Video AI Inference Serving - Decision Checklist

Fora Soft Learn · AI for Video Engineering

1 - The two ideas that make an LLM/VLM server fast

```
PagedAttention      KV cache in small pages on demand -> waste drops from 60-80% to under 4%.
Continuous batching new request slots in the instant one finishes -> GPU 30-40% busy becomes 75-90%.
Together: ~2-4x the throughput of the previous server generation, on the SAME GPU.
```

2 - The serving choices (open + NVIDIA + serverless)

```
vLLM            the default for LLMs & VLMs; multi-user; OpenAI-compatible; Docker; NVIDIA/AMD/TPU.
Ollama          one user, one machine (laptop / Apple Silicon). Prototyping only - NOT for a fleet.
SGLang          prefix-heavy work (shared rubric, RAG, multi-turn); ~29% over vLLM, up to 6.4x on prefixes.
TensorRT-LLM    max speed on NVIDIA (15-30% over vLLM). 2026: PyTorch backend, no engine compile step.
Triton          many models on one fleet; ensembles = pipelines in-server. Successor: NVIDIA Dynamo (2025).
Serverless      Modal / RunPod / Replicate; scale to zero; pay by the second; watch cold starts on big models.
```

3 - A video feature is a pipeline - match each stage to a server

```
Decode (frames+audio)        FFmpeg / decoder.
Detect / segment             CV models (YOLO, SAM) -> TensorRT, fixed input -> fixed result, one pass.
Transcribe (speech-to-text)  faster-whisper on CTranslate2 (~4x faster; far faster batched).
Describe (vision-language)   VLM -> vLLM (or SGLang / TensorRT-LLM).
Summarize (language)         LLM -> vLLM. Wire it all together; for one fleet, a Triton ensemble.
```

4 - The cost reality (one batching example)

```
85% busy / 35% busy = 2.4x more tokens from the same card.
$3.00/hr / 1.0M tok = $3.00 per million;  $3.00/hr / 2.4M tok = $1.25 per million  ->  58% cheaper, same GPU.
```

5 - Before you commit the serving plan, ask:

- [ ] Is every stage of the feature served by software that fits its model type, not one server tuned for one stage?
- [ ] Are language/vision-language stages on vLLM (or SGLang/TensorRT-LLM), with PagedAttention + continuous batching ON?
- [ ] Is Ollama confined to prototyping, and NOT the production architecture by inertia?
- [ ] Is speech served on a specialized engine (faster-whisper/CTranslate2) rather than the LLM server?
- [ ] Are computer-vision models compiled to TensorRT for real-time, single-pass inference?
- [ ] If many models share one fleet, are they wired as a Triton ensemble (or routed via Dynamo at datacenter scale)?
- [ ] Is spiky/low traffic on serverless (scale to zero), and steady/high traffic on a kept-warm dedicated cluster?
- [ ] Do we track cost PER TOKEN, not just latency - and have we measured the batching gain on our own workload?