GNX HRC

The Blue Book of GNX HRC
설명서·매뉴얼·참고서

운영자·개발자·파트너 어댑터 설계자를 위한 공식 청서

문서 등급	Enterprise Operation Manual & Technical Reference Blue Book
제품명	HRC_Heart-rate Resonance Control / 심박 공명 제어 시스템
버전	v1.0 / HRC Runtime v0.3.1 기준
배포 범위	라이선스 검토, 기술실사, 내부 평가, 운영 인수확인
특정 회사명 정책	공개 문서와 평가 산출물에서 특정 회사명은 의도적으로 배제함

본 청서는 HRC 의 작동 원리, 상태 기계, 입력 소스, Workbench 사용법, Partner Adapter SDK, signed LIVE packet, evidence 검증, 운영 명령어와 장애 대응 절차를 공식적으로 설명한다.

공식 문서 주의: 본 문서는 의료 성능 주장서가 아니라, HRC 런타임의 구조, 검증, 라이선스 검토, 파트너 연동 및 운영 체계를 설명하는 엔터프라이즈 문서이다.

문서 통제 및 배포 기준

문서명	The Blue Book of GNX HRC 설명서·매뉴얼·참고서
적용 엔진	HRC Runtime v0.3.1
핵심 경계	ENGINE_TEST, RECORDED_SESSION, LIVE_DEVICE 의 명시적 분리
라이선스 계층	Tenant license layer 및 HMAC-SHA256 signed LIVE_DEVICE packet 검증
검증 계층	Public evidence index, verifier, acceptance report, partner evaluation kit
대외 배포 원칙	특정 회사명 대신 vehicle_seat_partner, wellness_device_partner 와 같은 일반화 명칭 사용
주의	본 문서는 의학적 효능 확정, 질병 치료 주장, 최종 양산 인증 완료를 의미하지 않음

문서 독립성: 본 문서는 특정 파트너의 내부 설계, 특정 회사명, 특정 제품명에 종속되지 않는다. 모든 적용군은 파트너 어댑터와 프로파일의 문제이며, HRC Core 는 공명 허가 게이트와 증거 런타임을 중심으로 동일하게 유지된다.

목차

1. HRC 사용 설명 개요

1.1 독자와 사용 범위

1.2 시스템 개념

1.3 기본 용어

1.4 전체 흐름

2. Runtime 과 상태 기계

2.1 상태 목록

2.2 위상 오차

2.3 hold 와 ramp

2.4 safety return

3. 입력 소스와 패킷 원칙

3.1 ENGINE_TEST

3.2 RECORDED_SESSION

3.3 LIVE_DEVICE

3.4 signed packet

4. Workbench 사용법

4.1 프로파일 선택

4.2 샘플 선택

4.3 실행과 로그

4.4 evidence 다운로드

5. Partner Adapter SDK 사용법

5.1 API 계약

5.2 adapter registry

5.3 Python SDK

5.4 장치 능력 선언

6. License 와 Tenant 운영

6.1 public verifier

6.2 tenant license

6.3 키 전달 원칙

6.4 권한 오류 해석

7. Evidence 와 검증 페이지

7.1 manifest

7.2 verify

7.3 bundle

7.4 acceptance report

8. 운영 명령어와 배포 점검

8.1 hrcquick

8.2 hrcops

8.3 hrcaccept

8.4 hrceval

9. 장애 대응과 품질 관리

9.1 API 장애

9.2 HTTPS 와 Nginx

9.3 서명 오류

9.4 문서 품질 정책

1. HRC 사용 설명 개요

1.1 독자와 사용 범위

이 청서는 HRC_Heart-rate Resonance Control, 심박 공명 제어 시스템의 작동 원리, 운영 방법, API 사용, 파트너 SDK, 증거 검증, 장애 대응을 설명하는 공식 매뉴얼이다. 백서가 리더와 보안전문가를 위한 라이선스 검토 문서라면, 청서는 운영자, 개발자, 기술실사 담당자, 파트너 어댑터 설계자가 실제로 참고할 수 있는 설명서이다.

청서의 목적은 사용자가 HRC 의 상태 기계와 패킷 원칙을 이해하고, Workbench 를 실행하고, evidence bundle 을 검증하고, signed LIVE_DEVICE packet 을 구성하고, partner evaluation kit 을 사용하는 절차를 명확히 안내하는 것이다. 본 문서는 최종 의료기기 사용설명서가 아니며, 장치 양산 인증 문서도 아니다.

1.2 시스템 개념

HRC 는 공명 허가 엔진이다. 사용자가 특정 장비에서 감각 자극을 받는 상황을 가정할 때, HRC 는 생체 신호 위상과 자극 위상을 비교하고, 일정 조건이 충족되었을 때만 자극 증강을 허가한다. 이때 HRC 가 반환하는 것은 device driver 명령이 아니라 actuation_policy 이다. 파트너 adapter 는 이 정책을 자신들의 하드웨어 언어로 번역한다.

HRC 의 핵심 문장은 “Visuals follow runtime”이다. 화면은 엔진의 결정을 따라가야 하며, 런타임이 반환하지 않은 상태를 시각 효과로 꾸며내지 않는다. RECORDED_SESSION 은 recorded sample 로 표시되고, LIVE_DEVICE 는 signed packet 없이는 실행되지 않는다.

1.3 기본 용어

용어	의미
bio_phase_rad	생체 신호에서 유도된 위상
stimulus_phase_rad	자극 또는 자극 모델의 위상
phase_error	두 위상 차이를 정규화한 값
entry_threshold	동기 진입 판단 기준
exit_threshold	동기 이탈 판단 기준
hold_ms	동기 조건 유지 시간
ramp_ms	L2 허가 전 점진 증가 시간
evidence_hash	해당 decision 이 hash-chain 에 기록된 해시

1.4 전체 흐름

1. 사용자는 target_profile 과 input_source 를 선택한다.
2. 세션이 시작되면 session_id 와 license_scope 가 결정된다.
3. 각 패킷은 /api/session/step 으로 전달된다.
4. HRC 는 state 와 actuation_policy 를 반환한다.
5. 각 decision 은 evidence hash-chain 에 기록된다.
6. 검증자는 manifest, verify, bundle 을 통해 세션을 검토한다.

이 흐름은 Workbench, SDK, Partner Evaluation Kit 에서 동일하게 적용된다. 차이는 입력이 ENGINE_TEST 인지, RECORDED_SESSION 인지, LIVE_DEVICE 인지에 따라 필요한 경계가 다르다는 점뿐이다.

요약문

청서는 HRC 를 실제로 이해하고 운용하기 위한 공식 설명서이다. HRC 는 공명 허가 엔진이며, 화면은 runtime 을 따라가고, 입력 소스는 명시적으로 분리된다. 전체 흐름은 세션 시작, 패킷 입력, decision 반환, evidence 기록, bundle 검증으로 이어진다.

2. Runtime 과 상태 기계

2.1 상태 목록

HRC 상태 기계는 ASYNC, ENTERING_SYNC, SYNC, EXITING, SAFE_RETURN 으로 구성된다. ASYNC 는 위상 오차가 동기 기준 밖에 있거나 hold 조건이 시작되지 않은 상태이다. ENTERING_SYNC 는 phase_error 가 entry_threshold 이내로 들어와 hold_ms 가 누적되는 상태이다. SYNC 는 hold 조건을 만족하고 램핑이 진행 또는 완료된 상태이다. SAFE_RETURN 은 위상 이탈 또는 품질 실패가 발생하여 안전 복귀 정책을 내는 상태이다.

상태	조건	일반 정책
ASYNC	entry band 밖 또는 초기 상태	KEEP_L1
ENTERING_SYNC	entry band 안에서 hold 누적	KEEP_L1
SYNC	hold 충족 후 ramp 또는 L2 허가	RAMPING_TO_L2 또는 L2_ALLOWED
EXITING	동기 band 를 벗어나는 중간 상태	KEEP_L1
SAFE_RETURN	exit threshold 초과 또는 품질 실패	SAFE_RETURN_L1

2.2 위상 오차

phase_error 는 bio_phase_rad 와 stimulus_phase_rad 의 차이를 -pi 에서 pi 범위로 정규화한 값이다. HRC 는 abs_phase_error_rad 가 entry_threshold_rad 이하인지 확인한다. 프로파일별 threshold 는 다르며, 예를 들어 vehicle_seat 와 premium_recliner 는 서로 다른 hold_time 과 safe_return_ms 를 가질 수 있다.

위상 오차는 HRC 의 핵심 판단 입력이지만 유일한 입력은 아니다. signal_quality, actuator_latency_ms, source_type, license boundary, packet integrity 가 모두 함께 판단된다. 따라서 phase_error 가 작더라도 LIVE packet 서명이 유효하지 않으면 HRC 는 해당 패킷을 거절한다.

2.3 hold 와 ramp

hold_ms 는 phase_error 가 entry_threshold 이내로 유지되는 누적 시간이다. hold_target_ms 에 도달하기 전까지는 ENTERING_SYNC 상태이며, 정책은 KEEP_L1 이다. hold 가 충족되면 SYNC 로 전환되고 ramp_ms 가 증가한다. ramp_target_ms 에 도달하면 L2_ALLOWED 가 반환된다.

ramp 는 갑작스러운 자극 증강을 막기 위한 안전 및 사용자 경험 장치이다. 파트너 장비는 HRC 가 반환한 RAMPING_TO_L2 또는 L2_ALLOWED 를 자기 하드웨어의 실제 강도 변화에 매핑해야 한다. HRC 는 물리적 모터 출력을 직접 제어하지 않고 허가 정책을 반환한다.

2.4 safety return

SAFE_RETURN 은 HRC 의 중요한 안정성 기능이다. SYNC 상태에서 abs_phase_error 가 exit_threshold 를 초과하거나 signal_quality 가 FAIL 이면 HRC 는 SAFE_RETURN_L1 을 반환한다. Safety Return Proof 는 이 상태 전이를 evidence 로 보여준다.

운영자는 SAFE_RETURN 을 오류로 보아서는 안 된다. SAFE_RETURN 은 시스템이 위험 신호를 감지하고 더 낮은 허가 상태로 복귀한 정상적인 안전 판단이다. 따라서 dashboard 나 파트너 QA 에서는 safe_return_count 를 부정 지표이면서 동시에 안전 기능 작동 지표로 해석해야 한다.

요약문

HRC 상태 기계는 ASYNC, ENTERING_SYNC, SYNC, EXITING, SAFE_RETURN 으로 구성된다. hold 와 ramp 는 자극 증강 허가를 천천히 안전하게 만드는 장치이며, safety return 은 위상 이탈이나 품질 실패 시 L1 으로 복귀시키는 정상 안전 기능이다.

3. 입력 소스와 패킷 원칙

3.1 ENGINE_TEST

ENGINE_TEST 는 deterministic engineering vector 이다. 엔진 개발 및 공개 동작 확인에 사용되지만 live human data 가 아니다. 따라서 UI 와 문서에서 ENGINE_TEST 를 LIVE 처럼 표시해서는 안 된다. HRC 는 no_unlabelled_simulation 원칙을 갖고 있으며, ENGINE_TEST 벡터를 LIVE_DEVICE 나 RECORDED_SESSION 으로 재사용하지 않는다.

ENGINE_TEST 는 빠른 health check 와 latency benchmark 에 유용하다. 그러나 라이선스 검토에서는 RECORDED_SESSION proof 와 signed LIVE_DEVICE flow 를 함께 확인해야 한다.

3.2 RECORDED_SESSION

RECORDED_SESSION 은 recording_id 와 anonymized=true 를 포함해야 한다. Workbench 에는 Resonance Permission Proof 와 Safety Return Proof 두 샘플이 제공된다. 기록 샘플은 끝까지 소비되면 자동으로 반복되지 않고, 사용자에게 evidence bundle 다운로드를 안내한다. 이것은 기록 데이터가 살아 있는 것처럼 보이는 문제를 차단한다.

RECORDED_SESSION 은 공개 검증에 적합하다. 실제 파트너 live key 를 공개하지 않고도 HRC 의 상태 전이, L2_ALLOWED, SAFE_RETURN_L1, hash-chain verification 을 보여줄 수 있다.

3.3 LIVE_DEVICE

LIVE_DEVICE 는 파트너 장비 또는 파트너 sandbox 가 실시간 또는 준실시간으로 생성한 패킷을 의미한다. HRC v0.3.1 에서는 LIVE_DEVICE 세션 시작에 tenant license key 가 필요하고, 각 패킷에는 HMAC-SHA256 서명이 필요하다. 이 서명은 integrity.signature 를 제외한 전체 packet canonical JSON 에 대해 생성된다.

필수 integrity 필드는 device_id, timestamp, nonce, session_id, signature_alg, signature 이다. session_id 가 현재 HRC 세션과 다르면 live_packet_session_mismatch 가 발생한다. 서명 알고리즘이 다르면 unsupported_live_signature_algorithm 이 발생한다. 서명이 틀리면 live_packet_signature_invalid 가 발생한다.

3.4 signed packet

7. 파트너는 tenant license key 를 별도로 수령한다.
8. 파트너는 /api/session/start 로 LIVE_DEVICE 세션을 시작한다.
9. 반환된 session_id 를 packet.integrity.session_id 에 넣는다.
10. integrity.signature 를 비운 packet 을 canonical JSON 으로 직렬화한다.
11. tenant license key 로 HMAC-SHA256 을 계산한다.
12. 계산한 hex digest 를 integrity.signature 에 넣어 /api/session/step 으로 보낸다.

이 과정은 Python SDK 예제와 signed-live-packet-contract.json 에 명시되어 있다. 공개 evaluation kit 에는 license key 가 포함되지 않는다.

요약문

HRC 입력 소스는 ENGINE_TEST, RECORDED_SESSION, LIVE_DEVICE 로 분리된다. RECORDED_SESSION 은 origin metadata 를 요구하고, LIVE_DEVICE 는 tenant license 와 HMAC-SHA256 signed packet 을 요구한다. 이 분리는 HRC 의 기술적 정직성과 보안 신뢰를 만드는 핵심이다.

4. Workbench 사용법

4.1 프로파일 선택

Workbench 에서 사용자는 target profile 을 선택한다. 프로파일은 HRC Core 를 바꾸지 않고 adapter context 를 설정한다. 예를 들어 vehicle_seat, wellness_bed, massage_chair, body_healthcare_device, premium_recliner 와 같은 프로파일은 max_latency_ms, entry_threshold_rad, exit_threshold_rad, hold_time_ms, ramp_time_ms, safe_return_ms 를 다르게 가진다.

프로파일 선택은 파트너 장비의 폼팩터를 반영하기 위한 것이며, HRC Core 의 본질을 변경하지 않는다. Core 는 항상 phase error, gate, policy, evidence 를 생성한다.

4.2 샘플 선택

RECORDED_SESSION 을 선택하면 Recorded Proof Sample 을 선택할 수 있다. Resonance Permission Proof 는 SYNC 와 L2_ALLOWED 를 보여주고, Safety Return Proof 는 SAFE_RETURN_L1 을 보여준다. 두 샘플은 공개 evidence bundle 과 연결되어 있어, UI 실행 결과와 공개 증거 체계가 같은 원리를 따른다.

운영자는 RECORDED_SESSION 을 반복 재생하는 방식으로 보여주면 안 된다. 현재 UI 는 샘플 끝에 도달하면 자동 실행을 중지하고, evidence bundle 다운로드를 안내한다.

4.3 실행과 로그

Start session 은 /api/session/start 를 호출한다. Step once 는 하나의 packet 을 입력한다. Auto run 은 일정 간격으로 step 을 반복한다. 로그에는 tick, state, actuation_policy, phase_error, signal_quality, packet count 가 표시된다. L2_ALLOWED 는 PERMISSION flag 로, SAFE_RETURN 은 SAFETY flag 로 강조된다.

이 로그는 보조 시각화가 아니라 runtime decision 요약이다. 따라서 로그와 evidence bundle 이 일치해야 한다.

4.4 evidence 다운로드

evidence 버튼은 /api/session/{id}/bundle 을 호출하여 HRC-LICENSE-REVIEW-BUNDLE-v1 JSON 을 다운로드한다. 이 bundle 에는 manifest, verification, events 가 포함된다. 운영자는 bundle 을 보관하거나 파트너 실사 자료로 전달할 수 있다. 다만 개인정보 또는 실제 파트너 데이터가 포함될 경우 별도 보안 경로를 사용해야 한다.

요약문

> Workbench 는 프로파일, 입력 소스, proof sample 을 선택하고 런타임을 직접 호출하는 공개 검증 표면이다. 운영자는 UI 가 runtime decision 을 따라가는지, 샘플이 반복되지 않는지, evidence bundle 이 생성되는지를 확인해야 한다.

5. Partner Adapter SDK 사용법

5.1 API 계약

API 계약은 /api/contract 에서 확인할 수 있다. 계약에는 core endpoints, packet contract, decision contract, live signature contract, license contract 가 포함된다. 공개 SDK 파일 /sdk/api-contract.json 은 파트너 개발자가 오프라인으로 검토할 수 있는 형태이다.

핵심 endpoint 는 /api/session/start, /api/session/step, /api/session/{id}/manifest, /api/session/{id}/verify, /api/session/{id}/bundle, /api/license/verify, /api/benchmark/latency 이다. 파트너는 이 계약을 기준으로 adapter 를 작성해야 한다.

5.2 adapter registry

Adapter registry 는 /api/adapters 에서 확인한다. 각 adapter 는 input_contract, output_contract, partner_boundary 를 가진다. 예를 들어 vehicle_seat adapter 는 signed live packet, driver_or_passenger_mode, vehicle_state, bio_phase, stimulus_phase 를 입력 계약으로 가질 수 있고, actuation permission 과 safe-return policy 를 출력 계약으로 가진다.

adapter registry 는 HRC Core 를 변경하는 문서가 아니다. 파트너 장치가 Core 로 들어오기 위해 어떤 값을 준비해야 하는지 설명하는 계약이다.

5.3 Python SDK

공개 예제에는 hrc_partner_client.py, vehicle-seat-live-demo.py, wellness-device-live-demo.py 가 포함된다. 이 예제들은 tenant license key 를 직접 포함하지 않고 placeholder 를 사용한다. 파트너는 별도로 발급받은 키를 넣어 session/start 와 session/step 을 테스트할 수 있다.

Python SDK 는 양산용 언어를 강제하지 않는다. 파트너는 C, C++, Rust, Java, Kotlin, Swift, embedded C 등 자기 환경에 맞게 같은 packet contract 와 signature contract 를 구현할 수 있다.

5.4 장치 능력 선언

파트너는 adapter 설계 전에 device capability 를 선언해야 한다. 이 선언에는 bio_inputs, stimulus_outputs, feedback_inputs, latency feedback 지원 여부, amplitude ramping 지원 여부, safe return 지원 여부가 포함된다. HRC 는 이러한 능력을 전제로 정책을 반환하고, 실제 하드웨어 구동 책임은 파트너에게 있다.

능력 항목	예시
bio_inputs	PPG, ECG, respiration
stimulus_outputs	vibration, pressure, acoustic, piezo_haptic
feedback_inputs	accelerometer, actuator_phase_model
latency	driver delay and feedback delay
safety	safe return, thermal, contact quality

요약문

Partner Adapter SDK 는 파트너 장치 데이터를 HRC packet contract 로 번역하는 계층이다. API 계약, adapter registry, Python 예제, device capability 선언을 통해 파트너는 HRC Core 를 변경하지 않고 내부 평가를 시작할 수 있다.

6. License 와 Tenant 운영

6.1 public verifier

public verifier 는 공개 검증용 license context 이다. 이 context 는 ENGINE_TEST 와 RECORDED_SESSION 만 허용한다. LIVE_DEVICE 는 허용하지 않는다. 따라서 공개 사이트에서 누구나 live 처럼 보이는 데이터를 실행할 수 없고, recorded proof 와 engine test 만 확인할 수 있다.

이 구조는 공개 신뢰를 만든다. 검증자는 공개 화면에서 충분히 구조를 이해할 수 있지만, 실제 live integration 은 tenant license 를 받은 파트너만 진행할 수 있다.

6.2 tenant license

tenant license 는 특정 평가 범위에 대한 권한이다. tenant 는 allowed_profiles 와 allowed_sources 를 가진다. LIVE_DEVICE 가 포함된 tenant 는 유효한 license key 로 session/start 를 호출해야 한다. HRC 는 key hash 를 tenant store 에서 확인하고 scope 를 부여한다.

tenant key 는 공개 artifact 에 포함하지 않는다. 공개 evaluation kit 은 키 없이도 읽고 검토할 수 있지만 LIVE_DEVICE 실행은 할 수 없다. 이 설계는 영업과 보안을 동시에 만족시킨다.

6.3 키 전달 원칙

키는 NDA, 계약, 별도 보안 채널을 통해 전달한다. 문서에 키를 넣거나 공개 repository 에 넣으면 안 된다. 파트너 내부에서도 키는 secret manager 또는 제한된 secure storage 에 보관해야 한다. 키는 테스트 키와 상용 키를 분리하고, tenant 별 scope 를 최소화해야 한다.

운영 단계에서는 key rotation, revocation, expiration, usage audit 가 필요하다. v0.3.1 은 평가용 tenant layer 를 제공하지만, 상용 운영 단계에서는 key lifecycle 관리 기능을 확장해야 한다.

6.4 권한 오류 해석

오류	의미	대응
license_required	LIVE_DEVICE 에 license key 없음	tenant key 로 session/start 재시도
invalid_license_key	키 hash 불일치	키 확인 또는 재발급
profile_not_allowed	tenant scope 에 profile 없음	계약 범위 조정
source_not_allowed	tenant scope 에 source 없음	source 권한 조정
live_packet_integrity_required	필수 서명 필드 부족	integrity 필드 보완
live_packet_signature_invalid	서명 검증 실패	canonicalization 과 secret 확인

요약문

HRC 는 public verifier 와 tenant license 를 분리한다. 공개 검증은 recorded proof 중심이고, LIVE_DEVICE 는 tenant key 와 signed packet 이 있어야 한다. 키는 공개 kit 에 포함하지 않고 별도 보안 경로로 전달해야 한다.

7. Evidence 와 검증 페이지

7.1 manifest

manifest 는 세션 요약이다. session_id, target_profile, input_source, decision_count, sync_decision_count, safe_return_count, last_event_hash 등이 포함된다. manifest 는 긴 bundle 을 모두 읽기 전에 세션의 성격을 빠르게 파악하는 데 유용하다.

운영자는 manifest 를 파트너 내부 DB 에 저장할 수 있다. 이 경우 user-level 개인식별 정보와 분리하고, technical session summary 로 저장하는 것이 바람직하다.

7.2 verify

verify endpoint 는 hash-chain 의 무결성을 검사한다. 결과가 ok true 이면 각 이벤트의 prev_hash 와 event_hash 가 연쇄적으로 맞는다는 의미이다. 오류가 있다면 errors 배열에 mismatch 가 나타난다. 이 기능은 로그 조작 방지와 실사 신뢰에 중요하다.

7.3 bundle

bundle 은 전체 라이선스 검토 자료이다. manifest, verification, events 가 포함된다. events 에는 SESSION_STARTED 와 HRC_DECISION 들이 기록된다. 각 decision 은 input_packet 과 함께 저장되므로, 검증자는 특정 tick 에서 어떤 입력이 어떤 정책으로 이어졌는지 추적할 수 있다.

파트너 데이터가 포함된 bundle 은 공개 URL 에 올리면 안 된다. 공개 bundle 은 curated RECORDED_SESSION proof artifact 이며, 파트너 live data bundle 은 보안 저장소에 보관해야 한다.

7.4 acceptance report

Acceptance report 는 배포와 증거 상태의 요약이다. public proof count, all proofs verified, sync proof exists, safety return proof exists, bundle hashes match index 같은 항목을 제공한다. 이 report 는 내부 인수확인과 대외 검토 준비 상태를 빠르게 보여준다.

요약문

> HRC evidence 는 manifest, verify, bundle, acceptance report 로 구성된다. 검증 페이지는 evidence index 를 읽어 proof 상태를 보여주고, bundle 은 각 decision 과 packet, hash 를 포함한다. 파트너 live data bundle 은 공개하지 않는 것이 원칙이다.

8. 운영 명령어와 배포 점검

8.1 hrcquick

hrcquick 은 빠른 상태 점검 명령어이다. 서비스 상태, HTTPS root, API health, evidence index 를 짧게 확인한다. 운영자는 문서 수정이나 간단한 설정 변경 후 hrcquick 으로 1 차 확인을 수행할 수 있다.

8.2 hrcops

hrcops 는 전체 운영 점검 명령어이다. nginx 와 hrc-engine 상태, nginx syntax, ports, certificate, canonical redirect, security header count, public pages, API health, evidence index, evidence files, runtime integrity boundary, 최근 로그를 확인한다. 중요한 변경 후에는 hrcops 가 OK 로 끝나는지 확인해야 한다.

8.3 hrcaccept

hrcaccept 는 acceptance page 와 acceptance JSON 을 확인한다. 보고서가 존재하고 proof summary 가 true 인지 확인한다. 라이선스 검토 자료를 전달하기 전에는 hrcaccept 를 실행하여 public acceptance layer 가 최신인지 확인한다.

8.4 hrceval

hrceval 은 partner evaluation kit 점검 명령어이다. runtime version, partner evaluation page, evaluation kit files, evaluation pack summary, SDK archive checksum, latest benchmark, service state 를 확인한다. 파트너에게 SDK 링크를 보내기 전에는 hrceval 이 완료되어야 한다.

명령어	목적
hrcquick	빠른 상태 확인
hrcops	전체 운영 점검
hrcaccept	인수확인 상태 점검
hrceval	파트너 평가 패키지 점검

요약문

운영 명령어는 hrcquick, hrcops, hrcaccept, hrceval 네 가지로 정리된다. 각각 빠른 점검, 전체 점검, 인수확인, 파트너 평가 패키지 확인에 쓰인다. 배포 전후에는 이 명령어들을 통해 공개 상태와 엔진 상태를 확인해야 한다.

9. 장애 대응과 품질 관리

9.1 API 장애

API health 가 실패하면 먼저 hrc-engine service 상태를 확인한다. systemctl is-active hrc-engine 이 active 인지 확인하고, journalctl -u hrc-engine 으로 최근 오류를 본다. /api/health 가 local 에서는 되지만 public 에서 안 되면 Nginx proxy 설정을 확인한다.

session/start 오류는 license scope 또는 target_profile 문제일 수 있다. session/step 오류는 packet source mismatch, integrity field missing, signature invalid, recorded origin missing 일 가능성이 높다.

9.2 HTTPS 와 Nginx

HTTPS 오류는 인증서, Nginx syntax, canonical redirect, security headers 를 확인한다. www 는 apex 로 redirect 되어야 하며, 보안 헤더는 중복 없이 한 번만 나오는 것이 바람직하다. Certbot 인증서는 자동 갱신되지만, dry run 이 일시적으로 실패하더라도 실제 인증서 유효 상태를 먼저 확인해야 한다.

9.3 서명 오류

live_packet_signature_invalid 가 발생하면 세 가지를 확인한다. 첫째, canonicalization 이 ensure_ascii=false, sort_keys=true, compact separators 인지 확인한다. 둘째, integrity.signature 를 제외한 packet 전체를 서명했는지 확인한다. 셋째, session_id 와 license key 가 현재 세션에 맞는지 확인한다.

서명 후 quality_score, phase, timestamp, nonce 중 어느 하나라도 바꾸면 서명이 깨진다. 이는 정상이다. HRC 는 서명 후 변조를 차단해야 한다.

9.4 문서 품질 정책

공개 문서에서는 특정 회사명을 쓰지 않는다. 문서를 수정한 후에는 grep 으로 회사명 노출을 확인한다. Partner target 은 vehicle_seat_partner 와 wellness_device_partner 처럼 일반화된 라벨을 사용한다. 의료 효능 표현, 확정 치료 표현, 과장된 양산 완료 표현도 피해야 한다.

문서와 코드가 바뀌면 SDK archive 와 SHA-256 도 재생성해야 한다. evaluation pack JSON 의 artifact hash 가 실제 파일과 맞는지 확인해야 하며, hrceval 로 최종 확인한다.

요약문

장애 대응은 API, HTTPS, 서명, 문서 품질 네 영역으로 나뉜다. 특히 signed LIVE packet 은 canonicalization 과 session binding 이 중요하다. 공개 문서에서는 특정 회사명과 의료 성능 과장 표현을 배제하고, 변경 후 운영 명령어로 검증해야 한다.

부록 A. 현재 HRC v0.3.1 상태표

항목	상태	의미
Runtime	HRC v0.3.1	파트너 평가용 통합 런타임
License layer	enabled	tenant 기반 LIVE_DEVICE 범위 제어
Partner Adapter SDK	enabled	장치별 입출력 번역 계약
Signed LIVE packet	HMAC-SHA256	서명 없는 패킷 및 변조 패킷 차단
Evidence	hash-chain bundle	manifest, verify, bundle, acceptance report
Public pages	workbench, verifier, license review, operations, acceptance, SDK, integration, partner eval	검증자별 접근 표면
Partner labels	vehicle_seat_partner, wellness_device_partner	특정 회사명 배제 정책 적용

요약문

부록은 HRC v0.3.1 의 현재 구현 상태를 간결하게 정리한다. Runtime, license layer, adapter SDK, signed LIVE packet, evidence, public pages, company-name-neutral partner labels 가 모두 함께 존재해야 파트너 평가 패키지로서 의미를 가진다.

부록 B. 공개 경로 및 산출물 목록

경로	목적
/	HRC Verification Workbench
/verify/	Public Evidence Verifier
/docs/license-review.html	라이선스 검토 설명
/docs/operations.html	운영 점검 설명
/docs/acceptance.html	인수확인 리포트
/docs/sdk.html	SDK 설명
/docs/integration.html	signed LIVE packet 연동 가이드
/docs/partner-eval.html	Partner Evaluation Kit
/evidence/hrc_public_evidence_index.json	공개 evidence index
/evidence/hrc_partner_eval_pack.json	파트너 평가 패키지 JSON
/sdk/hrc_partner_evaluation_kit.tar.gz	파트너 평가 SDK 다운로드

요약문

공개 경로는 검증, 라이선스 검토, 운영, 인수확인, SDK 연동, 파트너 평가로 나뉜다. 각 경로는 특정 회사명을 포함하지 않으며, HRC 의 엔진 상품성을 단계별로 증명하는 표면이다.

부록 C. 용어집

용어	정의
Resonance Permission Engine	공명 조건이 충족될 때만 자극 증강을 허가하는 제어 엔진
Tenant License	파트너 평가 범위를 제한하는 라이선스 키 기반 권한
Signed LIVE Packet	tenant key 로 HMAC-SHA256 서명된 LIVE_DEVICE 입력 패킷
Evidence Bundle	세션 이벤트, decision, packet, verification 을 포함하는 검토용 JSON
SAFE_RETURN_L1	이탈 또는 품질 실패 시 낮은 허가 상태로 복귀시키는 정책
Company-name-neutral	공개 문서에서 특정 회사명을 제거하고 적용군 라벨만 사용하는 정책

요약문

용어집은 HRC 의 공식 표현을 고정하기 위한 장치이다. 모든 문서와 제안서는 같은 용어를 사용해야 하며, 특히 공명 허가 엔진, tenant license, signed LIVE packet, evidence bundle, company-name-neutral 정책을 일관되게 유지해야 한다.