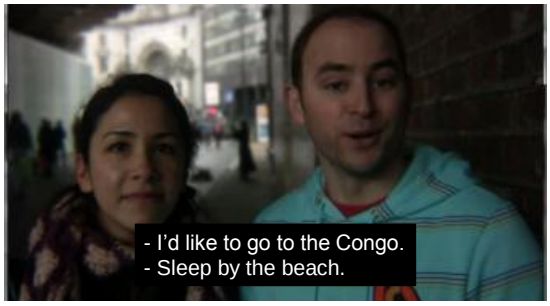
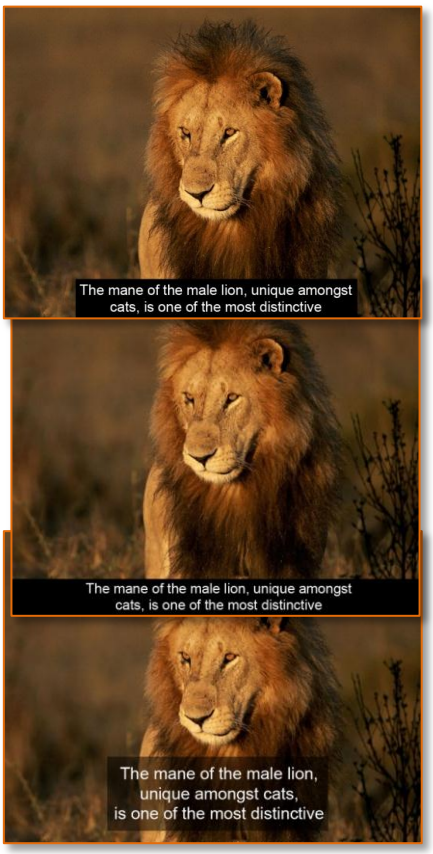
Screen has developed innovative solutions to the production and display of quality subtitles on the web for both recorded and live video.

Background

Most standard web-video players use a simple timed-text file for subtitling (e.g. SRT for YouTube, TTML/DFXP for Flash, WebVTT for iTunes and SMIL for Windows Media Player). Such files can be easily created in standard subtitle preparation tools such as WinCAPS and MediaMate then hosted on the server along with the video file.






```
<p xml:id="19" begin="00:05:20.320"
end="00:05:23.720" tts:textAlign="center" style="s1">
- I'd like to go to the Congo.<tt:br />
- Sleep by the beach.</p>
```

extract from a TTML timed-text file

The interpretation of this timed-text file, and hence the appearance of the subtitles, can differ between players and between devices. (Early implementations did not even overlay the subtitles within the video picture area, instead positioning text within a black box beneath the main video.) Positioning can be particularly inconsistent – a crucial factor where translation subtitles may be carefully positioned to obscure other on-screen text in the video. Other factors include anti-aliasing and variable font widths. The result is an inconsistent viewer experience where the quality may be well below that of broadcast subtitles.

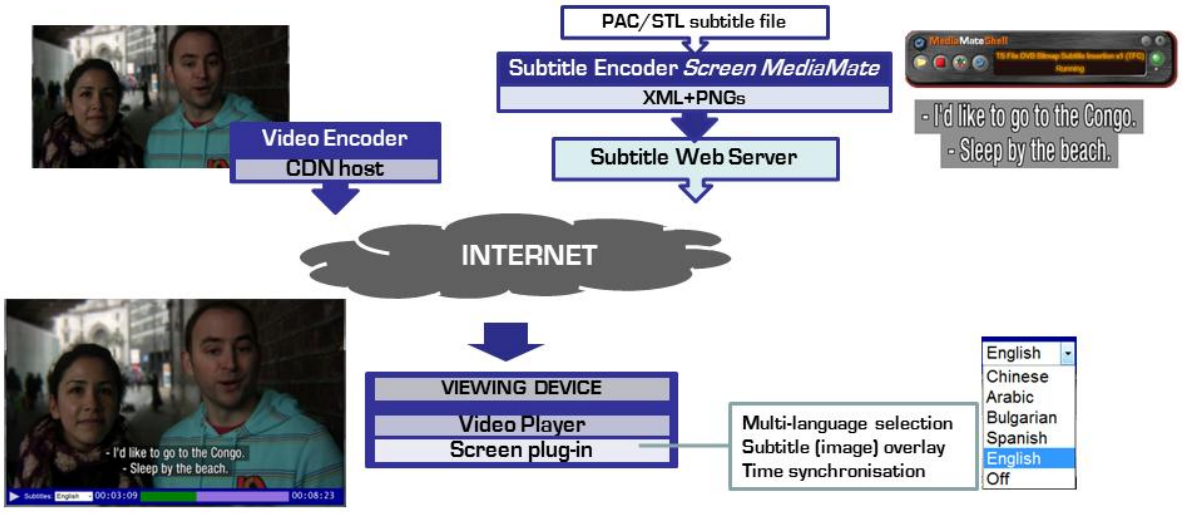
Most of these player implementations have been prioritised to satisfy demands for broadcast closed-captions to also be available with online versions, and in many ways this type of accessibility application does not pose a particular problem. Although there are still variations in presentation between the players there is plenty of room to avoid errors when working within the constraints of American closed captioning (including its limit of 32 characters per row) or its European teletext equivalent (up to 36 characters). However translation subtitles with more characters, varying text/background styles and quite precise positioning are more challenging.

Even with "standardised" timed-text formats, such as SMPTE TTML, a minimal implementation may be sufficient to satisfy a regulator concerned with accessibility (such as the FCC) but it's far from reproducing the intended TV viewing experience of open-captions used for translation subtitles.

The lack of consistency across players means that the subtitle-author cannot be sure as to how the subtitles will appear in font, style or position. For accurate and consistent presentation these uncertainties need to be overcome.

Streaming from a server (video-on-demand, catch-up TV)

Screen has developed a method to encode, store and display high-quality subtitles with served web video playback. The solution ensures that subtitles are always displayed in the style intended regardless of the player – while allowing the viewer to select from several language streams (and off) without the need to re-encode the video or host multiple copies.

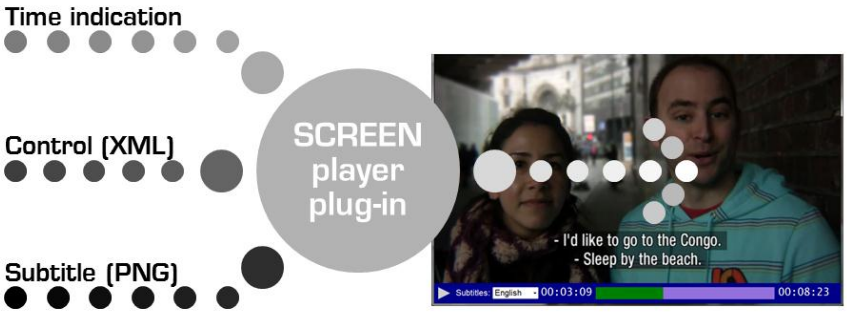


Subtitles for web video consistently displayed as the producer intended

Screen's new subtitle overlay layer solution for web video involves not only the engine to create the subtitles but also a means to display them in the player. It's an image based approach to ensure consistent presentation of the subtitles on all players while also improving the visual quality of the subtitles on low-bandwidth connections as the subtitles are not subject to the same compression as the underlying video picture.

- **Consistent display -** across any viewing device
- **Multi-language –** allowing the viewer to select from several language streams or turn captions off
- **Standard subtitle file format –** for ease of production, and allowing for automatic adjustment of broadcast version for the web via a simple EDL
- **Image-based –** avoids inconsistencies of browser implementations based on timed-text and gives producer full creative control over style and position
- **High-quality text -** regardless of video compression

Screen's MediaMate file-based subtitle encoder converts a standard subtitle file (per language) into timing data and images for the target resolution(s) applying an EDL if necessary to adjust for web-specific edits or commercial breaks. The resultant PNG and XML files are hosted on a web-server ready for use; for extra flexibility this web-server need not be the same as that hosting the video.



Open-source Java plug-ins for each media player allow clients to implement this form of subtitling into their website systems with ease. The plug-in displays the PNG subtitle sequence as an overlay to the main video in synchronisation with the video playback. The subtitles can be turned on/off at the viewer's control and will even allow the user to select from multiple languages where available.

MEDIAMATE	PLAYER PLUG-IN
• Converts industry standard STL/PAC subtitle file to PNG images & XML control code	• Source code freely available for partner customers to integrate into their own player solutions
• Control font, size, background style, safe area etc.	• Overlays correct subtitle for player's current time reference
• Output hosted on web-server - may be independent to video	• Resizes subtitle to match size of video player
• Automated process	• Multi-language selection

The following links show prototype implementations of this new subtitle overlay layer solution for various Internet media players. The players used are all simple versions, created on top of the generic publically available sample players with the subtitling functionality foremost in mind rather than the aesthetics of the control bar. We recommend viewing in Chrome or Firefox rather than Internet Explorer and of course some of these players will not work on mobile devices.

HTML5 (HD – High Profile H.264)
http://screen.sysmedia.com/demo496515/LondonPlayer.html

HTML5 (SD - 640x368 Baseline profile, MP4)
http://screen.sysmedia.com/demo496515/LondonPlayerSD.html

Silverlight
http://screen.sysmedia.com/test618773/LondonSilverlight.html

Flash
http://screen.sysmedia.com/demo496515/LondonFlash/player.html

Live-streams (webcast, broadcast/internet simulcast)

Live subtitling on broadcast TV programmes is most usually provided in only the source language, to improve accessibility for deaf and hard-of-hearing viewers. Transferring those subtitles to live web streams however is not easy. There is no timed-text delivery path for live streams and unlike served video the player does not provide an elapsed time indication so the variable latencies in web delivery make accurate synchronisation very difficult – and that's before allowing the viewer the chance to pause and resume the playback, or even rewind it. However Screen's image based approach described in the previous section can also be applied to live web-streaming with minor modifications to allow for a real-time workflow (rather than file-based) and to ensure subtitles synchronise to video regardless of buffering etc.

Screen therefore offers three solutions for subtitling of live webstreams ensuring reliable subtitles that are always in sync with the video regardless of the video player or platform. These work for both recorded programmes (with prepared subtitles played out as simulcast to TV and web), and for live programmes (with real-time captions).

- **Polistream image-based web-subtitling output with a timestamp in the active video picture** prior to web-encoding and Screen's plug-in for the browser's media player to handle synchronisation of the subtitles with the video using that timestamp (viewer can turn subtitles on/off and potentially select from multiple languages).
- **Polistream image-based open-subtitling output with interface to the web-encoder** which then burns the subtitle image into the picture allowing the use of a standard media player (without modification) at the viewer's end (however viewer has no control to turn the subtitles off, although it is relatively simple to encode separate streams both with and without subtitles if required).
- **Polistream SDI open-subtitling output** with subtitles burnt into the SDI video prior to web-encoding, also allowing the use of a standard media player (without modification) at the viewer's end (again viewer has no control to turn the subtitles off).

The suitability of the first two approaches may be dependent on encoder manufacturer, particularly for encoding direct from ASI rather than SDI. The final approach should work with any encoding method and adds no extra cost if the broadcast output from Polistream already uses open-subtitles (assuming the same output can be used for both broadcast and web delivery).

Existing Screen customers already using Polistream for broadcast output can easily upgrade the system to add an interface to their streaming video encoder so that real-time subtitling on any live broadcast is also conveyed on the simulcast web output.