

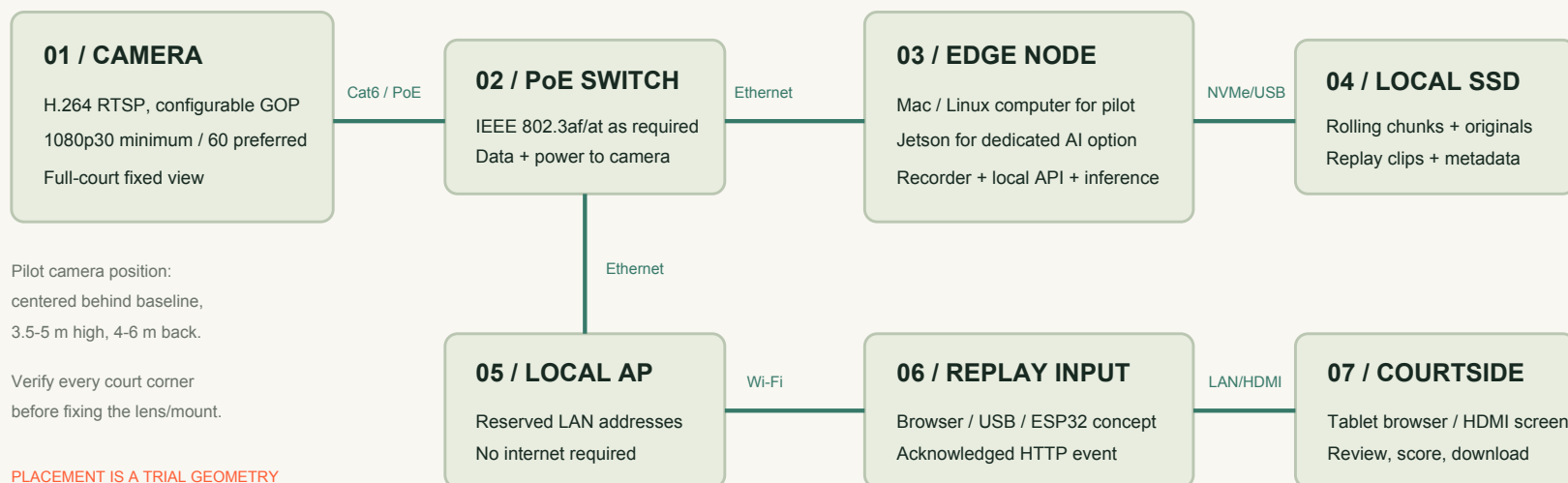
Uncle Ref / Courtside hardware blueprint

REVISION 0.1 | 14 OCTOBER 2026 | ENGINEERING PROTOTYPE

A local camera, rolling recorder and replay studio. This reference design is ready for a bench build; physical camera, thermal and venue tests remain outstanding.

UNCLE REF / SINGLE-COURT EDGE SYSTEM

Reference topology - revision 0.1 - 14 October 2026 - hardware not commissioned



POWER PATH: CERTIFIED UPS - SWITCH / ROUTER / EDGE POWER SUPPLY

Edge computer uses its approved supply. PoE is only for compatible powered devices. Do not feed raw PoE into GPIO.

Mount camera safely. Keep compute enclosure ventilated. Qualify outdoor protection, surge protection and thermal performance.

1. Build the first court

Use a wired IP camera and an existing Mac or Linux computer for the first real-court pilot. This keeps the camera, recording, inference and button independently replaceable. The complete local software runs on the supplied Mac workspace; no physical camera or embedded unit was available for commissioning.

The primary proposed dedicated build is a PoE camera plus Jetson Orin Nano Super and NVMe storage. A Raspberry Pi 5 is a capture/control alternative; equivalent neural inference throughput is not assumed. No camera SKU has been purchased or qualified.

2. Signal and power topology

1. Camera RJ45 -> outdoor-rated Cat6 -> IEEE 802.3af/at PoE switch. Confirm the selected camera's negotiated PoE standard and worst-case power.
2. PoE switch LAN -> edge computer Ethernet. The edge computer uses its own manufacturer-approved power supply; do not connect raw PoE voltage to GPIO or USB.
3. Edge computer -> internal NVMe or supported USB 3 SSD. Retain originals and exports outside the short replay ring.
4. Local router / access point -> tablet browser and optional Wi-Fi button. Reserve stable addresses; time-sync the camera, edge computer and button.
5. Optional HDMI -> courtside display. The present UI can play local video; unattended casting and TV switching are future software work.
6. AC -> certified UPS -> switch, router and edge supply. Qualified installers handle mains, earthing and outdoor surge protection.

3. Bill of materials

Quantities are for one court. Prices deliberately remain supplier quotes; taxes, availability, housings and region materially affect landed cost.

Item	Qty	Required specification	Purpose / qualification
IP camera	1	H.264 RTSP, manual exposure, configurable GOP, 1080p30 minimum; 1080p60 preferred	Require a real RTSP sample before buying; avoid cloud-only cameras
Lens / FOV	1	Capture all playing lines with margin from intended mount	Verify corners physically; wide angle reduces pixels per ball
PoE switch / injector	1	IEEE 802.3af/at compatible with selected camera, sufficient total power	Add capacity for IR/heater peak load; no passive-PoE assumption
Edge computer	1	Existing Apple Silicon Mac / Linux PC for pilot	Current implementation tested on arm64 Mac, not Jetson
Dedicated AI option	1	Jetson Orin Nano Super dev kit, supported JetPack and supply	CUDA/TensorRT deployment and throughput must be commissioned
Capture-only option	1	Raspberry Pi 5 8 GB, active cooler, suitable 5 V / 5 A supply	Use compressed RTSP stream copy; benchmark CPU inference separately
Storage	1	1 TB endurance-rated SSD, compatible adapter	Monitor free space and SMART; size for actual retention
Network cable	As needed	Outdoor-rated Cat6, weatherproof glands, strain relief	Use conduit where exposed; respect Ethernet distance limits
Mount and housing	1	Rigid rated mount, safety tether; outdoor enclosure for location	No prototype enclosure IP rating is claimed
Replay control	1	Browser for pilot; proposed ESP32-WROOM-32 DevKit for wireless	Physical button and battery life untested
Pushbutton	1	Normally open, momentary, enclosure-mounted	Debounce and feedback required
UPS	1	Power and runtime sized from measured total load	Graceful stop and power-loss recovery must be tested
Courtside screen	Optional	Tablet browser or HDMI display	Tablet and TV are not supplied

NVIDIA specifies the Orin Nano Super as an edge AI developer kit. Raspberry Pi lists a 2.4 GHz Cortex-A76 CPU, Gigabit Ethernet, USB 3, PCIe and a 5 V / 5 A supply requirement for Pi 5. These are manufacturer specifications, not observed performance of Uncle Ref. Sources are listed in section 11.

4. Camera placement and optics

Initial tennis trial geometry: centered 4-6 m behind the baseline, 3.5-5 m above the playing surface, subject to venue permissions and a safe mount. These are design starting points, not installation instructions for every venue. Capture the full near baseline, far baseline, doubles sidelines, and ball toss. Keep the camera rigid and disable digital stabilization that moves the court frame.

A camera at 4.5 m height and 4 m behind a baseline has a near-baseline slant distance of about 6.0 m. To see an 11 m width there, a first estimate is $2 * \arctan(11 / (2 * 6.0)) = 85$ degrees horizontal FOV. Add margin and confirm using a live image. Actual lens distortion, sensor cropping and pitch change this estimate. A roughly 90-110 degree horizontal field is a trial range, not a purchase guarantee.

Use manual shutter trials around 1/500-1/1000 s in adequate light to reduce motion blur. Indoor lighting may require more light or different exposure; test flicker, image noise and ball visibility. A 1080p frame spanning 20 m provides only about 96 pixels/m: a 67 mm ball is roughly 6 pixels across even before perspective reduces it. This is why general person detection cannot establish ball-level accuracy. 4K can increase sampling detail but does not itself solve occlusion, calibration or latency.

7. Mechanical / thermal drawing notes

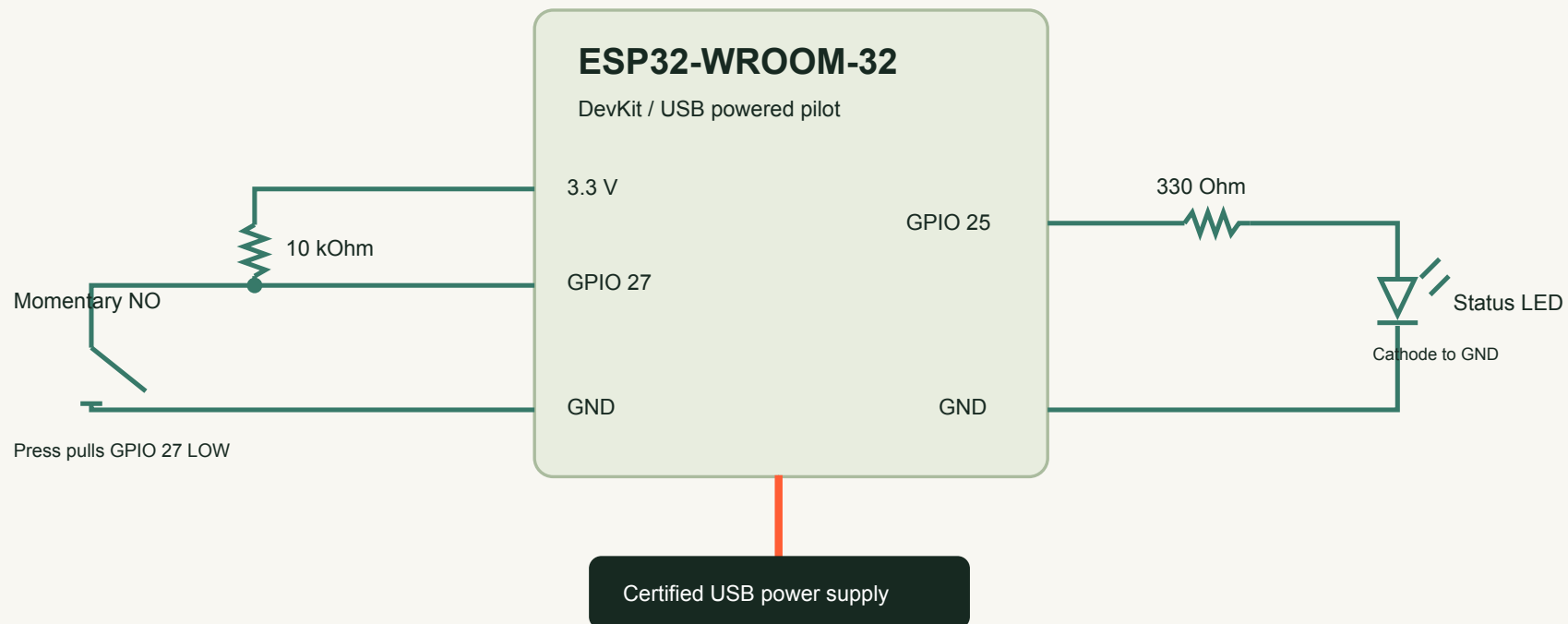
The product illustration is conceptual, not a manufactured enclosure. Use a rigid camera bracket with a secondary tether. Locate the compute node in a lockable ventilated indoor cabinet where possible. Keep fan intake and exhaust clear; isolate rain ingress from airflow. Use drip loops and glands for outdoor cables. Do not put a fan-cooled development kit into an unventilated sealed box.

Production mechanical drawings must be derived from the chosen board's official mounting-hole CAD, connector clearances and the selected camera. Do not drill from the landing-page illustration. Thermal soak, condensation, UV, water ingress and corrosion tests determine the final enclosure design.

5. Prototype replay button - wiring

UNCLE REF / REPLAY BUTTON PROTOTYPE

Classic ESP32-WROOM-32 DevKit concept - chip GPIO names, not header positions - verify board pinout



Optional 100 nF from GPIO 27 to GND. Add firmware debounce, visible acknowledgment and bounded retries.

Never connect 5 V to GPIO. Battery charging, deep-sleep firmware and enclosure qualification are outside this prototype.

Proposed board: classic ESP32-WROOM-32 DevKit; verify the exact board pinout before assembly. GPIO numbers below are chip GPIO names, not header positions.

- GPIO 27 -> normally-open momentary button -> GND.
- GPIO 27 -> 10 kOhm pull-up -> 3.3 V. Optional 100 nF from GPIO 27 to GND for noise filtering; firmware also debounces.
- GPIO 25 -> 330 Ohm resistor -> LED anode; LED cathode -> GND.
- Power the development board through its documented USB input using a certified low-voltage supply. Do not apply 5 V to a GPIO pin.
- Keep the prototype on USB power. Battery capacity, charger safety, deep-sleep wake design and outdoor housing are separate engineering tasks; no six-month battery claim is made.

`hardware/button_client.py` is an executable host-side button protocol simulator. It persists event IDs for retries. The current camera endpoint saves the buffer at request arrival. Therefore delayed offline button presses cannot recover their original event time in this release. The simulator marks stale events for review rather than silently submitting them as if they just happened. A production wireless button needs synchronized UTC timestamps, a persistent sequence, replay protection, mapped capture-time segments, acknowledgments and explicit expired-event errors.

6. Storage, bandwidth and power budget

For bitrate B in Mbps, recording H hours/day on N courts:

$$\text{GB/day} = B * 3600 * H * N / 8 / 1000$$

At 8 Mbps and 12 h/day, one court generates 43.2 GB/day; two courts generate 86.4 GB/day. Seven days for one court requires about 302.4 GB before filesystem, exports and safety margin. A 120 s rolling buffer consumes about 120 MB/court at 8 Mbps, plus keyframe and container overhead. Reserve at least 20% free storage as an operating policy.

8 Mbps video uses roughly 8 Mbps on the LAN plus transport overhead. Allocate headroom for replay downloads and multiple cameras. Prefer wired capture; Wi-Fi airtime is shared. No internet uplink is necessary for local capture. Remote video delivery would create a separate bandwidth and security budget.

Measure AC draw under recording, inference, display and worst ambient temperature before selecting a UPS. Required usable battery Wh is approximately $\text{watts} * \text{hours} / \text{conversion_efficiency}$. Add manufacturer-specific reserve and ageing allowances. Avoid estimating battery life from a neural accelerator's advertised TOPS.

8. Software commissioning

1. Install pinned Python dependencies, model weights and application before the offline trial.
2. Verify the RTSP feed is H.264 and decodes locally; set a two-second GOP where supported.
3. Start Uncle Ref on loopback. For tablet access, deploy authenticated TLS access on the local network; the prototype is not a public internet service.
4. In Camera lab, connect the RTSP address. Verify completed chunks appear and timestamps advance.
5. Save ten replays at known moments. Open each MP4, check the requested lookback, audio sync and last completed frame.
6. Record a consented match and manually label active intervals. Evaluate separately from the broadcast benchmark.
7. Check model throughput, CPU/GPU load, memory and temperature under simultaneous capture, export and inference.
8. Confirm retention, disk-full handling, camera interruption, restart, duplicate button press and power recovery before unattended operation.

The recorder currently reports a failed capture but does not automatically reconnect an RTSP camera. A Linux supervisor can restart the service, but that also interrupts its inference job. Per-camera reconnect/backoff and manifest recovery remain production work.

9. Acceptance gates before a venue pilot

Gate	Proposed measurable requirement	Current evidence
Continuous capture	8 h with no unexplained gaps over 1 s	Not run on hardware
Replay duration	Requested lookback within 0.25 s, clipped at recording start	Seven-second file simulation passed
Replay freshness	End lag no more than GOP + export latency	Need actual camera measurement
Save reliability	100 presses / 100 acknowledged clips; duplicates create no extra clip	API retry test only
Buffer retention	Old chunks removed; selected export remains valid during pruning	Local simulation
Thermal	No throttling / crashes through target ambient soak	Not measured
Network outage	Reconnect and explicit gap indication	Automatic reconnect not implemented
Storage exhaustion	Controlled stop, visible alarm, no corrupt existing clip	Quota handling not implemented
Tennis AI	Court-disjoint precision/recall and boundary error accepted by operator	Broadcast proxy benchmark only
User experience	Tablet can save, replay and export without cloud access	Desktop/mobile browser test; physical tablet not tested

10. Recommended build sequence

Pilot with an existing computer and one borrowed or returnable RTSP camera. Collect real footage before custom electronics. Validate latency and picture quality, then commission Jetson acceleration if CPU throughput is insufficient. Prototype the button over USB and authenticated LAN requests. Only then design battery firmware, outdoor enclosure, a custom PCB and production fleet updates.

11. Manufacturer and technical references

- Raspberry Pi 5 specifications: <https://www.raspberrypi.com/products/raspberry-pi-5/>
- NVIDIA Jetson Orin Nano Super: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/>
- NVIDIA developer-kit setup: https://docs.nvidia.com/jetson/orin-nano-devkit/user-guide/latest/quick_start.html
- Espressif sleep / wake behavior: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/sleep_modes.html
- FFmpeg segment muxer: <https://ffmpeg.org/ffmpeg-formats.html#segment>
- Ultralytics model documentation: <https://docs.ultralytics.com/models/yolo11/>

Specifications were checked on the research date. Reconfirm exact part compatibility and supplier documentation before procurement.