

SMPTE ENGINEERING GUIDELINE

Media Dispatch Protocol (MDP) — Engineering Guideline (Informative)



Table of Contents

	Page
Foreword	4
Introduction	4
1 Scope	5
2 MDP Document Structure	5
3 What Problem Does MDP Address?	6
4 How Does MDP Relate to...?	7
4.1 Applications	7
4.2 Transfer Protocols	7
4.3 File Formats	7
4.4 Security Technologies	7
4.5 Metadata	8
4.6 Web Services	8
4.7 Middleware	8
4.8 Proprietary Delivery Solutions	8
4.9 BXF	8
4.10 TV-Anytime, ADI, SIP/SDP, etc.	8
5 How Does MDP Work	9
5.1 Organizations and Agents	9
5.2 Projects	10
5.3 Transaction	10
5.4 Initiator Target Agents	10
5.5 Push-Mode, Pull-Mode and Use Cases	10
5.6 The Manifest Document (Part One)	11
5.7 Messages and Endpoints	12
5.8 Data Structures	13
5.9 The Manifest Document (Part Two)	13
5.9.1 manifest elements	14

5.9.2	transferoption elements.....	14
5.9.3	file elements.....	15
5.9.4	transferwith elements and status properties.....	15
5.9.5	details, updated and comment properties.....	16
5.10	Transfers, Controllers, Senders and Receivers.....	16
5.11	The Capabilities Document.....	17
6	What Happens During a Typical Delivery?.....	18
6.1	Before Initiation.....	18
6.2	Initiation.....	18
6.3	Negotiation.....	19
6.4	Start of Transfer.....	19
6.5	During Transfer.....	20
6.5.1	Status check.....	20
6.5.2	Pause and resume.....	20
6.5.3	Checking received files.....	22
6.6	Completion.....	22
7	What Happens When Things Go Wrong?.....	22
7.1	General.....	22
7.2	Error Responses and reason Properties.....	23
7.3	Stalled and Failed Transfers and updated Properties.....	23
7.4	Problems that MDP Cannot Express.....	24
8	How Will MDP Be Kept Both Interoperable and Future-Proof?.....	24
8.1	Additional Profiles.....	24
8.2	Additional Mappings.....	25
8.3	Transfer Protocols.....	25
8.4	Capabilities Document.....	26
9	Can MDP Carry Metadata?.....	26
10	What Must Be Implemented and What is Optional?.....	27
11	How Can Transfers Be Scheduled and Prioritized?.....	27
12	How Can Agents Discover Each Other?.....	28
13	How Can Good Transfer Performance Be Obtained?.....	28
14	How Can Deliveries Be Secured.....	28
14.1	General.....	28
14.2	Authentication.....	29
14.3	Integrity Check.....	29
14.4	Manifest Document Properties.....	29
15	Can Parts of Files Be Transferred?.....	29
16	How Can MDP Work through a Firewall or NAT?.....	30
17	How Can MDP Be Used within a Department or Facility.....	31

18 Can MDP Be Used for Streamed or Live Content?	31
19 Can MDP Be Used for Delivery to Multiple Recipients	31
Annex A Requirement for Post-Production File Transfer.....	32
A.1 General	32
A.2 Target Use Cases	32
A.3 Simplicity.....	32
A.4 Packaging	33
A.5 Content	33
A.6 Transfer	33
A.7 Security	33
A.8 Performance and Reliability.....	34
A.9 Control and Monitoring	34
A.10 Standards	34
Annex B Glossary	35

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in Part XIII of its Administrative Practices.

SMPTE Engineering Guideline EG 2032-4 was prepared by Technology Committee S22 on Committee on Television Systems Technology.

Introduction

The Media Dispatch Protocol (MDP) is a means of orchestrating the delivery of media files over IP networks. It defines a standard mechanism for implementations to initiate a delivery, to negotiate the details of the delivery, to provide information about the progress of the delivery, and to provide a confirmation of the outcome of the delivery.

It is important to note that MDP is not a transfer protocol. Instead it allows organizations to transfer files at agreed times, using agreed transfer protocols, and using an agreed set of secure technologies (where appropriate). Furthermore, MDP is not overly prescriptive in regards to which protocols or technologies shall be used; rather it provides a framework which allows organizations to choose those that best suit their own needs.

This document is the MDP Engineering Guideline. It provides an introduction to the protocol, explains what problem it is intended to solve, and outlines how it works. It also provides information what is contained in the normative MDP specification documents.

1 Scope

This Engineering Guideline gives an introduction to the Media Dispatch Protocol (MDP) for the orchestration of the delivery of media files over IP networks. It describes the purpose of the protocol, and presents an outline of the technology involved, including an overview of the messages and data structures of the protocol, and how the protocol can be extended to meet future scenarios.

The Guideline provides advice on how the protocol can be used in practice, and what should be implemented. It also outlines how common transfer protocols and security mechanisms might be used in conjunction with MDP.

2 MDP Document Structure

The MDP specification is split into a number of separate parts (see Figure 1) in order to create a document structure that allows new applications to be covered in the future. These parts are:

Part 1 (Normative), the MDP protocol specification (SMPTE 2032-1).

Part 2 (Normative), the MDP mapping specifications (e.g. SMPTE 2032-2 is the MDP/XML/HTTP mapping).

Part 3 (Normative), the MDP profile specifications (e.g. SMPTE 2032-3 is the MDP Basic Target Pull profile).

Part 4 (Informative), the MDP Engineering Guideline (this document).

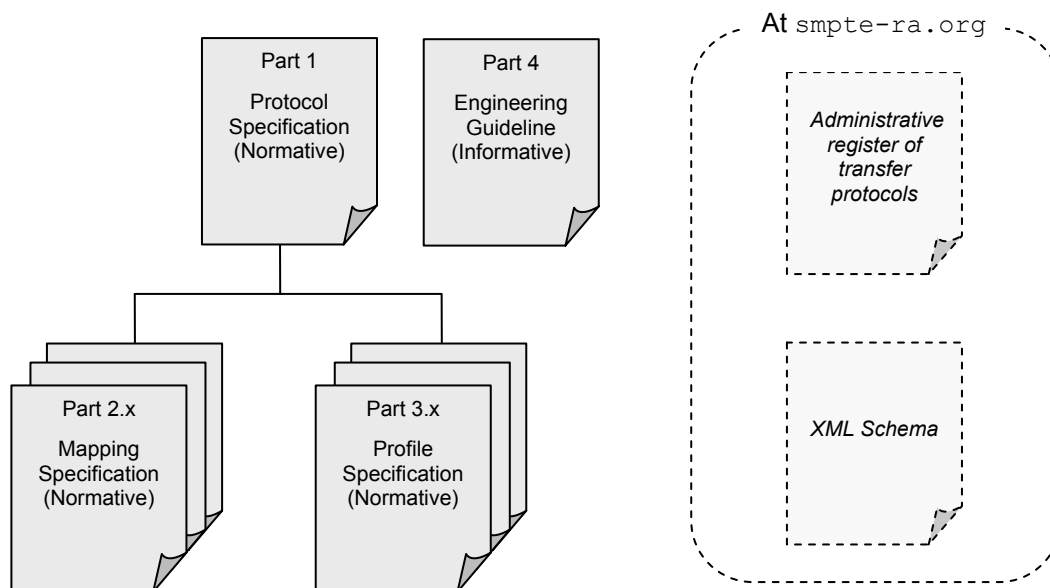


Figure 1 – MDP document suite

When implementing MDP, you should ensure that you have the latest version of all of these documents. The individual mapping and profile specifications will be independently updated.

This document is the MDP Engineering Guideline (Part 4). It provides an introduction and description. This document should be read first because it introduces many of the concepts and explains what problem MDP is intended to solve. It is presented as a set of questions (or an extended FAQ). Concepts are introduced gradually, and, where necessary, more detail is added later. This makes the document easier to read, but less good as a reference, so you are encouraged to make full use of the table of contents. Annex B of this document lists the defined terms in the specification.

The MDP protocol specification (Part 1) is at the core of the MDP standard. It defines the syntax and semantics of the messages and data structures used within the protocol.

The MDP mapping specifications (Part 2) define how the messages and data structures of Part 1 are represented on the network. At present there is only one defined mapping (MDP/XML/HTTP); however, as new user and technical requirements are identified, additional mappings will be produced. Each mapping specification will have its own document.

The MDP profile specifications (Part 3) define subsets of the messages and data structures of Part 1, and place constraints on the use of these subsets. The purpose of this is to make it easier to achieve interoperability between implementations. At present there is only one defined profile (Basic Target Pull); however, as new user and technical requirements are identified, additional profiles will be produced. Each profile specification will have its own document.

The specification also normatively references an “Administrative Register” document on the SMPTE Registration Authority website. This contains a list of transfer protocols that have defined names with MDP (for instance “HTTP”). This will be updated as required to incorporate new protocols without the need to issue a new version of a specification document.

The MDP document suite makes reference to several non-SMPTE standards. Unlike many SMPTE specifications, the majority of these are from the Internet and Web community. Many are “Request For Comment” documents (RFCs) from the Internet Engineering Task Force (IETF), or from the World Wide Web Consortium (W3C). A Bibliography for the document suite appears in the protocol specification.

To improve the clarity in all parts of the MDP document suite, `text in this typeface` is used to indicate the names or values of messages and data structures used in the protocol.

3 What Problem Does MDP Address?

The use of data files to represent audio and video content has revolutionized the television industry. For example, almost all editing occurs using non-linear systems, and most playout occurs from disk-based servers. However, at the time of writing (early 2007) transfer of content as files has so far largely been confined to scenarios where the content formats and transfer protocols can be carefully controlled. In particular, delivery between different organizations, or different departments of the same organization, still often occurs through the use of physical media such as videotapes, or via SDI and video lines.

As the cost of metropolitan- and wide-area connectivity falls, we can expect the demand to increase for file-based media delivery over IP-based networks. Examples include delivery of content between production facilities and broadcasters, and provision of agency news feeds. The success of such an approach depends on the availability of fit-for-purpose network interconnectivity, but it also depends on the systems at either end understanding one another. While the simple transfer of a file is a trivial problem (FTP is available on almost every platform), this only provides part of what is required. To avoid the need for manual intervention during every delivery, we need a mechanism to automatically initiate, supervise and audit the transfers. In addition because there are so many different transfer protocols in existence, and so many ways of configuring a transfer, the systems need to agree just what is going to happen.

There are on the market various proprietary file delivery solutions that solve these problems. However, differing systems do not interoperate, which in practice means that an organization wishing to exchange

content with multiple partners needs to deploy multiple systems. In many cases this would be unacceptable, and a standardized approach is needed.

MDP provides part of such a standardized approach. In particular it provides a mechanism to allow organizations to agree on the details of a delivery (what files are to be sent, when transfer will occur, what transfer protocol will be used), to manage its lifecycle, and to exchange information about its progress.

MDP was initially developed by a working group of the Professional MPEG Forum that investigated requirements and technologies for the exchange of large media files over IP networks. File delivery within the post-production community was seen as an important use case, and Annex A provides a summary of the relevant requirements identified by the group in this area.

MDP was therefore initially developed to be suitable for file transfer within the post-production community. However, it contains no features that tie it to such applications, and is suitable for use in other scenarios, for instance delivery of finished programs to playout centers. In fact, MDP is not specific to media files and could be used in any scenario requiring the transfer of large files between organizations.

4 How Does MDP Relate to...?

As MDP is in some ways an “invisible technology” it is helpful to clarify how it relates to other technologies.

4.1 Applications

A production or other media organization typically creates, manages and processes files using a selection of applications such as non-linear editing systems, media asset management systems and transcoding engines.

Sometimes these applications will need to exchange these files with applications belonging to other organizations. Because there are a wide range of possible mechanisms for performing these exchanges, it makes sense if these are not implemented in the applications themselves, and instead the tasks are delegated to specialized software agents. Then, as new transfer protocols, or other technologies, are adopted, it is not necessary to change the code within the applications themselves. As discussed in section 5, MDP provides a standard way for such agents to communicate.

4.2 Transfer Protocols

MDP is not a transfer protocol. Instead it carries information about what transfer protocol might be, or is being used. To ensure interoperability the standard requires all implementations to support HTTP and HTTPS, but allows the use of other protocols.

Although MDP was developed for deliveries over metropolitan- and wide-area networks, it can also be used with protocols more suitable for local-area networks, for example copying of files from a shared network file system such as NFS.

4.3 File Formats

Although MDP provides a logical grouping of files that are involved in the same delivery, it is not a file wrapper or container format like MXF (SMPTE 377M). It is expected that delivery of MXF files will be an important use of MDP, but the protocol makes no assumptions or requirements of the type of files transferred (they do not even have to be related to audio or video).

4.4 Security Technologies

Often it will be necessary to deliver files over public networks such as the Internet. MDP does not in itself ensure that deliveries are secured, but allows security technologies such as TLS (Transport Layer Security) to

be used, and enables implementations to exchange information about what security measures are supported before starting the transfers.

4.5 Metadata

MDP is not a metadata protocol, and it has no defined relationship with any metadata standard or recommended practice (e.g. SMPTE RP 210).

MDP can be used to transfer files containing metadata, e.g. within an MXF file or in a separate XML file.

The standard forbids the carriage of “dark metadata” within the protocol; this is discussed further in section 9.

4.6 Web Services

“Web service” simply means any software system that supports interoperable machine-to-machine interaction over a network. Therefore MDP implementations can be considered as web services.

However, “web service” is also often used to mean a service that uses SOAP (Standard Object Access Protocol) envelopes over the network and WSDL (Web Services Description Language) to describe its interface. At the time of writing, MDP does not use SOAP or WSDL; however, this could be added in a future mapping specification.

4.7 Middleware

“Middleware” can mean various things, but in general is software that connects other items of software. This could be for reasons of interoperability, or to provide a distributed architecture. MDP arguably could be seen as a type of middleware.

4.8 Proprietary Delivery Solutions

Many file delivery solutions currently on the market make use of high-performance, but often proprietary, transfer protocols. MDP is not intended to replace such solutions; the adoption of an MDP interface could aid the uptake of these new protocols, and is highly encouraged.

4.9 BXF

The Broadcast eXchange Format provides a standard for interchange of data and metadata among professional broadcast systems. It is intended for transferring schedule and playlist information between traffic, automation and similar systems. Although BXF includes messages to order content transfers, the intended application domain is different to that of MDP.

4.10 TV-Anytime, ADI, SIP/SDP, etc.

There are various other non-SMPTE protocols and interfaces that also cover some aspects of orchestration of content delivery, for example:

TV-Anytime defines standards for distribution of content for home storage.

ADI (Asset Distribution Interface) defines how to place content assets in context for video-on-demand distribution.

SIP (Session Initiation Protocol) and SDP (Session Description Protocol) manage and describe multimedia communication sessions, and the IETF is studying extension to support file transfer.

However, these are aimed at solving different problems to MDP, and are aimed at different application domains.

5 How Does MDP Work?

This section explains the main concepts within the protocol, with particular emphasis on the MDP/XML/HTTP mapping and the Basic Target Pull profile.

Figure 2 shows the context for which MDP is intended to be used, and what is defined in the protocol specification.

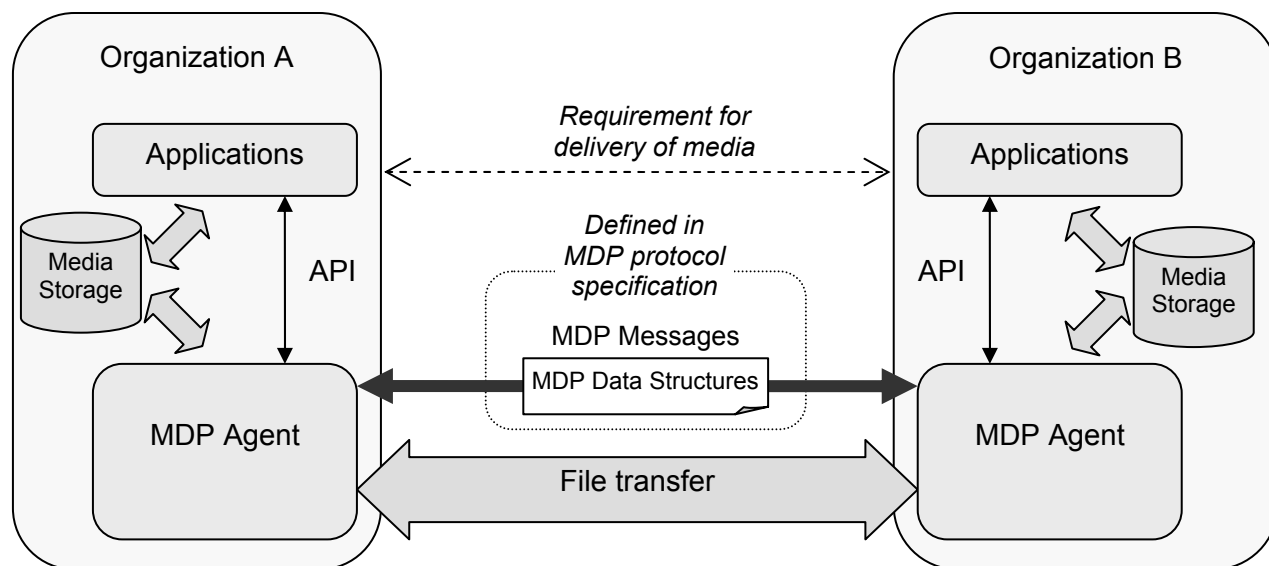


Figure 2 – Context for Media Dispatch Protocol

5.1 Organizations and Agents

The MDP specification uses the terms “organization” and “agent” extensively. The following summarizes their meaning:

An agent manages deliveries on behalf of an organization.

An organization is someone or something that has a requirement for file delivery; this might be a private company, a public corporation, a department, possibly even an individual person.

An organization is identified within MDP by a simple string. A domain name such as `broadcaster.com` has the advantage of global uniqueness, but the standard does not require the use of these, and other identifiers could be used, for instance based on X.500/LDAP distinguished names (e.g. “CN=Broadcaster”).

An agent will normally be a piece of software, typically running on a computer with network connectivity. Generally an agent will offer some form of interface. This might be a user interface such as a web page allowing a user to request and check on a delivery. More importantly it might be API allowing an application such as an editing system or media asset management system to automatically invoke a delivery, and receive information about its progress. Please note that the MDP specification does not define or require any interface; this is out of scope of the standard.

An agent is identified by its message endpoint URL, as discussed in section 5.6.

Note: In practice, a single instance of agent software might handle deliveries on behalf of more than one organization (e.g. `mdpagent.com` might handle deliveries for `aaa.com`, `bbb.com` and `ccc.com`). However, this is of no relevance to the MDP standard, even if the agent uses the same message endpoint URL when acting on behalf of more than one organization.

5.2 Projects

Whatever the organizations are, there will always be a reason for deliveries between them, for example the deliveries might be regarding program number `P12345678`. In MDP this reason is called a “project”, and is identified by a simple string. This string does not have to be globally unique; it just has to be sufficient to identify the project for the two organizations.

5.3 Transactions

The sequence of steps that the agents take with regards to a delivery (of any number of files) is called a “transaction”. This starts with one agent telling another that a delivery is required. The agents then negotiate the details of how the delivery will occur (for example which files will be sent, what transfer protocol will be used, when transfers will start), and initiate file transfer operations as appropriate. During transfer, the agents can exchange information about the progress of each transfer. After the transfers have completed, the agents exchange details about the outcome of each transfer. See section 6 for more detail about the lifecycle of a transaction.

A transaction is identified within MDP by a simple string. This does not have to be globally unique; it just has to be sufficient to identify the transaction for the organizations and project involved. However, use of a globally unique identifier such as a GUID is suggested as a straightforward means of ensuring this is the case.

5.4 Initiator Target Agents

The specification uses the terms “initiator agent” and “target agent” extensively:

A transaction starts with the initiator agent telling the target agent that a delivery is required.

5.5 Push-Mode, Pull-Mode and Use Cases

In the above statement, “A delivery is required” could mean several things. For the Basic Target Pull profile, the statement could be written as:

A transaction starts with the initiator saying that it would like the target agent to pull some files across the network.

In this profile the initiator agent is typically “at the sending end”. The target agent is “at the receiving end”, and initiates all the transfers, in other words the transfers are pull-mode. MDP calls this a “target pull use case”, hence the name of the profile. An example of a scenario that is well-suited to a target pull use case is where a production company delivers television programs to a broadcaster; the production company knows when the content is available and so is in the better position to initiate the transaction, while the broadcaster can initiate the transfers at times that are best suited to its own schedules.

Other use cases can be identified that may be better suited to other scenarios (see Figure 3):

In an initiator push use case, the transaction is again initiated “at the sending end”, but the initiator agent initiates push-mode transfers.

In an initiator pull use case, the transaction is initiated “at the receiving end”, and pull-mode transfers are used.

In a target push use case, the transaction is initiated “at the receiving end”, and push-mode transfers are used.

MDP can also (in theory) support mixed use cases, e.g. where the transaction is initiated “at the sending end” but some transfers are push-mode while others are pull-mode.

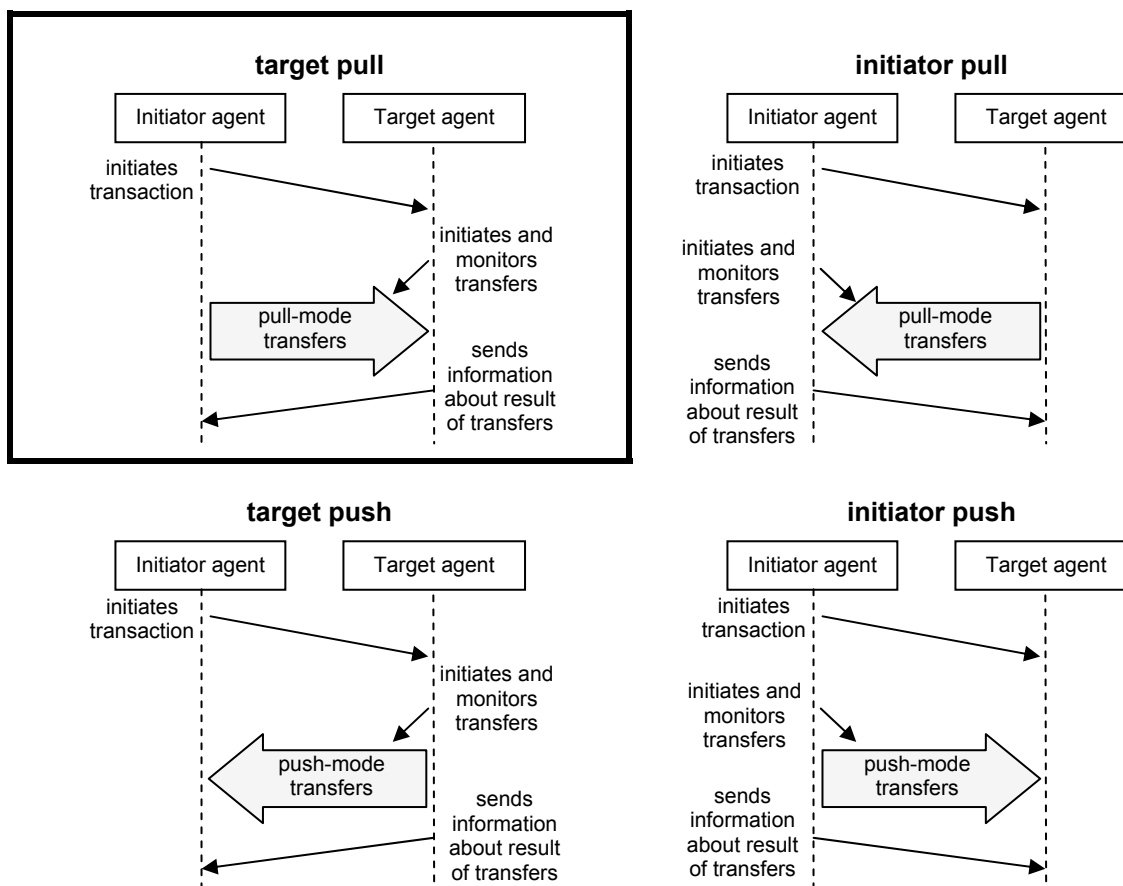


Figure 3 – Use cases

Although at the time of writing no profile exists that allows any use case other than target pull, the specification includes reserved values enabling these to be added in the future.

Note: For simplicity, transfers are shown between as occurring between the agents in Figure 3. As explained in section 5.10, this does not have to be the case. This is why the phrases “at the sending end” and “at the receiving end” are in quotation marks.

5.6 The Manifest Document (Part One)

The manifest document is fundamental to the way in which MDP works (in fact during development, the protocol was often just called “Manifest”). It is a document that is sent between the agents and provides information about the transaction. As the transaction progresses, the agents send one another manifest documents to negotiate the details of the transfers that will occur and provide information on their progress.

The information in the manifest document includes (among other things):