Getting off the Scan-and-Fix Hamster Wheel with Generative AI

Asankhaya Sharma

Asankhaya Sharma, Co-Founder & CTO, https://patched.codes

2003 — Institute of National Technology, Warangal

2007 — NUS National University of Singapore

2010 — Microsoft

2014 — [:] SourceClear

2018 — ca technologies

2019 — VERACODE

2022 — Securade.ai

2023 — patched

Code Analysis Tool for .NET v2.0

DIDAR – Database Intrusion Detection with Automated Recovery

HIP/SLEEK :Automatic Verification and Specification Inference System

Botwall4J

SRC CLR SCA Agent

GramTest

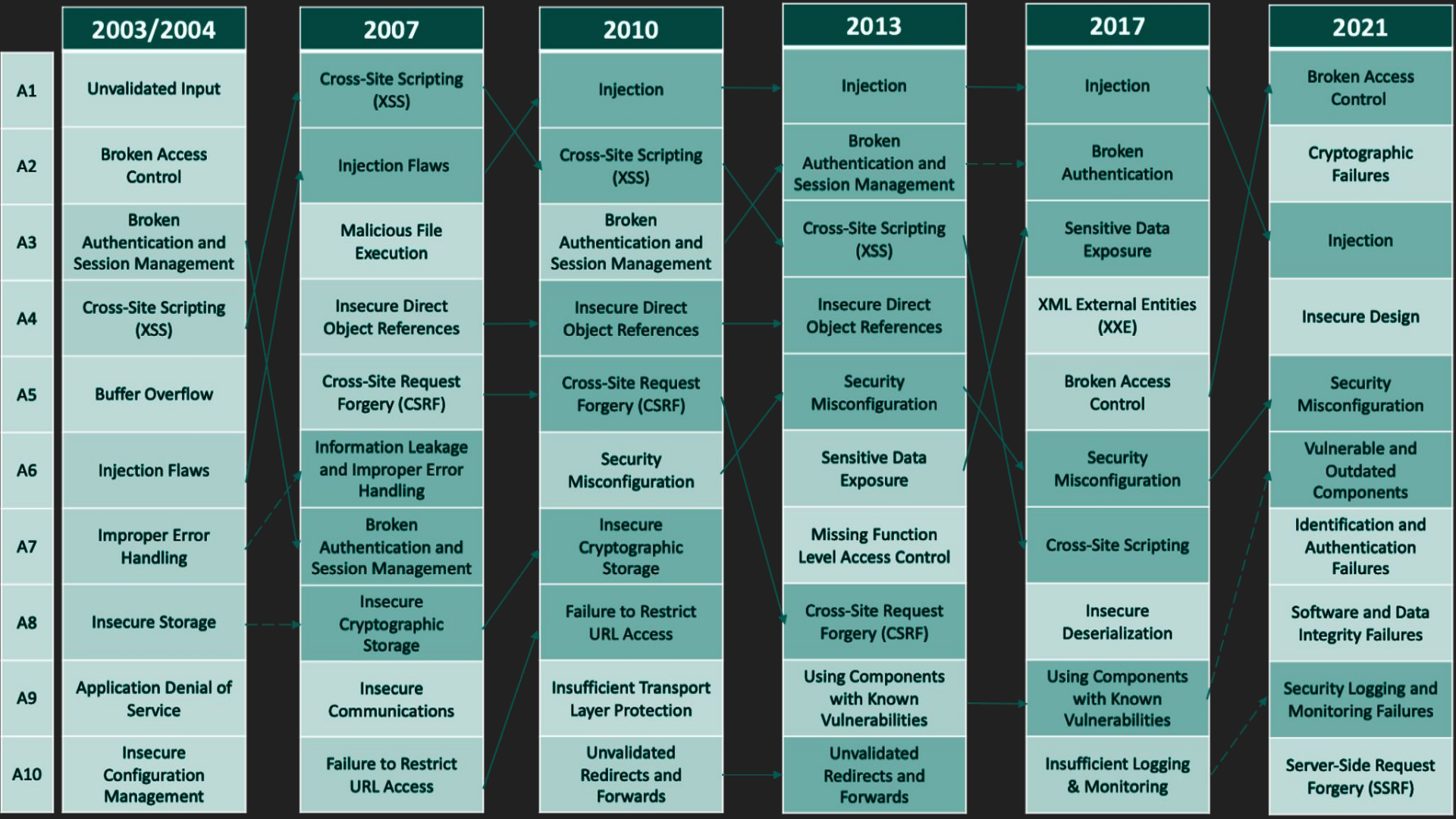
AutoFix

Static Analysis + LLM = AutoFix

building security tools for developers

v/s

developer tools for security


2003/2004
2007
2010
2013
2017
2021
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
Unvalidated Input
Broken Access Control
Broken Authentication and Session Management
Cross-Site Scripting (XSS)
Buffer Overflow
Injection Flaws
Improper Error Handling
Insecure Storage
Application Denial of Service
Insecure Configuration Management
Cross-Site Scripting (XSS)
Injection Flaws
Malicious File Execution
Insecure Direct Object References
Cross-Site Request Forgery (CSRF)
Information Leakage and Improper Error Handling
Broken Authentication and Session Management
Insecure Cryptographic Storage
Insecure Communications
Failure to Restrict URL Access
Injection
Cross-Site Scripting (XSS)
Broken Authentication and Session Management
Insecure Direct Object References
Cross-Site Request Forgery (CSRF)
Security Misconfiguration
Insecure Cryptographic Storage
Failure to Restrict URL Access
Insufficient Transport Layer Protection
Unvalidated Redirects and Forwards
Injection
Broken Authentication and Session Management
Cross-Site Scripting (XSS)
Insecure Direct Object References
Security Misconfiguration
Sensitive Data Exposure
Missing Function Level Access Control
Cross-Site Request Forgery (CSRF)
Using Components with Known Vulnerabilities
Unvalidated Redirects and Forwards
Injection
Broken Authentication
Sensitive Data Exposure
XML External Entities (XXE)
Broken Access Control
Security Misconfiguration
Cross-Site Scripting
Insecure Deserialization
Using Components with Known Vulnerabilities
Insufficient Logging & Monitoring
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification and Authentication Failures
Software and Data Integrity Failures
Security Logging and Monitoring Failures
Server-Side Request Forgery (SSRF)

Breaking the Cycle: Beyond Scan-and-Fix in AppSec

- Old Ways, New Challenges
  - Stuck in a "scan and fix" loop, traditional tools leave us chasing vulnerabilities instead of being proactive.
- Shift Left Illusion
  - Moving security earlier in the SDLC doesn't stop the cycle - it starts it sooner, overburdening developers.
- IDE Interruptions
  - Real-time scanning in IDEs disrupts developer workflow, compromising productivity with constant alerts and overhead.

This pattern is just wrong. It's broken. We've seen a history of the challenges following this pattern does in working with developers."
—Chris Romeo

How good are LLMs at fixing vuls?

Static Analysis Eval

A dataset of 76 Python programs taken from real Python open source projects (top 1000 on GitHub), where each program is a file that has exactly 1 vulnerability as detected by a particular static analyzer (Semgrep).

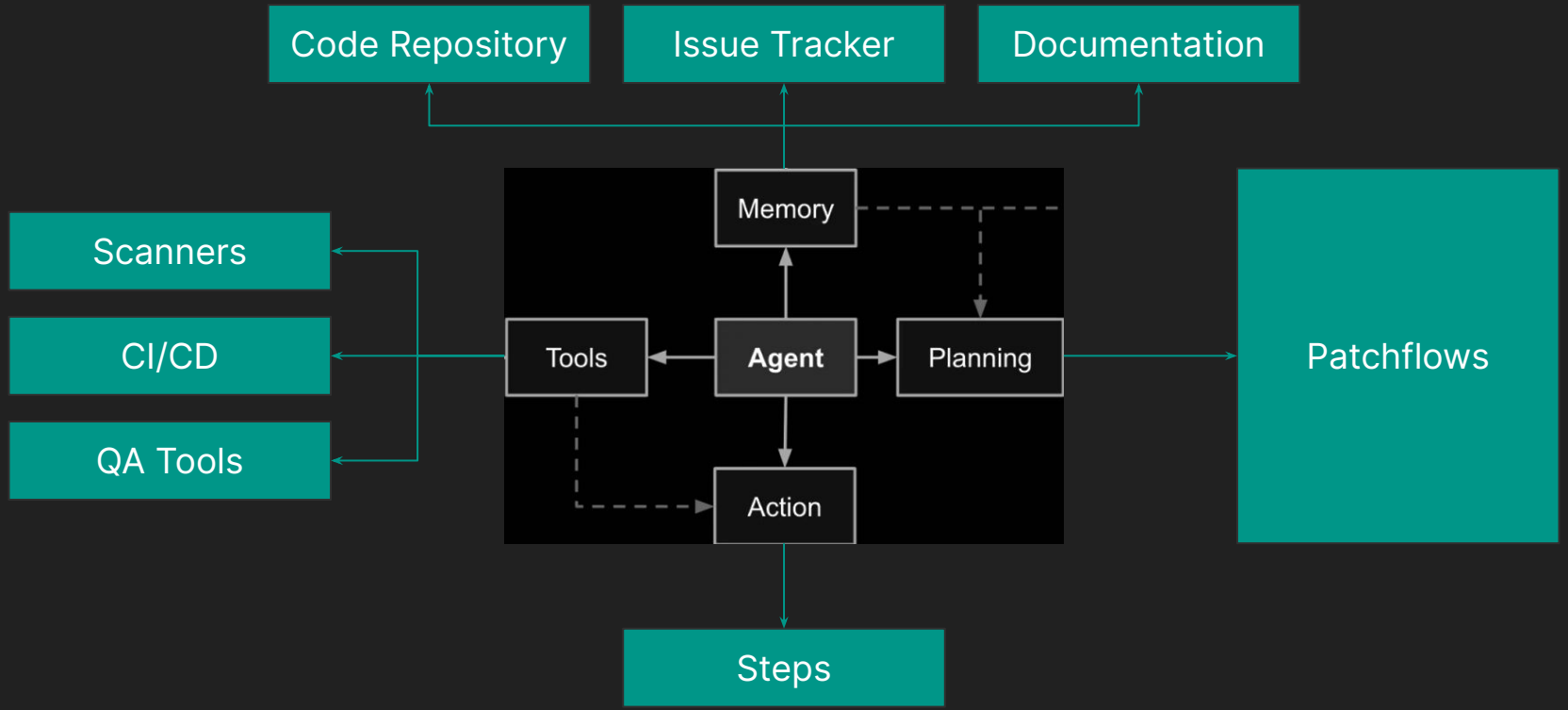
https://huggingface.co/datasets/patched-codes/static-analysis-eval

Model	StaticAnalysisEval (%)	Time (mins)
gpt-4o	69.74	23:0
gemini-1.5-flash-latest	68.42	18:2
Llama-3-70B-instruct	65.78	35:2
Llama-3-8B-instruct	65.78	31.34
gemini-1.5-pro-latest	64.47	34:40
gpt-4-1106-preview	64.47	27:56
gpt-4	63.16	26:31
gpt-4-0125-preview	53.94	34:40
patched-coder-7b	51.31	45.20
patched-coder-34b	46.05	33:58
Mistral-Large	40.80	60:00+
Gemini-pro	39.47	16:09
Mistral-Medium	39.47	60:00+
Mixtral-Small	30.26	30:09
gpt-3.5-turbo-0125	28.95	21:50
claude-3-opus-20240229	25.00	60:00+
Gemma-7b-it	19.73	36:40
gpt-3.5-turbo-1106	17.11	13:00

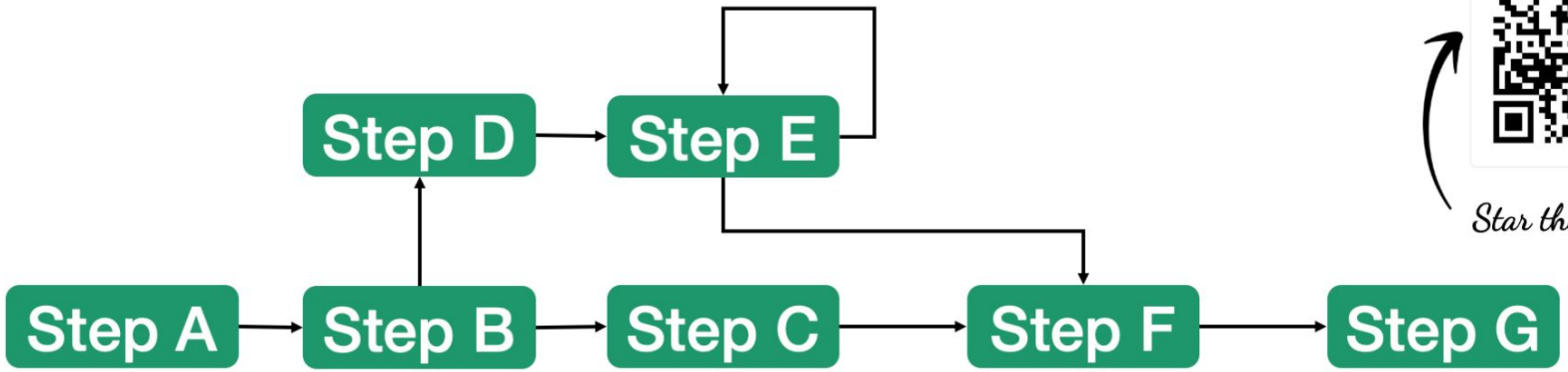
Introducing Patchwork

An open-source framework that effortlessly integrates into and automates tasks like vulnerability fixes, while giving you complete flexibility and control.

Patchwork Overview



Patchflow



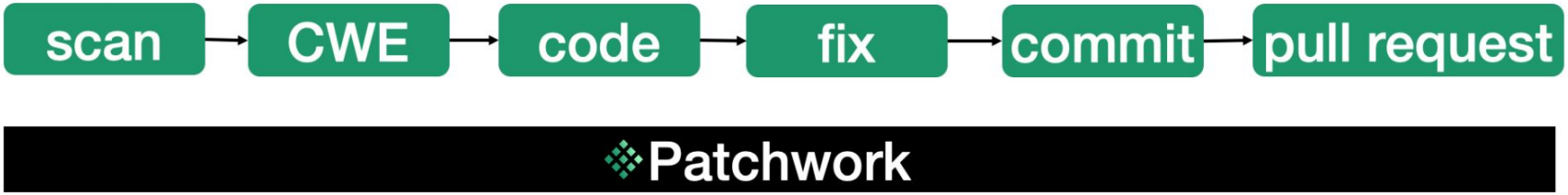
Star the GH Repo

Patchwork

https://github.com/patched-codes/patchwork

AutoFix

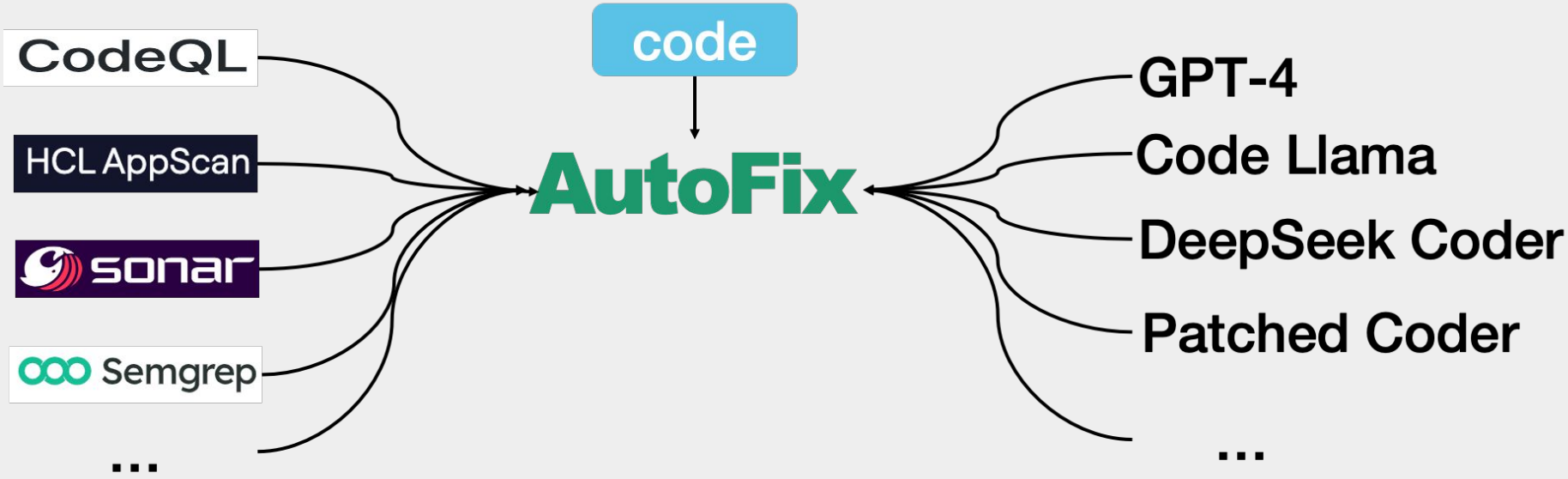
5,000+ Vuls Patched

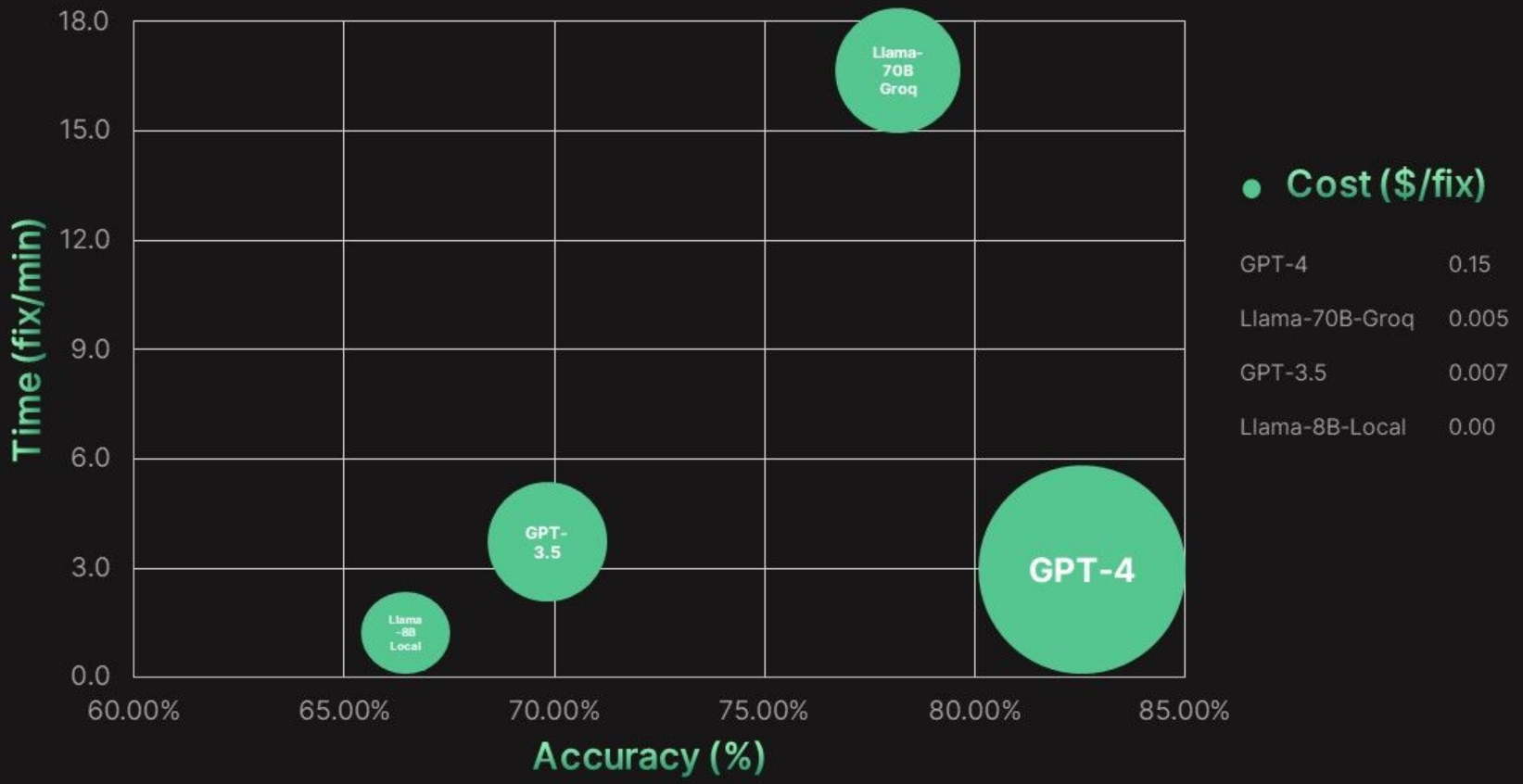



```
patchwork AutoFix sarif_file_path=results.sarif severity=critical patch_template_file=customprompts.json
```

https://github.com/patched-codes/patchwork

SAST + LLMs


CodeQL
HCL AppScan
sonar
Semgrep
...
code
AutoFix
GPT-4
Code Llama
DeepSeek Coder
Patched Coder
...


18.0
15.0
12.0
9.0
6.0
3.0
0.0
Time (fix/min)
60.00%
65.00%
70.00%
75.00%
80.00%
85.00%
Accuracy (%)
Llama-70B Groq
GPT-3.5
Llama-8B Local
GPT-4
Cost ($/fix)
GPT-4 0.15
Llama-70B-Groq 0.005
GPT-3.5 0.007
Llama-8B-Local 0.00

Why Patchwork?

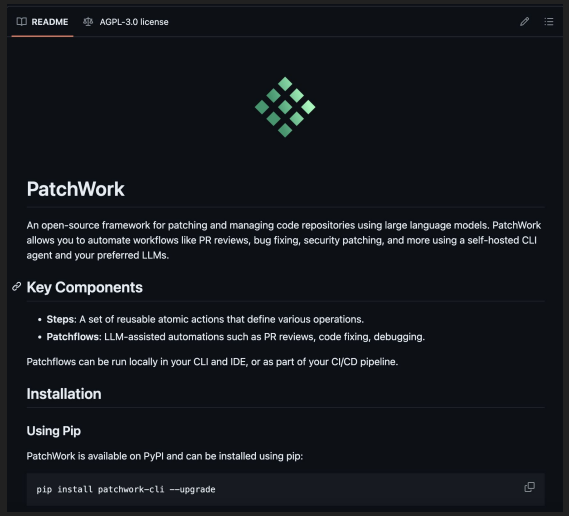
Integrated with IDE, CLI and CI

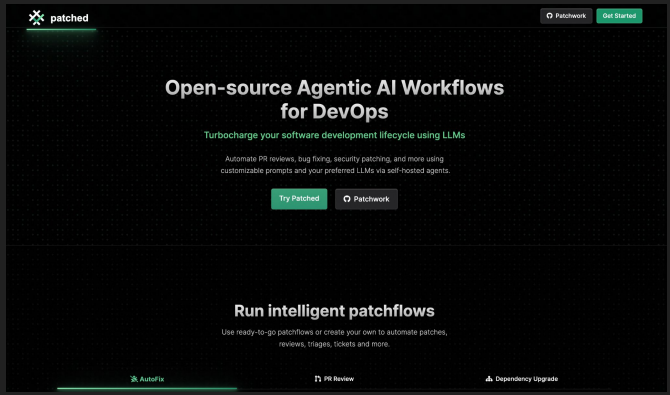
Extensible with Steps

Works with any LLM

Customizable with prompt templates

Demo





https://github.com/patched-codes/patchwork

https://patched.codes

Thank You!