

# 3GPP TR 29.864 V8.0.1 (2008-12)

---

*Technical Report*

## **3<sup>rd</sup> Generation Partnership Project; Technical Specification Group Core Network and Terminals; Study on IMS Application Server Service Data Descriptions for AS interoperability (Release 8)**



The present document has been developed within the 3<sup>rd</sup> Generation Partnership Project (3GPP<sup>TM</sup>) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPP only. The Organizational Partners accept no liability for any use of this Specification. Specifications and reports for implementation of the 3GPP<sup>TM</sup> system should be obtained via the 3GPP Organizational Partners' Publications Offices.

---

Keywords

---

IP, multimedia, network

**3GPP**

Postal address

---

3GPP support office address

---

650 Route des Lucioles – Sophia Antipolis  
Valbonne – France  
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

---

<http://www.3gpp.org>

---

**Copyright Notification**

---

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2008, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TTA, TTC).  
All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI currently being registered for the benefit of its Members and of the 3GPP Organizational Partners  
GSM® and the GSM logo are registered and owned by the GSM Association

# Contents

Foreword .....	6
Introduction .....	6
1 Scope .....	7
2 References.....	7
3 Definitions, symbols and abbreviations .....	8
3.1 Definitions .....	8
3.2 Abbreviations.....	9
4 Requirements for defining Application Server Specific Service Data.....	10
4.1 Introduction .....	10
4.2 Requirements for AS Interoperability .....	10
4.3 Flow of Information between HSS and AS .....	10
4.4 Role of Sh interface for IMS and non-IMS services .....	10
4.5 Reuse and Extensibility .....	10
4.6 User Re-homing and Load Balancing .....	11
5 Mechanisms for Transfer of Defined Service Data between Application Server and the HSS .....	11
5.1 Introduction .....	11
5.2 Defined features.....	11
5.3 Data sent over Sh in an uncompressed format .....	12
5.4 Data sent over Sh in compressed format .....	12
5.5 Location of compression options .....	13
5.6 Information Content vs. representation .....	13
5.7 Evaluation of compression options .....	15
5.7.1 Introduction.....	15
5.7.2 Architecture aspects.....	15
5.7.3 Compression algorithm discussion .....	15
5.8 Application Server Service Data Descriptions (ASSDD) Reference point.....	16
6 Telephony Application Server service data definition .....	17
6.1 Defined Data.....	17
6.1.0 Introduction.....	17
6.1.1 Explicit Communication Transfer (ECT).....	17
6.1.2 Communication Diversion (CDIV).....	17
6.1.2.1 CF Busy.....	17
6.1.2.2 CF No Reply /No Response.....	18
6.1.2.3 CF Not Logged In (CFNL) .....	18
6.1.2.4 CF All/CFU.....	19
6.1.2.5 CF Not Reachable (CFNRc) .....	20
6.1.2.6 Communication Deflection .....	20
6.1.2.7 Communication Diversion Notification .....	21
6.1.3 Three-Party (3PTY)/ CONFerence (CONF) .....	21
6.1.4 ID Presentation Services .....	21
6.1.4.1 OIP.....	21
6.1.4.2 OIR.....	21
6.1.4.3 TIP .....	22
6.1.4.4 TIR.....	22
6.1.5 Communication Barring.....	22
6.1.6 Call blocking .....	23
6.1.7 Malicious Communication IDentification (MCID).....	23
6.1.8 Anonymous Communication Rejection (ACR) .....	23
6.1.9 Communications Waiting (CW) .....	23
6.1.10 Communication HOLD (HOLD) .....	23
6.1.11 Message Waiting Indication (MWI) .....	24
6.1.12 Advice Of Charge (AOC).....	24

6.1.13	Reverse charging .....	24
6.1.14	Closed User Group (CUG) .....	24
6.1.15	Flexible Alerting (FA).....	24
6.1.16	Completion of Communications to Busy Subscriber (CCBS)/Completion of Communications on No Reply (CCNR) .....	24
7	Reuse and extension .....	25
7.1	Introduction .....	25
7.2	Extensibility of IMS Application Server Service Data Descriptions .....	25
7.2.1	Introduction.....	25
7.2.2	Service indication per MMTel group/suite of service .....	25
7.2.3	Extensibility within defined datasets.....	26
7.2.4	Service indications per service within MMTel group/suite of services.....	26
7.2.5	Recommendations .....	28
7.2.6	Syntactic and Semantic recognition .....	28
7.2.7	Extensibility for information efficient data definitions.....	28
7.2.8	Backward Compatibility for information efficient data definitions .....	30
7.2.8.1	Introduction.....	30
7.2.8.2	Releases in databases .....	30
7.2.8.3	New releases.....	31
8	Alternative solutions .....	31
8.1	Introduction .....	31
8.2	A Binary Storage Approach to Defining IMS Service Data .....	31
8.2.0	Service Data Description .....	31
8.2.1	Services Data Definition .....	33
8.2.2	DMTAS_FORWARD_TO.....	35
8.2.3	DMTAS_DN.....	36
8.2.4	DMULONG .....	36
8.2.5	DMUSHORT .....	37
8.2.6	DMUBYTE.....	37
8.2.7	DMTAS_VARBYTE .....	37
8.2.8	DMTAS_VARSTRING.....	38
8.2.9	DMTAS_URI.....	38
8.2.10	HEADER WITH LENGTH .....	38
8.2.11	TAS_SERVICE_STAT.....	39
8.2.12	DM_TAS_CALLING_ID_RESTRICT_ENUM .....	39
8.2.13	DM_TAS_CF_EDITING .....	40
8.2.14	DM_TAS_CF_MODE .....	41
8.2.15	DM_TAS_SERVICE_ID.....	42
8.2.16	DM_CALL_CATEGORY.....	43
8.2.17	DM_CALL_BARRING_PARM .....	44
8.2.18	DM_CALL_BLOCKING_PARM .....	44
8.2.19	DM_CALL_TRANSFER_PARM .....	45
8.2.20	DM_CALLING_ID_PARM .....	46
8.2.21	DM_CF_ALL_PARM .....	47
8.2.22	DM_REVEAL_URI_OPTIONS .....	48
8.2.23	DM_TAS_FORWARD_TO_PRESENTATION_OPTIONS .....	49
8.2.24	DM_CF_BUSY_PARM .....	49
8.2.25	DM_CF_NO_RESPONSE_PARM .....	50
8.2.26	DM_MULTIWAY_PARM .....	51
8.3	XML based data description.....	52
8.3.1	General proposal .....	52
8.3.2	Required changes to standards.....	52
9	Conclusions and recommendations .....	52
9.1	Discussion.....	52
9.1.1	Summary .....	52
9.1.2	Memory Consumption.....	53
9.1.3	AS Processing Efficiency .....	53
9.1.4	Service Flexibility and Interoperability Concerns .....	53
9.1.5	Future Standardization .....	54
9.2	Recommendations .....	54

**Annex A: Change history** .....55

---

---

## Foreword

This Technical Report has been produced by the 3<sup>rd</sup> Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

---

## Introduction

An Application Server service data representation is identified specific to an IMS telephony application server.

---

# 1 Scope

The present document studies requirements for application server service data definition. It also studies the transfer of the defined service data between an application server and the HSS. Further, it exposes a service data definition for a telephony application server for services as defined in 3GPP TS 22.173 [3]. Finally it provides recommendations for how the framework can be reused and extended for future IMS service definitions.

---

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
  - For a specific reference, subsequent revisions do not apply.
  - For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.
- [1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications"
- [2] ETSI TS 183 023 V1.3.1 (2007-04): "TISPAN PSTN/ISDN simulation services; Extensible Markup Language (XML) Configuration Access Protocol (XCAP) over the Ut interface for Manipulating NGN PSTN/ISDN Simulation Services"
- [3] 3GPP TS 22.173: "IMS Multimedia Telephony Communication Service and Supplementary Services, Stage 1";
- [4] 3GPP TS 24.604: "Communication Diversion (CDIV) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification"
- [5] 3GPP TS 24.605: "Conference (CONF) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification"
- [6] 3GPP TS 24.606: "Message Waiting Indication (MWI) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification"
- [7] 3GPP TS 24.607: "Originating Identification Presentation (OIP) and Originating Identification Restriction (OIR) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification"
- [8] 3GPP TS 24.608: "Terminating Identification Presentation (TIP) and Terminating Identification Restriction (TIR) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification"
- [9] 3GPP TS 24.629: "Explicit Communication Transfer (ECT) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification"
- [10] 3GPP TS 24.173 V7.1.0 (2007-06): "IMS Multimedia Telephony Communication Service and Supplementary Services; Stage 3, (Release 7)"
- [11] 3GPP TS 29.329: "Sh Interface based on the Diameter protocol; Protocol details"
- [12] IETF RFC 20: "ASCII format for Network Interchange, Oct 1969"
- [13] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies November 1996."
- [14] IETF RFC 3548: "The Base16, Base32, and Base64 Data Encodings, July, 2003"

- [15] 3GPP TS 24.616: " Malicious Communication Identification (MCID) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [16] 3GPP TS 24.611: " Anonymous Communication Rejection (ACR) and Communication Barring (CB) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [17] 3GPP TS 24.647: " Advice Of Charge (AOC) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [18] 3GPP TS 24.615: " Communication Waiting (CW) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [19] 3GPP TS 24.610: " Communication Hold (HOLD) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [20] 3GPP TS 23.228: " IP Multimedia Subsystem (IMS); Stage 2 "
- [21] 3GPP TS 24.239, " Technical Specification Group Core Network and Terminals; Flexible Alerting (FA) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [22] ITU I.256.3, " ISDN General Structure and Service Capabilities; Reverse Charging "
- [23] 3GPP TS 24.642 " Completion of Communications to Busy Subscriber (CCBS) and Completion of Communications by No Reply (CCNR) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [24] 3GPP TS 24.654, " Closed User Group (CUG) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification "
- [25] IETF RFC 3485: " The Session Initiation Protocol (SIP) and Session Description Protocol (SDP) Static Dictionary for Signaling Compression (SigComp) "
- [26] ITU-T X.891, " Information technology – Generic applications of ASN.1: Fast Infoset "
- [27] W3C Efficient XML Interchange (EXI) Format 1.0, W3C Working Draft 28 July 2008

---

## 3 Definitions, symbols and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Within GSM MAP, operations and parameters were defined. Parameters rely heavily on sub parameters.

In this service data description mapping, the operations correlate to diameter Sh Read/Write/Subscribe/Notify, and therefore will not be specified.



Table 3.1-1: Definitions

Term	Definition
Database	An independent table that is written to the HSS. This is a collection of data accessed through a common mechanism. It is stored and administered as one entity. Each database has a unique ServiceIndication. Each may be included within a Diameter Sh request, but recent Sh commands allow multiple queries within the same Request. Diameter Sh error scenarios do allow successful read operations with failures simply omitted, so care must be taken when separating data into distinct databases.
Release	The versioning of a database that defines new content definition. Each Release has a release ID and is accompanied by a length parameter so that multiple releases can be parsed even by an AS not familiar with the contents of a specific release. Data contained within a Release can be categorized into various structured types, bitmap, string, and varbyte are some examples.

Releases begin with a release identifier and release length. The length points to the end of data definition for the release. This is either the end of the defined database or the beginning of a subsequent release.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

3PTY	Three-Party Communication
AS	Application Server
ACR	Anonymous Communication Rejection
AOC-C	Advice Of Charge - Charging
AOC-D	Advice Of Charge - During the communication
AOC-E	Advice Of Charge - at the End of the communication
AOC-I	Advice Of Charge - Information
AOC-S	Advice Of Charge - charging information at communication Set-up time
CAT	Customized Alerting Tone
CB	Communication session Barring
CCBS	Completion of Communication sessions to Busy Subscriber
CCNR	Completion of Communication sessions on No Reply
CD	Communication Deflection
CDIV	Communication DIVersion
CDIVN	CDIV Notification
CFB	Communication Forwarding Busy
CFNL	Communication Forwarding on Not Logged-in
CFNR	Communication Forwarding No Reply
CFNRc	Communication Forwarding on Subscriber Not Reachable
CFU	Communication Forwarding Unconditional
COLP	COnnected Line identification Presentation
COLR	COnnected Line identification Restriction
CONF	CONFerence
CUG	Closed User Group
CW	Communication Waiting
DM	Data Module
ECT	Explicit Communication Transfer
FA	Flexible Alerting
GRUU	Globally Routable User agent URI
HOLD	Communication HOLD
ICB	Incoming Communications Barring
MCID	Malicious Communication IDentification
MMTel	MultiMedia Telephony
MWI	Message Waiting Indication
OCB	Outgoing Communications Barring

OIP	Originating Identification Presentation
OIR	Originating Identification Restriction
OSA	Open Service Architecture
PSAP	Public Safety Answer Point
PUID	Public User Identity
SCS	Service Capability Server
SDP	Session Description Protocol
TAS	Telephony Application Server
TIP	Terminating Identification Presentation
TIR	Terminating Identification Restriction

---

## 4 Requirements for defining Application Server Specific Service Data

### 4.1 Introduction

This clause contains a list of the requirements for the solution provided in this study. The primary aim is to define an Application Server service specific schema that facilitates ease of Application Server interoperability in a multi-vendor network. Additional benefits to management plane and service configuration can also be envisioned from an operator perspective.

### 4.2 Requirements for AS Interoperability

Currently, IMS standards define the role of Sh and very generically specify things like transport and protocol. This means things like command sets, schemas, and services are all undefined and proprietary to each vendor. The conclusion is that vendors may end up building proprietary solutions that cannot inter-work. Every other combination represents a separate development as desired by the operator and leads to operator supervising expensive and time consuming inter-operability events.

In summary, responsibility lies with both parties—operators and vendors—to encourage the adoption of standards and the preservation of interoperability. If this issue is handled correctly, both benefit: operators become more productive and drive down costs and vendors are able to compete on the quality of their product.

### 4.3 Flow of Information between HSS and AS

It is vital to have a clear definition of the services and the nature of service information exchanged between HSS and AS. This implies that specifying the service data schema will also provide guidance on information exchanged over the Sh interface.

### 4.4 Role of Sh interface for IMS and non-IMS services

Sh is defined as a reference Point between an AS (SIP-AS or OSA-SCS) and an HSS. Sh interface allows access to non-transparent data which represents the state, location of subscribers in CS, PS and IMS domains and has a defined schema providing IMS applications access to HLR/HSS service data. Additionally, it allows access to transparent data which can represent anything. 3GPP does not impose a schema on this data, it is treated at the HSS as data blob and mechanisms are defined to allow access and sharing by multiple application servers. There should be methods to define a standardised AS service data schema for services that are transported within the data blob, which will help in satisfying the architectural requirements.

### 4.5 Reuse and Extensibility

In solutions provided compatibility with existing service data definitions must be considered. Additional Application Server service extensions, if defined should allow co-existence with current service data schemes. Section 8 discusses data extension and backward compatibility issues in further detail.

## 4.6 User Re-homing and Load Balancing

To facilitate re-homing of users to application servers and load balancing, the definition of IMS application server service data descriptions increases the opportunity for hosting services on alternative application servers.

---

# 5 Mechanisms for Transfer of Defined Service Data between Application Server and the HSS

## 5.1 Introduction

There are several forms in which the service data could be transferred between the AS and HSS over Sh interface. The data sent may be in a compressed form or not depending on the nature of compressions algorithm supported by HSS and AS.

Definition refers to a set of attributes supporting application functionality and how they are to be commonly interpreted or encoded by a compliant AS. Compression, defines the mechanism of reducing the definition to transmit and store smaller versions of the data definition.

## 5.2 Defined features

The data to support Telephony application server features are identified in the following list taken from 3GPP TS 22.173 [3]. The feature list may be expanded in the future, but initially will contain the following capabilities:

- Originating Identification Presentation (OIP)
- Originating Identification Restriction (OIR)
- Terminating Identification Presentation (TIP)
- Terminating Identification Restriction (TIR)
- Malicious Communication Identification (MCID)
- Anonymous Communication Rejection (ACR)
- Communication DIVersion (CDIV)
- Communication Waiting (CW)
- Communication HOLD (HOLD)
- Communication Barring (CB)
- Completion of Communications to Busy Subscriber (CCBS)
- Completion of Communications No Reply (CCNR)

NOTE: CCNR is not listed in 3GPP TS 22.173 [3]

- Message Waiting Indication (MWI)
- CONFerence (CONF)
- Three-Party (3PTY)
- Advice Of Charge (AOC)
- Explicit Communication Transfer (ECT)
- Reverse charging

- Closed User Group (CUG)
- Flexible Alerting (FA)
- Customized Alerting Tone (CAT)

NOTE: CAT is not listed in 3GPP TS 22.173 [3]

Conveying this data from an application server to the HSS requires definition of the details for each of the features and is implemented within the application server.

### 5.3 Data sent over Sh in an uncompressed format

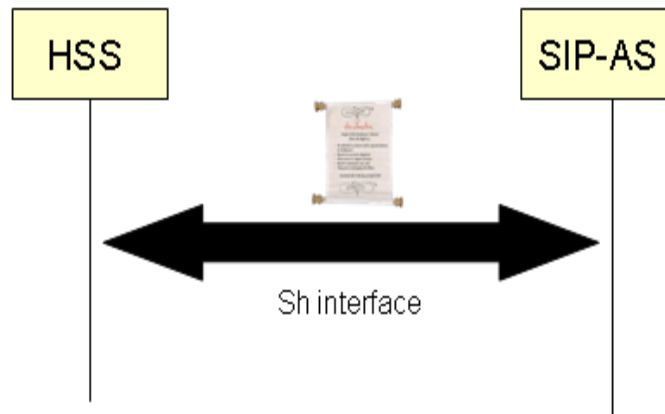


Figure 5.3-1: AS service data exchanged in uncompressed format

Tradeoffs between transmission rate and compression/decompression processing time should be considered. It is possible to operate an interface at a higher data rate by transmitting uncompressed data. This would be the case if transmitting compressed data required significant overhead time due to the invocation of the compression/decompression component. Transmitting uncompressed data may allow for faster session processing (system does not need to employ conversion algorithms before operating on the data) where the size of data being transferred over the interface is not critical. This compression/decompression activity may not be required if the AS stores this data in an uncompressed form (e.g. xml documents).

### 5.4 Data sent over Sh in compressed format

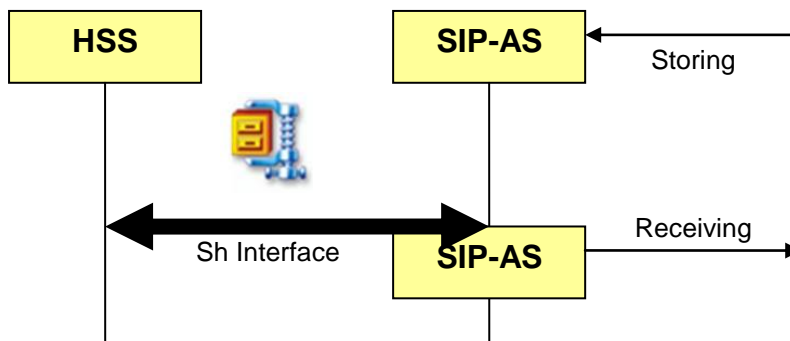


Figure 5.4-1: AS service data exchanged in compressed format

This mechanism works on the basis that the storing AS and the AS recalling data understand the encoding schemes and compression algorithm. Compression is useful in order to save memory space and transmission bandwidth. The possible fallout may include excessive processing which may cause problems with certain applications. The AS could carry out

this negotiation through transparent data content to indicate upon a commonly supported compression/decompression algorithm. An alternative method could be the use of O&M to ensure a common compression/decompression algorithm

The disadvantage of this proposal is that they could involve heavy standardisation work to co-ordinate compression/decompression algorithms across all Application Servers.

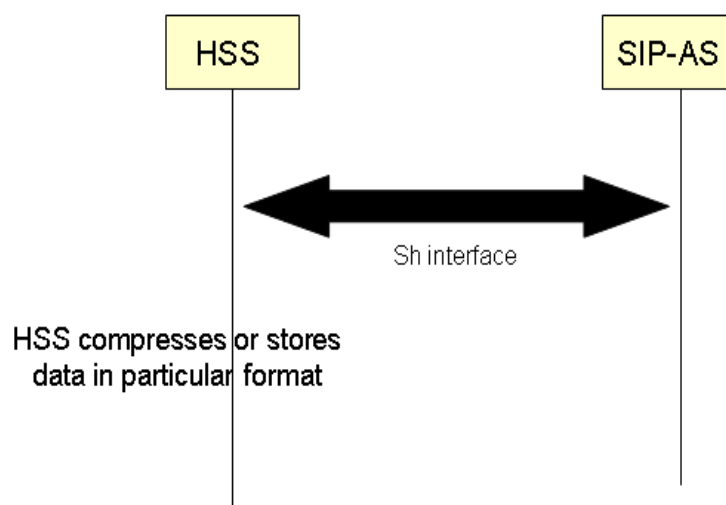
## 5.5 Location of compression options

There are two alternatives for the location of compression/decompression algorithms.

- Option A: compression/decompression algorithms at AS

This allows easy extension of XML increasing overall network efficiency. However to satisfy, AS interoperability requirement this would require heavy standardisation efforts to include different compression/decompression algorithms. This option is applicable to Figure 5.4-1.

- Option B: compression/decompression algorithms at HSS.



**Figure 5.5-1: HSS storage of AS service data**

This option is independent of the nature of data exchanged over the Sh interface. In this option the compression/decompression algorithm could operate on data received at the HSS over the Sh. The HSS could have knowledge of the data being received and could choose a compression algorithm based upon the type of data, or instance of data being transmitted. This is not advisable, as the App Server and HSS would need to be coordinated requiring a standardization/definition exercise for each application server or application server data increment. This is contrary to our rapid AS development goal in IMS.

In conclusion, the compression/decompression algorithms stored by HSS do not influence the nature of data sent over the Sh interface. Different compression/decompression and formats of storage mechanisms are HSS implementation specific and out of scope of this WID.

## 5.6 Information Content vs. representation

To facilitate centralized storage, the information required to create basic telephony services must be conveyed to the HSS. To facilitate session processing, it must be conveyed to the requesting application server. This information is stored co resident with session processing in the active application server. This co-resident version is considered a cache. The master copy is viewed as the version stored in the HSS.

One application server goal is to allow higher subscriber numbers/capacity per application server. This goal requires architecture of this data to be extremely compact, regardless of the size of data transmitted from/to the centralized storage facility in the HSS, this form can be retained after registration. Increased numbers of subscribers that are concurrently cached at an application server drives capacity.

The HSS is impacted with increased size of this transferred data. Available memory in an HSS is allocated to each subscriber. Higher memory use per subscriber inversely affects the HSS capacity. For this reason, the HSS will consider compression of the stored data if that is practical.

With transparent data, the HSS is not aware of the content of the data. If the HSS were aware of content, then it could use this knowledge to reduce the size of the stored image. Since it is not, only generic compression techniques are available for this use.

If extensibility mechanisms such as XML are used to store data in an HSS, the expected increase in transmitted then stored information between the HSS and the application server could be between 15 and 50 or even more. An example showing the XML expansion effect when it is used to represent communication forwarding busy follows. The XML shown is a representation of data related to communication forwarding busy.

```
<CallForwardingBusy Assigned="1" Activated="0" PerPuid="0">
<ForwardToType>FORWARD_TO_DN</ForwardToType>
<ForwardToDNAllowed>0</ForwardToDNAllowed>
<ForwardToDN>+19742400002</ForwardToDN>
<EditPermissions>EDIT_FULL</EditPermissions>
</CallForwardingBusy>
```

Notice that the information contained within the 246 characters (not counting 10 spaces) could be contained within

<b>Digits</b>	(12)
+19742400002	
<b>Binary flags</b>	(3)
Call forwardingBusy	
Assigned="1"	
Activated="0"	
PerPuid="0"	
<b>Enumerations</b>	(2)
ForwardToType	
FORWARD_TO_DN	
EditPermissions>	
EDIT_FULL	
<b>Total Bits</b>	
(12*4)+3+(2*4) = 59 bits	<b>(BINARY)</b>
vs	
246 x 8 = 1968	<b>(XML)</b>
<b>XML Expansion Ratio</b> = 1968 / 59	~33.

**Figure 5.6-1: Expansion ratio example**

This expansion ratio value may significantly vary from one example to another and according to the data encoding choice, but it does not change the overall conclusion on the XML expansion effect. However, if trying to add flexibility to the binary encoding (see e.g. the example in 3GPP TS 24.604 [4] clause 4.9.1.2) additional information such as length and type indicators is required which will also affect the ratio. These considerations (size and bandwidth vs. flexibility) need to be weighted in order to select the format.

The XML advantages that make it largely used, i.e. ease of readability, a generic standard, do have cost aspects that XML induces regarding memory size and bandwidth which drive to consider a more compact form of IMS service data description.

Compression is discussed in sub-chapter 5.7. Flexibility aspects are covered in section 7 dealing with extensions.

## 5.7 Evaluation of compression options

### 5.7.1 Introduction

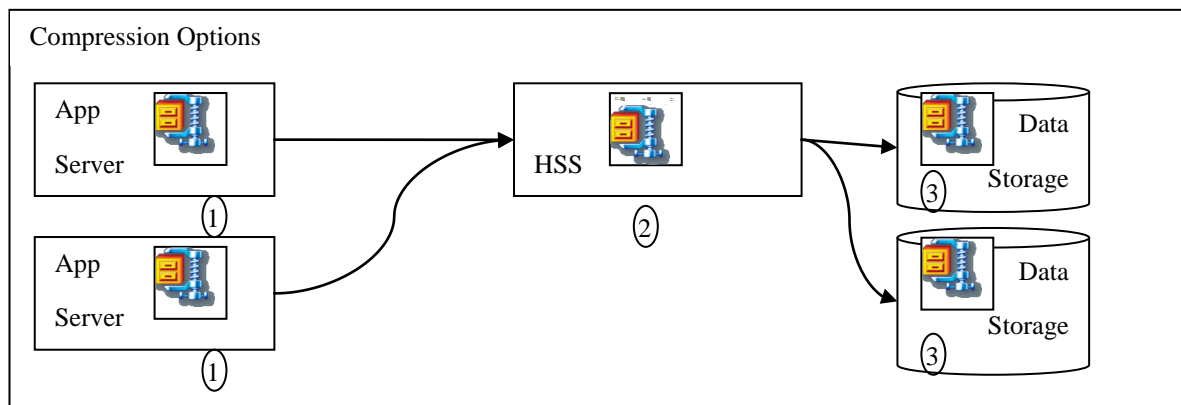
In case size is considered a problem, options for compression algorithms that would operate on the service data and their possible location options are described.

### 5.7.2 Architecture aspects

Figure 5.7.2-1 indicates three alternative locations for compression execution.

One option for compression is within the HSS complex, closest to the data storage. This would allow varying compression algorithms to be used in the network, and would leave the Transparent Data compression task and algorithm to be specified at each database. This is shown as option 3.

Another option (option 2) would be to identify a compression algorithm that would operate on Data received at the HSS. The HSS could have knowledge of the data being received (XML, encoded binary, .wav, text) and could choose a compression algorithm based upon the type of data, or instance of data being transmitted. This could even be determined and pre-configured per ServiceIndication. This begins to share specific contents of the HSS transparent Data in order to drive network efficiencies. As the ultimate progression of this HSS compression, the HSS may require specific knowledge of the XML contents so that it may efficiently store the received data. Potentially, this could require development on the HSS, or additions to HSS capabilities to allow a service to load and store a schema so that compression was available.



**Figure 5.7.2-1: Compression Algorithms**

Finally (option 1) we consider a mechanism wherein the Application servers agree upon a compression algorithm. This would allow any flexibility for XML extension, and may allow human readable visibility to the transmitted information, albeit prior to transmission. It appears as a data expansion to XML, followed by an almost immediate compression.

Notice that for increased network efficiency, compression should be performed closer to the left side of the figure. Compression schemes applied closer to the left also require more standardization or coordination to ensure the compression algorithm is similarly applied across all ASs.

### 5.7.3 Compression algorithm discussion

A binary encoded format, storing information using defined structures, is a form of compression that is extremely efficient. As indicated in clause 5.6 this efficiency will vary depending on the degree of flexibility required.

XML documents are flexible in specifying elements that can be combined freely, such as rules for the CDIV services. Normally, the XML elements have quite long names in order to be man-readable. If the size is considered a problem, compression may be an option. There are many examples of compression methods for text based messages. The simplest example is the compact form of headers as defined for SIP. Slightly more advanced is to build a dictionary of common strings, which has been defined for SIP and SDP in IETF RFC 3485 [25]. Using gzip is a third widely used method.

Compression algorithms for XML may negate one of its alleged values that is being man-readable. It will also be required to specify a common compression algorithm for AS interoperability.

XML compressors use different degrees of knowledge of the XML structure. Some XML compressors make use of the fact that they are operating on an XML file, some compressors also make use of a prior knowledge of the XML schema, and some compressors make no prior assumptions at all. Needless to say, more prior information generally results in more efficient algorithms.

When it comes to standards, some efforts have been done to standardize XML specific compression algorithms. ITU-T has specified the 'Fast Infoset' binary encoding in ITU-T Rec. X.891 [26].

W3C are in the late stages of developing a specification for Efficient XML Interchange (EXI) [27]. The current status of that work is that a complete working draft is out for review, and they plan to have a candidate specification in December 2008.

It seems that there are two options for compression. Either use the emerging W3C standard, or use the widely used gzip format. The latter is not as efficient, but as the service data are not that large in size, it will provide a reasonably good compression.

## 5.8 Application Server Service Data Descriptions (ASSDD) Reference point.

Binary data is not permitted within the Sh interface. When binary assumptions are made to define service data, this must be converted. Other Sh binary transfers have used the base64 encoding scheme. For this case, a service data definition is established as shown in Figure 5.8-1. This is at a point prior to base 64 encoding the binary data into an ASCII, IETF RFC 20 [12] format and including or extracting it from the Sh-Data>RepositoryData> ServiceData XML tags within the User-Data A VP. The defined format is not exposed on the Sh interface. Instead it is different by the deterministically recoverable base64 encoding scheme. See IETF RFC 2045 [13], IETF RFC 3548 [14].

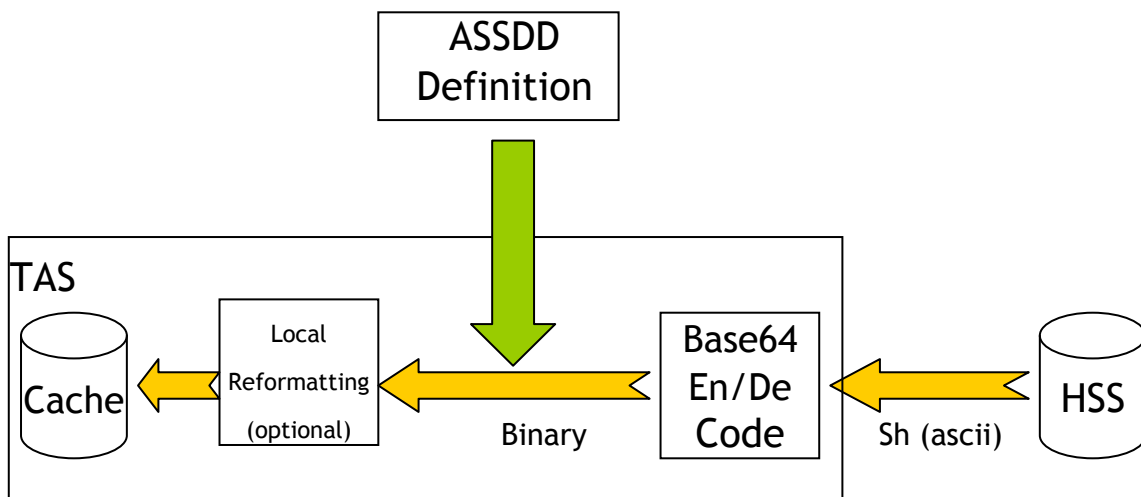


Figure 5.8-1: Definition point for TAS



---

## 6 Telephony Application Server service data definition

### 6.1 Defined Data

#### 6.1.0 Introduction

Data to support Telephony application server features is identified herein.

Conveying this data from an application server to the HSS requires definition of the details for each of the features and is implemented within the application server.

#### 6.1.1 Explicit Communication Transfer (ECT)

This feature is defined in 3GPP TS 24.629 [9], section 4.2.1 and in 3GPP TS 22.173 [3].

- Service Authorization – identifies if the subscriber has this service
- Consult Transfer - identifies if the transferred to party wants to be connected to the transferring party prior to the transfer
- Blind Transfer - identifies if the subscriber is allowed blind transfers.

#### 6.1.2 Communication Diversion (CDIV)

This feature is defined in 3GPP TS 24.604 [4] and has applicability to the following:

- CFU
- CFB
- CFNR
- CFNRc
- CFNL
- CD
- CDVN.

##### 6.1.2.1 CF Busy

The following variable length information is required to support this service:

- Forward-to-Destination – A SIP URI or TEL-URL that represents the available destination

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

The following are from 3GPP TS 24.604 [4], Table 4.3.1.1:

- Served user receives notification that a communication has been forwarded (indication of communication diversion to the diverting user) [notify-served-user]
- Originating user receives notification that his communication has been diverted (forwarded or deflected) [notify-caller]

- Served user receives reminder indication on outgoing communication that CDIV is currently activated. [notify-served-user-on-outbound-call].

The following enumeration fields are required to support this service:

- Served user allows the presentation of diverted to URI to originating user in diversion notification. [reveal-identity-to-caller]
- Served user allows the presentation of his/her URI to diverted-to user. [reveal-identity-to-target]
- Served user allows the presentation of his/her URI to originating user in diversion notification. [reveal-served-user-identity-to-caller].

Each enumeration field may contain any of:

- No
- Not Reveal as GRUU
- Yes (default).

### 6.1.2.2 CF No Reply /No Response

The following variable length information is required to support this service:

- Forward-to-Destination – A SIP URI or TEL-URL that represents the available destination

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

The following are from 3GPP TS24.604 [4], Tables 4.3.1.1 and 4.3.1.2:

- Served user receives notification that a communication has been forwarded (indication of communication diversion to the diverting user)
- Originating user receives notification that his communication has been diverted (forwarded or deflected)
- Served user receives reminder indication on outgoing communication that CDIV is currently activated.

The following enumeration fields are required to support this service:

- Served user allows the presentation of diverted to URI to originating user in diversion notification
- Served user allows the presentation of his/her URI to diverted-to user
- Served user allows the presentation of his/her URI to originating user in diversion notification.

Each enumeration field may contain any of:

- No
- Not Reveal as GRUU
- Yes (default).

### 6.1.2.3 CF Not Logged In (CFNL)

The following variable length information is required to support this service:

- Forward-to-Destination – A SIP URI or TEL-URL that represents the available destination.

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service.
- Service Activation – Identifies if the subscriber has activated this service.

The following are from 3GPP TS 24.604 [4], Tables 4.3.1.1 and 4.3.1.2:

- Originating user receives notification that his communication has been diverted (forwarded or deflected)
- Served user receives reminder notification on outgoing communication that forwarding is currently activated.

The following enumeration fields are required to support this service:

- Served user allows the presentation of diverted to URI to originating user in diversion notification
- Served user allows the presentation of his/her URI to diverted-to user
- Served user allows the presentation of his/her URI to originating user in diversion notification.

Each enumeration field may contain any of:

- No
- Not Reveal as GRUU
- Yes (default).

#### 6.1.2.4 CF All/CFU

The following variable length information is required to support this service:

- Forward to Destination – A SIP URI or TEL-URL that represents the available destination

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service.
- Service Activation – Identifies if the subscriber has activated this service.

The following are from 3GPP TS 24.604 [4], Tables 4.3.1.1 and 4.3.1.2:

- Served user receives notification that a communication has been forwarded (indication of communication diversion to the diverting user)
- Originating user receives notification that his communication has been diverted (forwarded or deflected)
- Served user receives reminder indication on outgoing communication that CDIV is currently activated
- Served user receives notifications (CDIVN) of their communication diversions.

The following enumeration fields are required to support this service:

- Served user allows the presentation of diverted to URI to originating user in diversion notification
- Served user allows the presentation of his/her URI to diverted-to user
- Served user allows the presentation of his/her URI to originating user in diversion notification.

Each enumeration field may contain any of:

- No
- Not Reveal as GRUU
- Yes (default).

### 6.1.2.5 CF Not Reachable (CFNRc)

The following variable length information is required to support this service:

- Forward to Destination – A SIP URI or TEL-URL that represents the available destination.

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

The following are from 3GPP TS 24.604 [4], Tables 4.3.1.1 and 4.3.1.2:

- Served user receives notification that a communication has been forwarded (indication of communication diversion to the diverting user)
- Originating user receives notification that his communication has been diverted (forwarded or deflected)
- Served user receives reminder notification on outgoing communication that forwarding is currently activated.

The following enumeration fields are required to support this service:

- Served user allows the presentation of diverted to URI to originating user in diversion notification
- Served user allows the presentation of his/her URI to diverted-to user
- Served user allows the presentation of his/her URI to originating user in diversion notification.

Each enumeration field may contain any of:

- No
- Not Reveal as GRUU
- Yes (default)

### 6.1.2.6 Communication Deflection

The following variable length information is required to support this service:

- Forward to Destination – A SIP URI or TEL-URL that represents the available destination

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

The following are from 3GPP TS 24.604 [4], Tables 4.3.1.1 and 4.3.1.2:

- Originating user receives notification that his communication has been diverted (forwarded or deflected).

The following enumeration fields are required to support this service:

- Served user allows the presentation of diverted to URI to originating user in diversion notification
- Served user allows the presentation of his/her URI to diverted-to user
- Served user allows the presentation of his/her URI to originating user in diversion notification.

Each enumeration field may contain any of:

- No
- Not Reveal as GRUU

- Yes (default).

### 6.1.2.7 Communication Diversion Notification

The following True/False information is required to support this service:

- Served user receives reminder notification on outgoing communication that forwarding is currently activated
- Served user receives notifications (CDIVN) of their communication diversions.

### 6.1.3 Three-Party (3PTY)/ CONFerence (CONF)

This feature is defined in 3GPP TS 24.605 [5], section 4.2.1 and in 3GPP TS 22.173 [3].

3PTY communication setup is equivalent to a three party conference created on request of the served user from existing communication sessions, this is equivalent to three party conference creation in CONF, For this reason 3PTY can be seen as a special case of CONF and most of service interactions for CONF apply also to 3PTY.

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

NOTE: 3GPP TS 24.605 [5] does not mention subscription options such as maximum number of conference callers.

### 6.1.4 ID Presentation Services

There are four separate ID presentation services: OIP, OIR, TIP and TIR.

#### 6.1.4.1 OIP

This feature is defined in 3GPP TS 24.607 [7] and in 3GPP TS 24.173 [10].

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

#### 6.1.4.2 OIR

This feature is defined in 3GPP TS 24.607 [7] and in 3GPP TS 24.173 [10].

The following enumerations are required to support this service:

This enumeration identifies the variant of Originating Identification Restriction which is in place. Note that this applies to both Line and Name Presentation. This enumeration specifies the treatment/labelling of IDs in INVITES and responses to INVITES. Unsubscribed is equivalent to PERM\_PUBLIC.

Table 6.1.4.2-1: OIR options

OIR Value	Description
RESTRICT_PERM_PRIVATE_INFO	Restrict all private information from appearing in headers
RESTRICT_PERM_PRIVATE	Indicate to the network the intent to never present this calling ID (To be honoured apart from OVERRIDE features).
RESTRICT_PRIVATE_INFO	The default state is to not present (restrict) the private information from appearing in a header, but the caller can change this per call with a dialled code.
RESTRICT_PRIVATE	The default state is to not present (restrict) the ID, but the caller can change this per call with a dialled code.
RESTRICT_PUBLIC	The default state is to present (not restrict) the ID, but the caller can change this per call with a dialled code.
RESTRICT_PERM_PUBLIC	Always" present this calling ID to the network (whether called party receives it of course depends on their subscription)

NOTE: Information options related to a user with an override capability of this feature are for further study.

### 6.1.4.3 TIP

This feature is defined in 3GPP TS 24.608 [8] and in 3GPP TS 24.173 [10].

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

### 6.1.4.4 TIR

This feature is defined in 3GPP TS 24.608 [8] and in 3GPP TS 24.173 [10].

The following enumerations are required to support this service:

Table 6.1.4.2-1: TIR options

TIR Value	Description
RESTRICT_PERM_PRIVATE	Indicate to the network the intent to never present this terminating ID (To be honoured apart from OVERRIDE features).
RESTRICT_PRIVATE	The default state is to not present (restrict) the ID, but the called party can change this per call with a dialled code.
RESTRICT_PUBLIC	The default state is to present (not restrict) the ID, but the called party can change this per call with a dialled code.

NOTE: Information options related to a user with an override capability of this feature are for further study

## 6.1.5 Communication Barring

This feature is defined in 3GPP 24.611 [16] and in 3GPP TS 22.173 [3].

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

Barring service for originations contains predetermined sets of user denied addresses, including denial of all calls, international calls, long distance calls, inter/intra lata calls, closed user group calls, and a variety of nuisance number combinations. Since origination barring is predefined, and applied the same for many subscribers, there is low likelihood of these enumerated services being common with the Ut interface barring defined in 3GPP 24.611 [16]. Therefore the intent is to enumerate the most used barring conditions, and allow either the enumeration or the Ut defined interface, as appropriate.

NOTE: The repeated filtering that occurs on the Ut interface, and the document size that would be required should this be statically stored for each user. Both can be avoided with this enumeration.

### 6.1.6 Call blocking

This feature is defined in 3GPP TS 24.611 [16] and in 3GPP TS 22.173 [3].

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

### 6.1.7 Malicious Communication IDentification (MCID)

This feature is defined in 3GPP TS 22.173 [3] and in 3GPP TS 24.616 [15]

The following True/False information is required to support the 2 variants of this service:

MCID:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

MCID\_Temp:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

### 6.1.8 Anonymous Communication Rejection (ACR)

This feature is defined in 3GPP TS 22.173 [3] and in 3GPP TS 24.611 [16].

The Anonymous Communication Rejection (ACR) service allows the served user to reject incoming communications on which the asserted public user identity of the originating user is restricted. In case the asserted public user identity of the originating user is not provided then the communication shall be allowed by the ACR service.

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

### 6.1.9 Communications Waiting (CW)

This feature is defined in 3GPP TS 22.173 [3] and in 3GPP TS 24.615 [18].

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

### 6.1.10 Communication HOLD (HOLD)

This feature is defined in 3GPP TS 22.173 [3] and in 3GPP TS 24.610[19].

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service.

### 6.1.11 Message Waiting Indication (MWI)

This feature is defined in 3GPP TS 22.173 [3] and in 3GPP TS 24.606 [6].

The following True/False information is required to support this service:

- Service Authorization – identifies if the subscriber has this service. Authorization implies activation.

### 6.1.12 Advice Of Charge (AOC)

This feature is defined in 3GPP TS 22.173 [3] and in 3GPP TS 24.647 [17].

The following True/False information is required to support each of the AOC-S, AOC-D and AOC-E services:

- Service Authorization – identifies if the subscriber has this service
- Service Activation – Identifies if the subscriber has activated this service.

### 6.1.13 Reverse charging

This feature is defined in 3GPP TS 22.173 [3] which heavily references ITU I.256.3 [22].

The following enumerations are required to implement either origination or termination party reverse charging services.

ITU I.256.3 [22] defined cases include: Each are potential capabilities for any user.

- a) [...a calling user, on a per-call basis, to request reverse charging at call set-up time...]
- b) [...calling user, during the active phase of the call, to request reverse charging...]
- c) [...called user, before disconnection of the call, to request reverse charging for the entire call...]
- d) [...called user to be charged for all incoming calls...].

### 6.1.14 Closed User Group (CUG)

This feature is defined in 3GPP TS 22.173 [3] and in 3GPP TS 24.654[24].

This feature is considered low penetration. It contains data should not be contained within a Transparent data block with other mainstream services. Instead, it should be considered as a Services 3 when viewing Figure 7.2.3-1.

### 6.1.15 Flexible Alerting (FA)

This feature is defined in 3GPP TS 24.239 [21].

This feature is considered low penetration. It contains data should not be contained within a Transparent data block with other mainstream services. Instead, it should be considered as a Services 3 when viewing Figure 7.2.3-1.

### 6.1.16 Completion of Communications to Busy Subscriber (CCBS)/Completion of Communications on No Reply (CCNR)

This service is defined in 3GPP TS 24.642 [23].

The following True/False information is required to support this service:

- Service Authorization for CCBS/CCNR – identifies if the subscriber has this service.



---

## 7 Reuse and extension

### 7.1 Introduction

Extensibility allows applications to function during changes in the surrounding architectures. Backward compatibility deals with the requirements that must be in place prior to extending data for services. This section addresses both.

Application servers must allow data extensions to coexist with an operational definition of the data. Restricting the definition of services is not viable, so it must be possible to add TAS data. Allowances for data extension and backward compatibility are discussed further in section 7.2.

### 7.2 Extensibility of IMS Application Server Service Data Descriptions

#### 7.2.1 Introduction

There are two ways of expanding AS service descriptions based on association of service indications and the transparent data blobs. They are as follows:

Service indication per MMTel group/suite of services, and Service indications per service within MMTel group/suite of services. Additionally, base services get extended over time. Extending a service and expanding the number of services can be approached differently. The following sections discuss initial expansion (definition) of services and within each the extension of the service is discussed.

#### 7.2.2 Service indication per MMTel group/suite of service

Even established telephony services must change over time. For example, law enforcement act, emergency services, and priority service have changed even the basic telephony services. This extensibility section deals with the definition of the data into blobs, and the options for extending services.

For simpler storage and retrieval, service features may be combined into a group or groups. Larger groups should be expected as a result of initial maturity of many service definitions. Since the data is communicated to the HSS via transparent data, these should be grouped into Transparent Data blobs, and identified by Service Indication name. The blob defined by a distinct service indication is herein referred to as a database.

The service data descriptions contained in one database may be a complete service suite definition, or can be viewed as a portion of a service definition. The remaining portion of a service definition may be comprised of elements specific to a particular architecture (e.g. WLAN, CDMA etc.), or a subsequent addition to standardised service definitions, or proprietary extensions. The principle of extensibility should be applicable to all afore-mentioned cases.

An example figure 7.2.3-1 below considers the database contents for a distinct service. Each of the darker services blocks (databases) represent an HSS transparent data blob with a distinct service indication. Multiple databases may be applied to represent a service suite on an application server.

On the left, the representation depicts the grouping of mature TAS services. It represents that additional services be located in an extension database, 'services2'. Note that Services2 does not repeat data from the Services1 and cannot serve a subscriber on its own, but instead extends the TAS services in a group.

On the right, notice that many of the services defined within the Services2 structure were also extended from the Services1 structure. With this association, if any Services1 data remains required, there will be dead data also conveyed. Instead, we should create data so that established features are collected in one database and outlier features, new features and low penetration features are separated. In this way, when/if data is made obsolete, it is more likely isolated in structures that aren't required by all users. In summary, data associated with a base functionality should be extended, rather than duplicately redefined.

Additionally, extensions to the basis of Services1 (e.g. Services4) if generated in a distinct database, should also be extended by reference. This could be within a structure like Services2 or within a new extension similar to Services5.

All ASs capable of operating in a network will semantically understand the agreed upon '3GPP Services' data. Other ASs may recognize the extensions if appropriate.

### 7.2.3 Extensibility within defined datasets.

Databases may be expanded to keep similar features intact. This is possible without predefining spare fields. Rules for the extensions are dependent upon the agreed upon schemas. Backward compatibility rules must also be built into the schemas.

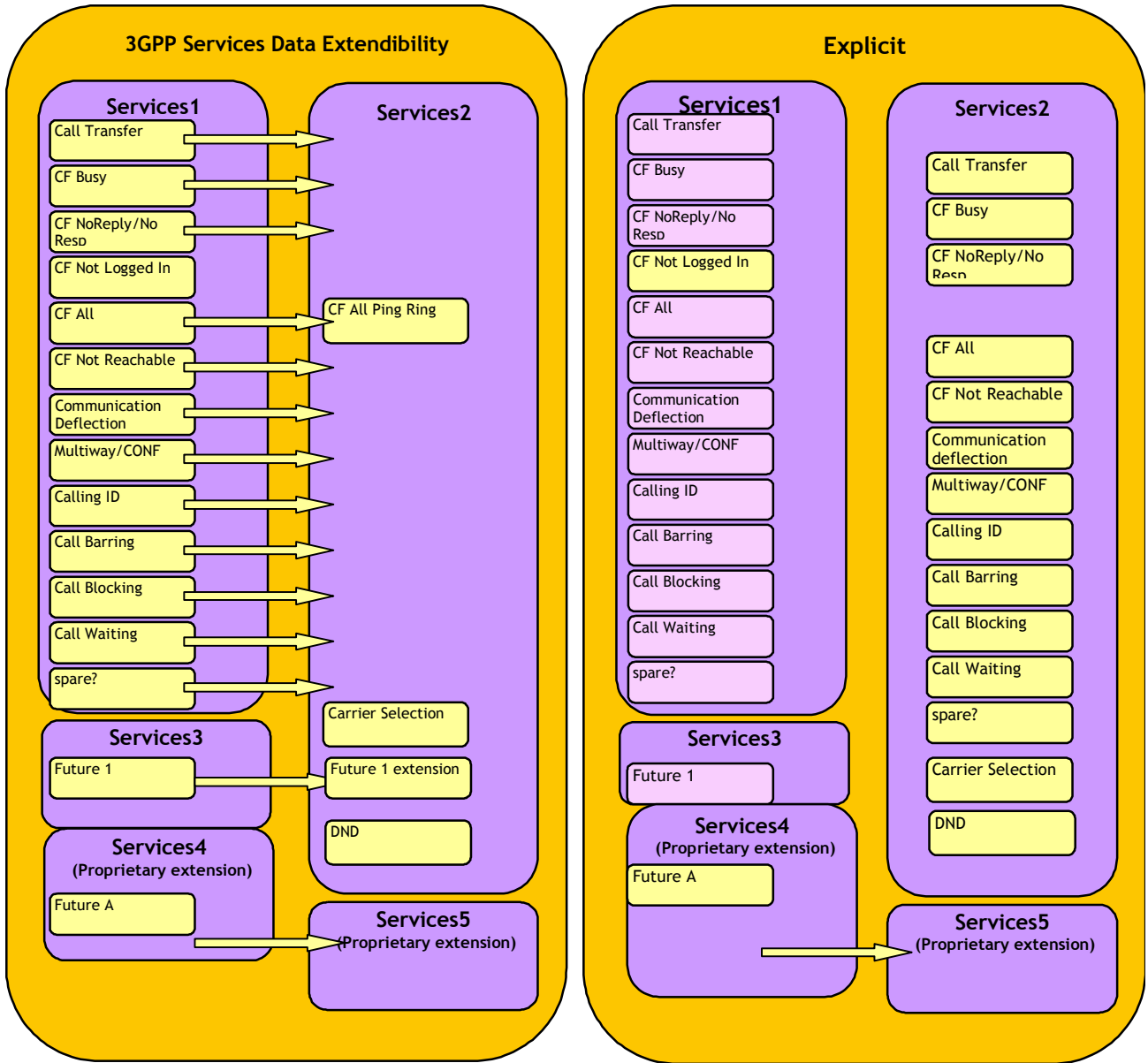
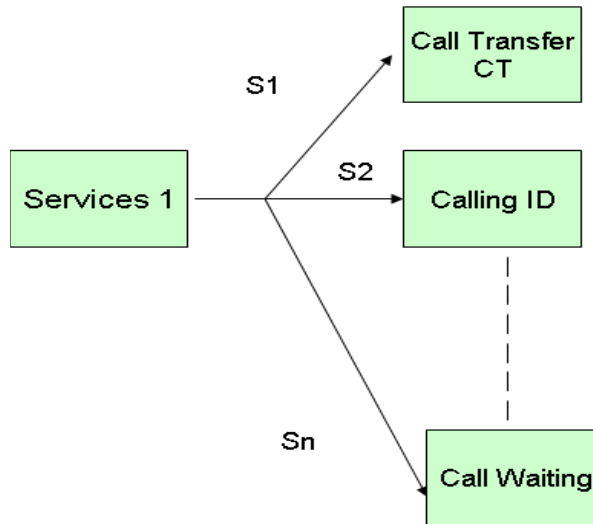


Figure 7.2.3-1: Analysis of Extension databases: Inference and explicit

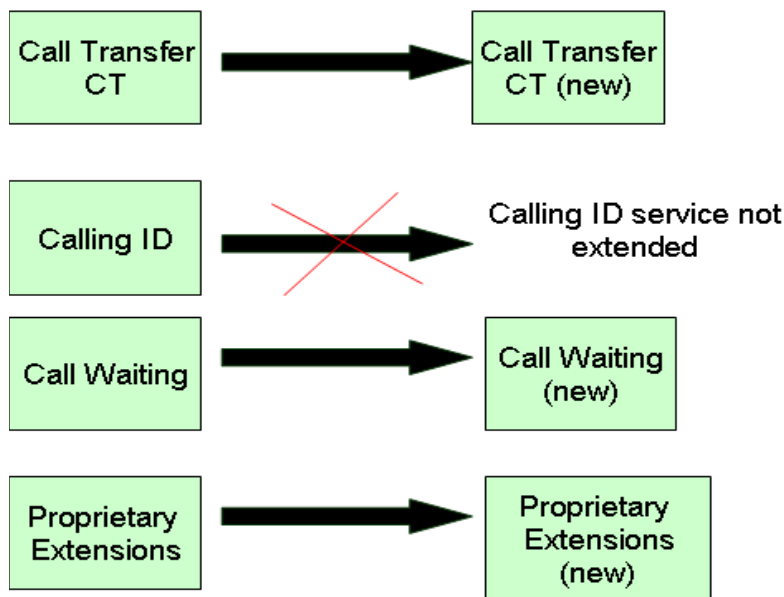
### 7.2.4 Service indications per service within MMTel group/suite of services

Another alternative (as shown in figure 7.2.4-1) is based upon assigning a service indication for each service within the MMTel group/suite of services. In the figure 7.2.4-1, S1, S2...Sn are service indications pointing to a specific service.



**Figure 7.2.4-1: Assigning Service Indications to each service within MMTel suite of services**

The number of service indications defined shall be determined by the number of logical service groupings. Services can be determined as unique and have their own Service-Indication; however it is still possible for common service data between two services. For example Call Forwarding All may have a different Service-Indication than Call Forwarding Not Reachable though the service data is similar. A further optimization can be achieved by using the same Service-Indication for similar style of services names/tags for example, Call forwarding Busy and Call Forwarding No Answer.



**Figure 7.2.4-2: Extensions to services within MMTel suite of services**

The figure 7.2.4-2 shows extensions to existing one of the services like Call transfer (CT). This extension may happen by reference wherever there is an existing instance of the service data i.e enhancement of an existing service. It is understood that AS will be aware of the service indications relevant to the services it supports. If the AS wants to read status of all services within the MMTel suite/group, it can do so over the Sh interface using Notif-Eff feature (See 3GPP TS 29.329 [11]).

As shown in figure, the Call Transfer CT (new) service data shall contain both the old and the new data. While this may mean that minor obsolete aspects of service data get carried over, the benefits of not forcing multiple queries to the HSS to get all the data and allowing for data to be changed, not just added are crucial systemic advantages.

The alternative described above allows greater flexibility for the operator who may want to use different service subsets of MMTel suite/group of services. In a distributed services environment, it is efficient for an AS to deal with its specific service data rather than parsing through contents of the entire MMTel blob. For example, if an AS responsible for Call forwarding services wants to change the call forward to number on behalf of an user, it shall not be forced to update the

entire MMTel blob. Additionally an operator who may want to extend one or two services should not be forced to define a whole new blob just to enable a new definition for the specific services.

## 7.2.5 Recommendations

A hybrid approach of combining Service indications to form subset of services may be possible. This optimization may be based on frequency of use, similarity of service data etc. For example the Call forwarding Busy and Call forwarding No Answer and Call Forwarding All services might be combined, since the frequency of updates is often common. Further transparent data containing Return Last Call information might have it's own Service Indication, since it may be rewritten at a different frequency.

## 7.2.6 Syntactic and Semantic recognition.

When detected in any application server, databases that are syntactically parsed, but not completely semantically understood by that application should follow strict extensibility rules. These must preserve unrecognized data for application servers that may later receive this data and understand it. This allows newer applications to extend data versions when new data is required, and requires the previous version of application server to permit the additions. Syntactically all ASs must be capable of identifying unrecognized data. Further, upon updating data in the HSS, an AS must also convey the unrecognized data integrated with any recognized data changes.

## 7.2.7 Extensibility for information efficient data definitions.

In section 7.2.3, we note that a database (associated with a specific ServiceIndication) can be expanded to keep related data together. This may be necessary when adding to a feature. For example, (referring to Figure 7.2.3-1) the "Call Forwarding All" addition of the ping ring feature, could be accomplished within the database (e.g. ServiceIndication=Services 1) that initially defined CFA. If this is done, the addition/definition of ServiceIndication=Services 2 would not be necessary.

The following is a set of rules that are necessary to facilitate extensions. These rely on the definitions in Section 3.1 and they apply for extensions within a ServiceIndication:

1. Additions to a database MUST occur either in
  - Additional releases, or
  - Re-use of existing spare fields within a defined release.
2. Each additional release MUST include an advertised release name and size
3. All applications must retain all release information. When unknown releases exist, and when writing known releases an AS must sequentially restore unknown releases with no changes.
4. New data can be in new release segments or new databases (Service Indication).
5. Additions to a release MUST not allow deletion of previously defined data.
6. Additions to a database must be 4-byte aligned.
7. Add on feature definitions should add additional releases to the same database to keep like data in the same Service Indication.
8. Proprietary/Significant unrelated functionality should identify new ServiceIndications.

Refer to Figure 7.2.7-1 and Figure 7.2.7-2 for distinctions on adding data in the same database (ServiceIndication) and adding additional databases.

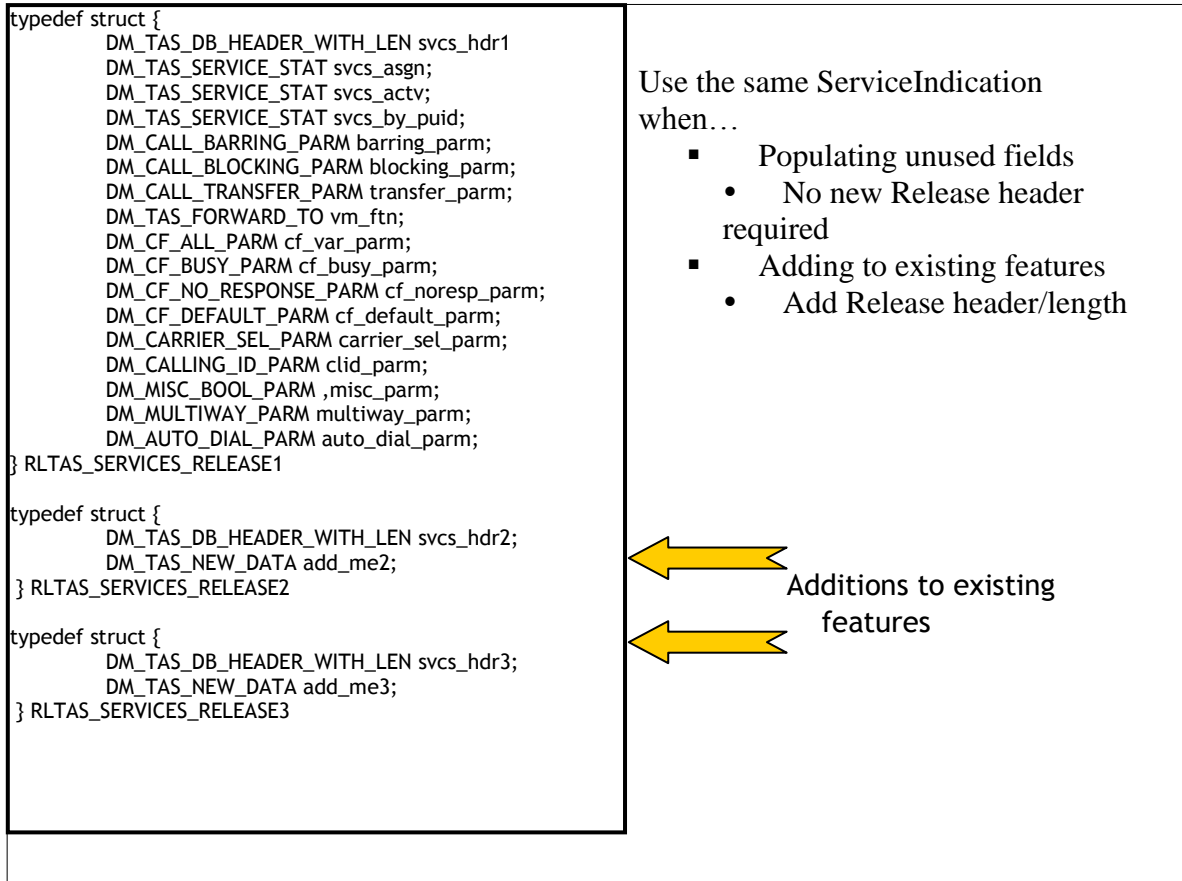


Figure 7.2.7-1: Adding to a Release

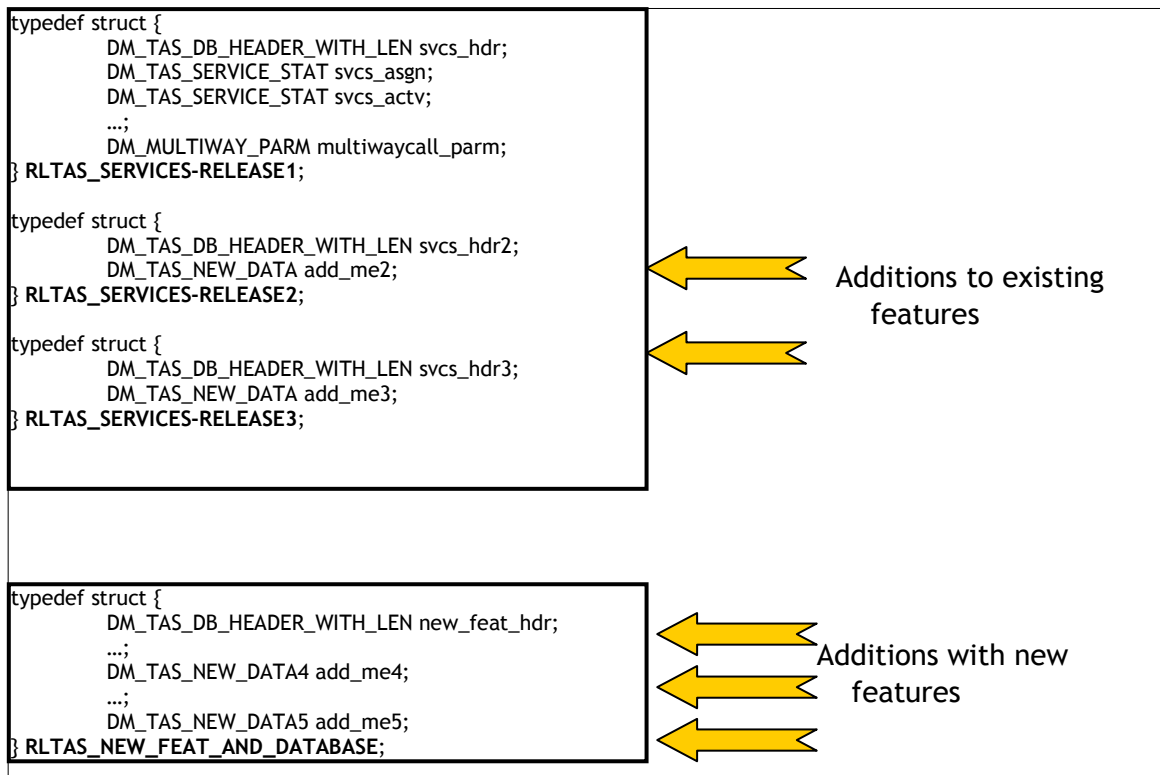


Figure 7.2.7-2: Adding new functionality.

A new database (ServiceIndication) should be considered when significant unrelated feature functionality is added to an application server. This is especially true if low subscriber penetration is expected for the new data, because this allows the additional data size to be attributed to only the subscribers requiring the data.

An alternate design that would not define a new database is to create an optional release within the existing database. The optional property is also available with a new release within an existing database.

The cohesion of the data along with its eventual growth needs (maximum possible size) may be the main criteria for selecting a new database versus extending the existing database with an optional release.

Additions to an existing database should be considered when the additional data is related to or used concurrently with data defined within the existing database (ServiceIndication). This allows the data to be stored within the same 'blob' and can ensure that related data is recalled concurrently. Database read/write exercises will atomically succeed or fail for each ServiceIndication, whereas multiple service indications may require retries in some error scenarios.

A release is not only a unit of new data, but is also a unit of optioning. For subsequent releases it should be possible to add multiple iterations of the same release into the database. This might be used for list management and growth. An example of this use and its efficiencies applies in the screening list example. If the "release" defines one or more numbers to screen against, but this subscriber has 3x the average number of screening entries, then the release that defines screening members may be repeated several times to accommodate the longer list within the same database.

## 7.2.8 Backward Compatibility for information efficient data definitions.

### 7.2.8.1 Introduction

All future revisions must contain a mechanism for backward compatibility. The following list contains the basic guidelines to be included. Refer to section 4.1 for terminology used in this section.

The data as presented have assumed a set of nested structures, as a means of encapsulating related data, but this feature set has so few attributes in many cases that we will have very sparse structures.

Since we don't go to extraordinary lengths to pack data (probably breaking local cohesion, as related data might be spread out based on its size), we will have bits and possibly bytes which are not used in the initial definition. The extensibility section implies that these bits may be used later, without going into detail about how. Given that releases (layouts) do not have versions, the only way this works is if:

- All bits in the fixed section of release data shipped to a TAS MUST be preserved by the TAS when rewritten. (Note: this doesn't require the TAS to store them; the database could be re-retrieved when an update is required, and the new data written over the old while preserving unchanged bits.)
- On creation, all those unused bits MUST be set to a known consistent value, and extensions which start to use the bits MUST accept that initial value as indicating disabled or inactive or whatever the appropriate default for the new use is. (Alternately, interpretation of those bytes can be controlled by some other data, without which those bits and bytes are ignored. There is a similar problem with the control data, but they could conceivably be in a new release.)

The inactive value MUST be 0, for bits and full bytes both. Proprietary extensions MUST NOT use bits (reserved or unused) in non-proprietary releases (segments).

### 7.2.8.2 Releases in databases

Releases in a database consist of a predefined set of data, defined at the time of a standard baseline version. An application server must process the first release. All subsequent release contents are optionally understood.

- a. Optional releases MUST NOT become mandatory.
- b. Mandatory releases MUST NOT become optional. The first release MUST be mandatory.
- c. Additional mandatory releases MUST NOT be added to an existing database.
- d. Additional optional releases may be added to an existing database.
- e. Existing mandatory releases MUST NOT be removed from existing databases.

- f. Application servers MUST ensure that any release content that is not understood when read is included intact if rewriting the database to the CSDB.
- g. The range of any release data for an existing sub-data component MUST NOT be reduced.
- h. The meaning of any defined database content value MUST NOT be changed, though reserved and unused contents may be defined.
- i. The contents of reserved and unused entries in a database MUST be preserved when rewriting the database.
- j. The minimum length of the data portion of a database is 4 bytes.
- k. Releases are required to be 4 byte aligned.
- l. The size of a database shall be a multiple of 4 bytes. This may be redundant, since releases are aligned to 4 byte boundaries. Base64 encoding has rules for determining the fill bits if these do not align on increments of 6 bits.
- m. Enumeration value ranges MAY be extended within the defined fixed space limitations.
- n. The order of subsequent release contents MAY be variable.
- o. The length of each database may be variable, although subsequent definition versions may reuse unused or reserved fields.
- p. The services bitmaps DM\_TAS\_SERVICE\_STAT svcs\_asgn, svcs\_actv, and svcs\_by\_puid MAY identify all features for a subscription. They do not need to be repeated in subsequent databases or in other releases of RLTAS\_SERVICES.
- q. Release structures may occur 0, 1 or more times. This allows variable number of (potentially) variable length byte definitions to exist.
- r. Releases are defined with a Network Byte order (Big Endian) wherein the Most Significant byte of any multibyte data field is stored at the lowest memory address, which is also the address of the larger field.

### 7.2.8.3 New releases

- a. New optional releases may be added to existing databases after a recommendation has been published; however, network entities that do not recognize these new releases MAY ignore the functionality represented. Ignoring the functionality does not imply that the data may be ignored when writing back to the CSDB. Information in a release MUST BE rewritten intact when replacing data. Reference Section 7.2.8.1 item f.
- b. New fields may be added to, or spare fields may be used in existing releases; however, network entities that do not recognize these new fields may ignore the feature functionality represented, but MUST retain these fields when re-writing the database.

---

## 8 Alternative solutions

### 8.1 Introduction

This section presents the possible solutions to store subscriber data supporting an IMS Telephony Application server and the changes required in the specifications to implement them.

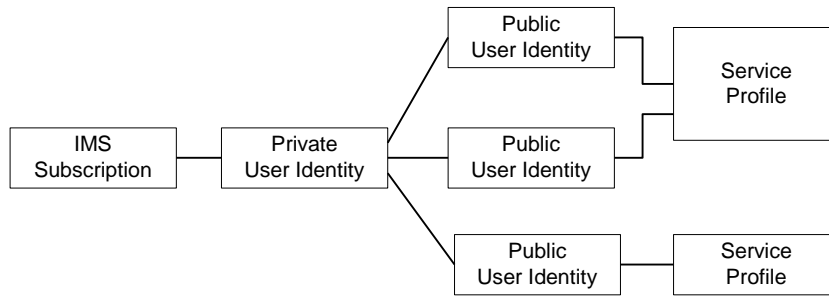
### 8.2 A Binary Storage Approach to Defining IMS Service Data

#### 8.2.0 Service Data Description

The definition of TAS service data contained in this section is one realizable mechanism of defining supplementary service data for a Telephony application server. It contains a space efficient, therefore transmission bandwidth efficient mechanism for defining data associated with Telephony supplementary services in an IMS environment.

With a telephony application server, some data pertains to the handset, whereas other data applies to each address tied to the handset.

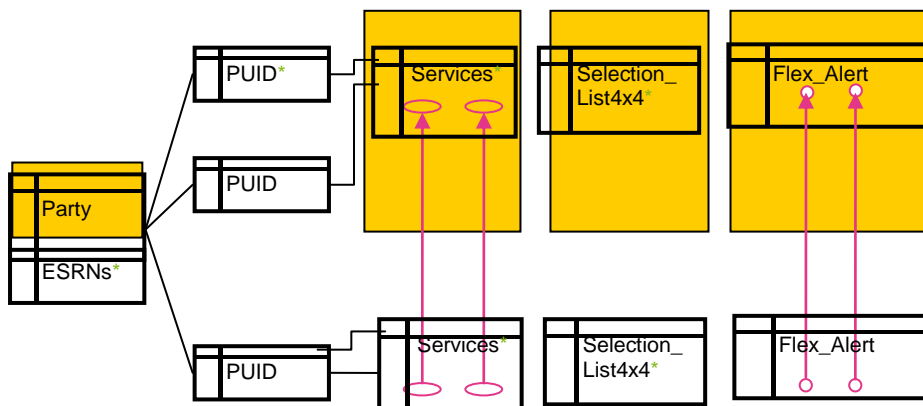
The 3GPP data model for IMS comes in 2 variants. This exercise selected the simpler of the 2 formats. This model is shown as (3GPP 23.228 [20] section 4.3.3.4). This is repeated in Figure 8.2.0-1.



**Figure 8.2.0-1: 3GPP Data Model Structure**

The AS may have requirements to keep data scoped to each of the boxes identified in the above figure. This assumes that each PUID may have independent features. The TAS structure to model is shown in Figure 8.2.0-2. Implementations have shown that this can be achieved with databases identified in boxes in this figure.

In the definition that follows, only the shaded portion of this figure may apply when the PUIDs share the same services. Alternately, different services or may apply for each PUID within a subscription, and in those instances, the additional services, (or selection list or simultaneous\_ring, etc.) databases may apply. Subscribers with more complex features and data additions, such as Selection features (screening, etc.) and Simultaneous Ringing (Flexible Alerting) require separate databases. Since the Sh interface provides a query for PUIDs within the subscription, the information defining the related PUIDs can be obtained via Sh query, and not maintained on the TAS. The Party database only contains data which applies across all PUIDs. Extensions to the defined list of databases are addressed in section 7.2.



**Figure 8.2.0-2: Association of Services across PUIDs**

The information listed in this section contains detailed data fields required to support IMS Telephony services defined with in 3GPP.

The data supporting the Services database is listed in tables complete with use-descriptions, representing the documentation of the necessary data. This is immediately followed within each section by the C style data definitions.



This clarifies the binary representation of this structure that will be base64 encoded into the Sh interface within the User-Data A VP.

Reserved indicates that this value is currently used for masked capabilities. Unused indicates the data assignment may be reassigned to new functionalities.

## 8.2.1 Services Data Definition

The database name for this Service definition, as requested from an HSS ServiceIndication is "SERVICES RELEASE1"

This is the main structure for services data. This is essentially per-subscriber data, but for aliases multiple PUIDs can point to a single entry. Data captured here is intended to reflect the data for features described in section 6, but is currently absent of data supporting some services. These services will be completely defined.

**Table 8.2.1-1: Service Data Main Structure**

Name	Description/Details
svcs_hdr	<p><b>Description</b></p> <p>Carries the release identifier for this structure (0001) and the overall length of this block (fixed + variable, including this header, padded to four bytes).</p> <p><b>Attribute Description</b></p> <p>DM_TAS_DB_HEADER_WITH_LEN</p>
svcs_asgn	<p><b>Description</b></p> <p>This carries the boolean indications per service whether it is assigned for this subscription or not (DBYES=YES...). Indexed by DM_TAS_SERVICE_ID.</p> <p><b>Attribute Description</b></p> <p>DM_TAS_SERVICE_STAT</p>
svcs_actv	<p><b>Description</b></p> <p>This carries the boolean indications per service whether it is currently activated for this subscription or not (DBYES=YES...). Indexed by DM_TAS_SERVICE_ID. Should be YES only if already assigned. (Note that it's a big bit field, not an array of dmubytes.)</p> <p><b>Attribute Description</b></p> <p>DM_TAS_SERVICE_STAT</p>
svcs_by_puid	<p><b>Description</b></p> <p>This carries the boolean indications per service whether the service data should come from the per-PUID definition or from the primary PUID are stored here. Initially, the per-PUID data are stored in the FEATIDX belonging to the Default Originating PUID, a.k.a. the Primary PUID. So that tuple will have all bits set to DBNO, as there's nowhere else to go. Another observation is that very few features are permitted to be per-PUID. Provisioning must enforce that. One can determine if a PUID has services specific only to the PUID by checking that its features are different than that of the "party" common feature set.</p> <p>This is the set of features that applies to all PUIDs in the subscription. If different features are detected, then those service parameters whose svcs_by_puid bits are set in the local PUID's SERVICES database apply specific to that PUID. The remainder of the SERVICES tuple consists of the common party features. Note that potentially the Party has features assigned which the PUID reserves but doesn't have assigned. Again indexed by DM_TAS_SERVICE_ID. (Note that it's a big bit field, not an array of DMUBYTEs.)</p>

	<p><b>Attribute Description</b></p> <p>DM_TAS_SERVICE_STAT</p>
Barring_parm	<p><b>Description</b></p> <p>Parameters for TAS_CALL_BARRING</p> <p><b>Attribute Description</b></p> <p>DM_CALL_BARRING_PARM</p>
blocking_parm	<p><b>Description</b></p> <p>Customization parameters for TAS_CALL_BLOCKING</p> <p><b>Attribute Description</b></p> <p>DM_CALL_BLOCKING_PARM</p>
transfer_parm	<p><b>Description</b></p> <p>TAS_CALL_TRANSFER</p> <p><b>Attribute Description</b></p> <p>DM_CALL_TRANSFER_PARM</p>
vm_forward_to	<p><b>Description</b></p> <p>Number to use when forwarding to voice mail. Normally also the number the user enters to retrieve voice mail. Stored here because it's used by several services.</p> <p><b>Attribute Description</b></p> <p>DM_TAS_FORWARD_TO</p>
cf_var_parm	<p><b>Description</b></p> <p>Parameters for TAS_CF_ALL (CFV)</p> <p><b>Attribute Description</b></p> <p>DM_CF_ALL_PARM</p>
cf_busy_parm	<p><b>Description</b></p> <p>Parameters for TAS_CF_BUSY (CFBL)</p> <p><b>Attribute Description</b></p> <p>DM_CF_BUSY_PARM</p>
cf_noresp_parm	<p><b>Description</b></p> <p>Parameters for TAS_CF_NO_RESPONSE (CFNR/CFDA)</p> <p><b>Attribute Description</b></p> <p>DM_CF_NO_RESPONSE_PARM</p>
carrier_sel_parm	<p><b>Description</b></p> <p>Parameters for TAS_CARRIER_SEL</p> <p><b>Attribute Description</b></p>

	DM_CARRIER_SEL_PARM
clid_parm	<p><b>Description</b></p> <p>Parameters for TAS_CALLING_LINE_ID_PRESENTATION</p> <p><b>Attribute Description</b></p> <p>DM_CALLING_ID_PARM</p> <p><b>Rules</b></p> <p>Default (unassigned) is edit_clir == DBNO edit_clir is only allowed in cases where per-call privacy override is also allowed. Calling ID edit_clir cannot be YES if clir is a Permanent type</p>
multiwaycall_parm	<p><b>Description</b></p> <p>Parameters for TAS_MULTIWAY_CALLING</p> <p><b>Attribute Description</b></p> <p>DM_MULTIWAY_PARM</p>

```
typedef struct {
    DM_TAS_DB_HEADER_WITH_LEN svcs_hdr;
    DM_TAS_SERVICE_STAT svcs_asgn;
    DM_TAS_SERVICE_STAT svcs_actv;
    DM_TAS_SERVICE_STAT svcs_by_puid;
    DM_CALL_BARRING_PARM barring_parm;
    DM_CALL_BLOCKING_PARM blocking_parm;
    DM_CALL_TRANSFER_PARM transfer_parm;
    DM_TAS_FORWARD_TO vm_forward_to;
    DM_CF_ALL_PARM cf_var_parm;
    DM_CF_BUSY_PARM cf_busy_parm;
    DM_CF_NO_RESPONSE_PARM cf_noresp_parm;
    DM_CARRIER_SEL_PARM
    DM_CALLING_ID_PARM clid_parm;
    DM_MULTIWAY_PARM multiwaycall_parm;
} SERVICES_RELEASE1;
```

As an example of a separate database, we could use the CUG example :

```
typedef struct {
    DM_TAS_DB_HEADER_WITH_LEN svcs_hdr;
    DM_TAS_SERVICE_STAT svcs_asgn;
    DM_TAS_SERVICE_STAT svcs_actv;
    DM_TAS_SERVICE_STAT svcs_by_puid;
    DM_CUG_TYPE1 cug_data1;
    DM_CUG_TYPE2 cug_data2;
} CUG_RELEASE1;
```

## 8.2.2 DMTAS\_FORWARD\_TO

This is a further definition of the above named structure.

Table 8.2.2-1: DMTAS\_FORWARD\_TO

Name	Description/Details
DM_TAS_FORWARD_TO	<p><b>Description</b></p> <p>This is the Forward-to numeric string, which will be processed by Digit Analysis and E.164 conversion. Might be in E.164 format already, but it could also be a sequence of digits, maybe even with an access code (account code, authorization code, carrier selection...). Not a URI - no prefix, no domain; it's not even an implied URI, unlike DMPUID_USER. This is a request URI, and so may have parameters and access codes which are not directly part of the destination address.</p> <p>The size is somewhat arbitrary, but should be large enough to handle the various codes besides a directory number which might appear. The size is not actually large enough to hold all possible combinations.</p> <p>SEE ALSO: DMTAS_DN, when no invocation or access codes are permitted (e.g. ESRNs) and it's got to be a number.</p> <p>The DM_TAS_FORWARD_TO should be composed of at most 32 bytes as variable length data.</p> <p><b>Attribute Description</b></p> <p>alphanumeric DMTAS_URIVARBYTE</p>

```
typedef DMTAS_URI DMTAS_FORWARD_TO;
```

### 8.2.3 DMTAS\_DN

This is a further definition of the above named structure.

Table 8.2.3-1: DMTAS\_DN

Name	Description/Details
DMTAS_DN	<p><b>Description</b></p> <p>An ASCII DN string; primarily a digit-string received from another entity, generally E.164. Note that this is not large enough to hold all user-dialed strings, which can include dialcodes and other prefixes. This is a real string, i.e. null-terminated. 16 characters are allocated to allow room for the '+' character used in SIP to indicate that a number is fully-resolved, i.e. E.164. E.164 numbers are at most 15 digits.</p> <p><b>Attribute Description</b></p> <p>type="VARCHAR" size="16" ioType="AlphaNumeric" presentation="Raw DMTAS_URIVARBYTE</p>

```
typedef char DMTAS_DN[16 + 1];
#define SZTAS_DN 16
```

### 8.2.4 DMULONG

```
typedef unsigned long DMULONG;
```

## 8.2.5 DMUSHORT

Typedef unsigned short DMUSHORT;

## 8.2.6 DMUBYTE

This is a further definition of the above named structure.

**Table 8.2.6-1: DMUBYTE**

Name	Description/Details
DMUBYTE	<p><b>Description</b></p> <p>An Unsigned Byte type, so that it's clear this is not a single character (e.g. 'a') but a numeric element. No type called byte is defined in the C/C++ Language, but an unsigned 8 bit char is defined. We use Unsigned Char to define a UByte.</p> <p><b>Attribute Description</b></p> <p>Type=UTINYINT</p>

typedef unsigned char DMUBYTE;

## 8.2.7 DMTAS\_VARBYTE

This is a further definition of the above named structure. It identifies the structural contents of a variable length Byte definition.

**Table 8.2.7-1: DMTAS\_VARBYTE**

Name	Description/Details
vdata_offset release	<p><b>Description</b></p> <p>This field is the starting byte count relative to the start of the release.</p> <p><b>Attribute Description</b></p> <p>unsigned short</p>
vdata_len rls_size	<p><b>Description</b></p> <p>The byte count of the variable length contents.</p> <p><b>Attribute Description</b></p> <p>unsigned short</p>

```
typedef struct { {
    unsigned short vdata_offset; /* byte offset from release start */
    unsigned short vdata_len; /* length in octets */
} DMTAS_VARBYTE;
```

## 8.2.8 DMTAS\_VARSTRING

This is a further definition of the above named structure. It is a VARBYTE subtype used to hold string data. Nominally these data are of type UTF-8, although the length is still specified in bytes not characters. Being a string, Null characters are not legal. Initially, the character set is restricted to printable ASCII, plus white space.

**Table 8.2.8-1: DMTAS\_VARSTRING**

Name	Description/Details
DMTAS_VARSTRING	<p><b>Description</b></p> <p>This field is the starting byte count relative to the start of the release.</p> <p><b>Attribute Description</b></p> <p>DMTAS_VARBYTE</p>

```
typedef DMTAS_VARBYTE    DMTAS_VARSTRING;
```

## 8.2.9 DMTAS\_URI

The following table contains further definition of the structure named in this heading.

**Table 8.2.9-1: DMTAS\_URI**

Name	Description/Details
DMTAS_URI	<p><b>Description</b></p> <p>A VARSTRING subtype used to hold URIs. Primarily sip: or tel:. The URI type is restricted to the general URI layout and character set. Semantically it must also be an acceptable request; requests may be strings of digits which are meaningful to a TAS but do not map to a destination (e.g. service access codes).</p> <p><b>Attribute Description</b></p> <p>DMTAS_VARSTRING</p>

```
typedef DMTAS_VARSTRING    DMTAS_URI;
```

## 8.2.10 HEADER WITH LENGTH

This is a further definition of the DM\_TAS\_DB\_HEADER\_WITH\_LEN structure. It identifies the structural contents of a release in a database.

**Table 8.2.10-1: HEADER WITH LENGTH**

Name	Description/Details
release	<p><b>Description</b></p> <p>This field is unique for a data definition. It identifies which data definition should be Applied in order to interpret the release</p> <p><b>Attribute Description</b></p>

	unsigned short
rls_size	<p><b>Description</b></p> <p>The byte count of the release contents length, including variable length contents that the release may contain.</p> <p><b>Attribute Description</b></p> <p>unsigned short</p>

```
typedef struct {
    unsigned short  release; /* Version ID */
    unsigned short  rls_size; /* length in octets */
} DM_TAS_DB_HEADER_WITH_LEN;
```

## 8.2.11 TAS\_SERVICE\_STAT

**Table 8.2.11-1: TAS\_SERVICE\_STAT**

Name	Description/Details
DM_TAS_SERVICE_STAT	<p><b>Description</b></p> <p>This array exists in order to encapsulate the size of the service list for reuse elsewhere. This should be accessed as a set of boolean bits, indexed by DM_TAS_SERVICE_ID.</p> <p><b>Attribute Description</b></p> <p>Ubyte, size = 8</p>

```
Typedef dmubyte DM_TAS_SERVICE_STAT[8];
#define SZ_TAS_SERVICE_STAT      8
```

## 8.2.12 DM\_TAS\_CALLING\_ID\_RESTRICT\_ENUM

This is a further definition of the DM DM\_TAS\_CALLING\_ID\_RESTRICT\_ENUM data. It describes the variant of Calling ID Restriction which is in place. Note that most of these apply to both Calling Line and Calling Name Presentation/Restriction (CLIP/CLIR, CNIP/CNIR). This enumeration specifies the treatment/labelling of IDs in INVITEs and responses to INVITEs. Unsubscribed is equivalent to PERM\_PUBLIC.

**Table 8.2.12-1: DM\_TAS\_CALLING\_ID\_RESTRICT\_ENUM**

Enumeration	Description/Details
DM_ID_RESTRICT_PERM_PRIVATE	<p><b>Description</b></p> <p>Permanent Private.</p> <p>"Never" present this calling ID (apart from OVERRIDE features).</p> <p><b>Attribute Description</b></p> <p>Enumeration</p>

DM_ID_RESTRICT_PRIVATE	<p><b>Description</b></p> <p>The default state is to not present the ID, but the caller can change this per call with a dialled code.</p> <p><b>Attribute Description</b></p> <p>Enumeration</p>
DM_ID_RESTRICT_PUBLIC	<p><b>Description</b></p> <p>The default state is to present the ID, but the caller can change this per call with a dialled code.</p> <p><b>Attribute Description</b></p> <p>Enumeration</p>
DM_ID_RESTRICT_PERM_PUBLI	<p><b>Description</b></p> <p>Always" present this calling ID to the network (whether called party receives it of course depends on their subscription).</p> <p><b>Attribute Description</b></p> <p>Enumeration</p>

```
typedef enum {
    DM_ID_RESTRICT_PERM_PRIVATE,
    DM_ID_RESTRICT_PRIVATE,
    DM_ID_RESTRICT_PUBLIC,
    DM_ID_RESTRICT_PERM_PUBLIC
} DM_TAS_CALLING_ID_RESTRICT_ENUM;
#define BF_TAS_CALLING_ID_RESTRICT_ENUM 8
```

### 8.2.13 DM\_TAS\_CF\_EDITING

Enumeration of permissions for user-originated changes of Call Forwarding data

**Table 8.2.13-1: DM\_TAS\_CF\_EDITING**

Enumeration	Description/Details
DM_CF_EDIT_FULL	<p><b>Description</b></p> <p>User has full permission, i.e. can activate and deactivate the service, and can change the forward-to destination.</p> <p><b>Attribute Description</b></p> <p>Enumeration</p>
DM_CF_EDIT_PARTIAL	<p><b>Description</b></p> <p>User has only partial permission, i.e. can only activate and deactivate the service. This means that other fields cannot be changed by the user, such as forward-to destination, bearer-based options, or changes to the timeout value. (Not all services have these attributes, some may have other attributes, and some services may have attributes they don't use.)</p>



	<b>Attribute Description</b> Enumeration
DM_CF_EDIT_NONE	<b>Description</b> User has no permission, so can neither activate/deactivate the service, nor change the forward-to destination.  <b>Attribute Description</b> Enumeration

```
typedef enum {
    DM_CF_EDIT_FULL,
    DM_CF_EDIT_PARTIAL,
    DM_CF_EDIT_NONE
} DM_TAS_CF_EDITING;
#define BF_TAS_CF_EDITING 2
```

## 8.2.14 DM\_TAS\_CF\_MODE

Enumeration of possible forward-to options for call-forwarding services.

Note that this implementation means that the forwarding options are mutually exclusive (a single call can only be forwarded to one place), and activations are memoryless (one can't activate one mode, activate a second, deactivate the second, and expect the first mode to be used).

**Table 8.2.14-1: DM\_TAS\_CF\_MODE**

Enumeration	Description/Details
DM_CF_TO_DN	<b>Description</b> Forwarding should use the PUID specified by this specific service (normally user-programmable).  <b>Attribute Description:</b> Enumeration
DM_CF_TO_VM	<b>Description</b> Forwarding should go to the VoiceMail Service number provisioned for this subscriber.  <b>Attribute Description</b> Enumeration
Reserved	<b>Description</b> Attribute Description Enumeration
DM_CF_TO_SPARE1	<b>Description</b> Attribute Description Enumeration

```
typedef enum {
    DM_CF_TO_DN,
    DM_CF_TO_VM,
    Reserved
    DM_CF_TO_SPARE1
} DM_TAS_CF_MODE;
#define BF_TAS_CF_MODE 2
```

## 8.2.15 DM\_TAS\_SERVICE\_ID

Enumerates the services supplied by the organic Telephony Application Server (TAS). Many services have feature parameters in SERVICES, in addition to their basic assigned/active status.

The enumerations are numbered, so that the bitmaps in data (DM\_TAS\_SERVICE\_STAT) won't shift when new members are added or deleted. They should stay in numeric order! Note the END entry (at the end of the list); this can be used both as a loop boundary and in size declarations.

Empty descriptions represent proposed but not committed/defined/implemented services. (Unused services, and holes in the numbering, are to some extent left around to provide spares for field use, as well as bidding time until the unused service is actually implemented.) Reserved values represent ranges that should not be reused.

If the number is listed as unused, it truly is, if the number is not listed, it's also unused. The size covers all possible values, although most uses are limited to the size implied by TAS\_END\_OF\_SERVICES.

```
typedef enum {
    VM_FTN = 1,
    TAS_CALLING_LINE_ID_PRESENTATION = 2,
    TAS_CALL_BARRING = 3,
    TAS_CALL_BLOCKING = 4,
    TAS_CALL_TRANSFER = 5,
    TAS_CALL_WAITING = 6,
    TAS_CF_ALL = 7,
    TAS_CF_BUSY = 8,
    TAS_CF_NO_RESPONSE = 9,
    TAS_DO_NOT_DISTURB = 10,
    TAS_MULTIWAY_CALLING = 11,
    TAS_CF_NOT_LOGGED_IN = 12,
    TAS_CF_NOT_REACHABLE = 13,
    TAS_COMMUNICATION_DEFLECTION = 14,
    TAS_CF_DIVERSION_NOTIFICATION = 15,
    TAS_ORIGINATION_ID_RESTRICTION = 16,
    TAS_TERMINATION_ID_RESTRICTION = 17,
    TAS_TERMINATION_ID_RESTRICTION = 18,
    TAS_MALICIOUS_COMMUNICATION_IDENTIFICATION = 19,
    TAS_MALICIOUS_COMMUNICATION_IDENTIFICATION_TEMP = 20
    TAS_COMMUNICATIONS_HOLD = 21,
    TAS_MESSAGE_WAITING_INDICATION = 22,
    TAS_REVERSE_CHARGING = 23,
    TAS_CLOSED_USER_GROUP = 24,
    TAS_FLEXIBLE_ALERT = 25,
    TAS_CCBS/CCNR = 26,
    TAS_AOC-S = 27,
    TAS_AOC-D = 28,
    TAS_AOC-A = 29,
    TAS_END_OF_SERVICES = 30,
} DM_TAS_SERVICE_ID;
#define BF_TAS_SERVICE_ID 30
```

## 8.2.16 DM\_CALL\_CATEGORY

It provides most of the variants of call classification determined by Digit Analysis, plus a few other categories. Most variants are not mutually exclusive, so the representation is bitfields rather than an enum. When the comment refers to "designated" numbers, it means digit strings entered in the dialling plan as aliases for that service; so service providers can choose different numbers for logical services like emergency and directory assistance as needed to accommodate differing country conventions. This particular structure provides category data for several different services, which is why it isn't a PARM all by itself. The most obvious service is Call Barring, which can bar many categories of call. Account codes are another using service, as the codes can be marked as required for certain call categories.

**Table 8.2.16-1: DM\_CALL\_CATEGORY**

Name	Description/Details
call_cat_all	<p><b>Description</b></p> <p>All outgoing calls EXCEPT EMERGENCY; really mutually exclusive with all other options, due to its scope.</p> <p><b>Attribute Description:</b> DMBOOL</p>
call_cat_local	<p><b>Description</b></p> <p>All outgoing local calls.</p> <p><b>Attribute Description:</b> DMBOOL</p>
call_cat_intra_lata	<p><b>Description</b></p> <p>All outgoing intra-LATA toll calls. Generalized Name: Local Toll.</p> <p><b>Attribute Description:</b> DMBOOL</p>
call_cat_inter_lata	<p><b>Description</b></p> <p>All outgoing inter-LATA toll calls. Generalized Name: Long Distance.</p> <p><b>Attribute Description:</b> DMBOOL</p>
call_cat_intl	<p><b>Description</b></p> <p>All outgoing International calls... (calls using country code/INTL dialling).</p> <p><b>Attribute Description:</b> DMBOOL</p>
Reserved1Unused1b	<p><b>Description</b></p> <p>Reserved for alternate use.</p> <p><b>Attribute Description:</b> DMBOOL</p>

```
typedef struct {
    DMBOOL call_cat_all : BFBOOL;
    DMBOOL call_cat_local : BFBOOL;
    DMBOOL call_cat_intra_lata : BFBOOL;
    DMBOOL call_cat_inter_lata : BFBOOL;
    DMBOOL call_cat_intl : BFBOOL;
    DMBOOL call_cat_UnusedbReserved1 : BFBOOL;
    DMBOOL call_cat_UnusedbReserved2 : BFBOOL;
    DMBOOL call_cat_UnusedbReserved3 : BFBOOL;
    DMBOOL call_cat_UnusedbReserved4 : BFBOOL;
    DMBOOL call_cat_UnusedbReserved5 : BFBOOL;
    DMBOOL call_cat_UnusedbReserved6 : BFBOOL;
}
```

```

DMBOOL call_cat_UsedbReserved7: BFBOOL;
DMBOOL call_cat_UsedbReserved8: BFBOOL;
DMBOOL call_cat_UsedbReserved9: BFBOOL;
DMBOOL call_cat_UsedbReserved10: BFBOOL;
DMBOOL call_cat_UsedbReserved11: BFBOOL;
DMBOOL call_cat_UsedbReserved12: BFBOOL;
DMBOOL call_cat_Used13b: BFBOOL;
DMBOOL call_cat_Used14b: BFBOOL;
DMBOOL call_cat_Used15b: BFBOOL;
DMBOOL call_cat_Used16b: BFBOOL;
DMBOOL call_cat_Used17b: BFBOOL;
DMBOOL call_cat_Used18b: BFBOOL;
DMBOOL call_cat_Used19b: BFBOOL;
DMUBYTE call_cat_Used1;
} DM_CALL_CATEGORY;

```

## 8.2.17 DM\_CALL\_BARRING\_PARM

Provides the variants of call barring (denial of outgoing calls); most variants are not mutually exclusive, so the representation is in fields rather than an enum. When the comment refers to "designated" numbers, it means digit strings entered in the dialling plan as aliases for that service; so service providers can choose different numbers for logical services like emergency and directory assistance as needed to accommodate differing country conventions.

**Table 8.2.17-1: DM\_CALL\_BARRING\_PARM**

Name	Description/Details
categories	<p><b>Description</b></p> <p>A set bit implies that barring is active for that category of call. Only the first 16 categories are valid for the call barring service ("all" through the end).</p> <p><b>Attribute Description</b></p> <p>DM_CALL_CATEGORY</p>
user_ctrl	<p><b>Description</b></p> <p>A set bit implies that the end user (and only the end user) can set and clear the active status of the corresponding category (categories.xxx). Only the first 16 categories are valid for the call barring service ("all" through the end). These bits may only be set if the subscriber's feature bundle includes the TAS_USER_CTRL_CALL_BARRING service.</p> <p><b>Attribute Description</b></p> <p>DM_CALL_CATEGORY</p>

```

typedef struct {
    DM_CALL_CATEGORY categories;
    DM_CALL_CATEGORY user_ctrl;
} DM_CALL_BARRING_PARM;

```

## 8.2.18 DM\_CALL\_BLOCKING\_PARM

Call blocking is a terminating feature. It performs call blocking (rejection) for incoming calls. This parameter identifies categories of call blocking.

Table 8.2.18-1: DM\_CALL\_BLOCKING\_PARM

Name	Description/Details
block_all	<p><b>Description</b></p> <p>Block all incoming calls.</p> <p><b>Attribute Description:</b> DMBOOL</p>
block_intl	<p><b>Description</b></p> <p>Block all incoming calls from extra-national endpoints (e.g. other country codes).</p> <p><b>Attribute Description:</b> DMBOOL</p>
User_ctl_all	<p><b>Description</b></p> <p>DBYES means the user has the ability to change the block-all setting. User changes can be performed either via web portal or by dial code. Dial code attempts are accompanied by a PIN.</p> <p><b>Attribute Description:</b> DMBOOL</p>
User_ctl_intl	<p><b>Description</b></p> <p>DBYES means the user has the ability to change the block-intl setting. User changes can be performed either via web portal or by dial code. Dial code attempts are accompanied by a PIN.</p> <p><b>Attribute Description:</b> DMBOOL</p>
Unused	<p><b>Description</b></p> <p>Available for future use.</p> <p><b>Attribute Description:</b> DMBOOL</p>

```
typedef struct {
    DMBOOL block_all : BFBOOL;
    DMBOOL block_intl: BFBOOL;
    DMBOOL user_ctl_all : BFBOOL;
    DMBOOL user_ctl_intl : BFBOOL;
    DMBOOL Unused4b : BFBOOL;
    DMBOOL Unused3b: BFBOOL;
    DMBOOL Unused2b : BFBOOL;
    DMBOOL Unused1b : BFBOOL;
    DMUBYTE Unused2 : BFBOOL;
    DMUSHORT Unused2 : BFBOOL;
} DM_CALL_BLOCKING_PARM;
```

## 8.2.19 DM\_CALL\_TRANSFER\_PARM

This data defines variants of Call Transfer, at least one must be enabled for the service to exist (but Multi-way calling can be used to transfer too).

Table 8.2.19-1: DM\_CALL\_TRANSFER\_PARM

Name	Description/Details
------	---------------------

blind	<p><b>Description</b></p> <p>Network blind transfer w/dialled code; blind means no checking with the transfer-to party before severing the current connection. The UE may implement call transfer On its own, and if it does, this wouldn't be controlled or regulated at the TAS. This is left intact in case there is a flash implementation that would allow the TAS to control if blind transfer should be implemented</p> <p><b>Attribute Description:</b> DMBOOL</p>
consult	<p><b>Description</b></p> <p>Network transfer w/consultation (flashhook). The transfer-to party is connected to the transferring party for discussion. Flashhook toggles between the two calls. Because flashhook is used, this option is mutually exclusive with Multiway Calling. The UE may implement call transfer on it's own, and if it does, this wouldn't be controlled or regulated at the TAS. This is in case there is a flash implementation that would allow the TAS to control if consultative transfer should be implemented.</p> <p><b>Attribute Description:</b> DMBOOL</p>

```
typedef struct {
    DMBOOL Unused7b: BFBOOL;
    DMBOOL Unused6b: BFBOOL;
    DMBOOL blind : BFBOOL;
    DMBOOL consult : BFBOOL;
    DMBOOL Unused5b: BFBOOL;
    DMBOOL Unused4b: BFBOOL;
    DMBOOL Unused3b: BFBOOL;
    DMBOOL Unused2b: BFBOOL;
    DMBOOL Unused1b: BFBOOL;
    DMUSHORT Unused1: BFBOOL;
} DM_CALL_TRANSFER_PARM;
```

## 8.2.20 DM\_CALLING\_ID\_PARM

Options of Calling Identification services. This parameter covers both Calling Number (CLIP/CLIR/CLIRO) and Calling Name (CNIP/CNIR) services, and both originations and terminations (some interactions require knowledge of both the origination and termination properties).

Unsubscribed action will be equivalent to public origination no caller-ID on termination.

**Table 8.2.20-1: DM\_CALLING\_ID\_PARM**

Name	Description/Details
Clir	<p><b>Description</b></p> <p>Originating and Terminating Feature] Specifies the standard treatment for this subscription. Note that this applies only to an INVITE; responses to INVITES are covered by .colr below (future).</p> <p><b>Attribute Description</b></p> <p>DM_TAS_CALLING_ID_RESTRICT_ENUM</p>
Clip	<p><b>Description</b></p> <p>[Terminating Feature] Calling Line ID number presentation</p>

	<b>Attribute Description:</b> DMBOOL
Cnip	<b>Description</b> [Terminating Feature] Calling Line ID name presentation  <b>Attribute Description:</b> DMBOOL

```
typedef struct {
    DM_TAS_CALLING_ID_RESTRICT_ENUM clir : BF_TAS_CALLING_ID_RESTRICT_ENUM;
    DMBOOL clip : BFBOOL;
    DMBOOL Unused8b: BFBOOL;
    DMBOOL Unused7b: BFBOOL;
    DMBOOL Unused6b: BFBOOL;
    DMBOOL Unused6b: BFBOOL;
    DMBOOL Unused4b: BFBOOL;
    DMBOOL Unused3b: BFBOOL;
    DMBOOL Unused2b: BFBOOL;
    DMBOOL Unused1b: BFBOOL;
    DMUBYTE Unused1f: 7;
    DMUSHORT Unused1;
} DM_CALLING_ID_PARM;
```

## 8.2.21 DM\_CF\_ALL\_PARM

Customizing parameters for the Call Forward All (CFA) service (a.k.a. CF Variable – CFV or CF Unconditional CFU). This service ignores the state of the endpoint, e.g. Busy, Idle, unregistered. The data are similar to other call forwarding services, but the VoiceMail handling is different. In particular, CFA/CFV is only assigned as a standalone feature - assigning VoiceMail won't do it. Thus there's no need for a dn\_fwd\_ok attribute, the service assignment bit carries that meaning.

Note that this structure is also used by the Selective Call services which may need to do forwarding (SCF, SCA).

**Table 8.2.21-1: DM\_CF\_ALL\_PARM**

Name	Description/Details
vm_fwd_ok	<b>Description</b> Voice Mail assigned, i.e. this service is allowed to forward to voice mail. Updated by provisioning, based on the global status of voice-mail subscription. Might be set even if this service isn't assigned.  <b>Attribute Description:</b> DMBOOL
edit_perm	<b>Description</b> Permissions the user has to change these call-forward settings.  <b>Attribute Description:</b> DM_TAS_CF_EDITING  Default: DM_CF_EDIT_FULL
fwd_to_where	<b>Description</b> IF this service decides to forward a call, send it to the ID specified. So this field selects between the various (mutually exclusive) options for routing this call.  <b>Attribute Description:</b> DM_TAS_CF_MODE
reveal_uri_options	<b>Description</b>

	Identifies the URI options that are specified for the terminating user with reference to the information conveyed or toward the forwarding user and the originating user.  <b>Attribute Description:</b> REVEAL_URI_OPTIONS
forward_to	<b>Description</b>  ID to forward calls to; may be a partial dial string, might be a handle, could include carrier access codes, so it's not necessarily a PUID, but it can create one.  <b>Attribute Description:</b> DM_TAS_FORWARD_TO

```
typedef struct {
    DMBOOL Unused11b: BFBOOL;
    DMBOOL vm_fwd_ok : BFBOOL; /* redundant, derivable from voicemail_#_provisioned */
    DM_TAS_CF_EDITING edit_perm : BF_TAS_CF_EDITING;
    DMBOOL Unused10b: BFBOOL;
    DM_TAS_CF_MODE fwd_to_where : BF_TAS_CF_MODE;
    DMBOOL Unused9b: BFBOOL;
    DM_REVEAL_URI_OPTIONS reveal_uri_options; BF_REVEAL_URI_OPTIONS
    DMBOOL Unused7b: BFBOOL;
    DMBOOL Unused6b: BFBOOL;
    DMBOOL Unused5b: BFBOOL;
    DMBOOL Unused4b: BFBOOL;
    DMBOOL Unused3b: BFBOOL;
    DMBOOL Unused2b: BFBOOL;
    DMBOOL Unused1b: BFBOOL;
    DMTAS_FORWARD_TOVARBYTE forward_to;
} DM_CF_ALL_PARM;
```

## 8.2.22 DM\_REVEAL\_URI\_OPTIONS

This data defines the enumerations of permissions for user-originated changes of Call Forwarding data.

**Table 8.2.22-1: DM\_REVEAL\_URI\_OPTIONS**

Enumeration	Description/Details
DM_GRUU_FALSE	<b>Description</b>  URI should not be revealed.  <b>Attribute Description:</b> Enumeration
DM_NOT_REVEAL_GRUU	<b>Description</b>  If the "gr" parameter is present and the served user has the subscription option "Served user allows the presentation of his/her URI to diverted to user", then Not reveal processing must occur attributes they don't use.  <b>Attribute Description:</b> Enumeration
DM_GRUU_TRUE	<b>Description</b>  GRUU should be conveyed in full.  <b>Attribute Description:</b> Enumeration



```
typedef enum {
    DM_GRUU_FALSE,
    DM_GRUU_NOT_REVEAL_GRUU,
    DM_GRUU_TRUE
} REVEAL_URI_OPTIONS_
#define BF_REVEAL_URI_OPTIONS_2
```

### 8.2.23 DM\_TAS\_FORWARD\_TO\_PRESENTATION\_OPTIONS

```
typedef struct {
    DMBOOL        notify-caller: BFBOOL;
    REVEAL_URI_OPTIONS reveal-identity-to-caller: BF_REVEAL_URI_OPTIONS;
    REVEAL_URI_OPTIONS reveal-served-user-identity-to-caller: BF_REVEAL_URI_OPTIONS;
    DMBOOL        notify-served-user: BFBOOL;
    DMBOOL        notify-served-user-on-outbound-call: BFBOOL;
    REVEAL_URI_OPTIONS reveal-identity-to-target: BF_REVEAL_URI_OPTIONS;
} DMTAS_FORWARD_TO_PRESENTATION_OPTIONS;
```

### 8.2.24 DM\_CF\_BUSY\_PARM

Service parameters for Call Forward Busy (CFBL). Voice Mail number to forward to is globally stored. (Same elements as DM\_CF\_UNREG\_PARM...).

**Table 8.2.24-1: DM\_CF\_BUSY\_PARM**

Name	Description/Details
vm_fwd_ok	<p><b>Description</b></p> <p>Voice Mail assigned, i.e. this service is allowed to forward to voice mail. Updated by provisioning, based on the global status of voice-mail subscription.</p> <p><b>Attribute Description:</b> DMBOOL</p>
fwd_to_where	<p><b>Description</b></p> <p>If this service decides to forward a call, send it to the ID specified. So this field selects between the various (mutually exclusive) options for routing this call.</p> <p><b>Attribute Description:</b> DM_TAS_CF_MODE</p>
edit_perm	<p><b>Description</b></p> <p>Permissions the user has to change these call-forward settings.</p> <p><b>Attribute Description:</b> DM_TAS_CF_EDITING</p> <p>Default: DM_CF_EDIT_FULL</p>
forward_to	<p><b>Description</b></p> <p>ID to forward calls to; may be a partial dial string, might be a handle, could include carrier access codes, so it's not necessarily a PUID, but it can create one.</p> <p><b>Attribute Description:</b> DM_TAS_FORWARD_TO</p>

```
typedef struct {
    DMBOOL vm_fwd_ok : BFBOOL;
```

```

DMBOOL Unused11b: BFBOOL;
DMBOOL Unused10b: BFBOOL;
DM_TAS_CF_MODE fwd_to_where : BF_TAS_CF_MODE;
DMBOOL Unused9b: BFBOOL;
DMBOOL Unused8b: BFBOOL;
DM_TAS_CF_EDITING edit_perm : BF_TAS_CF_EDITING;
DMBOOL Unused7b: BFBOOL;
DMBOOL Unused6b: BFBOOL;
DMBOOL Unused5b: BFBOOL;
DMBOOL Unused4b: BFBOOL;
DMBOOL Unused3b: BFBOOL;
DMBOOL Unused2b: BFBOOL;
DMBOOL Unused1b: BFBOOL;
DMUSHORT Unused1;
DMTAS_FORWARD_TO forward_to;} DM_CF_BUSY_PARM;

```

## 8.2.25 DM\_CF\_NO\_RESPONSE\_PARM

Service parameters for Call Forward No Response/Answer (CFDA/CFNA). Because this feature may be assigned either for forward-to-number service or forward-not-answered-to-voice-mail, extra bits are required inside this parameter to indicate which services are actually allowed once this feature is assigned (i.e. call processing must screen activation attempts based on this internal data).

This service provides not only no-answer timing once alerting has started, but other non-busy error responses as well (all SIP non-success responses except 486, 480 and 600, which are captured by Call Forwarding Busy).

**Table 8.2.25-1: DM\_CF\_NO\_RESPONSE\_PARM**

Name	Description/Details
noresp_timer	<p><b>Description</b></p> <p>Time to wait before declaring the "No Response" condition. Units are (seconds, rather than ring cycles?)</p> <p><b>Attribute Description:</b> DMTAS_VARBYTE</p>
vm_fwd_ok	<p><b>Description</b></p> <p>Voice Mail assigned, i.e. this service is allowed to forward to voice mail. Updated by provisioning, based on the global status of voice-mail subscription.</p> <p><b>Attribute Description:</b> DMBOOL</p>
fwd_to_where	<p><b>Description</b></p> <p>If this service decides to forward a call, send it to the ID specified. So this field selects between the various (mutually exclusive) options for routing this call.</p> <p><b>Attribute Description:</b> DM_TAS_CF_MODE</p>
edit_perm	<p><b>Description</b></p> <p>Permissions the user has to change these call-forward settings.</p> <p><b>Attribute Description:</b> DM_TAS_CF_EDITING</p> <p>Default: DM_CF_EDIT_FULL</p>
forward_to	<p><b>Description</b></p> <p>ID to forward calls to; may be a partial dial string, might be a handle, could</p>

	include carrier access codes, so it's not necessarily a PUID, but it can create one.  <b>Attribute Description:</b> DM_TAS_FORWARD_TO
--	---

```
typedef struct {
    DMUBYTE noresp_timer;
    DMBOOL vm_fwd_ok : BFBOOL;
    DMBOOL Unused11b: BFBOOL;
    DMBOOL Unused10b: BFBOOL;
    DM_TAS_CF_MODE fwd_to_where : BF_TAS_CF_MODE;
    DMBOOL Unused9b: BFBOOL;
    DMBOOL Unused8b: BFBOOL;
    DM_TAS_CF_EDITING edit_perm : BF_TAS_CF_EDITING;
    DMBOOL Unused7b: BFBOOL;
    DMBOOL Unused6b: BFBOOL;
    DMBOOL Unused5b: BFBOOL;
    DMBOOL Unused4b: BFBOOL;
    DMBOOL Unused3b: BFBOOL;
    DMBOOL Unused2b: BFBOOL;
    DMBOOL Unused1b: BFBOOL;
    DMUBYTE Unused1;
    DMTAS_FORWARD_TO forward_to; } DM_CF_NO_RESPONSE_PARM;
```

## 8.2.26 DM\_MULTIWAY\_PARM

Service parameters for Multiway Calling. This service does not provide advanced conferencing abilities, such as meet-me, recordings and the like.

**Table 8.2.26-1: DM\_MULTIWAY\_PARM**

Name	Description/Details
n_way	<p><b>Description</b></p> <p>The number of members allowed to be joined as a single call. The subscriber (near-end) leg is included in this number.</p> <p><b>Attribute Description:</b> DMUBYTE</p> <p><b>Default Value:</b> 3</p>
do_transfer	<p><b>Description</b></p> <p>Applies primarily to 3-way calling -- specifies the treatment of other legs when the controller of the call hangs up. If DBYES, then the remaining parties stay connected ("transfer from 3-way call"); if DBNO, all connections are torn down.</p> <p><b>Attribute Description:</b> DMBOOL</p>

```
typedef struct {
    DMUBYTE n_way;
    DMBOOL do_transfer : BFBOOL;
    DMULONG Unused1f: 7;
    DMUSHORT Unused1; } DM_MULTIWAY_PARM;
```

## 8.3 XML based data description

### 8.3.1 General proposal

One option for defining the AS Service Data Description is to use XML. XML is used over Ut interface for user configuration of the supplementary services. For the supplementary services defined in the 3GPP TS 24.6xx series, XML schemas are defined in each specification.

Using the service based schema definitions is thus one possibility. If this option is selected, there is no need to invent new encodings, when the simulated services are extended in future. With a possible increase in the complexity of rules supported it will be increasingly difficult to maintain a binary data model which has to accommodate for all possible combinations.

As data size may become a problem for some implementations for an XML data set, compression is needed as an option. The simplest solution for compression would be to use gzip. Using gzip, provides for a reasonably good compression, in particular for larger data sizes. Using EXI is another option. It is currently only a late draft specification, but it can be expected that this standard will gain a more general acceptance.

Using the XML as specified in the 3GPP TS 24.6xx series and adding the option of a standard compression algorithm provides for a simple definition of the services data model, which will be simple to maintain. When new services with XML schemas are defined in the future, the service data format can be taken as is, saving design time in the standardization bodies.

### 8.3.2 Required changes to standards

To specify XML as the data format some changes are required to the Sh interface. It is needed to separate the specified simulation services data from any possible proprietary extensions. In order to achieve this, different Service Indicators shall be used for

- the services that are standardized under simulation services and defined in TS 24.6XX series
- extension services. Different service indicators will produce different XML document (of type tTransparentData) that can be managed differently for the standardised services and the proprietary extension services.

The services authorization and activation information needs to be added in the information as it currently is not specified in the simservs document. It is also necessary to add schema for the subscription options for standardised 24.6xx series services.

A version control is needed in order to be able to handle new standard releases. A version control element can either be defined as a version per blob, or inside the blob. The former requires a change to Sh, but would allow for accessing the version without opening the XML document.

---

## 9 Conclusions and recommendations

### 9.1 Discussion

#### 9.1.1 Summary

In this Technical Report two options for standardizing the Service Data Description have been discussed; one based on a binary data description, and the other based on XML as specified for the simulation services. The following table summarizes the advantages and disadvantages of the alternative proposals.

Table 9.1.1-1: Discussion Summary

	Binary approach	XML based approach
Advantages	<ul style="list-style-type: none"> <li>- Efficient encoding in terms of memory</li> <li>- Efficient processing in terms of parsing, as already in a form easily read by low level languages.</li> <li>- Optimized when very large majority of users use only simple services</li> </ul>	<ul style="list-style-type: none"> <li>- Allows full flexibility in implementing services</li> <li>- Less standardization work for new services based on XML user service configuration.</li> </ul>
Disadvantages	<ul style="list-style-type: none"> <li>- Limits some services, in particular rule based combinations, as the supported combinations need to be specified and enumerated.</li> </ul>	<ul style="list-style-type: none"> <li>- Pure XML is memory consuming</li> <li>- XML parsing affects AS processing</li> </ul>

## 9.1.2 Memory Consumption

A binary approach is considered to be more efficient in terms of memory consumption. A binary approach is well optimized to handle users with simple services. It may be argued that this is important to the HSS performance for a network with a large proportion of users using only a few of these services.

When services become more advanced the ratio between XML memory sizes and binary memory size become smaller as it is expected that the actual service data takes a larger proportion of the data (as opposed to the XML containers), e.g. a white list is a list of URIs, and these URIs may then take a larger proportion of data irrespective of solution.

To reduce the memory size for the XML format compression may be used. The two main alternatives would be to use either gzip, which is widely used as a general compressor, or EXI under development by W3C as an XML based alternative.

## 9.1.3 AS Processing Efficiency

Much the same efficiency arguments as for memory efficiency apply for the binary approach when it comes to processing efficiency.

XML parsing affects AS processing efficiency, although there is currently a rapid development with hardware accelerators for improved efficiency.

## 9.1.4 Service Flexibility and Interoperability Concerns

The binary approach limits the rule and condition based services to the combinations of rules and conditions that are enumerated.

The XML approach allows full flexibility in defining the services.

Two sets of diverging arguments are:

- a) For the services as defined today, as well as for possible future services, the possibility is offered to the user (Ut Interface) to combine rules freely in a XML document, but such flexibility with large combinational possibilities for some service configuration by the user may be differently handled by various TAS and generate interoperability issues.

To practically ensure interoperability between Telephony Application Servers, there should be defined a content subset of the MMTel services to be supported by Telephony Application Servers, optionally completed by service extensions allowing additional flexibility or new services. Such a subset of services is also sufficient for a large majority of users that have a simple use of services, extensions allowing to serve other users with richer services. This is taken into account in the binary approach.

- b) It is unacceptable to standardize a data format that does not fully support the services as they are defined. This implies that proprietary extensions are needed in order to fulfil the services as specified.

It is unclear whether it is in CT4 remit to specify what the content subset of MMTel services should be.

The rules and conditions in an XML file are deterministic. Given a set of rules, each consisting of a set of conditions it is well defined how to handle them. For CDIV rules are executed in order, and a rule matches as soon as all its conditions evaluate to true. When a rule matches the associated action is performed and the rule evaluation is completed.

### 9.1.5 Future Standardization

The extension mechanism identified in the TR that applies to the binary approach allows service extensions to existing services or new services without impacting existing services.

For the binary approach addition of new services will require extra specification of the possible combinations of rules and conditions.

For the XML based approach the service related data can be taken as is from the service specifications, provided that the specifications continue to contain XML schemas for the services.

For both alternatives possible new subscription options need to be specified. Authorization and activation codes are a trivial extension regardless of data format.

## 9.2 Recommendations

Based on the analysis done in previous sections, it is recommended:

- to specify a definition of the IMS Telephony Supplementary service (MMTel) data understood by a variety of TASs;
- to define a binary format for these service data in the context of Rel 8;
- to define an XML format based on the one defined for the Supplementary Services in the 24.6xx series in the context of Rel 8;
- for the binary format include a service description specifying the services and impacts on the IMS Telephony Supplementary services for which this approach enables interoperability. For the binary approach it shall also be described how an optional extension mechanism works that supports more complex features for some of the services, or introduces new services;
- for the XML format, to include a possibility for compression, based on either gzip or EXI;
- this will be complemented by the necessary mechanism for transmission over the Sh interface;
- and finally, the inserting of this data into the Sh User Data AVP in a manner such that additional transparent data can also be transmitted.

---

## Annex A: Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
09-2008	CT#41	CP-080481			Approved in CT#41	2.0.0	8.0.0
12-2008	CT#42				Copyright Notification updated	8.0.0	8.0.1