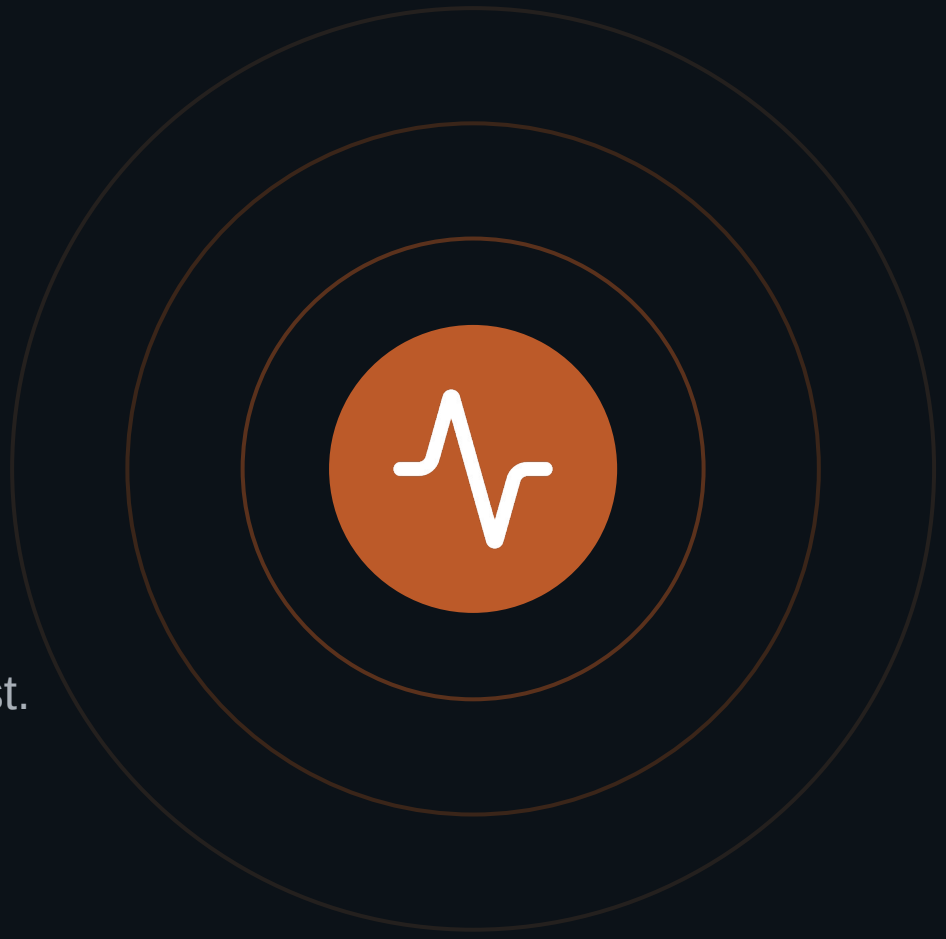
Architect

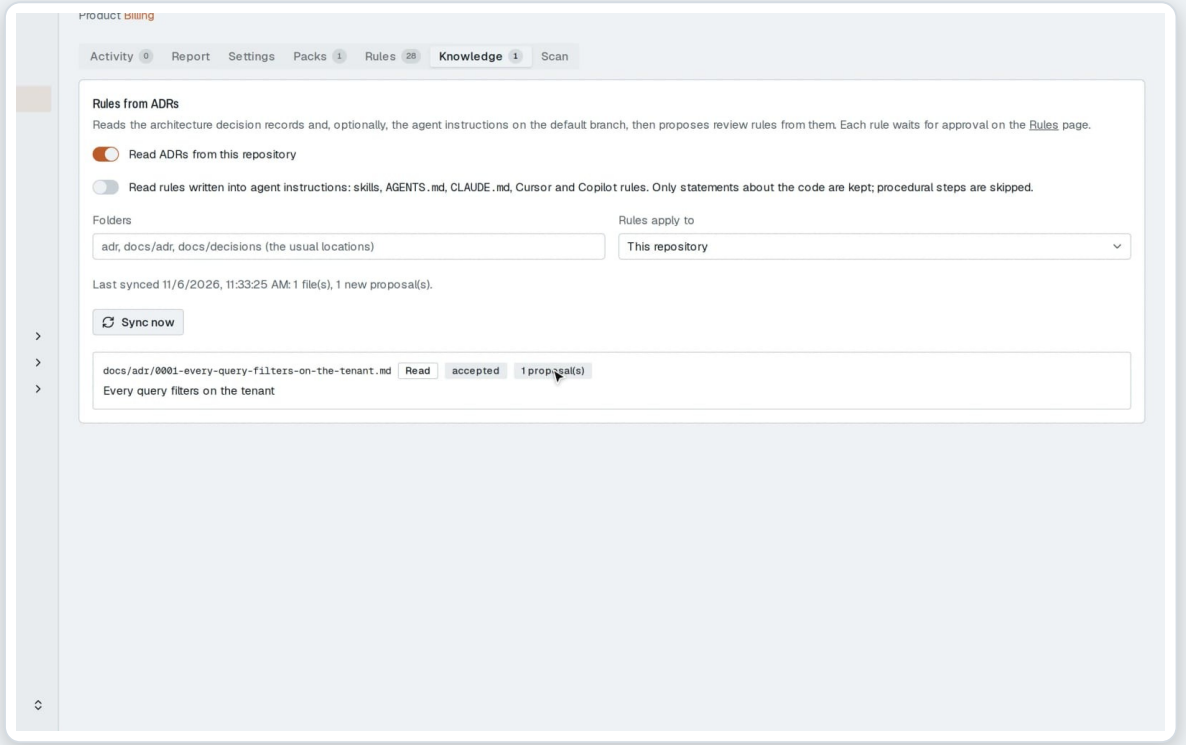


Your ADRs turned into rules that every pull request is checked against.

Your architecture decisions become rules

“As an architect, I want the decisions in our architecture decision records (ADRs) checked on every pull request, so the team still follows them a year later.”

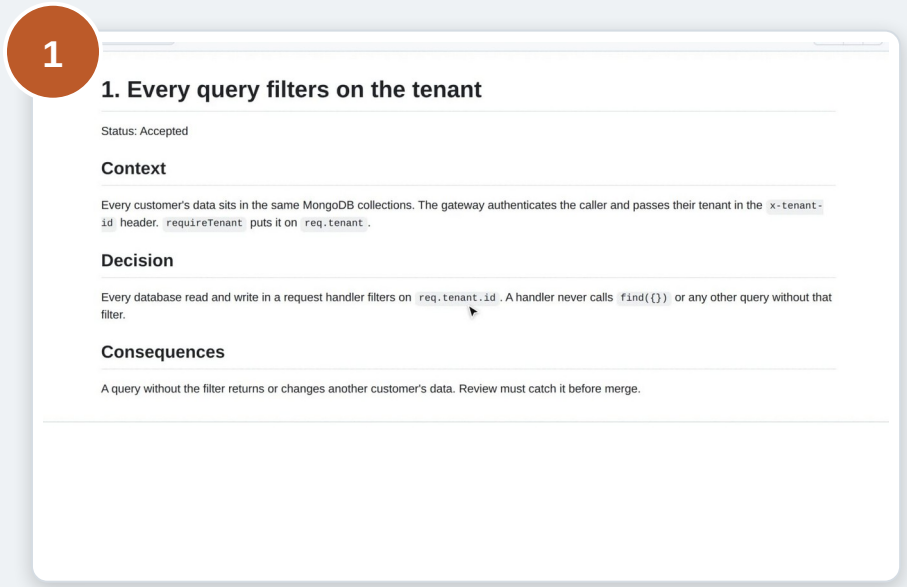
An ADR records a decision once. Nobody checks it later, and new team members never read it.



What to notice

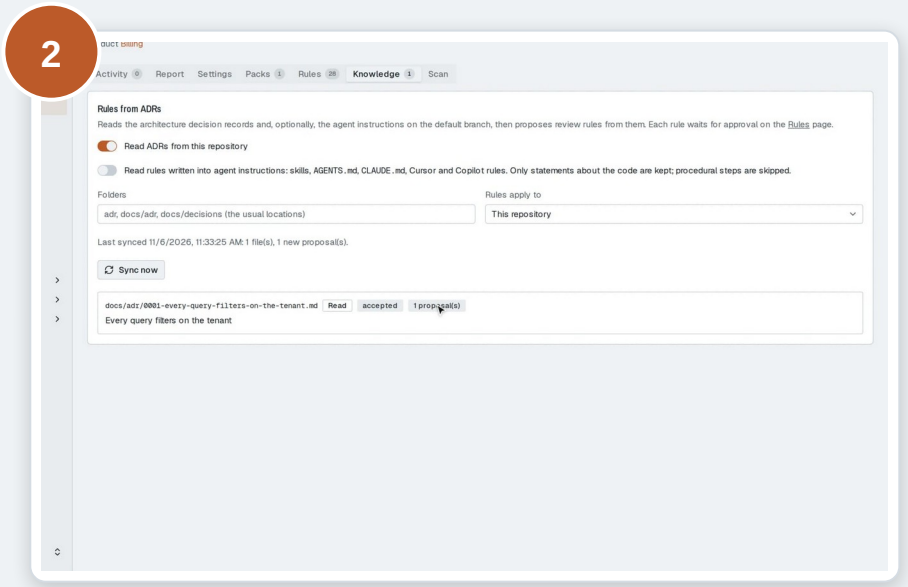
- The draft quotes the ADR, so a reviewer can compare it with the source.
- The rule does nothing until a person applies it.
- The rule shows which ADR it comes from.

Your architecture decisions become rules



Your team writes its decisions down, as ADRs in the repository.

ADR 0001 in the repository says: every database query must read and write only the caller's own customer data.

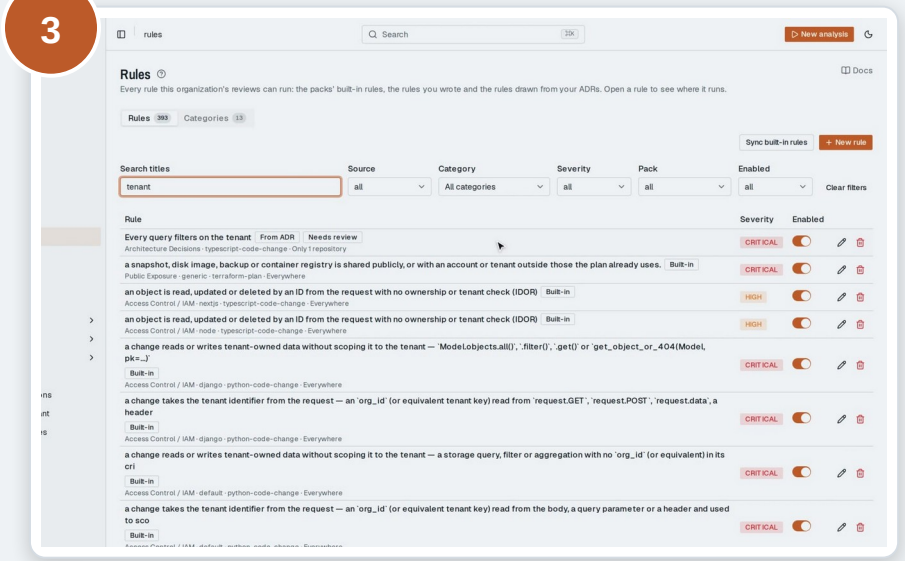


OpenTremor reads them and drafts a rule from each decision.

The setting "Read ADRs from this repository" is on. OpenTremor reads the file and drafts a rule. The draft quotes the sentence it comes from.

Your architecture decisions become rules

3



A person applies the rule. Nothing applies on its own.

A person reviews and applies the draft. It never applies by itself.

4



The next pull request is held to the decision.

The next pull request with a query that reads every customer's data gets a finding for that rule.

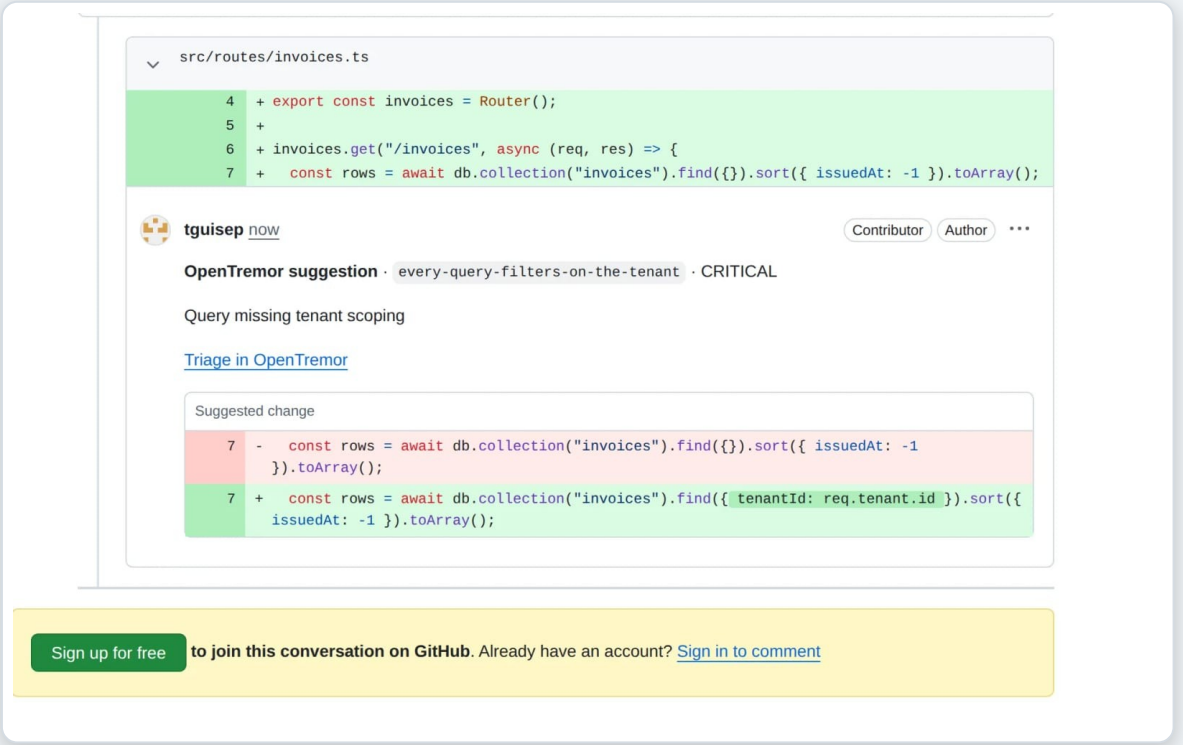
Every pull request checked against your team's rules

“As a tech lead, I want every pull request checked against our written standards, so reviewers stop repeating the same comments.”

Your standards are in a wiki and in people's heads. A pull request follows them only if the reviewer knows them and has time.

What to notice

- The rule is your team's own, written in plain words. You do not need a special rule language.
- The finding explains the risk, not only the line of code.
- The fix is a normal GitHub suggestion. OpenTremor never commits code for you.



Every pull request checked against your team's rules

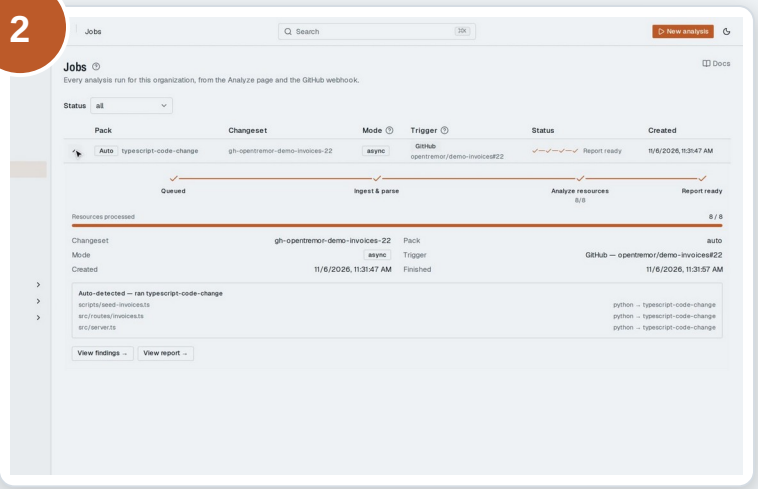
1



A developer opens a pull request.

The pull request adds a list of invoices. Its database query returns the invoices of all customers, not only the caller's.

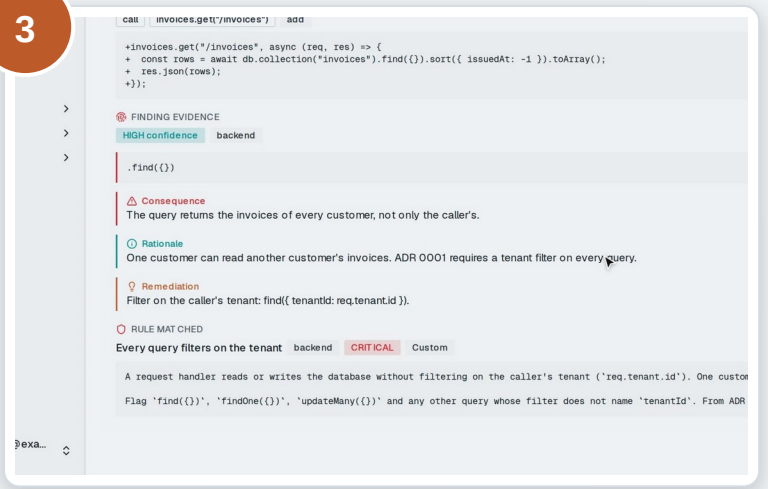
2



OpenTremor reviews it against your team's rules.

OpenTremor receives the pull request and checks the changed code against the organization's rules.

3

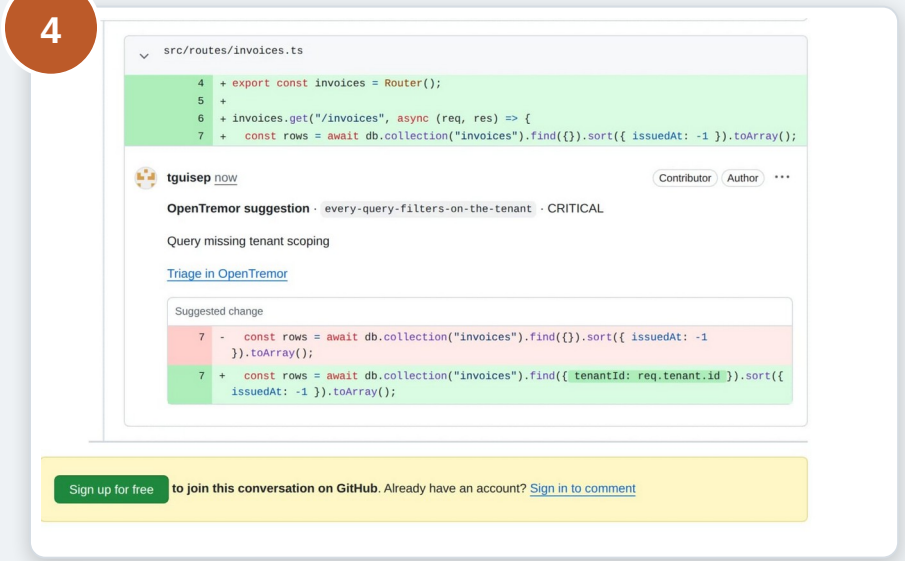


Each finding names the rule it breaks and why it matters.

The finding names the broken rule, "Every query filters on the tenant". It also explains the risk if the code goes to production.

Every pull request checked against your team's rules

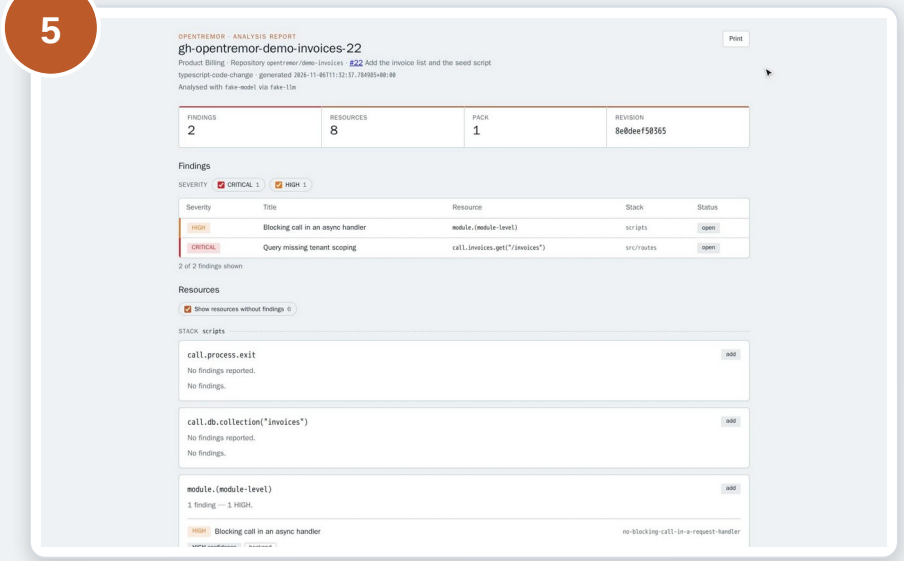
4



The fix comes back to the pull request, ready to commit.

For a critical finding, OpenTremor suggests the fix on GitHub. The developer accepts it from the pull request.

5



The full report, shared with one link.

The report covers the whole change. Anyone with the link can read it until the link expires.

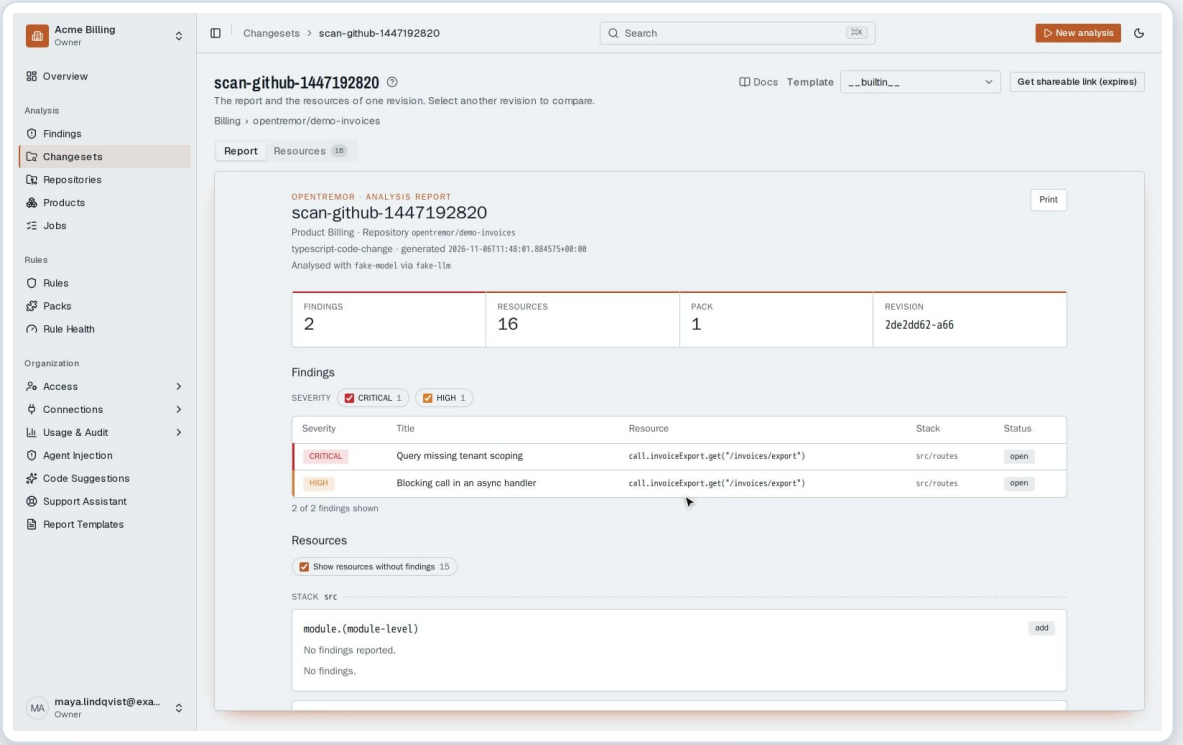
Find problems already in your code

“As a platform engineer, I want to check an existing repository against our standards, so I can find its problems before they cause an incident.”

Pull request review sees only the changed code. Nobody checks the older code, so nobody knows how much of it breaks the rules.

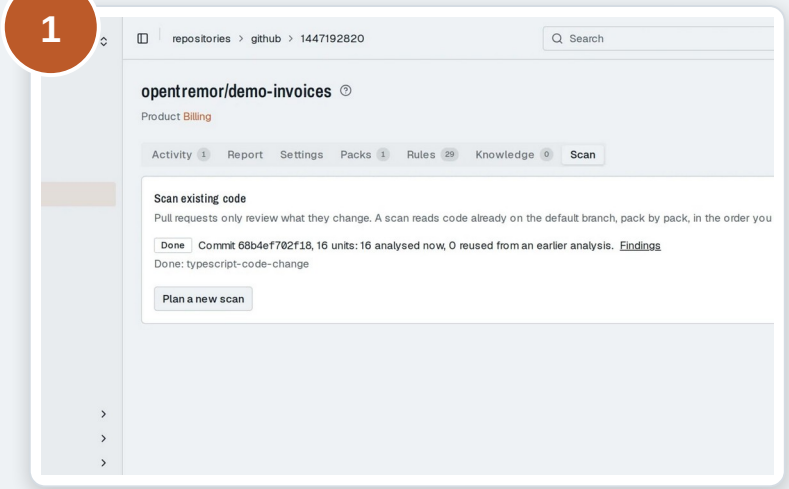
What to notice

- A scan never blocks a pull request. It only shows the current state of the code.
- OpenTremor never sends the same code to the AI model twice.



Find problems already in your code

1



A platform engineer scans the code already on the default branch.

The repository's Scan tab shows what a scan will check. One click scans the main branch.

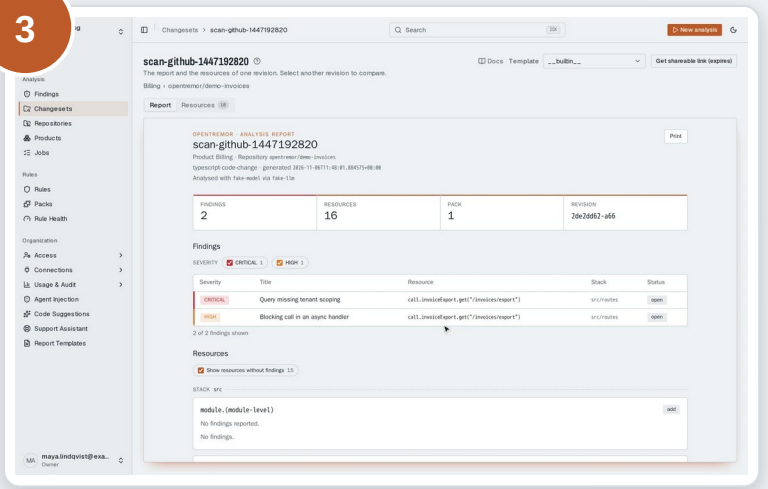
2



OpenTremor finds what the existing code breaks, rule by rule.

The scan finds an old CSV export that returns the invoices of all customers. Each finding names the rule and the risk.

3



The scan's report sorts the findings by severity: a backlog to work through.

The scan report counts the findings by severity: one critical, one high. The team can plan the work and assign it.

NEXT

Read more

Rules from ADRs

Which documents are read, and how a draft becomes a rule.

docs.opentremor.dev/configuration/rules-from-adrs/

How rules apply

Which rules a review runs, and where each setting lives.

docs.opentremor.dev/configuration/rules/

See it in motion



opentremor.dev/tour/for/architect · docs.opentremor.dev