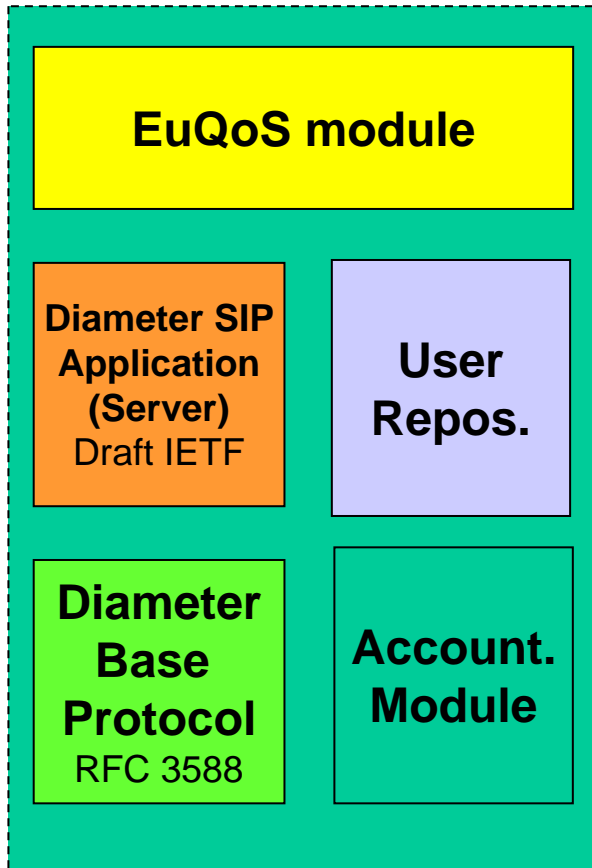
• Diameter SIP application protocol

Allows Diameter clients to request authentication and authorization in conjunction with SIP (IETF Draft)

Extends Diameter with new commands and new AVPs

In EuQoS SAAA it also performs accounting facilities

SAAA



• User Repository

Stores user data (including subscribed QoS level)

Is provisioned by Charging module

• Accounting Module

Provides Charging module with user accounting record

Builds IPDR data format

• EuQoS

Defines end to end QoS requirements

Thank you