Chapter 3 MPEG-4 IPMP-X Specification

This chapter will introduce MPEG-4 IPMP-X Specification[1] that extends the current IPMP capabilities of MPEG-4. It provides extensions to the currently specified IPMP specification(the “Hooks”), documented in ISO/IEC 14496-1:2001[1].

3.1 Hooks vs Extensions[1]

The following features compare MPEG-4 IPMP Hook with MPEG-4 IPMP Extension(IPMP-X):

(1) IPMP extensions support use of the “hooks” so that multiple tools by different vendors can cooperate through a secure and open interface.

(2) Using hooks, unable to consume the content protected by another IPMP system which is not supported by decoder.

(3) Extensions, specify interoperable ways of communication between IPMP subsystems, and where also how to obtain them.

(4) All IPMP subsystems, encryption algorithms will be replaceable plug-in under IPMP extension

3.2 Term and Definition[1]

We will briefly introduce the important terms about this specification. The later section will introduce in detail about the action of these terms

(1) IPMP Tool –IPMP tools are modules that perform IPMP functions such as authentication, decryption, watermarking ,etc.

(2) IPMP Message – Messages are directed to a given IPMP Tool to enable, assist or facilitate its operation.

(3) IPMP Tool Manager- The IPMP tool manager is a conceptual entity within the terminal that processes IPMP tools and retrieves the tools that specified therein.

(4) Message Router- A conceptual entity within the terminal that implements the terminal-side behavior of the terminal tool interface.

3.3 IPMP-X Architecture

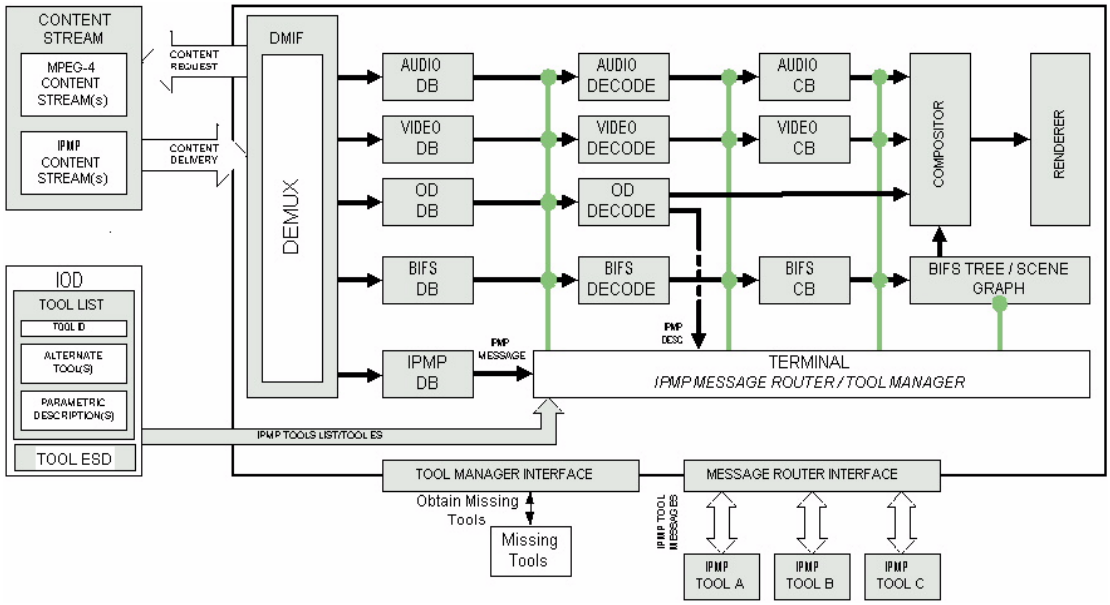


Figure 3-1 Mapping of IPMP Extensions to MPEG-4 Systems Architecture,[1]

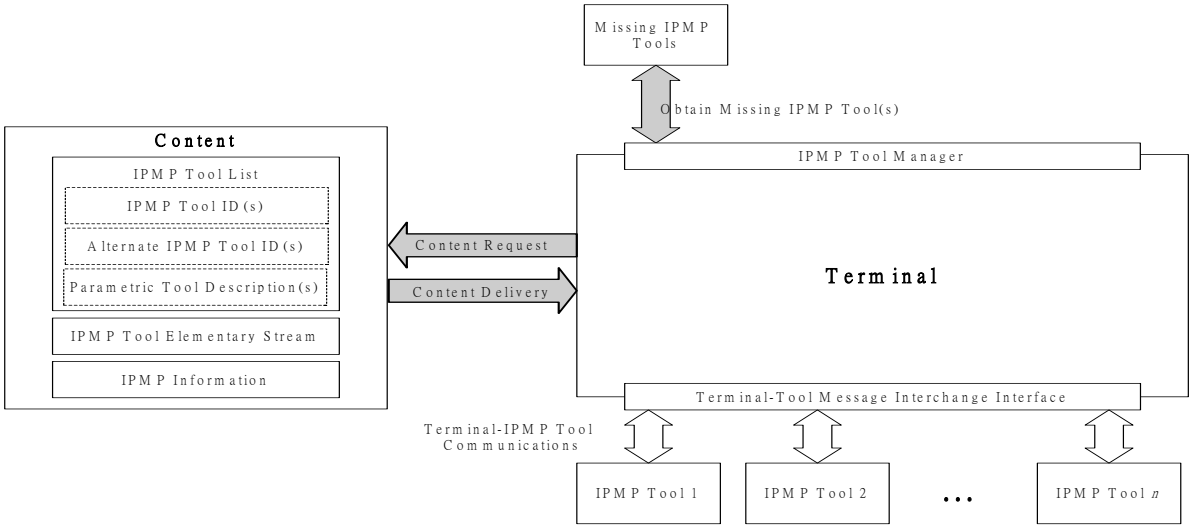


Figure 3-2 Architecture Diagram for walkthrough concept,[1]

The architecture of IPMP-X terminal is illustrated in figure 3-1.IPMP-X defines a set of mutual authentication messages that can be used to verify the trust of IPMP Tools/Terminal, and provides a secure channel for the exchange of messages between any pair of IPMP Tools or between IPMP Tools and the IPMP-X compliant terminal. So, MEPG-4 IPMP-X architecture provides a secure communication interface, but IPMP-X does not specify the interface to transfer messages. It is not MPEG standard. The following is characteristics of IPMP-X terminal.

(1) Message based using a communication interface

(2) The Message Router (MR) is conceptual component of the Terminal. It plays the role of receiving and delivering tool to tool and tool to terminal messages.

(3) To provide the design and use of any type of tool and the placement of tools at some control points

(4) To provide Mutual Authentication between tools.

(5) To provide the delivery of tools in the bitstream.

The architecture diagram in figure 3-1 shows the overview concepts about

MPEG-4 IPMP-X and provides a possible sequence of steps involved in the consumption of content protected by the IPMP Extensions in figure 3-2.

After we have seen that MPEG-4 IPMP-X architecture is shown in figure 3-1, we will focus on the terminal processing .The terminal processing contains the next stage.

(1) User request specific content[1]

1. IPMP requirements on the terminal should be placed with or before media requirements on the terminal.
2. Access Information and/or restrictions should precede media stream deliver information.

(2) IPMP Tools description access (refer to section 3.4)

(1) IPMP Tool retrieval (refer to section 3.5)

(2) Instantiation of IPMP Tools (refer to section 3.6)

(3) IPMP Information Routing and Consumption Permission (refer to section 3.7)

The next section , we will concentrate on the terminal processing including IPMP Tools description access, IPMP Tool retrieval, Instantiation of IPMP Tools and IPMP Information Routing and Consumption Permission.

3.4 IPMP Tool description access

3.4.1 Delivery of Tools via Content[1]

The IPMP Tool may be embedded in the terminal or carried within the bitstream for specific IPMP Tool implementation. A stream Type (Table 3-1) “IPMPToolStream” is defined to carry binary IPMP Tool .A given tool is carried as the payload of one IPMP Tool stream, the representation format, package information and IPMP Tool ID of which is specified in DecoderConfigDescriptor in the associated ESD as shown in figure 3-3.

Scheme 3A:

```
Class IPMPToolES_DecoderConfig extend DecoderSpecificInfo: bit(8) tag =DecSpecificInfoTag
{
bit(128)    IPMP_ToolID;
bit(32)          Tool_Format_ID;
bit(16)          Tool_Package_ID;
}
```

The class in scheme 3A indicates that the ID of the Tool(IPMP_ToolID) carried in this stream on some representation forms(Tool_Format_ID, eg:structure,binary) and on some package tool form.(Tool_Package_ID,eg Zip,Tar)

Table 3-1 Stream Type Table,[1]

StreamType value	StreamType description
0x00	Forbidden
0x01	ObjectDescriptorStream
0x02	ClockReferenceStream
0x03	SceneDescriptionStream
0x04	VisualStream
0x05	AudioStream
0x06	MPEG7Stream
0x07	IPMPStream
0x08	ObjectContentInfoStream
0x09	MPEGJStream
0x0A	InteractionStreamType
0x0B	IPMPToolStream
0x0B-0x1F	Reserved for ISO Use
0x20-0x3F	User Private

The DecoderConfigDescr is included in Es Descriptor. The relationship of Object descriptor and Es descriptor is described in figure 3-3.

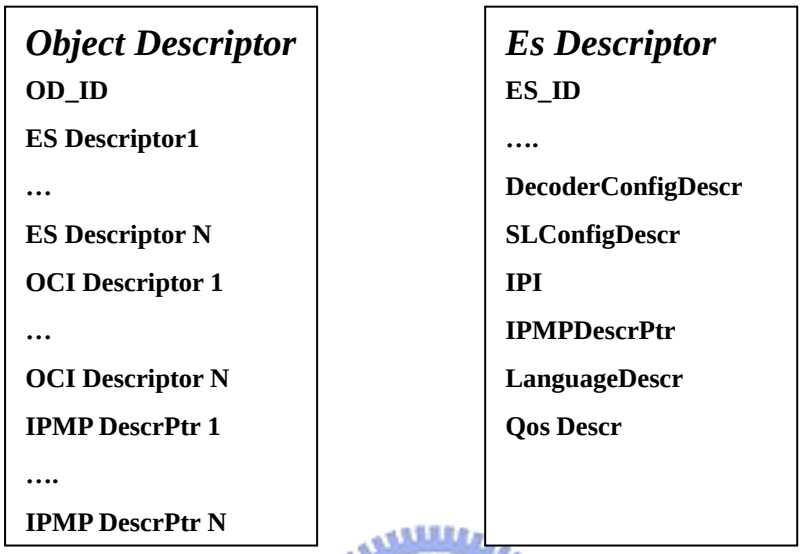


Figure 3-3 MPEG-4 essential parts of an Object descriptor and ES descriptor

3.4.2 IPMP Tool List Descriptor[1]

The figure 3-4 shows that IPMP Tool List Descriptor shall be in the Initial Object Descriptor. The Initial Object Descriptor is a key element necessary for accessing MPEG-4 content. It contains additional element to signal profile and level and at least two elementary stream descriptors. One is the scene description stream and others are object descriptor stream.

IPMP Tool List Descriptor

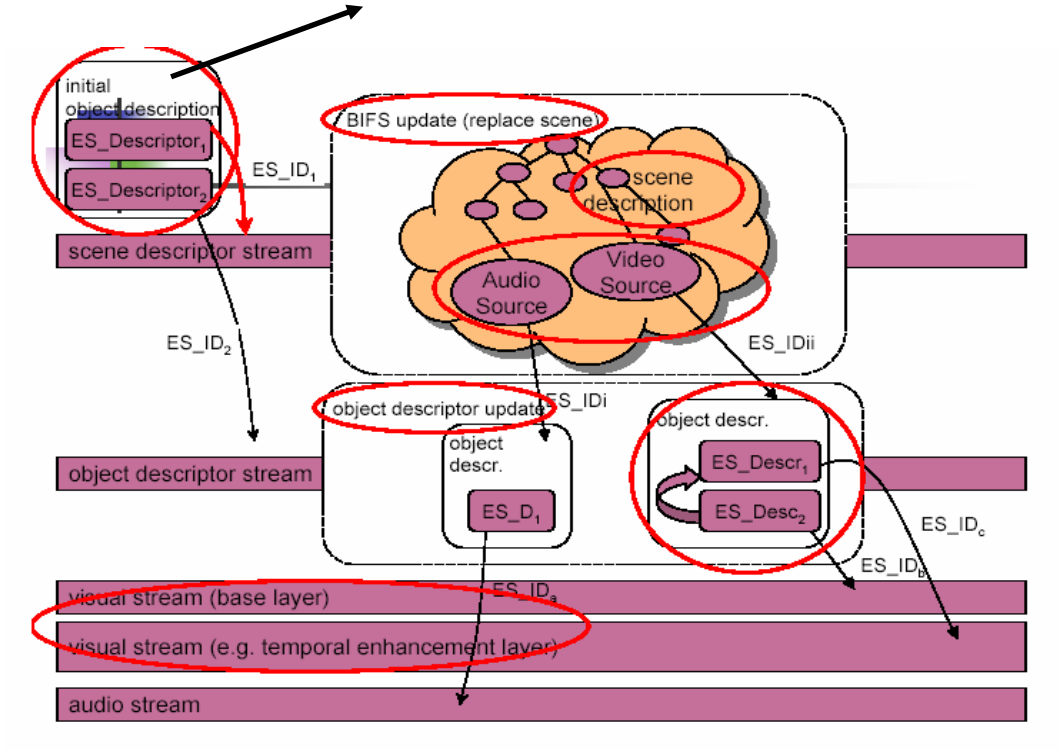


Figure 3-4 Initial Object Descriptor linking scene description to elementary stream, [31]

```
class IPMP_Tool extends BaseDescriptor:bit(80) tag=IPMP_ToolTag
{
        bit(128)    IPMP_ToolID;
        bit(1)      isAltGroup;
        bit(1)      isParametric;
        const bit(6)reserved=0b0000.00;
        if (isAltGroup){
                bit(8)numAlternates;
                bit(128)    specificToolID[numAlternates];
        }
        if (isParametric)
                ByteArry toolParamDesc;
        int(8)numURLs;
        ByteArray ToolURL[numURLs];
}
```

The main semantic of IPMP_ToolListDescriptor in scheme 3B describes a tool either as a unique implementation, as tools working together, or through a parametric description.

3.5 IPMP Tool retrieval[1]

This section describes IPMP Tool retrieval. The IPMP Tool Manager is a conceptual entity in a given IPMP terminal. Upon receipt of the Tool List, the IPMP Tool Manager parses the IOD for Tool retrieval. The following steps detail the process of parsing and retrieval of tools.

1. The IPMP Tool List arrives in the IOD and is processed by the Tool Manager.
2. The Tool Manager parsers information for the IPMP Tools.
3. The Tool Manager checks if the required Tools are available.
4. The Tool Manager is responsible for parsing the IPMP Tool ESD and retrieving the binary IPMP Tool that is carried inside IPMP Tool ES.

3.5.1 The Related Message Class for Tool Manager

This subsection, we collect the related message classes which will be processed by Tool Manager.

Table 3-2 The Related Message Class for Tool Manager

Related Message	Class
IPMP Tool List	IPMP_ToolListDescriptor, IPMP_Tool
IPMP parametric infrastructure	ToolParamDesc
IPMP descriptor pointer	IPMP_DescriptorPointer
IPMP tool descriptor	IPMP_ToolDescriptor
Delivery of tools via content	IPMPToolES_DecoderConfig IPMP_ToolES_AU

3.6 Instantiation of IPMP Tools[1]

This section, we will describe instantiation of IPMP Tools.

3.6.1 The Characteristics of IPMP Tool Instantiation

(1) Each instantiation of an IPMP Tool shall establish a new instance of the tool , for that particular scope of protection.

(1) The terminal assigns a context identifier for the logical instance of the tool , which maps to the specific tool instance.

(2) These context identifiers shall be unique to ensure unambiguous addressing.

3.6.2 The rules of protection scope

This subsection, we describe the rule of protection scope including the next rules.

(1) Instantiation of an IPMP Tool shall occur whenever an IPMP_DescriptorPointer in an OD or an ESD is given , and the associated IPMP_ToolDescriptor is known at the terminal.

(2) IPMP_DescriptorPointer in an OD, the IPMP tool protects all streams described in the OD.

(3) IPMP_DescriptorPointer in an ESD, the IPMP tool protects the stream described by the ESD.

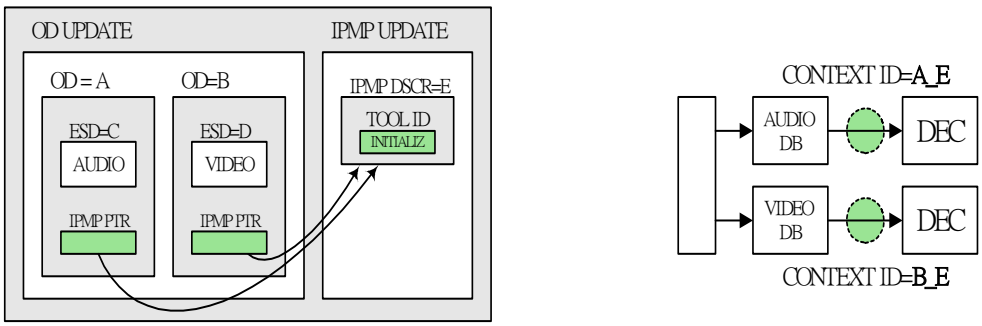
(4) IPMP_ToolDescriptorUpdate and no IPMP_DescriptorPointer referring to this IPMP_ToolDescriptor is found.

1. No impact on ESDescriptor or ObjectDescriptor protection

2 IPMP stream are only used to convey data to IPMP tool instance.

3.6.3 Tool Context Data

The tool context data may contain tool, the description of tool and the scope of protection. It is formed by two different ways: one is descriptor and the other is declarator. From the figure 3-5, we can be easy to understand relationship of the tool context data .The tool context data describes the tool associated with its scope. The relationship of declarator is shown in figure 3-6. The more examples is illustrated in the specification[1].



Descriptor Example

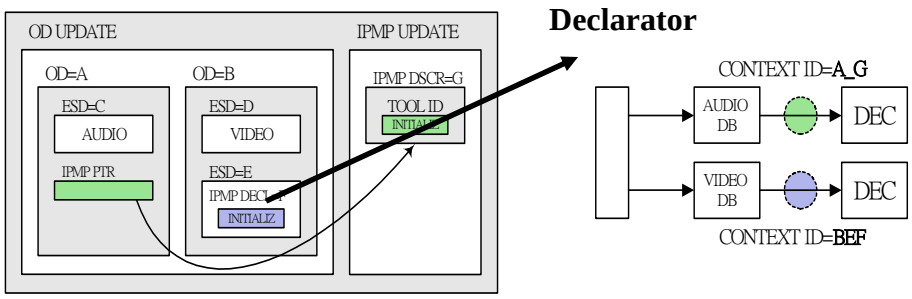


Figure 3-5 Examples of Use and Scope of IPMP Descriptors and Declarators,[1]

.

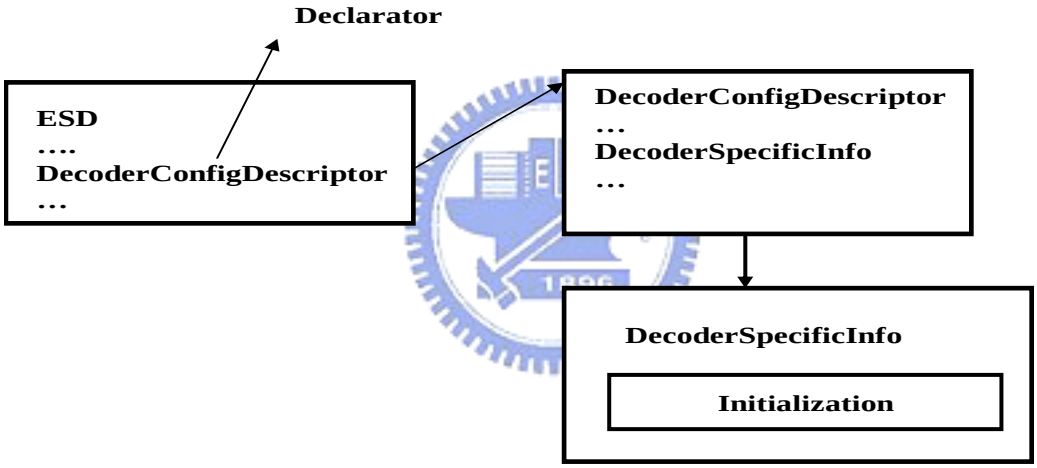


Figure 3-6 Relationship between ESD and DecoderConfigDescriptor

3.6.4 The meaning of IPMP Tool Control Point[1]

The Control Point means at which the IPMP Tool resides such that the IPMP Tool knows where it must perform its module. IPMP control point is defined in the IPMP_Initialize class of the IPMP_ToolDescriptor class[1]. It may be one(Table 3-3) of the following condition.

Table 3-3 Control Point Code Description, [1]

ControlPointCode	Description
0x00	No control point
0x01	Control point between the decode buffer and the decoder. This is between the decod buffer and class loader for MPEG-Jstreams
0x02	Control point between the decoder and the composition buffer
0x03	Control point between the composition buffer and the compositor
0x04	BIFS Tree
0x05-0xDF	ISO Reserved
0xE0-0xFE	User defined
0xFF	Forbidden

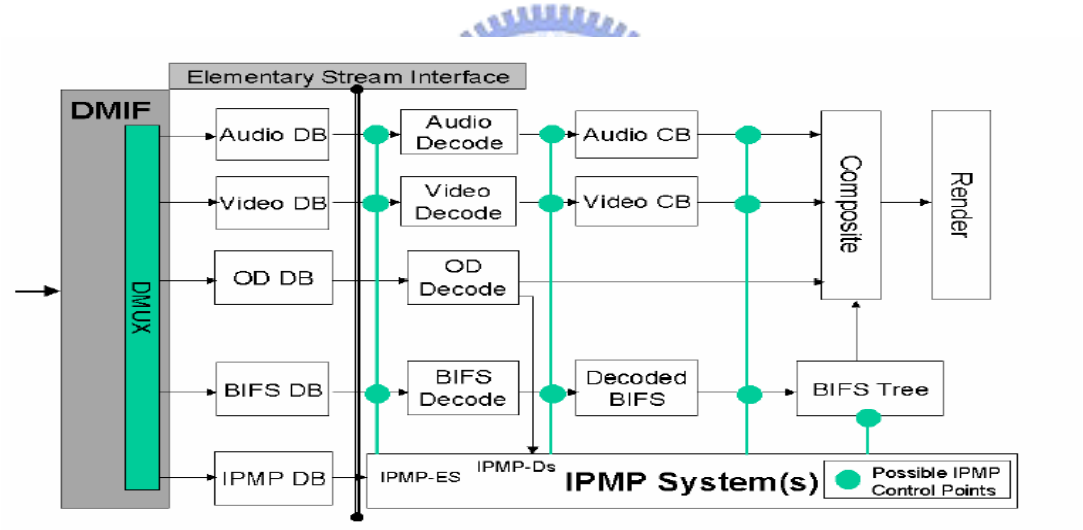


Figure 3-7 MPEG-4 IPMP,[5]

3.6.5 The message class for Tool Instantiation

Some message may affect tool instantiation such that the tool context is deleted ,instantiated or updated. We highlight these message in the following.

(1) IPMP_Initialize

(2) IPMP_ToolDescriptorPointer

(3) IPMP_ToolDescriptor

(4) IPMP_ToolDescriptorUpdate

(5) IPMP_ToolDescriptorRemove

(6) IPMP_StreamDataUpdate

(7) DecoderConfigDescriptor

3.7 IPMP Information Routing and Consumption Permission[1]

This section, we will describe the Message Router and highlight some features of IPMP message.

3.7.1 The characteristics of Message Router

(1) The Message Router (MR) is a conceptual component of the terminal plays the role of receiving and delivering tool to tool and tool to Terminal messages.

(2) Message Router implements the terminal-side behavior of the terminal-tool interface

(3) Parsing and sending the information to Tool Manager message interface handling

(3) Addressing

3.7.2 The Characteristics of IPMP Message

IPMP information may come from four sources.

(1) The content

(2) The terminal resource

(3) Remote resources

(4) Another IPMP Tool

Next, we will describe IPMP_ToolMessageBase because it is base class of the

Tool Message.

Scheme 3C:

```
Aligned(8) abstract expandable class IPMP_Tool Message Base:bit(8) tag = 0
{
    bit(8)      Version;
    bit(32)     Msg_ID;
    bit(32)     sender;
    bit(32)     recipient;
}
```

The class in scheme 3C indicates:

Sender – 0x00 is reserved for the terminal

Recipient – 0x00 is reserved for the terminal

The table 3-4 lists the messages extending IPMP_ToolMessageBase.

Table 3-4 Tags for Messages Extending IPMP_ToolMessageBase

8 bit Tag value	Symbolic Name
0x00	Forbidden
0x01	IPMP_MessageFromBitstream_tag
0x02	IPMP_ToolDescriptorFromBitstream_tag
0x03-0xCF	ISO Reserved
0XD0-0xFE	User Defined
0xFF	Forbidden

The detail semantics refer to IPMP-X specification[1].

On the other hand, the Message Router uses the delivery function transferring IPMP information in the terminal. The IPMP information is delivered to the Tools specified by the information. There are two delivery functions described: one is IPMP_MessageFromBitstream in scheme 3D and the other is IPMP_ToolDescriptorFromBitstream in scheme 3E. They deliver message received in

the content to the IPMP Tool instance.

Scheme 3D:

```
Class IPMP_MessageFromBitstream extends IPMP_ToolMessageBase: bit(8) tag =
IPMP_MessageFromBitstream_tag
{
    bit(8) numMessages;
    IPMP_StreamDataUpdate message[numMessages];
}
```

Scheme 3E:

```
Class IPMP_ToolDescriptorFromBitstream extends IPMP_ToolMessageBase:bit(8) tag =
IPMP_ToolDescriptorFromBitstream_tag
{
    IPMP_ToolDescriptor toolDescriptor;
}
```

From beginning introduction to now, We can conclude these components relationship to get control flow in the below figure.

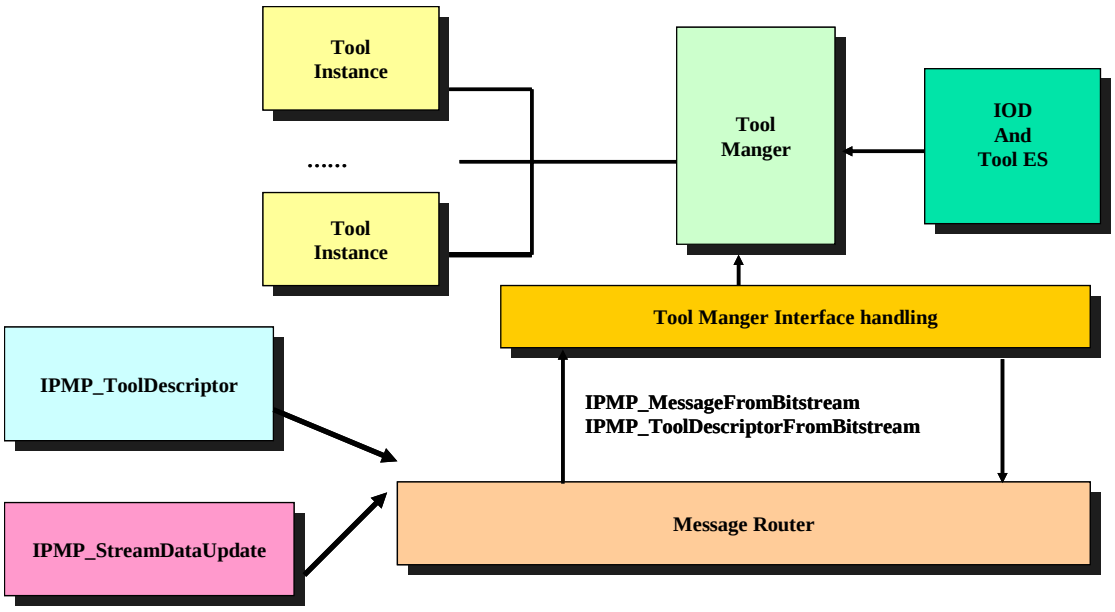


Figure 3-8 Control Flow for IPMP-X in the Terminal

The last of this section, we will discuss IPMP message. It extends to IPMP_Data_BaseClass.We know IPMP-X provides an open secure frame allowing them to cooperate with each other. The open secure frame depends on various extending Base Class Messages such that it is interoperable for the developer to implement IPMP-X application. IPMP data extending from IPMP_Data_BaseClass could be carried in the IPMP_ToolDesriptor or IPMP_StreamDataUpdate. We generalize different kind of IPMP message in below Table 3-5.

Table 3-5 IPMP Data Specification and Purpose, [7]

IPMP Tool Connection and Disconnection (Specification and Purpose)

IPMP_GetTools	This message is sent by a Tool to the Terminal to find all the tools, instantiated or not, that are available on the terminal
IPMP_GetTools Response	This message is sent by the Terminal to a Tool in reply of an IPMP_GetTools request
IPMP_ToolParam CapabilitiesQuery	This message allows a terminal to query al tool as for support for a specific parametric description
IPMP_ToolParam CapalilitiesResponse	This message is the response to the above parametric capabilities query and simply returns a boolean values.
IPMP_ConnectTool Context	This message allows a tool to request the Terminal to create a connection to a tool identified in the toolDescriptor
IPMP_DisconnectTo olContext	This message allows a tool to disconnect a tool it has previously connected at a control point

IPMP Tool Notification (Specification and Purpose)

IPMP_AddToolNotificati onListener	This message is sent from a Tool to the Terminal to request notification of event
IPMP_RemoveToolNotifi cationListener	This message is sent from a Tool to the Terminal to request stop notification of certain event
IPMP_NotifyToolEvent	This message notifies an IPMP Tool of an event for which it had previous registered as a listener

Mutual Authentication (Specification and Purpose)

AlgorithmDescriptor	This class if for specifying an identifier o an authenticaitin related algorithm
AuthCodes	Authentication codes for IPMP_MutualAuthentication message
BaseAuthenticationDescriptor	The base for authentication descriptors
Certificate	A generic certificate
DateClass	Contain the audit date of IPMP tool in question, in Universal Time ,Co-ordinated and Modified Julian Date
KeyDescriptor	This class is for specifying a cryptographic algorithm and a key conforming to the algorithm
IPMP_InitAuthentication	Message exchanged during mutual authentication process
IPMP_MutualAuthentication	Message exchanged during mutual authentication process
TrustSecurityMetadata	Message carry metadata for the verification of trust between two tools

IPMP Terminal to Terminal

IPMP_RequestContent	One IPMP Terminal requests a content fron the other IPMP terminal
IPMP_ReponseToContentRequest	The response to IPMP_RequestContent
IPMP_ContentTransfer	One IPMP terminal transfers the content to another IPMP terminal
IPMP_RequestTool	One IPMP terminal requests an IPMP Tool from the other IPMP terminal
IPMP_ResponseToToolRequest	The response to IPMP_RequestTool ,possibly containing th requested IPMP Tool
IPMP_DeviceID_Notification	One IPMP terminal transfers the content to another IPMP terminal

IPMP Processing (Specification and Purpose)

IPMP_Data_BaseClass	This class is used to carry IPMP data in the bitstream or from one Tool to another
IPMP_ByteArray	This class stores an array of bytes of know size, and convert it into binary format
IPMP_CanProcess	Sent from a Tool to terminal to allow or refuse content processing
IPMP_OpaqueData	This class is used for carriage of opaque data
IPMP_KeyData	This message carries key and synchronization information for decryption tools
IPMP_RightData	The IPMP_RightData contains rights information and can be carried in either the IPMP_ToolDescriptor or IPMP Stream
IPMP_SelectiveDecryptionInit	This message initialize a decryptor tool
IPMP_AudioWatermarkingInit	Delivers to an audio watermarking tool all the information about the characteristics of audio content
IPMP_SendAudioWatermark	An audio watermarking tool that has been required to perform payload extraction will construct this IPMP data
IPMP_VideoWatermarkinginit	This message delivers to a watermarking tool all the information about the characteristics of the video content, the type of action to be performed on it, and possibly other related proprietary data required by the watermarking tool
IPMP_SendVideoWatermark	

User Interaction (Specification and Purpose)

DTArray	Carries text to be displayed to the user
QTArray	Carries a prompt to be displayed to the user
RTArray	Carries an user's answer
OptionArray	Carries an option that may be selected by the user
IPMP_UserQuery	Used to query the user for information
IPMP_UserQueryResponse	Carries an user's answer to an IPMP_UserQuery

XML Schema (Specification and Purpose)

Schema for Terminal Platform	An XML schema to describe the termainal platform that an IPMP tool can run on
Schema for Parametric Description	The content provider can describe what type of tool is required to playback the content