

“

The Chisel Method for building parsers



Let's start with an experiment



66 I guess you are thinking

- Parsers? What a boring topic. Parsing is a solved problem!
- I learned what I needed to learn about parsing 10 or 20 or 30 years ago at the university



“



Like flying!



“



Flying: same but different



66 What is wrong with building parsers?

- It is perceived by many as a black magic art
- Clients have not a clear idea of what a parser does
- It is a trial and error process
- It takes a lot of time to train people
- There are repetitive tasks involved
- Parsers are not easy to integrate



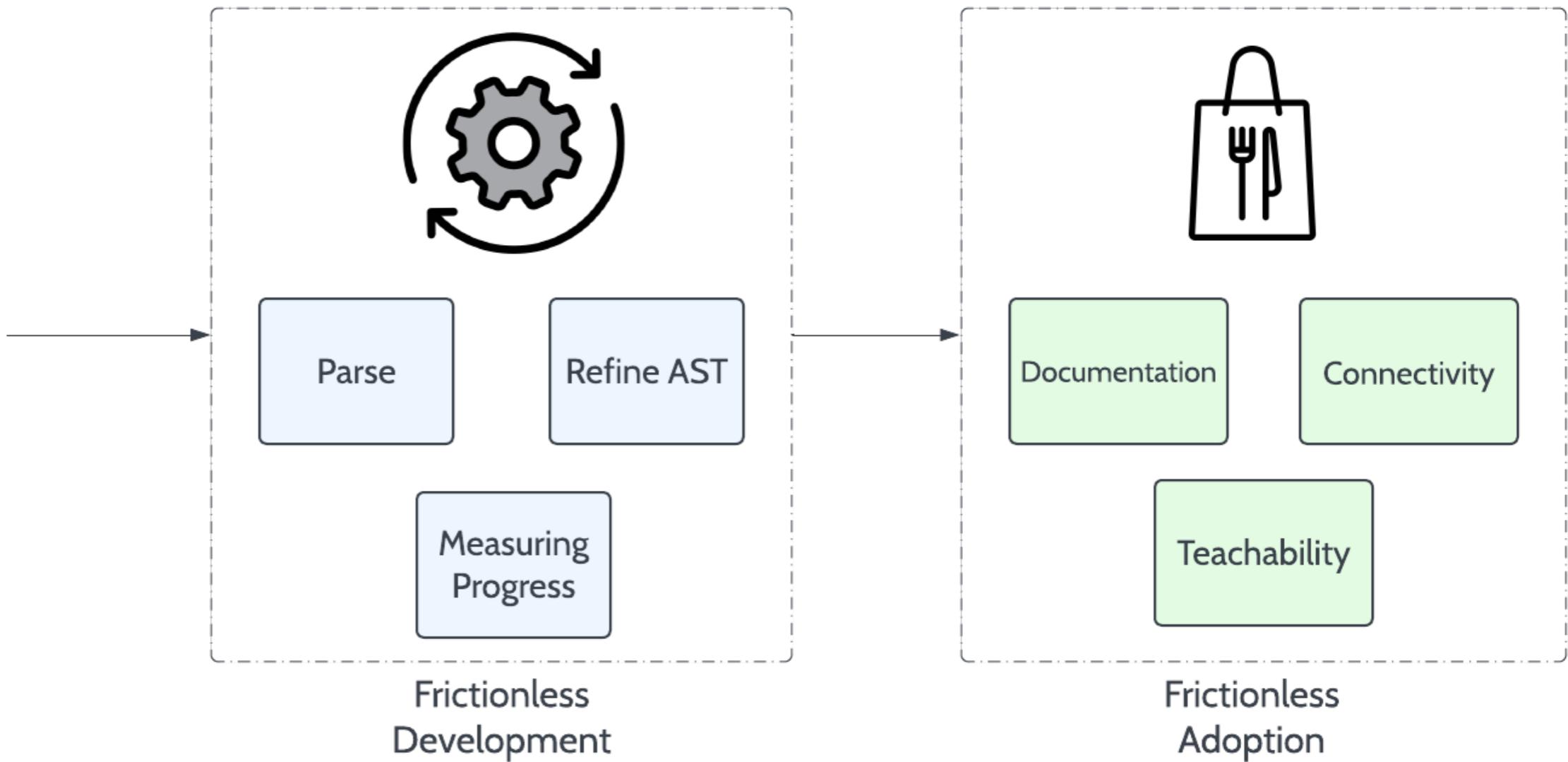
What can we do?



“



Define the goal



The Chisel Method



“



Define the goal

The Chisel Method - Goal setting



66 Start with the end in mind

The screenshot shows a GitHub search interface with the following parameters in the search bar: `language:Java stars:>1000 forks:>500`. The sidebar on the left includes sections for **VWikis**, **Topics**, **Marketplace**, **Languages** (with options for JavaScript, Python, Java, TypeScript, Go, C++, C, PHP, C#, and HTML), and **Advanced** filters (Owner, Size, Number of followers, Number of forks, Number of stars, Date created, Date pushed, Topic, License, Archived). The main area displays 3.3k results in 360 ms, sorted by Most stars. The results list includes:

- TheAlgorithms/Java**: All Algorithms implemented in Java. Tags: search, java, algorithm, algorithms, sort. Updated 1 hour ago.
- spring-projects/spring-framework**: Spring Framework. Tags: framework, spring, spring-framework. Updated 14 hours ago.
- google/guava**: Google core libraries for Java. Tags: java, guava. Updated 10 hours ago.
- ReactiveX/RxJava**: RxJava – Reactive Extensions for the JVM – a library for composing asynchronous and event-based programs using observable sequences for t... Tags: java, flow, rxjava, reactive-streams. Updated yesterday.
- NationalSecurityAgency/ghidra**: Ghidra is a software reverse engineering (SRE) framework. Tags: reverse-engineering, disassembler, software-analysis. Updated 1 hour ago.



66 Quality checks

1. We can produce an AST for each single valid example.

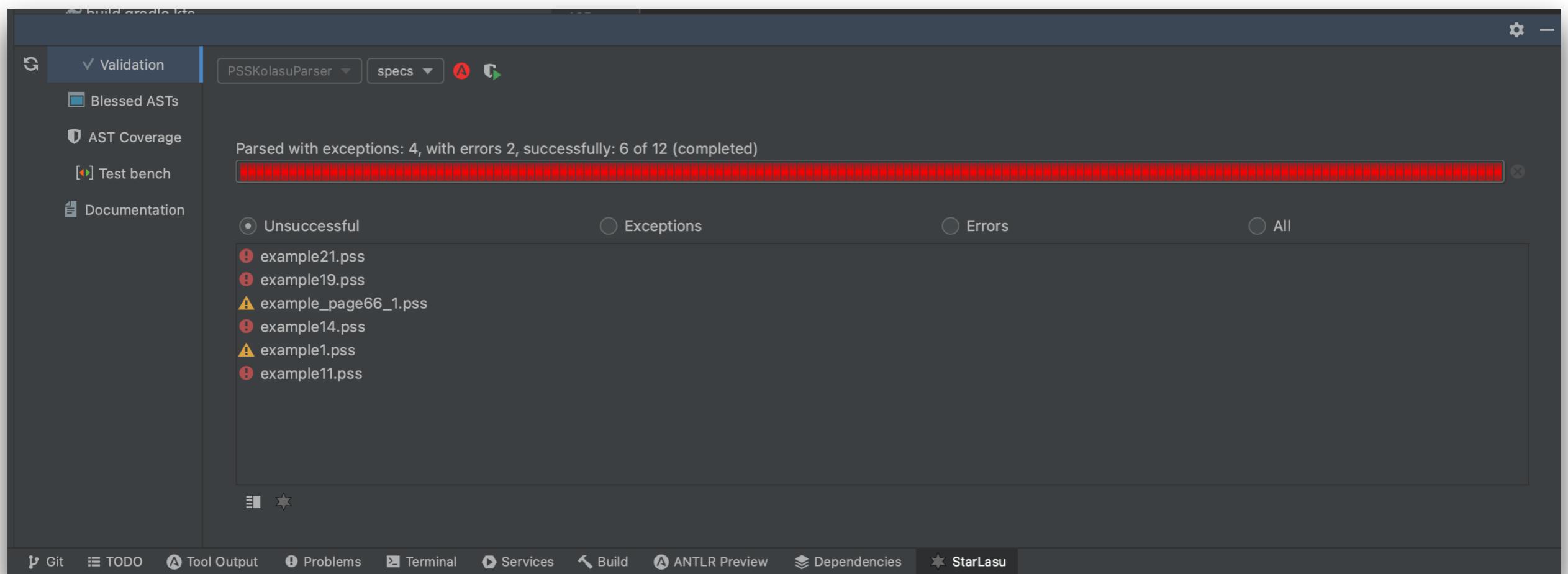
This ensures we can parse.

2. Each Concept is validated by at least one blessed example.

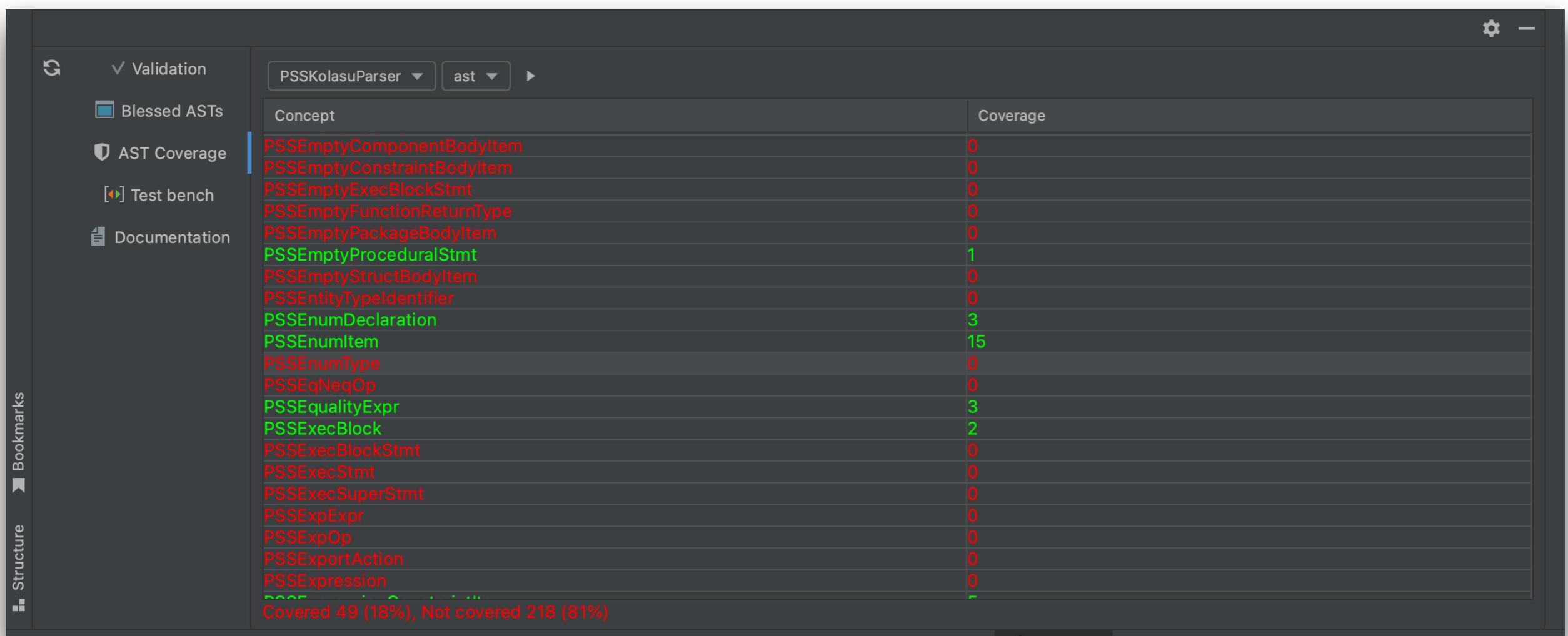
This ensures that the model of the code produced is valuable.



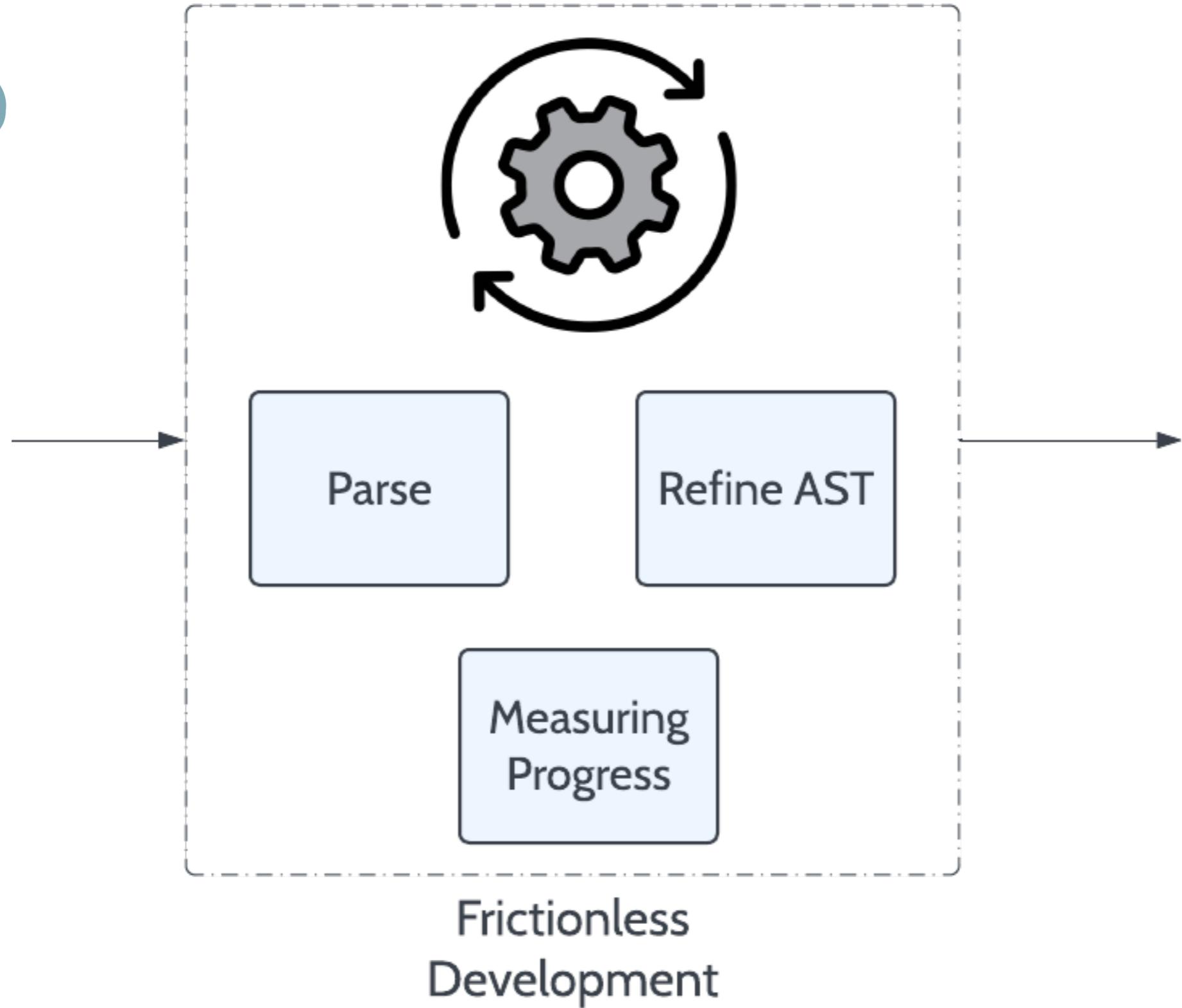
66 Quality checks



66 Quality checks



“



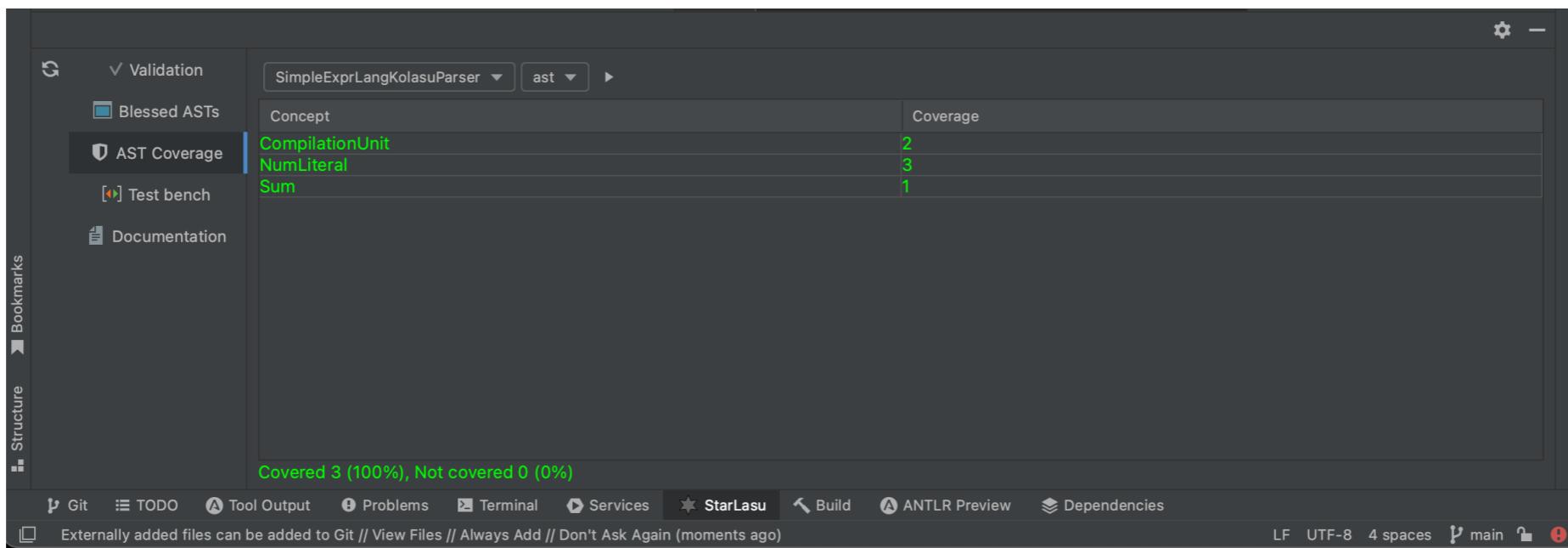
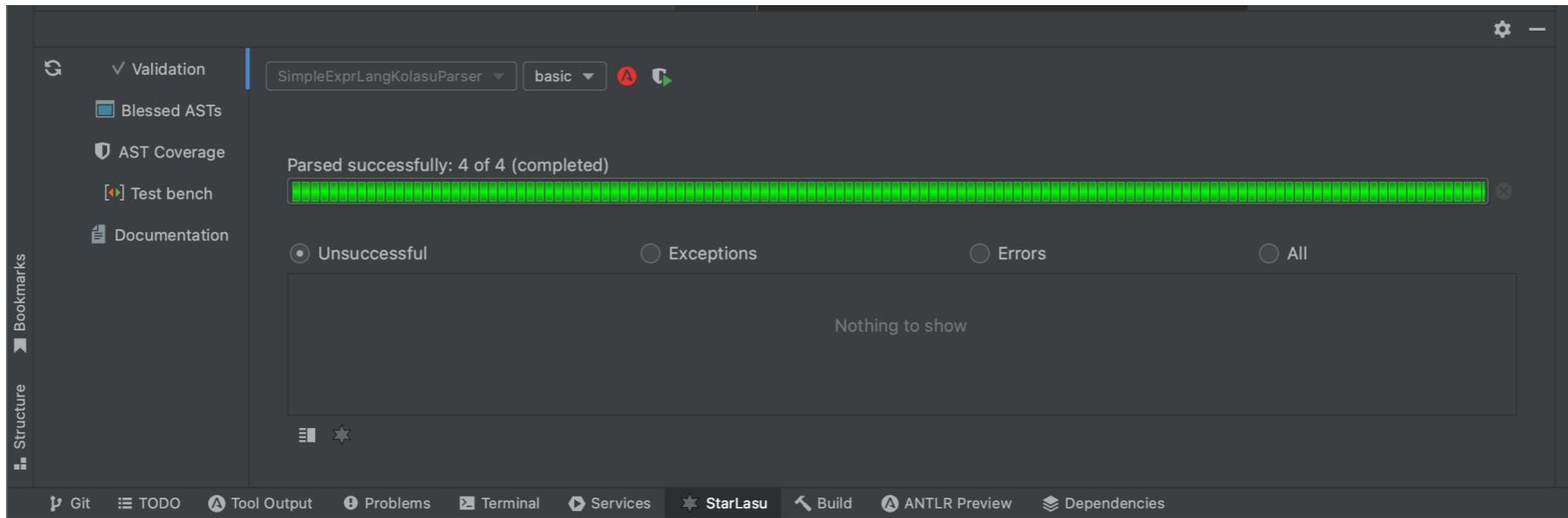
The Chisel Method - Workflow



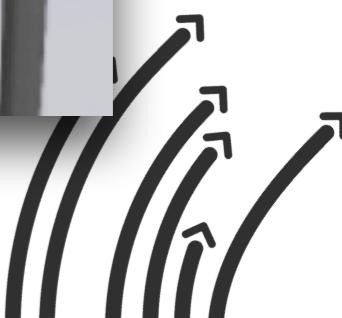
Workflow Demo



66 Show all checks are green



66 Where are we?



66 Where are we?

- Setup is now automated, so it does not take time
- Time is spent in the Parse Phase and the AST Refinement Phase
- We know by experience that the Parse phase takes 70% of the time
- We know that the progress is not linear, i.e., the last bits are the most difficult



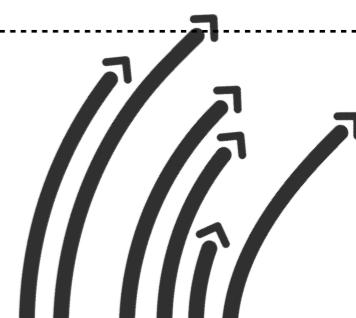
66 Where are we?

Parse Phase (0-70%) - 1,000 examples to cover

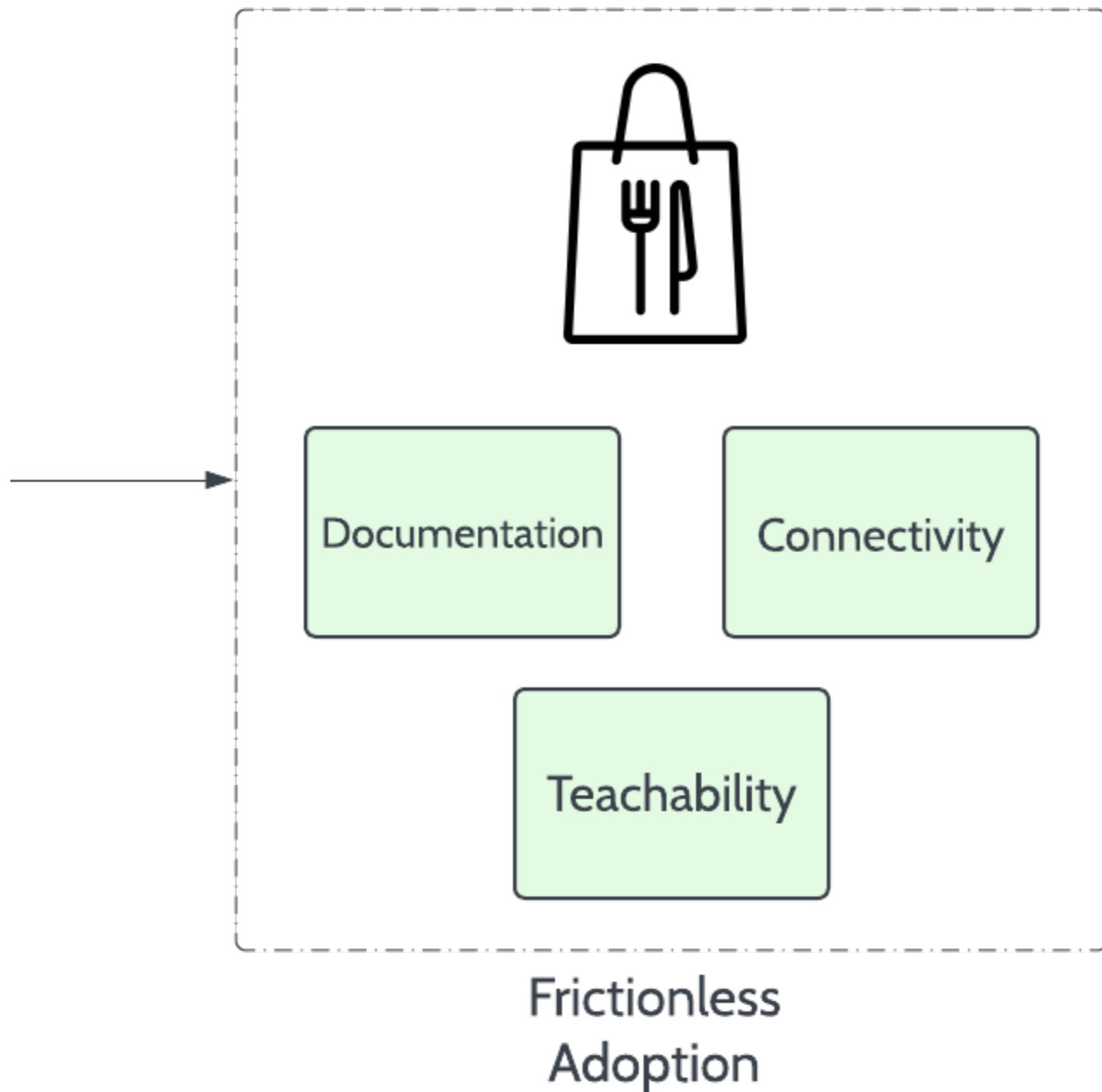
Examples covered	Progress
250	9 %
500	25 %
750	45 %
1,000	70 %

AST Refinement Phase (70-100%) - 100 constructs to cover

Constructs	Progress
25	74 %
50	81 %
75	89 %
100	100 %



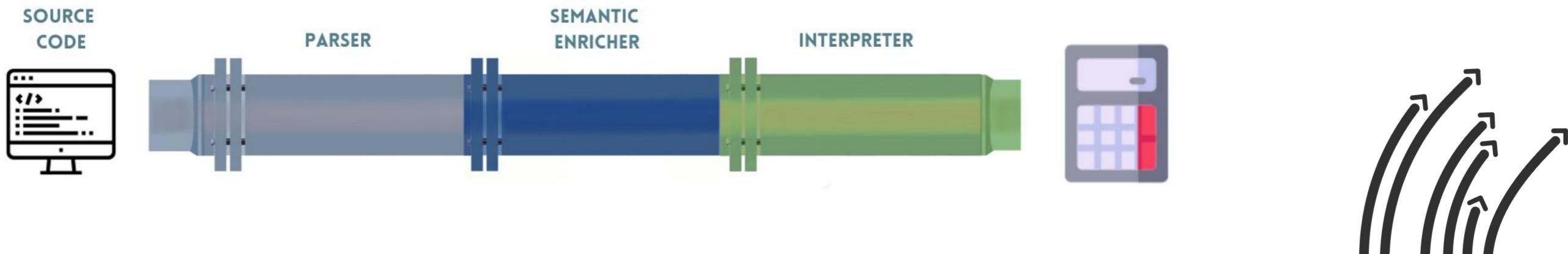
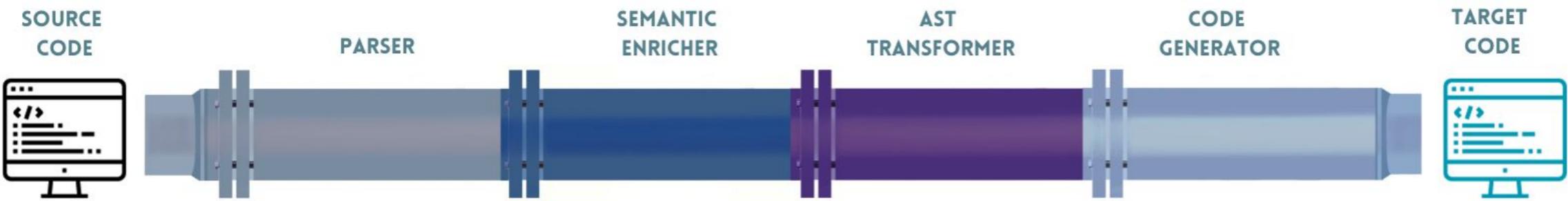
“



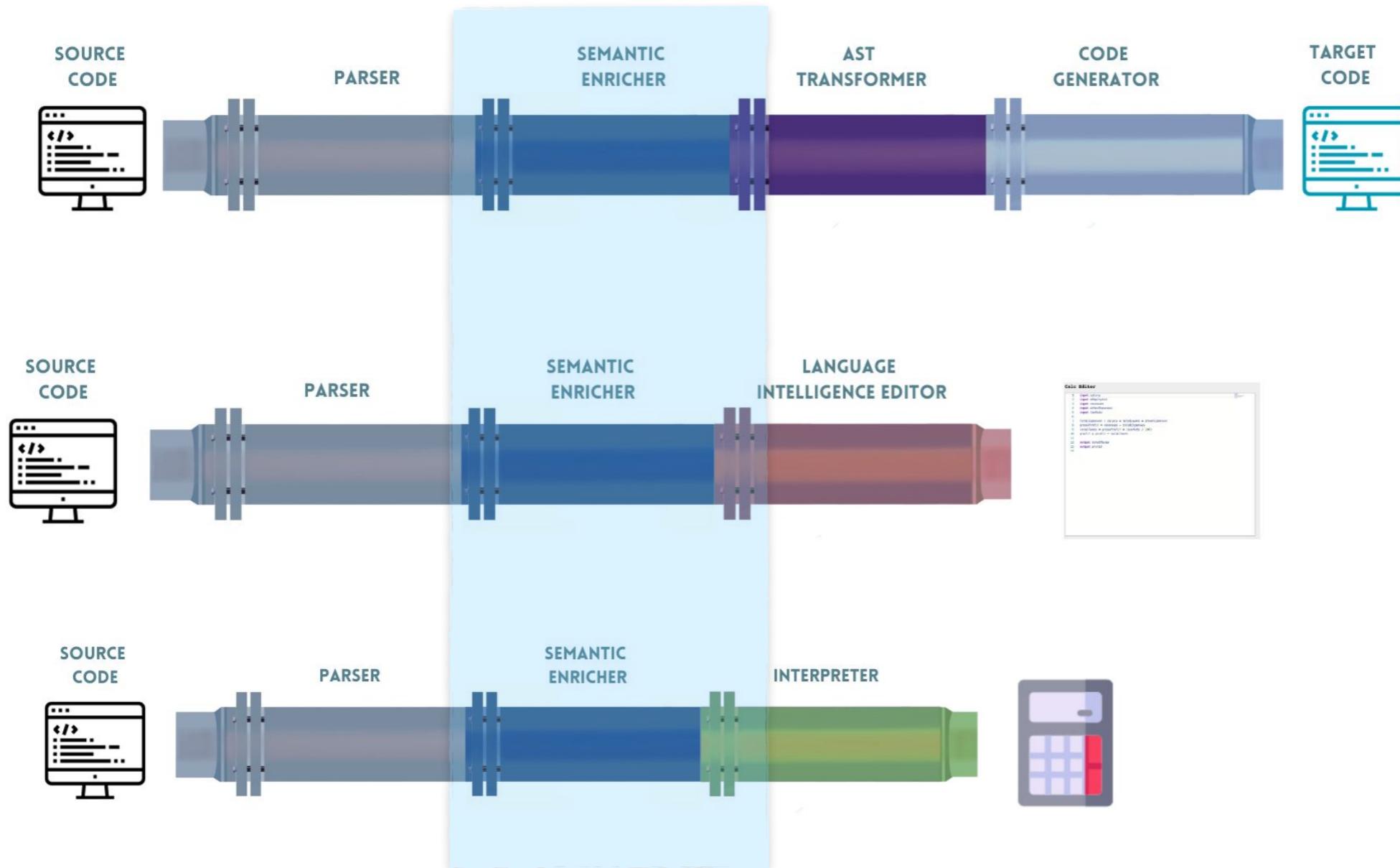
The Chisel Method - Adoption



66 Language Engineering Pipelines



66 Language Engineering Pipelines



66 Providing APIs

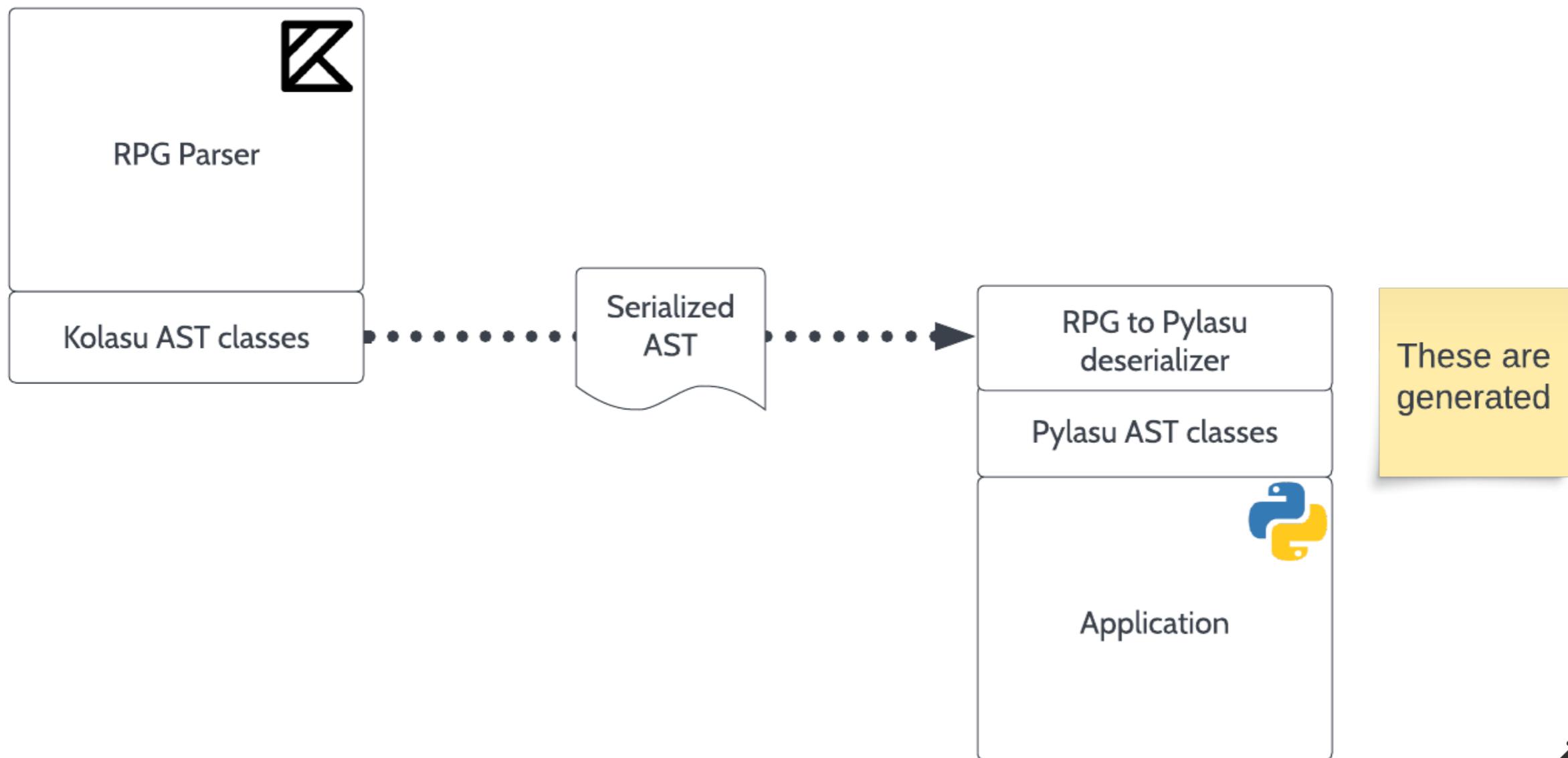
All of our parsers are based on the StarLasu libraries:

- Kolasu for Kotlin
- Pylasu for Python
- Tylasu for Typescript
- SharpLasu for C#

All those libraries provide APIs to navigate and process the AST



66 Generating adapters



66 Compatibility to the next level



11:00

LionWeb Initiative

Niko Stotz and Jos Warmer



66 Documentation

S SimpleExprLangDocs

Search the docs...  

Documentation

Introduction

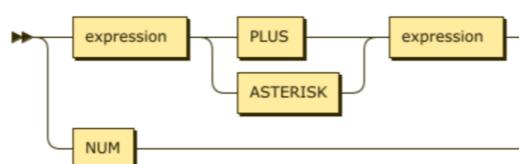
com.stumenta.SimpleExprLang.ast

Abstract Classes and Interfaces

Expressions

Classes

Expression Class (Abstract) A



An Expression in SimpleExprLang is a very core element. I wish I had something smarter to write here.

Inherited Features

Name	Type
position	Position?
origin	Origin?
destination	Destination?

Subclasses

[Multiplication](#) [NumLiteral](#) [Sum](#)

Used In

[Sum](#) [Multiplication](#) [Definition](#)



66 Teachability

Teachability reduce risks:

- the vendor providing the parser will be able to train new maintainers, if needed
- the client can take over maintenance of the parser, if needed
- complementary parsers, based on the same structure, can be developed (think of an HTML, JS, and CSS parsers that are compatible)



66 What are the deliverables

- A written method: currently described into 20 pages, later on it will become a video-course
- Supporting libraries: the ones necessarily at runtime are the StarLasu libraries, and they are open-source
- Tools: a gradle plugin and an IDEA plugin, that at the moment we use internally and we are refining. They just accelerate development



66 Why calling it the Chisel method?



Because it is about getting the information out of the code, as you use a chisel to take the statue out of the marble.

Also, Strumenta means tools in Latin, and Chisel is one tool.



66 What next?

- Refine this for the next 10 years
- ...while working on the method for building Transpilers
(hopefully in time for next year LangDev!)



66 Thank you!

Feedback at:

federico@strumenta.com

