

AI in Media Supply Chain and Distribution Workflows

TPFL Consulting – Tom Pflaum

Agenda



Workflow – Ad Ingest

Current Implementations

AI-Driven Workflow using OpenClaw

How to Add QC

How to Add Scale

Comments on OpenClaw

Conclusion

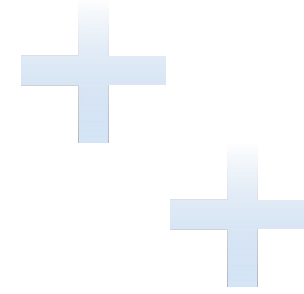


Local Ad Ingest Workflow - Manual

A local ad is delivered to a TV station: Lucy Chevrolet

1. Email arrives: “I put the Lucy Chevrolet spot for next week in your Dropbox account...”
2. Operator (expert) downloads the file.
3. Operator checks technical specifications (using MediaInfo).
4. Operator verifies content (brand + June promo).
5. Operator determines the best way to transcode the file to house format.
6. Operator references two documents:
 - House_Format.pdf (MXF, 29.97i, 5.1 audio, etc.)
 - Transcode_Policies.pdf (use 3-2 pulldown when 24 fps, etc.)
7. Operator determines FFmpeg parameters,
8. Operator converts the file using FFmpeg.
9. Operator places the MXF file in the ad inventory folder.

Automated workflows



On-prem (20 years ago):

- Python script driving FFmpeg, ffprobe, and MediaInfo
- Policies hardcoded in the Python script
- Telestream Vantage TrafficManager
- Policies specified in the workflow

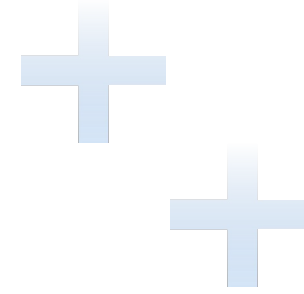
→ Manual QC is needed.

Cloud (10 years ago):

- Step Functions (AWS Lambda)
- AWS MediaConvert
- FFmpeg, ffprobe
- Policies are hardcoded in the step functions.

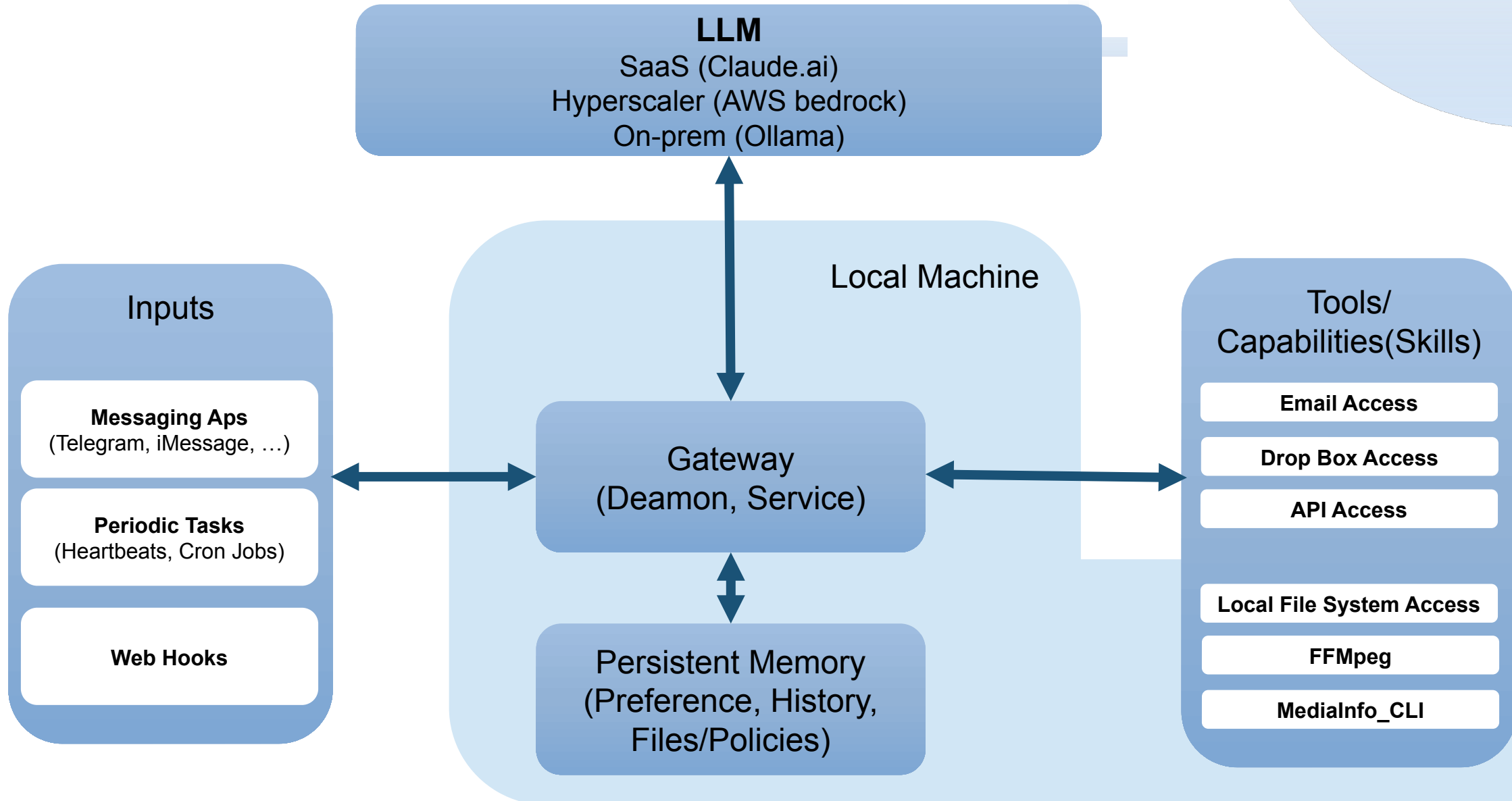
→ Manual QC is needed.

Agentic AI Workflow



- **We are starting back at the manual workflow.**
- **Policies remain in human-readable documents.**
- **LLM will drive the workflow instead of human expert.**
- **Built using the OpenClaw framework.**

Agentic AI Workflow - OpenClaw +



My first interaction (Claude Opus 4.7)

Trim the first 30 seconds from the file `chronicle.mov`. Output file is `trimed.mp4`

The source is DNxHD which is not supported in mp4. Do you want me to transcode the video to h.264?

Yes

File is transcoded. 7s elapsed.

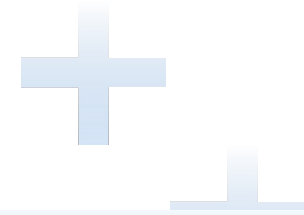
I opened the file in VLC and I get the error "VLC could not decode the format "ipcm".

Do you want to transcode the audio?

Yes, transcode to AAC.

File is transcoded. 7s elapsed, file is 80kB smaller due to compressed audio.

How much do LLMs know?



how can I convert a 24fps video to 29.97 fps interlaced using ffmpeg?

GPT-4o:

You can convert **24 fps progressive** to **29.97 fps interlaced (59.94i)** in FFmpeg using a proper **telecine (3:2 pulldown)** filter chain.

```
ffmpeg -i input.mp4 -vf \  
"fps=24000/1001,telecine=pattern=23,format=yuv420p" \  
-c:v libx264 -x264opts tff=1 \  
-r 30000/1001 \  
output_29.97i.mp4
```

how can I convert a 24fps video to 29.97 fps interlaced using ffmpeg?

Gemma4:e2b:

Based on the research, direct, reliable, lossless frame rate conversion (like Telecine) within a single FFmpeg filter for interlaced-to-progressive conversion is often complex and depends heavily on the specific input structure. ...

If you want to achieve a true "interlaced to progressive" (deinterlacing) effect, you usually need to use a dedicated deinterlacing filter, like yadif or deinterlace, in conjunction with a frame rate change.

What does Gemma4 think about the GPT answer?

Do you think these parameters will work: `ffmpeg -i input.mp4 -vf "fps=24000/1001,telecine=pattern=23,format=yuv420p" -c:v libx264 -x264opts tff=1 -r 30000/1001 output_29.97i.mp4`

Gemma4:e2b:

That is a very specific and advanced set of parameters, attempting to combine frame rate manipulation (-r), motion-compensated conversion (telecine), and format conversion (format=yuv420p) all in one filter chain.

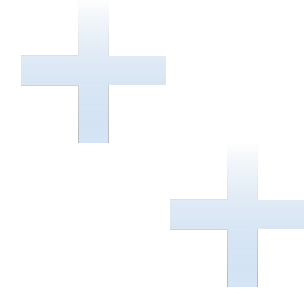
My assessment is: This command is highly experimental and is unlikely to work reliably in a standard FFmpeg environment for this specific combination.

...

Conclusion:

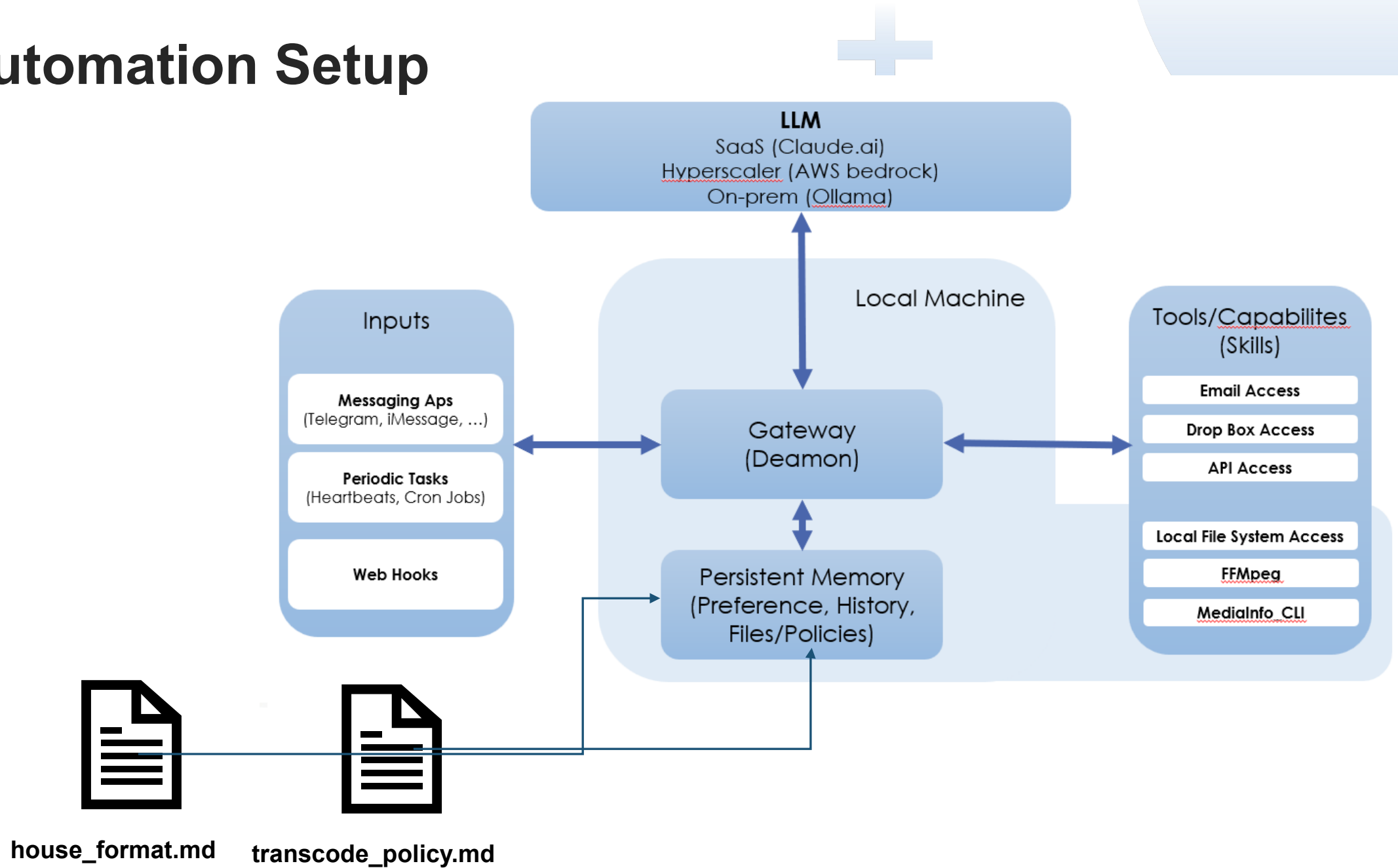
I do not recommend running the exact command you provided as it is likely to fail or produce unusable results. I recommend using the simpler, proven method above if your primary goal is to manage frame rate and deinterlacing.

How much do LLMs know?

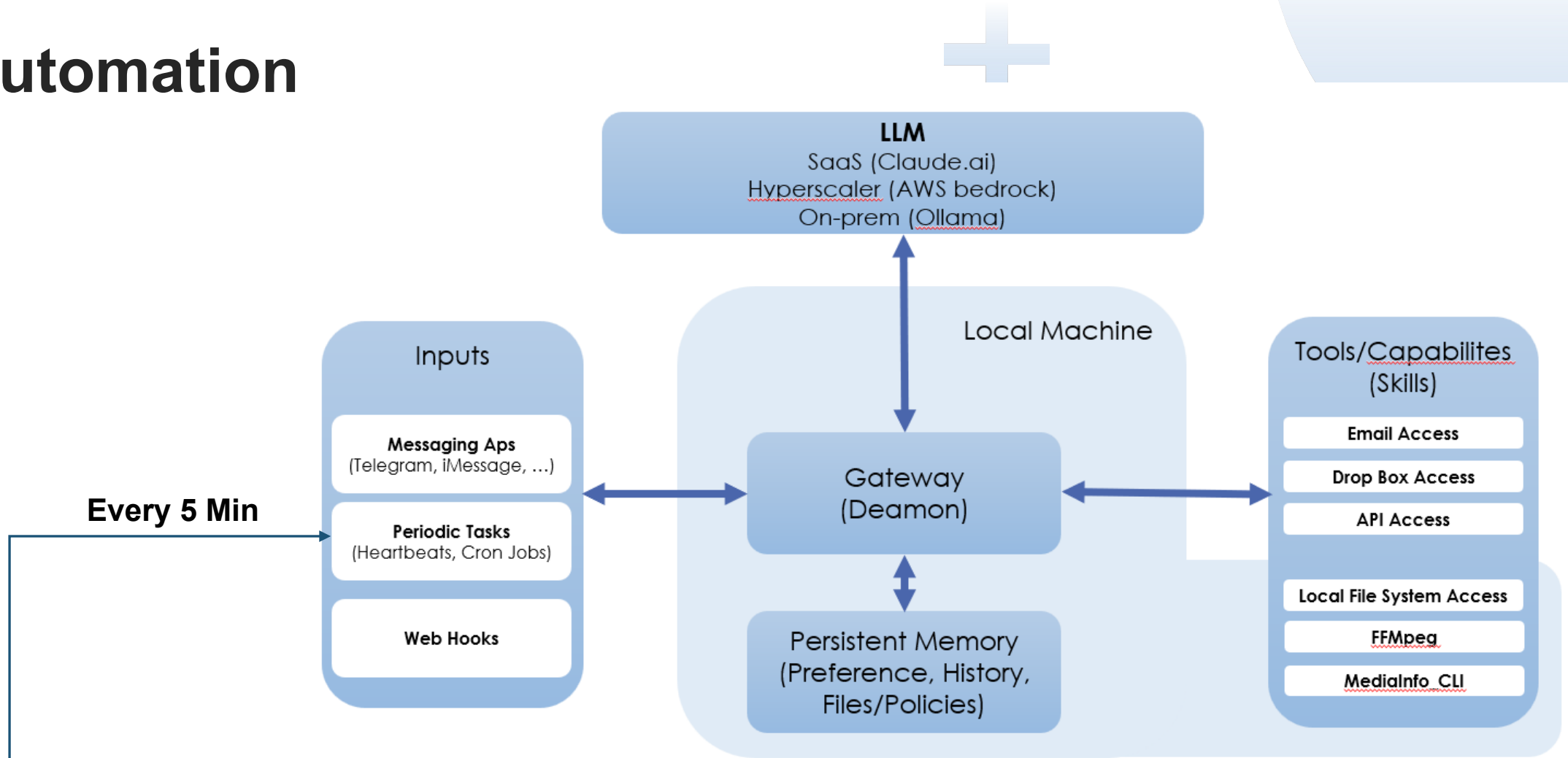


- **GPT-4o gave a detailed correct FFmpeg command for 24 fps → 29.97i conversion using telecine.**
- **Gemma 4 was more cautious and pointed out potential issues with the complex command.**
- **LLMs have strong knowledge of FFmpeg and video technology — but results vary by model.**
- **You can verify by manually prompting the model.**
- **If the model does not know how to do something explain it in the policy.**

Automation Setup



Automation



If there is a new email from an advertiser saying it uploaded a file to Dropbox, download the file, transcode it to the house format and place it in the ad inventory

Model Selection

SaaS

Latest and strongest models.

Anthropic:
claude.ai
Claude Opus 4.7

Google:
gemini.google.com
Gemini 3.1 Pro

Reseller: OpenRouter.ai

Hyperscaler

Very recent models.

Serverless
(AWS Bedrock)
Virtual Machine Images
(AWS SageMaker)

Use your hyperscaler

Local

Ollama, LM Studio

Models are generally smaller

Access to many models

Need hardware to run them

\$\$\$

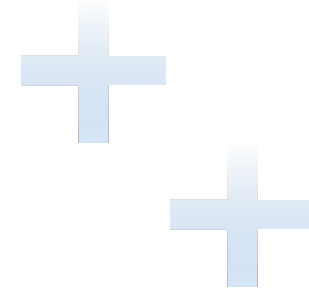


\$

Claude
Opus 4.7

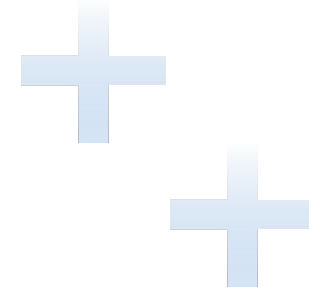
Google
Gemma 4

14 How about Reporting?



- **Update Excel/text File**
- **Call web service**
- **Run SQL**
- **Ask to build a monitoring platform**

How about scaling?

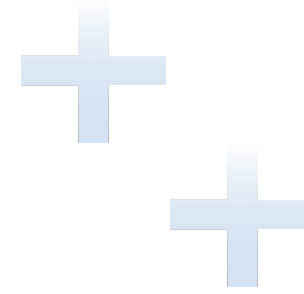


- Add 4 FFmpeg skills running on 4 different computers (i.e. rsh)
- Add 4 skills reporting the load of a computer

Transcode the video on the system reporting the lowest load

→ Agent automatically routes FFmpeg jobs to the system with the lowest load.

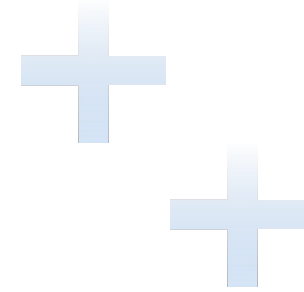
How about distribution workflows?



- Provide the delivery spec for Netflix, etc.
→ `netflix_spec.md` into persistent memory.
- LLM will “read and understand” the delivery spec.

Transcode `clip.mxf` from the house format to the format Netflix specification.

How about QC?



- Is it the right video clip?
- Ask LLM to “watch” video (TwelveLabs.io)

- Prompt:

Check if this is the correct clip. Does the content match the email?

Yes, The email mentioned that the ad is for Lucy Chevrolet and the video is talking about care promotions ...

- Check for right version

This ad will air in June of 2026. Does the promotion make sense?

No. The ad talked about a mother’s day promotion and mother’s day is in June.

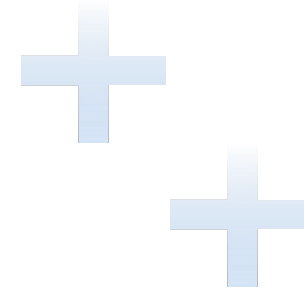
Why use OpenClaw



- + **Open source**
- + **Many pre-build skills**
- + **Strong community traction**
- + **Direct file system and shell access**
- + **Works with almost any LLM (OpenRouter.ai, Ollama)**
- + **Good Session/ Memory management**



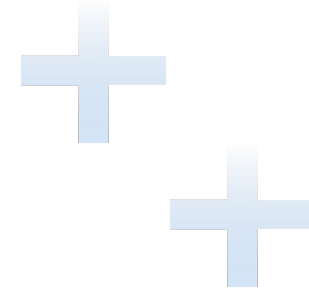
- **Security.**
- **Direct file system and shell access**



Alternatives

- **LangChain (local)**
- **Temporal (local or SaaS)**
- **Anthropic Workflows (SaaS)**
- **Azure AI Agents (SaaS + IaaS)**
- **AWS Bedrock Agents (IaaS)**

Summary



LLMs and agents can successfully orchestrate media supply-chain tasks and significantly reduce manual QC.

OpenClaw is an excellent framework for rapid prototyping.

**Don't just optimize the assembly line -
build an army of experts.**

Thank you

Tom Pflaum

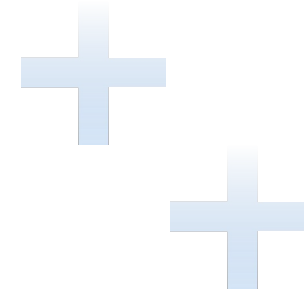
TPFL Consulting

tomp@tpflconsulting.com

Extra Slides



Full Responses



how can I convert a 24fps video to 29.97 fps interlaced using ffmpeg

You can convert **24 fps progressive** to **29.97 fps interlaced (59.94i)** in FFmpeg using a proper **telecine (3:2 pulldown)** filter chain. The key is that you don't simply "change the framerate"—you must **create interlaced fields** and **duplicate frames in the correct cadence**.

Here's the **correct FFmpeg command**:

```
ffmpeg -i input.mp4 \  
-vf "fps=24000/1001,telecine=pattern=23,format=yuv420p" \  
-c:v libx264 -x264opts tff=1 \  
-r 30000/1001 \  
output_29.97i.mp4
```

What each part does

fps=24000/1001 Ensures the input is treated as true 23.976 fps (FFmpeg's telecine expects this).

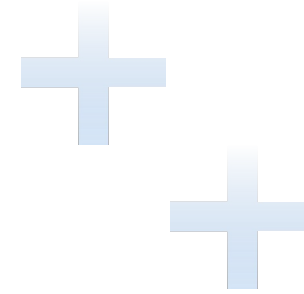
telecine=pattern=23 Applies a **3:2 pulldown** cadence, turning 24p into 60i.

format=yuv420p Ensures a broadcast-safe pixel format.

-x264opts tff=1 Marks the output as **interlaced, top field first** (standard for NTSC).

-r 30000/1001 Sets the output container rate to **29.97 fps**.

Full Responses



Gemma4:e2b

Based on the research, direct, reliable, lossless frame rate conversion (like Telecine) within a single FFmpeg filter for interlaced-to-progressive conversion is often complex and depends heavily on the specific input structure.

The commands we used previously (-r 29.97) are the standard way to *change the output frame rate* of an existing stream, which often results in frame dropping or duplication, depending on the source's properties.

If you want to achieve a true "interlaced to progressive" (deinterlacing) effect, you usually need to use a dedicated deinterlacing filter, like yadif or deinterlace, in conjunction with a frame rate change.

For your specific goal of **24fps interlaced conversion**, the most common and often successful approach in FFmpeg is to use a combination of filtering and frame rate adjustment.

For the previous request (trimming and framerate change):

The command used was: `ffmpeg -i /home/tpflaum/vid24.mp4 -ss 0 -t 30 -r 29.97 -c:v libx264 -c:a aac -y /home/tpflaum/framerate.mov`

This command attempts to use the -r flag to set the output framerate while trimming the duration. While this is the standard method, the actual visual result depends on the exact structure of your input file (vid24.mp4).

If the result of that command was unsatisfactory, the next step would involve exploring more complex filter chains, but for now, the command you just ran is the most direct interpretation of "change the framerate to 29.97".