

DAVID
s y s t e m s

WHITEPAPER

INSIDE **MOVES MEDIA**



This paper explains some technical details of elements that make up the Moves Media product family. Functional descriptions of the modules are available separately.

Table of Contents

Introduction	3
DigaReplicator:	4
High Speed WAN Filetransfer.....	4
DigaMailbox IP	6
Security.....	6
Robustness	7
DigaTransfer System	8
Common features.....	8
Broker features.....	9
Worker features	9
Analyzer features	9
TransferClient features.....	9
DigaPorter	10
Basic Workflow.....	10
Joint features	11
Enhanced metadata handling	11
Built-in transcoding functionality.....	12

Introduction

DAVID Systems Moves Media – more than just shifting files

DAVID Systems' Moves Media technology provides a powerful and easy-to-use set of modular applications to exchange media files and their associated metadata globally. The product family consists of applications to replicate data via WAN or LAN connections; an IP based mailbox system for highly secure and bi-directional file exchange ; and software for the load balanced transport of data within one large system or between any number of separate systems. All of these software modules are freely combinable, centrally administered, format independent and allow full background operation. The Moves Media applications are executing their tasks almost invisibly for the editorial staff and come with intelligent features such as powerful metadata handling, on-the-fly transcoding or automatic database registration. Nevertheless, all these applications are based upon standard IT infrastructure and integrate seamlessly into existing system environments. By being able to use the existing network infrastructure or IP connections to exchange media content also results in significant reductions to be achieved over conventional satellite transfer.

Originally developed and optimized for use in big broadcast networks, the Moves Media applications have broadened their reach to deliver proven and leading performance wherever huge amounts of content have to be distributed or where metadata integrity and synchronisation are essential.

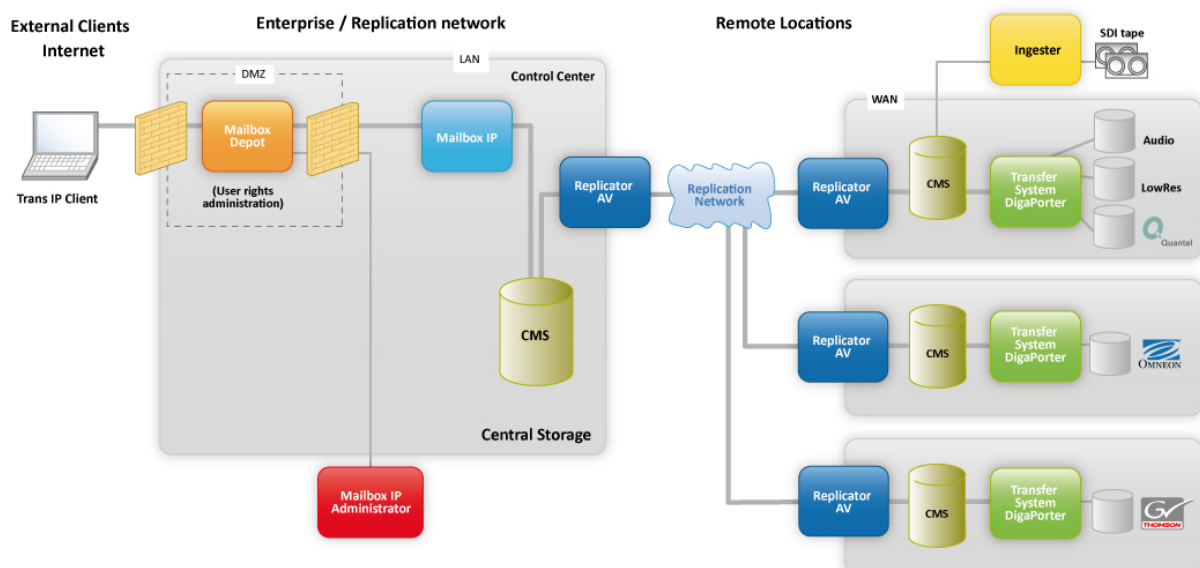


Figure 1: An example of how Moves Media applications are used in a broadcast organisation. An external client sends content to the headquarters from where it is redistributed to regional sites, each of which has its own format and metadata requirements

DigaReplicator:

DigaReplicator is a WAN optimized system to connect CMS Systems in different locations. It allows the bi-directional exchange of files and metadata as well as offering interfaces for external vendors to connect to it.

High Speed WAN Filetransfer

What is the challenge?

Various Tests and Experiences in WAN-environments showed that TCP significantly underutilizes the bandwidth in high speed long distance networks. As an example, on a 400 MBit/s WAN connection, an FTP transmission achieves only 60 MBit/s average speed, while the same application on a 1 GBit/s LAN connection reaches transfer speeds easily up to the available bitrate.

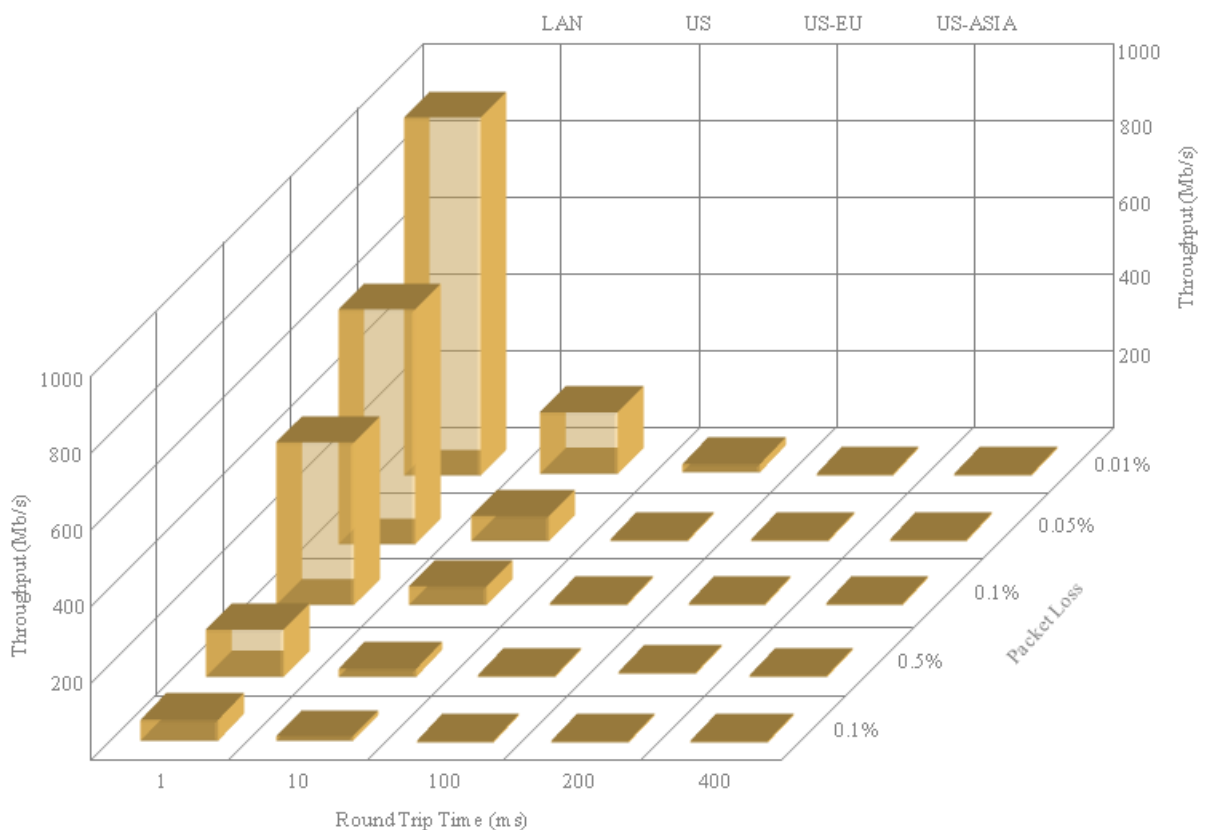


Figure 2: How TCP/IP Throughput Performance varies with increasing Round-Trip Time and Packet Loss

In figure 2, it shows how TCP/IP throughput performance varies depending upon RTT (Round-Trip Time) and Packet Loss. The effects are mainly caused by the non WAN friendly handshake mechanism used in TCP/IP, which is not optimized for long RTT values. This explains why TCP/IP based protocols like FTP and HTTP don't perform on file transfers over WAN with RTT > 10 ms and some Packet Loss. There are alternative methods of transfer which have been developed to try to patch this behaviour (different TCP/IP implementations or multi-socket transfers), but all of them have had some form of major drawback in the form of either specialist equipment requirements or a redesign of the applications. This is why DAVID Systems looked to solve this problem with its implementation of UDT transfer technology.

What is UDT?

UDT stands for UDP-based Data Transfer Protocol and is built above UDP at the application level. It provides similar functionalities to that of TCP, so that existing applications can be moved to UDT with little effort and application developers do not need much time to learn UDT's semantics. To address efficiency and fairness, UDT provides a new protocol design and implementation, as well as a new congestion control algorithm. Based on UDP it also adds connection-oriented reliable duplex unicast data streaming on the application level. From the developers' perspective, it can be used similarly to TCP/IP, but with a much better performance in Wide Area Networks. It is also firewall friendly, as in UDT-M mode, it only needs one open UDP-Port on the receiver and transmitter side.

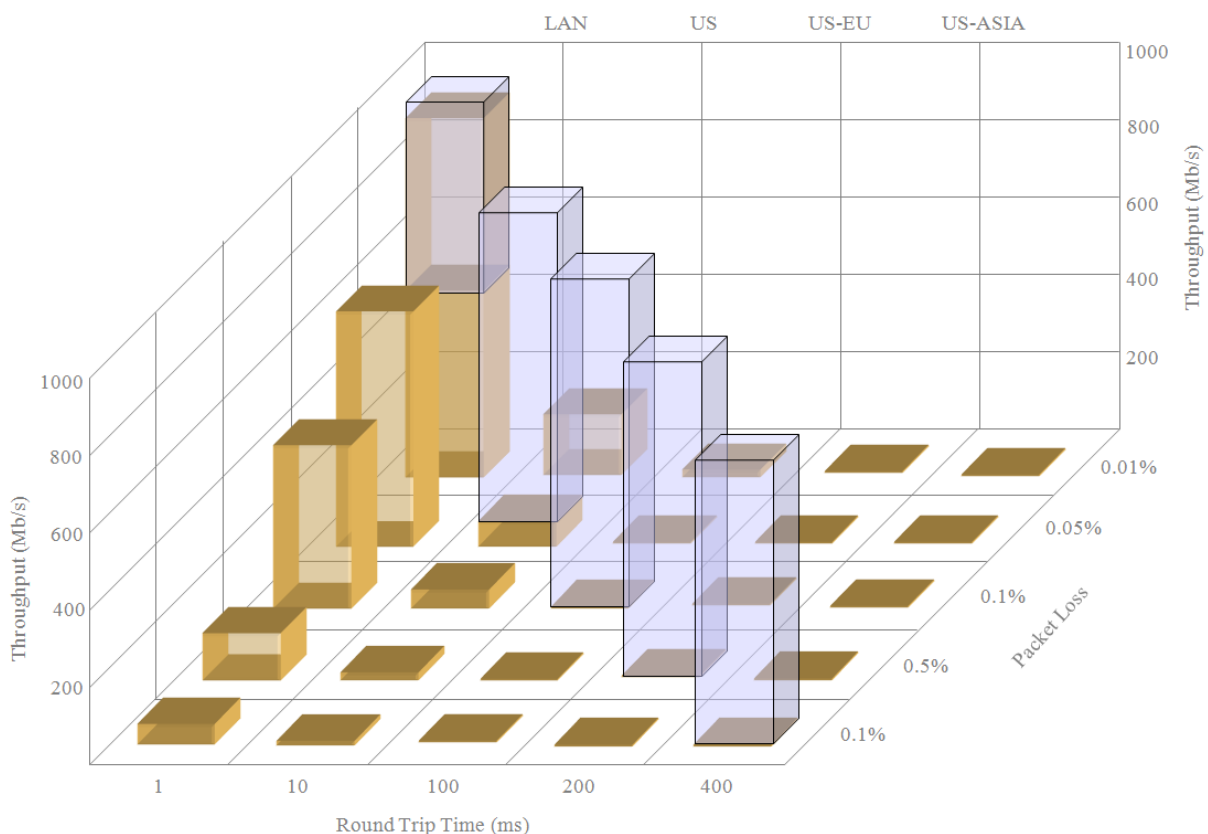


Figure 3: Showing the optimized behaviour of UDT in WAN environments.

DigaMailbox IP

DigaMailbox IP is an Internet optimized system for exchanging content between a station (e.g. Radio or TV) and individual organizations like studios, production companies or correspondents.

Security

Security is an important subject area when it comes to file transfers over the internet. Most commercial or public organisations look to avoid or minimize any external access through their firewalls from external locations, to reduce the risk of a cyber-attack or unauthorized access to confidential information and material. However, many of these same organisations, especially in the media and telecommunications industry want to still be able to offer an FTP or HTTP service through open ports in their firewall, so that external users can access these services from outside. There is always a certain degree of risk in this, because a bug in the implementation can lead to a security problem, leaving the system vulnerable to unauthorized access. A typical example is the so called 'buffer-overflow', where specially prepared data packets, can be used to execute malicious program code on the server machine. Once the server is under control, all other machines behind the firewall are in danger, because security-holes exist everywhere and will be used.

Because of these dangers, people use a so-called DMZ (Demilitarized Zone) to connect their own servers and services to the Internet. This means that all servers that are connected to the Internet are secured via a second firewall to the internal system. Every transfer to and from the servers within the DMZ should always be initiated from inside the firewall and never the other way around. As the internal firewall does not open any ports or services to the outside, even a hacked server in the DMZ cannot harm the internal system.

This is also the basic concept behind the DigaMailbox IP. Figure 4 below shows the different components a DigaMailbox IP system is made up of.

Access from the Internet is only allowed to the DigaMailbox Depot, which can be put in the DMZ. It buffers all data which goes to and from the internal system. Import and Export of Files is handled by the DigaMailbox IP, which exclusively connects from the internal network to the DMZ, like an ordinary web-browser. As with a browser it uses port 80 (http) or port 21 (ftp) for all communication, so no special firewall settings have to be applied. In addition to http, https is also possible for transfers, but this needs an official certificate placed on the server site. The inner firewall (firewall 2) does not need to open any ports from the DMZ to the internal network and so avoids opening any security holes. The outer firewall (firewall 1) needs to be opened only for http, ftp and if requested for https access. In the event of any of the services of DigaMailbox Depot being compromised, only the underlying service (the depot) itself will be out of order, without any of the services within the internal network being affected.

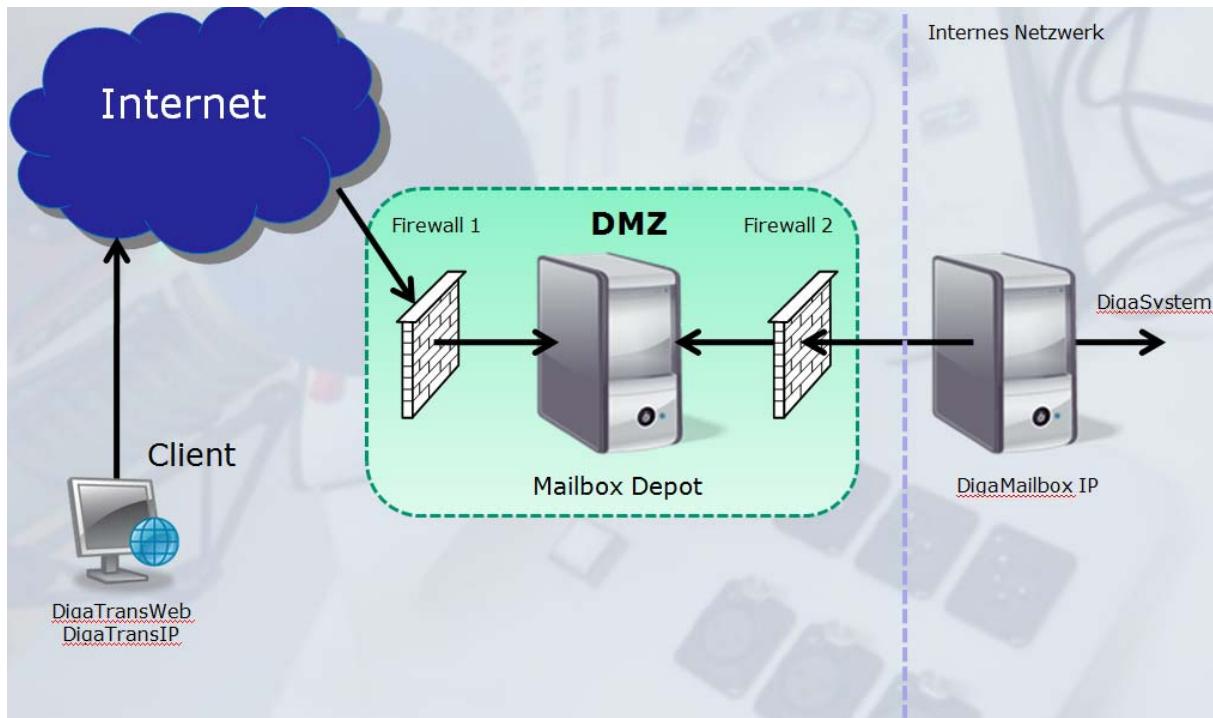


Figure 4: DigaMailbox IP and DMZ security concept

Robustness

The server applications on the DigaMailbox Depot run as Windows services and are capable of being clustered with a standard windows cluster system. Therefore it is possible to build a highly stable system, which will run even after the failure of one cluster node. Another possibility to secure the system is by using a backup depot. It is even possible to run the depot outside the DMZ, hosted by a normal internet service provider. The system can be configured so that clients automatically connect to the backup system if the main system is not reachable. Additionally, it is possible to configure the DigaMailbox IP so that sent data is also received from the Backup-Depot. Finally, the DigaMailbox IP can be secured even further using a second instance, which runs in parallel to the first instance with both applications doing their work simultaneously without disturbing each other.

DigaTransfer System

DigaTransfer is a LAN optimized system for copying, converting, uploading, downloading and analyzing files without blocking the client application.

Its architecture is structured into three layers as shown below in Figure 5:

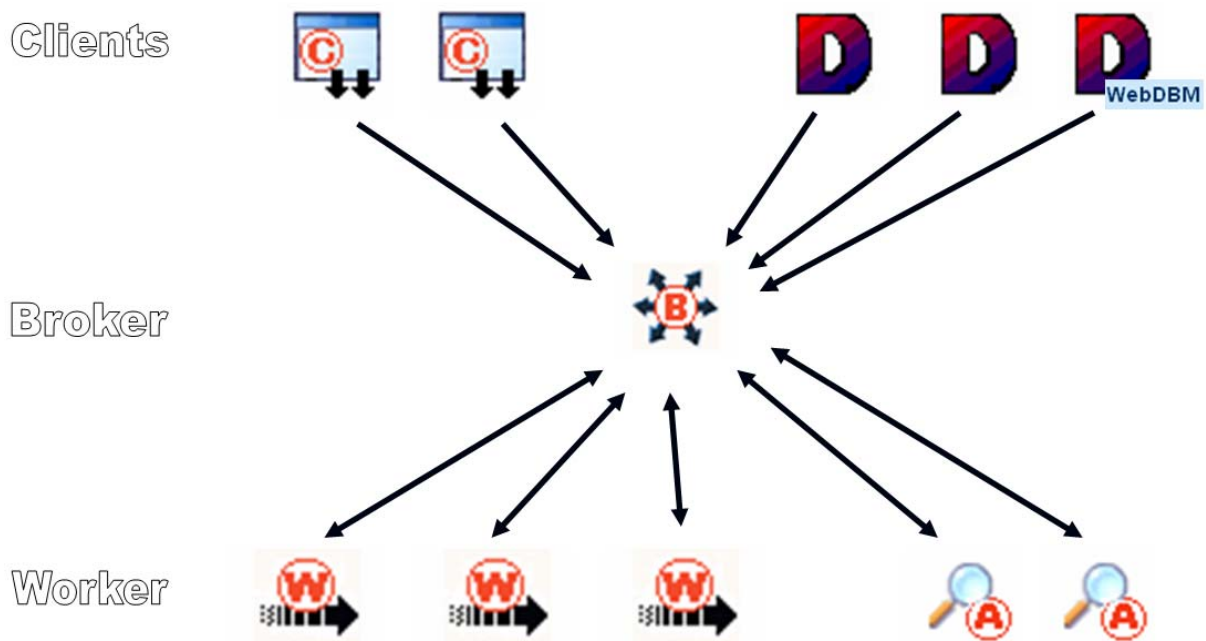


Figure 54: Overview of Transfer System Architecture

- the client application creates tasks
- the DigaTransfer Broker service manages tasks and creates/controls worker jobs
- the DigaTransfer Worker service copies/converts/downloads/uploads files
- the DigaTransfer Analyzer service analyzes or matches file formats

The layers are in communication via a SOAP interface (= web services). Third party clients can use the SOAP interface to interact with the DigaTransfer system. Different clients create tasks on a Broker and the Broker creates jobs on different Workers and/or queries different Analyzers.

Common features

The server applications (Broker, Worker) work multi-threaded and every activity is logged into a protocol file. They have a built-in web server, e.g. for displaying state, history and test pages.

Broker features



There are several routing options available for assigning jobs to Workers. It is possible to use load-balancing, to use a fixed Worker and select a Worker depending on path prefixes (e.g. directory names).

Fault tolerance is implemented by various retry options, and task states can be stored in persistent storage and can be used to restart tasks after a crash.

There is also (although not necessary) an interface to DigaSystem databases, which allows destination metadata to be changed at the end of a task (set state to EXISTING or set INVALID to 1), create new database entries for the transferred files or change source metadata (back-channel) to signal the transfer state.

Worker features



The worker is the actual workhorse that does the transfer and performs the transcoding conversion of audio and video files.

It is able to communicate, download and upload from FTP, File share and a variety of Video servers. (GVG ProfileXP, GVG K2, Omneon, Quantel sQ)

Analyzer features



The Analyzer allows the analyzing and matching of video formats to enable automatic format conversion if necessary.

TransferClient features



The client GUI can display pictures from the targets, so a user can drag-and-drop files from Windows Explorer or DigaSystem DMB database entries to initiate a task to a service provider via SOAP (e.g. Broker, Worker). It displays status & history of a task executed by a Broker and Worker and is able to display the content of various source and destination locations.

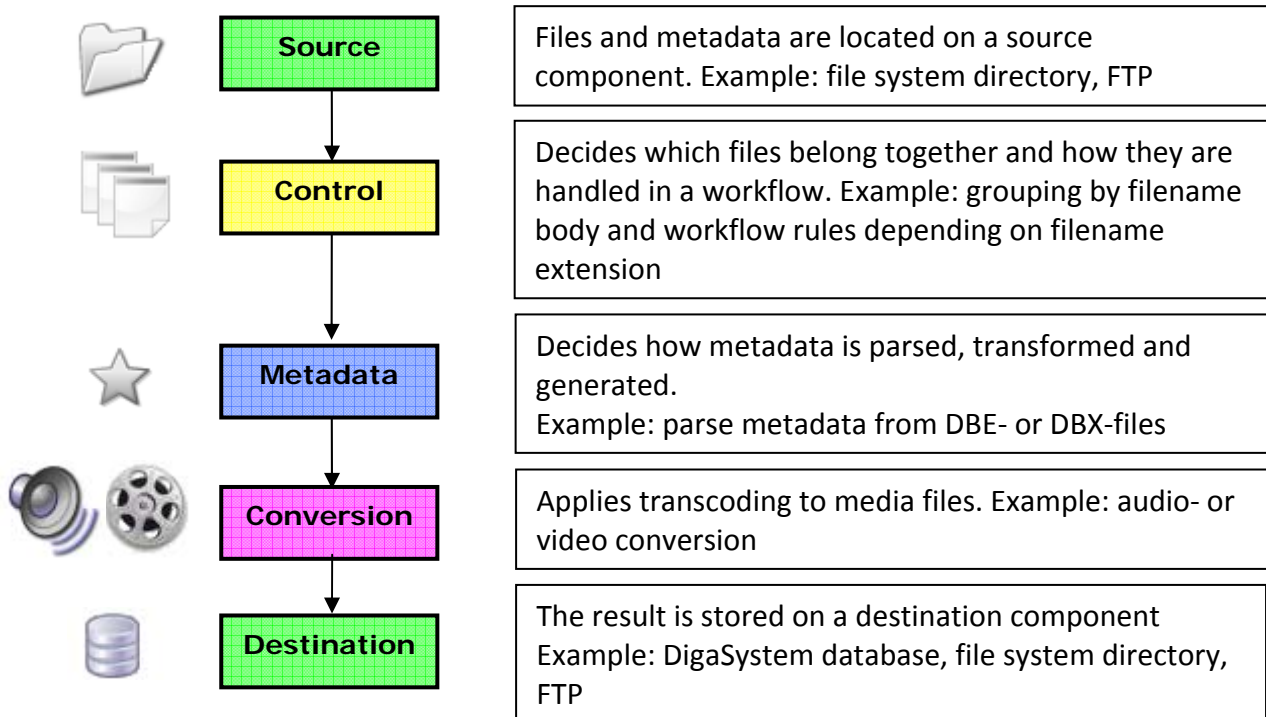
DigaPorter

DigaPorter is a SAF application module for transporting and transforming media essence and metadata from one place to another in an automated way

The DAVID Systems Station Application Framework executes application modules in a common environment. Application modules can make use of basic SAF library features and functionalities provided by available SAF extensions (a kind of plug-in).

Basic Workflow

The basic workflow is defined by 5 component types: source, control, metadata, conversion and destination. Each component type is implemented as a SAF extension (= plug-in). Thus, each component of the workflow can be replaced by a custom component.



Joint features

Enhanced metadata handling

Moves Media applications are designed not only to preserve metadata during file transfers but also to enrich them when necessary. Metadata can be added in every stage of the transfer process, either manually or automatically according to preset rules, e.g. transfer date and time or name of the sending station. Customizable metadata masks, which may contain mandatory fields as well, are available in every Moves Media application. An example of such a mask is shown in Figure 6 below.

The screenshot shows a window titled "Item Metadata" with a blue title bar and standard window controls. The main area is a form with the following fields and values:

- Title: 01 - Brain On A Stick - Belong
- Buttons: AFT (selected), VFT
- Author: BRAIN ON A STICK
- Owner: BRAIN ON A STICK
- Editor: Supervisor
- Place of Recording: Hobbs End
- Date of Recording: Dienstag, 28. August 2007
- Identifier: 123456789-987654321
- VSAT:
- Department: [dropdown]
- Language: English
- Priority: [dropdown]
- Class: MUSIC
- Source: CD
- Itemtype: [dropdown]
- Theme: NONE
- Information: [text box]

At the bottom, there are two tabs: "Comment" (selected) and "List of Files". The "Comment" tab contains the following text:

```
Belong  
(M+W by Thomas Diekmann)  
  
you're all I care  
you're all I know  
you're all I want, never letting you go  
you're all I see  
you're all to me
```

At the bottom right, there are "Save" and "Cancel" buttons.

Figure 6: Customized metadata mask for Mailbox IP

In a broadcast environment, metadata is often distributed in a separate file before the transfer of the respective essence file, thus allowing for scheduling a contribution in a rundown, reserving a free studio or planning human resources before or during the actual transfer. In this way Moves Media applications easily integrate into existing workflows.

Built-in transcoding functionality

Typical functionalities like wrapping/unwrapping, transcoding or audio extraction are integrated into all Moves Media applications. This offers the advantage that a file can be transcoded on-the-fly or wherever enough bandwidth is available. In most instances transcoding happens fully automatically according to predefined rules. A sending station can apply different rules for different recipients of the same source file (see figure 7). For every target in the list there exists a set of parameters which define recipient specific formats and metadata settings.

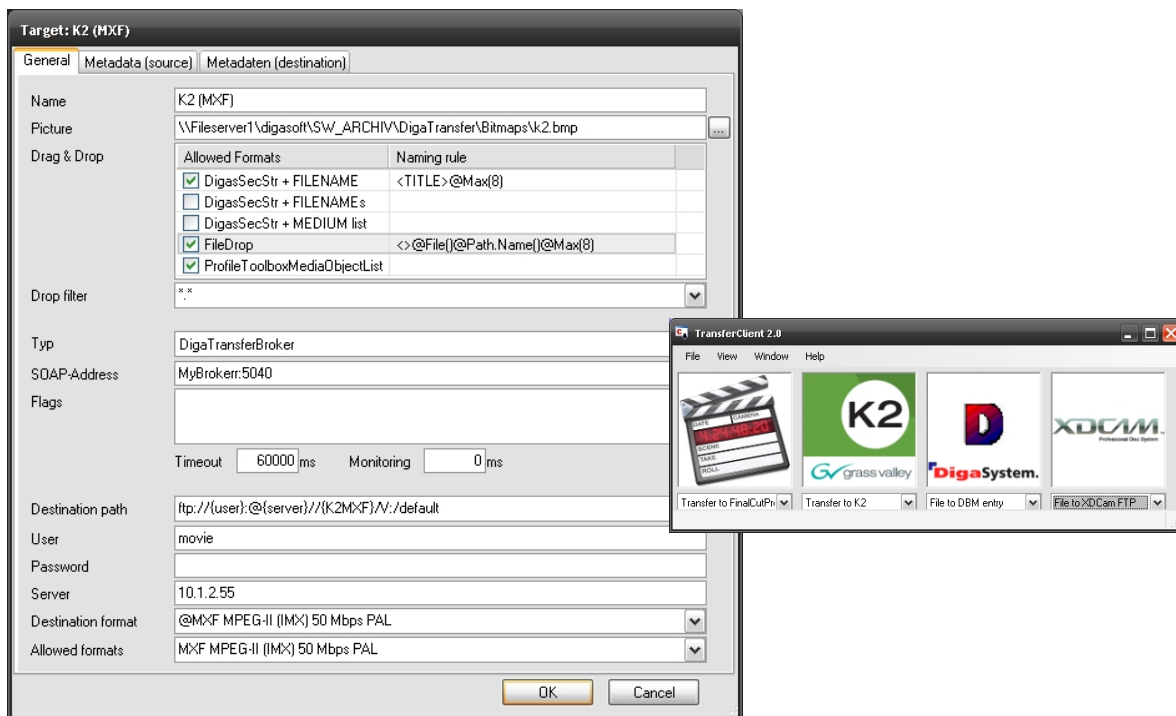


Figure 7: Target settings and list of typical targets in a Transfer System