LIVEWYER

Time-Addressable Media Open Source Store

TAMOSS is an open-source, Kubernetes-native implementation of the BBC TAMS API. One product, one deployment model, with the flexibility to run it based on your scenario — a single laptop, a server, a full hybrid cloud setup, or at enterprise scale across regions. Complete flexibility, all without vendor lock-in.

WHAT IS TAMOSS?

An open API specification, made deployable.

TAMS (Time-Addressable Media Store) is an open API specification, originally developed by the BBC. It defines how media is broken into time-stamped segments and how those segments are stored, retrieved, and reassembled.

The TAMS API deliberately doesn't define how the store is deployed.
That's where TAMOSS comes in.

TAMOSS (Time-Addressable Media Open Source Store) is the production-ready, deployable platform. We package the TAMS API with Kubernetes as a thin orchestration layer, plus the ingest, monitoring dashboard, authentication, and player you need to run a TAMS store in production.

The same product runs on a laptop, a single server, or a multi-region cluster. You choose the profile to match the workload. The product underneath is the same.

DEPLOYMENT PROFILES

The same product, sized to your workload.

Local · DEPLOYMENT 01

A single node. Runs on your laptop with Kind in just two commands, and you have the full stack.

Best for: remote editing, development, demos

Hardware: Your laptop

Single server · DEPLOYMENT 02

One dedicated machine. Choose between on-prem, a VM or in your chosen cloud.

Best for: single productions, OB trucks, small facilities

Hardware: one server, on-prem, cloud or virtual

Multi-server cluster · DEPLOYMENT 03

A single node. Runs on your laptop with Kind in just two commands, and you have the full stack.

Best for: facility-wide and multi-site operations

Hardware: any conformant Kubernetes

TAMOSS IN PRODUCTION

Built for media, built by platform engineers.

Tool choice flexibility

Vendor-neutral store
Any application that speaks the TAMS API can read from it and write to it. Whether that is your editor, your transcoder, your playout system, or your archive.

Camera-to-cloud ingest
Ingest content from cameras and devices in the field, without committing to one vendor's ecosystem.

One pattern, every workflow
News, sport, and post-production workflows can run on the same underlying infrastructure.

A cost shape you control
Pay for the storage and compute you actually use, not for a per-seat licence on the store itself.

Production-grade deployment

BBC TAMS v8.0, no extensions
No proprietary API surface. Only Kubernetes-native conventions (/healthz, /readyz) plus runtime concerns the spec leaves open.

Modern stack, no surprises
FastAPI backend. PostgreSQL for metadata. RustFS for S3-compatible object storage. Helm-deployable on EKS, GKE, AKS, or any conformant Kubernetes.

OpenTelemetry-ready
Conditional instrumentation. Bring your own collector, Prometheus, and Grafana, with no bundled observability stack to fight.

Durable async lifecycle
Workers for flow deletion, webhook delivery, and object cleanup. PostgreSQL-backed claim semantics with exponential-backoff retry. The operations work has already been done.

HOW TO DEPLOY

Open source, pilot, or partnership.

Self-serve

Free · MIT License · Community support

Clone the repository. Deploy with Helm by following the step-by-step guide on GitHub. This is a completely free TAMOSS deployment, with free support available via the TAMOSS community Slack.

Best for: engineering teams evaluating TAMOSS, smaller deployments, and contributors.

LiveWyer Pilot

Fixed price · Fixed scope · Real workloads

We deploy, validate, and hand over a working TAMOSS using your chosen real-world workload on your chosen target. A fixed-price end-to-end Pilot with zero obligation for any further delivery efforts.

Best for: organisations validating TAMOSS against a specific workflow or production.

LiveWyer ProServe

Fixed price or T&M · Personalised scope

If your team needs support with implementation, LiveWyer have the delivery capability to help you succeed, integrating within your existing engineering team to supplement delivery and add immediate value.

Best for: organisations looking to build on the pilot implementation and need additional engineering support.

Delivery Partnership

Fixed price or T&M · Delivery support

We collaborate with our partners and their customers to architect, build, validate solutions. If you have a customer who could benefit from an Open Source TAMS implementation, let's discuss how we can help.

Best for: Current and future partners with clients who need support with their TAMOSS implementations.

FREQUENTLY ASKED

The questions worth meeting head-on.

Isn't Kubernetes expensive?
The cost of Kubernetes scales with what you ask of it. A single-node TAMOSS deployment is cheap to run; a multi-region cluster costs what a multi-region cluster costs. The supervisor doesn't impose cost — the workload does.

Is this only viable for large deployments?
No. The open-source product runs on a laptop. The LiveWyer paid services target scale, but TAMOSS itself is designed to start small and grow.

How does this compare to running TAMS on AWS?
Complementary, not competitive. AWS provides a serverless TAMS implementation. TAMOSS provides a Kubernetes-native one. If your workload is happy on AWS, run it there. If you need on-prem, hybrid, multi-cloud, or portable infrastructure, that's where TAMOSS earns its place.

What's the relationship with the BBC?
TAMS is an open API specification stewarded by the BBC. TAMOSS is an independent open-source implementation by LiveWyer. We track the upstream spec faithfully — no proprietary extensions, no API drift.

Want to see it running?
Book a demo — live deployment, full walkthrough, your questions answered.

How to get involved
TAMOSS is open source. Try it, fork it, or contribute on Github.