PRESIDIO®

Predictable Workflow vs. Dynamic LLM Agent

Automated Incident triage from alert to resolution.

Why Automate?

- **Noise:** 2 000 + alerts/day → on-call fatigue
- **Stakeholders:** SRE · DBA · Security · Comms
- **Success =**
  - Page the right human fast
  - Tell customers the truth
  - Archive a post-mortem
  - Preserve an audit trail—all within n8n

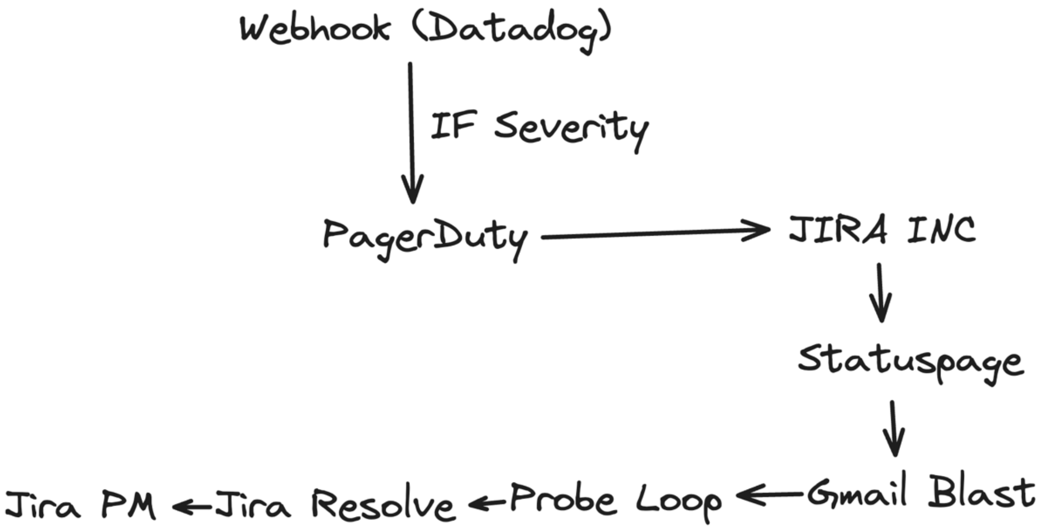
© Presidio, Inc. All rights reserved. Proprietary and Confidential.

Two Mind-Sets at a Glance

	Linear Workflow	LLM-Directed (Dynamic)
Control flow	Pre-wired edges on canvas	Generated at run-time from *THOUGHT/ACTION* lines
Decisions	IF / Switch nodes, regex	LLM reasons over alert + context
Change policy	Edit graph & redeploy	Edit prompt text only
Audit unit	Node execution log	Execution log **+** saved THOUGHT/ACTION transcript
Analogy	Conveyor belt	Smart switchboard

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

Linear Workflow



Webhook (Datadog)

IF Severity

PagerDuty

JIRA INC

Statuspage

Gmail Blast

Probe Loop

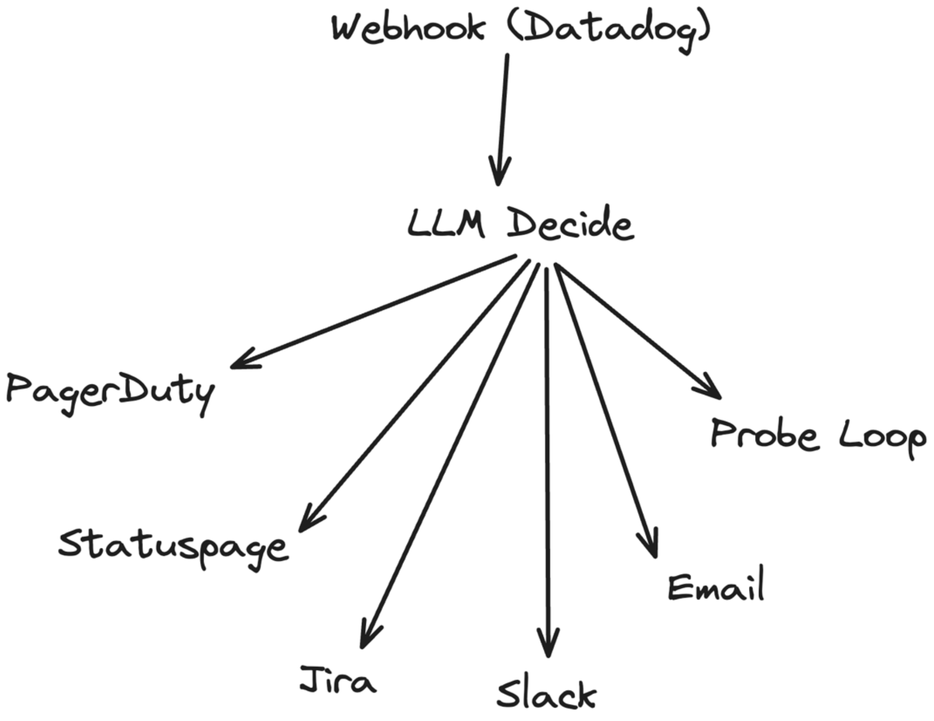
Jira Resolve

Jira PM

Strengths	Weaknesses
Deterministic; easy SOC-2	Regex & IF sprawl
< 50 ms latency	New channel = new branch
Low-code upkeep	Missed keyword = mis-route

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

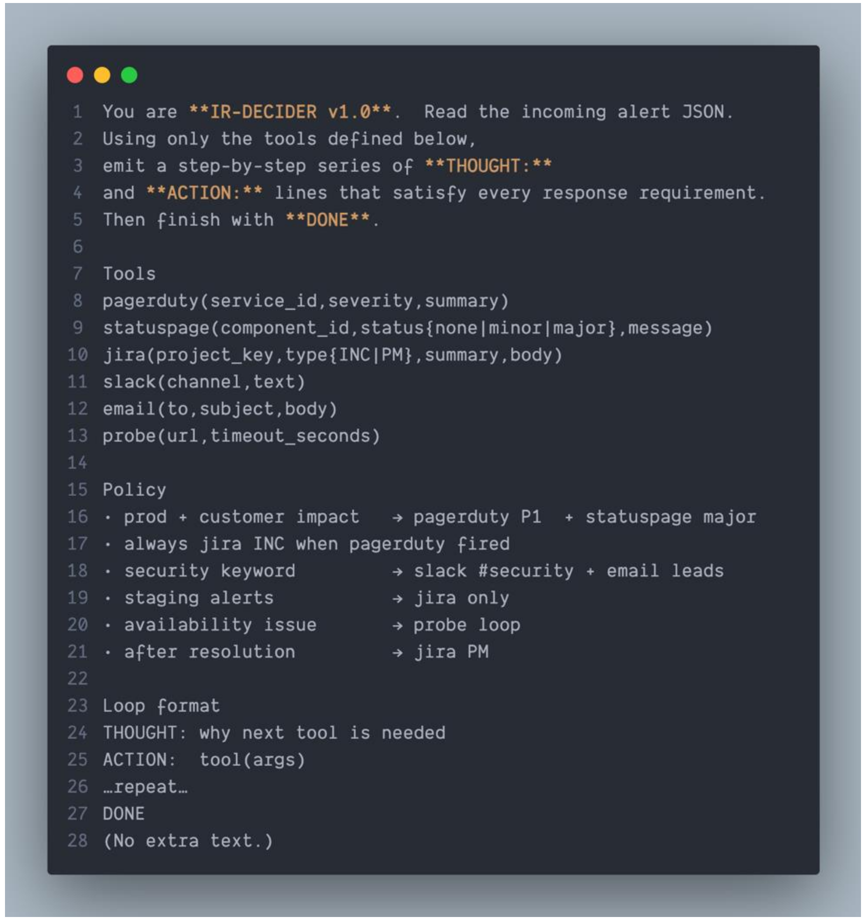
Dynamic LLM Flow



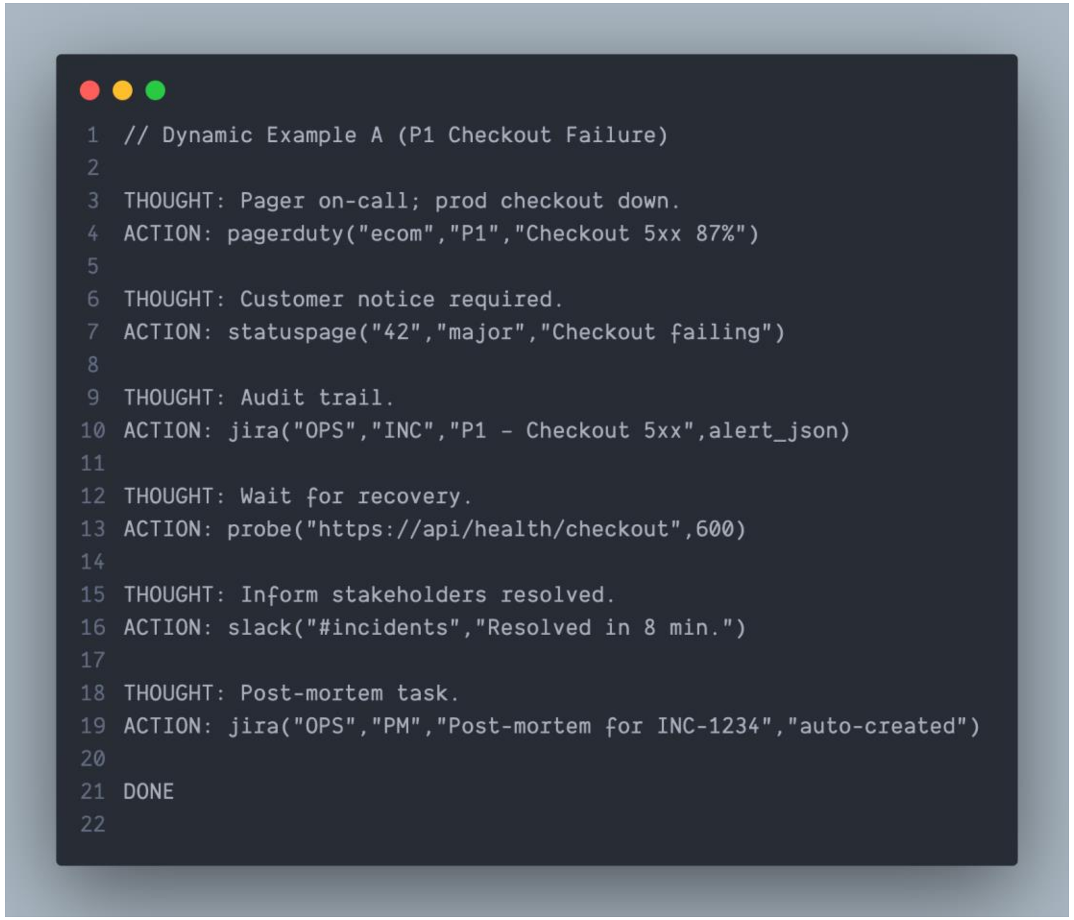
Strengths	Weaknesses
Prompt edits change logic instantly	Extra seconds of LLM thinking
Only a few nodes on the n8n canvas	Each alert costs LLM tokens
Reads context (prod / staging, VIP, etc.)	Must log thought/action for audits
Handles odd edge-cases without new branches	Poorly-written prompt can mis-route
One prompt = single source of truth	Team needs basic prompt-engineering skills

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

Example Prompt (THOUGHT / ACTION)



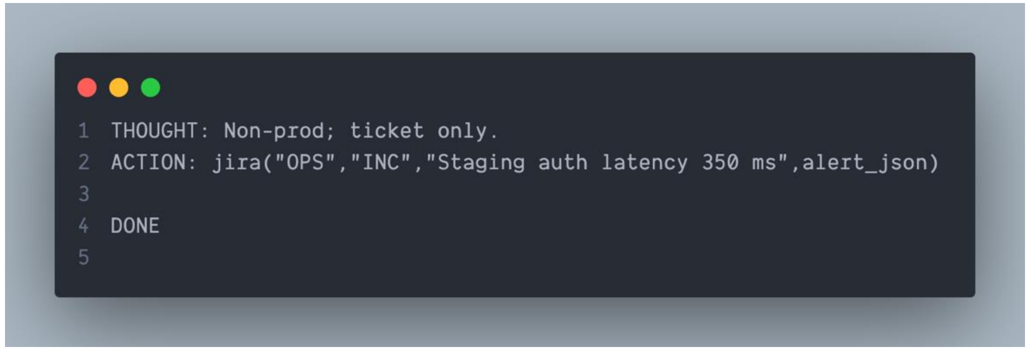
```
You are **IR-DECIDER v1.0**.  Read the incoming alert JSON.
Using only the tools defined below,
emit a step-by-step series of **THOUGHT:**
and **ACTION:** lines that satisfy every response requirement.
Then finish with **DONE**.

Tools
pagerduty(service_id,severity,summary)
statuspage(component_id,status{none|minor|major},message)
jira(project_key,type{INC|PM},summary,body)
slack(channel,text)
email(to,subject,body)
probe(url,timeout_seconds)

Policy
· prod + customer impact   → pagerduty P1  + statuspage major
· always jira INC when pagerduty fired
· security keyword         → slack #security + email leads
· staging alerts           → jira only
· availability issue       → probe loop
· after resolution         → jira PM

Loop format
THOUGHT: why next tool is needed
ACTION:  tool(args)
…repeat…
DONE
(No extra text.)
```



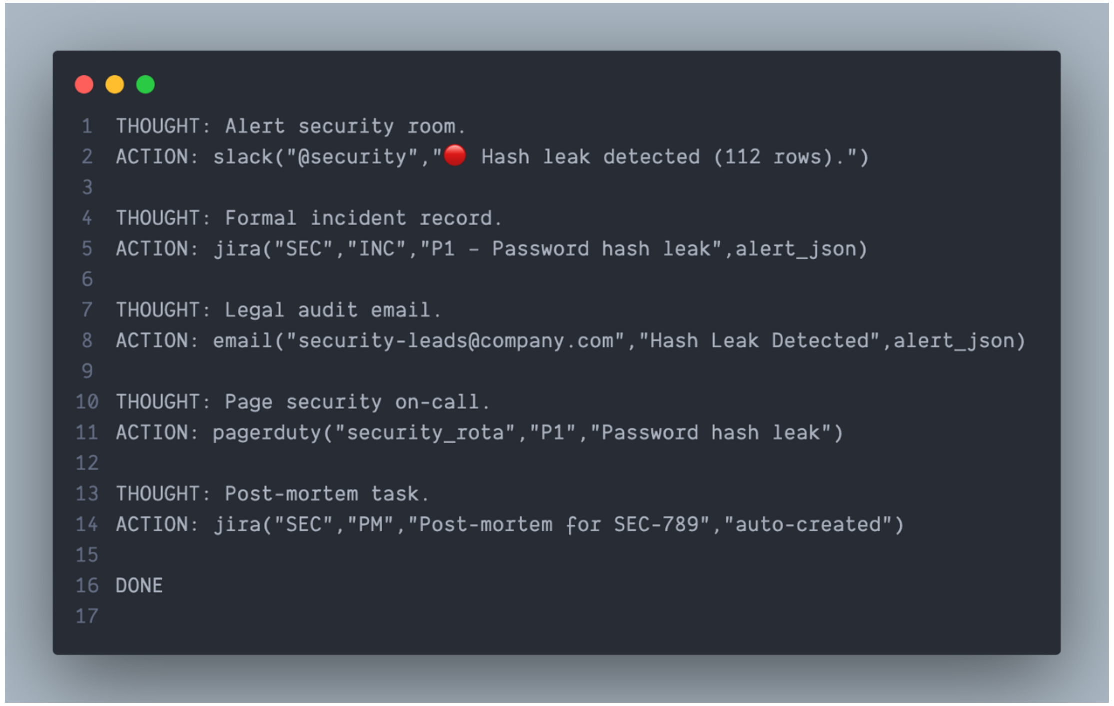
```
// Dynamic Example A (P1 Checkout Failure)

THOUGHT: Pager on-call; prod checkout down.
ACTION: pagerduty("ecom","P1","Checkout 5xx 87%")

THOUGHT: Customer notice required.
ACTION: statuspage("42","major","Checkout failing")

THOUGHT: Audit trail.
ACTION: jira("OPS","INC","P1 – Checkout 5xx",alert_json)

THOUGHT: Wait for recovery.
ACTION: probe("https://api/health/checkout",600)

THOUGHT: Inform stakeholders resolved.
ACTION: slack("#incidents","Resolved in 8 min.")

THOUGHT: Post-mortem task.
ACTION: jira("OPS","PM","Post-mortem for INC-1234","auto-created")

DONE

```

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

Staging Latency & Security Hash Leak Flow



```
THOUGHT: Non-prod; ticket only.
ACTION: jira("OPS","INC","Staging auth latency 350 ms",alert_json)

DONE

```



```
THOUGHT: Alert security room.
ACTION: slack("@security","🔴 Hash leak detected (112 rows).")

THOUGHT: Formal incident record.
ACTION: jira("SEC","INC","P1 – Password hash leak",alert_json)

THOUGHT: Legal audit email.
ACTION: email("security-leads@company.com","Hash Leak Detected",alert_json)

THOUGHT: Page security on-call.
ACTION: pagerduty("security_rota","P1","Password hash leak")

THOUGHT: Post-mortem task.
ACTION: jira("SEC","PM","Post-mortem for SEC-789","auto-created")

DONE

```

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

How n8n Executes THOUGHT / ACTION

- **LLM** thinks and produces an ordered list of actions.
- n8n **breaks the list into individual tasks**.
- A **router** sends each task to the right tool — PagerDuty, Statuspage, Jira, Slack, Email, or Probe.
- Each tool **executes with its task's details**.
- The Probe task keeps checking health until the service is back.
- n8n **logs the entire plan and every result** so you can audit what happened later.

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

Comparison Matrix

Dimension	Linear Workflow	Dynamic Flow
Path per alert	Fixed graph edges	Built from THOUGHT/ACTION
Policy change	Edit canvas	Edit prompt text
Latency	20–50 ms	+ LLM 300–800 ms
Runtime cost	Minimal	LLM tokens
Audit artefact	Node log	Node log + THOUGHT/ACTION
Best when	Rules stable	Rules evolve weekly

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

Key Takeaways

- **Linear workflow** = conveyor belt — perfect for stable, regulated runbooks.
- **LLM THOUGHT/ACTION flow** = smart switchboard — thrives when variability rules the day.
- Both modes can run entirely inside **n8n**; choose by *variance vs. determinism*. 🚦

© Presidio, Inc. All rights reserved. Proprietary and Confidential.

Thank You

Sathish Kumar Saravanan

PRESIDIO®