P2PTV PROJECT

P2P Based Video Delivery Infrastructure technical specifications

Introduction:

The goal is to get/develop a video delivery platform based on the Peer-to-Peer model in order to optimize delivery by using the full capacity of the local provider tail loop,. The international Internet network peering being very weak and expansive, the future solution needs to be optimized for local loop network distribution. The future infrastructure should be able to provide both On-Demand and Live streaming content, acquired from different sources types (video files, Terrestrial, Satellite, Camera, etc.).

This document will describe features requirements for each different block composing the final Video Delivery Infrastructure (VDI) end to end.

Because we are targeting a medium/long-term development, not all features are requested for D-1, but depending on the workforce needed and costs, we will be able to provide an implementation roadmap.

Drafted by Nicolas Mathey (nicolas@mathey.fr)

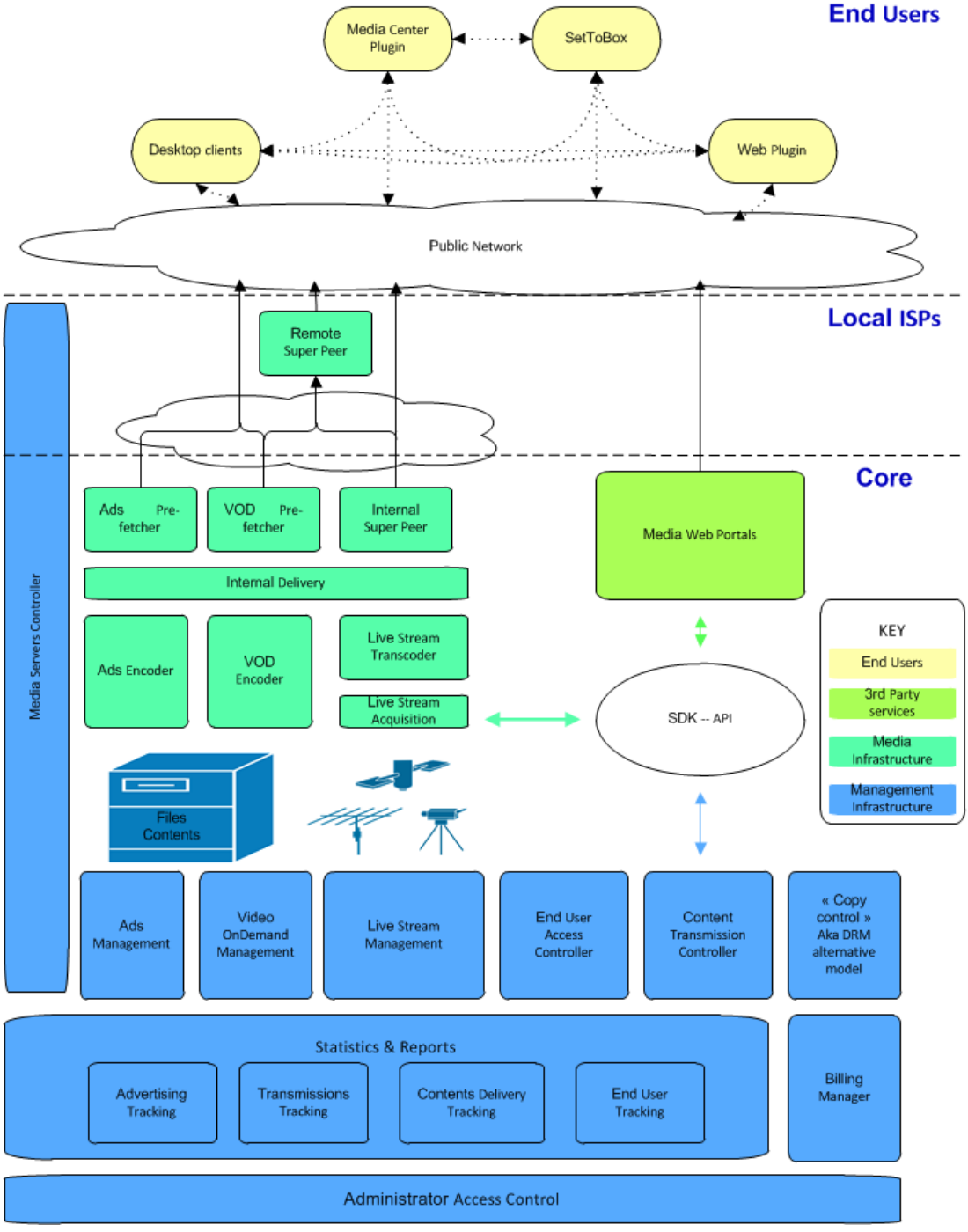
Glossary

P2P	Peer-To-Peer. It is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged and equipotent participants in the application.
P2PTV	Peer-To-Peer TV. It is a software application designed to redistribute video streams in real time on a P2P network. The distributed video streams are typically TV channels.
CDN	Content Delivery Network or Content Distribution Network. It is a system of computers containing copies of data, placed at various points in a network so as to maximize bandwidth for access to the data from clients throughout the network.
SDK	Software Development Kit. It is a set of development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platforms.
RTMP	Real Time Messaging Protocol. It is a protocol developed for streaming audio, video and data over the Internet, between a Flash player and a server.
STB	Set-Top Box. It is a device that connects to a television and an external source of signal, turning the signal into content which is then displayed on the television screen or other display device.
HTPC	Home Theater PC. It is a convergence device that combines the capabilities of a personal computer with a software application that supports video, photo, and music playback, and sometimes digital video recorder functionality.
VoD	Video on Demand. It is a system that allows users to watch video on Demand.
VDI	Video Delivery Infrastructure. This is the infrastructure components supporting the CDN.

P2PTV PROJECT

I. Overall VDI features requirements:

In order to easily run a medium term solution development, we decide to divide as much as possible the final solution into blocks.



Global Video Distribution Infrastructure overview

P2PTV PROJECT

1. Media Infrastructure

This major goal of this infrastructure is to manipulate contents video sources (acquiring, encoding, transcoding) and deliver them to the end-users infrastructure.
Sources may be files, terrestrial/satellite signal, camera, etc. Sources are out of the scope for this request for proposal.

Video contents, both VOD and Live, can have at least two audio channels (i.e. two different languages) and two subtitle stream (i.e. two different languages).

- **VOD Encoder:**
  This block is responsible of encoding the On-Demand contents to several encoding profiles in order to adapt the content to the final end-users infrastructure capabilities.

- **Ads Encoder:**
  This block is responsible of encoding the Ads contents to a unique and optimized profile.

- **Live Stream Acquisition:**
  This block receives the live content from different sources like an analog signal, existing streaming signal, stored video files, etc. It processes the signal and pushes it into the media infrastructure.
  This block is also responsible of simulating live channels by playing videos playlist (See "III. Near VOD channels").

- **Live Stream Transcoder:**
  This block is responsible of transforming the original signal acquired into an optimized content based on transcoding profiles.

- **VOD Pre-Fetcher:**
  This block is responsible of partially distributing specific VOD contents (i.e. including both audio and video) directly to compatible end-users infrastructure.
  Detailed features requirements are developed in "IV. Instant VOD feature".

- **Ads Pre-Fetcher:**
  This block is responsible of pushing advertising to each end-user infrastructure, in order to play instantly ads before playing requested content without consuming end-user bandwidth.
  This feature is similar to the "VOD Pre-fetcher" except than advertisement are published completely on the client side.

- **Internal Super Peer:**
  This block is responsible of receiving the content from sub-layer component described above, and processes it in order to push and/or stream it into the P2P network swarm. Thus it is acting as a broadcaster into the P2P swarm as part of the end-users content distribution framework. It first priority is to insert the content into the swarm, even if can also participating in the content distribution.

- **Internal Delivery:**
  This block is sharing and replicating the contents between nodes part of the end-users distribution layer. This component can be used for load-balancing such nodes.

- **Remote Super Peer:**
  Very similar to the Internal Super Peer except it is not acting as a broadcaster but only as peer having high upload bandwidth capabilities. It first priority is to actively distribute content in the swarm.
  In our model, this component will be place in local ISPs facilities in order to optimize bandwidth usage & costs, but would be remotely managed from the core infrastructure.

2. Management Infrastructure

This infrastructure layer is aimed to manage each components of the future VDI, and make sure each of them can interact properly.

- **Video On Demand Management:**
  This block manages VOD contents: source location, number of replicas for each videos, distribution on the end-users infrastructure, encoding profiles available for each content, number of viewers, assigned resources, etc.
  This block is interacting with following Media Infrastructure blocks: VOD Encoder, VOD Pre-fetcher, End-user delivery layers.

- **Ads Management:**
  This block manages advertising contents: source location, number of replicas for each video, distribution on the end-users infrastructure, number of view, assigned resources, etc.
  Each ads are tagged per ads customer (3rd Party paying to display their ads), category, price plan (should be played at least 10 times of day for instance), etc.

- **Live Stream Management:**
  This block manages Live contents being processed by the VDI: original signal source, content encoding & transcoding profiles, listing of viewers, assigned resources, etc.
  This block is interacting with following Media Infrastructure blocks: Live Stream Acquisition, Live Stream Transcoder, End-User delivery layers.

- **Media Servers Controller:**
  This block is responsible of controlling Media Infrastructures nodes components by adding, removing and controlling resources nodes assignments.
  It is also responsible of securing interactions between each node, using certificates.

- **End User Access Controller:**
  This block is managing and establishing permissions to end-users on specifics components.

- **Content Transmission Controller:**
  This block manages and controls content transmission process all over the VDI and the content distribution to end-users.

- **"DRM" Manager:**
  This block is responsible of managing Digital Rights when required, by making sure the end-users is viewing content as per their rights/permissions, but should not be platform dependent.
  –This need to be developed and name differently as I don't want to follow current DRM logic maintained supported by majors. "Rights /Permissions" definitions need to be develop in order to describe scope of that application block, and make sure media acquired/rent are readable on any platform own by the customer-

- **Statistics & Reports:**
  This block can be sub-divide into three main functions:

  Transmission Tracking: Monitor contents (Live and VOD) distribution over the network to the end-users. This may help for detecting transmission and streaming problem. This feature provides information on the quality of the contents distributions.

  Content Delivery Tracking: Provides information about contents popularity, availability, etc.

  Advertisement Tracking: Provides information about ads being viewed, availability, etc.

End User Tracking: Provides information about end-users profiles: contents requested, IPS, requests time, etc.

- **Billing Manager:**
  This block is computing results from different tracking module (Statistics & Reports block) in order to provide an order to invoice for end-users and advertising customers. Information can be exported into CSV format at least.

- **Administrator Access Control:**
  This block is controlling administrator access, privileges, and defining securities for the whole VDI.

3. Third Party Services Infrastructure

This infrastructure is basically including any 3rd party services, which may use the VDI using the SDK and APIs.
The best example would be a web media portal where end-users can subscribe to the service, browse the contents catalog, and watch a content, etc.
Another example might a 3rd party streaming server for Flash, Silverlight, or HTML5 legacy streaming technologies.

As part of this request for proposal, we include the following features in the scope:

- **Custom Media & Control Panel Portal:**
  This portal is teasing the VDI offers to end-users and a way to administrator to manage the whole infrastructure.
  End-Users may watch Live Stream and access to their personal account.

- **HTML5 streaming server:**
  Video legacy streaming server using HTML5 and distributing partially or totally the same contents catalog.

4. End Users Infrastructure

This infrastructure layer is on the end-users side and use to watch contents offered by the VDI.
Any of the following players should be able to choose any audio channel, and subtitle stream if any, and change this selection while playing the content.

The Graphical User Interface ("GUI") should be customized/branded to the future company owning this VDI. This GUI has to be similar to any type of end-user infrastructure detailed below:

- **HTPC Plug-in:**
  Embed version of the media player into XBMC Media Center software providing a complete media solution to the HTPC end-users.
  It will be available on Windows, MacOS X and Linux OS, and will be compatible with all the VDI features (instant VOD included)
  Note: We are open to discussion on the HTPC Software choice.

- **Desktop clients:**
  Directly install on the end-users OS as a standalone player application compatible with the all the VDI features (instant VOD included).
  It will be available on Windows, MacOS X and Linux OS, and should be able to run in full screen.

- **Set-To-Box ("STB"):**
  Embedded version of the player into a Media Center SetToBox compatibles with all the VDI features (instant VOD included).
  Note: the requested of development only includes the addition of the instant VOD features into your own current release, using your current STB model.
  In a mid-term plan (P4/P5) we will work on a custom STB development offering more capabilities to the ends-users.

- **Web Plug-in:**
  Browser embedded version of the player only compatible with live streaming solution only (i.e. not compatible with instant VOD feature).

II. Near VOD Channels:

As broadcasting legacy channels from non-IP sources (i.e. Terrestrial, Satellite, etc.) may request complex and costly infrastructure, we are targeting to simulate a live channel by playing a videos playlist channel type. However on a end-user point of view it will be shown as a normal live channel, playing looping content, similar as we could have on certain airline video system.
Each Near VOD channel playlist will be create and manage on the backend side.

III. Instant VOD feature:

Because local loop may be very poor quality, end-users may have to wait almost 10 minutes before even starting watching a movie. Plus they may have some glitch or have to re-buffer often.
To avoid such situation we would like to implement a way to instantly start playing the requested On-Demand content. However we may not be able, legally and even technically, to store completely all our movies library on the end-users side.
Thus partial distribution of each movie content (including both video/audio channels) on each end-users client may answer this problematic. Indeed, the end-user can start instantly watch his movie from the local storage, and download using P2P technologies the rest of the movie from end-users clients and our media infrastructure. Subtitles can be downloaded, if the end-users requested them, as soon as requested.

We think the solution as follow:

- Each end-user client will have partially movies on their local storage.
- The content of the library will be announced via RSS technology to each end-user client.
- New available movies announced via RSS will be simultaneously partially downloaded on all end-user clients.
- All end-users clients will have the first XX% (i.e. first 30 minutes for a 1h30m movie), plus random YY% of each movie.
- XX (default is 30%) and YY (default is 10%) are modifiable variables.
- Movies that are not announced any more in the RSS streams will be deleted from local storage.
- The end-user will be able to immediately start watching his movie, while in the same time last parts of this same movie is being downloaded from the end-users infrastructure swarm.
- The next 30% of the watched movie should be downloaded first when initiating video content play.
- When the end-users finished watching, the movie is being completely kept stored on the local storage for ZZ days and become a major seeder for such amount of time.
- After ZZ days the movie is being deleted partially to only keep first XX% plus random YY% of it, to become a minor seeder again.
- ZZ (default is 7) is a modifiable variable.

IV. Advertising concept:

Ads will be played to each end-users before he start watching the requested content (VOD or Live). More than just getting some revenues, and because ads will have been preloaded on the client side, it will able the remote player to start the streaming process in the background and take "some advance" to avoid any freeze/glitch.

V. Support requirements:

Depending on implementation and commercialization phase our support need may evolve.
On our point of view, we are seeing three phases:

- **Building phase:** we will build the VDI and will have beta users. We may need something similar to your current "Basic Support" offer.
- **Commercialization phase:** Public Go-Live, where we'll need to be the most reactive in case of any issue. Support model similar to your "Full support" offer may work, but might probably need to be customizing to fit our time zone difference.
- **Steady State phase:** Starting 1 or 2 year after public Go-Live when we will probably request some new future development. The support model requested may be a mixed of the two above.

VI. Timing versus Infrastructure Sizing

Size of the VDI will depend on the commercialization phase progression, and how many VOD contents and Live channels we will served to how many end-users.

1. Day 1 sizing:
- 500 users
- 8 Live channels
- 100 VOD contents in catalog

2. Month 6 sizing:
- 1000 users
- 8 Live channels
- 300 VOD contents in catalog

3. Year 1 sizing:
- 2000 users
- 10 Live channels
- 500 VOD content in catalog

4. Year 2 sizing:
- 5000 users
- 20 Live channels
- 1000 VOD content in catalog

VII. Pricing

Pricing can propose in work packages as per features priority requested, with estimate duration for development and implementation, depending on the timing described above.

The proposal should provide the features list being develop and implement with an estimate number of servers resource needed, with specifications recommendations.
Hardware infrastructure procurement is out of the scope of this request for proposal.

The proposal should also estimate the bandwidth required by the VDI for each commercialization sub-step (i.e. see timing definition above).

Components that don't have critical bandwidth, latency and jitter requirements can be hosted outside our facilities, but an in-house hosting (mitigating any International link failure proposal) should also be proposed for high-availability purpose.

VIII. Confidentiality

The Parties agree to keep confidential and not to disclose, communicate or divulge to anyone any confidential information received through or for the purposes of this project.

IX. Appendices – Price list format

In order to better understand effort needed and costs to plan our budget roadmap, we would like you to follow the format below as much as possible:

	Day 1			Month 6			Year 1			Year 2		
	Duration	Cost	Delivrables / Comments	Duration	Cost	Delivrables / Comments	Duration	Cost	Delivrables / Comments	Duration	Cost	Delivrables / Comments
1. Media Infrastructure												
1.1 VOD Encoder												
1.2 Ads Encoder												
1.3 Live Stream Acquisition												
1.4 Live Stream Transcoder												
1.5 VOD Pre-Fetcher												
1.6 Ads Pre-Fetcher												
1.7 Internal Super Peer												
1.8 Internal Delivery												
1.9 Remote Super Peer												
2. Management Infrastructure												
2.1 VOD Management												
2.2 Ads Management												
2.3 Live Stream Management												
2.4 Media Servers Controller												
2.5 End Users Access Controller												
2.6 Content Transmission Controller												
2.7 DRM Manager												
2.8 Statistics & Reports												
2.9 Billing Manager												
2.10 Administrator Access Control												
3. 3rd Party Services Infrastructure												
3.1 Customes Media & Control Panel orbit												
3.2 HTML 5 Streaming server												
4. End Users Infrastructure												
4.1 Media Center Plug-in												
4.2 Desktop clients												
4.3 STB												
4.4 Web Plug-in												
5. Support Models												
5.1 Building Phase												
5.2 Commercialization phase												
5.3 Steady state phase