THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA
Embedded Linux Conference

KUBERCA

Troubleshooting Like a Senior on Day 1
: ReAct Agents With Real-Time Cluster Evidence

From Recovery to Knowledge.

Speaker

Taeji Kim

DevSecOps

Bohyun Choi

NVIDIA SW Engineer

Woobin Hwang

DevOps & Blockchain Engineer

Agenda

1. Overview
2. KubeRCA at a Glance: Architecture and Operator Experience
3. Tech Deep Dive: From Alert Intake to Root Cause Analysis
4. Outcomes & Expected Impact

The 30-Minute Gap

Alert fires

Manual context gathering across tools

K8s | Prometheus | Loki | Tempo | Slack

Operators jump between consoles, copy logs, run kubectl, then write up findings.

RCA

Root cause Analysis

Tens of minutes to hours pass before an operator truly understands what broke.

What Changes When We Solve It

OPERATOR TIME

Spends 30 min collecting evidence

KNOWLEDGE

Past incidents stay in heads

REPORTING

Each RCA written from scratch

AFTER KUBERCA

Spends time on judgment, MTTR down

AFTER KUBERCA

Searchable history seeds new responses

AFTER KUBERCA

Standard format → cheap retrospectives

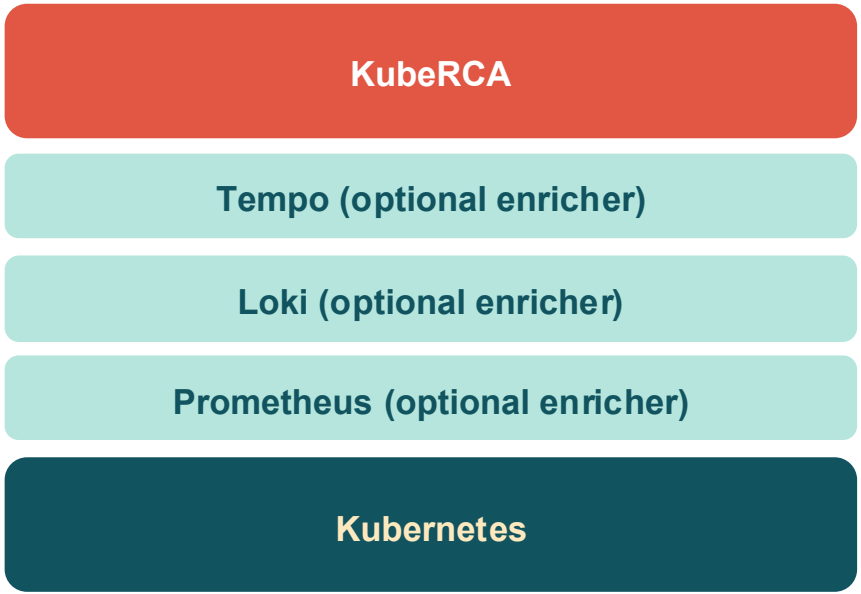
Automation moves operators from "gathering" to "deciding".

Project Goal

> *Close every step from the moment Alertmanager fires to the moment an RCA lands in a Slack thread, JUST inside ONE system.*

One pipeline · One context · One operator interface.

Layer On Top — Don't Replace



How graceful degradation works

Each Agent response includes a capabilities field:

k8s_core: ok
manifest_read: ok
prometheus: unavailable
loki: ok
tempo: unavailable
mesh_type: istio
routing_evidence: virtualservice

Prompt policy_block (literal):

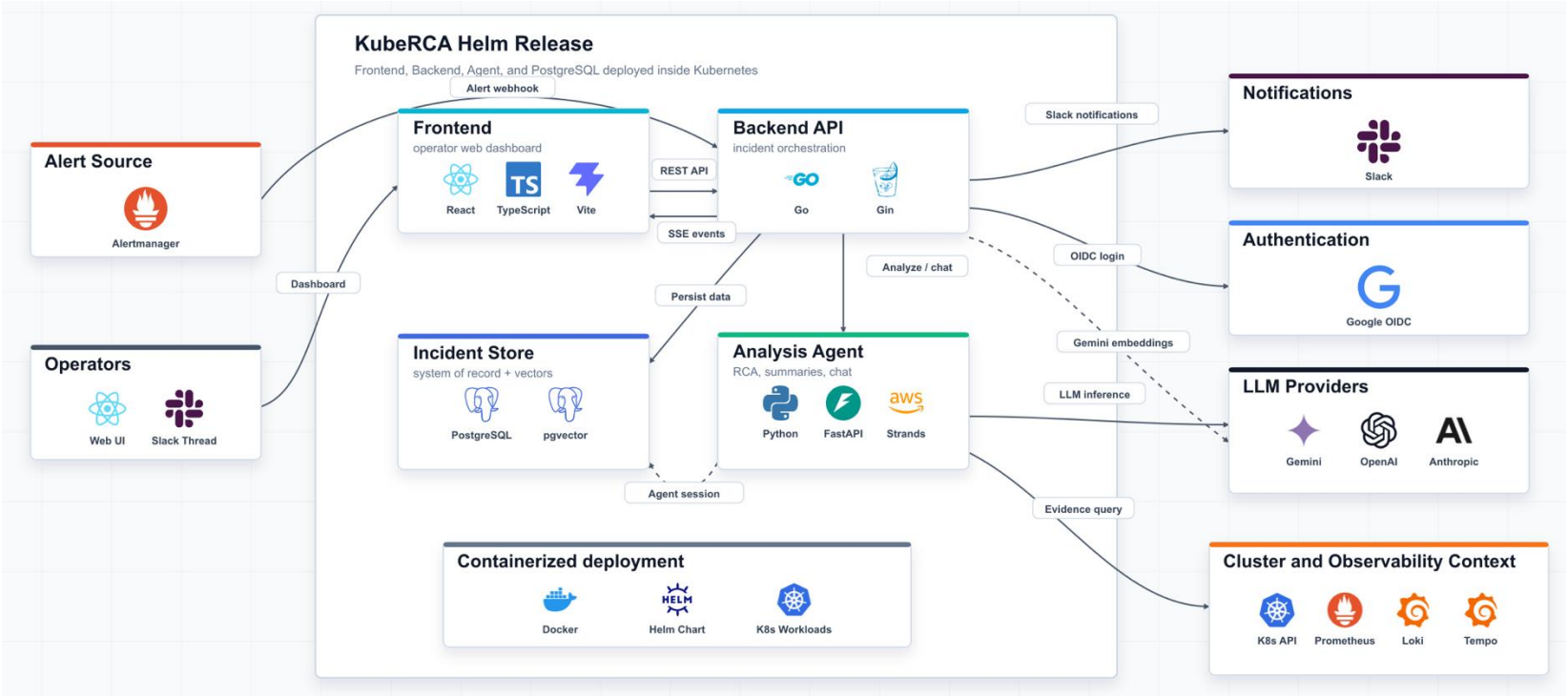
"Do not mention Prometheus, Loki, Tempo, or Istio as action items when the corresponding capability is unavailable."

PREVIEW

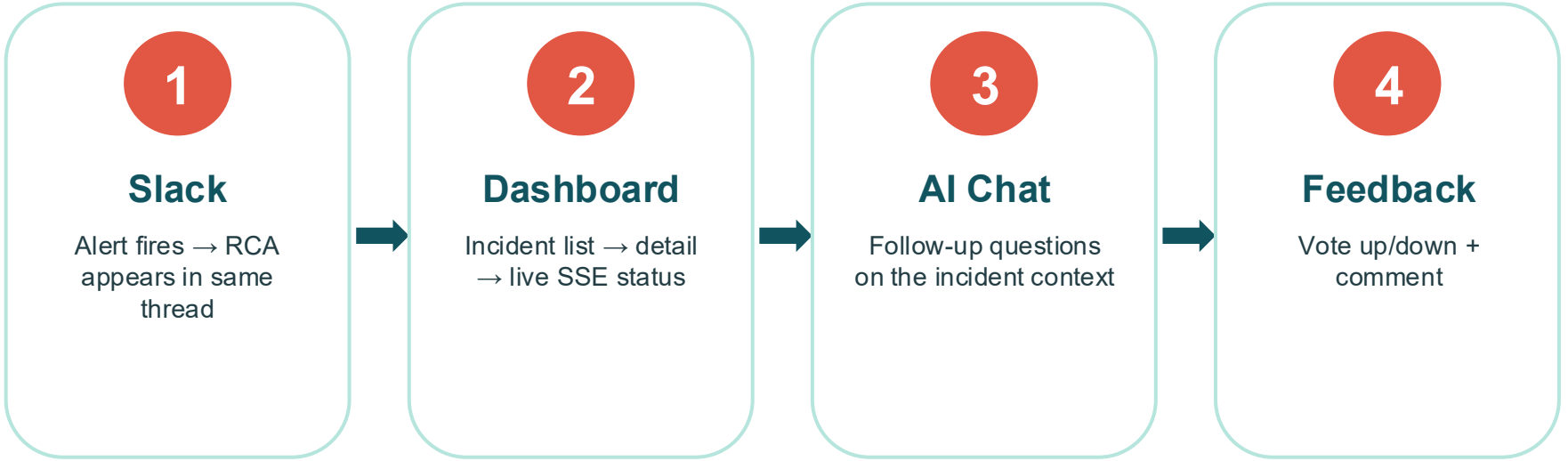
KubeRCA at a Glance

OPEN SOURCE SUMMIT NORTH AMERICA
Embedded Linux Conference

System Architecture at a Glance



Live Demo · 3-Minute Walkthrough



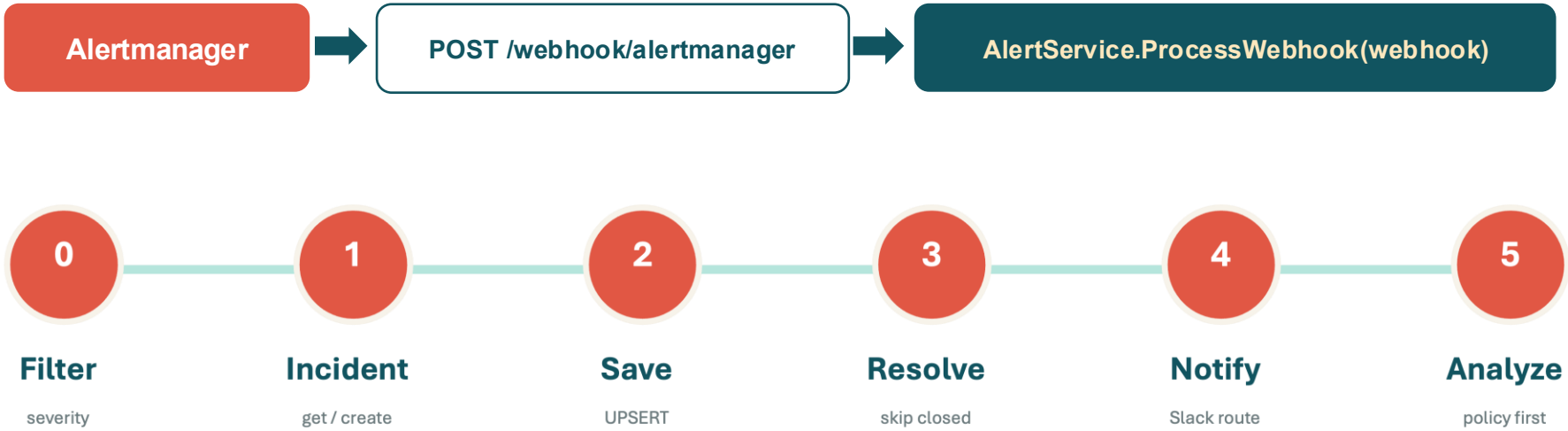
Same incident context flows through every step.

TECH DEEPDIVE 1

Alert Intake & Incident Model

OPEN SOURCE SUMMIT NORTH AMERICA
Embedded Linux Conference

Alert Intake Pipeline



rc

```
Context: kkamji-lab [RW]                <0> all
Cluster: cluster.local                  <1> external-se
User:    kubernetes-admin               <2> kube-rca-ch
K9s Rev: v0.50.18                       <3> openclaw
K8s Rev: v1.35.1                        <4> bookinfo
CPU:     20%                            <5> play-hub
MEM:     62%
```

pods(bookinfo)[7]

NAME↑	PF	READY	STATUS	RESTARTS	CPU	%CPU/R	%CPU/L	MEM	%MEM/R	%MEM/L	IP
details-v1-59c4d968dd-jf2bf	●	2/2	Running	0	107	71	26	62	32	8	10.233.64.54
load-generator-86bdf48648-vcpj7	●	2/2	Running	0	129	73	18	158	82	15	10.233.65.12
productpage-v1-5ddd7d89b5-9h4hr	●	2/2	Running	0	367	244	30	687	358	44	10.233.64.170
ratings-v1-c5b99f77f-vsjc4	●	2/2	Running	0	48	32	16	76	39	9	10.233.64.114
reviews-v1-769478b674-hppzj	●	2/2	Running	0	20	13	5	154	80	15	10.233.65.120
reviews-v2-bbbb5dc84-k2fdl	●	2/2	Running	0	46	30	11	174	91	17	10.233.65.24
reviews-v3-5b4ff46c8b-8jqrf	●	2/2	Running	0	43	28	10	166	86	16	10.233.65.119

<pod>

```
~/c/c/bun-terraform
❯
```

kkamji-lab (kube-rca) personal:kr

BookInfo Dashboard - Demo

Kube-RCA

grafana.kkamji.net/d/bookinfo-red-use/bookinfo-dashboard?orgId=1&from=now-5m&to=now&timezone=bro...

Home › Dashboards › Demo-App › BookInfo Dashboard

Search... ⌘+k

Edit Export Share

Datasource: Prometheus | Namespace: bookinfo | Service: All | Last 5 minutes | Refresh 5s

Overview

Total Request ...	Error Rate %	P95 Latency	Pod Restarts (1h)	Running Pods	Failed Pods
313 req/s	0%	21.9 ms	5.02	7	0

Golden Signals (RED)

Request Rate by Service

Name	Mean	Max
details	96.8 req/s	98.3 req/s
productpage	72.7 req/s	75 req/s
ratings	71.9 req/s	73.5 req/s

Latency Distribution (P50/P95/P99)

Name	Mean	Max
P50	2.81 ms	2.87 ms
P95	21.9 ms	21.9 ms
P99	24.5 ms	24.5 ms

Error by Destination

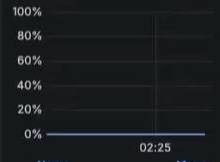
5xx Rate by Dest

Name	Mean
503 - details	0%
503 - productpage	0%
503 - ratings	0%

5xx Count by Dest

Name	Total
503 - details	0
503 - productpage	0
503 - ratings	0

4xx Rate by Dest



Name	Mean
404 - productpage	0%
405 - productpage	0%
408 - ratings	0%

4xx Count by Dest

Name	Total
404 - productpage	0
405 - productpage	0
408 - ratings	0

Error by Source

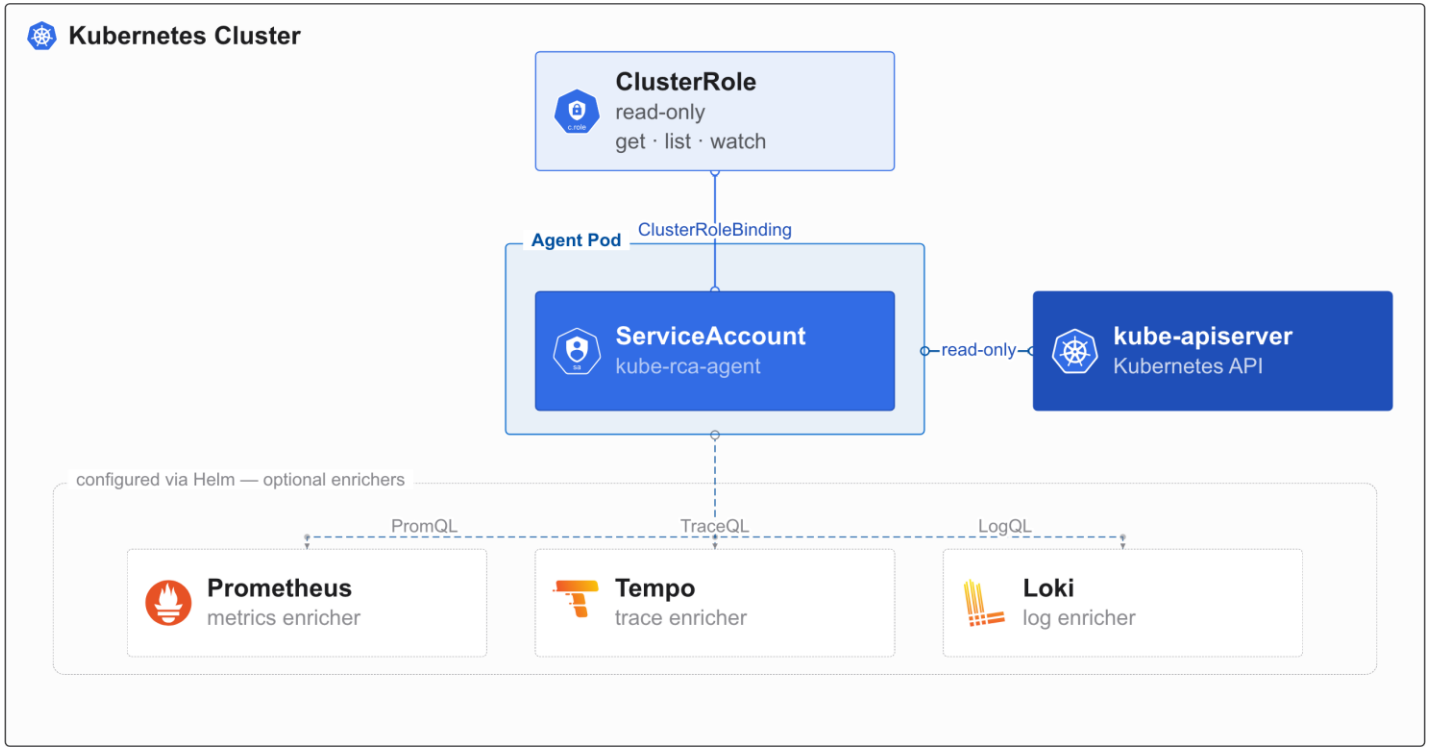
5xx Rate by Source | 5xx Count by Source | 4xx Rate by Source | 4xx Count by Source

TECH DEEPDIVE 2

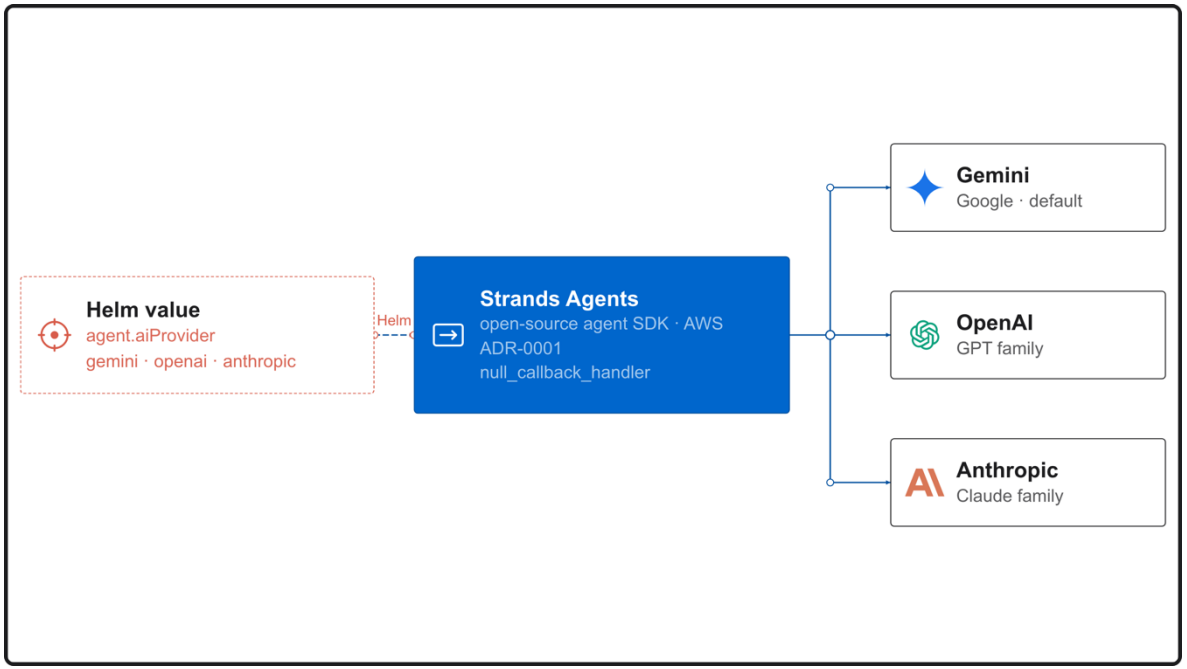
Context Collection Strategy

OPEN SOURCE SUMMIT NORTH AMERICA
Embedded Linux Conference

Where Does the Agent Live?



Strands Agents — open-source SDK



What this absorbs

Provider abstraction

agent.aiProvider · no image rebuild

Tool-use ergonomics

First-class tool calls · no boilerplate

Lightweight footprint

Minimal deps · fast pod start

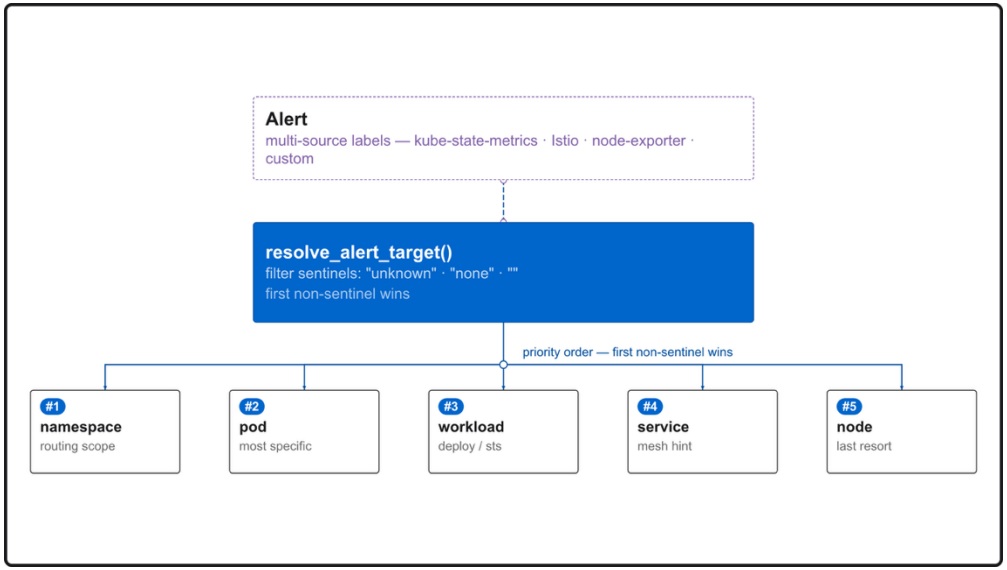
Tool Registry — Base + Dynamic Enrichers

Layer	Component	Gate	If disabled
base	K8s core	always registered	—
dynamic	Istio	agent.istio.enabled	tools removed from prompt
dynamic	Prometheus	non-null client	tools removed from prompt
dynamic	Tempo	non-null client	tools removed from prompt
dynamic	Loki	non-null client	tools removed from prompt

The prompt rewrites itself.

Unavailable enrichers disappear from the prompt entirely — so the LLM never produces useless advice like "run a PromQL query" when no Prometheus is connected.

Alert Routing & Operator Surface



Operator-facing surface

analysis_quality: high

analysis_quality: medium

analysis_quality: low

missing_data tracks only what the agent failed to fetch even with tools — "expected absences" stay out.

TECH DEEPDIVE 3

Agent / LLM Inference Pipeline

OPEN SOURCE SUMMIT NORTH AMERICA
Embedded Linux Conference

Multi-LLM Provider

Provider	Helm value	If API key missing
Anthropic	"anthropic"	→ Gemini default
Gemini	"gemini" (default)	(always)
OpenAI	"openai"	→ Gemini default

null_callback_handler

Mandatory for non-Gemini providers. Keeps K8s pod logs clean — the default Strands handler writes to stdout and corrupts structured pod logging.

Honesty: Helm flip = pod restart, not runtime auto-routing. Provider swap is operator-initiated, never silent.

Agentic Loop — 4 phases + self-loop

1 Observe

Receive alert payload + load dispatch context

2 Hypothesize

Form RCA candidate from current evidence

3 Tool-call

Fetch more evidence; loop back to Hypothesize if needed. Strands SDK manages iterations.

4 Synthesize

Final answer → DB · SSE · Slack fan-out. KubeRCA caps total time + memory only — no max_iterations knob.

Prompt Evolves Through Chaos

Test bed	Bookinfo + load generator; no optimal prompt on first try.	**Chaos mix**	9 scenarios: 2 Chaos Mesh · 5 Istio · 2 kubectl.
Chaos Mesh	OOMKilled · network delay.	**Istio faults**	404 · 500 · 503 · 504 · ratings-multi.
kubectl	CrashLoopBackOff · ImagePullBackOff.	**Agent run**	Analyze with tools; read output for evidence gaps.
Prompt diff	Update prompt, rerun the same scenario, compare.	**Principles**	Context-grounded · structured tool-use · least-words · evidence honesty.

The four principles are loop output — not design-time decisions.

Model Selection: Operational Trade-off

Fast model

Good for summary, chat, and simple follow-up. Lower latency and cost per call.

Balanced model

Good default for normal RCA workflows. Adequate reasoning without excessive latency.

High-reasoning model

Useful for complex incidents with many correlated signals. Higher cost per call.

This is a conceptual framework, not a benchmark result.

What the choice optimizes

Speed · Reasoning Quality · Cost

Balance all three for RCA. One axis alone is not enough.

Latency stacks

Higher-tier models can add significant latency across multiple tool-call rounds.

Defaults one tier down

Helm flip works today. Not tied to a specific model.

Honesty

Conceptual framework, not a benchmark result.

Token Pre-LLM Cuts

Before the LLM call, shrink context without losing target signal

1

Tool output line caps
Logs: 25 lines; events: 25 by default. Tunable per scenario via Helm.

2

Alert label scope
namespace / pod / workload narrows tool calls to the affected target only.

3

Resolved-alert halve
Logs 25 -> 12; events 25 -> 12. Tempo pivots from fires_at to ends_at.

Retry asymmetry

/analyze gets Tenacity retry.

/summarize and /chat are single-shot.

RCA is worth retrying. Chat is not.

Previous firing analysis is already in the prompt; duplicate raw context wastes tokens.

Token Budget Defense

5-stage progressive drop, when the cap still would not fit

Trigger: prompt > 32K tokens after pre-LLM cuts (~128K chars with conservative 4 chars/token).

Stage 1 Drop Tempo context	Stage 2 Drop previous logs	Stage 3 Drop all logs	Stage 4 Drop events	Stage 5 Compact fallback
Trace detail is expensive and least durable.	Prior firing evidence is summarized already.	Keep higher-level evidence and metadata.	Events go after logs and traces.	Keep five fields and final minimal context.

Least-valuable data drops first. This is the last guard before the LLM call.

TECH DEEPDIVE 4

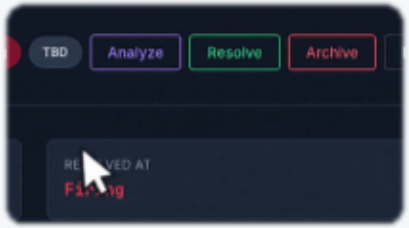
Semantic Knowledge Layer

OPEN SOURCE SUMMIT NORTH AMERICA
Embedded Linux Conference

Incident Summary — Async Pipeline

01 Trigger Summary

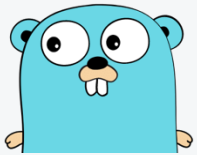
POST /incidents/:id/resolve
HTTP 200 returned



02 Async Goroutine

requestIncidentSummary()
Decouples LLM from API

Client never blocked by LLM latency



03 AI Agent Call

Builds prompt from context
Alerts + analyses + artifacts

5-min TTL dedup map prevents retry storms

04 Save Summary

DB UPDATE first
Embedding is best-effort

Fallback summary if LLM fails, failure can't lose data

Embedding & Similar-Incident Search

Embedding — Backend calls Gemini SDK directly

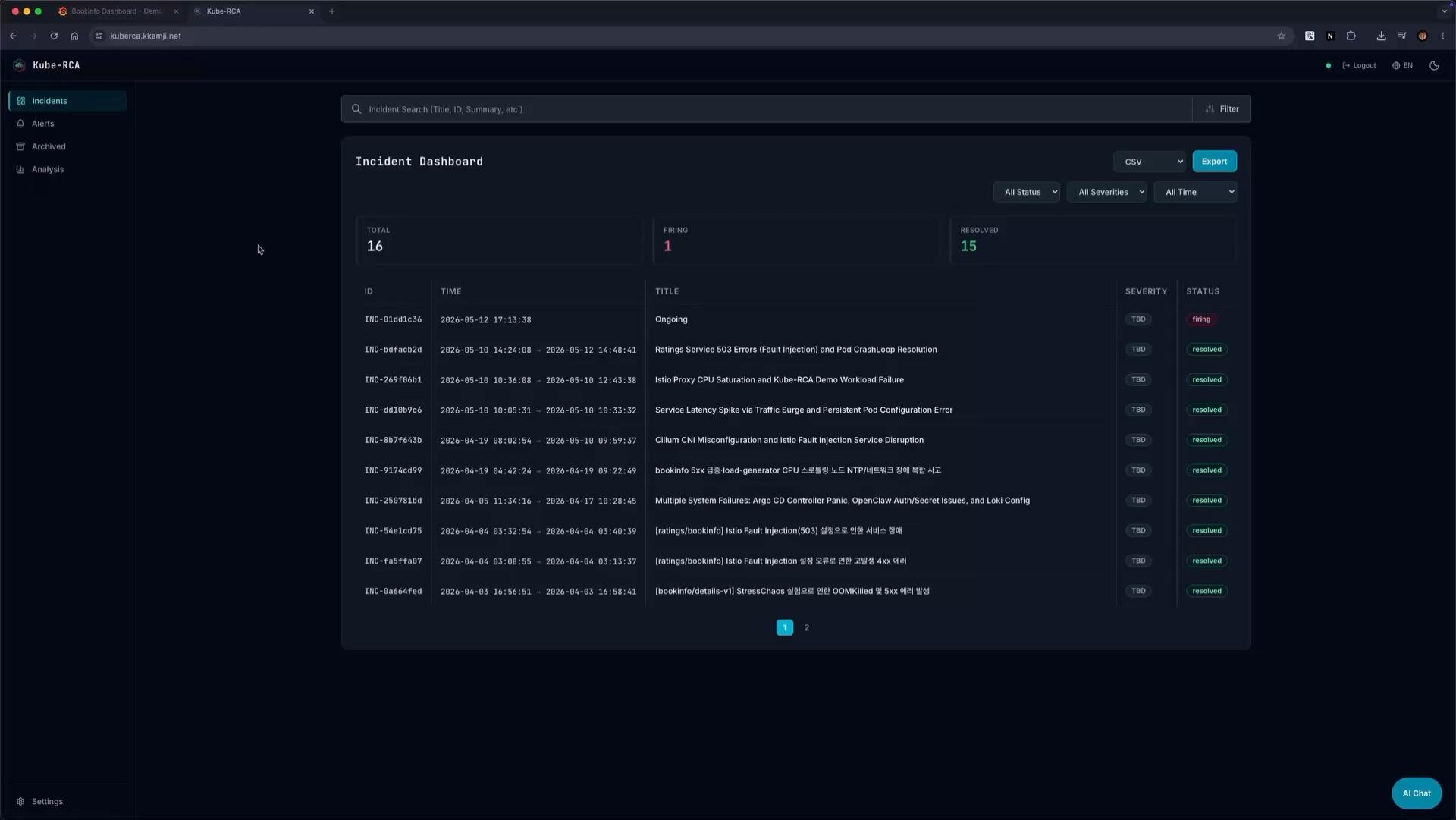
db/embedding.go

```
CREATE TABLE embeddings (
  id                  BIGSERIAL PRIMARY KEY,
  incident_id         TEXT NOT NULL,
  incident_summary    TEXT NOT NULL,
  embedding           vector(3072),
  model               TEXT NOT NULL,
  created_at          TIMESTAMPTZ
);
```

- 3072-dim Gemini embedding
- model column = traceability across migrations
- CREATE EXTENSION vector at boot — fail fast

Search — Backend re-embeds query, Top-K cosine

1. Query text ("pod CrashLoopBackOff")
2. EmbedText() — same model as index
3. `SQL: ORDER BY embedding <=> $1 LIMIT K`
4. Return similarity = 1 − cos_distance

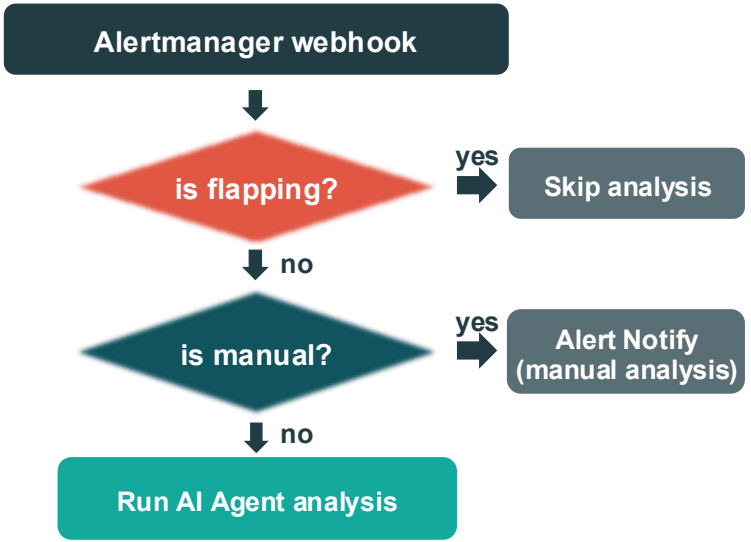

Bookinfo Dashboard - Demo
Kube-RCA
kuberca.kkamji.net
Kube-RCA
Logout
EN
Incidents
Alerts
Archived
Analysis
Settings
Incident Search (Title, ID, Summary, etc.)
Filter
Incident Dashboard
CSV
Export
All Status
All Severities
All Time
TOTAL
16
FIRING
1
RESOLVED
15
ID
TIME
TITLE
SEVERITY
STATUS
INC-01dd1c36
2026-05-12 17:13:38
Ongoing
TBD
firing
INC-bdfacb2d
2026-05-10 14:24:08 → 2026-05-12 14:48:41
Ratings Service 503 Errors (Fault Injection) and Pod CrashLoop Resolution
TBD
resolved
INC-269f06b1
2026-05-10 10:36:08 → 2026-05-10 12:43:38
Istio Proxy CPU Saturation and Kube-RCA Demo Workload Failure
TBD
resolved
INC-dd10b9c6
2026-05-10 10:05:31 → 2026-05-10 10:33:32
Service Latency Spike via Traffic Surge and Persistent Pod Configuration Error
TBD
resolved
INC-8b7f643b
2026-04-19 08:02:54 → 2026-05-10 09:59:37
Cilium CNI Misconfiguration and Istio Fault Injection Service Disruption
TBD
resolved
INC-9174cd99
2026-04-19 04:42:24 → 2026-04-19 09:22:49
bookinfo 5xx 급증·load-generator CPU 스로틀링·노드 NTP/네트워크 장애 복합 사고
TBD
resolved
INC-250781bd
2026-04-05 11:34:16 → 2026-04-17 10:28:45
Multiple System Failures: Argo CD Controller Panic, OpenClaw Auth/Secret Issues, and Loki Config
TBD
resolved
INC-54e1cd75
2026-04-04 03:32:54 → 2026-04-04 03:40:39
[ratings/bookinfo] Istio Fault Injection(503) 설정으로 인한 서비스 장애
TBD
resolved
INC-fa5ffa07
2026-04-04 03:08:55 → 2026-04-04 03:13:37
[ratings/bookinfo] Istio Fault Injection 설정 오류로 인한 고발생 4xx 에러
TBD
resolved
INC-0a664fed
2026-04-03 16:56:51 → 2026-04-03 16:58:41
[bookinfo/details-v1] StressChaos 실험으로 인한 OOMKilled 및 5xx 에러 발생
TBD
resolved
1
2
AI Chat

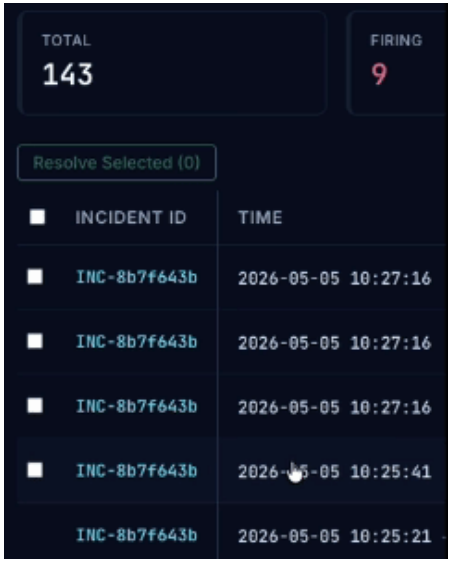
TECHDEEPDIVE 5

Convenience & Security

OPEN SOURCE SUMMIT NORTH AMERICA
Embedded Linux Conference

Manual / Auto Analysis Policy





Alert Bulk Resolve

LLM analysis	Alert notify
OFF	ON

- Default — all severities auto
- **Resolved waits up to 3 min for in-flight firing analysis**

4-Layer Masking — Defense in Depth

Built-in-Redactor

L1 Key Denylist	L2 Value Heuristic	L3 k8s-aware
password, token, secret, api_key… allowlist beats it	JWT, Bearer, ?token=, validated base64 ≥16B	env values, --password=, annotations

Regex-Masker

L4 User Regex
Org-specific patterns, validated at boot

Hash Mode — [MASKED] vs [HASH:xxxxxxxx]

[MASKED] — default

```
User A login fails:
  token=[MASKED]

User A retries:
  token=[MASKED]
```

Same? Different? Cannot tell.

[HASH:xxxxxxxx] — opt-in

```
User A login fails:
  token=[HASH:a3f1c0e2]

User A retries:
  token=[HASH:a3f1c0e2]
```

Same secret = same hash → repeat-failure pattern visible

⚠ **Tradeoff: weak secrets like "password123" yield well-known SHA-256 prefixes — rainbow-table lookup is feasible. Original is unrecoverable, but fingerprinting risk is non-zero.**

OUTCOMES

Outcomes & Expected Impact

Quantitative Outcomes

Mean Time to RCA

Before: 30–60 min

< 10 min

Mean Time to Recovery

Before: 1–4 hrs

< 5 min

Similar Incident Lookup

Before: 5–15 min

< 1 sec

Qualitative Effects

ON-CALL

First-response load drops

"Gathering" goes to zero — operators start from the judgment phase.
Less context-switching, less fatigue.

TEAM

Postmortems standardization

Every RCA arrives as Root cause / Evidence / Actions / Missing data.
Retros become comparison and tagging, not rewriting.

ORGANIZATION

Incident knowledge accumulates

Closed incidents auto-embed.
Next week's lookalike fires and last month's resolution surfaces in one second.

Questions & Contibutions Welcome.

Thank you

KubeRCA Github Repository

Troubleshooting Like a Senior on Day 1: ReAct Agents with Real-Time Cluster Evidence