

Timezer Edge — Technical architecture

Technical overview of how Timezer Edge connects to Timezer Cloud, local cameras, MediaMTX, clip generation, and Cloudflare R2.

1. Big picture

Timezer splits Cloud (business, bookings, admin, clip metadata) from Edge (venue appliance: RTSP capture, buffer, FFmpeg, overlay, upload). One Edge Agent runs per venue mini-PC; cameras share that LAN.

2. Components

- Timezer Cloud (Easybook MVC): Edge Admin, Court Replay, /api/edge/v1/*, R2 signing
- Edge Agent (.NET Worker): activate, heartbeat, command queue, buffer, CreateClip
- MediaMTX: local RTSP/HLS broker (optional; cameras may also be direct RTSP)
- FFmpeg: segment recording, [anchor-N, anchor] cut, 1080×1920 overlay, encode
- Cloudflare R2: MP4/thumb storage via presigned URLs

3. Activation flow

Admin creates EdgeDevice + activation code → mini-PC starts with ActivationCode → POST activate → Cloud returns device JWT → agent stores under data/ for heartbeats.

4. Heartbeat and commands

About every 30s the agent sends health (CPU, disk, MediaMTX, cameras). Cloud replies with queued commands (e.g. CreateClip). v1 uses heartbeat queueing — not WebSocket.

5. Court ↔ camera mapping

Camera.IdResource binds a camera to a Resource (court). A replay request resolves resource → camera → one CreateClip.

6. Clip window

anchorUtc = button/click time. Edge extracts [anchor – durationSeconds, anchor] from the rolling buffer. Heartbeat delay must not shift the scene into the future.

7. Overlay

Per Resource: top/bottom strips with height 0–200 px. Video is normalized to 1080×1920; each strip is cover+cropped into its band.

8. Upload

Agent requests upload-targets → presigned PUT to R2 → complete on Cloud. Download uses signed GET (Admin and booking owner).

9. Logical diagram

IP cameras (LAN)

- MediaMTX / direct RTSP
- Edge Agent (buffer + FFmpeg + overlay)
- Cloudflare R2 (video)
- Timezer Cloud (metadata, UI, commands)

Player / Admin ↔ Timezer Cloud (HTTPS)

10. Security

- Device JWT signed by Timezer (not Auth0)
- RTSP passwords encrypted in Cloud and pushed to Edge
- MediaMTX API preferably localhost-only in production compose