Automating Production of Cross Media Content for Multi-channel Distribution
www.AXMEDIS.org

DE4.4.1.2
Content Sharing and Production on P2P, first update

Version: 1.12
Date: 05/10/2005
Responsible: DSI (revised and approved by the 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: DE4.4.1.2 Contractual Date of Delivery: M24 Actual Date of Delivery: 29/09/2006 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

AXMEDIS Copyright Notice

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:

- how to share contents in a network of B2B participants and how this content can be searched and indexed.
- How to deliver large objects with an efficient solution.
- How to provide a technical user interface for editing queries.
- How to 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.

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)

Technical solutions for the AXEPTool. The architecture will integrate the 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, etc.
The R&D plan aims at the following research objectives: a protocol/architecture for automatic discovery of participants, a protocol/architecture for indexing and query distribution over the net, a protocol/architecture for content delivery across the net.
The above goals will be pursued by testing some existing protocols and eventually developing new ad-hoc protocols. Given the low degree of decentralization required by AXMEDIS some protocols/architectures like Napster/Gnutella2/Edonkey2K will be taken into account for the indexing and query distribution. This analysis will probably lead to the choice of one ad-hoc protocol for AXMEDIS. Regarding the protocol/architecture for content delivery the architecture of BitTorrent will be taken as reference model and it is likely to be developed an AXMEDIS client for BitTorrent or in alternative a client for AXMEDIS torrent-like protocol. The work on this area will be decomposed in small iterations cycles of spike solution/test definition/proof-of-concept/simulation where a spike solution consists in exploring a technology running mock-ups and developing on existing libraries, while the simulation is performed with conventional network analysers or ad-hoc models in order to establish the scalability and the performances.

The second objective of the R&D plan is the integration of P2P architecture in a content factory. This objective requires some sub-objectives to be reached:

- A technical solution/architecture to realize exchange databases (AXIN and AXOUT DB) meeting major requirements of AXMEDIS users
- A technical solution/architecture to realize a system of active selection of objects for publishing in the P2P network
- A technical solution to realize a system of active selection of objects for loading third-party AXMEDIS objects in the factory
- A complete integration of distribution operations on P2P with fingerprinting systems in order to check correctness of exchanged objects
- A complete integration of tools and engines designed with an external workflow system in order to automate publishing and download according to pre-definite activity models.

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

The research in this task will consist in the experimentation of GUI, prototypes, and in the integration of meta-data in the P2P message based architecture, the management of metadata mapping. This part is linked to the Loader and Saver and to the Query Support in the AXMEDIS Database. 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 have to include:

- costs and DRM rules, for each action a price, play, excerpts, redistribution, resizing, distribution on a different area, validity of the DRM rule and copyright coverage, etc.
- available languages if there is speech or text
- range of age suggested
- business model suggested
- time of delivery and availability in terms of first delivering, if not ready
- type of delivery: on-line, offline, etc.
- if on-line time of downloads or acquisition
- cultural level
- subject, description of content with simple metadata
- Textual description of subject and evolution

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.

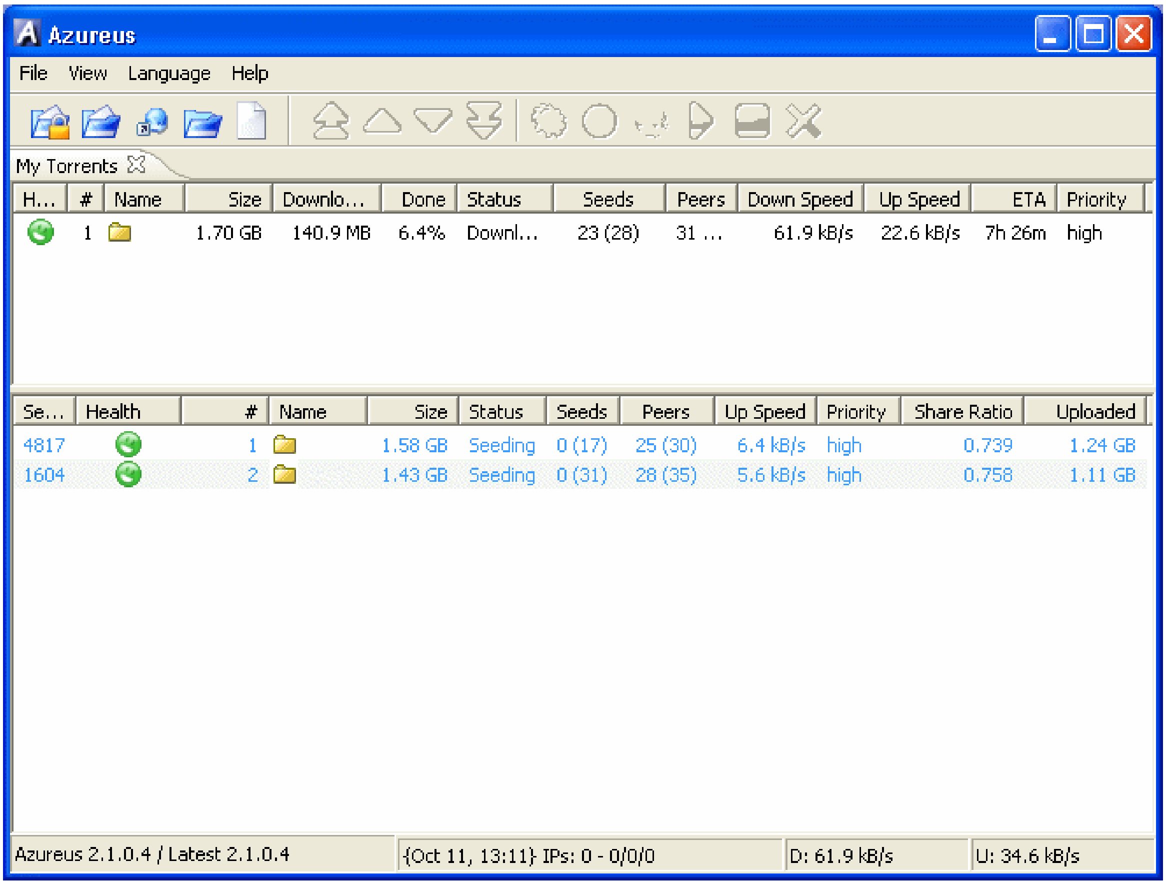
Clients who have finished downloading the file may also choose to act as seeders, providing a complete copy of the file. After the torrent file is created, a link to it is placed on a website or elsewhere, and it is registered with a tracker. BitTorrent trackers maintain lists of the clients currently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

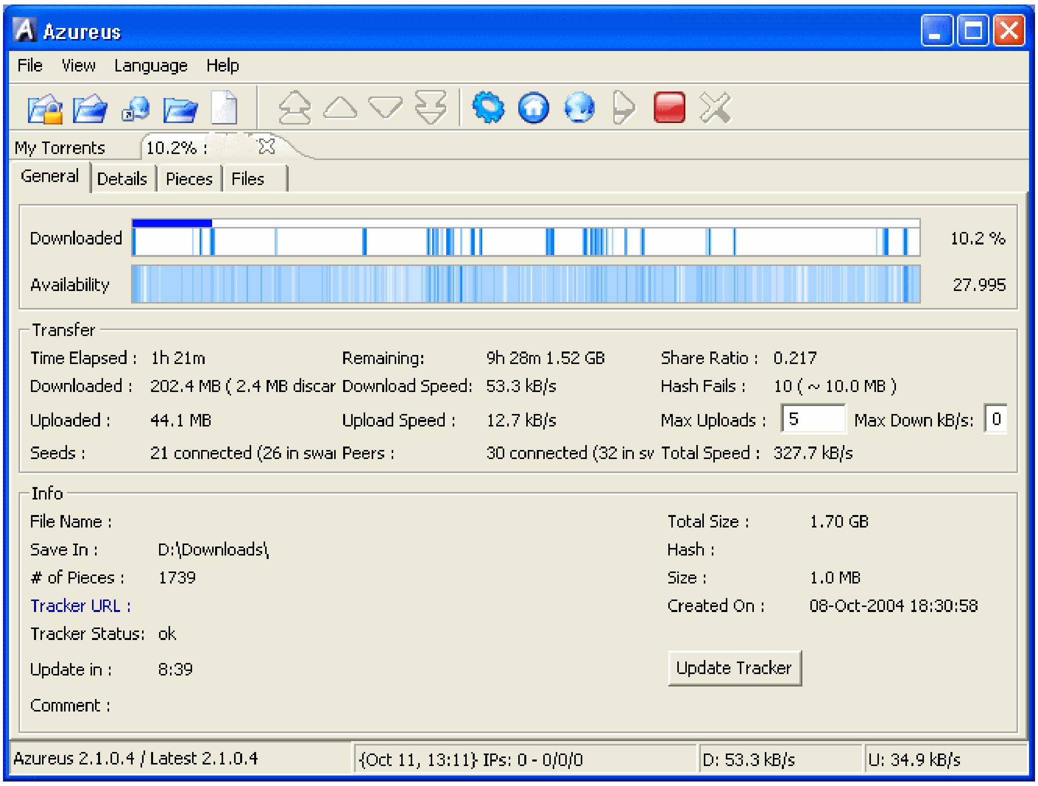


H...	#	Name	Size	Downlo...	Done	Status	Seeds	Peers	Down Speed	Up Speed	ETA	Priority
●	1		1.70 GB	140.9 MB	6.4%	Downl...	23 (28)	31 ...	61.9 kB/s	22.6 kB/s	7h 26m	high

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 swai Peers : 30 connected (32 in sv Total Speed : 327.7 kB/s

Info

File Name :

Save In : D:\Downloads\

# of Pieces : 1739

Tracker URL :

Tracker Status: ok

Update in : 8:39

Comment :

Total Size : 1.70 GB

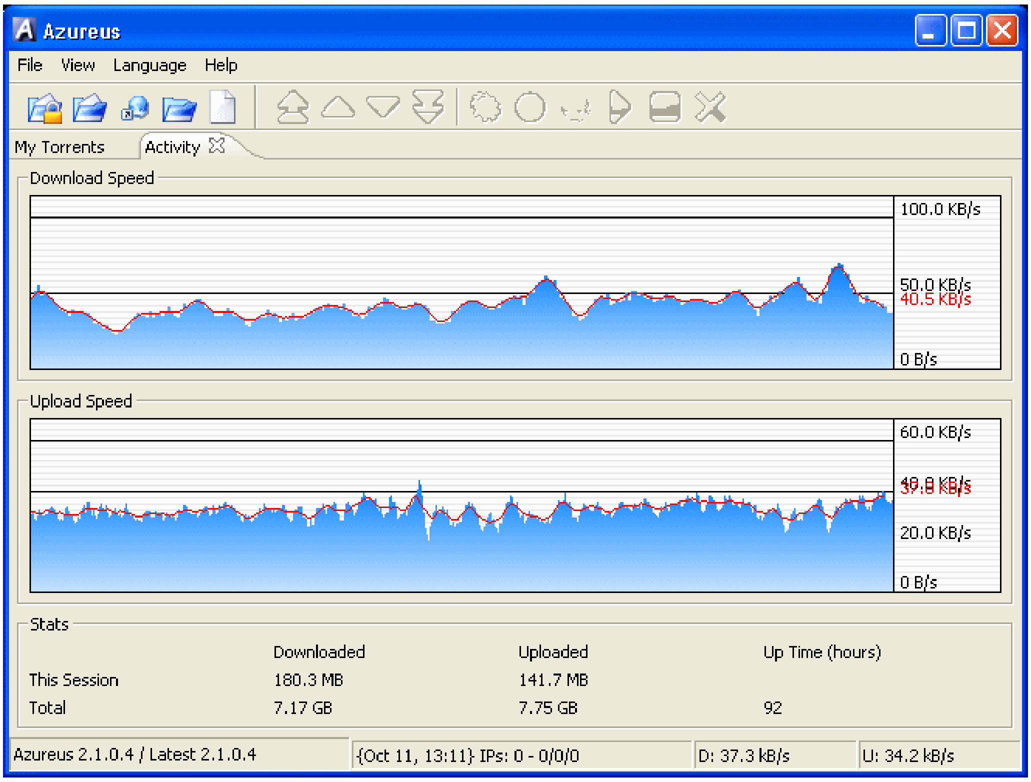
Hash :

Size : 1.0 MB

Created On : 08-Oct-2004 18:30:58

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.

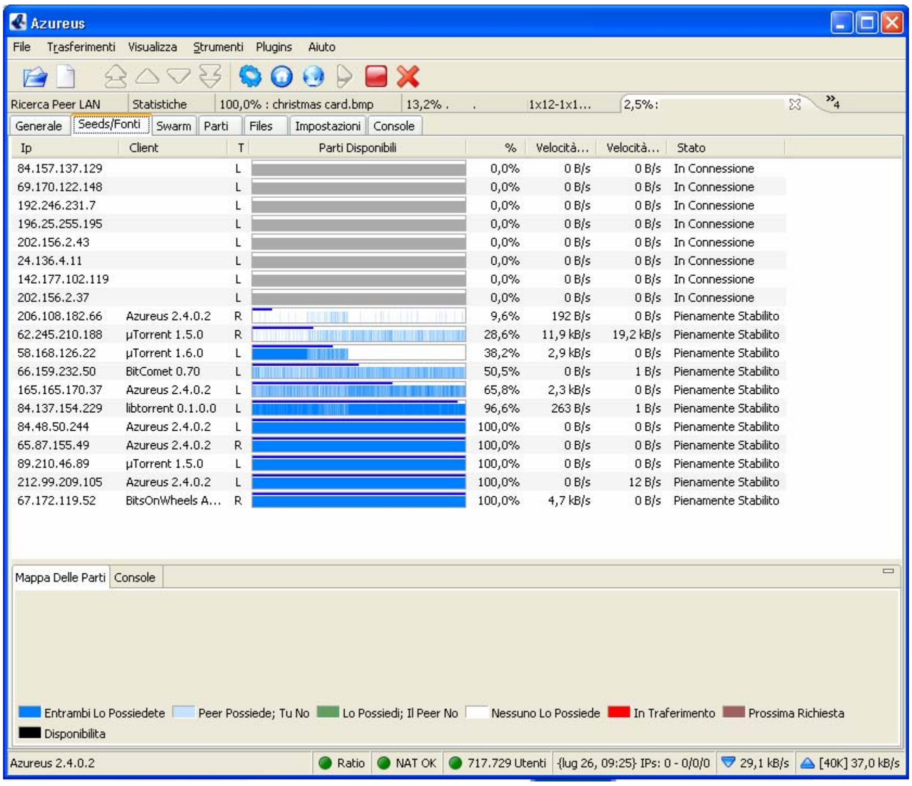


Stats

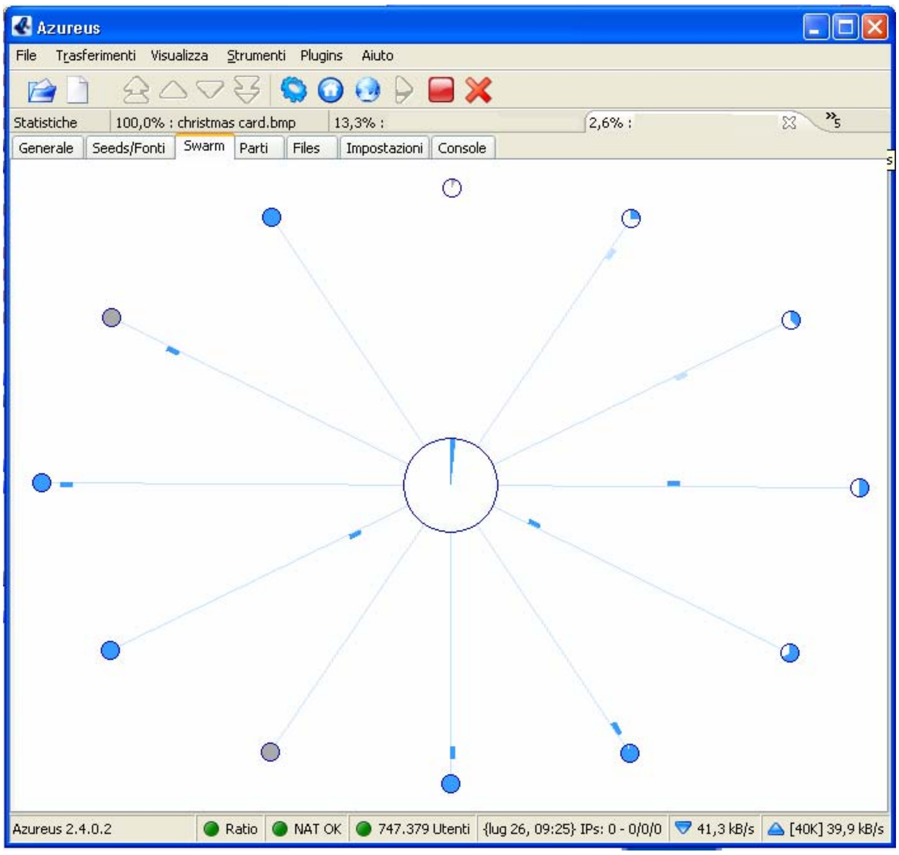
	Downloaded	Uploaded	Up Time (hours)
This Session	180.3 MB	141.7 MB	
Total	7.17 GB	7.75 GB	92

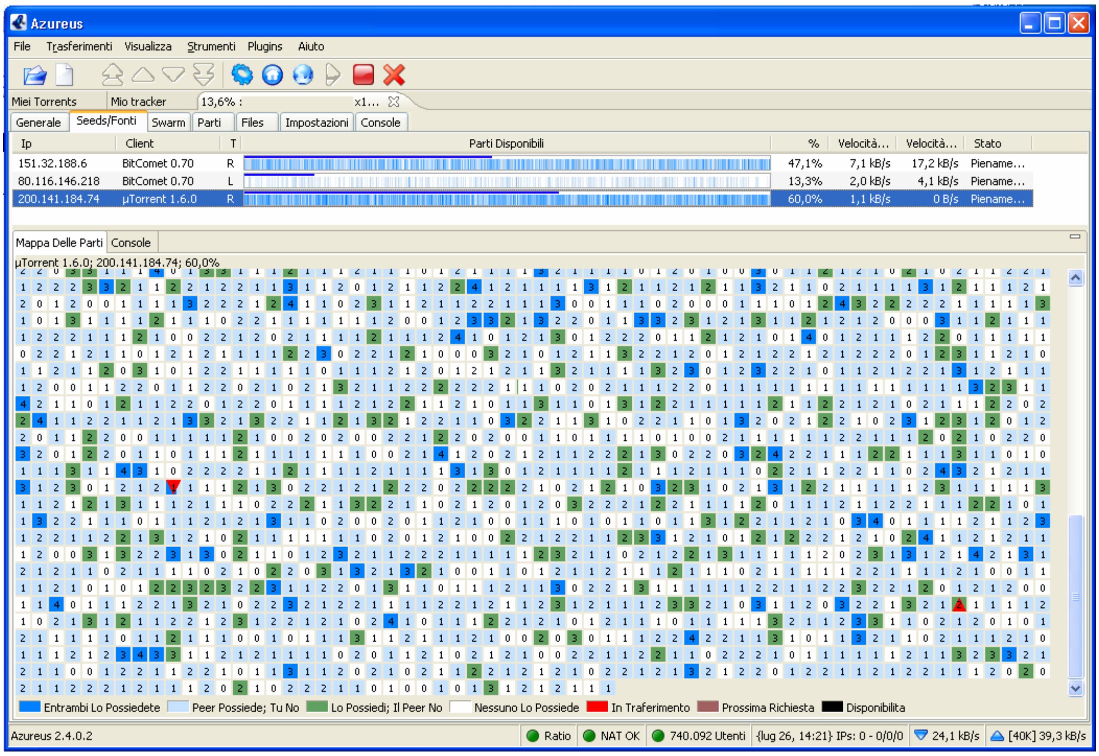
Azureus 2.1.0.4 / Latest 2.1.0.4 {Oct 11, 13:11} IPs: 0 - 0/0/0 D: 37.3 kB/s U: 34.2 kB/s

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.


Azureus
File Trasferimenti Visualizza Strumenti Plugins Aiuto
Ricerca Peer LAN | Statistiche | 100,0% : christmas card.bmp | 13,2% . . | 1x12-1x1... | 2,5% :
Generale | Seeds/Fonti | Swarm | Parti | Files | Impostazioni | Console
Mappa Delle Parti | Console
Entrambi Lo Possiedete
Peer Possiede; Tu No
Lo Possiedi; Il Peer No
Nessuno Lo Possiede
In Traferimento
Prossima Richiesta
Disponibilita
Azureus 2.4.0.2 | Ratio | NAT OK | 717.729 Utenti | {lug 26, 09:25} IPs: 0 - 0/0/0 | 29,1 kB/s | [40K] 37,0 kB/s

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


Azureus
File Trasferimenti Visualizza Strumenti Plugins Aiuto
Statistiche | 100,0% : christmas card.bmp | 13,3% : | 2,6% :
Generale | Seeds/Fonti | Swarm | Parti | Files | Impostazioni | Console
Azureus 2.4.0.2 | Ratio | NAT OK | 747.379 Utenti | {lug 26, 09:25} IPs: 0 - 0/0/0 | 41,3 kB/s | [40K] 39,9 kB/s



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%

Entrambi Lo Possiedete · Peer Possiede; Tu No · Lo Possiedi; Il Peer No · Nessuno Lo Possiede · In Traferimento · Prossima Richiesta · Disponibilita

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.

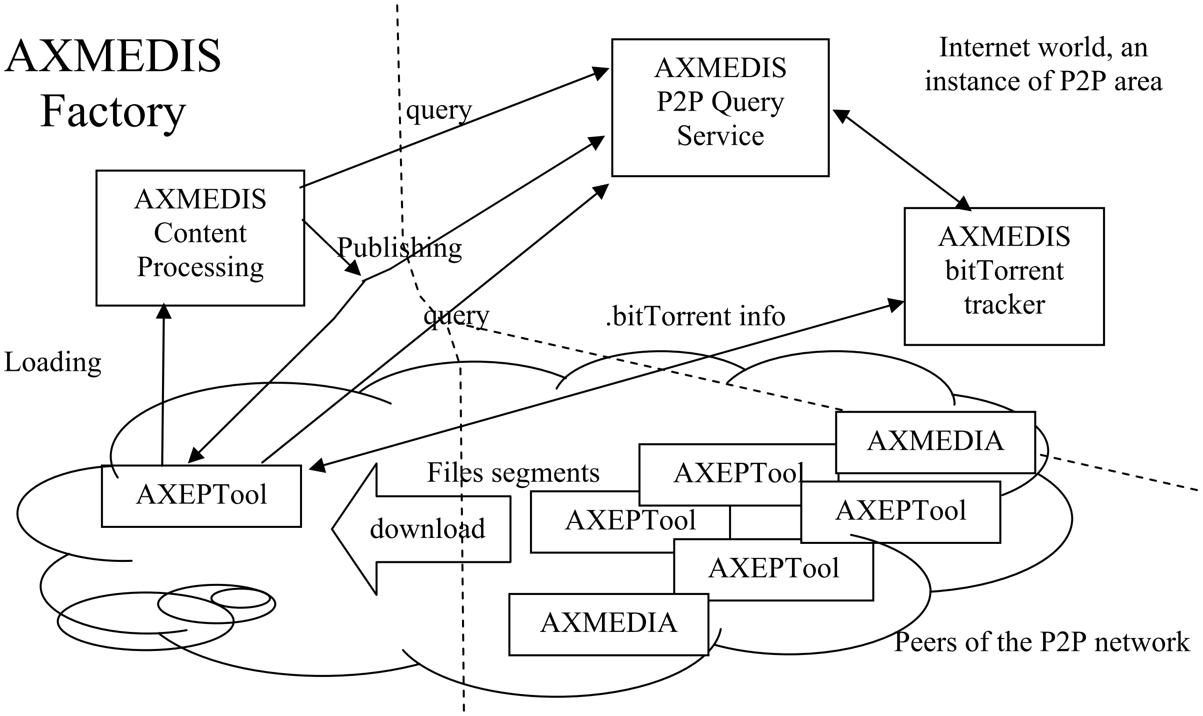
5.4.1) Must provide a simples query support that allows simple search queries composition through a simplified GUI, providing results in a simplified format, easy to understand for the final users.

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.



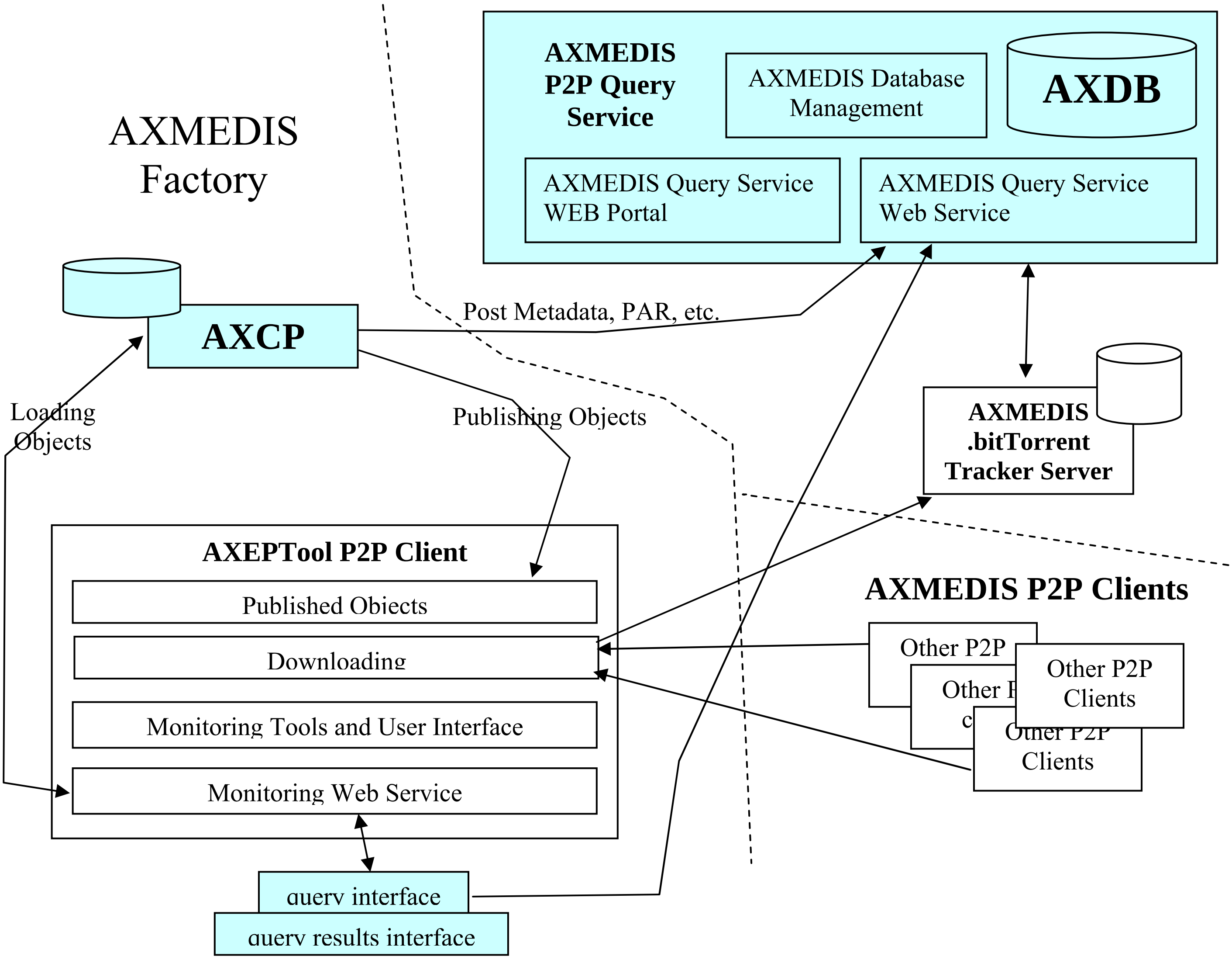
The P2P Clients can be both: AXEPTool and/or AXMEDIA P2P tools.

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

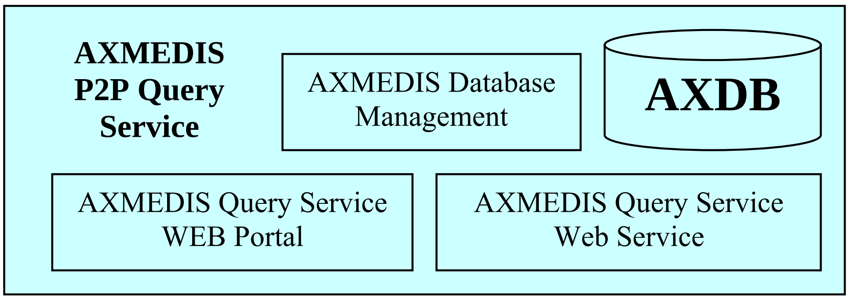
BitTorrent Maker is a tool that allows packaging objects and producing AXMEDIS .bitTorrent information files.
The **AXMEDIS .bitTorrent Maker** has to be usable as a simple executable file tool with its parameters from:

- 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.

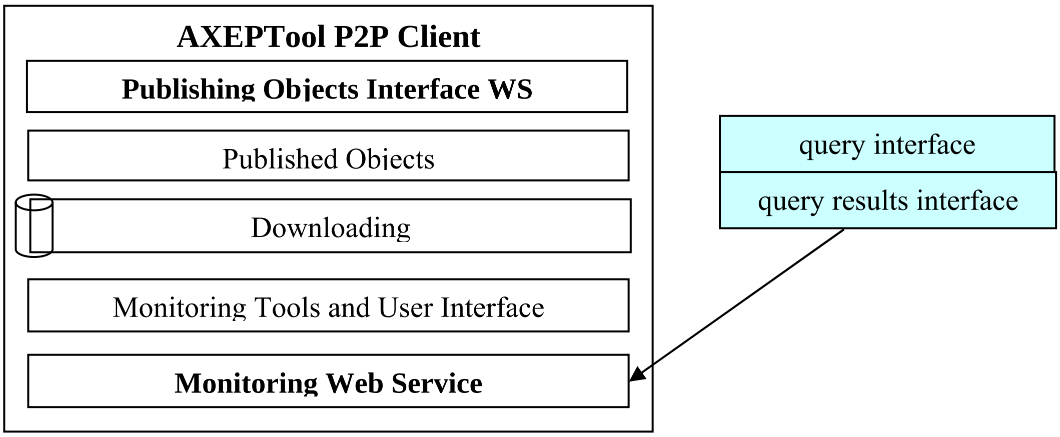
The tracker can be easily integrated into the AXMEDIS P2P Query Service Server or in alternative it is possible to use any active BitTorrent tracker if the protocol for AXMEDIS is not going to be changed.
If we maintain the same protocol we lose the feature of controlling the objects that are using the 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.

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 Can be realized with .bitTorrent Azureus client (see the list in the previous section), which has as major features those presented in the previous table that for the our purpose can be summarized as:

- Developed in Java
- Includes a tracker
- Support protected communication with the Tracker
- Presents a monitor interface

It seems that the present Plugin technology of Azureus does not allow to do much and for sure not to control the downloads and other facts that we need to enforce. So that the work is in effect to the modification of the P2P client.

The AXMEDIS AXEPTool P2P Client:

- P2P query Tool based on C++ tools of DSI
    - Provides a **Query Interface** *that can be realized by what is already present on some other AXMEDIS tools, see Query Support in the Specification of database*
    - Provides a **Query Results Interface** in a sophisticated manner allowing presenting:
        - Metadata of the AXMEDIS objects in response of the query, the so called AXInfo
        - A point/button for selecting the objects to be downloaded, for which the download has to start
- Provide a **Publishing Objects Interface WS** for receiving objects from another tool (for example from the AXMEDIS AXCP Egine) 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 an FTP or WEB service or Plug-in to:
        - **publicationObject(URI);** publication of an object, its metadata have been already posted on the AXMEDIS P2P Query Service Server:
            - post/save the AXMEDIS object passed into the **Published Object area of the P2**P environment of the P2P client tool.
        - **BitTorrentInfo URI createPublicationObject(URI)**, publication of an object, its metadata have been already posted on the AXMEDIS P2P Query Service Server. This function service has three functionalities:
            - create the BitTorrent Information file and return it to the AXCP

        - post/save the AXMEDIS object passed into the **Published Object area of the P2**P environment of the P2P client tool.
        - Post the BitTorrent Information file into the Tracker Server of reference identified by the P2P Client Tool.
    - the P2P client tool uses a repository or a directory with the **Published Objects** which is the Published Object area of the P2P Client
    - The objects can also be published manually from the user interface of the AXEPTool. Probably in the P2P Client there is no distinct areas for the Downloaded objects and the objects published. In this case we have a simplification.
- 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 **Monitoring Web Service** with the following functionalities:
    - **PubStatus monitorPublication()**, provides the current status of the objects which are in the **Published Object area of the P2**P environment. It returns a list if IDs, with the number of downloads for each segment, the so called PubStatus:
        - AXOID=45, 1(34), 2(345), ….. that means segment 1 has been downloaded 34 times.
        - AXOID=45225,
    - **AXOIDList getPublicationObjectsList()**, returns the ID of objects which are present in the directory of Published objects
    - **removePublicationObject(AXOID)**, 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.
    - **DwnStatus monitorDownload()**, provides the current status of the downloading objects in the P2P environment. It returns a list if IDs with the current percentage of download, the so called DwnStatus:
        - AXOID=45, status 56%, estimated time to completion (if possible)
        - AXOID=45225, status 100% estimated time to completion (if possible)
    - **AXOIDList getDownloadObjectsList(),** returns the ID of objects in the directory of Downloaded objects
    - **downloadObject(AXOID)**, starts the download of the object specified by the AXOID in the P2P environment; The AXOID field has to be added into the Tracker database.
    - **abortDownloadObject(AXOID)**, aborts/stops the download of the object specified by the ID.
    - **removeDownloadObject(AXOID),** deletes the object specified by the ID from the database/directory of Downloaded 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.
    - **URI getObjectURI(AXOID),** 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 directory/uri.

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

Integration into the AXMEDIS AXEPTool of:

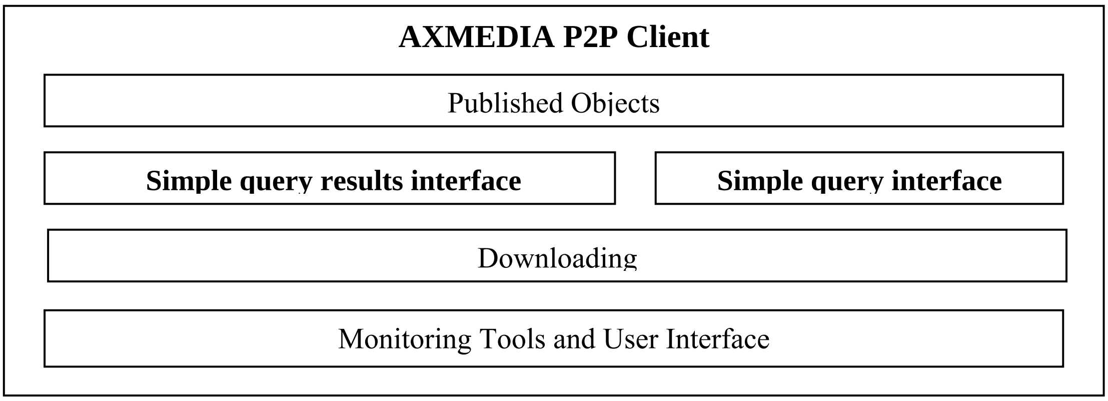
- certification and authentication mechanisms of AXMEDIS to be sure that the AXEPTool is a certified and authorized tool and person.
- A module for estimating fingerprints of resources, connection to the AXCS to verify if the estimated FP correspond to the resource effectively contained in the digital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

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

7.6 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

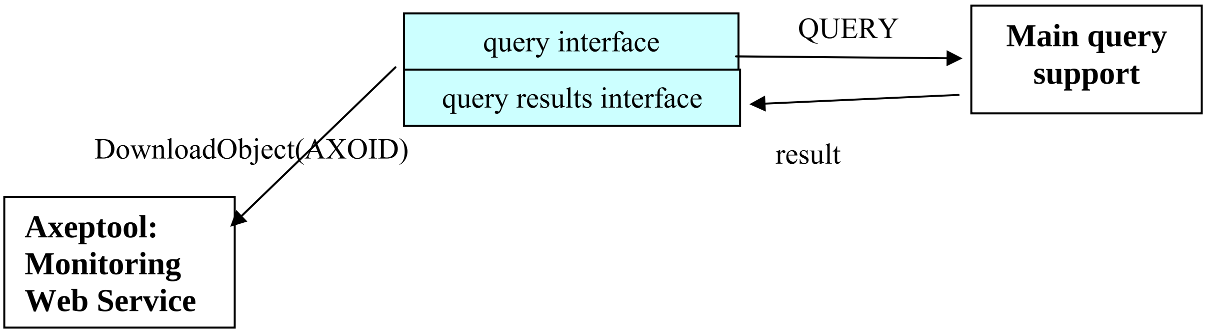
Downloading

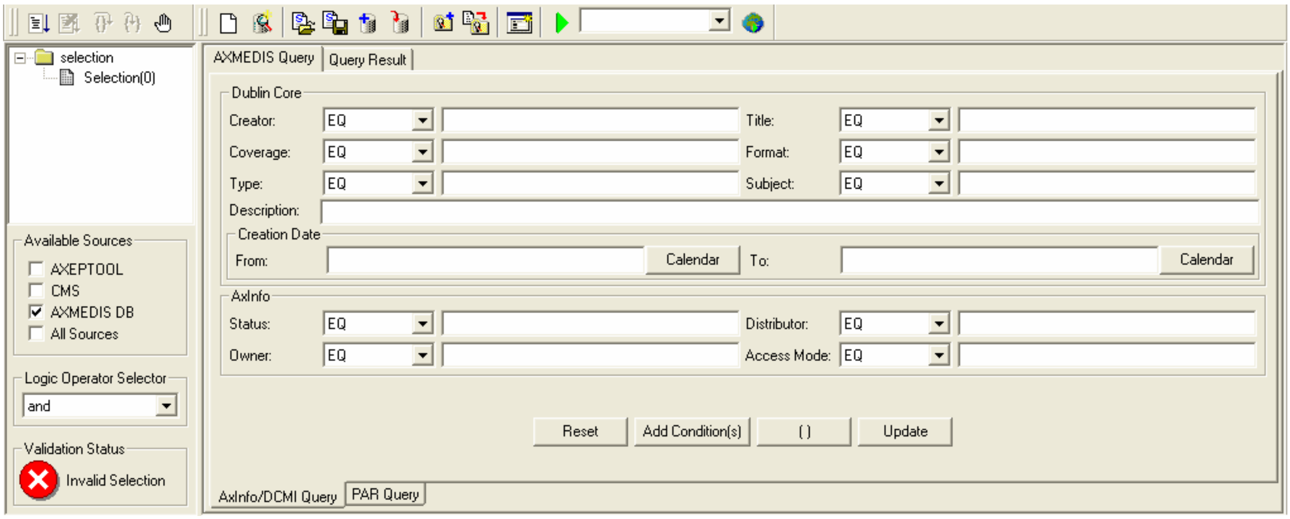
Monitoring Tools and User Interface

The AXMEDIA tools presents:

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

7.7 Query Interface/Query Results Interface (DSI resp)





Available Sources
☐ AXEPTOOL
☐ CMS
☑ AXMEDIS DB
☐ All Sources

Logic Operator Selector: and

Validation Status: Invalid Selection

Dublin Core
Creator: EQ
Title: EQ
Coverage: EQ
Format: EQ
Type: EQ
Subject: EQ
Description:

Creation Date
From: Calendar
To: Calendar

AxInfo
Status: EQ
Distributor: EQ
Owner: EQ
Access Mode: EQ

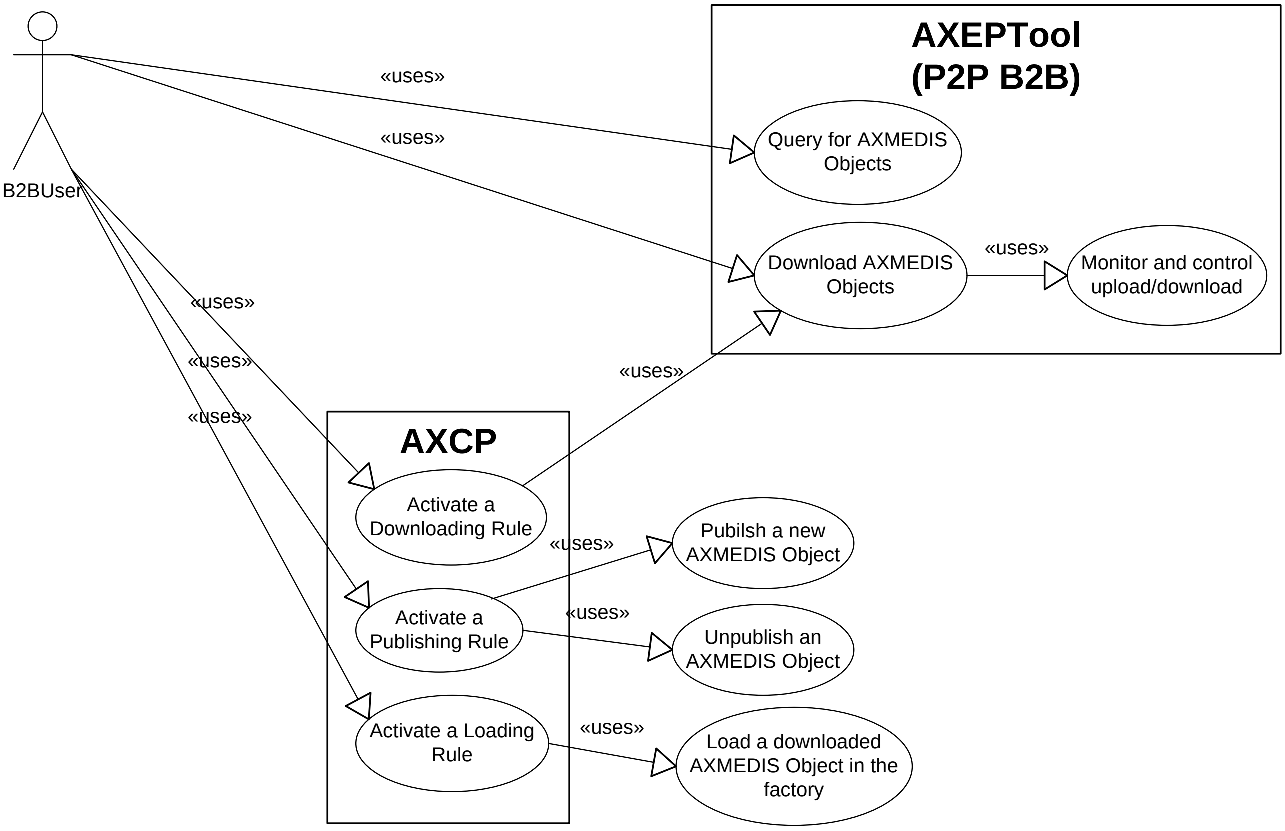
This will be derived from the AXCP query interface.

7.8 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:

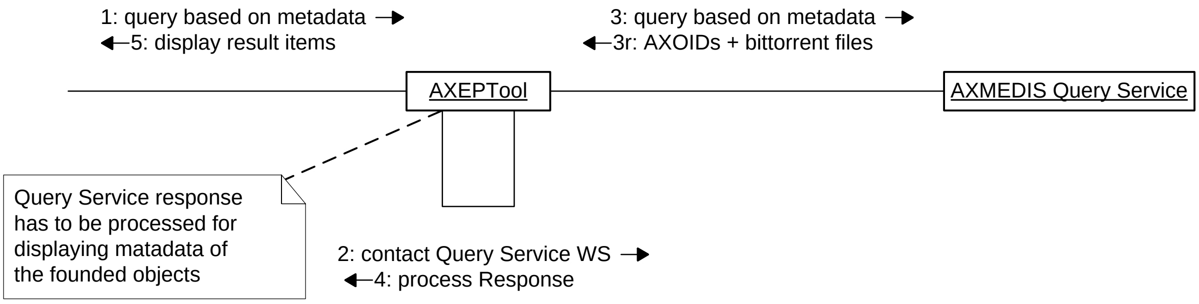
- The user can query for content directly on the client interface (even embedding a web-page rendering);
- The user can command to initiate the download process for a given query result; this use case include the typical action of monitoring and controlling the active downloading process.


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

The other use cases are reported for two main reasons:

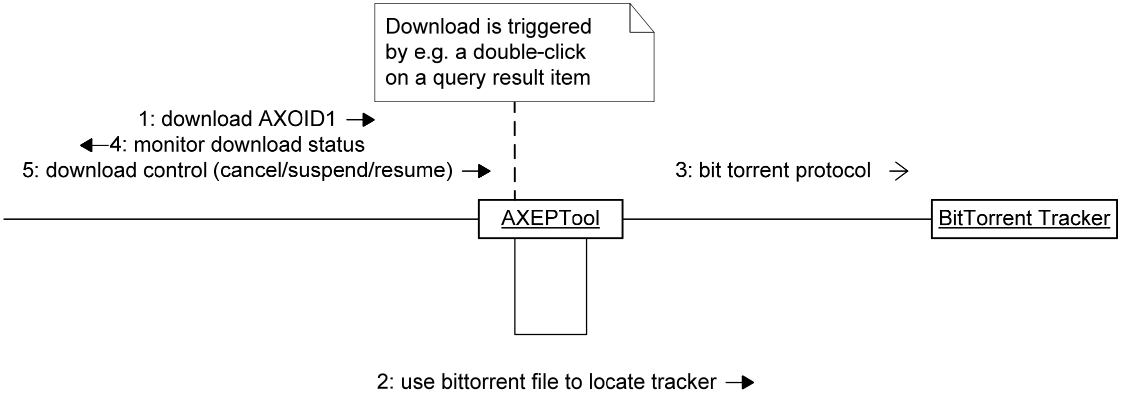
- The capability of controlling the AXEPTool (by a well-known WS interface) remotely. This action is particularly needed in B2B context since download processes can be managed in an automatic manner. This highlights that the downloading technology is reused, while it is accessed by a different way w.r.t. 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

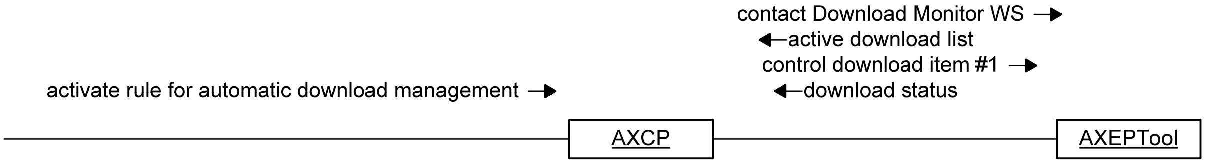
The first diagram shows the interaction between AXEPTool and AXMEDIS Query Service. The action is simply to redirect the user query to the Query Service collecting in the response information regarding the matching objects. The WS response is processed, since all the present metadata have to be attached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

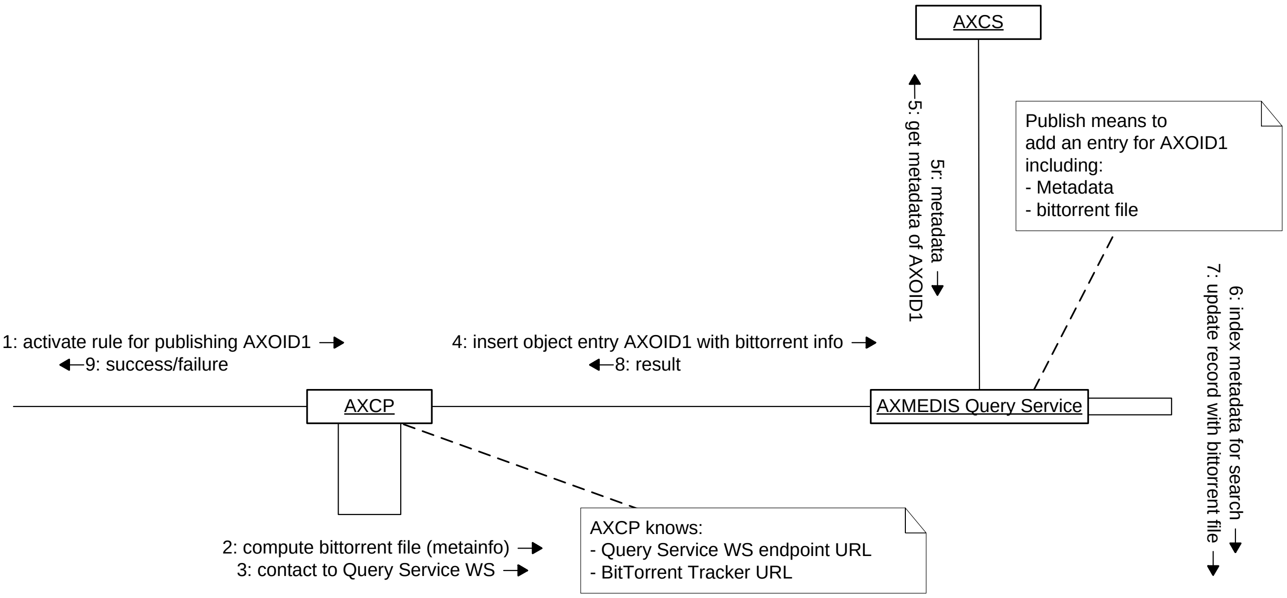
In the last diagram the AXCP is put as an intermediate in the AXEPTool control. In this case it allows the script function to operate as the user on the GUI (i.e. looking what is downloading and controlling those processes).

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.

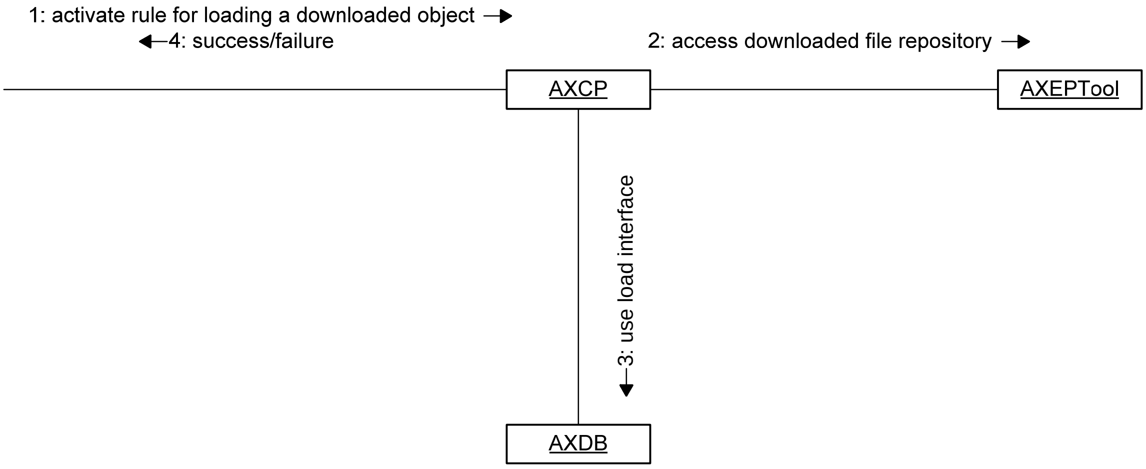
This solution can reuse the knowledge of the AXMEDIS infrastructure regarding an object metadata, since any object that can be distributed/published have been registered to AXCS (with metadata).

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
6: index metadata for search
7: update record with 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
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 Major Components and Related Work

A Company will provide a customisation of the BitTorrent P2P environment and tools for the usage and integrated with AXMEDIS framework for content sharing on PC.

The work has been described in the deliverable DE3.1.2.2.10 and consists of the following major activities:

- **Development of the AXMEDIS P2P Query Service Server (EXITECH)**
  - AXMEDIS Query WEB Portal (EXITECH)
  - AXMEDIS Query WEB Service, for queries (EXITECH)
    - Limitation of number of results especially for queries coming from AXMEDIA tools
  - To put together different communities managed each of them by an AXMEDIS Query Service Server. This will be managed by redistributing the query from one AXMEDIS Query Service Server to others collected in a list and merging the results (EXITECH)

- **Query Tool based on C++ query service in AXCP editor**
    - Interconnection of different AXMEDIS P2P Query Service Servers
    - Integration of the **Query Interface** towards the AXMEDIS P2P Query Service and Query Support into the P2P client, and related documentation.
    - Access to the **Monitoring Web Service** of the AXEPTool to ask for a download in terms of AXOID
- **Customisation of a BitTorrent P2P Client (Azureus) to play the role of AXEPTool** in the AXMEDIS Framework and thus:
    - Usage of the .bittorrent file maker, and customization to work with AXOID
    - Realization and integration of a **Publication Object Interface** Web Service Module to automatize the posting of AXMEDIS objects into the P2P network, and related documentation.
    - Realization and integration of a **Monitoring Web Service and tool Module** to control the status of the downloads and taking the AXMEDIS objects from the P2P network in connection with the AXCP GRID, and related documentation.
    - Provide a Monitoring user interface to see for each AXOID:
        - Progression of percentage downloaded, and number of Kbytes/Mbytes downaloaded, etc.
        - how many streams have been opened (in and out),
        - predicting the time to complete
        - number of completed sources available
        - ….
    - Verification and Authentication of tools (FUPF, DSI)
    - Certification/verification of objects, by using functionalities are already present in Azureus client.
    - Provide a simple HELP in English
    - Provide documentation in Java DOC, and general description of the changes performed into the tools, where and how in terms of AXMEDIS Specification format
- **Customisation of a BitTorrent P2P Client (probably Azureus) to play the role of AXMEDIA** in the AXMEDIS Framework and thus:
    - Realization of simplified **Query Interface t**owards the AXMEDIS P2P Query Service WS with simple interface for final users, direct download in terms of AXOID. This can be developed by starting from the AXMEDIS WEB query interface developed by EXITECH.
    - Realisation of a nice and appealing/simplified user interface to access at P2P data.
    - Provide a Monitoring user interface to see for each AXOID:
        - Progression of percentage downloaded, and number of Kbytes/Mbytes downaloaded, etc.
        - how many streams have been opened (in and out),
        - predicting the time to complete
        - number of completed sources available
        - ….
    - Provide a simple HELP in English
    - Provide documentation in Java DOC, and general description of the changes performed into the tools, where and how in terms of AXMEDIS Specification format
- **Customisation of the BitTorrent Tracker Server** (of PHPBT) to serve the P2P B2B and B2C networks for content distribution in AXMEDIS:
    - **Allowing working on AXOID**
    - Allowing to control the access of non AXMEDIS objects into the P2P network, enabling them or excluding them, accepting the upload of bittorrent files only coming form AXEPTools and/or AXMEDIA tool, or from all bittorrent clients.
    - Realisation of an User interface to access at the statistical information (production of related documentation):
        - how many downloads for each AXOID,
        - list of them, for each AXOID:

        - date and time in which each download has been performed,
        - duration of the download,
        - AXEPTools from where it has been downloaded, etc.
    - Realisation of an Web Service to access at the above mentioned statistical information (production of related documentation):
    - Integration of different Trackers for one AXMEDIS Query Service Server. This is not a problem since the bittorrent file has the tracker URL inside
    - Migration of information from one Tracker to others. This could be avoided
    - Provide documentation in Java DOC, and general description of the changes performed into the tools, where and how in terms of AXMEDIS Specification format
- **Set up, test and deployments of the above tools and components plus the whole P2P environment in collaboration with other AXMEDIS partners:**
    - Set up and deploy of some instances for the serves and many instances for the clients.
    - Testing the whole solution in presence of significant workload, and related documentation and in connection with the other related AXMEDIS tools.
    - For the above mentioned AXMEDIA and AXEPTool P2P clients simple packages to allow their easy installation will be also provided.

Trial, as:

- Set up and Installation of visible and accessible AXMEDIS P2P Query Service Servers
- Set up and Installation of visible and accessible .bitTorrent Tracker Servers
- Putting accessible for download AXEPTool clients
- Putting accessible for download AXMEDIA clients

Tests to be done:

- The possibility of using the P2P of AXMEDIA tools for the sharing of file for the AXEPTool for P2P B2B
- The possibility of allowing or not the sharing on the same community both AXMEDIS and other files in other formats. A clear mark has to be presented on the results of the query to put in evidence which are the AXMEDIS file with respect to those that are single resources…or that in any are not AXMEDIS file.

Since some partners of the AXMEDIS consortium will work on the same tools (for the protection aspects and for the integration aspects), a support for their usage and adaptation of the P2P Environment will be also provided by subcontractor to the AXMEDIS consortium.

The above mentioned tools will be:

- Developed in English for user interface, keywords, and configuration and help
- Derived from Open Source code. That are Azureus and BT Tracker.
    - source code of the derived tools will be posted into the AXMEDIS Framework repository since the beginning.
    - No IPR has to be pending on the code, no patent from software coder in this phase, no pending royalty for the future usage of the tools, etc.
- Using the following code provided by AXMEDIS:
    - Guidelines to work with the CVS/SVN of the AXMEDIS framework guidelines that are public on the AXMEDIS portal.
    - Java version of the query interface (EXITECH)
    - Web Service client for the WS of the AXMEDIS Query Service Web Service (EXITECH)
    - …

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.