

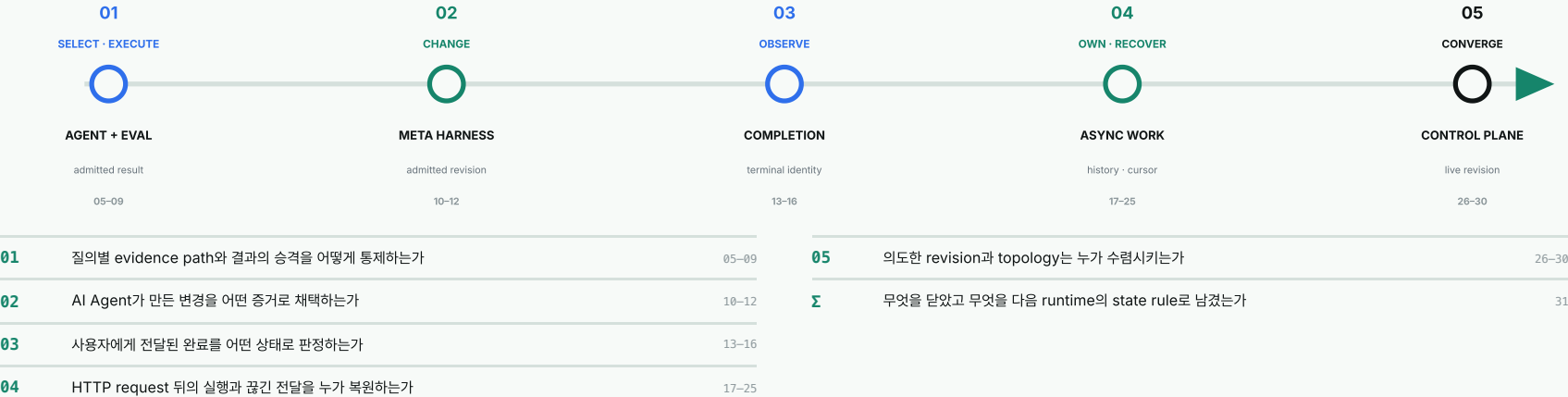
Eco²

WEB / MOBILE · AGENT WORKFLOW · AWS / KUBERNETES

목차

다섯 장은 LLM의 판단을 실행하고, 변경하고, 관측하고, 복구한 뒤 platform state로 수렴시키는 하나의 설계 경로입니다.

DECISION SPINE · EACH SECTION HANDS ONE UNRESOLVED OBSERVATION TO THE NEXT OWNER



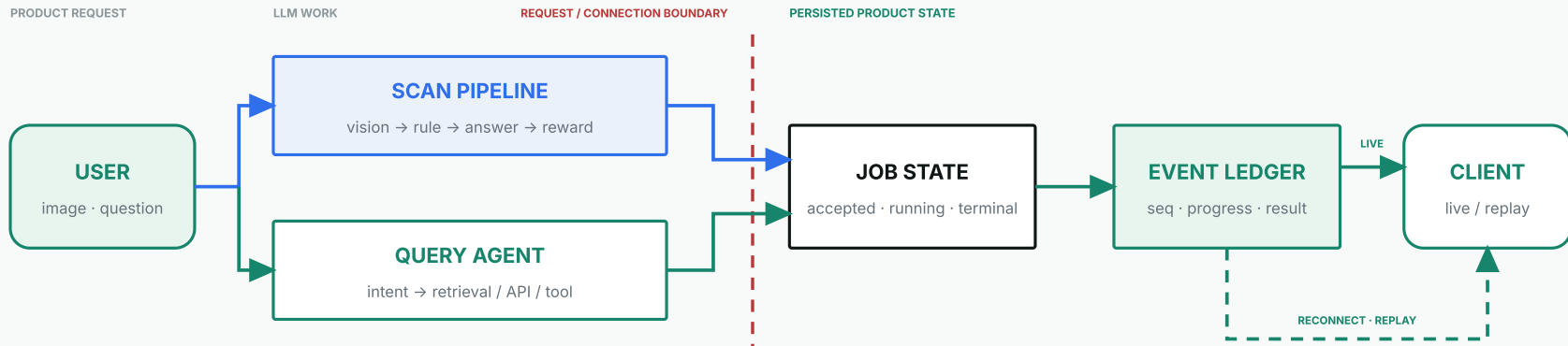
EVIDENCE CONTRACT

실행·검증 artifact
code/test · metric/log/trace · terminal ledger

승격 권한
human/CI keep-revise-revert · 미측정은 OPEN

긴 LLM 작업을 끝까지 전달하고자 실행·이벤트·복구를 제품 상태로 설계했습니다

Scan은 단단계 실행이 요청 수명을 넘고, 환경 질의는 intent에 따라 retrieval·API·tool 경로가 같았으며, SSE 단절 뒤에는 완료 결과를 다시 적용해야 했습니다. 접수·실행·진행·종료·복원을 같은 job lifecycle에 기록했습니다.



PRODUCT COMPLETION

accepted → running → ordered event → client-applied terminal; 연결이 먼저 끝나면 같은 event ledger에서 이어 받습니다.

문제 · 판정 질문

요청이나 SSE 연결이 실행보다 먼저 끝날 때, 어떤 상태를 남겨야 사용자가 마지막 결과를 다시 받을 수 있는가?

근거 · TEST / RUN ARTIFACT

[Public product portfolio](#) · [Scan application](#) · [Chat worker graph](#) · [Frontend SSE state](#)

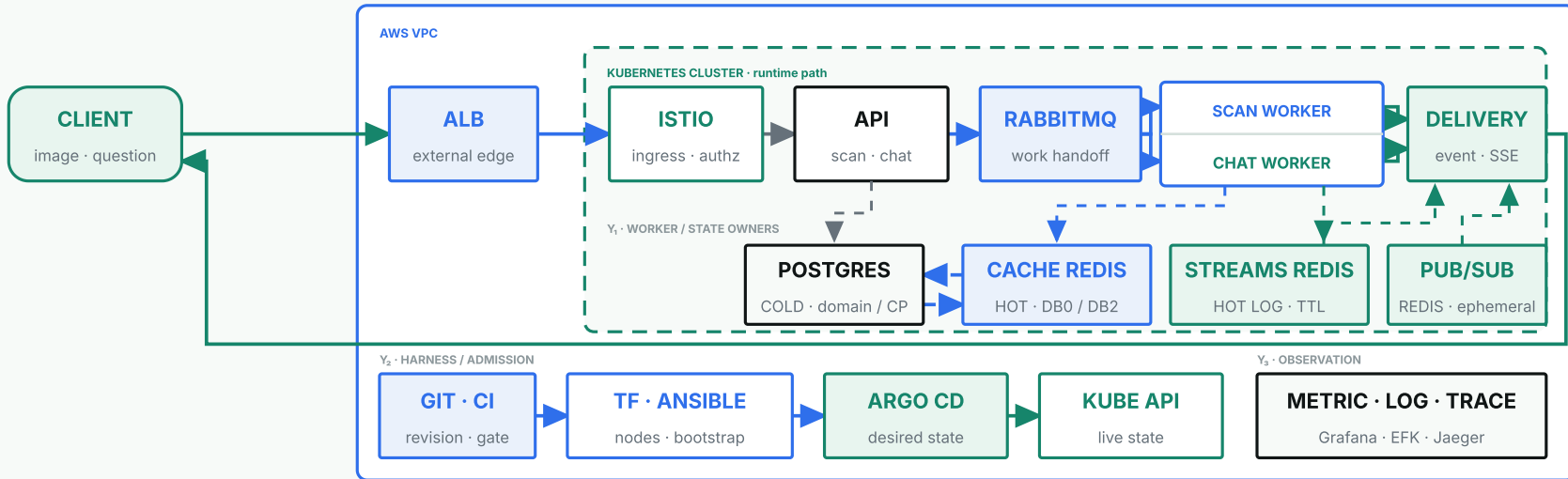
승격 권한 · 잔존 경계

현재 코드의 job-event-SSE recovery 경로를 근거로 한 설계 범위입니다. task 별 효과와 복구 성공률은 동일 조건의 대조 실행이 없어 주장하지 않습니다.

Eco²는 LLM 판단을 typed execution·state·admission owner에 결속합니다

API가 accepted job을 만들면 RabbitMQ가 Scan의 Celery DAG와 Chat의 LangGraph Agent에 실행 소유권을 넘깁니다. 두 worker는 model/tool 실행, checkpoint-retry, progress-terminal event 생성을 책임지고 Event Router→Pub/Sub→SSE가 사용자 화면까지 전달합니다. Cache Redis DB0/DB2는 hot execution state, PostgreSQL은 durable domain/cold checkpoint, Streams Redis는 TTL event history를 맡습니다.

X · REQUEST → WORK HANDOFF → EXECUTION → TERMINAL DELIVERY



LEGEND · solid green return = SSE progress/result/reconnect · dashed = state join · Cache→PG sync · PG→Cache miss promote

문제 · 판정 질문

열은 한 요청의 lifecycle, 행은 LLM Agent runtime-state-harness/admission-observation owner를 뜻합니다. 이후 색선은 이 개요의 연결 하나씩을 failure와 repair contract로 확대합니다.

근거 · TEST / RUN ARTIFACT

[Scan worker queues](#) · [Chat worker task](#) · [Event Router](#) · [Frontend SSE](#)

승격 권한 · 잔존 경계

고성능 LLM은 판단과 tool proposal을 넓히지만 side effect-배포-완료 판정 권한은 각 system owner에 남습니다. 현재 cluster health-traffic-retention은 별도 run evidence가 필요합니다.

01

LLM 판단을 실행 계약으로 묶습니다

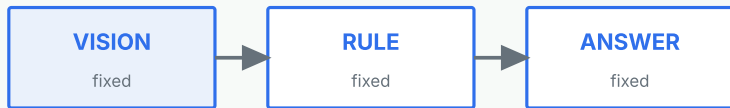
Fixed Scan DAG와 intent-conditioned query graph를 구분하고 proposal·command·reducer·evaluator의 권한을 실행 evidence에 결속합니다.

INTENT · TOOL · STATE · REDUCER · EVALUATION

고정 Scan DAG와 intent-conditioned query graph를 분리했습니다

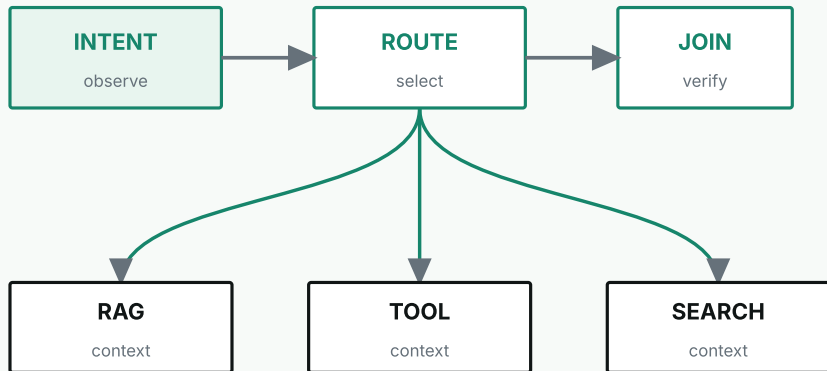
Image scan은 입력·출력 schema와 stage order가 반복 실행에서 유지돼 fixed DAG로 남겼습니다. 환경 질의는 intent와 tool observation에 따라 retrieval·domain API·search 조합이 달라져 conditional route를 열었습니다. Intent contract는 required field와 producer-fallback을 함께 고정하고 optional route는 응답을 유지하되 missing evidence를 남깁니다.

STABLE DOMAIN



scan · known stage order

VARIABLE QUESTION



INTENT CONTRACT

required · waste / bulk / location / collection point → producer + fallback

OPTIONAL

general / character / weather / web / price / image → fail-open + missing evidence

문제 · 판정 질문

모든 질의를 같은 DAG에 넣으면 branch가 불필요하게 늘고, 모든 결정을 모델에 맡기면 required evidence와 fallback의 owner가 사라집니다. 반복 순서가 안정적인 경로는 fixed stage로, 새 관측이 다음 operation을 바꾸는 경로는 intent-conditioned graph로 분리했습니다.

근거 · TEST / RUN ARTIFACT

[Scan task chain](#) · [Chat graph factory](#) · [Initial graph decision](#) · 164 · [Routing contracts](#)

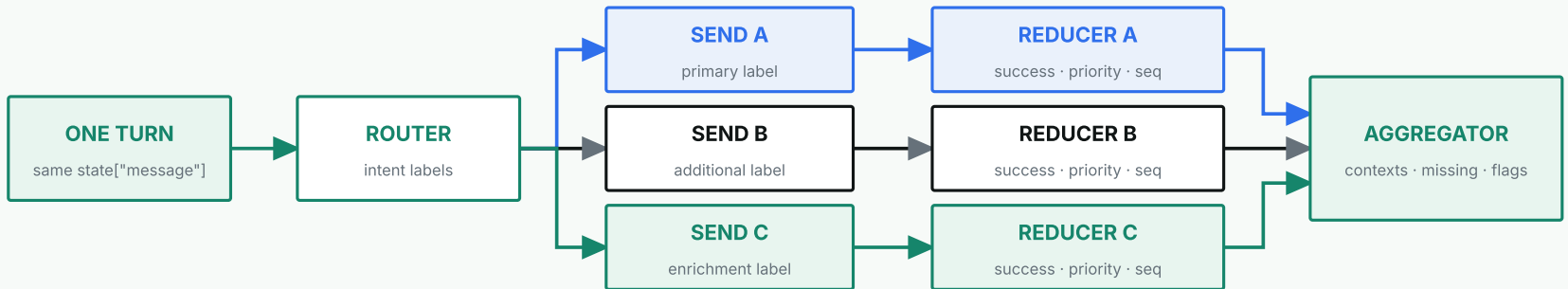
승격 권한 · 잔존 경계

Intent label 오류는 새 failure surface입니다. Required route는 missing field를 terminal diagnosis로 남기고 optional route는 availability를 유지하므로 enrichment 노력을 별도로 관측해야 합니다.

Reducer order는 고정됐지만 query→branch evidence binding은 OPEN입니다

Router는 동일 turn state를 선택된 intent node로 fan-out하고 **activated_nodes**가 같은 target의 중복 실행을 막습니다. 각 channel reducer는 비결정적 도착 순서를 success·priority·seq로 정규화하며 aggregator가 required context의 누락과 fallback flag를 기록합니다. 그러나 current main의 branch node는 **decomposed_queries/current_query** 대신 같은 **state[message]**를 읽어 query-specific evidence binding이 달하지 않았습니다.

BRANCH LIFECYCLE · fan-out → order normalization → missing-evidence flags

**IMPLEMENTED**

activated_nodes admits each target once

OUTPUT

needs_fallback · missing_required

OPEN 1query_i → branch_i evidence binding**OPEN 2**

consumer route for fallback flags

DECISION

arrival timing cannot change reducer order; missing evidence cannot silently become a complete answer

문제 · 판정 질문

병렬 node를 추가해도 같은 evidence가 각 branch에 결속됐는지, 늦게 도착한 결과가 어떤 순서로 합쳐지는지, required field가 빠졌을 때 누가 멈추는지는 정해지지 않습니다. Admission-reducer-aggregator를 분리해 각각의 판정 위치를 고정했습니다.

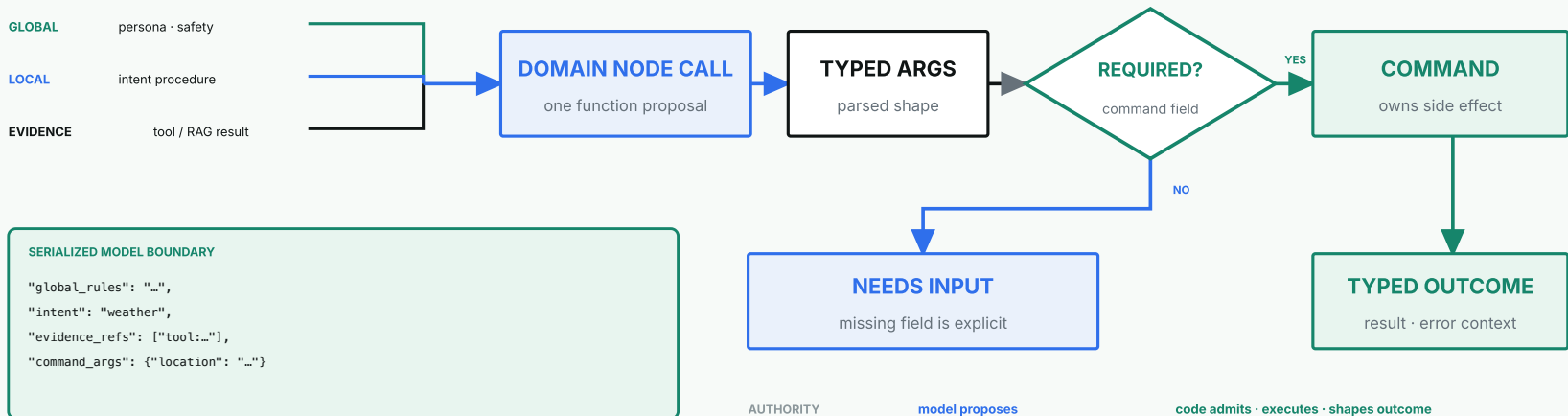
근거 · TEST / RUN ARTIFACT
[Dynamic router](#) · [State reducers](#) · [Aggregator](#) · [Context contracts](#)
승격 권한 · 잔존 경계

Current code proves Send(node, state), node de-duplication, reducer ordering과 output flags입니다. Query-specific branch binding과 fallback consumer route는 OPEN이며 dynamic replanning을 주장하지 않습니다.

Production domain node는 one-shot proposal 뒤 typed command만 실행합니다

Route는 global 규율·intent procedure·tool/RAG evidence를 한 domain-node input으로 조립합니다. Model은 해당 node에서 function-call argument까지만 제안합니다. Required field가 없으면 입력 요청으로 종료하고, 조건을 충족한 argument만 application command를 통과해 typed outcome이 됩니다. 이는 전체 graph가 한 번의 model call로 끝난다는 뜻이 아닙니다.

ROUTE-SELECTED CONTEXT → ONE MODEL PROPOSAL → CODE-OWNED SIDE EFFECT



Missing context stops before the side-effect boundary

문제 · 판정 질문

한 거대한 prompt는 route별 failure를 숨기고, 모델이 실행까지 소유하면 plausible argument가 실행 사실을 덮습니다. Context selection과 side-effect admission을 분리해 어느 입력이 어떤 command를 만들었는지 복원합니다.

근거 · TEST / RUN ARTIFACT

[Prompt loader](#) · [Collection point node](#) · [Weather node](#) · [Answer command](#)

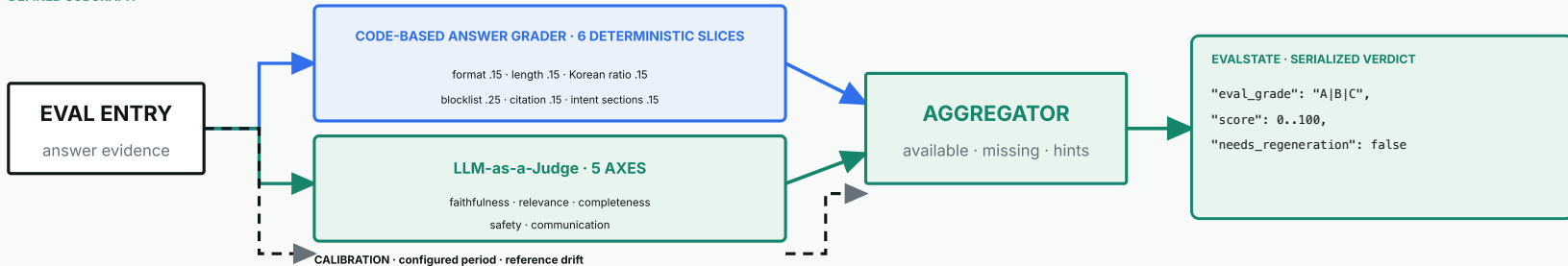
승격 권한 · 잔존 경계

이 구현은 bounded tool routing입니다. 반복 ReAct loop-동적 replanning-branch-local query binding은 현재 claim에 포함하지 않습니다.

Metric schema는 정의됐지만 실행 trajectory가 없어 승격을 멈췄습니다

CodeGraderService는 형식·길이·한국어 비율·금지 표현·출처·intent section을 6개 field와 명시된 가중치로 직렬화하며 unit test가 각 failure를 고정합니다. LLM-as-a-Judge는 faithfulness·relevance·completeness·safety·communication 5축을 정의합니다. 그러나 pinned parent assembly가 eval_counter signature에서 멈춰 task-level trajectory·metric·review artifact를 생성하지 못하며, 이 결손이 promotion을 차단합니다.

DEFINED SUBGRAPH



PINNED PARENT ASSEMBLY



NEXT BLOCKERS

missing adapter / asset · parent edge omits counter · intended test cannot collect

Earliest executable failure stops the evidence claim

문제 · 판정 질문

Task·model·workspace·grader 조건이 고정된 실행 trajectory가 있어야 metric과 reviewer 판단이 promotion evidence가 됩니다. 현재 가장 먼저 끊긴 executable edge에서 승격을 멈춥니다.

근거 · TEST / RUN ARTIFACT

[Code grader](#) · [Code grader tests](#) · [Eval graph factory](#) · [Dependency assembly](#) · [Parent factory](#) · [ACES](#) · [external eval frame](#)

승격 권한 · 잔존 경계

6-slice service와 unit test는 직접 근거입니다. Production evaluation, paired trial, ATIF normalization, human-grader calibration, regeneration 소비, model-quality improvement는 OPEN입니다.

02

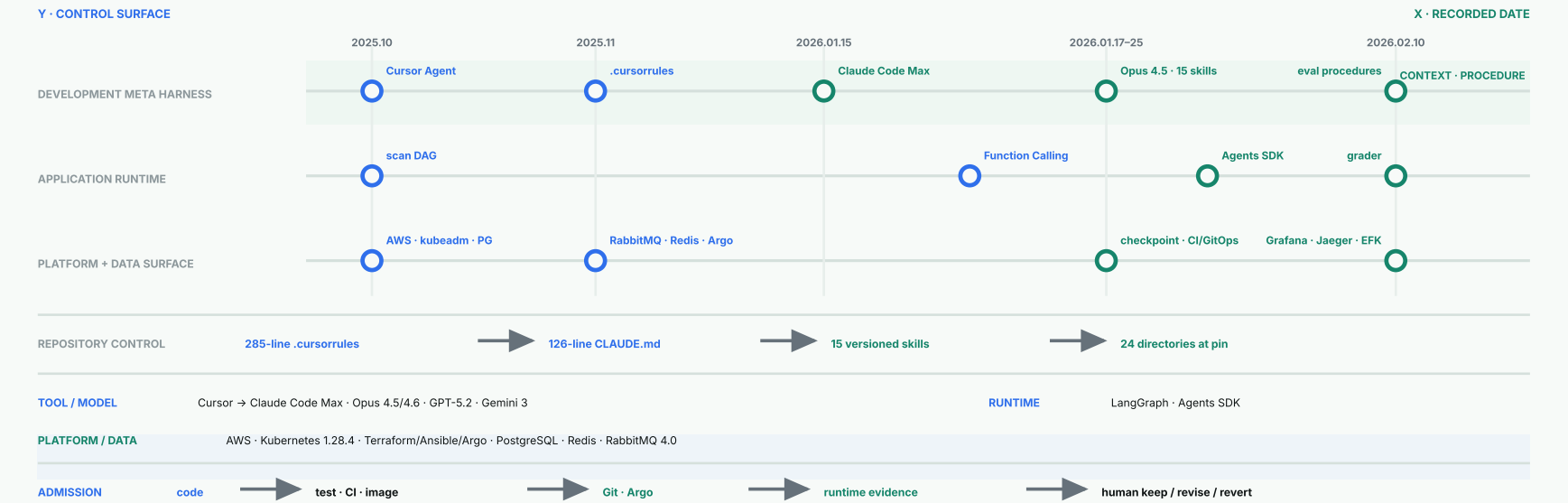
AI 변경의 제어면

대화의 지시를 저장소 procedure로 옮기고 code·test·run artifact를 통과한 변경만 다음 경계로 보냅니다.

RULE · PROCEDURE · CHANGE · TEST · ADMISSION

Repository procedure가 선택한 변경을 test·run artifact·human review로 admission합니다

Cursor session에 머물던 맥락은 `.cursorrules`, `CLAUDE.md`, 15개 versioned skill로 분해됐고 current pin에는 24개 procedure directory가 남았습니다. Procedure는 변경 후보의 context를 제공하고, path-specific test-image-GitOps-runtime evidence가 실행 산출물을 남기며, human review가 keep·revise·revert를 결정합니다. 이 세 표면을 같은 repository revision에서 읽어 개발 Agent의 change authority를 제한했습니다.



문제 · 판정 질문

Procedure 설명, code/test 산출물, runtime evidence, reviewer 판단이 같은 revision을 가리키는가? 이 결속이 성립할 때만 다음 변경이 이전 판단을 재사용할 수 있습니다.

근거 · TEST / RUN ARTIFACT

[Cursor→Claude Code record](#) · [Opus 4.5 change](#) · [15 versioned skills](#) · [Agents SDK path](#) · [Current runtime deps](#) · [ACES · external eval frame](#)

승격 권한 · 잔존 경계

Eco²는 ACES의 paired with/without-skill trial-ATIF·Skill Lift를 구현하지 않았습니다. Skill inventory의 marginal value, human-grader calibration, defect-rate-token 절감은 미측정입니다. Claude skills는 development control plane에 속합니다.

Current CI는 변경 경로를 막고 dated audit은 다음 test 범위를 정했습니다

Current workflow는 Black-Ruff-pytest, Go test, Terraform-Helm-Kustomize-Kubeconform을 변경 경로별로 실행하고 image tag-rendered manifest를 남깁니다. 별도 2026-01-14 audit은 unit test의 success-error-serialization-connection contract와 pytest-cov-Radon metric을 같은 실행 기록에 결속했습니다. 115 pass, 전체 49%, 선택 core 88-100%, 424 block 평균 A 2.17은 당시 test scope를 설명하며 current coverage로 읽지 않습니다.

X · SELECTOR → VALIDATION → ARTIFACT → ADMISSION

Y · CHANGED SURFACE

CHANGED SURFACE	PATH SELECTOR	STATIC / EXECUTABLE GATE	ARTIFACT	ADMISSION OUTPUT
API SERVICES	apps/{8 APIs} changed-files matrix	black · ruff · pytest format · lint · test	Buildx image revision-bound	tag + manifest next owner
WORKERS	apps/*_worker changed-files matrix	kustomize · black · ruff unit pytest	worker image revision-bound	manifest candidate next owner
SSE / EVENT	gateway · router changed-files matrix	Redis svc · black · ruff pytest	component image revision-bound	realtime candidate next owner
GO AUTHZ	apps/ext_authz changed-files matrix	go test ./... compile · unit	authz image revision-bound	manifest tag next owner
INFRA	tf · charts · overlays changed-files matrix	tf fmt/validate · Helm/ct Kustomize · Kubeconform	rendered manifests revision-bound	GitOps candidate next owner

DATED CHAT_WORKER AUDIT · 2026-01-14

115 / 115
pytest reported pass

49%
total measured coverage

88-100%
selected core paths

A · 2.17
424 blocks · Radon

TEST DESIGN · Port mocks · AAA · fixtures · success/error/serialization/connection

GAP · LLM/LangGraph/setup/presentation → integration/E2E

문제 · 판정 질문

Changed-files matrix가 test 환경을 고르고, executable gate가 artifact를 남기며, coverage-complexity metric과 reviewer가 다음 test scope를 결정합니다. 각 판정이 같은 revision을 가리키는지 확인합니다.

근거 · TEST / RUN ARTIFACT

[Automatic CI lanes](#) · [Dated test-quality audit](#) · [Radon / coverage config](#) · [Manual Sonar workflow](#) · [CodeClimate config](#)

승격 권한 · 잔존 경계

115 pass-49%-88-100%-A 2.17은 2026-01-14 dated artifact입니다. Current pin의 전역 coverage, paired baseline, live skill lift, calibrated human score는 없습니다. Radon-CodeClimate-SonarCloud는 자동 required gate가 아닙니다.

03

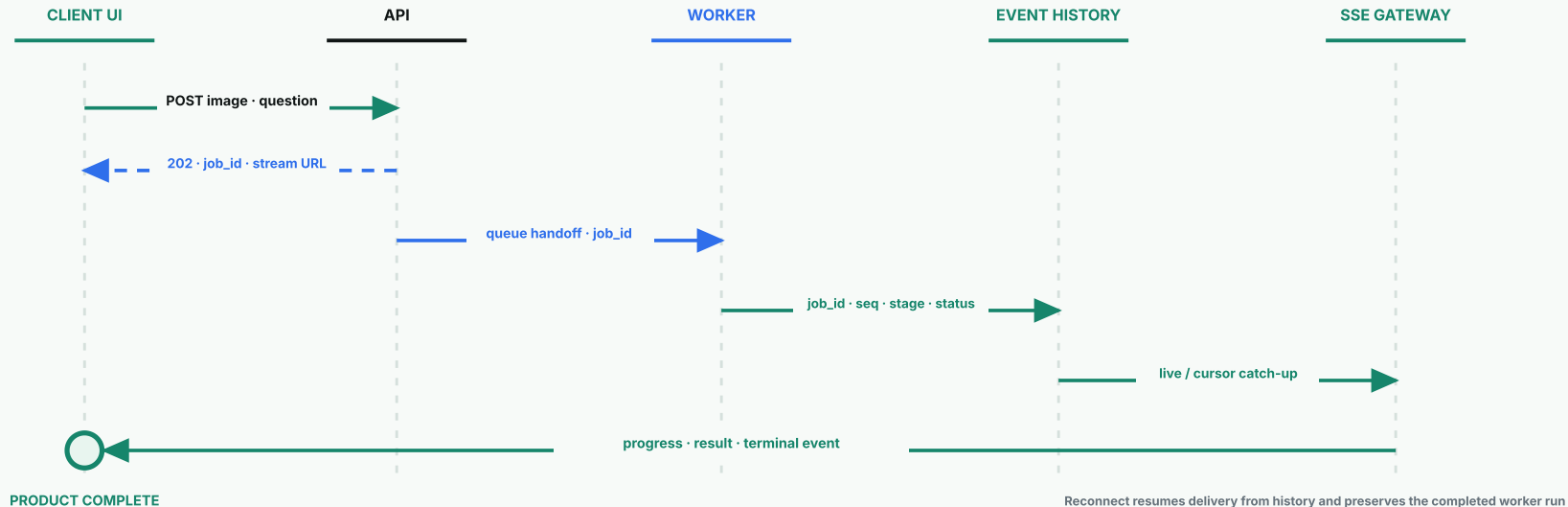
사용자 관측에서 완료를 닫습니다

Revision-request-terminal state를 결속해 client가 복원한 상태에서 workflow 완료를 판정합니다.

REVISION · REQUEST · TERMINAL STATE

Client가 terminal event를 적용한 시점이 제품 완료입니다

API 수락, worker 종료, ordered event 기록, SSE 전달은 서로 다른 관측입니다. `job_id·seq`로 연결된 terminal event가 live 또는 cursor catch-up을 통해 UI state에 반영될 때 이 workflow의 사용자 결과가 닫힙니다.



문제 · 판정 질문

API-worker-history-delivery 중 어느 관측이 사용자 완료를 판정할 수 있는지 분리합니다.
Client가 terminal event를 적용한 순간만 제품 완료를 씁니다.

근거 · TEST / RUN ARTIFACT

[Scan API contract](#) · [Event Router](#) · [SSE Gateway](#) · [Frontend SSE hook](#)

승격 권한 · 잔존 경계

추가된 event history와 catch-up 경로는 terminal 복원을 가능하게 하며,
`job_id·seq` 보존과 reconnect 운영 비용을 함께 요구합니다.

Signal·revision·terminal이 같은 run에 결속된 비교만 채택합니다

EFK는 component log를 ECS field로 정규화하고, Jaeger는 span parentage를, Prometheus·Grafana는 queue·worker·router pressure를 관측합니다. Dashboard 한 장은 failure location을 가리킬 수 있어도 어느 배포의 어느 요청이 어떤 terminal에 도달했지는 결정하지 못합니다. 따라서 artifact revision, request/job identity, timeout과 terminal denominator를 비교 자격 규칙으로 사용했습니다. 이는 배포된 gate가 아니라 artifact 해석과 human review의 admission 기준입니다.

OBSERVABILITY EVOLUTION · DATE / QUESTION ANSWERED

STRUCTURED LOG



12/17

Fluent Bit · ECS · EFK

DISTRIBUTED TRACE



12/17

OTel · Jaeger · Kiali

PIPELINE METRIC



12/27

ServiceMonitor · Grafana

TRACE JOIN AUDIT



01/19

queue parentage gap

CURRENT PUBLIC MAIN

SOURCE / CARRIER

NORMALIZE / CORRELATE

STORE / VIEW

COMPARISON ADMISSION

LOG

STDOUT JSON

raw component event



Fluent Bit DS

normalize ECS



EFK

service.* - kubernetes.*

TRACE

W3C CARRIER

traceparent



TASKIQ labels={}

parentage gap



worker span

new trace root

METRIC

SERVICE MONITOR

scrape target



Prometheus

time series



Grafana

queue · worker · router

COMPARISON RULE · ANALYTICAL

IDENTITY

revision · request/job · timeout

SIGNAL

log · span · metric

OUTCOME

completed · failed · OPEN

LIMIT

snapshot ≠ raw run ledger

Signal localizes mechanism; only identity + terminal denominator can authorize comparison

문제 · 판정 질문

Component signal만 남기면 배포 변경과 사용자 결과의 인과 경로를 복원할 수 없습니다. Signal은 mechanism localization을 맡기고, revision-request/job-terminal join이 comparison eligibility를 맡도록 authority를 분리했습니다.

근거 · TEST / RUN ARTIFACT

[Fluent Bit · ECS](#) · [Jaeger application](#) · [Event Bus dashboard](#) · [Chat trace handoff](#)

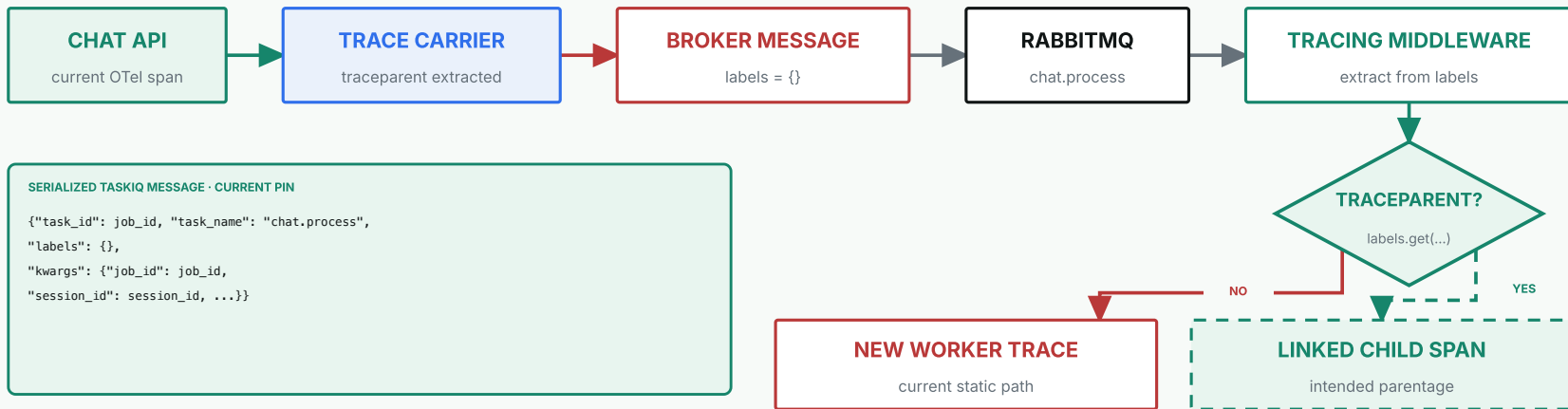
승격 권한 · 잔존 경계

Pinned main의 queue labels는 비어 있어 API→worker parentage가 닫히지 않습니다. Jaeger는 memory storage-15,000 trace cap, Elasticsearch는 single-node 구성이고 Grafana snapshot은 raw k6 ledger를 대체하지 않습니다.

빈 queue labels가 Chat API와 worker trace의 parentage를 끊습니다

Pinned main은 W3C carrier를 추출하지만 TaskIQ payload와 BrokerMessage의 `labels`를 모두 `{}`로 직렬화합니다. Worker middleware는 `traceparent`가 없으면 parent context 없이 consumer span을 시작하므로 API→MQ→worker의 단일 trace chain을 현재 코드로 주장할 수 없습니다.

CURRENT PINNED MESSAGE PATH



HISTORICAL LOG AUDIT

125,398 / 1,750,699 = 7.16% with trace.id

window · revision unavailable

문제 · 판정 질문

`job_id`는 작업과 Redis event를 결속하고, `traceparent`는 API span과 worker span의 latency parentage를 결속합니다. 두 식별자의 권한을 분리해 누락 위치를 코드 경계에서 판정했습니다.

근거 · TEST / RUN ARTIFACT

[Chat job submitter](#) · [Worker tracing middleware](#) · [Redis event trace fields](#) · [Historical trace audit](#) · 41

승격 권한 · 잔존 경계

Current finding is a static code-path audit, not measured runtime incidence. The historical 7.16% ratio lacks an exact window and revision.

04

요청 수명을 넘긴 실행을 복원합니다

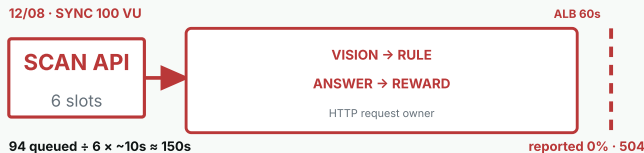
Deadline과 delivery pressure를 관측해 job·history·client recovery의 owner를 분리했습니다.

JOB IDENTITY · HISTORY · ACK · CURSOR · TERMINAL OUTCOME

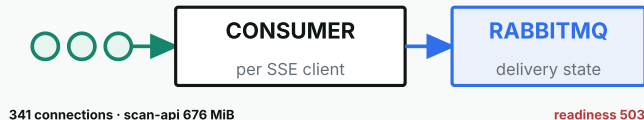
동기 timeout과 broker pressure가 실행·전달 소유권을 분리했습니다

동기 Scan은 Vision·Rule·Answer·Reward가 끝날 때까지 HTTP request가 실행을 소유했습니다. VU 100의 추정 대기열 약 150초가 ALB 60초를 넘자 API는 job reference를 반환하고 RabbitMQ·Celery가 실행을 이어받았습니다. 그러나 VU 50 비동기 v1은 per-client RabbitMQ consumer를 341 connections와 scan-api 676 MiB, readiness 503으로 드러냈습니다. Current path는 task-ordered history-live delivery-recovery를 서로 다른 owner에 배치합니다.

DATED PRESSURE · TWO OWNERSHIP FAILURES



12/28 · ASYNC V1 · 50 VU



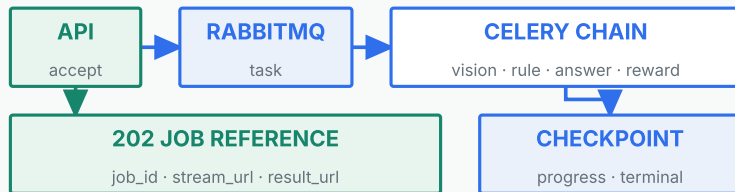
OBSERVATION

execution left HTTP; delivery retained broker state

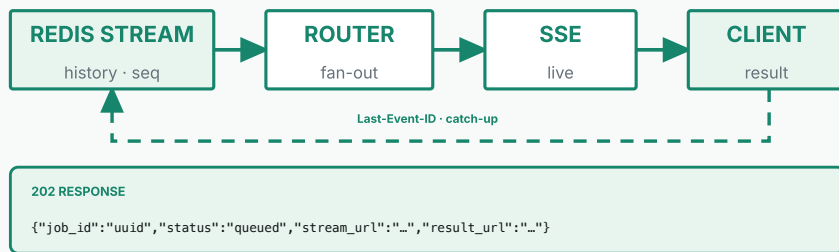
DECISION

split task, history, live delivery and recovery owners

ADOPTED OWNER GRAPH · CURRENT PIN



DELIVERY OWNERS · ordered history → fan-out → live → client



문제 · 판정 질문

request deadline과 per-client broker state를 각각 어떤 owner에게 넘겨야 retry-terminal-client recovery를 같은 job으로 복원할 수 있는가?

근거 · TEST / RUN ARTIFACT

[Synchronous Scan observation · 17](#) · [50-VU report · 78](#) · [Submit command](#) · [Event Router](#) · [SSE Gateway](#)

승격 권한 · 잔존 경계

The VU100 synchronous and VU50 asynchronous observations are unmatched historical runs. The figure explains the decision lineage and current owner path; it does not claim a matched capacity or latency improvement.

실패 owner를 분리한 뒤 500→1,000 VU 검증 workload를 열었습니다

동기 request의 deadline과 per-client broker state를 분리하고 reconnect contract를 추가한 뒤 polling workload를 500·600·700·800·900·1,000 VU로 넓혔습니다. 각 run의 completed·failed·OPEN과 E2E p95가 다음 부하의 조사 대상을 정합니다. 1,000 VU에서는 worker floor/ceiling 변경 뒤에도 probe restart가 잔존 병목으로 남았습니다.

X · DATED OBSERVATION → ADMITTED CHANGE

12/08 · SYNC 100 VU



~150s queue > ALB 60s

job identity · Celery Chain

12/22~28 · ASYNC 10~50 VU



341 broker conns · readiness 503

Streams · KEDA · Event Router

01/23~25 · DELIVERY REPAIR



stale listener · reconnect loss

Last-Event-ID · 4 shard channels

01/27 · POLLING LADDER



queue wait → probe restart

worker floor/ceiling retest

RUN LEDGER · 2026-01-27

C/F/O = completed / failed / OPEN

TARGET VU	500	600	700+	800	900	1000*
C / F / O	1336 / 0 / 0	1401 / 4 / 3	1313 / 11 / 172	1378 / 4 / 4	1532 / 4 / 4	1469 / 33 / 16
E2E p95	83.3s	108.3s	122.3s	144.6s	149.6s	173.3s

VU1000 · SAME TARGET/TIER/300s

06:12 · 1494 / 53 / 45

min 1→2 · max 3→5

worker node 1→2

20:06 · 1469 / 33 / 16

↑ 172 OPEN outlier · monotonic fit 제외 · * final run totals differ

문제 · 판정 질문

각 workload에서 드러난 deadline·connection·queue·probe owner를 보강한 뒤 다음 검증 부하를 열었는가?

근거 · TEST / RUN ARTIFACT

[VU500 report · 252](#) · [VU1000 report · 259](#) · [Current scaling overlay](#) · [Current shard delivery](#)

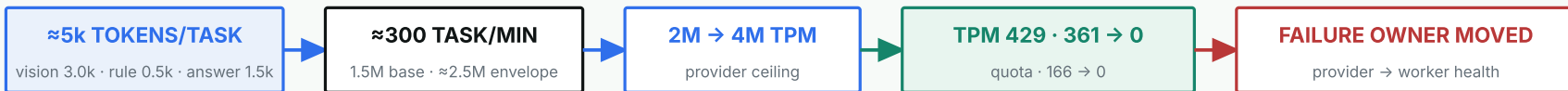
승격 권한 · 잔존 경계

VU는 k6 검증 부하이며 실제 사용자 증가가 아닙니다. 12월 run과 1월 run, 그리고 일부 1월 run도 config-submitted total이 달라 raw ledger를 시간순으로 배치했을 뿐 하나의 성능 곡선으로 연결하지 않습니다.

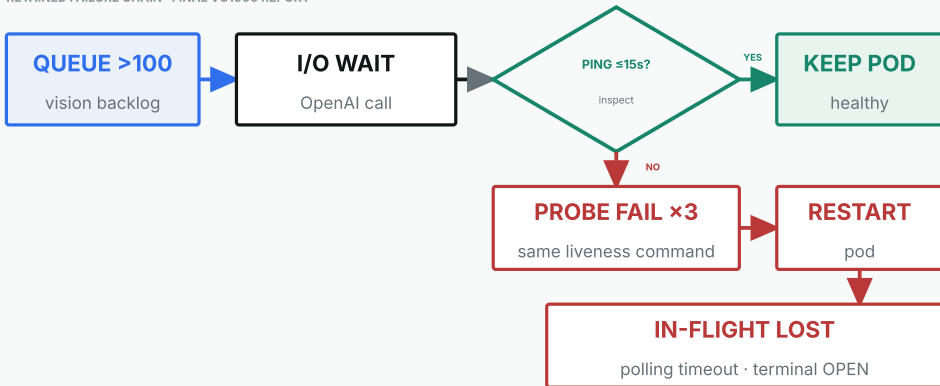
Tier 4에서 429가 닫힌 뒤 retained report는 probe restart를 추적했습니다

약 5k tokens/task의 demand가 2M TPM ceiling을 넘긴 기록은 4M·auto-recharge에서 429 0건으로 닫혔습니다. 같은 VU1000의 multi-change run에서 worker floor-ceiling과 node를 늘린 뒤에도 E2E p95는 173.3초였습니다. Retained report는 queue backlog에서 `celery inspect ping`이 늦어져 동일 command를 쓰는 liveness가 실패하고 재시작 중 task가 유실될 수 있는 경로를 다음 검증 대상으로 지목했습니다.

PROVIDER BOUNDARY · REPORT ESTIMATE



RETAINED FAILURE CHAIN · FINAL VU1000 REPORT



SAME VU1000 / TIER4 · MULTI-CHANGE OBSERVATION

	06:12 INITIAL	20:06 FINAL
worker min / max	1 / 3	2 / 5
worker nodes	1	2
failed	53	33
throughput · req/min	378.7	373.4
submit p95 · s	0.863	0.787
poll p95 · s	2.547	2.609
E2E p95 · s	166.7	173.3
E2E p99	OPEN	OPEN

문제 · 판정 질문

Provider ceiling은 429를, KEDA·node 수는 worker supply를, liveness는 restart를 각각 소유합니다. 실패 owner를 분리해 독립 HTTP health와 queue backpressure를 다음 검증 대상으로 남겼습니다.

근거 · TEST / RUN ARTIFACT

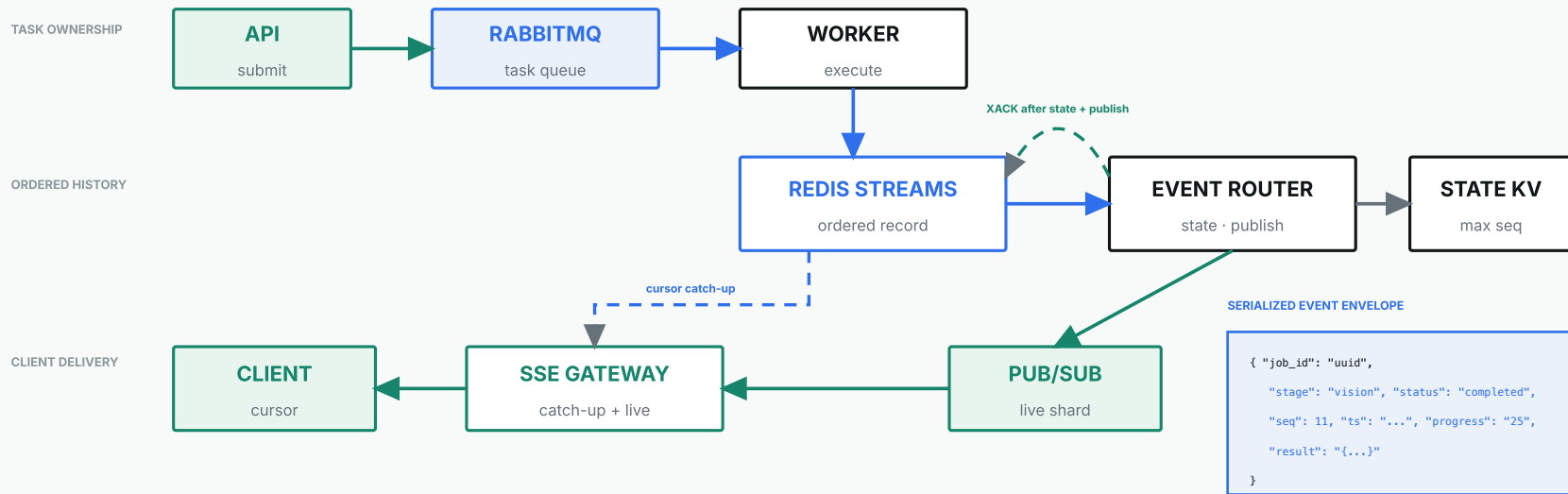
[Tier 3 report](#) · [Tier 4 report](#) · [Final run report](#) · [Current worker probes](#)

승격 권한 · 잔존 경계

두 Tier 4 run은 submitted total과 worker config가 달라 53→33 감소의 인과를 주장하지 않습니다. Probe chain과 task loss는 retained report의 진단이며 독립 /health는 권고·미구현입니다. E2E p99는 report에 없어 OPEN입니다.

Task·history·live delivery·recovery를 job_id·seq로 결속했습니다

초기 RabbitMQ task handoff에서 ordered Streams history, Pub/Sub live fan-out, PEL-Last-Event-ID recovery로 책임을 분리했습니다. Worker는 stage event를 Streams record로 직렬화하고, Router는 state update와 Pub/Sub publish가 성공한 뒤 XACK합니다. SSE Gateway는 cursor catch-up을 마친 뒤 live shard를 이어 받습니다.



EVOLUTION · RabbitMQ task → Streams history → Pub/Sub live → PEL / Last-Event-ID recovery

문제 · 판정 질문

Streams record의 job_id·seq가 ACK, state update, catch-up과 live resume을 같은 event lifecycle로 추적하게 하는가?

근거 · TEST / RUN ARTIFACT

[Event Router source](#) · [SSE Gateway source](#) · [Stage event serializer](#) · [HA architecture - 100](#)

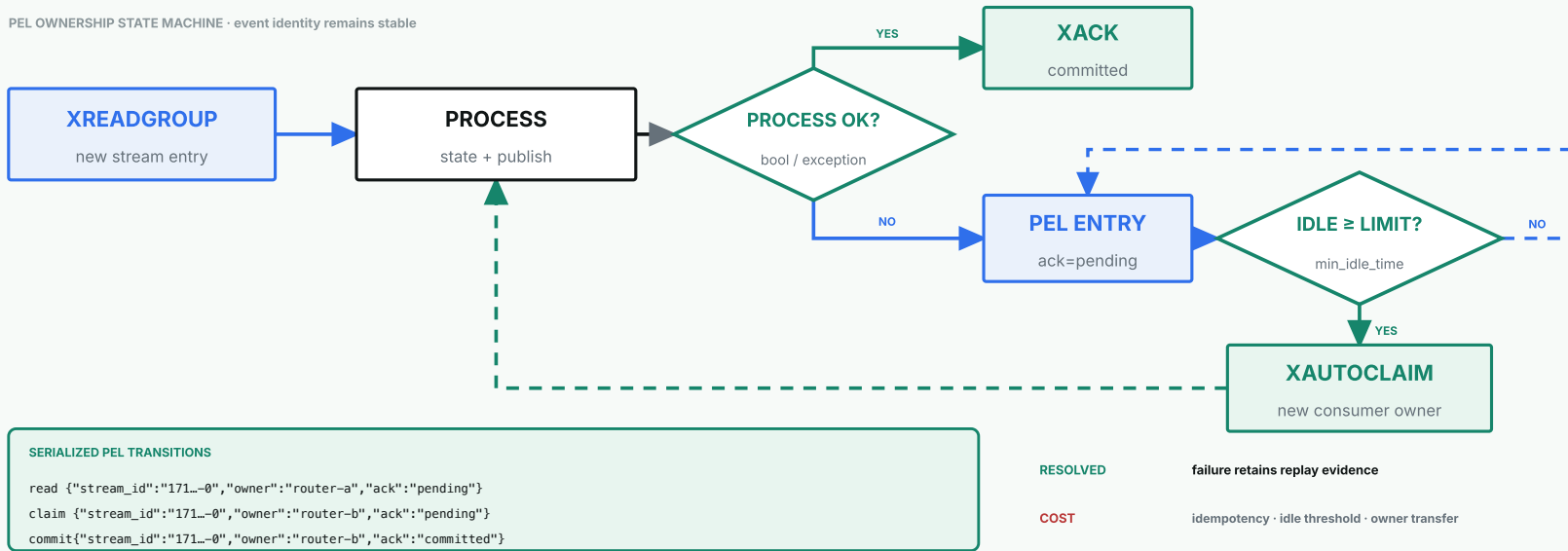
승격 권한 · 잔존 경계

Logical responsibility separation is shown; current public Streams storage is not claimed as long-term durable.

실패한 entry는 PEL에 남고 XAUTOCLAIM이 처리 소유권만 이전합니다

`process_event`가 false를 반환하거나 exception을 내면 XACK를 건너뛰어 같은 `stream_id`를 PEL에 보존합니다. Idle threshold를 넘긴 entry는 XAUTOCLAIM이 새 consumer에게 이전하며, 재 처리도 state update와 publish가 성공한 뒤에만 committed로 닫힙니다. Event identity는 유지되고 owner와 acknowledgement state만 전이합니다.

PEL OWNERSHIP STATE MACHINE · event identity remains stable



문제 · 판정 질문

처리 실패를 ACK하면 replay evidence가 사라지고, 원본을 복제하면 attempt identity가 갈라집니다. 같은 stream entry를 pending으로 보존하고 ownership만 이전해 failure-safe replay와 idempotency 판정을 함께 유지합니다.

근거 · TEST / RUN ARTIFACT

[Consumer success-before-XACK](#) · [XAUTOCLAIM reclaimer](#) · [Integrity repair](#) · [237](#)

승격 권한 · 잔존 경계

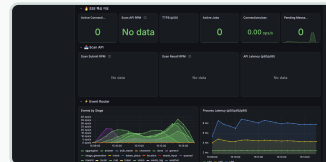
Current source proves pending, reclaim, and success-before-XACK branches. Idle threshold와 consumer identity를 운영해야 하며 pinned matched replay benchmark는 없습니다.

독립 queue snapshot은 terminal 분모와 병치하되 인과 비교에서 제외했습니다

600-800-900-1,000 VU의 독립 Grafana snapshot은 pending-active 최대값과 router p99의 mechanism 위치를 보여 줍니다. 별도 dated run report는 1,518개 제출을 1,469 completed, 33 failed, 16 OPEN으로 보존합니다. 두 artifact에 공통 revision-window identity가 없으므로 queue 수치가 outcome을 만들었다거나 VU 증가에 따른 추세라고 해석하지 않았습니다.

Y · QUEUE SNAPSHOT MAX (count)

X · OFFERED VU · BLUE pending · GREEN active/unacked



PUBLIC SNAPSHOT · 2026-01-27

TERMINAL DENOMINATOR

SUBMITTED	1,518
COMPLETED	1,469
FAILED / OPEN	33 / 16

96.8% all · 97.8% resolved

문제 · 판정 질문

Grafana는 pending-active-router latency가 위치한 mechanism을, k6 report는 submitted-completed-failed-OPEN의 분모를 소유합니다. 한 화면에 배치하되 공통 identity가 없으면 두 evidence owner 사이의 인과 화살표를 그리지 않습니다.

근거 · TEST / RUN ARTIFACT

[VU600 snapshot](#) · [VU800 snapshot](#) · [VU900 snapshot](#) · [VU1000 snapshot](#) · [VU1000 run report](#) · 259

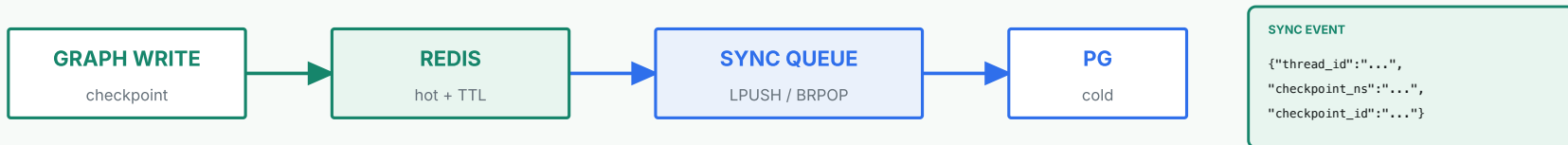
승격 권한 · 잔존 경계

Queue bars are independent per-snapshot maxima on a shared count axis; router p99 is a millisecond annotation. 96.8%는 전체 제출, 97.8%는 resolved 1,502건 분모입니다. Matched time-series와 causal before/after는 미측정입니다.

Redis checkpoint를 PostgreSQL cold recovery로 동기화합니다

Write는 Redis와 sync event를 남기고, read는 hot miss일 때만 PostgreSQL checkpoint를 읽어 Redis에 promote합니다. Sync event의 세 identity가 cold copy를 같은 graph state에 결속합니다.

WRITE PATH



READ PATH



checkpoint recovery and realtime event recovery retain separate cursors

문제 · 판정 질문

Redis는 hot write를, sync queue는 async persistence를, PostgreSQL은 cold recovery를 각각 소유합니다.

근거 · TEST / RUN ARTIFACT

[SyncableRedisSaver](#) · [Checkpoint sync service](#) · [Read-through checkpointer](#) · [Pool incident](#) · 242

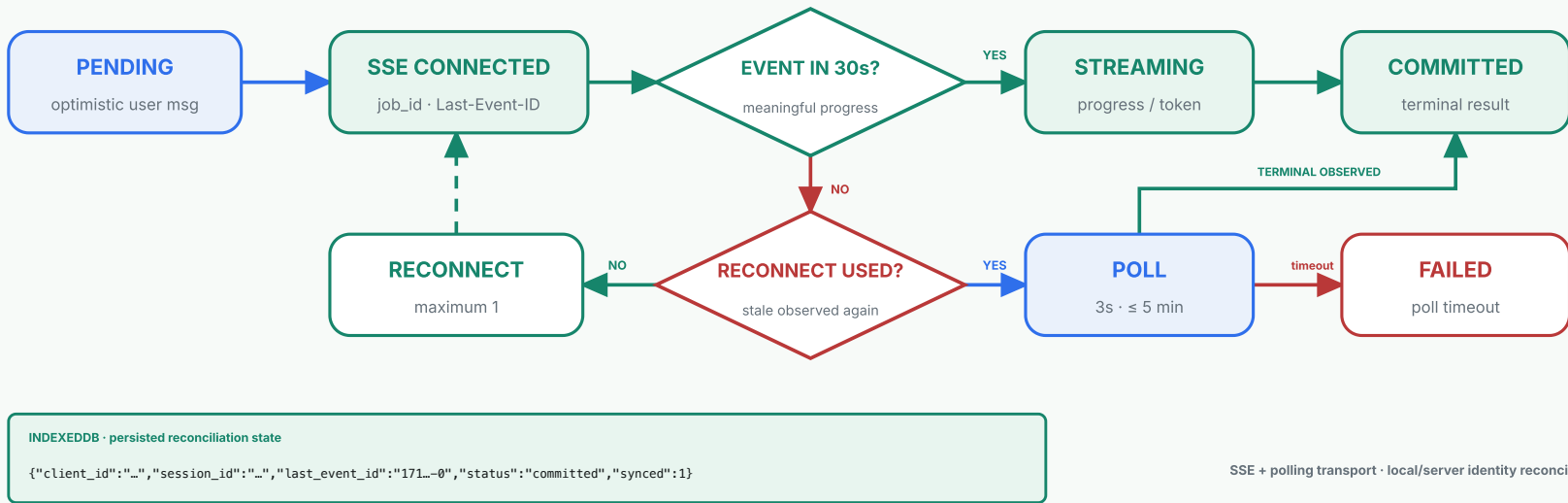
승격 권한 · 잔존 경계

현재 wiring과 명시된 artifact가 이 장의 판정을 지지합니다. 다음 상태·운영 비용·성과는 별도 실행에서 측정합니다.

Reconnect와 polling은 terminal state를 다시 관측하는 복구 action입니다

Client는 pending-streaming-committed-failed를 명시적으로 전이합니다. Last-Event-ID catch-up은 보지 못한 Redis Stream entry를 회수합니다. 30초 안에 meaningful event가 없으면 reconnect를 한 번 사용하고, 다시 stale이면 3초 polling을 최대 5분 수행합니다. Polling에서 terminal을 확인한 경우에만 committed로 전이하며 timeout은 failed로 답습니다.

CLIENT MESSAGE STATE MACHINE · predicate labels own every branch



문제 · 판정 질문

Reconnect와 polling이 추가 observation을 가져온 뒤 client state가 committed 또는 failed로 닫히는지 추적합니다.

근거 · TEST / RUN ARTIFACT

[SSE request parsing](#) · [Catch-up implementation](#) · [IndexedDB session hierarchy](#) · [Agent SSE recovery](#) · [Stale polling fallback](#)

승격 권한 · 잔존 경계

정보 획득 action은 transient gap을 흡수하지만 latency-API traffic-local/server identity reconciliation 비용을 추가합니다.

05

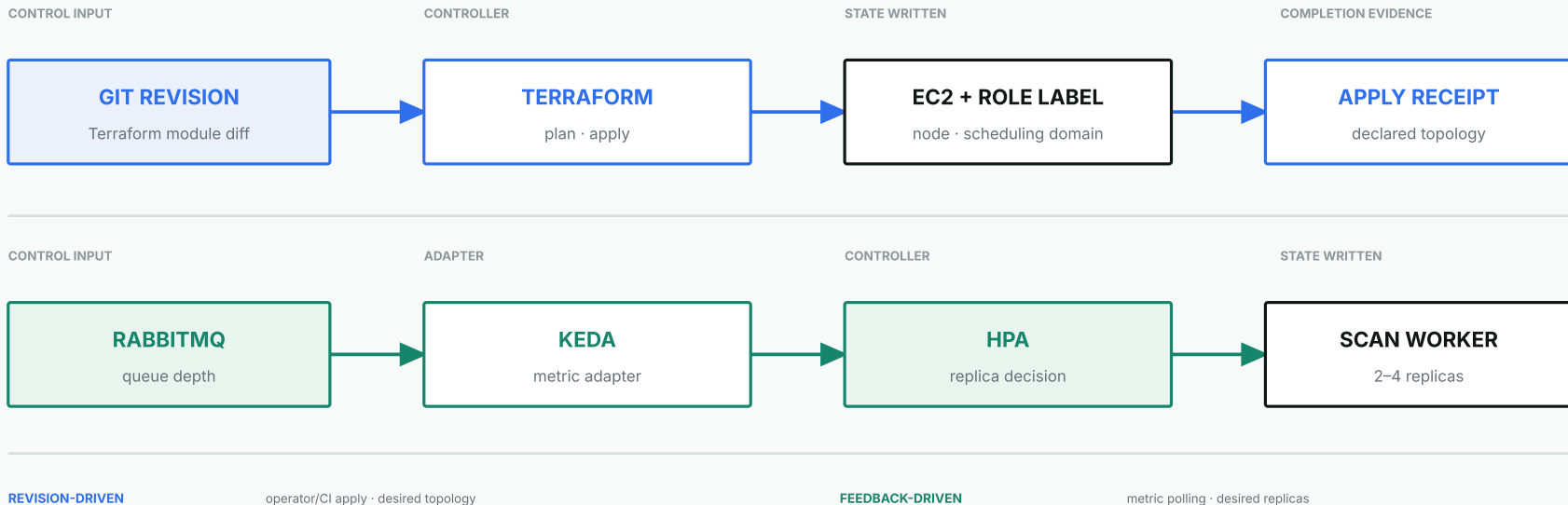
플랫폼 제어면

Resource별 controller가 revision과 topology를 수렴시킵니다.

PROVISION · BOOTSTRAP · RECONCILE · SCALE · ROUTE

Git revision은 node existence를, queue depth는 worker replica를 바꿉니다

Git revision→Terraform은 EC2와 role label을 선언하고 apply receipt에서 닫힙니다. RabbitMQ queue depth→KEDA→HPA는 scan-worker의 desired replica를 2-4 범위에서 조정합니다. 입력·controller·기록 상태가 다른 두 흐름을 별도 축으로 둡니다.



문제 · 판정 질문

물리 resource existence와 event-driven workload replica 중 어느 controller가 어느 변화에 책임지는가?

근거 · TEST / RUN ARTIFACT

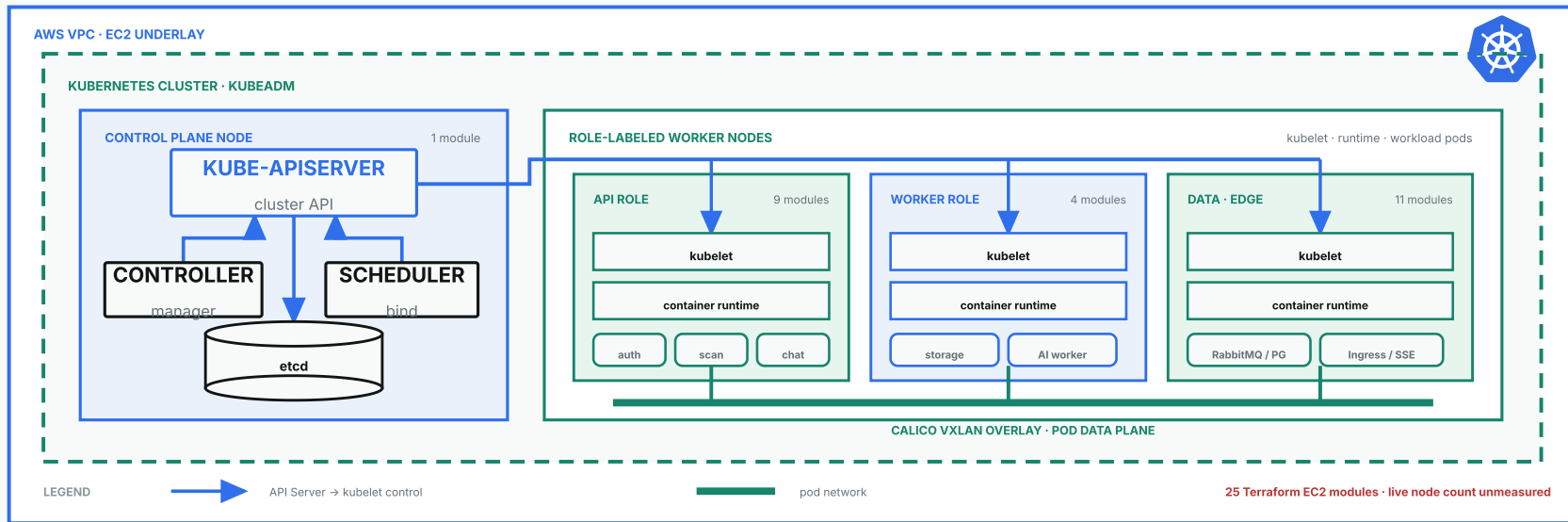
[Terraform current](#) · [Scan worker ScaledObject](#) · [KEDA RabbitMQ scaler](#) · [KEDA scaling model](#)

승격 권한 · 잔존 경계

Public desired-state wiring only; live node inventory and realized autoscaling outcomes are not measured here.

Terraform role label이 Kubernetes control path와 failure domain을 분리합니다

AWS VPC가 kubernetes cluster를 포함하고, cluster 안에서 control-plane node와 세 worker role group이 나뉩니다. Kube-apiserver가 controller-scheduler-etcd와 cluster state를 조정하고 각 node의 kubelet에 control request를 보냅니다. Pod traffic은 Calico VXLAN data plane을 사용합니다.



문제 · 판정 질문

Control-plane state, kubelet control traffic, pod data traffic을 각 boundary에 걸쳐 scheduling-node-network failure의 조사 시작점을 고정합니다.

근거 · TEST / RUN ARTIFACT

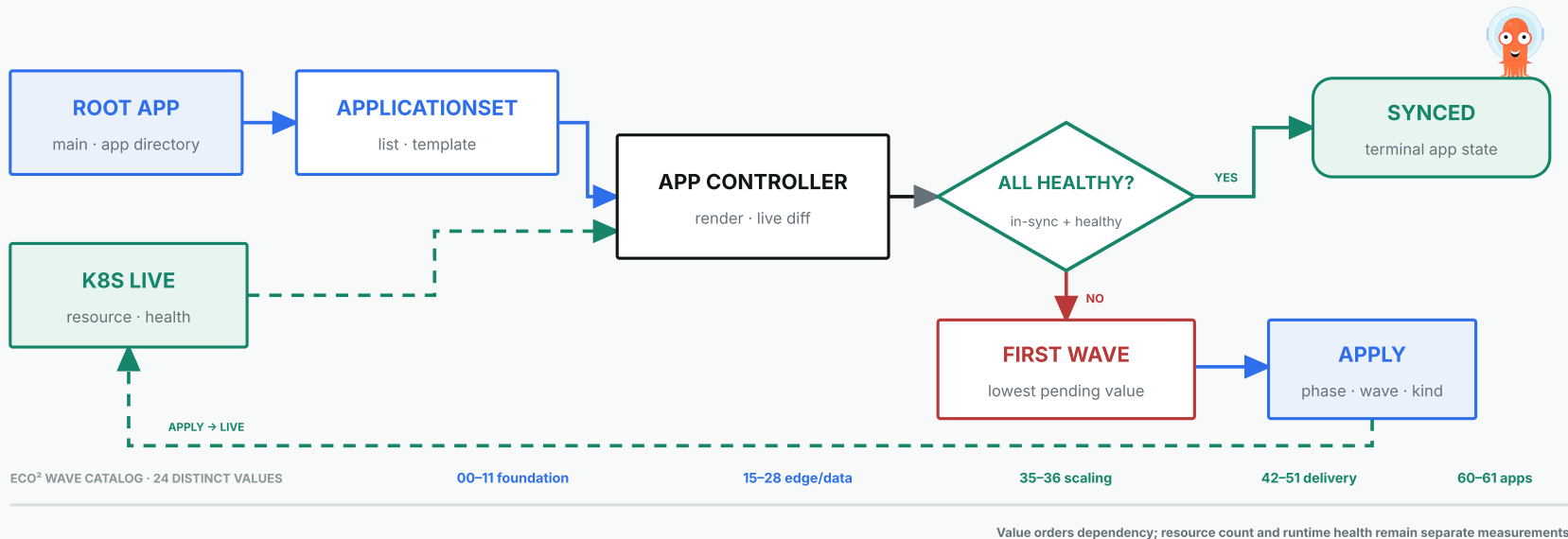
[Terraform main](#) · [Kubelet role labels](#) · [Kubernetes components](#) · [Public portfolio](#)

승격 권한 · 잔존 경계

Role label은 code-declared scheduling boundary를 제공합니다. 세 worker group과 25 Terraform module의 운영 비용은 늘어, live node inventory와 realized isolation outcome은 이 장에서 측정하지 않습니다.

Root App이 확장한 Application을 wave별 health loop로 수렴시킵니다

prod-root가 main의 app directory를 읽고 ApplicationSet이 child Application spec을 만듭니다. App controller는 rendered desired state와 Kubernetes live resource를 비교해 가장 낮은 pending wave를 적용하고 health를 다시 읽습니다. 24개 값의 dependency order이며 resource 수나 성능 지표가 아닙니다.



문제 · 판정 질문

Git path 발견·Application 생성·live reconciliation·health 판정을 다른 control phase로 분리하고, unmet prerequisite가 있는 첫 wave에서 반복합니다.

근거 · TEST / RUN ARTIFACT

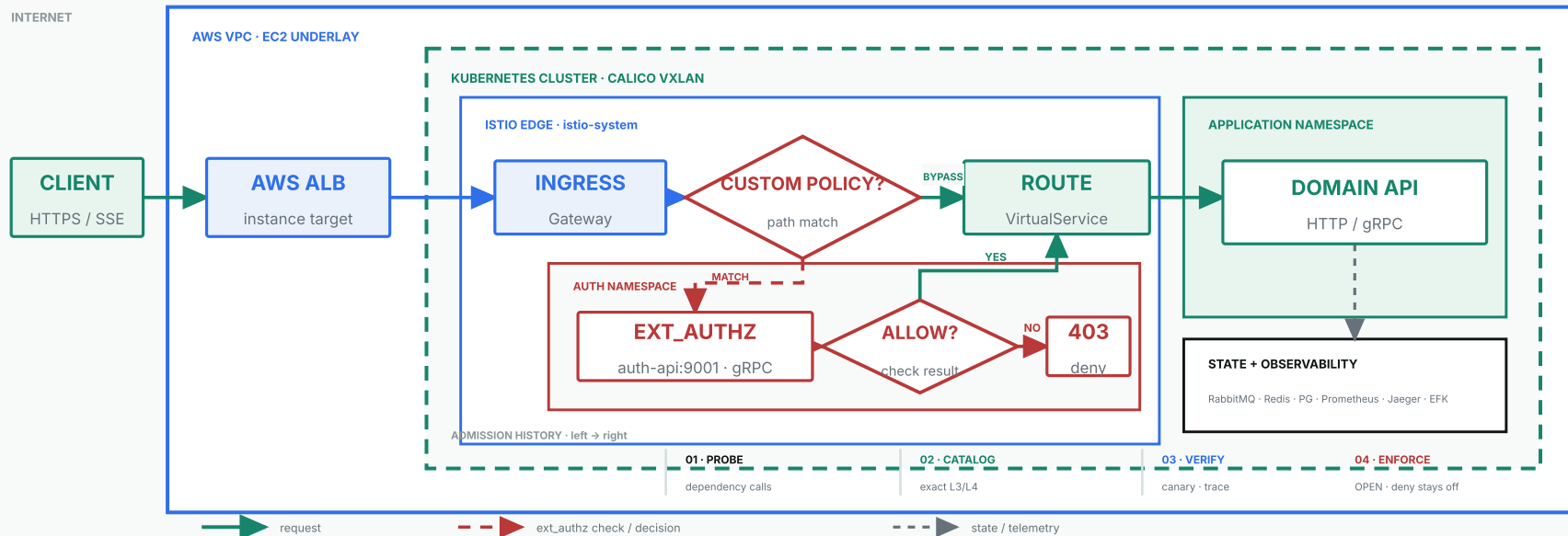
[Production Argo CD apps](#) · [Infrastructure validation](#) · [Argo sync waves](#) · [CNCF · configuration drift](#)

승격 권한 · 잔존 경계

재귀 수렴은 drift를 복구하지만 child catalog와 sync-wave 간격을 함께 운영해야 합니다. 값 이동은 dependency와 rollout 순서를 바꾸므로 실제 prerequisite만 남습니다.

Allow catalog를 정제했지만 default-deny enforcement는 보류했습니다

Client request는 ALB·Istio ingress를 지나 CUSTOM policy 대상만 ext-authz 판정을 받습니다. 별도의 default-deny probe는 DNS·DB·gRPC·event·telemetry 같은 내부 호출 경로를 드러냈고 확인된 의존성을 L3/L4 allow catalog로 정제했습니다. 그러나 live canary·packet trace가 달하지 않아 전면 default-deny는 비활성으로 유지했습니다.



문제 · 판정 질문

Reachability, external authorization, internal dependency는 서로 다른 owner와 실패를 갖습니다. 한 정책으로 동시에 강화하면 403과 통신 단절의 원인을 잃으므로 관측→exact rule→canary/trace 순으로 admission을 나눴습니다.

근거 · TEST / RUN ARTIFACT

[Istio Gateway](#) · [CUSTOM AuthorizationPolicy](#) · [Extension provider](#) · [Istio custom authz](#)

승격 권한 · 잔존 경계

Figure는 current manifest의 desired route입니다. 실제 packet path-latency, FQDN enforcement, 전면 default-deny는 OPEN이며 해당 evidence가 생길 때만 enforcement 후보로 승격합니다.

Eco²는 비동기 완료를 복구했고, 미결 관측은 다음 runtime의 state rule로 남겼습니다

Eco²에서 실행·이력·client terminal·platform controller의 권한은 구현 경계로 담았습니다. 반면 trace parentage, branch-local evidence, evaluation assembly, live enforcement는 current pin의 OPEN입니다. 이 구분을 유지한 채, 동일 관측이 다른 다음 행동을 요구할 때 paired history로 최소 separating field를 찾는 분석 규칙을 GEODE의 trajectory admission으로 전이합니다.

01

CLOSED IN ECO²

ASYNC HANDOFF

API deadline 뒤 job identity가 실행을 인계

ORDERED RECOVERY

Streams·PEL·Last-Event-ID가 누락 이력을 보존

CLIENT TERMINAL

reconnect·polling이 committed / failed를 복원

CONTROL OWNERS

revision·replica·route를 서로 다른 controller가 수렴

02

OPEN AT PIN

TRACE PARENTAGE

queue labels={} · API→worker join 미완

BRANCH BINDING

decomposed query→branch evidence 미결속

EVAL TRAJECTORY

parent signature에서 실행 assembly 중단

LIVE ENFORCEMENT

default·deny·p99·matched causal run 미측정

03

TRANSFER RULE · PROPOSED

같은 projection이 서로 다른 다음 action을 요구할 때만 state를 세분화합니다.

h₁ · terminal applied → STOP

h₂ · receipt missing → REPLAY / POLL

paired history를 같은 contract에서 재생하고 action을 가르는 최소 field만 후보 projection에 추가합니다.

후보 · job_id · seq · stream_id · client status

비용 · serialization · migration · storage · fallback traffic

문제 · 판정 질문

구현으로 닫힌 경계, current pin에서 재현된 결손, 다음 system에 적용할 분석 규칙을 같은 어휘로 섞지 않습니다.

근거 · TEST / RUN ARTIFACT

[Pinned backend](#) · [POMDP](#) · [Kaelbling et al.](#) · [State abstraction](#) · [AAAI](#) · [ReAct](#) · [ICLR 2023](#)

승격 권한 · 잔존 경계

POMDP·state abstraction·ReAct는 분석 배경입니다. Eco²는 POMDP solver·belief state·RL policy·projection·admission loop를 구현하지 않았습니다. Transfer rule은 proposed analytical framing입니다.

References

구현 질문은 pinned code, 성능 질문은 run artifact, 선택 배경은 dated narrative, 개념 해석은 external frame에서 시작합니다. 각 장의 footer는 여기서 고른 원본으로 직접 연결됩니다.

	SOURCE TYPE	PRIMARY REFERENCES	OWNS	DOES NOT PROVE
01	CODE AUTHORITY	Backend pinned tree · Frontend pinned tree · CHANGELOG	class-function-manifest-test wiring	live health-traffic는 별도 run이 소유
02	RUN ARTIFACT	VU1000 run · Grafana · pre-gevent · Grafana · VU1000 · Test-quality audit	terminal-latency-queue-coverage-complexity	unmatched run 사이의 인과는 OPEN
03	DATED NARRATIVE	Public portfolio · Engineering chronology · Debugging context · Pinned commit	문제 관측·선택 이유·변경 순서	당시 기록이 current wiring를 덮지 않음
04	EXTERNAL FRAME	Trustworthy Agents · OpenAI agent guide · ACES · POMDP · State abstraction · ReAct	agent-evaluation-state-refinement vocabulary	외부 연구는 Eco ² 구현 증거가 아님

READING ORDER

Code pin → run artifact → human/CI decision · dated narrative와 external frame은 이유와 해석을 보강하고 실행 사실을 대체하지 않습니다.