

Initial proposal of storage hierarchy design for video streams

All the data for video streams will be mounted under /data. We will keep raw feeds for about 1 week (This may change in future depending on us) along with the processed (Calibrated) data. Then after we will keep it under /archives (maybe even in compressed format using ZSTD).

Note: This hierarchy design is only for video feeds not for data which is stored in our time series database. I may well make another partition and mount there not with /data, again to make data retrieval faster and keep TMDB safe in case if /data may go corrupt.

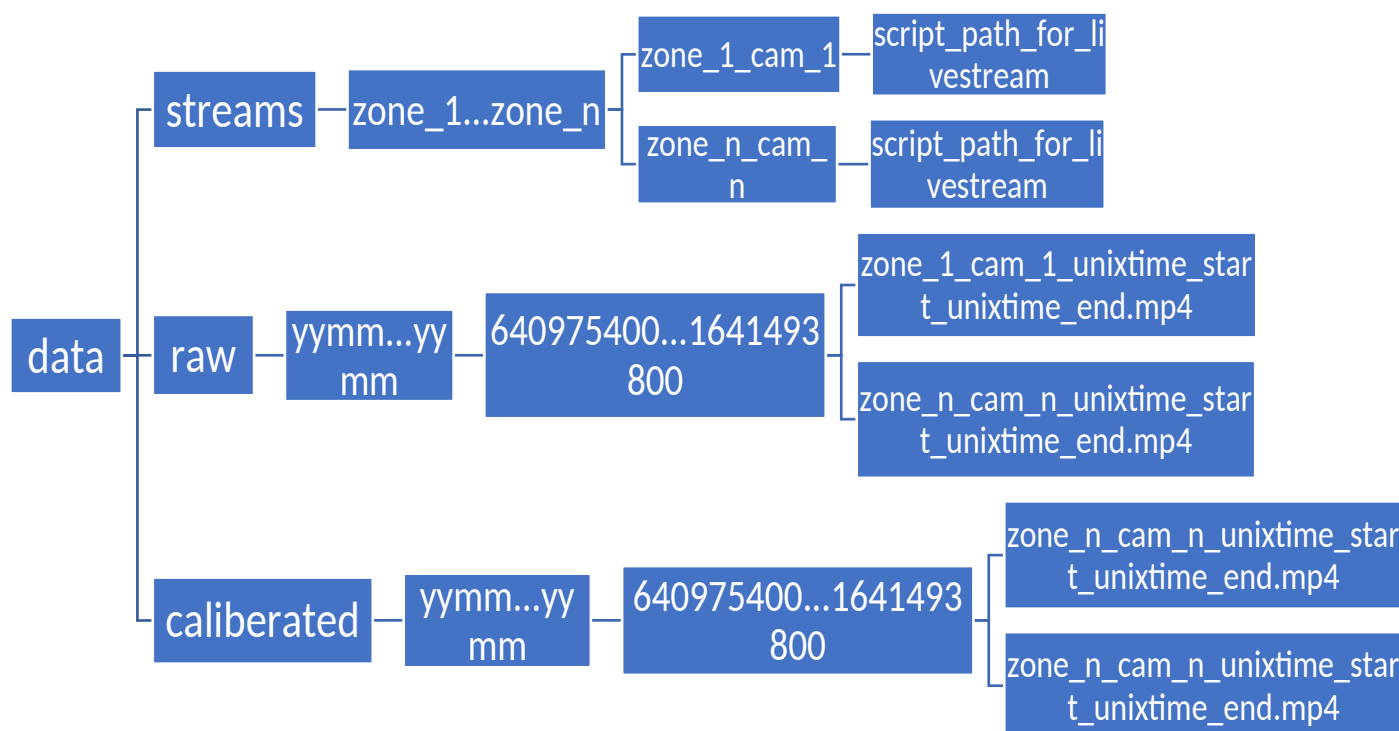


Figure: /data partition hierarchy

I have divided the streams into zones and will categorise the respective cameras along with their zones. And as this path will be a live stream so there will be a script or a binary which will access the live feed from that camera with GigE/RTSP protocol in a GUI window with the IP:PORT rather than having a video file over there.

We will cut the video feeds from the live streams continuously like in an interval of 5 min (may change) and will be stored under /data/raw/unixdatetime/zone_n_cam_n_unixtimestart_unixtimeend.mp4 for both the raw data (feeds) as well as for calibrated data but under /data/calibrated/*.

Sample data hierarchy:

```
data
├─ calibrated
│  └─ 202209
│     │  └─ 1641061800
│     │     │  └─ blast_furnance_zone_infrared_cam_1641062400_1641062700.mp4
│     │     │  └─ 640975400
│     │     │     │  └─ blast_furnance_zone_infrared_cam_1640976000_1640976300.mp4
│     │     │     │  └─ thermal_zone_monochrome_cam_1640976000_1640976300.mp4
│     └─ 202210
│        └─ 640975400
│           │  └─ blast_furnance_zone_infrared_cam_1640976000_1640976300.mp4
│           │  └─ thermal_zone_monochrome_cam_1640976000_1640976300.mp4
├─ raw
│  └─ 202209
│     │  └─ 1641061800
│     │     │  └─ blast_furnance_zone_infrared_cam_1641062400_1641062700.mp4
│     │     │  └─ thermal_zone_monochrome_cam_1641062400_1641062700.mp4
│     └─ 202210
│        │  └─ 1641061800
│        │     │  └─ blast_furnance_zone_infrared_cam_1641062400_1641062700.mp4
│        │     │  └─ thermal_zone_monochrome_cam_1641062400_1641062700.mp4
│        └─ 640975400
│           │  └─ blast_furnance_zone_infrared_cam_1640976000_1640976300.mp4
└─ streams
   └─ blast_furnance_zone
      │  └─ infrared_cam_stream
      │  └─ monochrome_cam_stream
   └─ other_zones_stream
      │  └─ infrared_cam_stream
      │  └─ monochrome_cam_stream
   └─ thermal_zone
      │  └─ infrared_cam_stream
      │  └─ monochrome_cam_stream
```

Storage hierarchy for archives

As we will archive the raw as well as processed data there will be a separate partition for archives which is /archives. The raw data will probably be kept for 1 year at max and 2 years max for processed data (which may change in future).

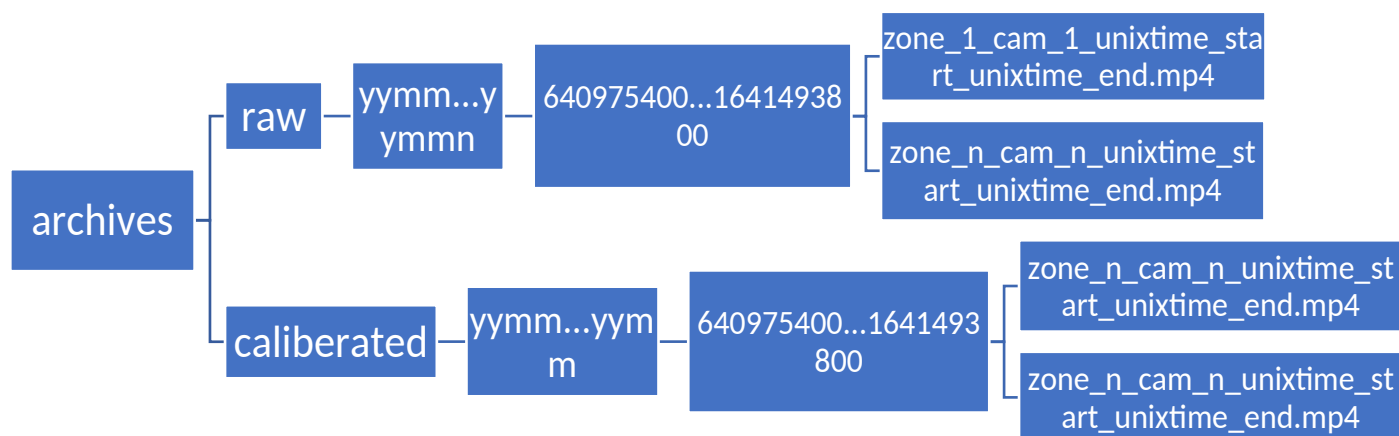


Figure: /archives partition hierarchy

Sample archive hierarchy:

```
archives
├── calibrated
│   ├── 202209
│   │   ├── 1641061800
│   │   │   ├── blast_furnance_zone_infrared_cam_1641062400_1641062700.mp4
│   │   │   └── thermal_zone_monochrome_cam_1641062400_1641062700.mp4
│   │   └── 640975400
│   │       ├── blast_furnance_zone_infrared_cam_1640976000_1640976300.mp4
│   │       └── thermal_zone_monochrome_cam_1640976000_1640976300.mp4
│   └── 202210
│       ├── 1641061800
│       │   ├── blast_furnance_zone_infrared_cam_1641062400_1641062700.mp4
│       │   └── thermal_zone_monochrome_cam_1641062400_1641062700.mp4
│       └── 640975400
│           ├── blast_furnance_zone_infrared_cam_1640976000_1640976300.mp4
│           └── thermal_zone_monochrome_cam_1640976000_1640976300.mp4
└── raw
    ├── 202209
    │   ├── 1641061800
    │   │   ├── blast_furnance_zone_infrared_cam_1641062400_1641062700.mp4
    │   │   └── thermal_zone_monochrome_cam_1641062400_1641062700.mp4
    │   └── 640975400
    │       ├── blast_furnance_zone_infrared_cam_1640976000_1640976300.mp4
    │       └── thermal_zone_monochrome_cam_1640976000_1640976300.mp4
    └── 202210
        ├── 1641061800
        │   ├── blast_furnance_zone_infrared_cam_1641062400_1641062700.mp4
        │   └── thermal_zone_monochrome_cam_1641062400_1641062700.mp4
        └── 640975400
            ├── blast_furnance_zone_infrared_cam_1640976000_1640976300.mp4
            └── thermal_zone_monochrome_cam_1640976000_1640976300.mp4
```

Storage Purging Logic

- I'm thinking to use the UNIX cron daemon for this regular purging process, as it was developed by AT&T and is part of the standard UNIX daemon so we can rely on that.

- Here is the basic cron expression:

```
# _____ minute (0 - 59)
# _____ hour (0 - 23)
# _____ day of the month (1 - 31)
# _____ month (1 - 12)
# _____ day of the week (0 - 6) (Sunday to Saturday;
#                               7 is also Sunday on some systems)
#
# * * * * * <command to execute>
```

- Let us take an example, we have millions of video feeds under /archive and we want to make sure the video feeds which are older than 2 years will be deleted regularly. So, we will make a script which will check whether there is any video or not if NULL or empty then return else check for videos whose DateTime are 2 years older than the current system time and all this process will be run every day, so to this in cron:

```
0 0 * * * <path_to_our_deleting_script_or_binary> // Every day at 12:00 AM
Or
@daily <path_to_our_deleting_script_or_binary> // Equivalent to above expression
```

- With cron daemon, we can have many combinations for our needs.
- crond also offers us to allow and deny the cron service for the users under /etc/cron.allow and /etc/cron.deny.
- And the same goes for /data, after 1 week or whatever time that we will define the raw and processed data will be moved safely to the archive in compressed format.

Some points

- There are many different ways in which we can define our storage hierarchy, for now, I have done it in the date-time manner in which all videos of the day1 are stored under the day1 folder and so on for dayⁿ and then just the video files with 5min of interval and to query the videos we can go through the DateTime folder and the name of the file. The name of the file will contain the zone and camera from which it was taken.
- Other ways can be categorizing it in zones or the camera from which was taken and then the DateTime folder or just appending that in the file. (/data/zone1...n/datetime/video_files or /data/cam1...n/datetime/video_files).
- Although it seems that there won't be any difference but ill go through it more about how the users will query the data.
- I have mentioned the limitations of file size, no. of files, file name size, volume size, etc of some file systems that we may use in the other document, please refer to that.
- As we are going to have a lot of I/O and CPU operations it's better to write asynchronous programs.
- I'll keep updating these documents on git.