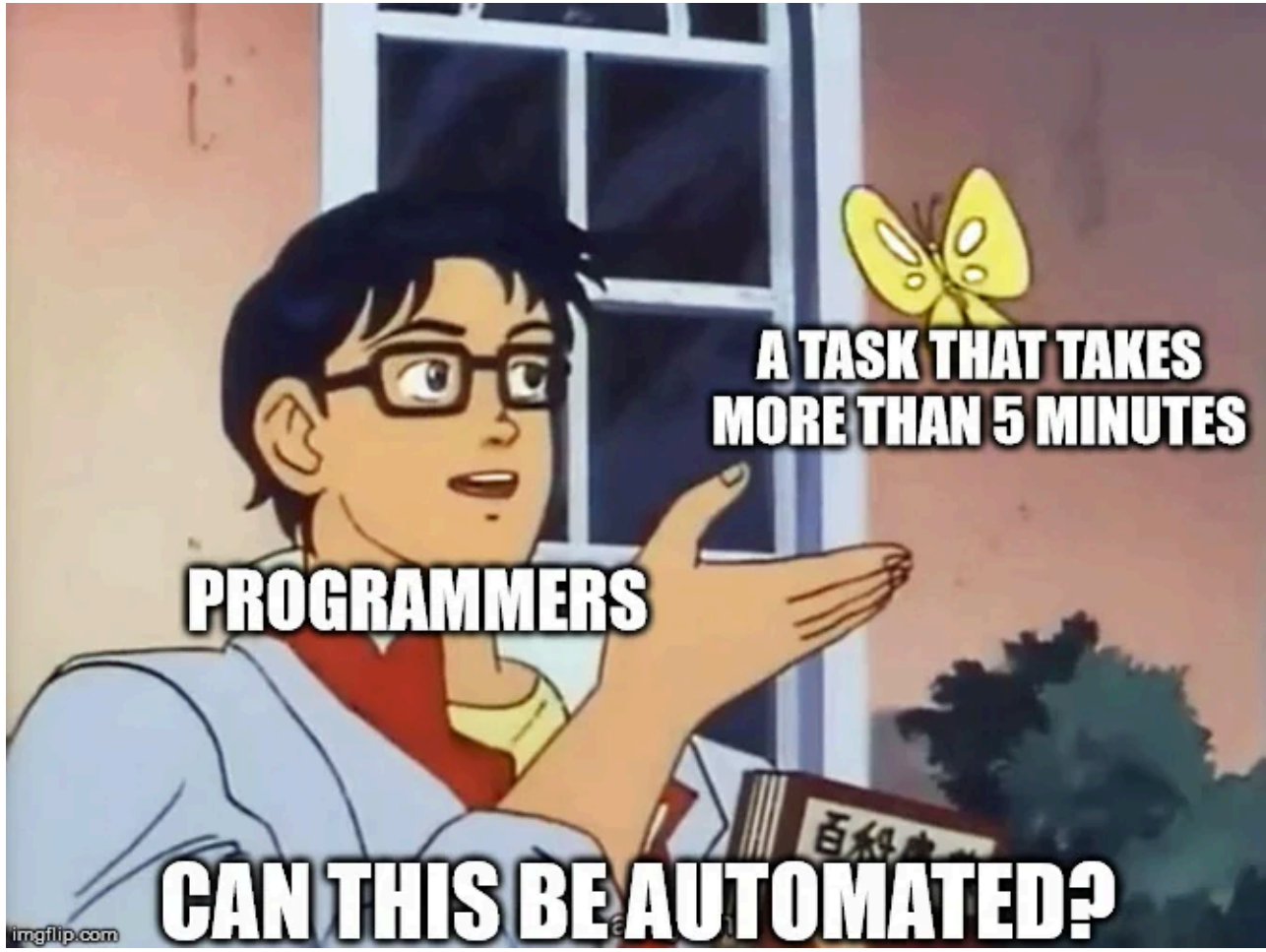
Rolling your own AI SRE agent with Github Copilot

Automate Github Copilot to be your incident triage teammate

Farid Nouri Neshat

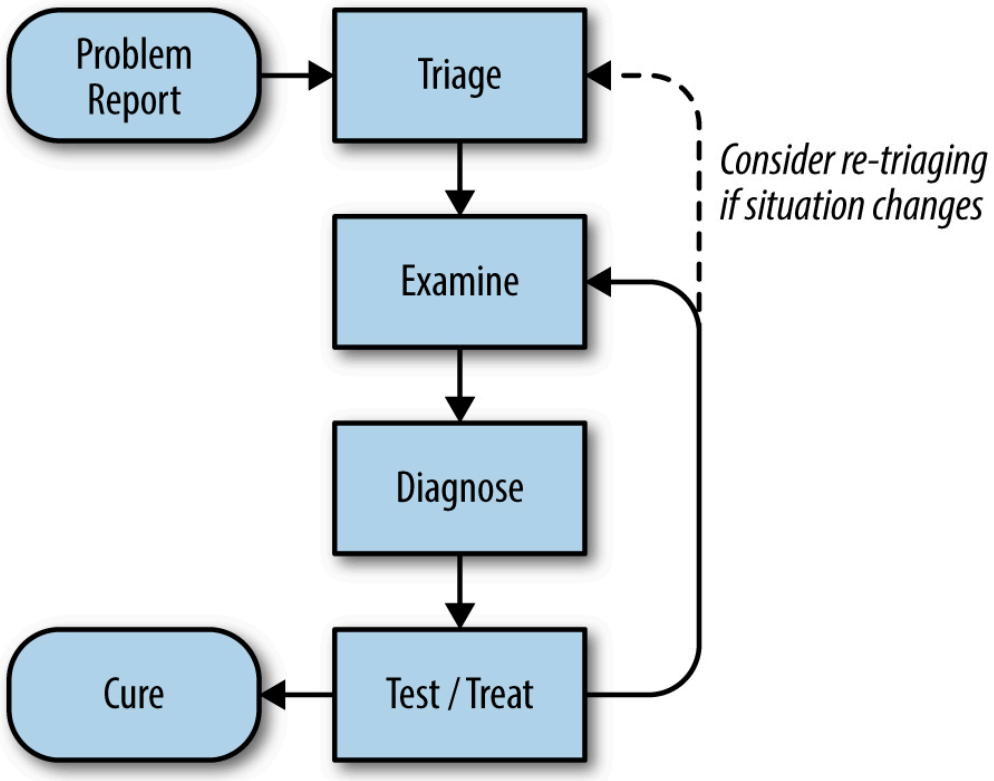
me?

- Senior vibe coder
- Software engineer at heart
- Otherwise AWS cloud engineer
- Unnecessarily automates processes!
- Horrible Guitar Player!



AN INCIDENT HAS HAPPENED

Troubleshooting



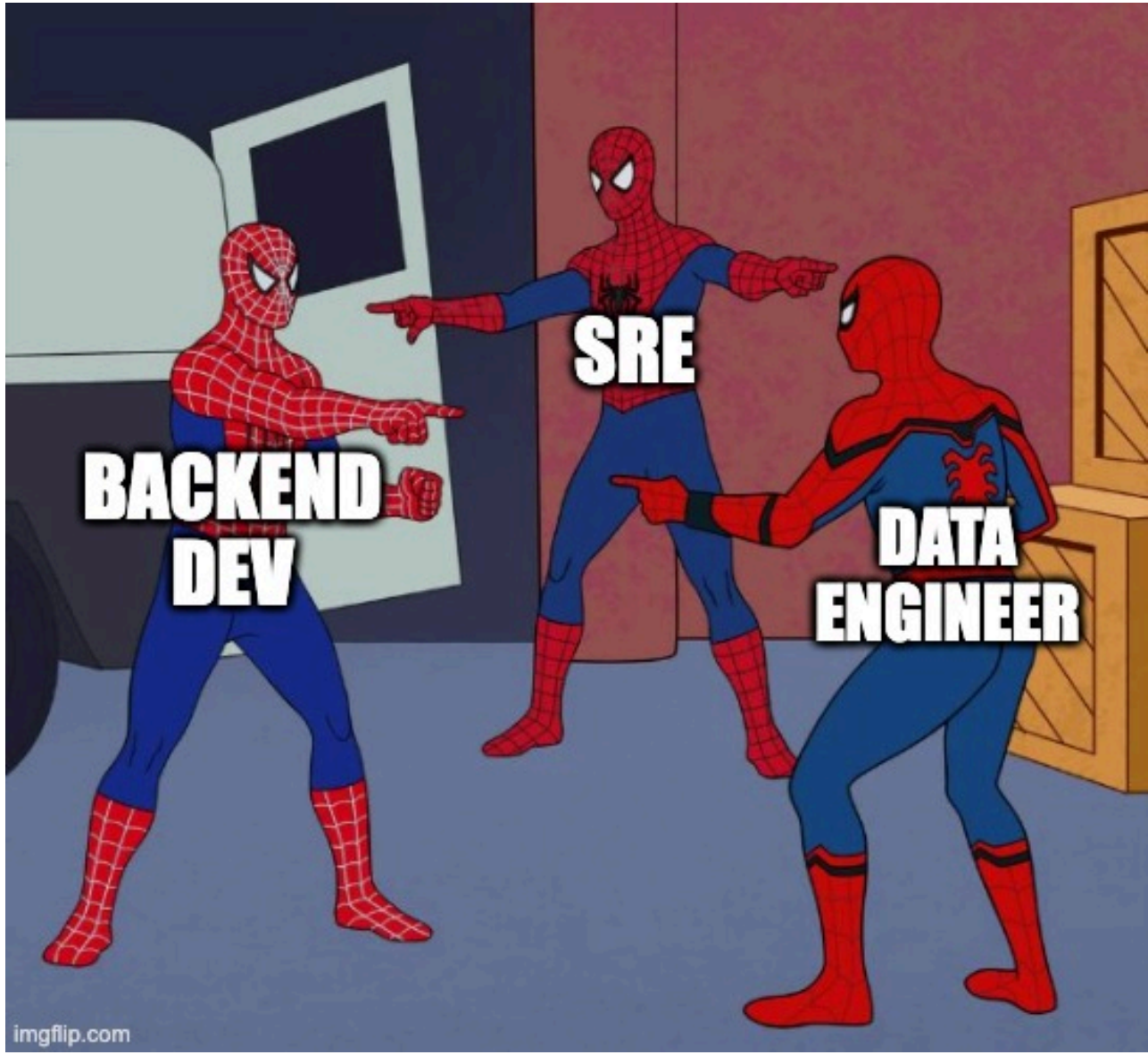
Source: "Site Reliability Engineering" by Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy. Copyright © 2016 Google, Inc.

Triage

First need to figure out a bunch of things:

- Severity
- Who to involve
- Any other alerts and incidents?
- What can we do mitigate quickly and safely

Troubleshooting the root cause comes afterwards.



What's next?

- Examine: Check the obsverablity tooling
- Diagnose: Follow data through the system
- Test & Treat: Verify hypotheses, develop solutions, verify
- Cure: Root cause, Postmortem time

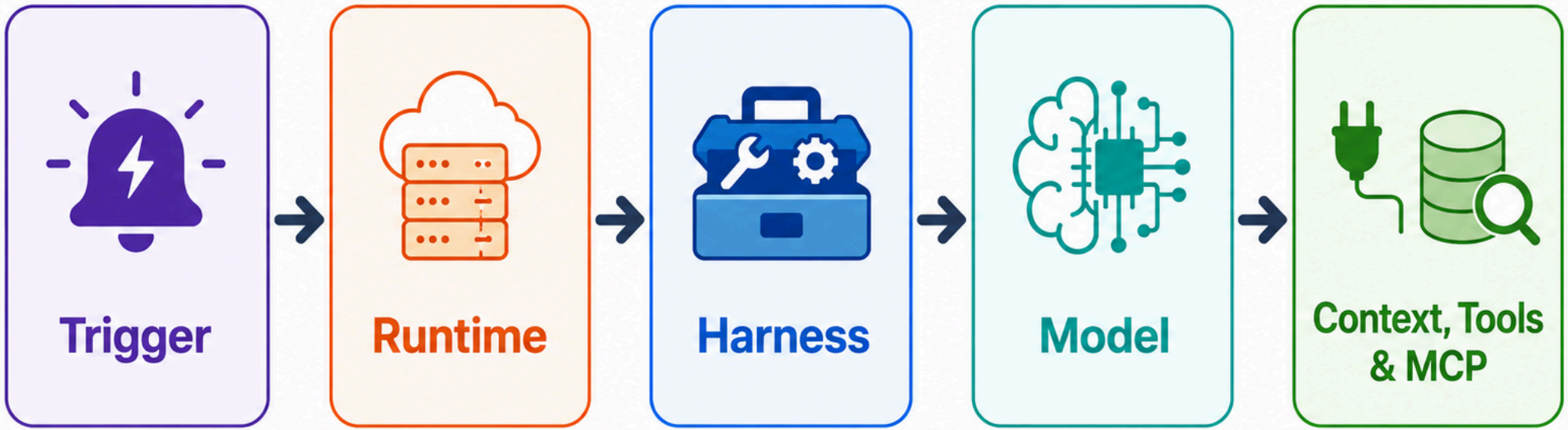
Why build this internally?

1. You can keep it simple.
2. Cheaper, you pay for the tokens only
3. Full customizability
4. Control
5. Build your own expertise in automatic agentic AI
6. You don't need a knowledge graph or self learning.

How do we automate?

1. Have Copilot agent to be triggered as soon as alert happens
2. Give it the right prompt and context
3. Access to tools for troubleshooting and testing
4. Profit?

What do we need?



Model

- Use small models for quick evaluations, classifications, simple tasks when no reasoning is required
- Use models which score higher in agentic bencmarks for troubleshooting
- Use models which score higher in reasoning bencmarks for verification
- Use sub agents for consuming token intensive tool responses for example reading logs

Harness

- Copilot Agent
- Copilot CLI
- Third party agents: Anthropic Claude & OpenAI Codex
- Copilot SDK

Agent Runtime

A place to run your code

- Copilot Cloud Agent
- Copilot Cloud Agent on self hosted Github Action runners
- Third party managed agent runtime
- Self managed on your own infra favorite container service.

Trigger

- Copilot API
- Creating an issue in Github
- Creating an issue in Jira
- Self managed? Ask your infra architect!

Context, Tools, MCP & Access

- Include the latest system changes or major events
- Include other recent alerts
- Access to relevant repositories
- Access to observability tooling
- Readonly access to cloud resources & databases
- Access to non-destructive actions for testing and verification

Security

Let's make sure a single typo and a malicious npm module will not leak your data!

- Ensure there's a outbound firewall. Don't forget DNS firewall.
- Point the agent to a private whitelisted artifact repository
- Ideally credentials the agent has access to only work within it's network
- A data perimeter for your cloud estate

How to enable Agent to mitigate

1. Ensure all config and infra are managed by code.
2. Agents can then just make pull requests for changes.
3. Extra CI checks can done automatically
4. Reviewed and merged by human
5. Pipeline applies the change

What about dangerous actions?

- Use governence layers.
- For dealing with sensitive data, use speclized agent with less access
- Create deterministic playbooks with guardrails included

Governance Layers?

1. Provided as an MCP / CLI / tool to the agent
2. Evaluates all actions based on configured rules
3. When human approvals are needed, sends it to the configured team/channel
4. (Optionally) have AI evaluation layers on risks of the action
5. (Optionally) masks sensitive data before returning

Test!

- Do regular chaos engineering and gamedays
- Evaluate the agent with mock MCP and resources & simulate previous events.

Continuous Improvement

- Every alert / incident should result in an action.
- Sometimes the alarm needs to be fixed.
- Improve the setup after every postmortem to see how various steps can be improved.
- Do not let it become noise.

Questions?

Also check out my website at faridnsh.ninja!