

# CPACHECKER Development in 2026

**Philipp Wendler**

SoSy-Lab @ LMU Munich

CPA'26 2026-10-16



# More Explicit Test Structure

3 basic categories of tests in CPAchecker:

- 1) Unit tests, written with JUnit
- 2) Small-scale integration tests, written with JUnit
  - a) some with stronger checks that BuildBot does not support
  - b) some redundant, just for convenience of developers
- 3) Large-scale integration tests, done with BuildBot+BenchExec+BenchCloud

# More Explicit Test Structure

3 basic categories of tests in CPAchecker:

- 1) Unit tests, written with JUnit
- 2) Small-scale integration tests, written with JUnit
  - a) some with stronger checks that BuildBot does not support
  - b) some redundant, just for convenience of developers
- 3) Large-scale integration tests, done with BuildBot+BenchExec+BenchCloud

Improvements:

- ▶ All JUnit tests categorized appropriately
- ▶ [Documentation](#) for test writers
- ▶ Distinction reflected in Ant tasks and CI, individually executable
- ▶ 1) + 2a) executed by default, 2b) not  
⇒ Can be added without increasing test time too much, but prefer BuildBot!

# Further Test and CI Improvements

- ▶ CI faster due to splitting and parallelization of slowest check
- ▶ Most pipelines run slightly less tests now, faster
- ▶ Merge trains run more tests now
- ▶ Improved [documentation](#)

Further suggestions and ideas welcome!

# BuildBot Integration Tests

- ▶ Prioritize adding tests for new analyses!  
Still several important cases without integration tests.
- ▶ Tests need maintenance and attention, too!
- ▶ Subscribe to [mailing lists](#)!
- ▶ Check BuildBot results for regressions!
- ▶ If not done recently:  
Check BuildBot results for expected state!
  - ▶ Meaningful selection of benchmarks?
  - ▶ Unknown problems visible in current set of results?

# Java Version Required for CPACHECKER

- ▶ Java 11 required since December 2019 (15 months after release)
- ▶ Java 17 required since January 2023 (16 months after release)
- ▶ Java 21 required since August 2025 (23 months after release)

# Java 25

- ▶ Java 25 (LTS) released 2025-09-16
- ▶ Not many relevant language features:
  - ▶ Unnamed variables
  - ▶ Foreign Function API as JNI alternative (only relevant if JavaSMT adopts it)
  - ▶ Markdown as JavaDoc alternative (ugly syntax?)
  - ▶ Stream Gatherers
  - ▶ Flexible Constructor Bodies
  - ▶ Scoped Values

# Switch to Java 25?

- ▶ In Debian 13 (stable), oldstable out of regular support
- ▶ In Ubuntu 22.04 and newer
- ▶ CI tools support Java 25.
- ▶ Proposals:
  1. Require Java 25 after this year's release
  2. Require Java 25 after next year's release
  3. Skip Java 25 and only require Java 29 in 2028

# General Java Upgrade Policy

- ▶ Java LTS releases in September of odd years
- ▶ Ubuntu backports LTS releases ([source](#))
- ▶ Debian releases in June-August of odd years, sometimes with latest Java LTS, sometimes without
- ▶ Potential policy:  
Require in CPAChecker after autumn release in even years?  
(12-14 months later)

# LLMs in CPAchecker development

## Guidelines:

- ▶ Usage according to personal preference
- ▶ General documentation instead of LLM-focused texts  
e.g., tutorials, development guidelines, etc.  
Submit if you have such texts!
- ▶ Own review before sending generated code to others
- ▶ No “meat proxies”: interactions between humans  
instead of forwarding LLM output

Written policy needed?

**Thank you for all your contributions  
and cooperation!**

# Long-Term Development Plans (No Progress)

- ▶ Migrate nullability annotations to use JSpecify
- ▶ Fully enforce nullability annotations
- ▶ Make CPAchecker a “modularized” Java project that is compatible with the JPMS ([#1344](#))
  - ⇒ Allows more uses of `sealed` and restricting reflection.

Feedback and contributions welcome!