Private media archives for the small budget

Egg, January 12, 2020: *Today's blog is a bit familiar, because it gives an insight into the private sector. Specifically, it's about a private media archive that was created in the last two months. To go with it, there is a new box called ArchivistaDom. This new media solution offers 10 TByte of storage space for a solid 1190 francs. And for those who prefer a bit more, the same device is available with 14 TByte for 1490 francs. Included in the new ArchivistaDom is ArchivistaDMS in the Rigi expansion stage, with which up to 30,000 files can be managed.*

That is why ArchivistaDom is needed and exists.

Some years ago it was planned to realize the private media archive with Kodi. The interface is graphically very appealing, the operation is simple and the access can be regulated with accounts. Soon we had to realize, that Kodi does not offer (apart from folder or file names) any possibilities to tag the contents. If, for example, certain movies should only be made available to children, then separate folders had to be created and assigned for them. In short, as nice and fancy Kodi looks, the handling is and remains impractical when setting up a media server.

https://archivista.ch/cms/wp-content/uploads/file/privates_medien_archiv_archivistadom.mp4

Over time, the decision matured to use ArchivistaDMS for this. At the end of October 2020 the time had come, the ArchivistaBox had both an internal video and audio player. First of all, it was a matter of capturing the private DVDs and CDs. For the DVDs **makemkv** was used, for the audio CDs **Asunder** in combination with **gnudb.org.**

Capturing "old stock'

Although makemkv is only available in a beta version, all DVDs ever acquired could be read in. It is important to mention that only copies for private use may be created, see **https://praxistipps.chip.de/kopiergeschuetzte-dvds-legal-kopieren_29960.** The created mkv files are very large, they can be converted to a MP4 file with the following script:

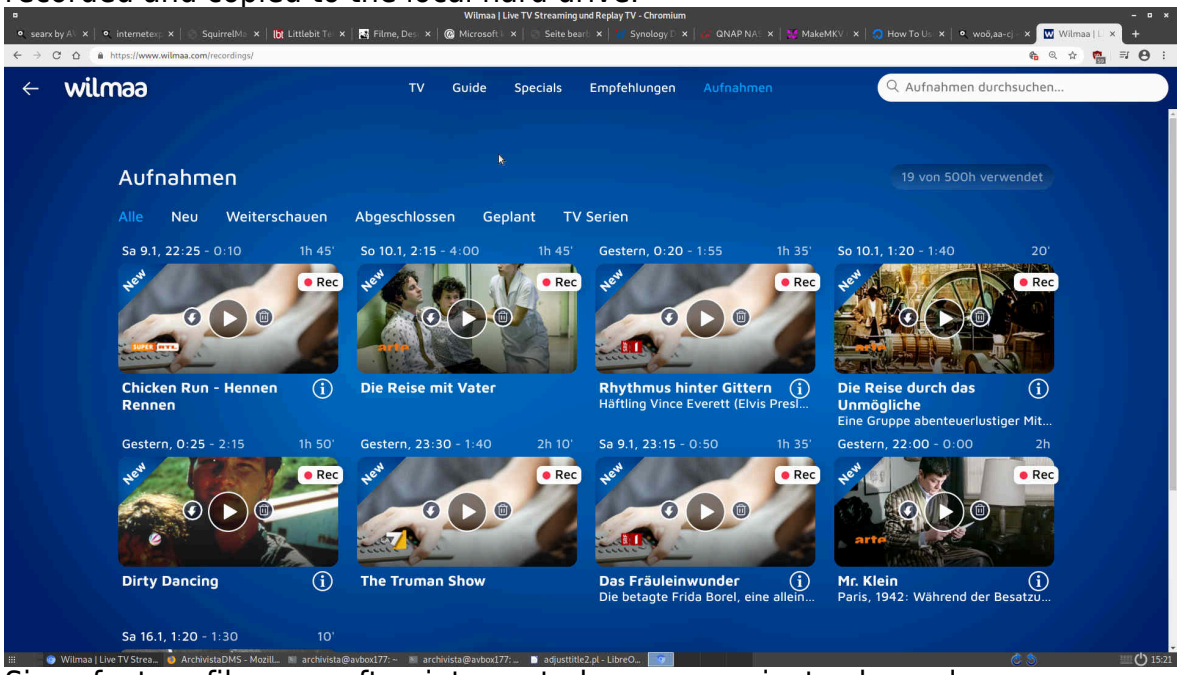
```
#!/usr/bin/perl
use strict;
my @folders = <./*>;
foreach my $folder (@folders) {
```

```
my @files = <$folder/*.mkv>;
foreach my $in (@files) {
  next if -d $in;
  my $out = $in;
  $out =~ s/\//_/g;
  $out =~ s/^(\._)//g;
  $out =~ s/(.mkv)$/.mp4/g;
  my $cmd = "ffmpeg -i \"$in\" -map 0:v -map 0:a ".
    "-disposition:a:0 0 -disposition:a:0 default -c:v
libx264 \"$out\"";
  my $res = system($cmd);
  print "$res=>$cmd\n";
 }
}
```

If the “console approach” is too complicated for you, you can also use **VLC** as an alternative. Capturing audio CDs is easy with ASunder. Insert the CD, ASunder searches the online database gnudb.org for the matching tracks and usually finds them. If this is not the case, the files can be tagged manually. The created files can then be copied into the ArchivistaDMS office folder or read in automatically.

Content from the Internet

The tools **MediathekView and JDownloader or MP3toCD** were already introduced last year. In order to record current films, the streaming service **wilmaa.ch** was used. This allows content from several hundred channels to be recorded and copied to the local hard drive.



Since feature films are often interrupted on many private channels, **LossLessCut** is recommended to remove the commercials. Besides that, there is another little “problem”. The file names at **Wilmaa.ch** start with date and contain underscores, only after that the title of the program is given. For sorting on the harddisk this is ideal, for later import or a simple search less so. Therefore a second script was created that copies the filenames “refined” into the title line.

```
#!/usr/bin/perl
use strict;
use lib qw(/home/cvs/archivista/jobs);
use AVJobs;
```

```
my $db = "moviestar";
my $titel = "";
my $subtitle = "";
my $thema = "";
my $dbh = MySQLOpen();
if ($dbh) {
  if (HostIsSlave($dbh)==0) {
    $dbh->do("use $db");
    my $sql = "select MediaName,Laufnummer from $db.archiv
".
      "where Titel is null or Titel=''";
    my $prow = $dbh->selectall_arrayref($sql);
    foreach my $prow1 (@$prow) {
      $titel="";
      $subtitle="";
      $thema="";
      my $genre="";
      my $genre2="";
      my $eigentuemer="";
      my $moviename = $$prow1[0];
      my $lnr = $$prow1[1];
      next if $lnr<=0;
      my @parts = split(/\./,$moviename);
      my $mext = pop @parts;
      my $mbase = join('.',@parts);
      $titel = $mbase;
      if (lc($mext) ne "ogg" && lc($mext) ne "mp3") {
        @parts = split('-',$mbase);
        my $anz = @parts-1;
        while ($anz>0) {
          if ($parts[$anz]>0) {
            pop @parts;
            $anz--;
          } elsif ($parts[$anz] eq "merged") {
            pop @parts;
            $anz--;
          } elsif ($parts[$anz] eq "cut") {
            pop @parts;
            $anz--;
          } else {
            $anz=0;
          }
        }
        shift @parts if $parts[0]>0;
        shift @parts if $parts[0]>0;
        shift @parts if $parts[0]>0;
        shift @parts if $parts[0]>0;
        $titel = join('-',@parts);
      } else {
        $genre = "Musik";
        $genre2 = "Kinder";
        $eigentuemer = "kinder";
        my ($titel1,$subtitle1,$thema1) = split(' --
',$titel);
        if ($titel1 ne "" && $subtitle1 ne "" && $thema1 ne
"") {
          $titel=$titel1;
```

```
            $subtitle=$subtitle1;
            $thema=$thema1;
          }
        }
        $titel =~ s/(\_)/ /g;
        $sql = "update $db.archiv set
Titel=".$dbh->quote($titel);
        $sql.=",Subtitel=".$dbh->quote($subtitle) if $subtitle
ne "";
        $sql.=",Thema=".$dbh->quote($thema) if $thema ne "";
        $sql.=",Genre1=".$dbh->quote($genre) if $genre ne "";
        $sql.=",Genre2=".$dbh->quote($genre2) if $genre2 ne
"";
        $sql.=",Eigentuemer=".$dbh->quote($eigentuemer) if
$eigentuemer ne "";
        $sql.="  where Laufnummer=$lnr";
        $dbh->do($sql);
        logit("$moviename goes to $titel in $lnr");
      }
    }
}
```

The script can be started either time-controlled via a recurring job or then directly when adding each file. The small program also does some other things. In particular, it makes all audio files available to all users (group ‘children’). This is an elegant transition to another topic, the rights system.

Media archive with rights management and keywording

There are currently four groups in the private media archive: Adult, Teen, Children and Guest. This is because certain films are not suitable for teens or children. Rights assignment in ArchivistaDMS is carried out via the ‘Owner’ field. Of course, a finer assignment of rights could also be made (e.g. U6, U10, U14, U18).

Over the last two months, several thousand videos were captured. Many broadcasts were taken from **MediathekView**. **Wilmaa.ch**, on the other hand, was primarily used for feature films. To enable a convenient selection, the moviestar database has the fields ‘Genre1’, ‘Genre2’ and ‘Genre3’ (up to five keywords would be possible). This way, the type of video (e.g. feature film) can

be specified, but also whether it is a classic, a thriller or a comedy. In contrast to other solutions (e.g. Kodi), the keyword fields can be freely defined.

Certificates with LetsEncrypt

Also in contrast to Kodi, ArchivistaDMS works completely web-based. If the ArchivistaBox is available via a public IP address and a DNS name (e.g. example.com), a certificate can be set up with Certboot to match:

```
certbot certonly --standalone --preferred-challenges http -d example.com
```

Important: Before setting up the certificate in this way, the web server must be disabled (/etc/init.d/apache2 stop). If successful, a message with 'Congrutaltions…' appears. The certificate must be included in the Apache configuration. The file can be found at /etc/apache/sites-availabe/pve.conf (below is the crucial excerpt);

```
<VirtualHost *:443>
...
SSLCertificateFile /etc/pve/pve-ssl.pem
SSLCertificateKeyFile /etc/pve/pve-ssl.key
...
</VirtualHost>
```

The reference to the certificates should be entered there:

```
<VirtualHost *:443>
...
SSLCertificateFile
/etc/letsencrypt/live/example.com/fullchain.pem
SSLCertificateKeyFile
/etc/letsencrypt/live/example.com/privkey.pem
...
</VirtualHost>
```

The generated keys must then be copied to the destkop folder:

```
cp -rp /etc/letsencrypt /home/data/archivista/cust/desktop
```

Now the keys must be activated correctly at startup. To do this, add the following commands to the desktop.sh file in /home/data/archivista/cust/desktop:

```
#!/bin/bash
fp="/home/data/archivista/cust/desktop"
cp -rpf $fp/index.htm /usr/share/pve-manager/root
cp -rpf $fp/letsencrypt /etc
cp -pf $fp/pve.conf /etc/apache2/sites-available
/etc/init.d/apache2 restart
```

The file desktop.sh must be executable (chmod a+x desktop.sh), otherwise it cannot be executed on reboot. If this is too complicated for you, you can delegate this to a professional. At the end of this procedure, the Meidenarchiv can be accessed encrypted with a public certificate (the message that the page is insecure no longer appears with all browsers).

ArchivistaDom with 10 to 32 TByte.
The current private archive comprises approx. 5 TByte. In this range, SSD hard disks, which are usually supplied with the ArchivistaBox, are currently somewhat expensive. This is why the ArchivistaDom was designed. Although the ArchivistaDom is somewhat larger (30x40x10 cm), up to two conventional 3.5″ hard disks with up to 32 TBytes can be installed.

Expandability with ArchivistaDom
The first hard drive is included, and there is a hot-swap slot for a second drive. The slot can also be used for backup as an alternative to removal. 10 TBytes are roughly enough for 5000 movies in HD quality. The ArchivistaDom contains a Ryzen CPU from AMD, just as high-quality hard disks from WesternDigital (Gold standard) are installed. With an optional hard drive (14 TB internal and 18 TB HotSwap) up to 32 TByte can be reached.

The ArchivistaDom is fully capable of editing or playing 4K videos. All applications from AVMultimedia and the ArchivistaBox are available. Various memory expansion levels between 4 and 64 GBytes are possible. The basic ArchivistaDom with 10 TByte costs 1190 francs, with 14 TByte it is 1490 francs (both figures without VAT). Existing ArchivistaBox customers can “upgrade” from their existing ArchivistaBox. The exact conditions will be calculated based on the age of the existing box.
The ArchivistaDom is sold exclusively in Switzerland. Customers from other countries purchase the ArchivistaBox Rigi Virtualized and install it on their own

computer. When ordering, please note Media Archive or ArchivistaDom. You will then receive the basic parameters for the purchase of your own suitable hardware as well as a voucher for two hours support which can be redeemed within 90 days of purchasing the solution.
At this point it should be noted that the ArchivistaBox itself can be operated in a virtualised (box-in-box) manner without any problems. However, it is not recommended that this be done for a multimedia archive in the higher TByte range. As the ArchivistaBox generally works in an extremely “low-maintenance” manner and also represents a fully-grown multimedia desktop, native operation makes much more sense in every respect. The **ArchivistaDom can be ordered in the web store.**