

서울

How AI Is Changing Open Source Communities Lessons from openEuler

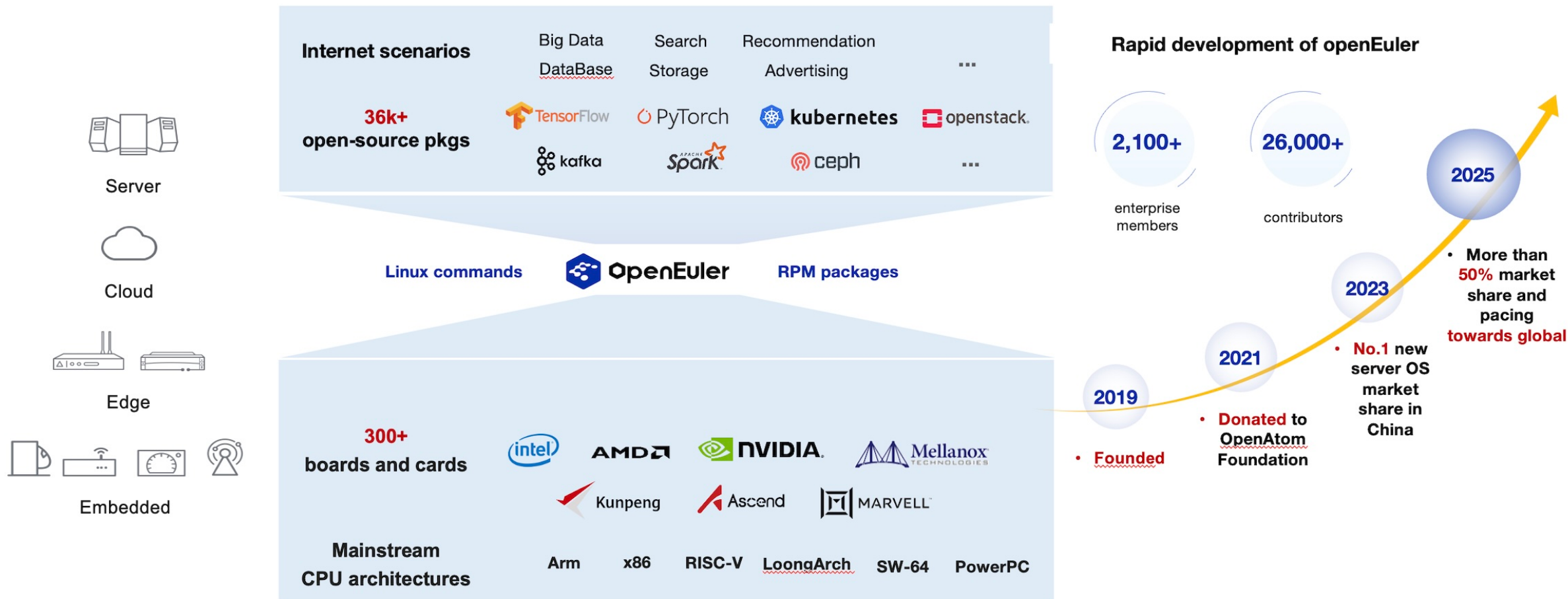
Jimmie Wang

Member of openEuler Technical Committee

Maintainer of CloudNative SIG and SBC SIG in openEuler



openEuler in Brief



From AI in the OS to AI in the Community

openEuler has explored AI for OS capability for years:

- [A-Tune](#) for intelligent tuning
- [A-Ops](#) for intelligent O&M
- [Intelligence BooM](#) for AI infrastructure
- ...

This year, AI is moving from OS optimization into community operations:

- Contributors create patches differently
- Maintainers review work differently
- Security issues are triaged differently

The new challenge is workflow change.

What We Are Seeing Now

Three changes are especially visible:

1. More security findings and faster vulnerability discovery
2. Rapid PR growth, including more invalid or low-quality submissions
3. More discussion about whether AI should participate in maintenance

Community tracking data shows:

- Vulnerability data: 2025 H2 -> 2026 H1
 - Public vulnerability awareness increased **55%** from 3,679 to 5,687
- PR data: 2025 Q1 -> 2026 Q1
 - PRs increased **72%** from 9,766 to 16,857
 - Invalid PRs increased **143%** from 2,290 to 5,568

Start with the Rules

As AI-generated code started to affect daily maintenance, maintainers faced inconsistent questions:

- Should this AI-generated code be accepted?
- How should quality be controlled?
- How can reviewers know AI was used?
- How should many AI-assisted PRs be reviewed?

Before automation scales, the community needs shared rules.

Governance should come before large-scale AI contribution.

Other Communities Are Defining Rules Too

References we reviewed:

- [Linux Foundation Generative AI Policy](#)
- [Apache Generative Tooling Guidance](#)
- [Fedora Council Policy](#)
- [GNOME Shell Extensions Review Guidelines](#)
- [QEMU Code Provenance](#)
- [NetBSD Commit Guidelines](#)

Common themes: responsibility, disclosure, community trust.

Guidance Becomes Process

In February, I proposed AI-assisted coding guidance for openEuler.

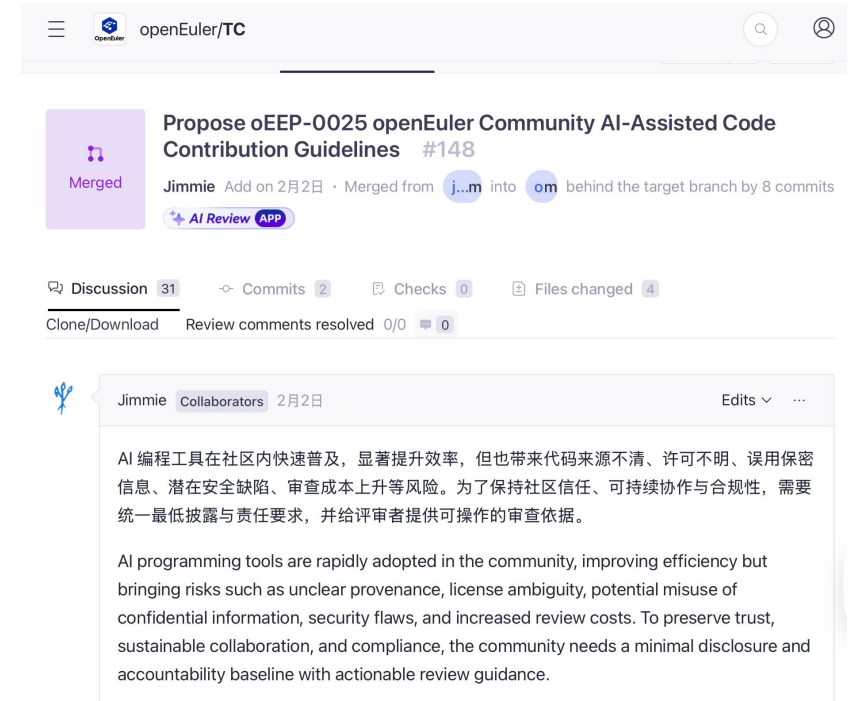
After community discussion and a vote, it was merged on April 15:

- April 15: proposal PR merged: <https://atomgit.com/openeuler/TC/pull/148>
- July 28: official announcement: <https://www.openeuler.org/en/community/ai-coding-assistants/>

The core design:

- Developer and reviewer responsibilities remain unchanged
- AI usage should be transparent and traceable
- PRs and commits can record AI usage details

AI may assist. People remain accountable.



Why This Matters

AI makes it easier to contribute, but it can also **make responsibility harder to see**.

The community needs to know:

- Who submitted the change?
- What did AI help with?
- What was verified by a human?
- What risk should reviewers pay attention to?

CVEs: Security Work at a New Scale

AI changes security work in two directions:

- Attackers and scanners can find more issues faster
- Defenders can also use AI to analyze and repair faster

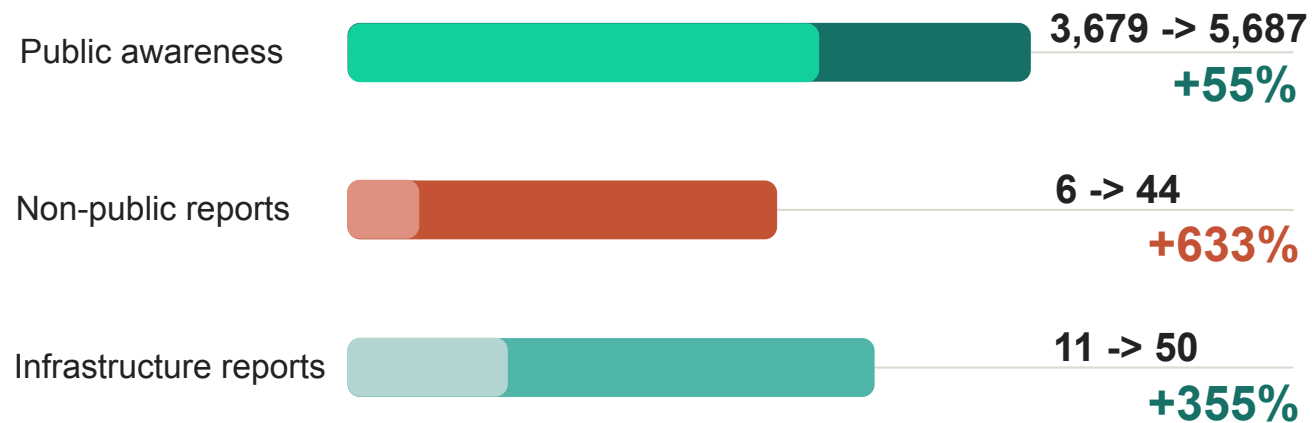
The challenge is not only discovery.

The challenge is turning findings into verified fixes.

CVE Pressure in openEuler

Community tracking data from 2025 H2 to 2026 H1 shows a sharp increase:

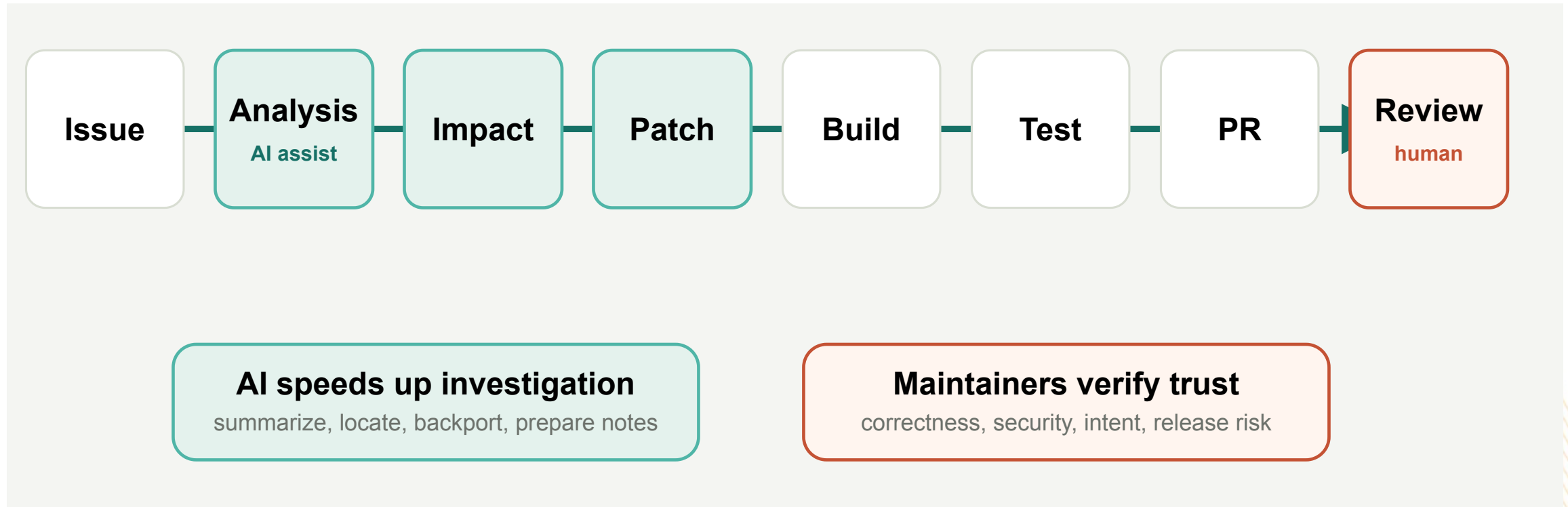
openEuler community tracking data: 2025 H2 to 2026 H1



AI can scale detection faster than a community can scale human judgment.

From CVE to Reviewed Fix

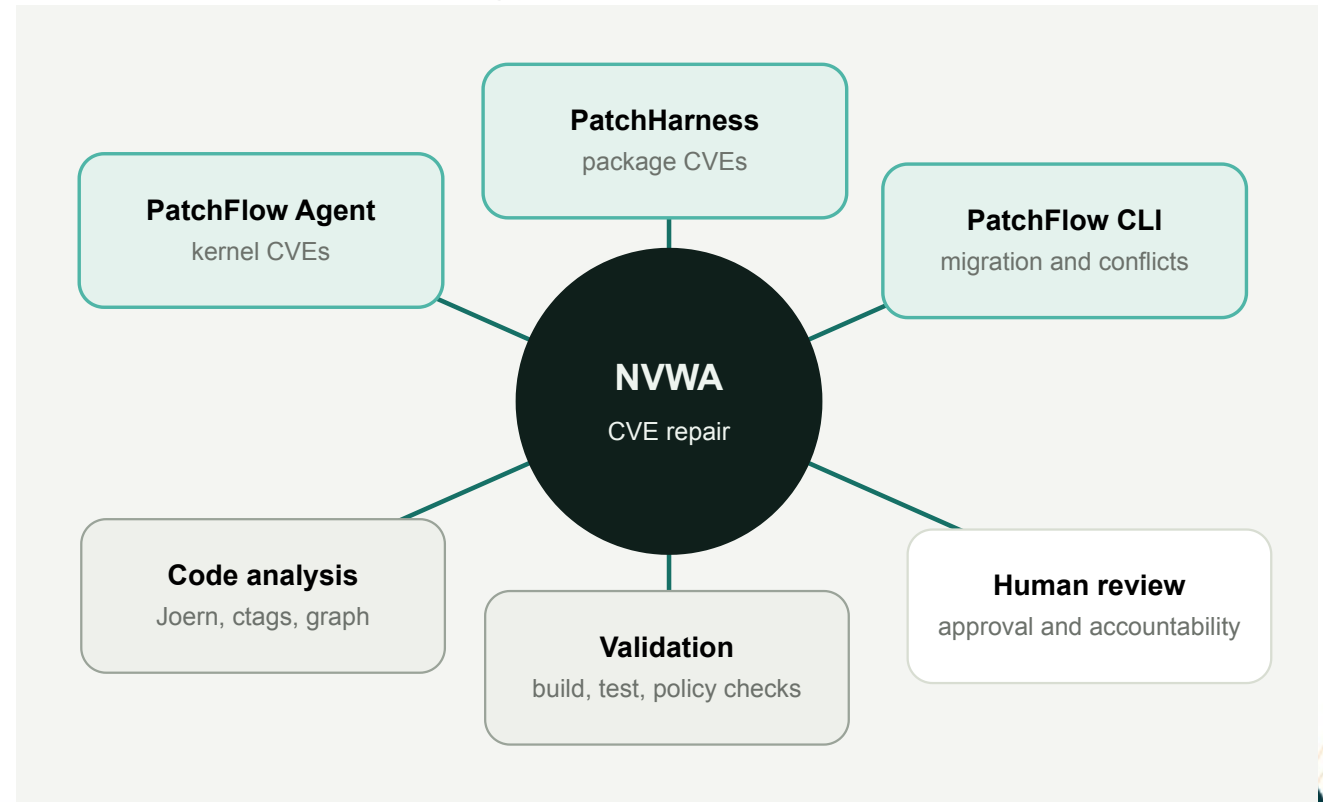
AI helps the workflow, but accountability stays with people.



NVWA: AI-Assisted CVE Repair

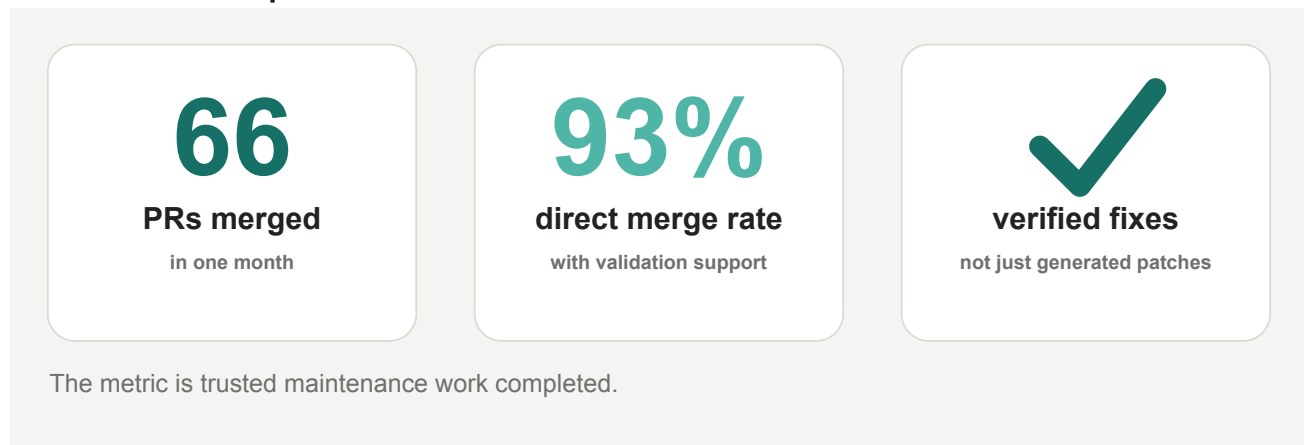
openEuler is exploring [NVWA](#) for CVE repair and conflict handling.

The design keeps AI inside a controlled workflow.



A Practical Result

Recent kernel CVE repair work shows practical results.

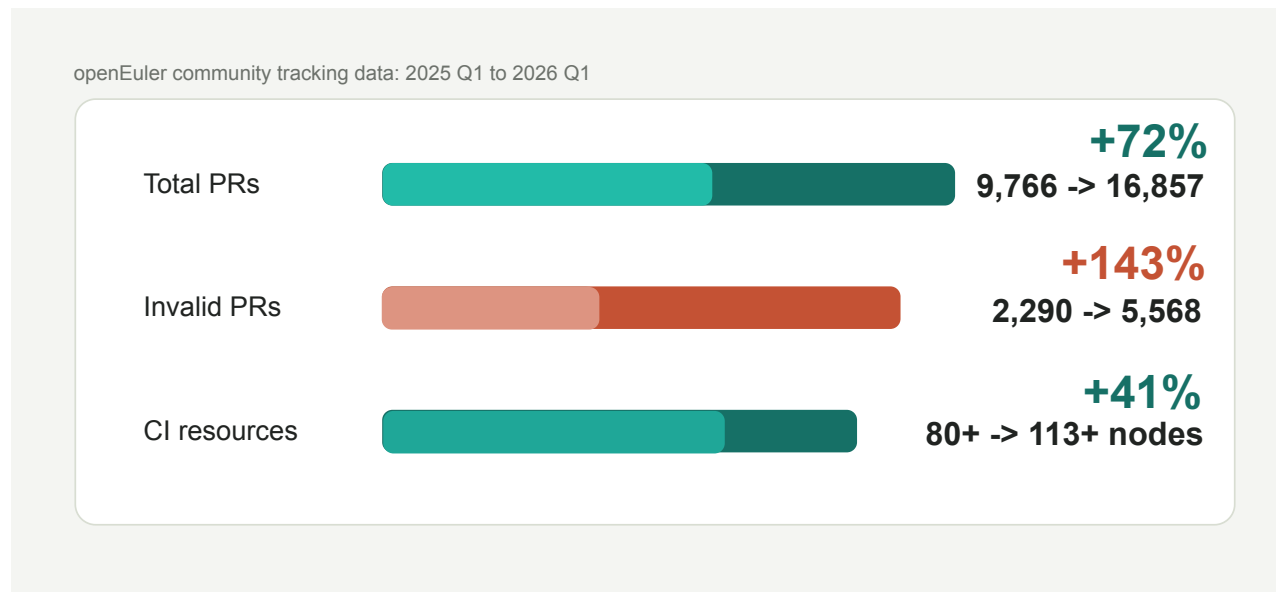


The key metric is not how much AI can generate.

It is how much trusted maintenance work the community can complete.

PR Volume Is Growing Too

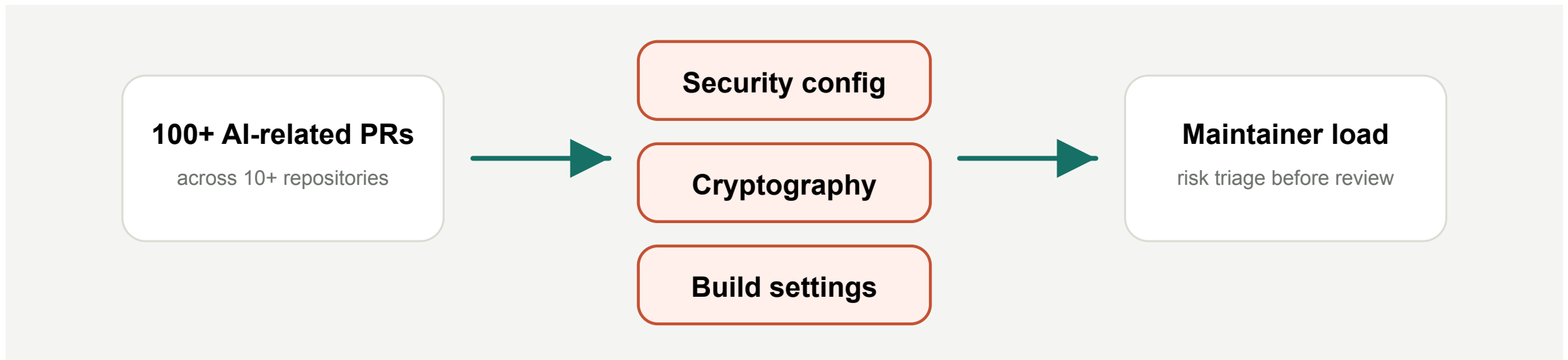
Community tracking data from 2025 Q1 to 2026 Q1 shows:



Generating a contribution is becoming cheaper. Reviewing it is not.

PR Flood Becomes a Governance Problem

In 2026 Q1, openEuler observed AI-related accounts submitting 100+ PRs across 10+ repositories.



Scaling Review and Maintenance

The response cannot be "use more AI" only.

Communities also need:

- Better contribution quality gates
- AI-related PR metadata
- CI checks for required declarations
- Maintainer dashboards
- Risk-based review routing
- Ways to limit low-value automated submissions

Package Maintenance: From Execution to Verification

Before

Discover

Analyze

Modify

Build

Test

Submit

Review

With AI

Detect

AI Analyze

AI Propose

Build/Test

Maintainer Verify

Merge

Deterministic scripts handle repeatable checks. Maintainers handle judgment.

AI on Both Sides of Review

There are two different review questions:

Review AI-generated code

How do maintainers check code that AI helped create?

Use AI to assist review

How can AI help reviewers find issues faster?

In both cases, final approval must remain human-owned.

Opportunities

AI creates real opportunities:

- Reduce repetitive maintainer work
- Improve security response
- Help contributors understand unfamiliar code
- Scale package maintenance
- Assist code review
- Lower barriers to contribution
- Make community knowledge easier to access

Used well, AI gives maintainers more time for work that requires judgment.

Challenges

Communities also need to prepare for:

- PR volume exceeding review capacity
- Plausible but incorrect generated code
- Security and privacy risks
- Licensing and provenance questions
- Over-reliance on automated decisions
- Loss of transparency in community workflows
- Agent intent, behavior, data, and supply chain safety

More automation does not automatically mean a healthier community.

Communities Become More Important

AI does not make open source communities weaker.

It makes trusted communities more important.

When software production becomes faster, the supply chain will ask more than:

What does this code do?

It will also ask:

- Who maintains it?
- How is it maintained?
- Who participates?
- Can the project stay healthy over time?

Three Lessons

1

Principles stay

openness
trust
accountability

2

Workflows evolve

contribution
security
maintenance

3

Governance adapts

rules
metadata
process

Steering AI

AI will continue to evolve faster than any single community policy.

We cannot predict every capability.

But open source communities can decide how those capabilities become part of their work.

The goal is not simply to adopt more AI.

It is to steer AI toward healthier, scalable, and trustworthy collaboration.

Thank You!

May the open force be with you