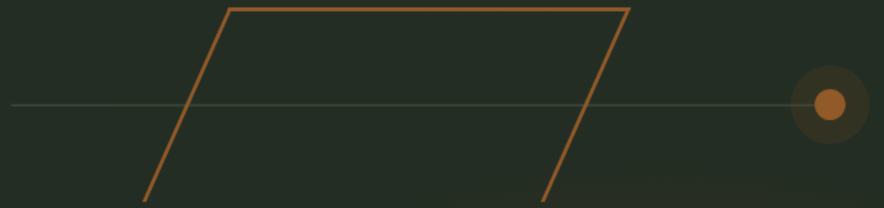
AI-POWERED INCIDENT INVESTIGATION

Evidence before action.

A governed DevOps investigation agent built with Oracle Private Agent Factory, Oracle Database, APEXLang, and APEX.



ACME RETAIL · SYNTHETIC INCIDENT 9001

INCIDENT RESPONSE TODAY

The evidence is everywhere. The decision is still yours.

01	02	03	04	05
Alerts	**Logs**	**Changes**	**Topology**	**Runbooks**
What crossed a threshold—and when?	Which error explains the symptom?	What moved before impact began?	Where did the failure propagate?	What action is actually approved?

ONE QUESTION STARTS THE INVESTIGATION

Checkout failures are elevated.

SEV-2 OPEN PRODUCTION

The operator knows the symptom—not the cause, the complete service scope, or the correct next step.

INCIDENT 9001	INITIAL EVIDENCE
Checkout failure rate	6.4%
Payment API 5xx rate	8.7%
Gateway p95 latency	2875 ms

SEPARATION OF RESPONSIBILITIES

The dashboard reads facts. The agent investigates them.

OPERATOR EXPERIENCE

Oracle APEX

Incident queue, dashboard, evidence detail, approval, and audit.

→

REASONING LAYER

Private Agent Factory

Scope discovery, tool orchestration, knowledge retrieval, and synthesis.

→

SYSTEM OF RECORD

Oracle Database

Synthetic incidents, alerts, deployments, dependencies, logs, and audit.

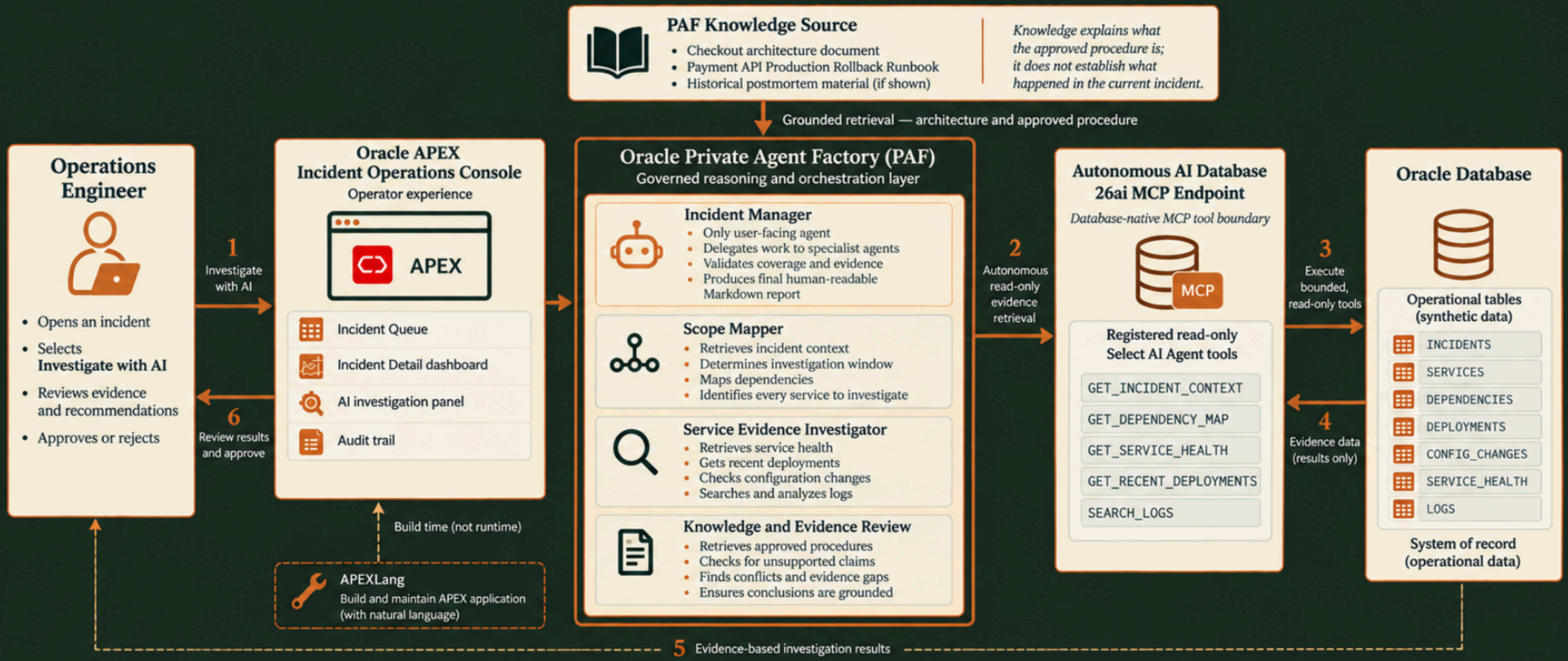
APEX → DB · direct SQL for deterministic UI

PAF → MCP → DB · autonomous read-only evidence collection

Governed Incident Investigation Architecture

APEX presents operational facts. PAF investigates them. Oracle Database remains the system of record.

OBSERVE
INVESTIGATE
UNDERSTAND
ACT WITH CONFIDENCE



Synthetic scenario. No production infrastructure, production credentials, customer data, or live observability system is involved.

A SMALL INVESTIGATION TEAM

One manager. Three focused responsibilities.

ONLY USER-FACING AGENT

Incident Manager

Delegates evidence collection, validates coverage, and delivers one human-readable incident brief.

01	**Scope Mapper**	Finds the time window and every service that must be investigated.
02	**Service Investigator**	Collects health, deployment, configuration, and log evidence per service.
03	**Knowledge + Review**	Grounds procedure in approved documents and challenges unsupported conclusions.

TOOLS THE AGENT CAN ACTUALLY USE

Bounded tools make the investigation auditable.

GET_INCIDENT_CONTEXT

Status, severity, impact, alerts, and investigation window.

GET_SERVICE_HEALTH

Current rolling health—kept distinct from incident status.

SEARCH_LOGS

Bounded log events with request and trace identifiers.

GET_DEPENDENCY_MAP

Direct dependencies, dependents, and criticality.

GET_RECENT_DEPLOYMENTS

Versions, configuration changes, and rollback target.

OPERATIONAL BOUNDARY

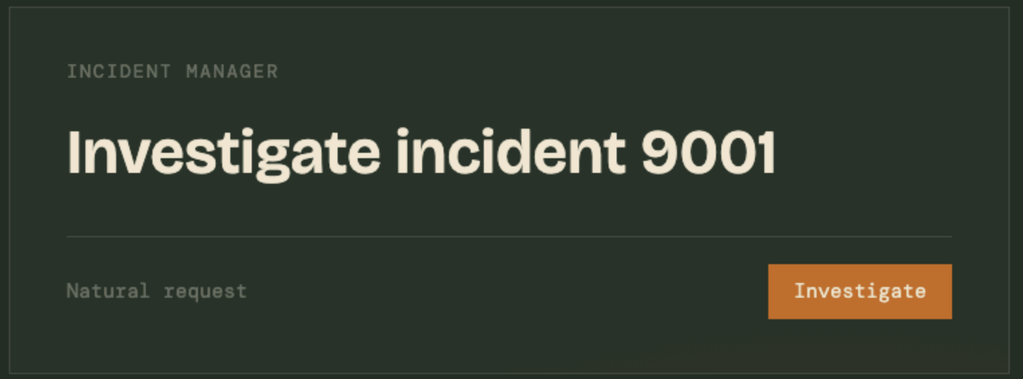
Read first. Recommend second. Act only with approval.

The webinar tools use deterministic synthetic data and cannot reach production infrastructure.

NO ANSWER HIDDEN IN THE PROMPT

Ask the incident, not the workflow.

The agent discovers the time window, affected services, evidence sources, and knowledge it needs.



- ✓ Scope discovery
- ✓ Parallel service evidence
- ✓ Knowledge grounding
- ✓ Evidence-linked synthesis

THE EVIDENCE FORMS A SEQUENCE

Fourteen minutes from change to customer error.

T+00

16:21:23Z

Timeout reduced

2500 ms → 250 ms

`config-change:92101`

T+12

16:33:23Z

5xx alert

payment-api reaches 8.7%

`alert:9101`

T+13

16:34:23Z

Timeout logged

Request exceeds 250 ms

`log:9401`

T+14

16:35:23Z

HTTP 502

Authorization fails

`log:9402`

T+14

16:35:23Z

Checkout fails

Failure crosses service boundary

`log:9403`

TIMING SUPPORTS THE HYPOTHESIS · IT DOES NOT PROVE IT ALONE

LEADING HYPOTHESIS

A timeout tuned below reality.

The payment-api deployment reduced the gateway timeout by 90%, producing gateway timeouts, HTTP 502 responses, and downstream checkout failures.

CONFIDENCE

HIGH

Configuration evidence, explicit timeout logs, alert metrics, and a cross-service trace all agree. Upstream gateway telemetry remains unavailable.

BEYOND TEMPORAL CORRELATION

The failure crosses the service boundary.

PAYMENT-API · LOG:9402

PAYMENT_GATEWAY_TIMEOUT

Authorization failed after gateway timeout; returning HTTP 502.

```
request_id = req-7a92
trace_id = trace-10002
```

MATCHED IDENTIFIERS

CHECKOUT-UI · LOG:9403

CHECKOUT_PAYMENT_FAILED

Checkout could not complete because payment-api returned HTTP 502.

```
request_id = req-7a92
trace_id = trace-10002
```

APPROVED KNOWLEDGE

Payment API Production Rollback Runbook

- ✓ Incident identifies payment-api
- ✓ Known-good target is verified
- ✓ Deployment and runtime evidence are cited
- ○ Explicit human approval is required

THE CONTROL POINT

The agent can recommend. It cannot authorize.

For the webinar, approval records a simulated audit event. No production infrastructure is changed.

ROLLBACK 2026.09.09.3 → .2

REVIEW PROPOSAL

ORACLE PRIVATE AGENT FACTORY

Governed intelligence

Discovers scope, collects evidence, reasons across services, consults approved knowledge, and returns a cited incident brief.

→

APEXLANG + ORACLE APEX

Operational workspace

Turns that capability into an incident queue, investigation experience, evidence viewer, approval flow, and audit trail.

APEXLANG

Prompts to a *Native APEX* Application

Building with Codex, SQLcl MCP, and APEXlang.

THE BASIC SETUP

Two capabilities, *one workflow*

MCP supplies live environment access. The skill supplies the language and the rules. Codex coordinates both.

01 / SQLCL MCP

Environment access

Connect to Oracle Database, inspect schema truth, validate APEXlang, and import into APEX.

DATABASE · SCHEMA · APEX VALIDATION · IMPORT

02 / APEXLANG SKILL

Build guidance

A GitHub-distributed skill with declarative syntax, templates, compiler rules, and workflows.

SOURCE LANGUAGE · COMPILER TRUTH · TEMPLATES

03 / CODEX

Orchestration

Reads the prompt and skill context, calls MCP tools, and maintains the local application source.

PROMPT · TOOLS · SOURCE · HANDOFF

BUILD THE FOUNDATION

Build the *operational view*

Establish the project, prove the schema, and create the complete Page 1 experience.

- Inspect the real schema before generating—tables, columns, relationships, types, and values are authoritative.
- Build native KPIs, the Incident Queue, paired charts, an Evidence Timeline, and supporting reports.
- Compiler-check, live-validate, import, and hand off a known-good baseline.

~22

MINUTES · APPROXIMATE BUILD TIME

Why longer? Project setup, schema checks, query validation, and the complete dashboard composition happen here.



ADD CONTROLLED INTELLIGENCE

APEXlang is *not just UI*

It can define shared components—and virtually any APEX object—including the AI Agent and its server-side tool.

- Continue from the validated dashboard and preserve the existing page behavior.
- Define the AI Agent, the `call_api` tool, and one required `PROMPT` parameter.
- Return evidence through the server-side result path and validate the shared components.

~8

MINUTES · APPROXIMATE BUILD TIME

01 **Agent**
Instructions + AI service

02 **Tool**
call_api(PROMPT)

03 **Evidence**
Preserved in the result

EXPOSE THE CAPABILITY

Add the *assistant launcher*

A deliberately small final increment connects the validated agent to native APEX behavior.

- Add one circular, icon-only launcher fixed at the bottom-right of Page 1.
- Connect its click Dynamic Action to native `showAiAssistant`.
- Verify keyboard access, minimal CSS, agent reference, validation, and import.

~4

MINUTES · APPROXIMATE BUILD TIME

The platform owns the assistant dialog. The application adds only the entry point and native action.

OPEN_ASSISTANT ICON-ONLY BUTTON	→	Dynamic Action CLICK EVENT
→		showAiAssistant NATIVE APEX

READABLE DECLARATIVE SOURCE

What an *.apx file* looks like

P00001-HOME.APX

```
region incident-queue (
    name: Incident Queue
    type: interactiveReport
    source {
        location: localDatabase
        type: sqlQuery
        sqlQuery:
            ```sql
            select i.incident_id as INCIDENT_ID, ...
            ```
    }
    layout {
        sequence: 30
        slot: BODY
    }
)
```

01 · NAMED DECLARATION

A stable source-level identity for the region.

02 · NATIVE TYPE

interactiveReport
maps to the APEX component.

03 · SOURCE + LAYOUT

SQL and page placement remain visible together.

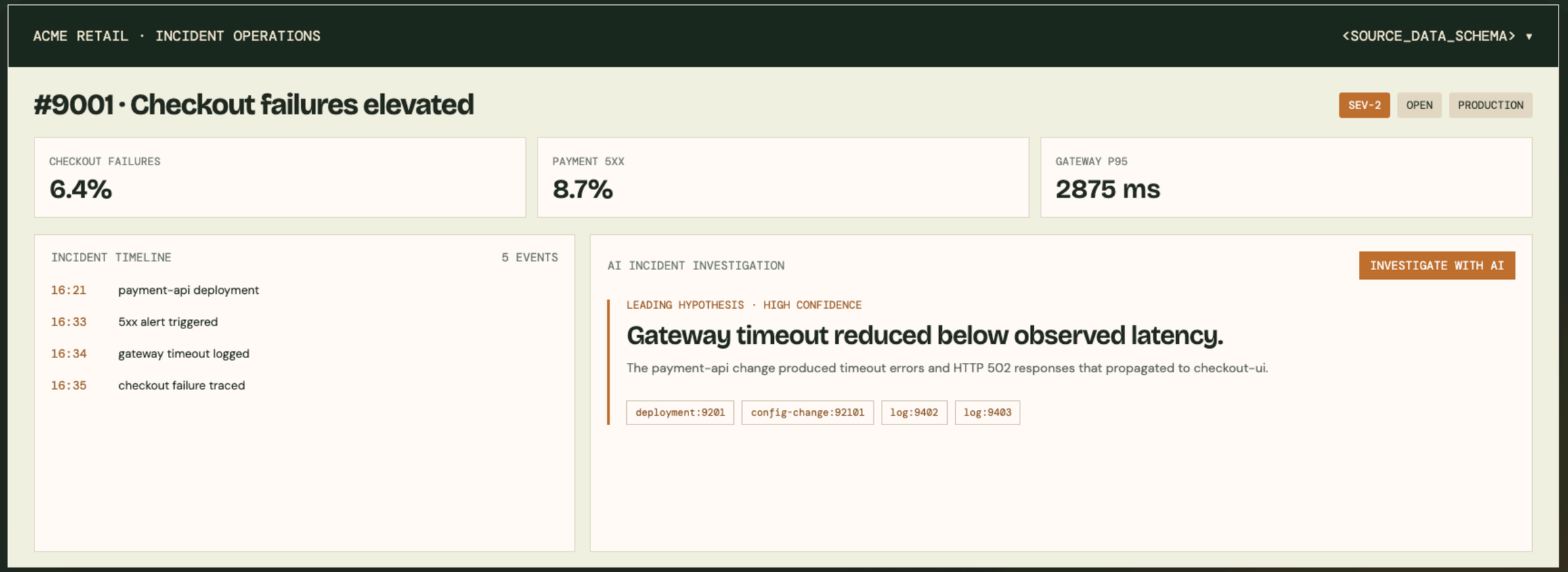
04 · CHECKED TWICE

Compiler truth, then live environment validation.

INCIDENT OPERATIONS CONSOLE

Not another chatbot.

The AI investigation lives inside the operator's incident workflow.



DASHBOARD DATA COMES DIRECTLY FROM ORACLE DATABASE

FOUR MOMENTS · ONE CONTINUOUS WORKFLOW

See it. Investigate it. Verify it. Control it.

01

Open the incident

APEX queries the operational system of record directly.

02

Investigate with AI

A server-side request sends incident context to PAF.

03

Inspect the evidence

Each citation opens its underlying database record.

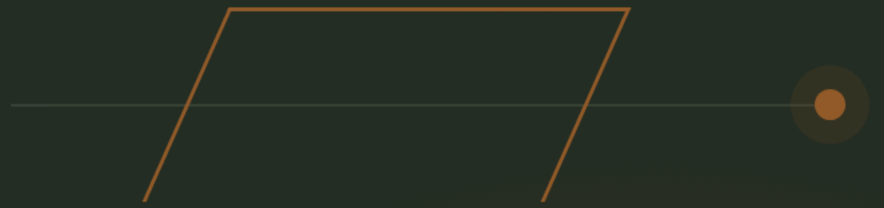
04

Review the proposal

The operator owns approval; the audit trail records the decision.

THE ENTERPRISE PATTERN

Faster reasoning. Visible evidence. Human control.



PAF supplies governed intelligence · **APEXLang + APEX** turn it into an operational experience.

EVIDENCE BEFORE ACTION