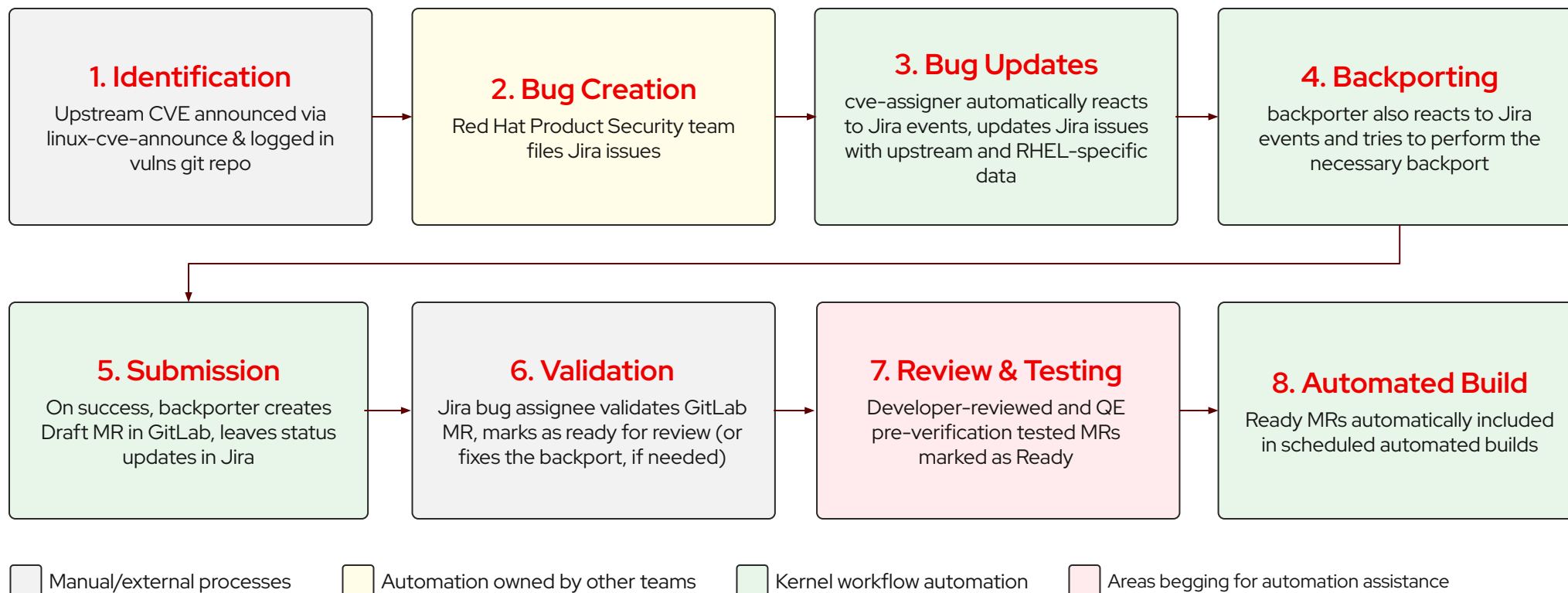


Red Hat Kernel Backport Automation

Accelerating kernel security fixes

Jarod Wilson, Senior Principal Software Engineer
Kernel Workflow Team Lead

Automated CVE Backport Workflow



This automated workflow handled 82.8% of all Red Hat Kernel CVE fixes in August & September 2026

Current Status: Kernel Backport Agent – Patchpal Integration

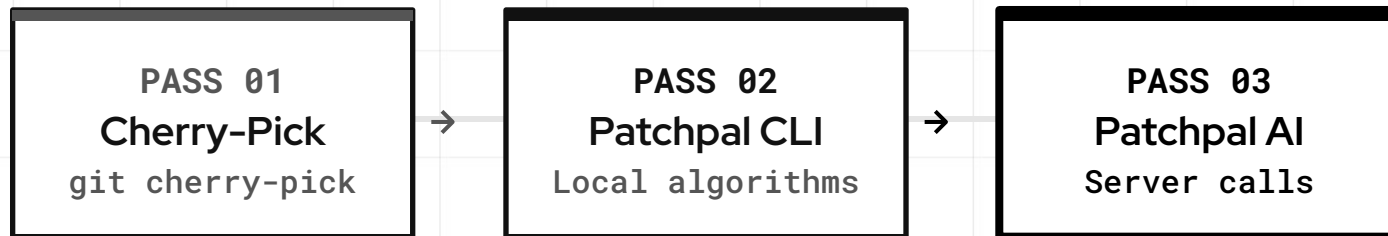
SYSTEM OVERVIEW

Standard git cherry-pick with Patchpal fallbacks

- 1) Try git cherry-pick
- 2) Try `patchpal-cli` dependency commit detection, if depchain short enough, try to apply everything
- 3) Throw out everything and try patchpal AI

PIPELINE EXECUTION

Three-Tiered Multi-Pass Strategy



August/September 2026: 86.5% of automated backports accepted as-is. But 96.7% of those were pure git cherry-pick, and AI-assisted backports have a much higher rejection rate.

\$0 TOKEN COST

Runs on hardware in Red Hat's own labs, the only ongoing cost is the power for the backend server and effort to keep it running.

LIMITATIONS

The backend has only a single 80GB VRAM GPU to work with. The largest and latest open models require more resources to run training, and may lag behind the frontier models from Anthropic, OpenAI, Google, etc., in terms of overall model functionality and capability.

The Planned Addition: Incorporation of Synthmerge

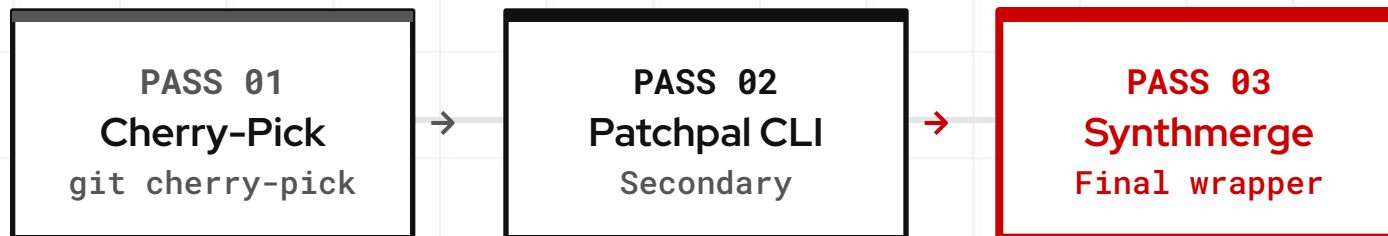
SYSTEM OVERVIEW

Multi-System Wrapper Strategy

Replaces prior step 3 with calls to **synthmerge**, configured to use an as yet to be determined combination of three AI backends to (ideally) reach consensus on a given AI backport and improve overall success rate.

PIPELINE EXECUTION

Three-Tiered Multi-Pass Strategy



COST INCREASE

Using a mix of models like Gemini 3.7 Flash and ChatGPT Sol 5.6 should keep costs fairly minimal per Merge Request, as well as continuing to use Patchpal as a backend, but it will be an increase over \$0 token spend. Earlier data suggests about \$0.45 per commit with conflict

INTEGRATION TIMELINE

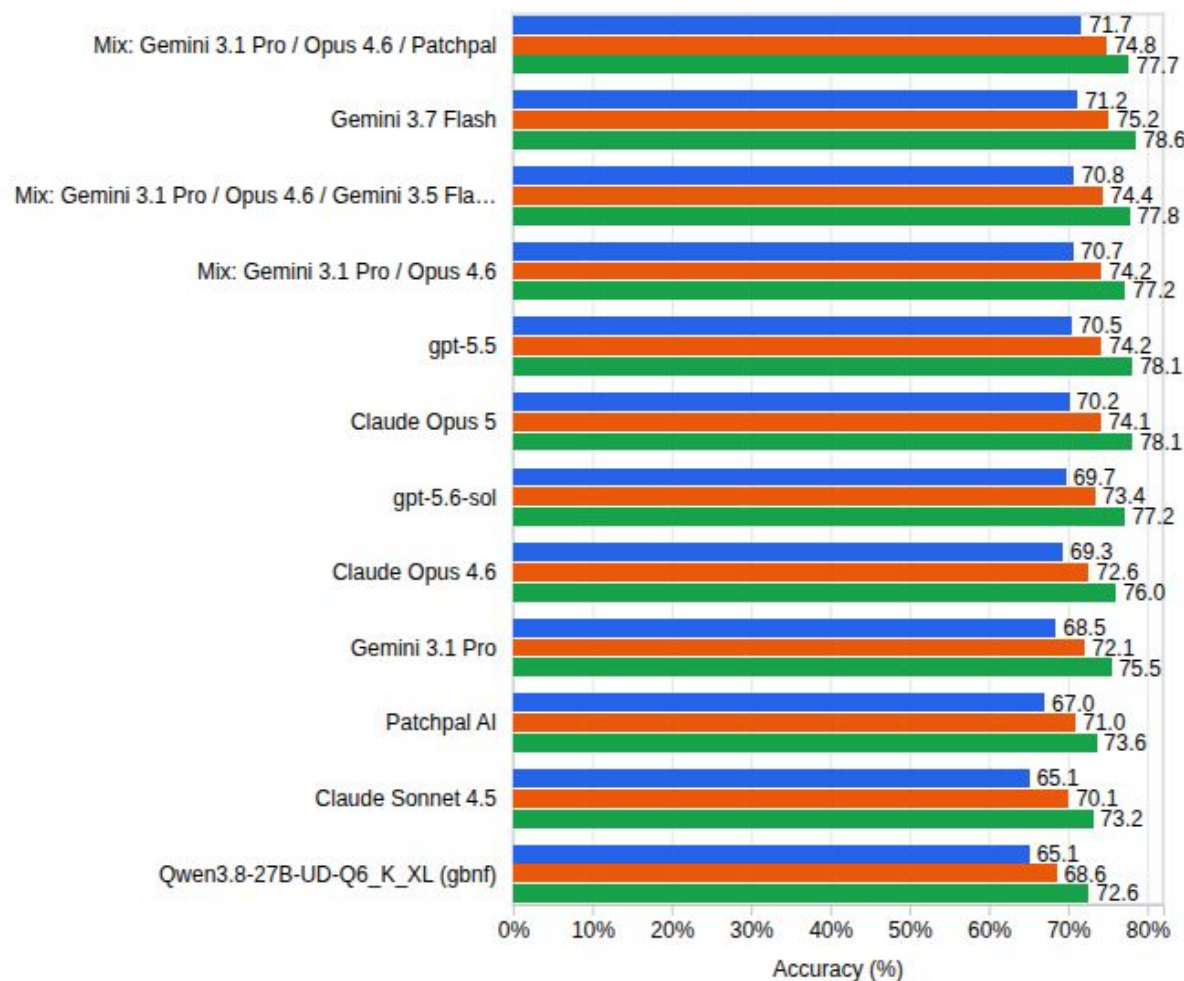
Incremental Rollout Plan

Due to resource constraints, we started with just the Patchpal CLI mode to establish a baseline before fully integrating Synthmerge. Synthmerge deployment is still a work in progress.

Model Benchmarks

Model accuracy comparison

Original Aligned Stripped



Notes

- More detailed numbers available from synthmerge repo
- Overall model success rate is not the only metric of importance, ability to handle complexity also matters
- We're still nowhere near 100%
- Patchpal holds up very well for how small the model is compared to others

CONCLUSIONS

A balanced mix of a high overall success rate model, a high success rate on complex backports model, and a third relatively low-cost model can theoretically increase our AI backport success rate by about 4%

Patchpal – <https://gitlab.com/patchpal-ai>

- An AI-assisted cherry-picking and merge conflict resolution application that simplifies the process of applying fixes and features across different code bases.
 - Public open-source project, headed by Red Hat's own Jiri Benc and Eric Garver
 - Backend written in python, frontend written in rust
 - Mostly model-independent, but internal production server currently uses a RHEL-trained and tuned QwenCoder model (license is compatible with RH's needs and use)
 - Backend runs on relatively modest 40-80GB GPU host, but with a correspondingly somewhat limited model size
 - Frontend has both a GUI and CLI available
 - Frontend has various parsing and patching algorithms in addition to the ability to request assistance from the AI backend
 - Frontend can both initiate a backport and be invoked to resolve merge conflicts
 - Frontend can handle a multi-patch backport with reordering, adding detected possible prerequisite patches, editing hunks, etc.

Synthmerge – <https://gitlab.com/aarcange/synthmerge>

- Synthmerge: a minimalistic command-line tool that leverages AI to automatically resolve conflicts arising from Git commands.
 - Public open-source project, headed by Red Hat's Andrea Arcangeli, built on Patchpal research
 - Written predominantly in rust
 - Instead of using its own model, runs a parallel inference engine to see the AI collective consensus
 - Can use almost any instruct LLM, has been successfully used with at least Claude, Gemini, GPT and Qwen-Coder LLMs, as well as the Patchpal backend
 - Typically recommended to be configured with three different backends
 - Multiple frontier models in agreement usually means higher certainty of backport validity
 - Operates entirely on git conflicts

Developer Feedback/Requests

- Try other linux-stable versions of the upstream fix if initial cherry-pick fails
 - Deterministic is usually better
- Ability to opt out of use of AI for backports per subsystem
 - Deterministic is usually better
- Ability to opt out of automated backports per subsystem
 - Complexity of backports, interaction with other work
- Ability to interact with the agent to instruct it for MR revisions
 - Security risks? Better to just do it w/a local/personal agent?
- Better progress notifications and logging
 - Is the queue busy? Is it stuck retrying because of ... something?
- MR creation isn't really the part where we need help
 - We need more help on the review and testing of MRs generated

Code Validation: Synthmerge MR Replayer & Sashiko Integration

Phase 01 // AI Code Reviews

■ Integrating Synthmerge & Sashiko to CentOS Stream

We are monitoring usage of [Sashiko](#) reviews (in combination with Synthmerge MR Replayer diffs) to implement an internal deployment to review CentOS Stream backports, while keeping token costs under-control.

Phase 02 // Upstream Integration

■ Commit Compare and LKML

CentOS and RHEL Merge Requests now include commit-compare evaluation tables with links to the original LKML threads and the upstream Sashiko review of backported patches, courtesy of [b4 dig](#).

Phase 03 // Reinforcement of Deterministic Evaluation***

■ Better Integration of Existing Tooling

The next phase of Sashiko deployments will look to better integrate existing internal tooling and exposed Synthmerge data to emphasize and review patches that are *the most likely* to contain errors.

*** We also do a TON of deterministic evaluation

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.



linkedin.com/company/red-hat



youtube.com/user/RedHatVideos



facebook.com/redhatinc



twitter.com/RedHat