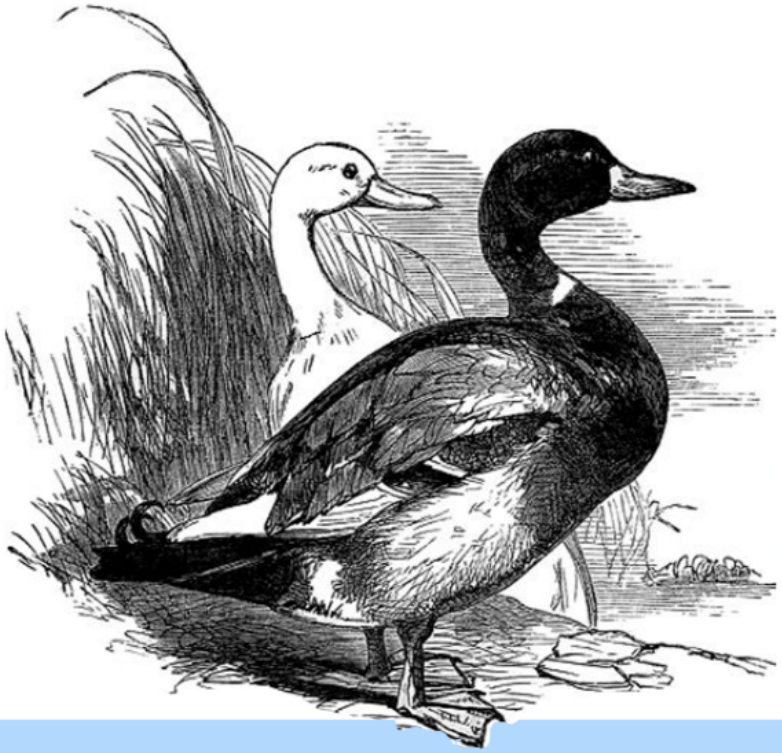
Conclusion

Writing Real Time Web Applications Without The Hassle



Programming in Pass

Rafael Castellanos, Project Manager
Peter Sugihara, Language Guru
Nicolo Pizzoferrato, System Integrator
Andrew Lamping, System Test
Cody De La Vara, System Architect

Group 17

Evolution of the Internet

Primitive

- simple static file server
- HTTP 1.0
  - non-persistent connection
  - only supported simple requests such as GET, POST, etc

Getting better...

- Client-side scripting: JavaScript!
- Persistent HTTP 1.1
- AJAX, polling, etc.

Current technologies

NodeJS

- capability to build a server entirely in JavaScript
- event-driven
- asynchronous

WebSocket

- bidirectional communication protocol
- designed for real-time communication

So...what's wrong with the way things are now?

- learning API for servers/sockets is time consuming
- even experienced programmers must deal with boilerplate configurations before getting to the core of the application
- HTTP's stretched metaphor

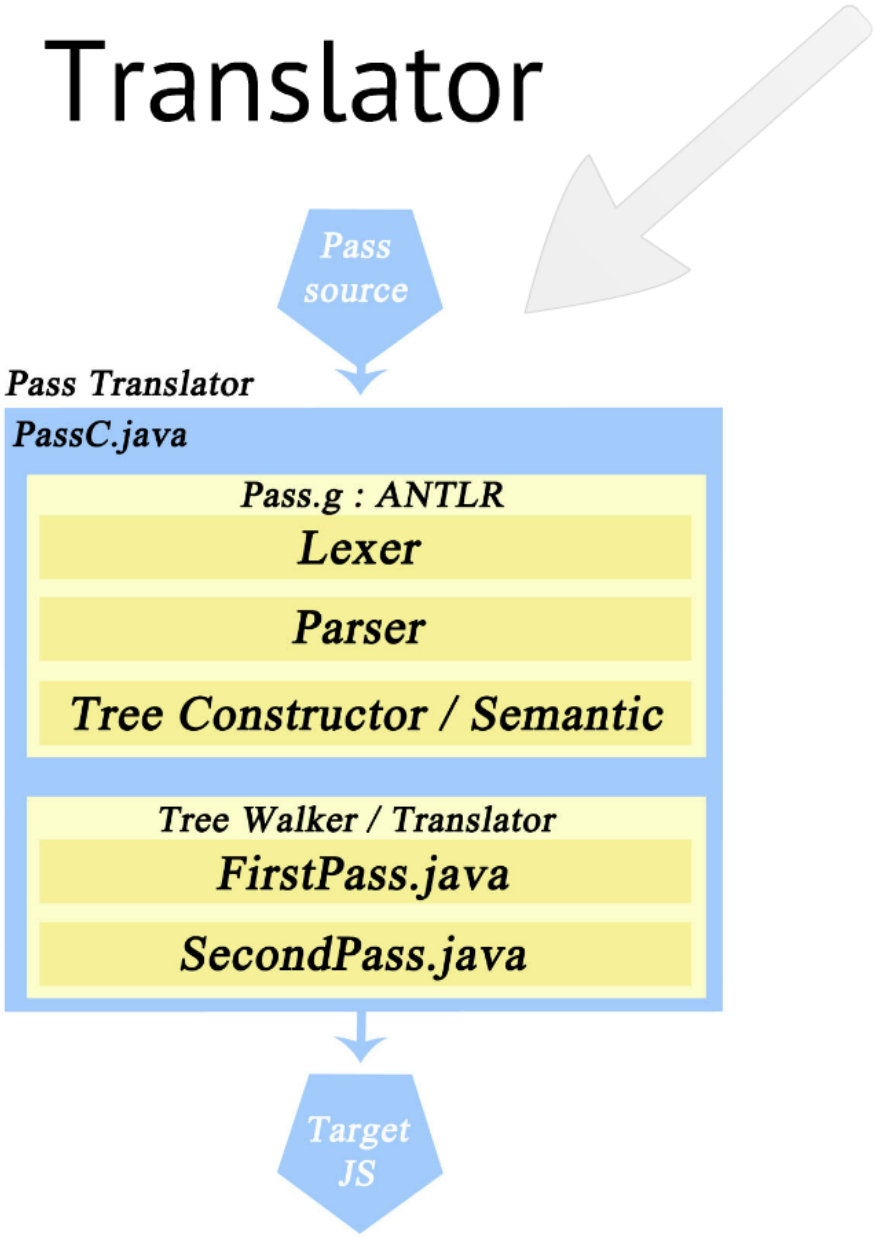
Introducing...

PASS

- DSL for writing real-time web applications
- abstracts away the details of setting up a server
- no dealing with socket connections
- immediately work on the application logic
- simple mechanism for function exposure and server-client communication
- a sleek, compact syntax without sacrificing functionality
- easy to learn for anyone with JavaScript (or other C-family language) background

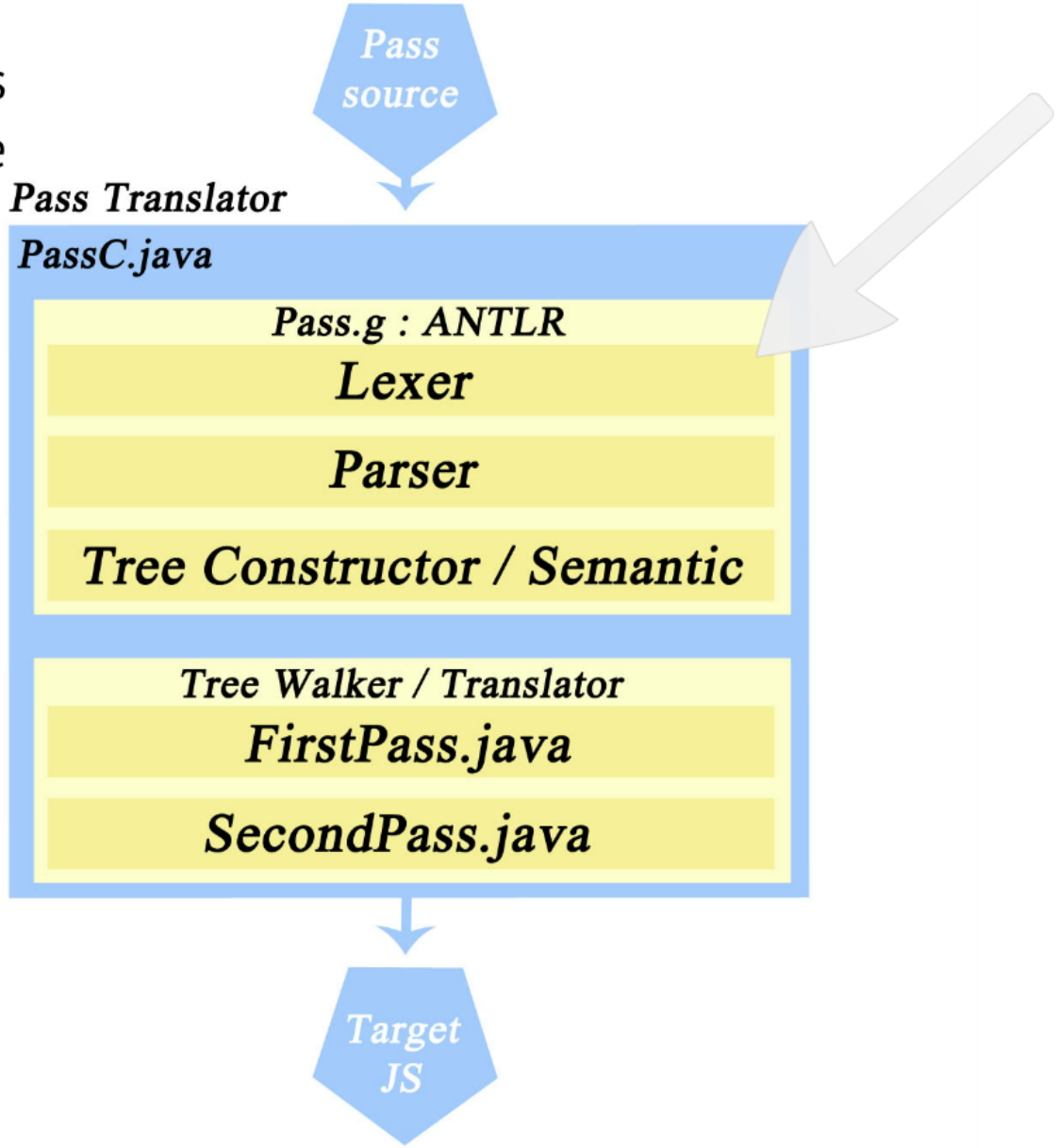
Translator

PassC passes the source into an ANTLRFileSteam, which is given to an ANTLR Lexer to break the source program into a Token stream.



Translator

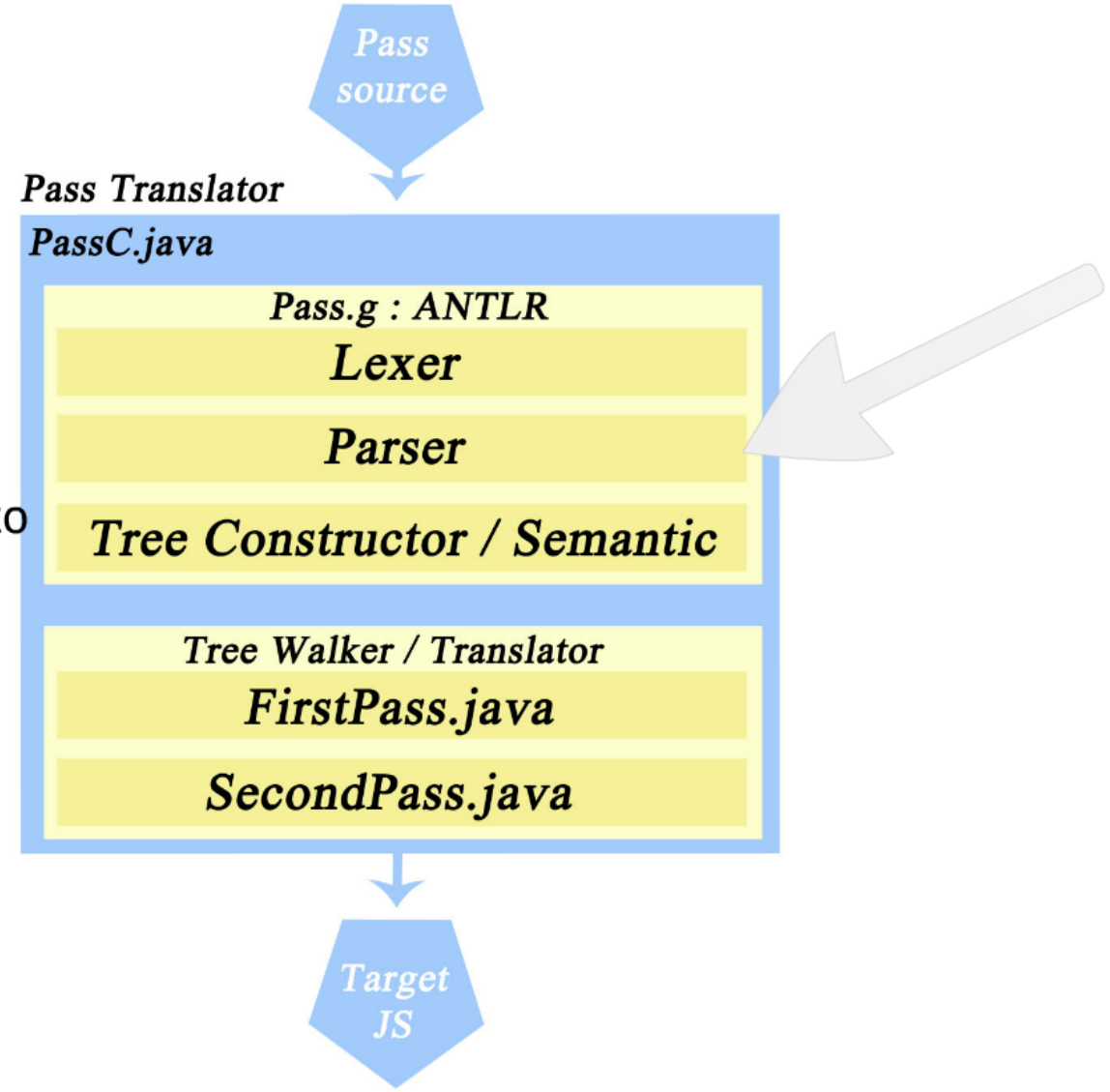
Leading whitespace is used to determine the grouping of statements in Pass. The lexer generates INDENT and DEDENT tokens



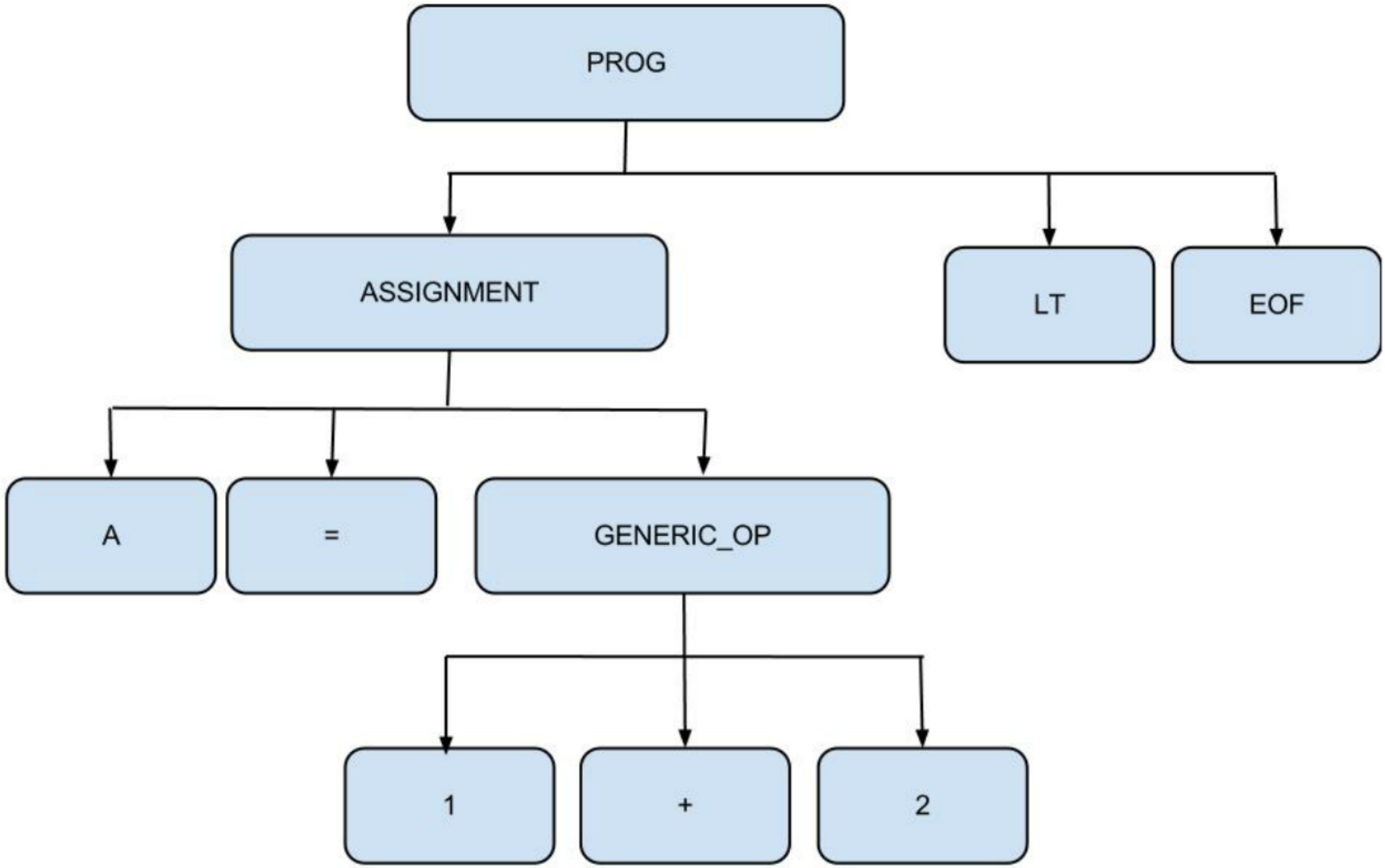
Translator

Parser ensures tokens conform
to our grammar.

ANTLR allows us to insert code in-between rule productions used to return semantic errors earlier on.

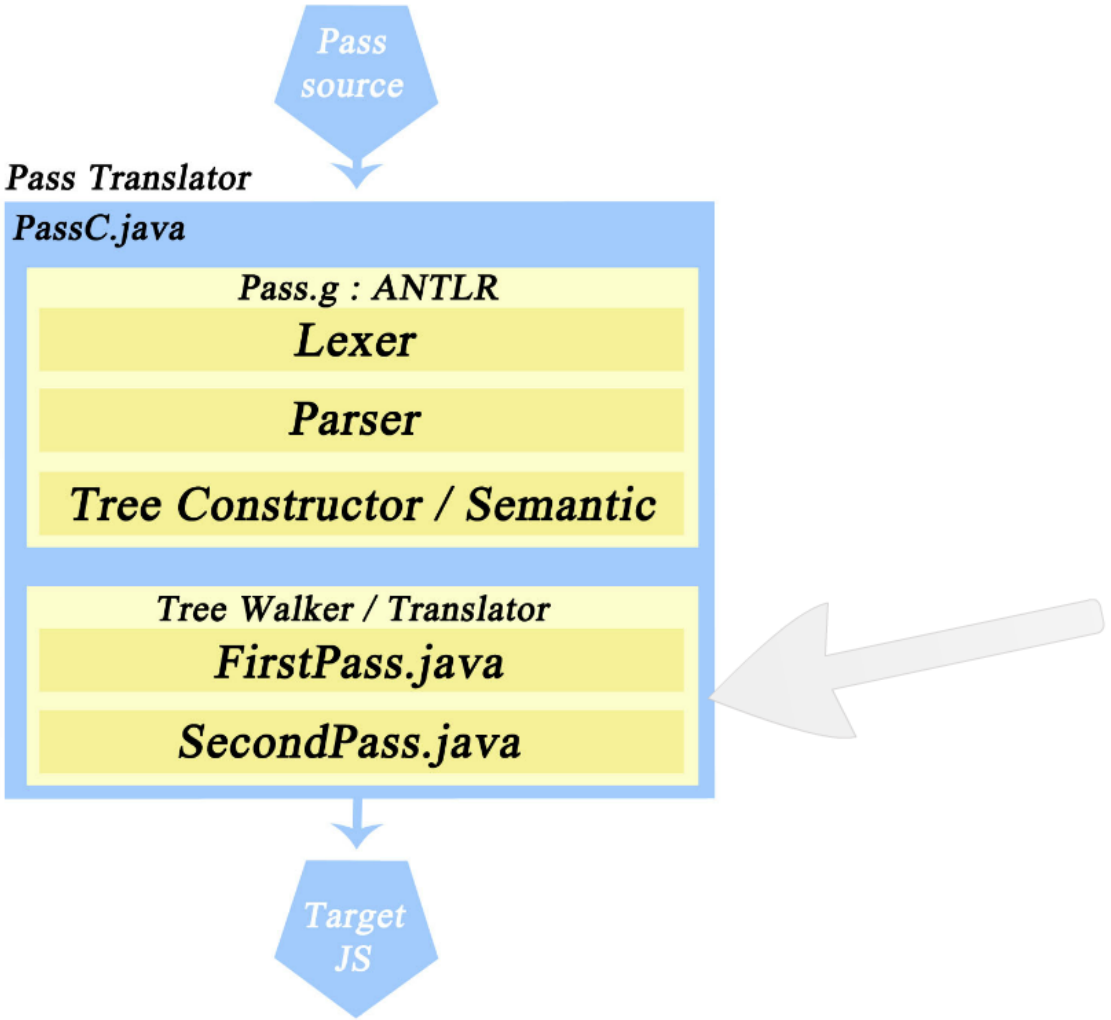


Example of AST for A = 1 + 2

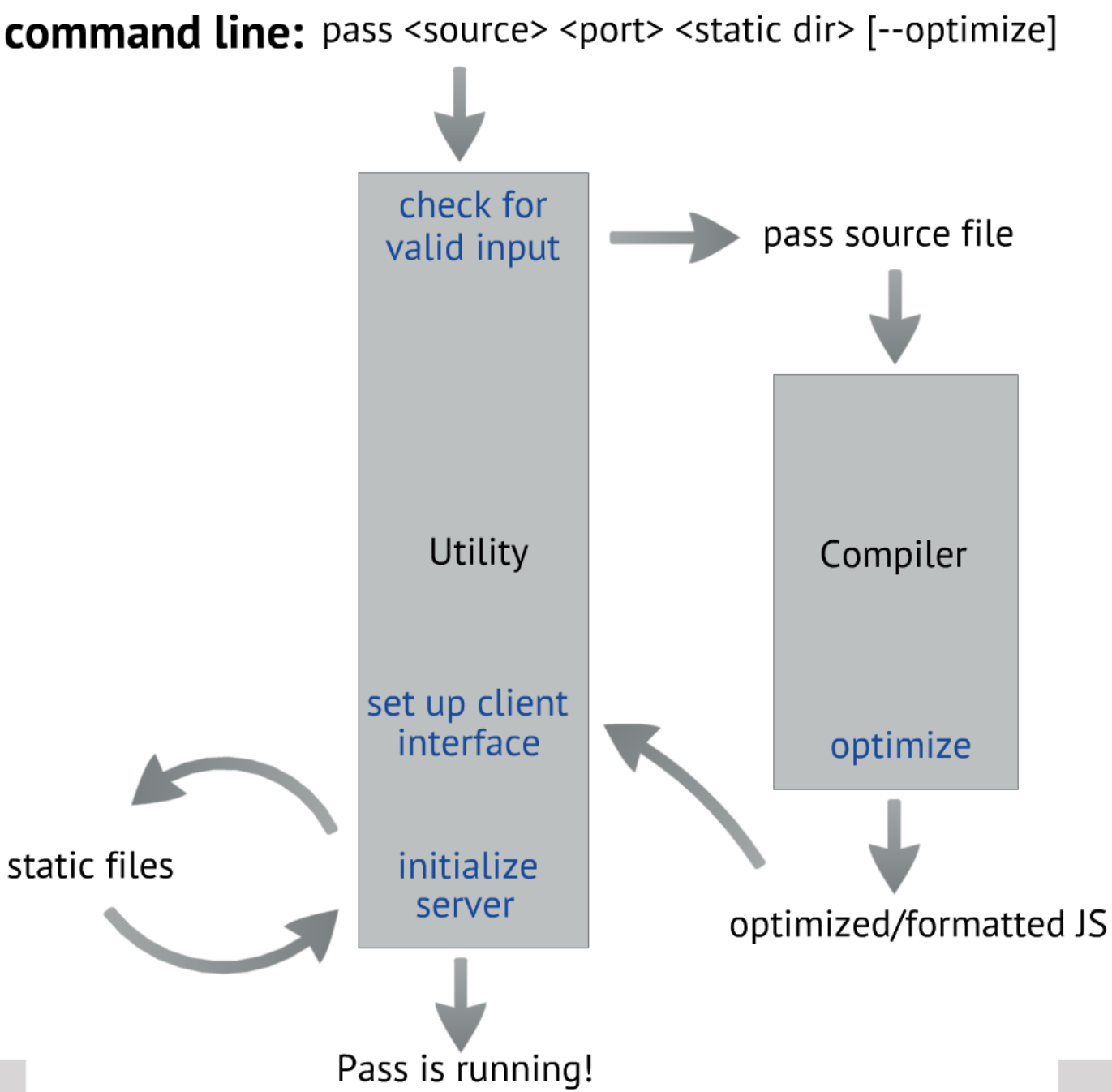


Translator

With the AST, we simply walk the tree and translate!



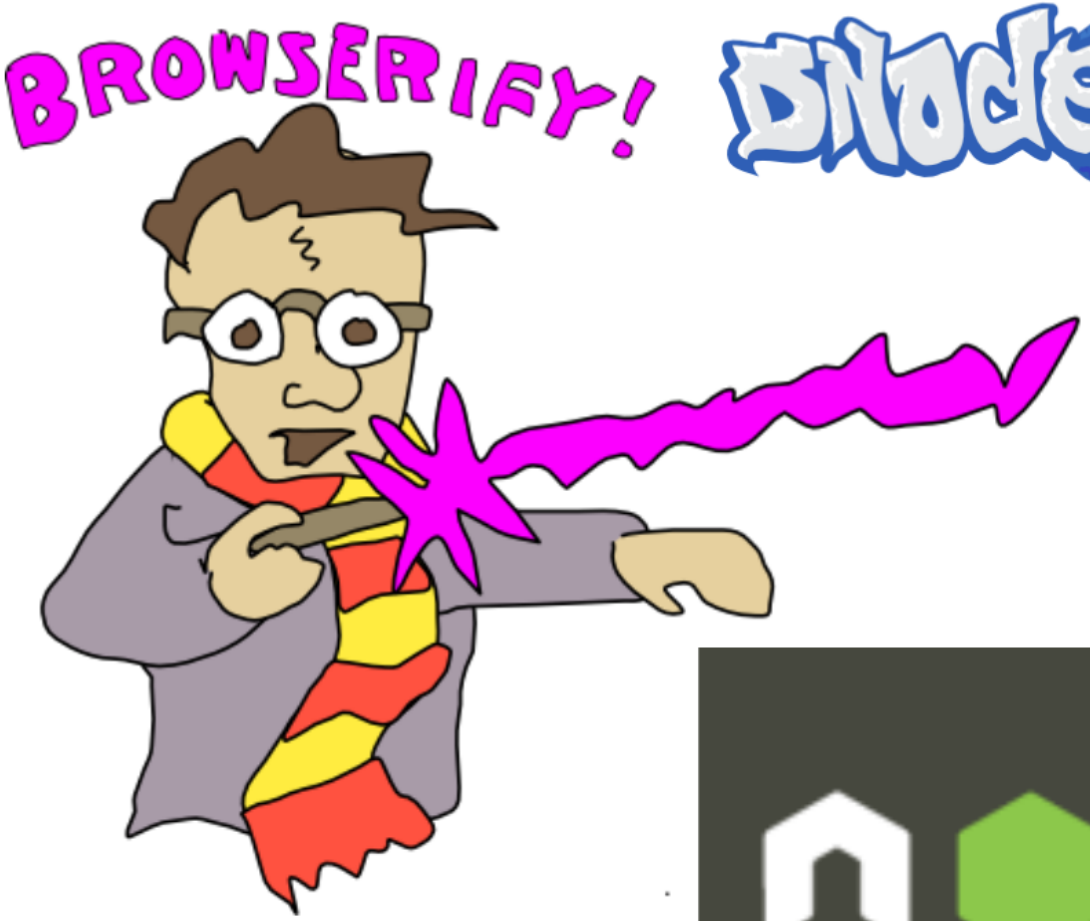
How it all fits together



Installation

npm install pass

depends on Node.js and Java


BROWSERIFY!


DNode
FREESTYLE
r.p.c.


node
JS
TM

Native functions

(client grouping support)

MULTIPLE TAGS:

- pushTag(connection, tag)
- popTag(connection, tag)
- getTags(connection)

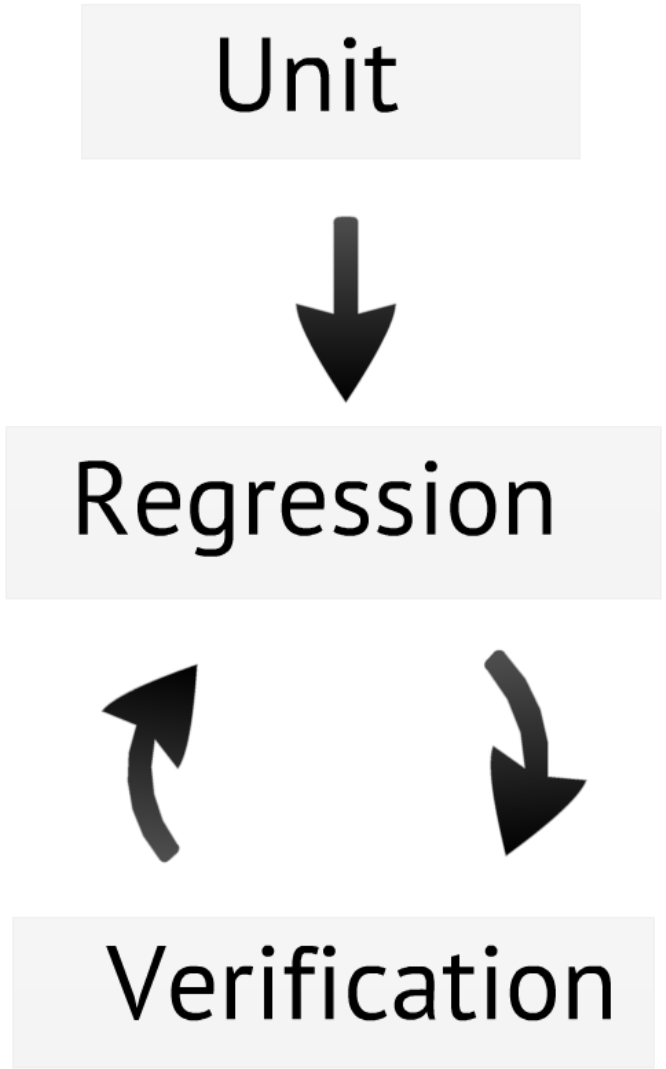
SINGLE TAG:

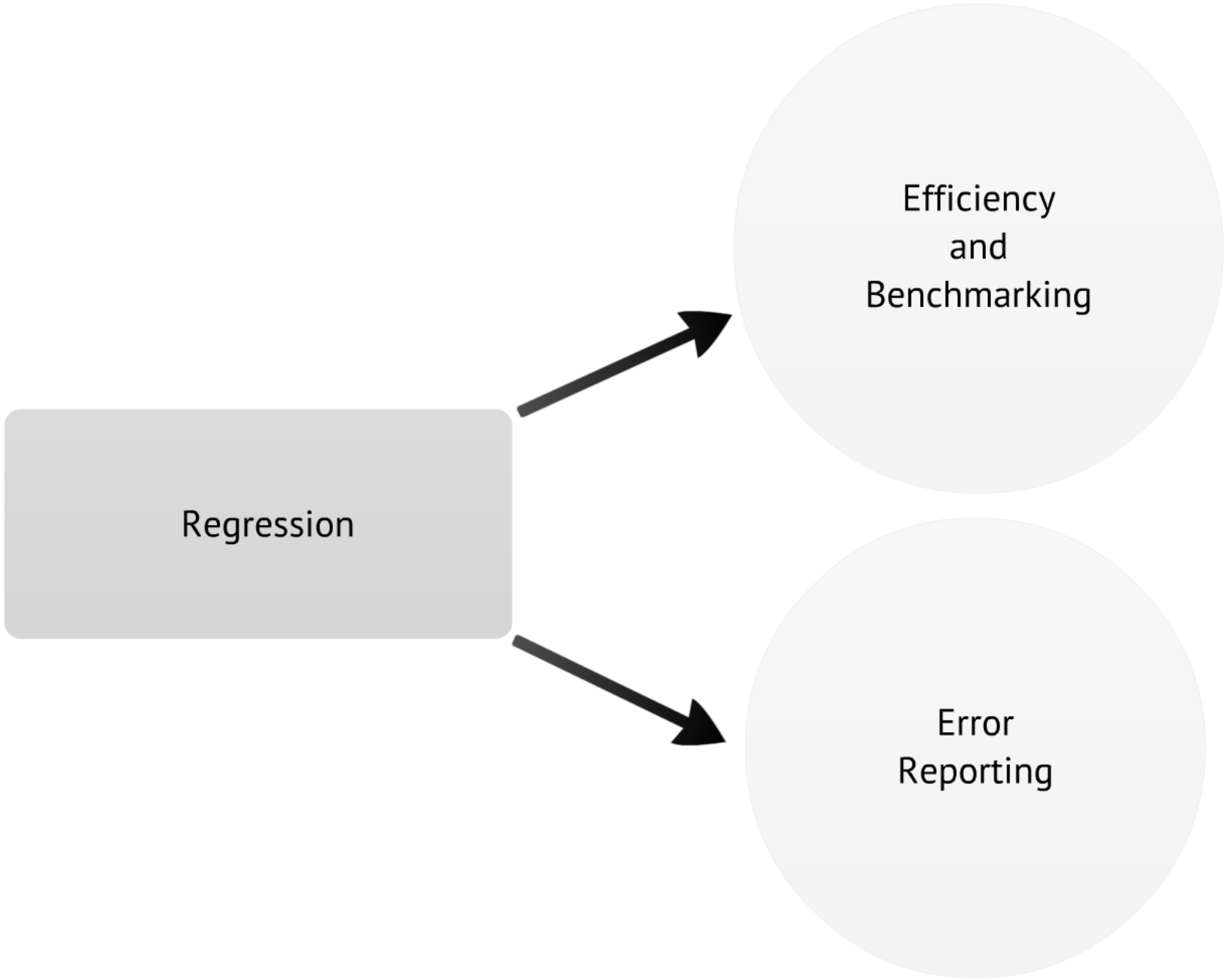
- setTag(connection, tag)
- getTag(connecton)

UNIVERSAL:

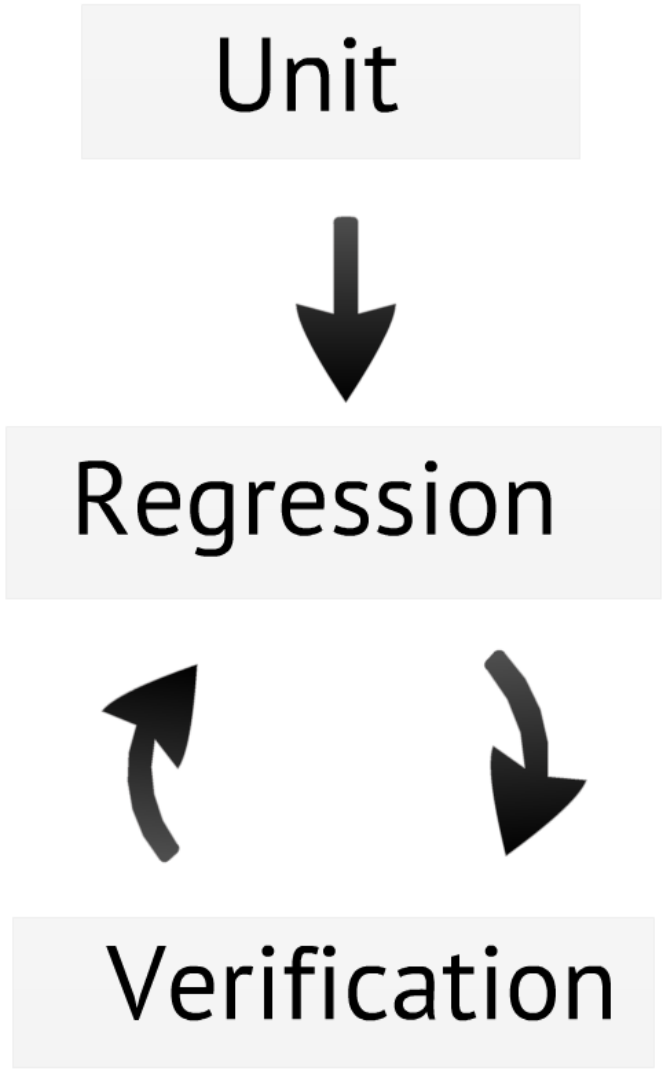
- conns(tag)
- clearTags(connection)
- tagIsLive(tag)
- hasTag(connection, tag)

Testing



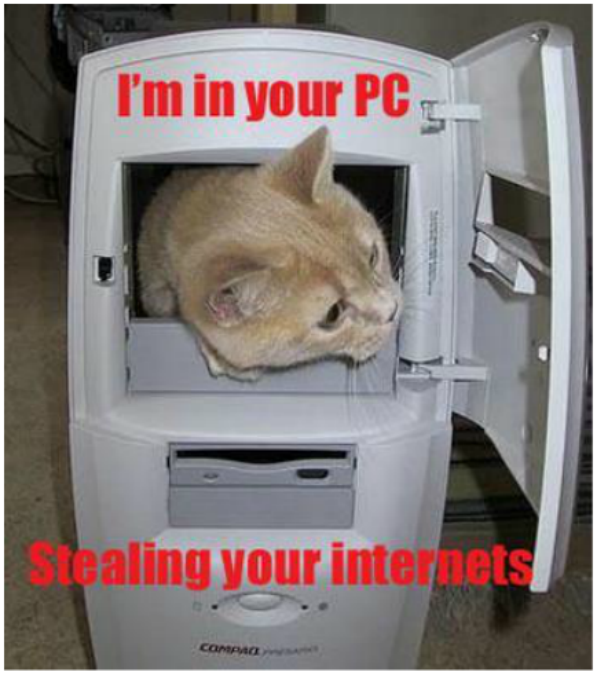

Regression
Efficiency
and
Benchmarking
Error
Reporting

Testing



Project Management

Security



Don't be too serious.

Conclusion