The Blue Book of GNX AAL

설명서 · 매뉴얼 · 참고서

목적 결속형 실행 객체 비생성 제어 기반 접근·세션·실행·반출 및 도구 호출 통제 체계

공식 엔터프라이즈 문서 | 2026. 6. 12. 기준 정합본

문서 등급	Enterprise / Official Reference
문서 성격	설명서·매뉴얼·참고서
작성 목적	설계자, 운영자, 보안엔지니어, 플랫폼팀, 감사팀이 GNX AAL을 구현·운영·시험할 수 있도록 구성요소, 상태기계, API, 저장구조, 운영 절차를 정리한 정합본
정합 기준	명세서, 도 1~8, 선행기술진보성조사보고서, 선행기술대비표, 우선심사신청설명서, AAL 기술자료, 운영 PoC 측정자료, 사업행정자료 및 보조자료
핵심 명제	non_birth = issued_objects 실행 가능 네임스페이스 미생성
표현 원칙	전 세계 전수 부존재를 단정하지 않고, 조사 범위와 기준일 하에서 확인된 자료의 단독 또는 결합으로부터 핵심 구성이 개시·시사되지 않는다는 한정형 표현을 사용함

본 정합본의 중심 문장

본 청서의 핵심은 GNX AAL을 운영 가능한 제어 계약으로 설명하는 데 있다. 고위험 행위 요청은 목적 결속형 실행 객체 발생 요청으로 정규화되고, 발급 전제조건이 실패하면 실행 참조값이 생성되지 않으며, 생성된 객체도 verify와 consume을 통과해야만 후단 실행이 가능하다.

GNX AAL

목차

본 목차는 청서의 존재 이유에 맞추어 구현·운영·시험·장애처리·용어 정합성 순서로 구성하였다.

공식 운영 서문

본 청서는 GNX AAL의 작동 원리, 구성 요소, 상태 전이, API 계약, 저장구조, 운영 절차, 장애 처리 및 검증 방식을 설명하는 공식 설명서·매뉴얼·참고서이다. 백서가 리더와 보안전문가에게 검증·라이선싱 관점의 핵심 명제를 제시하는 문서라면, 청서는 설계자, 운영자, 보안엔지니어, 플랫폼팀, 감사팀이 GNX AAL을 어떻게 구현하고 운용하며 시험해야 하는지를 설명한다.

청서의 용어는 구현자가 혼동하기 쉬운 지점을 먼저 분리한다. audit id는 감사 식별자이지 실행 식별자가 아니다. request_fingerprint는 추적과 멱등성 판단의 자료이지 실행권한이 아니다. non_birth_events는 감사 네임스페이스이지 issued_objects 실행 가능 네임스페이스가 아니다. shadow-review는 검토 상태이지 제한 실행권한이 아니다. shadow-executable은 제한 조건이 부여된 실행 가능 객체이므로 검증과 소비를 반드시 통과해야 한다.

문서 성격 및 표현 원칙

본 청서는 구현·운영 기준본이며, 소스 코드, credential, DB 세부 secret 또는 운영 비밀값을 포함하지 않는다. 실제 배포에서는 보안 경계와 비밀정보 보호 정책을 별도 운영 절차로 관리해야 한다.

요약문

청서는 GNX AAL을 구현·운영하는 실무자를 위한 문서이다. 핵심 원칙은 고위험 행위 요청을 실행 객체 발생 요청으로 정규화하고, 발급 전제조건이 실패하면 실행 참조값을 생성하지 않으며, 발급 객체도 verify와 consume을 통과해야 실행된다는 것이다.

제1장 시스템 개요와 구성

1.1 전체 구성

GNX AAL 시스템은 요청 수신부, 주체 확인부, 기준 구조체 참조부, 증거 형성부, 목적 결속부, 범위 해시 산출부, 정책 버전 결합부, 감사 사전 기록부, 객체 발생 결정부, 원자적 객체 발급부, 객체 검증부, 객체 소비 트랜잭션부, 감사 체인 기록부, 후단 강제 연동부, 제한 상태 검토부, 테넌트 격리부, 서명키 관리부, 불변 감사 저장부로 구성된다. 각 구성은 독립 모듈처럼 보일 수 있으나, 실제 보안 효과는 발급 전제조건과 발급 성패의 결합에서 발생한다.

부호·구성	운영 역할
110 요청 수신부	접근·세션·실행·반출·도구 호출을 고위험 행위 요청으로 수신
150 목적 결속부	목적 객체를 action, resource, scope, risk, approval, retention, evidence와 결합
160 범위 해시 산출부	후단 재검증 가능한 normalized_scope hash 산출
170 정책 버전 결합부	활성 정책 버전, 서명, 승인 상태, 배포 epoch, tenant 결합성 검증
180 감사 사전 기록부	PRE_DECISION을 발급 전제조건으로 생성
190 객체 발생 결정부	born, shadow-executable, shadow-review, non_birth 결정

부호·구성	운영 역할
200 원자적 객체 발급부	조건 충족 시 issued_objects 실행 가능 행 생성
210/220 검증·소비부	verify, reserve, lock, commit, finalize로 후단 실행 전 강제
230/280 감사 체인·WORM	audit_events 해시 체인 및 불변 저장

1.2 요청 수신과 정규화

요청 수신부는 고객정보 조회, 세션 개설, 명령 실행, 파일 반출, 외부 API 호출, 도구 호출을 모두 목적 결속형 실행 객체 발생 요청으로 정규화한다. 정규화 결과에는 object_type, subject_ref, tenant_id, action, resource_ref, normalized_scope, purpose_ref, evidence_refs, idempotency_key, request_fingerprint가 포함된다. object_type은 접근 객체, 세션 객체, 실행 객체, 반출 객체, 도구 호출 객체 중 하나 또는 확장 유형이 될 수 있다.

1.3 목적 결속과 증거 형성

목적 객체는 “왜 필요한가”라는 자유 입력 문장이 아니다. 목적 객체는 object_type, action, resource_ref, normalized_scope, risk_class, approval_ref, retention_rule, evidence_validity와 기계적으로 결합되어야 한다. 증거 형성부는 업무 티켓, 사례 식별자, 승인 참조값, 법적 근거, 에이전트 작업 식별자, 변경 요청, 보존 규칙, 서명 증거 등을 검증 가능한 evidence_units로 만든다.

1.4 범위 해시

범위 해시 산출부는 실행 또는 반출 범위를 결정론적으로 정규화한다. 데이터 반출의 경우 자원, 필드 집합, 행 수, 목적지, masking profile, canonical schema version이 포함될 수 있다. 도구 호출의 경우 API 경로, 메서드, 파라미터 집합, tool arguments, 호출 목적, 예상 출력 범위가 포함될 수 있다. 후단 시스템은 actual_scope_descriptor 또는 actual_scope_hash를 제출하여 발급 시 산출된 범위 해시와 비교한다.

> **요약문**
>
> 제1장은 GNX AAL의 전체 구성과 정규화 원칙을 설명하였다. 요청은 반드시 객체 유형, 주체, 테넌트, 목적, 범위, 증거, 멱등키, 지문으로 구조화되어야 하며, 목적과 범위는 후단 재검증 가능한 형태로 결속되어야 한다.

제2장 상태기계와 객체 발생 결정

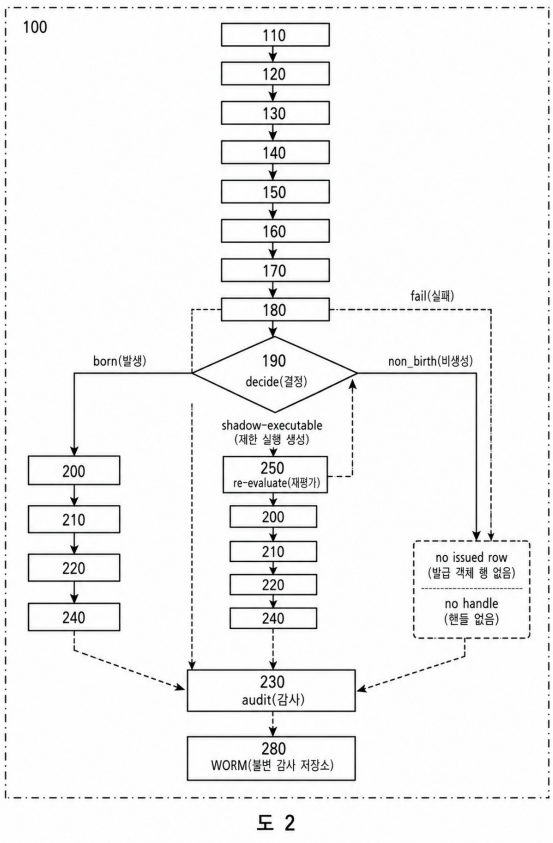


그림 1. 발급 전제절차 이후 born, shadow-executable, shadow-review, non_birth로 분기되는 처리 흐름

2.1 상태기계 개요

객체 발생 결정부는 입력 신호를 평가하여 네 가지 상태 중 하나를 산출한다. born은 정상 발급 상태이다. shadow-executable은 제한 조건이 부여된 실행 가능 객체를 생성하는 상태이다. shadow-review는 실행 가능 객체 없이 검토용 비실행 레코드만 생성하는 상태이다. non_birth는 실행 가능 객체 참조값을 생성하지 않는 상태이다. 이 네 상태는 allow, deny, pending 같은 단순 UI 상태가 아니다.

결정상태	실행 참조값	운영 의미
born	생성 가능	모든 발급 전제조건이 충족된 정상 발급
shadow-executable	제한 조건으로 생성 가능	read-only, masking, delay, co-sign, sandbox, rate limit 등 제한 강제
shadow-review	생성 금지	검토 레코드와 감사 증거만 남기고 재평가 필요
non_birth	생성 금지	실행 가능 네임스페이스에 no row/no handle 상태 형성

2.2 born 흐름

born은 목적 객체, 범위 해시, 정책 버전, 감사 사전기록, 테넌트 맥락, 멱등성, 서명키 준비가 모두 조건을 만족할 때 산출된다. 원자적 객체 발급부는 issued_objects 실행 가능 네임스페이스에 발급 객체 행을 삽입하고 object_id, object_jti, capability 또는 verification_ref를 생성할 수 있다. 단, 생성된 객체도 후단 실행의 자동 허가가 아니다. verify와 consume 절차를 통과해야 실행이 확정된다.

2.3 shadow-executable 흐름

shadow-executable은 불확실하지만 제한 조건 하의 실행이 허용되는 상태이다. 예를 들어 읽기 전용, 마스킹, 지연 실행, 공동서명, 샌드박스, 범위 축소, 1회성 소비, rate limit, 제한 목적지가 적용될 수 있다. 이 상태는 실행 가능 객체를 생성하므로 issued_objects에 행이 존재할 수 있다. 따라서 제한 플래그, 만료, 소비 조건, 범위 해시, 정책 버전, 서명 검증이 후단 verify와 consume에서 강제되어야 한다.

2.4 shadow-review 흐름

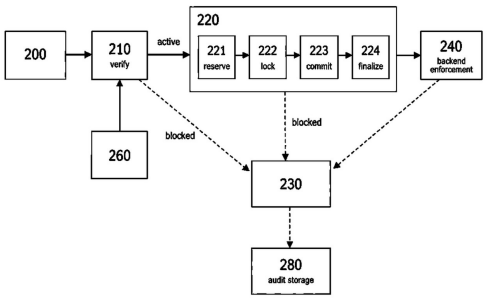
shadow-review는 검토가 필요하지만 실행 가능 객체를 아직 생성하지 않는 상태이다. 검토팀이 확인할 레코드와 감사 증거는 생성될 수 있으나 object_id, object_jti, signed capability, verification_ref는 생성되지 않는다. 운영자가 shadow-review 레코드에 임시 실행 핸들을 넣으면 설계 위반이다. 검토 결과 재평가가 필요하면 새 결정 절차를 거쳐야 한다.

2.5 non_birth 흐름

non_birth에서는 실행 가능 객체가 태어나지 않는다. non_birth_events 또는 audit_events에는 이벤트 식별자, 요청 지문, 사유 분류, 정책 버전, 감사추적 식별자, 증거체인 해시가 남을 수 있다. 그러나 이 값은 후단 실행 입력으로 사용할 수 없다. verify API는 non_birth 감사 식별자를 받더라도 active를 반환하지 않아야 한다.

> **요약문**
>
> 제2장은 born, shadow-executable, shadow-review, non_birth의 차이를 설명하였다. 특히 shadow-review와 non_birth는 실행 참조값을 만들지 않으며, shadow-executable은 제한된 실행 가능 객체로서 검증과 소비 강제를 반드시 필요로 한다.

제3장 API 계약과 후단 실행



도 4

그림 2. verify 이후 reserve, lock, commit, finalize로 이어지는 소비 트랜잭션 및 blocked audit 흐름

3.1 /decide API

/decide API는 객체 발생 결정의 입구이다. 입력은 object_type, subject_ref, tenant_id, action, resource_ref, normalized_scope, purpose_ref, evidence_refs, idempotency_key, request_fingerprint를 포함한다. 출력은 decision_state, audit_trace_id, policy_version_id, proof_chain_hash, reason_class를 포함한다. decision_state가 born 또는 shadow_executable이고 발급 전제절차와 발급 완료절차가 성공한 경우에만 object_id, object_jti, capability 또는 verification_ref가 포함될 수 있다. non_birth와 shadow_review에서는 이러한 실행 참조값이 없어야 한다.

3.2 /verify API

/verify API는 후단 실행 시스템이 실행 전에 제출한 object_id, object_jti, capability 또는 verification_ref를 검증한다. 검증은 행 존재, 서명 유효성, 테넌트 일치, 범위 해시 일치, 만료 여부, 폐기 여부, 소비 여부, 정책 버전, 제한 플래그를 확인한다. active는 issued_objects 실행 가능 네임스페이스의 실제 행과 모든 검증 조건이 일치할 때에만 반환된다.

3.3 /consume API

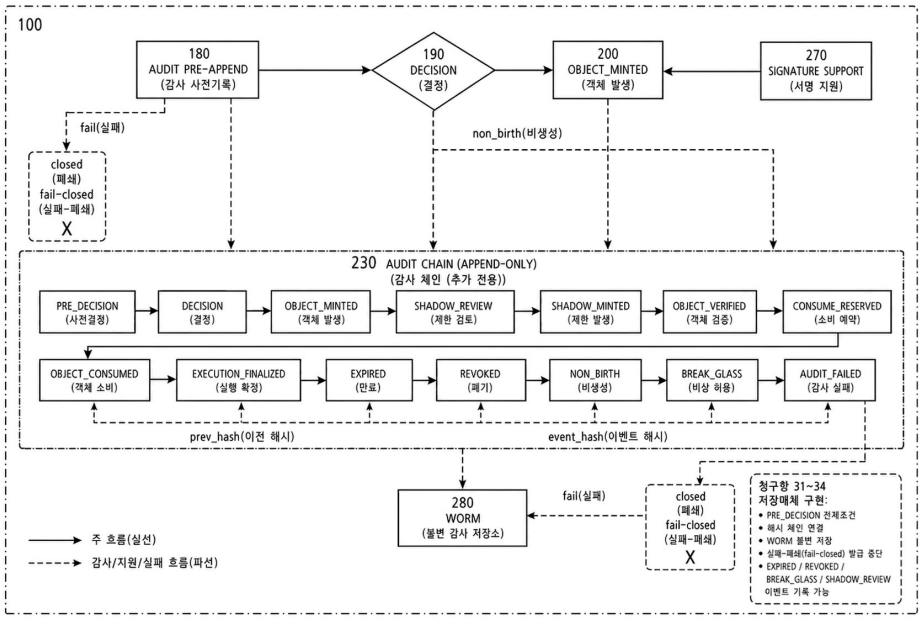
/consume API는 1회성 또는 제한 사용 객체의 소비를 원자적으로 확정한다. 소비 트랜잭션은 reserve, lock, commit, finalize 단계로 구성될 수 있다. 구현 방법은 row-lock, compare-and-swap, conditional write, lease fencing token 등으로 달라질 수 있으나 핵심은 동일하다. consumed_at, execution_nonce, execution_result, execution_commit_hash 또는 감사 추가기록이 커밋된 경우에만 후단 실행 또는 실행 완료를 인정한다.

3.4 /audit API

/audit API는 PRE_DECISION, DECISION, OBJECT_MINTED, SHADOW_REVIEW, SHADOW_MINTED, OBJECT_VERIFIED, CONSUME_RESERVED, OBJECT_CONSUMED, EXECUTION_FINALIZED, EXPIRED, REVOKED, NON_BIRTH, BREAK_GLASS, AUDIT_FAILED 이벤트를 audit_events와 WORM 저장부에 기록한다. 각 이벤트는 prev_hash와 event_hash로 연결되어 감사 체인을 형성한다. 정책상 필수 감사 단계가 실패하면 신규 born 및 shadow-executable 발급은 fail-closed로 중단되어야 한다.

> **요약문**
>
> 제3장은 /decide, /verify, /consume, /audit의 기본 계약을 제시하였다. non_birth와 shadow-review 응답에는 실행 참조값이 없어야 하며, active 응답은 issued_objects 행 존재와 범위·테넌트·정책·제한 조건이 모두 일치할 때에만 허용된다.

제4장 저장구조와 감사 체인



도 7

그림 3. 감사 이벤트 추가기록, prev_hash/event_hash 연결, WORM 저장 및 fail-closed 구조

4.1 저장구조 원칙

GNX AAL의 저장구조는 보안 모델의 일부이다. issued_objects는 실행 가능 네임스페이스이다. non_birth_events는 비실행 감사 네임스페이스이다. audit_events는 추가기록 전용 감사 체인이다. policy_versions는 활성 정책, 승인 상태, 배포 상태, 단조 증가 활성화 순서, restriction_flags, usage_policy를 보유한다. purpose_objects는 검증 가능한 목적 구조체를 보유한다. evidence_units는 목적과 결부된 증거를 보유한다. idempotency_keys는 중복 요청과 재시도를 제어한다.

4.2 issued_objects

issued_objects에는 born 또는 shadow-executable 상태에서만 실행 가능한 객체 행이 삽입된다. 주요 필드는 object_id, object_jti, tenant_id, object_type, subject_ref, purpose_ref, scope_hash, policy_version_id, restriction_flags, issued_at, expires_at, revoked_at, consumed_at, status, verification_ref_hash, capability_key_ref, audit_trace_id가 될 수 있다. non_birth 또는 shadow-review의 레코드가 이 테이블에 실행 가능 상태로 들어가면 설계 위반이다.

4.3 non_birth_events

non_birth_events에는 감사와 설명을 위한 비실행 정보만 저장한다. 예시는 non_birth_event_id, tenant_id, request_fingerprint, reason_class, policy_version_id, audit_trace_id, evidence_chain_hash, created_at이다. object_id, object_jti, signed capability, verification_ref, session handle, export handle, file creation handle, tool invocation id는 저장하지 않는다. 임시 예약행이 필요하면 별도 non_executable namespace에 두고 verify 입력으로 해석되지 않게 해야 한다.

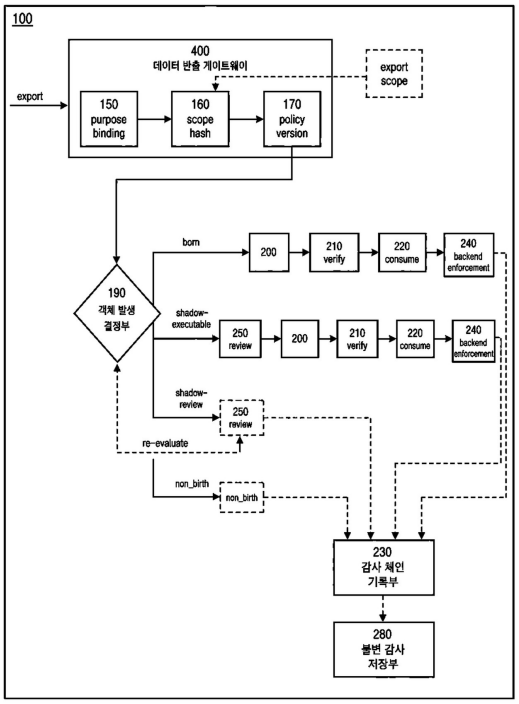
4.4 audit_events와 WORM

저장구조	역할	주의사항
issued_objects	born 또는 shadow-executable 실행 가능 객체 행	non_birth 또는 shadow-review 실행 행 금지
non_birth_events	비생성 감사·추적 정보	object_id, capability, verification_ref 저장 금지
audit_events	PRE_DECISION부터 AUDIT_FAILED까지 감사 체인	prev_hash/event_hash 연결 검증
policy_versions	활성 정책과 restriction_flags 관리	배포 상태·epoch·서명 확인
purpose_objects	검증 가능한 목적 구조체	free-text 단독 목적 금지
evidence_units	승인·티켓·법적 근거·증거 유효성	purpose와 결속 필요
idempotency_keys	중복 발급·재시도 제어	실행 참조값으로 사용 금지

요약문

제4장은 저장구조가 보안 모델 그 자체임을 설명하였다. issued_objects와 non_birth_events의 네임스페이스 분리, audit_events의 해시 체인 및 WORM 저장, policy_versions와 purpose_objects의 결합이 운영 불변식을 형성한다.

제5장 대표 운영 시나리오



도 6

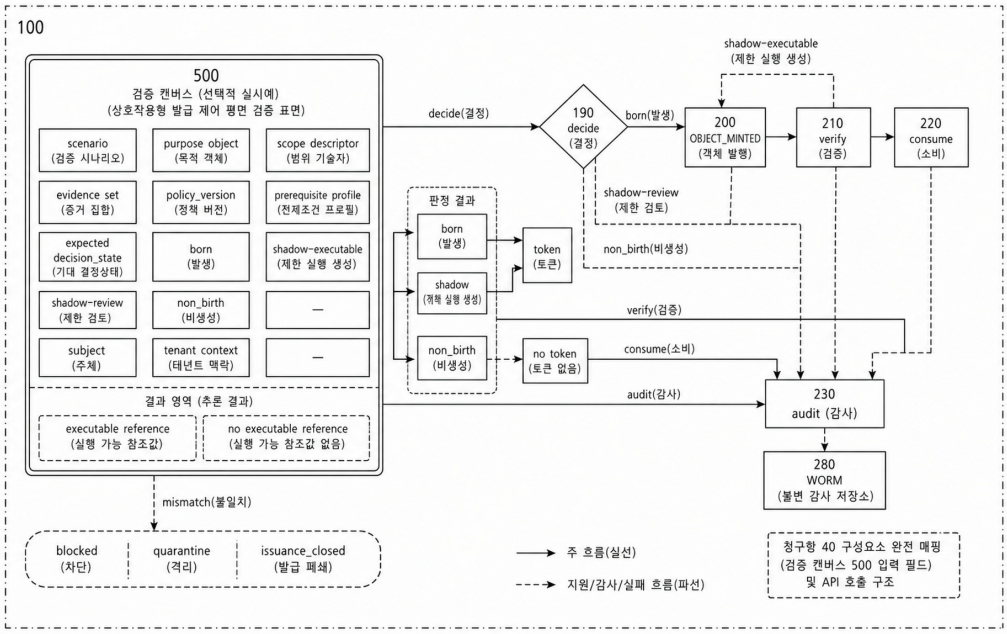
그림 4. 데이터 반출 게이트웨이에서 export scope, purpose binding, policy version을 결합하는 흐름

5.1 데이터 반출 운영

데이터 반출 운영에서 AAL은 export 요청을 실행 객체 발생 요청으로 정규화한다. purpose binding은 반출 목적과 승인 참조를 결합하고, scope hash는 필드 집합, 행 수, 목적지, masking profile, schema version을 결정론적으로 표현한다. policy version은 반출 정책과 배포 epoch를 고정한다. born이면 제한된 export handle이 생성될 수 있고, shadow-executable이면 마스킹·범위 축소·지연 실행이 적용된다. non_birth이면 file creation handle, download URL, external transfer handle, BI extract reference가 생성되지 않는다.

5.2 AI 에이전트 도구 호출 운영

AI 에이전트 도구 호출 운영에서는 prompt/tool context가 AAL Gateway로 들어오고, tool call이 독립 실행 객체 발생 요청으로 변환된다. 외부 API, 결제, 파일 쓰기, 데이터베이스 쿼리, 배포 도구 호출은 모두 tool invocation id 또는 execution handle의 발생을 요구한다. AAL은 purpose object, scope descriptor, evidence set, policy version, tenant context, pre-audit을 결합해 born, shadow-executable, shadow-review 또는 non_birth를 산출한다.



도 8

그림 5. 검증 캔버스: 시나리오, 목적 객체, 범위 기술자, evidence set, policy version과 판정 결과

5.3 특권 세션과 관리자 예외

특권 세션과 관리자 예외는 AAL의 우회 경로가 아니라 AAL의 입력 조건이다. break-glass는 별도 감사 이벤트와 정책 버전에 결합되어야 하며, 예외 승인 후에도 verify와 consume이 우회되어서는 안 된다. 운영자는 break-glass를 실행 객체의 자동 출생 허가가 아니라 엄격히 감사되는 결정 신호로 취급해야 한다.

5.4 배치·서비스 계정 운영

서비스 계정, 배치 작업, BI 추출, 자동화 job은 사람의 UI를 거치지 않기 때문에 AAL의 적용 필요성이 더 크다. 배치 job id, extraction reference, scheduled execution handle, service token은 모두 실행 가능 객체 참조값이 될 수 있다. AAL은 이러한 참조값을 동일한 목적 결속형 실행 객체 발생 모델로 통제한다.

> **요약문**
>
> 제5장은 데이터 반출, AI 도구 호출, 특권 세션, 배치·서비스 계정 운영을 설명하였다. 공통 원칙은 사람이든 에이전트든 서비스 계정이든 후단 실행 참조값이 태어나는 시점에는 동일한 발급 전제조건과 후단 검증이 적용되어야 한다는 것이다.

제6장 배포·장애·관측성

6.1 배포 기준

배포 기준은 API 서버의 응답 여부만으로 판단하지 않는다. AAL 배포는 정책 버전 활성화, purpose_objects/evidence_units 스키마 적용, issued_objects와 non_birth_events의 네임스페이스 분리, audit_events 해시 체인 초기화, WORM 저장 연결, RLS 또는 tenant isolation 적용, verify/consume 후단 연동을 함께 확인해야 한다.

6.2 테넌트 격리

테넌트 격리는 subject_ref 필터링 수준이 아니라 저장구조와 검증 단계 전반에서 강제되어야 한다. tenant_id는 request normalization, purpose object, policy version, issued_objects, non_birth_events, audit_events, verify 응답, consume 트랜잭션에 모두 결합되어야 한다. RLS가 사용되는 경우 force_rls 또는 동등한 정책 강제가 운영 기준에 포함되어야 한다.

6.3 장애 처리

장애 유형	운영 기준
정책 저장소 장애	신규 born/shadow-executable 발급 중단, 기존 검증은 정책에 따라 제한
감사 사전기록 장애	fail-closed, incident 또는 fallback audit 채널 기록
서명키 장애	capability 외부 반환 금지, 필요한 경우 non_executable 상태
DB 부분 커밋 의심	issued_objects 행 rollback 또는 active 불가 상태 전환
WORM 저장 장애	감사 필수 정책이면 신규 발급 중단
후단 verify 장애	backend execution fail-closed, 임의 실행 금지

6.4 관측성과 대시보드

관측성 지표는 단순 요청 수가 아니라 decision_state 분포, non_birth reason_class, born/shadow-executable 발급 수, verify active 비율, consume finalize 비율, replay rejected 수, scope_mismatch 수, tenant mismatch 수, audit chain validity, WORM write success, fail-closed count를 포함해야 한다. 운영 대시보드는 non_birth를 실패로만 보지 말고 정상 보안 결정상태로 분리해야 한다.

요약문

제6장은 배포, 테넌트 격리, 장애 처리, 관측성 기준을 제시하였다. AAL 운영의 기본 원칙은 의심 상황에서 실행 참조값을 임의로 생성하거나 후단 실행을 허용하지 않는 fail-closed이다.

제7장 시험·검증·운영 점검

7.1 단위 시험

단위 시험은 각 구성부의 정상 동작뿐 아니라 실패 상태를 검증해야 한다. purpose normalization 실패, scope hash empty, policy inactive, evidence expired, pre-audit failure, key signing failure, idempotency collision, tenant mismatch를 각각 독립 케이스로 만들고, object_id 또는 object_jti가 응답·DB·로그·캐시에 남지 않는지 확인한다.

7.2 통합 시험

통합 시험은 /decide -〉 /verify -〉 /consume -〉 backend enforcement -〉 /audit의 전체 경로를 검증한다. born 정상 경로에서는 minted, verified, consume_reserved, object_consumed, execution_finalized가 audit chain에 연결되어야 한다. replay 요청은 rejected 또는 already_consumed로 처리되어야 하며, 신규 실행은 허용되지 않아야 한다.

7.3 침투·우회 시험

- non_birth audit id를 object_id처럼 제출하여 verify active가 반환되는지 시험한다.
- request_fingerprint 또는 idempotency_key를 execution handle처럼 재사용해 후단 실행을 시도한다.
- shadow-review 레코드에 임시 handle을 주입하여 후단이 실행하는지 시험한다.
- 범위 해시를 축소 발급 후 확대 실행 범위로 바꾸어 scope_mismatch가 발생하는지 시험한다.
- 테넌트 ID를 바꾸어 issued_objects 행을 조회하거나 verify하도록 시도한다.
- 감사 저장 장애 상황에서 시스템이 fail-open하지 않는지 확인한다.

7.4 운영 점검 체크리스트

점검 항목	기준
정책 버전	active, approved, deployed, signed, tenant-bound 상태 확인
감사 체인	prev_hash/event_hash 연속성 및 WORM write success 확인
non_birth	reason_class별 통계와 no object/no handle 불변식 확인
verify	active, inactive, non_executable, expired, revoked, consumed, scope_mismatch 분포 확인
consume	reserve/lock/commit/finalize 및 replay rejected 확인
후단 강제	verify 실패 상태에서 backend execution 금지 확인

요약문

제7장은 단위 시험, 통합 시험, 침투·우회 시험, 운영 점검 기준을 정리하였다. 테스트의 핵심은 정상 발급보다 실패 케이스에서 실행 참조값이 생성되지 않는지 확인하는 것이다.

제8장 운영 원칙과 참고 용어

8.1 공식 운영 원칙

- 실행 가능 객체의 존재 여부는 UI가 아니라 저장구조와 후단 검증 계약으로 결정한다.
- 감사 식별자, 요청 지문, 멱등키는 실행 참조값이 아니다.
- non_birth와 shadow-review는 실행 참조값을 생성하지 않는다.
- shadow-executable은 제한 실행 객체이므로 verify와 consume에서 제한을 강제한다.
- 감사 필수 경로가 실패하면 신규 born 및 shadow-executable 발급은 fail-closed한다.
- 모든 고위험 행위는 사람, 에이전트, 서비스 계정, 외부 시스템 여부와 무관하게 동일한 객체 발생 모델로 정규화한다.

8.2 구현자가 피해야 할 반패턴

반패턴	문제점
non_birth인데 object_id를 내부에 먼저 만들고 숨김	존재하지만 전달되지 않음에 불과하여 AAL 핵심 위반
shadow-review에 임시 실행 핸들 부여	검토 상태와 제한 실행 상태 혼동
audit_trace_id를 verify 입력으로 허용	감사 네임스페이스와 실행 네임스페이스 혼합
프론트엔드 deny만으로 후단 실행 차단 대체	직접 API, batch, cache, DB 경로 우회 가능
감사 장애 시 임시 발급 허용	fail-closed 원칙 위반
토큰 active 검증만으로 non_birth 대체	이미 존재하는 토큰을 검증하는 post-birth 계열

8.3 참고 용어

용어	정의
purpose-bound execution object request	고위험 행위 요청을 실행 객체 발생 여부 판단 단위로 정규화한 요청
non_birth	실행 가능 네임스페이스에 후단 실행 참조값이 생성되지 않는 결정상태
issued_objects	born 또는 shadow-executable의 실행 가능 객체 행이 존재하는 네임스페이스
non_birth_events	비생성 결정의 감사·추적 정보를 저장하는 비실행 네임스페이스
scope_hash	정규화 범위를 후단 재검증 가능하게 표현한 해시
pre-audit	객체 발급 전에 생성되어 결정상태 산출의 필수 전제조건이 되는 감사 기록
consume transaction	reserve, lock, commit, finalize를 통해 1회성 또는 제한 실행을 확정하는 원자 트랜잭션

8.4 최종 운영 결론

AAL 운영자는 non_birth를 장애나 거절 화면으로만 취급해서는 안 된다. non_birth는 목적·범위·정책·증거·감사 전제조건이 충족되지 않았을 때 실행 참조값의 출생을 금지하는 정상 보안 결정상태이다. 운영의 성패는 이 상태가 API, 저장구조, 감사 체인, 후단 실행에서 일관되게 유지되는가로 판단한다.

> **요약문**
>
> 제8장은 운영 원칙, 반패턴, 참고 용어를 정리하였다. 핵심은 감사·추적 값과 실행 참조값을 엄격히 분리하고, 실행 가능 네임스페이스의 행 존재 여부를 보안 판단의 중심으로 삼는 것이다.

제9장 참조 구현·조직·변경관리

9.1 참조 구현 계층

참조 구현은 gateway layer, policy layer, purpose/evidence layer, issuance layer, verification layer, consumption layer, audit/WORM layer, backend enforcement adapter로 구분할 수 있다. 각 계층은 독립 배포될 수 있지만, security invariant는 계층 간 계약으로 유지되어야 한다. 특히 issuance layer는 policy layer와 audit layer의 성공 신호 없이는 issued_objects 행을 생성하지 않아야 한다.

9.2 이벤트 명명 규칙

이벤트	의미
PRE_DECISION	객체 발생 결정 전 감사 사전기록
DECISION	born, shadow-executable, shadow-review, non_birth 결정 기록
OBJECT_MINTED	born 실행 객체 발급 기록
SHADOW_REVIEW	비실행 검토 상태 기록
SHADOW_MINTED	제한 실행 객체 발급 기록
OBJECT_VERIFIED	후단 실행 전 검증 기록
CONSUME_RESERVED	소비 예약 기록
OBJECT_CONSUMED	소비 확정 기록
EXECUTION_FINALIZED	실행 확정 기록
NON_BIRTH	실행 참조값 미생성 결정 기록
BREAK_GLASS	비상 허용 또는 관리자 예외 기록
AUDIT_FAILED	감사 실패 및 fail-closed 근거 기록

9.3 운영자 역할 분리

정책 승인자, 운영 배포자, 감사 관리자, 키 관리자, 보안 검증자, 후단 시스템 소유자는 역할이 분리되어야 한다. 한 사용자가 정책 활성화, 키 교체, 감사 체인 수정, 후단 adapter 우회를 모두 수행할 수 있으면 AAL의 통제 효과가 약화된다. break-glass 권한도 별도 승인과 사후 감사 검토가 필요하다.

9.4 변경관리

정책 버전, 목적 객체 스키마, evidence schema, scope canonicalization, verify response schema, consume state machine, audit event schema가 변경되면 기존 객체와의 호환성 및 재검증 가능성을 검토해야 한다. 정책 버전은 단조 증가 활성화 순서, 서명, 승인자, 적용 테넌트, 배포 epoch를 기록해야 한다. 변경 중에는 이전 정책과 신규 정책이 혼합되어 실행 참조값을 잘못 active로 판단하지 않도록 해야 한다.

9.5 최종 참조 원칙

최종 참조 원칙

GNX AAL의 구현은 “정책 판단이 있는가”가 아니라 “정책 판단, 목적 객체, 범위 해시, 정책 버전, 감사 사전기록이 실행 가능 객체의 생성 여부와 같은 트랜잭션 경계에서 결합되는가”로 평가한다. 조건 실패 시 남는 것은 실행 가능한 객체가 아니라 비실행 감사 사실이어야 한다.

요약문

제9장은 참조 구현 계층, 이벤트 명명 규칙, 역할 분리, 변경관리 원칙을 제시하였다. 최종 운영 기준은 조건 실패 시 실행 참조값을 만들지 않는 것과 생성된 객체도 verify/consume을 통과해야만 실행되도록 하는 것이다.