# Source-Level Dataflow-Based Fixes

## Experiences from using IntraJ and MagpieBridge

Idriss Riouak – Lund University

# EXAMPLE DATAFLOW-BASED ANALYSES

```
1  void foo(boolean b){
2    String x = null;
3    if(b) x = "Hello World";
4    x.toString();
5  }
```

## Null pointer Analysis (NPA)

⚠ Possible `NullPointerException` at line 4

```
1  private int hash = 0;
2  int hashFunc(){
3    if(hash==0){
4      int hash = 10;
5      //Complex operations on  hash
6      hash += 10;
7    }
8    return hash;
9  }
```

## Dead Assignment Analysis (DAA)

SIMPLIFIED EXAMPLE FROM APACHE FOP (90 KLOC)

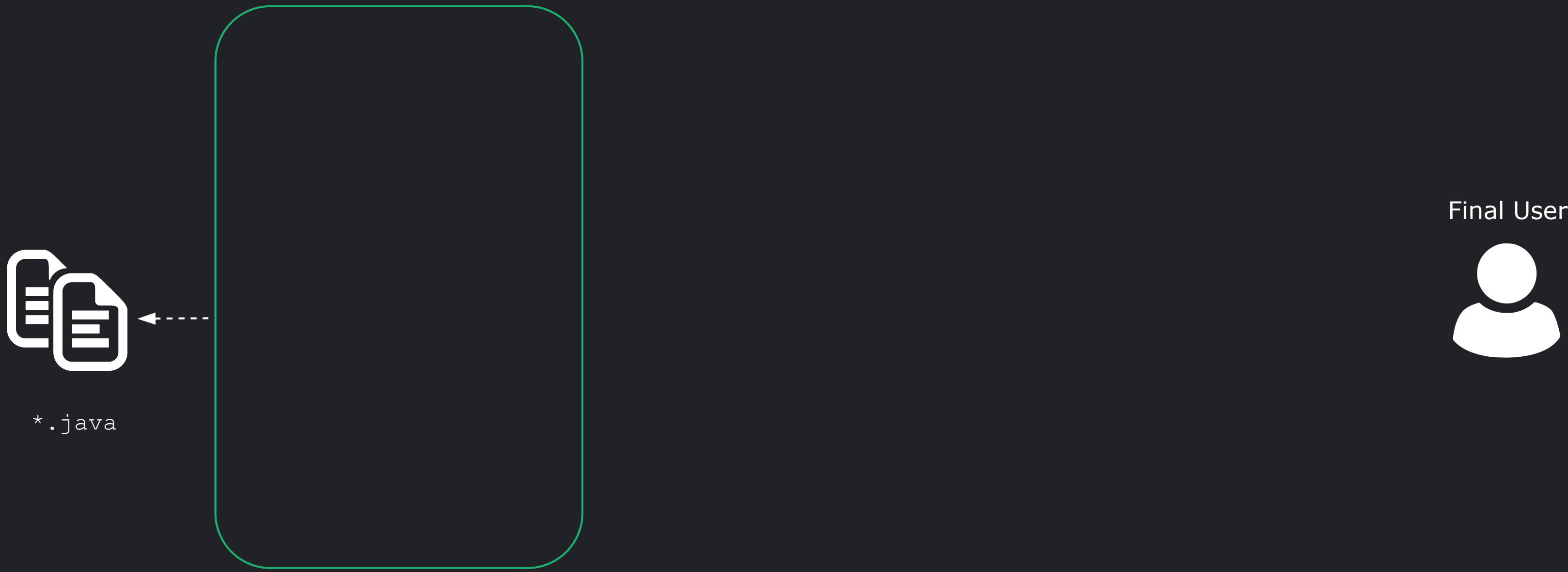
⚠ `Dead Assignment` at line 6. The value of `hash` is never read.

## THE BIG PICTURE

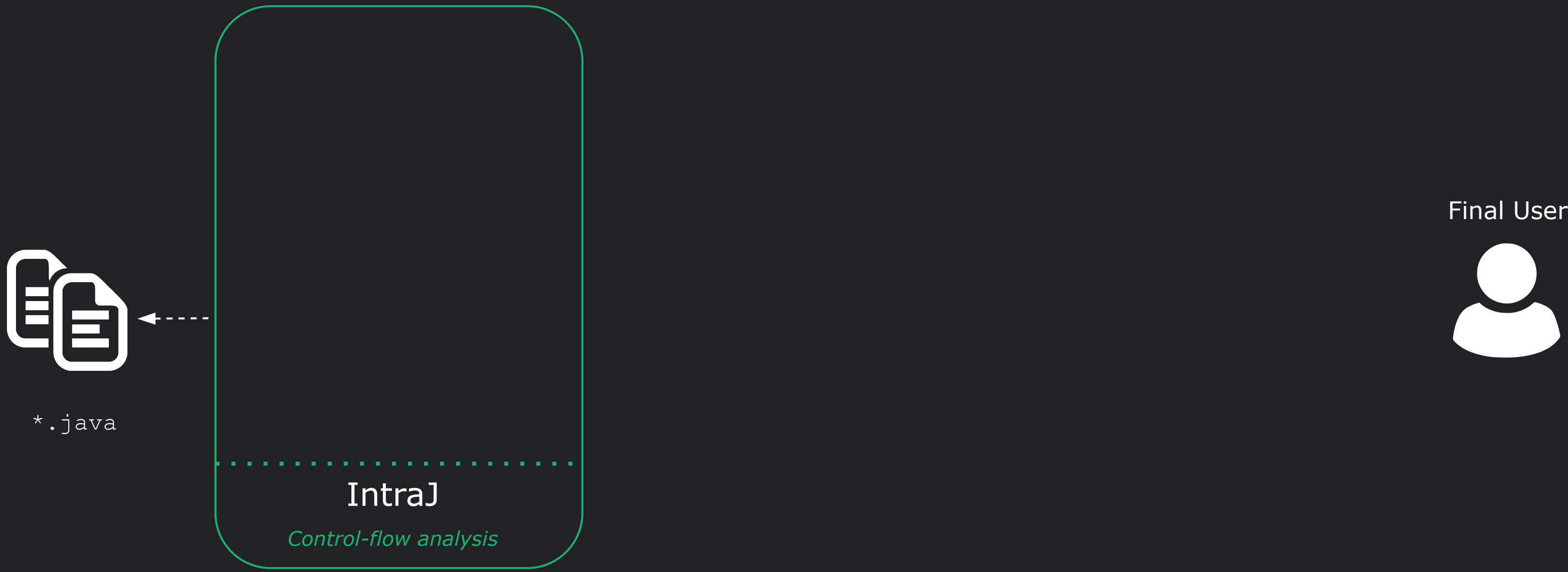
Final User

`*.java`

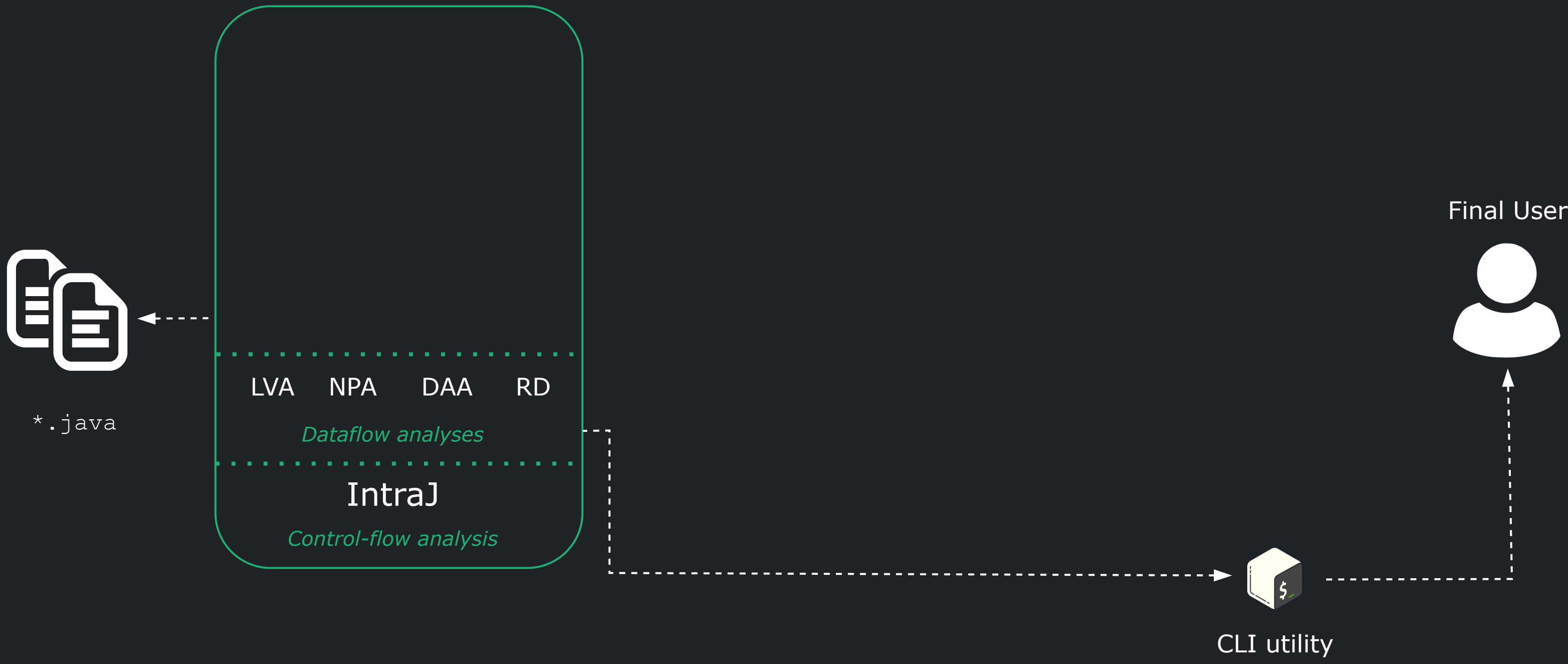
## THE BIG PICTURE



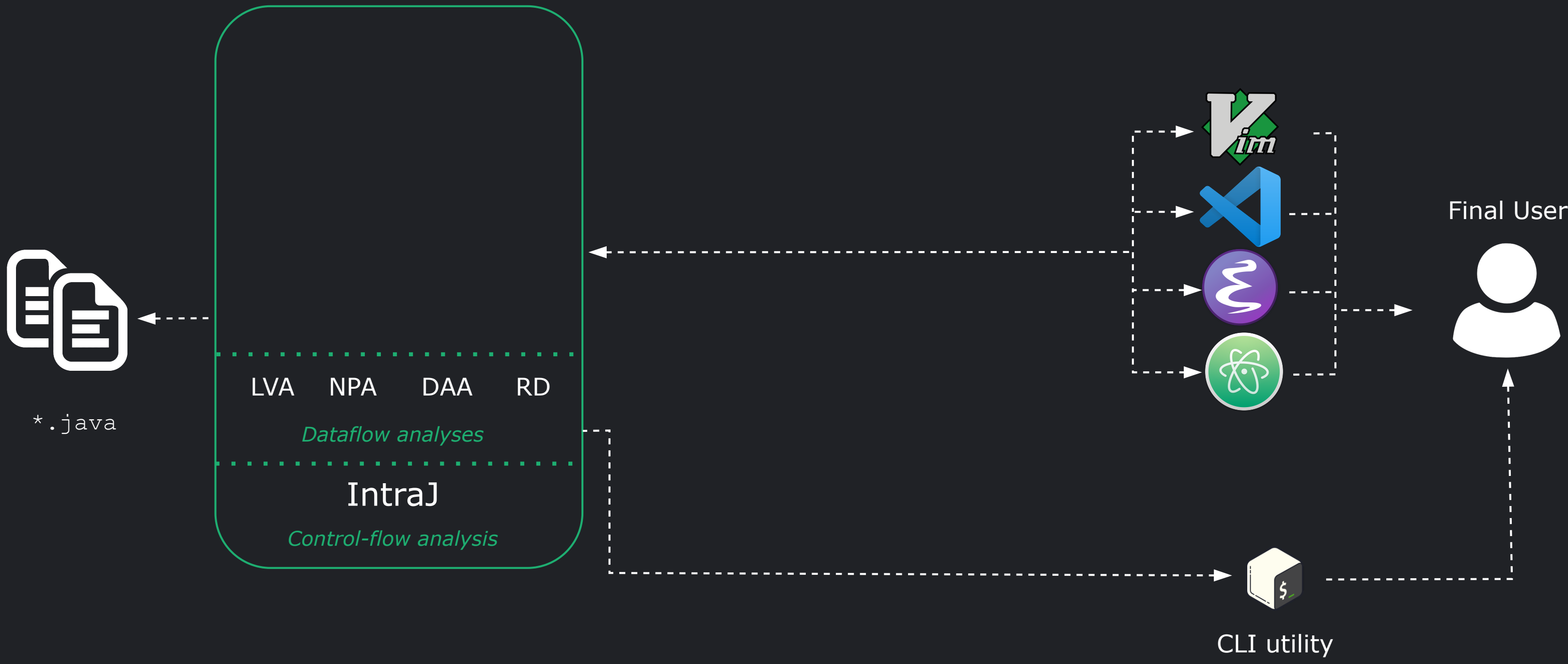
## THE BIG PICTURE



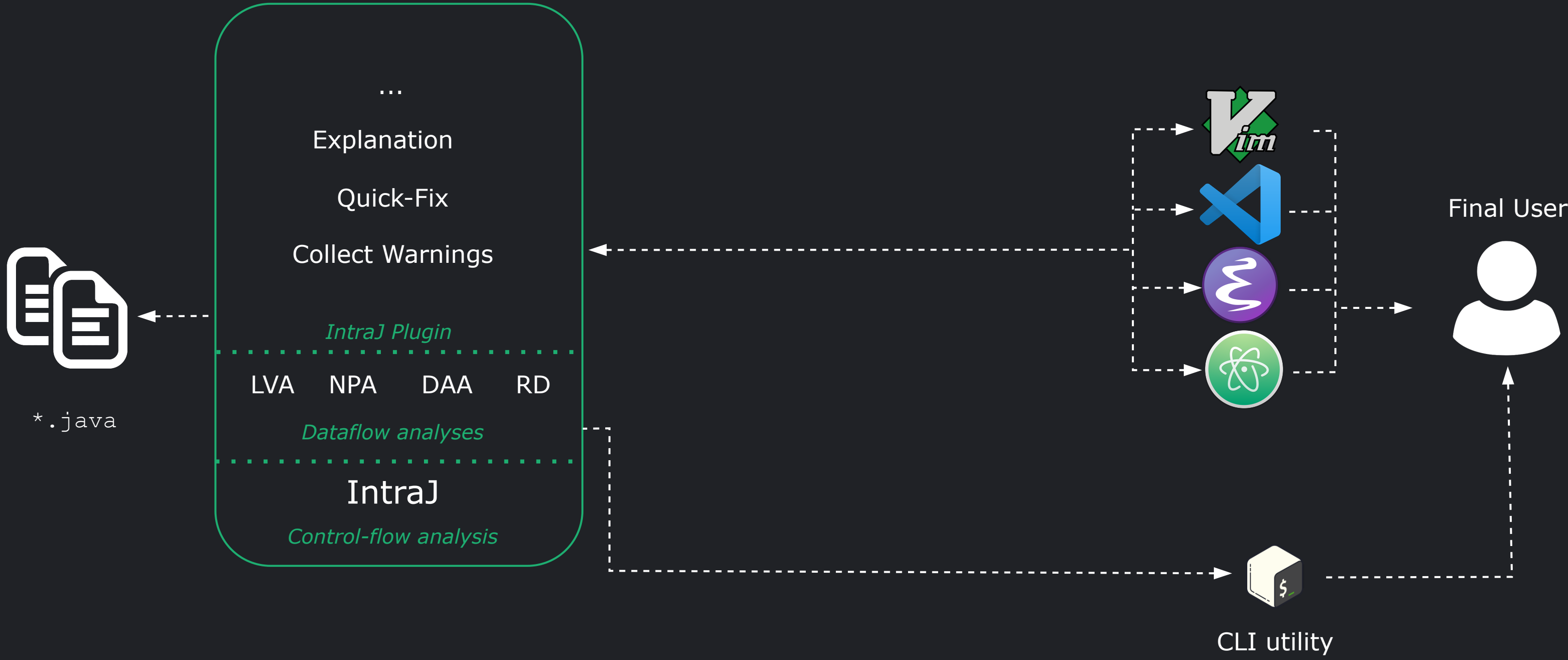
## THE BIG PICTURE



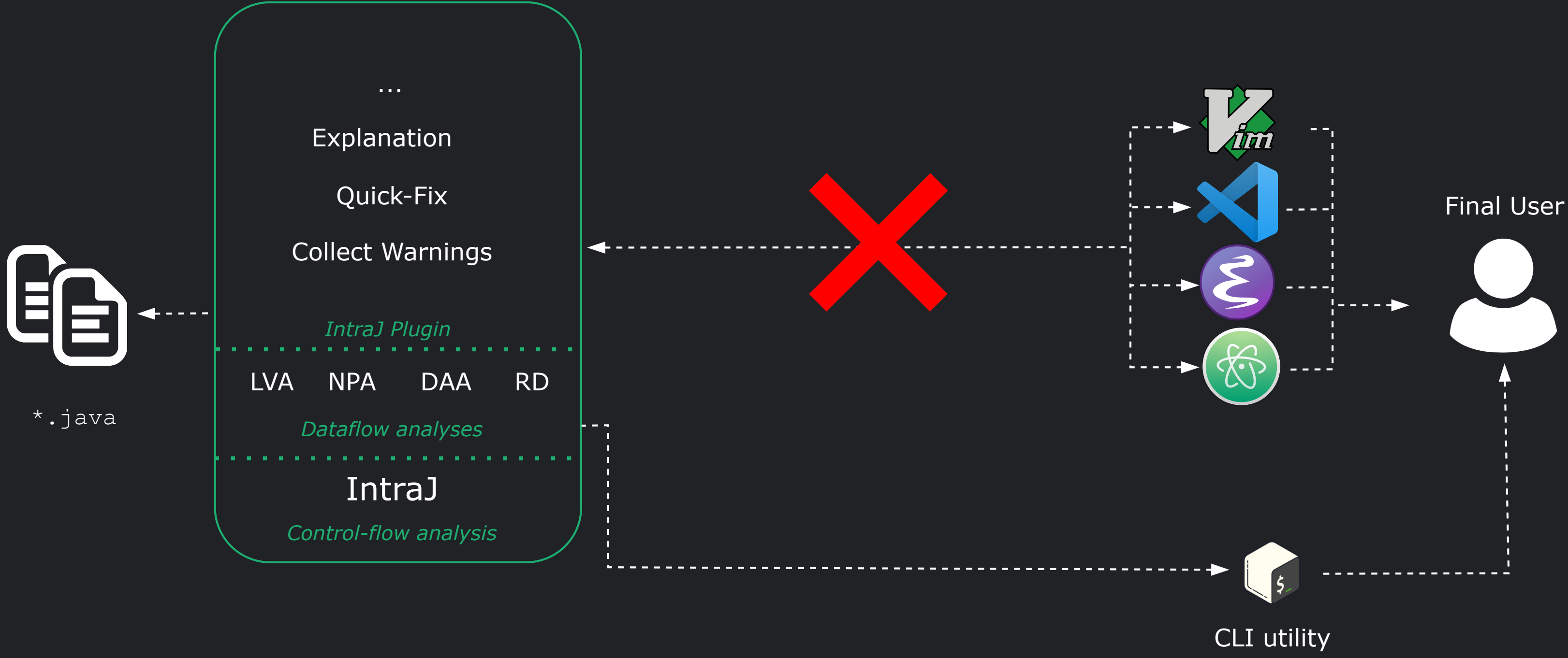
# THE BIG PICTURE



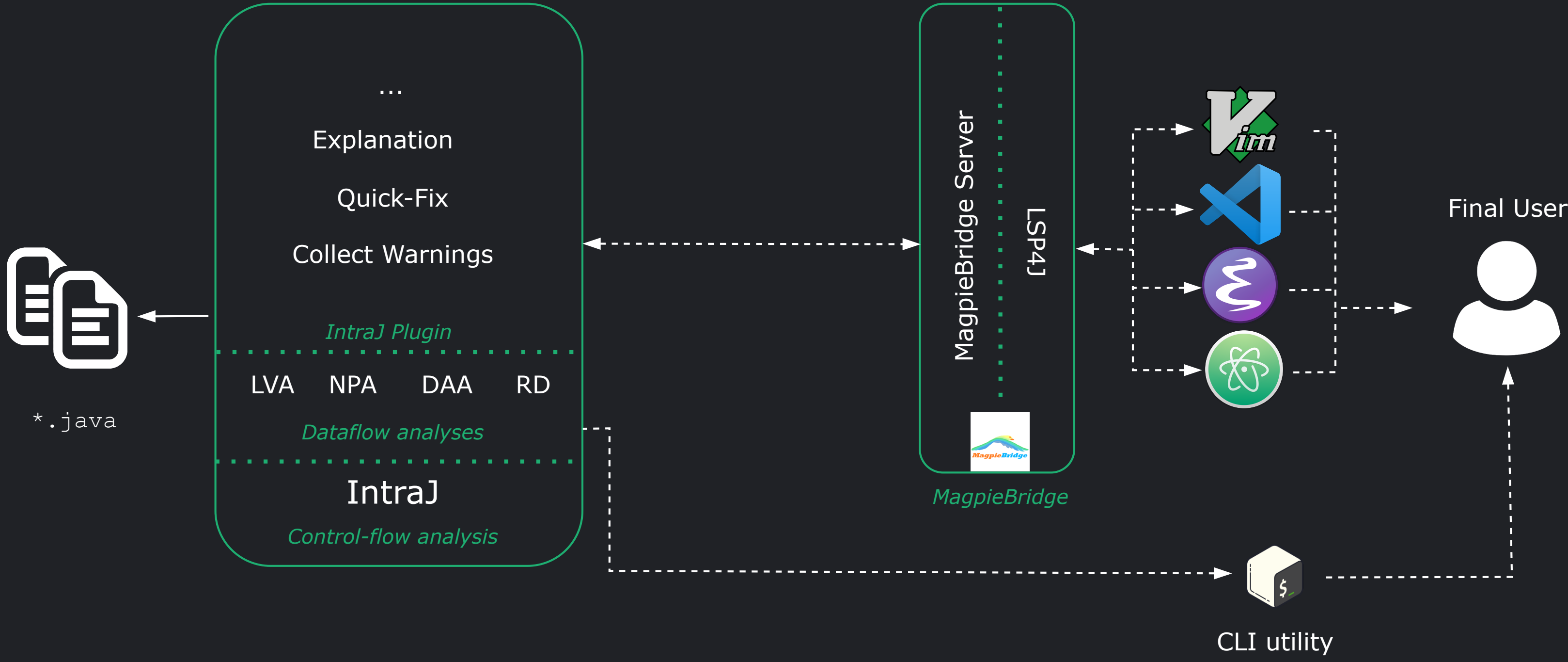
# THE BIG PICTURE



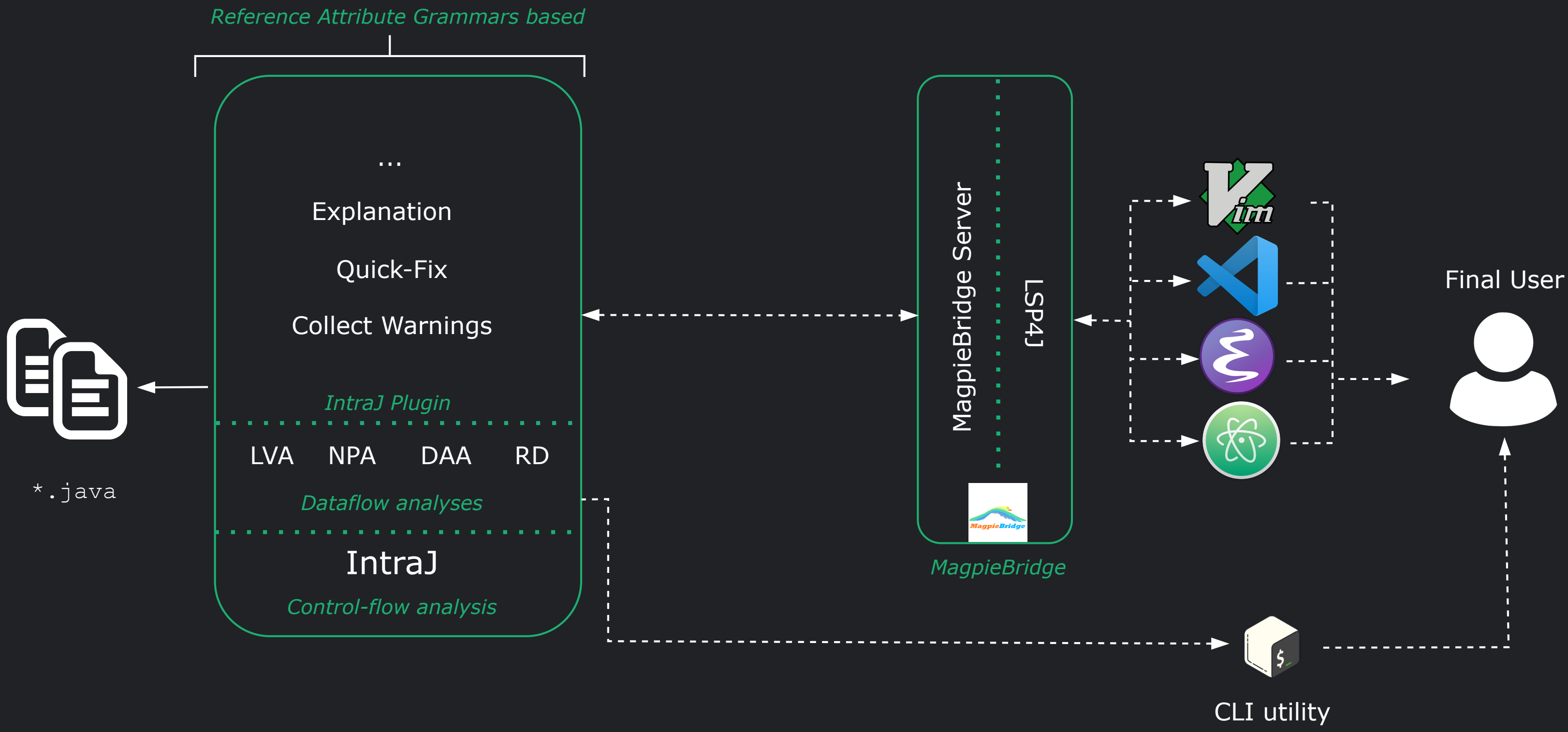
# THE BIG PICTURE



# THE BIG PICTURE

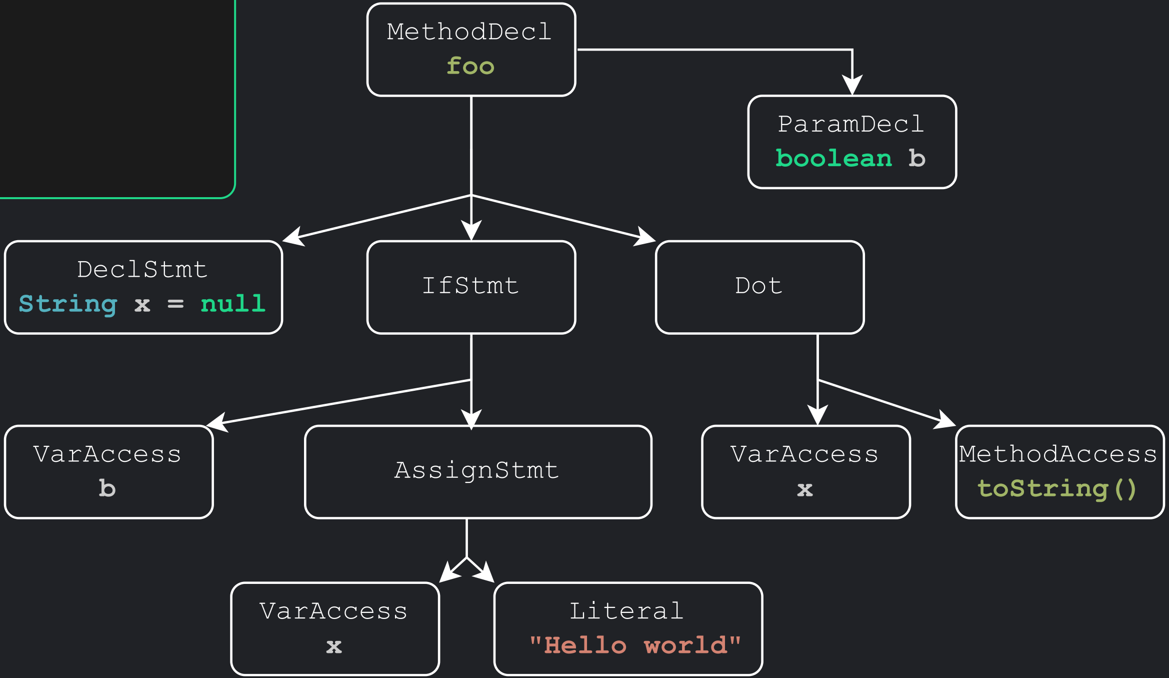


# THE BIG PICTURE



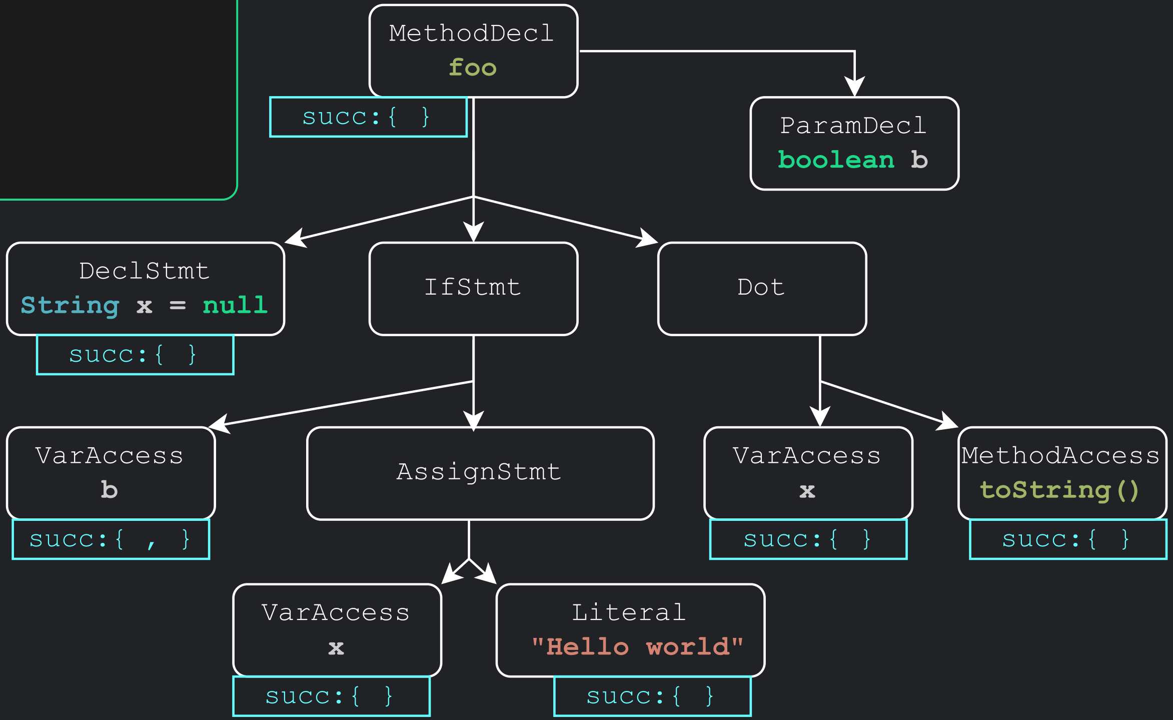
# REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



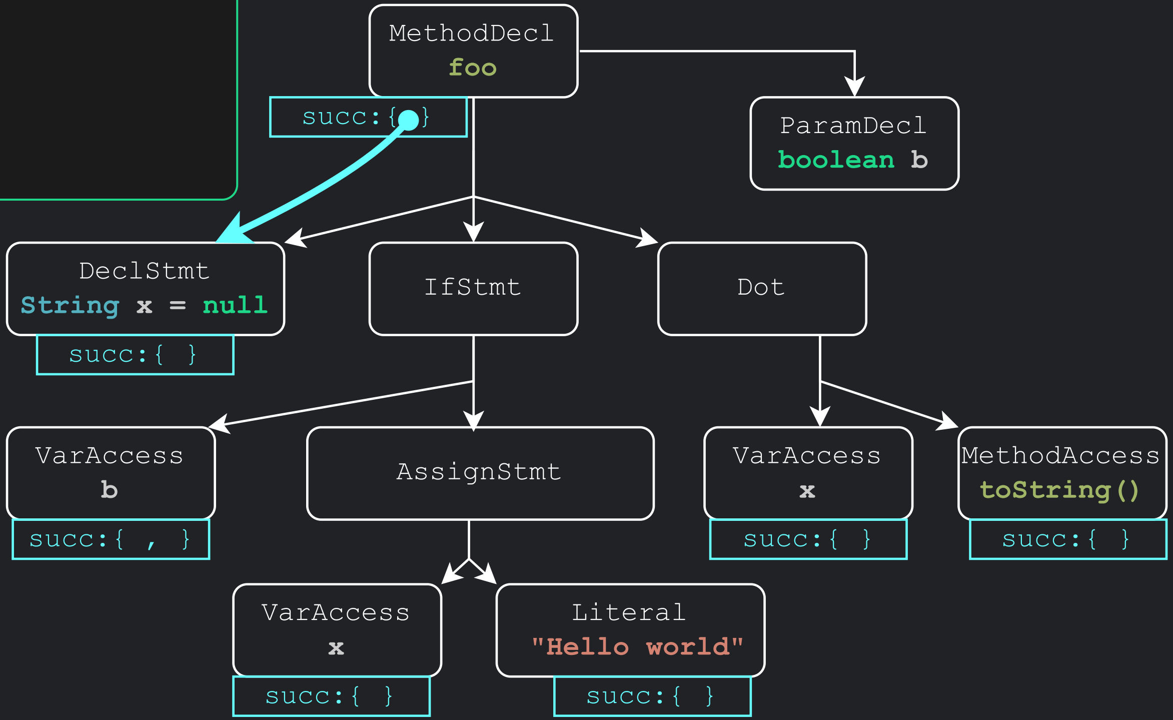
# REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



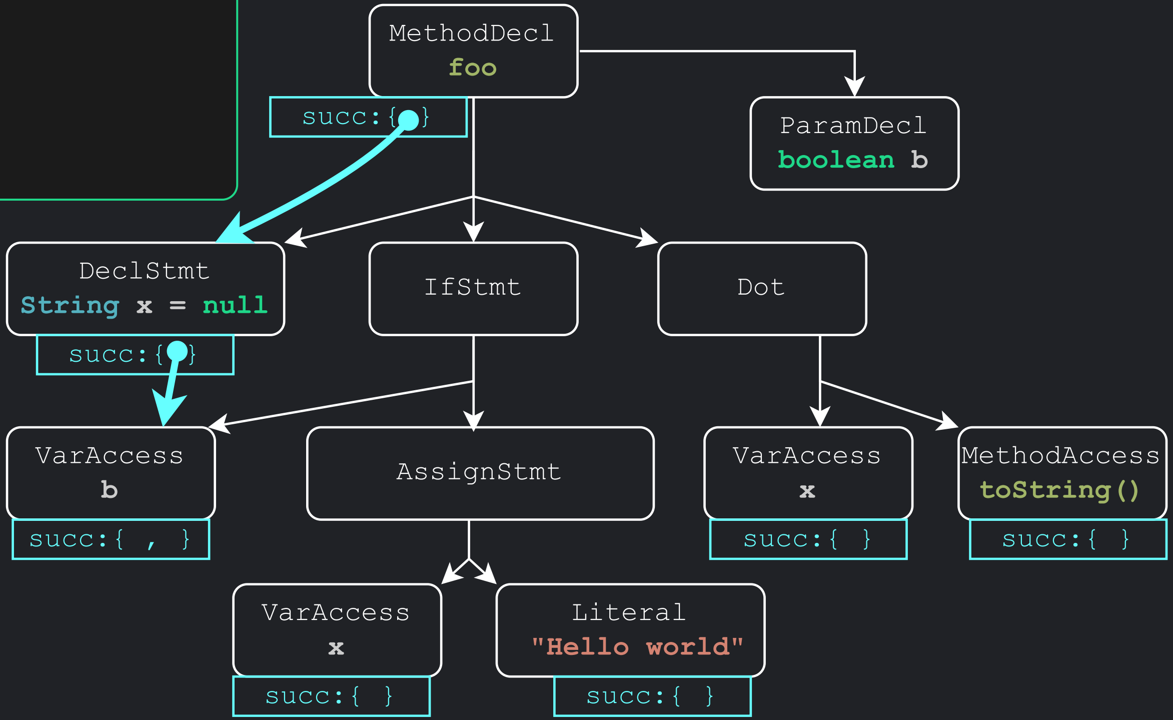
## REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



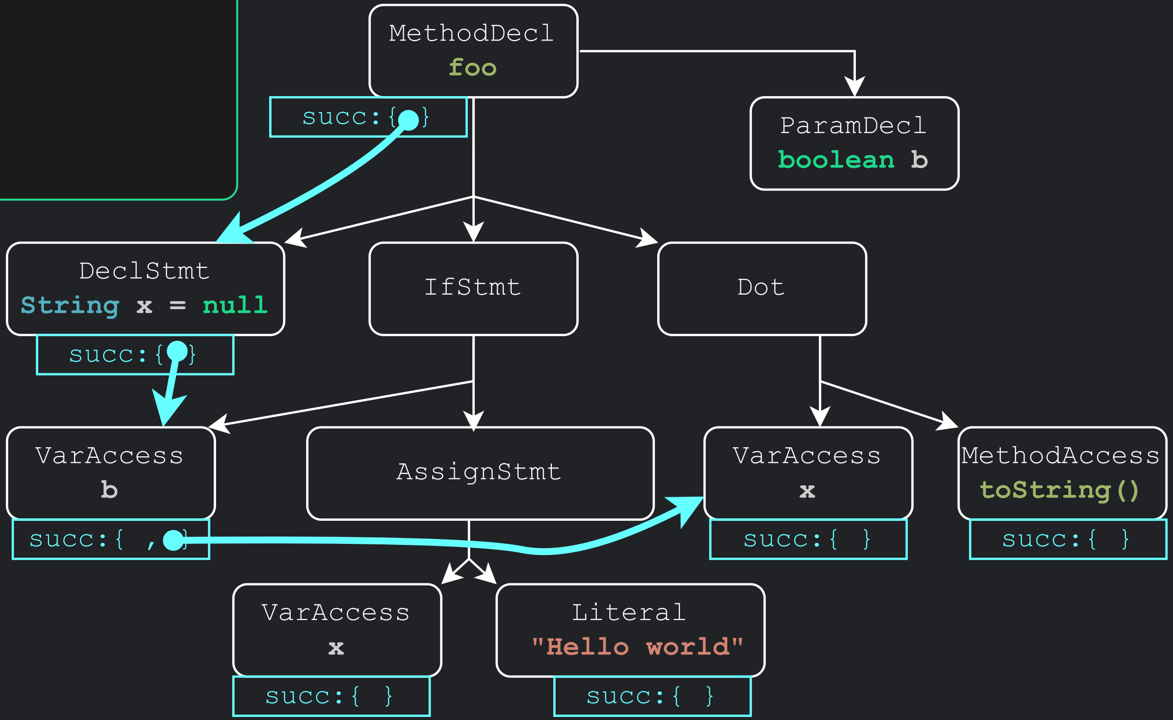
# REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



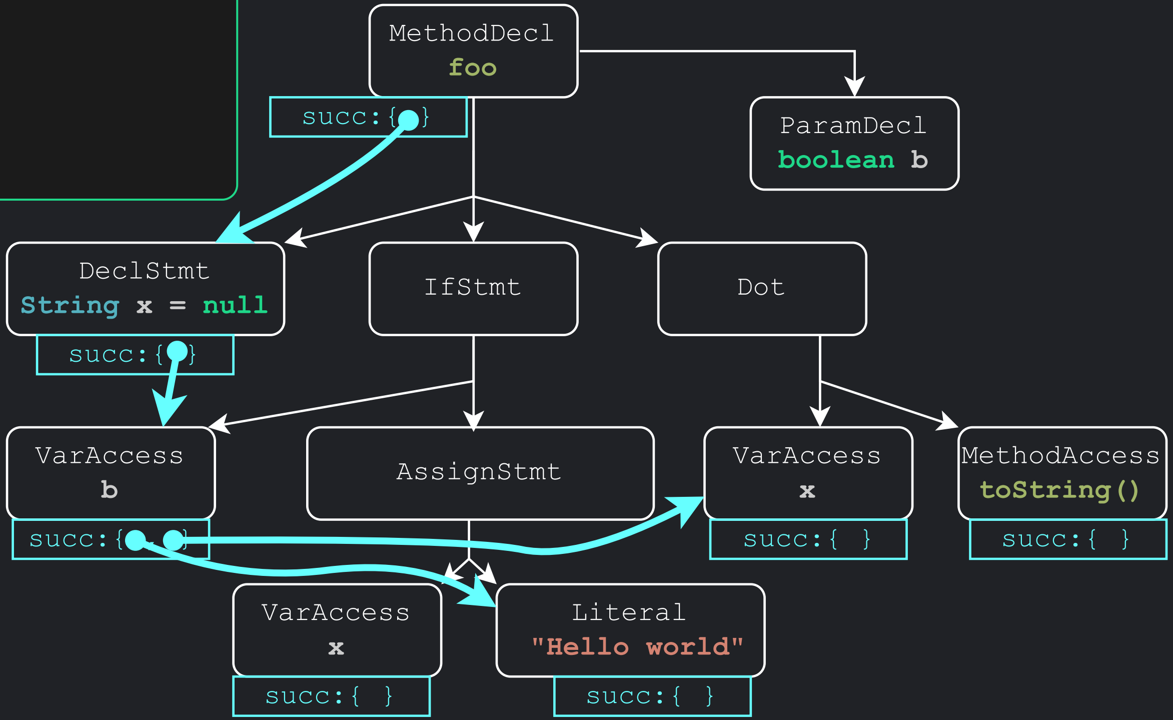
# REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



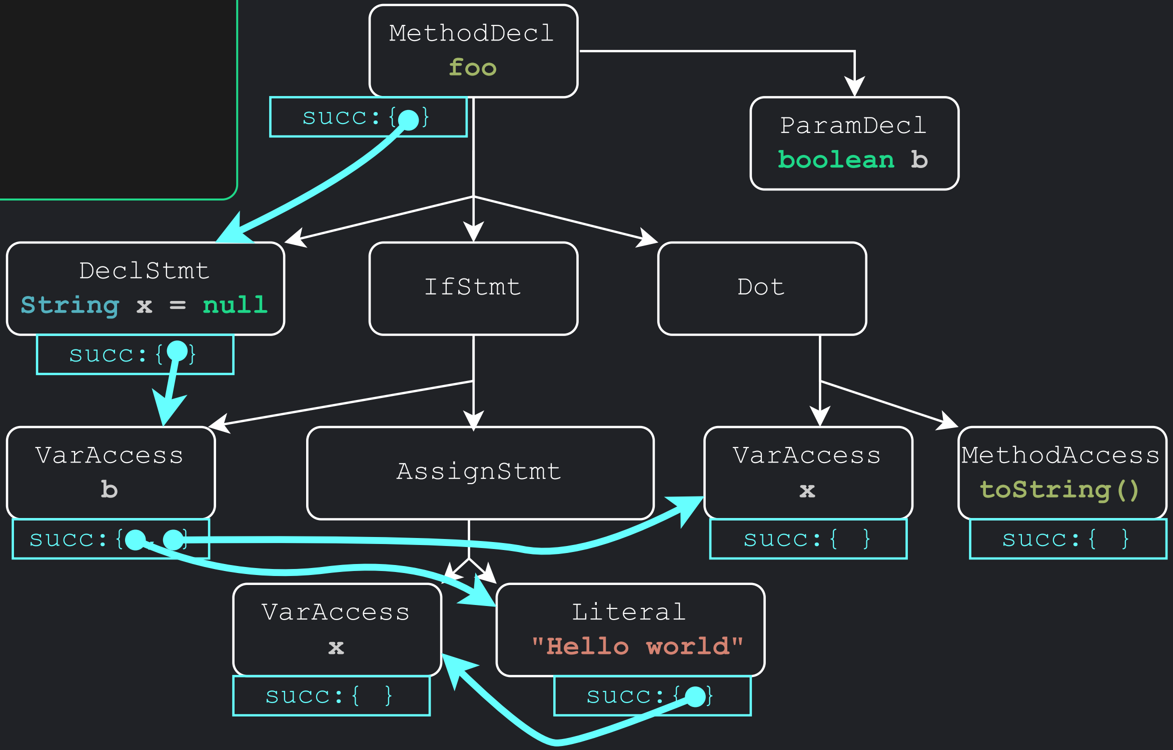
## REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



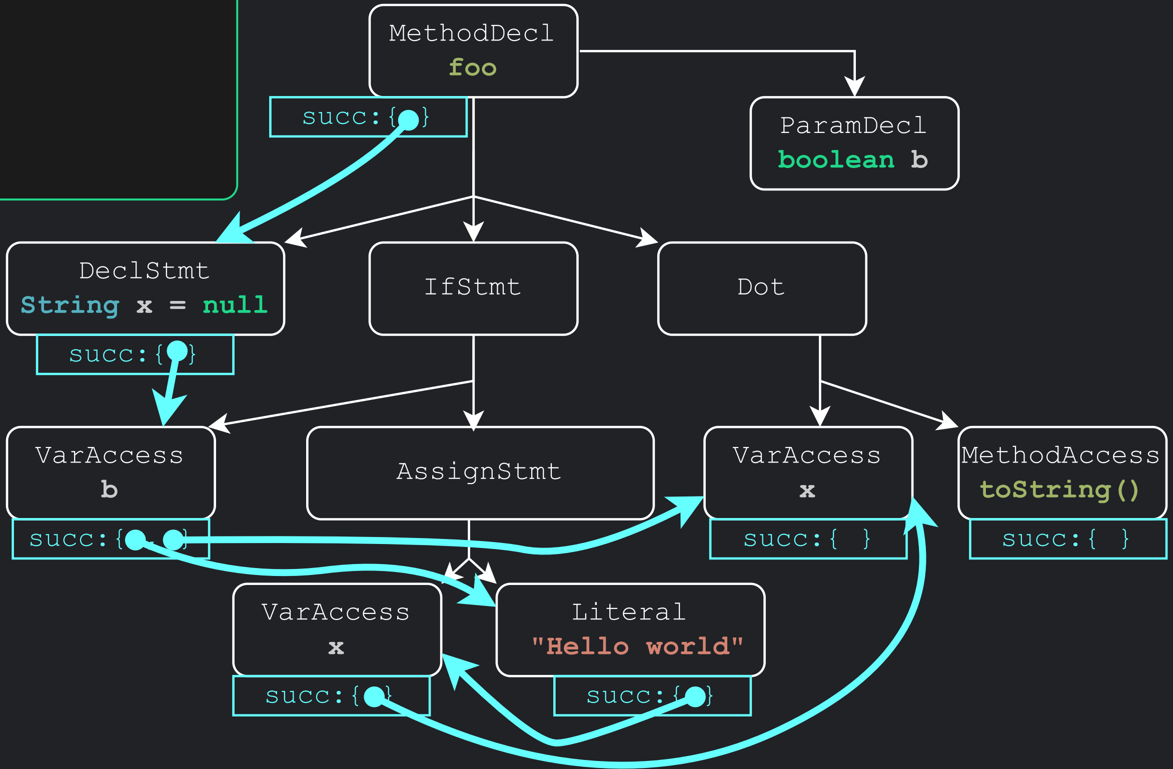
## REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



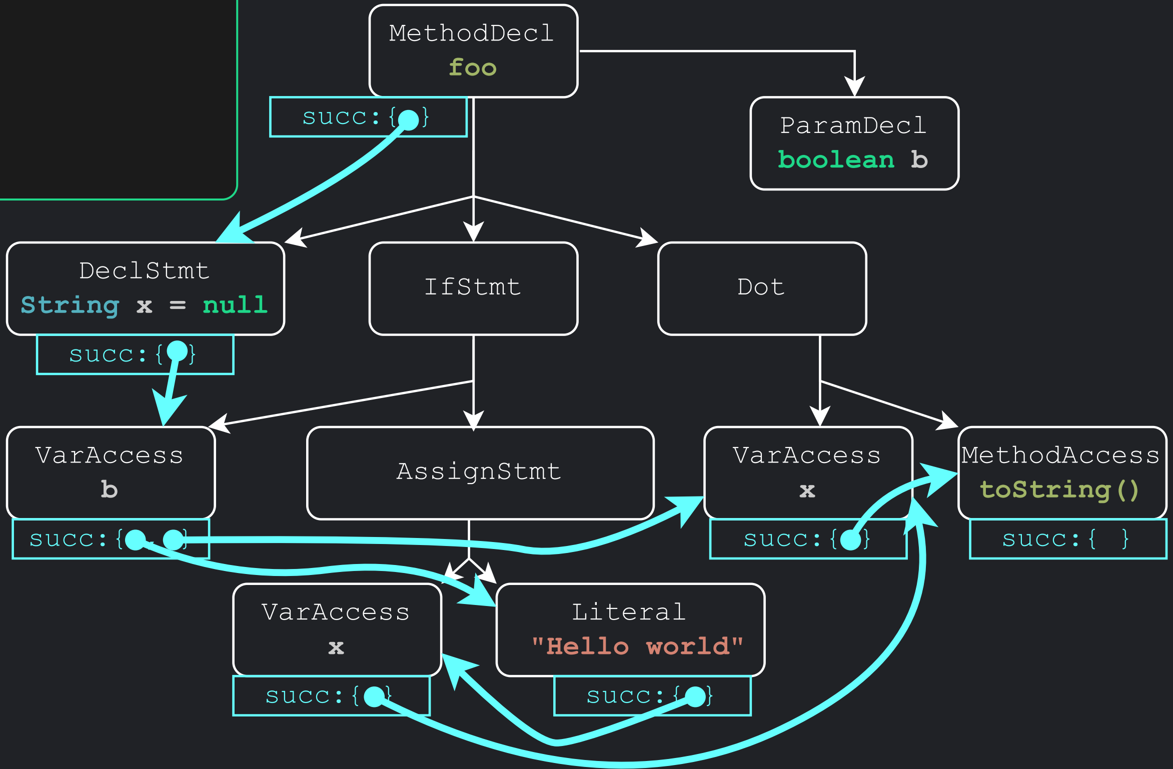
# REFERENCE ATTRIBUTE GRAMMARS

```java
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



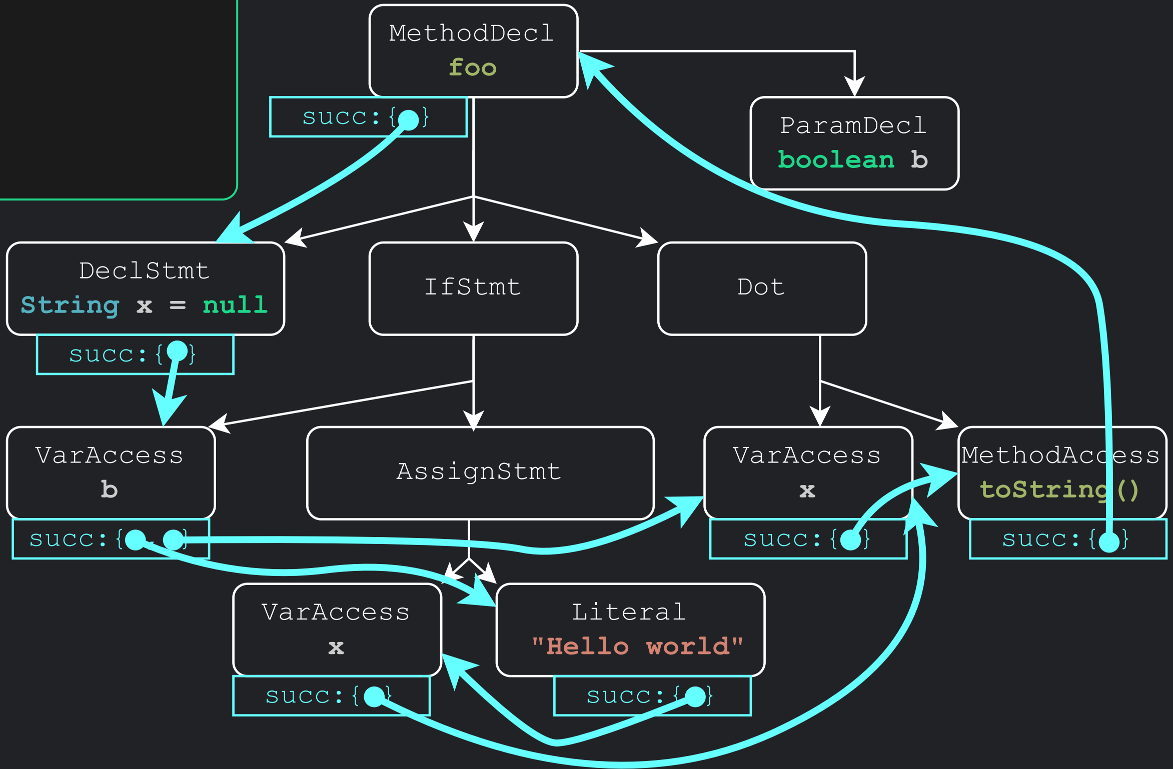
# REFERENCE ATTRIBUTE GRAMMARS

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



# REFERENCE ATTRIBUTE GRAMMARS

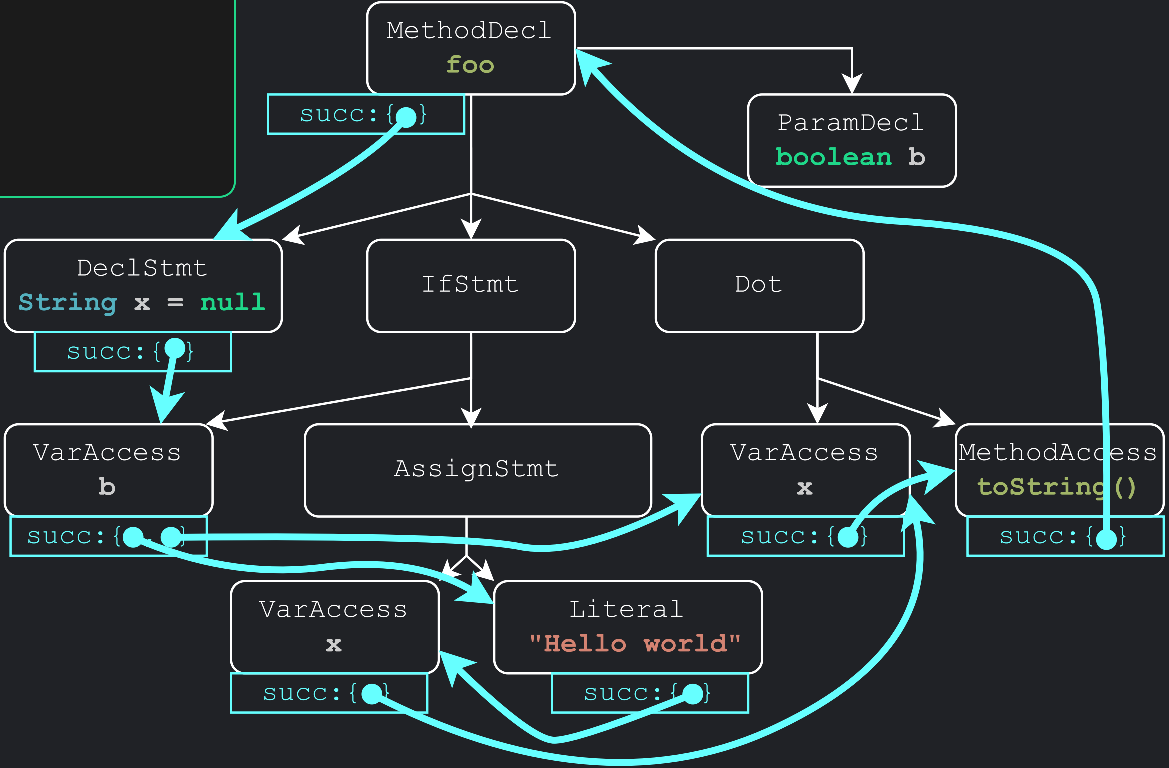
```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```



# REFERENCE ATTRIBUTE GRAMMARS

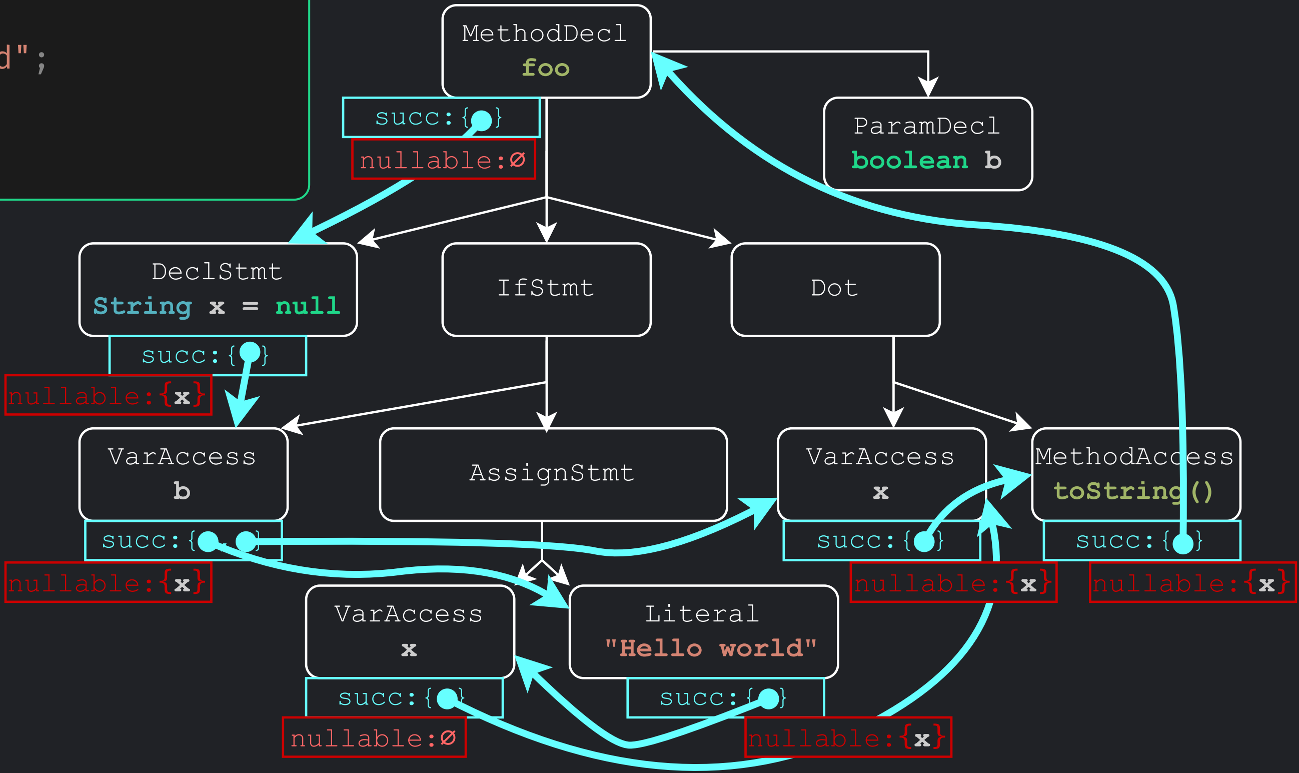
```java
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```

- JastAdd ecosystem
  - On-demand evaluation
  - Fix point computation
  - Higher-Order Attributes



# NULL POINTER ANALYSIS

```java
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```

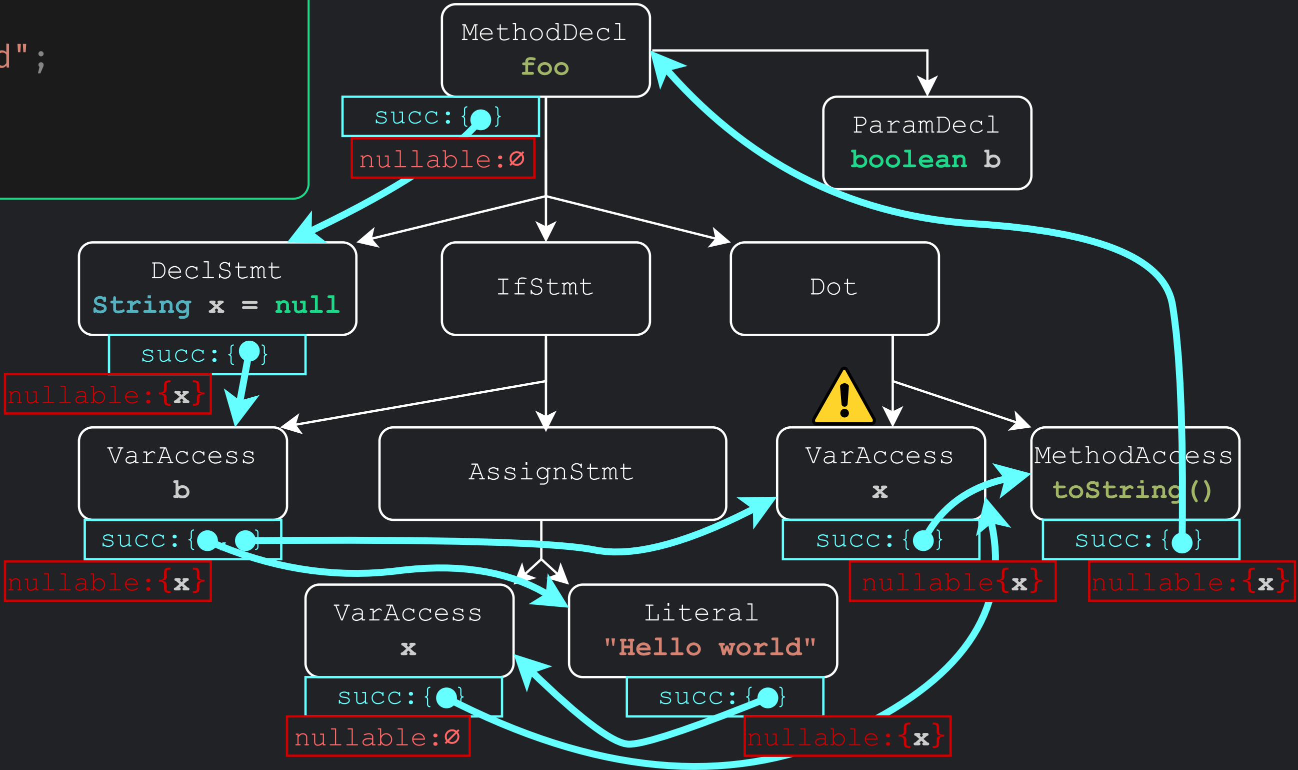


# NULL POINTER ANALYSIS

```java
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```

⚠ WARNING

A 'NullPointerException' could be thrown; 'x' is nullable here.



## INTRAJ

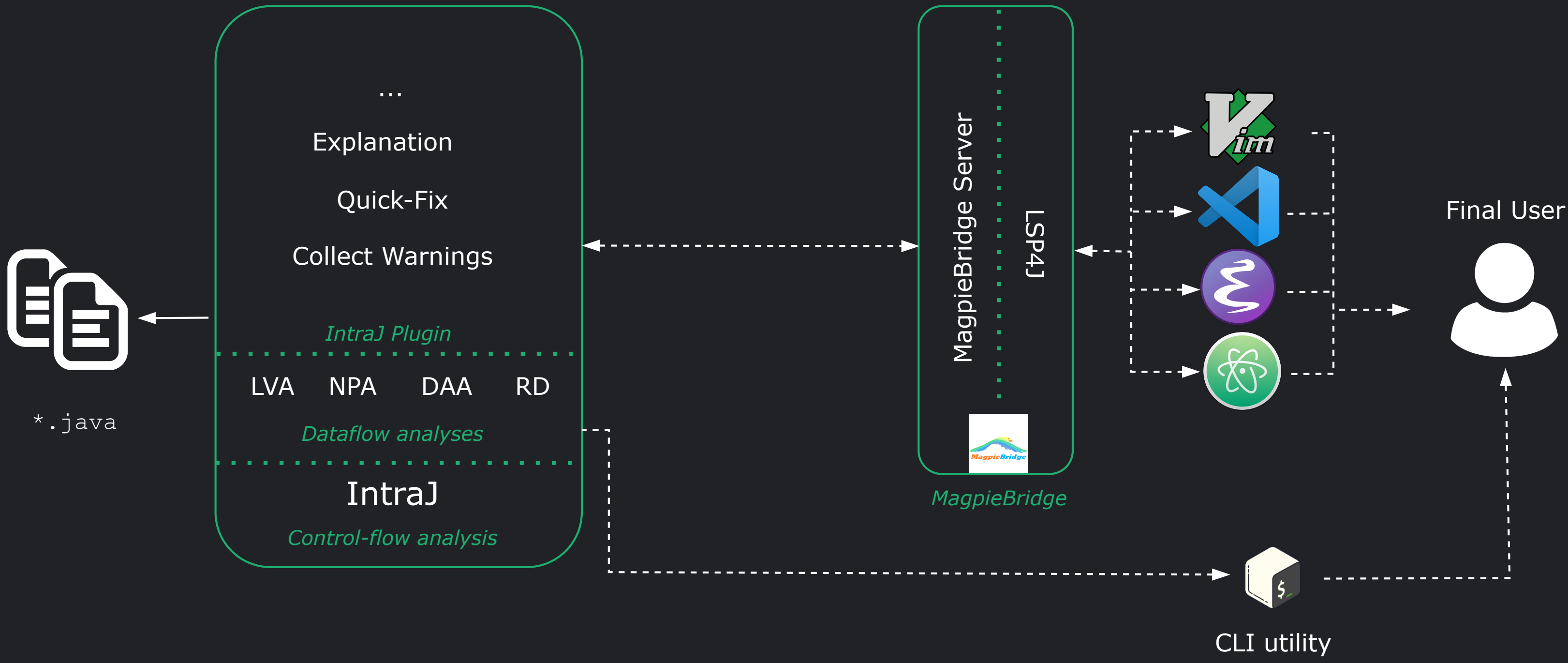
- Builds the CFGs on the AST
- Handles *implicit control-flows*
- Analyses competitive with existing tools e.g., *SonarQube*

**If you want to know more ...**

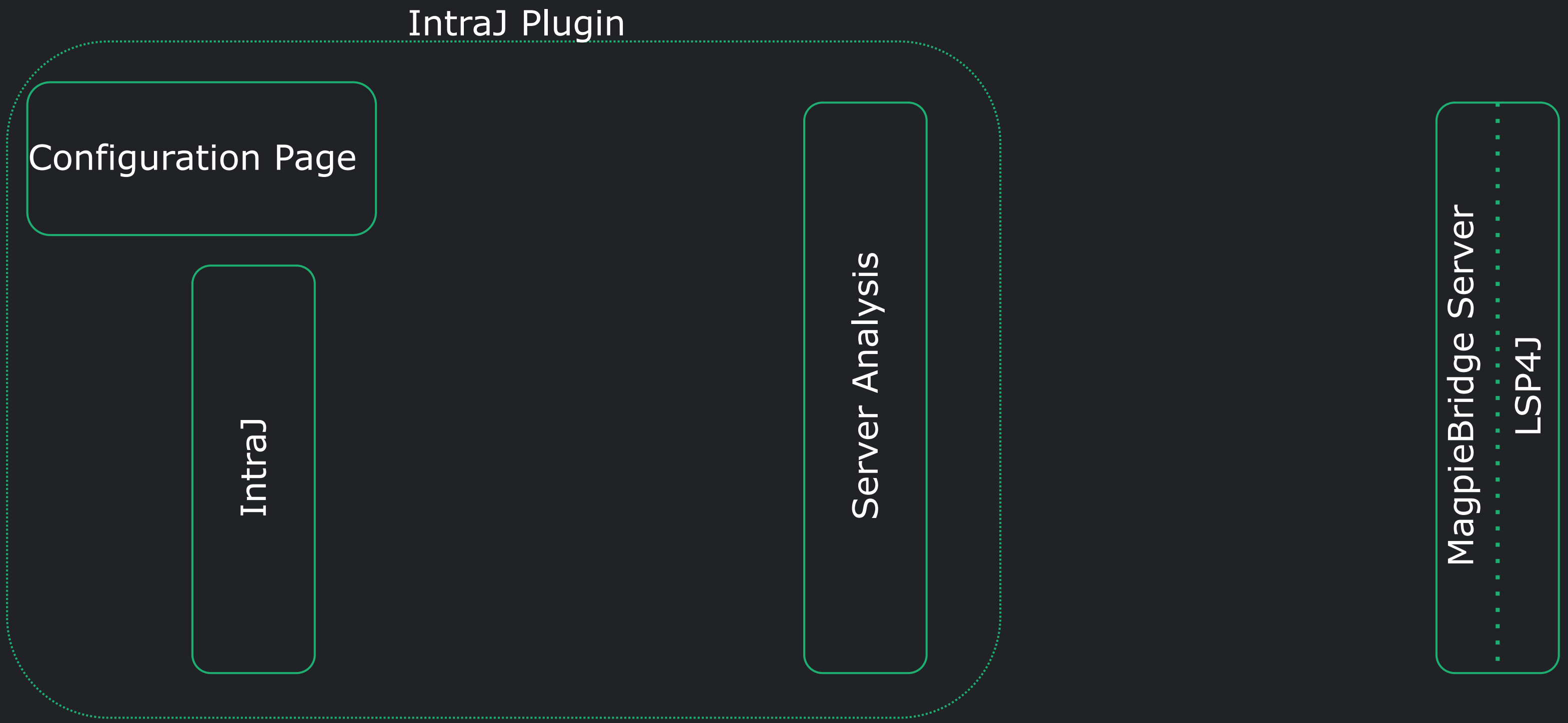
GitHub

Paper

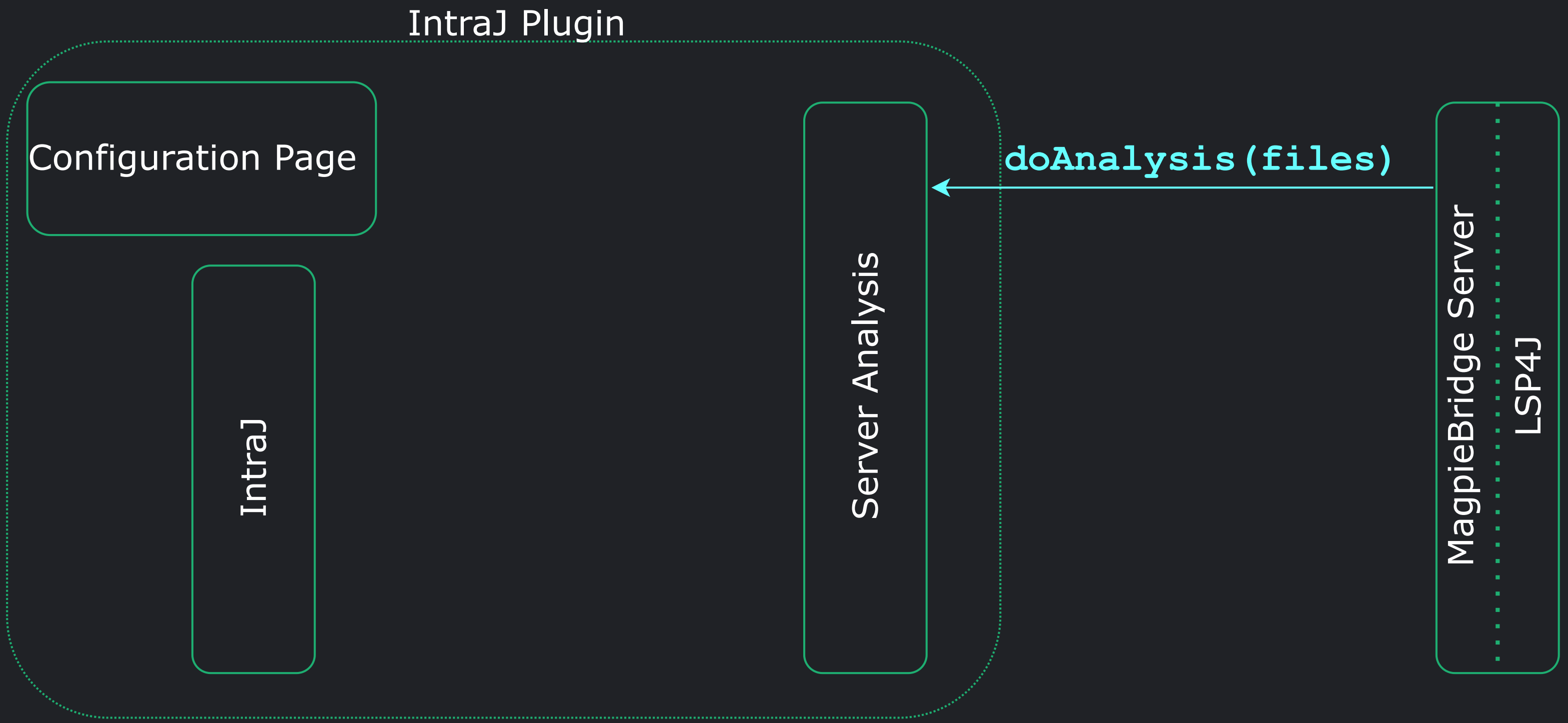
# THE BIG PICTURE, AGAIN



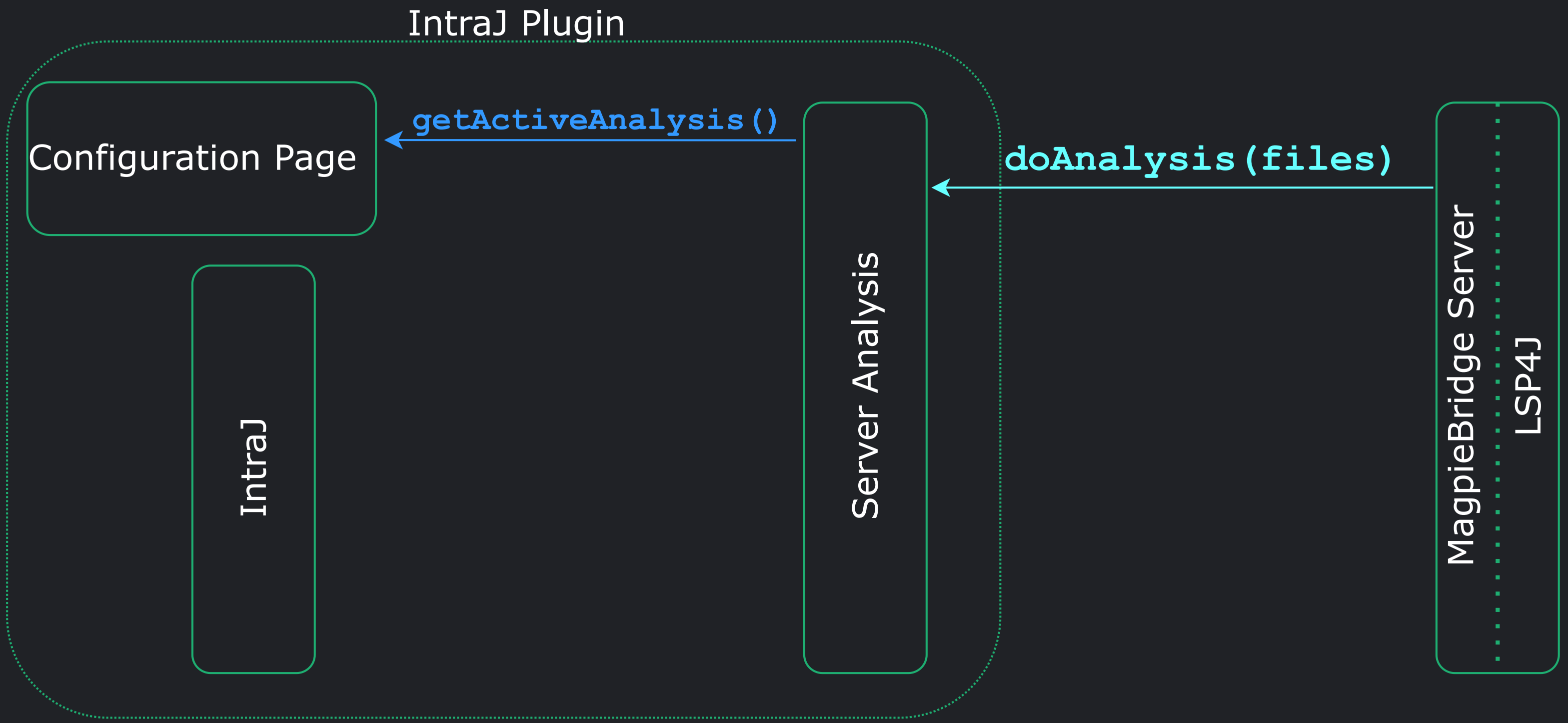
## ZOOM-IN


IntraJ Plugin
Configuration Page
IntraJ
Server Analysis
MagpieBridge Server
LSP4J

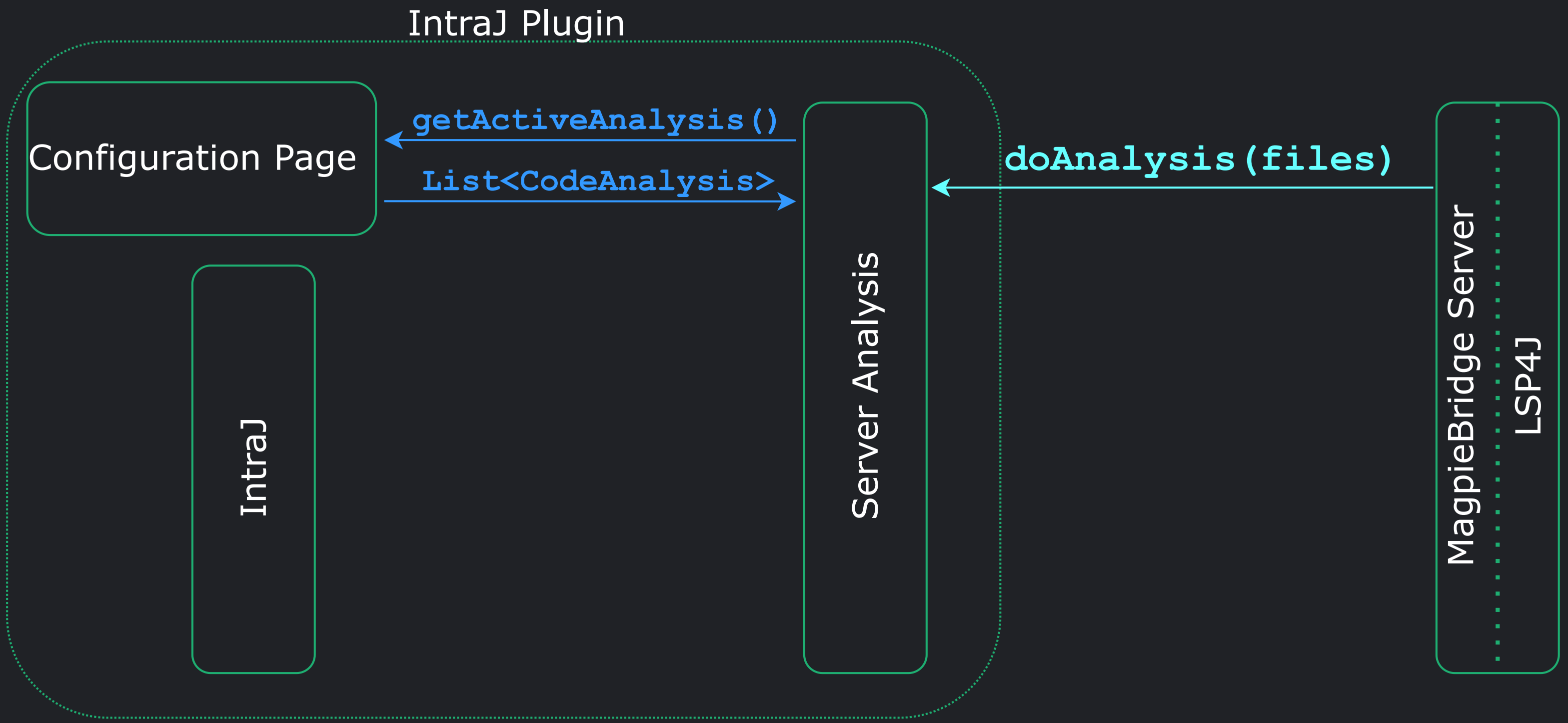
# ZOOM-IN


IntraJ Plugin
Configuration Page
IntraJ
Server Analysis
doAnalysis(files)
MagpieBridge Server
LSP4J

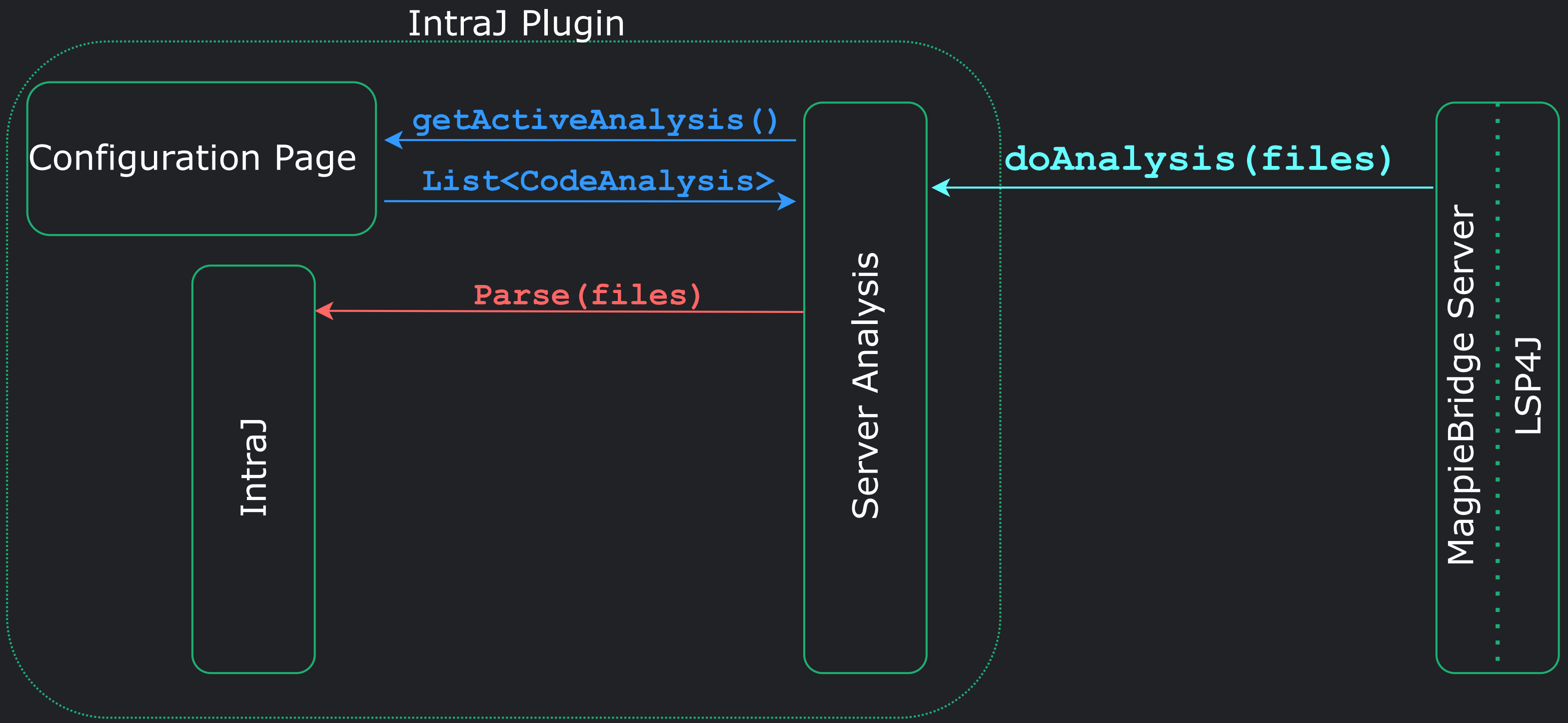
## ZOOM-IN


IntraJ Plugin
Configuration Page
getActiveAnalysis()
IntraJ
Server Analysis
doAnalysis(files)
MagpieBridge Server
LSP4J

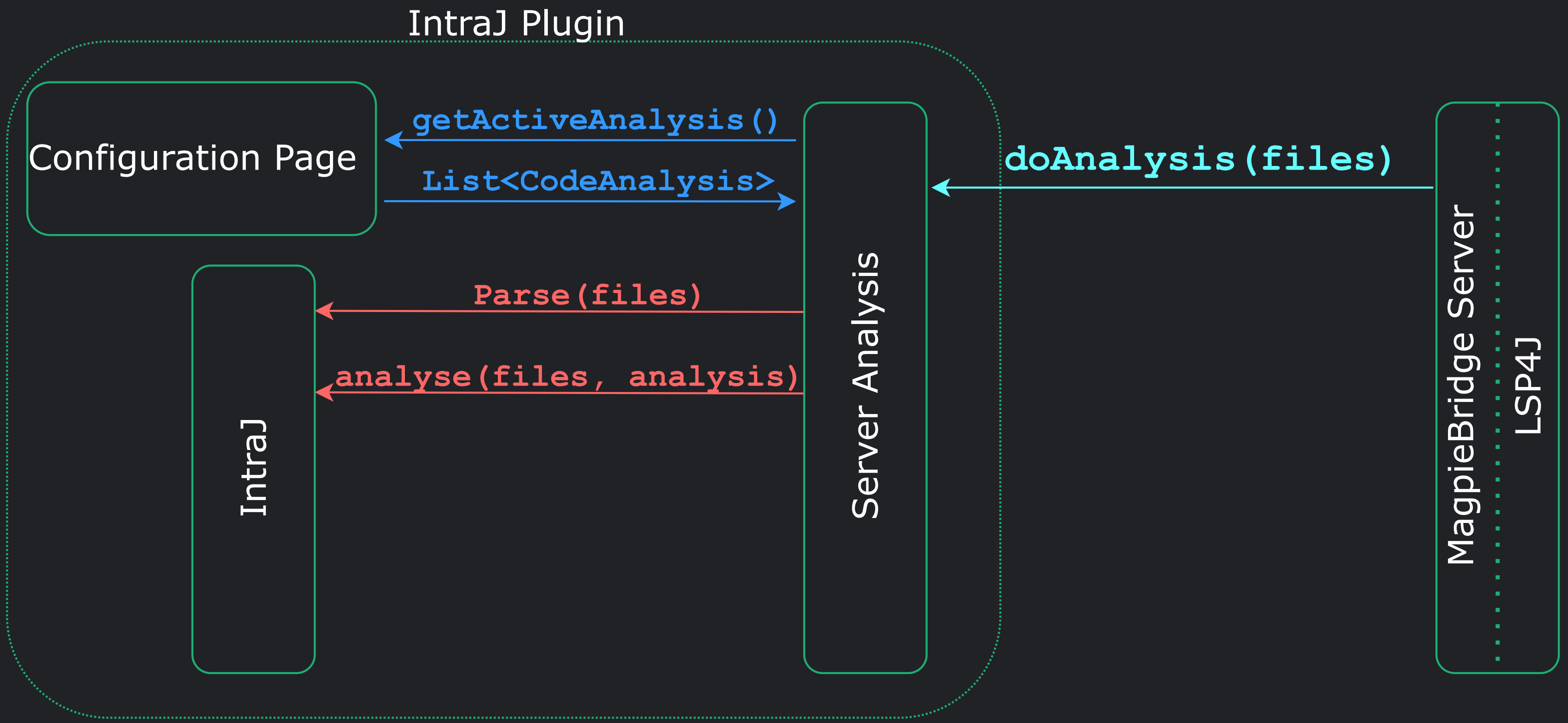
# ZOOM-IN


IntraJ Plugin
Configuration Page
getActiveAnalysis()
List<CodeAnalysis>
IntraJ
Server Analysis
doAnalysis(files)
MagpieBridge Server
LSP4J

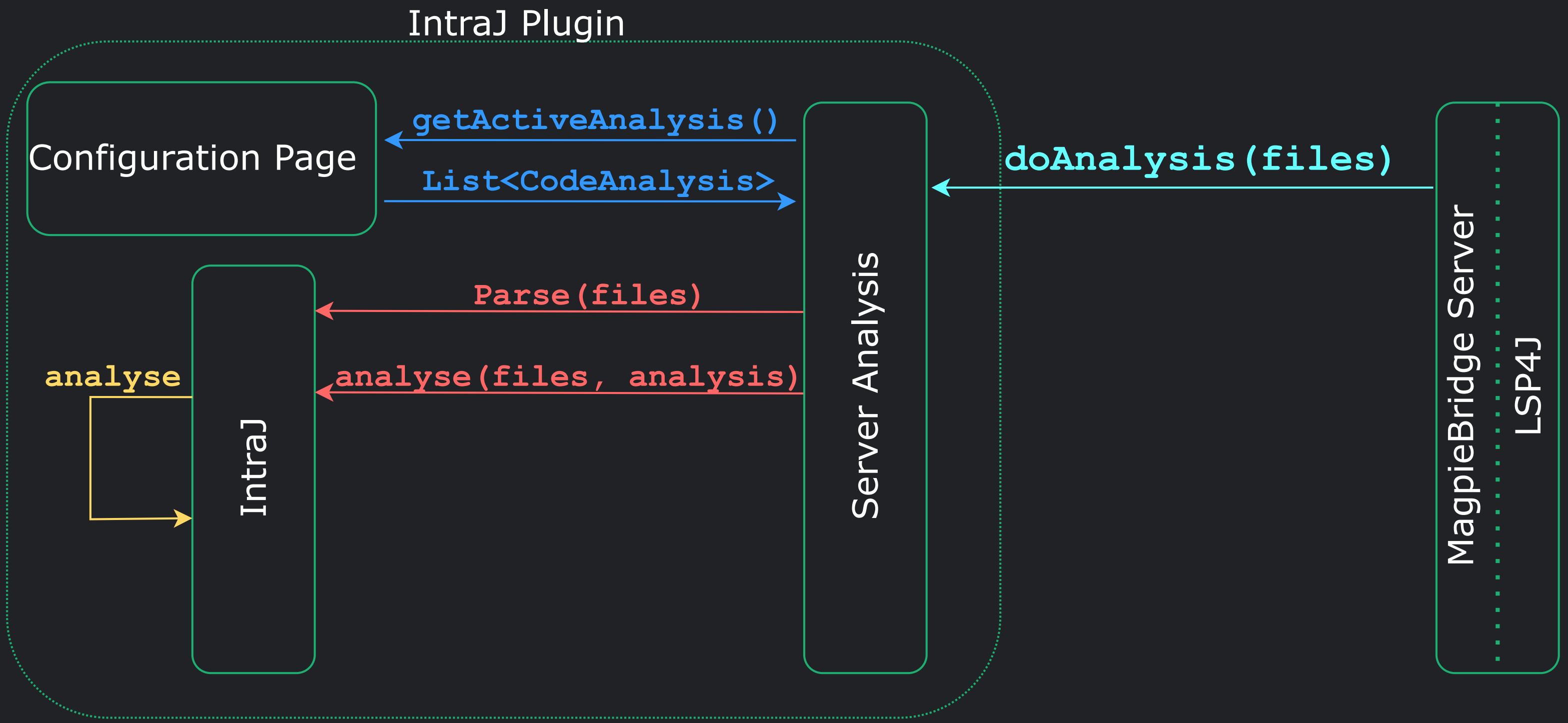
# ZOOM-IN


IntraJ Plugin
Configuration Page
getActiveAnalysis()
List<CodeAnalysis>
doAnalysis(files)
Parse(files)
IntraJ
Server Analysis
MagpieBridge Server
LSP4J

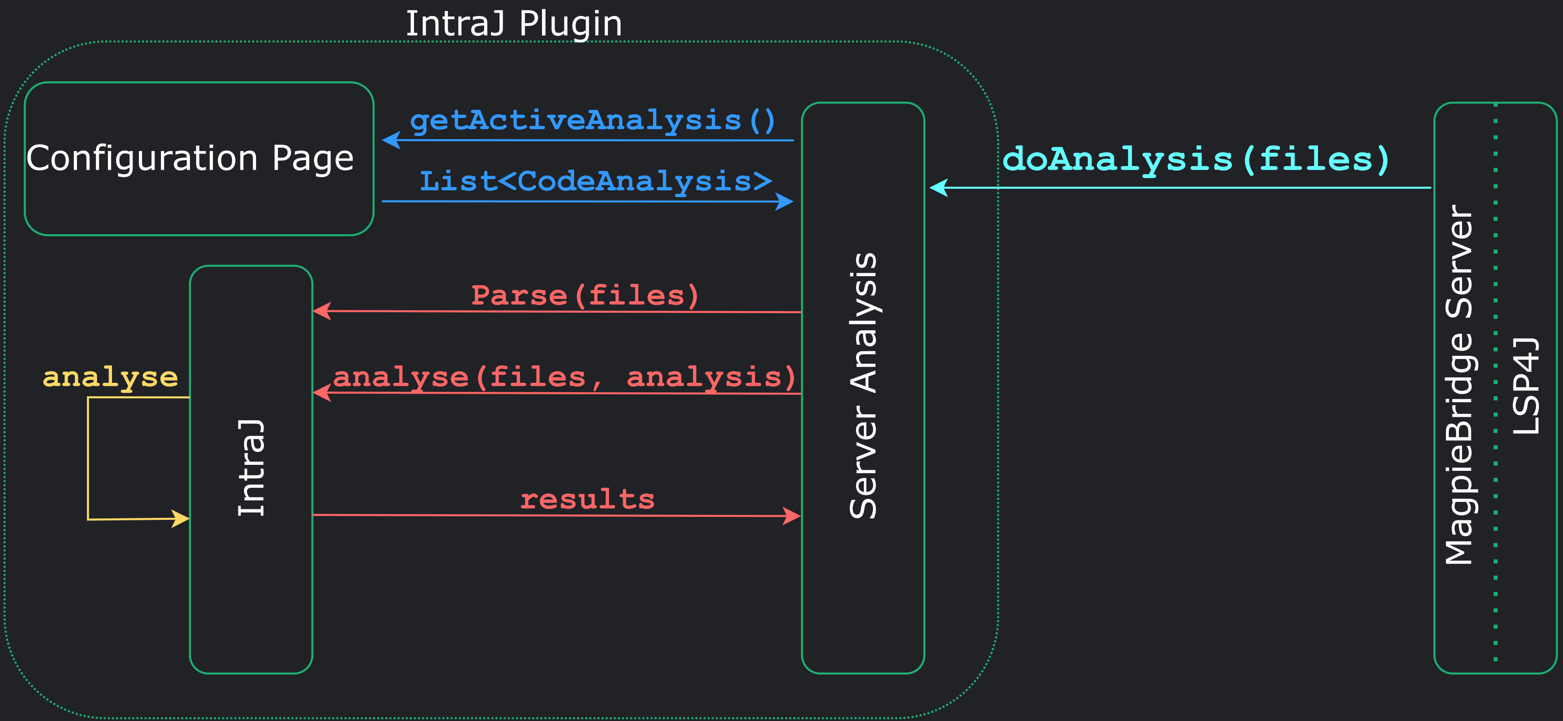
## ZOOM-IN


IntraJ Plugin
Configuration Page
getActiveAnalysis()
List<CodeAnalysis>
doAnalysis(files)
Parse(files)
analyse(files, analysis)
IntraJ
Server Analysis
MagpieBridge Server
LSP4J

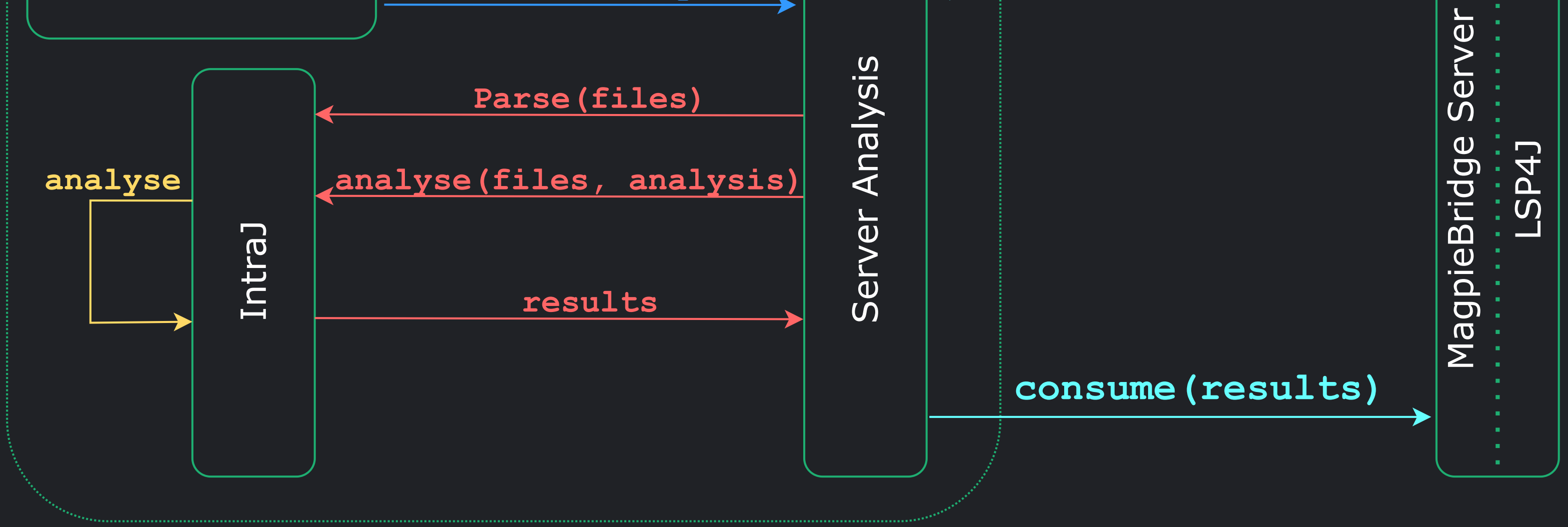
# ZOOM-IN


IntraJ Plugin
Configuration Page
getActiveAnalysis()
List<CodeAnalysis>
doAnalysis(files)
Parse(files)
analyse
analyse(files, analysis)
IntraJ
Server Analysis
MagpieBridge Server
LSP4J

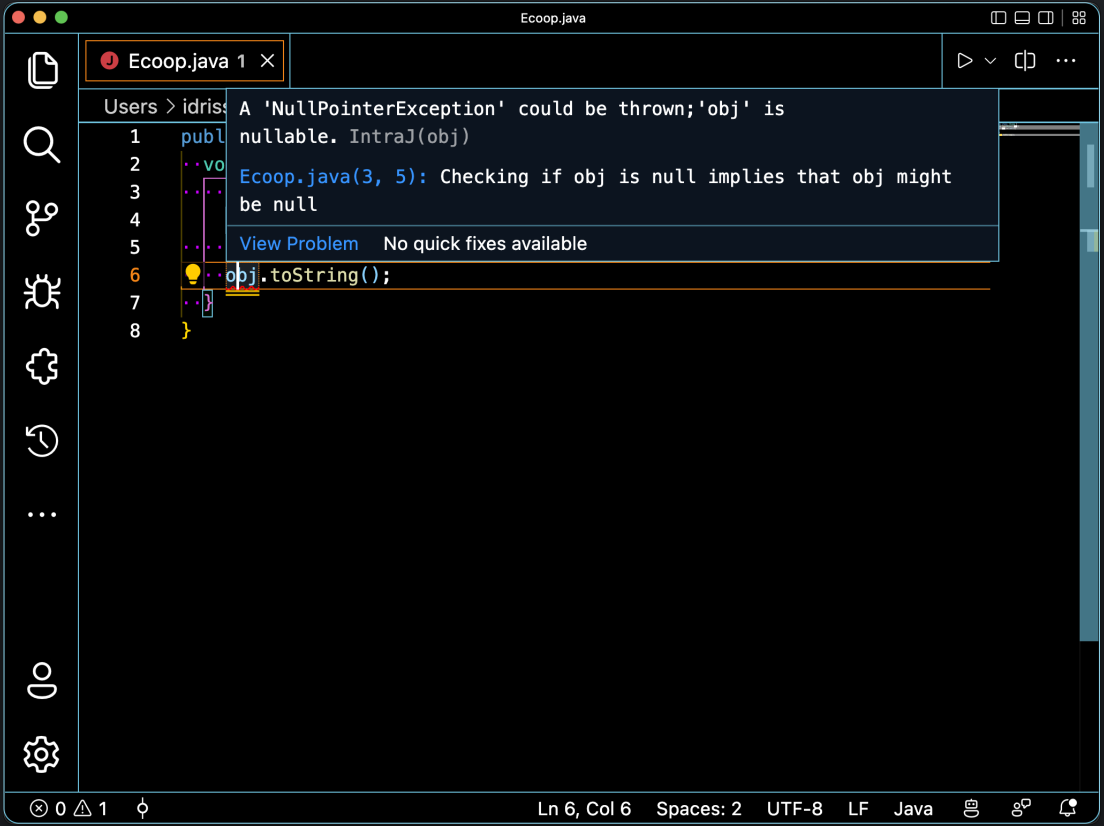
# ZOOM-IN


IntraJ Plugin
Configuration Page
getActiveAnalysis()
List<CodeAnalysis>
doAnalysis(files)
Server Analysis
MagpieBridge Server
LSP4J
Parse(files)
analyse(files, analysis)
results
analyse
IntraJ

## ZOO


Parse(files)
analyse
analyse(files, analysis)
IntraJ
results
Server Analysis
consume(results)
MagpieBridge Server
LSP4J

# EXAMPLE: QUICK FIX (WARNING)



Ecoop.java

Ecoop.java 1

Users > idris

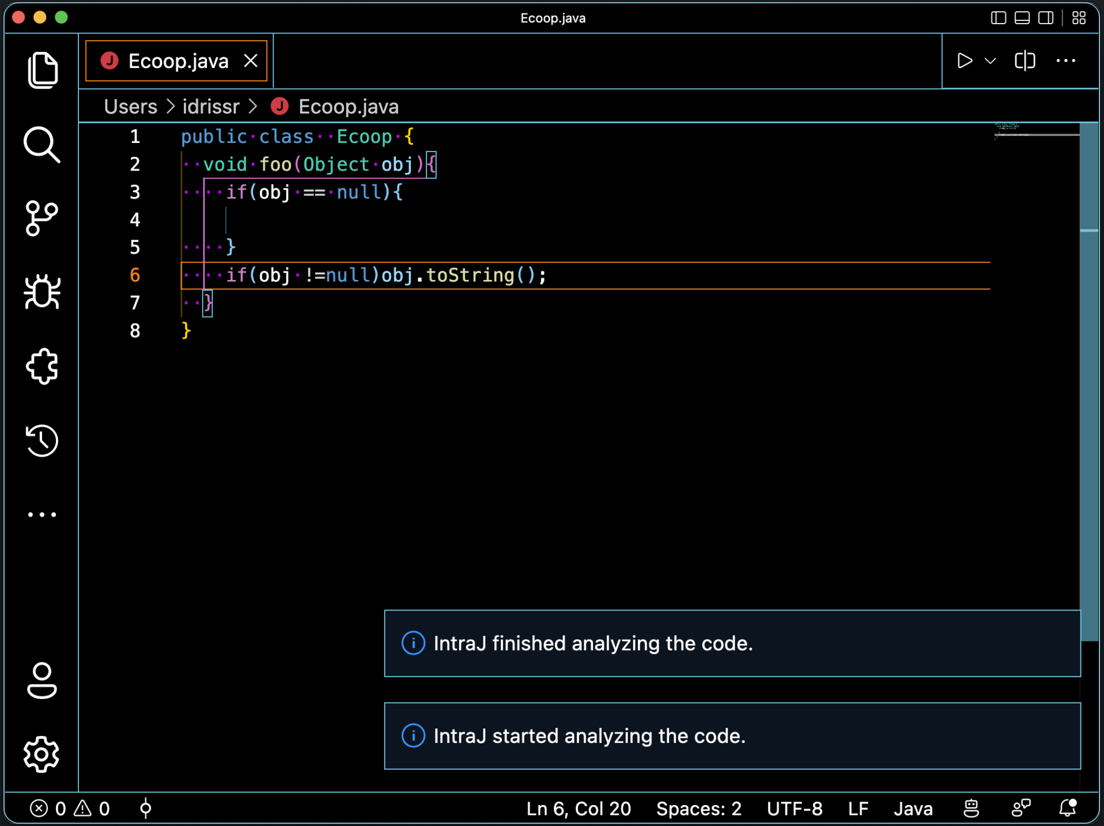
```
publ
  vo
  ...
  ...
  obj.toString();
  }
}
```

```
A 'NullPointerException' could be thrown;'obj' is
nullable. IntraJ(obj)

Ecoop.java(3, 5): Checking if obj is null implies that obj might
be null
```

View Problem    No quick fixes available

# EXAMPLE: QUICK FIX



Ecoop.java
Ecoop.java
Users > idrissr > Ecoop.java
Ln 6, Col 20   Spaces: 2   UTF-8   LF   Java
0 0

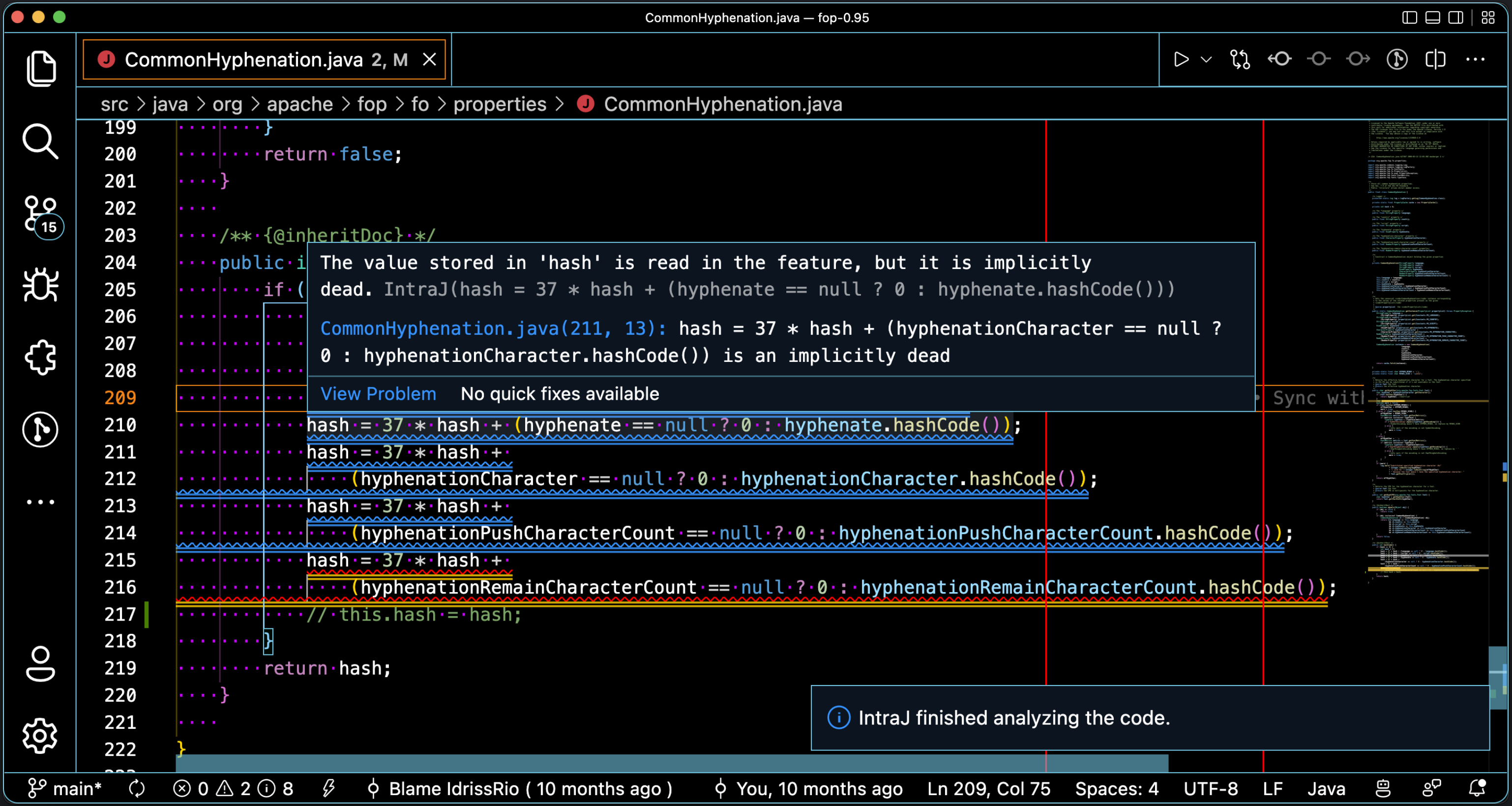
```java
public class  Ecoop {
  void foo(Object obj){
    if(obj == null){

    }
    if(obj !=null)obj.toString();
  }
}
```

IntraJ finished analyzing the code.

IntraJ started analyzing the code.

# EXAMPLE: BUG EXPLANATION



CommonHyphenation.java — fop-0.95

CommonHyphenation.java 2, M

src > java > org > apache > fop > fo > properties > CommonHyphenation.java

```
199             }
200             return false;
201         }
202     
203         /** {@inheritDoc} */
204         public i
205             if (
206     
207     
208     
209     
210                 hash = 37 * hash + (hyphenate == null ? 0 : hyphenate.hashCode());
211                 hash = 37 * hash +
212                     (hyphenationCharacter == null ? 0 : hyphenationCharacter.hashCode());
213                 hash = 37 * hash +
214                     (hyphenationPushCharacterCount == null ? 0 : hyphenationPushCharacterCount.hashCode());
215                 hash = 37 * hash +
216                     (hyphenationRemainCharacterCount == null ? 0 : hyphenationRemainCharacterCount.hashCode());
217                 // this.hash = hash;
218             }
219             return hash;
220         }
221     
222     }
```

The value stored in 'hash' is read in the feature, but it is implicitly dead. IntraJ(hash = 37 * hash + (hyphenate == null ? 0 : hyphenate.hashCode()))

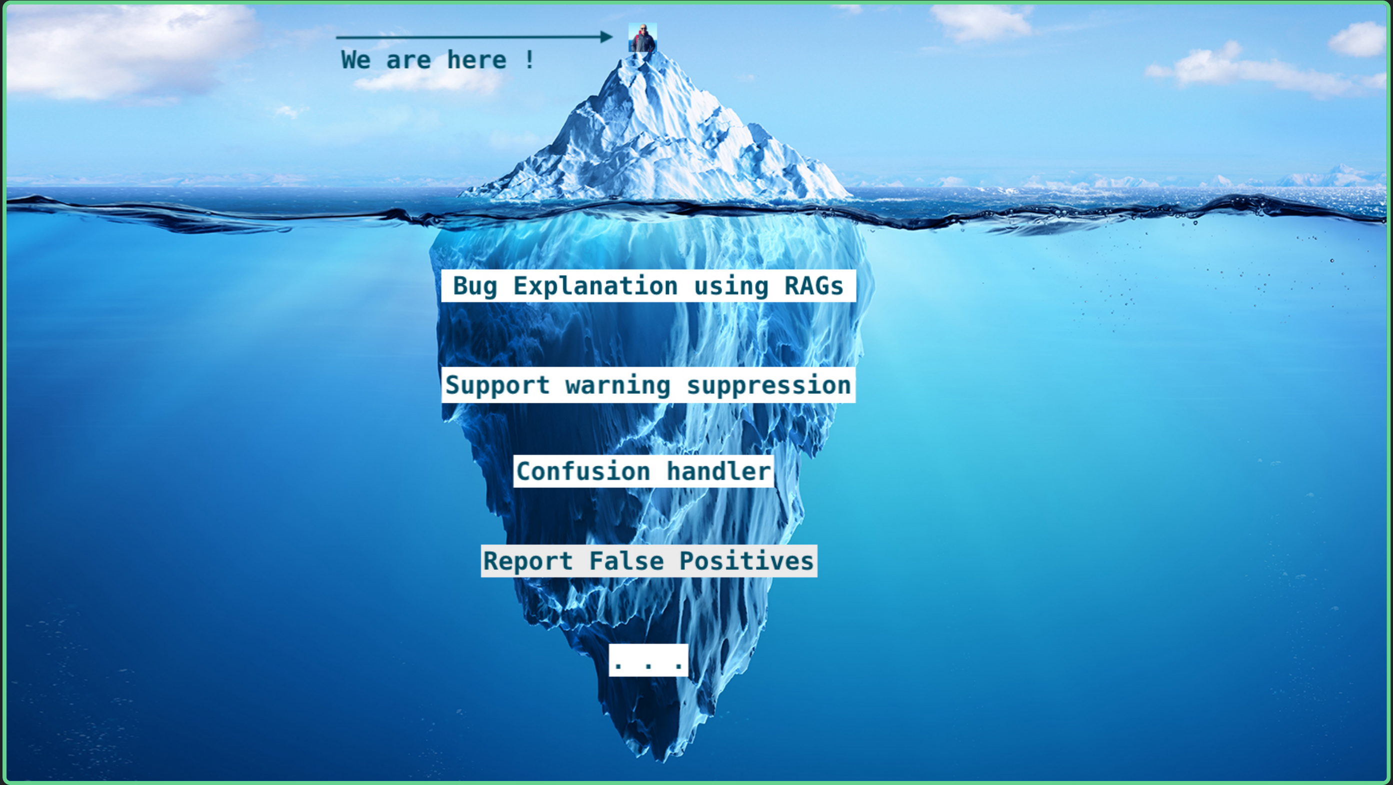
CommonHyphenation.java(211, 13): hash = 37 * hash + (hyphenationCharacter == null ? 0 : hyphenationCharacter.hashCode()) is an implicitly dead

View Problem    No quick fixes available

IntraJ finished analyzing the code.

main*   ⊗ 0 ⚠ 2 ⓘ 8   Blame IdrissRio ( 10 months ago )   You, 10 months ago   Ln 209, Col 75   Spaces: 4   UTF-8   LF   Java

## TIP OF THE ICEBERG



## OVERALL EXPERIENCE

- Intuitive and easy to use
- Concise specification of the server
- Well documented
- With the scaffolding we provide, adding support for a new analysis is trivial:

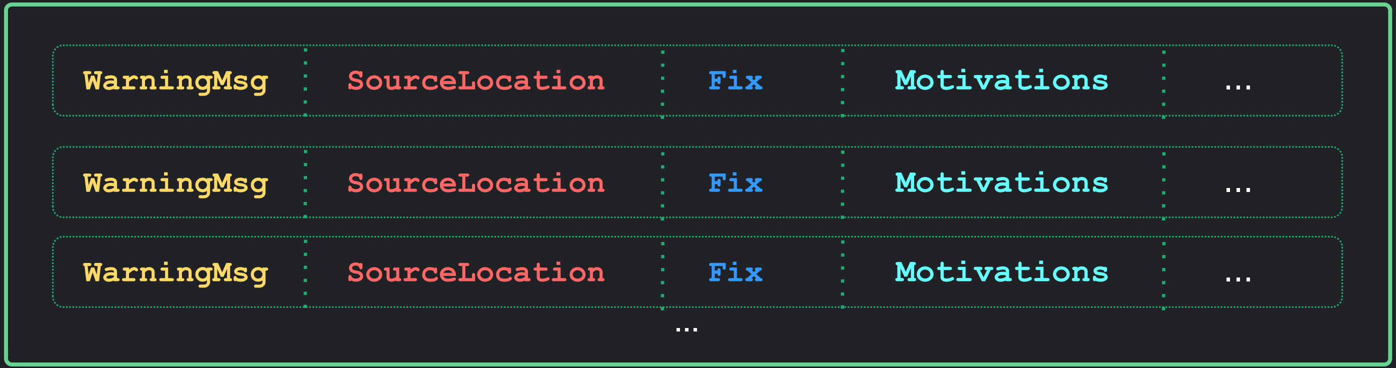
```java
public class YourAnalysis extends CodeAnalysis {
  public String getName() { return "YourAnalysis"; }
  protected Set<Warning> getWarnings(CompilationUnit cu)
    { return cu.yourAnalysis(); }  //⟵  Property triggered by the analysis
}
```

```java
activeAnalyses.put(new YourAnalysis(), true); //Register the analysis
```

- Plugin V 0.0.1 made by Charlie Mrad (Master Student @ LU)

## ON-DEMAND EVALUATION

We are able to run analyeses on-demand 👍



| | | | | |
|---|---|---|---|---|
| WarningMsg | SourceLocation | Fix | Motivations | ... |
| WarningMsg | SourceLocation | Fix | Motivations | ... |
| WarningMsg | SourceLocation | Fix | Motivations | ... |

...

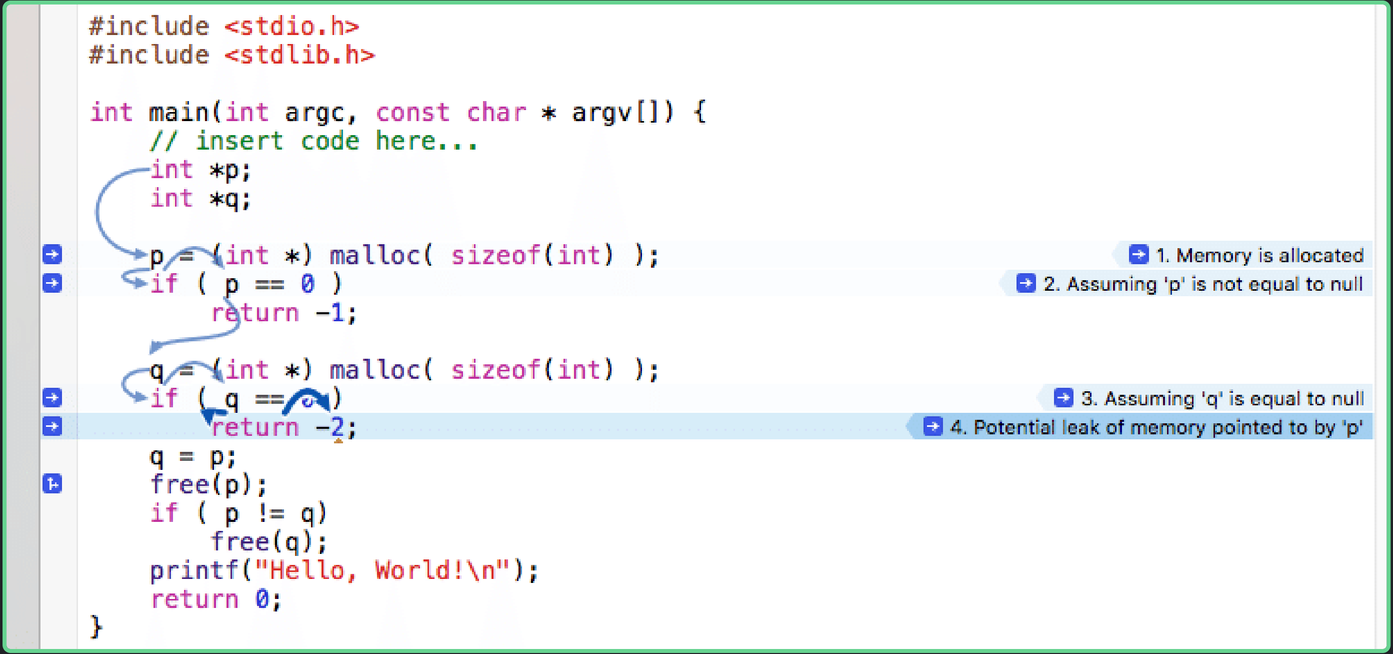
But we construct all *fixes* and *motivations* ahead-of-time because

- Hover
- CodeLens

are not exposed to **ServerAnalysis**

## LOOKING FORWARD FOR ...

### Not only warnings



```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, const char * argv[]) {
    // insert code here...
    int *p;
    int *q;

    p = (int *) malloc( sizeof(int) );        1. Memory is allocated
    if ( p == 0 )                             2. Assuming 'p' is not equal to null
        return -1;

    q = (int *) malloc( sizeof(int) );
    if ( q == 0 )                             3. Assuming 'q' is equal to null
        return -2;                            4. Potential leak of memory pointed to by 'p'
    q = p;
    free(p);
    if ( p != q)
        free(q);
    printf("Hello, World!\n");
    return 0;
}
```

## THANK YOU FOR YOUR ATTENTION !

GitHub

Paper

Extension

## MOTIVATIONS: SOURCE-LEVEL

```
void foo(boolean b){
  String x = null;
  if(b) x = "Hello World";
  x.toString();
}
```

### Advantages

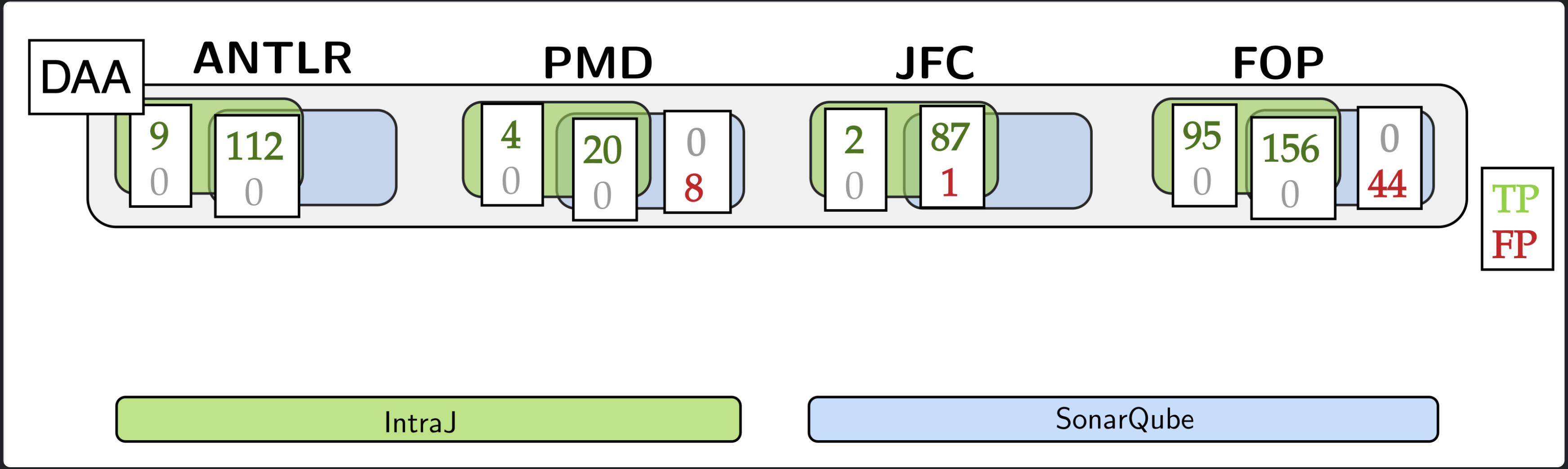
1. Error are directly linked to the source code
2. Works with broken code
3. Easier integration with IDEs

### Disadvantages

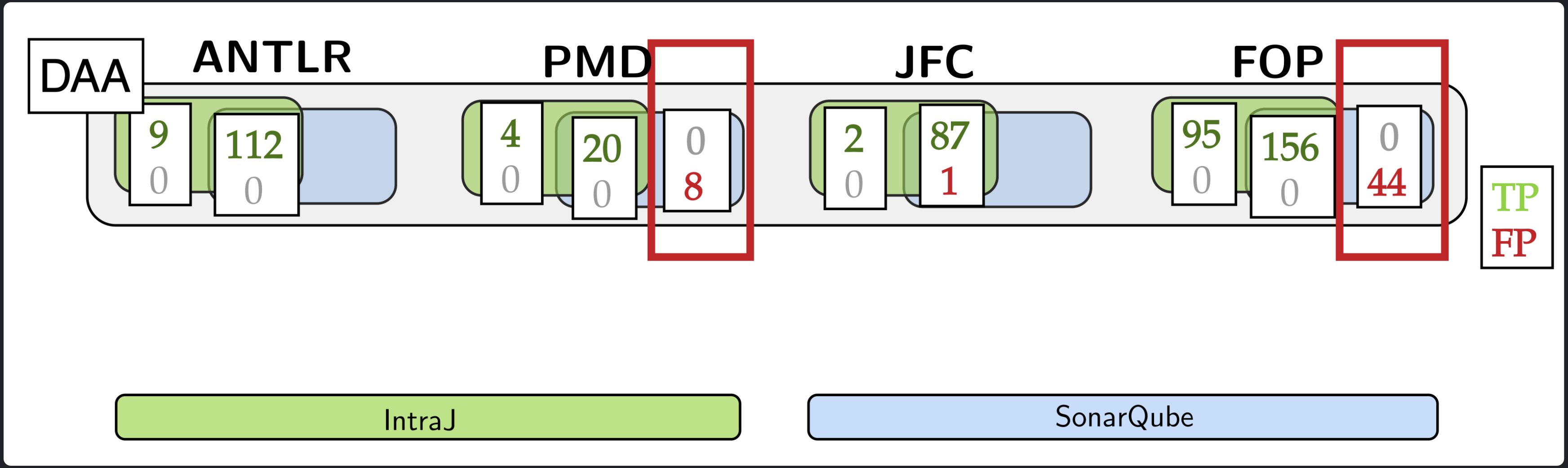
1. Bigger language
2. Source-code contains implicit facts

```
    void foo(java.lang.boolean);
      Code:
         0: aconst_null
         1: astore_2
         2: aload_1
         3: invokevirtual #2
         6: ifeq          12
         9: ldc           #3
        11: astore_2
        12: aload_2
        13: invokevirtual #4
        16: pop
        17: return
```

# PRECISION: NUMBERS

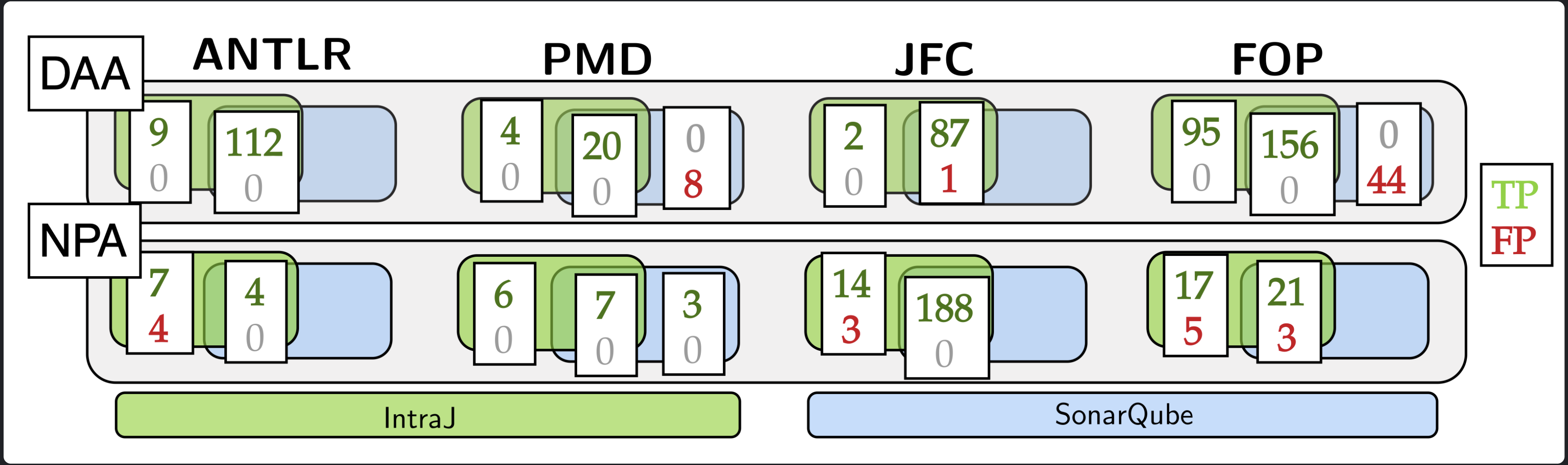


## PRECISION: NUMBERS



DeadAssignmentAnalysis: IntraJ detects everything that SonarQube detects

# PRECISION: NUMBERS

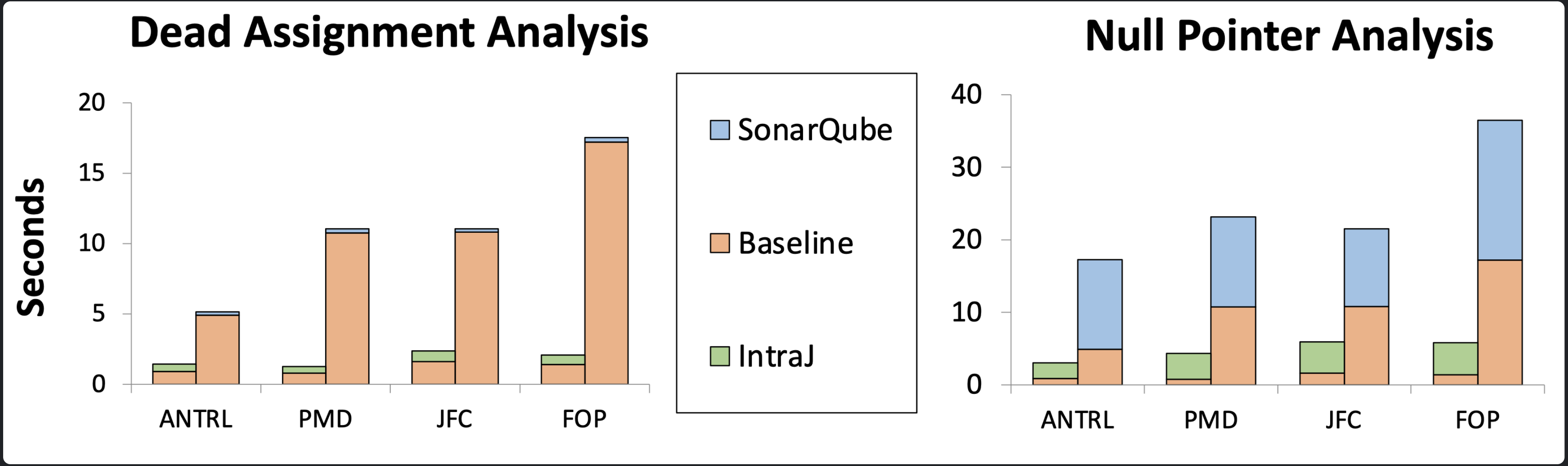


DeadAssignmentAnalysis: IntraJ detects everything that SonarQube detects

NullPointerAnalysis: SonarQube is more precise but IntraJ remains competitive

# PERFORMACE

1. No dealy in the previous demo


Dead Assignment Analysis
Null Pointer Analysis
Seconds
ANTRL
PMD
JFC
FOP
SonarQube
Baseline
IntraJ