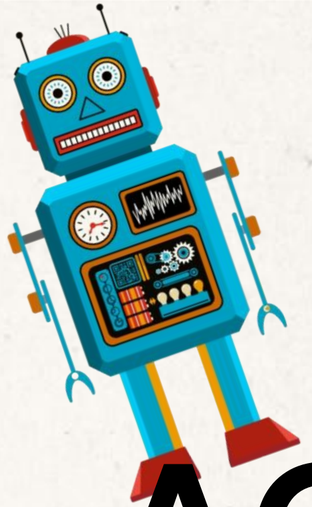
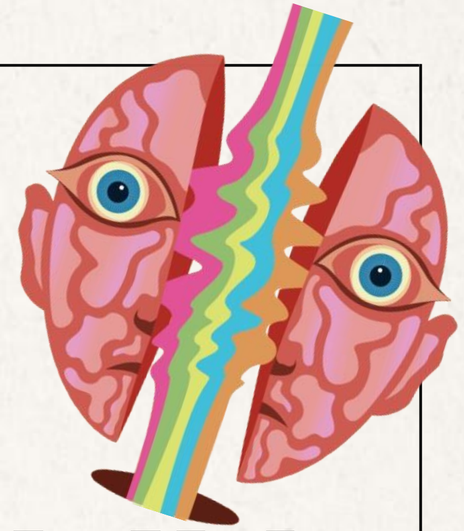
Research Fellow	MSR	Lightening talk





AGENTS TO ANOTOMY
Code doesn’t bleed

A Presentation by: Sathvik Joel

05/13/2026

Research Fellow	MSR	Lightening talk

Research works

1. **Code researcher: Deep research agent for large systems code and commit history**

arxiv
R Singh, S Joel*, A Mehrotra, N Wadhwa, RB Bairi, A Kanade, N Natarajan*

2. **Exposing Weak Links in Multi-Agent Systems under Adversarial Prompting**

Accepted at SE @ AAMAS 2026
Nirmit Arora, Sathvik Joel*, Ishan Kavathekar, Rohan Gandhi, Yash Pandya, Tanuja Ganu, Aditya Kanade, Akshay Nambi*

3. **CheXAlpha: Making Radiology VLMs Clinically Trustworthy through Auxiliary Supervision, Reward-Aligned Learning, and Tool-Augmented Measurement**

Under review at MLHC 2026
Anirban Porya, Mercy Prasanna Ranjit*, Niharika Vadlamudi*, Nikhilesh Chowdary Eathamukkala*, Prasanth V V*, Sathvik Joel*, Abhyuday Kumara Swamy, Pranay Narhari Umredkar, Pradeep Narayan, Vivek Rajagopal, Tanuja Ganu*

CODE RESEARCHER

Given a crash report (C) and a reproducer (R) that triggers a crash in the buggy kernel K_{buggy} find a patch (P) such that when applied to K_{buggy} it produces K_{fixed}, where $K_{fixed}(R)$ doesn't result in crash

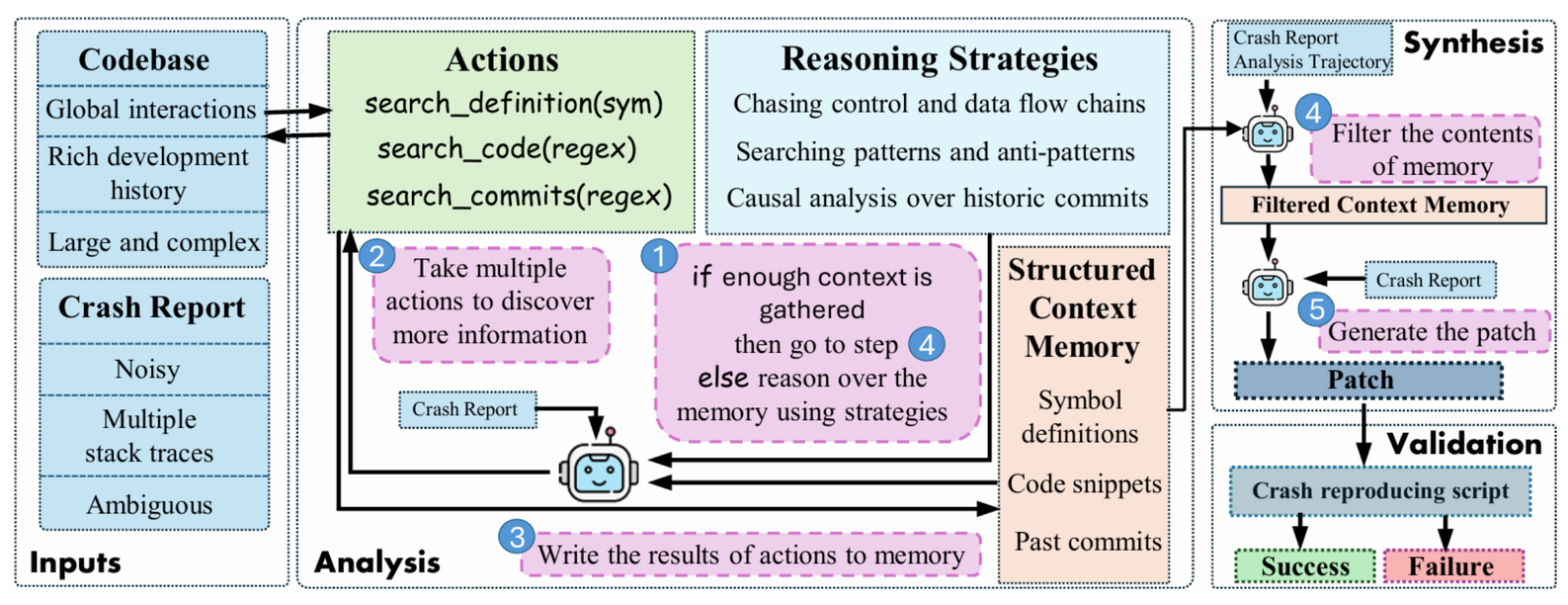
```
jump label: negative count!
WARNING: CPU: 1 PID: 3629 at kernel/jump_label.c:235
static_key_slow_try_dec+0xca/0xe0 kernel/jump_label.c:235Modules linked in:CPU:


<TASK>
__static_key_slow_dec_cpuslocked kernel/jump_label.c:243 [inline]
__static_key_slow_dec kernel/jump_label.c:255 [inline]
static_key_slow_dec+0x5c/0xc0 kernel/jump_label.c:270
nf_tables_chain_destroy+0x250/0x640 net/netfilter/nf_tables_api.c:1880
nf_tables_addchain.constprop.0+0xc56/0x16e0 net/netfilter/nf_tables_api.c:2329
nf_tables_newchain+0x16d1/0x1ef0 net/netfilter/nf_tables_api.c:2593

<TASK>
```

Crash Report

PROBLEMS

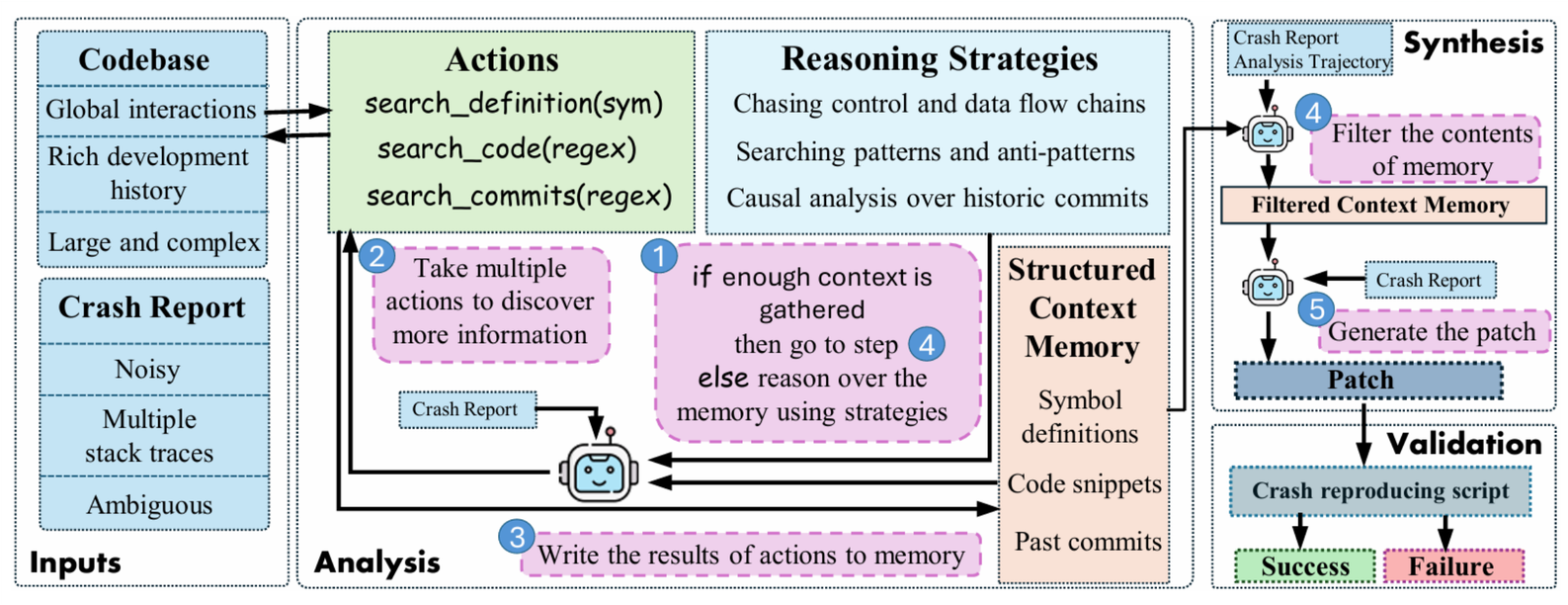
- ❑ Universal: Linux is critical software
- ❑ Massive: 28M LOC, 75K+ files
- ❑ High cost: 2k kernel developers, ~70 days (median) to fix crash
- ❑ Low-level Systems code: Mostly written in C, Assembly, Bash, Rust
- ❑ Ambiguous Intent of Crash report
- ❑ Diverse coding styles: Many subsystems and coding conventions

CODE RESEARCHER



- This architecture in some ways resembles modern Coding Agents like Claude Code (recently released by mistake), but we build this before May 2025 (CC release date).
- Differs in some important ways: Reasoning strategies mimic the deep research behavior of expert developers when navigating large, complex systems codebases

CODE RESEARCHER



- We took advantage of the rich history these large codebases have, “Causal analysis over historic commits” and “search_commits(regex)” action are good examples. We see a 10% drop in crash resolution rate (CRR) without these
- Testing this benchmark is a nightmare, testing each patch involves compiling the kernel

2. Exposing Weak Links in Multi-Agent Systems under Adversarial Prompting

Problem

Architectural Vulnerability Gap: Existing security research focuses on single-agent systems, leaving a critical gap in understanding how multi-agent design choices introduce new security risks.

Emergent "Weak Links": Vulnerabilities arise from design choices, such as how context is shared or how plans are constructed—that inadvertently create blind spots for safety.

SOLUTION

Introducing SAFEAGENTS

We developed a unified, modular framework that allows for the systematic security evaluation of different MAS architectures.

Identifying Failure Patterns

Our analysis reveals critical vulnerabilities, such as "Context Fragmentation," where sub-agents receive only tiny, atomic instructions and therefore cannot recognize a larger, harmful objective.

DHARMA Diagnostic Metric

To solve the measurement gap, we created DHARMA (Design-aware Harm Assessment Metric for Agents). This hierarchical metric localizes rejections to specific components—like the planner or sub-agents—to pinpoint the root cause of a failure.

Path Toward Security-Aware Design

Our findings prove that building secure MAS requires more than just safe models; it requires deliberate architectural decisions, such as robust fallback mechanisms and better context organization.


AI SAFETY

What next? Building an AI Matchmaking startup !

Our product sits between Tinder and Shaadi : no swiping, no marriage pressure — built exclusively for urban Indian professionals aged 25 to 40 who are ready to commit but not ready to compromise.

Our conversational AI gets to know you the way a close friend would, then arranges curated dates at handpicked venues across the city. You do not swipe, you just show up.

And unlike anything in the market, our product gets smarter after every date, learning what worked, refining the next match, building a system that compounds with every human interaction it touches

PITCH SLIDE: WELCOME TO THE FUTURE OF FINDING LOVE ☺

THANK YOU MSR !