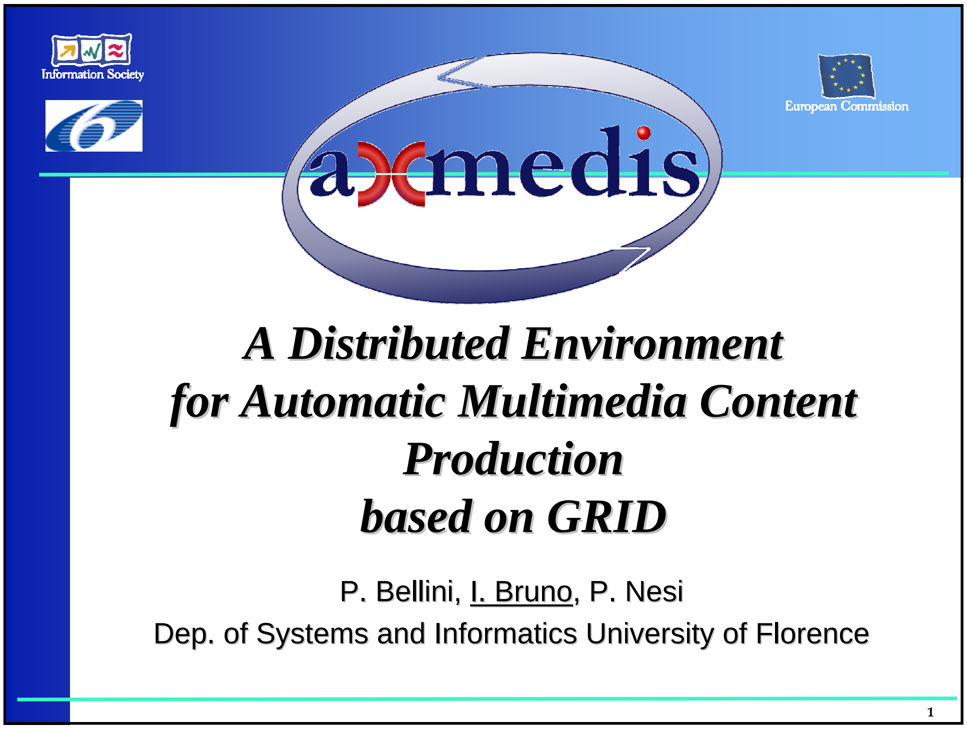
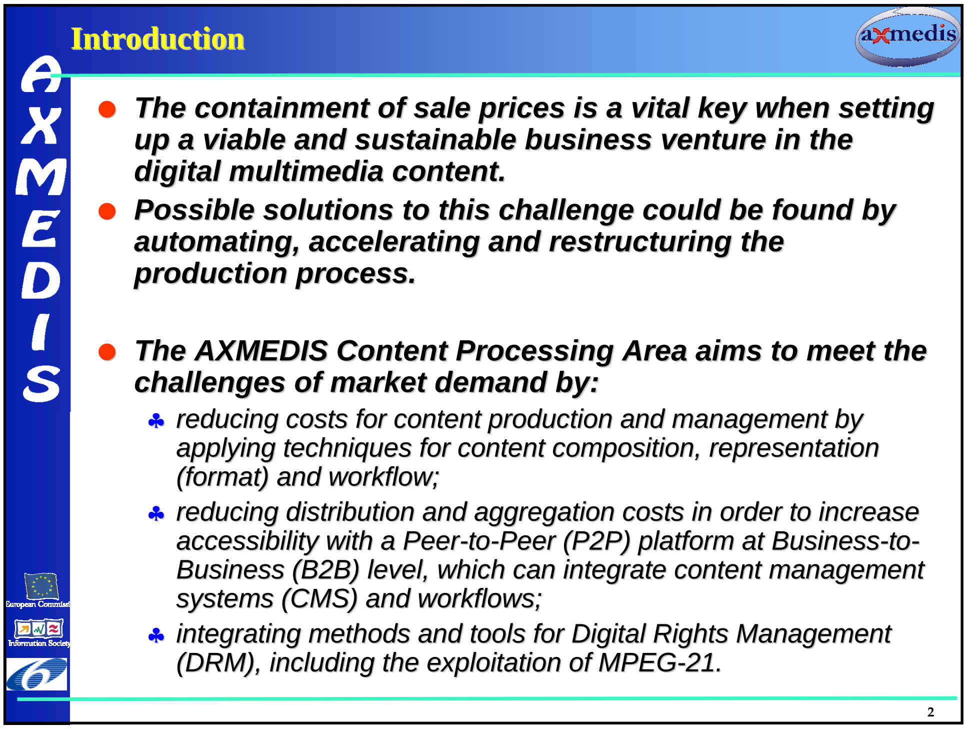


A Distributed Environment for Automatic Multimedia Content Production based on GRID

P. Bellini, I. Bruno, P. Nesi
Dep. of Systems and Informatics University of Florence

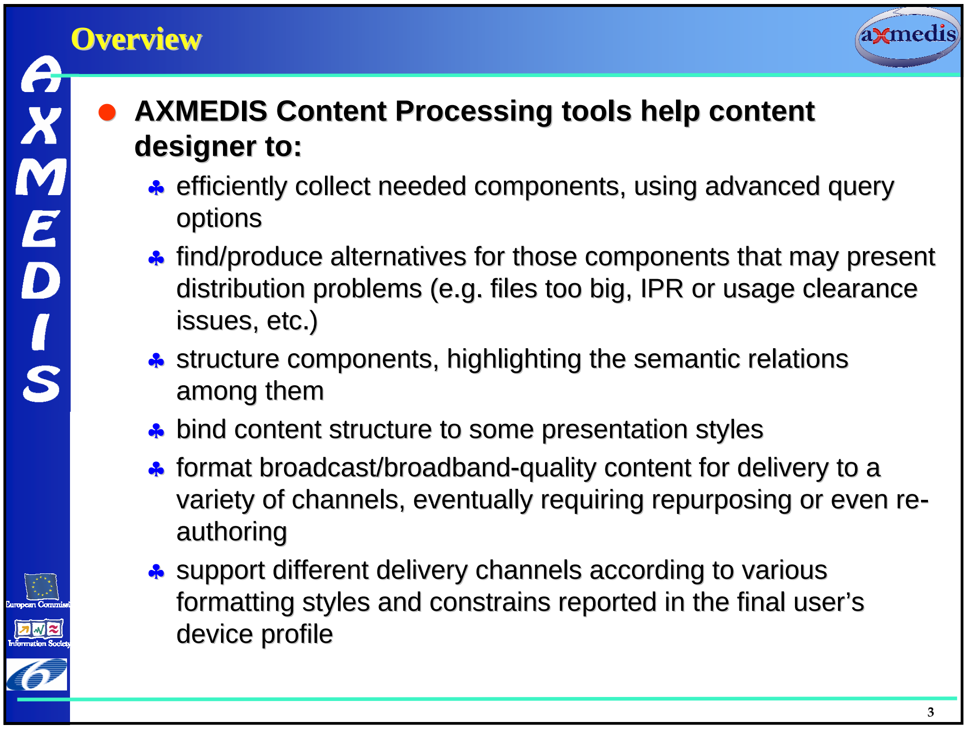
1



Introduction

- ***The containment of sale prices is a vital key when setting up a viable and sustainable business venture in the digital multimedia content.***
- ***Possible solutions to this challenge could be found by automating, accelerating and restructuring the production process.***
- ***The AXMEDIS Content Processing Area aims to meet the challenges of market demand by:***
  - *reducing costs for content production and management by applying techniques for content composition, representation (format) and workflow;*
  - *reducing distribution and aggregation costs in order to increase accessibility with a Peer-to-Peer (P2P) platform at Business-to-Business (B2B) level, which can integrate content management systems (CMS) and workflows;*
  - *integrating methods and tools for Digital Rights Management (DRM), including the exploitation of MPEG-21.*

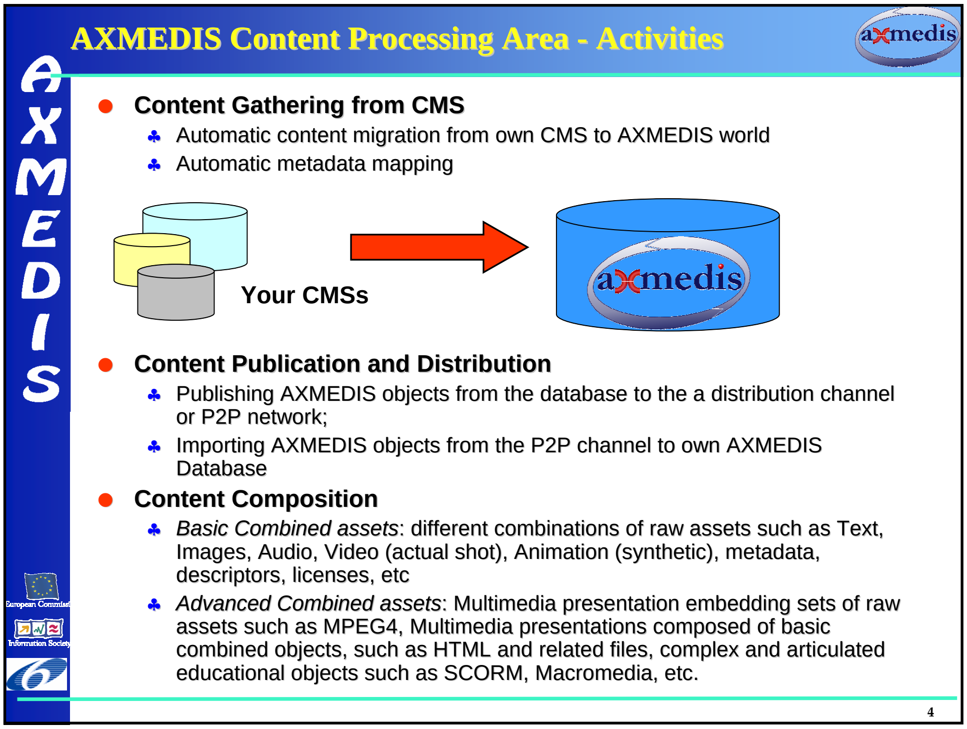
2



Overview

- **AXMEDIS Content Processing tools help content designer to:**
  - efficiently collect needed components, using advanced query options
  - find/produce alternatives for those components that may present distribution problems (e.g. files too big, IPR or usage clearance issues, etc.)
  - structure components, highlighting the semantic relations among them
  - bind content structure to some presentation styles
  - format broadcast/broadband-quality content for delivery to a variety of channels, eventually requiring repurposing or even re-authoring
  - support different delivery channels according to various formatting styles and constrains reported in the final user's device profile

3



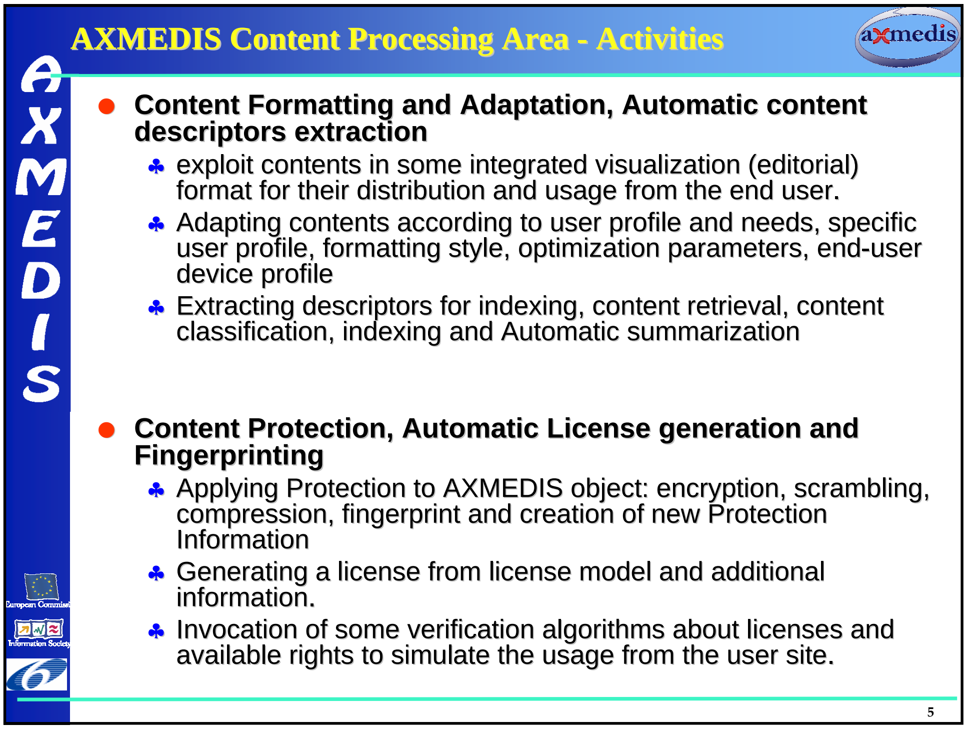
AXMEDIS Content Processing Area - Activities

- **Content Gathering from CMS**
  - Automatic content migration from own CMS to AXMEDIS world
  - Automatic metadata mapping

Your CMSs
axmedis

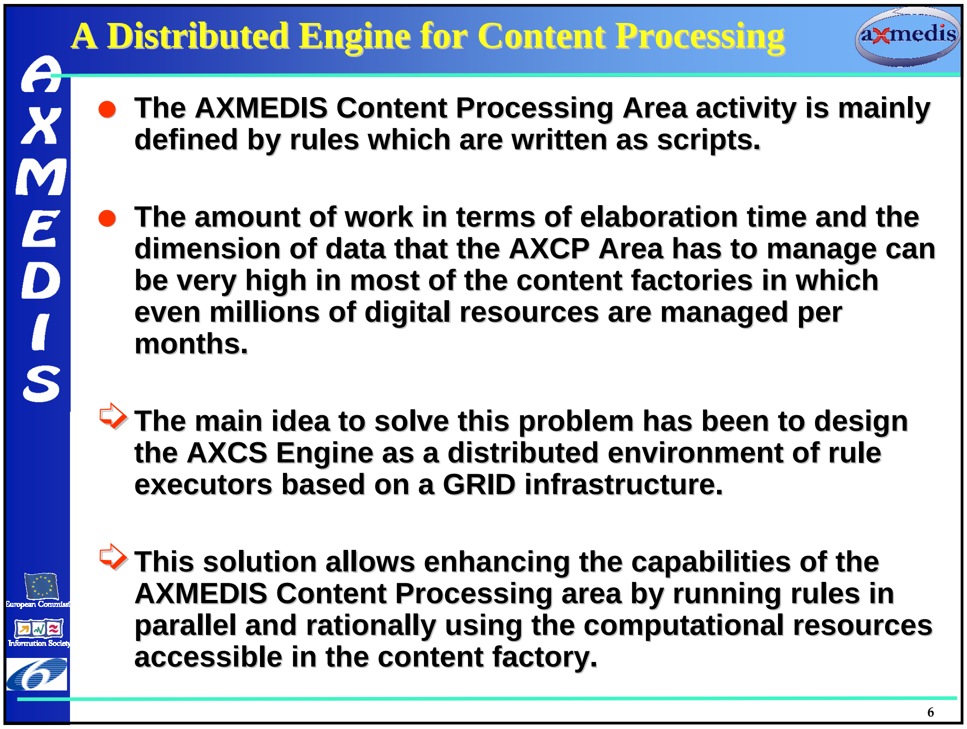
- **Content Publication and Distribution**
  - Publishing AXMEDIS objects from the database to the a distribution channel or P2P network;
  - Importing AXMEDIS objects from the P2P channel to own AXMEDIS Database
- **Content Composition**
  - *Basic Combined assets*: different combinations of raw assets such as Text, Images, Audio, Video (actual shot), Animation (synthetic), metadata, descriptors, licenses, etc
  - *Advanced Combined assets*: Multimedia presentation embedding sets of raw assets such as MPEG4, Multimedia presentations composed of basic combined objects, such as HTML and related files, complex and articulated educational objects such as SCORM, Macromedia, etc.

4



AXMEDIS Content Processing Area - Activities

- **Content Formatting and Adaptation, Automatic content descriptors extraction**
  - exploit contents in some integrated visualization (editorial) format for their distribution and usage from the end user.
  - Adapting contents according to user profile and needs, specific user profile, formatting style, optimization parameters, end-user device profile
  - Extracting descriptors for indexing, content retrieval, content classification, indexing and Automatic summarization
- **Content Protection, Automatic License generation and Fingerprinting**
  - Applying Protection to AXMEDIS object: encryption, scrambling, compression, fingerprint and creation of new Protection Information
  - Generating a license from license model and additional information.
  - Invocation of some verification algorithms about licenses and available rights to simulate the usage from the user site.



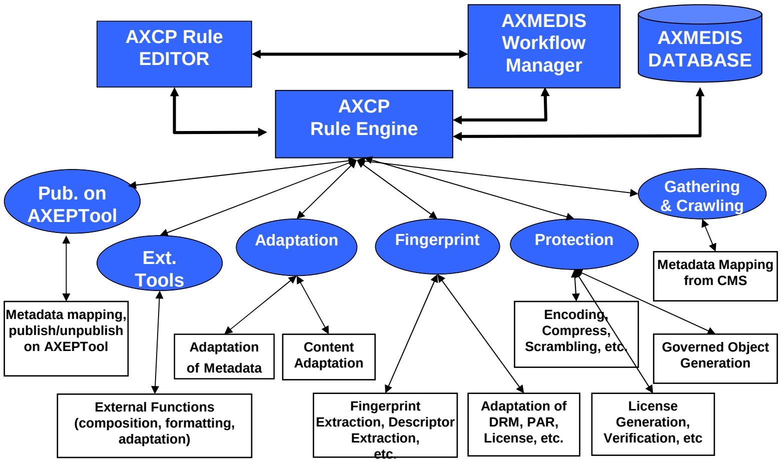
A Distributed Engine for Content Processing

- **The AXMEDIS Content Processing Area activity is mainly defined by rules which are written as scripts.**
- **The amount of work in terms of elaboration time and the dimension of data that the AXCP Area has to manage can be very high in most of the content factories in which even millions of digital resources are managed per months.**

⇨ **The main idea to solve this problem has been to design the AXCS Engine as a distributed environment of rule executors based on a GRID infrastructure.**

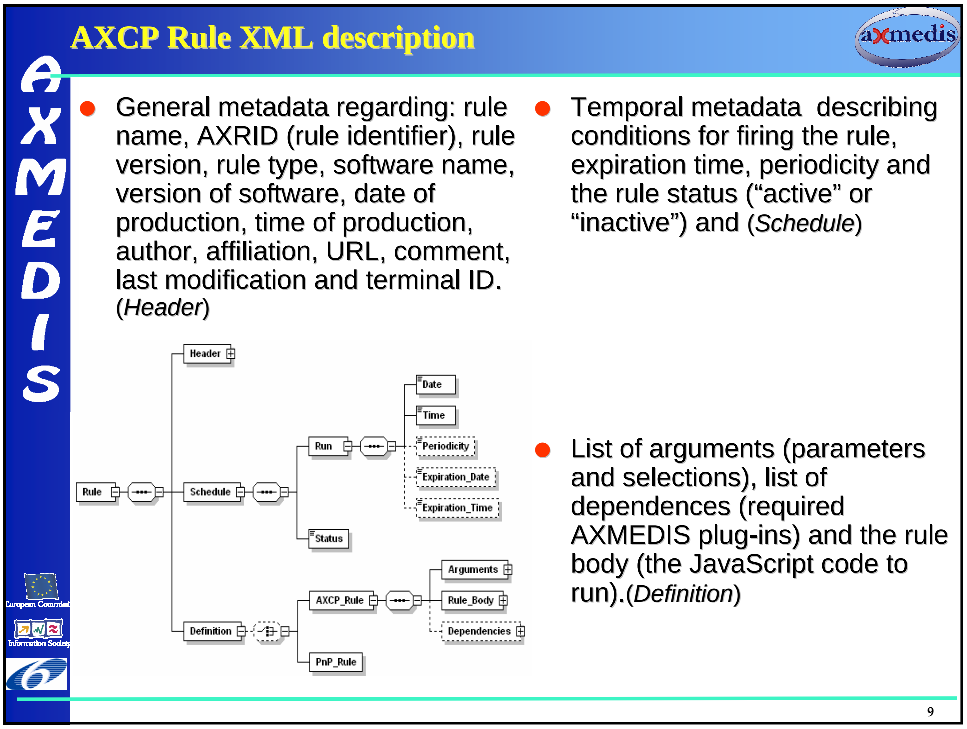
⇨ **This solution allows enhancing the capabilities of the AXMEDIS Content Processing area by running rules in parallel and rationally using the computational resources accessible in the content factory.**

AXMEDIS Content Processing Area - Architecture


AXCP Rule EDITOR
AXMEDIS Workflow Manager
AXMEDIS DATABASE
AXCP Rule Engine
Pub. on AXEPTool
Gathering & Crawling
Ext. Tools
Adaptation
Fingerprint
Protection
Metadata Mapping from CMS
Metadata mapping, publish/unpublish on AXEPTool
Adaptation of Metadata
Content Adaptation
Encoding, Compress, Scrambling, etc
Governed Object Generation
External Functions (composition, formatting, adaptation)
Fingerprint Extraction, Descriptor Extraction, etc.
Adaptation of DRM, PAR, License, etc.
License Generation, Verification, etc

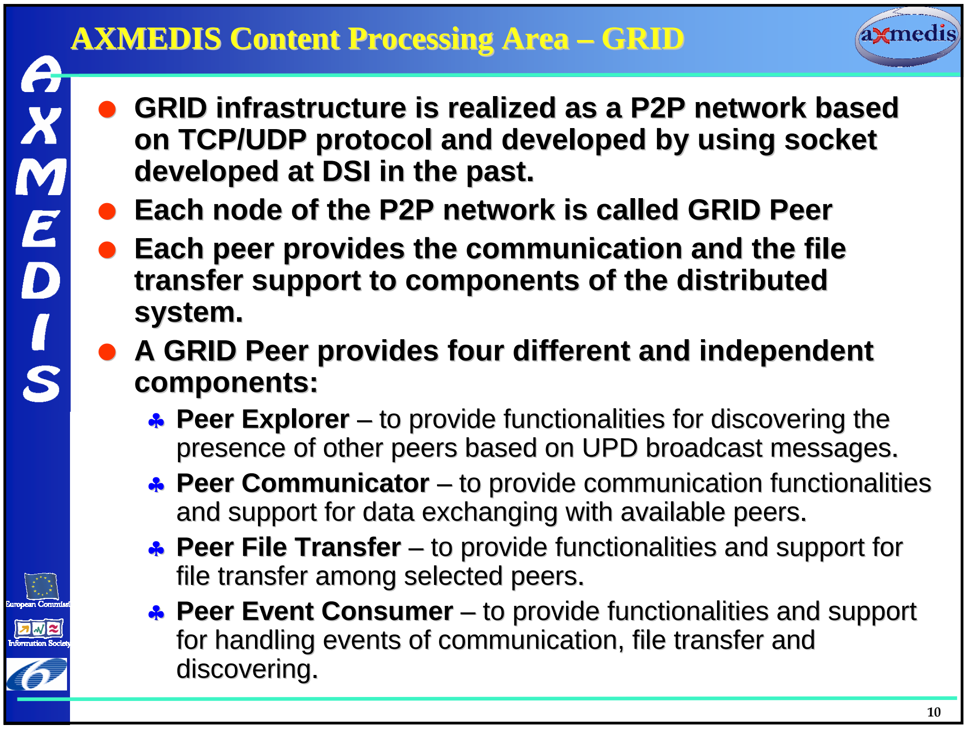
AXCP Rule Formalisation

- **An AXCP Rule is a procedure:**
- $R = f(S_1,S_2,..,S_n,P_1,\ldots,P_m)$
- **Where:**
- $\boldsymbol{S_i}$ is a database Selection, it is a sequence of queries to be sent to the AXMEDIS Database to retrieve objects IDs or a set of object IDs to AXMEDIS objects or a mix of them;
- $\boldsymbol{P_i}$ 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.
- $\boldsymbol{f}$ is the identifier of rule (name of rule or an ID);
- $\boldsymbol{R}$ 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 scheduler, etc…



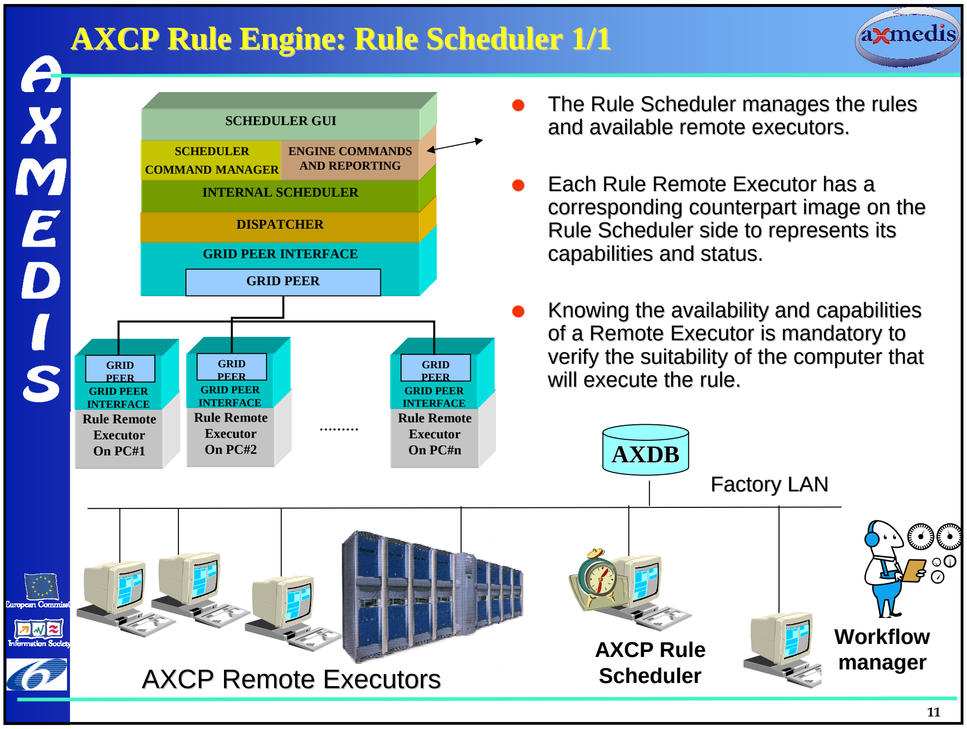
AXCP Rule XML description

- 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 JavaScript code to run).(*Definition*)



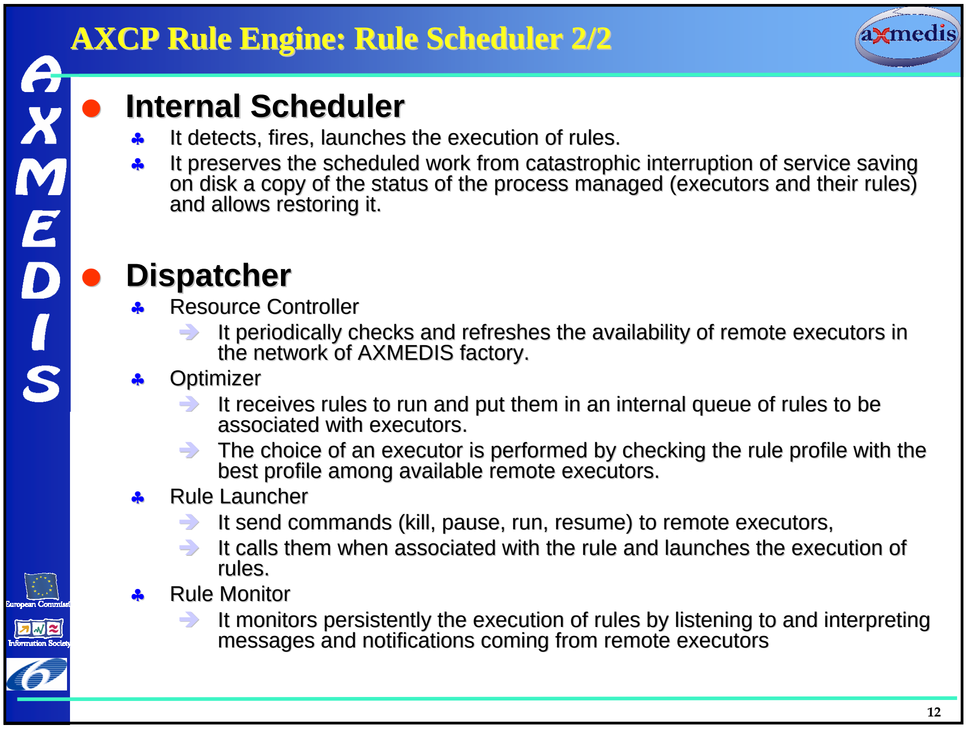
AXMEDIS Content Processing Area – GRID

- **GRID infrastructure is realized as a P2P network based on TCP/UDP protocol and developed by using socket developed at DSI in the past.**
- **Each node of the P2P network is called GRID Peer**
- **Each peer provides the communication and the file transfer support to components of the distributed system.**
- **A GRID Peer provides four different and independent components:**
  - **Peer Explorer** – to provide functionalities for discovering the presence of other peers based on UPD broadcast messages.
  - **Peer Communicator** – to provide communication functionalities and support for data exchanging with available peers.
  - **Peer File Transfer** – to provide functionalities and support for file transfer among selected peers.
  - **Peer Event Consumer** – to provide functionalities and support for handling events of communication, file transfer and discovering.



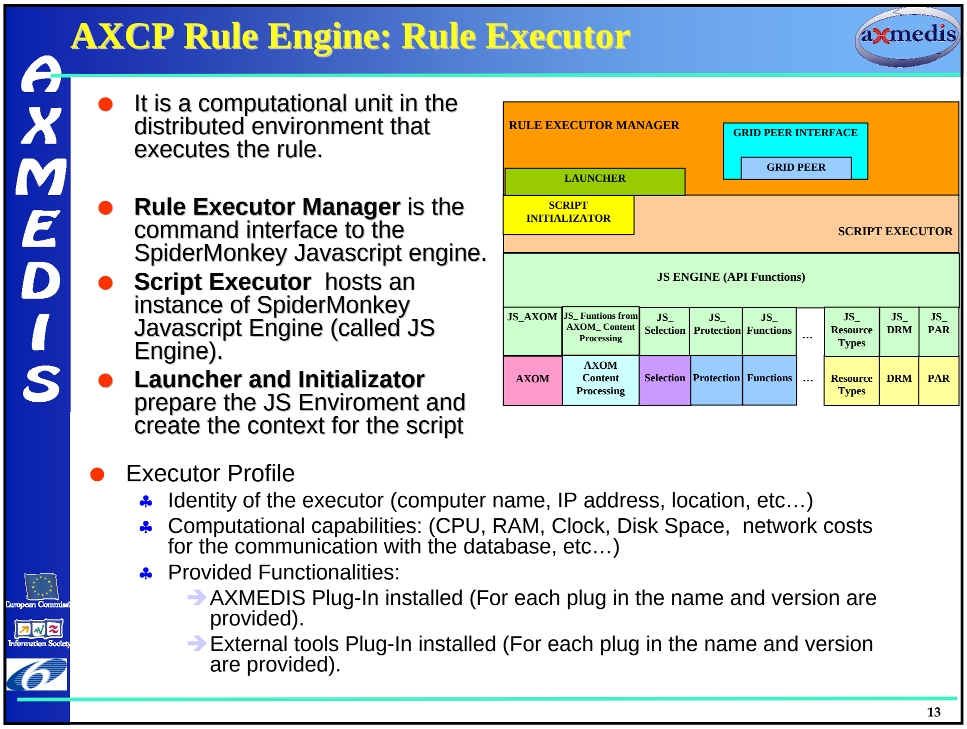
AXCP Rule Engine: Rule Scheduler 1/1

- The Rule Scheduler manages the rules and available remote executors.
- Each Rule Remote Executor has a corresponding counterpart image on the Rule Scheduler side to represents its capabilities and status.
- Knowing the availability and capabilities of a Remote Executor is mandatory to verify the suitability of the computer that will execute the rule.



AXCP Rule Engine: Rule Scheduler 2/2

- **Internal Scheduler**
  - It detects, fires, launches the execution of rules.
  - It preserves the scheduled work from catastrophic interruption of service saving on disk a copy of the status of the process managed (executors and their rules) and allows restoring it.
- **Dispatcher**
  - Resource Controller
    - It periodically checks and refreshes the availability of remote executors in the network of AXMEDIS factory.
  - Optimizer
    - It receives rules to run and put them in an internal queue of rules to be associated with executors.
    - The choice of an executor is performed by checking the rule profile with the best profile among available remote executors.
  - Rule Launcher
    - It send commands (kill, pause, run, resume) to remote executors,
    - It calls them when associated with the rule and launches the execution of rules.
  - Rule Monitor
    - It monitors persistently the execution of rules by listening to and interpreting messages and notifications coming from remote executors



AXCP Rule Engine: Rule Executor

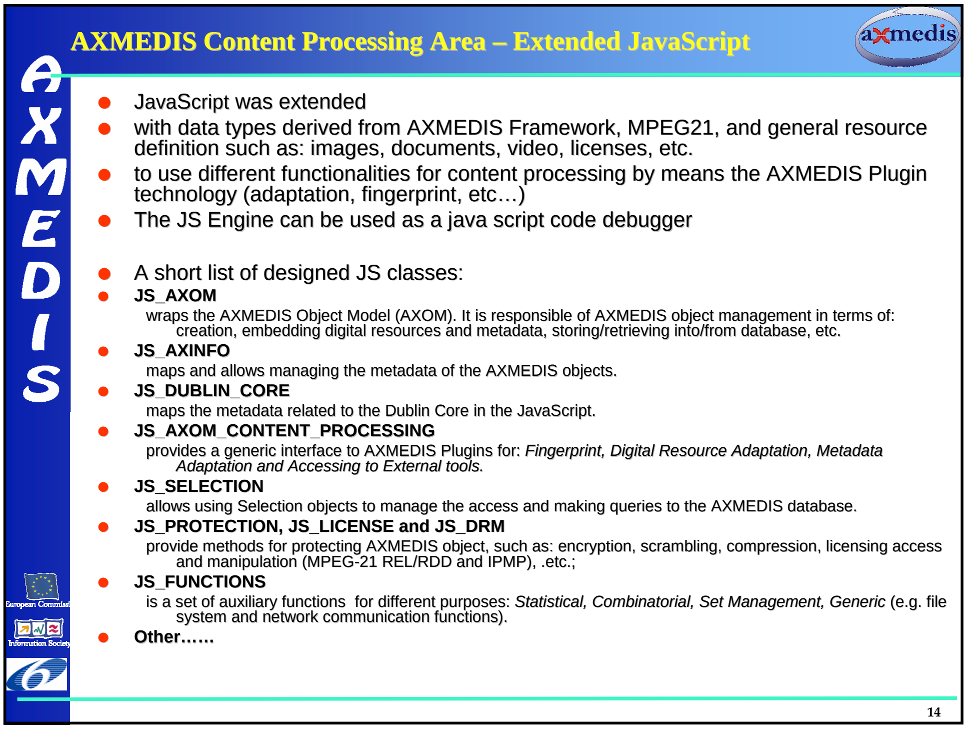
- It is a computational unit in the distributed environment that executes the rule.
- **Rule Executor Manager** is the command interface to the SpiderMonkey Javascript engine.
- **Script Executor** hosts an instance of SpiderMonkey Javascript Engine (called JS Engine).
- **Launcher and Initializator** prepare the JS Enviroment and create the context for the script

RULE EXECUTOR MANAGER
GRID PEER INTERFACE
GRID PEER
LAUNCHER
SCRIPT INITIALIZATOR
SCRIPT EXECUTOR
JS ENGINE (API Functions)

JS_AXOM	JS_ Funtions from AXOM_ Content Processing	JS_ Selection	JS_ Protection	JS_ Functions	…	JS_ Resource Types	JS_ DRM	JS_ PAR
AXOM	AXOM Content Processing	Selection	Protection	Functions	…	Resource Types	DRM	PAR

- Executor Profile
  - Identity of the executor (computer name, IP address, location, etc…)
  - Computational capabilities: (CPU, RAM, Clock, Disk Space, network costs for the communication with the database, etc…)
  - Provided Functionalities:
    - ➔AXMEDIS Plug-In installed (For each plug in the name and version are provided).
    - ➔External tools Plug-In installed (For each plug in the name and version are provided).

13



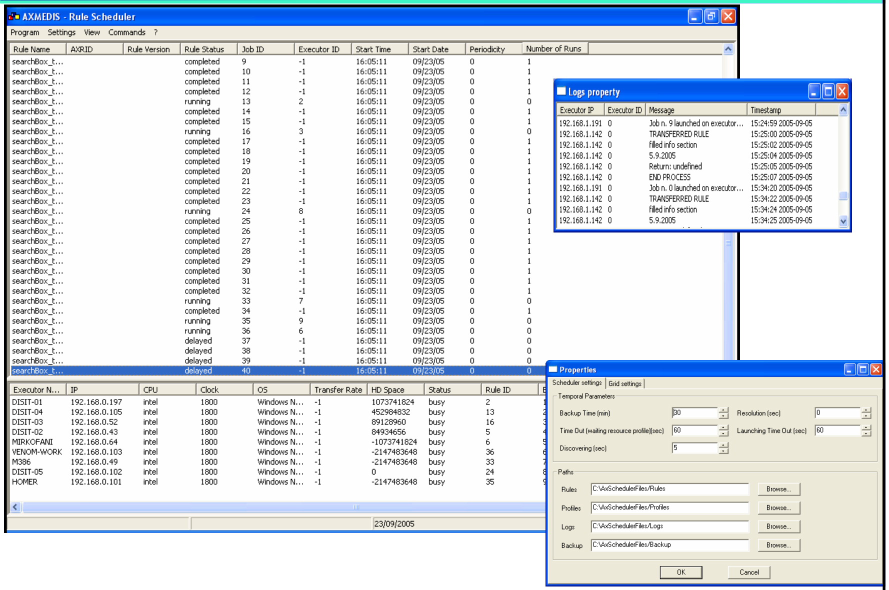
AXMEDIS Content Processing Area – Extended JavaScript

- JavaScript was extended
- with data types derived from AXMEDIS Framework, MPEG21, and general resource definition such as: images, documents, video, licenses, etc.
- to use different functionalities for content processing by means the AXMEDIS Plugin technology (adaptation, fingerprint, etc…)
- The JS Engine can be used as a java script code debugger

- A short list of designed JS classes:
- **JS_AXOM**
  wraps the AXMEDIS Object Model (AXOM). It is responsible of AXMEDIS object management in terms of: creation, embedding digital resources and metadata, storing/retrieving into/from database, etc.
- **JS_AXINFO**
  maps and allows managing the metadata of the AXMEDIS objects.
- **JS_DUBLIN_CORE**
  maps the metadata related to the Dublin Core in the JavaScript.
- **JS_AXOM_CONTENT_PROCESSING**
  provides a generic interface to AXMEDIS Plugins for: *Fingerprint, Digital Resource Adaptation, Metadata Adaptation and Accessing to External tools.*
- **JS_SELECTION**
  allows using Selection objects to manage the access and making queries to the AXMEDIS database.
- **JS_PROTECTION, JS_LICENSE and JS_DRM**
  provide methods for protecting AXMEDIS object, such as: encryption, scrambling, compression, licensing access and manipulation (MPEG-21 REL/RDD and IPMP), .etc.;
- **JS_FUNCTIONS**
  is a set of auxiliary functions for different purposes: *Statistical, Combinatorial, Set Management, Generic* (e.g. file system and network communication functions).
- **Other……**

14

AXCP Rule Engine: Rule Scheduler GUI



Rule Name	AXRID	Rule Version	Rule Status	Job ID	Executor ID	Start Time	Start Date	Periodicity	Number of Runs
searchBox_t...			completed	9	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	10	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	11	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	12	-1	16:05:11	09/23/05	0	1
searchBox_t...			running	13	2	16:05:11	09/23/05	0	0
searchBox_t...			completed	14	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	15	-1	16:05:11	09/23/05	0	1
searchBox_t...			running	16	3	16:05:11	09/23/05	0	0
searchBox_t...			completed	17	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	18	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	19	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	20	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	21	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	22	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	23	-1	16:05:11	09/23/05	0	1
searchBox_t...			running	24	8	16:05:11	09/23/05	0	0
searchBox_t...			completed	25	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	26	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	27	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	28	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	29	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	30	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	31	-1	16:05:11	09/23/05	0	1
searchBox_t...			completed	32	-1	16:05:11	09/23/05	0	1
searchBox_t...			running	33	7	16:05:11	09/23/05	0	0
searchBox_t...			completed	34	-1	16:05:11	09/23/05	0	1
searchBox_t...			running	35	9	16:05:11	09/23/05	0	0
searchBox_t...			running	36	6	16:05:11	09/23/05	0	0
searchBox_t...			delayed	37	-1	16:05:11	09/23/05	0	0
searchBox_t...			delayed	38	-1	16:05:11	09/23/05	0	0
searchBox_t...			delayed	39	-1	16:05:11	09/23/05	0	0
searchBox_t...			delayed	40	-1	16:05:11	09/23/05	0	0

Executor N...	IP	CPU	Clock	OS	Transfer Rate	HD Space	Status	Rule ID
DISIT-01	192.168.0.197	intel	1800	Windows N...	-1	1073741824	busy	2
DISIT-04	192.168.0.105	intel	1800	Windows N...	-1	452984832	busy	13
DISIT-03	192.168.0.52	intel	1800	Windows N...	-1	89128960	busy	16
DISIT-02	192.168.0.43	intel	1800	Windows N...	-1	84934656	busy	5
MIRKOFANI	192.168.0.64	intel	1800	Windows N...	-1	-1073741824	busy	6
VENOM-WORK	192.168.0.103	intel	1800	Windows N...	-1	-2147483648	busy	36
M386	192.168.0.49	intel	1800	Windows N...	-1	-2147483648	busy	33
DISIT-05	192.168.0.102	intel	1800	Windows N...	-1	0	busy	24
HOMER	192.168.0.101	intel	1800	Windows N...	-1	-2147483648	busy	35

23/09/2005

Logs property

Executor IP	Executor ID	Message	Timestamp
192.168.1.191	0	Job n. 9 launched on executor...	15:24:59 2005-09-05
192.168.1.142	0	TRANSFERRED RULE	15:25:00 2005-09-05
192.168.1.142	0	filled info section	15:25:02 2005-09-05
192.168.1.142	0	5.9.2005	15:25:04 2005-09-05
192.168.1.142	0	Return: undefined	15:25:05 2005-09-05
192.168.1.142	0	END PROCESS	15:25:07 2005-09-05
192.168.1.191	0	Job n. 0 launched on executor...	15:34:20 2005-09-05
192.168.1.142	0	TRANSFERRED RULE	15:34:22 2005-09-05
192.168.1.142	0	filled info section	15:34:24 2005-09-05
192.168.1.142	0	5.9.2005	15:34:25 2005-09-05

Properties

Temporal Parameters

Backup Time (min) 30
Resolution (sec) 0
Time Out (waiting resource profile)(sec) 60
Launching Time Out (sec) 60
Discovering (sec) 5

Paths

Rules C:\AxSchedulerFiles/Rules
Profiles C:\AxSchedulerFiles/Profiles
Logs C:\AxSchedulerFiles/Logs
Backup C:\AxSchedulerFiles/Backup

Conclusion

- The architecture of the AXMEDIS Content Processing Area has been described.
- This Area is a core subsystem of the AXMEDIS Framework and architecture for automatic content production, protection, formatting, metadata adaptation, etc.,
- The adopted solution was based on GRID Computing.
- The whole activity of the AXMEDIS Content Processing Area is performed by a distributed Rule Engine that runs different AXCP rule in parallel.

- Future work
- The rational choice of the computational resources will be one of the main future activities inside the AXEMDIS project.
- This will be focused on the optimizer module inside the Scheduler
- optimisation algorithm and artificial intelligence (e.g. Taboo Search, Genetic Algorithm, etc…) will be investigated to allow using efficiently the GRID infrastructure

AXMEDIS
axmedis
European Commission
Information Society

Thanks for your attention!

17