ระบบยืนยันตัวตนด้วยใบหน้า แบบ client–server

ตรวจความมีชีวิต · ตรวจการปลอมแปลง · ลงทะเบียน / ยืนยัน / ระบุตัวบุคคล — เซิร์ฟเวอร์บน Railway

ชื่อระบบ	eKYC Server (FastAPI) + แพ็กเกจ `@ekyc/react-native-ekyc` + แอป `apps/example`
เวอร์ชันเอกสาร	2.0 (server deploy 19 ส.ค. 2569, แอป build 21)
วันที่จัดทำ	19 สิงหาคม 2569
สถานะ	ฉบับส่งมอบเพื่อทดสอบภายใน
เซิร์ฟเวอร์ทดสอบ	https://ekyc-api-production-1c11.up.railway.app (backend onnx, API key ทดสอบฝังในแอป)
แหล่งโค้ด / ไฟล์ติดตั้ง	https://github.com/watcharaponthod-code/ekyc (branch `master`, Releases `server-v1.0.0-b21`)

ตัวเลขทุกค่าในเอกสารเป็นค่าที่วัดได้จริงในเงื่อนไขที่ระบุ ไม่ใช่การรับรองสมรรถนะโดยหน่วยงานภายนอก

1. ระบบทำอะไร

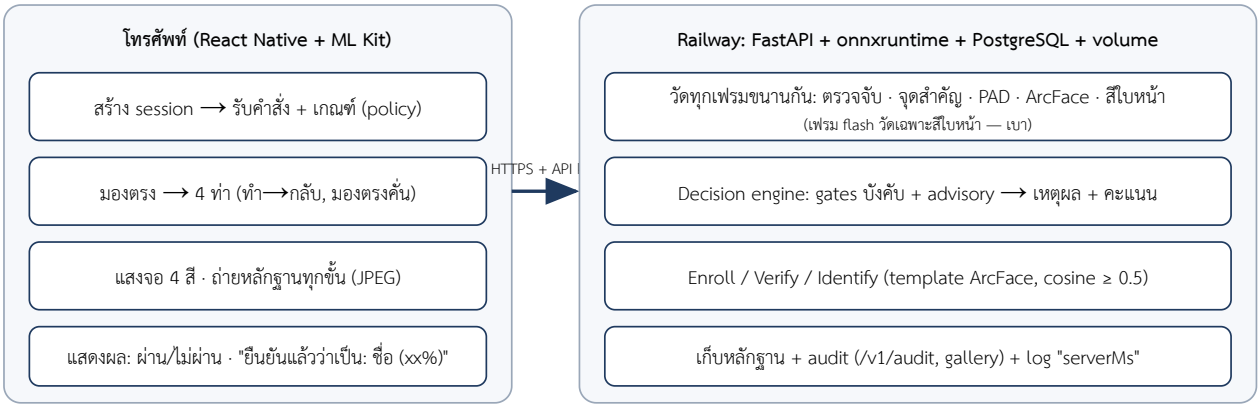
แอปบนโทรศัพท์นำผู้ใช้ทำท่าตามคำสั่ง ถ่ายหลักฐานทุกขั้น แล้วส่งให้เซิร์ฟเวอร์ตัดสินว่า (1) เป็นคนจริงที่ทำท่าตามคำสั่งขณะนั้น (2) ไม่ใช่สื่อปลอม (รูป/จอ/หน้ากาก) (3) เป็นคนเดียวกันตลอดรอบ และ (4) **เป็นใคร** — ลงทะเบียน (enroll), ยืนยันกับบุคคลที่ระบุ (verify) หรือค้นว่าตรงกับใคร (identify) ผลลัพธ์แสดงชื่อบุคคลและคะแนนตรงบนหน้าจอ

ตารางที่ 1 ขอบเขตและสถานะ

ความสามารถ	สถานะบนเซิร์ฟเวอร์ทดสอบ (backend onnx)
คำสั่งท่าทาง: อ้าปาก (ทุกครั้ง) + สุ่ม 3 จาก กระพริบตา · หันซ้าย · หันขวา · ขยับเข้า · ขยับออก	บังคับ — ทุกท่าเป็นการเคลื่อนไหวเทียบท่ามองตรง; เกณฑ์มุมหันส่งจากเซิร์ฟเวอร์ให้แอป (22° + margin 3°)
ตรวจการปลอมแปลง 2 มิติ (PAD) — MiniFASNet บนเฟรมท่าทาง	บังคับ (≥ 0.5) · ค่าที่วัดได้จริงกับคนจริง 0.995+
มุมหัน/ก้ม จากจุดสำคัญ 5 จุด เทียบเฟรมมองตรง	บังคับ · ค่าที่หันได้จริง 52–63°
ความเป็นคนเดียวกันตลอดรอบ (ArcFace ทุกเฟรม)	บังคับ (≥ 0.35) · วัดได้จริง 0.61–0.72
คุณภาพภาพ (ความคม สว่าง ขนาด มุมเอียง)	บังคับ — สาเหตุตกที่พบจริง: `QUALITY_SHARPNESS` ภาพมองตรงเบลอ
ตาหลับ/ลืม (EAR)	**advisory บน onnx** — จุดสำคัญ 5 จุดวัดตาเป็นค่าประมาณ (ตกทั้ง 4 รอบทดสอบทั้งที่กระพริบจริง) ⟶ เปลี่ยนเป็นบันทึกอย่างเดียว; บังคับเมื่อใช้ backend deepface (MediaPipe)
แสงสีจากหน้าจอ 4 สี สะท้อนบนใบหน้า	advisory · วัดได้จริง 0.82–0.92 (เกณฑ์ 0.5) — พร้อมเปิดบังคับเมื่อมีข้อมูลโจมตี
ชีพจรจากสีผิว (rPPG), ระนาบภาพ, ร่องรอยฉีดภาพ, attestation ของเครื่อง	advisory / ตรวจการมีอยู่
อ้าปาก/ยิ้ม วัดจาก blendshape (MediaPipe)	เฉพาะ backend deepface — บน onnx แอปบังคับท่าเอง (contours ML Kit) เซิร์ฟเวอร์ไม่ซ้ำ

หน้ากาก 3 มิติ / ซิลิโคน	ชั้นกัน: อ้าปากบังคับ (หน้ากากแข็ง), PAD, rPPG (advisory), flash (advisory) — **ไม่มีการรับรอง ISO/IEC 30107-3** (ต้องห้องปฏิบัติการที่ได้รับการรับรอง); มีเครื่องมือประเมินแบบเดียวกับมาตรฐาน
เก็บหลักฐานทุก shot	เปิดอยู่ (`EKYC_RETAIN_FRAMES=all`) ลง volume ถาวร `/data/retained` ดูผ่านหน้า gallery

2. สถาปัตยกรรมและขั้นตอน



รูปที่ 1 โครงสร้างระบบและลำดับหนึ่งรอบ

ตารางที่ 2 API หลัก (ทุก endpoint ยกเว้น /health ต้องมี `X-API-Key`)

Endpoint	หน้าที่
`POST /v1/sessions`	สร้างรอบ: purpose (enroll/verify/identify), tier ⟶ คืน sessionId, รายการท่า, จำนวนเฟรม flash, policy (เกณฑ์มุม), nonce
`POST /v1/sessions/{id}/submit`	multipart: เฟรมชื่อ neutral / turnLeft / … / flash_0..3 (+ pulse) ⟶ `decision, reasons, scores, personId, displayName, match, serverMs`
`GET /v1/audit`, `/v1/audit/gallery?key=`, `/v1/audit/{id}/frames`	สรุปรอบล่าสุด (อัตราผ่าน เหตุผล การกระจายคะแนน) · หน้าเว็บ thumbnail ทุกภาพต่อรอบ · ภาพรายรอบ
`GET /health`	สถานะ + backend

2.1 ตั้งค่าเซิร์ฟเวอร์ทดสอบ (ตัวแปรสภาพแวดล้อมบน Railway)

`EKYC_BACKEND=onnx` · `EKYC_API_KEYS=…` · `EKYC_NEUTRAL_YAW_MAX_DEG=45` · `EKYC_FLASH_FRAMES=4` · `EKYC_FLASH_RULE=advisory` · `EKYC_EYE_RULE=advisory` · `EKYC_RETAIN_FRAMES=all` · `EKYC_FRAMES_DIR=/data/retained` · `DATABASE_URL` (PostgreSQL) — โมเดลดาวน์โหลดตอน build image (`scripts/fetch_models.py onnx`) · redeploy: `cd server && railway up --service ekyc-api --detach`

3. เกณฑ์การตัดสิน

ตารางที่ 3 gates และเกณฑ์ (ปรับผ่าน env / `config.py` ; ทุกค่ามีที่มา ไม่ใช่ค่าเดา)

gate	เงื่อนไข	ที่มา / ค่าที่วัดได้จริง
โครงสร้าง	เฟรมครบทุกท่าที่สั่ง 1 ใบหน้าหลักต่อเฟรม nonce ตรง	—
คุณภาพ	ความคม (Laplacian var) ≥ 60, สว่าง 40–220, ใบหน้า ≥ 112 px, roll ≤ 25°	ตกจริง: ภาพมองตรงเบลอ 2/8 รอบ ⟶ คำแนะนำ "ถือนิ่ง"
PAD	ค่าเฉลี่ย MiniFASNet บนเฟรมท่าทาง ≥ 0.5	คนจริง 0.995–0.999
ท่าทาง	Δyaw ≥ 22° (ซ้าย/ขวา) เทียบมองตรง; มองตรง |yaw| ≤ 45°	หันได้จริง 52–63°; เกณฑ์ส่งให้แอป (+3°) จึงไม่ขัดกัน
ตา	EAR ปิด ≤ 0.18 ลืม ≥ 0.22	onnx = advisory (ดูตารางที่ 1)
คนเดียวกัน	cosine ArcFace ทุกเฟรมเทียบมองตรง ≥ 0.35 (รวม pulse frames)	0.61–0.72 (ท่าหันทำให้ต่ำกว่าภาพตรง เป็นปกติ)
แสงจอ	correlation ต่อ channel ≥ 0.5 (advisory)	0.82–0.92
ชีพจร	rPPG prominence ≥ 7 dB (advisory, ต้อง pulse burst)	ยังไม่มีตัวเลขจากเครื่องจริงพอ
Match	cosine กับ template ≥ 0.5	identify จริง 0.84

3.1 ท่าทางฝั่งแอป (ทำไมค้างท่าเดียวไม่ผ่าน)

ทุกท่าเทียบกับ baseline ของคนคนนั้น ท่า 2 ช่วง (กระพริบ อ้าปาก ขยับเข้า/ออก) ถ่ายหลักฐานที่ช่วงแรกและต้องผ่านช่วงกลับ (ลืมตา หุบปาก กลับระยะเดิม) ระหว่างท่าต้องกลับมามองตรง (±12°) และทุกขั้นบันทึก "ทำได้/ต้องการ" ลง telemetry — ชุดทดสอบ `motion.test.ts` ยืนยันว่าไม่มีท่าใดผ่านด้วยท่าค้างหรือรูปนิ่ง

4. ผลการทดสอบอ้างอิง

ตารางที่ 4 การทดสอบอัตโนมัติ

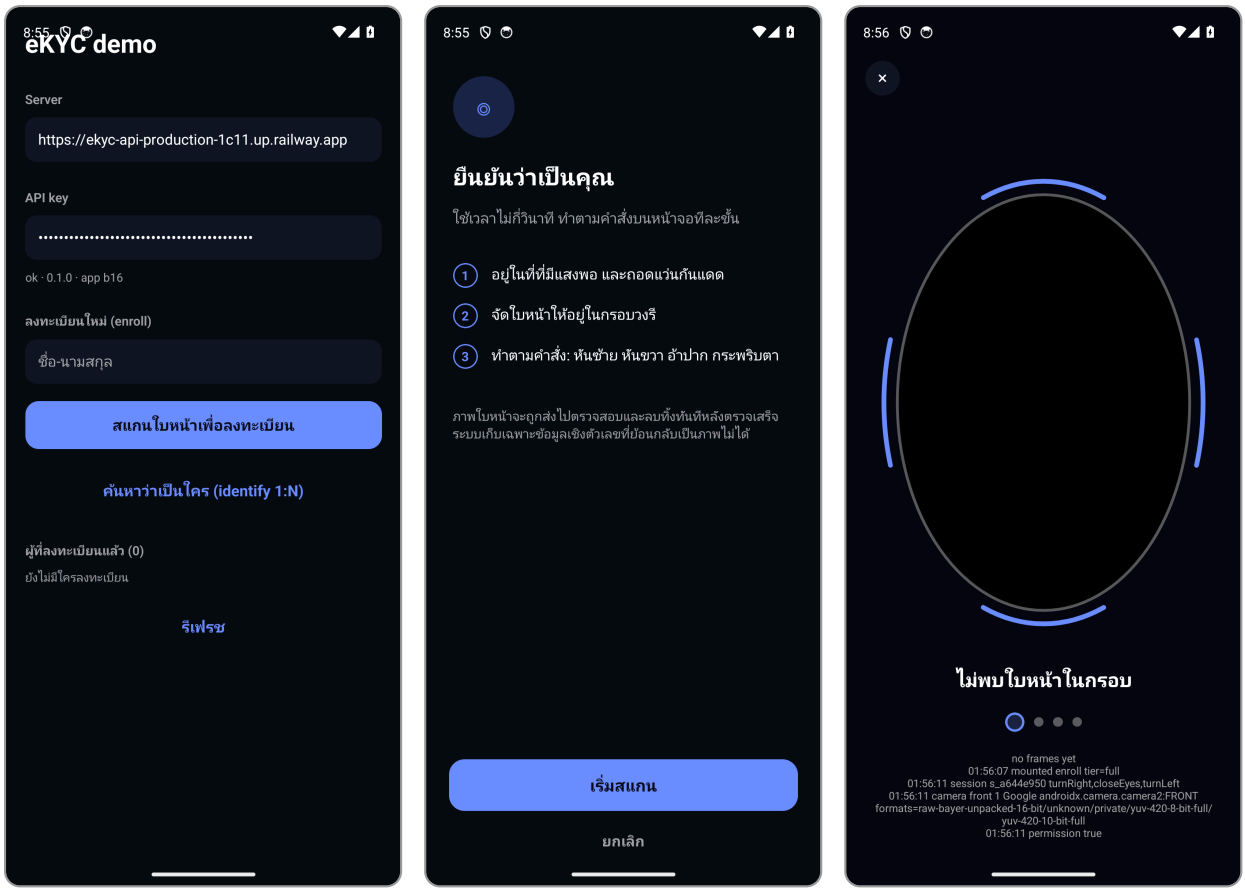
ชุดทดสอบ	ผล	ครอบคลุม
Server (pytest, backend fake)	154 ผ่าน	ทุก gate, นโยบายสุ่มท่า, API key, retention, audit/gallery, flash เบา + วัดขนาน, displayName ใน enroll/verify/identify, serverMs
Core client (jest)	129 ผ่าน	LivenessSession, ทุก Challenge, ท่า 2 ช่วง, recenter, tuning จาก policy, telemetry
PAD harness `scripts/pad_eval.py`	พร้อมใช้	APCER / BPCER / ACER / NRR ต่อชนิดการโจมตี พร้อม Wilson CI — แบบเดียวกับ ISO/IEC 30107-3 (ผลจริงต้องใช้ชุดโจมตีจริง)

4.1 ข้อมูลจากเซิร์ฟเวอร์จริง (Railway, 18–19 ส.ค. 2569)

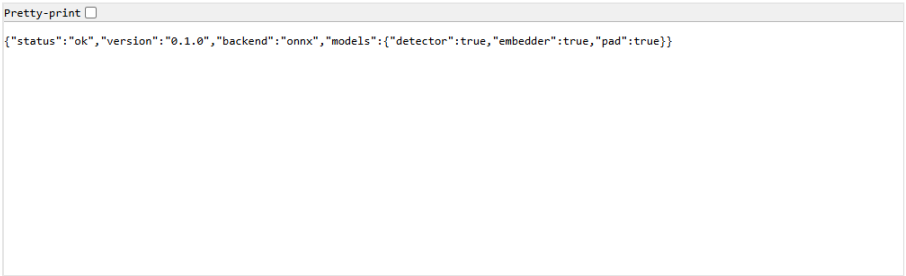
ตารางที่ 5 รอบทดสอบจากโทรศัพท์จริง (จาก `/v1/audit` และ log)

ชุด	รอบ	ผล	ค่าที่วัดได้
build 20 ก่อนแก้ตา	4	ตกทั้งหมดด้วย `EYES_NOT_CLOSED` แม้กระพริบจริง	PAD 0.995+, flash 0.82–0.92, identity 0.61–0.72, หัน 52–63°
หลังตั้ง eye=advisory	4	enroll ผ่าน · identify ผ่าน (match 0.84) · 2 รอบตก `QUALITY_SHARPNESS`	ทุก gate อื่นผ่าน

ความเร็ว: รอบ build 20 รอผล ~10 วินาที สาเหตุ: เฟรม flash 4 ภาพถูกวัดเต็ม (PAD + ArcFace) และวัดทีละเฟรม build 21 + deploy ล่าสุด: เฟรม flash วัดเฉพาะสีใบหน้า, วัดทุกเฟรมขนาน 4 threads, เฟรม flash ส่งเป็น JPEG คุณภาพ 60; เซิร์ฟเวอร์รายงาน `serverMs` ในผลลัพธ์และ log ทุกรอบ เพื่อใช้ตัดสินรอบถัดไปว่าส่วนที่เหลือคือประมวลผลหรือการอัปโหลด (ถ้าเป็นอัปโหลด: ย่อภาพฝั่งแอปก่อนส่ง)



รูปที่ 2 แอปตัวอย่างบน Android Emulator: หน้าแรก (เลือก enroll/verify/identify) / หน้าแนะนำ / หน้าจับภาพ (build 21 ฝัง API key ทดสอบ ไม่ต้องกรอก; แสดงชื่อบุคคลบนหน้าผลลัพธ์)



รูปที่ 3 `/health` ของเซิร์ฟเวอร์บน Railway (backend onnx)

5. การใช้งาน ทดสอบ และตรวจหลักฐาน

1. ติดตั้ง `ekyc-example-v1.0.0-b21-arm64.apk` — เซิร์ฟเวอร์และ API key ทดสอบถูกตั้งไว้แล้ว
2. **Enroll** ใส่ชื่อ ⟶ ทำท่า ⟶ ผ่าน = บันทึก template ของชื่อนั้น
3. **Identify** (ไม่ใส่ชื่อ) ⟶ ผลบอก "ยืนยันแล้วว่าเป็น: ชื่อ (xx%)" หรือ `NO_MATCH` · **Verify** เลือกบุคคลแล้วยืนยัน
4. ดูหลักฐานทุก shot: เปิดเบราว์เซอร์ `https://ekyc-api-production-1c11.up.railway.app/v1/audit/gallery?key=<API key>` — thumbnail ทุกภาพ (neutral / ท่า / flash) ต่อรอบพร้อมผล เหตุผล และคะแนน; สรุปอัตราผ่านที่ `/v1/audit` (header `X-API-Key`)
5. ปรับเกณฑ์จากตัวเลข: `scripts/audit_report.py` รวมการกระจายคะแนนต่อ gate; gate advisory (flash, pulse, eye บน onnx) เปลี่ยนเป็นบังคับเมื่อ p10 ของคนจริงผ่านสบายและมีรอบโจมตีไว้ประเมิน APCER

ตารางที่ 6 กรณีทดสอบแนะนำ

#	กรณี	ผลที่คาดหวัง
1	ทำท่าจริง ถือเครื่องนิ่งตอนมองตรง	ผ่าน; PAD ≥ 0.99, flash ≥ 0.8; enroll/identify แสดงชื่อ
2	รูปถ่ายหรือจอมือถืออีกเครื่อง	ท่าเคลื่อนไหวไม่ผ่านฝั่งแอป; ถ้าหลุดไปถึงเซิร์ฟเวอร์ PAD ต่ำ ⟶ `PAD_SPOOF` ; flash score ต่ำ (บันทึก)
3	คนละคนกลางรอบ (สลับคน)	`IDENTITY_INCONSISTENT`
4	Identify คนที่ไม่เคย enroll	`NO_MATCH` (ไม่มีชื่อ)
5	เรียก API โดยไม่มี key	401
6	หน้ากากกระดาษ/แข็ง	อ้าปากไม่ผ่านฝั่งแอป (หมดเวลา) — บันทึกเป็นรอบโจมตีสำหรับ `pad_eval.py`

ข้อจำกัดและแผนถัดไป

- ยังไม่มีชุดการโจมตีจริงมากพอสำหรับ APCER/BPCER ที่มีนัยสำคัญ; ไม่มีการรับรอง ISO/IEC 30107-3
- onnx backend วัดตาเป็นค่าประมาณ (advisory) และไม่วัดอ้าปาก/ยิ้ม — ต้องการความเข้มสูงสุดให้ใช้ backend deepface (MediaPipe) ซึ่งหนักกว่า
- ความเร็วหลังแก้ต้องยืนยันด้วย `serverMs` จากรอบจริง; โมเดลยังเป็น CPU

- API key ทดสอบฝังในแอป — ใช้เฉพาะทดสอบ; ของจริงต้องออก key ต่อผู้ใช้/แอปและหมุนเวียนได้

ภาคผนวก: รหัสเหตุผลและไฟล์สำคัญ

รหัส / ไฟล์	ความหมาย
`MISSING_FRAME` · `NO_FACE` · `MULTIPLE_FACES` · `FRAME_UNREADABLE`	โครงสร้างหลักฐาน
`QUALITY_SHARPNESS/BRIGHTNESS/FACE_SIZE/ROLL`	คุณภาพภาพ
`PAD_SPOOF` · `POSE_NOT_MET` · `EYES_NOT_CLOSED` · `MOUTH_NOT_OPEN` · `IDENTITY_INCONSISTENT` · `FLASH_SPOOF` · `PULSE_ABSENT` · `NO_MATCH`	gates (บางตัว advisory ตามตารางที่ 1)
`server/app/decision.py` · `config.py` · `services/verification.py` · `main.py`	engine ตัดสิน · เกณฑ์ · วัดเฟรม (flash เบา, ขนาน) · API + displayName + serverMs
`server/app/services/{retention,audit,evidence}.py`	เก็บหลักฐาน · สรุป · gallery
`server/scripts/pad_eval.py` · `audit_report.py`	ประเมิน PAD แบบ ISO · รายงานจาก audit
`packages/react-native-ekyc/src/liveness/*` · `ui/EKYCCamera.tsx` · `client/EKYCClient.ts`	ท่าทาง · หน้าจอ · เรียก API

ผู้จัดทำ	ผู้ตรวจสอบ	ผู้อนุมัติ