

Itview Modules

List of internal modules used by itview. Those that are already implemented as re-usable modules and used outside the itview application are [highlighted in blue](#). All the modules are written in python 2.x unless specified otherwise.

Techstack

- Python, PyQt, Cython, C++, Scala

GUI

- PyQt v4 (PySide compatible)

Viewport

- OpenGL based "layer stack"

Playback Engine

- Timecube (timing module) that takes over the Qt event loop
- Videostreamer, manages read-ahead logic for video and invoking the read codecs
- Audiostreamer, same as videostreamer but for audio

I/O Layer

- [OIIO](#) for individual image files (C++)
- [FileSequence](#) (C++)
 - Understands individual image files for a rendered sequence
- [SpMovie](#) (C++)
 - Wrapper around third party libraries and internal movie formats (h264, J2K codecs)
 - Internal movie file format using J2K with DCP compliant encoding (12 bit depth)
 - Ffmpeg based libraries for most third party movie file formats
- MediaCache
 - Finds the physical media on disk based on tracking number
- VirtualMedia
 - Programmatically generate frames for blackframes, error card, contact sheet, title cards
- Media layer that abstracts all video, audio I/O
 - Support pixel probing
 - Recognize colorspace, aspect ratio, stereo

Playlist

- Manages a list of clips (media with video and/or audio)
- Clip can have one video and/or audio
- Manages metadata with playlists, clips and the media objects themselves
- Transitions: cross-dissolves and wipes
- Timewarps: customize the order of frames for playback, holding frames, speed changes etc
- Playlist files are in XML format

Database Layer

- [Shotgun](#) (Production tracking)
- [Edbot](#) (Editorial tracking: proprietary, maybe replaced with OTIO in the future)
- [VFO](#) (review notes)
- [VnP](#) (Versioning and Publishing - proprietary asset management system)

Annotations

- Supports
 - Scribbles with various Brushes, Text, Color Corrections (rectangles, ellipses and lasso)
- Vector format (resolution independent), stored as XML in database

- Internal API for storing and retrieving
- Would be great to move to an interchangeable format+API (OTIO, USD)

Color Pipeline

- [OCIO](#) (C++)
- Color correction operations are sent as OCIO requests and rendered
- Aware of display devices being used and will choose the correct look

Misc

- [FCurve](#)
- [FTGL/Freetype](#)
- [FormWidgets](#)
- [ResolutionTable](#)
- [StereoConvergence](#)

Plugins

- Python based
- Internal “pyCore” module exposes most of core itview functionality to plugins

Messaging

- PyQt signals/slots
- ZeroC ICE api
- Sockets

Remote Sync

- [Sync Server](#) (Scala)
 - Separate project, written in Scala for high performance
- Itview clients talk to it via ZeroC ICE api protocol
- Manages permissions and access controls for remote users
- Host itview Qt events and API calls are broadcast to remote clients
- Sync sessions provide a “headless itview” with it’s entire state backed by a database
- Public (show based) /private user based sessions as well as internal “context view playlists”
 - Aware of editorial information for sequence/shots