

Audio- und Video im Internet ab HTML 5 (Freeware) (Stand 24. Mai 2010)
#####

Nachfolgend werden 2 Varianten der Mediendarstellung im Internet behandelt, die gerade Einzug in die Welt der WWW gehalten haben und auch noch frei sind.

Theora-Ogg und VP8-Ogg

1. Theora und ogg von www.xiph.org (beide Codecs) (Stand Anfang Mai 2010)
=====

1.1. Ansatz

Etliche Jahre später als es Microsoft getan hat, wurde nun Sound und Video als HTML-Tag eingeführt und dabei die Schnittstelle von Microsoft unnötig gemacht.

Im Zuge der Verbreitung von HTML5, das Mediadaten anzeigen kann, wurde Codec (Vorschriften zur Art der Medien) benötigt, der ebenfalls verbreitbar ist, wobei entweder keine oder zeitweise oder immer Lizenzgebühren fällig werden.

Es gibt Browserhersteller, die haben sich der Implementation von HTML 5 angeschlossen oder auch nicht. Zu letzteren gehört Microsoft als Vertreter von Unternehmen, die ihre Produkte direkt oder indirekt gegen Entgelt verwerten und Kunden an Produkte binden wollen.

Die Verbreitung von Media per HTML 5 ist nur in Form der Codecs von www.xiph.org möglich (oga, ogv, theora).

Microsoft und andere Lizenzgeber planen, HTML 5 um den H264-Codec zu erweitern, der auf Verwertung per Lizenzrecht basiert und zum Zweck der Verbreitung wie folgt offeriert wird:

zeitweise von Entgelten befreit,

Einbindung des Codes in Windows, das Payware ist (indirekte Lizenzverwertung).

Zu diesem Zweck plant Microsoft sogar, auf den eigenen Codec V1 zu verzichten, damit der massenhaften Verbreitung von H264 (ähnlich wie von DivX) wenig im Wege steht - außer eben Ogg und Theora.

Microsoft ist z.Z. DER bekannteste Browserhersteller, der HTML-5-Media nicht unterstützt und das schon seit einigen Jahren in Microsoft-HTML-DOM eingebaute BGSOUND-Tag nicht erweitert. Im Gegenteil: Microsoft hat HTML-5 bereits in Elemente so eingebaut, dass das Objekt als leere Hülle vorhanden sind, aber das Ansprechen der Hülle zu Fehlern der Webseite führt, fängt man die leere Hülle nicht ab. Mit anderen Worten: Microsoft macht sich also auch noch lustig über HTML-5. Z.Z. ist der Internet Explorer 8 am Markt, der HTML-5-Media fehlerhaft implementiert hat. Der Internet Explorer 9 wird HTML-5

per H264 anbieten, wobei die Lizenzgebühr per Kauf von Windows abgedeckt wird.

Eine weitere Strecke zur Mediendarstellung per Webseite ist inzwischen das massenhaft verbreitete Flash von Macromedia, das von Adobe geschluckt wurde und mit hochpreisigen Produkten angepriesen wird - nur der Player nebst Codec sind kostenfrei beziehbar. Die Flash-Verbreitung ist derart massiv, dass wie Google-Werbung Webseiten regelrecht und aggressiv zugemüllt sind: Es soll Verwertung stattfinden. Webseiten mit abgeschalteter Flash-Fähigkeit sind eine Ansammlung von leeren Containern, so dass die maschinelle Erstellung der Webseite (nebst Datenbankbindung) geringe Kosten der Entwicklung erlauben, um Verwertung am Markt sichtbar wird. Man sieht diesen Webseiten den sterilen Flash-Stil oft an - optimierte Uniformität für Verwertung.

HTML-5 mit ogg und theora erlaubt es, kostengünstig, müllfrei und individuelle Webseiten zu gestalten, ohne sich an Hersteller, Betriebssystem und Lizenzregelungen binden zu müssen. Diese innovativen Möglichkeiten sind schlichtweg divergent zu o.g. Auffassungen bezüglich H264 in HTML5. Ob ein Verwertungsanbieter sich erbarmt, und HTML-5 unverändert zulässt, ist stark zu bezweifeln - erst recht bei Microsoft und Adobe. Das zeigt auch die Entwicklung bezüglich DivX (analog die von Blu-Ray als DVD-Nachfolger: Nur die AVCHD-Variante lässt sich kostengünstig oder mit Freeware nachbilden, nicht jedoch das eigentliche Blu-Ray).

1.2. HTML-5-Media auf Basis ogg und theora (www.xiph.org)

Z.Z. sind nicht alle Webbrowser aktuell bezüglich HTML-5. Der Microsoft Internet Explorer 8 ist komplett veraltet: Er kennt kein HTML-5-Media, egal was Microsoft über HTML-5 lobend von sich gibt. Microsoft hat mit HTML-5-Standard wenig am Hut und diesmal extrem kontraproduktiv, dafür bestens sich selbst ausgrenzend.

Browser wie Opera und Firefox kennen HTML-5, wobei Firefox traditionell etwas verspielter ist.

Wenn also HTML-5-Media im Browser implementiert ist, dann z.Z. auf Basis ogg und theora (www.xiph.org). Die HTML-Elemente <VIDEO> und <AUDIO> werden vom Browser direkt verarbeitet, ohne dass ein Codec installiert sein muss. Der Browser bietet einen Player an.

Der Codec-Anbieter liefert zugleich Software mit, die z.B. in Windows den Microsoft Windows Media Player zur ogg/theora Fähigkeit verhilft, nicht jedoch aber den Internet Explorer (nur Google arbeitet gern für Microsoft und erweitert den Internet Explorer um Mediawiedergabefähigkeit). Der Codec erweitert auch Freewareplayer wie Media Player Classic (<http://mpc-hc.sourceforge.net/>). Kommerzielle Software wird eventuell nicht um die Abspielfähigkeit erweitert, wenn der Anbieter der kommerziellen Software das unterbindet und nur seine Codecs offerieren will. Unter Windows wird DirectShow benutzt.

Im Bereich Freeware gibt es wenige, aber brauchbare Tools zu Konvertierung von Medien nach ogg bzw. theora, so dass man die Medienerstellung auch mit anderer Freeware vollziehen kann.

z. B. OGV erstellen per FFCode (über ffmpeg2theora) z. B. aus AVI oder mpg.
Damit gibt es eine Adobe-Flash-Konkurrenten mehr und das als HTML-Standard.

Audio in HTML-5

wave also vermutlich nur unkomprimierte PCM
definitiv kein flac (getestet) obwohl flac zu Ogg gehört
kein a3 (getestet)
kein mid (getestet)

Video

nur Ogg Theora also OGV oder OGG oder OGA

<http://theora.org/downloads/>

AVI

keine komprimierten Formate wie mpg, wmv, mp4

Ogg Theora (ogv) können

Opera
Firefox
Chrome
Safari kann nur H.264 codec (mp4)
Internet Explorer kann nur Plugins wie Flash etc.

Internet Audiostreams per OGG

Streaming ist der permanente Datenfluss über das Internet, also können permanent Kosten des Datenflusses entstehen. Warnung: Es gab in der Vergangenheit Provider, die haben Flatrate angeboten, aber bei tatsächlicher Nutzung mit hohem Transferumfang den Kunden gefeuert. Flatrate ist nicht Flatrate, zumindest in Deutschland nicht.

Wer LiveStream für OGG sucht, um es problemlos im Browser per HTML5 anzuhören, der muss ausgiebig suchen.

Die Radiosender des WDR (dradio und dfunk) bieten ogg-Streaming an: Kostenlos das Programm live im Browser mithören. Warnung: GEZ-Gebühr für Internetzugang fällig (Es reicht die reale Möglichkeit des Zuganges, jedoch ob auch wirklich benutzt, ist egal).

Üblich ist es, den Zugang per m3u-Datei anzubieten und je nach Qualität des Datenflusses. m3u-Datei ist wie eine Textdatei

lesbar und enthält den Internet-Link, der dem Player per <AUDIO>-Tag im Attribut SRC übergeben wird. Das Streamen selbst erledigt der Browser komplett allein, welcher dazu mit dem Internet kontaktiert sein muss.

Wer vermutet, dass ARD und Co. Livestreaming für OGG anbieten, liegt komplett daneben: Null Angebot für OGG-TV. ARD und Co. benötigen z.Z. ihre volle Konzentration und Kraft für ein HD-Fernsehen, das bewusst nicht im internationalen Standard gesendet wird (produziert schon eher). So gesehen ist es nicht verwunderlich, wenn ARD und Co. auch bezüglich Nutzerfreundlichkeit auf das falsche Pferd bei Livestream gesetzt haben: WMA und Flash und Co. "Begeisterte" Zuschauer von HD-TV des ARD und Co. haben schon mal eine Internet-Petition ausgelöst, die ARD und Co. zur Einhaltung des internationalen Standards und Qualität auffordert.

1.3. HTML-5-Audio

HTML-5-Audio und HTML-5-Video wurden spartanisch implementiert: Z.B. gibt es keine JavaScript-Schnittstelle .stop(), dafür gibt es aber .play() und .src (belegt man .src mit dem Dateinamen, dann werden Daten erst per .play() wiedergegeben; belegt man .src mit Leerkette also "", dann wird ein .stop() emuliert - Man sieht: Konsequent nicht objektorientiert implementiert).

Audio ähnelt dem Video.

Video umfasst mehr Möglichkeiten und Eigenschaften von Audio.

Video kann auch Audio abspielen (ohne Bild)

Achtung: Design-Fehler: Ein leerer Anzeigebereich wird eventuell angezeigt, auch wenn Audio kein Bild haben kann.

bietet per rechter Maus, also per Kontextmenü, Möglichkeiten der Steuerung an.

Achtung: Es gibt ev. Browser, die zeigen Kontextmenü in der Reaktion abweichend an.

HTML-Syntax- Beispiel für Audio

```
<HTML>
```

```
<BODY>
```

```
<!-- Quelltext wurde mit Windows notepad.exe erstellt -->
```

```
<H3> OGA-Audio im Browser laden und sofort starten</H3>
```

```
<IMAGE ID="ersatz_fuer_poster"
```

```
  WIDTH="300"
```

```
  HEIGHT="240"
```

```

        SRC="demo_audi o. j pg"
    >
<BR>
<!--      Player-Control -Lei ste-höhe ist 50 Pixel  -->
<!--      Player-Control -Lei ste-Brei te ist 300 Pixel  für Audi o-Tag  -->
<!--      ist in Brei te l aut width für Vi deo-Tag  -->
<AUDIO ID="test"
    CONTROLS
    >
mp3-Audi o <A HREF="demo.mp3">demo.mp3</A>
<BR>
Der Microsoft Internet Explorer erkennt Audi o-Tag nicht, so dass der Link auch nicht
<BR>
erkannt wird, aber dafür der Text angezeigt wird.
<BR>
Sol l te der Codec zu ogg unter Windows installiert worde sein,
<BR>
kann der Windows Medi a Player auch ogg / theora wi edergeben.
</AUDIO>
    <!-- width und height werden ignoriert -->
    <!-- Control -Lei ste-Brei te ist 300 Pixel  für Audi o-Tag, das ni chts anzei gt und ohne Di mensi on ist  -->
    <!--      ist in Brei te l aut width für Vi deo-Tag (ni cht des Vi deos sel bst)  -->
    <!-- src muss angegeben werden, Vi deo oder Audi o  -->
    <!--      auch per source-Tag kodi erbar  -->
    <!--      z. B.  < audio ..... > < source src="demo.ogv" type="vi deo/ogv" > ... < / audi o >  -->
    <!--      Es wi rd nur das Audi o aus demo.ogv wi edergeben  -->
    <!-- poster wi rd i rgnriert  -->
    <!-- autobuffer optional, Laden der Medi adaten aktivi eren gleichzei tig mi t Laden der Websei te  -->
    <!-- autoplay optional, abspi elen der Medi adaten di rekt nach dem Laden des Audi o-Obj ektes i n der Websei te  -->
    <!-- controls optional, Playerl iste unten anzei gen wobei paral lel per Kontextmenü auch Player-Buttons verü gbar si nd
-->
    <!--      Playerlei ste kann auch per Kontextmenü ein- und ausgebl endet werden  -->
    <!--      Control -Lei ste ist di e ei nzi gste mögl ich Anzei geform  -->
    <!-- Loop Wi ederholung der Wi edergabe endl os  -->
<SCRIPT>
// +++++ Zei ger auf das AUDIO-Obj ekt hol en
var HTML5Obj ekt=document. getEl ementByI d("test");

// +++++ Audi o-Datei als Source
HTML5Obj ekt. src="demo.oga"; // src-Zuwei sung startet kei n play

// +++++ zugewi esene Audi o-Datei abspi elen
HTML5Obj ekt. pl ay();

// HTML5Obj ekt. stop(); // gi bt es ni cht

```

```
// HTML5objekt.src=""; // das bewirkt stop
</SCRIPT>
</BODY>
</HTML>
```

1.4. HTML-5-Video

HTML-5-Audio und HTML-5-Video wurden spartanisch implementiert: Z.B. gibt es keine JavaScript-Schnittstelle `.stop()`, dafür gibt es aber `.play()` und `.src` (belegt man `.src` mit dem Dateinamen, dann werden Daten erst per `.play()` wiedergegeben; belegt man `.src` mit Leerkette also "", dann wird ein `.stop()` emuliert - Man sieht: Konsequent nicht objektorientiert implementiert).

Audio ähnelt dem Video.

Video umfasst mehr Möglichkeiten und Eigenschaften von Audio.

Video kann auch Audio abspielen (ohne Bild)

Achtung: Design-Fehler: Ein leerer Anzeigebereich wird eventuell angezeigt, auch wenn Audio kein Bild haben kann.

bietet per rechter Maus, also per Kontextmenü, Möglichkeiten der Steuerung an.

Achtung: Es gibt ev. Browser, die zeigen Kontextmenü in der Reaktion abweichend an.

HTML-Syntax- Beispiel für Video

```
<HTML>
<BODY>
<!-- Quelltext wurde mit Windows notepad.exe erstellt -->

<H3> OGV-Video im Browser laden und sofort starten</H3>
<VIDEO WIDTH="320"
  HEIGHT="290"
  SRC="demo.ogv"
  POSTER="demo_video.jpg"
  AUTOBUFFER
  AUTOPLAY
  CONTROLS
  LOOP
>
<!-- Player-Control -Leiste-höhe ist 50 Pixel -->
<!-- Player-Control -Leiste-Breite ist 300 Pixel für Audio-Tag -->
<!-- ist in Breite laut width für Video-Tag -->
```

Der Microsoft Internet Explorer erkennt Video-Tag nicht, so ein Link auch nicht

erkannt wird, aber dafür der Text angezeigt wird.

Sollte der Codec zu ogg unter Windows installiert worden sein,

kann der Windows Media Player auch ogg / theora wiedergeben.

</VIDEO>

```
<!-- width optional, Breite des Containers, kann entfallen da dann Videobreite benutzt wird -->
<!-- wenn Dimension des Containers > Dimension des Videos, dann wird Video NICHT vergrößert -->
<!-- Video wird immer unter Einhaltung der Proportionen gerendert, wobei auch gezoomt wird -->
<!-- Ändert sich eine Dimension (z.B. Breite oder Höhe) dann wird die andere derart angepasst, -->
<!-- dass die Proportion behalten wird. Damit muss gezoomt werden. -->
<!-- Control-Leiste-Breite ist 300 Pixel für Audio-Tag, das nichts anzeigt und ohne Dimension ist -->
<!-- ist in Breite laut width für Video-Tag (nicht des Videos selbst) -->

<!-- height optional, Höhe des Containers, kann entfallen da dann Videohöhe benutzt wird -->
<!-- wenn Dimension des Containers > Dimension des Videos, dann wird Video NICHT vergrößert -->
<!-- Video wird immer unter Einhaltung der Proportionen gerendert, wobei auch gezoomt wird -->
<!-- Ändert sich eine Dimension (z.B. Breite oder Höhe) dann wird die andere derart angepasst, -->
<!-- dass die Proportion behalten wird. Damit muss gezoomt werden. -->
<!-- Achtung: Player-Control-Leiste erscheinen auf dem Video, wenn die -->
<!-- Höhe des Players nicht um die Control-Höhe erweitert wird -->
<!-- Control-Leiste-Höhe ist 50 Pixel -->
<!-- Control-Leiste-Breite ist 300 Pixel für Audio-Tag, das nichts anzeigt und ohne Dimension ist -->
<!-- ist in Breite laut width für Video-Tag (nicht des Videos selbst) -->

<!-- src muss angegeben werden, Video oder Audio -->
<!-- auch per source-Tag kodierbar -->
<!-- z.B. < video width ..... > < source src="demo.ogv" type="video/ogv" > ... < / video > -->

<!-- poster optional, Bild das angezeigt wird wenn Mediadaten nicht vorhanden oder kein autoplay aktiv -->

<!-- autobuffer optional, Laden der Mediadaten aktivieren gleichzeitig mit Laden der Webseite -->

<!-- autoplay optional, abspielen der Mediadaten direkt nach dem Laden des Video-Objektes in der Webseite -->
<!-- beachte poster -->

<!-- controls optional, Playerliste unten anzeigen wobei parallel per Kontextmenü auch Player-Buttons verfügbar sind
-->
<!-- Playerleiste kann auch per Kontextmenü ein- und ausgeblendet werden -->
<!-- Controls erscheinen auf dem Video, wenn width und height in der Video-dimension bzw. -->
<!-- width und height nicht angegeben wurden -->

<!-- Loop Wiederholung der Wiedergabe endlos -->
```

</BODY>
</HTML>

2. VP8 und ogg von www.google.de und www.xiph.org (getrennte Anbieter)

=====

2.1. Ansatz

Kurz nach Einführung von Theora und Ogg in HTML 5 wird HTML-5 um VP8 erweitert.

VP8 als Videocodec stammt von Google (Projekt WebM)

Ogg als Audiocodec stammt von xiph.

Download per

<http://code.google.com/p/webm/downloads/list>
<http://www.webmproject.org/code/specs/container/>
<http://www.webmproject.org/tools/>

<http://www.xiph.org/>

<http://matroska.org/>

Die Einbindung von VP8-video ist in HTML 5 IDENTISCH mit der von Theora-Video und Ogg-Audio.

Es gibt aber einen wesentlichen Unterschied:

Mit Theora-Codec kann gerade für kleindimensioniertes Internet-Video eine erheblich kleinere Datenmenge als mit VP8 erzeugt werden, wenn man Abstriche an der Videoqualität macht und eher dem Audio Vorzug gibt.

VP8-Dateien (webm) sind erheblich größer als ogv.

Sollte sich VP8 durchsetzen, ist das ein Geschäft für Internetprovider und deren Internetdatenleitungen: Das Netz (z.B. Cloude-Netz) wird vermüllt mit massiver Werbung z.B. die von Google (Werbung als Netzwährung mit Wirkung wie Geld drucken) oder zur Datenbeschaffung und -verwertung per Sozialer Netze als Massenmedium z.T. Minderjähriger und deren rechtlicher Fremdbestimmung bei Datenübergang in anderes Hoheits- und Rechtsgebiet.

Dass der OGG-Audio-Codec gut sein muss, sieht man daran, dass Google OGG-Audio für WebM verwendet, aber anstelle des bereits verfügbaren Theora nun das Google-eigene VP8. Z.B. kann damit Google das einverleibte Youtube als Internet-Offerte bei höherer Anzeigenqualität kostengünstiger verwerten.

2.1. Webm-video

Das Video besteht aus folgenden Komponenten

Video-Codec VP8 von Google (freier Codec)
Audio-Codec von xiph (freier Codes)
Container von Matroska (freier Container)
Besonderheit: Der Container darf NUR mit obigen Komponenten befüllt werden.
Video darf ohne Audio sein. Video darf nur mit Audio sein.

Die Qualität von VP8 soll auf unterstem Niveau des H.264, also auf Basis-Variante des H.264 liegen und ebenfalls schlechter sein als x264 (freier Pendant von h.264).
VP8 hat die leicht bessere Qualität von H.264 Baseline also der H.264-Basis-Variante und 72% der Qualität von H.264 Highline.

Alle 264-Varianten sind kein Standard im Internet per HTML.

H.264 ist in der Basisvariante z.T. von Lizenzgebühren freigestellt, ansonsten kostet alles z.B. im Blu-Ray-Bereich, der auf H.264 basiert (neben Digital-Audio).

H.264 ist Video-Teil von MPEG-4, wobei Audio AAC sein muss und der Container MP4 ist.
wird ab 2016 auch für jede nicht-kommerzielle Nutzung kostenpflichtig (Lizenzgebühr)

Es gibt Tools, die bringen obige Komponenten von Webm mit - allerdings NUR in der Funktionalität der Tools. z.B. FFmpeg oder Browser Opera

Will man betriebssystemweit VP8 nutzen, muss bei Windows DirectX-Show von Google installiert werden (siehe oben).

HTML-seitig wird identisch zu Theora vorgegangen, aber es gibt z.Z. eine Besonderheit, wenn man <SOURCE> benutzt.

Der HTML-MIME-Typ für Webm-Videos lautet bei Opera	video/webm mit Audio
	audio/webm nur Audio
ansonsten	video/x-webm
	audio/x-webm

Um dieses Elend zu umgehen, sollte man SRC nur im VIDEO-Tag kodieren.

```
<html >  
<head>
```

```

<body>
<video controls="controls"> <!-- Variante 1 -->
  <source src="sunflower.webm" type='video/webm; codecs="vorbis, vp8"' > <!-- auch codecs="vp8, vorbis"
-->
  <p><a href="sunflower.webm">Download the video</a>.</p>
</video>
<br>
<video src="sunflower.webm" controls> <!-- Variante 2 -->
  <p><a href="sunflower.webm">Download the video</a>.</p>
</video>
</body>
</html>

```

Hinweis für Theora

```

<source src="theora.ogv" type='video/ogg; codecs="vorbis, theora"' > auch codecs="theora, vorbis"

```

HTML-Programmierung mit Javascript ist wie bei HTML-5 für Theora und Ogg-Vorbis (ogv und oga).

2.3. Google-Projekt WebM

Das Projekt stellt Schnittstellen bereit, die VP8 verbreiten lassen und Konditionen von Google.

Google hat VP8 vom einverleibten Hersteller On2 übernommen, wobei On2 sichtbar vernachlässigt wird (Ausschlachten geht dem Ende zu Richtung Unbedeutsamkeit von On2).

Auf Basis des von Google bereitgestellten DirectX-Show-Controls kann der VP8-Codec unter Windows bereit gestellt werden, so dass z.B. der Media Player Classic auch webm-Dateien abspielt.

Es gibt einen Haken: Google kümmert sich NUR um VP8. Der Nutzer muss also den Audiocodec Ogg bei xiph.org besorgen, wobei xiph.org den Theora-Video-Codec gleich mit anbietet. Google holt sich also die Konkurrenz Theora ins Haus. Klingt scheinbar fair.

Das Software-Development-Kit (SDK) von Google richtet sich NUR an C-Programmierer. Selbst die Anpassung von FFmpeg muss man selbst reinkompilieren.

Google stellt keine angepasste Version von FFmpeg bereit - die muss man selbst erstellen, oder ergooglen.

Für reine Nutzer, die nur Video bzw. Audio erstellen wollen, um es in HTML einzubinden, ist das Google-Projekt komplett sinnlos.

Der User muss extrem hochpreisige Videoerstellung kaufen, die so den

freien Videocodec VP8 verwertet,
oder der User muss sich auf die Suche Freeware machen - und die fehlt massiv oder
taugt wenig.

Die Situation im Bereich Theora ist für User erheblich einfacher - auch
weil nicht von Google bestimmt.

2.4. Browserverfügbarkeit von VP8

Bekannte Browserhersteller bieten lauffähige Browser an, die VP8 so anzeigen als wäre es
das bereits bekannte Theora - eben auf HTML-5-Basis.

Nutzer des Internet Explorers werden durch den Hersteller Microsoft gezielt ausgegrenzt:

ab Internet Explorer 9 soll zukünftig VP8 unterstützt werden, aber Theora weiterhin nicht.

Microsoft hat sich entschlossen, Video zu unterstützen, das mehr Datenverkehr benötigen
und gibt z.B. Webseitenbetreibern, die Datentransfer-Kosten per Theora sparen wollen, keine einzige
Chance auf Theora, obwohl die Konkurrenz gnadenlos sich an HTML-Standard wie Theora halten.
Microsoft hat andere Interessen: Zusammen anderen Geschäftspartner wird die Verwertung von Windows
angestrebt. Da Google und z.B. Pornoseitenanbieter auf VP8 setzen, zieht Microsoft genau da
nach. Was Internet-Nutzer wollen, interessiert Microsoft wenig, solange Nutzer zahlen.
Der Internet Explorer hat damit seine Nischenberechtigung ohne Frage erreicht, aber als
Massenware ist er bedingt bzw. gewollt untauglich.

2.4. Videoerstellung mit Free-und Payware

Die Verfügbarkeit von Softwares zu Erstellung per VP8 ist z.T. katastrophal. Google ist nicht in der
Lage, sein VP8 zu verbreiten, wenn es außerhalb von Youtube benutzt wird (Der Sinn von VP8
liegt auf der Hand).

Player-Softwares (Decoder) bringen entweder Codecs mit, oder lassen sich unter Windows per DirectX-Show
VP8-fähig machen (siehe oben). Google bietet einen extrem primitiven Player im SDK an.

Erzeugung (Encoder) per Software ist Mangelware und zeigt den Sinn von VP8:

Extrem hochpreisige Softwares sind am Markt, die den freien Codec VP8 verwerten.
Es gibt Freewares als Gnadensbrot von Hochpreisanbietern oder als Produkt von
Triittbrettfahrern.

Z.z. gibt es nur 1 freie sinnvolle Software, die VP8 erstellen per Konvertierung (allerdings nicht aus Theora zu Webm) erzeugen kann: FFMpeg.

On2 Flix Payware, kein VP8-Tool

Sorenson Squish Payware

Wildform Flix Payware bzw. Kunden als Betatester eines Encoders, der mehr als schlecht FFMpeg benutzt (Trittbrettfahrer)

Der Download erfolgt nach Zwangsregistrierung per E-Mail, wenn man den Zugang nicht kennt:

<http://www.wildform.com/flixwebm/downloadsite>

Passwort ist Wildform4Flix

Sinn des Programmes ist ansonsten unklar, denn erzeugt Video ohne Audio stürzt bei Play der Quelle z.B. mpg ab kann selbst erzeugte webm nicht abspielen enthält Debug-informationen im C-Code

Der Anbieter teilt aber mit

Flix WebM creates the new WebM video format that uses VP8, a high performance video codec optimized for the web. (The VP8 codec was originally developed by On2 Technologies and was recently acquired by Google.) WebM offers many benefits for developers, publishers and end users including the fact that: it is royalty free, it's designed to provide high-quality video across resolutions, bitrates and multiple systems, it's efficient (which reduces power consumption), and it's optimized for Internet and network video delivery.

Zencoder Payware

FFMpeg free, siehe unten

Miro Videoconverter free

Die Software wird google-weit als erste freie Software gepriesen, die VP8 kann. Die Verbreitung dieser Anpreisung basiert auf gegenseitigem Abschreiben - eine massive Gepflogenheit im Internet (extreme Vermüllung zum Zweck der Schal tung von z.B. Google-Werbung).

Real gilt aber folgendes:

<http://www.mirovideoconverter.com/>

geht mit jedem Start ungefragt online auf <ftp://ftp.osuosl.org/> extrem langsame Encodierung obwohl es FFMpeg nutzt überhaupt keine Parameter einstellbar

Alle angebotenen Ausgabe-Formate sind vordefinierte und nicht erklärte
Parametern für ffmpeg.
bezüglich Theora wird das Original von Theora benutzt.
Theora wird 50% kleiner gegenüber der Quelle.
VP8 wird 50% kleiner gegenüber der Quelle bei 64 kbps Audio, also
z.T. unbrauchbar.

2.5. FFmpeg mit VP8-Unterstützung

Google erwartet von seinen Nutzern, dass diese programmieren können, oder gut gefüllten
Geldbeute haben, oder Tonnen von Google-Werbung während googlen nach VP8-Tools ertragen.
Google selbst ist so verarmt und unwissend, das keine Webseite betrieben wird, die dem Nutzer
zugig, verständlich und wärmstens VP8 ans Herzen legt: Man soll Youtube schauen, da wo der Rubel rollt.

Auf der Webseite <http://micksam7.com/blog/index.php/?p=743>

gibt es den Download der aktuellen FFmpeg-Version mit VP8-Fähigkeit: Als
Kommandozeilenversion. Ob es eine angepasste GUI für FFmpeg mit VP8
gibt? (googlen :-))

FFmpeg wird beispielweise wie folgt aktiviert:

z. B. `encoding a 720p video`

```
ffmpeg -i input_file -vcodec libvpx_vp8 -vpre 720p output_file.webm
```

`libvpx_vp8` ist der VP8-Codec und nur beziehbar per
<http://code.google.com/p/webm/downloads/list>
um dann per C-Compiler in den Quellcode von FFmpeg
reinkompiliert zu werden, also für Jedermann im Internet
selbstverständlich machbar.

2.6. on2 als Erfinder VP8

Ähnlich dem Verhalten von Google gegenüber Usern, die VP8-software suchen, geht es On 2.
Google bestimmt fremd und ist solange freundlich wie Geld fließt bzw. fließen wird.

VP8 selbst stammt von On2 <http://www.on2.com/index.php?564>

wobei On2 von Google geschluckt wurde

wobei Encoder-Tools z.T. kostenpflichtig sind (Flix-Serie) für Java bzw. Adobe Flash.

Java benutzt FXM-File
Adobe benutzt FLV-Files

2.7. aktueller Kontext zur Einführung von VP8 durch Google

Google steuert den mit der Übernahme von On2 erworbenen Videocodec VP8 bei, der ab sofort komplett als Open Source unter einer BSD-Lizenz zur Verfügung steht. Der Code kann via Git heruntergeladen werden.

Ab 19. Mai 2010 wird WebM von Entwicklerversionen von Chromium, Firefox und Opera per HTML-5 unterstützt. Microsoft Internet Explorer bleibt weiterhin veraltet und außen vor. Chrome soll WebM im Rahmen des Early Access Release Channel ab 24. Mai 2010 beherrschen.

Adobe wird WebM in Flash integrieren und will dafür sorgen, dass die meisten Rechner im Web innerhalb eines Jahres WebM per Flash-Plugin unterstützen (falls Flash benötigt wird z.B. bei Interaktionen).

Google selbst wird VP8 auf Youtube einsetzen, um HTML5-Videos in Auflösungen ab 720p abzuspielen. Um dies mit einem VP8-fähigen Browser zu testen, muss die HTML5-Beta von Youtube aktiviert werden. Bei den Videos muss dann der Parameter "&webm=1" in der URL eingefügt werden. So lassen sich dann beispielsweise Trailer (Werbung) in WebM anschauen.

Um Videos im WebM-Format zu erstellen, bietet Google diverse Werkzeuge an. So stehen ab sofort Patches für FFmpeg in der Revision #23165 zur Verfügung, die FFmpeg um Unterstützung für VP8 und WebM erweitern. Für Windows stehen Directshow-Filter zur Verfügung. Ein VP8-SDK erlaubt es zudem, eigene Encoder und Decoder zu entwickeln.

Auch einige kommerzielle Tools wie On2 Flix, Sorenson Squish, Wildform Flix und Zencoder beherrschen WebM.

Zum Thema Hardwarebeschleunigung von WebM und VP8 machte Google bislang kaum Angaben. Auf der Partnerliste des WebM-Projekts finden sich aber diverse Chiphersteller, darunter AMD, ARM, Broadcom, Freescale, Marvell, Mips, Nvidia, Qualcomm und Texas Instruments. Es ist also damit zu rechnen, dass es nur eine Frage der Zeit ist, bis die ersten Chips WebM und VP8 in Hardware beschleunigt decodieren und bei Bedarf auch codieren können.

Mozilla hat derweil Code zur Hardwarebeschleunigung von Ogg Theora veröffentlicht.

Obwohl Microsoft VP8 und Theora abgelehnt hat, wurde nun beschlossen, dass der Internet Explorer 9 auch VP8-Videos abspielen können, wenn der User den Codec SELBST in Windows einbindet. Microsoft will einen Codec, der mit rechtlichen Risiken verbunden ist (Patentansprüche), nicht mit Windows verteilen.

Broadcom will mit seinen Videocore-Chips in Kürze die hardwarebeschleunigte Wiedergabe von WebM erlauben und GStreamer kann ab sofort Videos im neuen Format codieren und abspielen.

Die MPEG LA wird einen Patentpool gründen, der mögliche Patentverletzungen durch VP8 und Theora prüfen soll, um Lizenzgebühren aus VP8 und Theora zu erzielen, wenn Codecs auf Patentverletzungen basieren. Dabei spielt es keine Rolle, dass Google VP8 freigegeben hat. Partner des Pools ist Apple.

Auch das größte Erotik-Videoportal Youporn wird in Zukunft auf den Standard HTML5 setzen (Theora bzw. VP8), sodass die Auslieferung der Videos nicht länger unter Anwendung von Adobe Flash erfolgen wird. Youporn gilt neben Youtube zu den populärsten Webseiten im Internet, die ein Großangebot auf Basis von Flash angeboten hat. Laut diversen US-Medienberichten haben die Betreiber von Youporn bereits mit der notwendigen Konvertierung der Videos begonnen. Ein vollständiger Wechsel wird jedoch noch einige Wochen beanspruchen. Durch die Aussperrung von Flash vom Apple iPad und Apple iPhone muss der Betreiber letztlich reagieren, damit auch auf diesen Geräten der Dienst weiterhin genutzt werden kann.

Apple-Chef Steve Jobs hatte bereits vor wenigen Tagen bestätigt, dass ein Patentpool gegründet werden soll, um gegen das Open-Source-Projekt Ogg Theora vorzugehen. Gleichzeitig wurde Flash von Apple-Produkten verbannt, so dass z. B. kostenpflichtige H.264 begünstigt werden kann.