AXMEDIS Tutorial on Content Processing

Version 1.4
December 2006

Martin Schmucker – Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, Germany

Ivan Bruno – Department of Systems and Informatics (DSI-DISIT), University of Florence, Italy

Maulik Sailor – Intelligent Media Systems & Services (IMSS), University Reading, United Kingdom

1

AXMEDIS attacks the challenges of digital content processing and distribution

Automating Production of Cross Media Content for Multi-channel Distribution

2

Table of Content

- **Tutorial Rationale**
- **Tutorial Objectives & Outcomes**
- **AXMEDIS Content Processing (AXCP) – Part I**
  - Overview
  - Automation
  - Life-Cycle
  - Core Functionality
  - Extended Functionality

3

Table of Content (II)

- **AXMEDIS Content Processing (AXCP) – Part II**
  - AXCP Engine
  - AXCP Grid
  - Usage Scenarios
- **Summary, Conclusion and Outlook**
- **Discussion, Questions and Answers**

4

Tutorial Presentation

- **Rationale**
  - Recommended: General Tutorial
  - Focus: Automatic Content Processing
    - Underlying basics concepts
    - Usage examples of tools
  - Discussion of limits and constraints
- **Documents are available at http://www.axmedis.org**

5

Tutorial Presentation (II)

- **Intended audience**
  - Decision makers
  - Technical managers
  - Programmers
  - People in the areas of
    - Content production
    - Content distribution and content protection
- **Prerequisites**
  - Basic knowledge of production cycle and tools
  - Basic knowledge of distribution and protection tools

6

Tutorial Objectives

- Recall overall automatic content processing process
- Overview on individual tools
- AXMEDIS content processing framework usage know-how
- Interaction with tools
- Description of usage scenarios

7

Tutorial Outcomes

- How to use content processing tools in the AXMEDIS framework
- How to automate content production and processing using AXMEDIS
- Interaction of workflow and content processing
- Distribution of workload by using the AXMEDIS GRID distributed environment

8

AXMEDIS Tutorial on Content Processing - Part I

- **Current Challenges and Key Issues**
- **AXMEDIS General Overview**
- **AXMEDIS Content Processing**

9

Current Challenges

- **Very fast growing digital-content market**
- **Major limitations include:**
  - Convergence of the media, interoperability of content
  - DRM applications and introduction in several distribution channel
  - Massive processing of content processing and distribution
- **Real challenges that are currently being discovered**
  - Business-to-Consumer Scenarios (B2C)
  - Business-To-Business Scenarios (B2B)
- **Required: innovative means for various scenarios**

10

Key Issue: Flexibility

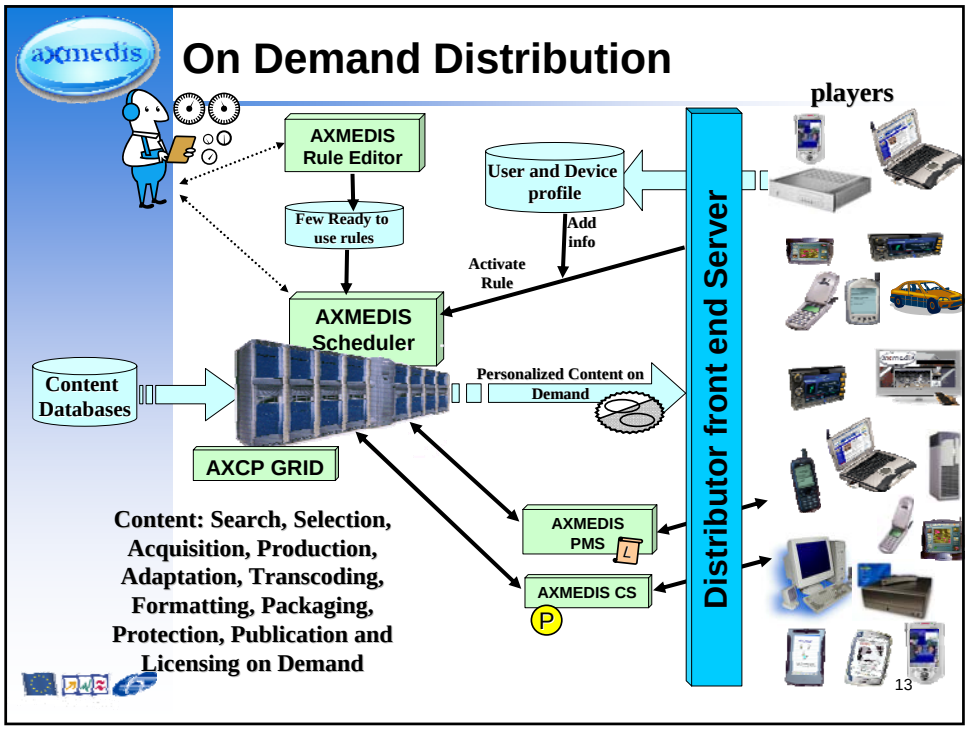
- **Devices and content delivery formats are not static**
  - Emerging devices and formats
  - Dynamic market in terms of possibilities and content types and formats
- **Required: Flexible Software Tools**
  - Support of numerous content types and formats
  - Support of different devices

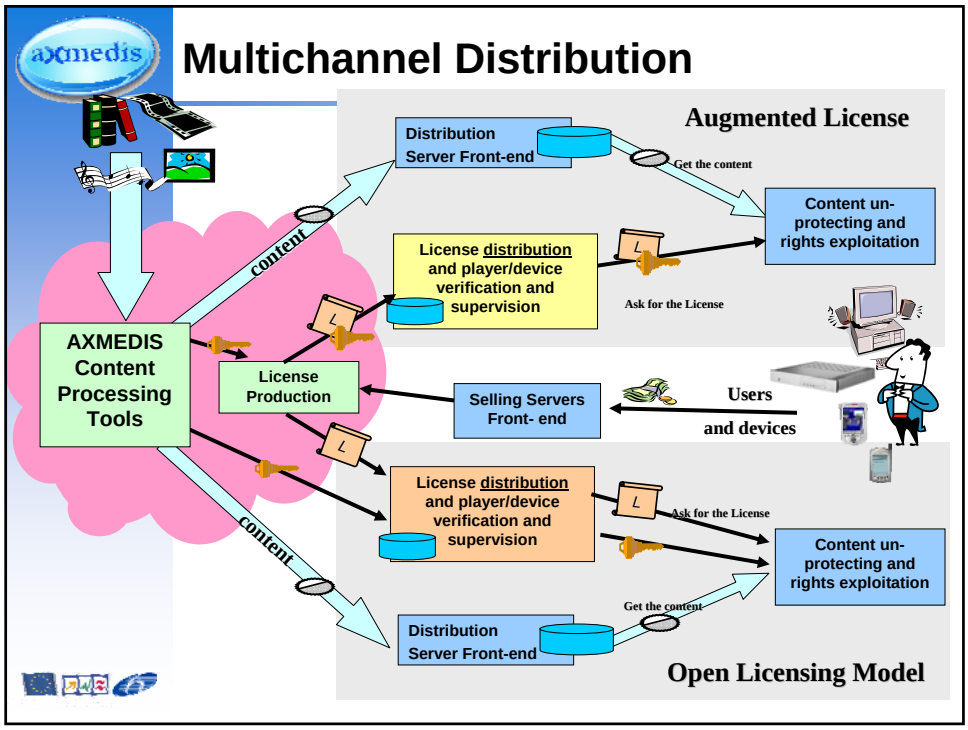
11

Key Issue: Automated Processing

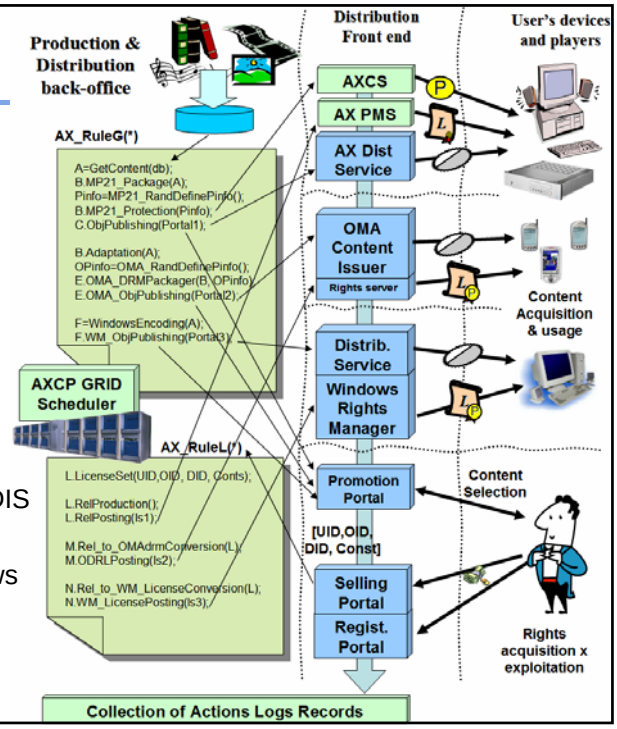
- **On-demand distribution:**
  - Production on the basis of requests and profiling (user device, network, etc.), etc.
  - Request depending adaptation and processing
- **Multi-channel distribution**
  - Differing receiving devices
  - Differing distribution modalities
  - Multiple interoperable DRMs, license chain processing
- **Content monitoring**
  - Broadcast channels and networks,
  - Peer-To-Peer networks, Websites, etc.

12


On Demand Distribution
players
AXMEDIS Rule Editor
Few Ready to use rules
User and Device profile
Add info
Activate Rule
AXMEDIS Scheduler
Content Databases
Personalized Content on Demand
AXCP GRID
Distributor front end Server
AXMEDIS PMS
AXMEDIS CS
Content: Search, Selection, Acquisition, Production, Adaptation, Transcoding, Formatting, Packaging, Protection, Publication and Licensing on Demand
13


Multichannel Distribution
Augmented License
Distribution Server Front-end
Get the content
Content un-protecting and rights exploitation
License distribution and player/device verification and supervision
Ask for the License
content
AXMEDIS Content Processing Tools
License Production
Selling Servers Front- end
Users and devices
License distribution and player/device verification and supervision
Ask for the License
Content un-protecting and rights exploitation
Get the content
Distribution Server Front-end
content
Open Licensing Model

- **Different channels**
- **Different DRMs**
  - AXMEDIS
  - OMA
  - Windows Media



AXMEDIS Advantages

- **Flexible and innovative solutions for**
  - Content production and management
  - Content distribution and aggregation
  - Digital Rights Management (DRM)
- **Cost Efficiency through underlying principles**
  - Automatic massive content production and processing
  - GRID technology
  - Extensibility through plug-ins technology
- **Cost Efficiency through the integration with**
  - Existing Content Management Systems (CMS)
  - Existing e-Service, e.g. back-office and workflow support

16

AXMEDIS Advantages (II)

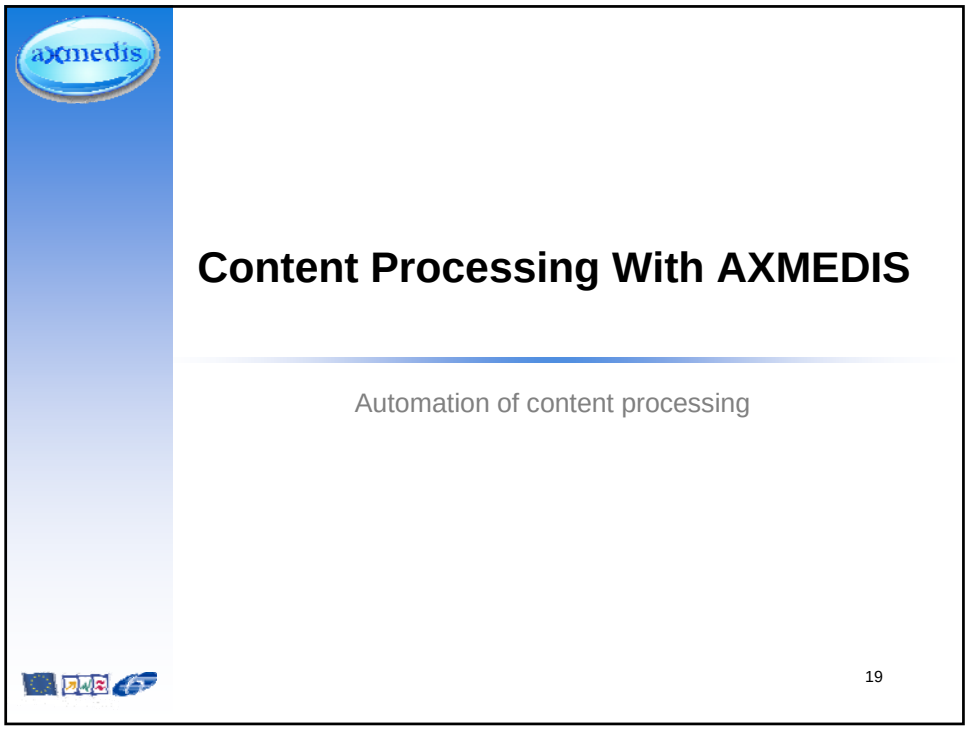
- **AXMEDIS Automated Content Processing**
  - Massive and small scale processing
  - Locally performed or Workflow controlled
  - On any kind of Digital Resource not only AXMEDIS objects
- **AXCP Applications for massive processing as**
  - Production/packaging platform for producers and integrators
  - Protection of objects, and protection information processing
  - Transcoding/adaptation platform for distributors
  - License Production, or as License Sever/processor
  - etc.

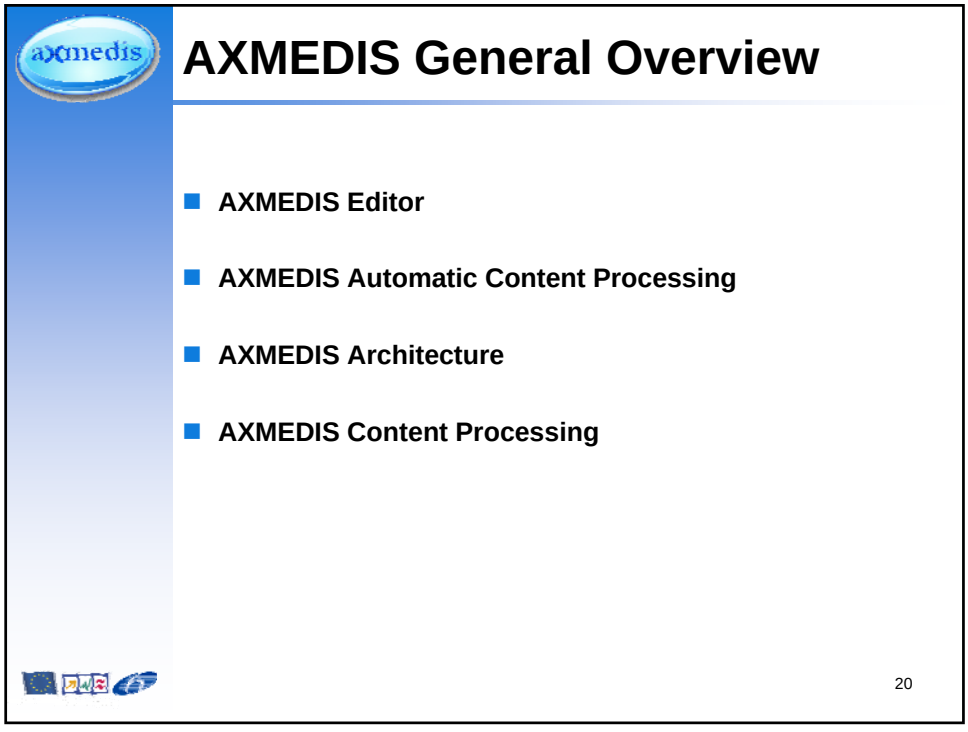
17

Summary: AXMEDIS

- **Intelligent and flexible solution**
  - Automated content processing and production
  - Efficient usage of available resources
  - Interoperability with existing infrastructure
  - Automatic provision of new service

18


Content Processing With AXMEDIS
Automation of content processing
19


AXMEDIS General Overview
AXMEDIS Editor
AXMEDIS Automatic Content Processing
AXMEDIS Architecture
AXMEDIS Content Processing
20

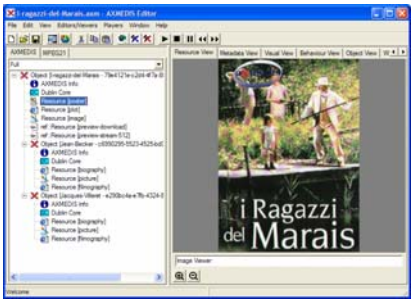
AXMEDIS Editor

- **manual production of AXMEDIS objects**
- **inspection of automatically produced objects**
- **finishing AXMEDIS objects pre-produced automatically**
- **AXMEDIS Tutorial on Content Production,**
  - ➔LT1: Tuesday, 12th (yesterday)

21

AXMEDIS Editor

- **It integrates many Editors & Viewers to handle all the aspects of the AXMEDIS Objects production**
  - Resource
  - Metadata
  - DRM
  - Protection
  - Presentation
  - Workflow
  - Annotation
  - …



- **AXMEDIS Tutorial on Content Production,**
  - ➔LT1: Tuesday, 12th (yesterday)

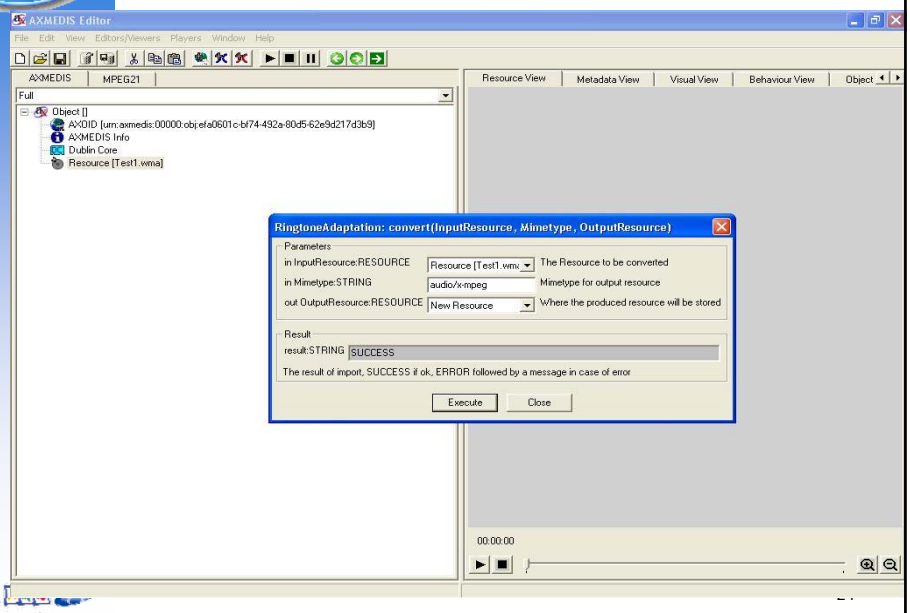
22

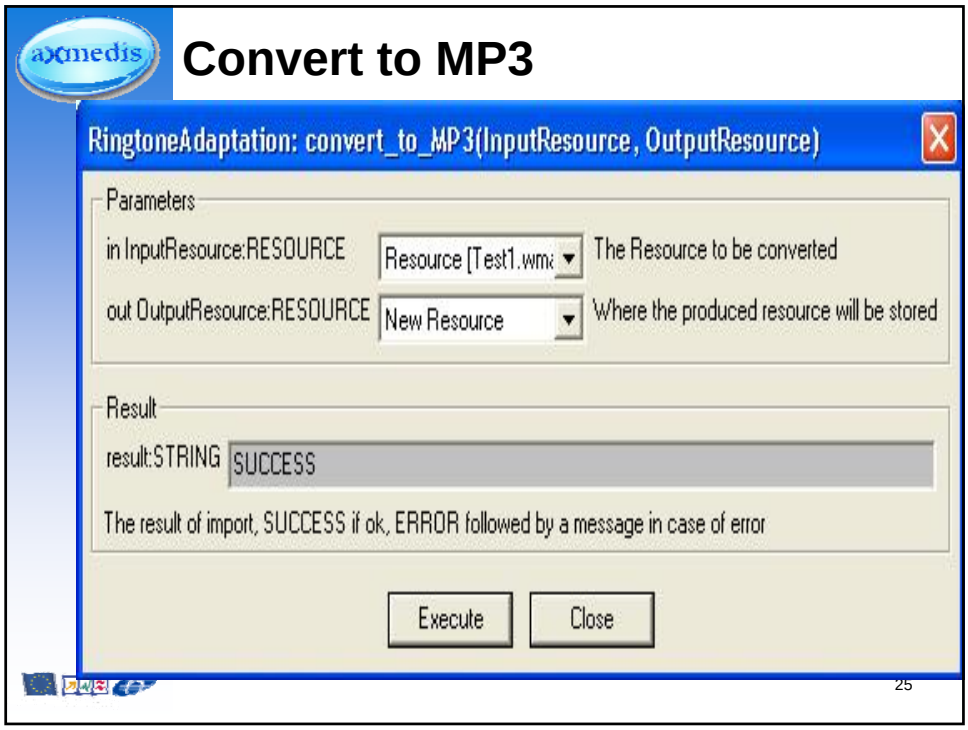
Ringtone Adaptation Plug-in

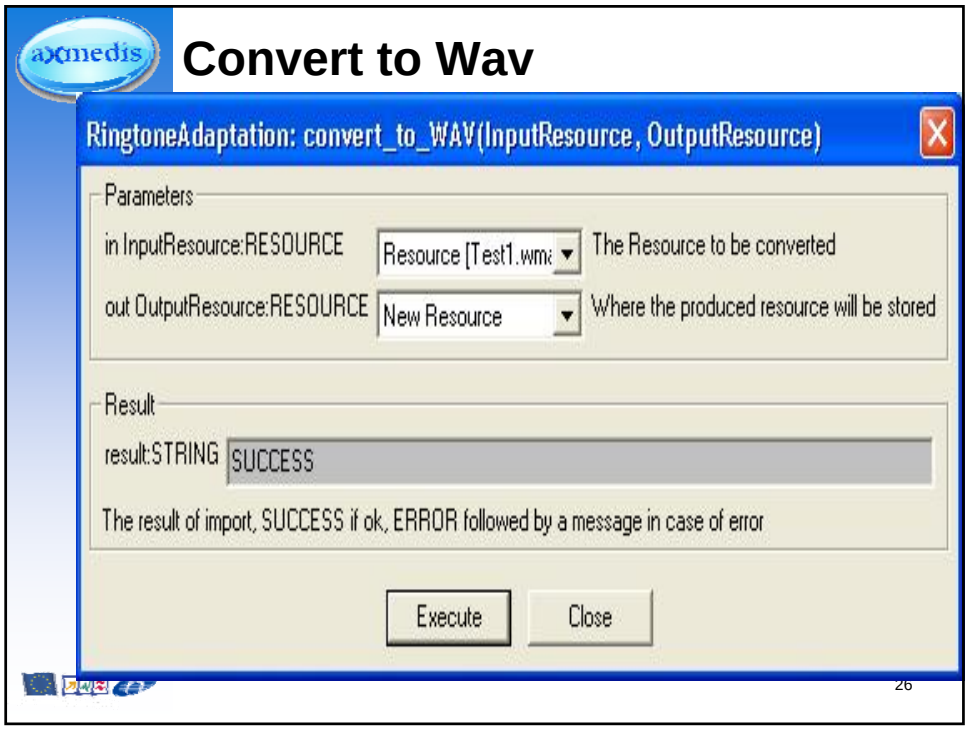
- **Adaptation of Ringtones of popular formats**
- **Transcoding on-demand.**
- **Main functions**
  - Convert to different formats
    - MP3, Wav, etc
  - Resample the Frequency, Bit rate etc
  - Clip to 5 sec, 30 sec clip etc

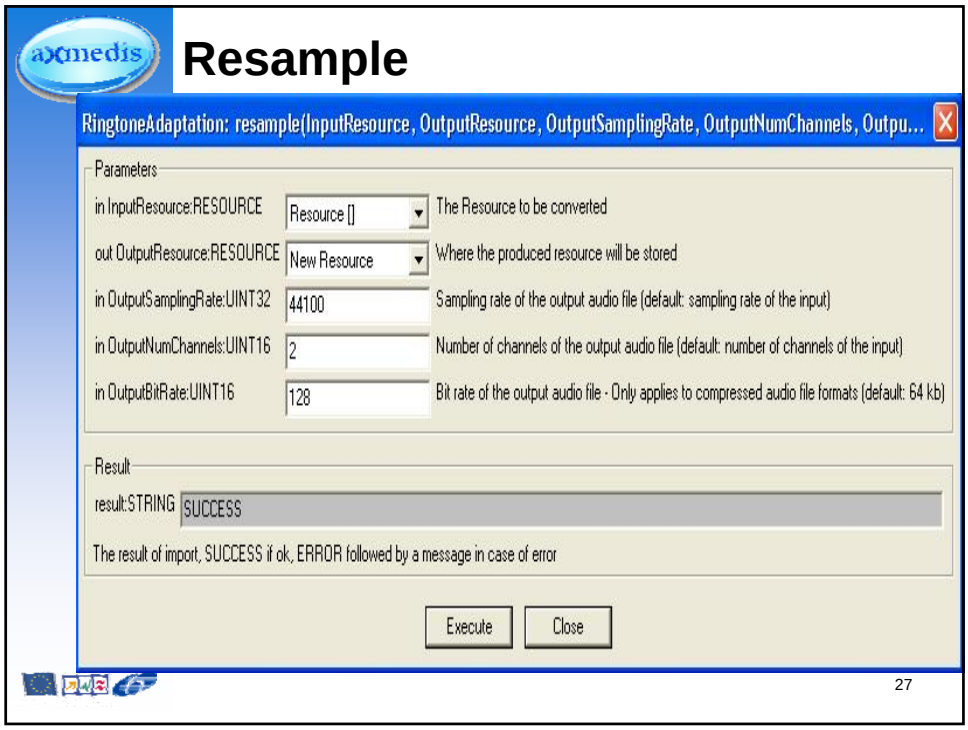
23

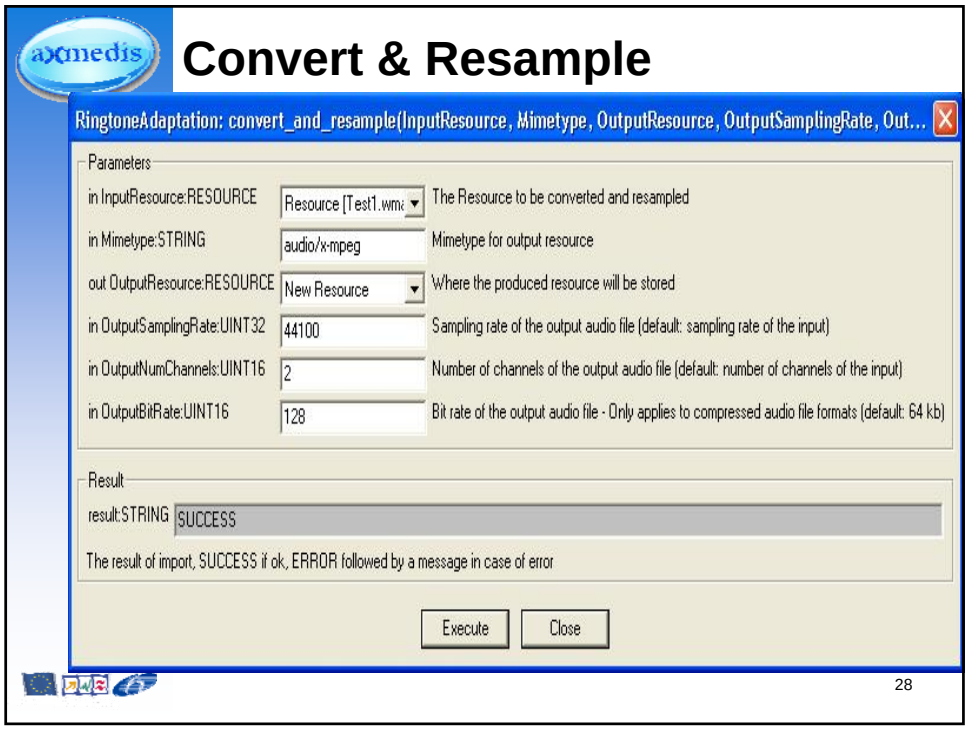
Convert



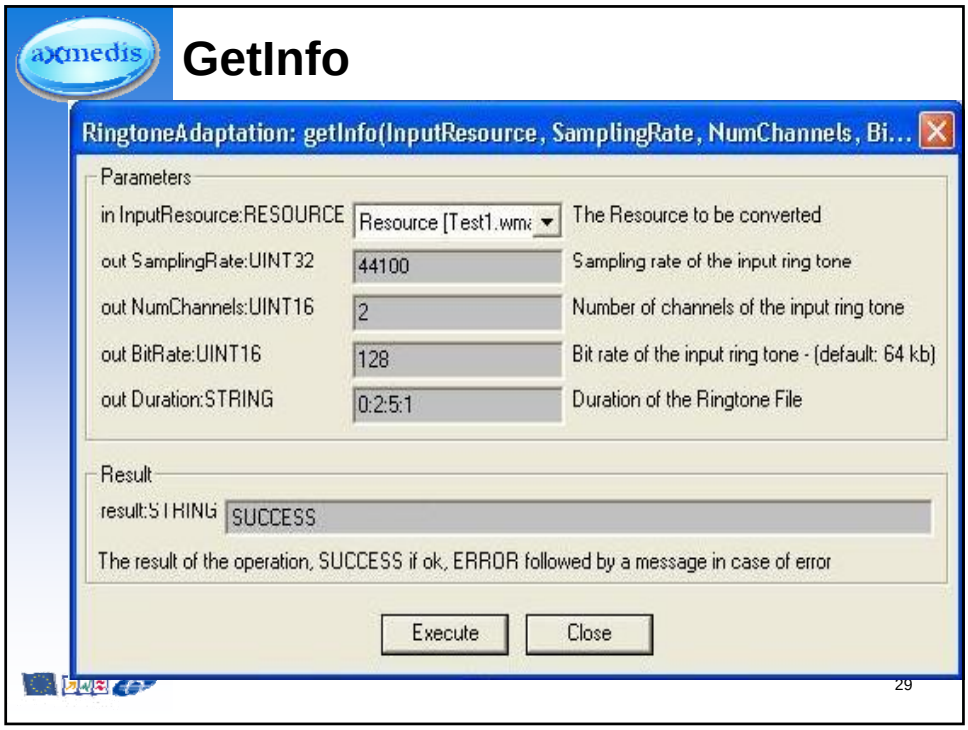

Convert to MP3
RingtoneAdaptation: convert_to_MP3(InputResource, OutputResource)
Parameters
in InputResource:RESOURCE
Resource [Test1.wma
The Resource to be converted
out OutputResource:RESOURCE
New Resource
Where the produced resource will be stored
Result
result:STRING
SUCCESS
The result of import, SUCCESS if ok, ERROR followed by a message in case of error
Execute
Close
25


Convert to Wav
RingtoneAdaptation: convert_to_WAV(InputResource, OutputResource)
Parameters
in InputResource:RESOURCE
Resource [Test1.wma
The Resource to be converted
out OutputResource:RESOURCE
New Resource
Where the produced resource will be stored
Result
result:STRING
SUCCESS
The result of import, SUCCESS if ok, ERROR followed by a message in case of error
Execute
Close
26

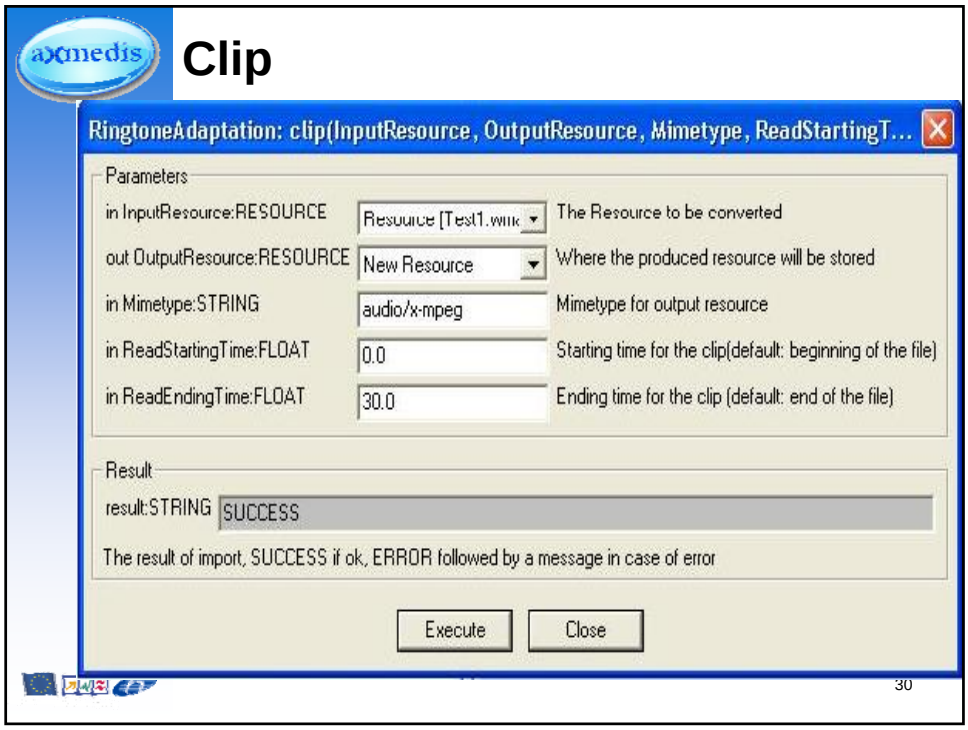

Resample
RingtoneAdaptation: resample(InputResource, OutputResource, OutputSamplingRate, OutputNumChannels, Outpu...
Parameters
in InputResource:RESOURCE
Resource []
The Resource to be converted
out OutputResource:RESOURCE
New Resource
Where the produced resource will be stored
in OutputSamplingRate:UINT32
44100
Sampling rate of the output audio file (default: sampling rate of the input)
in OutputNumChannels:UINT16
2
Number of channels of the output audio file (default: number of channels of the input)
in OutputBitRate:UINT16
128
Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)
Result
result:STRING
SUCCESS
The result of import, SUCCESS if ok, ERROR followed by a message in case of error
Execute
Close
27


Convert & Resample
RingtoneAdaptation: convert_and_resample(InputResource, Mimetype, OutputResource, OutputSamplingRate, Out...
Parameters
in InputResource:RESOURCE
Resource [Test1.wma
The Resource to be converted and resampled
in Mimetype:STRING
audio/x-mpeg
Mimetype for output resource
out OutputResource:RESOURCE
New Resource
Where the produced resource will be stored
in OutputSamplingRate:UINT32
44100
Sampling rate of the output audio file (default: sampling rate of the input)
in OutputNumChannels:UINT16
2
Number of channels of the output audio file (default: number of channels of the input)
in OutputBitRate:UINT16
128
Bit rate of the output audio file - Only applies to compressed audio file formats (default: 64 kb)
Result
result:STRING
SUCCESS
The result of import, SUCCESS if ok, ERROR followed by a message in case of error
Execute
Close
28

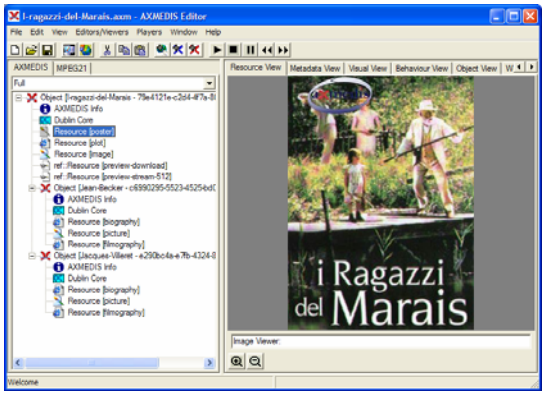
GetInfo



Clip



AXMEDIS Editor



- **AXMEDIS Tutorial on Content Production,**
  - ➔LT1: Tuesday, December, 12th, 2006 (yesterday)

31

AXMEDIS Automated Content Processing Tools

Available functionalities in the AXMEDIS Framework

32

AXMEDIS Automated Content Processing Area

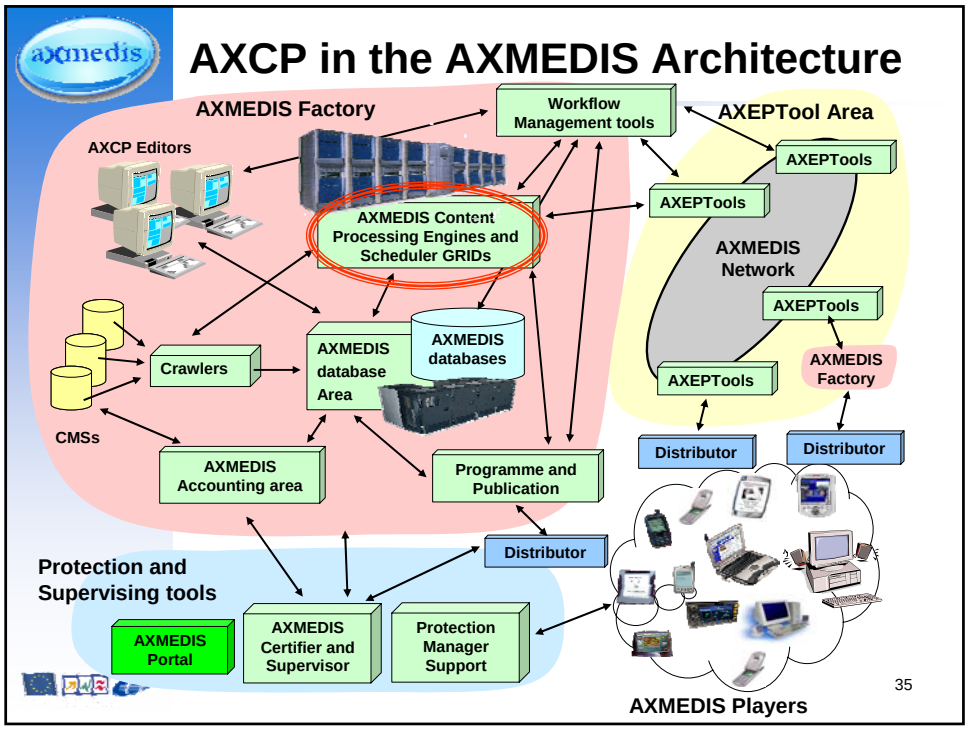
- **Properties**
- **Architecture**
- **AXMEDIS Content Processing Engine**
- **AXMEDIS GRID**

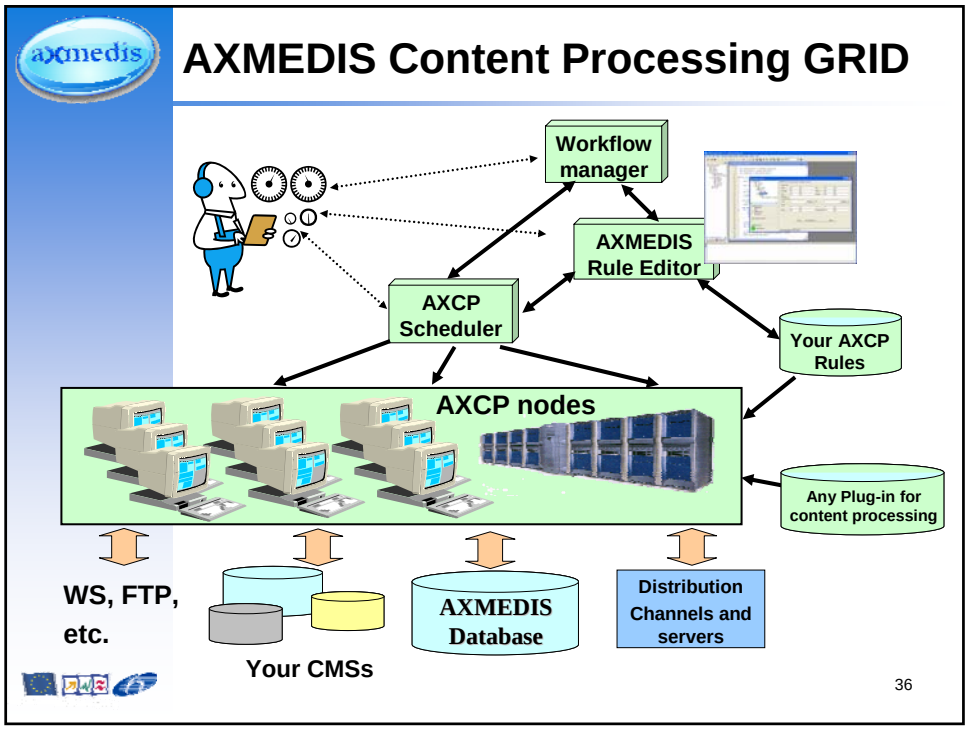
33

Automatic Content Processing

- **Content distribution and production characteristics:**
  - Content lifetime
  - Demand is time dependent
  - …
- **Example:**
  - Creating a content for a current event
  - Now: winter scenario … started to snow (… or not)
  - ➔ Manual production is (too) time consuming
- **How can the AXMEDIS Automatic Content Processing (AXCP) Area support you?**
  - Scalability
  - Extensibility
  - ➔ Essential functionality to fulfil real world's requirements

34


AXCP in the AXMEDIS Architecture
AXMEDIS Factory
Workflow Management tools
AXEPTool Area
AXCP Editors
AXEPTools
AXEPTools
AXMEDIS Content Processing Engines and Scheduler GRIDs
AXMEDIS Network
AXEPTools
AXMEDIS Factory
AXEPTools
Crawlers
AXMEDIS database Area
AXMEDIS databases
CMSs
AXMEDIS Accounting area
Programme and Publication
Distributor
Distributor
Distributor
Protection and Supervising tools
AXMEDIS Portal
AXMEDIS Certifier and Supervisor
Protection Manager Support
AXMEDIS Players
35

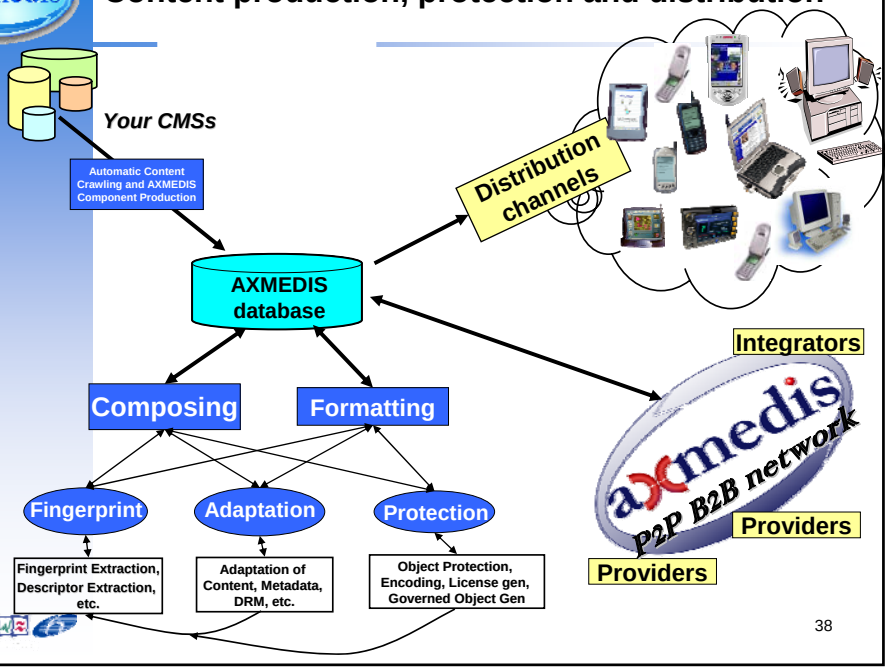

AXMEDIS Content Processing GRID
Workflow manager
AXMEDIS Rule Editor
AXCP Scheduler
Your AXCP Rules
AXCP nodes
Any Plug-in for content processing
WS, FTP, etc.
Your CMSs
AXMEDIS Database
Distribution Channels and servers
36

AXMEDIS Automated Content Processing Capabilities

- **Automated Content and Metadata Retrieval**
  - Content and Metadata Ingestion and Gathering
  - Content Query, Retrieval and Storage
- **Automated Content and Metadata Processing**
  - Content Adaptation and Transcoding
  - Metadata Generation and Mapping
  - Content Composition and Formatting
  - Content Protection and Licensing
- **Automated Content Distribution**
  - Content Publication
  - Content Distribution
  - Profile management and processing
  - Production of Content on Demand

37

Content production, protection and distribution



38

Content and Metadata Retrieval

- **Access to numerous existing DBs and CMSs for**
  - Content and Metadata Ingestion and Gathering
  - Content Query, Retrieval and Storage
  - Automatic migration of digital contents
- **Access to several different resources**
  - File Systems: Win, Linux, MAC, etc.
  - ODBC, JDBC, etc.
  - Native DB: DB2, Oracle, MS-SQL, MySQL, etc.
  - Protocols: IMAP, POP, Z39.50, etc.
  - XML databases
- **Automation of**
  - Loading and saving of AXMEDIS objects

39

Content and Metadata Processing

- **Metadata Generation**
  - Calculation of Low and High Level Descriptors
  - For AXMEDIS objects and included resources
- **Metadata Mapping**
  - Managing generic, AXInfo and DublinCore Metadata
- **Content Composition**
  - Putting together content different kinds of raw assets to create a new digital item
    → AXMEDIS Object (or MPEG-21 format)
  - Composing different AXMEDIS objects selected from the AXMEDIS database

40

Content and Metadata Processing (II)

- **Content Adaptation and Transcoding**
  - Adaptation of content (digital item adaptation, DIA)
  - For distribution via different channels to users' platforms such as i-TV, mobile, PC, etc…
- **Content Formatting**
  - Modifying digital resources according to a specific format
    - File format and properties
  - Applying a (content dependent) formatting style
    - graphic layout
    - spatial constraints
    - quality limitations
    - synchronization between audio and images, etc.

41

Content and Metadata Processing (III)

- **Content Protection**
  - Preventing un-authorized content access
  - Preventing un-authorized content manipulations
- **Content Licensing**
  - DRM and License production
  - DRM and Licence processing
  - Offline and on-demand services

42

Content Distribution

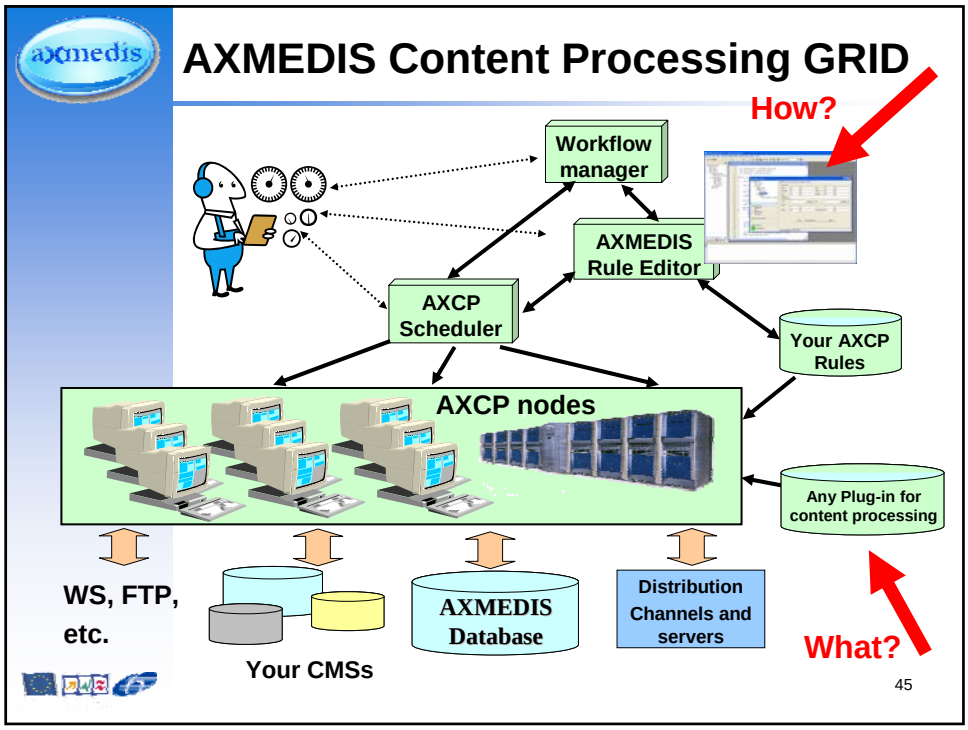
- **Content Publication**
  - metadata adaptation and mapping
  - publication of AXMEDIS objects on external channels in the B2B distribution (AXEPTools)
- **Content Distribution**
  - AXMEDIS objects distribution via available channels in the B2C scenarios
- **Profile management and processing**
  - Considering the characteristics of transmission channels and receiving device
- **Production of Content on Demand**
  - Creating new content upon customer‘s request

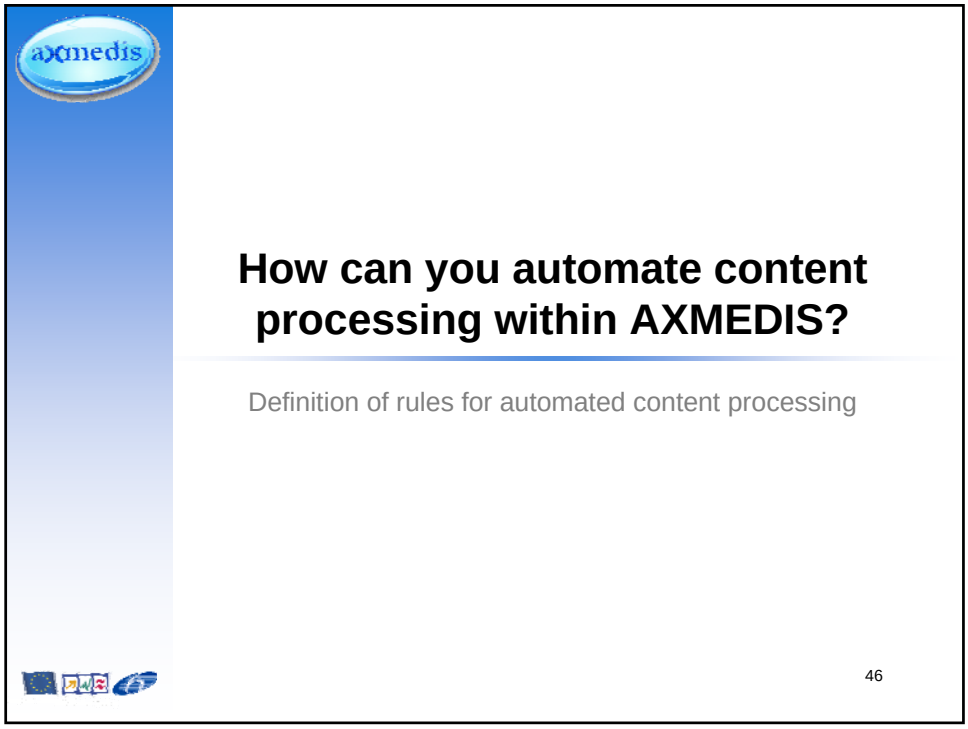
43

Extensibility

- **Provided functionality is extensible**
- **Plugin Interface**
  - To include further existing tools
  - To include future tools
- **Plugin SDK (Software Development Kit)**
  - For easy integration of your algorithms as plugins
  - Including documentation and source code examples

44


AXMEDIS Content Processing GRID
How?
Workflow manager
AXMEDIS Rule Editor
AXCP Scheduler
Your AXCP Rules
AXCP nodes
Any Plug-in for content processing
WS, FTP, etc.
Your CMSs
AXMEDIS Database
Distribution Channels and servers
What?
45


How can you automate content processing within AXMEDIS?
Definition of rules for automated content processing
46

AXMEDIS Scripting Language

- **Rules are expressed in a scripting language**
  - What is a rule?
  - What can be done with a rule?
  - How to write, test and finalize a rule?
  - How to execute a rule?

47

Rules within AXMEDIS

- **An AXCP Rule is a scripted procedure:**

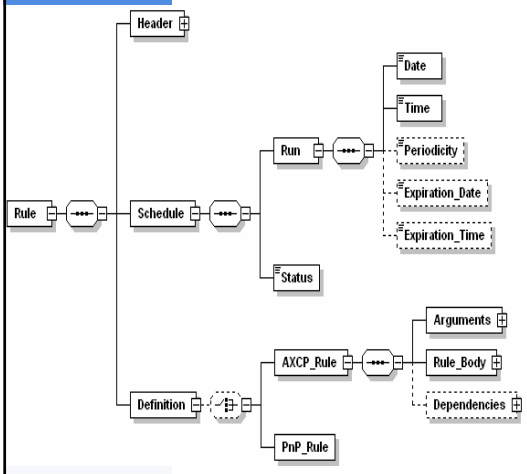
$$R = f(S_1,S_2,..,S_n,P_1,\ldots,P_m)$$

Where:

- S_i is a database Selection
  - **It is a sequence of queries to be sent to the AXMEDIS Database to retrieve objects IDs;**
- P_i is a parameter (basic type as integer, string, Boolean, etc.);
  - For example, coordinates for a formatting, size or value of object collection to be created, destination of the objects, name of the author, etc.
- f is the identifier of rule (name of rule or an ID);
- R is the result of the rule application.
  - **It can be a new AXMEDIS object, or a metadata manipulation, the license of an AXMEDIS object, a message to be returned to invokers, etc…**

48

AXCP Rule XML description



- General metadata regarding: rule name, AXRID (rule identifier), rule version, rule type, software name, version of software, date of production, time of production, author, affiliation, URL, comment, last modification and terminal ID. (*Header*)
- Temporal metadata describing conditions for firing the rule, expiration time, periodicity and the rule status ("active" or "inactive") and (*Schedule*)
- List of arguments (parameters and selections), list of dependences (required AXMEDIS plug-ins) and the rule body (the script code to run).(*Definition*)

49

AXCP Script Language

- **AXCP Rules are formally based on a script language**
- **Basis: Javascript Language (Spidermonkey by Mozilla)**
- **Extended functionality that is provided by the AXMEDIS framework for managing**
  - AXMEDIS Objects, Digital Resources
  - License, Metadata
  - Network, Database Access and Web Service
  - Profiling (user, device, context)
  - ...
- **Additional functions/utilities**
  - File System management
  - Invoking/Execution of external tools
  - MimeType management
  - Zip/Unzip Archive

➔ **Everything that can be done manually with AXEditor can be done automatically with the AXCP**

➔ **... and more ☺**

50

AXCP Tools: Rule Editor

Where and how to write, test and execute rules

51

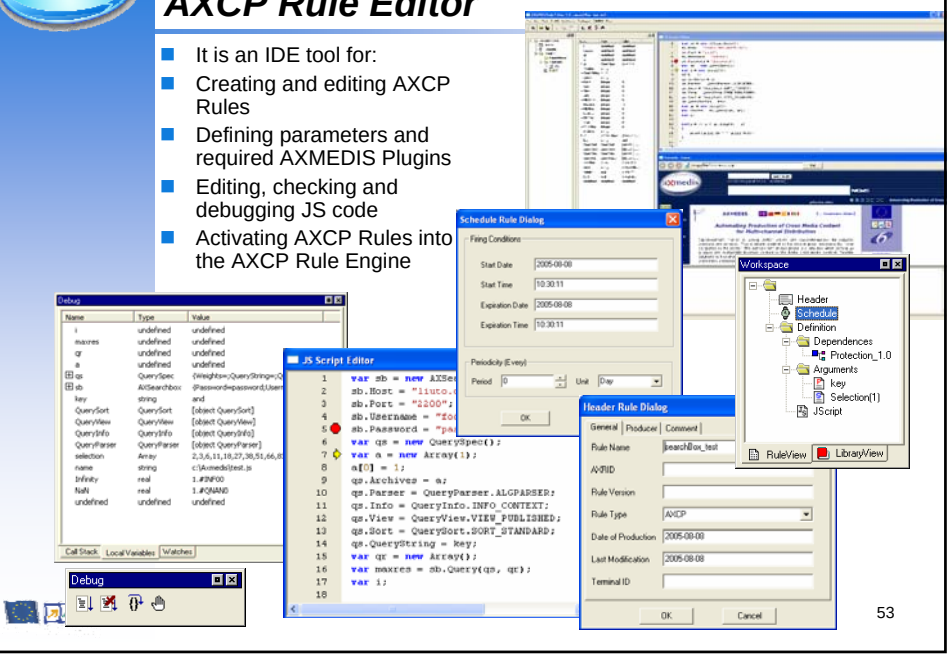
AXCP Rule Editor

- **IDE (integrated development environment) for creating, testing and managing AXCP rules**
- **Provides a set of tools and views to help the user during the editing and building of rule.**
- **Integrates the AXCP rule executor in order to provide functionalities for:**
  - executing, debugging, testing and validating
- **Provides an online help**
  - Documentation about script functions and classes of the AXCP Script Language.
  - List of AXCP Plugins and documentation about exposed functionalities

52

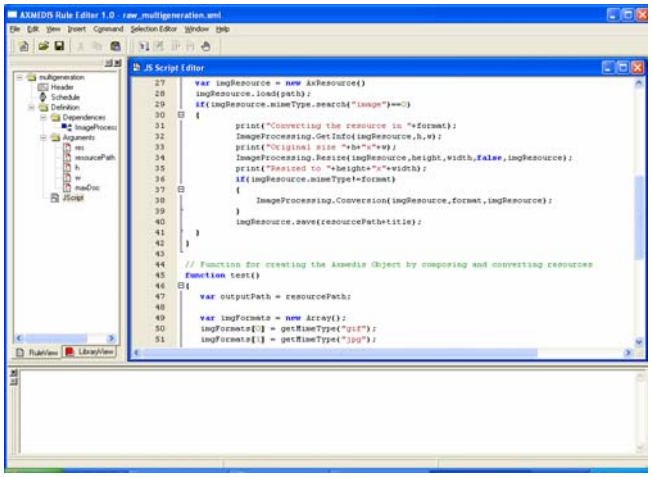
AXMEDIS Content Processing Area: *AXCP Rule Editor*

- It is an IDE tool for:
- Creating and editing AXCP Rules
- Defining parameters and required AXMEDIS Plugins
- Editing, checking and debugging JS code
- Activating AXCP Rules into the AXCP Rule Engine



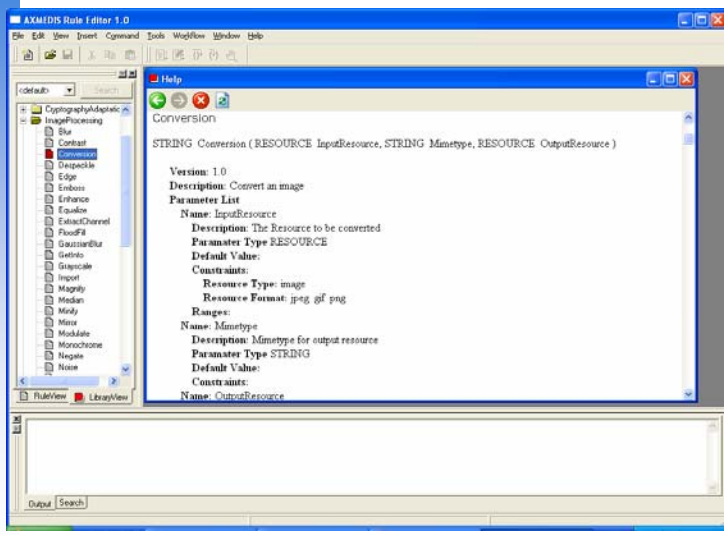
53

The Javascript Editor



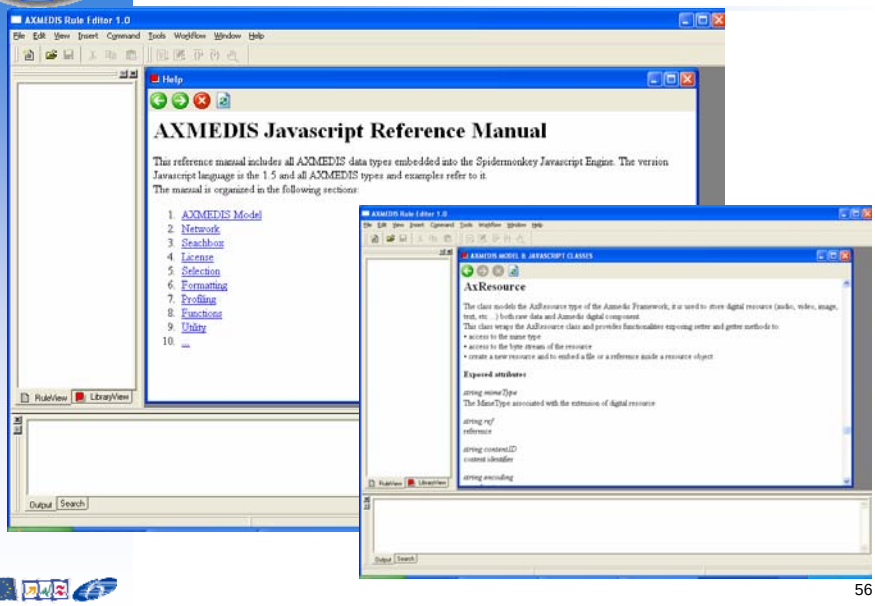
54

The Help for Plugins

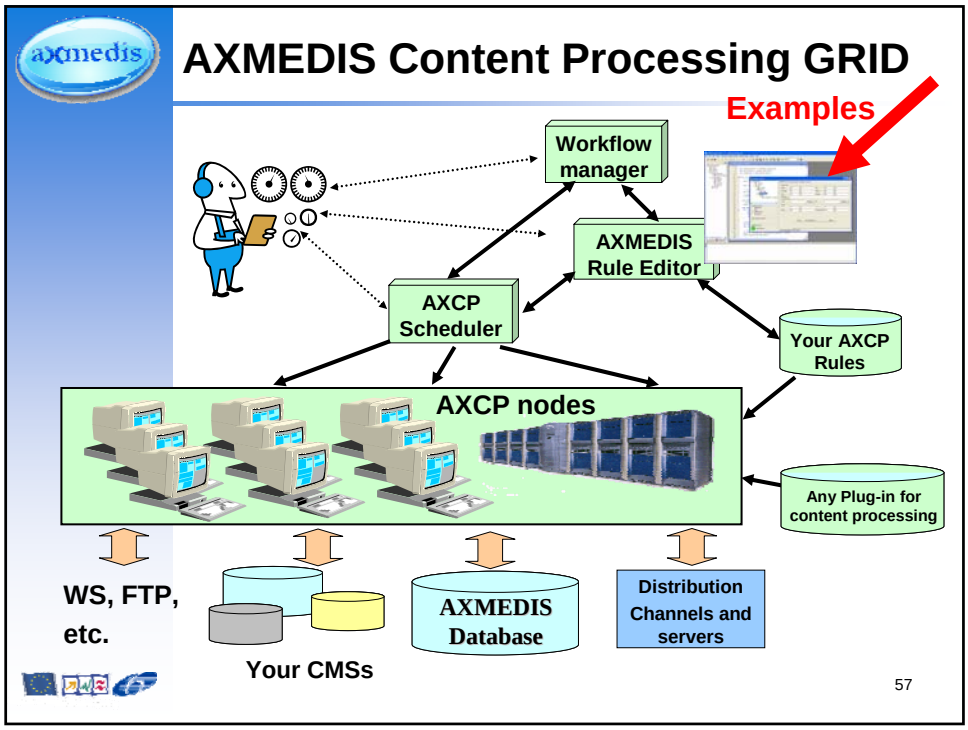


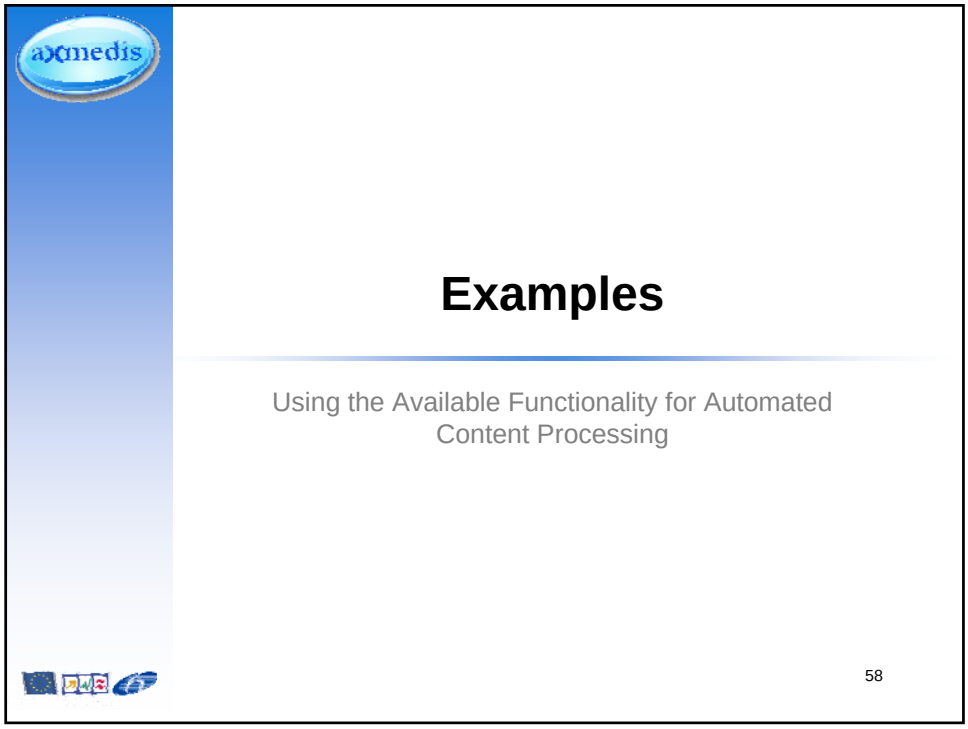
55

The Help for AXCP JS Classes



56


AXMEDIS Content Processing GRID
Examples
Workflow manager
AXMEDIS Rule Editor
AXCP Scheduler
Your AXCP Rules
AXCP nodes
Any Plug-in for content processing
WS, FTP, etc.
Your CMSs
AXMEDIS Database
Distribution Channels and servers
57


Examples
Using the Available Functionality for Automated Content Processing
58

Simple Rule without parameters

- **Example 1 of the User Manual DE5.0.1.1**
- **Goal:**
  - Create a simple script for resizing an image resource named AXMEDIS_logo.png stored in the C:\\ path and saving the new resized image on the disk.
- **Method:**
  - Writing and debugging the script
  - Comparing the original and new to see the result

59

Steps

1. **Create an empty digital resource (image)**
2. **Load the image file into the resource**
3. **Use the Image Processing plug-in for scaling the image resource**
4. **Saving the scaled image resource**

60

Script Language

```
print("Creating adapted image");

var resource = new AxResource()

resource.load("C:\\AXCPToolsDir\\resource\\axmedis_logo.png");

ImageProcessing.Resize(resource,100,100,false,resource);

print("Resized to 100 x 100");

ImageProcessing.Conversion(resource,"image/jpeg",resource);

resource.save("C:\\AXCPToolsDir\\output\\Example_100_100");

print("End of job");
```

61

Rule Parameters

- **Example 2 of the User Manual DE5.0.1.1**
- **Goals:**
  - Create a simple script for resizing an image resource stored on file system and saving the new resized image on the disk.
  - Generalize the previous script inserting some arguments in the script.
    - Location for loading and saving the image
    - Resulting image dimensions
    - Output image format
  - Obtain a rule that will remain valid and will be not modified in the future.
- **Method:**
  - Writing and debugging the script

62

Steps

Parameter definition

1. Definition of rules arguments (*input_path*, *output_path*, *width*, *height*, *out_mime_type*)

2. Create an empty resource

Parameter usage

3. Load the image file by the *input_path* argument
4. Use the Image Processing plugin for *scaling* the image
5. Use the Image Processing plugin for *conversion* the image
6. Resulting image is saved in the location specified by the *output_path* argument

63

Script Language

```
print("Creating adapted image");
var resource = new AxResource()
resource.load(inputPath+resourceFileName);
ImageProcessing.Resize(resource,width,height,false,resource);
print("Resized to "+width+" x "+height);
ImageProcessing.Conversion(resource,outFormat,resource);
resource.save(outputPath+"Example");
print("End of job");
```

64

Creating AXMEDIS Objects

- **Example 3 of the User Manual DE5.0.1.1**
- **Goal:**
  - Example 2 (Loading manipulating image resource)
  - Converting the resource into a different format
  - Create an AXMEDIS object
  - Filling metadata
  - Embedding an image resource
  - Storing on disk and/or uploading on DB
- **Method**
  - Write and debugging the script
  - Show of the produced AXMEDIS Object with AXEditor

65

Steps:

1. **Definition of rules arguments (*input_path, output_path, width, height, out_mime_type*)**
2. **Create an empty resource**
3. **Load the image file by the *input_path* argument**
4. **Use the Image Processing plugin for *scaling* the image**
5. **Use the Image Processing plugin for c*onversion* the image**

AXMEDIS objects

6. **Creating an empty AXMEDIS object**
7. **Adding meta data and the image resource**
8. **Save the AXMEDIS object**
9. **Open the AXEditor for visualisation**

66

Script Language

```
print("Creating adapted image");

var resource = new AxResource()

resource.load(inputPath+resourceFileName);

ImageProcessing.Resize(resource,width,heigth,false,resource);

print("Resized to "+width+" x "+heigth);

ImageProcessing.Conversion(resource,outFormat,resource);

// Create an empty Axmedis Object
var masterObj = new AxmedisObject();

// Add the resource in the Axmedis Object
masterObj.addContent(resource);

// Retrieve the Dublin Core metadata section from the Axmedis
// Object
var dc = masterObj.getDublinCore();
```

67

Script Language

```
// Start to fill DC elements
dc.addDCElement("creator","AXCP Tool");
dc.addDCElement("title","Demo2");
dc.addDCElement("description","Embedding a resource");

// Store the Axmedis Object on Disk
masterObj.save(outputPath+"Example.axm");

// Defining parameters for accessing to the Axmedis Database
var tempFileName ="";
var AXDBF_saverEndPoint =
    "http://192.168.0.108:8080/LoaderSaverF/save";
var AXDBF_user = "test";
var AXDBF_passwd = "test";
var AXDBF_usingftp = true;
var AXDBF_externalurl = "ftp://192.168.0.108";
var AXDBF_internalurl = "";
var AXDBF_lockEndPoint =
    "http://192.168.0.108:8080/LockUnlockWSF/lockunlock";

// Storing on Axmedis Database
masterObj.uploadToDB(AXDBF_saverEndPoint,AXDBF_user,
    AXDBF_passwd,AXDBF_usingftp,
    AXDBF_externalurl,tempFileName,
    AXDBF_internalurl,AXDBF_lockEndPoint);

print("End of job");
```

68

"Full Round" AXMEDIS Automated Object Production

- **Steps:**
  - Example 3 (previous example: AXObject creation)
    - Filling metadata
    - Embedding a resource
    - Adapting the resource into a different format
  - Registering Object
  - Protecting Object
  - Creating and storing licenses
  - Storing on disk and/or uploading on DB
- **Method**
  - Write and debugging the script
  - Show of the produced AXMEDIS Object with AXEditor

69

Steps:

1. **Setting of the license related parameters**
2. **Create an empty resource and AXMEDIS object**
3. **Load the resource**
4. **Adding meta data and the resource in the AXMEDIS object**

DRM operations

5. **Save the AXMEDIS object on disk and AXMEDIS database**
6. **Creating licences for distribution**
7. **Creating license for the end user**
8. **Fruition – Using the AXMEDIS object**

70

Script Language (I)

```
// IDs Definition
// Id of the Creator
var AXCID = "URN:AXMEDIS:00001:BUS:00000000-0000-0000-0000-
   000000000000";

// Id of the Distributor
var AXDID = "URN:AXMEDIS:00001:BUS:3A2F35F1-5E91-314C-A51A-
   D07D75C216C7";

// Id of the Final User
var AXUID = "URN:AXMEDIS:00001:USR:00000000-0002-0000-0000-
   000000000000";

// AXCS Client configuration for object definitive AXOID
// request & registration
var AXCS_WS_URI =
   "http://192.168.0.108:8080/AXCSObjectRegistrator/services/O
   bjectRegistrator";
var AXCS_USER = "mario";
var AXCS_PWD = "xxxxx";

//PMS Client configuration for posting licenses
var PMSClient_Endpoint="https://192.168.0.108:8502/PMS/";
var PMSClient_Certificate="client.pem";
var PMSClient_CertificatePsw="axmedis";
var PMSClient_CA="cacert.pem";
```

71

Script Language (II)

```
//AXDB configuration
var AXDBF_loaderEndPoint =
   "http://192.168.0.108:8080/LoaderSaverF/load";

var AXDBF_saverEndPoint =
   "http://192.168.0.108:8080/LoaderSaverF/save";
var AXDBF_user = "test";
var AXDBF_passwd = "test";
var AXDBF_usingftp = true;
var AXDBF_externalurl = "ftp://192.168.0.108";
var AXDBF_internalurl = "";
var AXDBF_lockEndPoint =
   "http://192.168.0.108:8080/LockUnlockWSF/lockunlock";

// Temporary Directory
var tempDir = outputPath;
```

72

Script Language (III)

```
function createAxmedisObject()
{
  // Section 1: Axmedis object creation

  // Create an empty Resource
  var resource = new AxResource();

  // Load the raw asset in the resource
  resource.load(inputPath+resourceFileName);

  ImageProcessing.Resize(resource,width,heigth,false,resource);
  print("Resized to "+width+" x "+heigth);
  ImageProcessing.Conversion(resource,outFormat,resource);

  // Create an empty Axmedis Object
  var masterObj = new AxmedisObject();

  //Add the resource in the Axmedis Object
  masterObj.addContent(resource);
```

73

Script Language (IV)

```
  // Retrieve the Dublin Core metadata section from the Axmedis
  // Object
  var dc = masterObj.getDublinCore();

  //Start to fill DC elements
  dc.addDCElement("creator","AXCP Tool");
  dc.addDCElement("title","Demo 3");
  dc.addDCElement("description","Protection and Licenses");

  // Retrieve the AxInfo metadata section from the Axmedis Object
  var axInfo = masterObj.getAxInfo();

  // Set the Distributor ID (AXDID)
  axInfo.DistributoAXDID = AXDID;

  //Store a copy on Disk
  masterObj.save(tempDir+"\\step1.axm");
```

74

Script Language (V)

```
//Section 2: Protection and Registration of the Axmedis
// Object

// Request of a definitive AXOID to the AXCS
if(!masterObj.obtainDefinitiveAXOID(AXCS_WS_URI,AXCS_USER,AXCS_PWD))
{
  var error = "Definitive AXOID request failure: "+AXOID;
  print(error);
  return false;
}
print(masterObj.AXOID);

var AXOID = masterObj.AXOID;

//Store a copy on Disk
masterObj.save(tempDir+"\\step2.axm");
```

75

Script Language (VI)

```
//Protecting Axmedis Object
var params = new Array();
var order = 1;
var toolID = "urn:axmedis:ipmp:tool:id:0003";
params[0]=20;
masterObj.setProtectionInfo(toolID,order,params);

//Store a copy on Disk
masterObj.save(tempDir+"\\step3.axm");

// Registering object to the AXCS
print("Registering object to AXCS");
masterObj.registerToAXCS(AXCS_WS_URI,AXCS_USER,AXCS_PWD);

masterObj.save(tempDir+"\\pub-object.axm");
```

76

Script Language (VII)

```
// Upload the Axmedis Object on Axmedis Database
var tempFileName ="";
if(!masterObj.uploadToDB(AXDBF_saverEndPoint,AXDBF_user,
 AXDBF_passwd,AXDBF_usingftp,AXDBF_externalurl,tempFileName,
 AXDBF_internalurl,AXDBF_lockEndPoint))
{
  var error = "Upload request failure: "+AXOID;
  print(error);
  return false;
}

//Section 3:Creating Mother License
// This license allows the distributor to generate license
// for opening the Axmedis object by a registered end user (grant)
// The issuer is the Creator and the Principal is the Distributor
// the righ is to play; no payment or limit are imposed

//creating the grant (mx:play for AXOID)
var endGrantGroup = new GrantGroup();
var endGrant = new Grant();
```

77

Script Language (VIII)

```
// Define the Right
var endRight = new Right();
endRight.setRight("mx:play");

// Setting the resource: the current Axmedis Object
var endRes = new Resource();
endRes.resourceId = AXOID;
endRes.diType = 1;

// Applying Resource and Right to the grant
endGrant.setResource(endRes);
endGrant.setRight(endRight);

// Adding the grant to the Grant Group
endGrantGroup.addGrant(endGrant);
```

78

Script Language (IX)

```
//Creating the issue grant (mx:issue for GrantGroup)
// Defining the mother License
var motherLicense = new License();

// Define the cretor of object
var creator = new Issuer();

// Create the distribution Grant Group
var distGrantGroup = new GrantGroup();

// Create the distributor
var distributor = new Principal();

// Create the Resource for the mother license
var distRes = new Resource();
distRes.setGrantGroup(endGrantGroup);
```

79

Script Language (X)

```
var distGrant = new Grant();

// Set the right type for distribution
var distRight = new Right();
distRight.setRight("r:issue");

// Set distributor ID
distributor.setName(AXDID);
distGrant.setPrincipal(distributor);
distGrant.setResource(distRes);
distGrant.setRight(distRight);

distGrantGroup.addGrant(distGrant);

creator.setIssuer(AXCID);
motherLicense.setGrantGroup(distGrantGroup);
motherLicense.setIssuer(creator);
```

80

Script Language (XI)

```
// Retrieve the xml description of Mother license
var mxmllic = motherLicense.getXMLLicense();

// Store the XML on disk
writeToFile(tempDir+"\\MLic.xml", mxmllic);

// Posting the mother license
print("Start Posting Mother License on PMS");

// Creating the Client for posting license into the PMS
var pms = new PMSClient(PMSClient_Endpoint,
 PMSClient_Certificate,PMSClient_CertificatePsw,
 PMSClient_CA);

// Sending license to PMS
var res = pms.sendLicense(mxmllic);
print("License posted as "+res+" in the PMS Server");
```

81

Script Language (XII)

```
// Section 4: Creating the enduser license (Child License)
// This license allows the end user associated with AXUID opening
// the Axmedis Object.
// The Issuer is the distributor and Principal is the End User. The
// grant including the right is what has been decided by creator in
// the Mother License
var enduserLicense = new License();
var dist_issuer = new Issuer();

var enduser = new Principal();
enduser.setName(AXUID);
endGrant.setPrincipal(enduser);

dist_issuer.setIssuer(AXDID);

// Set the Grant Group with the right (mx:play)
enduserLicense.setGrantGroup(endGrantGroup);
enduserLicense.setIssuer(dist_issuer);
```

82

Script Language

```
  // Retrieve the xml description of the end use license
  var exmllic = enduserLicense.getXMLLicense();

  // Store the XML on disk
  writeToFile(tempDir+"\\ELic.xml", exmllic);

  //Using the same PMSClient for sending the end user license
  var res = pms.sendLicense(exmllic);
  print("License posted as "+res+" in the PMS Server");

  return true;
}

var result = createAxmedisObject();
```

83

Automated Content Processing

- **What you have seen**
  - What is a rule and what can be done with it?
  - Writing a rule in the AXCP Rule Editor for automated content processing?
  - How to execute and testing a rule in the AXCP Rule Editor?
- **Next:**
  - Extended functionalities for content processing

84

Which further functionalities are available?

Extended functionalities provided by the AXMEDIS Framework for Automated Content Processing

85

Examples: Extended Functionalities

- Content Description and Meta Data Creation
- Meta Data Mapping
- Adaptation and Transcoding
- Content Authentication

86

Content Description and Meta Data Creation

- **Automatic extraction/calculation of**
  - Low Level Descriptors (LLDs)
  - Perceptual Hashes/Content Fingerprints
  - High Level Descriptors (LLDs)
- **List of available functionalities**
  - Audio - LLDs
    - MPEG-7 AudioWaveform
    - MPEG-7 AudioPower
    - MPEG-7 AudioSpectrumEnvelope
    - MPEG-7 AudioSpectrumCentroid
    - MPEG-7 AudioSpectrumSpread
    - MPEG-7 AudioSpectrumFlatness
    - Mel-Frequency Cepstrum Coefficients

87

Content Description and Meta Data Creation (II)

- **Continued list of available functionalities**
  - Audio – HLDs
    - Rhythm
    - Audio File Segmentation
    - Music Genre
  - Video
    - Basic Integration of MPEG-7 reference implementation
  - Text
    - Language Guesser
    - Keyword Extraction

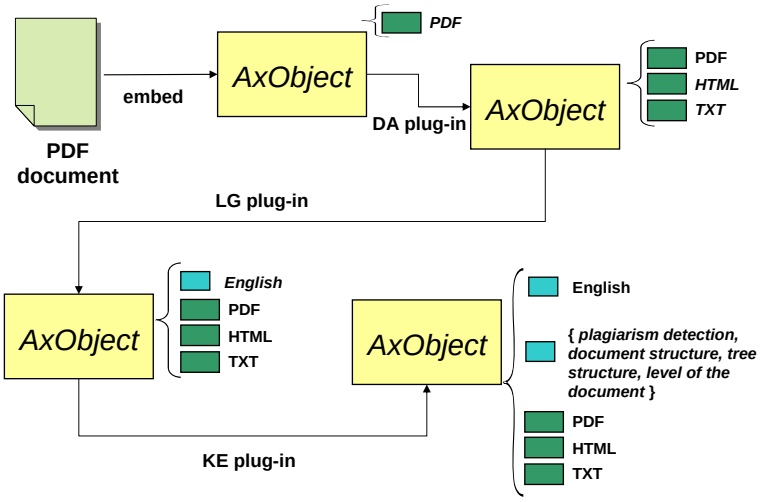
88

Demonstration: Text processing

- **Document Adaptation (DA)**
  - transcoding between various text formats (PDF, HTML, Postscript, RTF, plaint text)
- **Language guesser (LG)**
  - Retrieving the main language from a plain text document
- **Keyword extractor (KE)**
  - Extraction of mono-term and multi-term keywords from an plain text (English, Italian, German, Spanish and French, only English is implemented so far)
- **Automatically extracted descriptors can be used to fill AXMEDIS objects metadata, allowing better search results.**

89

Text Processing



90

Steps

1. Loading resource
2. Transcoding to HTML
3. Embedding the HTML resource in the object
4. Transcoding to plaint text
5. Embedding the plain text resource in the object

91

Text Processing Script (I)

```
// path where the data is stored
var path="C:\\axmedis\\";

// loading raw resource – PDF
var pdfRes = new AxResource();
pdfRes.load(path + "Zini-et-al_AXMEDIS2006.pdf");

// embedding the resource in a new objectvar obj = new
   AxmedisObject();
obj.addContent(pdfRes);

// Document Adaptation
// 1. Transcoding to HTML
var htmlRes = new AxResource();htmlRes.contentID =
   "HTMLResource";TextDocsAdaptation.DocumentConversion(pdfRes,"te
   xt/html",htmlRes);

// 2. Embedding the HTML resource in the object
obj.addContent(htmlRes);
var textRes = new AxResource();
textRes.contentID = "PlainTextResource";

// 3. Transcoding to plaint text
TextDocsAdaptation.DocumentConversion(pdfRes,"text/plain",textRes)
   ;

// 4. Embedding the plain text resource in the object
obj.addContent(textRes);
```

92

Text Processing Script (II)

```
// Language Guessing
// 5. Guessing resource language
var language = new Array(1);
language[0]="";
LanguageGuesser.LanguageGuesser(textRes,language);

// 6. Adding language to Dublin Core metadata
var dublinCore = obj.getDublinCore();
dublinCore.addDCElement("language", language[0]);

// Keyword Extraction
// 7. Retrieving keyowrds
var keywords = new Array(1);
keywords[0]="";
TextDescriptors.KWFromComparisons(textRes,4,false,keywords);
var commaSepKeywords = keywords[0].replace(/[\n\r]+/g, ", ");
commaSepKeywords = commaSepKeywords.replace(/_{3}/g," ");

// 8. Adding keywords to Dublin Core metadata
dublinCore.addDCElement("subject", commaSepKeywords);

// 9. saving object to file
obj.save(path + "textproc.axm");
true;
```

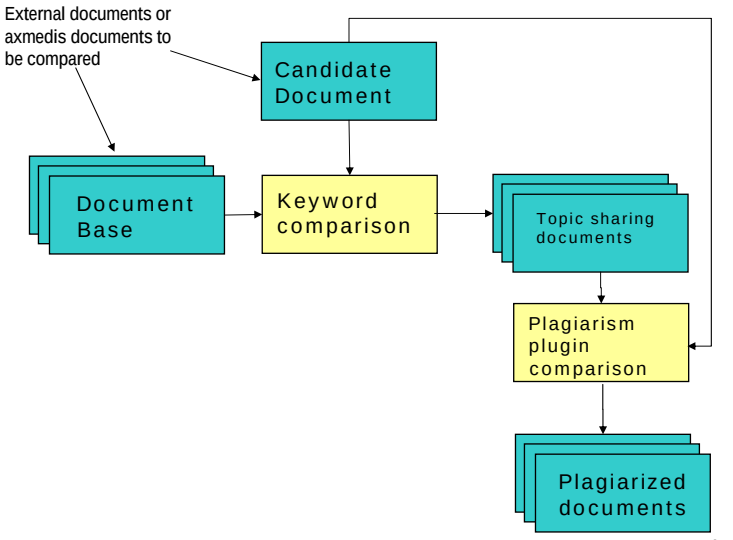
93

Plagiarism detection plug-in

- Detection of plagiarism behaviour: plagiarism between two textual documents
- Basic model: a set of modifications (insertion, deletion, substitution) performed
- Return value: normalised similarity value
- Can be combined with other tools, e.g. keyword extractor

94

Plagiarism plugin usage scenarios



95

Steps

1. Defining the reference document
2. Defining the documents to compare with
3. Comparing the documents

96

Plagiarism Detection Script

```
var path="c:\\axmedis\\"
var originalName = "original.txt";
var txtFilesArray = getAllFiles(path,"*.txt");
var threshold = 0.3;var i=0;var resources = new
   Array(txtFilesArray.length);

for (i in txtFilesArray) {

   var originalRes = new AxResource();
   originalRes.load(path + "original.txt");

   resources[i] = new AxResource();
   resources[i].load(txtFilesArray[i]);

   var score = Plagiarism.Compare(originalRes, resources[i]);

   if (score < threshold) {
     print(originalName + " <--> " + txtFilesArray[i].substring (
       txtFilesArray[i].lastIndexOf("\\") + 1 ,
       txtFilesArray[i].length ) + " --> Test OK");
   } else {
     print(originalName + " <--> " + txtFilesArray[i].substring (
       txtFilesArray[i].lastIndexOf("\\") + 1 ,
   txtFilesArray[i].length ) + " --> WARNING!!!
       Score is: " + score);
   }
}
true;
```

97

Adaptation and Transcoding

- **Transcoding to and from numerous formats**
- **Supported media types:**
  - Audio
  - Images
  - Text
  - Video
- **Metadata Adaptation**
- **Rights Information Adaptation**
  - Licenses: MPEG-21 ←→ OMA
- **Dynamic Transcoding**
  - ➔ See: Workflow Tutorial, today, starting 2pm

98

Audio Adaptation

- **Goal:**
  - Genre detection (only possible with WAVs)
- **Steps**
  - Loading the input resource (MP3 format)
  - Converting the input resource to WAV
  - Apply a genre detection
  - Output: genre

99

Adaptation Script

```
// two resources are needed for the transcoding
var res = new AxResource();
var res2 = new AxResource();

// loading the input resource (MP3)
res.load("C:\\axmedis\\uforobot.mp3");

// converting the input resource to WAV
RingtoneAdaptation.convert_to_WAV(res,res2);

// apply a genre detection
var genre = AudioDescriptor.MusicGenreRecognition(res2);

// output: genre
print(genre);
```

100

Meta Data Mapping

- **Manages**
  - Generic, AXInfo and DublinCore Metadata
- **Provides functionalities for**
  - Extracting Metadata from an AXMEDIS Object
  - Adapting Metadata
    - Loading adaptation style sheet
    - Transforming extracted metadata
  - Embedding the transformed metadata into an AXMEDIS Object

101

Steps

1. **Open AXMEDIS object**
2. **Initialize the Metadata Mapper**
3. **Extract the Metadata from the AXMEDIS object**
4. **Load the saved XSLT file from the disk**
5. **Transform the metadata producing the new Metadata**
6. **Embed the new metadata into an AXMEDIS Object**
7. **Save the AXMEDIS Object**

102

Metadata Mapper Script

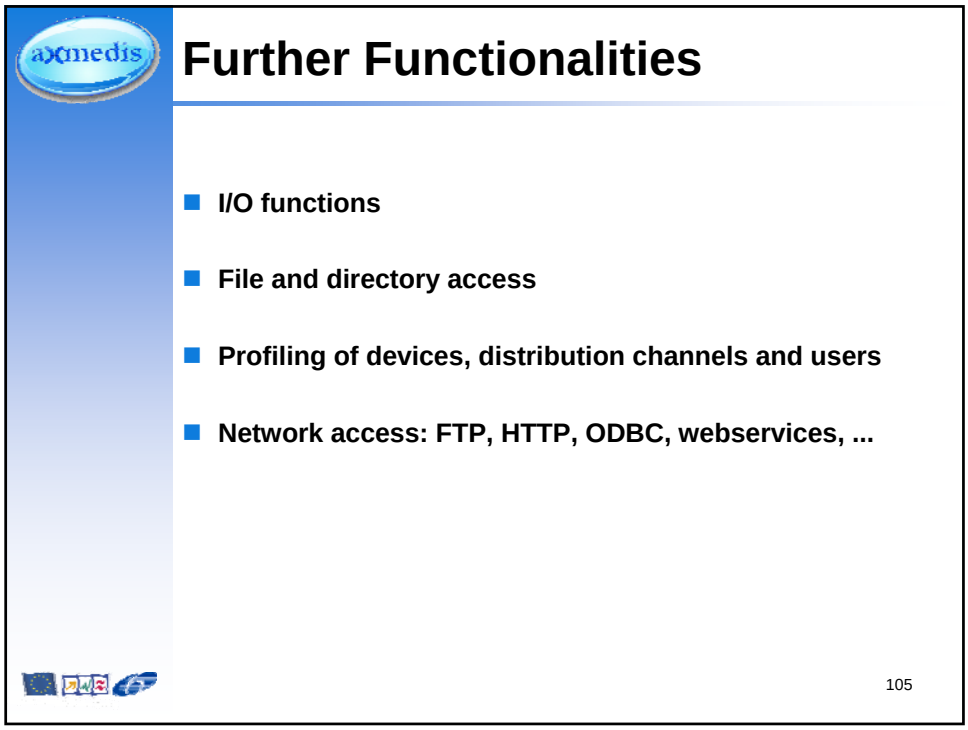
```
function main()
{
    // Open AXMEDIS object
    var axom = new AxmedisObject("C:\\axmedis\\tiscali-mediaclub-dottor-
    mabuse.axm");

    // Initalise the Metadata Mapper
    var mdm = new MetaDataMapper();

    // Extract the Metadata from the AXMEDIS object
    var extractedString = mdm.ExtractMetadata(axom);

    // Load the saved xslt file
    mdm.LoadXSL("C:\\axmedis\\tiscali-test-style2.xsl");

    // Transform the metadata returning the new metadata string
    var out = mdm.Transform(extractedString);

    // Embed the new metadata into an axmedis object
    mdm.EmbedMetadata(axom, out);

    // Save AXMEDIS object
    axom.save("C:\\axmedis\\tiscali-mediaclub-dottor-mabuse-adapted.axm")
        return 0;
}

// entry point
main();
```

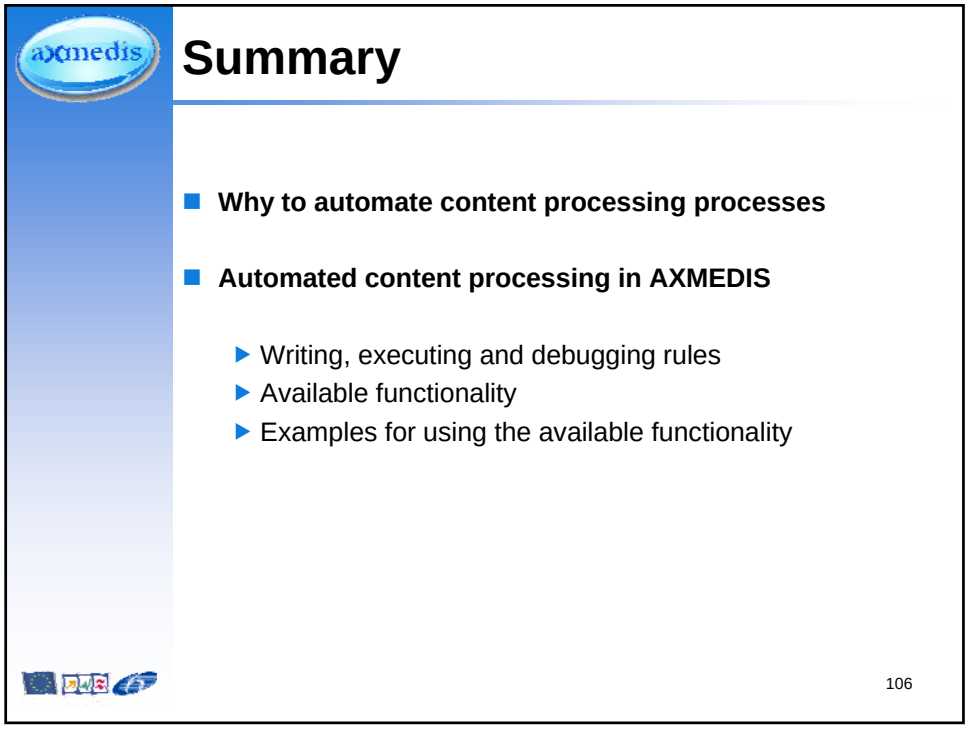
103

Content Authentication

- **Cryptographic Hash Functions**
  - All kind of content
- **Perceptual Hash Functions**
  - Audio
  - Images
  - Video

104


Further Functionalities
I/O functions
File and directory access
Profiling of devices, distribution channels and users
Network access: FTP, HTTP, ODBC, webservices, ...
105

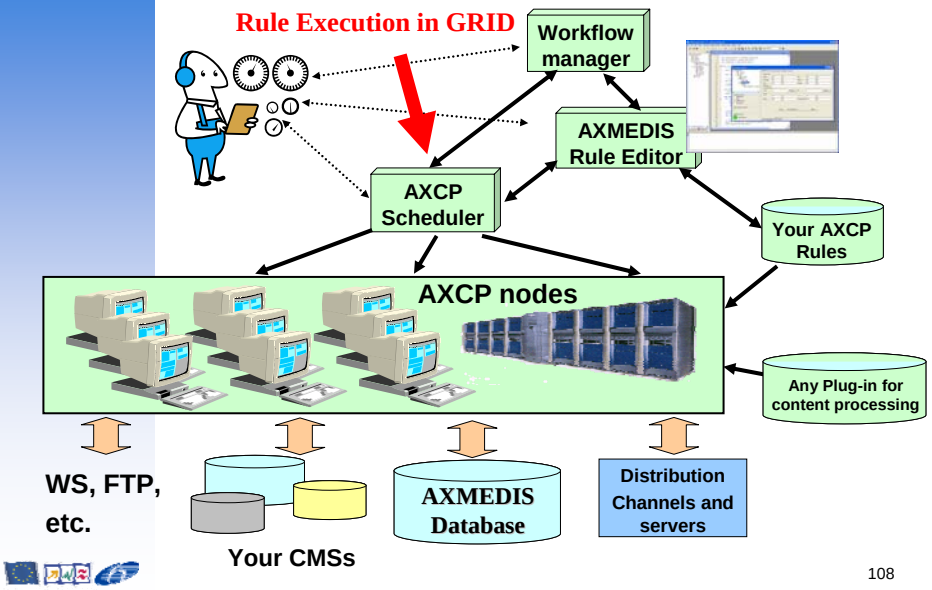

Summary
Why to automate content processing processes
Automated content processing in AXMEDIS
Writing, executing and debugging rules
Available functionality
Examples for using the available functionality
106

AXMEDIS Tutorial on Content Processing - Part II

- AXCP Rule Scheduler
- AXCP GRID
- Complex Scenarios
- Summary and Conclusion
- Discussion, Questions and Answers

107

AXMEDIS Content Processing GRID



108

AXCP Rule Engine

Managing the Rule Execution in the AXMEDIS Grid

109

AXCP Rule Engine

- **Consists of**
  - Rule Scheduler (Server Side)
  - Rule Remote Executor (Client Side): AXCP GRID Node
- **Rule Scheduler: Internal Scheduler and Dispatcher for**
  - rule installation
  - rule firing
  - rule executor discovering and management
  - rules scheduling and dispatching according to the executor profile
  - communication with the AXMEDIS environment (workflow)
  - ...
- **Rule Remote Executor**
  - Consisting of the same JavaScript engine

110

AXCP Scheduler: Functionalities

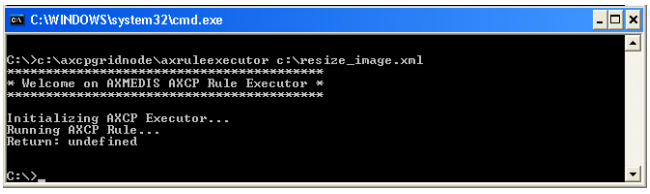
- Install & Activate a Rule in AXCP Grid
- Run a Rule in AXCP Grid
- Deactivate a Rule
- Suspend a Rule
- Pause a running Rule
- Resume a previously suspended Rule
- Kill a Rule immediately
- Remove a Rule from the AXCP Grid
- Determine the Status for any Rule in the Grid
- Retrieve the Logs for any Rule in the Grid
- Retrieve the List of Rules in the Grid
- Retrieve a particular Rule in the Grid

➔ manually or with a remote client (e.g. workflow)

111

Remote Rule Executor

- **Application**
  - Standalone



  - Node of the GRID
- **Properties**
  - CPU Monitoring
  - CPU Workload Constraints
  - Communication with the Scheduler (GRID Node)

112

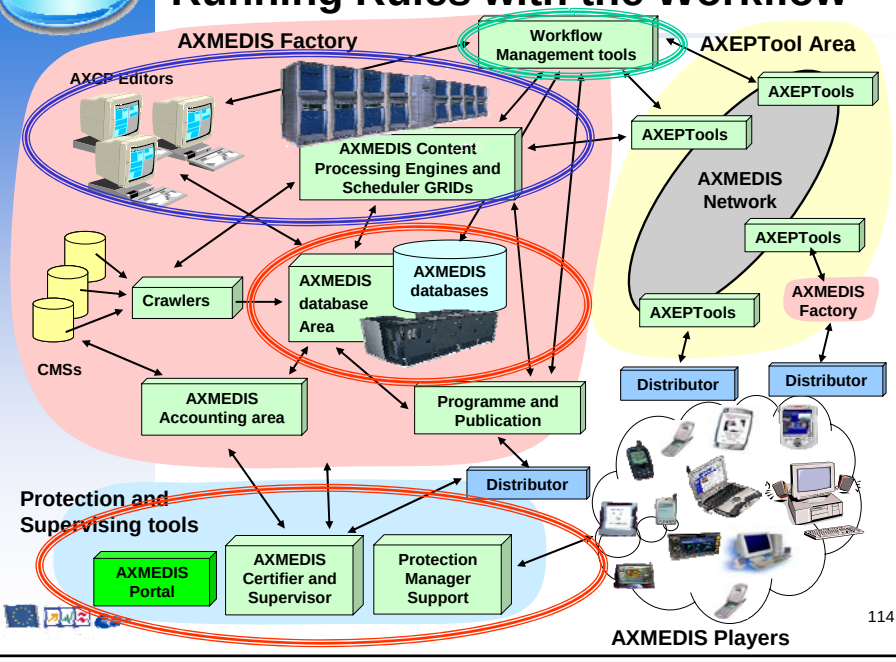
AXCP GRID Node

- **Workload setting (see configuration.xml)**

```
<Module category="" id="WORKLOAD_SETTINGS">
    <Parameter name="MON" type="string">
        0;0;0;0;0;0;30;50;30;30;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="TUE" type="string">
        0;0;0;0;0;0;30;30;30;30;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="WED"  type="string">
        0;0;0;0;0;0;30;30;50;60;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="THU" type="string">
        0;0;0;0;0;0;60;60;60;60;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="FRI" type="string">
        0;0;0;0;0;0;50;50;50;50;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="SAT" type="string">
        0;0;0;0;0;0;50;50;50;50;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
    <Parameter name="SUN" type="string">
        0;0;0;0;0;0;50;50;30;70;20;30;45;50;20;30;45;50;30;30;0;0;0;0
    </Parameter>
</Module>
```

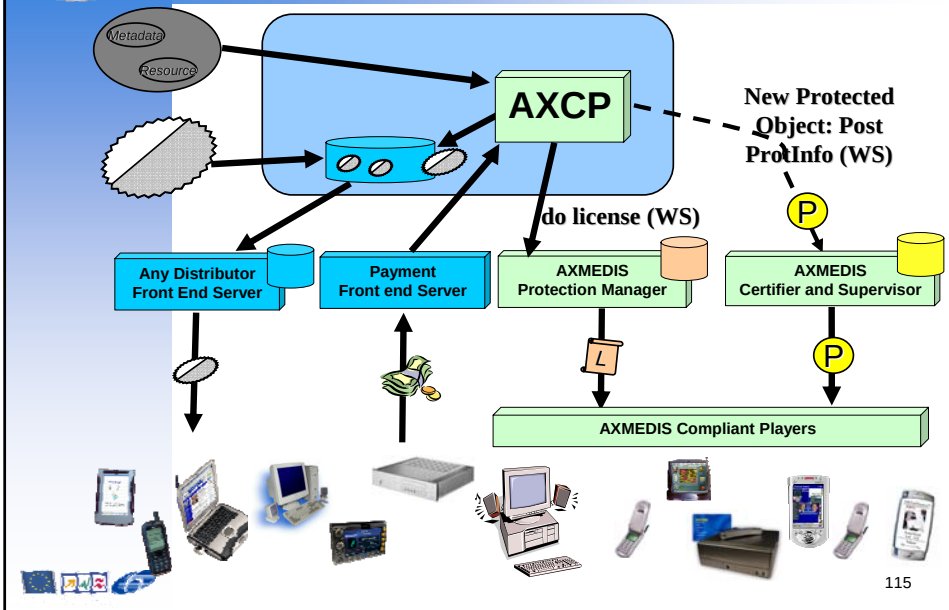
113

Running Rules with the Workflow



114

AXMEDIS The Protection and Control Process

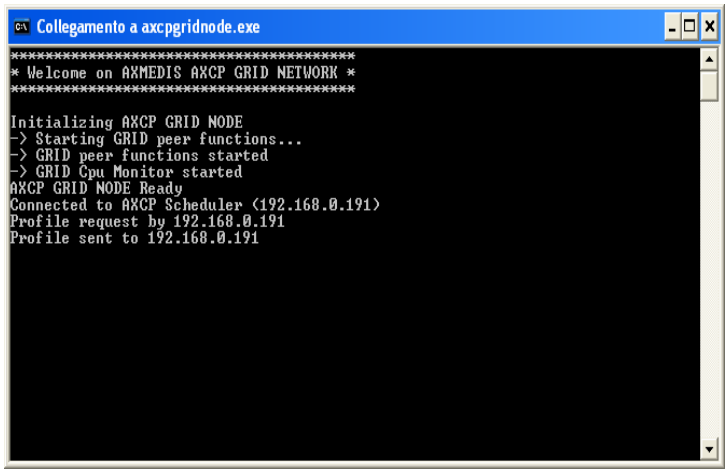


GRID in action

- Put in Execution the Scheduler
- Put in Execution the single GRID Nodes
- Joining of nodes to the GRID
- On the AXCP Editor: Installing Rules
- On the Scheduler: Running Rules with the Workflow

116

Discovered AXCP GRID Node

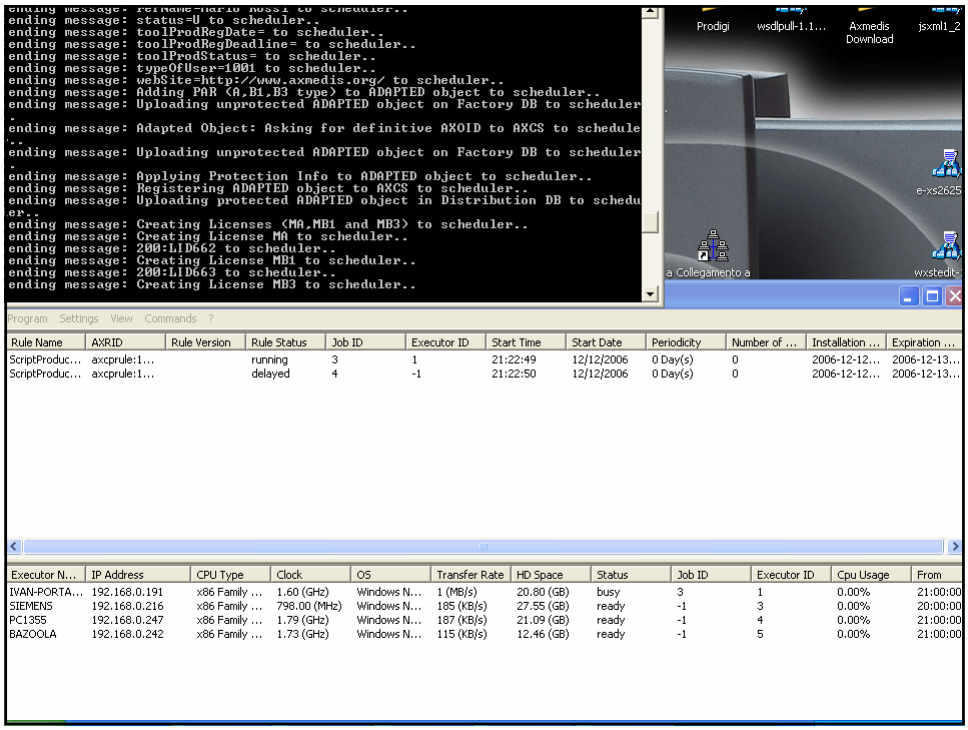


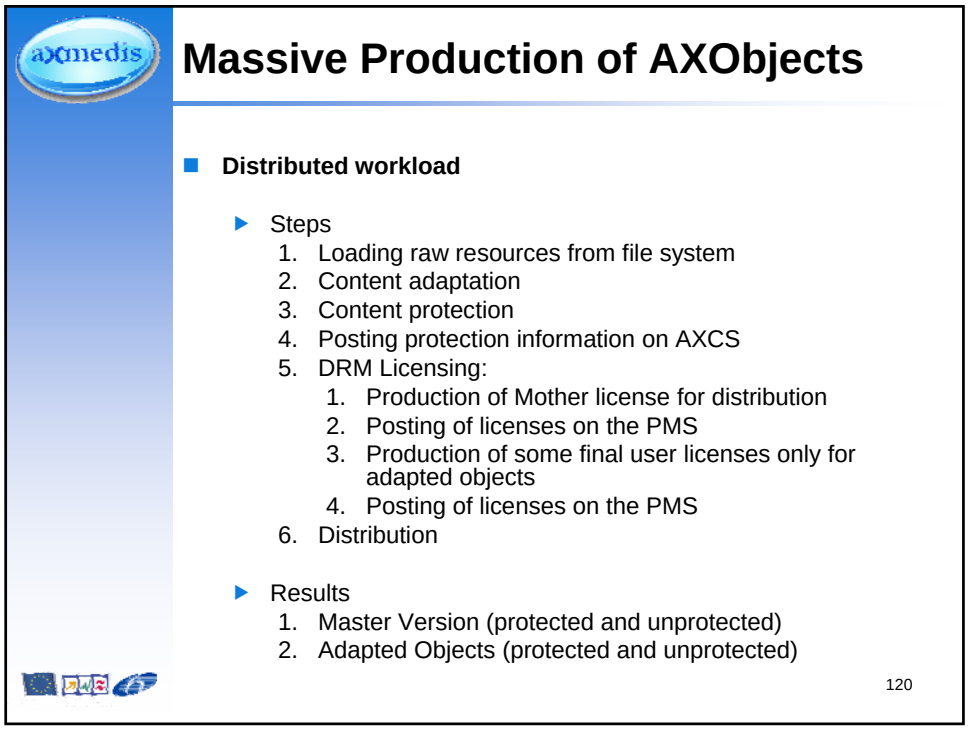
117

Installing Rules

- **Manually using the scheduler**
- **Installing rules with the AXCP Editor**
  - Script for Automated Production of content and production of licenses
- **Activating rule using external application**
  - AXCP Rule Editor, Workflow, …
  - Workflow is activating them
- **Running a rule**
  - On-demand execution

118


ending message: status=U to scheduler..
ending message: toolProdRegDate= to scheduler..
ending message: toolProdRegDeadline= to scheduler..
ending message: toolProdStatus= to scheduler..
ending message: typeOfUser=1001 to scheduler..
ending message: webSite=http://www.axmedis.org/ to scheduler..
ending message: Adding PAR (A,B1,B3 type) to ADAPTED object to scheduler..
ending message: Uploading unprotected ADAPTED object on Factory DB to scheduler
ending message: Adapted Object: Asking for definitive AXOID to AXCS to schedule
ending message: Uploading unprotected ADAPTED object on Factory DB to scheduler
ending message: Applying Protection Info to ADAPTED object to scheduler..
ending message: Registering ADAPTED object to AXCS to scheduler..
ending message: Uploading protected ADAPTED object in Distribution DB to schedu
er..
ending message: Creating Licenses (MA,MB1 and MB3) to scheduler..
ending message: Creating License MA to scheduler..
ending message: 200:LID662 to scheduler..
ending message: Creating License MB1 to scheduler..
ending message: 200:LID663 to scheduler..
ending message: Creating License MB3 to scheduler..
Prodigi
wsdlpull-1.1...
Axmedis Download
jsxml1_2
e-xs2625
wxstedit-
Program Settings View Commands ?
Rule Name | AXRID | Rule Version | Rule Status | Job ID | Executor ID | Start Time | Start Date | Periodicity | Number of ... | Installation ... | Expiration ...
ScriptProduc... | axcprule:1... | | running | 3 | 1 | 21:22:49 | 12/12/2006 | 0 Day(s) | 0 | 2006-12-12... | 2006-12-13...
ScriptProduc... | axcprule:1... | | delayed | 4 | -1 | 21:22:50 | 12/12/2006 | 0 Day(s) | 0 | 2006-12-12... | 2006-12-13...
Executor N... | IP Address | CPU Type | Clock | OS | Transfer Rate | HD Space | Status | Job ID | Executor ID | Cpu Usage | From
IVAN-PORTA... | 192.168.0.191 | x86 Family ... | 1.60 (GHz) | Windows N... | 1 (MB/s) | 20.80 (GB) | busy | 3 | 1 | 0.00% | 21:00:00
SIEMENS | 192.168.0.216 | x86 Family ... | 798.00 (MHz) | Windows N... | 185 (KB/s) | 27.55 (GB) | ready | -1 | 3 | 0.00% | 20:00:00
PC1355 | 192.168.0.247 | x86 Family ... | 1.79 (GHz) | Windows N... | 187 (KB/s) | 21.09 (GB) | ready | -1 | 4 | 0.00% | 21:00:00
BAZOOLA | 192.168.0.242 | x86 Family ... | 1.73 (GHz) | Windows N... | 115 (KB/s) | 12.46 (GB) | ready | -1 | 5 | 0.00% | 21:00:00

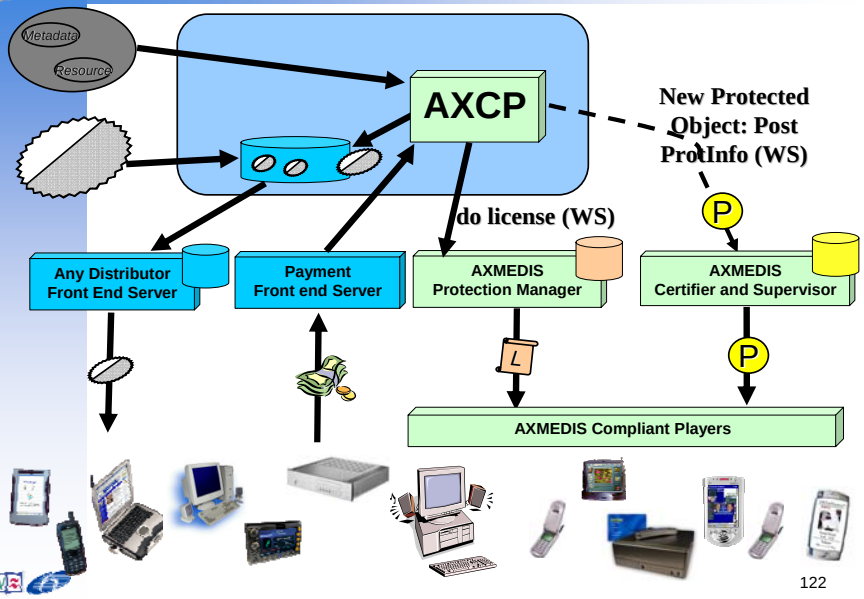

Massive Production of AXObjects
Distributed workload
Steps
1. Loading raw resources from file system
2. Content adaptation
3. Content protection
4. Posting protection information on AXCS
5. DRM Licensing:
1. Production of Mother license for distribution
2. Posting of licenses on the PMS
3. Production of some final user licenses only for adapted objects
4. Posting of licenses on the PMS
6. Distribution
Results
1. Master Version (protected and unprotected)
2. Adapted Objects (protected and unprotected)
120

What is produced

- **Some Different GRID nodes**
  - only two of them have capabilities to manage the rules in terms of plugins installed
- **Two rules activated on two different nodes**
- **The rules work on some resources**
  - a PDF is adapted to produce two objects in TXT and HTML
  - From an MP3 is produced an object
  - A wav file is adapted to produce an object with an MP3
  - From images in different formats, some adaptation resizing are performed
- **More than 30 AXMEDIS objects are created**

121

AXMEDIS The Protection and Control Process



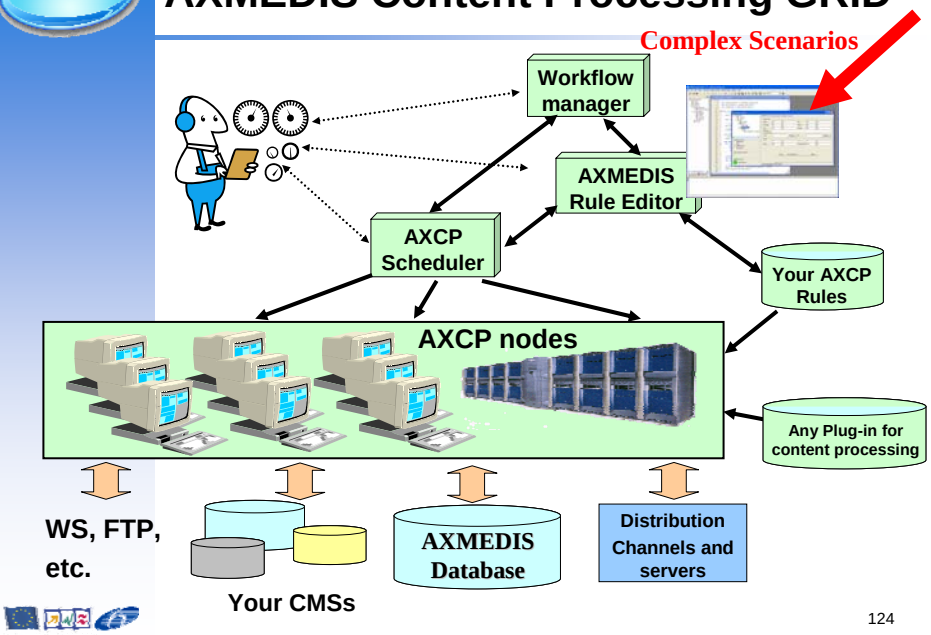
122

Complex Scenarios

Real Scenario taken from
Accademia Nazionale di Santa Cecilia

123

AXMEDIS Content Processing GRID



124

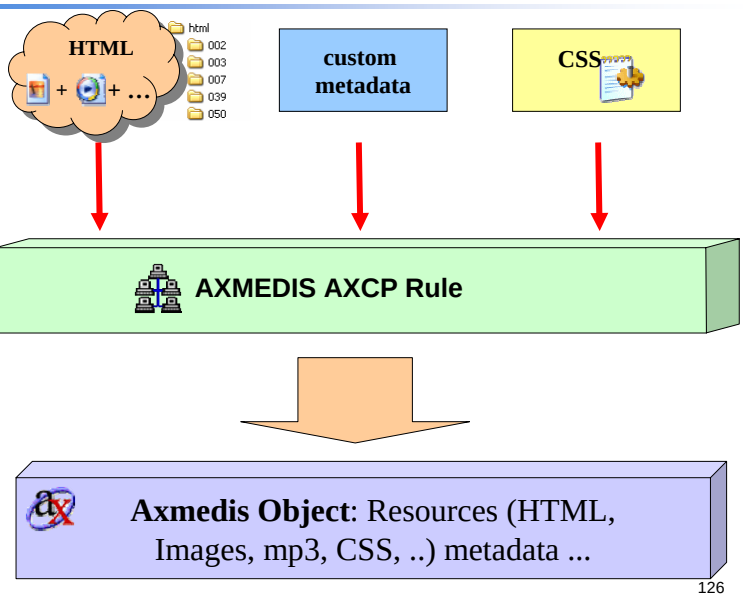
Complex Scenarios

- **Accademia Nazionale di Santa Cecilia,** http://www.santacecilia.it

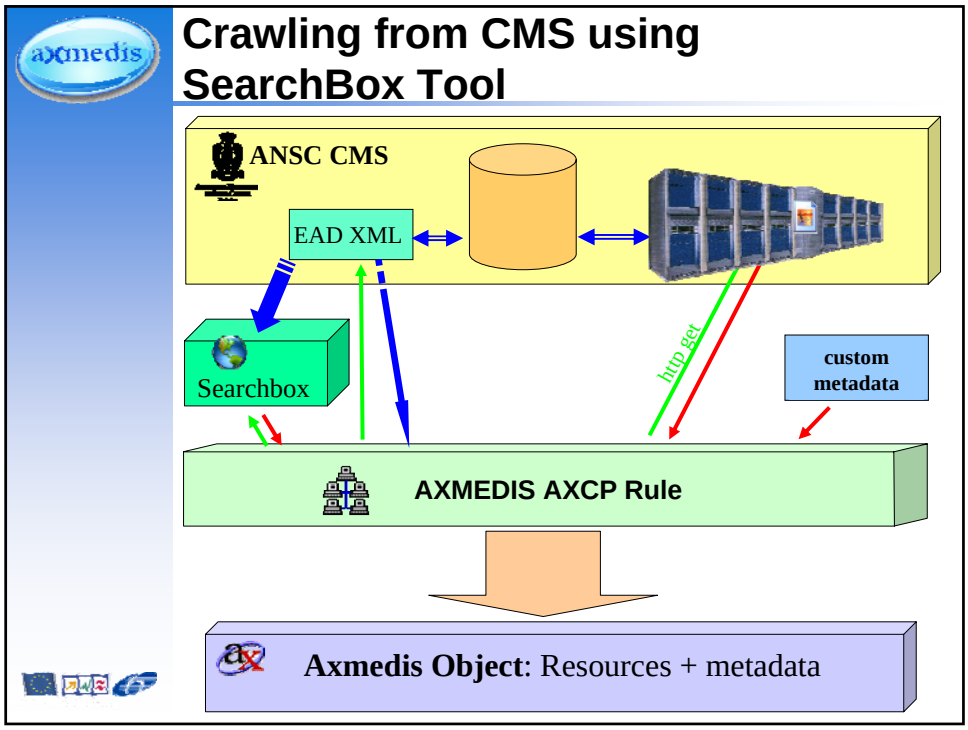
- **Examples:**
  - Content gathering from file system
  - Crawling from CMS using SearchBox Tool
  - Automatic SMIL representation

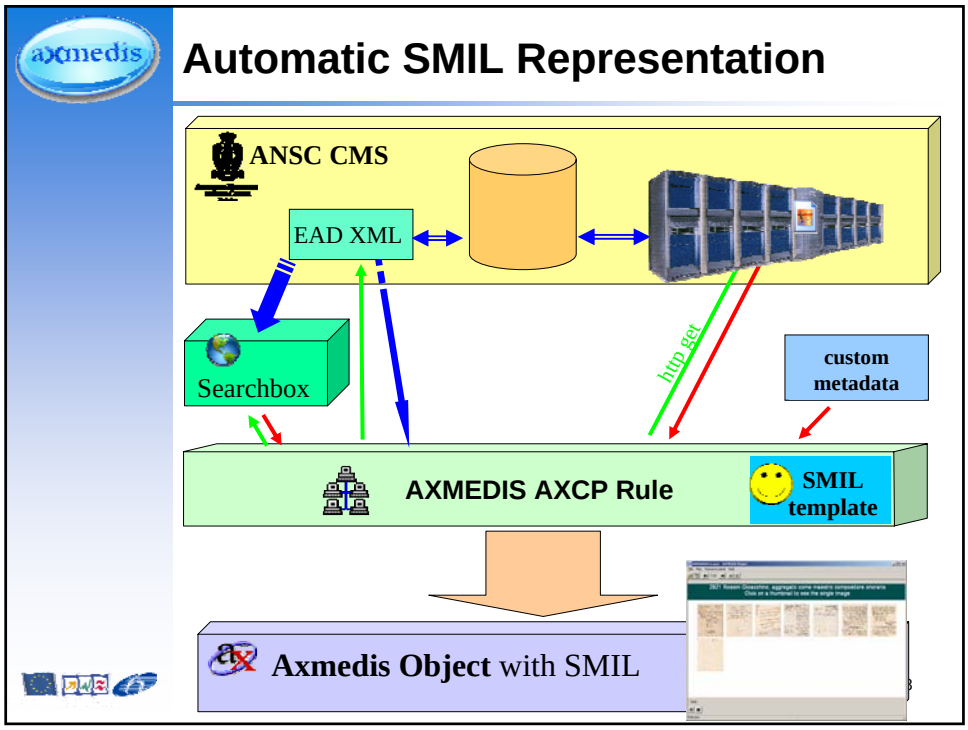
125

Content gathering from file system



126


Crawling from CMS using SearchBox Tool
ANSC CMS
EAD XML
Searchbox
http get
custom metadata
AXMEDIS AXCP Rule
Axmedis Object: Resources + metadata


Automatic SMIL Representation
ANSC CMS
EAD XML
Searchbox
http get
custom metadata
AXMEDIS AXCP Rule
SMIL template
Axmedis Object with SMIL

Summary

- Rules Definition, test and validation
- Rules installation and Execution
- Rules Scheduling
- AXMEDIS GRID management

129

Adding Your Plugins

- Flexible Plugin Interface already used to integrate the extended functionality
- Examples:
  - ImageMagick,
  - FFMPEG,
  - CryptLib,
  - MPEG-7 reference implementation,
  - ...
- External tools
  - M2ANY: AudioID (integrated)
  - ISHCE: Audio Watermarking (under development)

130

Conclusions

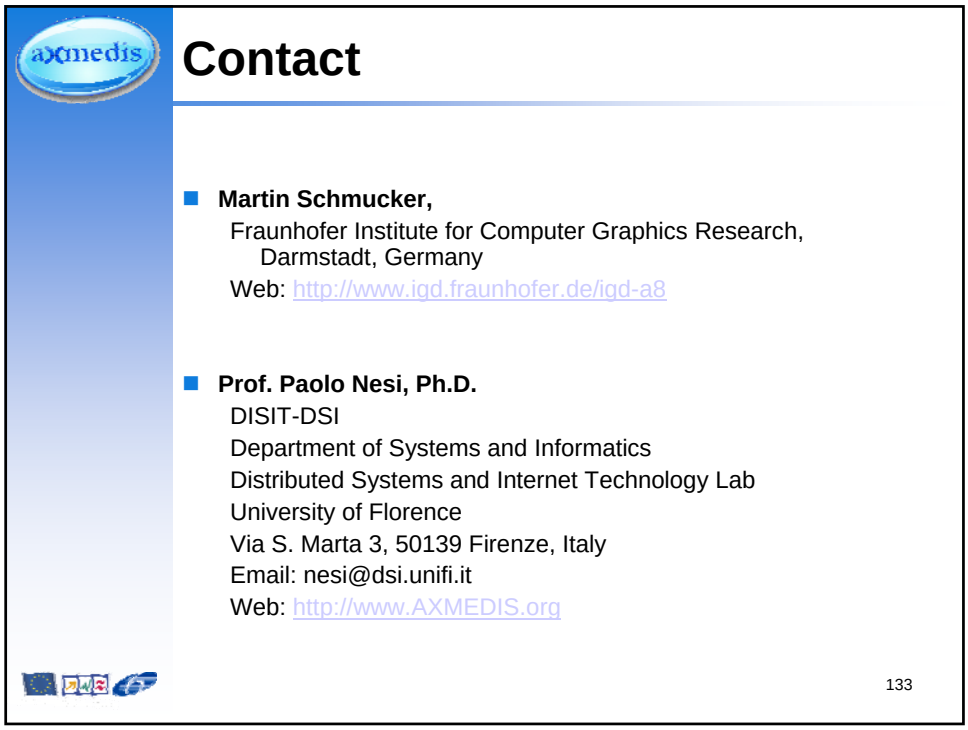
- **The AXMEDIS Scripting Language provides a flexible solution for automatic content processing.**
- **The AXMEDIS Core Functionality provides the basic method for the management of digital content and corresponding rights.**
- **External Functionality included in AXMEDIS enriches the basic functionality to cover daily tasks.**
- **The AXMEDIS GRID allows a dynamic load balancing.**
- **The extensibility of the AXMEDIS Framework provides the flexibility for future scenarios:**
  - scripting and creation of new AXMEDIS plugins.

131

Conclusions

- **AXMEDIS reduces costs for content management by providing a solution for automating the content processing, production, protection and distribution.**
- **AXMEDIS reduces distribution and aggregation costs in order to increase accessibility with a**
  - Peer-to-Peer platform at Business-to-Business level
  - Integration of content management systems
  - Integration of workflows

132


Contact
Martin Schmucker,
Fraunhofer Institute for Computer Graphics Research,
Darmstadt, Germany
Web: http://www.igd.fraunhofer.de/igd-a8
Prof. Paolo Nesi, Ph.D.
DISIT-DSI
Department of Systems and Informatics
Distributed Systems and Internet Technology Lab
University of Florence
Via S. Marta 3, 50139 Firenze, Italy
Email: nesi@dsi.unifi.it
Web: http://www.AXMEDIS.org
133