

智慧體育館：環場多視角情蒐
**Smart Stadium:
Multi-View Surrounding Camera System for Scouting**

指導教授：連震杰

組員：馮品叡

1.1 Introduction: Motivation and Objective

1. Motivation :

- **運動賽事畫面變化快速**：在運動賽事中，場上動作轉換頻繁，單一視角難以完整捕捉比賽過程與關鍵瞬間。
- **多機位影像管理困難**：實際應用中，多台 PTZ 攝影機同時運作時，往往缺乏統一的集中式管理介面，導致操作流程複雜，且容易發生關鍵畫面遺漏的情況。

2. Objective :

- **多機位 PTZ 整合拍攝系統**：整合多台 PTZ 攝影機之即時影像預覽、遠端控制與同步錄影功能，以因應運動賽事拍攝需求。
- **提升賽事影像紀錄品質**：透過穩定的影像串流與同步錄影機制，確保關鍵畫面完整保留，作為後續分析與回放使用。

1.2 Contribution

GIL : Python全域直譯器鎖

1. 多行程併發架構 (Per-Camera Multiprocessing) :

- 每相機獨立 multiprocessing 取代 threading，突破 Python GIL，8 路 1080p120 Parallelism。
- 透過 go2rtc 對每台相機單連線分流，預覽（QSV 硬解）與錄影（-c:v copy）解耦為兩條獨立管線，錄影時可同時保留預覽；UI 與 worker 解耦，錄影 I/O 不會造成UI凍結。

2. 零負載錄影與容錯儲存 (Zero-Overhead Recording & Fault-Tolerant Storage) :

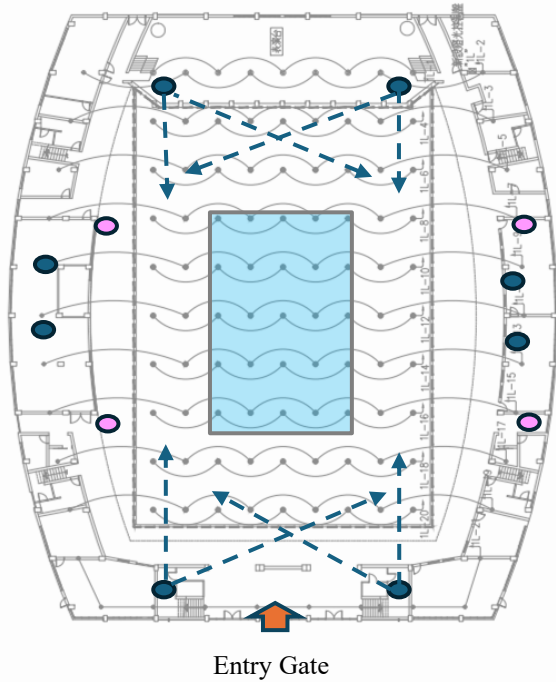
- FFmpeg -c:v copy 直接複製 H.264 封包，CPU 編碼負載趨近零、時間軸保真。錄影先寫本機暫存(TEMP) → 停錄後背景複製到 NAS，遇 NAS 不可用自動轉存桌面，資料零遺失。

3. 同步啟動與同步回放 (Synchronized Capture & Playback):

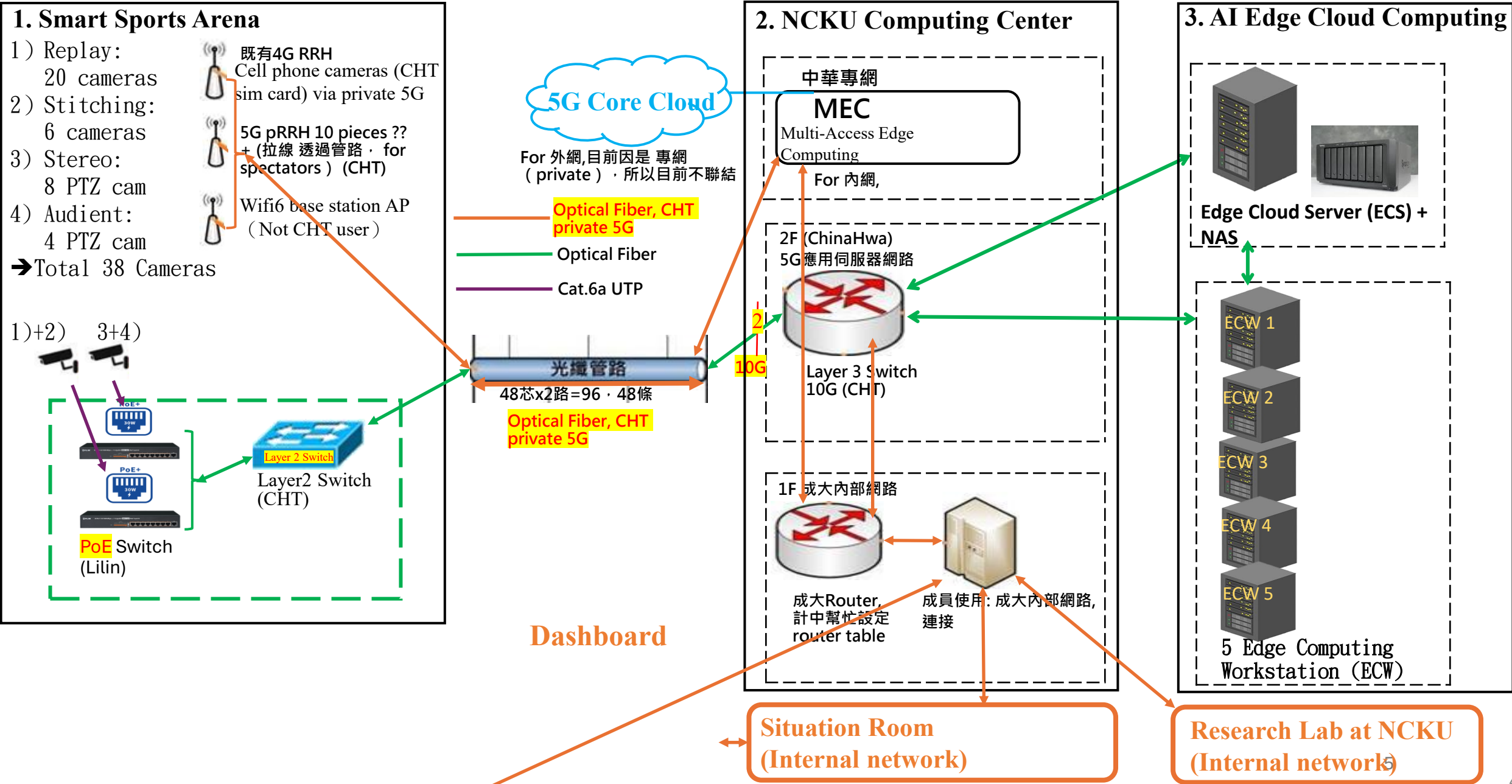
- 8 機共用 start_at + 2s preamble 預熱起錄，go2rtc 單連線會使起錄分散 ~1s，故用 ONVIF SetSynchronizationPoint 送同步 IDR 收斂起錄點，停錄端則透過 stop_at 機制強制同時收尾，回放端再以時長對齊(duration-align) + wall-clock 主時鐘把跨相機誤差壓到 < 1 frame(≈17ms @ 60fps)。

2.1.1 AI Edge Cloud Computing Network: Map

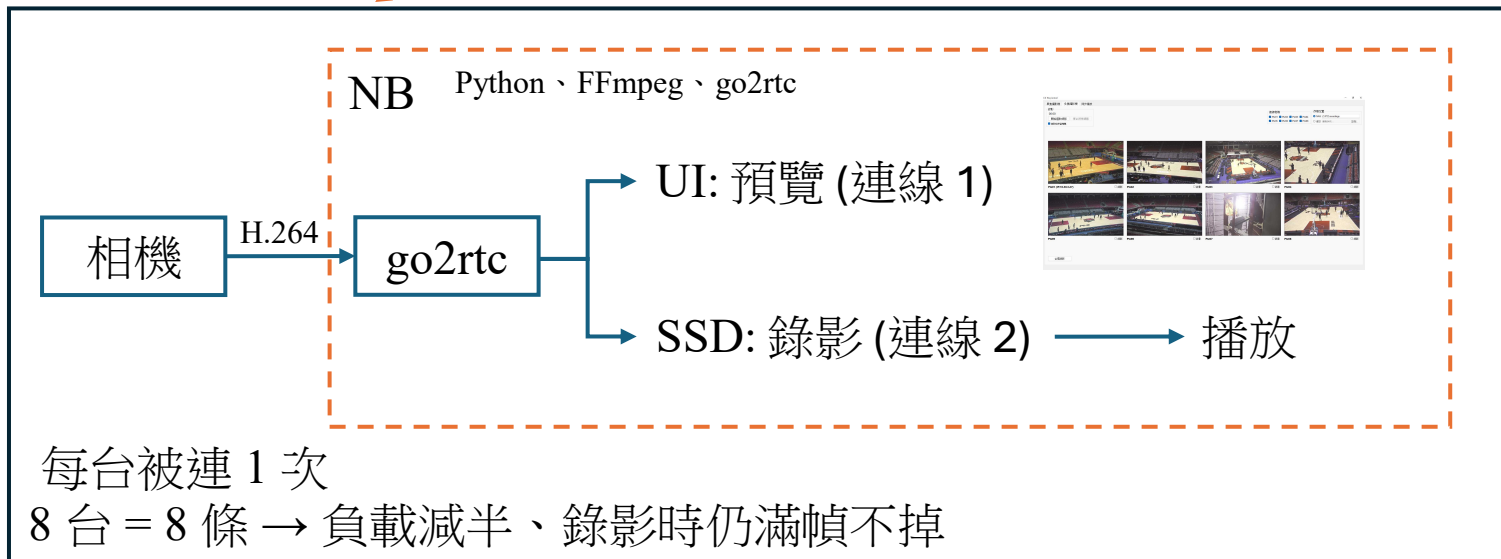
- 8 PTZ cameras: Stereo
- 4 PTZ cameras for interaction with audiences



2.1.2 AI Edge Cloud Computing Platform



x8 camera



Software	用途	說明
FFmpeg (ffmpeg.exe)	錄影、串流解碼、硬體加速	程式碼直接呼叫 ffmpeg.exe，找不到才 fallback用系統 PATH
go2rtc (go2rtc.exe + go2rtc.yaml)	RTSP restream中介（預覽+ 錄影並存）	USE_GO2RTC=True 時必 須有，缺少會無法啟動預 覽

2.2.1 Camera Core Parameter Settings

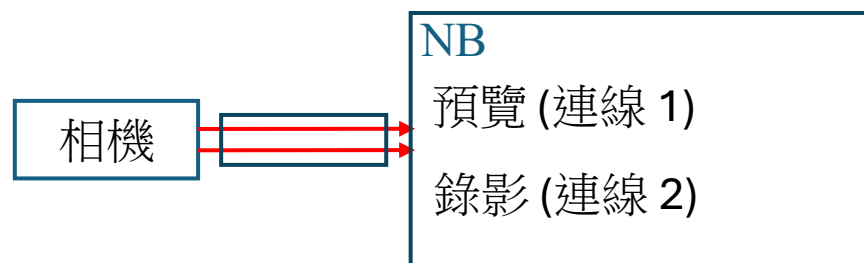
項目	規格
型號	LILIN S7R5724EX30
數量 / IP	8 , 10.30.3.27 ~ 10.30.3.34
解析度	1920 × 1080
影格率 (FPS)	120
視訊編碼	H.264
位元率 (bps)	8M
串流	RTSP (stream0)
傳輸 / port	RTSP over TCP , 554

3.0 GO::go2rtc Restream Architecture

- 採用 go2rtc 建立本機 RTSP 串流代理 (Restream) 機制。
- 若預覽與錄影同時向實體相機要求畫面，相機需承受 2 倍負載 (強制相機同時輸出 2 條串流)，導致設備過載、網路壅塞與嚴重掉幀。
- 透過 go2rtc 對相機僅維持 1 條連線，於電腦本機端無損複製串流給「預覽軌」與「錄影軌」，將 8 台相機連線數從 16 條降回 8 條，達成錄影滿幀不掉。

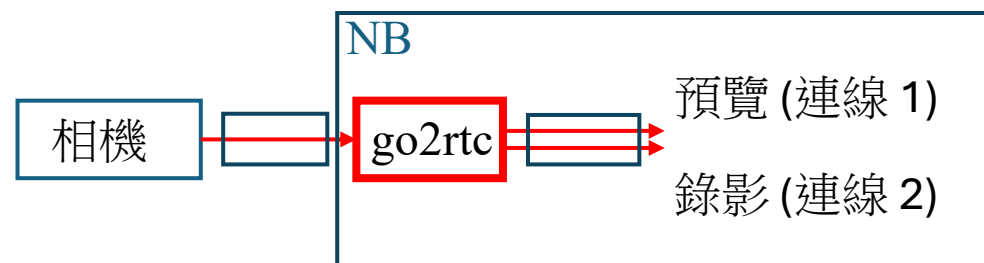
//本專案不使用其 WebRTC 功能，僅取用其本機 RTSP restream 代理能力。

直接連線（無 go2rtc）：



每台camera 被連 2 次
8 台 = 16 條 1080p120 主流 → 相機/網路負載
翻倍、錄影掉幀

透過 go2rtc 連線：



每台被連 1 次
8 台 = 8 條 → 負載減半、錄影時仍滿幀不掉

IN : rtsp://...@10.30.3.27~34 : 554 / stream0

OUT : rtsp://127.0.0.1 : 8554 / Ptz01 ~ 08

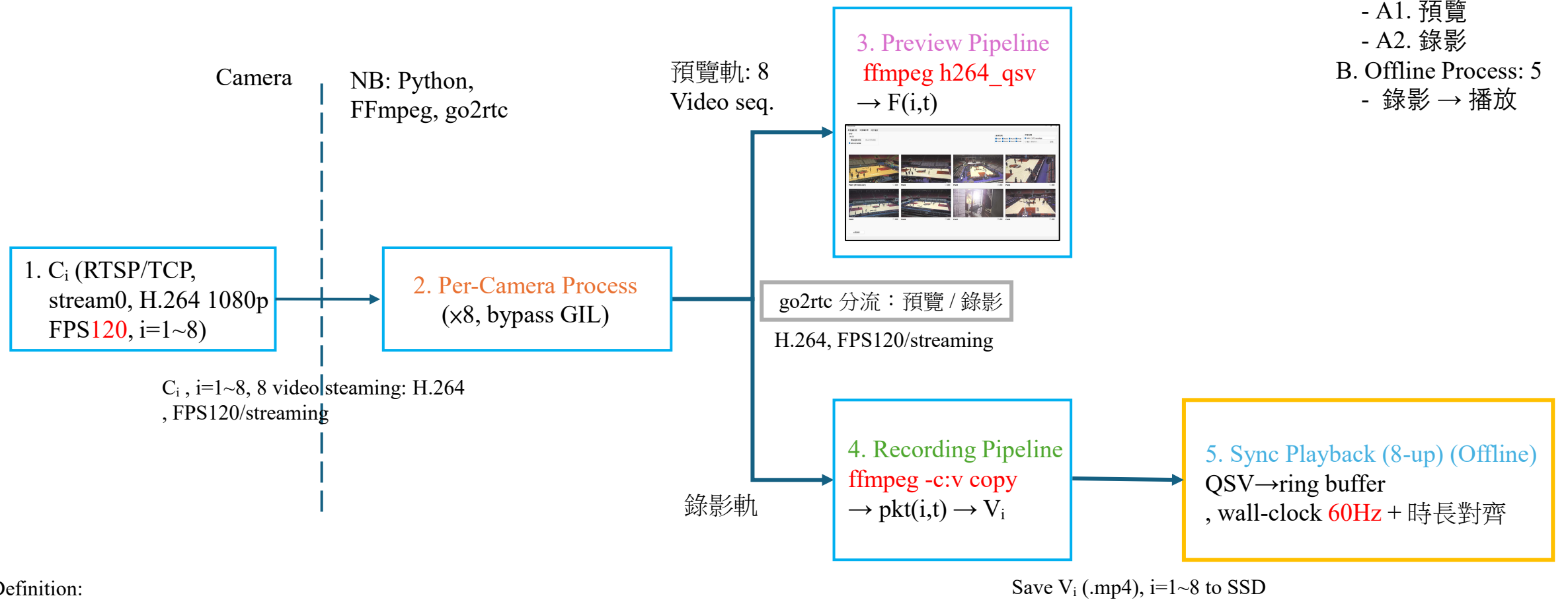
API : http://127.0.0.1 : 1984

← 8 台實體相機

← App 改連這裡

← 除錯用網頁

3.1 go2rtc Global Framework



- A. Online Process: 1~4
- A1. 預覽
 - A2. 錄影
- B. Offline Process: 5
- 錄影 → 播放

Definition:

- C_i = camera_i, i = 1, ..., 8
- F(i,t) = C_i 第 t 幀的 QSV 解碼預覽影像 (預覽軌)
- pkt(i,t) = C_i 第 t 個 H.264 packet (錄影軌, -c:v copy)
- V_i = {pkt(i,1)...pkt(i,T)} = C_i 完整錄影檔 (.mp4), T=sec
- start_at = 8 台同步起錄的 wall-clock 時間戳 (= now + 3s)
- Preamble = 2.0s, ffmpeg 預熱長度 (RTSP handshake + keyframe)

4.1 Data Collection

1. Capture Architecture (擷取架構):

- 1) 每相機獨立 multiprocessing (突破 GIL)
- 2) 預覽軌 → ffmpeg h264_qsv (Intel iGPU 硬體解碼) → 460×258 JPEG(q50) → mp.Queue(12) 有界 FIFO + 30fps 節拍器 → UI
- 3) 錄影軌 → ffmpeg -c:v copy (無重新編碼) → mp4
- 4) go2rtc 單連線後分流：預覽 / 錄影解耦，錄影、預覽同時進行

2. Synchronization (同步):

- 1) 8 機共用 start_at = now + 3s(含 2s preamble)，並用 ONVIF 同步 IDR 收斂起錄；停錄端精準同刻收尾，回放再做時長對齊
- 2) mp.Queue(maxsize=12) 有界 FIFO jitter buffer IPC
- 3) session_name = YYYYMMDD_HHMMSS 統一資料夾

3. Storage (儲存):

- 1) 本機暫存 TEMP\PTZ_staging
- 2) 停錄後背景 thread 複製到 NAS Z:\
- 3) NAS 不可用 → 自動搬桌面 (資料不遺失)

4.2 Experimental Results

1. High Concurrency via Multiprocessing :

- 1) 8 路 1080p120 並行、連續錄影 30 分鐘全程無中斷、無 worker process crash。
- 2) 多行程架構下 UI 主執行緒與錄影解耦，錄影期間介面持續可操作、無凍結 / 當機 (預覽模式下維持 30fps 刷新)。

2. Timeline Accuracy & Multi-Camera Sync :

- 1) 滿幀實測 (ffprobe Verified) : 8 台平均皆為 120fps，實際幀數與理論值 (時長 x 120) 誤差 < 2.5 幀 → 掉幀率 $\approx 0\%$ 。
- 2) 起錄端以 start_at + ONVIF 送同步 IDR 收斂；停錄端透過精準同刻收尾確保影片終點一致，回放端再以 ALIGN_BY_DURATION (時長差 = 起錄差) 做二次 trim 校正。實測 8 台批次觸發延遲 < 30ms，同步回放跨相機誤差 < 1 frame ($\approx 17\text{ms @ } 60\text{fps}$)。

3. Zero-Overhead & Storage Strategy :

- 1) -c:v copy 實測 CPU 編碼負載趨近於零 (無解碼 / 重編碼)。
- 2) 直寫 NAS 因 SMB 逐次寫入受限於網路傳輸延遲 (RTT) 導致寫入瓶頸而掉幀，改採本機暫存 → 背景複製到 NAS 達成滿幀零掉幀，證明 staging 策略的必要性。

5.0 Conclusion and Future Works

1. Conclusion :

- 1) 整合式中控平台：PyQt5 整合 8 路網格預覽、選取錄影、儲存切換、同步回放於單一介面。
- 2) 突破 GIL 之高並行架構：多行程取代 threading，8 路 1080p120fps 並行（go2rtc 單連線分流使預覽 / 錄影解耦）下 UI 持續響應。
- 3) 零負載錄影與時間軸驗證：Stream Copy (-c:v copy) 達 CPU 無重新編碼；ffprobe 驗證 MP4 影片總時長誤差極小，幀數偏差源自相機端位元率控制。

2. Limitations :

- 1) 頻寬與 I/O 依賴：8 路無損原始串流寫入對 Gbps 網路與磁碟 I/O 要求嚴苛。
- 2) 缺乏設備自動探索：採靜態 IP，移至新網段需手動配置。

3. Future Works :

- 1) 整合既有 PTZ 模組 + 自動追蹤: 結合既有 ONVIF AbsoluteMove 模組與 YOLO 物件偵測，使相機自動追蹤偵測。
- 2) 對接姿態估計 (MediaPipe / OpenPose) 邊錄影邊產出 3D 骨架的即時平台

6.1.1 System UI Design – 多機攝影機

■ 多機攝影機

開始 / 停止錄影

控制錄影時是否保持預覽

選取要錄製的相機
全選 / 個別選取

控制PTZ相機連線

選擇存檔位置

連線相機

存檔位置

Ptz01 (IP:10.30.3.27)

Ptz02

Ptz03

Ptz04

Ptz05

Ptz06

Ptz07

Ptz08

☐ 錄影

☐ 錄影

☐ 錄影

☐ 錄影

☐ 錄影

☐ 錄影

☐ 錄影

☐ 錄影

全選錄影

Ptz_Control

單隻攝影機 多隻攝影機 同步播放

錄影
00:00
開始選取錄影 停止所有錄影
☒ 錄影時保留預覽

Ptz01 (IP:10.30.3.27)

☐ 錄影

Ptz02

☐ 錄影

Ptz03

☐ 錄影

Ptz04

☐ 錄影

Ptz05

☐ 錄影

Ptz06

☐ 錄影

Ptz07

☐ 錄影

Ptz08

☐ 錄影

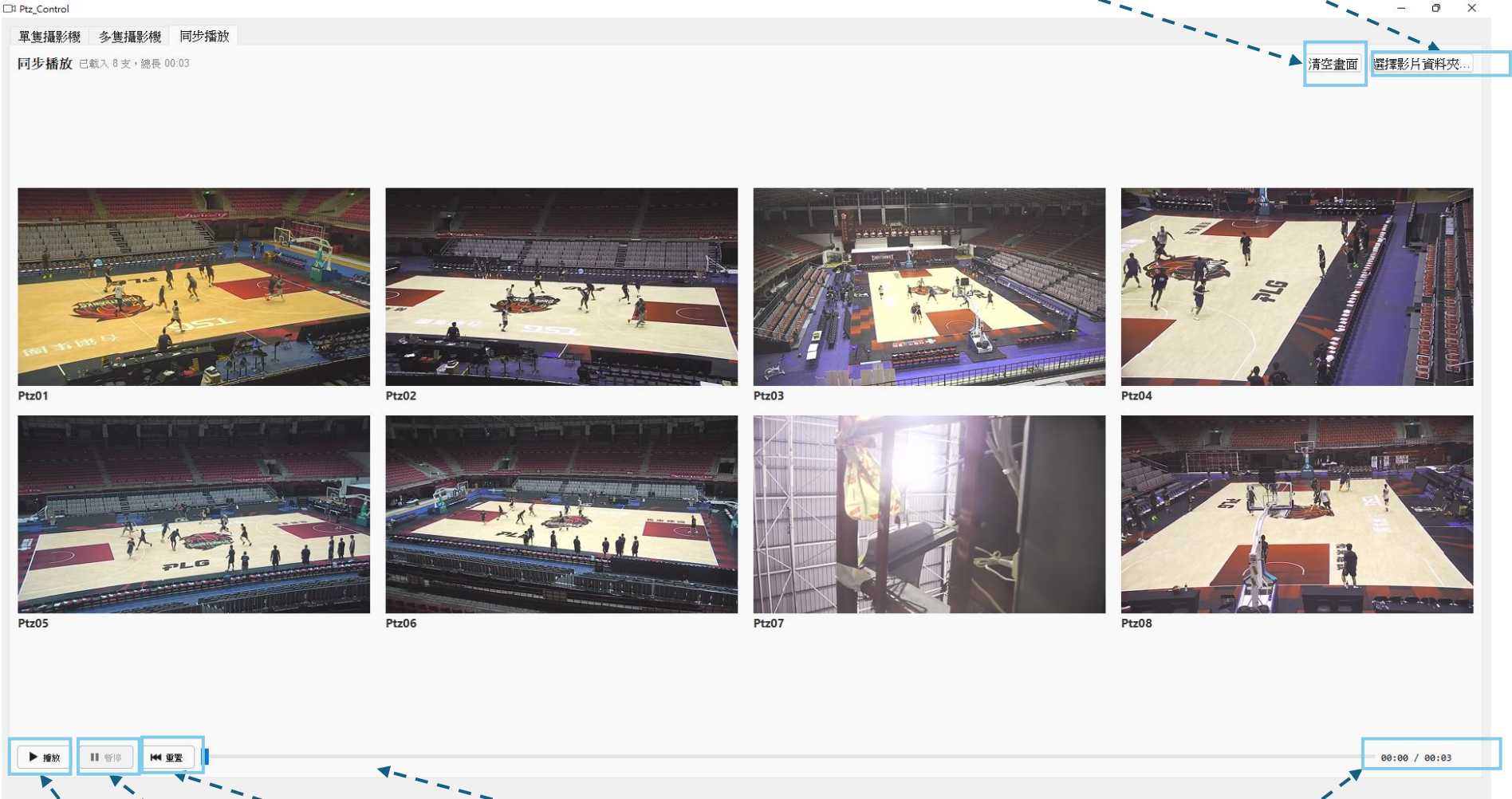
全選錄影

6.1.2 System UI Design – 播放頁面

■ 播放頁面

清空畫面按鈕

選擇播放資料夾按鈕
選擇資料夾中的影片會被載入



播放按鈕

暫停按鈕

重置按鈕
回到影片第一秒

影片時間進度條

時間 / 影片時長

6.1.3 Demo video

