Automating Production of Cross Media Content for Multi-channel Distribution

www.AXMEDIS.org

DE4.4.1.3
Content Sharing and Production on P2P

Version: 1.1
Date: 20/09/2007
Responsible: DSI (revised and approved by coordinator)

Project Number: IST-2-511299
Project Title: AXMEDIS
Deliverable Type: Report and Prototype
Visible to User Groups: Yes
Visible to Affiliated: Yes
Visible to Public: Yes

Deliverable Number: DE4.4.1.3
Contractual Date of Delivery: M36
Actual Date of Delivery: 20/09/2007
Work-Package contributing to the Deliverable: (WP4)
Task contributing to the Deliverable: WP4.4
Nature of the Deliverable: Report and Prototype
Author(s): DSI, EXITECH

Abstract:
This document reports on research activities and demonstrator implementation related to WP4.4 Content Sharing and Production on P2P

Keyword List:
Peer-to-peer, content sharing, content search, content download, automated activities, query user interfaces, integration of P2P with AXMEDIS Factory, bitTorrent.

Table of Contents

1 EXECUTIVE SUMMARY AND REPORT SCOPE 5

2 INTRODUCTION 6

2.1 T4.4.1 AXEPTOOL, P2P TOOL FOR B2B TRANSACTION AND CONTENT DISTRIBUTION (DSI) 6
2.2 T4.4.2: THE QUERY SUPPORT INTO THE AXEPTOOL DATABASE (EXITECH) 7

3 PERFORMED AND PLANNED ACTIVITY 8

4 STATE OF THE ART OF P2P BITTORRENT 9

4.1 BITTORRENT TERMINOLOGY 14
4.2 GENERAL INFORMATION ON BITTORRENT TRACKER 14
4.2.1 Lists of Trackers 15
4.3 SERVERS FOR LISTING AND SEARCHING TORRENT FILES 16
4.4 SOME ADDITIONAL ISSUES OF BITTORRENT 16
4.4.1 Alternative approaches 16
4.4.2 Legal defenses 16
4.5 OTHER FEATURES OF BITTORRENT 17
4.5.1 Utilities 17
4.5.2 Broadcatching 17
4.5.3 APIs 17
4.5.4 Multitracker 18

5 REVISED REQUIREMENTS FOR P2P TOOLS IN AXMEDIS 20

5.1 SYSTEM AND IPR REQUIREMENTS FOR P2P TOOLS OF AXMEDIS 20
5.2 GENERAL P2P REQUIREMENTS VALID FOR BOTH AXMEDIA AND AXEPTOOL P2PS 20
5.3 SPECIFIC AXEPTOOL FOR P2P ON B2B 21
5.4 SPECIFIC AXMEDIA TOOL FOR P2P ON B2C 21

6 REVISED GENERAL B2B SCENARIOS 21

7 INTEGRATION OF P2P IN AXMEDIS FOR B2B 23

7.1 AXCP (AXMEDIS CONTENT PROCESSING) AND RELATIONSHIPS WITH THE OTHER TOOLS 24
7.2 AXMEDIS .BITTORRENT MAKER TOOL 24
7.3 AXMEDIS P2P QUERY SERVICE SERVER 25
7.4 AXMEDIS BITTORRENT TRACKER SERVER 26
7.5 AXMEDIS AXEPTOOL P2P CLIENT 30
7.6 AXMEDIS P2P NETWORK STATISTICS AND PERFORMANCES OVERVIEW 32
7.7 AXMEDIS CONTENT PROCESSING GRID 38
7.8 BITTORRENT SOLUTION PROS AND CONS 38
7.9 AXMEDIS P2P SOLUTION 39
7.10 SERVICES OF THE P2P CLIENTS TOWARD THE FACTORY 41
7.11 P2P NETWORK CONTROL EXPERIMENTS 42
7.12 INTEGRATION OF P2P IN AXMEDIA FOR C2C 43
7.13 AXMEDIS P2P CLIENT FOR MOBILE 46
7.14 QUERY INTERFACE/QUERY RESULTS INTERFACE (DSI RESP) 47
7.15 USAGE WALK-THROUGH 49
Usage walk-through for AXMEDIS Query Service and AXCP (EXITECH) 50

8 CONCLUSIONS 52

9 BIBLIOGRAPHY 52

10 GLOSSARY 54

AXMEDIS Copyright Notice

The following terms (including future possible amendments) set out the rights and obligations licensee will be requested to accept on entering into possession of any official AXMEDIS document either by downloading it from the web site or by any other means.

Any relevant AXMEDIS document includes this license. PLEASE READ THE FOLLOWING TERMS CAREFULLY AS THEY HAVE TO BE ACCEPTED PRIOR TO READING/USE OF THE DOCUMENT.

1. **DEFINITIONS**

   i. "**Acceptance Date**" is the date on which these terms and conditions for entering into possession of the document have been accepted.

   ii. "**Copyright**" stands for any content, document or portion of it that is covered by the copyright disclaimer in a Document.

   iii. "**Licensor**" is AXMEDIS Consortium as a de-facto consortium of the EC project and any of its derivations in terms of companies and/or associations, see www.axmedis.org

   iv. "**Document**" means the information contained in any electronic file, which has been published by the Licensor's as AXMEDIS official document and listed in the web site mentioned above or available by any other means.

   v. "**Works**" means any works created by the licensee, which reproduce a Document or any of its part.

2. **LICENCE**

   1. The Licensor grants a non-exclusive royalty free licence to reproduce and use the Documents subject to present terms and conditions (the **Licence**) for the parts that are own and proprietary property the of AXMEDIS consortium or its members.

   2. In consideration of the Licensor granting the Licence, licensee agrees to adhere to the following terms and conditions.

3. **TERM AND TERMINATION**

   1. Granted Licence shall commence on Acceptance Date.

   2. Granted Licence will terminate automatically if licensee fails to comply with any of the terms and conditions of this Licence.

   3. Termination of this Licence does not affect either party's accrued rights and obligations as at the date of termination.

   4. Upon termination of this Licence for whatever reason, licensee shall cease to make any use of the accessed Copyright.

   5. All provisions of this Licence, which are necessary for the interpretation or enforcement of a party's rights or obligations, shall survive termination of this Licence and shall continue in full force and effect.

   6. Notwithstanding License termination, confidentiality clauses related to any content, document or part of it as stated in the document itself will remain in force for a period of 5 years after license issue date or the period stated in the document whichever is the longer.

4. **USE**

   1. Licensee shall not breach or denigrate the integrity of the Copyright Notice and in particular shall not:

      i. remove this Copyright Notice on a Document or any of its reproduction in any form in which those may be achieved;

      ii. change or remove the title of a Document;

      iii. use all or any part of a Document as part of a specification or standard not emanating from the Licensor without the prior written consent of the Licensor; or

      iv. do or permit others to do any act or omission in relation to a Document which is contrary to the rights and obligations as stated in the present license and agreed with the Licensor

5. **COPYRIGHT NOTICES**

   1. All Works shall bear a clear notice asserting the Licensor's Copyright. The notice shall use the wording employed by the Licensor in its own copyright notice unless the Licensor otherwise instructs licensees.

6. **WARRANTY**

    1. The Licensor warrants the licensee that the present licence is issued on the basis of full Copyright ownership or re-licensing agreements granting the Licensor full licensing and enforcement power.
    2. For the avoidance of doubt the licensee should be aware that although the Copyright in the documents is given under warranty this warranty does not extend to the content of any document which may contain references or specifications or technologies that are covered by patents (also of third parties) or that refer to other standards. AXMEDIS is not responsible and does not guarantee that the information contained in the document is fully proprietary of AXMEDIS consortium and/or partners.
    3. Licensee hereby undertakes to the Licensor that he will, without prejudice to any other right of action which the Licensor may have, at all times keep the Licensor fully and effectively indemnified against all and any liability (which liability shall include, without limitation, all losses, costs, claims, expenses, demands, actions, damages, legal and other professional fees and expenses on a full indemnity basis) which the Licensor may suffer or incur as a result of, or by reason of, any breach or non-fulfilment of any of his obligations in respect of this Licence.

7. **INFRINGEMENT**

    1. Licensee undertakes to notify promptly the Licensor of any threatened or actual infringement of the Copyright which comes to licensee notice and shall, at the Licensor's request and expense, do all such things as are reasonably necessary to defend and enforce the Licensor's rights in the Copyright.

8. **GOVERNING LAW AND JURISDICTION**

    1. This Licence shall be subject to, and construed and interpreted in accordance with Italian law.
    2. The parties irrevocably submit to the exclusive jurisdiction of the Italian Courts.

Please note that:

- You can become affiliated with AXMEDIS. This will give you the access to a huge amount of knowledge, information and source code related to the AXMEDIS Framework. If you are interested please contact P. Nesi at nesi@dsi.unifi.it. Once affiliated with AXMEDIS you will have the possibility of using the AXMEDIS specification and technology for your business.
- You can contribute to the improvement of AXMEDIS documents and specification by sending the contribution to P. Nesi at nesi@dsi.unifi.it
- You can attend AXMEDIS meetings that are open to public, for additional information see WWW.axmedis.org or contact P. Nesi at nesi@dsi.unifi.it

1 Executive Summary and Report Scope

The main activities that have supported the production of this deliverable are related to:

WP4.4 – Content sharing and production on P2P.

In this deliverable we cover the following main aspects of research about how to:

- share contents in a network of B2B and C2C participants and how this content can be searched and indexed.
- deliver large objects with an efficient solution.
- provide a technical user interface for editing queries.
- integrate the P2P with the production factory of AXMEDIS

The B2B environment is a fundamental aspect which drives technical choices. Moreover, one of the aspect is the automation of content distribution over the P2P network. In fact, differently from P2P for consumer-to-consumer, in B2B we need integration with a workflow system and the possibility to make content publication and content loading as part of a pipelined process which involves content production, content searching, and content distribution.

Main differences with respect to the past version (M24 version) of this deliverable are:

- Detailed description of the AXTracker, section 7.4
- Detailed description of the AXEPTool P2P client, section 7.5
- Possible network statistics that can be performed and related results Section 7.6
- Detailed description of the relationships of control of the AXCP on the P2P network in section 7.7
- Critical comments on the BitTorrent solution taken, section 7.8
- Overview of the AXMEDIS P2P solution, section 7.9
- P2P network control experiments
- AXMEDIA P2P client tool
- Study on the P2P for mobiles
- conclusions

-

2 Introduction

AXMEDIS will pursue an integrated solution to content distribution, providing flexibility and scalability to support any kind of content over any kind of network, and configurable to support terrestrial, satellite and mobile transport protocols.

The activities that have supported the production of this deliverable are related to the analysis of content sharing and distribution on P2P. The main topics presented and analyzed in this document are about:

- Query User Interfaces for B2B into P2P solution, Searching contents in a network of business peers
- Downloading content with state-of-the-art P2P protocols
- Automation of publication and loading of contents, connection with the AXMEDIS database
- Metadata mapping when the content coming from other AXMEDIS factories have to be loaded and they could present different mapping of metadata.
- Integration of security and DRM aspects into P2P solutions
- Integration of P2P network for both B2B and C2C

2.1 T4.4.1 AXEPTool, P2P tool for B2B transaction and content distribution (DSI)

The architecture will integrate the customisation of the BitTorrent P2P solution to AXMEDIS, including content verification by fingerprint, support for complete technical queries, DRM control of the certification of authority, tracking content exploitation, rights clearance, optimisation of file downloads, connection with the AXCP for automating publication and downloading, etc. According to the requirements and performed analysis of P2P state of the art, and past experiments performed by CRS4, AXMEDIS consortium decided to realise for the P2P Networks of AXMEDIS an architecture based on BitTorrent. The study and the definition of the new architecture has been performed by DSI in the M20-M22 and reported in the specification document DE3.1.2.2.10 (ver.: 1.7). The realization of the prototype will be performed in this period recovering the present delay provoked by CRS4. The work will consists mainly in: customization of the Azureus P2P client to work with AXCP, AXOID and AXMEDIS objects, integration in AXMEDIS Azureus (alias AXEPTool) the web services for the communication with AXCP tools, customisation of the bittorrent tracker (access to statistics and other information, control the access, certifications), test of the tools, validation, installation in several locations, performance analysis.

- M30: control of the access of non AXMEDIS objects into the P2P network will be defined and implemented into P2P client and Tracker Server sides. In this phase, the P2P network will be functioning and the first test will be performed; Integration of the Query Interface and the Publication Object Interface will be ready in the AXEPTool/AXMEDIA tools; AXEPTool and AXMEDIA tool packages will be realized in first version and installation packages; first release of the Monitoring Web Service and tools with at least the downloading functionalities. Realisation of the first version of the Web service clients to access at the Web Services of the AXEPTool (Azureus AXMEDIS P2P client) according to the specification document;
- M36: first test and verification of the most relevant functionalities for the P2P. massive test and verification of all the needed functionalities for the P2P tools. AXEPTool and AXMEDIA tool packages will be completed with their final installation packages, and related documentation. Final version of the Web service clients to access at the Web Services of the AXEPTool (Azureus AXMEDIS P2P client) according to the specification;
- M36: final version of the AXEPTool based on bittorrent solution and performance analysis for the P2P network and for the aspects related to the AXCP integration with the P2P network.

2.2 T4.4.2: The query support into the AXEPTool database (EXITECH)

The research in this task will consist in the study and realization of an efficient B2B and B2C query support for the Bittorent-based P2P networks of AXMEDIS (AXEPTool and AXMEDIA based solutions). The Query Support in the AXEPTool will allow the specification of technical/professional query including metadata, technical information, business and licensing aspects, content based, DRM rules, etc. Studying a specific user interface integrated into the AXEPTool for defining technical queries and it is integrated into WP4.4 Complex technical queries as described in the WP4.2, including query for complex objects and single components, reusable components. Technical queries are those of other AXMEDIS tools for B2B. The evolution of the P2P network based on bitTorrent may see the whole network managed by clusters controlled by an AXMEDIS P2P Query Service Server (including a web portal for giving access at users to make queries and a Web Service to give the possibility at other application to perform queries). Different clusters and thus of AXMEDIS P2P Query Service Server can communicate each other in order to propagate the information and the queries.

- M30: realization of the first version of the AXMEDIS P2P Query Service Server without sophisticated user interface and information migration among clusters;
- M36: realization of the improved version of the AXMEDIS P2P Query Service Server with a more sophisticated user interface and information migration (queries and results) among clusters (AXMEDIS P2P Query Service Server), early version of the performance analysis;
- M36: final version of the AXMEDIS P2P Query Service Serve, with final version of performance analysis.

3 Performed and planned Activity

The work on this WP has been in charge to CRS4 that abandoned the consortium without completing the work and without leaving a functional solution for the P2P.

To this end, the AXMEDIS consortium has reallocated the activity to other partners and it is going to assign a subcontract. The Consortium is waiting from the EC the start of the reallocated tasks. So that the activity in this area is waiting for several months to be restarted. Despite this fact, the AXMEDIS consortium has performed all its best to recover a the delay left by CRS4.

The activities performed an planned by AXMEDIS consortium are related to:

- Downloading content with state-of-the-art P2P protocols, performed and reported in the DE4.4.1 and in the following with an addition on the BitTorrent Techology (DSI)
- Definition of the P2P architecture for B2B and C2C (DSI, already performed as visible into the DE3-1-2-2-10-Spec-of-AXEPTool-and-AXMEDIA-tools document)
- Query User Interfaces for B2B into P2P solution, Searching contents in a network of business peers (EXITECH, largely already implemented, missing aspects are those related to the integration of different clusters of P2P)
- Automation of publication and loading of contents, connection with the AXMEDIS database (DSI, partially realised as AXCP elements, Publisher and Loader, to be completed in the next period)
- Metadata mapping when the content coming from other AXMEDIS factories have to be loaded and they could present different mapping of metadata (UNIVLEEDS, see metadata mapper, and related JS metadata mapper to be realised and based on XSLT).
- Integration of security and DRM aspects into P2P solutions, includes authentication, certification, the establishing of a protected channel from P2P client to the Query Server and from the P2P client to the BitTorrent Tracker, (FUPF, DSI, to be realised in the next period)
- Integration of P2P network for both B2B and C2C (DSI and its subcontract, to be realised in the next period)
- Customisation of the AZUREUS P2P client to become AXEPTool (DSI with its subcontract, to be realised in the next period)
- Customisation of the AZUREUS P2P client to become AXMEDIA tool (DSI with its subcontract, to be realised in the next period)
- Customization of a bitTorrent Tracker to become a reference Tracker for AXMEDIS P2P networks (DSI, to be realised in the next period)
- Analysis of performance, optimisation of solution, integration of B2B and C2C area for P2P
- Connection and harmonisation of different P2P clusters.

4 State of the art of P2P BitTorrent

Largely extracted from WEB pages and from Wikipedia: www.wikipedia.org. On the other hand, even if the following information could be accessible on the WIKIPEDIA, due to the instability/evolution of that WEB site we preferred to include and instance of that information in the document.

BitTorrent is the name of a peer-to-peer (P2P) file distribution client application and also of its related file sharing protocol, both of which were created by programmer Bram Cohen. BitTorrent is designed to distribute large amounts of data widely without incurring the corresponding consumption in costly server and bandwidth resources. CacheLogic suggests that BitTorrent traffic accounts for ~35% of all traffic on the Internet while other sources are skeptical.
The original BitTorrent application was written in Python. Its source code, as of version 4.0, has been released under the BitTorrent Open Source License, which is a modified version of the Jabber Open Source License. There are numerous compatible clients, written in a variety of programming languages, and running on a variety of computing platforms.
BitTorrent clients are programs which implement the BitTorrent protocol. Each BitTorrent client is capable of preparing, requesting, and transmitting any type of computer file over a network using the BitTorrent protocol. This includes text, audio, video, encrypted content, and other types of digital information.

Creating and publishing torrents
To share a file or group of files through BitTorrent, clients first create a ".torrent" information file. This is a small file which contains meta information about the files to be shared, and about the host computer that coordinates the file distribution (http://www.bittorrent.com). The exact information contained in the tracker file depends on the version of the BitTorrent protocol. However, a torrent file always has the extension .torrent. Torrent files contain an "announce" section, which specifies the URL of the tracker, and an "info" section which contains (suggested) names for the files, their lengths, the piece length used, and a SHA-1 hash code for each piece, which clients should use to verify the integrity of the data they receive.

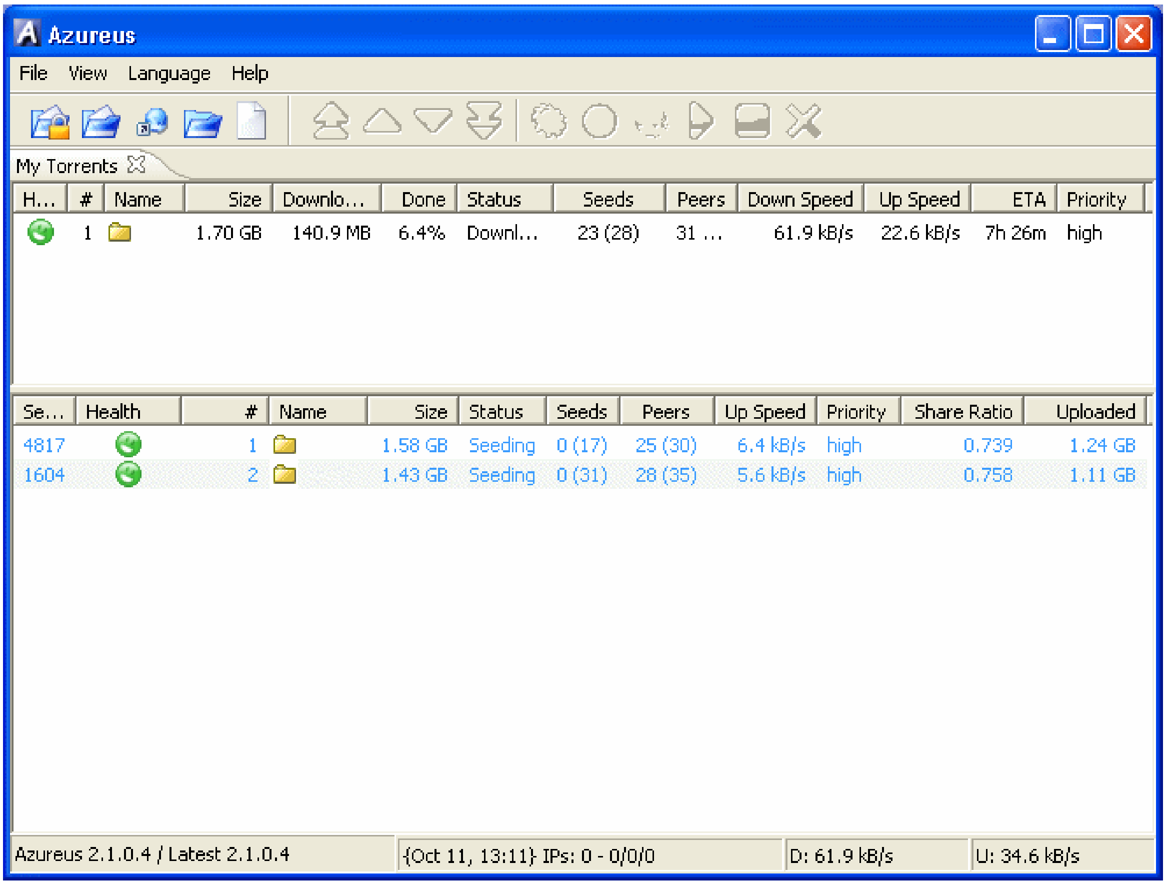
Clients who have finished downloading the file may also choose to act as seeders, providing a complete copy of the file. After the torrent file is created, a link to it is placed on a website or elsewhere, and it is registered with a tracker. BitTorrent trackers maintain lists of the clients currently participating in the torrent. The computer with the initial copy of the file is referred to as the initial seeder.

Downloading torrents and sharing files
Using a web browser, users navigate to the site listing the torrent, download it, and open it in a BitTorrent client. After opening the torrent, the BitTorrent client connects to the tracker, which provides it with a list of clients currently downloading the file or files. A group of peers on a BitTorrent or P2P connected with each other to share a particular torrent is generally referred to as a swarm.

Initially, there may be no other peers in the swarm, in which case the client connects directly to the initial seeder and begins to request pieces. The BitTorrent protocol breaks down files into a number of much smaller pieces, typically a quarter of a megabyte (256 KB) in size. Larger file sizes typically have larger pieces. For example, a 4.37 GB file may have a piece size of 4 MB (4096 KB). Pieces are checked as they are received using a hash algorithm to ensure that they are error free.

As peers enter the swarm, they begin sharing pieces with one another, instead of downloading directly from the seeder. Clients incorporate mechanisms to optimize their download and upload rates, for example using a tit for tat scheme. Peers download pieces in a random order, to increase the opportunity to exchange data, which is only possible if two peers have a different subset of the file.

The effectiveness of the peer-to-peer data exchange depends largely on the policies used by clients to determine whom to send data to. Clients will prefer to send data to peers that send data back to them, which encourages fair sharing, but strict policies often result in suboptimal situations, where newly joined peers are unable to receive any data (because they don't have any pieces yet to share themselves) and two peers with a good connection between them do not exchange data simply because neither of them wants to take the initiative. To counter these effects, the official BitTorrent client uses a mechanism called “optimistic unchoking”, where the client will reserve a portion of its available bandwidth for sending pieces to random peers (not necessarily known-good partners, so called preferred peers), in hopes of discovering even better partners and to ensure newcomers get a chance to join the swarm.

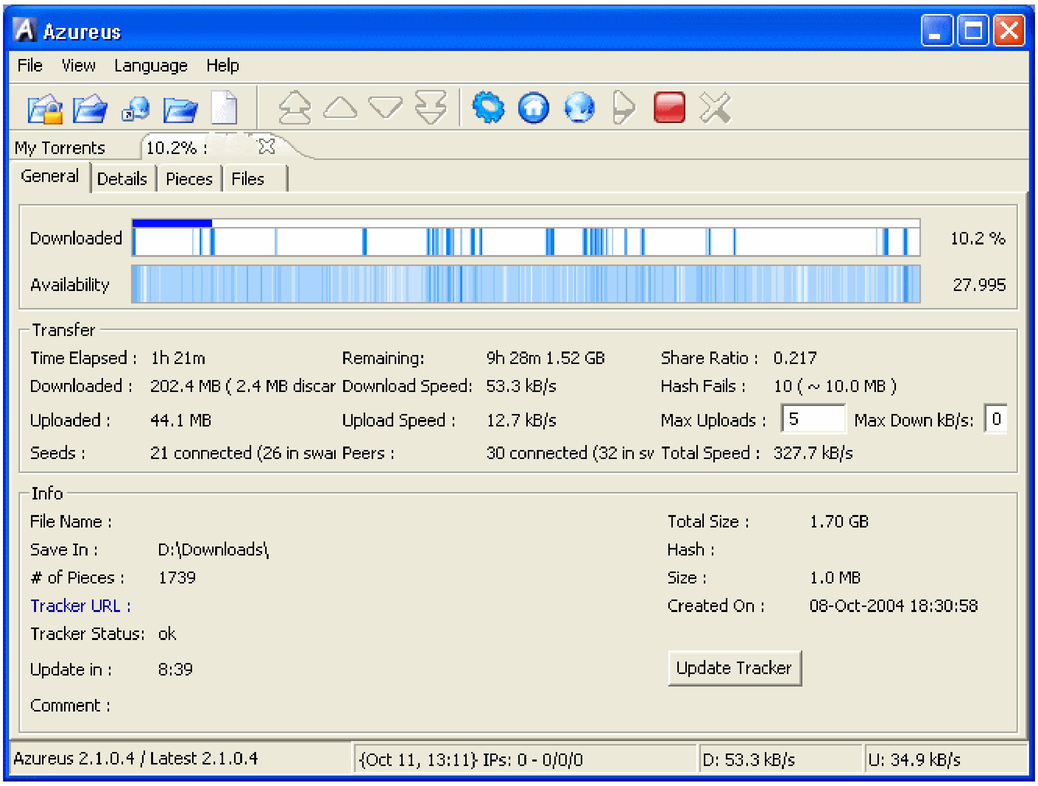


H...	#	Name	Size	Downlo...	Done	Status	Seeds	Peers	Down Speed	Up Speed	ETA	Priority
●	1		1.70 GB	140.9 MB	6.4%	Downl...	23 (28)	31 ...	61.9 kB/s	22.6 kB/s	7h 26m	high

Se...	Health	#	Name	Size	Status	Seeds	Peers	Up Speed	Priority	Share Ratio	Uploaded
4817	●	1		1.58 GB	Seeding	0 (17)	25 (30)	6.4 kB/s	high	0.739	1.24 GB
1604	●	2		1.43 GB	Seeding	0 (31)	28 (35)	5.6 kB/s	high	0.758	1.11 GB

Azureus 2.1.0.4 / Latest 2.1.0.4 {Oct 11, 13:11} IPs: 0 - 0/0/0 D: 61.9 kB/s U: 34.6 kB/s

In Azureus bittorrent client, while a file is downloading, right click and click **Show Details**. As you can see I've blocked out some information on mine as I don't want to go requesting permission from the tracker I was using before I post this guide. Therefore I have removed the **filename, Tracker URL, Hash** and the file **Comment**. On your screen you will see these details. Certain areas are important here like **Share Ratio**. As you can see my Share Ratio for this file is just 0.217. This is a bad ratio but expected while the file is still downloading. When the file finishes downloading, users are expected to keep the seeding going until the ratio goes over **1.000**. You will notice the **Update Tracker** button. Clients have to update the tracker on the parts of files they have and other such things, this button is just here in case you wish to update immediately, however if you look across the way, you will that there is already a countdown to next update. When you are seeding, there is a slightly different detail as you will see now.



Downloaded 10.2 %
Availability 27.995

Transfer

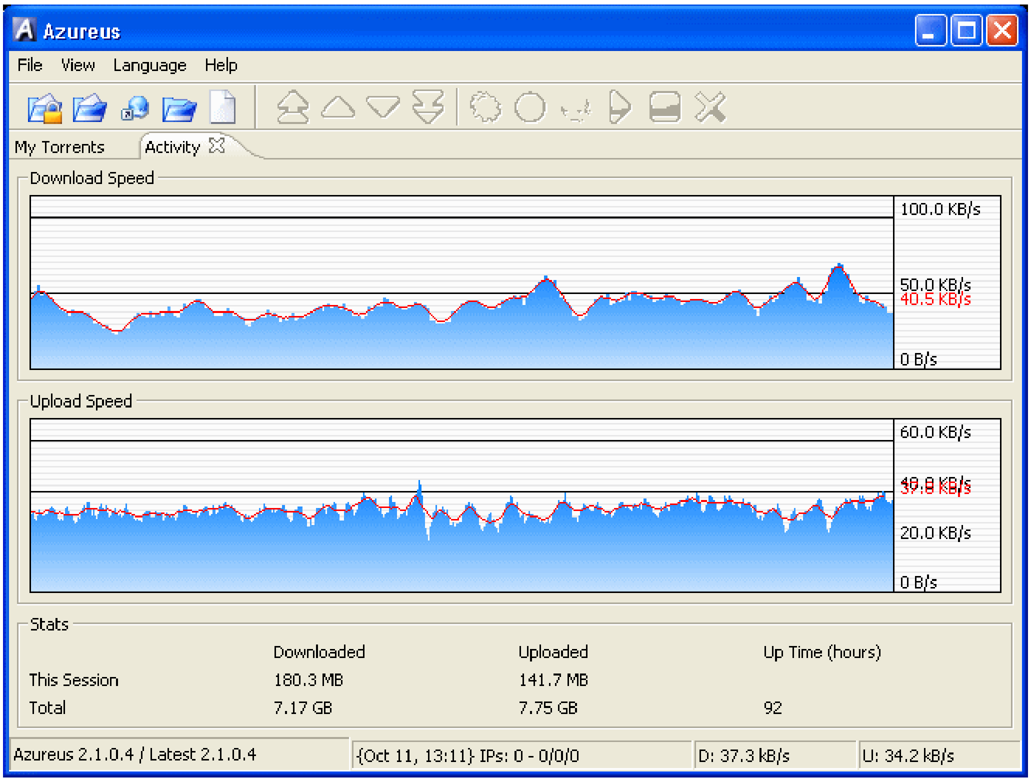
Time Elapsed :	1h 21m	Remaining:	9h 28m 1.52 GB	Share Ratio :	0.217
Downloaded :	202.4 MB (2.4 MB discar	Download Speed:	53.3 kB/s	Hash Fails :	10 (~ 10.0 MB)
Uploaded :	44.1 MB	Upload Speed :	12.7 kB/s	Max Uploads :	5 Max Down kB/s: 0
Seeds :	21 connected (26 in swa	Peers :	30 connected (32 in sv	Total Speed :	327.7 kB/s

Info

File Name :		Total Size :	1.70 GB
Save In :	D:\Downloads\	Hash :	
# of Pieces :	1739	Size :	1.0 MB
Tracker URL :		Created On :	08-Oct-2004 18:30:58
Tracker Status:	ok		
Update in :	8:39		
Comment :			

Azureus 2.1.0.4 / Latest 2.1.0.4 | {Oct 11, 13:11} IPs: 0 - 0/0/0 | D: 53.3 kB/s | U: 34.9 kB/s

In Azureus bittorrent client, if you click **View --> Statistics**, you get some nice information on your upload and download bandwidth and how stable it has been. As you can see it also shows you your overall downloaded data since you began using Azureus, your uptime and the bandwidth transfers for this session.

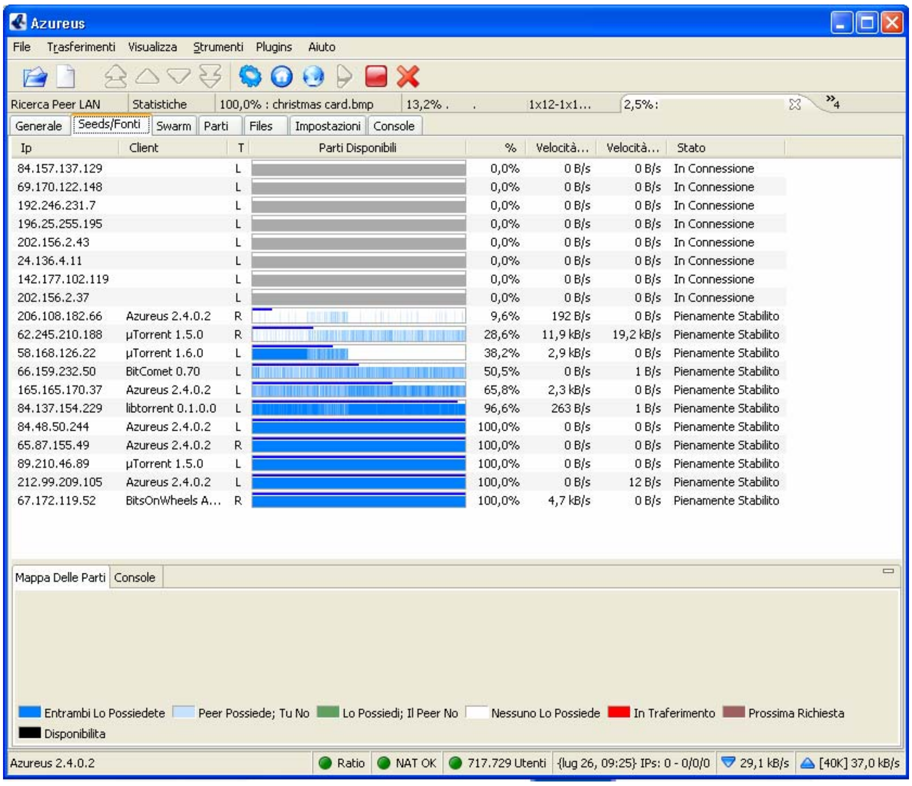


Stats

	Downloaded	Uploaded	Up Time (hours)
This Session	180.3 MB	141.7 MB	
Total	7.17 GB	7.75 GB	92

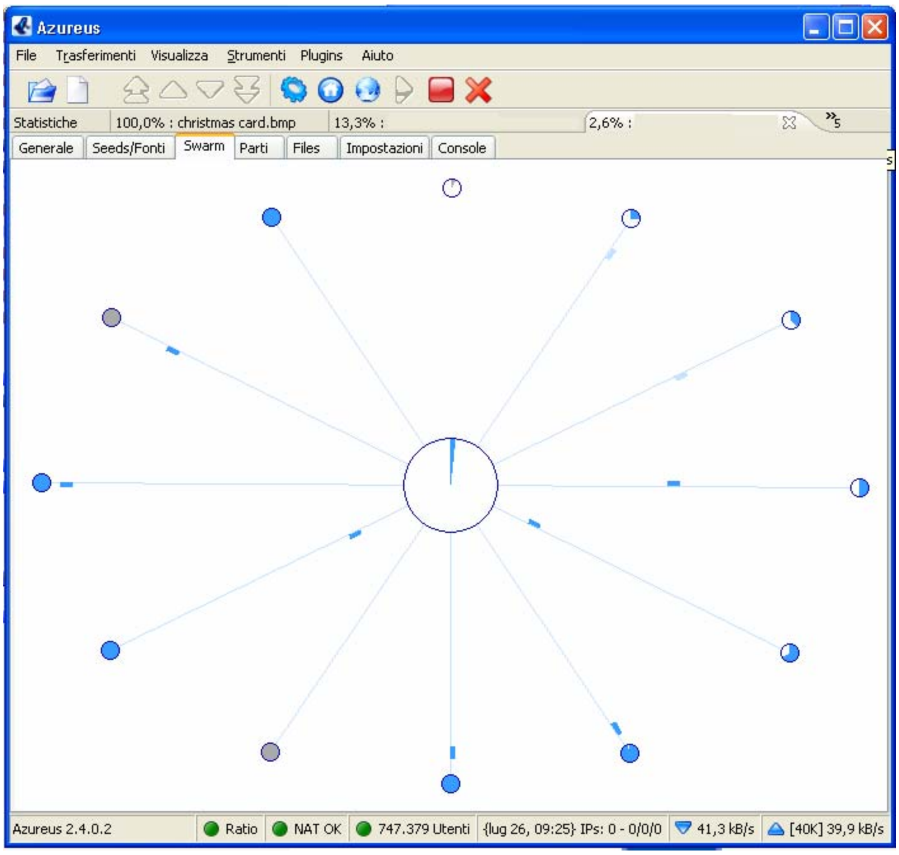
Azureus 2.1.0.4 / Latest 2.1.0.4 | {Oct 11, 13:11} IPs: 0 - 0/0/0 | D: 37.3 kB/s | U: 34.2 kB/s

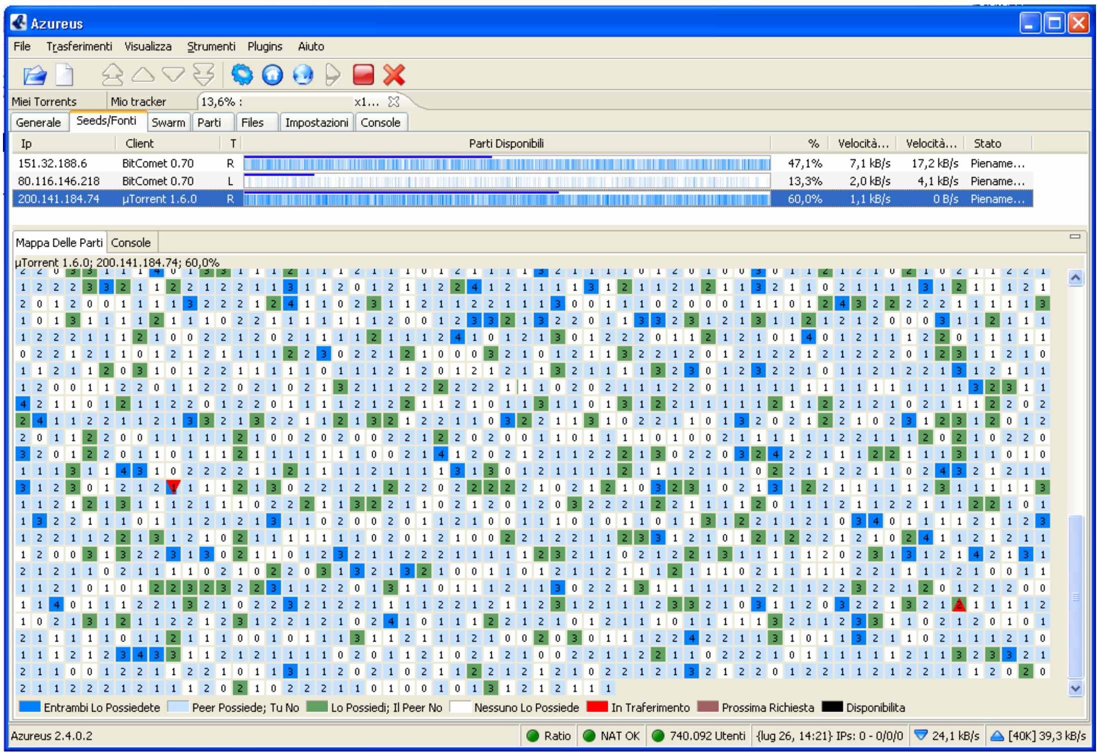
In the following, different representations of the status of the download are reported. The first one report the fonts with the rate of transfer for each of them. The second (the swarm) is a picture of the fonts with a simpler and immediate status of their capability. The latter is the status of the transfer for a given file in which the status of each segment of the file is marked with a different color depending if the file segment is accessible, present, under download, etc.



Ip	Client	T	Parti Disponibili	%	Velocità...	Velocità...	Stato
84.157.137.129		L		0,0%	0 B/s	0 B/s	In Connessione
69.170.122.148		L		0,0%	0 B/s	0 B/s	In Connessione
192.246.231.7		L		0,0%	0 B/s	0 B/s	In Connessione
196.25.255.195		L		0,0%	0 B/s	0 B/s	In Connessione
202.156.2.43		L		0,0%	0 B/s	0 B/s	In Connessione
24.136.4.11		L		0,0%	0 B/s	0 B/s	In Connessione
142.177.102.119		L		0,0%	0 B/s	0 B/s	In Connessione
202.156.2.37		L		0,0%	0 B/s	0 B/s	In Connessione
206.108.182.66	Azureus 2.4.0.2	R		9,6%	192 B/s	0 B/s	Pienamente Stabilito
62.245.210.188	µTorrent 1.5.0	R		28,6%	11,9 kB/s	19,2 kB/s	Pienamente Stabilito
58.168.126.22	µTorrent 1.6.0	L		38,2%	2,9 kB/s	0 B/s	Pienamente Stabilito
66.159.232.50	BitComet 0.70	L		50,5%	0 B/s	1 B/s	Pienamente Stabilito
165.165.170.37	Azureus 2.4.0.2	L		65,8%	2,3 kB/s	0 B/s	Pienamente Stabilito
84.137.154.229	libtorrent 0.1.0.0	L		96,6%	263 B/s	1 B/s	Pienamente Stabilito
84.48.50.244	Azureus 2.4.0.2	L		100,0%	0 B/s	0 B/s	Pienamente Stabilito
65.87.155.49	Azureus 2.4.0.2	R		100,0%	0 B/s	0 B/s	Pienamente Stabilito
89.210.46.89	µTorrent 1.5.0	L		100,0%	0 B/s	0 B/s	Pienamente Stabilito
212.99.209.105	Azureus 2.4.0.2	L		100,0%	0 B/s	12 B/s	Pienamente Stabilito
67.172.119.52	BitsOnWheels A...	R		100,0%	4,7 kB/s	0 B/s	Pienamente Stabilito

🟦 Entrambi Lo Possiedete ▭ Peer Possiede; Tu No 🟩 Lo Possiedi; Il Peer No □ Nessuno Lo Possiede 🟥 In Traferimento ▨ Prossima Richiesta
■ Disponibilita





Ip	Client	T	Parti Disponibili	%	Velocità...	Velocità...	Stato
151.32.188.6	BitComet 0.70	R		47,1%	7,1 kB/s	17,2 kB/s	Piename...
80.116.146.218	BitComet 0.70	L		13,3%	2,0 kB/s	4,1 kB/s	Piename...
200.141.184.74	µTorrent 1.6.0	R		60,0%	1,1 kB/s	0 B/s	Piename...

µTorrent 1.6.0; 200.141.184.74; 60,0%

Azureus 2.4.0.2 | Ratio | NAT OK | 740.092 Utenti | {lug 26, 14:21} IPs: 0 - 0/0/0 | 24,1 kB/s | [40K] 39,3 kB/s

The importance of Uploading

I had to write a small bit about uploading. Uploading on BitTorrent is vital. If you connect to a torrent that has just 3 seeds and 800 peers, then most of the sharing will be done between peers. If you download the whole file and have uploaded just 10% of that file and then leave, you are hurting the performance of that torrent. This kind of usage is very bad because if a lot of people begun doing it, then there would be very little seeds and eventually the seeds could disappear and there may be nobody left with 100% of the file. The full file still may be available as files are traded in small pieces, but if all users stopped uploading as much as they downloaded, torrents life wouldn't last long and when it was fully working, it would be very slow. Always make sure you upload as much as you download if not more. Someone who download 700MB and uploads 700MB still in the eyes of BitTorrent is not sharing properly. For the life of a torrent to go on for a long time at high speeds, all users should make sure they upload at least 150% of what they downloaded. When your torrents are done downloading keep them active until you reach this amount, or if you need to use the files, you can stop the torrent activity, use them (but do not alter them) and then click the same torrent again and you would be added to the tracker as a seed and would continue uploading again. Trackers has begun banning leechers, or keeping records of leechers as a way to deter their behavior, if you like BitTorrent, don't try and cheat it.

Limitations

BitTorrent does not offer its users anonymity. It is possible to obtain the IP addresses of all current, and possibly previous, participants in a swarm from the tracker. This may expose users with insecure systems to attacks to discover their identity. This is not a problem in a legal service for content sharing.

Another drawback is that BitTorrent file sharers compared to users of client/server technology often have little incentive to become seeders after they finish downloading. The result of this is that torrent swarms gradually die out, meaning a lower possibility of obtaining older torrents. Some BitTorrent websites have attempted to address this by recording each user's download and upload ratio for all or just the user to see, as well as the provision of access to older torrent files to people with better ratios.

BitTorrent is typically best suited in continuously connected broadband environments. Dial-up users find it less efficient due to frequent disconnects and slow download rates.

4.1 BitTorrent Terminology

- **Availability**: (also distributed copies) The number of full copies of the file available to the client. Each seed adds 1.0 to this number, as they have one complete copy of the file. A connected peer with a fraction of the file available adds that fraction to the availability, if no other peer has this part of the file. (ie. a peer with 65.3% of the file downloaded increases the availability by 0.653. However, if two peers both have the same portion of the file downloaded - say 50% - and there is only one seeder, the availability is 1.5).
- **Choked**: Describes a peer to whom the client refuses to send file pieces. A client chokes another client in several situations: The second client is a seed, in which case it does not want any pieces (ie. it is completely uninterested) The client is already uploading at its full capacity (ie. the value for max_uploads has been reached)
- **interested** Describes a downloader who wishes to obtain pieces of a file the client has. For example, the uploading client would flag a downloading client as 'interested' if that client did not possess a piece that it did, and wished to obtain it.
- **leech** A leech is usually a peer who has a negative effect on the swarm by having a very poor share ratio - in other words, downloading much more than they upload. Most leeches are users on asymmetric internet connections and do not leave their BitTorrent client open to seed the file after their download has completed. However, some leeches intentionally avoid uploading by using modified clients or excessively limiting their upload speed. The term leech, however, can be used simply to describe a peer - or any client that does not have 100% of the data.
- **peer** A peer is one instance of a BitTorrent client running on a computer on the Internet to which other clients connect and transfer data. Usually a peer does not have the complete file, but only parts of it. However, in the colloquial definition, "peer" can be used to refer to any participant in the swarm (in this case, it's synonymous with "client").
- **scrape** This is when a client sends a request to the tracking server for information about the statistics of the torrent, such as with whom to share the file and how well those other users are sharing.
- **seeder** A seeder is a peer that has a complete copy of the torrent and still offers it for upload. The more seeders there are, the better the chances are for completion of the file.
- **snubbed** An uploading client is flagged as snubbed if the downloading client has not received any data from it in over 60 seconds.
- **superseed** When a file is new, much time can be wasted because the seeding client might send the same file piece to many different peers, while other pieces have not yet been downloaded at all. Some clients, like ABC, Azureus, BitTornado, TorrentStorm, and µTorrent have a "superseed" mode, where they try to only send out pieces that have never been sent out before, making the initial propagation of the file much faster. This is generally used only for a new torrent, or one which must be re-seeded because no other seeds are available.
- **swarm** Together, all peers (including seeders) sharing a torrent are called a swarm. For example, six ordinary peers and two seeders make a swarm of eight.
- **torrent** A torrent can mean either a .torrent metadata file or all files described by it, depending on context. The torrent file contains metadata about all the files it makes downloadable, including their names and sizes and checksums of all pieces in the torrent. It also contains the address of a tracker that coordinates communication between the peers in the swarm.
- **tracker** A tracker is a server that keeps track of which seeds and peers are in the swarm. Clients report information to the tracker periodically and in exchange receive information about other clients to which they can connect. The tracker is not directly involved in the data transfer and does not have a copy of the file.

4.2 General Information on BitTorrent Tracker

A BitTorrent tracker is a server which assists in the communication between peers using the BitTorrent protocol. It is also, in the absence of extensions to the original protocol, the only major critical point, as clients are required to communicate with the tracker to initiate downloads. (Clients that have already begun downloading also communicate with the tracker periodically to negotiate with newer peers and provide statistics, however, after the initial reception of peer data, peer communication can continue without a tracker.)

A tracker should be differentiated from a BitTorrent index by the fact that it does not necessarily list files that are being tracked. A BitTorrent index is a list of .torrent files (usually including descriptions and other information). Trackers merely coordinate communication between peers attempting to download the payload of the torrents.

Many BitTorrent websites act as both tracker and index. Sites such as these publicize the tracker's URL and allow users to upload torrents to the index with the tracker's URL embedded in them, providing all the features necessary to initiate a download.

Trackers are the primary reason for a damaged BitTorrent 'swarm'. (Other reasons are mostly related to damaged or hacked clients uploading corrupt data.) The reliability of trackers has been improved through two main innovations in the BitTorrent protocol:

Multi-tracker torrents feature multiple trackers in the one torrent. This way, should one tracker fail, the others can continue supporting file transfer.

There are two incompatible 'trackerless' BitTorrent transfer (aka. decentralized tracking) methods: DHT-based implementations, and Azureus's 'Distributed Database'.

The term 'trackerless' is something of a misnomer, as decentralized or distributed tracking essentially treats every peer in the swarm as a tracker. Original BitTorrent was the first client to offer decentralized tracking through its DHT method. Later, Azureus, µTorrent and BitComet adopted this feature, although Azureus's method of implementation is incompatible with the DHT offered by all other supporting clients.

4.2.1 Lists of Trackers

C/C++ Trackers

- BitCometTracker - C++(?), Windows Only
- Extended BitTorrent client and Tracker (XBTT) - Windows/Linux

BNBT Based

- BNBT - Port of the original Python BT tracker with many additional features
- BNBT Trinity Edition - A modified version of BNBT with a windows installer
- CBTT - Another tracker written in C++, based on BNBT
- =Xotic= Edition of BNBT - Based on BNBT with extra features; focused on Linux users
- BNBT - This is a C++ implementation of a BitTorrent tracker. It should compile under most any Unix with GCC available, as well as MS Windows with MSVC (binaries included.) It includes all of the functionality of the reference Python tracker, but it also includes many enhancements: user accounts, improved web interface, statistics, etc. See also the TrackPak for a bundled BNBT and installer that's easy to use.

PHP Trackers

- Blog Torrent - Supports webseed, does not require MySQL(?). ABANDONED
- Broadcast Machine - Successor to Blog Torrent, Focuses on Torrentcasts
- BT phpTracker Plus - Coded from scratch; lightweight
- Btittracker - A frontend for phpBTTracker
- ByteStats Tracker - External scrape support, installation script. ABANDONED
- PHP Nuke BitTorrent Module - BitTorrent tracker module for the popular PHPNuke CMS
- phpMyBitTorrent - Successor to the PHPNuke Tracker Module, this is a standalone tracker with many features

- BTChange 0.94a - For modifying tracker info in an existing .torrent file. Use this if the tracker changes, so that you don't have to recreate the file. See also: Sourceforge page.

PHPBTTRACKER - Installer, MySQL. ABANDONED

- PHPBTTRACKER+ - Tracker based on PHPBTTRACKER. Many extra features
- PHPBTTracker-Admin - Another tracker based on PHPBTTRACKER, also with many additional features.
- PHPBTTracker - a free (GPL) tracker implementation in PHP with a MySQL back-end, includes built-in statistics collection and reporting.

4.3 Servers for listing and searching Torrent Files

They are servers that allow you to make some query. There are many sites that list torrent files. To use these sites, all you would have to do is click on a file listed, and it should download automatically and your client should immediately connect to the tracker and start the downloading. Some known sites that list torrent files are. Sites that list torrent files (like suprnova.org) have their trackers to handle the downloading and sharing that the users are doing. A torrent file would have the details on the tracker so you can only use a torrent file on the tracker it was made for. Software exists for anybody to setup their own trackers and build their own torrent files.

- http://home.quicknet.nl/qn/prive/romeria/bittorrentsites.htm list of BitTorrent trackers, servers with list of files, server in which it is possible to search for BitTorrent file.
- Suprnova.org - (Only works with original BT client)
- Torrentbits.org
- TorrentReactor.com
- FileList.org - (Requires registration and read FAQ)

4.4 Some additional issues of BitTorrent

4.4.1 Alternative approaches

The BitTorrent protocol provides no way to index torrent files. As a result, a comparatively small number of websites have hosted the large majority of torrents linking to copyright material, rendering those sites especially vulnerable to lawsuits. In response, some developers have sought ways to make publishing of files more anonymous while still retaining BitTorrent's speed advantage. The Shareaza client, for example, provides three alternatives to BitTorrent: eDonkey2000, Gnutella, and Shareaza's native network, Gnutella2. If the tracker is down, it can finish the file over the other protocols, and/or find new (Shareaza) peers over G2. The use of distributed trackers is also one of the goals for Azureus 2.3.0.2 and BitTorrent 4.1.2. Another interesting idea that has surfaced recently in Azureus is virtual torrent. This idea is based on the distributed tracker approach and is used to describe some web resource. Right now, it is used for instant messaging. It is implemented using a special messaging protocol and requires an appropriate plugin. Anatomic P2P is another approach, which uses a decentralized network of nodes that route traffic to dynamic trackers.

4.4.2 Legal defenses

There are two major differences between BitTorrent and many other peer-to-peer file-trading systems, which advocates suggest make it less useful to those sharing copyrighted material without authorization. First, BitTorrent itself does not offer a search facility to find files by name. A user must find the initial torrent file by other means, such as a web search. Second, BitTorrent makes no attempt to conceal the host ultimately responsible for facilitating the sharing: a person who wishes to make a file available must run a tracker on a specific host or hosts and distribute the tracker address(es) in the .torrent file. While it is possible to simply operate a tracker on a server that is located where the copyright holder cannot take legal action, this feature of the protocol does imply some degree of vulnerability that other protocols lack. It is far easier to request that the server's ISP shut the site down than it is to find and identify every user sharing a file on a traditional peer-to-peer network. However, with the use of a distributed hash table (DHT), a tracker is no longer required, although they are often still used so that clients that do not support DHT can still connect to the swarm.

4.5 Other features of BitTorrent

4.5.1 Utilities

- TorrentSpy - An MS Windows tool which allows you to query a tracker about a specific torrent, view metadata info, check a file's hashes, etc. A new feature is the "create" tab for making torrent files to upload.
- maketorrent - A utility for creating torrent files, by the author of the burst! client. Version 1.x of MakeTorrent was a modified version of the Python 'completedir' program with extra features. Version 2.x is a complete rewrite in Delphi.
- completedir 1.0.1 - A utility for creating new .torrent files, part of the official BitTorrent client package. This is packaged as a Windows installer, get the source code for use with other platforms.
- Java BitTorrent Tools - A metafile viewer/editor, availability checker, and a basic tracker.
- DumpTorrentCGI - Use this handy web page to parse and output the contents of a .torrent file (from either your local system or a link URL.) Use this to quickly determine a torrent's hash ID or which tracker is hosting it.
- trackerlyze.pl 1.11 - A free (GPL) Perl script that analyzes the logfile of a tracker and creates graphs and reports of the bandwidth used and number of peers/seeds. See also: Home page.
- libbt - A library implementation of the BitTorrent protocol in C. This project is still under development, and is not suitable to end-users at this point.

4.5.2 Broadcatching

Another proposed feature combines RSS and BitTorrent to create a content delivery system dubbed broadcatching. Since a Steve Gillmor column for Ziff-Davis in December 2003, the discussion has spread quickly among many bloggers (Techdirt, Ernest Miller, and former TechTV host Chris Pirillo, for example). As Scott Raymond explained:

"I want RSS feeds of BitTorrent files. A script would periodically check the feed for new items, and use them to start the download. Then, I could find a trusted publisher of an Alias RSS feed, and 'subscribe' to all new episodes of the show, which would then start downloading automatically — like the 'season pass' feature of the TiVo."
While potential illegal uses abound as is the case with any new distribution method, this idea lends itself to a great number of ideas that could turn traditional distribution models on their heads, giving smaller operations a new opportunity for content distribution. The system leans on the cost-saving benefit of BitTorrent, where expenses are virtually non-existent; each downloader of a file participates in a portion of the distribution. One early adoption of this concept is IPTV show mariposaHD, which uses BitTorrent to distribute large (1-2 GB) WMVHD files of high-definition video.

RSS feeds layered on top keep track of the content, and because BitTorrent does cryptographic hashing of all data, subscribers to the feed can be sure they're getting what they think they're getting, whether that winds up being the latest Sopranos episode, or the latest Sveasoft firmware upgrade. (Naturally, however, ensuring that the same data reaches all nodes neglects the possibility that the original, source file may be corrupted or incorrectly labeled.)

One of the first open source attempts to create a client specifically for this was Democracy Player. The idea is already gaining momentum however, with other Free Software clients such as PenguinTV and KatchTV also now supporting broadcatching.

4.5.3 APIs

The BitTorrent web-service Prodigem has made available a feature to any web application capable of parsing XML through its standard Representational State Transfer (REST) based interface. Additionally, Torrenthut is developing a similar torrent API which will provide the same features, as well as further intuition to help bring the torrent community to Web 2.0 standards. Alongside this release is a first PHP application built using the API called PEP which will parse any Really Simple Syndication (RSS 2.0) feed and automatically create and seed a torrent for each enclosure found in that feed.

4.5.4 Multitracker

Another unofficial feature is an extension to the BitTorrent metadata format proposed by John Hoffman. It allows the use of multiple trackers per file, so if one tracker fails, others can continue supporting file transfer. It is implemented in several clients, such as BitTornado and µTorrent. Trackers are placed in groups, or tiers, with a tracker randomly chosen from the top tier and tried, moving to the next tier if all the trackers in the top tier fail.

Relevant Features and solutions matching the interests and requirements of AXMEDIS:

client	FLOSS	Runs on Linux/Unix	Runs on Windows	Runs on Mac OS X	Max active torrents	Supports Super-seeding	Tracker	Malware-free	Supports UPnP Port Mapping [1]	Supports NAT Port Mapping Protocol	Supports NAT traversal	Supports DHT [2]	Supports BitTorrent protocol encryption	Programming Language	Basis	Interface	Supports Web Seeding [14]	Supports Broadcatching (RSS)	Supports Prioritization	Supports Selective Downloading	Supports SOCKS for outgoing connections	Web Remote Control	Torrent Search Engine
Arctic Torrent	Yes	No	Yes	No	∞	No	No	Yes	No	?	No	No	No	C++	libtorrent	GUI	No	No	No	No	No	No	No
Azureus	Yes	Yes	Yes	Yes	∞	Yes	Yes	Yes	Yes	?	DHT only	Yes [3]	Yes	Java and SWT	-	GUI, CLI, Telnet, Web, XML over HTTP remote control API	Yes[15]	Plugin	Yes	Yes	Yes	Plugin[16]	3rd party plugins
Localhost	Yes	Yes	Yes	No	∞	Yes	Yes	Yes	Yes	?	Yes	Yes, required.	No	Java and SWT	Azureus	Web	No	Plugin	Yes	Yes	Yes	Plugin[16]	Yes
Shareaza	Yes	No	Yes	No	10	No	No	Yes	Yes	?	No	Yes [8]	No	C++	Shareaza core	GUI and web	No	No	Yes	No	No	Yes	Yes
XBT Client	Yes	No [12]	Yes	No	∞	Yes	Yes	Yes	Yes	?	?	No	No	C++	-	GUI and Web (alpha)	?	?	Yes	Yes	?	Yes	?

5 Revised Requirements for P2P tools in AXMEDIS

The AXEPTool is the application that allows business users to share a common environment in which AXMEDIS Objects can be published and loaded without the need of a centralized infrastructure which would be costly to maintain and a potential bottleneck in the whole production chain.

In the following, only the most relevant and mandatory requirements have been reported. Please note that some of them area satisfied by means of the AXMEDIS Content Processing tools that actively and periodically can send queries to the P2P network and tools. These issues will be clearer observing the architecture of the AXEPTool into the AXMEDIS framework.

The AXEPTool for P2P on B2B must meet requirements divided into the following categories: system and IPR, general requirements, discovery and connection to the AXMEDIS community, query to search AXMEDIS objects, loading remote AXMEDIS objects, publishing local AXMEDIS objects.

Requirements that are not mandatory are commented in a specific manner.

5.1 System and IPR requirements for P2P tools of AXMEDIS

AXMEDIS consortium major system and IPR requirements about the P2P support and tools in AXMEDIS:

5.1.1) to have of the P2P tools source code accessible and stored into the AXMEDIS Framework, located into the CVS of AXMEDIS;

5.1.2) to have the possibility of manipulating the source code for the future without any restriction, and without the need to have any specific authorization;

5.1.3) to have the possibility of improving the source code for integrating DRM aspects;

5.1.4) to have the source code in open source license, so as to allow the usage of tools without extra costs (NOT MANDATORY).

The Consortium Agreement of AXMEDIS may clarify other aspects of IPR.

5.2 General P2P Requirements valid for both AXMEDIA and AXEPTool P2Ps

The P2P infrastructure of AXMEDIS:

5.2.1) Must accept at least AXMEDIS objects in protected and non protected forms.

5.2.2) May accept in addition to AXMEDIS objects any files (NOT MANDATORY): not only AXMEDIS objects can be ”shared” and queried but files of any type or content.

5.2.3) Must allow performing some verification of consistency of the AXMEDIS object certification, to guarantee consistency of the objects with its metadata and its integrity, signature. Specific technology is present in the AXMEDIS that can be enforced into the P2P tools to cover this requirement.

5.2.4) Must allow performing the estimation of some fingerprint on the AXMEDIS objects and/or on eventual single audio and video tracks. Specific technology is present in the AXMEDIS that can be enforced into the P2P tools to cover this requirement (NOT MANDATORY if the solution already have some hash estimation to solve this)..

5.2.5) Must provide an efficient monitoring tool to present in the user interface the status of the download. The P2P tools must provide a real-time monitor for downloads of files in order to give to the consumer the status of all operations under run. The related GUI should present the monitored traffic as more as possible in an intuitively view.

5.2.6) Must provide support for queries in according to the AXMEDIS Query interface

5.2.7) Must support multiple sources download in an efficient manner such as .bitTorrent or other solutions

5.2.8) Must support the creation of multiple sources downloads where the peers are both those signed to B2B and C2C P2P networks, dependently on the content they have only.

5.2.9) The IDs used into the systems should be the AXOID defined in the AXMEDIS Framework with UUID model, if not possible an additional identification/descriptor has to be added into the AXMEDIS model.

5.3 Specific AXEPTool for P2P on B2B

The AXEPTool and solution has the above mentioned system and general requirements plus the following.

5.3.1) Must provide user registration and certification of clients

5.3.2) Must allow to establish some SSL connection with the tracker

5.3.3) Must provide integration for automatic loading and publishing of AXMEDIS content and content in general. These features have to be exported as WEB Services

5.3.4) Must provide support for sophisticated queries in according to the AXMEDIS Query interface *already developed and that can be reused into it.*

5.3.5) Music provide fully complete results exposing AXMEDIS AXOID and metadata, PAR, etc. as in the spirit of AXMEDIS.

5.4 Specific AXMEDIA Tool for P2P on B2C

The AXMEDIA Tool is the P2P application used in B2C via P2P. More requirements about Client/Server Distribution via PC are available in the specification of WP4 related to "**AXMEDIS for Distribution via Internet**". The AXMEDIA tool and solution has the above mentioned system and general requirements plus the following.

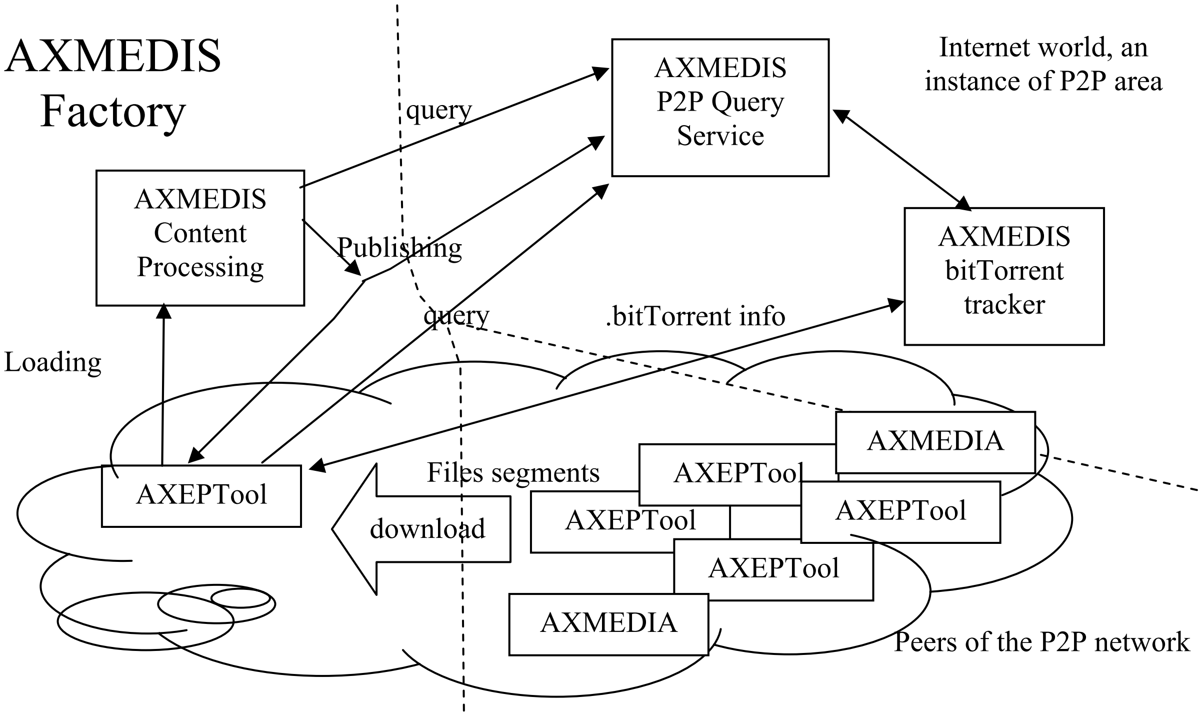
5.4.1) Must provide a simples query support that allows simple search queries composition through a simplified GUI, providing results in a simplified format, easy to understand for the final users.

5.4.2) Music provide complete results exposing AXMEDIS AXOID and metadata.

5.4.3) Must be easily installable on a wide range of computers and possibly on different platforms such as Windows and MAC.

6 Revised General B2B scenarios

The general solution and architecture of AXMEDIS framework tools integrated with P2P tools is reported in the previous section. Technically it can be depicted as follows.



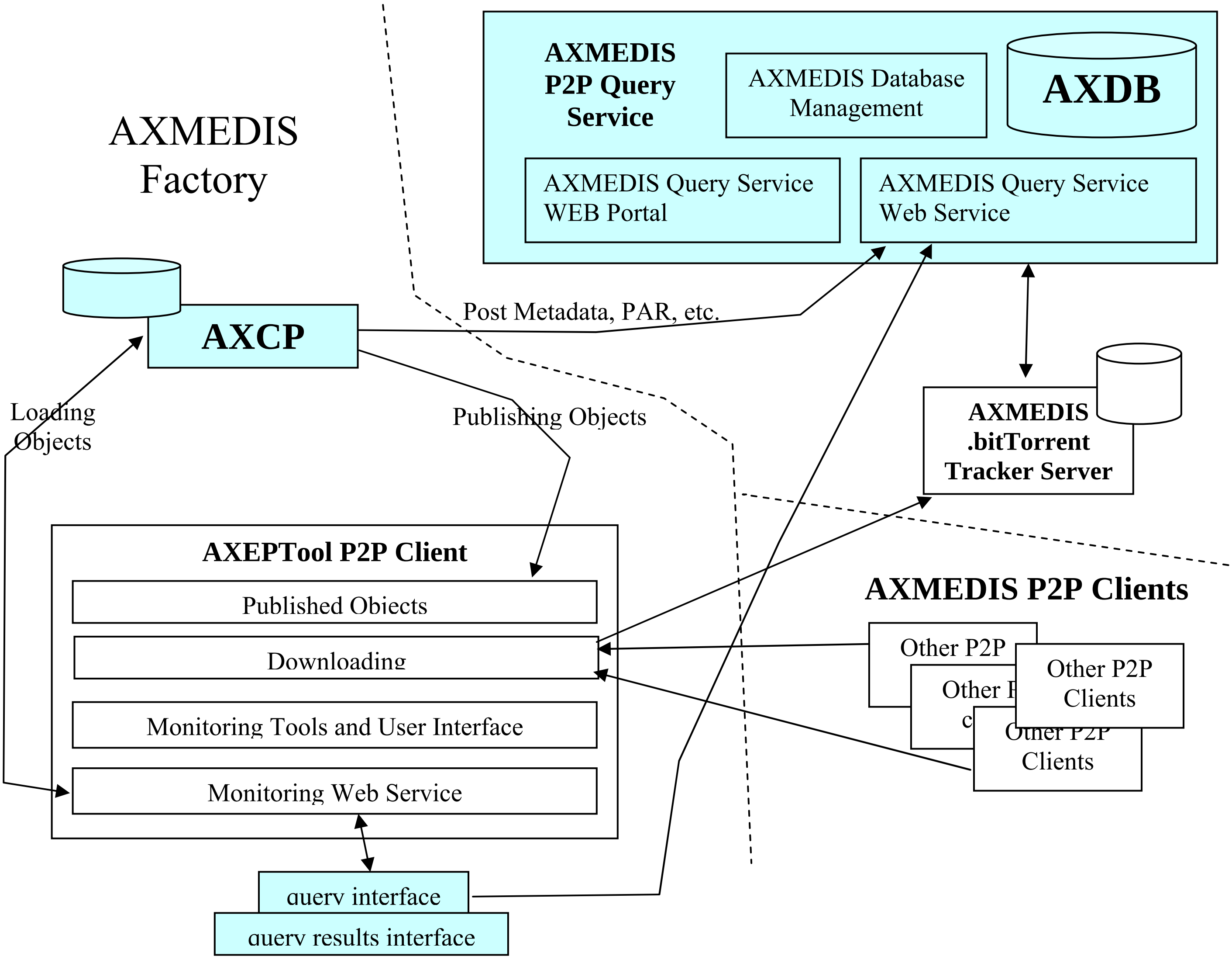
The P2P Clients can be both: AXEPTool and/or AXMEDIA P2P tools.

In this architecture for the B2B level we have:

- **AXMEDIS factory** (under responsibility of DSI), a place in which the AXMEDIS objects are created and/or are used to create other more complex objects, or are distributed towards other distribution channels, for example, broadcast,
- **AXMEDIS P2P Query Service** (under responsibility of EXITECH): a server located to cover a P2P community in which the database of metadata are located and that collect also the .bitTorrent information for the related AXMEDIS objects. AXMEDIS P2P Query Service may be connected with other AXMEDIS P2P Query Services to create a higher level of P2P sharing of content.
- **AXMEDIS BitTorrent Tracker**: a server derived from BitTorrent Tracker technology and solution to host BitTorrent information, update them, and provide them to AXMEDIS P2P clients according to the BitTorrent protocol. In a global geographic system, many AXMEDIS BitTorrent Trackers may survive to provide services.
- **.bitTorrent info:** BitTorrent information file containing the information created by a bitTorrent Maker processing the file to be shared into the P2P area.
- **AXEPTool**: a P2P client tool derived from a BitTorrent Client for B2B P2P sharing of AXMEDIS objects.
- **AXMEDIA**: a P2P client tool derived from a BitTorrent Client for C2C AXMEDIS object sharing.

In sharing files, the AXEPTool may use the files contained into the local repositories of the AXMEDIA tools and vice versa. So that the P2P network is practically defined by the “AXMEDIS P2P Query Service” and by clients that get connection to the AXMEDIS BitTorrent Tracker to get the specific BitTorrent information associated to a download.

7 Integration of P2P in AXMEDIS for B2B



In the above figure the parts in Cyan are already present and available in the AXMEDIS Framework even if not fully used for this kind of context.

The P2P Clients can be both: AXEPTool and/or AXMEDIA P2P tools.

7.1 AXCP (AXMEDIS content processing) and relationships with the other tools

AXCP (AXMEDIS Content Processing) it is a set of AXMEDIS tools described in the introduction of this document. The AXCP Editor and Executor allow to write scripts and execute them. Thus, the AXCP tools allow scripting a large set of content processing activities including accessing at web services, producing objects (MPEG-21 and AXMEDIS and OMA formats), manipulating XML, and thus in connection of the P2P tools the AXCP has to permit the control of upload and download of files from the P2P area.
See the following document for details about the AXCP tools:
http://www.axmedis.org/documenti/view_documenti.php?doc_id=1728

In the AXMEDIS architecture for the content sharing at B2B level with the usage of the P2P AXEPTools the AXCP (AXMEDIS content processing) tool (engine processing scripts) has to be capable of:

- publishing the AXMEDIS objects having AXOID=ABCDX by:
    i. Processing the AXMEDIS object with the “AXMEDIS .bitTorrent Maker” to produce the “AXMEDIS .bitTorrent Information” for file AXMEDIS object ABCDX
    ii. Sending metadata (an simple AXMEDIS object without digital resources) to the AXMEDIS P2P Query service, including the “AXMEDIS .bitTorrent Information”
    iii. Posting the AXMEDIS object processed by the “AXMEDIS .bitTorrent Maker” into the AXEPTool database for published objects.
- performing queries on the AXMEDIS P2P Query Service to have as a results a list of AXMEDIS objects with their metadata and their “AXMEDIS .bitTorrent Information”.
- interacting with the AXEPTool via a WEB Service called “Monitoring Web Service” to:
    - ask for the downloads of some AXMEDIS objects passing to the AXEPTool the “AXMEDIS .bitTorrent Information” and AXOIDs
    - ask for the status of the objects under downloads/publishing, receiving back:
        - listing of objects published
        - list of objects under download including those already at the 100%
        - for each object the percentage of download performed with respect to their completion, and an estimated time to complete the download, for example: in 34 Minutes.
    - stop the download for some AXMEDIS object identified by its AXOID

7.2 AXMEDIS .bitTorrent Maker tool

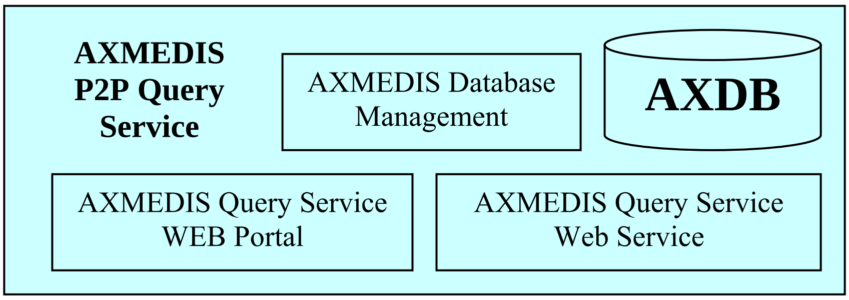
BitTorrent Maker is a tool that allows packaging objects and producing AXMEDIS .bitTorrent information files.
The **AXMEDIS .bitTorrent Maker** has to be usable as a simple executable file tool with its parameters from:

- the AXCP
- the AXEPTools
- the AXMEDIS tools and
- any other tools.

The .torrent file is produced as a result of a process that is typically performed by P2P BitTorrent Clients and can be passed to other tools as mentioned above to reach at the end the AXMEDIS BitTorrent Tracker Server.
The AXMEDIS .bitTorrent Maker is produced by customizing (if needed) a classical .bitTorrent maker, in many cases that functionality is directly present into the BitTorrent Client. .

7.3 AXMEDIS P2P Query Service Server

AXMEDIS P2P Query Service is a server that provides information about the objects indexed into the AXDB and at disposal of a community. Several Communities can be created, for example one for each distributor, by using and installing a different AXMEDIS P2P Query Service SERVER.
A query service that could be a simple redirection of the AXMEDIS database query on other AXMEDIS factories (see http://www.axmedis.org/documenti/view_documenti.php?doc_id=1728)
For more details on the structure of the AXMEDIS query support see:
http://www.axmedis.org/documenti/view_documenti.php?doc_id=1932


AXMEDIS P2P Query Service
AXMEDIS Database Management
AXDB
AXMEDIS Query Service WEB Portal
AXMEDIS Query Service Web Service

AXMEDIS P2P Query Service Server includes

1. **AXDB: AXMEDIS Database** which is practically an AXMEDIS database without the objects but only with the metadata. Plus a table that associates the AXOID with .bitTorrent initial files mentioned before as: "AXMEDIS .bitTorrent Information". This information is passed in a field of the AXMEDIS metadata.
2. **AXMEDIS Query Service Web Service** allows to establish a protected channel for:
    a. publishing/loading a new AXMEDIS object metadata and additional information (.torrent)
    b. receiving a query and providing a result in terms of AXOID, metadata and .torrent file
    c. deleting/un-publishing of an AXMEDIS object.
3. **AXMEDIS Query Service WEB Portal** allows to common users to make queries on the portal directly, the results are those described above. This is a simple and direct interface to make queries, probably used only by final users.

Please note that the **AXMEDIS Query Service Web Service** can be used by:

- the AXCP for automating the Publishing of objects, SSL connection or not: this can be implemented by posting the AXOID of objects to be indexed. In other cases, the posting could be performed only by sending the AXOID while the metadata can be recovered form the AXCS.
- the AXCP for automating the search for new objects in the P2P network, SSL connection or not
- the AXMEDIS query support for making a query on the Virtual Network of P2P tools of the several factories.

7.4 AXMEDIS BitTorrent Tracker Server

The AXMEDIS BitTorrent Tracker Server contains a table that contains for each AXOID all the AXMEDIS BitTorrent information about who has the file segments, etc. This server works with the P2P client according to the standard BitTorrent protocol with trackers. The communication from the P2PClient and the Tracker can be protected with SSL.

When and user on a P2P Client has received the query result has in the hands the references to the BitTorrent Tracker Server and from it can get the information about the peers that have the file segments and thus can start the download according to the BitTorrent policies.
Please note that the communication from the AXMEDIS Query Service Server and the BitTorrent Tracker could be used to communicate to the Tracker the list of objects (list of AXOID) permitted for that Tracker. Another way to enforce this can be to authenticate the P2P Clients, thus nobody that is authenticated would have interest in putting a wrong .torrent file in the Tracker.

The AXMEDIS BitTorrent Tracker Server:

- Can be realized from one of the server accessible, see the list in the previous section
- Some P2P BitTorrent Clients, such as Azureus, include their own integrated Tracker Server. This means that potentially each peer may have its tracker server. A more specific tracker server capable to support higher performances is needed. So that it is preferred to install one of the solutions listed in the previous sections.
- Probably starting from a standard Tracker Server none or marginal changes are needed
- Eventual change and/or modification could be needed in the protocol in order to host only AXMEDIS BitTorrent information files. This has to be decided yet, and probably can be performed in other manners.
- It could be useful to have statistical information about (some of the tracker server are capable to provide statistical information):
    - How many downloads of the BitTorrent File
    - How many updates of the BitTorrent File
    - Any other information that could be collected by the Tracker Server
    - Etc.

It should be avoided to provide many trackers as the number of Peers. It could be better to have few centralized trackers for all. May be one for each AXMEDIS P2P Query Service Server.

The tracker can be easily integrated into the AXMEDIS P2P Query Service Server or in alternative it is possible to use any active BitTorrent tracker if the protocol for AXMEDIS is not going to be changed.
If we maintain the same protocol we lose the feature of controlling the objects that are using the tracker and thus we can take the risk that some piracy objects can be indexed by our tracker server and that biTorrent files of our AXMEDIS objects are hosted in many and unknown Tracker Servers (this may always happen). The best solution is to control the tracker to allow the insertion and update of BitTorrent files and information only from AXMEDIS P2P tools.

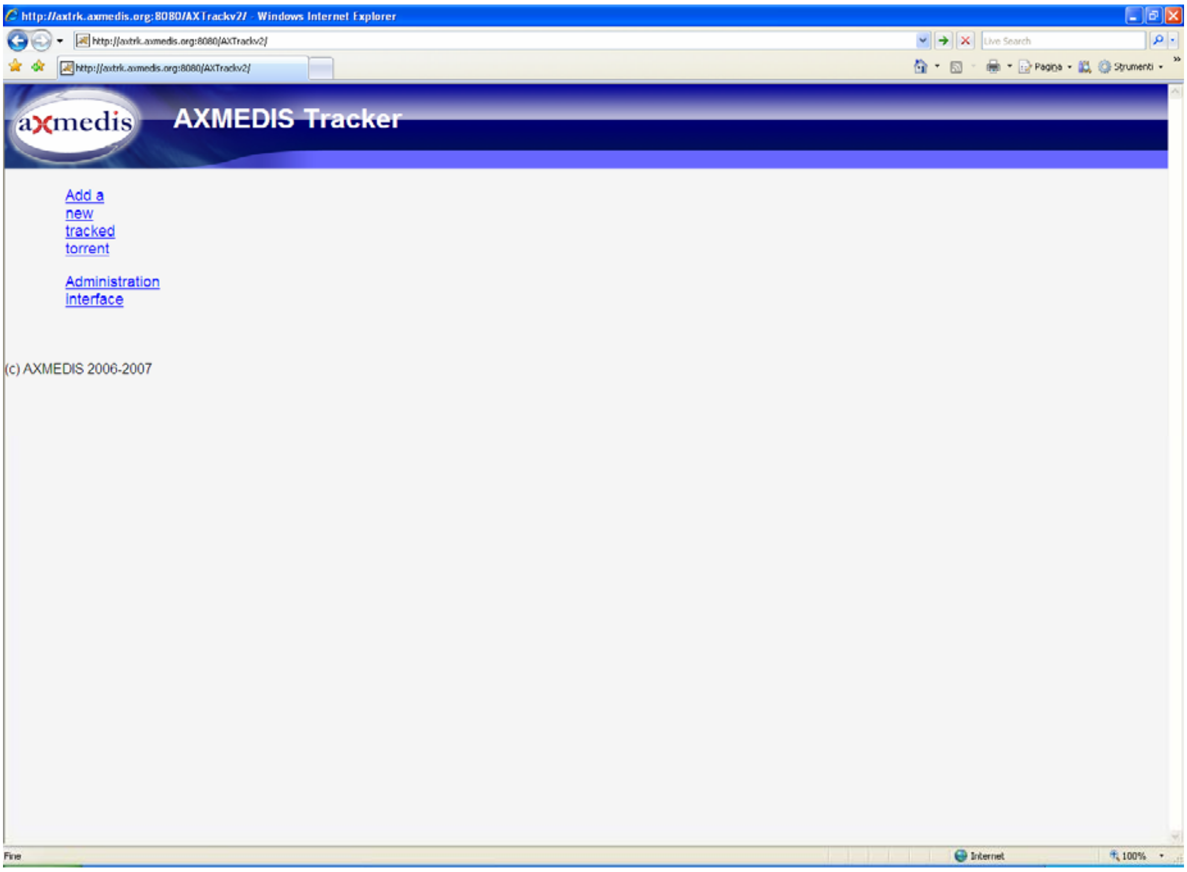
One possibility to create a restricted community is to:

- encrypt BitTorrent Files with some specific keys and protocols depending on the AXMEDIS certification model.
- Use the same BitTorrent protocol and Tracker Servers capable to open those files.
- Use specific P2P Client Tools.

The AXTrack has a password protected administration panel that lets to monitor its activity and performs some operations. It is possible to:

- **delete peers informations,**

- **erase all tracked documents (or some of them) and peer data,**
- **upload a tracked torrent directly,**
- **toggle on/off the status of the flag "accept non axmedis objects" (so to allow or deny the uploading of non AXMEDIS content on the tracker's catalogue),**
- **view the peers and torrents statistics with all the download events and**
- **check the whole catalogue list.**

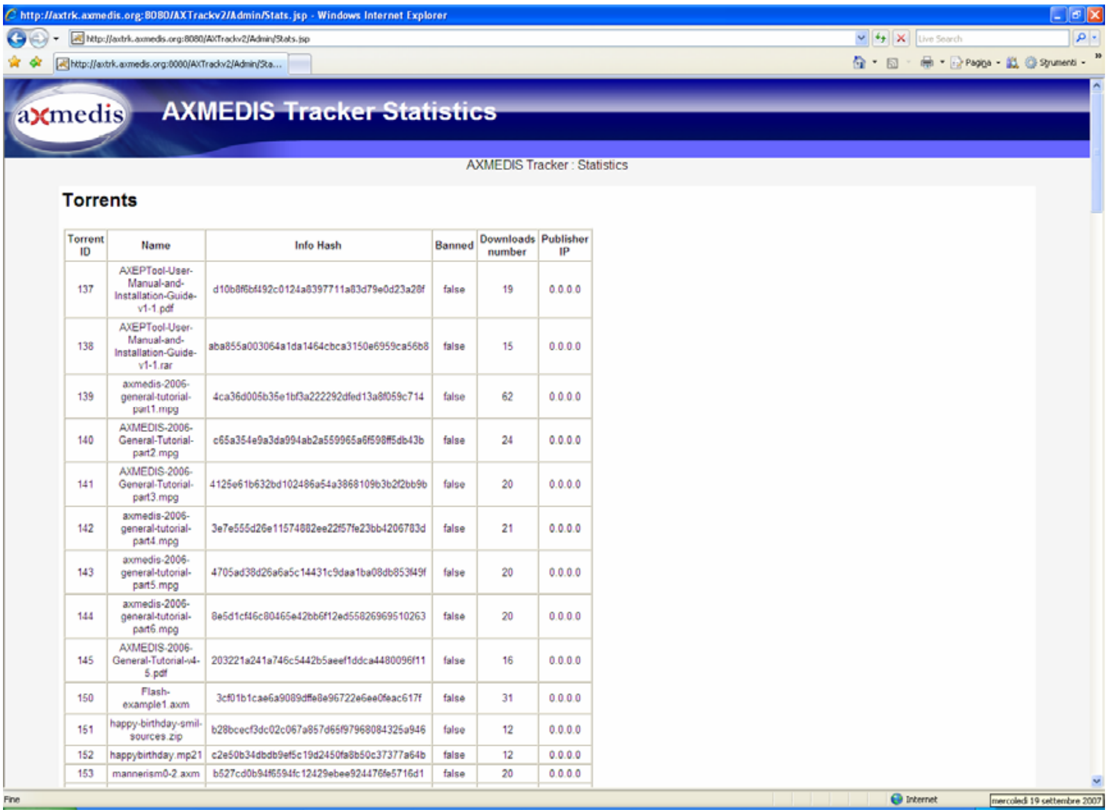

http://axtrk.axmedis.org:8080/AXTrackv2/ - Windows Internet Explorer
http://axtrk.axmedis.org:8080/AXTrackv2/
Live Search
Pagina
Strumenti
axmedis
Fine
Internet
100%

AXMEDIS Tracker

Add a new tracked torrent

Administration interface

(c) AXMEDIS 2006-2007

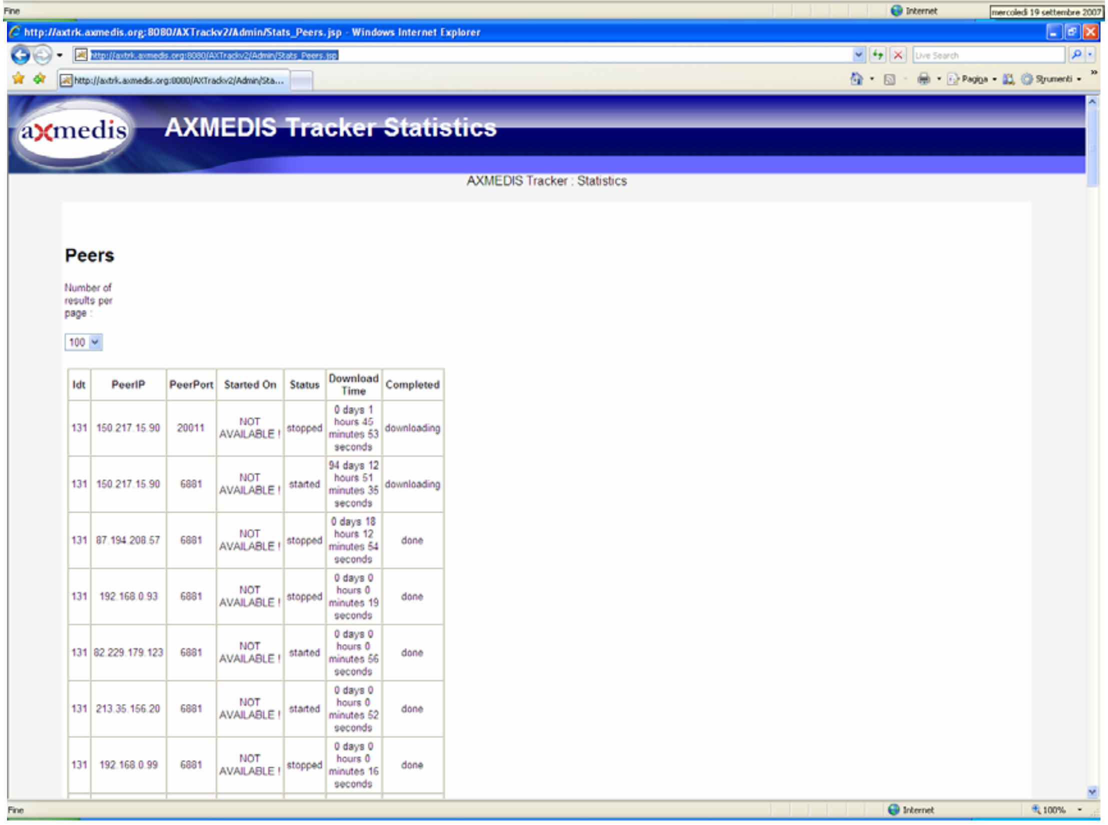

http://axtrk.axmedis.org:8080/AXTrackv2/Admin/Stats.jsp - Windows Internet Explorer
http://axtrk.axmedis.org:8080/AXTrackv2/Admin/Stats.jsp
Live Search
http://axtrk.axmedis.org:8080/AXTrackv2/Admin/Sta...
Pagina
Strumenti
axmedis
Fine
Internet
mercoledì 19 settembre 2007

AXMEDIS Tracker Statistics

AXMEDIS Tracker : Statistics

Torrents

Torrent ID	Name	Info Hash	Banned	Downloads number	Publisher IP
137	AXEPTool-User-Manual-and-Installation-Guide-v1-1.pdf	d10b8f6bf492c0124a8397711a83d79e0d23a28f	false	19	0.0.0.0
138	AXEPTool-User-Manual-and-Installation-Guide-v1-1.rar	aba855a003064a1da1464cbca3150e6959ca56b8	false	15	0.0.0.0
139	axmedis-2006-general-tutorial-part1.mpg	4ca36d005b35e1bf3a222292dfed13a8f059c714	false	62	0.0.0.0
140	AXMEDIS-2006-General-Tutorial-part2.mpg	c65a354e9a3da994ab2a559965a6f598ff5db43b	false	24	0.0.0.0
141	AXMEDIS-2006-General-Tutorial-part3.mpg	4125e61b632bd102486a54a3868109b3b2f2bb9b	false	20	0.0.0.0
142	axmedis-2006-general-tutorial-part4.mpg	3e7e555d26e11574882ee22f57fe23bb4206783d	false	21	0.0.0.0
143	axmedis-2006-general-tutorial-part5.mpg	4705ad38d26a6a5c14431c9daa1ba08db853f49f	false	20	0.0.0.0
144	axmedis-2006-general-tutorial-part6.mpg	8e5d1cf46c80465e42bb6f12ed55826969510263	false	20	0.0.0.0
145	AXMEDIS-2006-General-Tutorial-v4-5.pdf	203221a241a746c5442b5aeef1ddca4480096f11	false	16	0.0.0.0
150	Flash-example1.axm	3cf01b1cae6a9089dffe8e96722e6ee0feac617f	false	31	0.0.0.0
151	happy-birthday-smil-sources.zip	b28bcecf3dc02c067a857d65f97968084325a946	false	12	0.0.0.0
152	happybirthday.mp21	c2e50b34dbdb9ef5c19d2450fa8b50c37377a64b	false	12	0.0.0.0
153	mannerism0-2.axm	b527cd0b94f6594fc12429ebee924476fe5716d1	false	20	0.0.0.0


http://axtrk.axmedis.org:8080/AXTrackv2/Admin/Stats_Peers.jsp - Windows Internet Explorer
http://axtrk.axmedis.org:8080/AXTrackv2/Admin/Stats_Peers.jsp
Live Search
http://axtrk.axmedis.org:8080/AXTrackv2/Admin/Sta...
Pagina
Strumenti
axmedis
Fine
Internet
100%

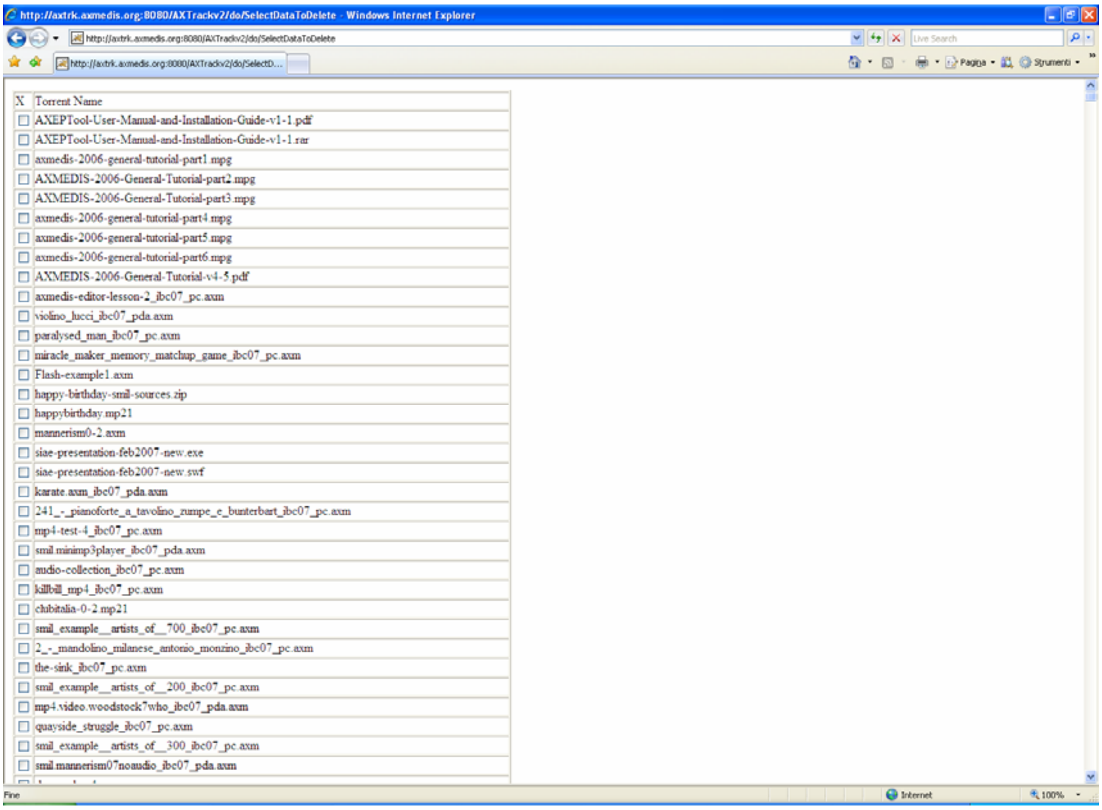
AXMEDIS Tracker Statistics

AXMEDIS Tracker : Statistics

Peers

Number of results per page : 100

Idt	PeerIP	PeerPort	Started On	Status	Download Time	Completed
131	150.217.15.90	20011	NOT AVAILABLE !	stopped	0 days 1 hours 45 minutes 53 seconds	downloading
131	150.217.15.90	6881	NOT AVAILABLE !	started	94 days 12 hours 51 minutes 35 seconds	downloading
131	87.194.208.57	6881	NOT AVAILABLE !	stopped	0 days 18 hours 12 minutes 54 seconds	done
131	192.168.0.93	6881	NOT AVAILABLE !	stopped	0 days 0 hours 0 minutes 19 seconds	done
131	82.229.179.123	6881	NOT AVAILABLE !	started	0 days 0 hours 0 minutes 56 seconds	done
131	213.35.156.20	6881	NOT AVAILABLE !	started	0 days 0 hours 0 minutes 52 seconds	done
131	192.168.0.99	6881	NOT AVAILABLE !	stopped	0 days 0 hours 0 minutes 16 seconds	done



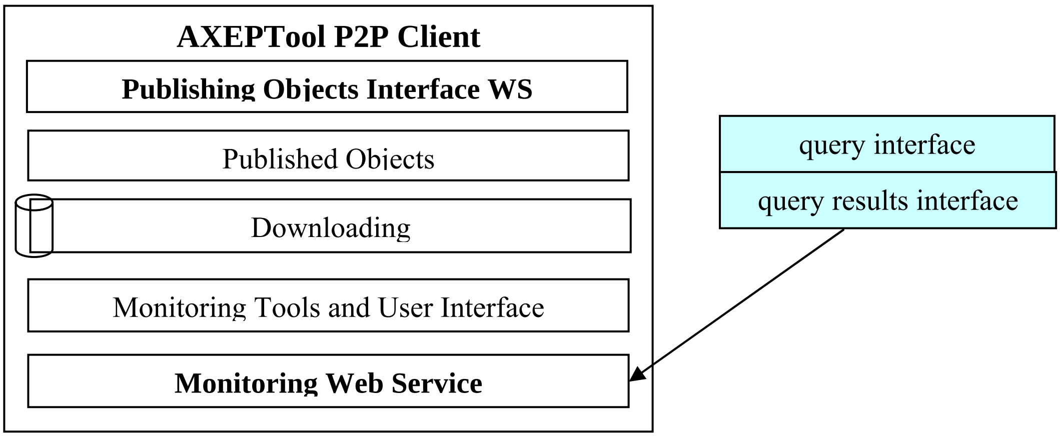
http://axtrk.axmedis.org:8080/AXTrackv2/do/SelectDataToDelete - Windows Internet Explorer

http://axtrk.axmedis.org:8080/AXTrackv2/do/SelectDataToDelete

X	Torrent Name
□	AXEPTool-User-Manual-and-Installation-Guide-v1-1.pdf
□	AXEPTool-User-Manual-and-Installation-Guide-v1-1.rar
□	axmedis-2006-general-tutorial-part1.mpg
□	AXMEDIS-2006-General-Tutorial-part2.mpg
□	AXMEDIS-2006-General-Tutorial-part3.mpg
□	axmedis-2006-general-tutorial-part4.mpg
□	axmedis-2006-general-tutorial-part5.mpg
□	axmedis-2006-general-tutorial-part6.mpg
□	AXMEDIS-2006-General-Tutorial-v4-5.pdf
□	axmedis-editor-lesson-2_ibc07_pc.axm
□	violino_lucci_ibc07_pda.axm
□	paralysed_man_ibc07_pc.axm
□	miracle_maker_memory_matchup_game_ibc07_pc.axm
□	Flash-example1.axm
□	happy-birthday-smil-sources.zip
□	happybirthday.mp21
□	mannerism0-2.axm
□	siae-presentation-feb2007-new.exe
□	siae-presentation-feb2007-new.swf
□	karate.axm_ibc07_pda.axm
□	241_-_pianoforte_a_tavolino_zumpe_e_bunterbart_ibc07_pc.axm
□	mp4-test-4_ibc07_pc.axm
□	smil.minimp3player_ibc07_pda.axm
□	audio-collection_ibc07_pc.axm
□	killbill_mp4_ibc07_pc.axm
□	clubitalia-0-2.mp21
□	smil_example__artists_of__700_ibc07_pc.axm
□	2_-_mandolino_milanese_antonio_monzino_ibc07_pc.axm
□	the-sink_ibc07_pc.axm
□	smil_example__artists_of__200_ibc07_pc.axm
□	mp4.video.woodstock7who_ibc07_pda.axm
□	quayside_struggle_ibc07_pc.axm
□	smil_example__artists_of__300_ibc07_pc.axm
□	smil.mannerism07noaudio_ibc07_pda.axm

7.5 AXMEDIS AXEPTool P2P Client

AXEPTool is a P2P client tool derived from bitTorrent Client for B2B P2P sharing and thus also for interacting with the AXMEDIS Factory and specifically with the AXCP tool.

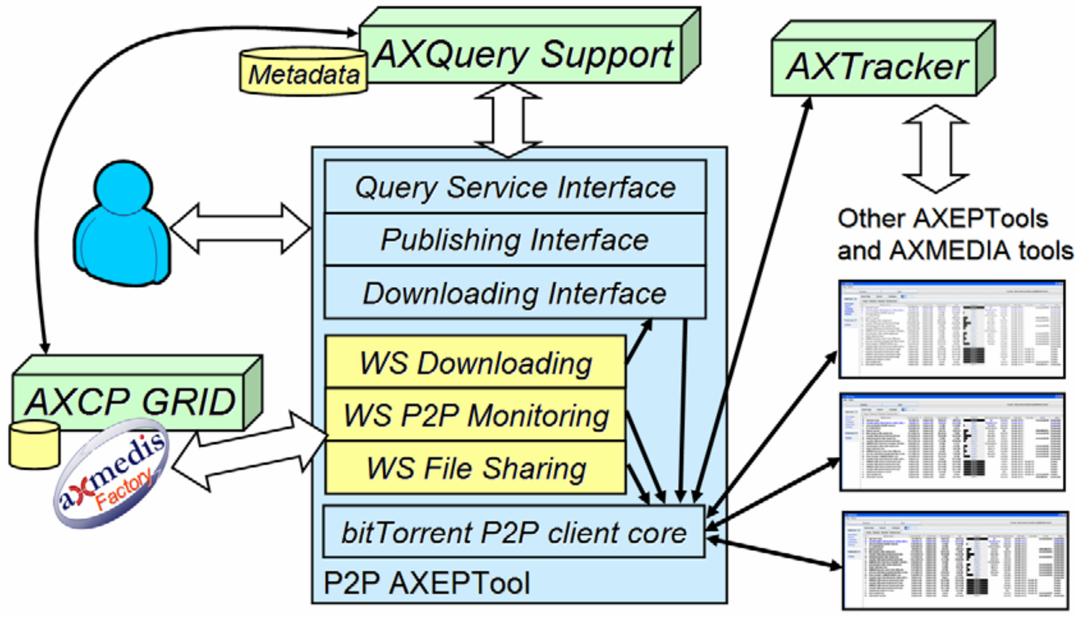


The AXMEDIS AXEPTool P2P Client:

- Include an Integrated Web Browser and an Internal Web Server in order to easily communicate with the Web Portal realized by P2P Query Service.
- Provide a **P2PPublisher WS** for receiving objects from another tool (for example from the AXMEDIS AXCP Engine) to be published on the P2P network and thus for posting the BitTorrent information file and/or making accessible the full version of the file for other peers. It provides a functionality via WEB service to **bitTorrentURL publishObject(localFileName or filepath, AXOID, trackerURL)** publication of an object, its metadata have been already posted on the AXMEDIS P2P Query Service Server; the contract consist in posting/saving the AXMEDIS object passed into the **Published Object repository** of the P2P client tool and create a bittorrent file, thus posting bittorrent file on the selected tracker URL. If you have already published an object, and you call this functionality with a local file name it tries to post the correspondent .torrent file on the Tracker. Using filepath instead provides object publishing too on the Tracker's catalogue.
- The objects (AXMEDIS or not) can also be published manually from the user interface of the AXEPTool.
- Include two different repository one for Downloaded (or downloading) and one for Published objects. The bittorrent will automatically make available a downloaded media (or media) part, the distinction is only for clearly distinguish which content has been introduced in the P2P network from the client itself.
- Provide a **Monitoring Tools User Interface** for:
  - Observing the status of the plan to Downloaded files, % of completion, who is involved, download rate, etc.
  - Estimation of the time to completion
  - All the examples related to the Azureus Client have been depicted in the previous sections.
- Provides a **P2PMonitoring WS** with the following functionalities:
  - **P2PObjectList getPublishedObjectsList()**, returns a list of objects which are present in the directory of Published objects. The list depicts all media (both AXMEDIS or general), including filename, AXOID (if any) and infoHash.
  - **PublishStatus getPublicationStatus(AXOIDFileName selectedMedia)**, provides the current status of a given object which is in the **Published Object area of the P2P** environment and selected by the proper identification (AXOID or localFileName).

It returns the amount of bytes, seeds and peers for each segment, and the list of contacting peer for a given media.

- **result removePublishedObject(AXOIDFileName selectedMedia)**, deletes the object specified by the ID from the database/directory of Published objects. This is something that should not to be done in P2P systems, but when they are used for B2B the directory could reach the limit of their size and housekeeping is needed.
- **result downloadObject(AXOID)**, starts the download of the object especified by the AXOID in the P2P environment. The AXOID field has to be added into the Tracker database.
- **result downloadObjectURL(bittorrentURI)**, starts the download of the object specified by the torrent file URL in the P2P environment.
- **DownloadStatus monitorDownloadingStatus(AXOIDFileName selectedMedia)**, provides the current status a downloading objects in the P2P environment. It returns the current percentage of download, estimated time to completion (if possible), peers and seeds.
- **P2PObjectList getDownloadingObjectsList(),** returns the list of objects in the directory of Downloaded objects.
- **result controlDownloadingObject(AXOIDFileName selectedMedia, boolean start/stop)**, start/stops the download of the object specified by identification (AXOID or localFileName).
- **result removeDownloadedObject(AXOIDFileName selectedMedia),** deletes the object specified by the ID from the directory of Downloaded objects. This is something that should not to be done in P2P systems, but in some cases the directory could reach the limit of their size and housekeeping is needed.
- **mediaURL getDownloadedObjectURI (AXOIDFileName selectedMedia),** retrieves the object URI from which the downloaded at 100% object can be copied into the AXMEDIS database by the AXCP. This action will be performed by the AXCP directly accessing to that URL.
- **ObjectStatus getStatus (AXOIDFileName selectedMedia),** retrieves the status of an object in the AXEPTool's transfer list. This can be Downloading, Paused, Checking, Seeding, Error, Queued.
- **Sysinfo getSystemInformations(),** retrieves the main system informations and profile of the AXEPTool: CPU type, Operating System, RAM, AXEPTool's configuration file in XML format, User Name, AXEPTool's version, free HD space where AXEPTool is running.



In the following sections the WSDL formalization of the above two web services are reported.

Integration into the AXMEDIS AXEPTool of:

- certification and authentication mechanisms of AXMEDIS to be sure that the AXEPTool is a certified and authorized tool and person.
- A module for estimating fingerprints of resources, connection to the AXCS to verify if the estimated FP corresponds to the resource effectively contained in the digital object.

In order to operate, AXEPTool requires the use of the AXTrack. The AXTrack is a tracker server located on the network, usually in a well connected server room. It keeps tracks of files currently distributed on the peer to peer network. As a result, it also serves as an authoritative source for listing the available files on the network and tracking statistics.

As a result, part of the interface was implemented inside the AXTrack as a web interface, but it is not designed to be used directly. This is why the AXEPTool integrates a native web browser (either managed by the internet explorer engine or by mozilla's gecko engine) which displays the web pages served by the AXTrack.

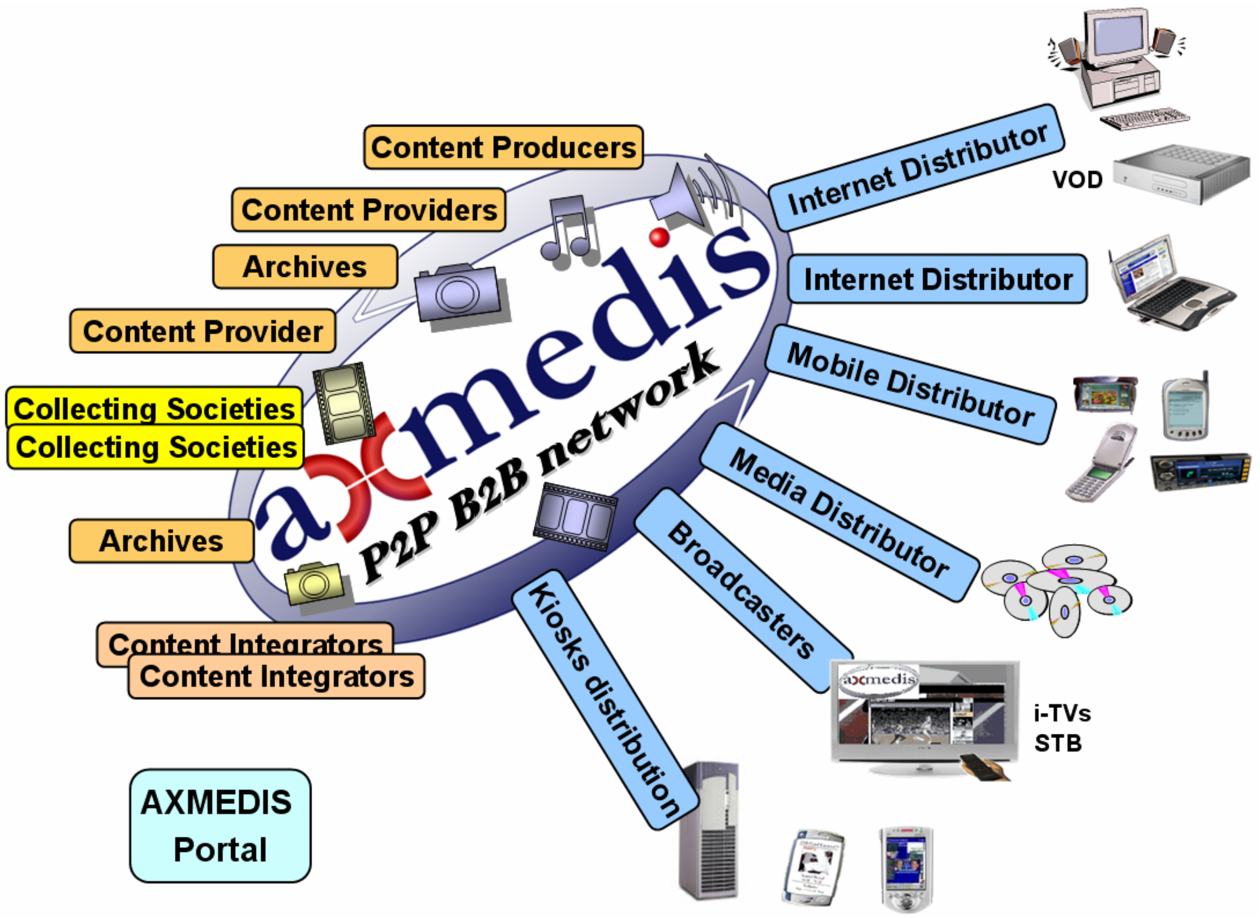
The AXEPTool also integrates an internal web-server. It is used as a replacement of the LiveConnect technology to notify the AXEPTool of events inside the browser window. The result is fully cross-platform which is not the case with LiveConnect. Connection to this internal web-server will be restricted to connections from localhost. For the moment, it is mainly used to automatically launch downloads directly when browsing the catalog from the browser window connected to the AXTrack.

The application uses a customized version of the Azureus engine. We have a way to use the Azureus jar file (which contains a full executable Azureus instance) as a library. As a result, the AXEPTool contains a standard compliant and full featured BitTorrent stack.

7.6 AXMEDIS P2P Network statistics and performances overview

The main aim of business actors such as content providers is to find a fast and safe solution for digital media publishing and distribution over the Internet. An important requirement is in fact to allow an efficient geographical distribution of digital contents, maintaining at the same time the highest level of security, controlled access, an appropriate network redundancy (fault tolerance) and prevention of node failures.

The advantages of P2P networks for B2B are related to the possibility of creating a network in which content producers, archives, providers, integrators, aggregators and distributors *can exchange resources at lower costs* for the content sharing, network set up and maintenance. On the other hand, in order to support Business to Business, B2B, activities for content distribution and sharing among business actors a set of specific requirements have to be taken into account.



The B2B P2P Network depicted in Figure 1 is used for sharing content among actors of the value chain. One of more P2P rings/areas of B2B can be created to share different kinds of objects or for geographical purposes.

The P2P B2B network can also share the content with peers of the final user network, increasing in this manner the number of peers and thus the advantages of the network.

Therefore, the P2P solutions for B2B have to satisfy a set of *functional requirements* and in particular they have to provide support for the:

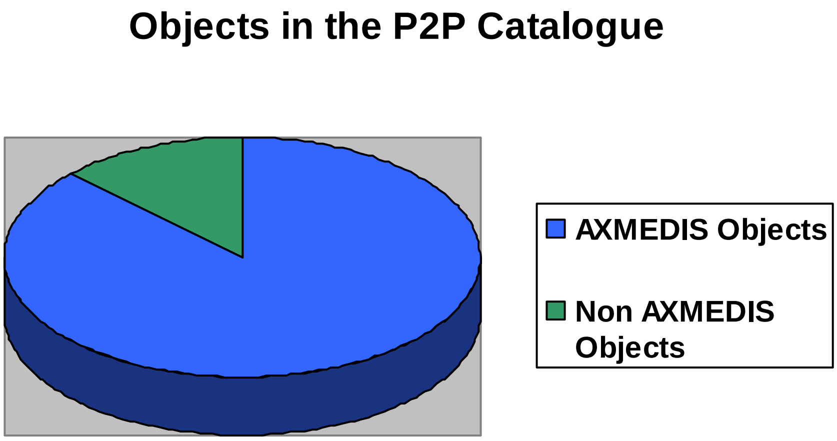
- Sharing/publishing and thus allowing the distribution of protected and non protected content according to one or more DRM technologies. In this context, the content has to be protected independently on the user identification and license, so that the license and the content adopt the so called Open Model and support super distribution. The file/content publication has to include the automated indexing of content on a query support server;
- control of the P2P network (i) avoiding the distribution of non certified/authorized content (e.g., content with metadata inconsistent with resources), (ii) avoiding the sharing of illegal files, those that are shared without the corresponding authorizations of the content owner), (iii) avoiding the access to P2P B2B facilities to non authorized (or malicious) actors/users;
- monitoring the activities of the P2P network (tools and users) in terms of (i) performances, (ii) queries performed and (iii) content shared, to extract statistical information that may be used for a better tuning of the service, and to understand the user's behavior;
- querying on content on the basis of a large set of metadata, including those to make search and queries on business/trading aspects: complex metadata, licensing rules and conditions, costs, etc.;

- set up of high quality services of content distribution/sharing. This implies to guarantee the content download according to predictable performance, QOS (Quality Of Service); even when (i) a new content object/file is shared, thus when the P2P network may not contain enough replicas; (ii) faults occur in the network and/or in the nodes;
- automating the above mentioned activities and harmonizing them with the factory workflow.

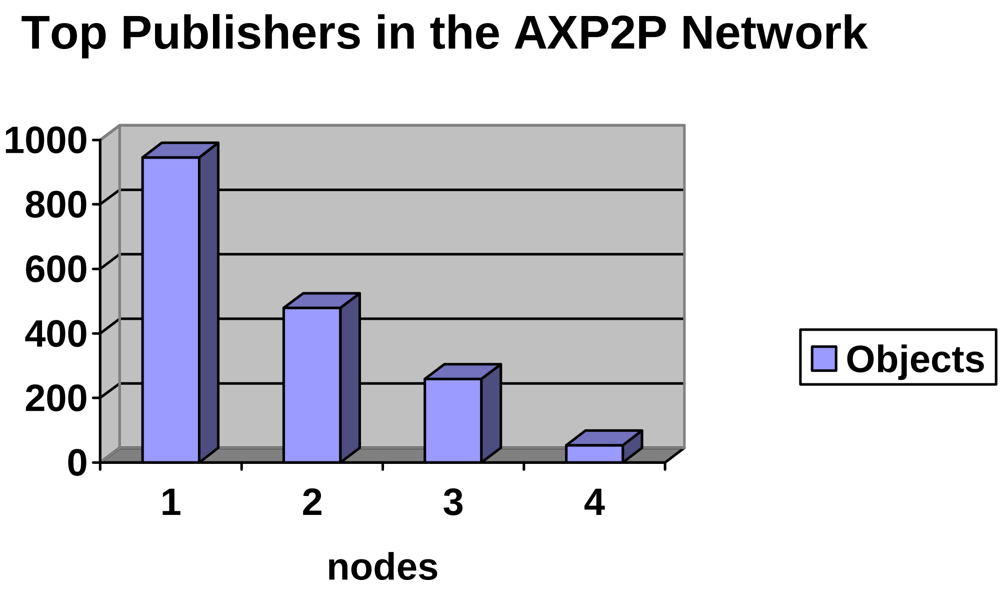
In P2P networks for B2B, the typical demand of user privacy of the P2P network and communities among final users are relaxed. In fact, in the B2B the user identification is preferred and thus it is a value to accelerate the trading and the conclusion of transactions.

Content producers and aggregators also need an effective integration of such P2P system with other business resources and processes, from content definition to digital media processing and production workflow. The AXMEDIS framework supports all these activities in an integrated manner. In the following section, the P2P architecture of AXMEDIS for digital content distribution at B2B level is presented. The AXMEDIS framework solution allows integrating GRID technology and P2P for content distribution. The AXMEDIS Content Processing (AXCP) GRID platform provides an infrastructure with a suitable set of tools to automate the above described adaptation process and harmonize it with the other back office activities such as access to servers with databases and Web Services in general.

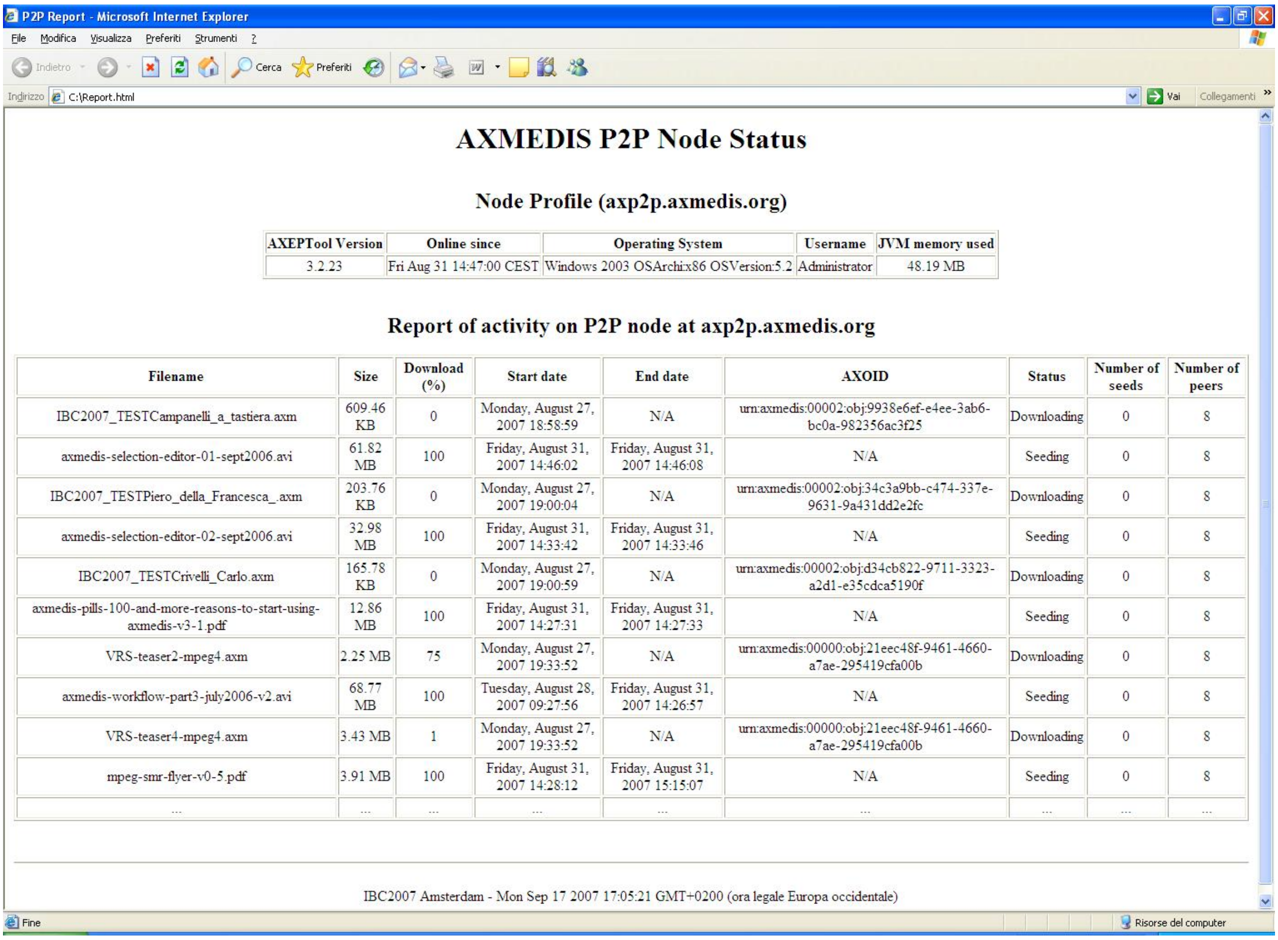
AXTrack's catalogue, at the time of this document (August 2007), consists of 1932 objects (16.27 GB of digital contents). The average P2P traffic, measured as number of peers involved in a transfer per hour is about 3.6. The number of unique peers appeared on the tracker for a download request since the start of the network is 280.
The average number of objects shared by a master node AXEPTool is 2046 while the average uptime of a master node AXEPTool is 9.2 days. Main operating system is Windows XP (~78%) followed by Windows 2003 Server (~22%). The average memory consumption by JVM process is an AXEPTool is 48.43 MB.



By the use of AXCP Rule editor AXMEDIS Tool it is possibile to monitor the network status, start objects download and log P2P traffic for statistical purposes. With javascript is possible to check a master node profile using the AXEPTool, and output a HTML results page. You can see a sample downloading object list with each download object's current status (seeding, downloading, paused etc…), start downloading time, number of seeds and peers involved in the transfer, filesize and AXOID and download percentage. Operating system, memory usage and current uptime for each node can be monitored too.



In the pictures below a node profile and the average Java Virtual Machine memory usage by a sample list of master nodes are showed.


P2P Report - Microsoft Internet Explorer
File Modifica Visualizza Preferiti Strumenti ?
Indietro
Cerca
Preferiti
Indirizzo C:\Report.html
Vai
Collegamenti
Fine
Risorse del computer

AXMEDIS P2P Node Status

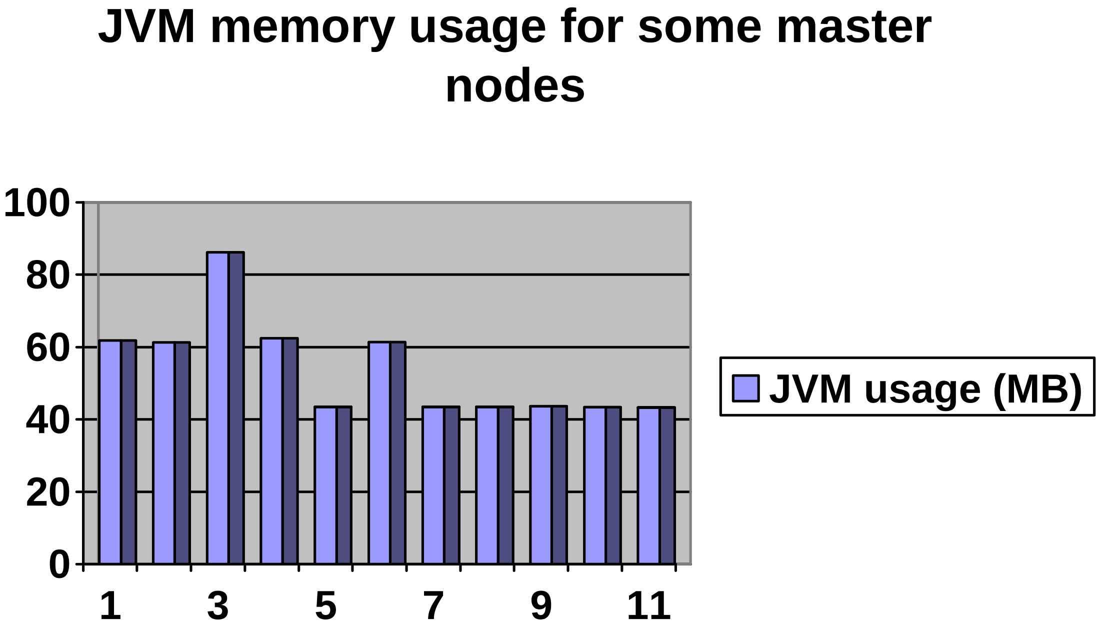
Node Profile (axp2p.axmedis.org)

AXEPTool Version	Online since	Operating System	Username	JVM memory used
3.2.23	Fri Aug 31 14:47:00 CEST	Windows 2003 OSArchi:x86 OSVersion:5.2	Administrator	48.19 MB

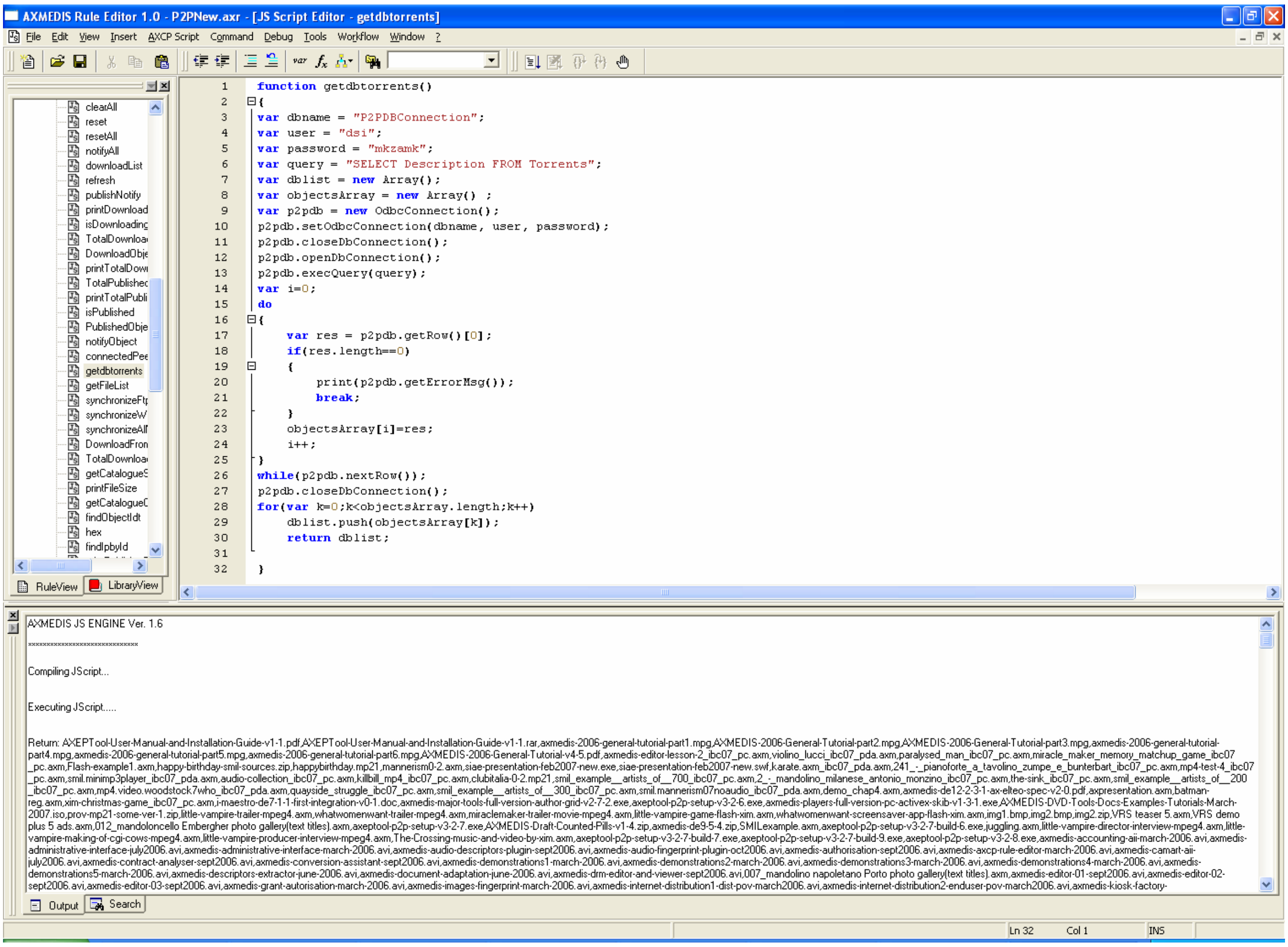
Report of activity on P2P node at axp2p.axmedis.org

Filename	Size	Download (%)	Start date	End date	AXOID	Status	Number of seeds	Number of peers
IBC2007_TESTCampanelli_a_tastiera.axm	609.46 KB	0	Monday, August 27, 2007 18:58:59	N/A	urn:axmedis:00002:obj:9938e6ef-e4ee-3ab6-bc0a-982356ac3f25	Downloading	0	8
axmedis-selection-editor-01-sept2006.avi	61.82 MB	100	Friday, August 31, 2007 14:46:02	Friday, August 31, 2007 14:46:08	N/A	Seeding	0	8
IBC2007_TESTPiero_della_Francesca_.axm	203.76 KB	0	Monday, August 27, 2007 19:00:04	N/A	urn:axmedis:00002:obj:34c3a9bb-c474-337e-9631-9a431dd2e2fc	Downloading	0	8
axmedis-selection-editor-02-sept2006.avi	32.98 MB	100	Friday, August 31, 2007 14:33:42	Friday, August 31, 2007 14:33:46	N/A	Seeding	0	8
IBC2007_TESTCrivelli_Carlo.axm	165.78 KB	0	Monday, August 27, 2007 19:00:59	N/A	urn:axmedis:00002:obj:d34cb822-9711-3323-a2d1-e35cdca5190f	Downloading	0	8
axmedis-pills-100-and-more-reasons-to-start-using-axmedis-v3-1.pdf	12.86 MB	100	Friday, August 31, 2007 14:27:31	Friday, August 31, 2007 14:27:33	N/A	Seeding	0	8
VRS-teaser2-mpeg4.axm	2.25 MB	75	Monday, August 27, 2007 19:33:52	N/A	urn:axmedis:00000:obj:21eec48f-9461-4660-a7ae-295419cfa00b	Downloading	0	8
axmedis-workflow-part3-july2006-v2.avi	68.77 MB	100	Tuesday, August 28, 2007 09:27:56	Friday, August 31, 2007 14:26:57	N/A	Seeding	0	8
VRS-teaser4-mpeg4.axm	3.43 MB	1	Monday, August 27, 2007 19:33:52	N/A	urn:axmedis:00000:obj:21eec48f-9461-4660-a7ae-295419cfa00b	Downloading	0	8
mpeg-smr-flyer-v0-5.pdf	3.91 MB	100	Friday, August 31, 2007 14:28:12	Friday, August 31, 2007 15:15:07	N/A	Seeding	0	8
...	...	...	...	...	...	...	...	...

IBC2007 Amsterdam - Mon Sep 17 2007 17:05:21 GMT+0200 (ora legale Europa occidentale)



AXMEDIS Ruleeditor is it possible to query the AXTrack database and the AXEPTool integrated WebService to get many informations, for example number of objects in the catalogue and their filename, (see picture below), objects filesize, number of peers involved in a transfer, peer activity on the tracker in the last n minutes, download object list for an AXEPTool, download and upload speed etc…



```
function getdbtorrents()
{
var dbname = "P2PDBConnection";
var user = "dsi";
var password = "mkzamk";
var query = "SELECT Description FROM Torrents";
var dblist = new Array();
var objectsArray = new Array() ;
var p2pdb = new OdbcConnection();
p2pdb.setOdbcConnection(dbname, user, password);
p2pdb.closeDbConnection();
p2pdb.openDbConnection();
p2pdb.execQuery(query);
var i=0;
do
{
    var res = p2pdb.getRow()[0];
    if(res.length==0)
    {
        print(p2pdb.getErrorMsg());
        break;
    }
    objectsArray[i]=res;
    i++;
}
while(p2pdb.nextRow());
p2pdb.closeDbConnection();
for(var k=0;k<objectsArray.length;k++)
    dblist.push(objectsArray[k]);
    return dblist;

}
```

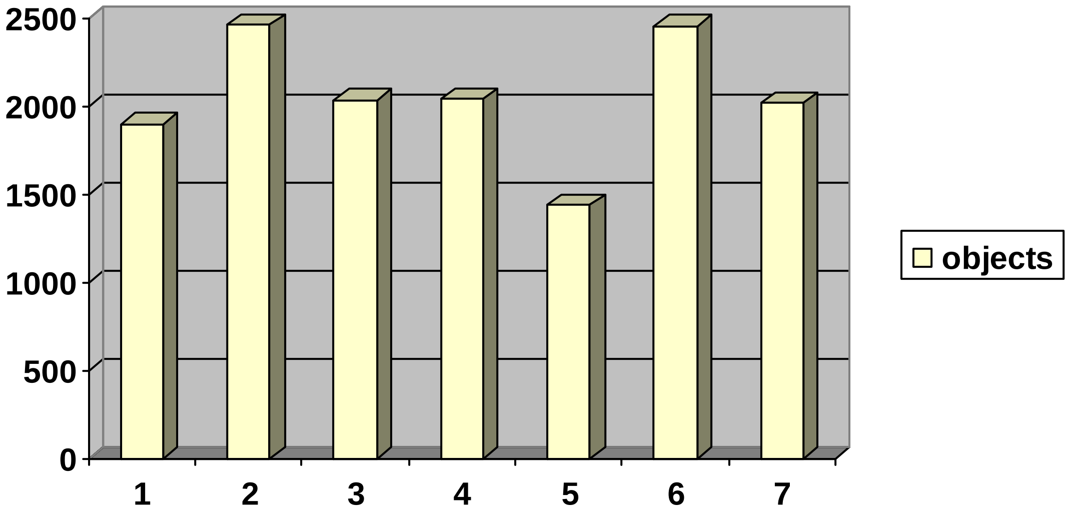
```
AXMEDIS JS ENGINE Ver. 1.6

********************************

Compiling JScript...

Executing JScript.....

Return: AXEPTool-User-Manual-and-Installation-Guide-v1-1.pdf,AXEPTool-User-Manual-and-Installation-Guide-v1-1.rar,axmedis-2006-general-tutorial-part1.mpg,AXMEDIS-2006-General-Tutorial-part2.mpg,AXMEDIS-2006-General-Tutorial-part3.mpg,axmedis-2006-general-tutorial-
part4.mpg,axmedis-2006-general-tutorial-part5.mpg,axmedis-2006-general-tutorial-part6.mpg,AXMEDIS-2006-General-Tutorial-v4-5.pdf,axmedis-editor-lesson-2_ibc07_pc.axm,violino_lucci_ibc07_pda.axm,paralysed_man_ibc07_pc.axm,miracle_maker_memory_matchup_game_ibc07
_pc.axm,Flash-example1.axm,happy-birthday-smil-sources.zip,happybirthday.mp21,mannerism0-2.axm,siae-presentation-feb2007-new.exe,siae-presentation-feb2007-new.swf,karate.axm_ibc07_pda.axm,241_-_pianoforte_a_tavolino_zumpe_e_bunterbart_ibc07_pc.axm,mp4-test-4_ibc07
_pc.axm,smil.minimp3player_ibc07_pda.axm,audio-collection_ibc07_pc.axm,killbill_mp4_ibc07_pc.axm,clubitalia-0-2.mp21,smil_example__artists_of__700_ibc07_pc.axm,2_-_mandolino_milanese_antonio_monzino_ibc07_pc.axm,the-sink_ibc07_pc.axm,smil_example__artists_of__200
_ibc07_pc.axm,mp4.video.woodstock7who_ibc07_pda.axm,quayside_struggle_ibc07_pc.axm,smil_example__artists_of__300_ibc07_pc.axm,smil.mannerism07noaudio_ibc07_pda.axm,demo_chap4.axm,axmedis-de12-2-3-1-ax-elteo-spec-v2-0.pdf,axpresentation.axm,batman-
reg.axm,xim-christmas-game_ibc07_pc.axm,i-maestro-de7-1-1-first-integration-v0-1.doc,axmedis-major-tools-full-version-author-grid-v2-7-2.exe,axeptool-p2p-setup-v3-2-6.exe,axmedis-players-full-version-pc-activex-skib-v1-3-1.exe,AXMEDIS-DVD-Tools-Docs-Examples-Tutorials-March-
2007.iso,prov-mp21-some-ver-1.zip,little-vampire-trailer-mpeg4.axm,whatwomenwant-trailer-mpeg4.axm,miraclemaker-trailer-movie-mpeg4.axm,little-vampire-game-flash-xim.axm,whatwomenwant-screensaver-app-flash-xim.axm,img1.bmp,img2.bmp,img2.zip,VRS teaser 5.axm,VRS demo
plus 5 ads.axm,012_mandoloncello Embergher photo gallery(text titles).axm,axeptool-p2p-setup-v3-2-7.exe,AXMEDIS-Draft-Counted-Pills-v1-4.zip,axmedis-de9-5-4.zip,SMILexample.axm,axeptool-p2p-setup-v3-2-7-build-6.exe,juggling.axm,little-vampire-director-interview-mpeg4.axm,little-
vampire-making-of-cgi-cows-mpeg4.axm,little-vampire-producer-interview-mpeg4.axm,The-Crossing-music-and-video-by-xim.axm,axeptool-p2p-setup-v3-2-7-build-7.exe,axeptool-p2p-setup-v3-2-7-build-9.exe,axeptool-p2p-setup-v3-2-8.exe,axmedis-accounting-aii-march-2006.avi,axmedis-
administrative-interface-july2006.avi,axmedis-administrative-interface-march-2006.avi,axmedis-audio-descriptors-plugin-sept2006.avi,axmedis-audio-fingerprint-plugin-oct2006.avi,axmedis-authorisation-sept2006.avi,axmedis-axcp-rule-editor-march-2006.avi,axmedis-camart-aii-
july2006.avi,axmedis-contract-analyser-sept2006.avi,axmedis-conversion-assistant-sept2006.avi,axmedis-demonstrations1-march-2006.avi,axmedis-demonstrations2-march-2006.avi,axmedis-demonstrations3-march-2006.avi,axmedis-demonstrations4-march-2006.avi,axmedis-
demonstrations5-march-2006.avi,axmedis-descriptors-extractor-june-2006.avi,axmedis-document-adaptation-june-2006.avi,axmedis-drm-editor-and-viewer-sept2006.avi,007_mandolino napoletano Porto photo gallery(text titles).axm,axmedis-editor-01-sept2006.avi,axmedis-editor-02-
sept2006.avi,axmedis-editor-03-sept2006.avi,axmedis-grant-autorisation-march-2006.avi,axmedis-images-fingerprint-march-2006.avi,axmedis-internet-distribution1-dist-pov-march2006.avi,axmedis-internet-distribution2-enduser-pov-march2006.avi,axmedis-kiosk-factory-
```



7.7 AXMEDIS Content Processing GRID

The AXMEDIS Content Processing (AXCP) architecture consists of a GRID Engine where the processes are executed on GRID nodes and specified in terms of the AXCP Script Language, which is an extension of ECMA Script language [3]. AXCP provides tools for editing, debugging and processing of scripts and a scheduler for the allocation and control of processes executed on the multiprocessor GRID architecture.

The tools and scripts allow to automate all the phases of content production, protection, distribution and coping with:

- processing of Metadata, mapping/transcoding and adaptation with XSLT profiles associated devices and/or user preferences;
- reasoning about user, device and network profiles to identify parameters for the adaptation according to the standard MPEG-21 DIA;
- processing of digital resources for the extraction of fingerprint, synchronization; adaptation (change in resolution and format, transcoding, estimation of descriptors etc.;
- processing of cross media content structure for packaging (MPEG-21, OMA), production of SMIL, HTML, layouting with SMIL template, style sheets;
- processing and production of DRM licenses according to MPEG-21 REL and OMA DRM;
- content protection according to MPEG-21 IPMP, AXMEDIS and OMA, allowing the dynamic selection of protection tools and algorithms.
- content and information gathering/crawling and access via AXCP tools and language; from/to content management systems such as ODBC, ORACLE, MSSQL, MySQL, XML databases; from/to operating system files; via communication protocols such as Web Services, WEBDAV, HTTP, FTP;
- multichannel distribution of content with license posting and protection information according to different business models: internet distribution; satellite data broadcast distribution (towards kiosk, PCs and STBs); mobile distribution (to smart phones and PDAs); P2P networks (automating both publication and download of digital content from and to a P2P network for B2B distribution as discussed in this paper).

All mentioned activities and flows can be managed by external workflows management systems such as Open Flow, towards single tools and the AXMEDIS content processing platform.

The AXMEDIS P2P architecture is based on BitTorrent protocol and can be used for setting up P2P networks among Business and Consumers actors or for a mix of them.

7.8 BitTorrent Solution pros and cons

BitTorrent is a peer to peer (P2P) file sharing protocol proposed by Bram Cohen to allow publishing and sharing of digital contents over the Internet. Traditional BitTorrent solutions are based on a few main components: the BitTorrent Tracker, the BitTorrent P2P Client and on related protocol for exchanging data and information among them.

Mainly, the Tracker identifies the community which the BitTorrent P2P Clients, called Nodes, can refer to for the publication and download. When a new file is uploaded/published on the P2P network the Node creates a so called BitTorrent file with all the information related to the file publication and posts this data into the Tracker. When a file is published over a BitTorrent network, it is split in segments of fixed size, each of them with a fingerprint calculated using SHA-1 hash.

The Tracker assigns an ID to each Node and an identification to each file uploaded (e.g., a hash code). A Node can start the download having a BitTorrent file (that can be obtained in several manners, e.g. from the tracker, via email, HTML pages, ftp, etc.). In the BitTorrent file the Tracker is referred so that the Node can contact the Tracker itself (via a HTTP protocol) to have updated information about the list of Nodes that

have segments of the file to be downloaded. In some BitTorrent implementations every peer can behave as a Tracker too, through the use of distributed hash tables (DHT). This information is continuously updated on the Tracker by the Nodes receiving/exchanging segments.
The policy of the BitTorrent protocol among Nodes for the download is based on few rules that facilitate the diffusion of the file segments assigning a higher priority for download to segments that have a lower number of replicas in the network, and contributes to fight the so called leech resistance by forcing each user to share a part of his bandwidth capacity during downloads.
In BitTorrent terminology there are two different types of Nodes: Peers and Seeds that constitute the so called swarm. A Peer is a Node which is not yet sharing all the segments of a file, since it's still downloading the file itself; a Seed is a Node which has completed the download of the file and it's sharing all its segments over the network.

The classical BitTorrent solution in its traditional implementation does not address the above mentioned B2B functional requirements and in particular those regarding the:

- integration of protection support and DRM, so that to control the distribution and sharing of non certified content;
- querying/indexing of content on the basis of the metadata and related object/file cataloguing and querying for B2B and/or C2C (Consumer to Consumer). In classical BitTorrent solutions, the queering/indexing is delegated to external services. Typically, the Tracker has only capabilities of presenting the list of objects, the called catalogue and the metadata are limited to the file name;
- control of the network allowing:
  - removal of content/files from the network, that would mean removing them at least from the Tracker;
  - fast notification of new files and thus of seeding of files among the network;
  - automating the control of P2P network for publishing and downloading files in an automatic manner, via the integration of the P2P network facilities with the content production facilities;
- monitoring activities and user behavior on Nodes. In addition to the classical P2P network monitoring that can be performed on Tracker in a limited manner.

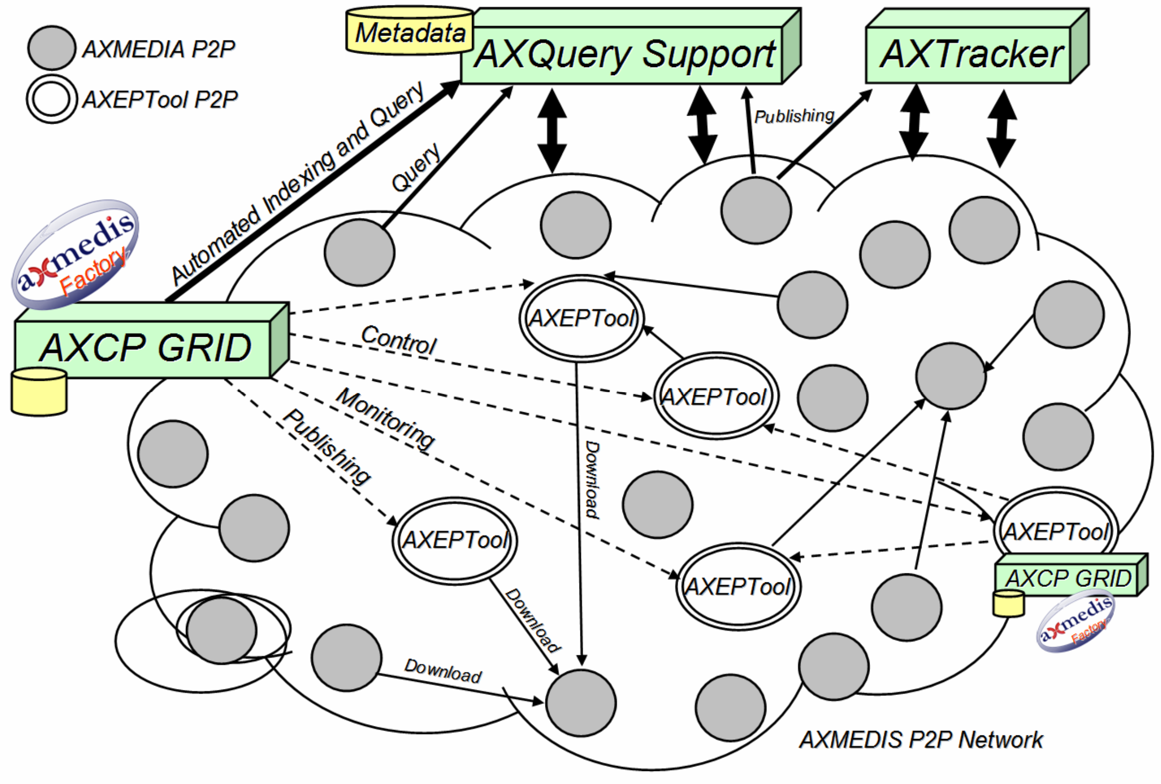
The above limitations have been overcome by the AXMEDIS P2P solution proposed. In the following section, the main aspects of the AXMEDIS P2P Architecture are described and better motivated.

7.9 AXMEDIS P2P solution

The AXMEDIS P2P solution is based on the following active elements.

AXTracker is a modified BitTorrent Tracker that manages the AXMEDIS P2P network and community. The tracker is used to publish new content by posting on it BitTorrent information and obtaining IDs for nodes and files. Main AXMEDIS Tracker's features are: tracking of BitTorrent files; catalog of tracked files; tracker's download statistics. AXTracker serves as an authoritative source for listing the available files on the network and for tracking statistics. In the global geographic system, many AXMEDIS BitTorrent Trackers may survive to provide services and share each other information regarding nodes and content files, so that to distribute the workload and making it fault tolerant. Thanks to the AXTracker, the AXEPTool/AXMEDIA P2P client tools are able to know where they can retrieve the segments constituting a file.

AXEPTool is a special P2P BitTorrent Client Node, suitable to play the role of a P2P Node for B2B activities such as producers, distributors, integrators, etc., for B2B content distribution. It is capable to accept automated requests of publishing, download, monitoring, etc., from other tools; for example, from an AXCP GRID node in the AXMEDIS Content Production/Distribution Factory. These facilities have been included to allow the P2P network controllability. In addition, the AXEPTool accepts also manual commands of publishing, downloading and monitoring via a graphical user interface.



AXMEDIA is a specific P2P BitTorrent Client Node for final users content sharing and B2C (Business to Consumer) content distribution. Internet distributors may use the P2P Network to reduce costs for scaling the infrastructure for content distribution. AXMEDIA does not include some of the AXEPTool's features regarding publication, downloading and monitoring of the activities. Thus, it is easy to use, lighter to be executed and more easily acceptable by final users.

AXCP GRID is an instance of the AXCP GRID tools. By the AXCP node it is possible to automate the activities of publishing, downloading, monitoring and control of the P2P Network calling the AXEPTool Web Services for that purpose. An AXCP node may control one or more AXEPTools as depicted in Figure 2.

AXQuery Support is a server on which the user and the AXCP may perform queries. Queries can be in a complex format and include classification, identification and legal aspects such as details regarding the potential licenses that can be associated with content, the so called Potentially Available Rights, that in AXMEDIS are formalized in MPEG-21 REL. This solution has been defined to solve the problem of content indexing. Once a new content is published on the network by a P2P Network node or by an AXCP GRID, the metadata are passed to the AXQuery Support. In the global geographic system, many AXQuery Support servers may exist and exchange the indexing information each other, so that to facilitate the query, distribute the workload, scaling the architecture and making it fault tolerant.

Therefore, according to the above presented architecture, the AXMEDIS P2P Network presents different kinds of P2P Nodes:

- content production, publications and sharing nodes in which a controlling tool (e.g., AXCP GRID) and at least one AXEPTool are joined;
- content sharing and distribution nodes which are constituted by an AXEPTool only (controlled and supervised by other AXCP GRID nodes);
- AXMEDIA P2P nodes for content sharing.

The AXTracker keeps track of each file segment shared on the network and can be turned ON/OFF to be exclusively used for AXMEDIS/MPEG-21 content files. When the limitation is not activated the AXTracker may interact with any legacy BitTorrent clients. Once the AXTracker is constrained to work only on AXMEDIS certified objects, the information regarding other objects types is not provided to clients and the AXTracker does not accept to publish non AXMEDIS objects. Thus, the download may start from a Node only by using an AXOID (AXMEDIS object ID) which is a unique ID for the AXMEDIS objects. This has been realized defining a modified BitTorrent protocol which is based on a specific BitTorrent File that includes an AXOID. This modified BitTorrent file is fully compatible with standard BitTorrent Files used by any BitTorrent application so that it can be uploaded and managed by any BitTorrent Tracker and not only by AXTrackers.

Thus, when the user publishes a new AXMEDIS object in a protected and/or non protected manner:

- metadata are automatically extracted (from protected and/or non protected objects) and used to feed the AXQuery Support via Web Service. This solution avoids the passing of large files to the query support server for their indexing;
- the BitTorrent information is generated by the Node, saved locally and passed to the AXTracker.

The same activities can be performed in a scheduled manner via script on the AXCP node.

When a user (or a Factory node with its AXCP GRID) makes a query the result is provided as a list of AXOID. They can be used to get the BitTorrent file from the AXTracker by means of the above mentioned modified BitTorrent protocol.

7.10 Services of the P2P Clients toward the Factory

The AXEPTool is a special P2P BitTorrent Client Node, suitable to cover the above mentioned requirements and roles in the P2P Network for B2B. It is used by content producers, distributors, integrators, aggregators, and expert users, etc., for automating activities of publishing, downloading and monitoring of the P2P network. Therefore, it accepts formal requests of publishing (file sharing), downloading, and monitoring via Web Services as depicted in Figure 3. In the AXMEDIS solution, the AXCP GRID Node may use those Web Services from the Java Script code which can be executed by any GRID node of the AXCP GRID. In addition, specific tools may be created for interacting with the AXEPTool via those Web Services.

In more details, the Web Services exposed by the AXEPTool P2P Clients, the AXQuery Support and the information provided by the AXTracker allow the AXCP GRID to automate procedures for:

publishing(AXEPToolURL, FileURI, AXTrackerURL) to publish objects/files on a reference AXEPTool, automating the activities of indexing and posting of metadata on the Query Support and uploading the BitTorrent information file on the AXTracker, making also accessible a full file on the AXEPTool itself;

- *download(AXEPToolURL, FileName or AXOID or BitTorrentURL)* requesting to a controlled AXEPTool to perform the download of a file (via file name) or of an AXMEDIS object (via AXOID) or passing a BitTorrent file. Each Node has a reference tracker, that may be changed on the basis of the BitTorrent file or by changing the configuration;

- *listContent listPublished(AXEPToolURL)* to get the list of files/objects that have been published on a specific AXEPTool. The list depicts all media (either AXMEDIS object or file), including filename, AXOID (if any) and info Hash. They are kept in a different area with respect to the downloaded objects/files;
- *listContent listDownloaded(AXEPToolURL)* to get the list of files/objects that have been downloaded and are currently downloading on a specific AXEPTool. All of them are kept in a different area with respect to the published objects/files;
- *infoStatus status(AXEPToolURL, FileName or AXOID)* to monitor the status of an indicated object/file on a specific AXEPTool in terms of its percentage of completed download, start time and date, peers involved in the transfer, download rate, estimated time to completion, status (seeding, download, waiting, paused, etc.);
- *controlDownload(AXEPToolURL, FileName or AXOID)* to start/stop the download of the specified object (AXOID or file name) on a specific AXEPTool;
- *listAXOID query(AXQuerySupportURL, Query)* to get the list of AXMEDIS objects that satisfy a given query on an AXQuery Support;
- *listContent catalogue(AXTrackerURL)* to get the list of content from a Tracker;
- *delete(AXEPToolURL, FileName or AXOID)* deletes the specified object (AXOID or file name) from the database/directory of published objects on a specific AXEPTool. This is something that should not be done in P2P systems, while when they are used for B2B the objects/files may be deleted if deprecated, or for optimizing the object/file distribution in the network;
- *FileURI get(AXOID or FileName)* to get objects/files URI of downloaded objects/files and to move them into some database accessible by the AXCP GRID, or to process them with the AXMEDIS content processing capabilities, e.g. adaptation, transcription, filtering, formatting, composition, integration, aggregation, etc., [3];
- *listAXEPTool listAXEPTool(AXTrackerURL)* to get the list of active and non active AXEPTools with their status;
- *infoNode statusNode(AXEPToolURL)* to obtain the capabilities of the node in terms of software version, HD space, CPU, networking capabilities, etc., and in second instance the user behavior too;
- *infoTracker statusTracker(AXTrackerURL, period)* to get the status log regarding the activities of peers in a given period of time (from to).

7.11 P2P Network Control Experiments

On the basis of the above reported basic functionalities of the AXMEDIS P2P network it is possible to create procedures/scripts for:

- Publishing an object on all the AXEPTools of the P2P Network for example to accelerate the seeding of a given object on the network:

```
Axoid=getAXOID(MyFile);
publishing(MyAXEPTool, MyFile, TheAXTracker);
LA = listAXEPTool(TheAXTracker);
For each la of LA:  download(la, Axoid);
```

- Notifying the publication of new objects/files in the network among the different AXCP GRIDs/AXEPTools controlled;
- Removing/deleting of objects/files from the network, at least from the AXEPTool nodes, AXTrackers and AXQuery Support:

```
    Axoid=getAXOID(MyFile);
```

```
axeptoolList=listAXEPTool(AXTrackerURL);
For each axeptool of axeptoolList:
    delete(axeptool, AXOID);
```

- Monitoring the status of the P2P Network, discovering which are the most active/virtuous AXEPTools, their capabilities, how many downloads have been performed, how many segments have been provided and for whose objects/files, if they are active, etc. This allows to perform specific analysis to assess the reputation of Business actors on the basis of their behavior on the corresponding AXEPTool;
- Controlling the content seeded by the AXEPTools, for example constrained them to become an exact replica of each other (for an uniform seeding distribution), or imposing some distribution for the content in the network of the AXEPTools on the basis of the content distribution and statistical analysis:

```
LA=listAXEPTool(AXTrackerURL);
rootNode=LA[0];
List= listDownloaded(rootNode);
For each la of LA:
    For each list in List:
        download(la, list);
```

- Activating automated queries for obtaining, downloading and posting these objects into the database of specific content collection on the basis of complex queries. So that, these active queries can be periodically activated to verify if some new content satisfies the criteria and, in the positive case, the automated download can be activated as well;
- Activating automated publishing on the P2P Network of accessible collections from the AXCP GRID and crawling facilities etc.

In this case study a simple AXMEDIS P2P Network constituted by eight AXEPTool nodes distributed across Europe was analyzed to monitor content dissemination speed, download bandwidth and some other useful network parameters. Measures and data were obtained by querying each node in the network, and collecting correspondent results through the use of the AXCP GRID, hosting a set of JavaScript rules implemented as a combination of the available above mentioned functionalities.

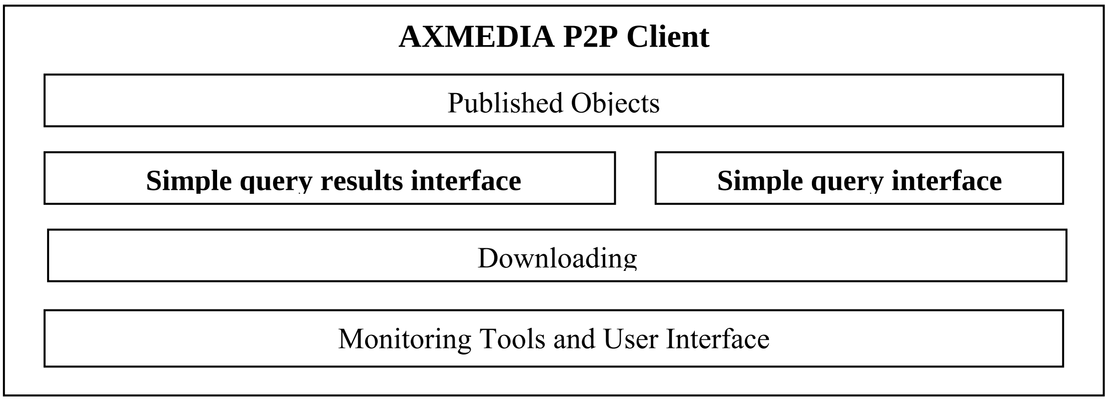
The performed tests are based on the functionalities described in the previous list. One of them assessed the impact of the first algorithm implementation (*Publishing an object on N AXEPTools)*, as reported in the previous section. It is based on the measure of the average time for content seeding as a function of the number of AXEPTools. It has been performed by forcing selected AXEPTool nodes to download a published and shared file by a single root node. Thus the early download for seeding was simultaneous on different nodes. Please note that a diffuse seeding of a give content lead to a higher content distribution performance and thus for any further download.

7.12 Integration of P2P in AXMEDIA for C2C

A stand alone Java based P2P tool without Web Service server.
It has to be capable to access to the AXMEDIS P2P Query Service Server by means of a Web Service client.
What has to be done:

- It has to work as the AXEPTool on AXOID

- It has to allow to make simple queries on the AXMEDIS P2P Query Service Server by means of a Web Service client. The query has to expose that is performed by an AXMEDIA tool.
- It has to allow to provide results of the query, direct selection of the file to be downloaded, selecting the line in the query result



AXMEDIA P2P Client

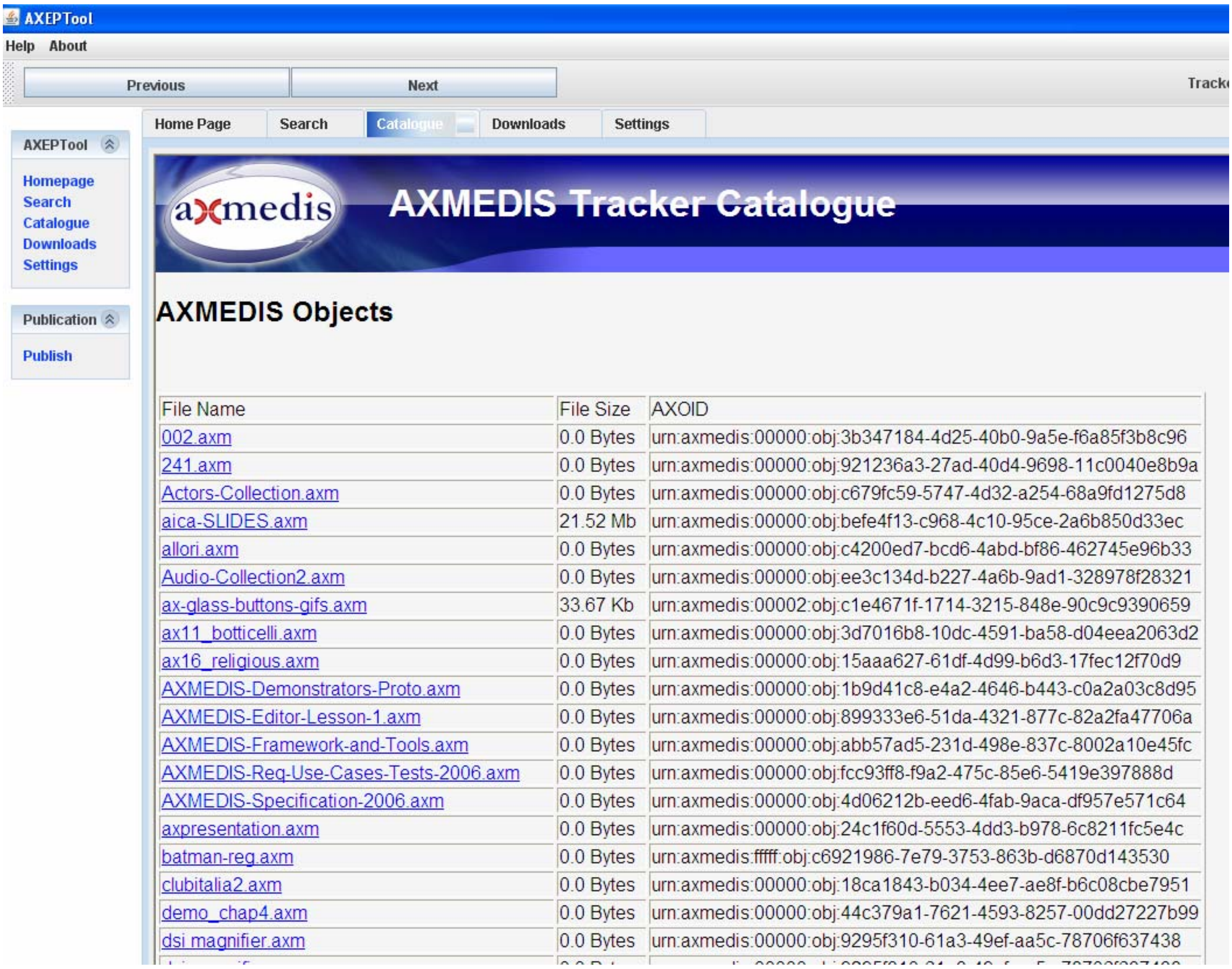
Published Objects

Simple query results interface

Simple query interface

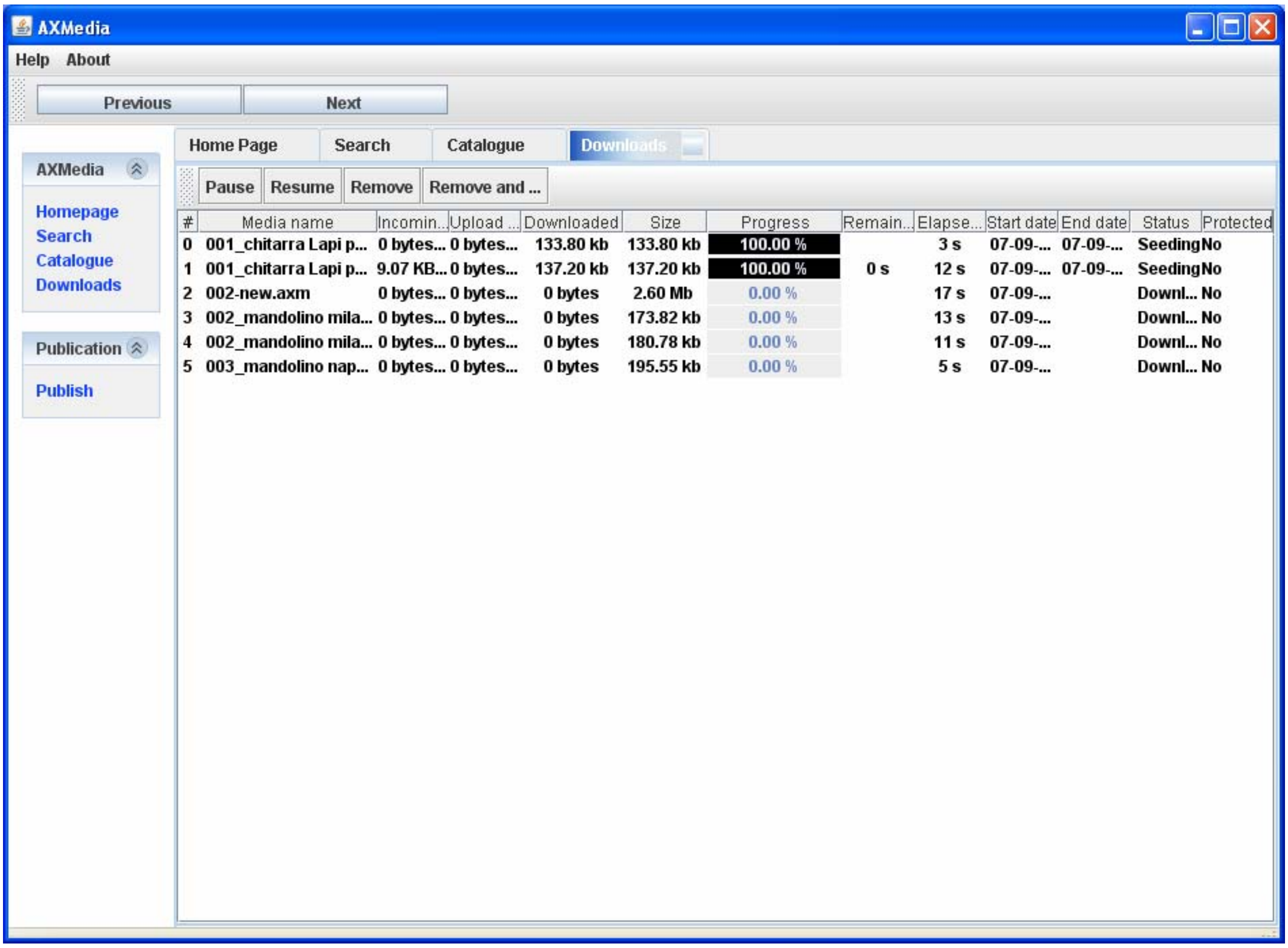
Downloading

Monitoring Tools and User Interface



AXMEDIS Objects

File Name	File Size	AXOID
002.axm	0.0 Bytes	urn:axmedis:00000:obj:3b347184-4d25-40b0-9a5e-f6a85f3b8c96
241.axm	0.0 Bytes	urn:axmedis:00000:obj:921236a3-27ad-40d4-9698-11c0040e8b9a
Actors-Collection.axm	0.0 Bytes	urn:axmedis:00000:obj:c679fc59-5747-4d32-a254-68a9fd1275d8
aica-SLIDES.axm	21.52 Mb	urn:axmedis:00000:obj:befe4f13-c968-4c10-95ce-2a6b850d33ec
allori.axm	0.0 Bytes	urn:axmedis:00000:obj:c4200ed7-bcd6-4abd-bf86-462745e96b33
Audio-Collection2.axm	0.0 Bytes	urn:axmedis:00000:obj:ee3c134d-b227-4a6b-9ad1-328978f28321
ax-glass-buttons-gifs.axm	33.67 Kb	urn:axmedis:00002:obj:c1e4671f-1714-3215-848e-90c9c9390659
ax11_botticelli.axm	0.0 Bytes	urn:axmedis:00000:obj:3d7016b8-10dc-4591-ba58-d04eea2063d2
ax16_religious.axm	0.0 Bytes	urn:axmedis:00000:obj:15aaa627-61df-4d99-b6d3-17fec12f70d9
AXMEDIS-Demonstrators-Proto.axm	0.0 Bytes	urn:axmedis:00000:obj:1b9d41c8-e4a2-4646-b443-c0a2a03c8d95
AXMEDIS-Editor-Lesson-1.axm	0.0 Bytes	urn:axmedis:00000:obj:899333e6-51da-4321-877c-82a2fa47706a
AXMEDIS-Framework-and-Tools.axm	0.0 Bytes	urn:axmedis:00000:obj:abb57ad5-231d-498e-837c-8002a10e45fc
AXMEDIS-Req-Use-Cases-Tests-2006.axm	0.0 Bytes	urn:axmedis:00000:obj:fcc93ff8-f9a2-475c-85e6-5419e397888d
AXMEDIS-Specification-2006.axm	0.0 Bytes	urn:axmedis:00000:obj:4d06212b-eed6-4fab-9aca-df957e571c64
axpresentation.axm	0.0 Bytes	urn:axmedis:00000:obj:24c1f60d-5553-4dd3-b978-6c8211fc5e4c
batman-reg.axm	0.0 Bytes	urn:axmedis:fffff:obj:c6921986-7e79-3753-863b-d6870d143530
clubitalia2.axm	0.0 Bytes	urn:axmedis:00000:obj:18ca1843-b034-4ee7-ae8f-b6c08cbe7951
demo_chap4.axm	0.0 Bytes	urn:axmedis:00000:obj:44c379a1-7621-4593-8257-00dd27227b99
dsi magnifier.axm	0.0 Bytes	urn:axmedis:00000:obj:9295f310-61a3-49ef-aa5c-78706f637438



#	Media name	Incomin...	Upload ...	Downloaded	Size	Progress	Remain...	Elapse...	Start date	End date	Status	Protected
0	001_chitarra Lapi p...	0 bytes...	0 bytes...	133.80 kb	133.80 kb	100.00 %		3 s	07-09-...	07-09-...	Seeding	No
1	001_chitarra Lapi p...	9.07 KB...	0 bytes...	137.20 kb	137.20 kb	100.00 %	0 s	12 s	07-09-...	07-09-...	Seeding	No
2	002-new.axm	0 bytes...	0 bytes...	0 bytes	2.60 Mb	0.00 %		17 s	07-09-...		Downl...	No
3	002_mandolino mila...	0 bytes...	0 bytes...	0 bytes	173.82 kb	0.00 %		13 s	07-09-...		Downl...	No
4	002_mandolino mila...	0 bytes...	0 bytes...	0 bytes	180.78 kb	0.00 %		11 s	07-09-...		Downl...	No
5	003_mandolino nap...	0 bytes...	0 bytes...	0 bytes	195.55 kb	0.00 %		5 s	07-09-...		Downl...	No

The AXMEDIA Tool is the P2P application used in B2C via P2P. More requirements about Client/Server Distribution via PC are available in the specification of WP4 related to “AXMEDIS for Distribution via Internet”. The AXMEDIA tool and solution has the above mentioned system and general requirements plus the following.

- Must provide a simple query support that allows simple search queries composition through a simplified GUI, providing results in a simplified format, easy to understand for the final users.

- Must provide complete results exposing AXMEDIS AXOID and metadata.

- Must be easily installable on a wide range of computers and possibly on different platforms such as Windows and MAC.

A stand alone Java based P2P tool without Web Service server and Web Server. AXMEDIA must not include the Web Service and Web Server code, neither in a deactivated form. Both AXEPTool and AXMEDIA must be created with conditional compilation and AXMEDIA project must not be a duplication of AXEPTool’s one.

It has to be capable to access to the AXMEDIS P2P Query Service Server by means of a Web Service client.

What has to be done:

- It has to work as the AXEPTool on AXOID

- It has to allow making simple queries on the AXMEDIS P2P Query Service Server by means of a Web Service client. The query has to expose that is performed by an AXMEDIA tool.
- It has to allow to provide results of the query, direct selection of the file to be downloaded, selecting the line in the query result

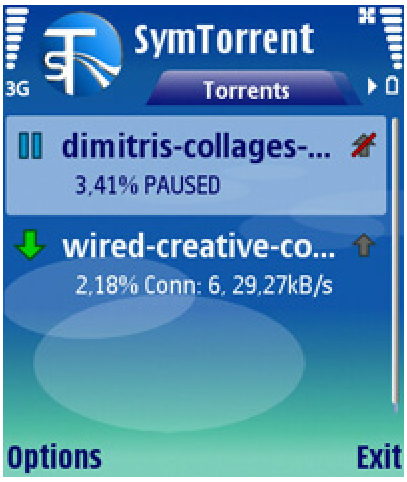
The AXMEDIA tools presents:

- a nicer user interface
- a simplified query user interface
- a simplified monitoring tool and user interface

AXMEDIA does not include an integrated WebService and can manage a less number of connections (4 up/4 down) than the AXEPTool (50 up/10 down). AMXEDIA does not provide a settings page.

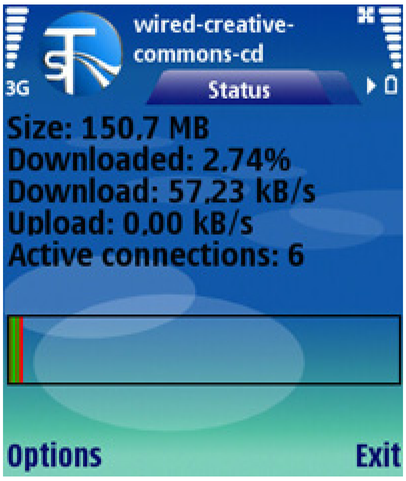
7.13 AXMEDIS P2P Client for Mobile

AXMEDIS P2P Network can be accessed by mobile devices too, featuring a bittorrent client. A test case was conducted using a Nokia 6630 (GSM/UMTS) mobile phone running Symtorrent Symbian OS client (http://symtorrent.aut.bme.hu/). Connection with the AXTrack was obtained with a standard APN configuration.



dimitris-collages-...
3,41% PAUSED

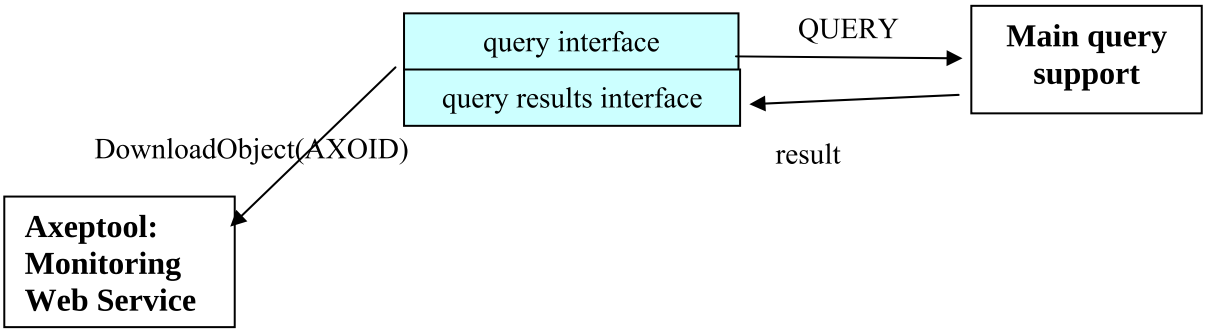
wired-creative-co...
2,18% Conn: 6, 29,27kB/s



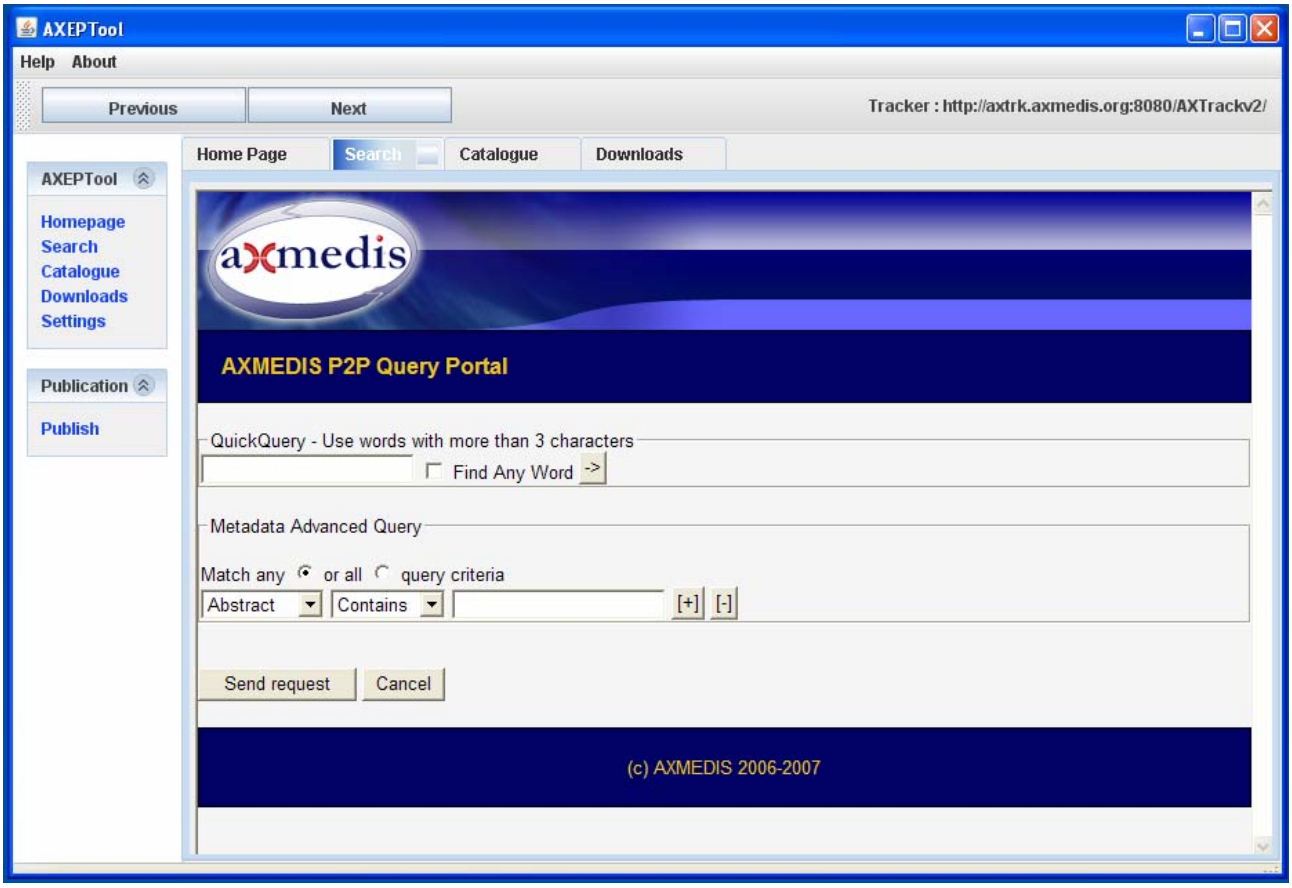
Size: 150,7 MB
Downloaded: 2,74%
Download: 57,23 kB/s
Upload: 0,00 kB/s
Active connections: 6

Interoperability between Symtorrent and Mediaplayer is being developed so to allow video/audio rendering on mobile devices of downloaded digital contents.

7.14 Query Interface/Query Results Interface (DSI resp)



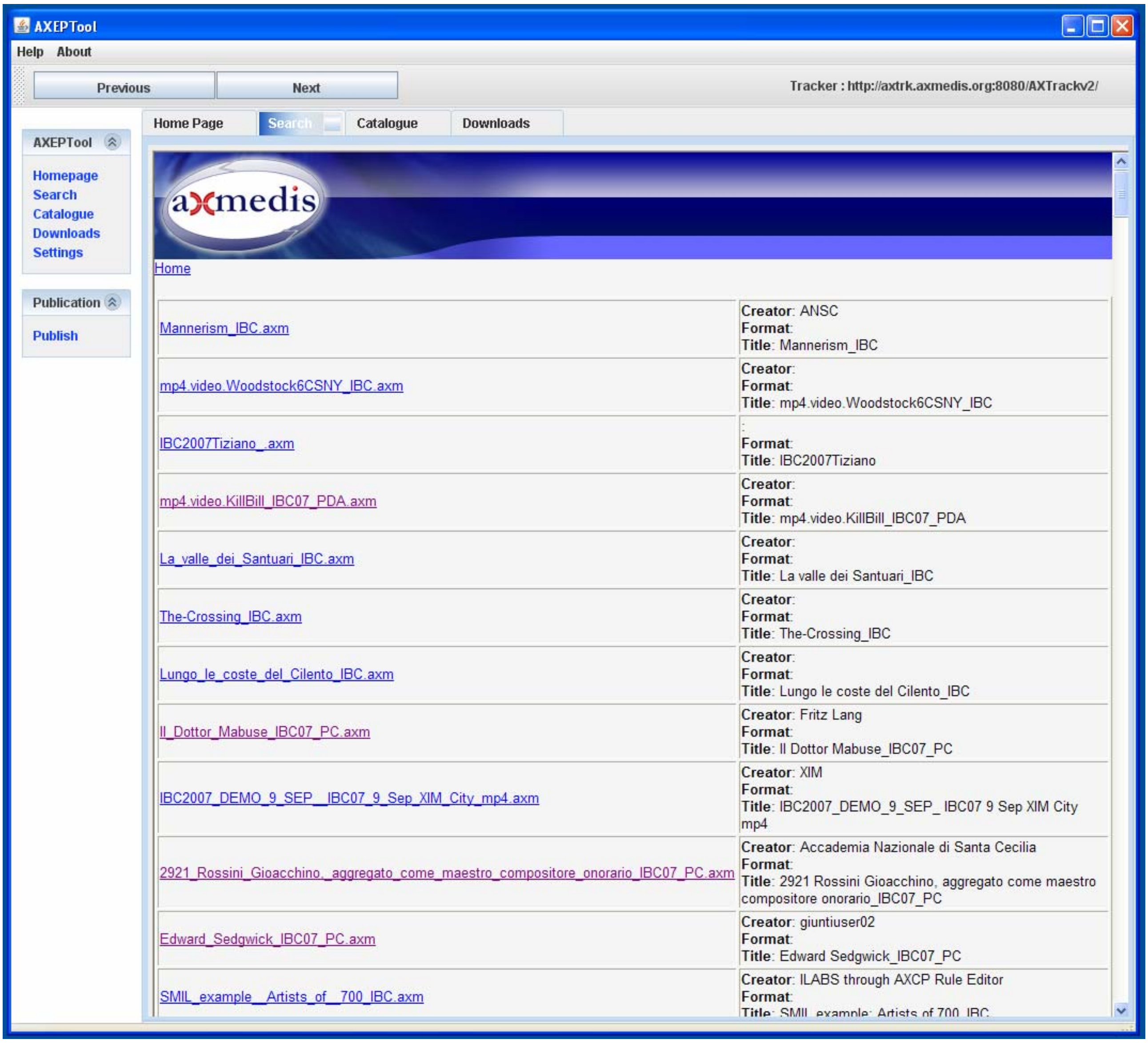
The interface for the AXMEDIS p2P Query portal is reported in the following.
The interface allow to make simple queries on textual basis or even more complex queries in which one may select the fields on which the query has to be performed.



Tracker : http://axtrk.axmedis.org:8080/AXTrackv2/

QuickQuery - Use words with more than 3 characters

Metadata Advanced Query



Mannerism_IBC.axm	**Creator**: ANSC **Format**: **Title**: Mannerism_IBC
mp4.video.Woodstock6CSNY_IBC.axm	**Creator**: **Format**: **Title**: mp4.video.Woodstock6CSNY_IBC
IBC2007Tiziano_.axm	: **Format**: **Title**: IBC2007Tiziano
mp4.video.KillBill_IBC07_PDA.axm	**Creator**: **Format**: **Title**: mp4.video.KillBill_IBC07_PDA
La_valle_dei_Santuari_IBC.axm	**Creator**: **Format**: **Title**: La valle dei Santuari_IBC
The-Crossing_IBC.axm	**Creator**: **Format**: **Title**: The-Crossing_IBC
Lungo_le_coste_del_Cilento_IBC.axm	**Creator**: **Format**: **Title**: Lungo le coste del Cilento_IBC
Il_Dottor_Mabuse_IBC07_PC.axm	**Creator**: Fritz Lang **Format**: **Title**: Il Dottor Mabuse_IBC07_PC
IBC2007_DEMO_9_SEP__IBC07_9_Sep_XIM_City_mp4.axm	**Creator**: XIM **Format**: **Title**: IBC2007_DEMO_9_SEP_ IBC07 9 Sep XIM City mp4
2921_Rossini_Gioacchino,_aggregato_come_maestro_compositore_onorario_IBC07_PC.axm	**Creator**: Accademia Nazionale di Santa Cecilia **Format**: **Title**: 2921 Rossini Gioacchino, aggregato come maestro compositore onorario_IBC07_PC
Edward_Sedgwick_IBC07_PC.axm	**Creator**: giuntiuser02 **Format**: **Title**: Edward Sedgwick_IBC07_PC
SMIL_example__Artists_of__700_IBC.axm	**Creator**: ILABS through AXCP Rule Editor **Format**: **Title**: SMIL example: Artists of 700_IBC

The results of the query are the main metadata, in addition on the left side it is possible to perform a click to have the activation of the P2P downloading.

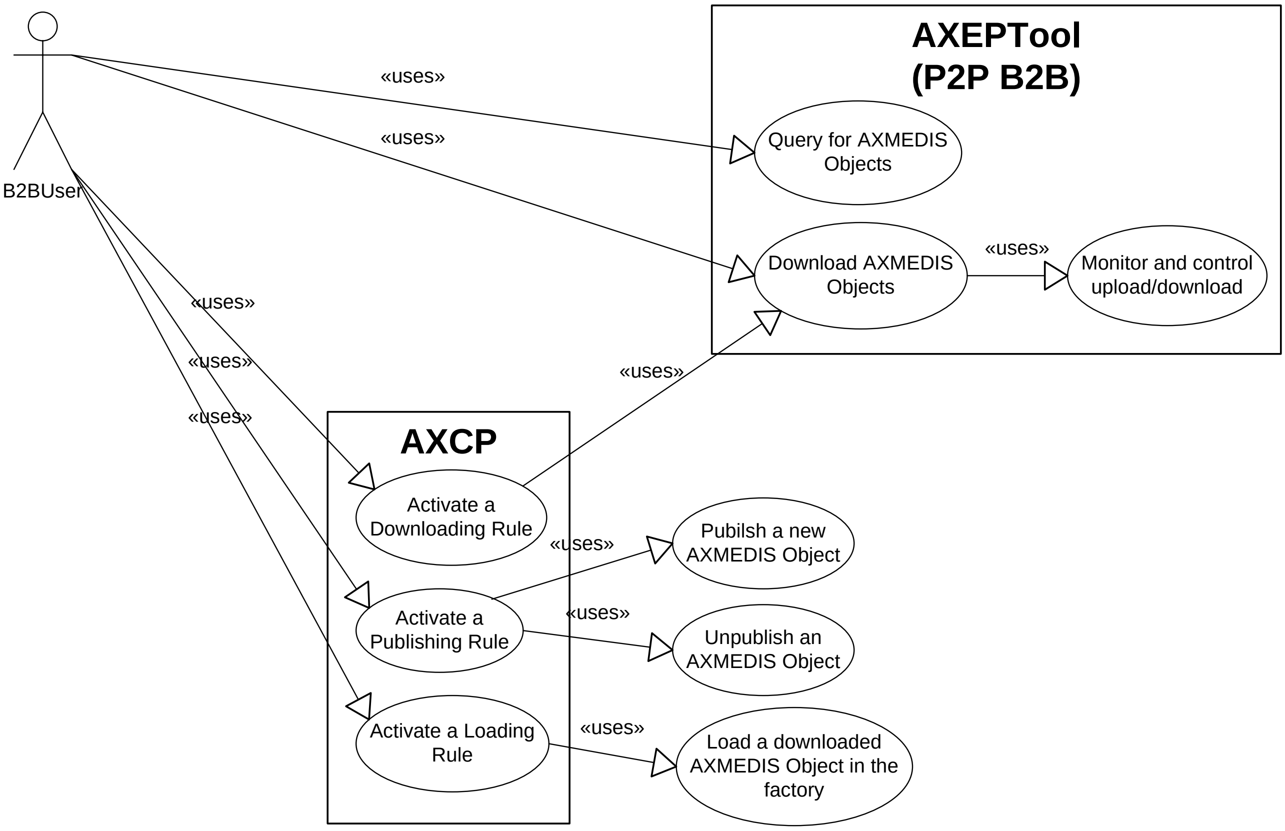
The window of the Query Support can be very easily substituted with any other web page, for example that of TISCALI portal for the download from p2P starting from their Media Club content distribution and sale portal.

7.15 Usage walk-through

In this section, the main use cases have been depicted in order to clarify functionalities to be exposed by AXEPTool (P2P B2B client).
In the following use-case diagram the main B2B user action have been included.

The use cases of main interest are those considered for AXEPTool:

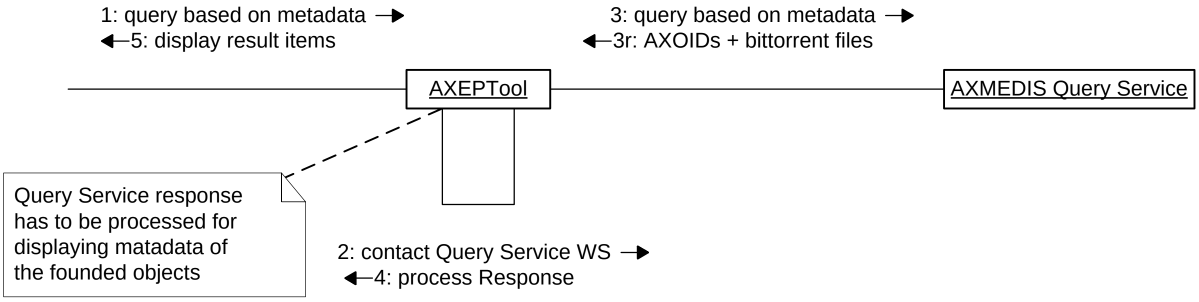
- The user can query for content directly on the client interface (even embedding a web-page rendering);
- The user can command to initiate the download process for a given query result; this use case include the typical action of monitoring and controlling the active downloading process.


AXEPTool
(P2P B2B)
«uses»
«uses»
Query for AXMEDIS Objects
B2BUser
Download AXMEDIS Objects
«uses»
Monitor and control upload/download
«uses»
«uses»
«uses»
«uses»
AXCP
Activate a Downloading Rule
«uses»
Pubilsh a new AXMEDIS Object
Activate a Publishing Rule
«uses»
Unpublish an AXMEDIS Object
Activate a Loading Rule
«uses»
Load a downloaded AXMEDIS Object in the factory

The other use cases are reported for two main reasons:

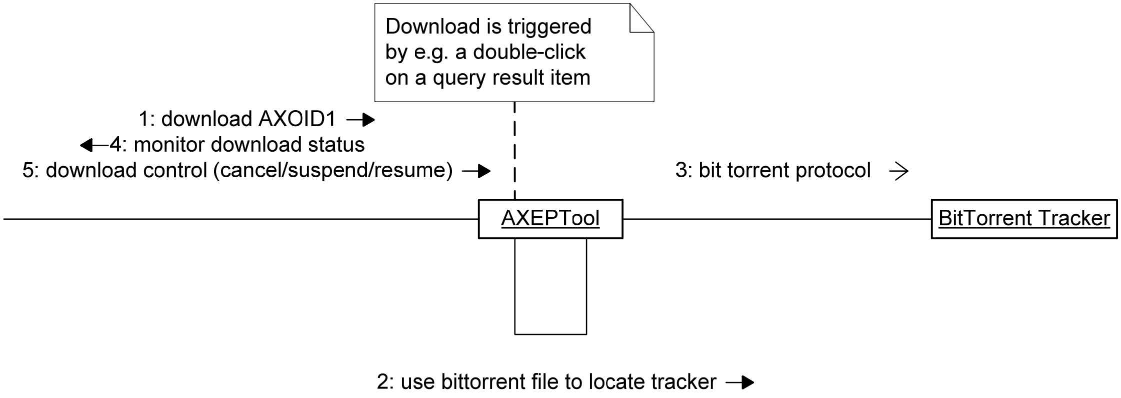
- The capability of controlling the AXEPTool (by a well-known WS interface) remotely. This action is particularly needed in B2B context since download processes can be managed in an automatic manner. This highlights that the downloading technology is reused, while it is accessed by a different way w.r.t. to the GUI.
- The role of the AXCP in completing the full P2P distribution life-cycle. A B2B user can actually create procedures for publish/unpublishing AXMEDIS objects by using scripting language. It can also “script” (i.e. program) what to import from the downloaded objects in the factory database for further usage.

In the following basic collaboration diagram are reported in order to show AXEPTool P2P client functionalities for query and download.


1: query based on metadata →
←5: display result items
3: query based on metadata →
←3r: AXOIDs + bittorrent files
AXEPTool
AXMEDIS Query Service
Query Service response has to be processed for displaying matadata of the founded objects
2: contact Query Service WS →
←4: process Response

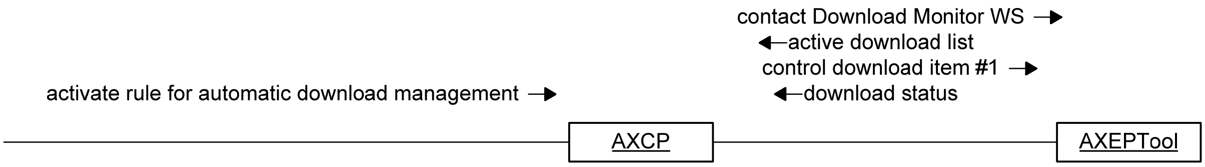
The first diagram shows the interaction between AXEPTool and AXMEDIS Query Service. The action is simply to redirect the user query to the Query Service collecting in the response information regarding the matching objects. The WS response is processed, since all the present metadata have to be attached to the query result items.

Please note that in the response also bittorrent metainfo files are returned.


Download is triggered by e.g. a double-click on a query result item
1: download AXOID1 →
←4: monitor download status
5: download control (cancel/suspend/resume) →
3: bit torrent protocol →
AXEPTool
BitTorrent Tracker
2: use bittorrent file to locate tracker →

In the second diagram the typical action, being consequent to a query, has been presented.

In fact the user can decide of starting the download of an object. In this case the interaction is based on bittorrent protocol established among AXEPTool and the interested BitTorrent Tracker.


contact Download Monitor WS →
←active download list
control download item #1 →
←download status
activate rule for automatic download management →
AXCP
AXEPTool

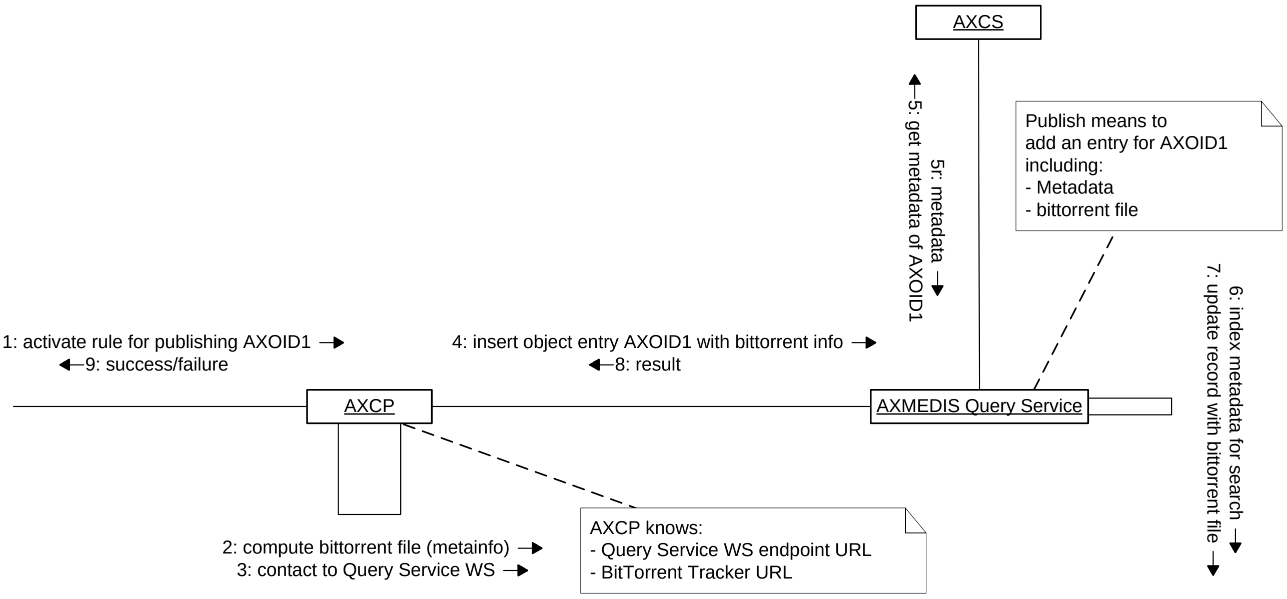
In the last diagram the AXCP is put as an intermediate in the AXEPTool control. In this case it allows the script function to operate as the user on the GUI (i.e. looking what is downloading and controlling those processes).

Usage walk-through for AXMEDIS Query Service and AXCP (EXITECH)

In this diagram a possible solution for avoiding posting of metadata by the B2B user is presented.

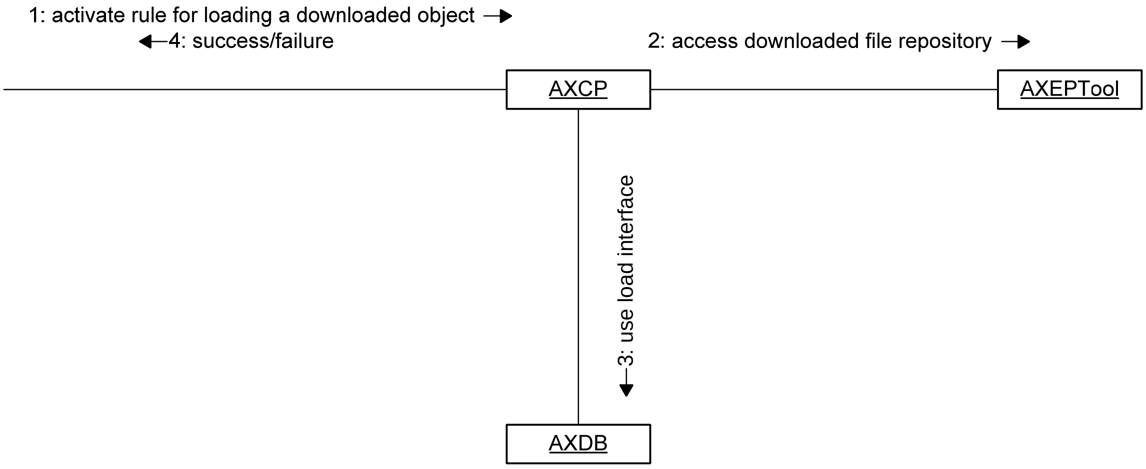
This solution can reuse the knowledge of the AXMEDIS infrastructure regarding an object metadata, since any object that can be distributed/published have been registered to AXCS (with metadata).

The solution of using the information located into the AXCS can be useful especially if the AXCS and the AXMEDIS P2P Query Service are located and maintained by the same organization. That can be AXMEDIS.ORG or by the channel distributor.


AXCS
5: get metadata of AXOID1
5r: metadata
Publish means to add an entry for AXOID1 including:
- Metadata
- bittorrent file
1: activate rule for publishing AXOID1
9: success/failure
4: insert object entry AXOID1 with bittorrent info
8: result
AXCP
AXMEDIS Query Service
6: index metadata for search
7: update record with bittorrent file
2: compute bittorrent file (metainfo)
3: contact to Query Service WS
AXCP knows:
- Query Service WS endpoint URL
- BitTorrent Tracker URL

Please note that publishing task is totally carried out by AXCP and its functionalities of accessing WS. AXCP has also to compute bittorrent file.

Another diagram has been produced to explain how import of newly downloaded object can be realized by only using AXCP functionality.
The latter has to access downloaded file repository of AXEPTool, and to use AXDB loading service.


1: activate rule for loading a downloaded object
4: success/failure
2: access downloaded file repository
AXCP
AXEPTool
3: use load interface
AXDB

8 Conclusions

The development of the P2P has been almost completed. The P2P network has been made accessible to all partners since April 2007. A lot of traffic has been performed for sharing content among partners, publication of content and obviously download.

Work that remain to perform:

- A more accurate analysis of performance is needed. To this end some activities of enforcing into the client tools and into AXTRACKER some monitoring and log point has been started. Also the catalogue should be usable providing the possibility of making some ordering and filtering on posted content.
- An additional work is needed to port the P2P solution also on mobile. This activity has not been planned on the early version of the AXMEDIS project.

9 Bibliography

[1] "SETI@home",
http://setiathome.ssl.berkeley.edu/, University of California. Space Sciences Laboratory,1999.

[2] “Distributed.net”, *http://www.distributed.net*

[3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. “Chord: A scalable peer-to-peer lookup service for Internet applications”. Technical Report TR-819, MIT, March 2001

[4] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. “Freenet: A Distributed Anonymous Information Storage and Retrieval System”. *Lecture Notes in Computer Science*, volume 2009, year 2001.

[5] B. Cohen, “Incentives Build Robustness in BitTorrent”, *Proceedings of the Workshop on Economics of Peer-to-Peer Systems.* 2003.

[6] Article on Napster published in Wikipedia. http://en.wikipedia.org/wiki/Napster

[7] Gnutella Protocol Development. http://www.the-gdf.org

[8] http://www.gnutella2.com/index.php/Node_Types_and_Responsibilities

[9] Limewire Web site. http://www.limewire.com/

[10] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM. Volume 13 , Issue 7 (July 1970), Pages: 422 - 426. Year of Publication: 1970 by ACM Press.

[11] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001

[12] P. Maymounkov and D. Mazieres. Kademlia: A peerto -peer information system based on the xor metric. In Proceedings of IPTPS02, Cambridge, USA, March 2002. http://www.cs.rice.edu/Conferences/IPTPS02/.

[13] Matthew Harren, Joseph M. Hellerstein, Ryan Huebsch, Boon Thau Loo, Scott Shenker, and Ion Stoica. Complex Queries in DHT-based Peer-to-Peer Networks. In Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS), March 2002.

[14] GNU WGET Web page. http://www.gnu.org/software/wget/wget.html

[15] Onion Networks. Swarming Technology. http://onionnetworks.com/technology/swarming/

[16] eMule-Project.net - Official eMule Homepage. http://www.emule-project.net/

[17] wxMozilla - Embedding Mozilla in wxWindows. http://wxmozilla.sourceforge.net/

[18] JRex - The Java Browser Component. http://jrex.mozdev.org/

10 Glossary

- **AXCP**: AXMEDIS Content Production
- **AXDB**: AXMEDIS DataBase.
- **AXDBIN**: AXMEDIS DataBase IN, a DB provided with query support where objects loaded from the network are indexed
- **AXDBOUT**: AXMEDIS DataBase OUT, a DB provided with query support where local objects shared in the p2p network are indexed. In Hub peers the AXDBOUT can index objects of other peers.
- **AXEPTool**: In the AXMEDIS framework, the modules/components/tools aimed at production and distribution over a P2P network.
- **AXOB**: AXMEDIS OBject
- **B2B**: Business-to-Business
- **B2C**: Business-to-Consumer
- **CA**: Certification Authority
- **CSS**: Cascading Style Sheets
- **DBIN**: see *AXDBIN*
- **DBOUT**: see *AXDBOUT*
- **DHT**: Distributed Hash Table
- **DRM**: Digital Rights Management
- **GUI**: Graphical User Interface
- **JSP**: Java Server Pages
- **JWSDP:** Java Web Services Development Pack
- **NAT**: Network Address Translator. A service which dynamically re-assign IP addresses to IP packets in order to use one IP public address with more that one local machine. It is often used in conjunction with firewalling to hinder access from outside a local network.
- **P2P**: Peer-to-Peer. An architecture in which all participants can act both as servers and as clients.