The White Book of GNX WNS

검증·라이선스 계약용 설명서

Enterprise Verification and Licensing Brief

문서 코드	GNX-WB-WNS-2026-04
문서 등급	Restricted / NDA 권고 / 검증 및 운영 참조용
작성 기준	theLogoofGnxcokr.txt 단일 소스 코드, WNS 공식 특허 문서, 우선심사결정서
배제 기준	기존 백서 및 청서 초안, 정합성 미흡 판정 자료, 검증되지 않은 주장
버전	v1.0 / 2026-04-29
권리자 표기	GNX Co., Ltd. / 주식회사 지엔엑스

본 문서는 라이선스 협상, 보안 검토, 운영 인수인계에 사용할 수 있도록 재구성한 공식 설명서입니다. 법률 의견, 특허 등록 확정 문서, 외부 보안 인증서는 아니며, 계약 전에는 별도 법률 검토와 독립 보안 검증을 병행해야 합니다.

목차

목차는 고정 목차입니다. 최종 계약본에서는 승인 페이지, 서명란, 문서번호 체계를 반영해 페이지 번호형 목차로 재발행할 수 있습니다.

1. 문서 목적과 라이선스 검증 범위

1.1 수신자와 의사결정 용도

본 백서는 GNX WNS 를 라이선스 테이블에 올리기 위한 경영진, CISO, 보안 아키텍트, 법무 검토자, 기술 실사 담당자용 설명서이다. 목적은 기술의 독창성을 과장해 판매 문구로 포장하는 것이 아니라, 현 소스 코드가 무엇을 수행하고 어떤 조건에서 검증되어야 하며 계약서상 어떤 권리와 의무로 분해되어야 하는지를 명확히 정리하는 데 있다.

GNX WNS 는 본 문서에서 식별자(Identity), 데이터 처리(Data), 실행 제어(Control)를 하나의 검증 가능한 통제 체계로 결합한 문서상 명칭으로 사용한다. 실제 코드명은 nanognx_engine 이며, 공개 표면은 GNX Logicnoid WNS 의 제품·체험·통제 API 로 노출된다. 검토자는 본 백서를 통해 소스 리뷰 범위, 특허 정합성 범위, 보안 검증 범위, 배포 책임, 감사 증적 요구사항을 한 번에 확인할 수 있다.

1.2 검증 대상과 비대상

검증 대상은 업로드된 단일 소스 코드의 엔진 구조, Express 라우팅, PostgreSQL 및 Redis 의존성, 키 파생 및 세션 처리, Zero-Lodger 로그 및 메모리 소거 정책, WNS 정규화와 Bident Resonance 계산, AIPhone 계열 디스플레이 락 및 터널링 상호 의존 트랜잭션, 운영 헬스체크와 데이터베이스 스키마이다. 라이선스 검증자는 코드가 선언한 보안 속성이 실제 운영 환경에서도 동일하게 충족되는지 확인해야 한다.

비대상은 폐기 지시된 기존 백서와 청서, 외부에서 검증되지 않은 마케팅 주장, 특허 등록 확정 주장, 제 3 자 인증 취득 주장, 모든 공격에 대한 절대 방어 보장이다. 본 백서는 독립 보안감사, 침투 테스트, 법무 검토, 특허 권리범위 의견서를 대체하지 않는다. 다만 계약 협상 전에 검토해야 할 항목을 체계화해 실사 비용과 쟁점 누락을 낮추기 위한 기준 문서로 기능한다.

구분	라이선스 검증 기준	계약 반영 항목
소스 범위	theLogoofGnxcokr.txt 의 monolith engine 구조와 라우트	소스 리뷰 가능 범위, NDA, escrow 여부
운영 범위	AWS EC2, Node.js, PostgreSQL, Redis, SMTP, TLS/쿠키 정책	배포 책임, 운영권한, 장애 책임
보안 범위	KDF, WNS, ZKV, 세션, rate limit, audit, health	보안 보증 한계, 검증 산출물, SLA
권리 범위	WNS 특허 문서 및 AIPhone 관련 언급의 기술적 정합성	실시권 범위, 지역, 기간, 서브라이선스
비범위	기존 폐기 문서, 외부 인증 미확보 주장, 등록 확정 주장	면책, 비보증 조항, 조건부 효력

요약문

본 장은 백서의 용도와 경계를 고정한다. GNX WNS 는 계약상 검증을 위해 식별자, 데이터 처리, 실행 제어를 결합한 문서상 명칭이며, 본 문서는 현 소스와 공식 WNS 자료만을 기준으로 작성된다. 절대 보안, 특허 등록 확정, 외부 인증 완료와 같은 주장은 본 백서의 범위 밖이다.

2. GNX WNS 기술 포지션

2.1 WNS 의 문서상 정의

GNX WNS 의 핵심 포지션은 문자열을 주소나 단순 로그인 식별자로 취급하지 않고, 실행 제어 입력으로 정규화한 뒤 내부 토큰, 상태, 세션, 게이트 판정으로 전환한다는 점이다. WNS 문서의 기술적 골자는 문자열 식별자를 네트워크 위치, 도메인 이름, 외부 레지스트리, 소유권 식별과 독립적으로 처리하고, 정규화·규칙 기반 토큰 생성·상태 결합 판단을 거쳐 실행의 개시 또는 차단을 결정하는 구조이다.

코드 구현은 이 포지션을 보안 런타임으로 확장한다. 입력된 ID 와 비밀번호는 WNS 정규화 파이프라인을 거쳐 구조적 토큰이 되고, MCSS 와 Phantom Token 을 통해 Bident Resonance 증거로 결합된다. 그 결과 브라우저나 API 응답에 원문을 반환하지 않고 receipt, fingerprint, session ledger, gate state, audit signature 와 같은 검증 표면을 제공한다. 이 구조는 기존 IAM, DB 보안, SIEM 을 대체하는 제품이 아니라 고위험 실행 지점 앞단에 붙는 통제 회로로 해석해야 한다.

2.2 WNS, ZKV, Bident Resonance 의 결합

WNS 는 입력 문자열을 정규화하고 파쇄하는 전처리 계층이다. 한글 완성형은 자모 단위로 분해되고, 영문·숫자·허용 기호는 NFKD 정규화와 소문자화 이후 whitelist 를 통과한다. 그 다음 각 입자에는 원래 위치를 반영한 positional entropy 가 결합되며, 정렬 이후에도 서로 다른 입력 구조가 동일하게 보이는 anagram 충돌 위험을 줄이는 방향으로 설계되어 있다.

ZKV 는 저장소에서 원문 ID 와 비밀번호를 직접 다루지 않고, user_id_hash 와 zkv_anchor 를 기준으로 검증하는 계층이다. 등록 시에는 WNS 파쇄 결과, MCSS, anchor salt, PBKDF2 결과를 통해 앵커를 만들고 PostgreSQL 에 저장한다. 인증 시에는 동일한 파생 절차로 재계산한 앵커를 가져온 값과 timingSafeEqual 로 비교하며, 존재하지 않는 사용자의 경우 dummy anchor 를 만들어 사용자 존재 여부가 응답 시간으로 드러나는 위험을 낮춘다.

Bident Resonance 는 MCSS 와 Phantom Token 을 결합해 세션 증거와 실행 허용 표면을 생성하는 계산 계층이다. Phantom Token 은 시간, 메모리 상태, 난수, 클라이언트 엔트로피를 포함하고 HMAC 으로 서명된다. 최종 세션은 HttpOnly, Secure, SameSite=strict 쿠키로 결합되며 raw token 은 JSON 응답으로 반환하지 않는 것이 설계 원칙이다.

기술 구성	역할	라이선스 관점의 의미
WNS Normalizer	문자열 입력을 정규화·파쇄·위치 엔트로피 결합	실행 제어 입력의 차별화된 핵심 기능
ZKV Anchor	원문이 아닌 hash/anchor 기반 검증	개인정보 보관 부담과 인증 검증 책임의 분리
Bident Resonance	MCSS 와 동적 Phantom Token 결합	세션과 게이트를 상태 증거로 계약서에 정의 가능
Zero-Lodger	로그 마스킹, 메모리 소거, 감사 서명	운영 증적과 개인정보 최소화 요구사항의 핵심
AIPhone Interlock	디스플레이 준비 신호와 터널 개방의 상호 의존	통신 세션 라이선스 범위와 별도 모듈화 가능

요약문

본 장은 GNX WNS 의 기술 포지션을 정리한다. 주소 해석이 아니라 실행 제어 입력으로 문자열을 다루고, WNS-ZKV-Bident Resonance 가 입력 처리, 검증, 세션 결합을 담당한다. 라이선스 협상에서는 이 세 계층을 하나의 통합 실시권으로 묶을지, 모듈별 실시권으로 분리할지 결정해야 한다.

3. 핵심 아키텍처와 통제면

3.1 부트, 키, 메모리

엔진은 기동 직후 환경변수를 검증한다. GNX_MASTER_KEY_B64, GNX_PROOF_HMAC_KEY_B64, GNX_COOKIE_SIGN_KEY_B64, GNX_AUDIT_HMAC_KEY_B64 는 Base64 형식과 최소 엔트로피 조건을 통과해야 하며, PostgreSQL 과 Redis URL, SMTP 설정도 사전 검증 대상이다. 일부 hardened 운영 규약과 기존 env 규약을 연결하기 위해 alias normalization bridge 가 구현되어 있으나, 이는 누락 시 보조 값을 채우는 호환 계층이지 비밀값 관리를 대체하는 구조가 아니다.

키 파생은 hardware entropy 와 HKDF-SHA512 를 사용해 목적별 서브키를 만든다. WNS MCSS master KDF, Phantom mutator sign, Bident session auth, ZeroLodger audit log 가 분리되어 있으며, 파생이 끝나면 key vault 를 잠그고 원본 IKM 과 salt 를 소거하도록 설계되어 있다. 메모리 관리 계층은 Buffer 단위의 난수 덮어쓰기, 보수 덮어쓰기, zero fill 3-pass 절차와 객체 deep scrub 을 제공한다.

3.2 실행 제어, 세션, 터널

실행 제어면은 API middleware 와 상태 저장소로 구현된다. sanitizeRequest 는 prototype pollution payload 와 user-agent 부재를 차단한다. evaporateMemoryHook 은 응답 완료 후 req.body 와 gnxContext 의 민감 필드를 deep scrub 하고 GC 를 유도한다. requireBidentSession 은 Authorization Bearer 또는 __Host-gnx_bident 쿠키에서 세션 토큰을 찾아 무결성을 검증하고, 유효하지 않으면 401 로 중단한다.

터널 계층은 Redis Pub/Sub 이벤트 버스, DisplayInterlockController, GNXOperationalStateStore, GNXTunnelTicketIssuer, ZeroTrustTunnelingGateway 로 구성된다. 발신 요청은 즉시 연결되지 않고 target identity hash 에 대한 display ready 신호를 기다린다. 신호가 확인되면 터널 ID 와 ticket proof 를 발행하고, 실패하면 locked-by-design 상태로 차단한다. 이 부분은 라이선스 계약에서 실시간 통신 게이트, 인증된 디스플레이 락, 상호 의존 세션의 범위로 별도 식별될 수 있다.

통제면	주요 구현	검증 질문
기동 통제	EnvironmentValidator, alias bridge, bootstrap	필수 secret 누락 시 fail-closed 되는가
키 통제	HKDF-SHA512 purpose separation, vault lock	키 회전과 KMS 연동 절차가 있는가
메모리 통제	SecureMemoryManager, deep scrub, GC hint	Node.js 런타임 한계가 문서화되었는가
API 통제	sanitizeRequest, Bident session, no-store	라우트별 인증과 캐시 정책이 일관적인가
터널 통제	Redis Pub/Sub, display interlock, ticket issuer	락 실패 시 우회 경로가 없는가
운영 통제	health/live, health/ready, schema guard	준비성 실패 시 트래픽 전환을 막는가

요약문

본 장은 GNX 엔진의 실제 통제면을 라이선스 검증 언어로 재배열한다. 부트, 키, 메모리, API, 터널, 운영 준비성이 각각 별도의 검증 포인트이며, 어느 하나라도 계약상 인수 기준에서 빠지면 엔터프라이즈 검증 문서로서 충분하지 않다.

4. 보안 검증 관점

4.1 보호 자산과 위협 모델

보호 자산은 원문 식별자, 인증 비밀, 세션 토큰, audit signing key, ZKV anchor, tunnel ticket, display interlock state, database schema, operational health surface 이다. 주요 위협은 원문 재구성, 사용자 존재 여부 추론, replay, prototype pollution, raw token 노출, Redis/DB 장애에 따른 fail-open, 로그를 통한 개인정보 유출, 운영자가 준비성 검증 없이 트래픽을 전환하는 오류이다.

소스는 여러 위협에 대해 방어 의도를 갖는다. 예를 들어 ZKV 는 dummy anchor 와 constant-time 비교를 통해 사용자 존재 여부와 byte-by-byte timing leakage 를 줄이려 한다. Attestation 은 nonce timestamp 와 Redis NX/PX 로 재사용 서명을 막으려 한다. AnomalyDefenseMatrix 는 Redis Lua sliding window 로 마이크로 윈도우의 요청 수를 제한한다. 그러나 이러한 방어 의도가 실제로 효과를 내는지는 부하 테스트, 장애 주입, clock skew 테스트, Redis failover 테스트, API fuzzing, memory snapshot 검증으로 확인해야 한다.

4.2 검증 체크리스트

검증 항목	필수 테스트	인수 기준
키/비밀값	Base64, KMS, rotation, env leakage 점검	누락·형식 오류 시 프로세스 기동 실패
WNS 정규화	한글, 영문, 숫자, 기호, 유니코드 혼동 입력	동일 입력은 동일 결과, 구조 변경 입력은 구별
ZKV 인증	존재/비존재 사용자 시간차 측정	실무 허용 편차 안에서 통계적 구별 어려움
세션 쿠키	HttpOnly, Secure, SameSite, no-store 확인	브라우저 JS 에서 raw token 접근 불가
Display lock	timeout, replay, 위조 signature, Redis 장애	락 실패 시 tunnel 미개방
Rate limit	Lua script load 실패, burst 트래픽, 다중 IP	정책 위반 요청 차단 및 로그 남김

감사 로그	전화번호, 이메일, IP, JWT, secret key 패턴	민감 문자열이 평문 로그에 남지 않음
Readiness	PostgreSQL/Redis/KMS/migration table 장애	ready=false 및 배포 차단

검증 산출물은 소스 리뷰 보고서, API 테스트 리포트, 보안 구성표, 데이터 흐름도, 로그 샘플, 위협 모델 매트릭스, 침투 테스트 요약본, 운영 runbook, 장애 복구 절차서로 나뉘어야 한다. 라이선스 계약서에는 각 산출물의 제출 주체, 제출 시점, 수정 기한, 불합격 시 보정 절차, 재검증 비용 부담을 명시해야 한다.

요약문

본 장은 보안 검증의 질문을 명확히 한다. 코드에는 보안 방어 의도가 다수 구현되어 있으나, 계약상 효력은 독립 검증으로 확인된 항목에만 부여해야 한다. 특히 fail-closed, timing leakage, replay 방어, 로그 마스킹, readiness gate 는 라이선스 인수 기준의 중심 항목이다.

5. 라이선스 계약 구조

5.1 실시권 객체와 배포 경계

권장 계약 구조는 핵심 엔진 실시권, 운영 배포 실시권, 검증 표면 사용권, 문서 및 상표 사용권을 분리하는 방식이다. 핵심 엔진 실시권은 WNS 정규화, ZKV anchor, Bident Resonance, Zero-Lodger 감사, AIPhone interlock 구현의 사용 범위를 규정한다. 운영 배포 실시권은 특정 tenant, 리전, 인스턴스, DB schema, Redis namespace, 도메인, API prefix 를 기준으로 한정한다.

소스 제공 범위는 특히 중요하다. 전부 제공, escrow 제공, binary/API 제공, review-only 제공은 서로 다른 위험과 가치를 가진다. 라이선시가 직접 운영한다면 키 관리와 DB 접근 권한을 넘겨야 하므로 책임 분장과 사고 통지 조항이 강화되어야 한다. GNX 가 운영하고 라이선시가 API 를 사용하는 경우에는 uptime, data processing, support escalation, audit access, incident disclosure 가 핵심 조항이 된다.

5.2 인수 기준과 운영 책임

인수 기준은 기술 시연 성공만으로 정하면 안 된다. 등록, 인증, replay rejection, session binding, interdependent gate lock, health ready, audit masking, database migration, 장애 복구까지 end-to-end 시나리오로 구성해야 한다. 각 시나리오는 입력, 전제 조건, API route, 예상 상태, 허용 응답, 실패 조건, 증거 로그, 보정 절차를 포함해야 한다.

운영 책임은 키 회전, 로그 보존, DB 백업, Redis persistence, secret rotation, 패치 적용, 취약점 대응, SLA 보고, 법적 요청 대응으로 나뉜다. Zero-Lodger 라는 명칭이 로그와 메모리 처리 정책을 설명하더라도, 법률상 보존 의무와 감사 의무가 사라지는 것은 아니다. 따라서 원문 최소화 정책과 법정 보존 책임은 별도의 데이터 처리 부속계약으로 조율해야 한다.

계약 조항	권장 명시 내용	주의점
권리 범위	WNS/ZKV/Bident/Interlock 모듈별 실시 범위	특허 등록 전후 조건부 문구 필요
소스 접근	review-only, escrow, delivery, API-only 중 선택	영업비밀과 보안위험 동시 고려
검증 절차	PoC, 보안 테스트, 인수 시험, 재검증	성공 기준을 수치화해야 함
데이터 처리	원문 미반환, 해시/앵커 저장, 로그 마스킹	익명정보 단정 금지
운영 책임	키, DB, Redis, KMS, 백업, 장애 대응	운영 주체별 책임 분리
면책/한계	외부 공격 절대 방어 비보증, 법률 의견 비대체	과장 표현 제거

요약문

본 장은 라이선스 계약의 골격을 제안한다. GNX WNS 는 단일 제품명으로 협상할 수 있지만 계약서는 권리, 소스, 운영, 검증, 데이터 처리, 면책을 분리해야 한다. 특히 소스 제공 범위와 운영 주체가 달라지면 책임과 가격 구조도 달라져야 한다.

6. 심사, 실사, 도입 로드맵

6.1 PoC 와 파일럿

1 단계 PoC 는 라이선시의 실제 개인정보나 운영 계정을 사용하지 않고 synthetic identity 와 isolated tenant 로 진행한다. 목표는 WNS commit, resonance 인증, session binding, display interlock failure, health readiness, audit masking 이 문서화된 대로 작동하는지 확인하는 것이다. PoC 기간에는 엔진 성능 수치보다 처리 경로의 정합성과 보안 경계가 더 중요하다.

2 단계 파일럿은 제한된 사용자 그룹과 제한된 데이터 범위에서 운영 유사 환경을 구성한다. 이때 PostgreSQL migration, Redis TTL, KMS round-trip, secret rotation rehearsal, 장애 주입, 로그 보존, 모니터링 대시보드, 관리자 권한 분리가 포함되어야 한다. 파일럿 종료 조건은 기능 성공률, 보안 테스트 통과율, 운영 장애 대응 시간, 법무 검토 완료, 계약 조항 합의로 정의한다.

6.2 엔터프라이즈 전환

엔터프라이즈 전환은 technical acceptance 와 commercial acceptance 를 분리해 승인한다. technical acceptance 는 소스 리뷰, API 테스트, 보안검증, readiness 검증, 데이터 처리 검토로 구성된다. commercial acceptance 는 실시권 범위, 가격, 유지보수, 지원, 업그레이드, 권리 귀속, 배상 제한, 종료 후 처리로 구성된다.

출시 전에는 운영 중단 계획도 필요하다. Redis 또는 PostgreSQL 장애, KMS key invalid, env misconfiguration, attestation nonce skew, tunnel lock timeout, rate limit false positive, audit sink failure 가 발생할 때 fail-closed 정책을 유지하면서 고객 영향도를 어떻게 통제할지 문서화해야 한다. 이 절차가 없으면 기술적으로 우수한 엔진도 엔터프라이즈 조달 심사에서 불합격할 수 있다.

1. PoC 승인: synthetic tenant, synthetic identity, 원문 미보관 검증, no-store 응답 확인.
2. 파일럿 승인: 제한 사용자, 제한 데이터, health/ready 통과, 장애 주입 완료.
3. 보안 승인: 소스 리뷰, SAST/DAST, API fuzzing, 세션 쿠키 검증, 로그 샘플 검토.
4. 법무 승인: 권리범위, 실시권 범위, 데이터 처리, 비보증, 종료 조항 확정.
5. 운영 승인: runbook, on-call, backup, key rotation, incident notice, SLA 확정.

요약문

본 장은 도입 순서를 제시한다. GNX WNS 는 PoC, 파일럿, 보안 승인, 법무 승인, 운영 승인을 순차적으로 통과해야 라이선스 테이블에서 계약 가능한 엔터프라이즈 자산이 된다. 기술 시연만으로 계약을 마무리하지 않는 것이 핵심이다.

부록. 근거 자료와 사용 제한

본 문서의 기술 서술은 업로드된 진성 엔진 소스 코드(theLogoofGnxcokr.txt)와 WNS 특허 명세서(Kipo WNS(1).pdf), WNS 우선심사결정서(WNS_우선심사결정서.pdf)를 기준으로 새로 작성하였다. 사용자가 폐기 지시한 기존 백서 및 청서 초안은 구조, 문장, 논리, 표현, 주장 모두에서 참고하지 않았다.

우선심사결정서는 해당 출원이 우선심사 대상으로 인정되었다는 절차적 사실을 보여주며, 특허 등록 또는 침해 비침해 판단을 확정하지 않는다. 라이선스 계약 문서에는 등록 여부, 권리범위, 실시권 범위, 소스 공개 범위, 검증 환경의 책임 분장을 별도 조항으로 명시해야 한다.

요약문

본 부록은 문서의 근거와 한계를 명시한다. 본 문서는 공식 검토용 설명서이지만 외부 인증서, 법률 의견서, 특허 등록 확정 통지서를 대체하지 않는다.