Integrating ECLAIR static analysis in IDEs using the Language Server Protocol

Integrazione dell'analizzatore statico ECLAIR in IDE tramite il Language Server Protocol

Nicolò Fuccella

8 settembre 2022

The Goal

- ECLAIR is a powerful platform for software verification with a strong focus on the development of high-integrity systems, including safety-and security-critical systems.

- At the moment the check is performed as a separate task, instead we want to integrate the quality assurance from the earlier phases of the development, giving feedback to developers about their code correctness while they're writing the code, directly in the IDE.

- This thesis aims at providing a proof-of-concept of "immediate feedback software verification" using ECLAIR.

The Issues

The transition from the traditional way in which static analysis results are consumed to this “immediate feedback” modality presents some challenges:

- analysis time becomes critical
- proliferation of coding environments can lead to a maintenance nightmare
- analysis output must be stored in an convenient format in order to be queried

Preliminary Notions

In order to fully grasp the project, it is important to briefly introduce:

- program correctness verification
- shift-left movement
- ECLAIR
- the Language Server Protocol

Program correctness verification

- empirical testing limitations
- static analysis pros and cons
- static analysis integrations

Shift-left movement

"Shift-left testing is how I refer to a better way of integrating the quality assurance (QA) and development parts of a software project. By linking these two functions at lower levels of management, you can expand your testing program while reducing manpower and equipment needs - sometimes by as much as an order of magnitude." [1]

"Here is the dilemma in software development: defects are expensive, but eliminating defects is also expensive. However, most defects end up costing more than it would have cost to prevent them." [2]

[1] Larry Smith. Shift-left testing. Dr. Dobb's J., sep 2001.

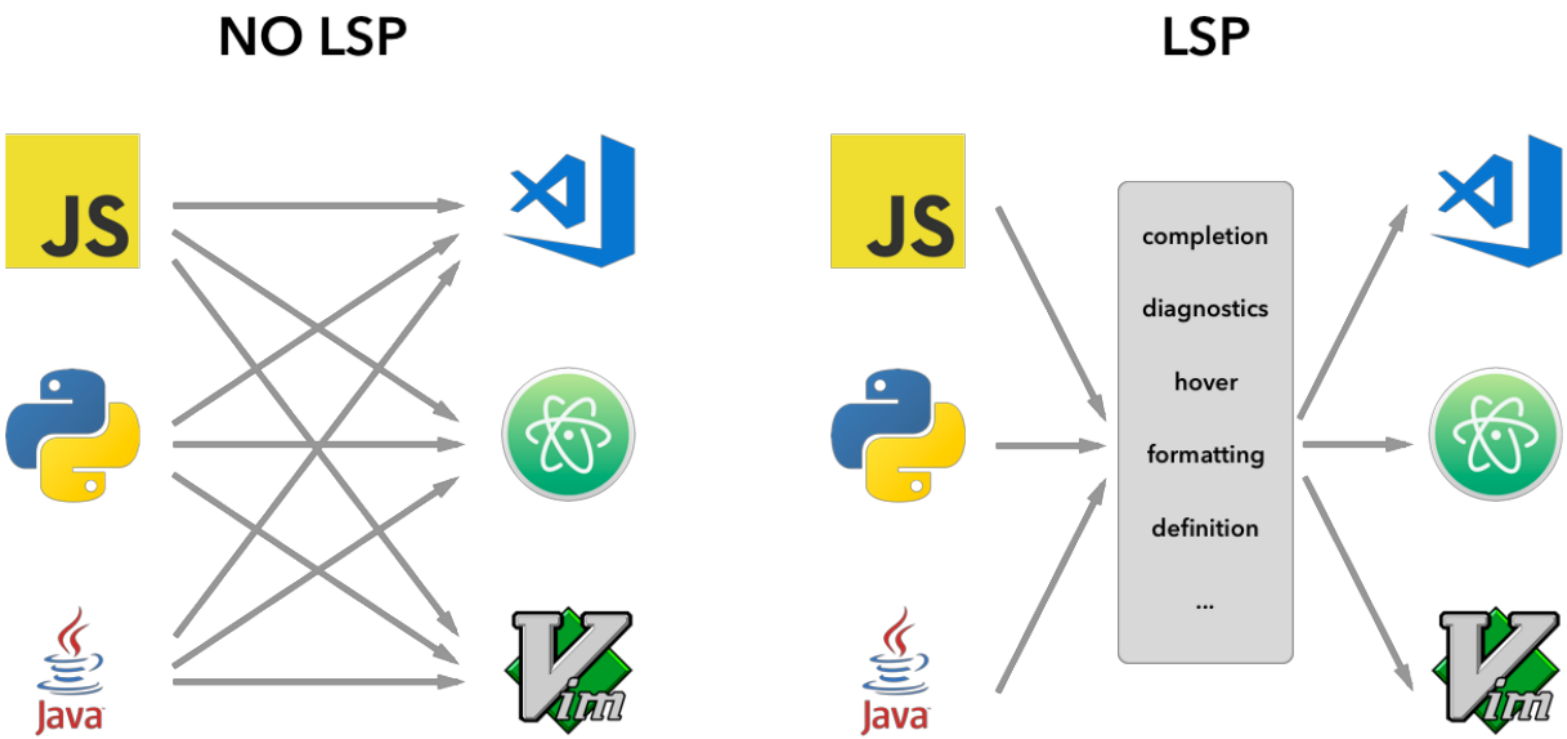
[2] K. Beck, C. Andres, and E. Gamma. Extreme Programming Explained: Embrace Change., 2004.

ECLAIR

ECLAIR is a powerful platform for software verification. It works on the desktop and on the server to analyze entire projects to:

- automatically detect important classes of software errors
- validate coding rules, with a particular emphasis on the MISRA coding standards and others
- compute software metrics
- automatically generate tests

The Language Server Protocol



Starting point

- The first step was to understand the ECLAIR architecture on top of which we were going to build the prototype.
- Then we got our hands dirty with a first experiment to fully grasp the potentialities of the Language Server Protocol.
- Finally, we had a clear idea of the issues that needed a more in depth analysis.

The ECLAIR architecture

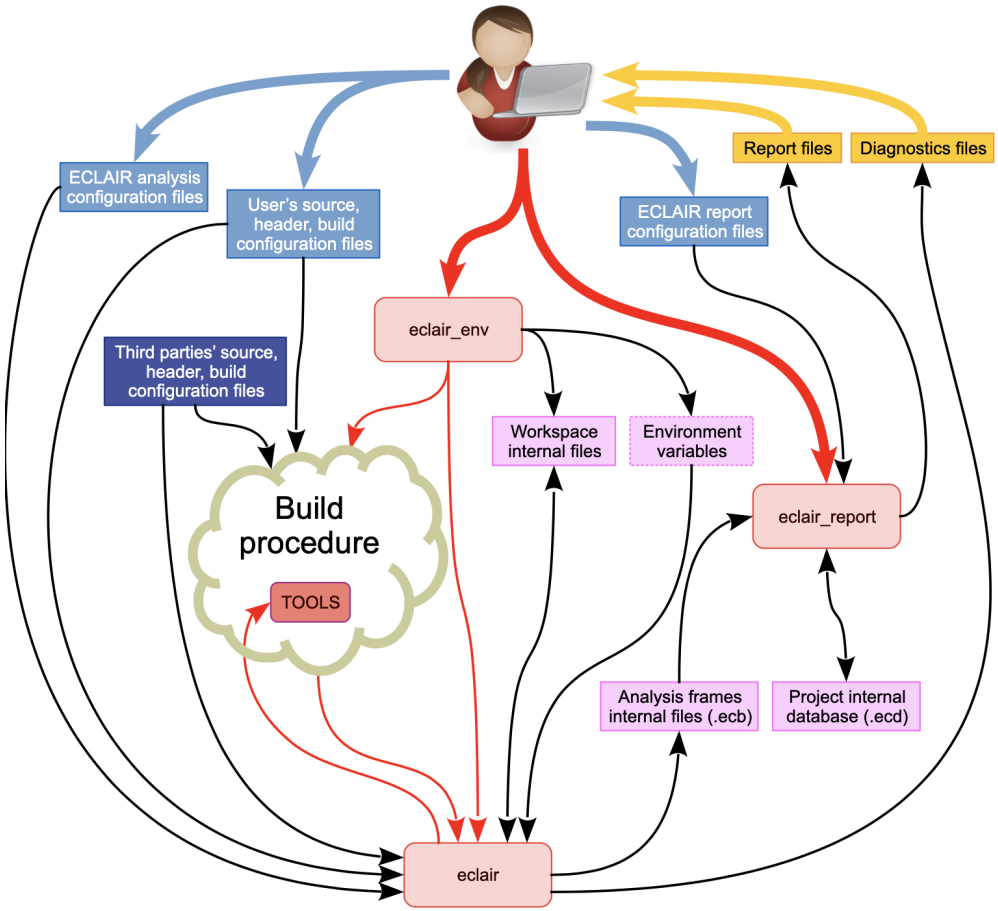
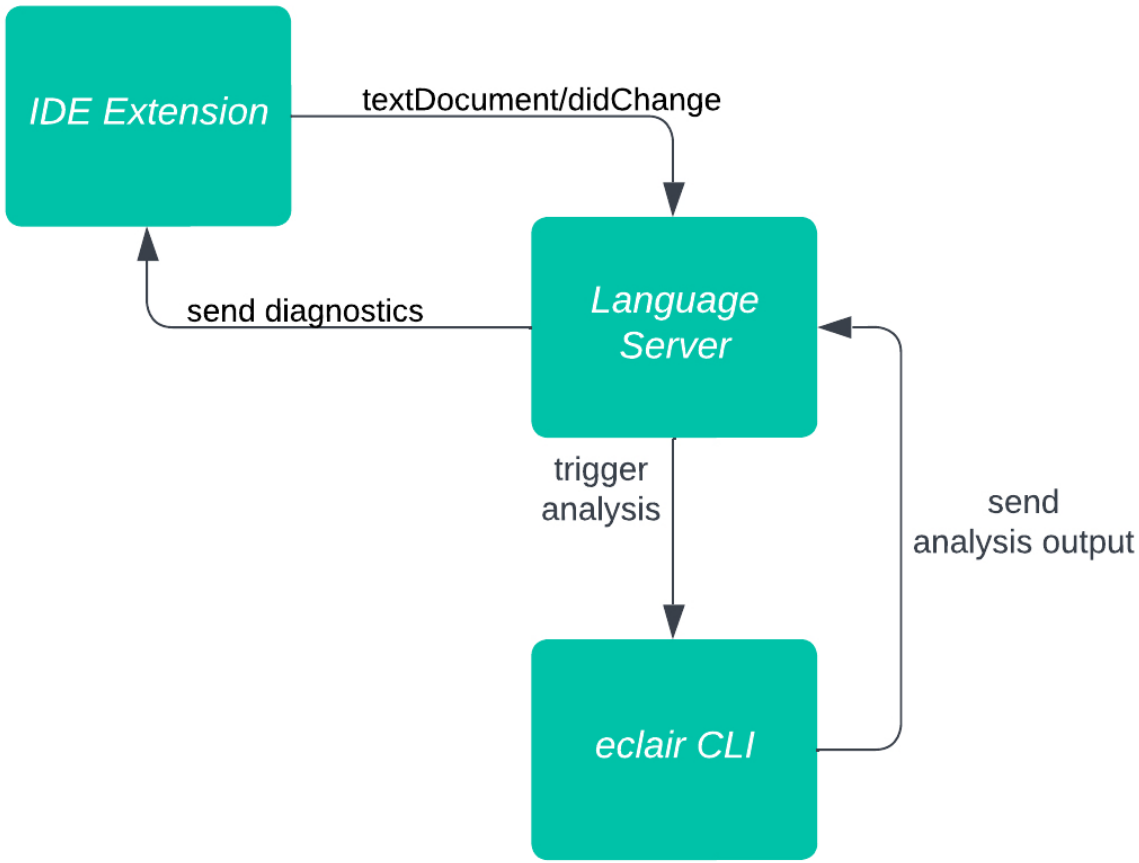


Figure: Image copyright by BUGSENG srl, reproduced with permission.

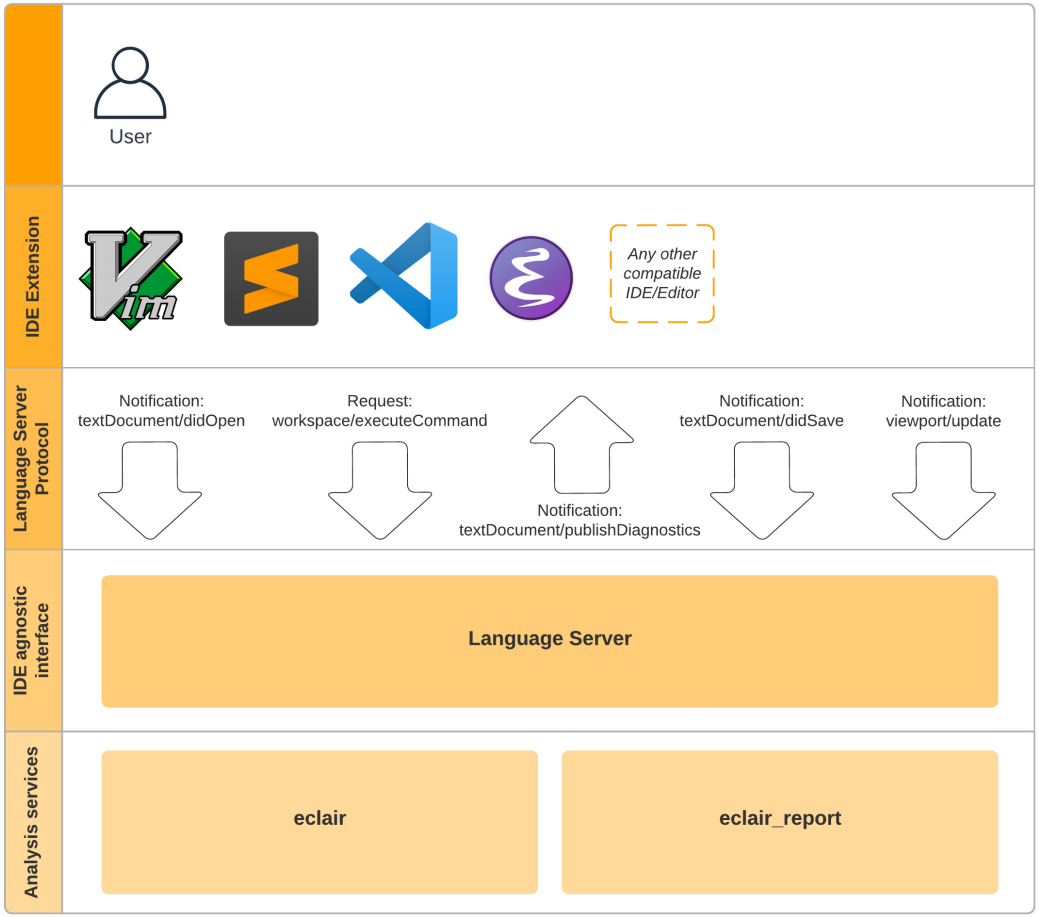
The first experiment - Overview



The first experiment - Lessons learned

- the analysis cannot be performed whenever the user changes something
- violations cannot be returned all at once
- the Language Server Protocol reduces dramatically the lines of code needed
- some IDE specific features must still be implement from scratch

Project architecture

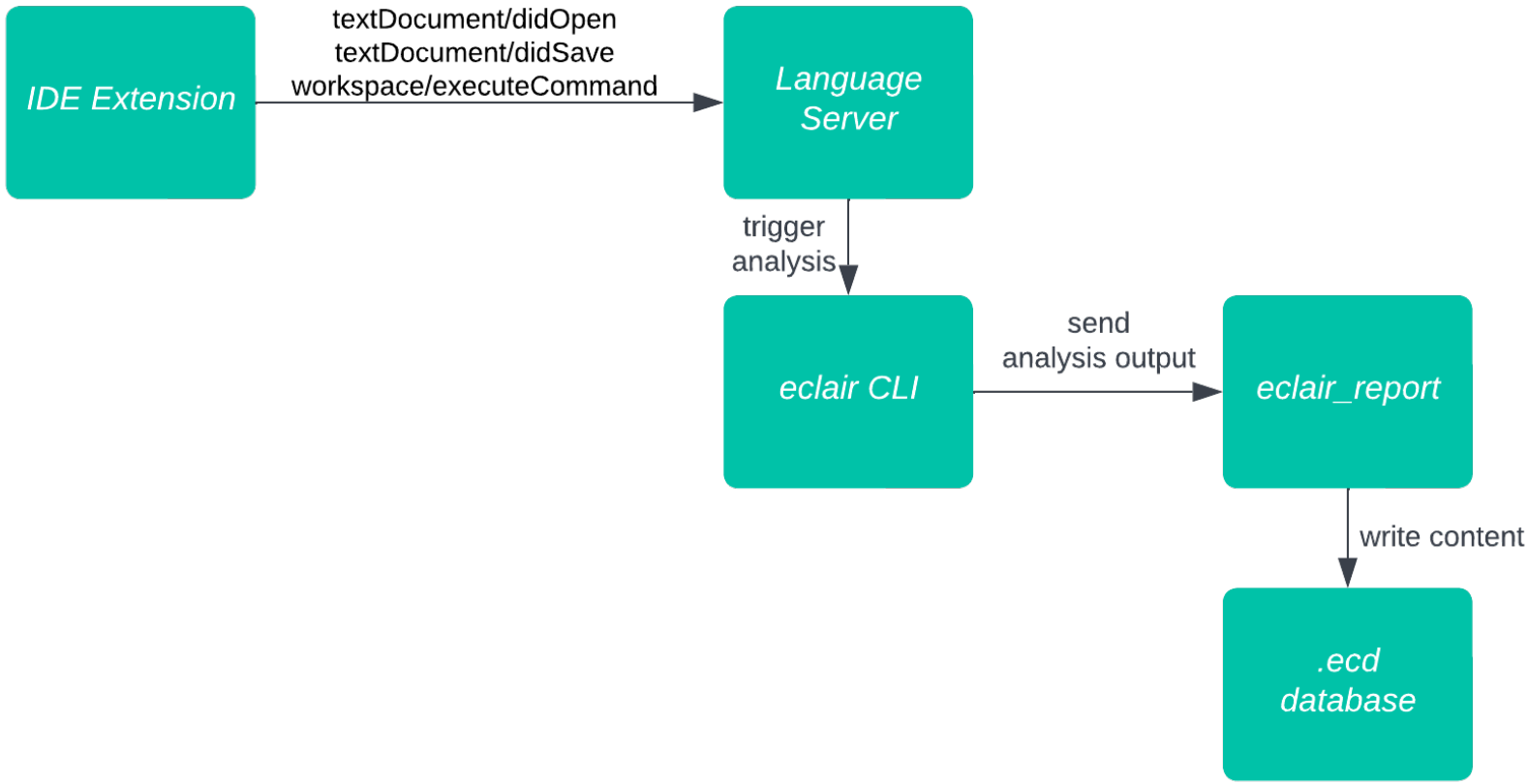


Project architecture - Components

Follows an analysis of each component's role:

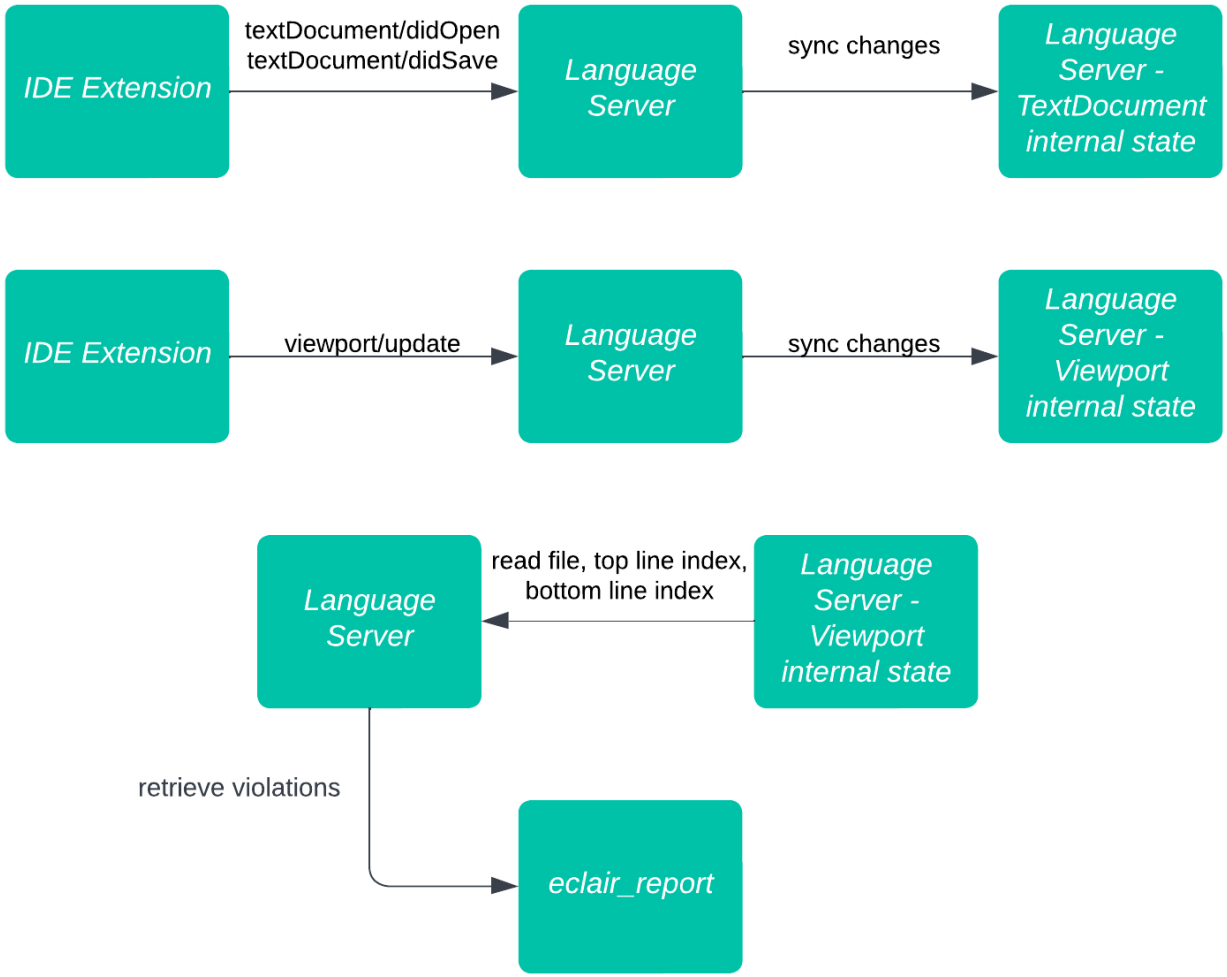
- *eclair* CLI
- *eclair_report*
- Language Server

eclair CLI

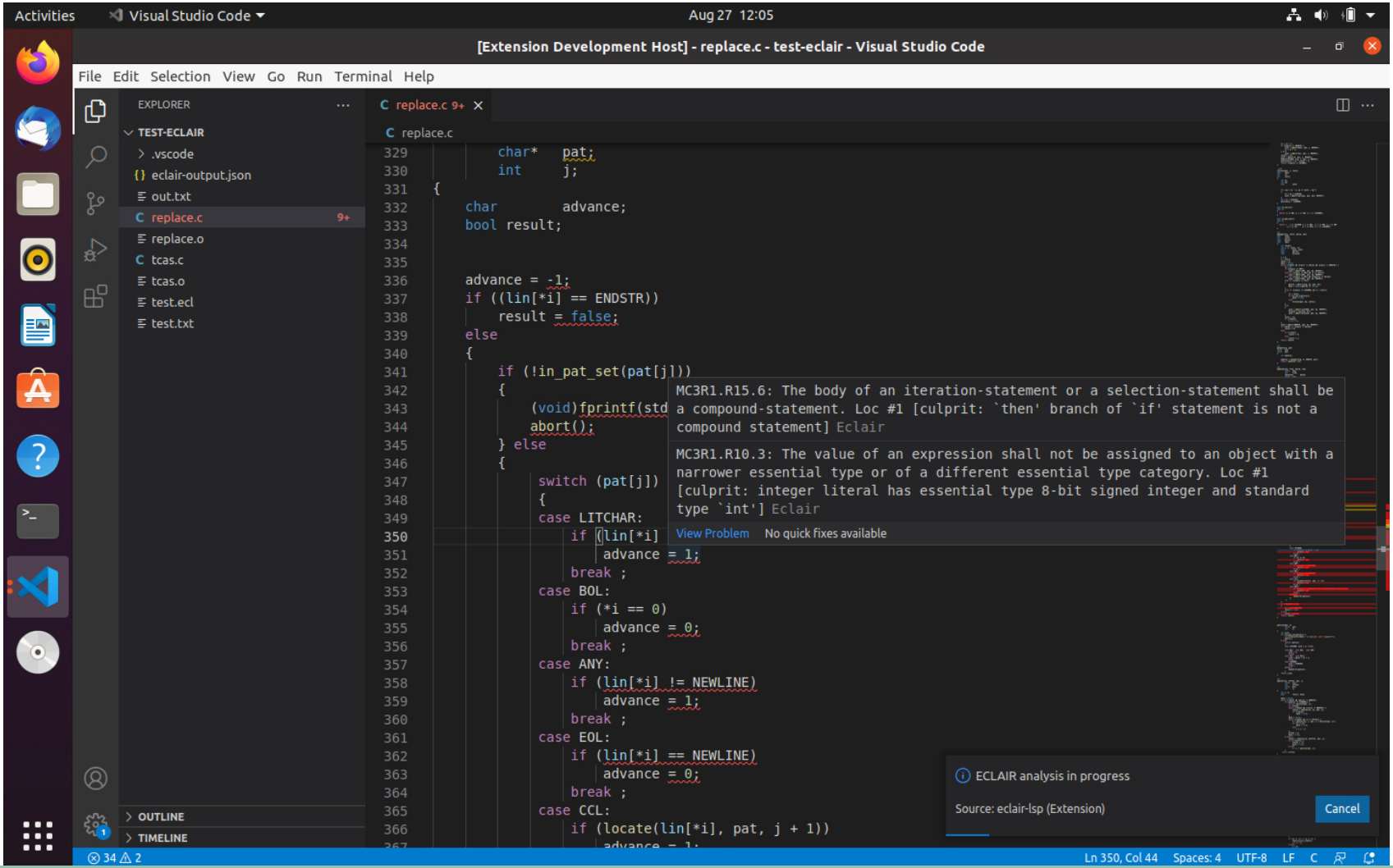

IDE Extension
textDocument/didOpen
textDocument/didSave
workspace/executeCommand
Language Server
trigger analysis
eclair CLI
send analysis output
eclair_report
write content
.ecd database

eclair_report

Language Server



The final result



The End

Thank you for your attention.